# CHAPTER ONE
# INTRODUCTION

## 1.1 Background of the Study

The 21st century business environment is characterized by increasing competition slimmer profit margins, more demanding customers and more stringent regulations. (Omiyi 2008)
He stated further that companies, especially multi-national enterprises, face greater pressures to pay attention to ethical, environmental and social issues. In addition, the Boston Consulting Group's Innovation 2005 Survey identified globalization and organizational issues as the biggest challenges facing many organizations today.

These issues also affect the oil and gas industry. In the last few years, the oil and gas industry has witnessed mergers involving BP and Amaco; Exxon and Mobil, Fina and Elf as well as Chevron and Texaco. These merges have significantly changed the landscape in the industry and have made the industry a lot more competitive globally. In Nigeria, the Shell Petroleum Development Company (SPDC) has completed a major re-organization tagged Securing Our Future (SoFU). Similarly, the NNPC has undergone a major re-structuring called Positioning and Aligning for Competitive Advantage (PACE) (Omiyi 2008).
On-line monitoring of manufacturing process is extremely important in modern manufacturing for plant safety, maximization of the production and consistency of the product quality Song et al, (2003). The development of diagnostic systems for the industrial applications started in early 1970s. The recent developments in microelectronics have increased their intelligence and let them found many industrial applications in the last two decades Mendonca et al (2009), Shi & Sepehri, (2004). Intelligent data analysis techniques are one of the most important components of the fault diagnosis methods Uppal et al (2002); Uppal & Patton (2002). This work proposes a scheme for monitoring the process involved in the production of gas/oil using the artificial neural networks (ANN).

Omiyi stated that the oil and gas industry is a multi-faceted industry that encompasses a wide range of activities. Key activities undertaken by the industry include:

- Exploration for and production of petroleum and allied products (mainly oil and gas)

- Refining and processing crude oil and gas
- Transporting, marketing and distributing petroleum products
- Power generation (including the utilization of alternative energy sources)
- Research

In Nigeria, key operators in the Petroleum Industry include Shell, ExxonMobil Chevron, Total, Agip, NDPC, etc. In the upstream (Exploration and Production) sector are Conoil, Oando, African Petroleum and a host of others in the downstream (Marketing and Distribution) sector. (Omiyi, 2008). Of course, the NNPC also plays a role in regulation, transportation and refining.

By its very nature, the petroleum industry requires innovation from any operator that wishes to be successful. The world will need much more energy in the years ahead and most of it will come from fossil fuels (mainly oil and gas). Demand is likely to increase more over the first three decades of this century than it did over the previous 30 years according to Omiyi.
Fossil fuels are a depleting resource; there is therefore the need to constantly replace produced reserves.

The industry is highly capital intensive, involving the use of advanced and complex technologies. In addition, investors often have to wait for a long time before their investments yield any returns. This long gestation period is exemplified by the fact that it took Shell 21 years, from commencement of exploration activities in Nigeria in 1937, till 1958 when commercial production and export commenced.
The upstream sector has the responsibility of finding and producing a resource that is located thousands of meters underground (or under the sea, as the case may be). There is no direct way of confirming the presence of hydrocarbon and measuring the volumes in any reservoir. There is no option but to rely on various reservoir modeling and estimation techniques in order to confirm the presence of a reservoir and to provide reasonable certainty that the volumes of oil or gas present can be profitably produced.

In addition, exploration and production take place in different environments such as arid land location in the deserts of the Middle East, the swampy environment of the Niger Delta, and offshore in the Gulf of Mexico, the North Sea and of course off Nigeria's Atlantic Coast. It is the responsibility of the oil companies to ensure minimal impact of their activities on the environment in all these different eco-systems. The implication of all this is that the upstream sector has to deal with a high level of risk, uncertainty and responsibility. Therefore the firm faces a challenge of developing and deploying innovative means of finding, quantifying, developing and producing oil and gas, in a profitable, yet safe and environmentally responsible manner. **S**imilarly, in the downstream sector, refining margins are extremely thin and competition very high. The most successful operators are the ones that are able, through innovation, to increase their competitive advantage. This could be in the development of more efficient fuels and lubricants, increasing the efficiency of the refining process etc.

The tools and techniques required to meet the challenge of the Oil and Gas Industry, cannot be bought off the shelf, plugged in and switched on. Rather they depend on advanced skills and continuing, long-term investment to choose, apply and integrate the best technologies to suit particular conditions. The industry's remarkable advances into deep water have involved many new and improved technologies.

Through this process an organization develops and deploys a wide range of innovative solutions that could help sustain its leadership position in the industry. Using Shell Oil Company Limited as a specific example of the use of this process in Nigeria, one of the challenges it faces operating in the Niger Delta is the need to monitor its oil production facilities for early detection of oil spill incidents. This is critical due to incidents of crude oil and sabotage which often lead to major oil spills. It has deployed a range of monitoring techniques but has encountered problems, especially in the swamp areas.

A team was set up to come up with a solution to the challenge. It could not use conventional systems, involving high cost technology and solar panels, because they were regularly stolen or sabotaged. The system had to be low cost, able to operate on power (from batteries), transmit over a range of up to 8 kilometers and be maintenance free. The team, in partnership with a Shell

Technology venture developed micro wireless units with the capacity to monitor pressure, temperature and flow in its pipelines. After a year of research and development, a low power, battery operated sensor was developed. The units are small enough to fit into a 10cm diameter conclave and do not attract the attention of thieves and saboteurs. This innovation was tested early in 2005 and proved successful, with the ability to monitor remote manifolds up to 8.6 km away from the flow station. SPDC is confident of efficient operation of its facilities, better safety and environmental performance and the ability to deploy smart well technology.

Today industries are the main power of the country. It increases the economic growth of the county and provides many benefits. Remote monitoring and intelligent maintenance is one of the most important criteria for maximizing production and process plant availability. Wireless media has been undergoing a rapid innovation process in search for a reliable, simple and business viable technology for fast, easy and inexpensive diagnosis of faults in process plants.

The use of monitoring and control technology in the oil/gas industry has experienced a steady evolution that has paralleled the advances in supervisory control and data acquisition (SCADA) and related technologies during the last 30 plus years. Monitoring and control of remote facilities initially meant a crew (or crews) spending countless hours driving from one remote facility to the next, often on a full-time basis. When they saw a problem at a facility, they fixed it, which often meant making an adjustment to a piece of equipment such as a pump or a valve. The first leap forward was equipping these crews with radios to call in to headquarters when a problem was located, or to call for additional crews when needed. The first steps in automating this process were simple monitoring and alarm systems. These were typically electro- mechanical devices at remote sites that would send a signal back to a control center via radio or leased telephone lines. While these systems were a step forward, the high cost of the computing technology often made them a tough sell to management on a purely cost/benefit basis. After all, once a problem was reported, a crew still needed to be dispatched to correct the deficiency.

In the recent past, a plant would typically house nearly all controllers in a building room centrally located in the plant facility. All measurement data taken in the plant (field) was communicated to the control room over pairs of wires carrying a 4 to 20mA signals with one pair for each sensor. They were connected to the controllers. The feedback control signals to the final control element were sent back to the field in other pairs of wires carrying 4 to 20mA signals. The 4 to 20mA range was an open standard that all manufacturers of controllers, sensor and final control element agreed upon. A petrol chemical plant extends over several areas so that wire runs could be very long.

Eventually, these systems evolved into SCADA systems with capabilities that enable remote control of facilities, not just monitoring. This added ability to control facilities (open/close, on/off, up/down, etc.) within specific limits via an automated system enabled SCADA to make sense to corporate management and not just operations staff. As the price (and size) of computing power continued to be driven lower, and as computing standards emerged, the market for SCADA systems grew significantly. Now, even small- and medium sized oil and gas companies and utilities could afford these systems.

While these systems were a step forward, the high cost of the computing technology often made them a tough sell to management on a purely cost/benefit basis. After all, once a problem was reported, a crew still needed to be dispatched to correct the deficiency. As the price (and size) of computing power continued to be driven lower, and as computing standards emerged, the market for SCADA systems grew significantly. Lower computing cost also opened the door to added functionality, such as modeling and analysis software, at relatively small incremental costs. The quest for SCADA users today is for dependability, scalability, and flexibility. One avenue for this powerful mix that has appeared on the horizon is the use of the mobile phones, Internet and Web-based technology.

Mobile phones and Internet-based data monitoring brings new capabilities and unprecedented access to process measurement and control. Using standard process sensors, such as thermocouples or RTDs (temperature), pressure transducers, flow meters or other sensors that produce a standard analog or pulse output, one can monitor, control or log data in almost any location – across the hall, on the other side of the street, across the town, on opposite ends of the country, or anywhere around the world where one is connected to the internet or have mobile phone.

Using the power of the internet or mobile phone and similar technology, remote monitoring takes on new meaning. With Ethernet-based, internet-enabled instrumentation, remote access can be anywhere a Smartphone has a signal. From the simplest application, viewing data through the web-browser on your iPhone, Blackberry device or laptop, to more sophisticated uses, such as sending a text or e-mail message when an alarm occurs, or transmitting a data log file over the internet from a remote location to a central office. A user can access this data anytime, anywhere, 24 hours a day, 365 days a year, wherever one has internet access.

The "passive monitoring" that's built-in to Ethernet-based systems really does bring a new meaning to job security. One is secure in the knowledge that the process measurement and control system, no matter how large, is functioning optimally. How does one know this? One can confirm it with just a glance at his web browser. Many instruments can now log data to card; the data can be read directly on a PC or Mac, or can be downloaded remotely, over your Ethernet network or the internet.

Thanks to stand-alone units with built-in sensors, one can monitor temperature, humidity, even dew point, without needing anyone on-site. With common alarm sensors, one can be alerted if a door is opened, a window broken, or a fire sprinkler going off. The stand alone unit can send an e-mail to you, as well as to a distribution list; it can also send a text message to a cell phone. If one has an IP camera or web cam, one can see what is happening at site.

The internet and mobile phones has made it possible to send a lot of data from one side of the world to the other side in almost no time. The use of the mobile phones and the Internet for real-time interaction of the remote monitoring and controlling of the plants would give us many advantages. This technology cannot only be used in the industry, but also in the field of medicine, education, etc.

Although all this looks promising two main problems should be faced before the mobile phone and internet based control and monitoring can be implemented. The first one is the aspect of time delay, which can lead to irregular data transmission and data loss. In the worst-case this can make the whole system unstable. The other one is the problem of security. When malicious hackers gain access to a system the consequences can be catastrophic. Other problems concern the distance or logistics. If something goes wrong with the system, a lot of time and preparation can be needed before somebody can intervene to requirement specifications and system implementation.

**1.2 Statement of the Problem.**

It is not always feasible to monitor industrial processing unit physically in extreme environmental conditions of plant. This leads to improper maintenance causing break down. The delay in identifying causes of failure renders industrial units unusable. Such problems can be solved by constantly monitoring the plant and environment condition from a remote location.. This work seeks to solve this problem using Artificial Neural Networks and short message services (SMS)

### 1.3 Aim and Objectives.

The aim of this dissertation is to design Remote Industrial Process Monitoring and Control Systems using Artificial Neural Networks.

This aim will be realized by pursuing the following objectives:

- To design a system that will monitor industrial variables namely pressure, temperature, level and flow rate in an Oil and Gas plant and alert remote user in case of departure from set point...
- To design a system that can control industrial process plant from remote location.
- To design system that is self learning, self adaptive and self diagnosing using Rule based expert system.
- To design a multiprocessor-based control system that will keep each of the monitored variables within set limits using Artificial Neural Network approach.
- To design an SMS-Based software to alert industrial personnel when and if any of these variables goes out of bounds inspite of the control systems effort.
- To demonstrate the potential use by a layperson of a commercially available artificial neural network to monitor and control industrial process in Oil and Gas industry.

To realize the objective of this project the overview of the design methods and architectures developed so far in SHELL for industrial process monitoring and control was studied. Thereafter an SMS-based industrial process monitoring and control systems remotely operated which is simple and economically attractive for the oil and gas sector will be designed. This system will monitor and control many processes such as the temperature, pressure, level, flow rate, motor speed and heaters which if not monitored can cause serious damage under abnormal condition.
The above objectives will introduce dependability engineering in industrial process control.

Dependability engineering is concerned with techniques that are used to enhance the dependability of both critical and non-critical systems.

This technique supports three complimentary approaches namely:

i. Fault avoidance: Few faults means less chance of run-time failure.
ii. Fault detection and correction: The verification and validation process are designed to discover and remove faults in a program.
iii. Fault tolerance: Fault or unexpected system behavior is handled or managed in such a way that the system failure does not occur.

**1.4 Expected Results of the Research**

This research seeks to provide the following expected results:

i. re-engineering of industrial process monitoring and control in Nigeria oil industries
ii. an engineering solution on how to move our industries forward and bring it at par with modern technology
iii. Application of artificial neural networks and web agents to industrial process monitoring and control.
iv. Designing of a prototype model for Nigeria industries.
v. Developing a model that can be remotely operated, thereby ameliorating the problems of hazard and security in the Niger Delta which has been a major headache to Nigeria Government.

This will in turn increase productivity and lead to high quality products. It will also improve performance of system and devices as well as the reduction of down time of industrial plants.

- Improving plant profitability
- Process uptime
- Maintenance
- Energy cost
- Predictive monitoring predictive maintenance program

A system can change from data collection to analysis and improvement activities.

By continuous monitoring of equipment status one can be warned of certain machinery problems. Real-time control and protection' system can run longer and eliminate unplanned activities. The goal is to reduce maintenance costs.

Cost to repair a piece of machinery is higher when it fails unexpectedly and requires unplanned repair. This system will help plan maintenance activities by warning of accelerating failures that requires immediate attention and reduce unnecessary preventive maintenance activities that could introduce more failure potential of electronic and computer engineering equipment.

### 1.5 Motivation for the Research.

The demand for faster, simple, efficient and cost effective large scale system that will increase productivity, reduce cost, increase quality and flexibility in production process motivated this work.

### 1.6 Significance of the Study

The significance of this work cannot be over-emphasized. The work is both economically and technologically significant since Nigeria is lagging behind in technology. This problem has become so serious considering the fact that more than 95% of Nigeria foreign earning comes from Gas and Oil.

It is also more serious when one considers the fact that even though there is not enough gas in Nigeria gases are still flared. When one considers the fact that most of our oil wells in hazardous areas, in addition, to the problem of the militants in the Niger Delta, this work becomes significant. In order words the economic implication of this work is very enormous.

The project will assist to re-engineer the shell industrial process using artificial neural networks and rule-based expert system.

The need for this work is very important now that Nigeria is talking about engineering/ technological breakthrough as a necessary tool for vision 20-2020.

It is also important since 98% of Nigeria income depends on gas and oil, this work will increase productivity reducing down time of industrial plant..

Considering the fact that the world has become a global village, it is essential that one can stay anywhere in the world, monitor and control operations in industries.

Further significance includes:

- It will improve efficiency of the management and maintenance of field devices assigned to plants.
- Automatic creation of certain management documents and devices so that they can be managed electronically.
- Execution of adjustment and diagnosis available with device.
- Documenting of faults and their remedies for use in future repair.

Further significance of SMS- based monitoring and control application include:

• Scalability: Application of SMS makes the information available to all in the organization who need this data to do their job. By providing this data via their GSM, the cost of personnel doing their job (and the time to perform tasks) are greatly reduced, opening the door to new business applications for this data.

• Cost: The cost of using simple handheld devices greatly reduces the investment in proprietary hardware and software. SMS-based system can also turn a large capital investment in communication hardware and analytical software into an affordable monthly expense.

• Legacy Device Capabilities:

Solutions exist that allow existing field equipment to leverage GSM connectivity. Devices that were not designed to utilize the GSM can now be accessed, controlled, and managed securely using GSM technologies.

• Faster deployment: The use of industry standard networking technologies that are familiar to a broader group of engineers means it is easier to make in-house modifications and deployments or find local expertise that can learn the system.

. Operational significance: Consider, for example, an interstate pipeline system. With facilities, equipment and custody transfer points across literally hundreds of miles, the ability to access and manage real-time system operating data and customer data 24/7 is critical. Accomplishing all of this via the GSM network increases the availability of this data quickly and cost-effectively.

**1.4 Scope of Work.**

Remote Process monitoring and control is a combination of architectures, mechanisms, and algorithms used in the industrial factory for monitoring and controlling the activities of a specific process to achieve the goal.

A remote monitoring system for oil and gas industry is a combination of both software and hardware. This scope of this work is the design of a multiprocessor based monitoring and control system embedded in oil /gas environment with nine subsystems each of which has the need for monitoring of process temperature, pressure, level and flow rate. The software which characterizes the features of the intended artificial neural network would be developed using assembly language and C sharp. It will also feature an SMS based interface to the personnel if and when things go wrong. In addition to self leaning, self adaptive and self diagnonising features.

### 1.5 Organization of Dissertation.

This report is divided into six chapters as shown in figure 1.1 in order to make the understanding and appreciation of the work easy.

Chapter one, is the introductory part of the work and deals with the topic under research. It looks at the background of the study, the objective of the project, the significance of study, the background of case study and thesis organization.

Chapter two focuses on the review of related literatures carried out on the topic of the Dissertation.

Chapter three handles the systems analysis and design; the various methods used in collecting necessary data, such as observations, interview etc.

Chapter four deals with the system design.

Chapter five handles the system implementation, integration and testing.

Chapter six focuses on the summary of achievement, conclusion, recommendation, suggestion and contribution to knowledge.

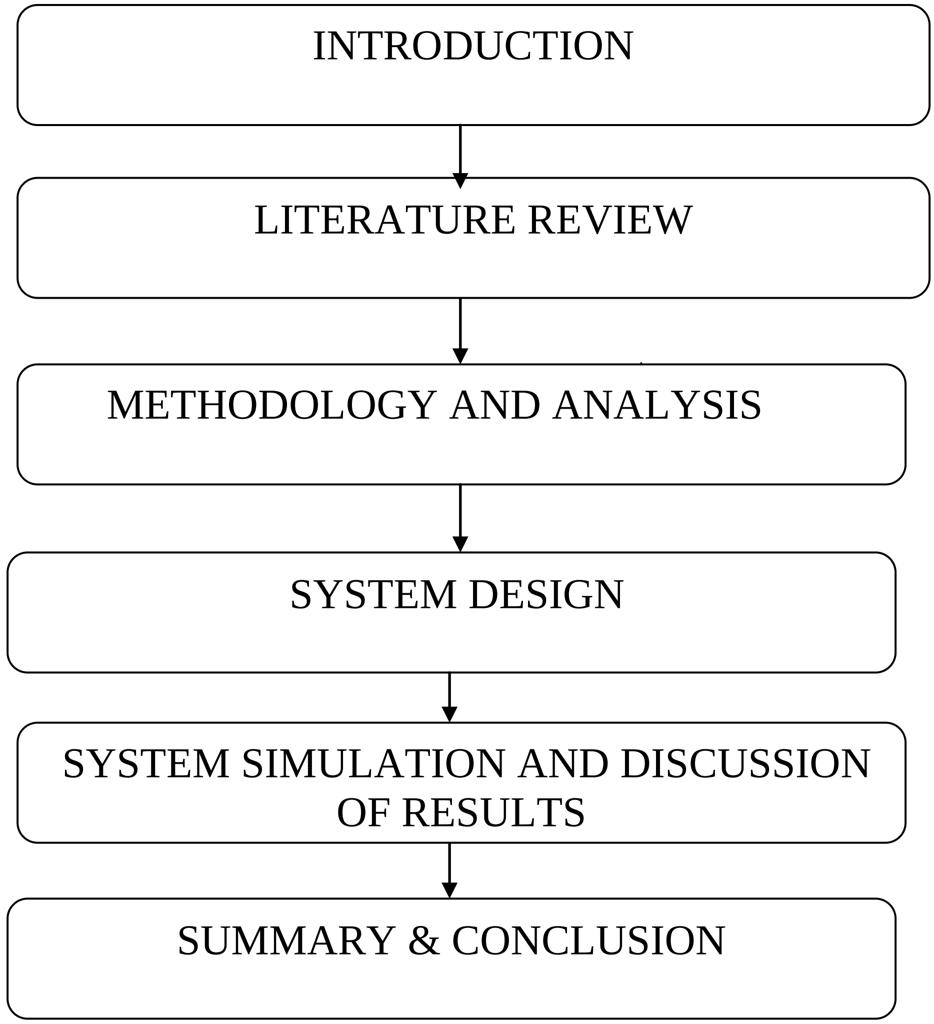



**Figure 1.1: Organization of Dissertation.**

# CHAPTER TWO
# LITERATURE REVIEW

## 2.1 Artificial Neural Networks

In recent years, many published papers have shown the results of research on Neural Networks (NN) and their applications in solving problems of control, prediction, and classification in industry, environmental sciences, and meteorology.

WU et al (2004) investigated the Model Reference Adaptive Neural Network Control approach based on back propagation algorithm to implement the water level control system.

Uraikul et al (2006) described the application of artificial intelligence for monitoring and supervisory control of process systems. Mitchell et al (2000) worked on using a neural network to predict the dynamic frequency response of a power system to an under- frequency load shedding. Lee et al (2011) proposed a neural SPSA on-line decoupled control scheme by using a PID neural network for a class of non-linear systems. Seema et al (2007) designed a neural network tuned fuzzy controller for multiple input multiple output system comprising a neural based tuned fuzzy controller for controlling the degree of freedom for MIMO system

Christos and Dimitrois (2001) defined Artificial Neural Network (ANN) as an information processing paradigm that is inspired by the way biological nervous system (such as the brain) process information. It is composed of highly interconnected application of neural (neurons) working in unison to solve specific problems. It is a powerful data-modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following ways:

- A neural network acquires knowledge through learning.
- A neural network's knowledge is stored without interneuron connection strengths known as synaptic weights.

The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics. Neural networks are widely used for classification, approximation, prediction, and control problems. ANN methodology has been reported to provide reasonably good solutions for circumstances where there are complex systems that are (a) poorly defined and understood using mathematical equations, (b) problems that deal with noisy data or involve pattern recognition, and (c) situations where input data is incomplete and ambiguous by nature. It is because of these characteristics, that it is believed an ANN could be applied to industrial process monitoring and control.

Based on biological analogies, neural networks try to emulate the human brain's ability to learn from examples, learn from incomplete data and especially to generalize concepts. A neural network is composed of a set of nodes connected by weights. The nodes are partitioned into three layers namely input layer, hidden layer and output layer Teo (2005) Teo.(2004). The neural network architecture used in this study is a simple 3-layer feed-forward ANN, where the data flows in a single direction, which are from the input layer through to the hidden layer and finally to the output layer. Many researchers have worked on the development of ANNs. Generally, most of the ANNs are ready to take the advantage of future parallel hardware. By considering these facts ANNs will be used for the classification in this study.

### 2.1.1 Structure of Biological Neural Networks.

It has long been known that learning in animals and humans can be achieved through observation of examples. The exact mechanism by which this learning takes place is still unknown, but science has yielded some clues. The vertebrate brain consists of an enormous number of interconnected cells called neurons. It has since become widely accepted that these neurons are the fundamental information processing elements of brains. The cell body of neuron is called the soma. The extensions of the cell are called dendrites. The axon extends away from the cell body to provide a pathway for outgoing signals. According to Toning (2010) signals are transferred from one neuron to another through a contact point called a synapse. It has been found that neurons respond to electrical impulses collected from other neurons through connecting fibres called axons and dendrites. Figure 2.1 shows a typical biological neuron.

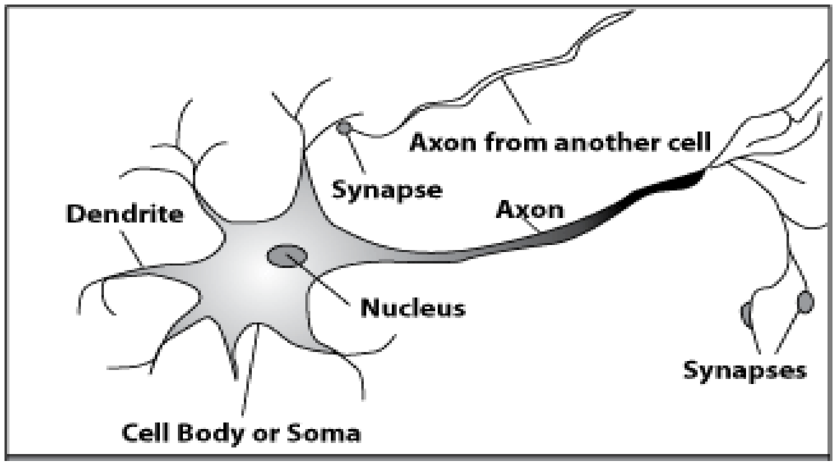



**Figure 2.1: A Biological Neuron.**[http://www.disagroup.com]

Prompted by studies in neuroscience, McCulloch and Pitts (1943) developed a simple mathematical model for a neuron as shown in Figure 2.2

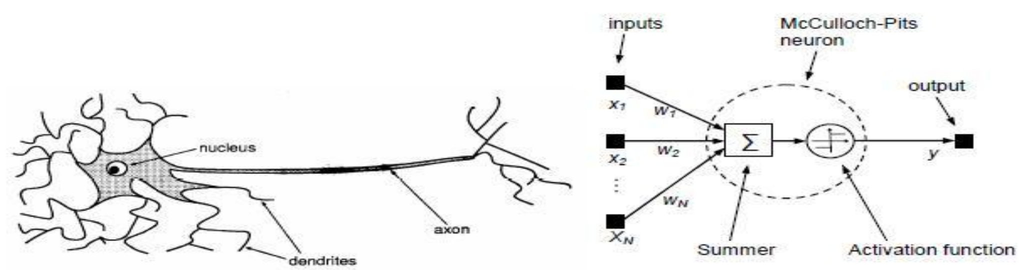



**Figure 2.2: A biological neuron and its model (McCulloch-Pitts neuron).**

Mainly, there are two types of ANNs: supervised and unsupervised. The supervised ANNs require an initial training. Unsupervised ones may start to monitor the signals without any training. Among the supervised ANNs, the feed-forward ANNs (FFNN) have been widely used. The Back-propagation (BP) algorithm is the most popular one for estimation of the weights and was used in many applications Huang et al (2007); Lu et al (2000) Tansel et al., (2009); Aykut et al (2010); Tansel et al (2009); Demetgul et al (2009). Quasi-Newton approaches such as Levenberg-Marquardt was developed to increase the speed of the estimation and is available in the MATLAB

ANN Toolbox Beale et al (2010). Carpenter et al (1992) allowed the use of the Adaptive Resonance Theory (ART) for the supervised learning. Among the unsupervised Artificial Neural Networks, Adaptive Resonance Theory 2 (ART2) (Grossberg (1987), Carpenter& Grossberg (1987), Rajakarunakaran et al., (2008), Lee et al (2003), Belforte et al., (2004) has been successfully used for classification in many applications.

### 2.1.2 The artificial neuron

A network consists of a number of elements or nodes, denoted as $x_i$. Each node receives signals from other nodes, processes and forwards them to other nodes. There is at every moment an activity in each node, here denoted as $x_i$. Nodes are connected by directed connections, denoted as w, which has a weight or strength. Node i is connected with node j with connection $w_{ij}$.

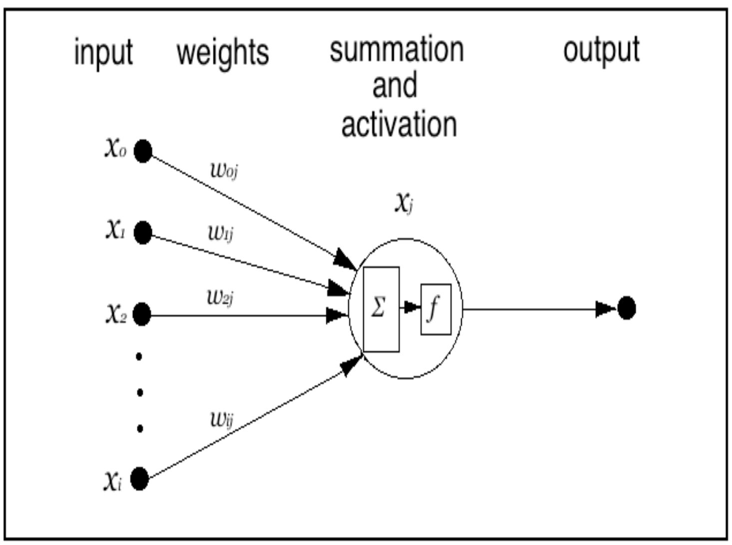



**Figure 2.3: An Artificial Neuron [**Tom & Lovric(2011)]

Signal dynamics of a network can be modeled either as continuous or discrete. The discrete is easier to explain. Input or other elements' activities are transformed into signals and proportionally strengthened by the weights. When inside the node, all signals are summarized.

The inner sum corresponds to: $\Sigma = x - in_j = \sum_{i=j}^{n} x_i w_{ij}$ (1)

Where

x = input to the network

w = weight of the input

n = node.

The activation function takes the summarized input as argument and the output value of this function is the nodes' resulting activity or output. The activating function is denoted with *f* and the resulting activity with xj.

### 2.1.3 Composition of Artificial Neural Network

Figure 2.4 is diagram showing the composition of Artificial Neural Network. ANN is composed of three layers of function. They consist of (a) an input layer, (b) a hidden layer, and (c) an output layer. The hidden layer may consist of several hidden layers.

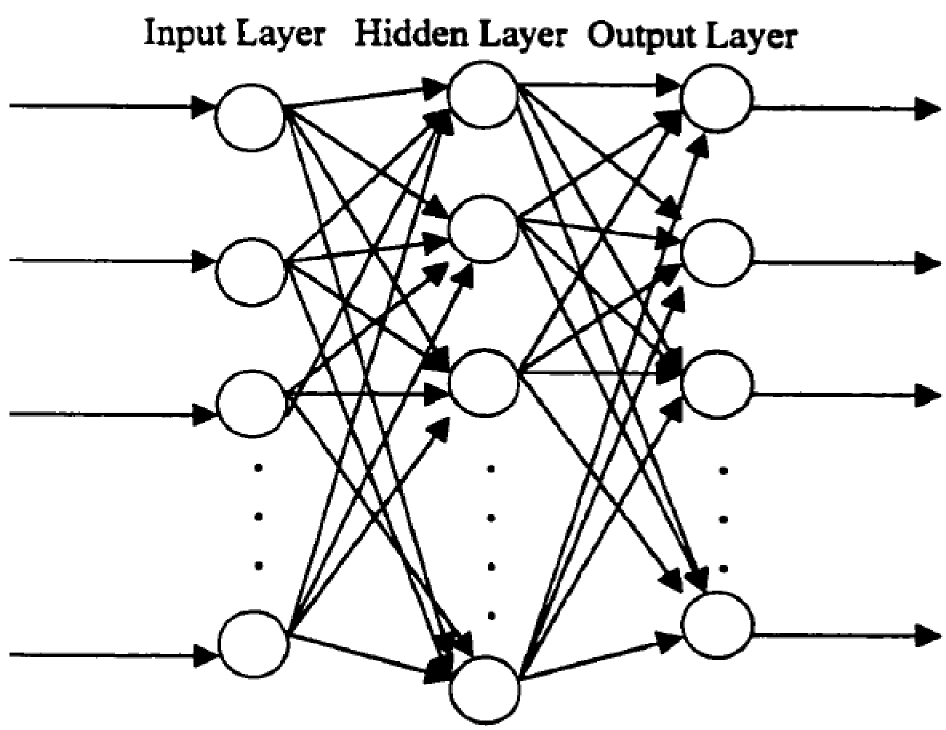



**Figure 2.4: Diagram of ANN (Mashudi, 2001)**

The input layer receives or consists of the input data. It does nothing but buffer the input data. The hidden layers are the internal functions of the NN. The output layer is the generated results of the hidden layers.

The two types of ANN are (a) feed-forward network and (b) a feedback network. The feed-forward ANN has no provision for the use of output from a processing element (hidden layer) to be used as an input for a processing unit in the same or preceding hidden layer. A feedback network allows outputs to be directed back as input to the same or preceding hidden layer. When these inputs create a weight adjustment in the preceding layers, it is called back propagation. An ANN learns by changing the weighting of inputs. During training, the ANN sees the real results and compares them to the ANN outputs. If the difference is great enough, the ANN then uses the feedback to adjust the weights of the inputs. The feedback learning function defines an ANN.

The general procedure for network development is to choose a subset of the data containing the majority of the process variables, train the network, and test the network against the remaining process variables. In this situation, the recorded variables will be divided into two sets—one large training set and a second smaller testing set. Once the ANN has been trained and tested for accuracy, it can be updated on a continuing basis using data provided through data acquisition sub systems from temperature, pressure, level and flow rates sensors.

As has happened in many fields, ANNs have generated their own terms and expressions that are used differently in other fields. To prevent confusion, the following are the definitions of specific terms used in ANNs (Markus, 1997):

Activation is the process of transforming inputs into outputs.

Architecture is the arrangement of nodes and their interconnections, (structure).

Activation Function is the basic function that transforms inputs into outputs.

Bias and Weights are the model parameters (Biases are also known as shifters. Weights are called rotators).

Layers are the elements of the ANN structure (input, hidden, and output).

Learning is the training and parameter estimation process.

Learning Rate is a constant (or variable) which shows how much change in error affects change in parameters.

### 2.1.4 Activation functions

Activation functions are functions that are involved in deciding how the nodes and the network process signals. The function controls when the neuron should be active and that depending on if a given threshold is reached or not.

In the standard theory of neural network a few basic types of activation functions are elaborated. We mention here some of the simplest like linear, threshold functions, activating trough competition and sigmoid functions ( Stevens & Lovric 2011)

. When choosing the activation function some important factors need to be considered. Using linear functions in multilayer network is pointless because the biological correspondence is nonlinear. Another important factor in recurrent network is the stability aspect. Since the signals are strengthened and summarized, they can get arbitrarily high values and for that reason the activation function has to be of limiting nature. For the training procedure, especially in the error back-propagation network, it is important that the functions are of derivative nature. Some useful activation functions are signum, log-sigmoid transfer and tan-hyperbolic functions.( Stevens & Lovric 2011)

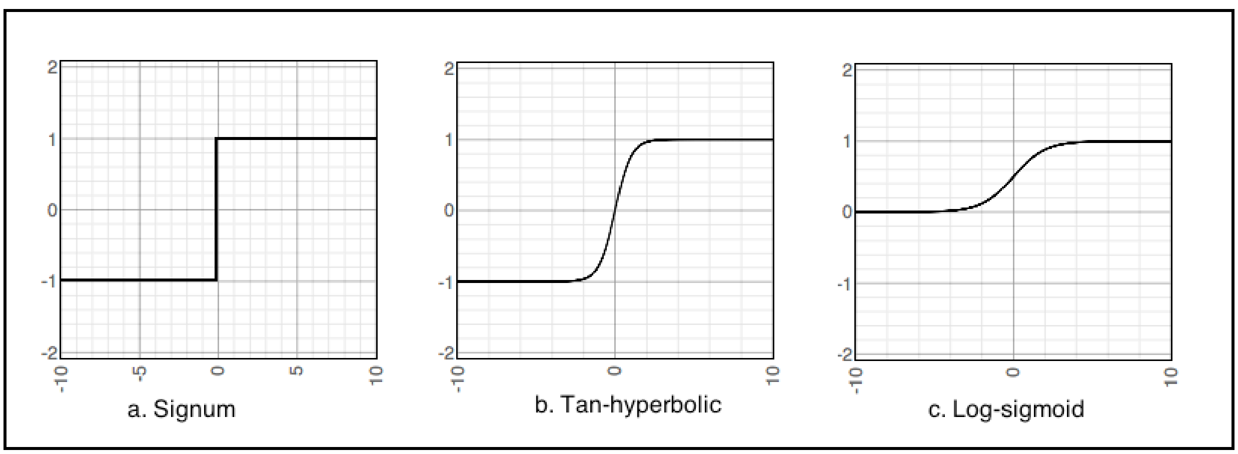



**Figure 2.5: Activation functions** [ Stevens & Lovric (2011)]

Graphical representations of useful activation functions are depicted in Figure 2.5. The log-sigmoid activation function (Figure 2.5c) should only be used with positive values, since it blocks out negative ones.

$$f(x) = \frac{1}{1+e-x} \quad (2)$$

Where,

e =Exponential (log sigmoid)
x = input node

### 2.1.5 Supervised ANN

The Feed Forward Neural Network (FFNN) belongs to the category of supervised learning ANN method. In this method, information moves in only one direction.

Feed Forward Neural Network (FFNN) became popular with the widespread use of the back propagation (BP) (Bryson & Ho, (1969) Rumelhart et al. (1976) algorithm. The FFNN have multiple layers. Generally, single hidden layer is used. The user determines the number of the hidden neurons of this layer by trial and error. The number of the neurons of the input Conditioning Monitoring and Fault Diagnosis for a Servo-Pneumatic System with Artificial Neural Network Algorithms and output layers depends on the application. The BP estimates the weights of the neurons by updating them after the forward and backward propagation of error. The learning rate and the momentum are two important parameters of the BP for training the network successfully (Chen & Mo, (2004); McGhee et al (1997)). Levenberg-Marquardt algorithm Beale et al (2010) generally estimates the parameters of the FFNNs. It finds the best weights by minimizing the function. It works effectively for many applications. (Beale et al (2010). Carpenter et al (1991),and Carpenter et al (1992) used the fuzzy logic and ART ANNs. It evaluates the similarity by considering the fuzzy subset hood and ART category choice. The vigilance is used to determine the size of the “category boxes” or sensitivity of the ANN. One of the very important advantages of the ARTMAP with or without the fuzzy component over the FFNNs is the use of the vigilance based on our experience. Aaron Garrett’s Garrett (2003) code was used for the training and testing of the fuzzy ARTMAP method.

### 2.1.6 Unsupervised ANN

The self organizing map belongs to the family of unsupervised leaning ANN methods. ART2 type ANN evaluates the characteristics of the inputs and assign them a category (Carpenter & Grossberg, (1987); Lee et al., (2003), Yang et al., (2004), Na et al (2008)). If the signal looks like one of the previously presented signals, it will be classified in the same category. On the other hand, if the signal is different from the previously presented ones a new category is assigned for it. The sensitivity of ART2 depends to the vigilance. At the low vigilances, it has higher tolerance. When the vigilance approaches to one it will be more selective. Fuzzy ART use fuzzy set theory in the ART1 type ANN structure. With the help of the MIN operator of the fuzzy set theory the classification of the binary and analog input patterns is possible. The vigilance parameter adjusts the selectivity of the ANN. In this study Aaron Garrett Garrett, (2003) implementation of the fuzzy ART was used. The most common neural network model is the multiplayer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when desired output is unknown. A graphical representation of an MLP is shown in figure 2.6

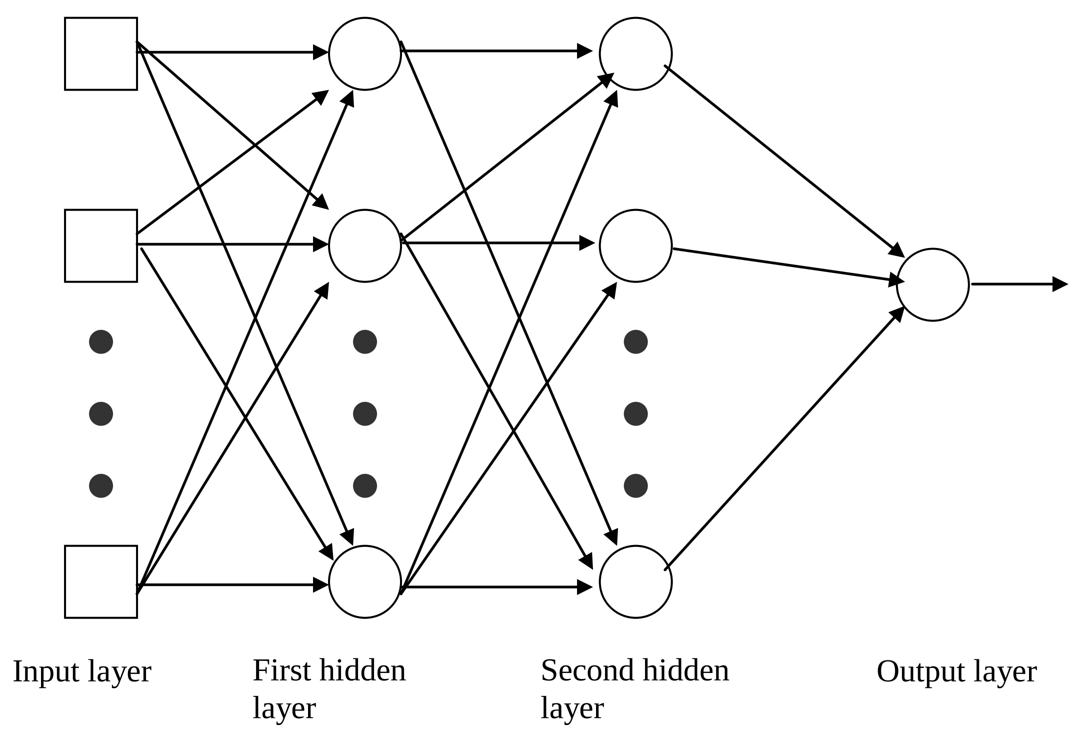



**Figure 2.6: Two Hidden Layer Multiplayer Perceptron**

The inputs are fed into the input layer and get multiplied by interconnected weights, as they are passed from the input layer to the first hidden layer. Within the first hidden layer, they get summed then processed by a non-linear function (usually the hyperbolic tangent). As the processed data leaves the first hidden layer, again it gets multiplied by interconnection weights, then summed and processed by the second hidden layer. Finally the data is multiplied by interconnection weights then processed one last time within the output layer to produce the neural network output.

The MLP and many other neural networks learn using an algorithm called back-propagation. With back-propagation, the input data is repeatedly presented to the neural networks. With each representation, the output of the neural network is compared to the desired output and an error is computed. This error is then fed back (back-propagated) to the neural network and used to adjust the weights such that the error decreased with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "training". (www.neurosolutions.com/products/ns/whatsNNhtml)

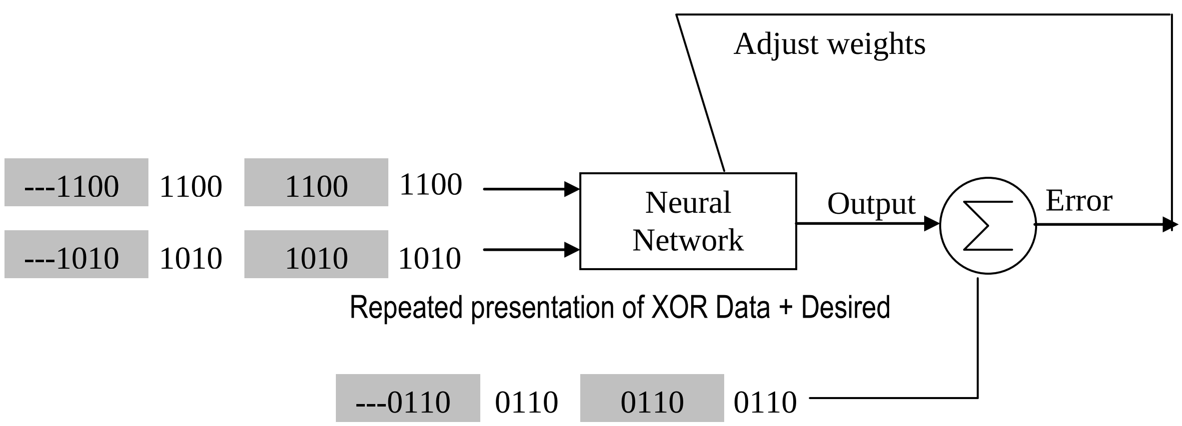



**Figure 2.7: Using Neural Network Learning To Model XOR [Source: www.neurosolutions.com/products/ns/whatsNNhtm**l].

The XOR data is repeatedly presented to the neural network as shown in figure 2.7. With each presentation, the error between the network output and the desired output is computed and fed back to the neural network. The neural network used this error to adjust its weight such that the error will be decreased. This sequence of events is usually repeated until an acceptable error has been reached or until the network no longer appears to be learning. A good way to introduce the topic is to take a look at a typical application of neural networks. Many of today's document scanners for the PC come with software that performs a task known as optical character recognition (OCR). OCR software allows you to scan in a printed document and then convert the scanned image into an electronic text format such as a word document, enabling you to manipulate the text. In order to perform this conversion, the software must analyse each group of pixels (0s and 1s) that form a letter and produce a value that corresponds to that letter. Some of the OCR software in the market uses a neural network as the classification engine.

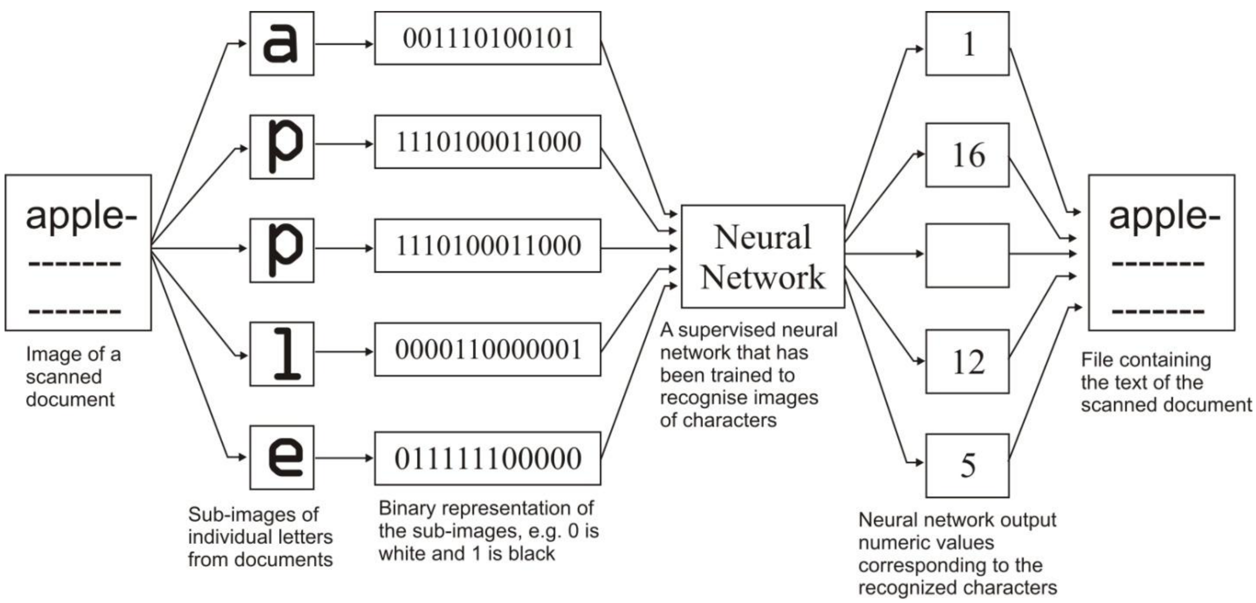



**Figure 2.8: Demonstration of a neural network used within an (OCR) application. [Source:www.neurosolutions.com/products/ns/whatsNNhtml]**

The original document is scanned into the computer and saved as an image. The OCR software breaks the image into sub-images, each containing a single character. The sub-images are then translated from an image format into a binary format, where each 0 and 1 represents an individual pixel of the sub-image. The binary data is then fed into a neural network that has been trained to make the association between the character image data and a numeric value that corresponds to the character. The output from the neural network is then translated into ASCII text and saved as a file as shown in figure 2.8

### 2.1.7 Summary of Related Research Efforts

Eliana (2002) used artificial neural networks (ANN's) to describe the behavior of a process. In this research, a soft sensor is developed for a batch distillation column, in order to estimate product compositions using available temperature measurements. A nonlinear deterministic model of the process, which was obtained in a previous study, was used to simulate the dynamic behavior of the batch column under different operating conditions. The databases generated in this way are employed to develop the ANN-based model of the process to be implemented on line. Various feed forward neural networks are considered and investigated for ease of training and accuracy of composition estimation. It is shown that a direct correspondence exists between optimal network structure and data characteristics, since it is possible to identify the most suitable ANN using information retained by the available process data together with limited a-priori knowledge about the process.

Kirubashankar etal (2007) proposed a remote monitoring system as part of distributed control system of process plant. The following parameter of DCS were monitored continuously: Alarm, graphic display, trend display, system and diagnostic display, control, bar chart, sequence display and fault analysis display. A database is created in MS Access/MY SQL in SCADA for monitoring alarm code, date and time of occurrence of alarm. A SCADA VB application is created for transferring generated alarm report to MS Outlook express with user ID CODE. It takes GPRS radio connection and internet technology to communicate between remote monitoring computer and field monitoring computer. M2M gateway is used to build wireless GPRS networks running independent of mobile operator. The researcher did not use ANN.

Taylor and Sayda (2003) proposed the use of Expert system in abnormal event management (AEM) in large manufacturing plants. They introduced different computational agents embodied in a three-layered cognitive hierarchy, which offers intelligent behavior at the system level, as well as at the level of specialized task agents. This researcher used expert system only to implement this work.

Ahmad et al (2001) proposed the application of artificial neural networks in the area of process monitoring, process control and fault detection. This study focused on the malfunctions of the process caused by the failure of the pressure, temperature and cooling water temperature sensors in the reactor. Faulty conditions are simulated using the Tennessee Eastman Plant model coded in

MATLAB language. Sensor failures are created causing the normal process operation to shift to a faulty operation mode. For the pressure sensor, deviation of 4.0% or greater from the normal condition is assumed to cause malfunction to the process. Similarly, for temperature and cooling water temperature sensors the figures are 6.0% and 3.0% respectively.

In the neural network-based control systems, a neural network is often trained to estimate the unknown nonlinear process and a controller is then formulated based on the neural network. In the fault detection Sub System Neural networks used for process fault detection generally use sensor measurement and process alarms as inputs, while the outputs represent particular fault types, or categories. In the ideal situation, if the value of a neuron in the output layer of the network is equal to one, then the fault represented by that particular neuron is considered to be present. They concluded that the application of neural network in monitoring reactor temperature gave a successful result. This research effort concentrated more on sensor fault not the control of industrial variables..

Ramamurthy et al (2010) developed a Low-Cost GSM SMS-Based Humidity Remote Monitoring and Control system for Industrial Applications proposes a wireless solution, based on GSM (Global System for Mobile Communication) networks for the monitoring and control of humidity in industries. This system provides ideal solution for monitoring critical plant on unmanned sites. The system is Wireless, utilizing Humidity sensor HSM-20G, ARM Controller LPC2148 and GSM technology. Historical and real time data can be accessed worldwide using the GSM network. The system can also be configured to transmit data on alarm or at preset intervals to a mobile phone using SMS text messaging. The proposed system monitors and controls the humidity from the remote location and whenever it crosses the set limit the LPC2148 processor will sends an SMS to a concerned plant authority(s) mobile phone via GSM network. The concerned authority can control the system through his mobile phone by sending AT Commands to GSM MODEM and in turn to processor. This work was in the number of parameters that can be measured and only a processor was used for this work

Popoola1 et al (2013) investigated the expert system design and control of crude oil distillation column (CODC) using artificial neural network model which was validated using experimental data obtained from functioning crude oil distillation column of Port-Harcourt Refinery, Nigeria. MATLAB program was written for the artificial neural network back-propagation algorithm using the implementation steps of the artificial neural network. Out of the one-hundred and thirty (130) experimental data sets obtained, ninety percent (90%) were used for training the network while the remaining ten percent (10%) were used for testing the network to determine its prediction accuracy. The accuracies obtained for the design were 94%, 99%, 92%, 93%, 81%, 95% and 90% for temperature at which 100% (T100) of Kerosene, 90% (T90) of Diesel and 10% (T10) of AGO were distilled; and naphtha, kerosene, diesel and AGO flow rates respectively. The researchers concentratd more project quality.

Zalizawati (2008) proposed the development of multiple-input multiple-output (MIMO) and multiple-input single-output (MISO) neural network models for continuous distillation column. The input-output data for the neural network model was generated from the validated general first principle model. Based on the input-output analyses, boiler heat duty, reflux flow rate and tray temperatures were selected as the inputs for the neural network model. Seven different profiles were designated to excite the first principle model to generate the input-output data. These sets of data were then divided into training, validation and testing data. The results showed that the first principle and the neural network models which were developed were in good agreement with the experimental data. The parameters that were monitored were limited.

Huffman (2007) used Neural Network to predict flood disaster.

The river system chosen for the research was the Big Thompson River, located in North-central Colorado, United States of America. The Big Thompson River is a snow melt controlled river that runs through a steep, narrow canyon. In 1976, the canyon was the site of a devastating flood that killed 145 people and resulted in millions of dollars of damage.

Using publicly available climatic and stream flow data and a Ward Systems Neural Network, the study resulted in prediction accuracy of greater than 97% in +/-100 cubic feet per minute range. The average error of the predictions was less than 16 cubic feet per minute.

In this dissertation, a daily rainfall-runoff model for two flow-measuring stations, Drake and Loveland, on the Big Thompson River in Colorado, was developed using a Ward System NN program called the NeuralShell Predictor. The study demonstrated the feasibility of using a commercially available NN to accurately predict day-to-day normal flows of a river and to predict extreme flow conditions commonly called flood events. The focus of the research was not on industrial process.

Umair and Usman(2006) developed an Automatic Irrigation System using Artificial Neural Network based Controller to schedule irrigation system so as to know when to irrigate and how much water to be applied. The input variables to the system were:

• Soil (ground) humidity;

• Temperature

• Radiation

• wind speed

• Air humidity

• Salinity (amount of salt in the ground).

The output parameters were:

• opening/closing the valves for water and/or fertilizer, and adjusting their amounts in combination;

• Turning energy systems on/off (lights, heating, ventilation);

• Opening/closing walls and roofs of hothouses.

. This research effort was not centered on industrial process monitoring and control.

Demetgul (2007) used Artificial Neural Networks to monitor the faults of a pneumatic system. Experimental data was collected by using the National Instrument (NI) compact Field Point measurement system with control modules. The Lab VIEW program environment controlled the measurement system. The values of four analog parameters were monitored. Three of these parameters were the pressure readings of the cylinders creating the motion in the x and y directions and the overall system. The Fourth analog input was the readings from the linear potentiometer. The gripper action was monitored from the digital signals coming from data acquisition card. The sensors provided long data segments during the operation of the system. To represent the characteristics of the system the sensory signals were encoded by selecting their most descriptive futures and presented to the ANNs. Two gripper sensor signals were monitored one for pick and one for place .Their outputs were either 0 V or 1V. The grippers pick and place signals were encoded by identifying the time when the value raised to 1V and when it fell down to 0V. The signals of the pressure of x axis, pressure of y axis and main pressure were encoded by calculating their averages. The work was centered mainly on monitoring the faults of a pneumatic system

Haydary et al. (2009) performed steady-state and dynamic simulation of pre-flash and atmospheric column (Pipestill) in a real crude oil distillation plant using ASPEN simulations. Steady-state simulation results obtained by ASPEN plus were compared to real experimental data. Experimental ASTM D86 curves of different products were compared to those obtained by simulations. Steady-state simulation results were in good agreement with experimental data.

Non-linear model predictive control (NMPC) of a distillation column using Hammerstein model and nonlinear autoregressive model with exogenous input (NARX) was proposed by

Kanthasamy (2009) in which he concluded that the model results showed a high level of consistency with the experimental results. The focus of this work was on product quality

Stevens and Lovric ( 2011) used neural networks in fault diagnosis of sand moulding machines manufactured by DISA. A description of the sand moulding process and DISA's current fault detection and diagnosis procedure is given together with testing the potentials of feed-forward neural networks for recognizing patterns represented in control charts based on data of 16 sampled channels on the DISAMATIC moulding machine. The testing in Matlab and Encog environment proved that neural networks can learn to recognize periodic patterns in presented data but accepts too large deviations in patterns. The concluding part in this work reveals that an application based solely on neural networks, is not the sustainable solution and some prior signal processing of the sampled input is necessary.

Kaparthi and Suresh (1991) presented a neural network clustering method for the part-machine grouping problem. This method is based on a neural network algorithm to support procedures like production flow analysis. A neural network clustering algorithm using similarity coefficients is used to solve the part-machine grouping problem. The neural network method is based on unsupervised learning. In the part-machine matrix, each row (part) is considered as a vector in a higher dimensional space and every dimension corresponds to a machine type, and the number of dimensions is given by the total number of machine types required for all parts Kaparthi and Suresh, (1992). The authors have shown that the neural network is capable of handling large data sets. The researchers used ANN more for grouping and analysis

Lee et al.(1992) presented a method for part family formation, machine cell identification, bottleneck machine detection and the natural cluster generation is done using a self-organizing neural network. The authors argue that the generalization ability of the neural network makes it possible to assign the new parts to the existing machine cells without repeating the entire computational process. The authors show that neural networks can learn from a given set of patterns and are able to generalize this knowledge to other similar problems. This property makes them useful in small and medium-size batch manufacturing systems where training data are limited and new parts are continuously encountered. The authors point out that their method based on neural networks is not significantly influenced by the size of the machine-part matrix and hence it is appropriate for solving large-scale industrial problems. The work was not on monitoring and control of industrial process variables.

In condition monitoring the ANN models, taking data from an online condition monitoring system, can predict tool life that would help to generate an appropriate maintenance schedule. Kong and Nahavandi (2002) developed such a model for the forging process that uses a multilayer error back propagation network. The inputs of the model were force, acoustic emission signals, process parameters (such as tool temperature, stroke rates, and surface lubrication condition of in-feed material), and expected life. The model helps to predict the tool condition, maintenance schedule, and tool replacement. Similar techniques can be applied to other metal forming processes. The researchers were limited in the number of parameters monitored.

. Huang and Chen (2000) developed an in-process tool breakage detection system using a neural network for an end mill operation. The inputs of the model were cutting force and machining parameters such as spindle speed, feed rate, and depth of cut. The output was to detect the tool breakage conditions. Per their report, the neural networks were capable of detecting tool condition accurately. This work was not on the control of industrial variables.

Brophy et al (2002) proposed a two-stage neural network model to detect anomalies in drilling process. The network was used to classify drilling operations as normal or abnormal (e.g, tool, breakage or missing tool). The network used spindle power signal (acquired over all or part of the operation) as the input. A limitation of the approach is that it requires the full signal before a classification is made.

### 2.1.8 Applications

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

The tasks artificial neural networks are applied to tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.
- Robotics, including directing manipulators, Computer numerical control.

Application areas include system identification and control (vehicle control, process control), quantum chemistry, game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications (automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

### *2.2 Types of Models*

Many models are used in the field defined at different levels of abstraction and modeling different aspects of neural systems. They range from models of the short-term behavior of individual neurons, models of how the dynamics of neural circuitry arise from interactions between individual neurons and finally to models of how behavior can arise from abstract neural modules that represent complete subsystems. These include models of the long-term and short-term plasticity, of neural systems and their relations to learning and memory from the individual neuron to the system level.

While initial research had been concerned mostly with the electrical characteristics of neurons, a particularly important part of the investigation in recent years has been the exploration of the role of neuron modulators such as dopamine, acetylcholine, and serotonin on behavior and learning.

Biophysical models, such as BCM theory, have been important in understanding mechanisms for synaptic plasticity, and have had applications in both computer science and neuroscience. Research is ongoing in understanding the computational algorithms used in the brain, with some recent biological evidence for radial basis networks and neural back propagation as mechanisms for processing data.

Computational devices have been created in CMOS for both biophysical simulation and neuromorphic computing. More recent efforts show promise for creating Nan devices for very large scale principal components analyses and convolution. If successful, these efforts could usher in a new era of neural computing that is a step beyond digital computing, because it depends on learning rather than programming and because it is fundamentally analog rather than digital even though the first instantiations may in fact be with CMOS digital devices.

### 2.2.1 Application of ANN

Artificial Neural networks have been successfully applied to broad spectrum of data-intensive applications, such as:

- **Process modeling and control:** Creating a neural network model for a physical plant then using the model to determine the best control settings for the plant.
- **Machine Diagnostic:** Detect when a machine has failed so that the system can automatically shut down the machine when this occurs.
- **Target Recognition:** Military application which uses video and/or infrared image data to determine if an enemy target is present.
- **Portfolio Management:** Allocate the assets in a portfolio in a way that maximizes return and minimizes risk.
- **Medical Diagnosis:** Assisting doctors with their diagnosis by analysing the reported symptoms and/or image data such as MRLs or X-rays.
- **Credit Rating:** Automatically assigning a company's or individual's credit rating based on their financial condition.
- **Target Marketing:** Finding the set of demographics which have the highest response rate for a particular marketing campaign.
- **Voice Recognition:** Transcribing spoken words into ASCII text.
- **Financial Forecasting:** Using the historical data of a security to predict the future movements of that security.

- **Quality Control:** Attaching a camera or sensor to the end of a production process to automatically inspect for defects.
- **Intelligent Searching:** An Internet search engine that provides the most relevant content and banner ads based on the user's past behaviour.
- **Fraud Detection:** Detect fraudulent credit card transactions and automatically decline the charge.

As neural networks have application areas so does it have the major advantages which include:

- They do not need to be programmed, they can simply be taught the desired response. This eliminates most of the cost of programming.
- They can improve their response by learning. A neural network designed to evaluate loan applications, for example, can automatically adapt its criteria based on loan failure feedback data.
- Input data does not have to be precise, because the network works with the sum of the inputs. A neural network images recognition system, for example, can recognize a person even though he or she has a somewhat different hairstyle than when the "learning" image was taken. Likewise, a neural network-based speech recognition system can recognize words spoken by different people. Traditional digital techniques have a very hard time with these tasks.
- Information is not stored in a specific memory location, than is in a normal digital computer. It is stored associatively as a network of interconnections and weightings. The result of this is that the "death" of a few neurons will usually not seriously degrade the operation of the system. This characteristic is also fortunate for us human.

## 2.3 Neural Networks in Process Control

According to Hunt.et-al (1992), a neural controller is an automatic controller in which an artificial neural network is utilized in the process of calculating the control signal as shown in figure 2.9. As described previously, the functionality of an ANN is usually wholly defined through learning by example. This implies that an important part of the design of a neural controller is the design of a mechanism by which learning exemplars can be collected and used to adjust the parameters of the ANN. This mechanism is very similar to that of adaptive controllers, so most neural controllers in fact are adaptive by nature.

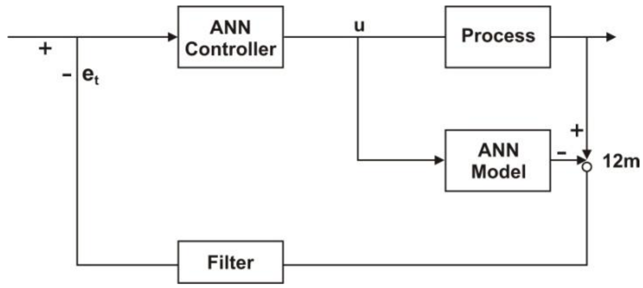



**Figure 2.9: Artificial Neural Network [hunt & Sharbero (1991)]**

### 2.3.1 Network function

The word *network* in the term 'artificial neural network' refers to the inter–connections between the neurons in the different layers of each system. An example system has three layers. The first layer has input neurons, which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations.

An ANN is typically defined by three types of parameters:

1. The interconnection pattern between different layers of neurons
2. The learning process for updating the weights of the interconnections
3. The activation function that converts a neuron's weighted input to its output activation.

### 2.4 Monitoring System

According to Summerville, (2010), monitoring systems compare observation of system behavior with standards that seem crucial for successful goal attainment. In modern software implementations of artificial neural networks, the approach inspired by biology has been largely abandoned for a more practical approach based on statistics and signal processing (Summerville, 2010). In some of these systems, neural networks or parts of neural networks (such as artificial neurons) are used as components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such adaptive systems is more suitable for real-world problem solving, it has far less to do with the traditional artificial intelligence connectionist models. What they do have in common, however, is the principle of non-linear, distributed, parallel and local processing and adaptation. Neural network models in artificial intelligence are usually referred to as artificial neural networks (ANNs); these are essentially simple mathematical models defining a function $f:X\rightarrow Y$ or a distribution over $X$ or both $X$ and $Y$, but sometimes models are also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase ANN model really means the definition of a *class* of such functions (where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity).

### 2.4.1 Remote Monitoring and Control

Remote monitoring and intelligent maintenance is one of the most important criteria for maximizing production and process plant availability. Kirubasanka et-al (2009).Enhancing the safety, efficiency and security of production and process control infrastructure is a growing requirement for many organizations with infrastructure located in remote parts of the world. A common problem facing these organizations is the difficulty in establishing cost effective methods of monitoring and controlling their remote assets.

Remote monitoring and control refers to a field of industrial automation that is entering a new era with the development of wireless sensing devices. Initially limited to SCADA technology, remote monitoring and control refers to the measurement of disparate devices from a network operations center or control room and the ability to change the operation of these devices from that central office. Kirubashankar et al, (2009) states that remote control allows clients to control their homes or installations from any place, whereas remote monitoring provides the clients the ability to monitor their installations, home or premises when they are away from the immediate environment in question.

Process monitoring and control applications range from data sensing, measurement, record and diagnosis, to machine/equipment operation and emergency action. These operations are classified by the ISA100 committee into six different classes with increased priority as following

Class 5: Monitoring without immediate operational consequences.

Class 4: Monitoring with short-term operational consequences

Class 3: Open-loop control

Class 2: Closed-loop, supervisory control

Class 1: Closed-loop, regulatory control

Class 0: Emergency action

Data communication in these process control applications includes continuous data request and asymmetric or spontaneous data traffic. For smooth operation, the major concern is the quality of service, which requires the correct data at the right time, i.e., the reliability of the data and the real-time guarantee.

Process monitoring and control is a combination of architectures, mechanisms, and algorithms used in the industrial factory for monitoring and controlling the activities of a specific process to achieve the goal, Gang (2010). Let us explain this by illustrating a simple wired application. For example, cooling down a reactor by adjusting the flow rate through the cooling jacket is a process that has the specific, desired outcome to reach: maintaining a constant predefined temperature over time. Here, the temperature is the controlled variable. At the same time, it is the input variable since it is measured by a temperature sensor and used in a special function to decide the adjustment of a valve to manipulate the flow rate through the cooling jacket. The desired temperature is the set point. The valve opening position (e.g. the setting of the valve allowing cooling material to flow through it) is called the manipulated variable since it is subject to control actions. In practice, the temperature value is transmitted to the controller; the controller implements the functions and calculations, transmits the output to control the valve and issues alarm if there are faulty conditions. In the meantime, all data information can be archived for future reference when a review of process trends could provide additional improvements. Utilizing Wireless Sensor Network technology, sensing and action devices will communicate wirelessly with an access point (e. g., a gateway or router), which is connected to the control station wirelessly or through wired methods. (Zhao, 2010).

Most of the process control applications are mission critical and have stringent requirements. Failure of a control loop may cause unscheduled plant shutdown or even severe accidents in process-controlled plants. Today the application of remote monitoring and control extends from fields such as:

- smart grid

- positive train control
- Structural health monitoring
- Pipeline sensors
- Patient monitoring
- Desktop/server monitoring

### 2.4.2 Embedded Systems

Embedded system is very important economically because almost every electrical device now includes software. There are therefore many more embedded software systems than other types of software system. If one looks around the house, three or four personal computers may be observed but one could probably have 20 or 30 embedded systems, such as systems in phones, cookers, microwaves etc.

Responsiveness in real time is the critical difference between embedded systems and other software systems, such information systems, web-based systems, or personal software system, whose main purpose is data processing (Summerville 2000). For non-real-time systems, the correctness of a system can be define by specifying how system inputs map to corresponding output that should be produced by the system. In response to an input, a corresponding output should be generated by the system and, often, some data should be stored. For example, if one choose a create command in a patient information system; the correct system response is to create a new patient record in database, and to confirm that this has been done. Within reasonable limits, it does not matter how long this takes.

However, in a real-time system, the correctness depends both on the response to an input and the time taken to generate that response (Summerville 2000). If the system takes too long to respond, then the required response may be ineffective. For example, if embedded software controlling a car braking system is too slow, then an accident may occur because it is impossible to stop the car in time. Therefore, time is inherent in the definition of a real-time software system:

A reel-time software system is a system whose correct operation depends on both the results produced by the system and the time at which these results are produced. A 'soft real-time system' is a system whose operation is degraded if results are not produced according to the specified timing requirements. If results are not produced according to the timing specification in a 'hard real-time system, this is considered to be a system failure.

Timely response is an important factor in all embedded systems but not all embedded systems require a very fast response. For example, the insulin pump software is an embedded system. Although it needs to check the glucose level at periodic intervals it does not need to respond very quickly to external events. Figure 2.10 is a Real-Time Control Systems.

In addition to the need for real-time response, there are other important differences between real systems and other types of system:

1. Embedded systems generally run continuously and do not terminate. They start when the hardware is switched on and must execute until the hardware is switched off. This means that techniques for reliable software engineering, to ensure continuous operation. The real-time system may include update mechanisms that support dynamic reconfiguration so that the system can be updated while it is in service.
2. Interactions with the system's environment are uncontrollable and unpredictable. In interactive systems, the pace of the interaction is controlled by the system and, by limiting user options, the events to be processed are known in advance. By contrast, real-time embedded systems must be able to respond to unexpected events at any time. This leads to a design for real-time systems based on concurrency, with several processes executing in parallel.
3. There may be physical limitations that affect the design of a system. Examples of these include limitations on the power available to the system and on the physical space taken up by the hardware. These limitations may generate requirements for the embedded software, such as the need to conserve power and so prolong battery life. Size and weight limitation may mean that the software has to take over some hardware functions because of the need to limit the number of chips used in the system.
4. Direct hardware interaction may be necessary. In interactive systems and information systems, there is a layer of software (the device divers) that hides the hardware from the operating system. This is possible because you can only connect a few types of device to these systems such as keyboards, mice, displays, etc. by contrast, embedded system may have to interact with a wide range of hardware device that do not have separate drivers.
5. Issues of safety and reliability may dominate the system design. Many embedded systems control devices whose failure may have high human or economic costs.

Therefore, dependability is critical and the system design has to ensure safety-critical behavior at all times.

Embedded systems can be thought of as reactive systems: that is, they must react to events in their environment at the speed of the environment, 1989; Lee, 2002) response time are often governed by the laws of physics of rather than chosen for human convenience. This is in contrast to other types of software where the system controls the speed of the interaction.

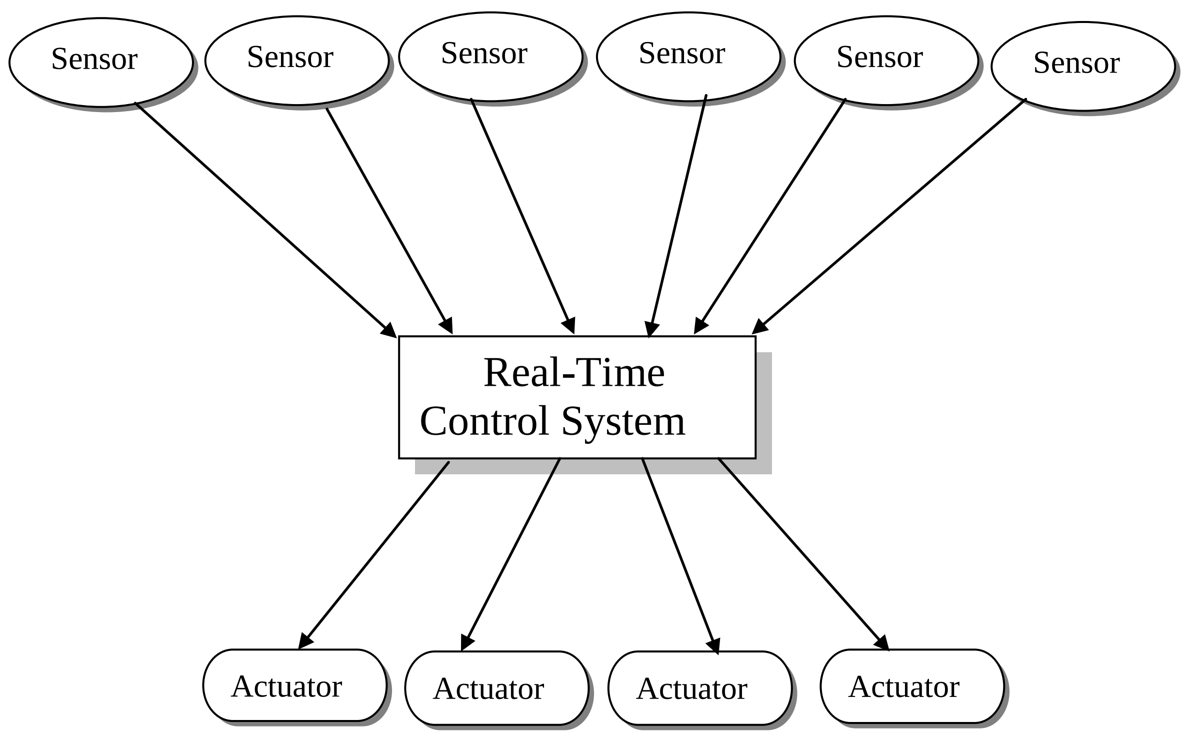



**Figure 2.10: Real-Time Control System**

**2.5 Control Engineering**

Control Engineering or Control systems engineering is the engineering discipline that applies control theory to design systems with predictable behaviours. Dorf (2004) states that control engineering is concerned with the analysis and design of goal-oriented systems. The practice uses sensors to measure the output performance of the device being controlled (often a vehicle) and those measurements can be used to give feedback to the input actuators that can make corrections toward desired performance. When a device is designed to perform without the need of human inputs for correction it is called automatic control such as cruise control for regulating a car's speed. Multi-disciplinary in nature, control systems engineering activities focus on implementation of control systems mainly derived by mathematical modeling of systems of a diverse range. Arnold. (2006)

Modern day control engineering (also called control systems engineering) is a relatively new field of study that gained a significant attention during 20th century with the advancement in technology. It can be broadly defined as practical application of control theory Christopher (2006). Control engineering has an essential role in a wide range of control systems, from simple household washing machines to high-performance F-16 fighter aircraft. It seeks to understand physical systems, using mathematical modeling, in terms of inputs, outputs and various components with different behaviors; use control systems design tools to develop controllers for those systems; and implement controllers in physical systems employing available technology. A system can be mechanical, electrical, fluid, chemical, financial and even biological, and the mathematical modeling, analysis and controller design uses control theory in one or many of the time, frequency and complex-s domains, depending on the nature of the design problem, Tan et al.(2001).

Process control is extensively used in industry and enables mass production of continuous processes such as oil refining, paper manufacturing, chemicals, power plants and many other industries. Process control enables automation, with which a small staff of operating personnel can operate a complex process from a central control's room. The basic objective of process control is to regulate the value of some quantity and maintain it at some desired value. This desired value is the set point, Curtis (2006)

For example, heating up a room is a process that has the specific, desired outcome to reach and maintain a defined temperature (e.g. 20°C), constant over time. Here, the temperature is the **controlled variable**. At the same time, it is the **input variable** since it is measured by a thermometer and used to decide whether to heat or not to heat. The desired temperature (20°C) is the **set point**. The state of the heater (e.g. the setting of the valve allowing hot water to flow through it) is called the **manipulated variable** since it is subject to control actions.

A commonly used control device called a programmable logic controller, or a PLC is used to read a set of digital and analog inputs, apply a set of logic statements, and generate a set of analog and digital outputs. Using the example in the previous paragraph, the room temperature would be an input to the PLC. The logical statements would compare the set point to the input temperature and determine whether more or less heating was necessary to keep the temperature constant. A PLC output would then either open or close the hot water valve, an incremental amount, depending on whether more or less hot water was needed. Larger more complex systems can be controlled by a Distributed Control System (DCS) or SCADA system.

## 2.6 Concept of Automatic Close Loop System

In its most general usage, automatic system can be defined as a technology concerned with carrying out a process by means of programmed command combined with the automatic feed back of data relating to the execution of those commands. The resulting system is capable of operating without human intervention. The development of this technology has become increasingly dependent on the use of computer and computer related technologies.

As consequences, automatic system have become sophisticated and complex Advanced System of this sort now represent a level of capability and performance that surpass in many ways the abilities of humans to accomplish the same activities. A closed loop control system uses a measurement of the output and feedback of this signal to compare it with the desired input Dorf (2004)

### 2.6.1 Principles and Theory of Automatic Close Loop System.

There are four building Block of automatic close loop feedback system; there are:

(1) A Source of power to perform some actions

(2) Feed back control

(3) Machine programming

(4) Decision making

**1. Source of Power to Perform Some Action**

An automated system is designed to accomplish some useful action, and that action invariably requires power. There are many source of power available, but the most commonly used energy in present-day automated system is electricity. Electrical energy is the most versatile because it can be generated from many sources e.g. fossil fuel, hydroelectricity, solar, and nuclear and to perform useful work and can be converted readily into several types of power e.g. mechanical, hydraulic, and pneumatic.

In addition electricity can be stored in high performance, long-wired batteries.

**2. Feed Back Control**

Feedback controls are widely used in modern automated system. Figure 2.7 is a typical diagram of feedback control systems. It consists of five basic components as shown below:

(a) Input

(b) Process being controlled

(c) Output

(d) Sensing element

(e) Controller and actuating devices.

- The five components of a feed back control system is represent

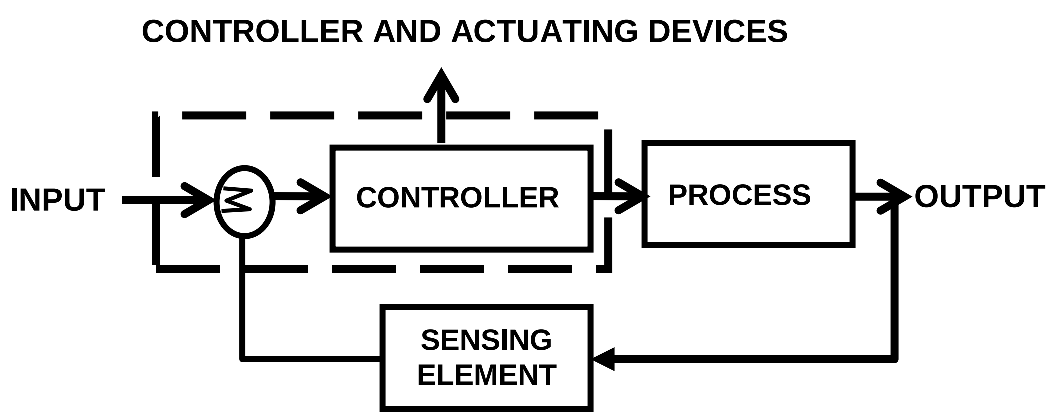



**Figure 2.11: Feedback Control System.**

- The input to the system is the reference value, or set point, for the system output. This represent the desired operating value output
- The output of the variable of the process that is being measured and compared to the input.
- The sensing element is the measuring devices used in the feedback loop to monitor the value of the output variable.

- The purpose of the controller and actuating devices in the feedback system is to compare the measured output value with the reference-input value and to reduce the difference between them. In general the controller and actuator of the system are the mechanism by which changes in the process are accomplished to influence the output variable.

## 2.7 Concept of SMS-Based Remote Monitoring and Control

The introduction of the Global System for Mobile Communication (GSM) and particularly the use of hand-held mobile phones brought the innovation of distance communication at remote location. The case of remote control capability and the possibility of achieving it at a reasonably low cost have motivated the need to use the GSM. GSM systems are becoming increasingly popular and have found application in almost all wireless applications because one can use it at any point and time. The capability of controlling industrial processes and many other numerous processes in a wireless and remote fashion has provided a great convenience to many people in life. The GSM can be used to monitor and control plants/field devices via short message service (SMS). SMS stands for Short Message Service. It is a mobile technology that allows for sending and receiving text or even binary messages to and from a Mobile phone (Hachenburg 1997). With an SMS based computer control system, monitoring and control can be achieved at all times. This is as a result of the ease of accessibility that comes with the use of a mobile phone. Therefore in this project the GSM has many advantages over other wireless systems. It is easy to install and covers a wide range. It can also monitor the signal strength and is more adaptable.

This system will be a powerful and flexible tool that will offer this service at any time, and from anywhere with the constraints of the technologies being applied. The proposed approach for designing this system is to implement a microcontroller-based control module that can receive instructions and commands from a cellular phone over the GSM network. The microcontroller then will carry out the issued commands and then communicate the status of a given parameter or device back to the Cellular phone.

## 2.8 Control System Network

There have been two major revolutions in process control and control system in the last 20 years. The first was the replacement of analogue controller with digital controller using embedded computer to perform the control function. The second revolution is the replacement of the standard 4 to 20mA analogue signals for communication throughout a process plant by serial communication via a network. In this recent past, a plant would typically house nearly all controllers in building room centrally located in the plant facility.

All measurement data taken in the plant (field) was communicated to the control over pair of wires carrying a 4 to 2mA signals with one pair for each sensor. They were connected to the controllers. The feedback control signals to the final control element were sent to the field in order pairs of wires carrying 4 to 20mA signals.

The 4 to 20mA range was an open standard that all manufacturers of controllers, sensors and final control element agreed upon.

A petrol chemical plant extends over several areas so that wire runs could be very long. In the early development of computers as controllers, data was still carried back and forth between the control room and field over the bus of 4 to 20mA wires pairs. With the present advancement in computer and computer technology, industrial process monitoring and control has moved to a very advanced level.

### 2.9 Direct Digital Control (DDC)

As computers have become more reliable and miniaturized, they have taken over the controller function. The operations of the controller have been replaced by Software in the computer.

The ADC and DAC provide interface with the process measurement and control action.

### 2.10 Industrial Control Systems

Industrial control system **(ICS)** is a general term that encompasses several types of control systems, including supervisory control and data acquisition (SCADA) systems, distributed control system configurations such as skid-mounted programmable logic controllers (PLC) often found in the industrial sectors and critical infrastructures.

ICSs are typically used in industries such as electrical, water, oil, gas and data. Based on information received from remote stations, automated or operator-driven supervisory commands can be pushed to remote station control local operations such as opening and closing valves and breakers, collecting data from sensor systems, and monitoring the local environment for alarm conditions.

**Programmable Logic Controller (PLC)** evolved, out of a need to replace racks of relays in ladder form. The latter were not particularly reliable, were difficult to rewrite, and were difficult to diagnose. PLC control tends to be used in very regular, high-speed binary controls, such as controlling a high-speed printing press. Originally, PLC equipment did not have remote 1/0 racks, and many couldn't even perform more than rudimentary analog controls.

**SCADA's** history is rooted in distribution applications, such as power, natural gas, and water pipelines, where there is a need to gather remote data through potentially unreliable or intermittent low-bandwidth/high-latency links SCADA systems use open-loop control with sites that are widely separated geographically. A SCADA system uses RTUs (remote terminal units, also referred to as remote telemetry units) to send supervisory data back to a control center. Most RTU system always did have some limited capacity to handle local controls while the master station is not available. However, over the years RTU systems have grown more and more capable of handing local controls.

The boundaries between these system definitions are blurring as time goes on. The technical limits that drove the designs of these various systems are no longer as much of an issue. Many Plc platforms can now perform quite well as a small DCS, using remote 1/0 and analog control loops, and are able to communicate supervisory data. It is not uncommon to have telecommunications infrastructure that is so responsive and reliable that some SCADA systems actually manage closed loop control over long distances. With the increasing speed of today's processors, many DCS products have a full line of PLC-like subsystems that weren't offered when they were initially developed.

This led to the concept of a PAC (programmable automation controller or process automation controller), that is an amalgamation of these three concepts. Time and the market will determine whether this can simplify some of the terminology and confusion that surrounds these concepts today.

### 2.11 Distributed Control System

DCSs are used to control industrial processes such as electric power generation, oil and gas refineries, water and wastewater treatment, and chemical, food, and automotive production. DCSs are integrated as a control architecture containing a supervisory level of control, overseeing multiple integrated sub-systems that are responsible for controlling the details of a localized process.

Product and process control are usually achieved by deploying feed back or feed forward control loops whereby key product and / or process conditions are automatically maintained around a desired set point. To accomplish the desired product and/or process tolerance round a specified set point, only specific programmable controllers are used.

## 2.12 A GENERALIZED VIEW OF THE PROCESS MODEL

The concept of representing process behavior with purely mathematical expressions is called modeling and the resulting form is called a model.

The most important model form in process Dynamic and control studies still remains the transfer function form. Ogunaike (1994) and Stephanopolous (1984) discussed some fundamental principles of how transfer functions are used for dynamic analysis.

**Types of models**

**1. State-space model**

The input, u (t), is a time-domain function, and the output, the time-domain function y(t) is generated by the process of solving differential or difference equations that model the process.(Ogunaike, 1994)

**2. Transfer-domain model**

The input is u(s), a function of the Laplace transform variables and output, y(s) (also a function of s), is generated by the process of multiplying u(s) by the function G(s) is also called as transfer function.(Dorf et al 2004)

**3. Frequency-response model**

The input, u ($j\omega$) is a function of the indicated complex frequency variable, and the input y ($j\omega$) (also a function of the same variable), is generated by the process of multiplying u(jw) by the function G($j\omega$).(Stephapolous, 1984)

**4. Impulse-response model**

The input, u (t) is a function of time, and the output, y(t) (also a function of time), is generated by a convolution operation of G(t) on u(t).(Ogunaike, 1994)

### 2.12.1 Concept of a Transfer Function (TF)

The underlying principles in each case discussed above may now be stated as follows. The function, by way of the indicated operation, "transfers" (or transforms) the input u(.) to the output y(.) as depicted in the equation :(Ogunaike,1994)

$$\mathbf{Y\ (.) = G\ (.)*u(.)} \qquad \textbf{(3)}$$

Where * represent the particular operation demanded by the specific model form and G(.) represent the function employed for the "transfer" of the input u(.) to the output y(._) via the indicated operation. For any model form, the function G(.) performs the noted task and is called the "transfer function" of such a model form if we include the effect of disturbances:

$$Y\ \mathbf{(.) = G\ (.)*u(.) + gd\ (.)*d(.)} \qquad \textbf{(4)}$$

The transfer function is usually represented by:

$$G(s) = K[Y(s)/x] \qquad (5)$$

Where,

G(s) = the general representation of a transfer function

Y(s) = Laplace transform of the output variable

X(s) = Laplace transform of the forcing function or input variable

### 2.12.2 Poles and zeros of a Transfer Function

With the exception of time-delay system, G(s) is generally a ratio of two polynomials in **s**, i.e.:

$$G(s) = N(s) / D(s) \qquad (6)$$

Where the numerator polynomial N(s) is of order r, and the denominator point D(s), is of order n. For real processes, we note that r is usually strictly less than n.

If these polynomials are factorized as follows:

$N(s) = k1\ (s – z1)(s-z2)\ \ldots\ldots\ldots\ldots\ldots\ldots\ (s – zr)$

$D(s) = k2\ (s – p1)(s-p2)\ \ldots\ldots\ldots\ldots\ldots\ldots\ (s – pn)$

hence

$G(s) = k\ [\ (s – z_1)\ (z – z_2)\ \ldots\ldots\ldots\ldots\ldots\ (s – z_r)] / [(s – p_1)(s – p_2)]\ldots\ldots(s-p_n)$

Where k = k1 /k2

Note

For s = $z_1$ or $z_2$ ,………………………………., or $z_r$, g(s) = 0

For s = $p_1$, or $p_2$…………………………or $p_n$, g(s) =infinity

The roots of the numerator polynomial, N(s), i.e,$z_1$,$z_2$ ……, $z_r$ are thus called the zeros of the transfer function, while $P_1$,$P_2$,………..$P_n$, the roots of the denominator polynomial, D(s), are called the transfer function poles.

Some important properties of transfer function are:

- In the TF of real physical system the highest power of s in the numerator is never higher than that of the denominator. In other words, $r \leq n$
- The TF relate the transforms of the deviation of the input and output variables from some initial steady state. Otherwise the nonzero initial conditions would contribute additional terms to the transform of the output variable.
- For stable systems the steady-state relationship between the change in output variable and the change in input variable can be obtained by: lim s $\rightarrow$ 0 [G(s)]

### 2.13 Expert system

An expert system is a software system that captures human expertise for supporting decision-making; this is useful for dealing with problems involving incomplete information or large amounts of complex knowledge. McCarthy (1984) described expert system as a computer system employing expert knowledge to attain high levels of performance in solving the problems within a specific domain area. Riley (1993) opined that Expert systems apply expertise to provide solutions for many complex systems in recent years. Expert system has been popular in most large and medium-sized organizations as a major tool for improving productivity and quality. Nedovic and Devedzii (2002); Nurminan et al (2003). An expert system is software that uses a knowledge base of human expertise for problem solving, or to clarify uncertainties where normally one or more human experts would need to be consulted. According to Giarattano (1993) Expert system apply expertise to provide solution for many complex systems in recent years. Expert systems are most common in a specific problem domain, and are a traditional application and/or subfield of artificial intelligence (AI). A wide variety of methods can be used to simulate the performance of the expert; however, common to most or all are:

1. The creation of a knowledge base which uses some knowledge representation structure to capture the knowledge of the Subject Matter Expert (SME);

2 A process of gathering that knowledge from the SME and codifying it according to the structure, which is called knowledge engineering; and

3. Once the system is developed, it is placed in the same real world problem solving situation as the human SME, typically as an aid to human workers or as a supplement to some information system. Expert systems may or may not have learning components.

Olabiyiosi et al (2004) in their paper titled web intelligence technologies: challenges and trends in new information age- web conference proceedings vol. 15, 2004 explained that an expert system is a type of application program that make decisions or solves problems in a particular field such as medicine, finance, geology etc. by using knowledge and analytical rules defined by experts in the field. Lydon (2003) defined ES as a "knowledge based system which imitates human experts usually by a very narrow way. He further illustrated that in an expert system there are two major components: a knowledge base and an inference Engine.

Efraim etal (2007) stated that expert systems can be classified in several ways – one way is by the general problem areas they address. For example diagnosis can be defined as inferring system malfunction from observations. Diagnosis is a general activity performed in medicine, computer studies and other fields.

According to Efraim , Expert system can adequately govern the overall behavior of a system. To do this the control system must repeatedly interpret the current situation, predict the future, and diagnose the cause

. Expert systems are particularly useful for on-line operations in the control field because they incorporate symbolic and rule-based knowledge that relate situation and action(s), and they also have the ability to explain and justify a line of reasoning Chiang et al., (2001).A common application of expert system technology in process control is for fault diagnosis. Typically, the basic components of an expert system include a knowledge base an inference engine and user interface. The knowledge base contains either shallow knowledge based on heuristics, or deep knowledge based on structural, behavioral or mathematical models. Chiang et al (2001). Various types of knowledge representation schemes can be used, including production rules, frames, and semantic networks Xia and Rao, (1999b).Since performance of the expert system is highly dependent on the correctness and completeness of the information stored in the knowledge base, updates to the knowledge base is necessary should the industrial process changes. The inference engine provides inference mechanisms to direct use of the knowledge, and the mechanisms typically include backward and forward chaining, hypothesis testing, heuristic search methods, and meta-rules Prasad et al., (1998); Norvilas et al.(2000);Rao et al.( 2000). Finally, the user interface translates user input into a computer understandable language and presents conclusions and explanations to the user. .Early applications of expert systems primarily focused on medical diagnosis Clancey and Shortliffe, (1984). Currently, expert systems have been adopted in many industrial applications, including equipment maintenance, diagnosis and control, plant safety, and other areas in engineering. For example, Srihari (1989) discussed a framework of knowledge-based system in industrial applications, using it for the tasks of diagnosis, supervision, and control; Xia and Rao (1999) built an expert system for operation support of pulp and paper manufacturing industries; Sun et al. (2000) and Uraikul et al. (2000) developed an expert system for optimizing natural gas pipeline network operations; Kritpiphat et al. (1998) implemented an expert system for intelligent monitoring and control of municipal water supply and distribution; Norvilas et al. (2000) developed an intelligent process monitoring and fault diagnosis environment by interfacing knowledge-based systems with multivariate statistical process monitoring techniques; Rao et al.(2000) developed an intelligent system for operation support for a boiler system and a chemical pulping process; Viharos and Monostori (2001) developed a hybrid system combining expert system and simulation for optimizing process chains and production planning; Wang et al. (1998, 2000) described the combination of expert system with neural networks for fault diagnosis of a transformer; and Prasad et al. (1998) applied the technology for constructing an operations support system for diagnosis and maintenance of a fluidized catalytic cracking unit and a paraxylene production unit. Although expert systems have been widely adopted for process control, some well-known limitations include the following Power and Bahri, (2004)

(i) Control over inference application is implicit in the structure of the knowledge base, e.g. in the ordering of rules for a rule-based system.

(ii) As the size of the knowledge base increases, the inference engine may be unable to identify the solutions in a timely fashion.

(iii) Most expert systems are domain specific and typically an expert system is only developed for an individual application.

(iv) Knowledge from experts is difficult to acquire and represent, and most often involves uncertainties.

To overcome the above limitations, a promising approach is the integration of expert systems with other solution approaches such as fuzzy logic, machine learning, and pattern recognition techniques, see for example (Rengaswamy and Venkatasubramanian, (1992); Venkatasubramanian,(1994); Power and Bahri, (2004). The uncertain knowledge can be handled by incorporating fuzzy logic into the knowledge representation.

### 2.13.1 Rule-Based Systems

Conventional problem-solving computer programs make use of well-structured algorithms, data structures, and crisp reasoning strategies to find solutions (Abraham, 2005). For the difficult problems with which expert systems are concerned, it may be more useful to employ heuristics: strategies that often lead to the correct solution, but that also sometimes fail. Conventional rule-based expert systems use human expert knowledge to solve real-world problems that normally would require human intelligence. Expert knowledge is often represented in the form of *rules* or as *data* within the computer. Depending upon the problem requirement, these rules and data can be recalled to solve problems. Rule-based expert systems have played an important role in modern intelligent systems and their application in strategic goal setting, planning, design, scheduling, fault monitoring and diagnosis and so on. With the technological advances made in the last decade,

Abraham, (2005).stated further that conventional computer programs perform tasks using a decision-making logic containing very little knowledge other than the basic algorithm for solving that specific problem. The basic knowledge is often embedded as part of the programming code, so that as the knowledge changes, the program has to be rebuilt. Knowledge-based expert systems collect the small fragments of human know-how into a knowledge base, which is used to reason through a problem, using the knowledge that is appropriate. An important advantage here is that within the domain of the knowledge base, a different problem can be solved using the same program without reprogramming efforts. Moreover, expert systems could explain the reasoning process and handle levels of confidence and uncertainty, which conventional algorithms do not handle Giarratano and Riley, (1989).

# CHAPTER THREE

# RESEARCH METHODOLOGY AND SYSTEM ANALYSIS

## 3.1 Research Methodology

The description of methods used to realize the goal of applying neural networks in design of a monitoring and control system is given in this chapter. Furthermore, the justification for the use of these methods is also presented.

The logic for embedded systems could be described by different methods. Embedded system is unique because it is a hardware-software co-design problem. The hardware and software must be designed together to make sure that the implementation not only functions properly but also meets performance, cost, and reliability goals. A number of methodologies that pertains to these work are hereby reviewed, followed by where they were used in the dissertation:

- Rapid application development (RAD)
- Waterfall methodology
- Top-down design methodology
- Level oriented design
- Data Flow-Oriented Design
- Data Structure-Oriented Design
- Object-Oriented Design
- Preferred Design methodology

### Rapid Application development

Rapid application development (RAD) is a development methodology, which involves iterative development and the construction of prototypes (Wolf, 2010). Figure 3.1 shows a design methodology for a combined hardware/software project. It begins with Front-end activities such as requirements and specification followed by architecture which decomposes the functionality into major components. The hardware and software aspects are considered simultaneously. Similarly at the back-end are integration and testing of the entire system. In the middle, however, development of hardware and software components can go on relatively independently—while testing of one will require stubs of the other, most of the hardware and software work can proceed relatively independently. This approach was used in this work

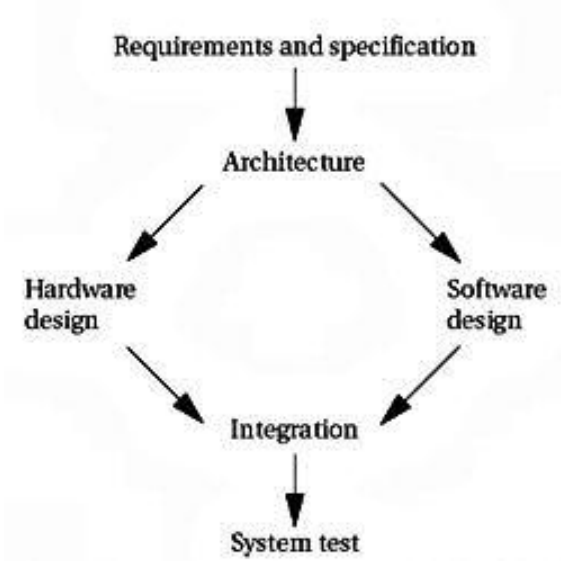



**Figure 3.1: Design Methodologies for a Combined Hardware/Software Project. [Source: Wolf, 2010]**

**Waterfall Development**

The Waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, determines the basic characteristics of the system architecture design decomposes the functionality into major components; implementation, testing (validation), integration, and maintenance (www.embededd.com/design/prototyping&development). The waterfall model gets its name from largely one way flow of work and information from higher level of abstraction to more detailed design steps. The waterfall model has the following methodology:

Requirement Definition

System and software design

Implementation and unit testing

Integration and system testing

Operation and Maintenance

**Level-Oriented Design**

In the level-oriented design approach, there are two general or broad strategies that can be used. The first strategy starts with a general definition of a solution to the problem then through a step-by-step process produce a detailed solution (this is called Stepwise Refinement). This is basically dependent on the system requirements and is a top-down process. The other strategy is to start with a basic solution to the problem and through a process of modeling the problem, build up or extend the solution by adding additional features (this is called design by composition).

The top-down process starts at the top level and by functional decomposition, breaks down the system into smaller functional modules (userpages.umbc.edu/~khoo/survey1.htm). Smaller modules are more readily analyzed, easier to design and code. But, inherent in the top-down process is the requirement that there must be a complete understanding of the problem or system at hand. Otherwise, it could lead to extensive redesign later on. The top-down process also is dependent on decisions made at the early stages to determine the design structure. Different decisions made at the early stage will result in different design structures. Functional decomposition is an iterative "breakdown" process called stepwise refinement, where each level is decomposed to a more detailed lower level. Thus, at each decomposition, there have to be a way to determine if further decomposition is needed or necessary, that is, if the atomic level has been achieved. There are no inherent procedures or guidelines for this. There is also a possibility of duplication if stepwise refinement is not done carefully or "correctly"; this will occur toward the end of the process, that is, at the lower levels. This can be costly, especially if there are many different designers or programming teams working on a single system. As a result, the top-down process is often used in the initial phase of the design process to break down the different components or modules of a system. The top-down process has also been used as a preliminary step in the other design methodologies. Once the modules of the system have been determined, they can be divided amongst the different designers or design teams.

The design by composition strategy involves the evolution of a solution by building upon the solution from the previous stage. Using this technique, additional features are added as the solution evolves. This strategy uses as its origin, the basic or simple initial solution and through an iterative composition process add or expand the solution to include additional modules. This approach will also encompass the bottom-up design, where the lowest level solution is developed first and gradually builds up to the highest level. Freeman (1983) has added a few models, such as the outside-in model where what the end-users sees (external functions of the system) are defined as the top-level decisions and the implementation (the inside of the system) as the lower-level decisions. This model was created to overcome the tendency of designers to pay insufficient attention to the needs of end-users. The alternative to the outside-in model is the inside-out model, where decisions relating to the implementation (inside) of the system are made before the external function of the system. Another model is based on the most-critical-component-first approach, where one first design the components of the systems that are the most constrained so that these critical parameters are satisfied. Then the rest of the system components are designed. Often these models are conceptual, not to be rigorously enforced because in a real design effort, integration of models is often necessary.

**Data Flow-Oriented Design**

In the data flow-oriented design approach, which is often called Structured Design, information flow characteristic is used to derive program structure (userpages.umbc.edu/~khoo/survey2.htm). In the data flow-oriented approach, emphasis is on the processing or operations performed on the data. Design is information driven. Information maybe represented as a continuous flow that is transformed, as it is processed from node to node in the input-output stream. As software can ideally be represented by a data flow diagram (DFD), a design model that uses a DFD can theoretically be applied in the software development project. The data flow-oriented approach is especially applicable when information is processed without hierarchical structure. A DFD can be mapped into the design structure by two means - transform analysis or transaction analysis. Transform analysis is applied when the data flow in the input-output stream has clear boundaries. The DFD is mapped into a structure that allocates control to three basic modules - input, process and output. Transaction analysis is applied when a single information item causes flow to branch along one of many paths. The DFD is mapped to a substructure that acquires and evaluates a transaction; another substructure controls all the data processing actions based on a transaction. A few examples of structured design or data flow-oriented design methodologies are Structured Analysis and Design Technique (SADT), Systematic Activity Modeling Method (SAMM) and Structured Design (SD).

**Data Structure-Oriented Design**

The data structure-oriented design approach utilizes the data structures of the input data, internal data (for example databases) and output data to develop software (userpages.umbc.edu/~khoo/survey2.htm). In the data structure-oriented approach, the emphasis is on the object, which is the data. The structure of information, called data structure, has an important impact on the complexity and efficiency of algorithms designed to process information.

As both data flow and data structure oriented design approaches are based on considerations in the information domain, there are similarities between both approaches. Both depend on the analysis step to build the foundation for later steps. Both attempt to transform information into a software structure; both are driven by information. In data structure-oriented design information structure are represented using hierarchical diagrams; DFD has little relevance; transformation and transaction flows are not considered. Data structure-oriented design have a few tasks - evaluate the characteristics of the data structure, represent the data in its lowest form such as repetition, sequence or selection, map the data representation into a control hierarchy for software, refine the control hierarchy and then develop a procedural description of the software. Some examples of the data structure-oriented design approach are the Jackson System Development (JSD) and the Data Structured Systems Development (DDSD) which is also called the Warnier-Orr methodology.

**Object-Oriented Design**

Traditional approaches to the design of software have been either *data oriented* or *process oriented*. (userpages.umbc.edu/-know/supreme/html) Data-oriented methodologies emphasize the representation of information and the relationships between the parts of the whole. The actions which operate on the data are of less significance. On the other hand, process-oriented design methodologies emphasize the actions performed by a software artifact; the data are of lesser importance.

It is now commonly held that *object-oriented* methodologies are more effective for managing the complexity which arises in the design of large and complex software artifacts than either data-oriented or process-oriented methodologies (userpages.umbc.edu/-know/supreme/html). This is because data and processes are given equal importance. *Objects* are used to combine data with the procedures that operate on that data. The object oriented design approach is unique in its usage of the three software design concepts: abstraction, information hiding and modularity. Objects are basically a producer or consumers of information or an information item. The object consists of a private data structure and related operations that may transform the data structure. Operations contain procedural and control constructs that may be invoked by a message, that is, a request to the object to perform one of its operations. The object also has an interface where messages are passed to specify what operation on the object is desired. The object that receives a message will then determine how the requested operation is to be performed. By this means, information hiding (that is, the details of implementation are hidden from all the elements outside the object) is achieved. Also objects and their operations are inherently modular, that is, software elements (data and process) are grouped together with a well-defined interface mechanism (that is, messages). Object oriented design is based on the concepts of: objects and attributes, classes and members, wholes and parts. All objects encapsulate data (the attribute values that define the data), other objects (composite objects can be defined), constants (set values), and other related information. Encapsulation means that all of this information is packaged into a single name and can be re-used. The object oriented design is rather new and as such it is still evolving even at this present moment. Object oriented design encompasses data design, architectural design and procedural design. By identifying classes and objects, data abstractions are created; by coupling operations to data, modules are specified and a structure for the software is established..

### 3.2 Preferred Design methodology

SMS Based Industrial Process Monitoring and Control using Artificial Neural network is a reactive system, it should be based on stimuli response model. Such a system involves a combination of both software and hardware components. In this work the use of a microcontroller based design with ANN control software will be used. It will adopt the combination of top-down, bottom-up and Rapid application development (RAD) approaches and methodology which will come under five major approaches namely:

i. Analysis and Design
ii. Selection of appropriate tools

iii. Development(prototype)

iv. Refining

v. Testing

Given the complex nature of the oil/gas plant, a hardware configuration must be chosen that will support the number of subunits and a software design must be created to efficiently make use of that hardware.

Figure 3.2 shows a design methodology for this work. Front-end activities such as specification and architecture simultaneously consider hardware and software aspects. Similarly, back-end activities namely integration and testing consider the entire system. In the middle, however, development of hardware and software components can go on relatively independently—while testing of one will require stubs of the other. The implementation phase itself is a complete flow from specification through testing. The system in view comprises both software and hardware. In the software approach the Artificial neural Network was programmed using Mat lab programming Language. The control system was also programmed with assembly language, C language and Visual Basic. The hardware is made up of the control unit, the input interface and the output interface. There are nine units in the gas supply Plant and each unit is controlled by a microcontroller. The nine units are again linked to a personal computer for the purpose of communicating with personnel and getting a feedback when any system parameter defies control processes

A GSM module will be designed that will be interfaced with the Microcontroller through a personal computer and a voltage converter.

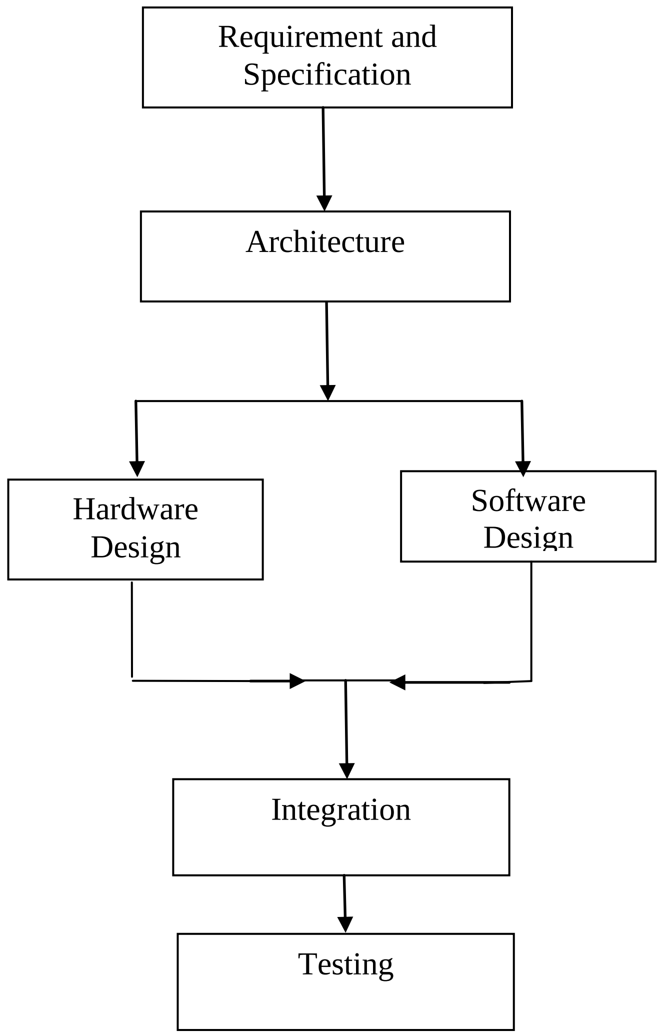



**Figure 3.2: Flow Diagram Of The Design Methodology For The Project**

### 3.3 Analysis of the Current System

Several visitations were made to Shell Petroleum Development Company for the purpose of studying current system and gathering information. A schematic representation of a typical crude oil processing facility is shown in figure 3.3. The hydrocarbon from the well head is routed to the central production facility which gathers and separates the produced fluids (oil, gas and water). The production facility processes the hydrocarbon fluids and separates oil, gas and water. The oil must usually be free of dissolved gas before export. Similarly, the gas must be stabilized and free of liquids and unwanted components such as hydrogen sulphide and carbon dioxide. Any water produced is treated before disposal. Appendices D, 1 to 9 are the pictures of the different sections of SHELL plant.

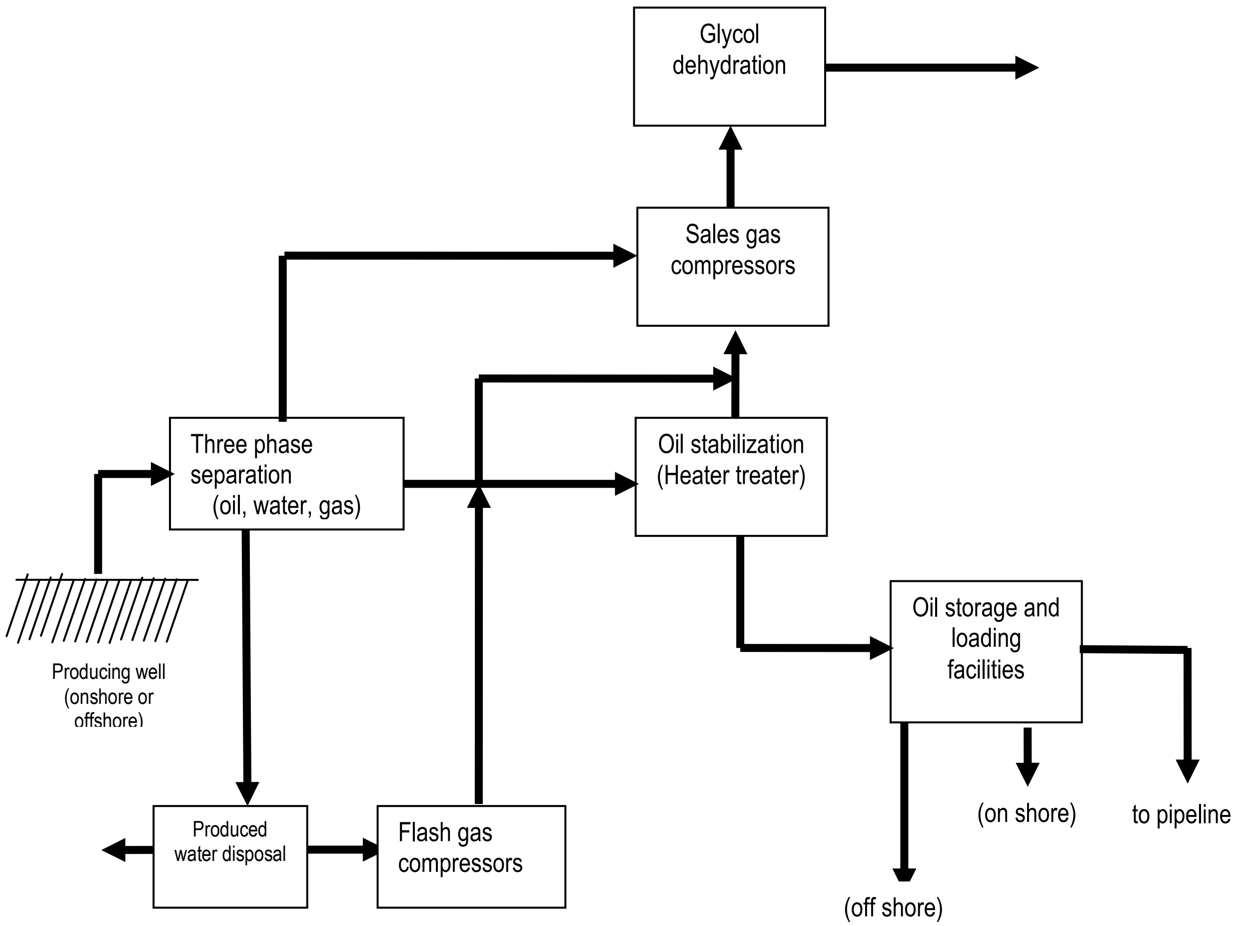



**Figure 3.3: A Schematic Representation of a Typical Crude Oil Processing Facility**

Figures 3.4a and 3.4b are block diagrams that show the process control systems in shell. Figure 3.4a is an element of Distributed Control Systems while figure 3.4b is control mechanism. The LAN connects the computer in the plant together. Through the LAN the respective reporting agents code named here as Engineer 1 with their computers can receive information.

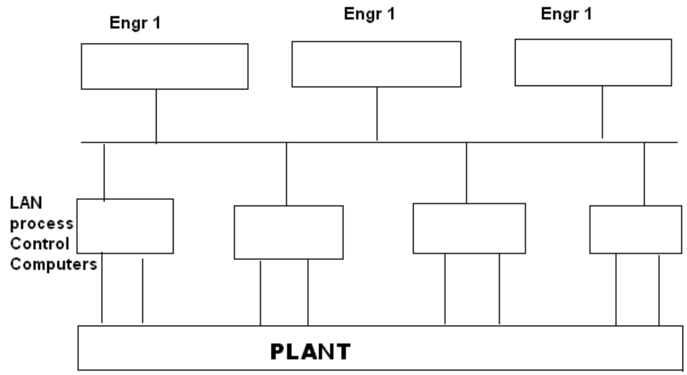



**Figure 3.4a: Block Diagram of LAN in Shell**

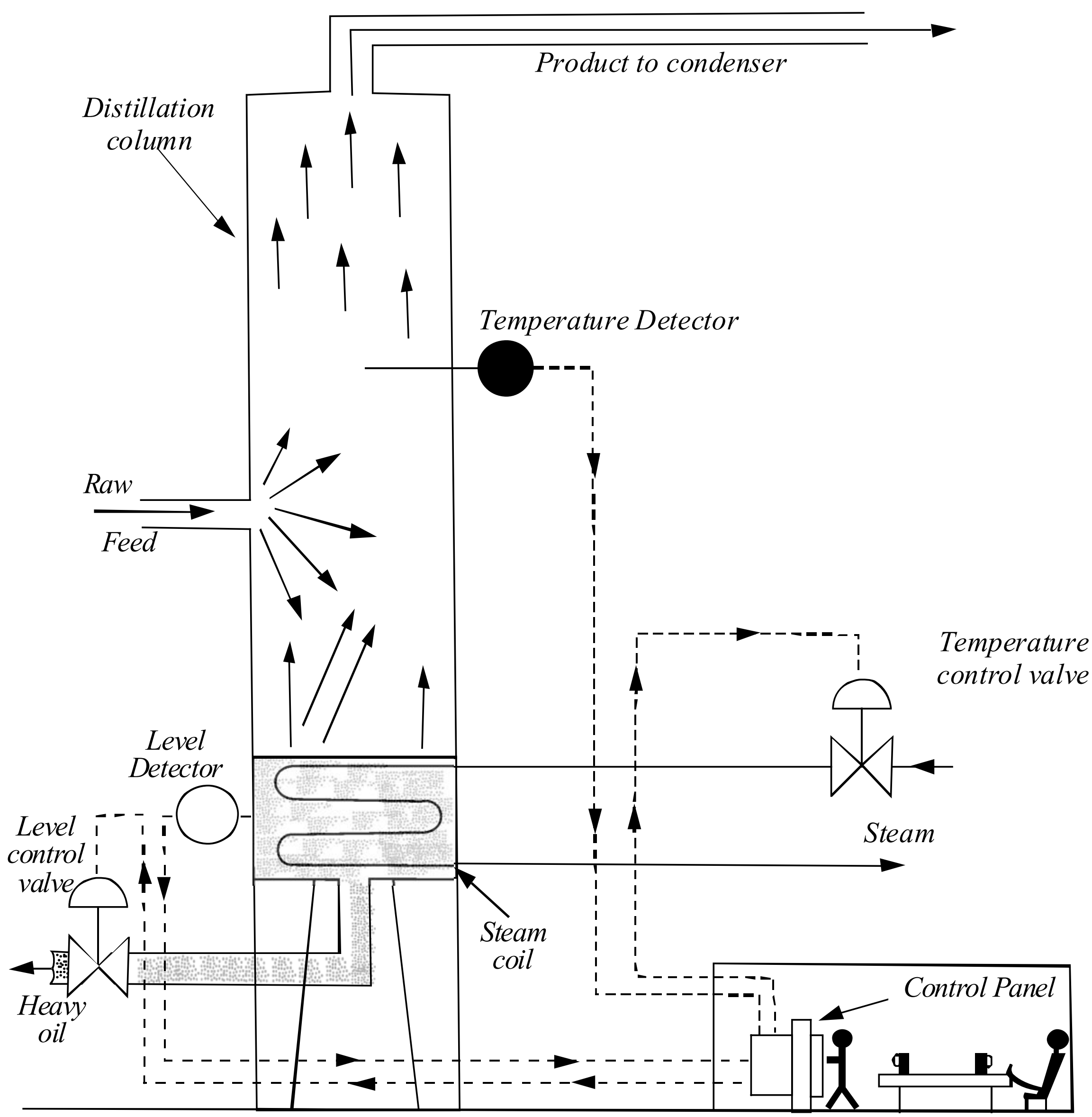



**Figure 3.4b: Diagram of Control Mechanism in Shell.**

### 3.3.1. System Investigation:

The system investigation is concerned with the careful study of the current system to know its strength and weakness.(Adams and Wagner, 1986). The shortcomings of the existing system are taken into consideration and the areas that prove unsatisfactory to the needs of the user are worked upon in order to design a new system that will meet the user's needs. System investigation is an in-depth and comprehensive study carried out upon an existing system so as to come up with relevant facts that will be of great help in the designing of the new system. The purpose of the study is to understand the existing system and to identify the basic information requirement. System investigation involves the task of fact-findings and analysis of the facts.

The investigation was carried out in three parts:

1 Investigation of existing system.

2 Data capturing/collection

3 Analysis of data collected/captured.

### 3.3.2 Objective of the Investigation

The following objectives were established for efficient investigations of the system:

I. To determine the operations under the current system and how they are preformed, in relation to old system that was in use.

II. To determine the strength and weakness of the current system.

III. To know the cost effectiveness of implementing a new system.

### 3.3.3 Limitation of Current System.

Wang .and Tan (2006) stated that condition monitoring systems for natural gas pipeline network is either handled in a manual or semi automatic manner. The field Monitoring and control of remote facilities initially meant a crew (or crews) spending countless hours driving from one remote facility to the next, often on a full-time basis. When they saw a problem at a facility, they fixed it, which often meant making an adjustment to a piece of equipment such as a pump or a valve. The first leap forward was equipping these crews with radios to call in to headquarters when a problem was located, or to call for additional crews when needed. The first steps in automating this process were simple monitoring and alarm systems. These were typically electromechanical devices at remote sites that would send a signal back to a control center via radio or leased telephone lines.

In this method as shown in Figure 3.4b, a plant would typically house nearly all controllers in a building room centrally located in the plant facility. All measurement data taken in the plant (field) was communicated to the control room over pair of wires carrying a 4 to 20mA signals with one pair for each sensor. They were connected to the controllers. The feedback control signals to the final control element were sent back to the field in other pairs of wires carrying 4 to 20mA signals. A petrol chemical plant extends over several areas so that wire runs could be very long. This is a centralized method and that is the present system in SHELL with addition of SCADA system using local area networks. The limitation of the present system in SHELL includes:

i. Control of industrial process is done in one location.
ii. The operator has to do most of the work.
iii. The operator being a human being is prone to make mistakes.
iv. It poses a security risk especially when it is in a hazardous area.
v. A lot of energy is spent in effecting control.
vi. Time is wasted moving from one point to the other to monitor operation.
vii. Many of the abnormal conditions may not be monitored in real time
viii. It lowers productivity.
ix. Down time is greatly increased

### 3.3.4 Control Process and Objective in SHELL

The overall objective of the process control philosophy of Shell is the control of Sales gas flow rate from the three non associated gas trains from individual well sites and also the associated Gas from Soku flow station to a specified range of the Daily Contract Quota (DCQ). This is accomplished by modulating the quantity of well fluid raised from Non Associated Gas wells. Shell currently has 18 wells located at Soku, Nembe Creek, Ekulama and Belema flow station.

The three major controls are:

- ✓ Level Control
- ✓ Temperature Control
- ✓ Pressure Control

Pressure control of the Inlet Separator forms an integral part of the overall Gas Plant control system. Distributed Control System means that the control system of a plant (Refinery, gas plant) is split into small units, which are distributed around the plant.

Figure 3.5 is a typical example of Distributed Control System in shell.

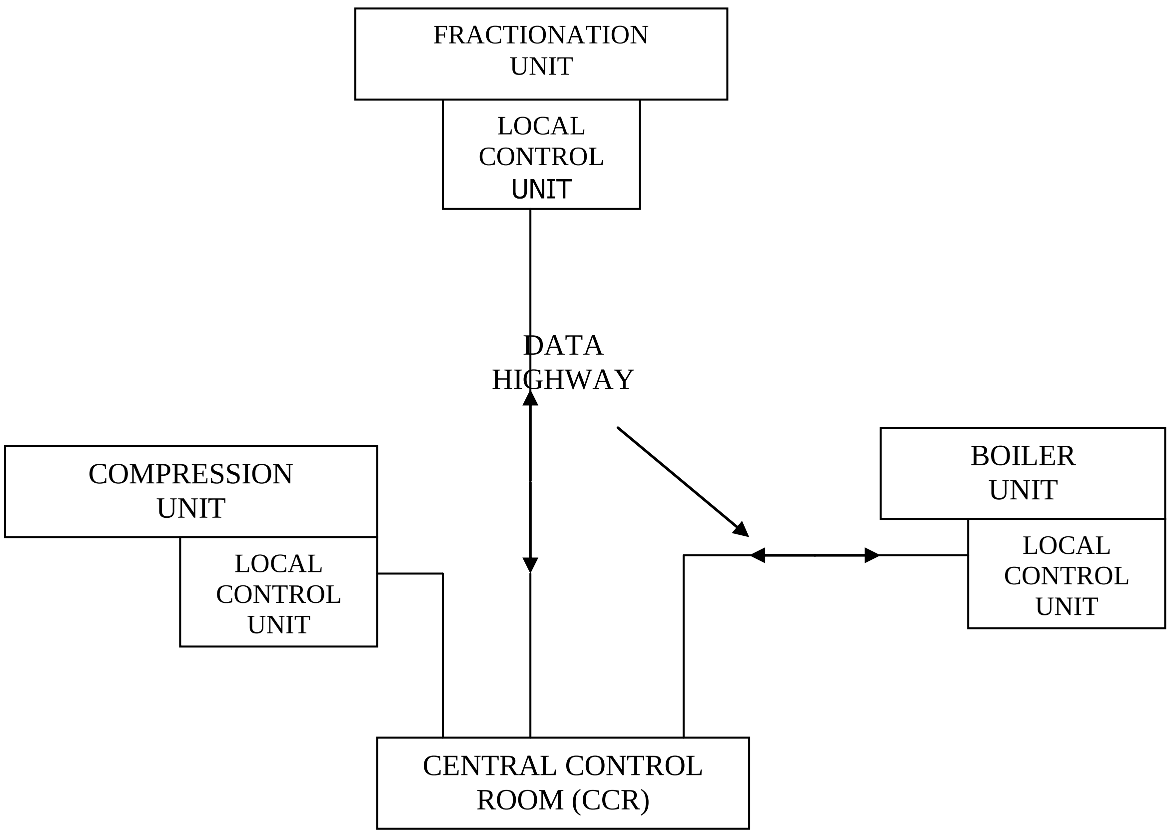



**Figure 3.5: A simple Distributed Control System**

From the simple DCS system, the plant consists of three separate (distributed) Local Control units:

- ✓ Fractionation
- ✓ Compression and
- ✓ Boiler

The loops for each unit are controlled by a local control unit. The information required by the operator is sent by a simple cable (data highway) to the Central Control Room (CCR). Here the information is shown at the workstation Visual/Video Display Unit (VDU). From the workstation, the operator can adjust set points Motor Operated Valves (MOV) etc. using the same data highway.

The DCS has five level concepts:

| | |
|---|---|
| LEVEL 5 | Group Manager workstation VDU |
| LEVEL 4 | Area Manager |
| LEVEL 3 | Supervision |
| LEVEL 2 | Control |
| LEVEL 1 | Field Services |

The Shell Petroleum Development Company (SPDC) of Nigeria has a major plant known as SOKU GAS PLANT.

The overall system control philosophy includes the specific control requirement of SOKU Gas Plant for the Non-Associated Gas (NAG) from the individual well sites and also for the Associated Gas (AG) from the SOKU Flow-Station when the Nembe Greek and Ekulama pipeline are in place. This research/design is essential since it will re-engineer the whole processes by the use of Neural Network to monitor and control the Oil/Gas process. It will in turn send report to the necessary agents using a SMS-based system. This will ensure:

1) Stable plant operation
2) Minimum shutdown times
3) Automatic control of operations faults and remedies.

### 3.4 Information Gathering

The nature of this research does not require much sampling from any population; it is technical and therefore requires on the spot visitation of the industry being used as case study.

Initial search for materials has been conducted from SHELL, Internet books, journals professional magazines, files, papers, literatures on the topic and related topics, relatively published and unpublished materials.

Shell Development Company of Nigeria was visited several times to observe operations and interact with staff of Control department.

Detailed study of their operations and interviews with those who are versed in the subject area was conducted. For this reason only those in control engineering / ICT professionals at the company and elsewhere were consulted and used. The collection of data for the research work was multi-dimensional, involving a wide range of research instruments. These instruments include:

i. Questionnaires surveys
ii. Personal Interviews
iii. Examination (Statistical reviews) and
iv. Personal Observation

The primary data was collected from shell through:

- Observation
- Personal interviews and
- Study of materials

**Interview Method**: This is a person to person communication with a basic decision making purpose. In other words, this is a method of data collection that involves the obtaining of data by conversation. This method helps in obtaining unseen information or data. The researcher tactfully asked questions depending on the objectives stated.

**Examination Method**: Some Relevant documents were downloaded from the internet. Some other documents were presented to the researcher which includes: Mutilated documents, research statistics and manuals and operational guidelines provided by the staff of the company though not official because of security reasons.

**Other materials** downloaded by the researcher include:

(i) IDC's white papers and research reports on Control system
(ii) Articles and professional opinion on control and monitoring system.
(iii) Forrester's research publications on Control and monitoring system, remote control, internet/web, rule based Expert system etc.
(iv) Cisco's Networking on ISP Materials

(a) **Observation Method**: this is another method used by the researcher during the investigation. The researcher visited the shell gas plant in Oyigbo three times to observe the process of Gas/Oil exploration from the oil wells to the exportion point at the liquefied natural Gas in Bonny.

Data is based on the fact that it is hoped that they will disclose and identify all the problems and solutions associated with SMS-based monitoring and control systems in Oil/Gas industries.

### 3.5 Data Analysis

Data analysis of the system was made on the information gathered for Proper analysis and critical examination of them for correctness and completeness. The data structure was examined and analyzed to ensure that there is no data conflict and that data are logically sound to produce the desired or expected results. The data structure was also examined to make sure that the information entered conformed to the objective and storing the data together with the acquired knowledge in a form that could be used in database.

### 3.6 Overview of Envisaged System

The envisaged system should have facilities to sense the condition of the process plant, analyze the conditions of the plant and process variables and act on the system with the aim of achieving set goals or as the case may be report to remote agents. Figure 3.6a is the envisaged diagram of such a system while figure 3.6b is the block diagram. It consists of several hardware components designed around a central microprocessor. These hardware components include input and output interfaces interfaced with the controller component. The input and output interfaces link the environment being monitored, the process plant and the reporting agent. The software is made of the artificial neural network developed using assembly and C languages. The control system is centered on the AT89C51 microcontroller. The output interface is made up of LCD and a provision or interfacing to a collating PC via a MAX 232 and GSM module.

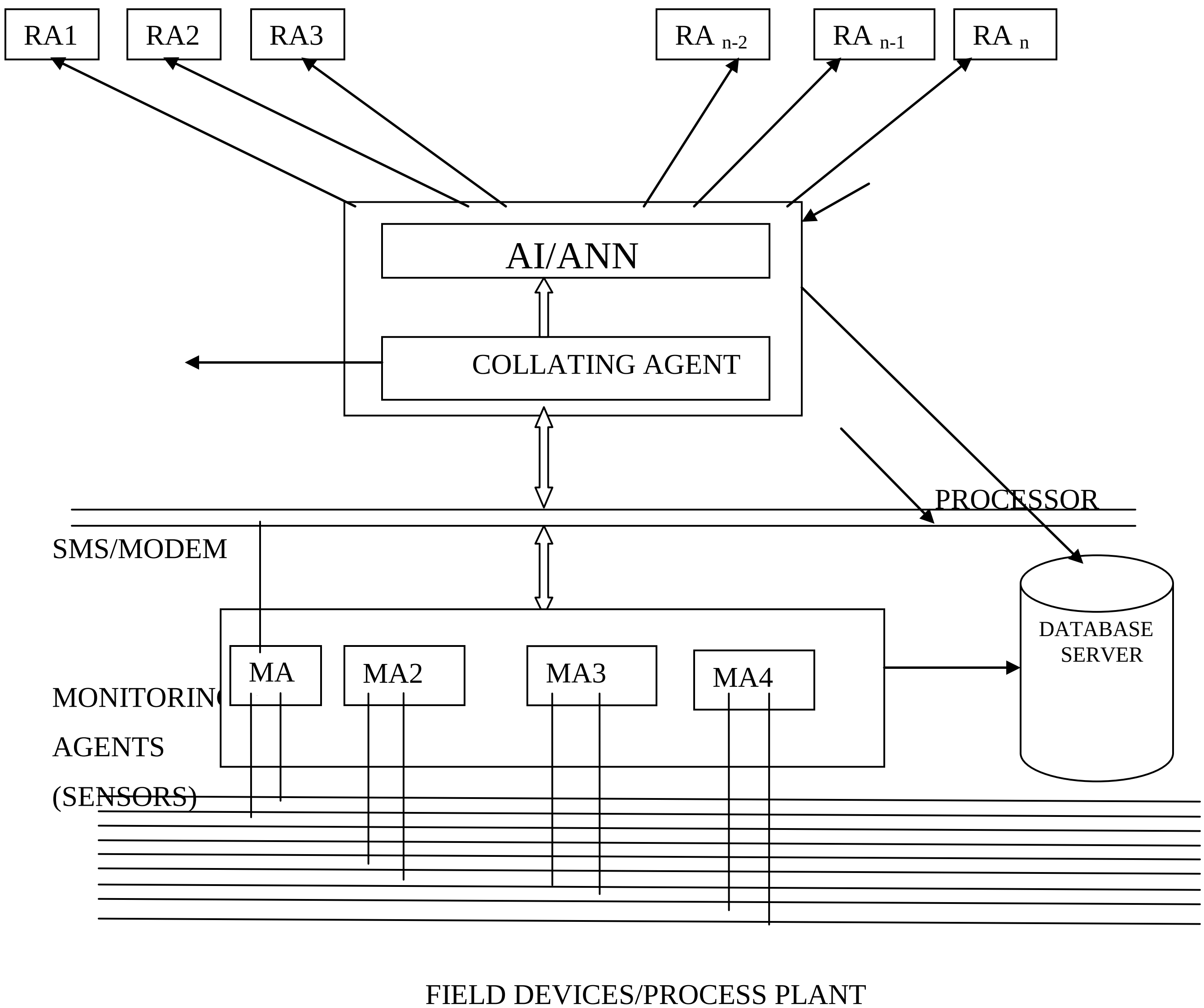



**Figure 3.6a: Overview of SMS-Based Remote Process Monitoring and Control Using ANN**

**KEY:**

**RA =Reporting Agent**

**MA = Monitoring Agent**

**AI = Artificial Intelligence**

**ANN = Artificial Neural Networks**

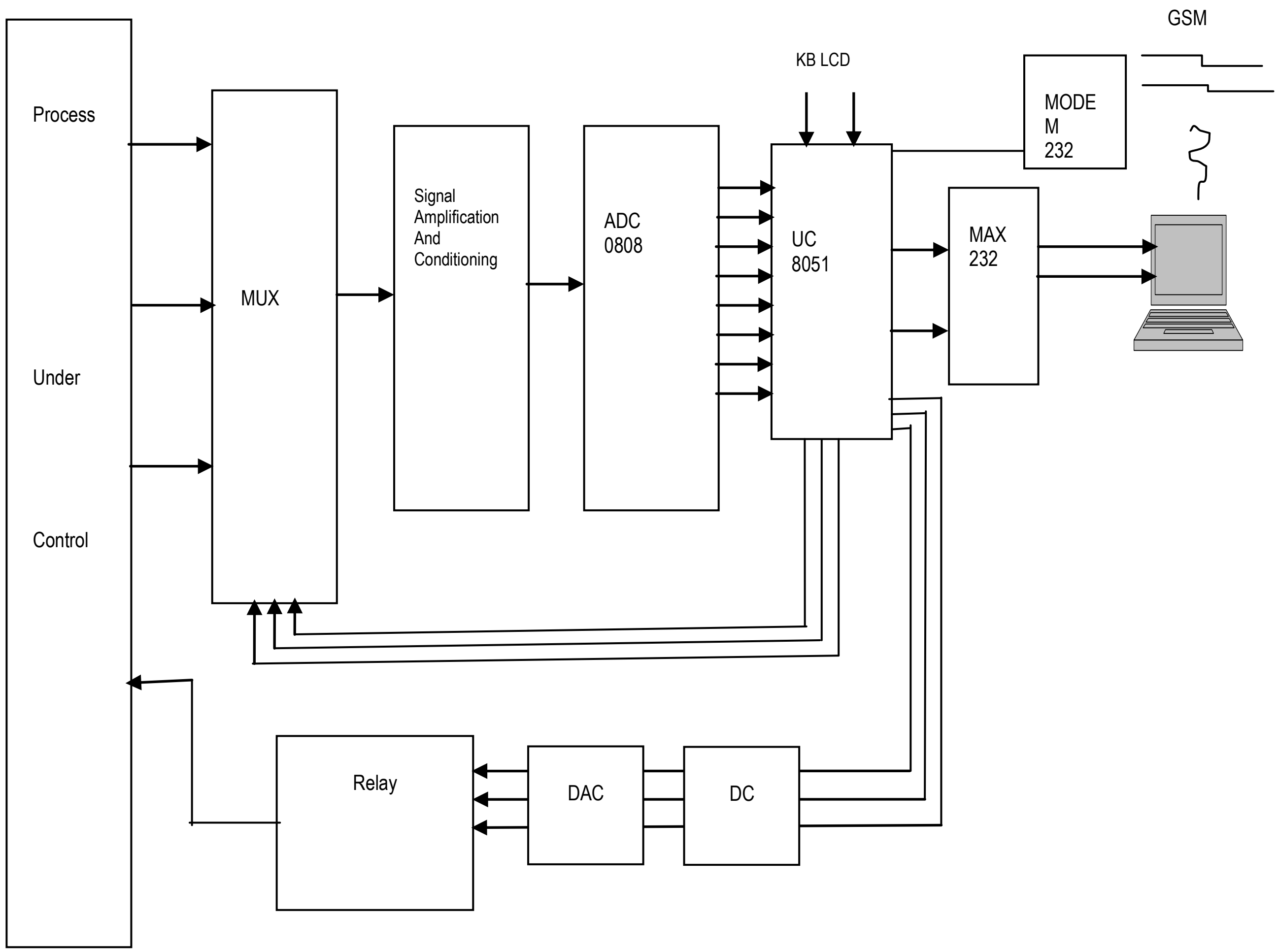



**Figure 3.6b: Block diagram of SMS-Based Remote Process Monitoring and Control Using ANN**

# CHAPTER FOUR

# SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. With the advent of computer based tools design of embedded systems has taken a new dimension. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

## 4.1 **Requirement Specification**

This is the process of writing down the user and system requirements. The user requirements of a system should include the functional and non functional requirements. System requirements are expanded versions of the user requirement that are used by engineers as starting point for the system design. The requirements specifications of this system involve:

- Real-Time Control of 9 separate Units that make up the Gas Plant Facilities

- In each unit, Liquid level, Pressure level, Flow rate and Temperature are the four parameters that may need to be controlled depending on the exact unit.
- Some unit require that only three of the four listed parameters need to be controlled, while some require the control of only two.

- It is desirable to meet the timing requirements of the application while using controllers optimally.

- Thus the 4, 3 or 2 parameters that need to be controlled at a unit may be handled by only one microcontroller. Thus, only 9 microcontroller should be used, one per unit (of 4, 3 or 2 parameter) in the Gas Plant facility of 9 units.

- This multiprocessor system must be interfaced to the a central PC both for the accumulation and maintenance of a control database (Comprised of fault diagnostics, actions taken, commands received and obeyed and system status information with time stamps), and for man-machine interaction between any of the nine units and the maintenance crew as desired.
- The central PC is interfaced to the Internet/GSM network to facilitate direct and urgent communication of plant status to the particular personnel needed to take remedial actions as and when due
- The Microprocessor Based Gas Plant Control System must do everything possible to eliminate completely or reduce to the barest minimum, unplanned plant shutdown due to fault

The nine units where the control of level, pressure, temperature and flow rate, or subsets of these are needed are:-

1. Well Stream Heater
2. Test Separator
3. Inlet Separator
4. Gas Cooler
5. Propane Refrigerator
6. Condensate Processing Unit
7. Gas Compression
8. Condensate Export Pump
9. TEG Storage and Transfer.

**4.2 System Specification**

System specification is statement that clearly states what the device or product is to be and do as well as the design criteria. The systems monitor and control the process variable in the oil and Gas Process pl.ant

### 4.2.1 Functionality of Industrial Automation System

According to Akpado (2011), modern industrial automation systems should be capable of conducting real-time online data acquisition and manipulation, recentralized system resources management, and networked data sharing. It must have a flexible configuration capability. It should be capable of flexible setting up of general local area network (LAN) capability and wide area network (WAN) capability in order to meet specific industrial measurement and control requirements. It should also be able to build comprehensive monitoring network, integrating various functions such as data collection, condition monitoring, fault diagnosis, resource management, and decision-making. Such an industrial automation system should be suitable for operation and management at different levels such as workshop, branch factory, and corporation. The basic requirements for an industrial automation system are listed as follows:

- It should be able to effectively conduct the desired measurement and control tasks in order to ensure the proper operation of the industrial process. By uninterrupted system monitoring and recording, the database stores gathered information on plant operation status. These data can be used latter on the further analysis and diagnosis of plant conditions.
- It should be able to effectively utilize various signal processing techniques to analyze the gathered data from different measurement points (channels). Moreover, appropriate and effective data processing algorithms need to be incorporated into the industrial automation software so as to fully exploit the merits of computational resources provided by modern computers as well as satisfy real-time constraints on data manipulation. By doing so, real-time measurement and through data analysis can be effectively accomplished.
- It should be able to increase the software versatility by allowing for flexible configuration of a variety of system parameters. The principles and main functions of industrial automation software may remain unchanged for different industrial applications. However, the details for any specific application can be redefined by modifying the configuration database according to any specific user requirement. Finally, by combining configuration database with the fixed system modules, system configuration for the specific application is accomplished and thus the industrial automation software with desired functionality is built.

- Human-machine interfaces should be designed according to the current popular development trends. User-friendly graphical user interfaces (GUIs) are always beneficial to improve software quality because they make user operations more convenient and pleasant. For instance, using the multimedia provided by the modern computer technology, all of the plant statuses can be displayed in an animated form as their corresponding industrial parameters are updated in real time.
- It should have comprehensive alarming and reporting capability. The alarm module in the industrial automation software compares the gathered data with the user-set parameters. Audiovisual alarms and exception reports are generated for immediate remedial action if the data levels detected exceed the preset parameters. The alarming function should be able to provide various alarming patterns in order to promptly inform the corresponding technical and management personnel of the presence of emergency situations. These flexible alarming modes include vivid screen indicator, speaker, automatic telephone dialing, beeper, e-mail, fax, and so on. E-mails can be sent to the cell phones of corresponding people in the form of SMS message, informing them of plant emergencies in a timely manner, which cuts machine downtime. All of these functions can be made available without needing extra prohibitive telemetry investments.
- It should be able to directly perform various measurement and control tasks using commonly used web browsers. Previously, the special purpose industrial automation software package had to be installed on the industrial computer before hand in order to conduct the tasks. The networked system provides the network server, which allows the user to accomplish industrial measurement and control from anywhere through the network web browser (for example, Internet Explorer of Netscape). It avoids the installation of any special-purpose software. Thus, software maintenance becomes more convenient, and such systems should be more economically priced. Using the network technologies, an industrial automation system is no longer an "island of automation" that only confines to stand-alone, local or dedicated network. The remote management activity allows operators anywhere to access the real-time data from the factory floor. Internet-enabled industrial automation system also allow for automatic software upgrades and remote maintenance. However, the challenges of handling security issues for such internet-enable systems should be well considered in adopting it.

In line with the above recommendations and requirement for a process control system that would meet the challenges for optimum productivity, the software and hardware specifications of our system are shown in Table 4.1 and 4.2. Table 4.1 is the set point values of the nine units.

**Table 4.1 Set point values**

| | **Temp** | **Press** | **Level** | **Flow** |
|---|---|---|---|---|
| Well Steam Heater | 150 | 90 | 850 | NA |
| Production and Test M | 45 | 105 | NA | NA |
| Inlet separator | 45 | 102.8 | NA | NA |
| Liquid Cooler | 20 | 50 | 65 | NA |
| Gas Compression | 35 | 98 | NA | 65.9 |
| Gas Cooler | 25 | 50 | NA | NA |
| Condensation Stabilization | 23 | 15 | A | NA |
| Dehydration Tag | 45 | 102 | NA | NA |
| Pump | 42 | 52 | NA | NA |

**[Source SPDC GP manual)**

### 4.2.3 Brief Description of the Design Architecture.

This design is microcontroller-based interfaced with the ANN and Rule-based expert system. The ANN is for the classification of the different values of the process variables while the rule based expert systems contain rules in the knowledge base. The rules are used to generate questions for the user and to provide recommendation using the knowledge and interference engine. One of the reasons for the development of expert system is its potential to provide knowledge and advice to large number of users. The expert is implemented as knowledge servers. Expert systems running on a PC/server can support a large group of users and communicate with the system over the GSM network.

### 4.2.4 Database.

A database is provided for the management of information, such as calibration data setting, the parameter for calibration and calibration status. The parameters are read and compensation calculation for the measured values are made in order words necessary adjustments are made to control the system. It also stores maintenance records, diagnosis records, and steps taken to remedy the faults as well as related information. By monitoring the message history, the system can continuously manage and control operation and remedy future repairs. In addition a database for temperature, pressure, level and flow rate is created.

### 4.2.5 Further Specifications

Before designing the architecture for SMS Based Remote Monitoring and Control (SBRMC) the requirements should be specified. The major task in the requirement specification is to identify and resolve trade-offs between goals and constraints of the system that are conflicting or not completely achievable.

Important requirements concerning SBRMC are as follows:

- Tasks, which require a deterministic timing regime, should be avoided because they may not be achievable due to GSM-related traffic delay.
- It is necessary to minimize the communication load between the GSM network level and the existing control level.
- Direct access to a controller (for example a PLC) is not a requirement and is probably not desirable because of security reasons. Information exchange between Process plants and GSM-based clients can be achieved through corporate systems – such as relational databases or real-time databases, instead of control units.
- Reliability and the possible system failures should be considered in order to keep the system robustness from the system architecture point of view.
- The different requirements will lead to different control structures having its own advantages and disadvantages

### 4.2.6: Hardware Subsystem Specification:

Table 4.2 is the hardware subsystem specification of the envisaged system. It comprises of the components shown in this table.

**Table 4.2 Main Features of the Hardware System**

| ITEM | CONTENT |
|---|---|
| Controller IC | Microcontroller |
| Interface IC | Bus Expander |
| Power Supply | 5-10V D.C/220V Ac |
| Dedicated Input Lines | 8 |
| Dedicated Output Lines | 16 |
| Input/Output Lines | 36 |
| Control Lines | 17 |
| Input Mode | Digital (3-5V) |
| Output Mode | Digital (3-5v) |
| Display Type | LCD |
| Text Format | Alphanumeric |
| Operating Temperature | 60-70$^0$ C |
| Power Extension | Yes, 3 |

### 4.3 The Gas Processing Plant Facilities.

It is important in a dissertation like this before detailed design issues are commenced to first of all discuss the steps involved in gas processing and the facilities required.

The gas plant takes in associated gas from a field reservoir and flow stations and processes it to produce dry gas for the LNG plant and stabilized (gas free) condensate. This is shipped to Soku flow station. The essential features of a gas supply plant are as shown in figure 4.1, and are explained hereunder.

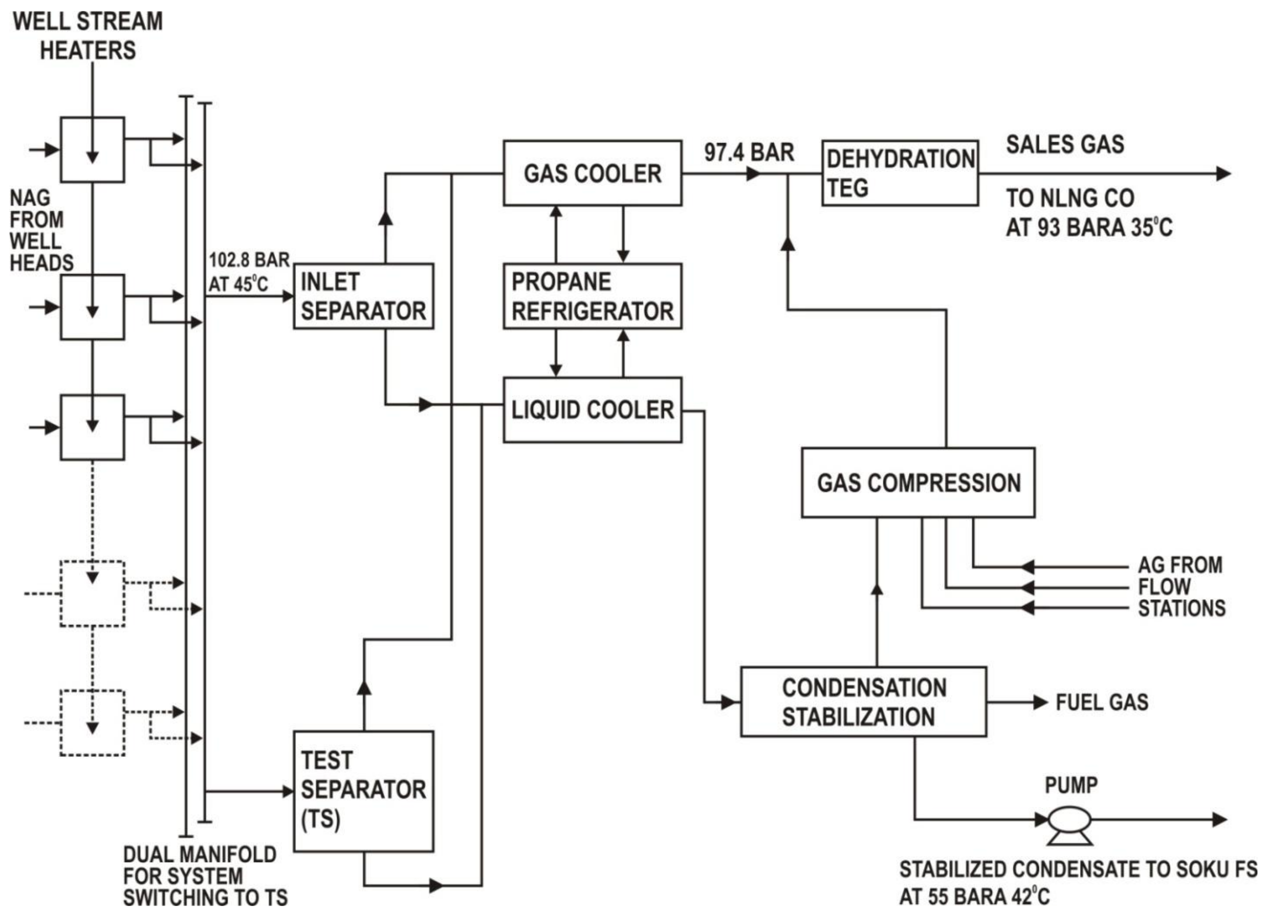



**Fig 4.1: Block Diagram of SPDC Gas Supply Plant.**

**a. Well stream heaters**

These heats up the incoming flow stream using hot water produced by a control boiler unit.

The purpose is to ensure that the gas/liquid mixture of the stream do not form hydrates as they pass through the inlet pressure reducing valves as shown in figure 4.1

The hydrates are solid in form and are a compound of the hydrocarbon gas and they are produced when the well stream pressure and temperature fall too low.

**b. Dual inlet manifold**

The regulated inlet incoming stream are connected to a dual inlet manifold so that using stop valves, any well stream can be passed to text separator so that the liquid/ gas ratio can be checked and composition analyzed.

The expected operating pressure and temperature of the flow line and of the inlet manifold to the operating inlet separator is 102.8 bar and $45^{o}$c.

**c. Inlet separator**

This vessel has sufficient volume to allow the well to separate into its liquid and gas components so that they can be separately treated to produce a saleable product. (Figure 4.1) Here temperature, pressure and level is important. Gas and oil are separated. The gas is up and oil is down.

**d. Test Separator**

The inlet heater manifold is of the dual type and the flow lines from the well head are valved to allow for each to be switched to the test separator (Figure 4.1). The test separator is added as the compositions of the flow stream vary considerably and will change with time. Regular testing is essential to ensure good gas quality.

e. **Propane Refrigeration**

In this subsystem, hot propane gas returning from both the gas and liquid cooler is scrubbed to remove contamination and passes via a heat exchanger to a lobe type compressor which increases the pressure to raise the liquefying temperature. The pressurized propane then passes through a propane/lube oil separator which removes the compressor lubricant and on to a fin fan cooler then condenses the propane

**f. Condensate processing.**

The produced liquids from the inlet separator are successively reduced in pressure until the atmosphere pressure is reached. This ensures that the fuel produced liquid (condensate) is stabilized and safe for pumping to the Soku flow station.

The pump increases the pressure to about 55bar at $42^{o}$C and thus ensures that there is no dangerous vapor emitted when the condensate reaches the flow station. The quality of the export condensate is controlled using analyzers and if it is not up to standard the system recycles until it is.

**g. Gas compression.**

A three stage centrifugal compressor driven by a gas turbine is used to increase the pressure of the gas produced during condensate stabilization to a level equal to the NAG from the gas cooler (around 98ba:;)

The Gas Compression also has an input from three locally situated flow stations which produce gas from other fields.

These flow stations are SOKU, EKULAMA AND NEMBE CREEK.

The input pressure to the compressor is around 6 bar and the output pressure around 98 bar which gives a stage compression ratio of 2:5. The estimated flow of gas from the stabilization process and flow station is 65.9mmsefd.

**h. Condensate Export Pump**

The condensate export pumps are high capacity/high heat pumps. The condensate export booster and export pumps take suction from either the condensate surge drum or the condensate storage tank. The pumps are provided with pressure relieve valves located downstream of the pump relieving of the suction of the export pump. They are set at a design pressure and temperature of 55bar and $42^{o}$C respectively.

**i. Dyhedration Teg and Transfer**

The Teg tanks are vented to atmosphere. The Teg transfer pumps are local manual stop/start with low low liquid level operation shut down three (OSD 3)pump trip,

In industrial gas process, we have:

i. Normal operating condition

ii. Alarm condition

iii. Trip condition

The process control system provides:

i. Centralized control and monitor of facilities from the control room.

ii. Remote control of valves, motors and process equipment.

iii. Monitor Emergency Shut Down & blow down valve station (open, close). Color of indicator shall be red in an abnormal situation.

iv. Electric drivers status (run, stop) (green indicate operation

v. process measurements

- All alarm condition
- All start & stop switches for operation of equipment
- All open & close switched for operation of valves

Open loop response: In open loop no controllers are present.

Close loop response: In closed loop response, feedback controllers are present

**4.4 The Proposed Design Process**

The design methodologies used for the computer-based control systems are not appropriate for the SMS-based control systems, as they do not consider the GSM environment issues such as time delay caused by the call traffic, concurrent user access, GSM-based interface, and SMS-related safety. For example, an SMS-based control system has uncertainty about who the users are, how many users there will be, and where they are. In contrast for a typical distributed control system (DCS) the system load has been determined from conception. SMS-based process control systems have a variable working load. Few of the existing implementations in SMS- based process controls discuss the limitation caused by the GSM environment features such as GSM transmission latency and user isolation. Actually, GSM time delay and multiple users' collaboration are two essential issues, which must be addressed in the design of SMS-based control system. The objective of establishing SMS-based process control systems is to enhance rather than replace computer-based process control systems by adding an extra GSM-level in the hierarchy Yang et al (2003).

When designing the architecture for Remote SMS Based Monitoring Control system it would be interesting to bring modularity into the design by using layers as entities that can perform different tasks. So each layer can perform its own jobs and communicate closely with others to make the whole system work seamlessly. Different layers have different specializations, and are responsible for different tasks. Each layer acts independently Yeung and Huang (2001). Figure 4.2 shows the development process of the system. The process began with requirement specification. The requirement is that the product should function properly by being able to control industrial variables and bring them to desired points. The specifications are defined. The components are analysed and the requirement further modified. The proposed system is designed using co-design approach involving hardware and software, The design system is simulated using a simulation tool called proteus ISIS simulation software,. The simulated work is tested for correctness using test data. If the result is correct and meet the specification the aim is achieved. If not the design step is fine toned again.

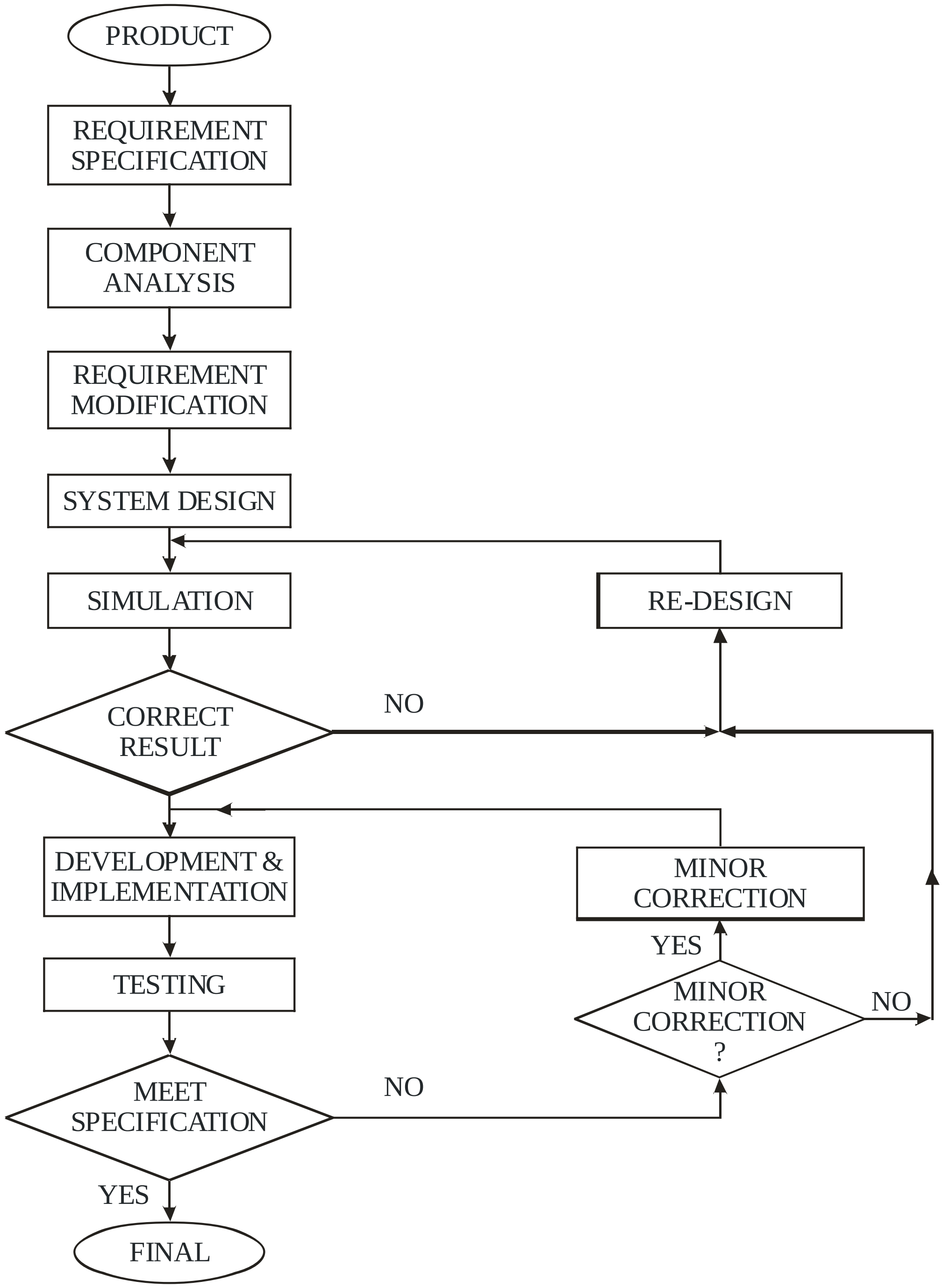



**Figure 4.2: The Development Process**

### 4.4 Steps in Designing the Control Systems.

Figure 4.4 shows the steps involved in developing the control system. Firstly establish the control goal, Identify the variables. In this case the variables are temperature, pressure, level and flow rate. Write the specifications. Establish the system configuration and identify the actuators. Obtain a model for the process, the actuator and sensor, Describe the controller and the parameter to be adjusted.

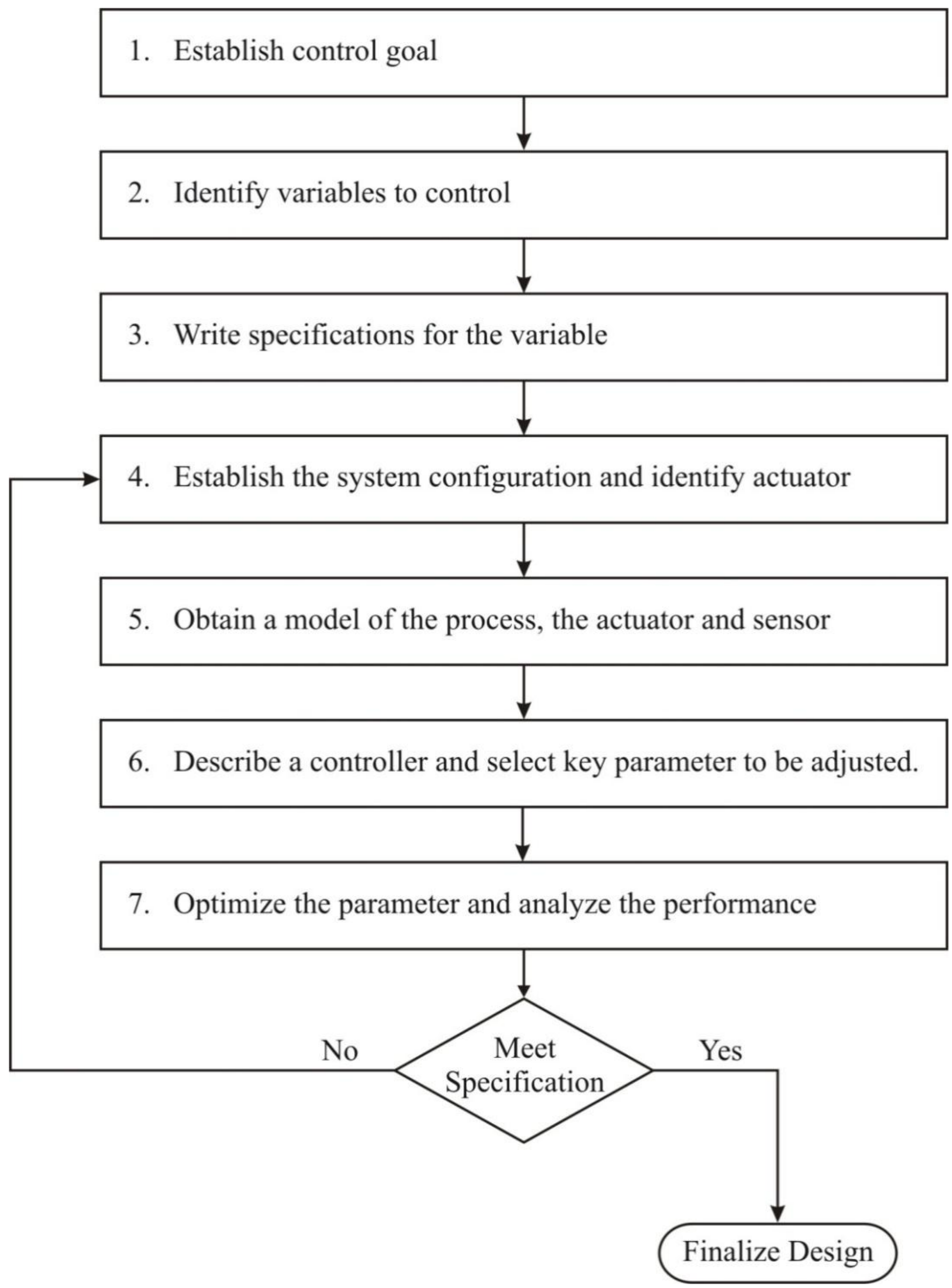



**Figure 4.3: Steps in Designing Control Systems.**

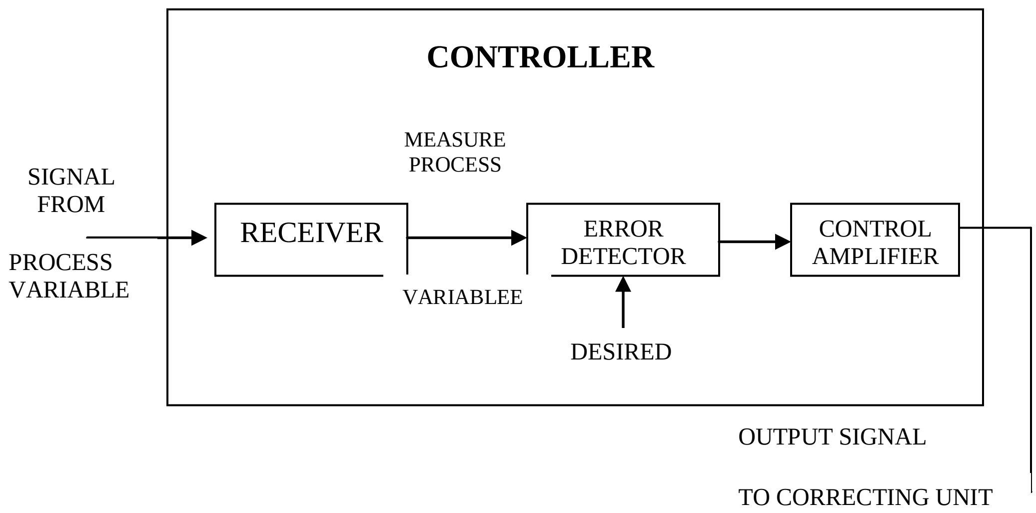



**Fig 4.4: The Controller Block Diagram**

The block diagram in figure 4.4 shows the main parts of a process loop controller:

- **Receiver**

  The receiver converts the signal from the process variable (flow, pressure, etc.) into a signal which is suitable for the controller operating system

- **Error Detector**

  The error Detector detects (finds) any difference between the measured process variable (measured value) and the set point (desired value).

- **Control Amplifier**

  This unit adjusts the output signal to the correcting unit (final control element e.g. control valve).the correcting unit corrects the error until the error signal is reduced to zero. When there is no error the control amplifier keeps the correcting unit at a fixed position. The operators can switch the system to manual and adjust the output signal by hand.

- **Controller Functions**

  There are up to three ways to adjust the controller. The older systems can be adjusted by a screwdriver. The new systems can be adjusted by changing the computer program.

The adjustments are:

- Proportional band (gain). This controls how much the error signal is amplified.
- Integral (reset). This is adjusted to cancel the final error which may be left after proportional action has finished.
- Derivative (rate). This is only used on slow moving loops (for example, temperature). It gives the system a quick start when an error occurs.

### 4.5 The Hardware Subsystem Design.

The hardware subsystem of the Gas/Oil Process Control System has data acquisition systems that convert the analog signals from various sensors to digital values that can be read in and processed by the microcontroller. The acquired data of any of the variables is selected, amplified, and converted into a form required by the microcontroller. Attached to the microcontroller are keyboard and the display unit which allow the user to enter set- point values, to read the current values of processed variables and to issue commands. Relays solenoids, D/A converters and other actuators are used to control the process variable under program command.

#### 4.5.1 System Block Diagram

The aim of this work is to develop an SMS based system that will remotely monitor and control industrial process using ANN**.** Such a system must be equipped with DAS to monitor the condition of the plant, control the variables and in exceptional case report to remote personnel. The block diagram in figure 4.5 shows the design concept. It has a number of hardware module designed around a standard microcontroller towards achieving this goal. The modules include the input and output interface modules linking the plant being monitored and the controller modules which include the artificial neural network that does the classification and pattern matching.

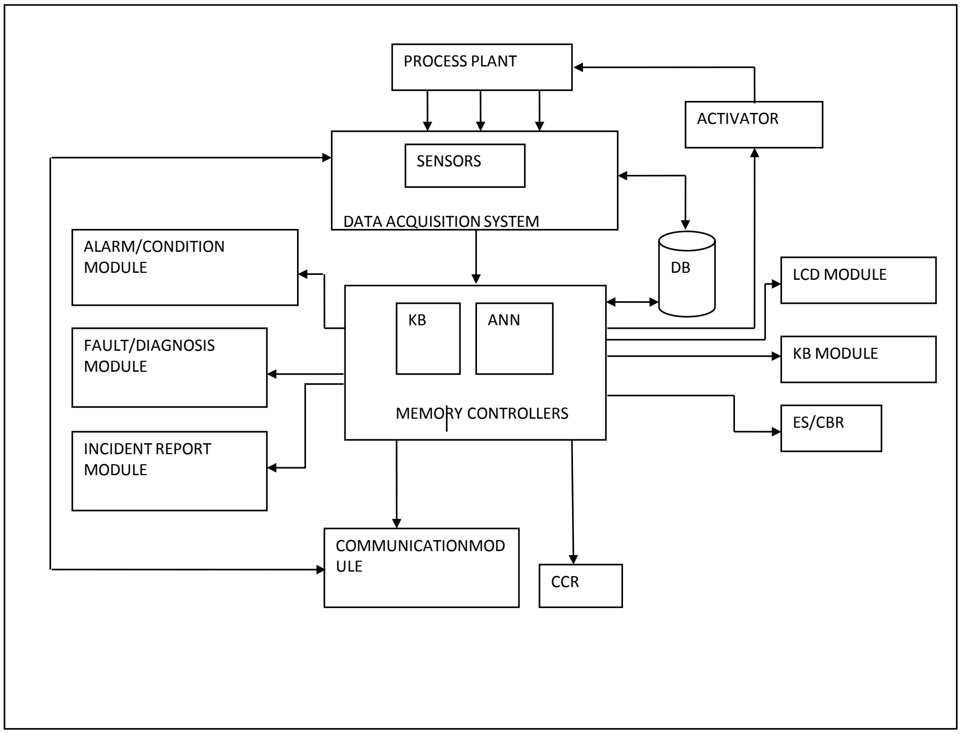



**Figure 4.5: The block diagram of the design concept of the system [Author]**

## 4.6 The Architecture of the Control System.

This system is meant to monitor and control industrial process of oil and gas industries using artificial neural networks. Therefore the hardware subsystem comprises of device that monitor the process plant in real time, process the variable data and effect a change in the plant to achieve a set point. The main block diagram of the design architecture of SMS based Remote industrial process monitoring and control using artificial neural network is shown in figure 4.6 This architecture is applied to the 9 units that make up the Gas/oil plant process but the specific unit is mentioned in the box labeled 'process under control' in figure 4.6 As a result of the complex nature of the process plant of oil and gas industries the entire plant was divided into nine sub-units. Nine dedicated processors handling each of the different sub-unit were used. Each of them is handling four, three or two variables. In all there are twenty seven inputs (variables) to handle. An input is read and classified by the neural network.

The block diagrams of these sub-units by the researcher are shown from figure 4.7a to figure 4.7i.

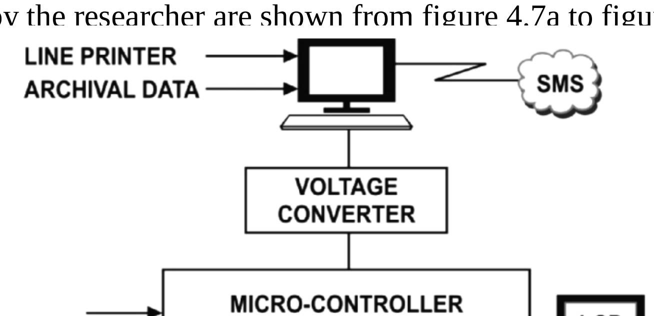


**Figure 4.6: The Architecture of a Unit Control System[Author]**

## 4.7 Hardware Configuration of the block Diagram

The Block diagram of the control system for each of the nine units comprises of the following:

i. Micro controller
ii. The data I/O
iii. Multiplexer and Demultiplexer
iv. Signal amplification and conditioning sub-unit
v. Voltage converter
vi. PC
vii. LCD and keyboard
viii. SMS Interface.
Ix .OP AMP

The system contains an 8 channels single ended analog signal multiplexer. A particular input channel is selected by using the address decoder. There are three input lines from the process namely. Temperature, Pressure and Level:

The input states for the address line is from 000 – 111 as shown in Table 4.3

**Table 4.3: Address Line of Multiplexer**

| **SELECTED ADDRESS CHANNEL** | **ADDRESS LINES** | | |
|---|---|---|---|
| | **C** | **B** | **A** |
| **IN0** | **L** | **L** | **L** |
| **IN1** | **L** | **L** | **H** |
| **IN2** | **L** | **H** | **L** |
| **IN3** | **L** | **H** | **H** |
| **IN4** | **H** | **L** | **L** |
| **IN5** | **H** | **L** | **H** |
| **IN6** | **H** | **H** | **L** |
| **IN7** | **H** | **H** | **H** |

The address is latched into the decoder on the low to high transition of the address latch enable signal. The variable signal selected is passed through the signal amplification and conditioning device for conditioning and amplification. The signal is passed through an analogue to digital converter which converts the continuous signal to discrete digital number.The signal is read by the micro controller through its input port.

The micro controller consist of 128byte of RAM, 4k byte of on-chip ROM, 2 timers, one serial port, 64 Kbytes external code memory space, 210 addressable location and four ports (each 8 bit wide).

The micro controller is programmed with C, Visual Basic and assembly Language. This software acts as industrial supervisory and control system supervising the different processes. The micro controller reads in and processes the data. An LCD and keyboard attached to the micro controller allows the user to view the measured values of process variables and to issue commands when necessary.

Process variables are controlled under program direction. A programmable timer in the system determines the rate/interval at which loops are serviced.

A PC is connected to the micro controller through voltage converter (MAX 232).

The voltage converter is connected to the TxD and RxD of the micro controller. The max 232 converts the voltage level to the Rx 232 level to provide interfacing with the PC. The max 232 provides RX 232 voltage level output from a single +5v supply via on chip charge pumps and external capacitors. The max 232 converts the 5v regulated voltage in the circuit to +12v and -12v.

Software is embedded in the PC to create a database to store different records in order to provide detail lists of the changes along with time and date so that it improves cost efficiency. Through a modem, a remote user using GSM network can monitor and control the process.

The microcontroller is programmed to read data from a single channel and solve the process equation..

Steps required are as follows:

i. Input the data
ii. Linearize it
iii. Determine the error
iv. Solve the control mode equation
v. Output the feedback to the appropriate final control element

**4.8 Block Diagrams for each of the nine Sub-units.**

Figures 4.7a to 4.7i are the block diagram of nine sub-units designed by the researcher. As explained earlier, instead of using just one processor, nine dedicated processors were used each for one sub-unit to control the processes. Each of the processors monitors and controls 4 or 3, 0r 2 parameters in the particular sub- unit it is interfaced to. The use of nine processors, one per sub unit was done to meet the timing constraints of the applications

**4.8.1 Well Stream Heater Control System**

The non associated gas produced from the various well heads is fed into the inlet header through the flow stream heaters. These heaters are used to stop hydrates forming as the gas flows through the choke valve which is adjusted by a flow controller driven by pressure controller fitted in the system. Temperature of the incoming steam is 150$^{o}$C while pressure and level is 90 bar and 850mm respectively. Figure 4.8a is the block diagram of well stream heater. In this sub-unit we have three variables to handle namely temperature, pressure and level. LM 35 temperature sensor was used to measure the temperature of the steam. This sensor is fixed at specified location, for the application to be performed. The Sensor gathers the values and sends the information to the microcontroller continuously. The microcontroller receives the values and sends the corresponding values to the LCD to display. This is a cyclic process which performs till the requirement is done. Whenever the temperature exceeds the set point, the controller drives the fan ‘ON’ and if it goes below set point it turns the Heater ‘ON’.

In order to chose the set point the user needs to send the value by SMS message to the GSM subsystem and then set point is automatically assigned to the system. As soon as the new Set-Point gets activated the previous Set-Point gets deleted and the process runs depending on the new Set-Point criteria.

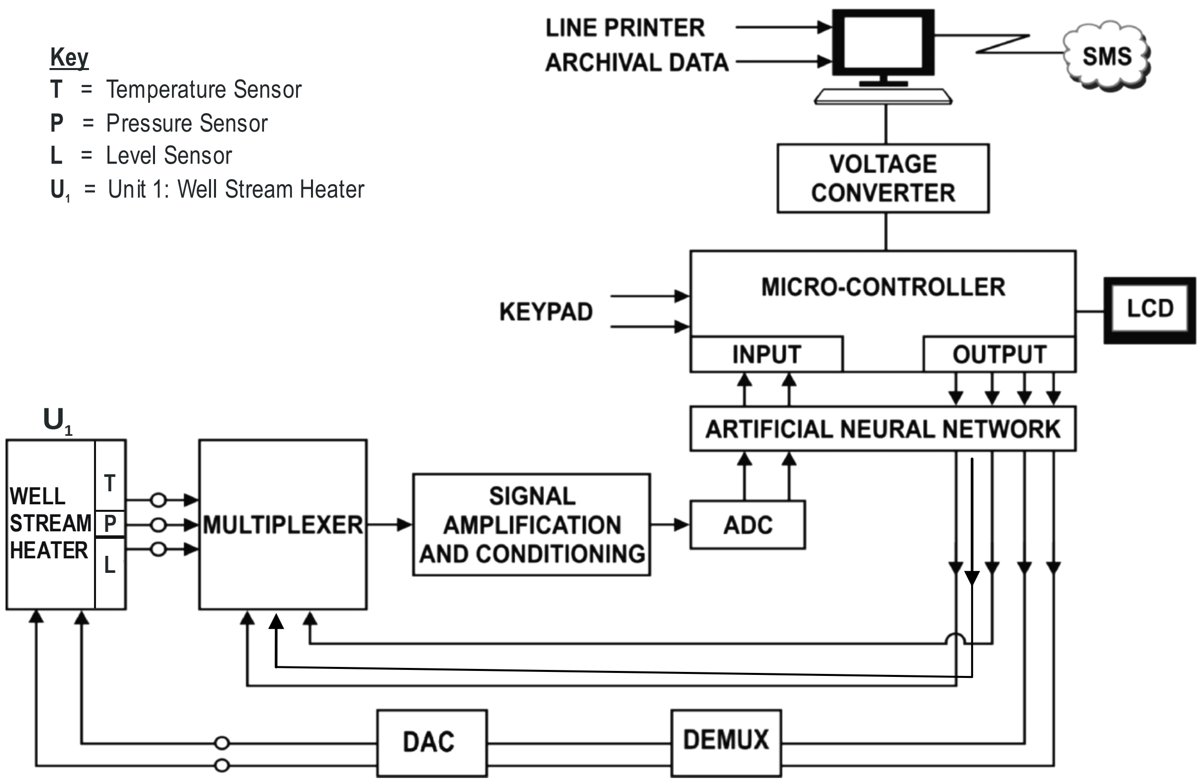



**Figure 4.7a: Block Diagram of Well Stream Heater Control System [Author]**

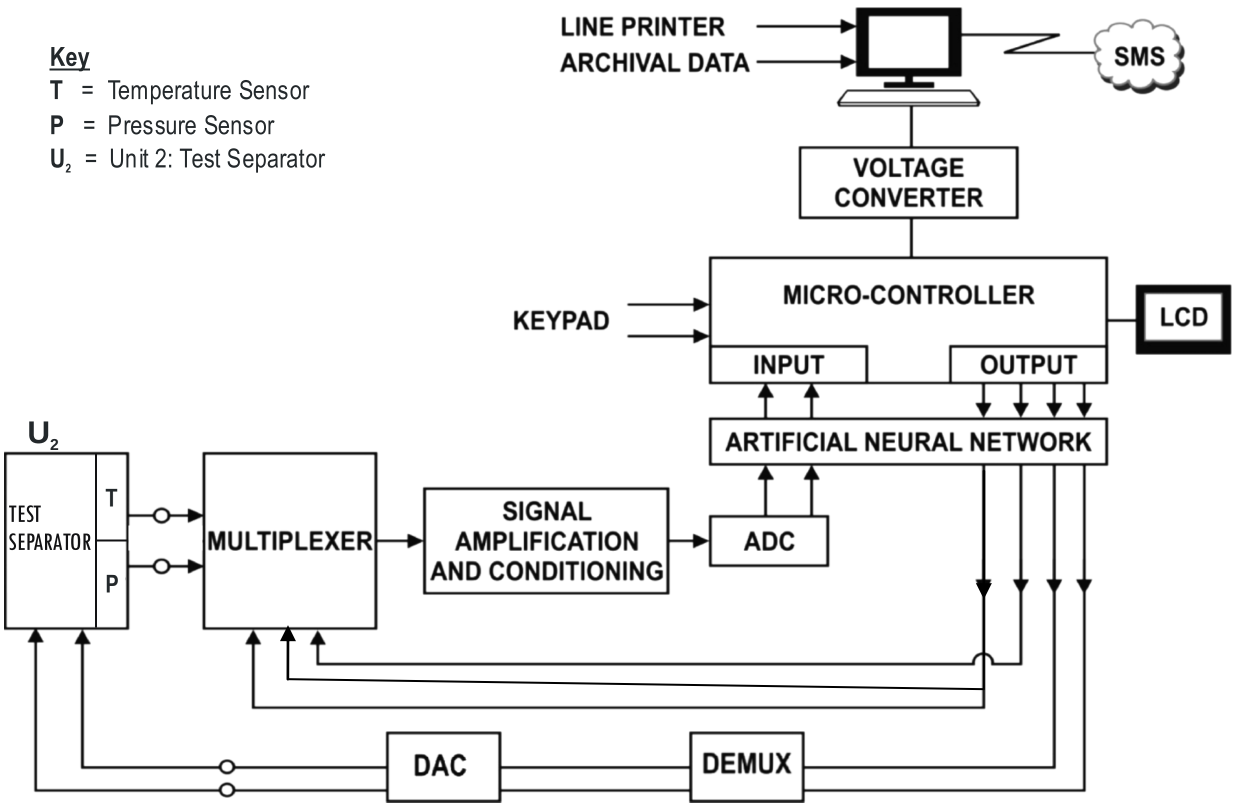



**Figure 4.7b: Block Diagram of the Test Separator Control System [Author]**

### 4.8.2 Test Separator Unit Control System.

The inlet header manifold is of the dual type and the flow lines from the well head are valved to allow each to be switched to a a test separator. Regular testing is essential to ensure quality. We are concerned here with temperature and pressure which are 45$^{o}$C and 102.8 respectively. Figure 4.8b is the test separator subsystem.

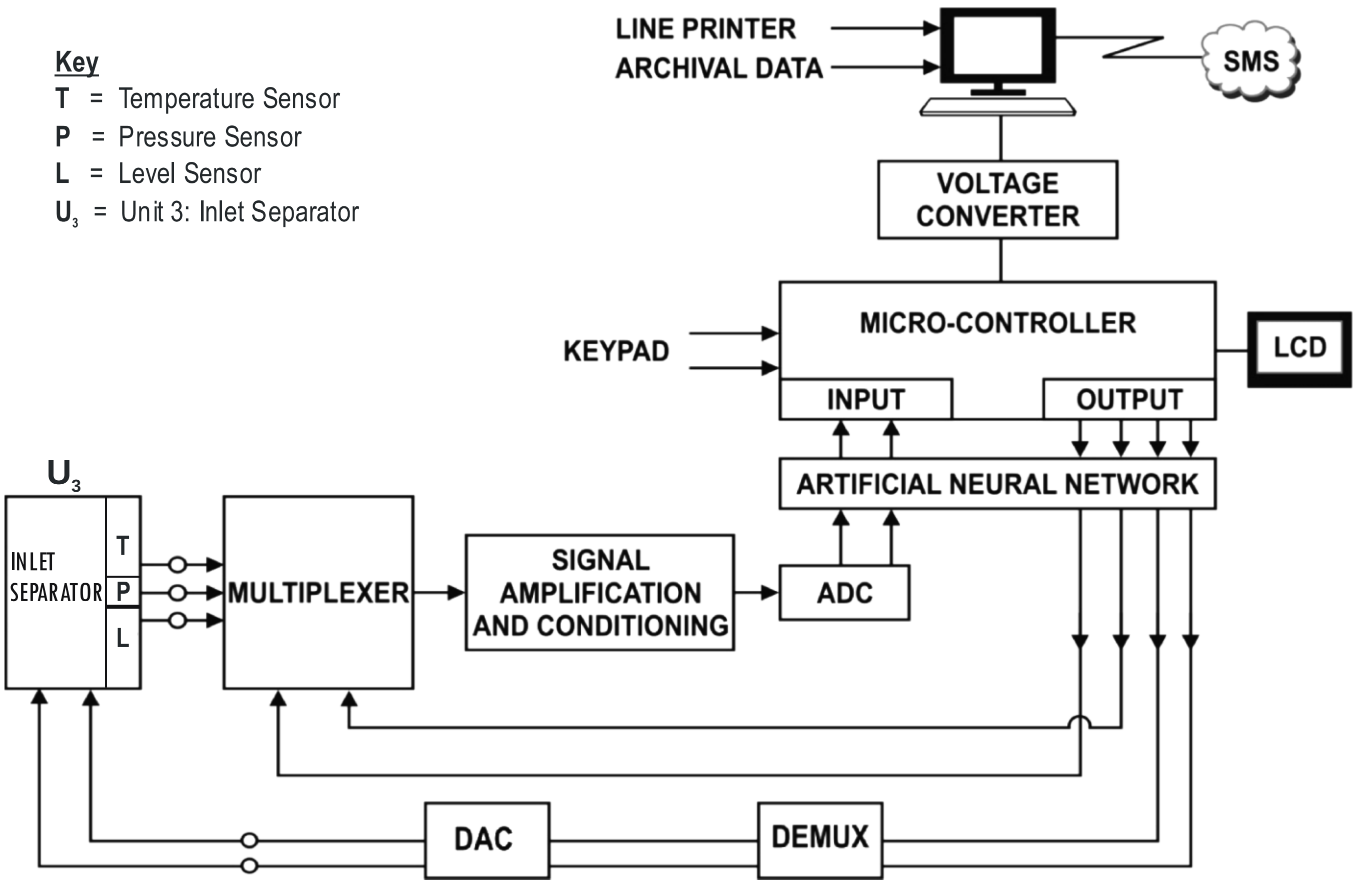



**Figure 4.7c: Block Diagram of Inlet Separator Control System [Author]**

### 4, 8.3 Inlet Separator Control System

The inlet separator (heat exchange) has three variables to monitor. They are temperature 45$^{o}$ C, pressure 192 bar and level 260mm. figure 4.8c is the Inlet Separator Control System. Three parameters are of paramount importance, namely: Temperature, Pressure and Level.        .

### 4.8.4 Gas Cooler Control System

The cooling medium is propane gas from the propane refrigeration unit. The cooled gas from the low temperature separator is then heated by the incoming gas before passing the glycol contactor for drying. The produced liquid from the inlet separator is cooled in the same way Inlet temperature and pressure is 25$^{o}$C and 50barrespectively. Outlet temperature and pressure are 35$^{o}$C and 93 bar respectively

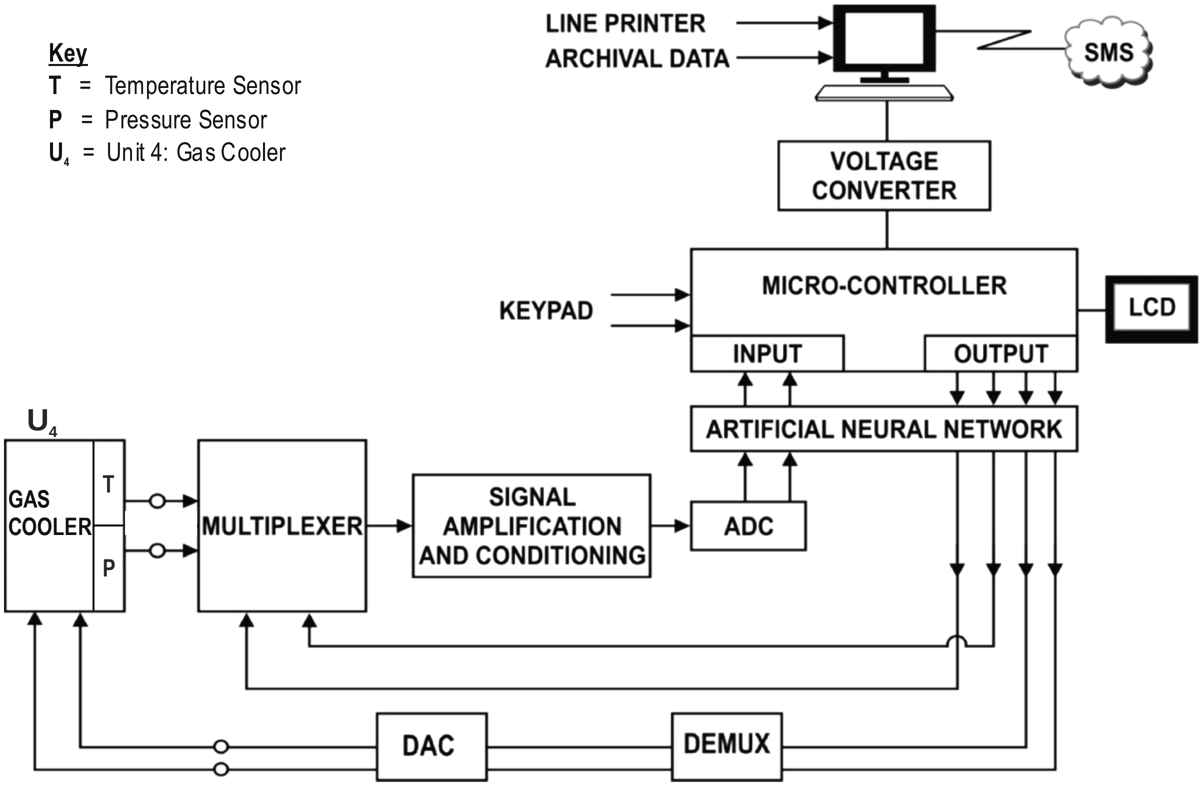



**Figure 4.7d: Block Diagram of Gas Cooler Control System[Author]**

### 4.8.5 Propane Refrigeration Control System

In this subsystem, hot propane gas returning from both the gas and liquid cooler is scrubbed to remove contamination and passes via a heat exchanger to a lobe type compressor which increases the pressure to raise the liquefying temperature Figure 4.8e is the block diagram of this subsystem.

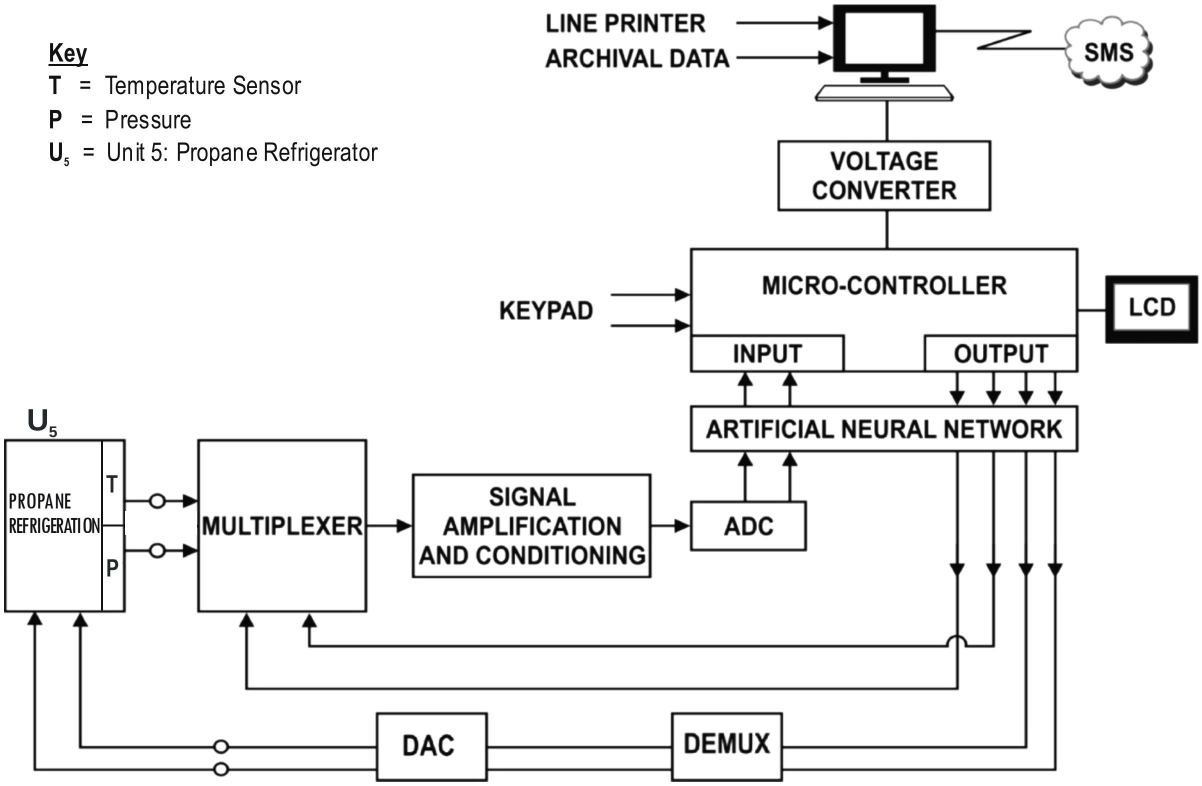



**Figure 4.7e: Block diagram of Propane Refrigeration Control System [Author]**

### 4.8.6 Condensate Processing Control System

The purpose of this unit is to stabilize the condensate produced from the NAG and AG supplies and ship it. The condensate is quality controlled to ensure that the vapour above the stabilized condensate is below the agreed figure. Pressure is atmospheric pressure while temperature is 23$^{o}$C. Figure4.8f is the block diagram of the subsystem.

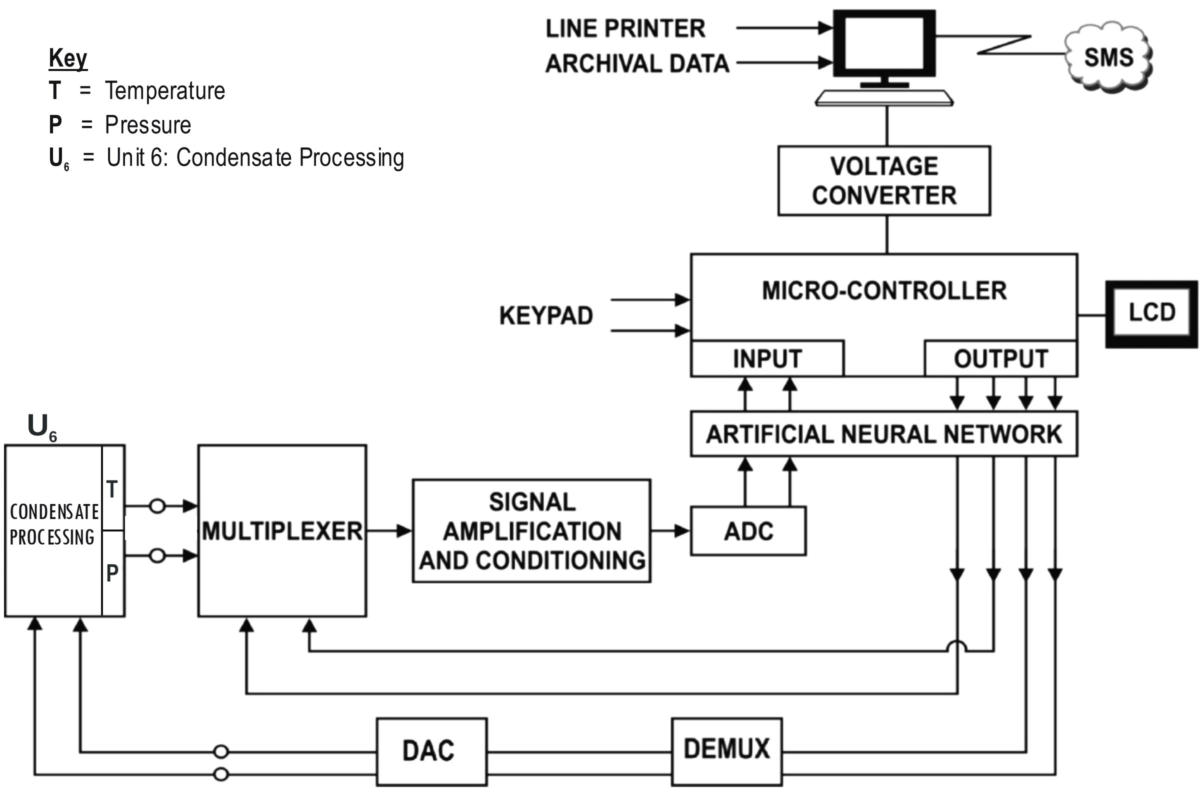



**Figure 4.7f: Block Diagram of Condensate Processing Control System [Author]**

The purpose of this subunit is to raise the pressure of associated gas and flash gas produced by the condensate unit to a value so that the gas can be mixed with the NAG at the inlet of the TEG drying unit. The inlet pressure will be 6 bar and the outlet about 95 bar Temperature is 35$^{o}$C while flow rate is 65.9 mmscf/d..

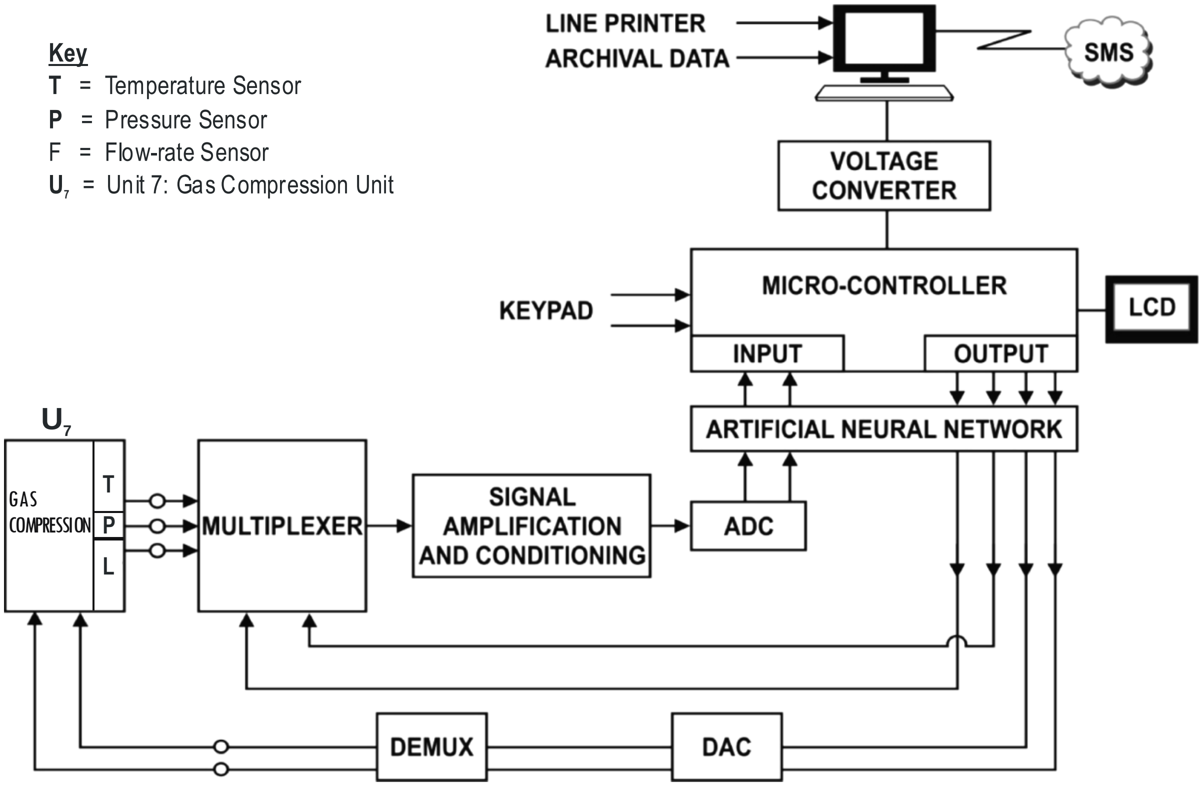



**Figure 4.7g: Block Diagram of Gas Compression Control System Author]**

**4.8.7**The condensate export pumps are high capacity/high heat pumps. The condensate export booster and export pump take suction from either the condensate surge drum or the condensate storage tank. The pump are provided with pressure relieve valves located downstream of the pump relieving of the suction of the export pump. They are set at a design pressure and temperature 55bra and 42$^{o}$ C respectively.

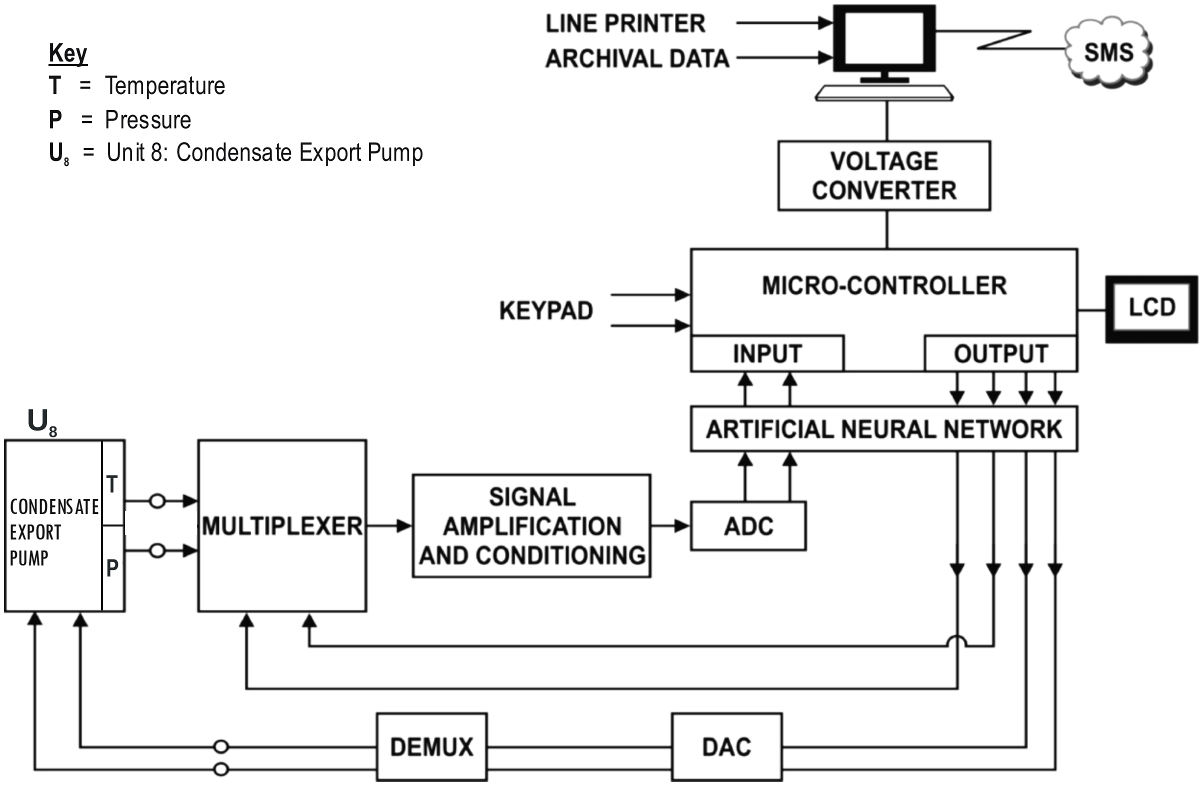



**Figure 4.7h: Block Diagram of Condensate Export Pump Control System [Author]**

### 4.8.8 Teg Storage and Transfer Control System

The Teg tanks are vented to atmosphere. The Teg transfer pumps are local manual stop/start with low low liquid level operation shut down three (OSD 3) pump trip, figure 4.18i is the block diagram of Teg Storage and Transfer. The parameters to monitor and control in these unit are Temperature, Pressure and Level.

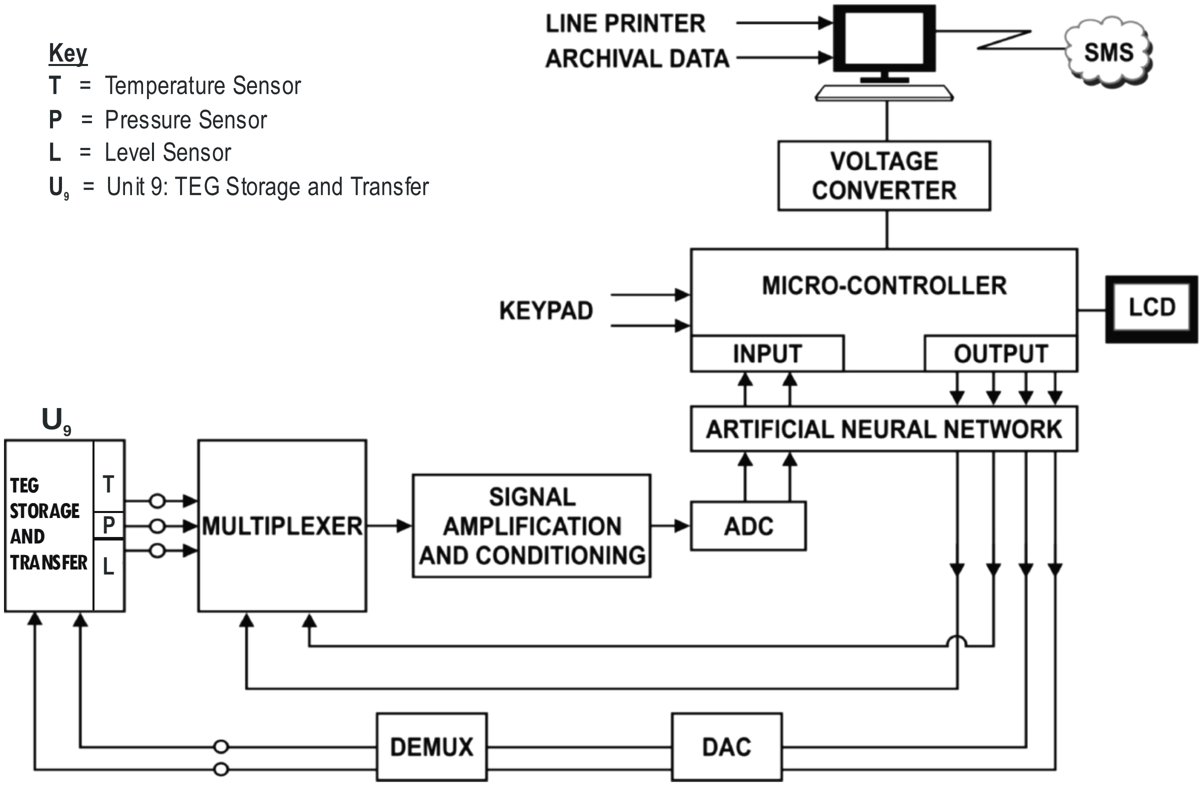



Figure **4.7i: Block Diagram of TEG Storage and Transfer Control System[Author]**

### 4.8.9 Test Separator Unit Control System.

The inlet header manifold is of the dual type and the flow lines from the well head are valved to allow each to be switched to a a test separator. Regular testing is essential to ensure quality. We are concerned here with temperature and pressure which are 45$^{o}$C and 102.8 respectively. Figure 4.8b is the test separator subsystem.

### 4.8.10. The Integration of the 9 Units Multiprocessor Control System

Figure 4.8 is the block diagram of the 9 units control systems integrated together as one system. As indicated earlier nine dedicated microprocessors were used in the design. The different subsystems are connected to a host computer and they communicate through a GSM network.

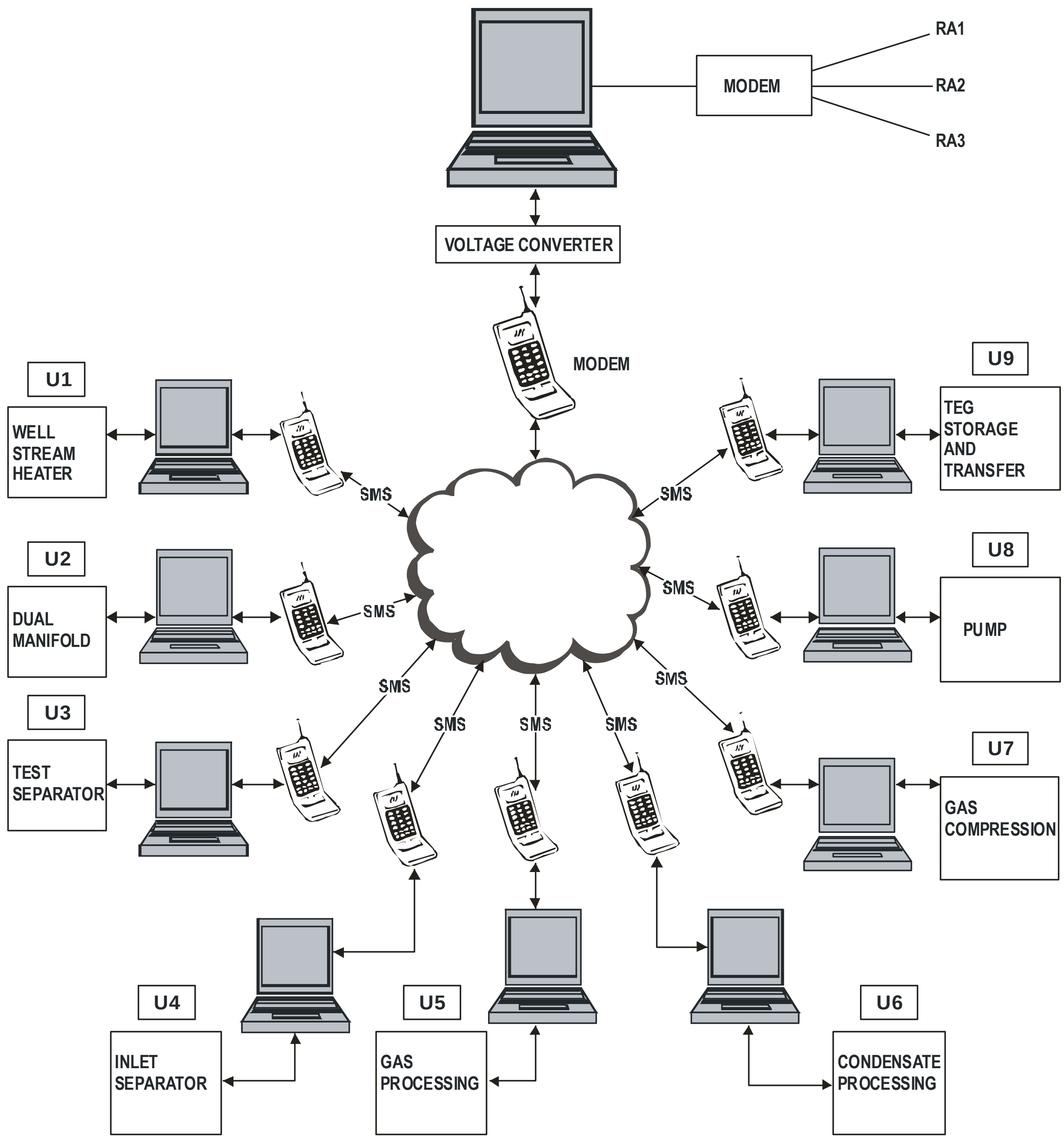



**Figure 4.8: Block Diagram of the Integration of the 9 Units control Systems**

### 4.9 Software Requirements

Various types of programs were written at different points in the design. To solve control equation, update various set points and when necessary report to an agent, the microcontroller was programmed with C, assembly language, and artificial neural networks. The GSM module was also in addition to ATM commands programmed with visual basic.

### 4. 10 The Input Interface

The input interface unit is designed around a standard analog to digital controller (ADC) chips. This ADC converts the analog signal to digital form so that it can be processed by the micro computer. Others devices include an analog multiplexer and the sensors.

The sensors are used to monitor the parameters of interest in the plant.

### 4.10.1 Sensors

The sensors are often regarded as primary element. Wireless sensor nodes are installed on industrial devices and equipments to monitor the process data such as pressure, temperature, flow, and level. They acquire information about the status of the process variables. Typical examples are LM 35 and thermocouple for temperature measurements and differential cells and ultrasonic for liquid level measurements. Figure4.9a is a temperature sensor interfaced to ADC Circuit.

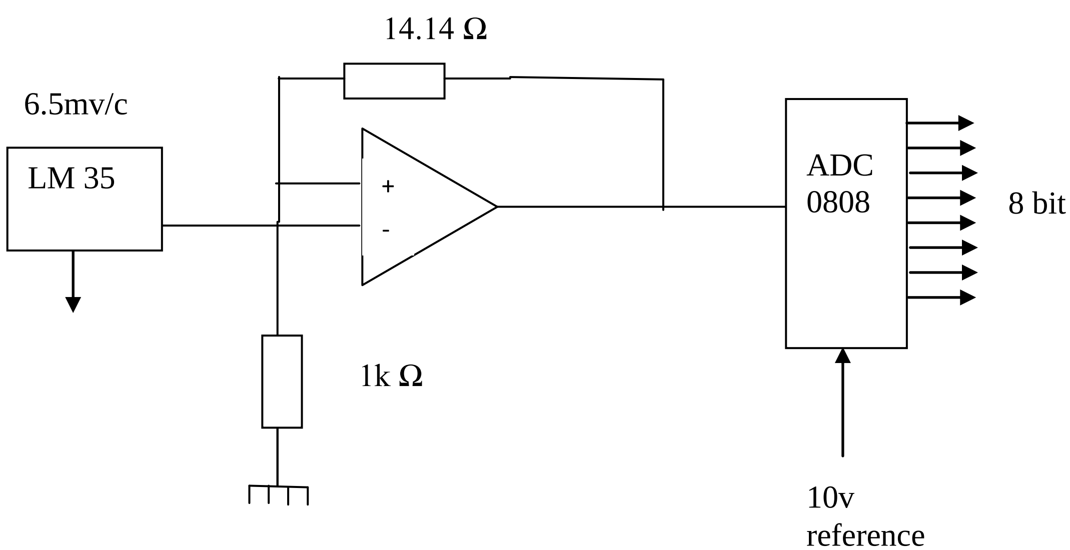



**FIG 4.9a: Temperature Sensor interfaced to ADC Circuit**

**es:**

There are four input variables that are used by the system namely:

•Temperature

• Pressure

• Flow rate

• Level.

These four input variables are used by the system to determine the quality of the product.

**4.10.2 The output parameters are**:

•Opening/closing the valves/pumps for water and/or Gas/Oil, and adjusting their amounts in combustion. Regulating the speed of electric motor

• Turning energy systems on/off (example: heating,);

**4.10.3 Temperature Sensing.**

There are various types of temperature sensors. In this design a semiconductor temperature sensor LM 35 is used. Figures 4.9b is the circuit diagram of LM 35. In figure 4.9b the LM 35 is connected to a converter. A sensor measures a variable and converts it to independent variable. The sensor output varies from 0.2 to 0.6 but the equipment to which the sensor output is connected varies from 0 to 50. The required signal condition is effected by changing the temperature to zero when the output is 0.2v by subtracting 0.2 from sensor output which is called a zero shift or a bias adjustment. We now have a voltage that varies from 0 to 0.4 since we need to make the voltage larger, we modify by12.5. The new output will vary from 0-5 as required (this is called amplification) and the 12.5, is called the gain.

**Figure 4.9b: LM 35 Sensor [Source WWW.Sensor.net]**

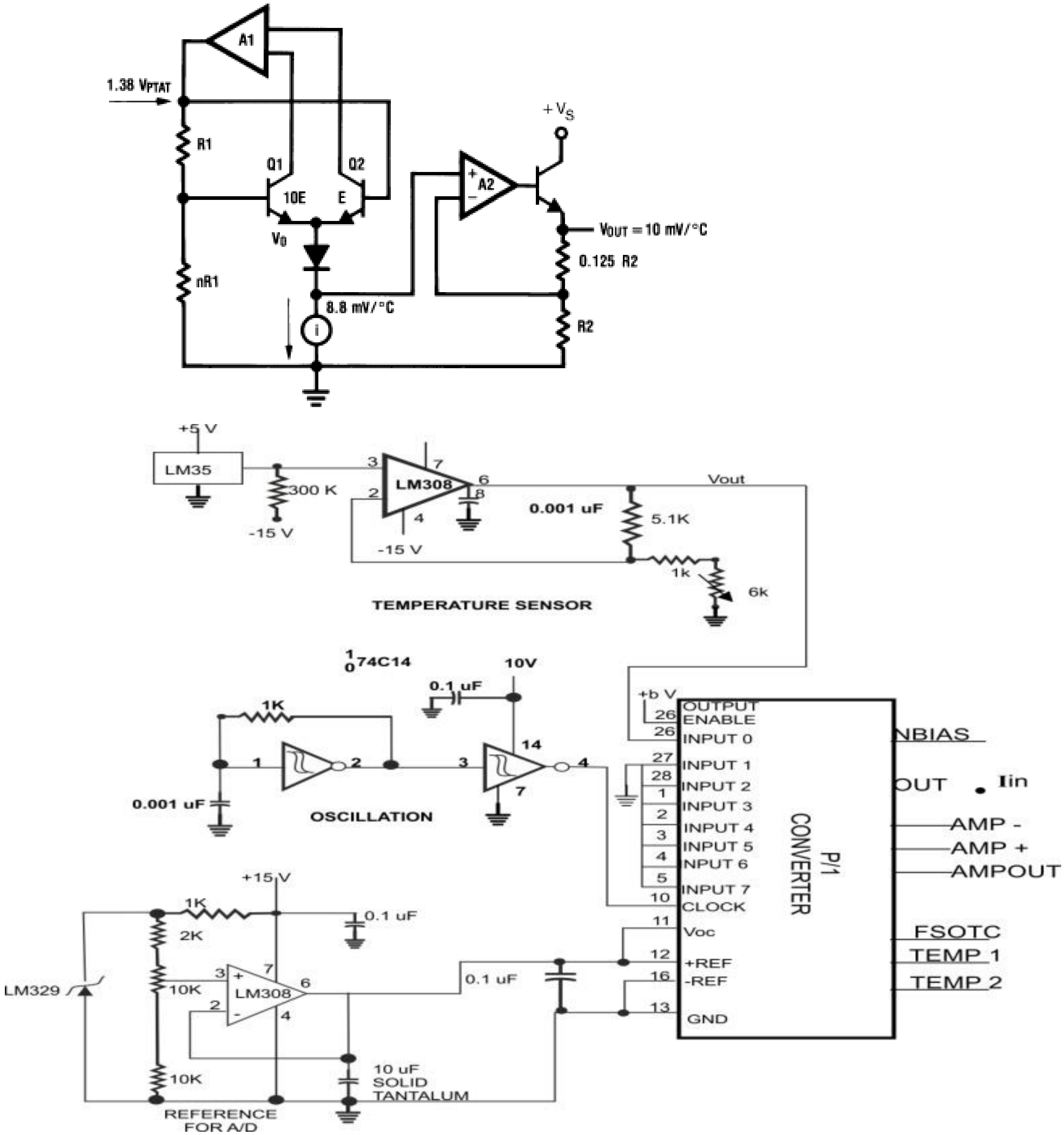



**Figure 4.9c: LM 35 Sensor Connected to a Converter.[Hall, 1986 ]**

### 4.10.4 Level sensing

This is the measurement of liquid level in tank which may be gas, oil or water. The ultrasonic level sensor is preferred because it consists of separate transmitter and receiver elements. It measure the height of the liquid column directly, the transducer can be mounted in the bottom of the tank. In this work the ultrasonic sensor was used. Figure 4.9d is the circuit diagram of the ultrasonic sensor.

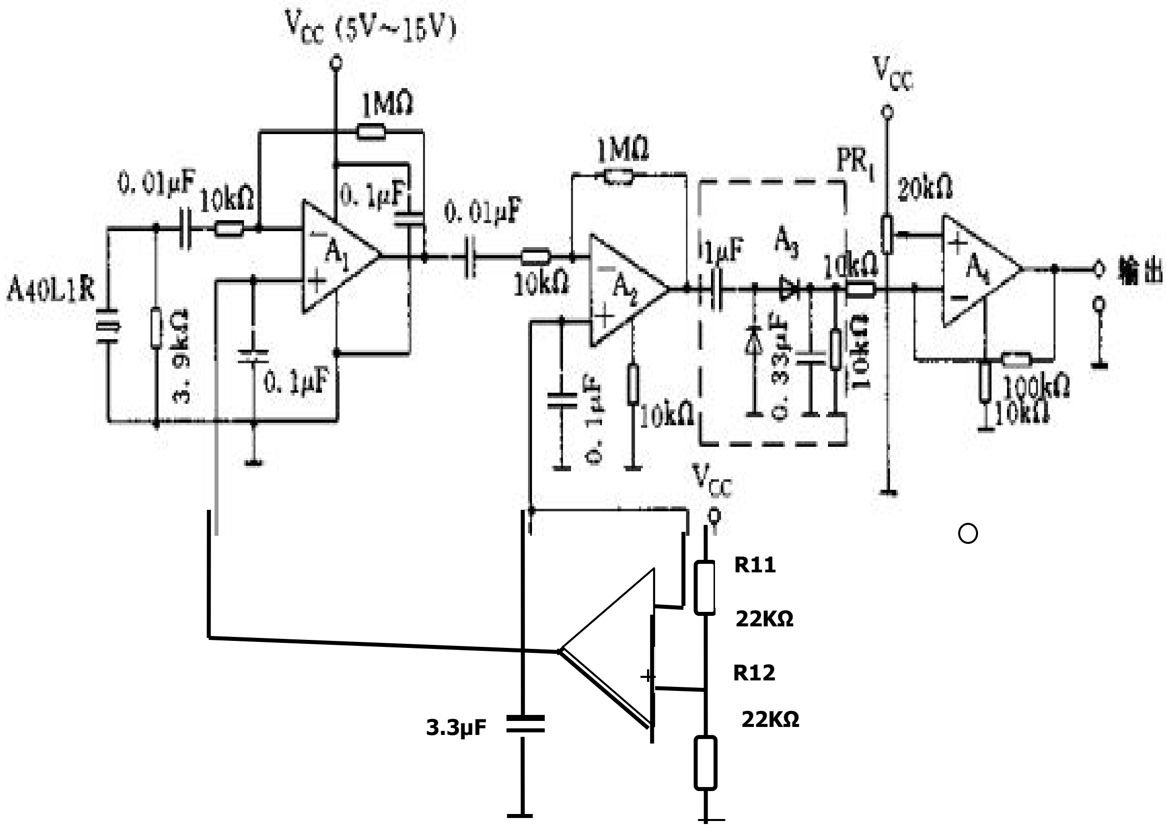



**Figure 4.9d: Circuit Diagram of Ultrasonic sensor [Source www.sensor.net]**

### 4.10.5 Flow Sensing

Flow is at the heart of process industries. The flow of gas from the well head to the process plant and then to Bonny (LNG) is expressed either as volume flow rate or flow velocity.

For velocity flow rate

V = Q /A where

V = flow velocity

Q = volume flow rate

A = Cross Sectional Area of flow carrier

One of the most common methods of measuring the flow of liquids in pipe is by introducing a restriction in the pipe and measuring the pressure drop that result across the restriction. In this work the differential pressure sensor was used. Figure 4.9e is a differential pressure measurement sensor for measuring flow.

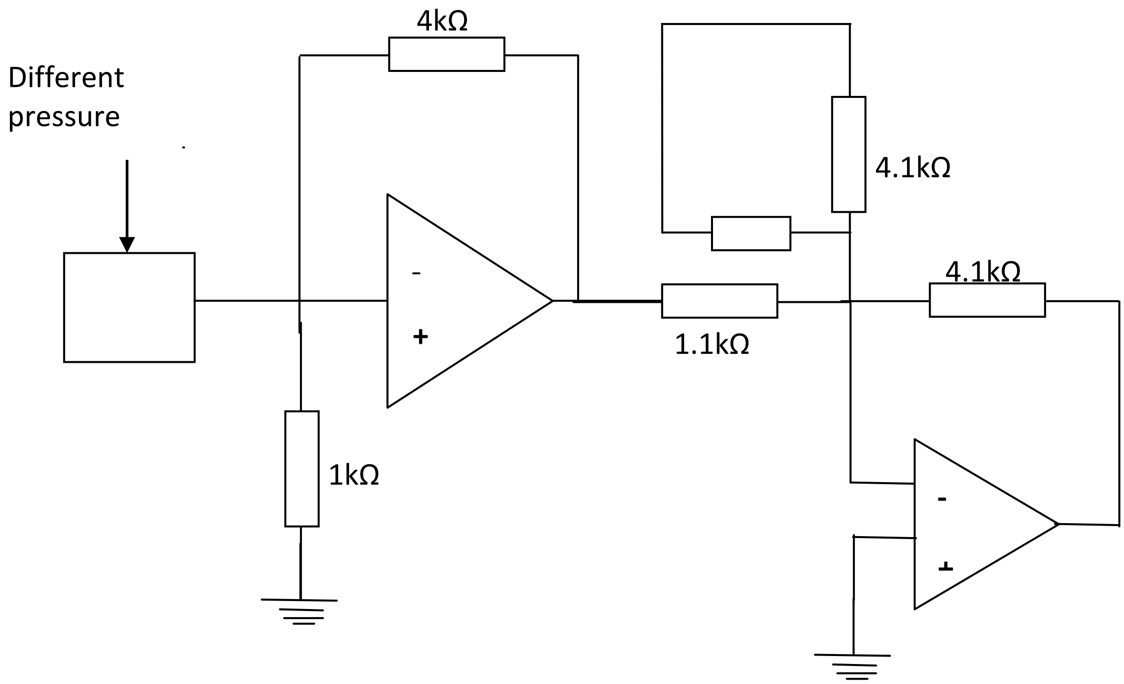



**Figure 4.9e: Pressure Differential Sensor.**

### 4.10.6 Pressure Sensing

Measurement and control of liquid and gas pressure is common in oil and gas industries like any other process industries. Pressure information is converted to the form measured by a sensor. There is solid state pressure sensor. Thermocouple can be used as pressure transducers. Figure 4.10 is a pressure sensor connected to the microcontroller for the measurement of pressure for the system.

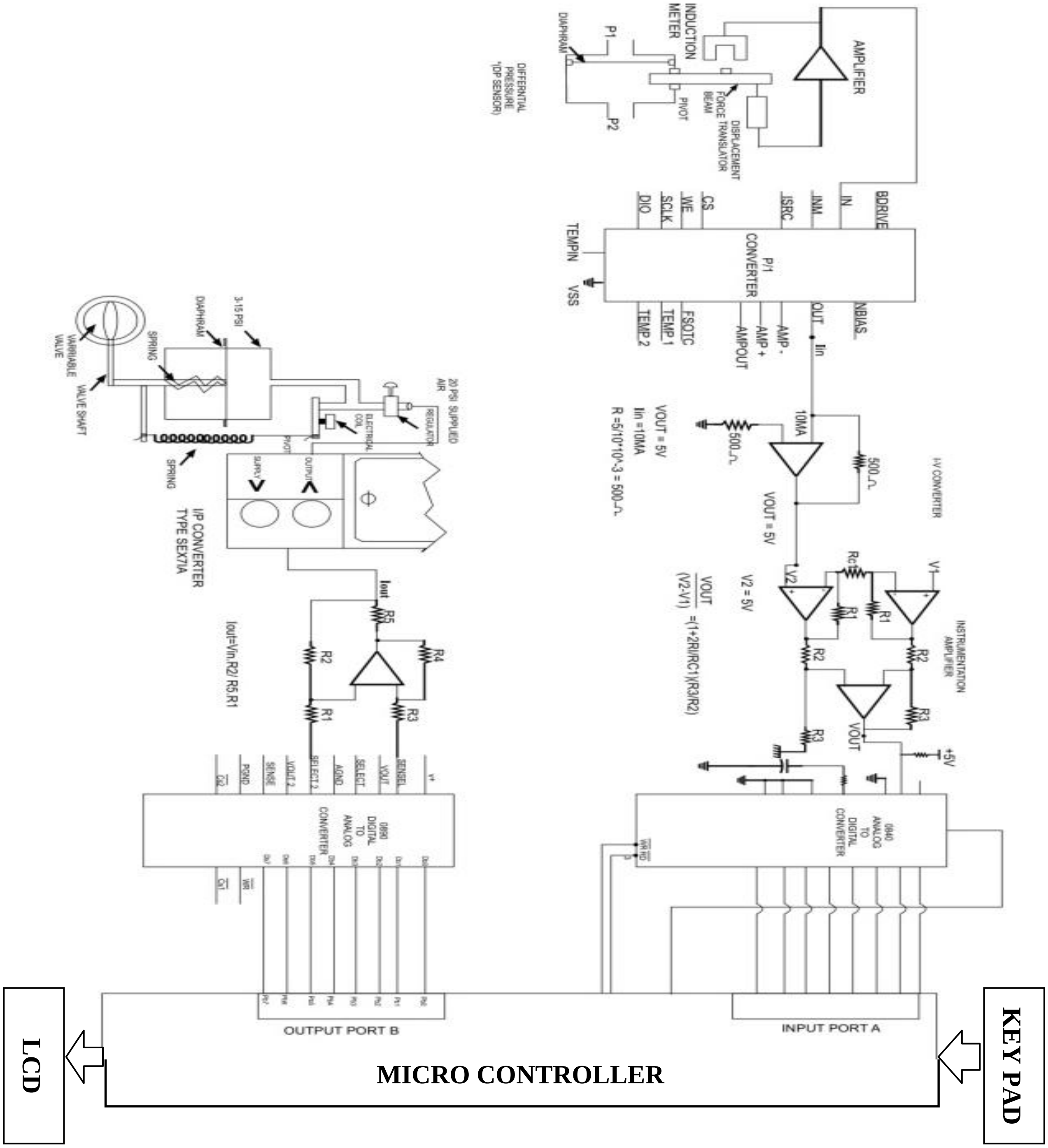



Figure 4.10: Pressure sensor connected to the microprocessor [Author]

### 4.11 Multiplexer:

The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder.

Table 4.4 shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

Table 4.4 Analog Address Line

| CHANNEL | ADDRESS LINE | | |
|---|---|---|---|
| **ANALOG CHANNEL** | **C** | **B** | **A** |
| **IN0** | L | L | L |
| IN1 | L | L | H |
| IN2 | L | H | L |
| IN3 | L | H | H |
| IN4 | H | L | L |
| IN5 | H | L | H |
| IN6 | H | H | L |
| IN7 | H | H | H |

### 4.11.1. Signal Conditioning

Signal conditioning refers to operation performed on signals to convert them to a form suitable for interfacing with other elements in the process control loop. Signal conditioning usually involves the following:

- Sampling (and conditioning) of continuous signals
- Appropriate mathematical description of sampled-data systems.
- Amplification of weak signals
- noise suppression by filtering
- Possible multiplexing.

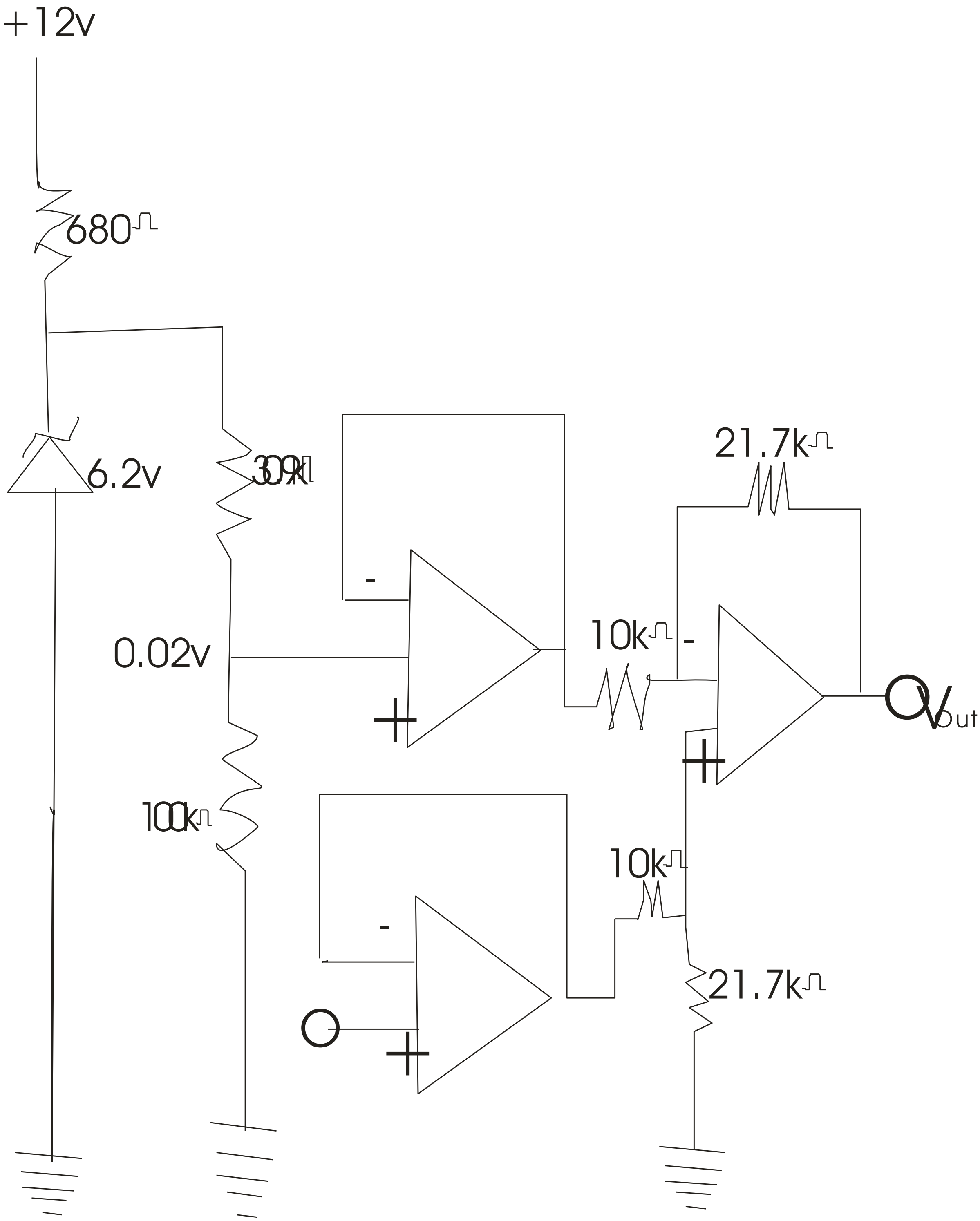



**Figure 4.11: Signal Conditioning Circuit**

In figure 4.11 the signaling conditioning circuit of this sensor outputs a range of 20 to 250mV.

To develop a signal conditioning that becomes 0 to 5v,.the circuit must have high input impedance.

Developing an equation for the output in terms of the input of the op amp circuit,

We can use the eqation

$V_{OUT} = - [R_2/R_1V_1 + R_2/R_1V_2]$ (7)

$V_{OUT} = GainV_{in} + V$

Selection of the values of the resistor is based n keeping the currents in milliamps

Make $R_1 = R_{2\,+}\, R_3$

$R_{2\,=}\;R_1/R_3$

Because this a differential amplifier $V_{OUT} = - A (V_{2\,-}\, V_1)$

$V_{OUT} = - R_2/R_1 (V_2 - V_1)$ (8)

We can now develop a current to provide the variation of the equation. The equation is that of a straight line

$V_{out} = Mv_{in+}\; V_{in}$ (where m is the slope and represent the gain when m> 1while m<1 is the attenuation)

$V_{o\,is}$ the intercept that is $V_{out\,=}\, V_o$ if $V_{in0}$

$0 = m (0.02) +V_o$

$5 = m(0.25) + V_0$

$M = 21.7$ and $V_0 = - 0.434V$

using standard algebra the equation is

$V_{out\,=}\, 21.7V_{in}- 0.434$

$V_{out}= 21.7(V_{in} -0.02)$

21.7 is the gain while 0.02 is the fixed input.

### 4.12 Interfacing ADC 0808 to 8051 Micro controller.

In many embedded systems microcontrollers need to take digital input. Most of the sensors and transducers such as temperature, humidity and pressure, produce analog output. Before interfacing these sensors to micro controllers we require to convert the analog output of these sensors to digital so that the controller can read it. Some micro controllers have built in Analog to Digital Converter (ADC) so there is no need of external ADC. For controllers that don't have internal ADC external ADC is used. One of the most commonly used ADC is ADC0808. ADC 0808 is a Successive approximation type with 8 channels i.e. it can directly access 8 single ended analog signals.

**Table 4.5: Analog channel selection**

| Analog Channel | ADDRESS LINES | | |
|---|---|---|---|
| | C | B | A |
| IN0 | 0 | 0 | 0 |
| IN1 | 0 | 0 | 1 |
| IN2 | 0 | 1 | 0 |
| IN3 | 0 | 1 | 1 |
| IN4 | 1 | 0 | 0 |
| IN5 | 1 | 0 | 1 |
| IN6 | 1 | 1 | 0 |
| IN7 | 1 | 1 | 1 |

- **ADDRESS LINE A, B, C**

The device contains 8-channels. A particular channel is selected by using the address decoder lines C, B, A. The table 4.5 shows the input states for address lines to select any channel.

- **Address Latch Enable ALE**

The address is latched on the Low – High transition of ALE.

- **START**

The ADC's Successive Approximation Register (SAR) is reset on the positive edge i.e. Low-High of the Start Conversion pulse. Whereas the conversion is begun on the falling edge i.e. High – Low of the pulse.

- **Output Enable**

Whenever data has to be read from the ADC, Output Enable pin has to be pulled high thus enabling the TRI-STATE outputs, allowing data to be read from the data pins D0-D7.

- End of Conversion (EOC)

This Pin becomes high when the conversion has ended, so the controller comes to know that the data can now be read from the data pins.

- **Clock**

**External clock pulses are to be given to the ADC; this can be given either from LM 555 in a stable mode or the controller can also be used to give the pulses.** Figure 4.12 shows the connection of 8051 to 0808

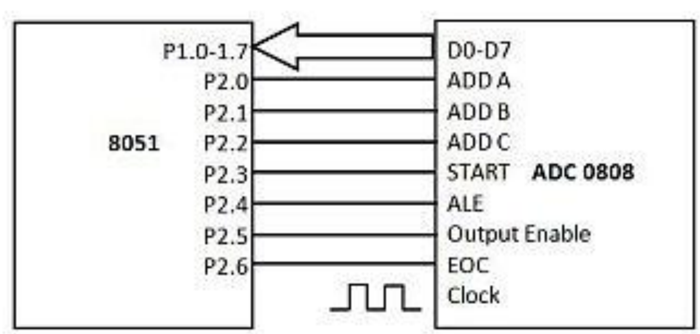



**Figure 4.12: Connections of 8051 to 0808**

**Algorithm**

1. Start.
2. Select the channel.
3. A Low – High transition on ALE to latch in the address.
4. A Low – High transition on Start to reset the ADC's SAR.

5. A High – Low transition on ALE.
6. A High – Low transition on start to start the conversion.
7. Wait for End of cycle (EOC) pin to become high.
8. Make Output Enable pin High.
9. Take Data from the ADC's output
10. Make Output Enable pin Low.
11. Stop

The total numbers of lines required are:

- data lines: 8
- ALE: 1
- START: 1
- EOC:1
- Output Enable:1

. One can directly connect the OE pin to $V_{cc.}$ Moreover instead of polling for EOC just put some delay so that instead of 12 lines one will require 10 lines.

One can also provide the clock through the controller thus eliminating the need of external circuit for clock.

### 4.12.1 Calculating Step Size

ADC 0808 is an 8 bit ADC i.e. it divides the voltage applied at $V_{ref+}$ & $V_{ref-}$ into $2^8$ i.e. 256 steps.

Step Size = ($V_{ref+}$ - $V_{ref-}$)/256

Suppose $V_{ref+}$ is connected to $V_{cc}$ i.e. 5V & $V_{ref-}$ is connected to the Gnd then the step size will be

Step size= (5 - 0)/256= 19.53 mv.

**Calculating $D_{out}$.**

The data we get at the D0 - D7 depends upon the step size & the Input voltage i.e. $V_{in}$.

$D_{out} = V_{in}$ /step Size.

If we want to interface sensors like LM35 which has output 10mv/°C then one would suggest that we set the $V_{ref+}$ to 2.56v so that the step size will be

Step size= (2.56 - 0)/256= 10 mv.

So now whatever reading that one gets from the ADC will be equal to the actual temperature.

### 4.13 Understanding the operation of Controller

There are two types of controllers: **Open loop controller** which is also called non-feedback controllers. This type of controller is designed on the following principles:

- It takes input and computes output for the system.
- It does not have any feedbacks to determine whether the desired output goal is achieved or not

**Closed loop Controller** is based on pre-defined concept and utilizing feedback from controlled object/system in some manner. The control unit continuously receives feedback from different sensors placed in the field. It enables the control unit to update its data about important system parameters. The Control Unit decides how much valve to open in accordance with the data collected from sensors and predefined parameters.

Figure 4.13 is a typical example how the microcontroller does its work. The diagram is a typical automatic level control of the surge tank. The differential pressure transmitter measures the value of a parameter in this case the level of liquid in the tank. The value is compared to a pre-set desired value and if there is a difference, some function is controlled to vary a property of the system. In industrial process, we use process instruments to control the process. Process instruments are devices that can measure, indicate record and control the properties of the process. An example is a separator vessel in flow station.

The Input signal from the transmitter is converted to a form usable by the controller. Here current signal is converted to a voltage signal for internal use.

The input is compared to the set point to determine if an error exists. If an error signal exists, the error signal is produced and then used by the controller to compute and produce the correct output signal, The computations that are done by the controller are determined by the control mode being used.

The output of the controller is 4mA – 20mA signal which is used to operate the control valve. The control valve responds to a signal and regulates the flow to the tank. As the flow to the tanks changes, the tank level changes. The level change is reflected by a change in the differential pressure transmitter output. The new measured value is sent to the controller which produces a new error value. The process continues repeatedly until no error value exists. The four basic functions of a close loop automatic system are: - measurement, comparison, computation and correction. The control signal operate the final control elements which correct the process

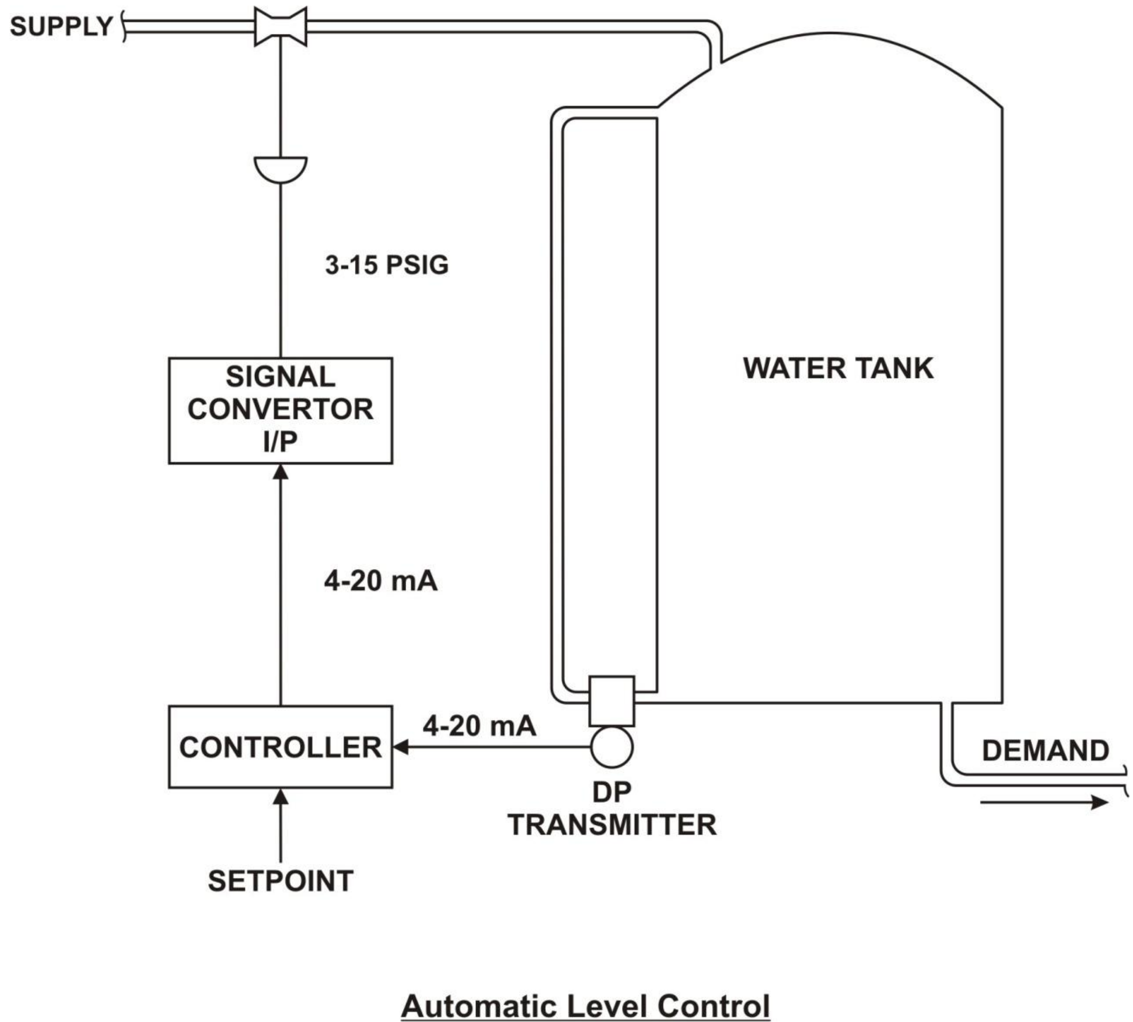



**Figure 4.13: Automatic Level Control**

**4.14 Temperature and flow rate in Stream Well as shown in the Diagrams**

Figure 4.14 is temperature control in the gas plant. The temperature-sensing element in the circuit is an LM35 precision Celsius temperature sensor. The voltage between the output pin and the ground pin of this device is OV at O$^{o}$C and will increase by 10mV for each increase of 1$^{0}$C above that. The 240k Ω resistor connecting the output of the LM35 to -1Ω allows the output to go negative for temperature below 0$^{0}$c. The circuit is able to measure -55 to 850$^{0}$c. . The Sensor gathers the 8bit data value which represents the temperature values and sends the information to the microcontroller continuously. The microcontroller receives the values and sends the corresponding values to the LCD to display. This is a cyclic process which performs till our requirement is done.

An LM308 buffers and amplifies the signal from the sensor by 2 so that the signal uses a greater part of the input range of A/D converter.

The A/D converter used here is ADC0808 system. It is an 8-input data acquisition system. The ADC can support 8 processes at a time, of which one is selected using three address lines. It uses a 3-bit address to select which input signal you want to select and digitized.

Schematic trigger inverters in a 74C14 are connected as an oscillator to produce a 300KHZ clock for the DAS. The voltage drop across an LM329 low drift zener is buffered by an LM308 amplifier to produce $V_{cc\ and}$ $V_{ref}$ of 5-12V for the A/D converter. With this reference voltage, the A/D converter will have 256 steps of 20mv each.

In this process to reach our requirement one assigns a certain Set-Points for the sensor to monitor the temperature (Example; Temperature-39 degrees). If the Temperature Sensor crosses its specific Set-Point the microcontroller drives the Fan 'ON' and starts cooling to reach the set point. If the value from the sensor gets below the Set point, the microcontroller drives the heater 'ON' and starts the heating to reach the set point. This is a cyclic process. To control the power delivered to the heater we use a 25A, 0V turn on, solid state relay.To control the amount of heat put out by the heater, we vary the duty cycle of pulses sent to the relay. The driver transistors on the input of the relay supplies the drive for the relay, isolate the port pin from the relay and holds the relay in the off position when the power is turned on.

The amount of heat output by the heater is controlled by the duty cycle of a pulse wave form sent to the solid state relay.

The binary value of the temperature is stored in a memory location for future refine. The value for the temperature is converted to a BCD value. The current temperature is compared with the set-point value. If the temperature is lower than the set-point the heater will be turned on. If the temp is at or above the set point the heater will be turned off.

Fig 4.15 is the diagram for cooling down a reactor by adjusting the flow rate through the cooling jacket. The process has the specific, desired outcome to reach: maintaining a constant predefined temperature over time. Here, the temperature is the controlled variable. At the same time, it is the input variable since it is measured by a temperature sensor and used in a special function to decide the adjustment of a valve to manipulate the flow rate through the cooling jacket. The desired temperature is the set -point. The valve opening position (e.g. the setting of the valve allowing cooling material to flow through it) is called the manipulated variable since it is subject to control actions. In practice, the temperature value is transmitted to the controller; the controller implements the functions and calculations, transmits the output to control the valve and issues alarm if there are faulty conditions. In the meantime, all data information can be archived for future reference when a review of process trends could provide additional improvements. By utilizing WSN technology, sensing and action devices will communicate wirelessly with an access point (e. g., a gateway or router), which is connected to the control station wirelessly or through wired methods (e.g., Ethernet, Modbus).

Most of the process control applications are mission critical and have stringent requirements. Failure of a control loop may cause unscheduled plant shutdown or even severe accidents in process-controlled plants.

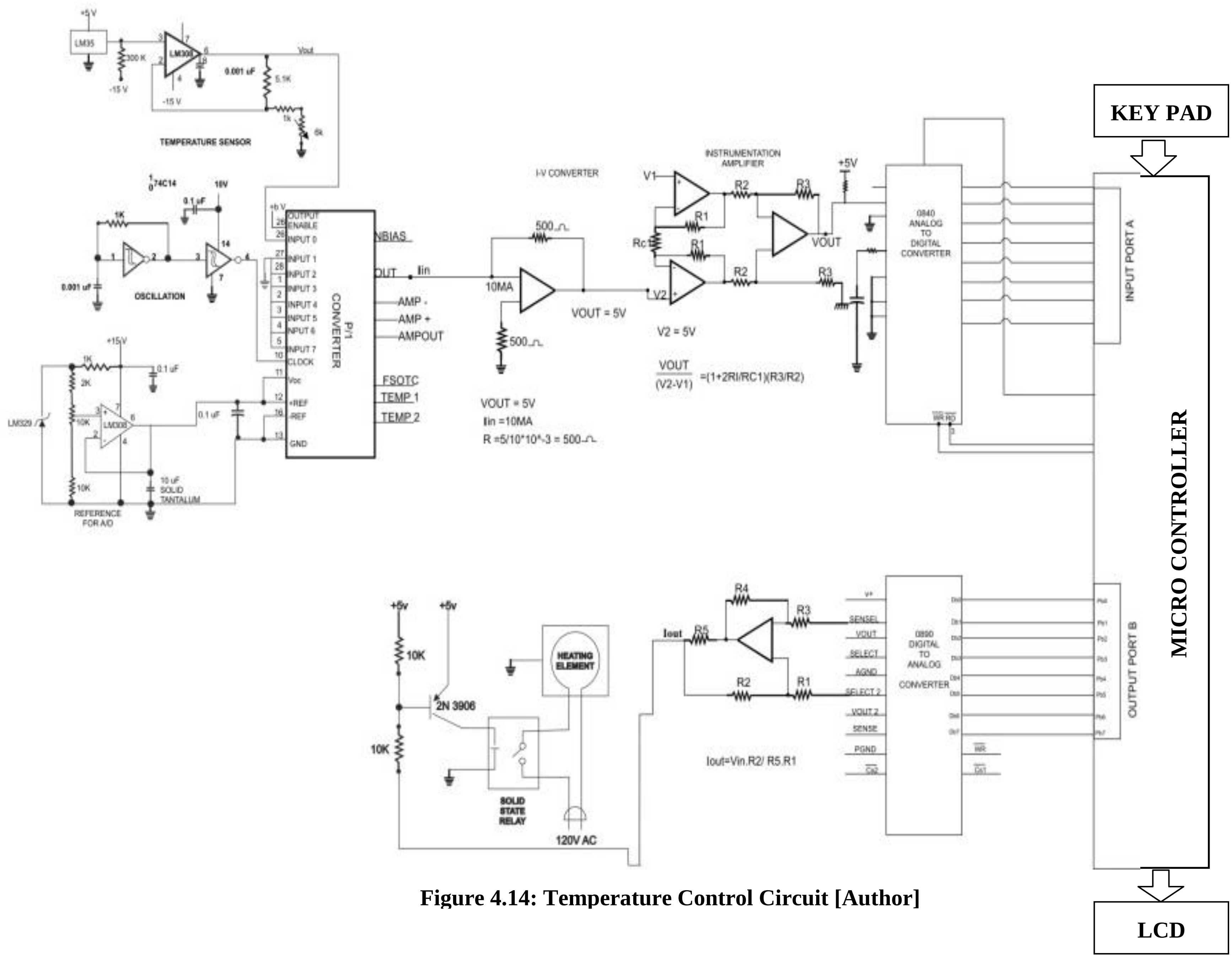



**Figure 4.14: Temperature Control Circuit [Author]**

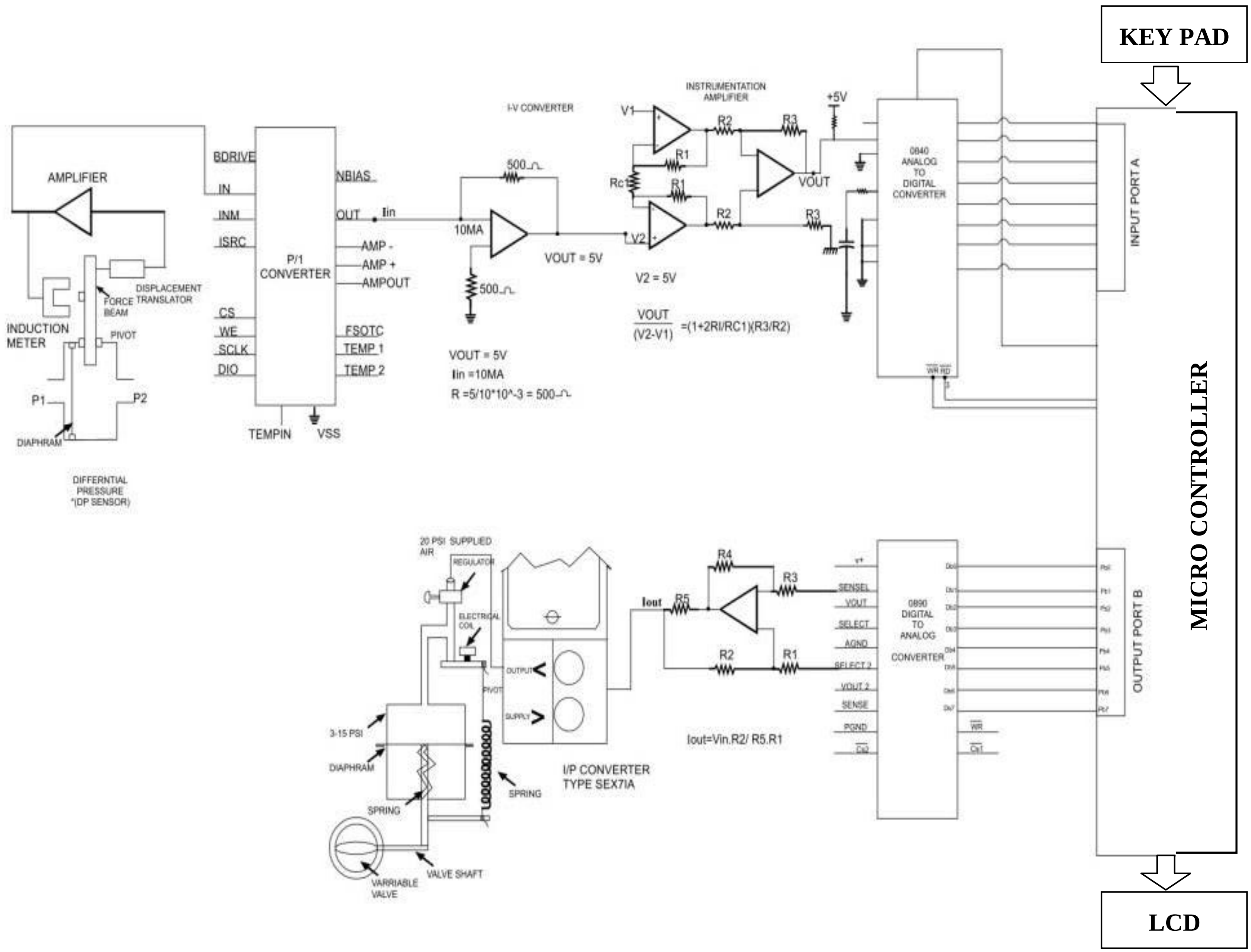



**Figure 4.15: Flow rate microcontroller Circuit**

## 4.15 Development of Intelligent Industrial Process Control Systems.

An intelligent system should be capable of monitoring the process going on, getting information from the system, analyzing such information and giving a feed back or taking actions to achieve the expected results.

In designing this system a number of design approaches can be adopted. The level of intelligence built in the system determines the complexity of the machine which determines the most valuable approach to use

A simple microcontroller based design can achieve a very good level of automation and intelligence but will lead to a lot of complexity with higher level of intelligence. A good approach to ease the design of a complex system is to break the system into a number of smaller but simpler systems which can be linked together to give the desired complex system.

## 4.16 Design of Control unit Using ANN

The Control unit consist of ANN based controller. The neural networks classify the input into any one of the seven possibilities. The decision to be taken is in the hidden layer. The ANN is able to classify the range where the neural logic lies. When it is classified, an action corresponding to the class is performed. The control unit interfaces the required values and measured values. The main function of this stage is to keep the actual value close to the required value. The output of this stage is control input for valves and other actuators. In this design, dynamic ANN is used because dynamic ANN is more powerful than static networks because they have memory. They can be trained to learn sequential and time varying factors.

The system should be able to group data into clusters and then analyze similarities between the clusters. This function is useful for identifying abnormal conditions. The acquired data is initially processed at the data classification module, which uses neural networks and statistics to perform data pre-processing, data compression and data format transfer. It has condition monitoring module, coupled with forward chaining inference, which alerts the user to abnormal process conditions. Then, the incident report module uses the case-based reasoning technique to determine whether the current conditions had previously occurred. If they had, the fault diagnosis module will generate a solution. Otherwise backward chaining will be invoked to detect faults.

The neural network Controller is represented as a block diagram as shown in figure 4.16a The block diagram shows operation of measurement, error detection, controller and final control element. The NNC controller uses the error output to determine an appropriate output signal which is provided as input to the control element. Figure 4.16b is the entire system consisting of of all the different units tht make up the system. The intelligent unit consist of the Neural Network and Expert system components.

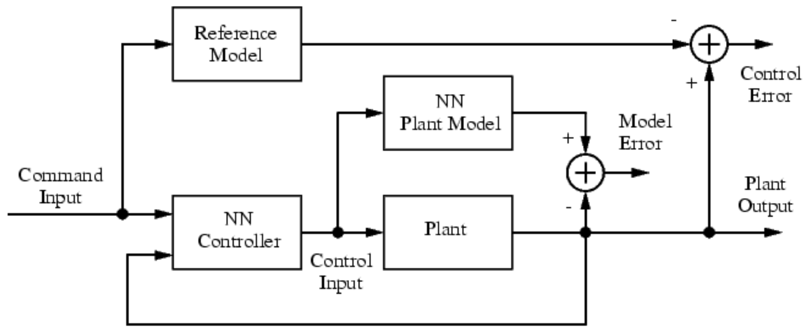



**Figure: 4.16a: NNC Control Loop Block Diagram**

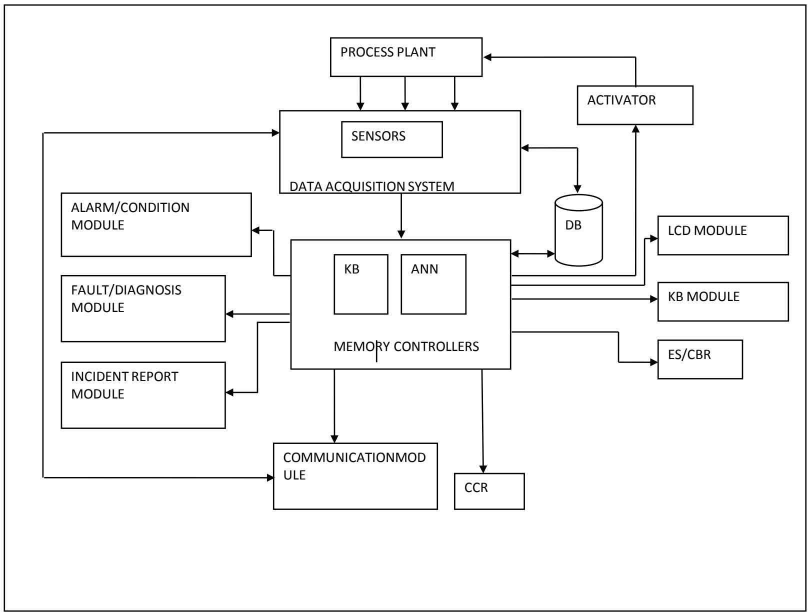



**Figure 4.16b: Block diagram of ANN Based System**

## 4.17 Modeling Equation using the back-propagation Algorithm

The activation function of the artificial neurons in ANNs implementing the back propagation algorithm is a weighted sum (the sum of the inputs x multiplied by their respective weights w)

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^{n} x_i w_{ji} \qquad (9)$$

We can see that the activation depends only on the inputs and the weights.

Using the common output function, the sigmoidal transfer function: for each node is thus

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{A_j(\bar{x}, \bar{w})}} \qquad (10)$$

The sigmoidal function is very close to one for large positive numbers, 0.5 at zero, and very close to zero for large negative numbers. This allows a smooth transition between the low and high output of the neuron (close to zero or close to one). We can see that the output depends only in the activation, which in turn depends on the values of the inputs and their respective weights.

Now, the goal of the training process is to obtain a desired output when certain inputs are given. Since the error is the difference between the actual and the desired output, the error depends on the weights, and we need to adjust the weights in order to minimize the error. We can define the error function for the output of each neuron:

$$E_j(\overline{x}, \overline{w}, d) = \left(O_j(\overline{x}, \overline{w}) - d_j\right)^2 \qquad (11)$$

We take the square of the difference between the output and the desired target because it will be always positive, and because it will be greater if the difference is big, and lesser if the difference is small. The error of the network will simply be the sum of the errors of all the neurons in the output layer:

$$E(\overline{x}, \overline{w}, \overline{d}) = \sum_j \left(O_j(\overline{x}, \overline{w}) - d_j\right)^2 \qquad (12)$$

The back-propagation algorithm now calculates how the error depends on the output, inputs, and weights. After we find this, we can adjust the weights using the method of *gradient descendent*:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \qquad (13)$$

This formula can be interpreted in the following way: the adjustment of each weight ji(w) will be the negative of a constant eta( n) multiplied by the dependence of their previous weight on the error of the network, which is the derivative of E in respect to w . The size of the adjustment will depend on n, and on the contribution of the weight to the error of the function. This is, if the weight contributes a lot to the error, the adjustment will be greater than if it contributes in a smaller amount. (5) is used until we find appropriate weights (the error is minimal). So, we "only" need to find the derivative of E in respect to w. This is the goal of the back-propagation algorithm, since we need to achieve this backwards. First, we need to calculate how much the error depends on the output, which is the derivative of E in respect j to O (from (5)).

$$\frac{\partial E}{\partial O_j} = 2\left(O_j - d_j\right) \tag{14}$$

And then, how much the output depends on the activation, which in turn depends on the weights (from (1) and (2)):

$$\frac{\partial O_j}{\partial w_{ji}} = \frac{\partial O_j}{\partial A_j}\frac{\partial A_j}{\partial w_{ji}} = O_j(1 - O_j)x_i \tag{15}$$

And we can see that (from (6) and (7)):

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial w_{ji}} = 2\left(O_j - d_j\right)O_j(1 - O_j)x_i \tag{16}$$

And so, the adjustment to each weight will be (from (7) and (10)):

$$\Delta w_{ji} = -2\eta\left(O_j - d_j\right)O_j(1 - O_j)x_i \tag{17}$$

### 4.18 Implementation of the procedure For the Artificial Neural Network

The back propagation algorithm was used as a part of implementation in building the artificial neural network for controlling industrial process in oil and Gas process. The neural network program was trained using the back propagation algorithm. This is done by adjusting the weight coefficient until the difference between the desired value and he measured value is within accepted limit. Figure 4.17a is the steps involved in implementing the Artificial Neural Network classifier..

### 4.18 Implementation of the procedure For the Artificial Neural Network

The back propagation algorithm was used as a part of implementation in building the artificial neural network for controlling industrial process in oil and Gas process. The neural network program was trained using the back propagation algorithm. This is done by adjusting the weight coefficient until the difference between the desired value and he measured value is within accepted limit. Figure 4.17b is the flow diagram of training the Artificial Neural Network using the back propagation of the error signal..

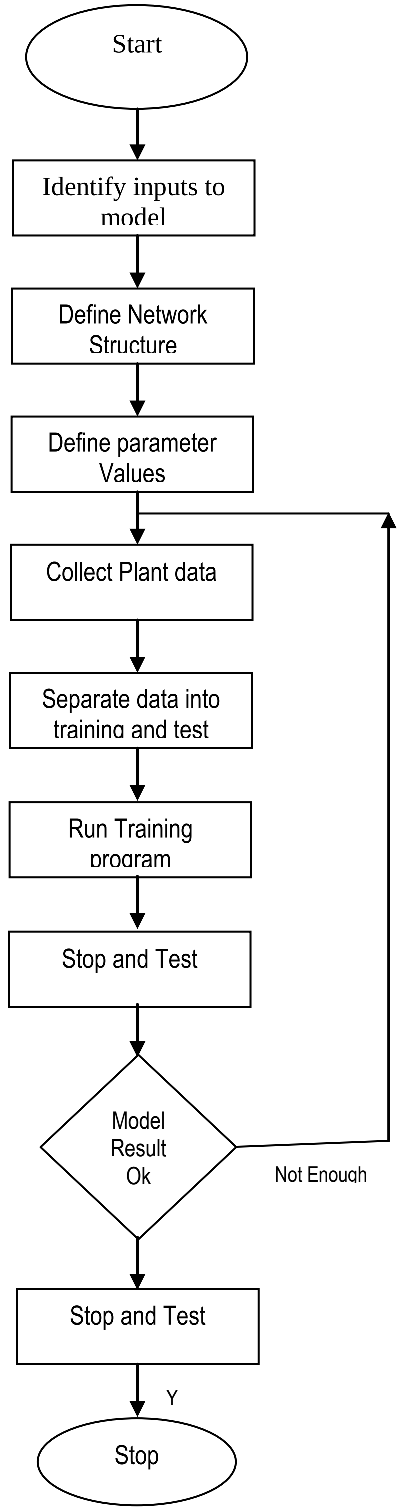



**Figure 4.17b: Flow diagram of the development process of ANN**

### 4.19 The Architecture of the Artificial Neural Network

The architecture for the design of the Work is three inputs, with one hidden layer of seven nodes each and one output layer making a total of eleven nodes. This is shown in figure 4.18a - 18c three neurons receive inputs to the network. Seven neurons are the weights and seven neurons give outputs from the network. The inputs to the network are Process temperature, pressure level, flow-rate and other manipulated variables. The outputs to the Networks are the-seven possible values of the process variables. The topology of the network is 3-7--1.

There are weights assigned with each arrow which represent information flow. These weights are multiplied by the values which go through each arrow to give more or less strength to the signal which they transmit. The neurons on the output layer receive the outputs of both input neurons, multiplied by their respective weights and sum them.

In other to classify the neurons chose weight and set them randomly along the network. Compute outputs for some inputs. Change the weights and see how the behavior of the network changes. Which weights are more critical that is if those weights are changed, the output will change more dramatically. Figure 4.17b is a flow diagram of the network.

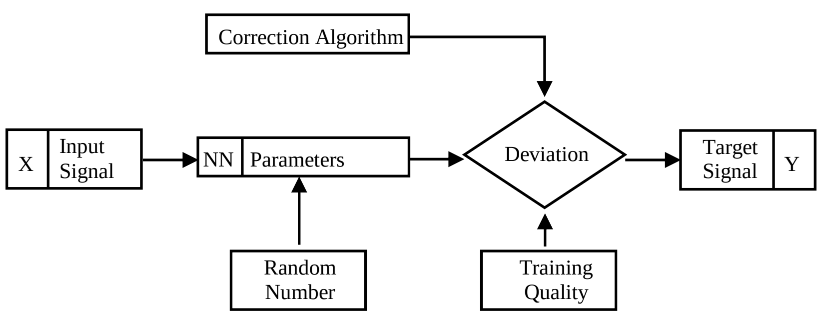



**Fig 4.17b: Flow Diagram of NNC training**

In training the neutral network using the back propagation error signal between the output and the hidden layer we use the expression

$\delta_k = T_k - Y_k$

Where $T_k$ is the target pattern

$Y_k$ is the actual output and $\delta_k$ = back propagation error signal.

We provide the network with example of inputs and outputs we want the network to compute the error which is the difference between actual and expected results and is calculated.

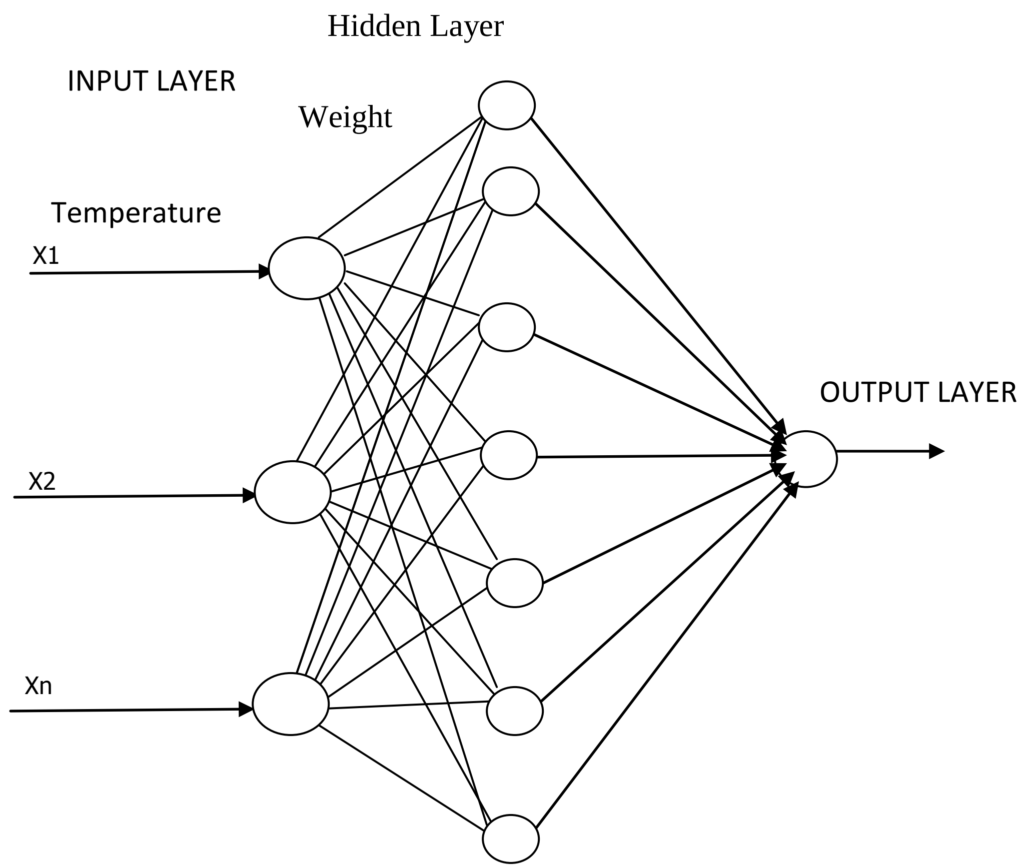
The idea of the back propagation algorithm is to reduce this error until the ANN learns the training data. The training begins with random weights and the goal is to adjust them so that the error will be minimal.

The activation function of the artificial neurons is ANNs implementing the back propagation algorithm is a weighted sum (the sum of the inputs) $X_i$ multiplied by their respective weights ($W_{ji}$):

$$A_j\left(\bar{x}, \bar{w}\right) = \sum_{i=0}^{n} x_i w_{ji} \qquad (18)$$

The activation function used during training is the sigmoid function defined in equation 11

$$\text{Output} = \frac{1}{1 + e^{-sum}} \qquad (19)$$




.

. . ..

. **Fig 4.18a: ANN Classification Diagram for Temperature**

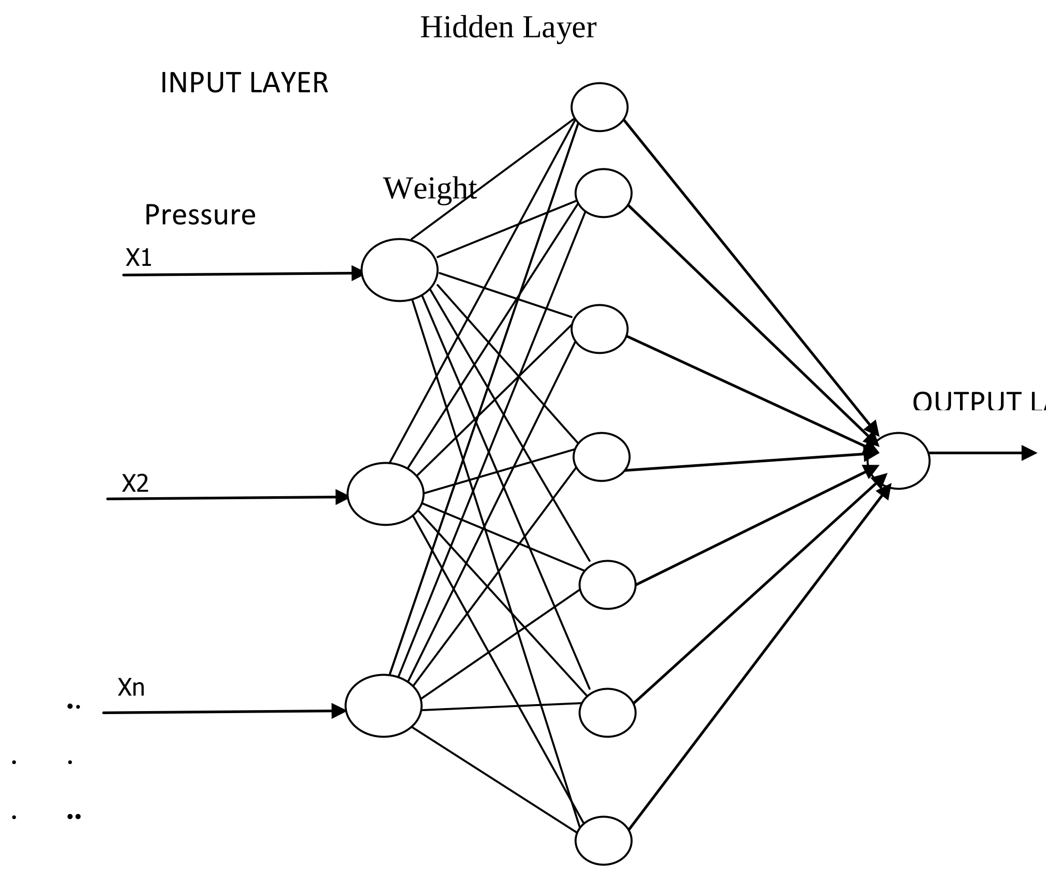



**Fig 4.18b: ANN Classification Diagram for Pressure**

**Hidden Layer**

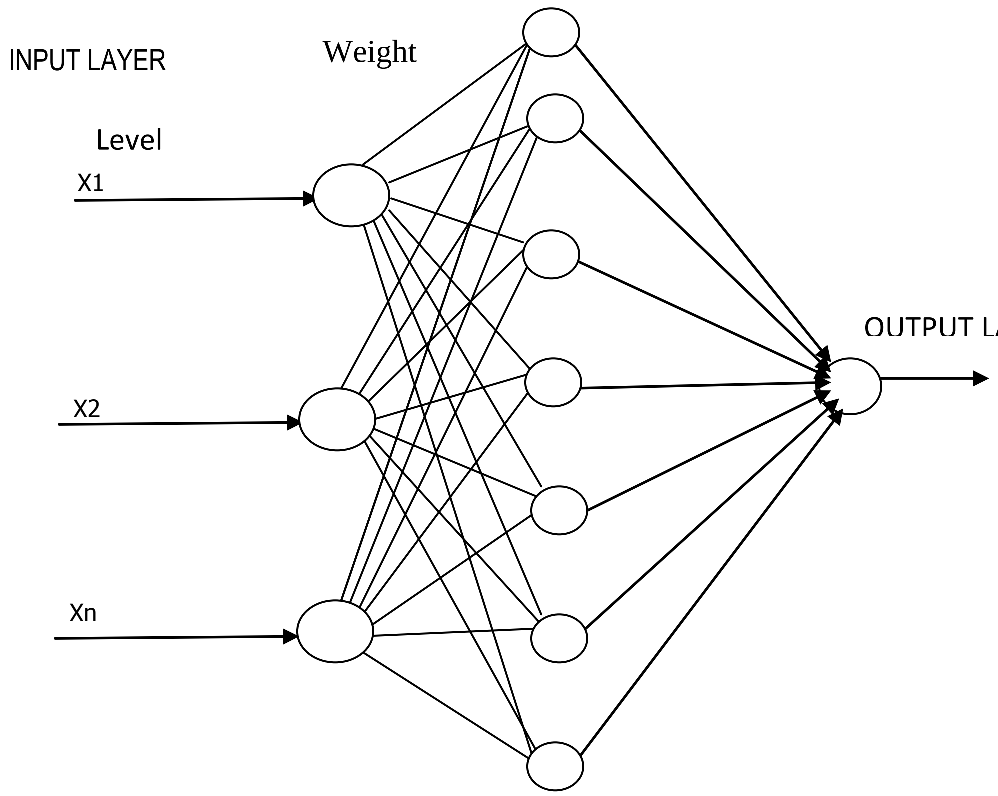



**Fig 4.18c: ANN Classification Diagram for Level**

In training the neutral network using the back propagation error signal between the output and the hidden and joining figures 4.17b and 4.18a-c, the following steps were followed.

Input signal X.

The output from the network is Y,

Reference Output Signal = T

Compare Y with T

The Deviation Value =$\sum(y_j - T_j)$

In training the neutral network using the back propagation error signal between the output and the hidden layer we use the expression

$\delta_k = T_k - Y_k$ (20)

Where $T_k$ is the target pattern

$Y_k$ is the actual output and $\delta_k$ = back propagation error signal.

It is expressed as

$\delta_k = f^1(net_j) \sum \delta_k . N_{kj}$ (21)

Here $f^1$(net) is the derivative of the activation function = f($net_j$)

$$f(net_j) = \frac{1}{1+\exp f(-net_j)} \quad (22)$$

The weights between the inputand hidden layer are

$$\Delta w_{kj}(t+1) = ɳ . \delta_k . y_k + \alpha . \Delta w_{ji}(t) \quad (23)$$

And the weight between the hidden and output layer are updated as

$$\Delta w_k(t+1)\ ɳ . \delta_k . y_k + \alpha \ . \ \Delta w_{ji}(t) \quad (24)$$

Where,

$y_i^i$ and $y_j^i$ = output of the input and the hidden layer.

ɳ is the learning rate, $\alpha$ is the momentum coefficient.

Adaptation of the weight of NNC between the hidden and output layer can be derived as,

$$\Delta w_{kj}^c = ɳ . \frac{\delta E^{ri}}{\delta w_j^c} \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (25)$$

Where,

$E^{ri} = 1/2 \ . \ (y_p - r)^2$

$y_p$ and r are the actual and desired output respectively

Using chain rule equation (25) can be expanded as

$$\Delta w_{kj}^c = ɳ \ . \ \frac{JE^{ri}}{\delta net_k^c} . \frac{\delta net_k^c}{\delta w_{kj}^c} \quad (26)$$

So $\Delta w_{kj}^c = ɳ . \delta_k^c \ . \ y_j^c$, (27)

$$\delta_k(\text{error signal}) = \frac{\delta E^{ri}}{\delta y_k^c} . \frac{\delta E^{ri}}{\delta net_k^r} = \frac{\delta E^{ri}}{\delta y_k^r} \quad (28)$$

$$\therefore \frac{\delta E^{ri}}{\delta y_{net_j^r}} = \delta_k^c \quad (29)$$

So $\delta k = \delta_j^i \ . \ w_{ji}^i$ (30)

The error between the input and hidden layer of the NNC can be derived as follows

$$\delta j = \frac{\delta E^{ri}}{\delta net_j^c} = \frac{\delta E^{ri}}{\delta y_k^c} . \frac{\delta y_j^c}{\delta net} \quad (31)$$

-Using chain rule

$$\frac{\delta E^{ri}}{\delta y_j^c} = \sum \frac{\delta E^{ri}}{\delta net_k^c} \quad . \quad \frac{\delta net_k^c}{\delta y_i^c} \tag{32}$$

$$\delta_l^c = N_j^c \ . \ (1 - y_j^c) \ . \ (\sum \delta_k^c \ . \ w_{kj}^c) \tag{33}$$

Where,

$y_p$ and $y_i$ are the output of the actual plant and neural network controller (NNC)

$$C = {}^{1}/_{2} \ . \ (y_p - y_i)^2 \tag{34}$$

$\eta$ and $\alpha$denote the Learning rate and momentum term respectively.

In training the network we initialize the weight to 0 (w → 0)

Let the Learning rate = 0.02

Net = new netlin([-11], 1, [0.1], 0.02);

Net .iw(1,1) = [00];

Net.biasconnect=0; Net.trainparam.epoch= 1;

To train the network

P1 = (1)

P = (2,3,4);

T1 = (3, 5, 6)

Net = train (net, P, T, P1);

W = [1,2]

Net .iw[1, 1)]= [1, 2];

To Simulate the N/W

P = (12,34)

A = Sim( Net, $P_1$)

The training data is displayed as shown in figure 4.18d depicting the Input and Output to the plant.

The upper curve of figure 4.18d shows the random plant Input steps of random length and width used for the training of the Neural Network model. This was done by sending random plant inputs-steps of random heights and width, while the one below is the Output of the plant. The error curve in figure 4.18e is the difference between the plant output and Neural Network Controller model. A set of data was created that indicated the behavior of the plant response to a characteristic input signal.

Once the training portion of the model is completed, it needs to be verified with the training results. A selection of similar data was created for testing the trained network.
The network was trained with C and Visual Basic programming Languages and normal PC

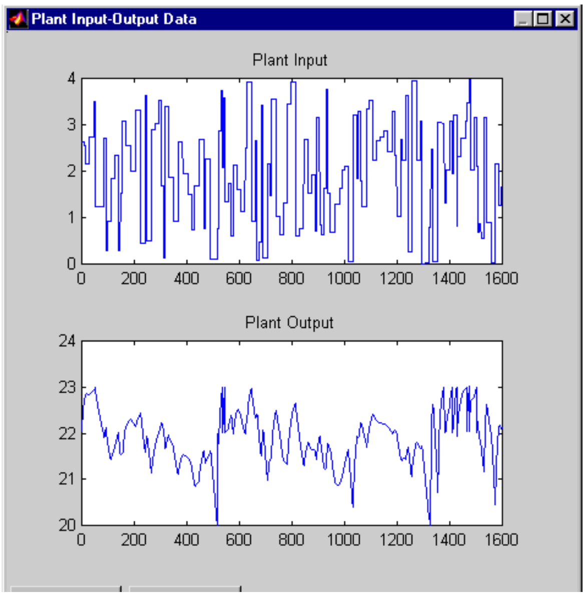



**Figure 4.18d: Input and Output Data (signal) of the plant**

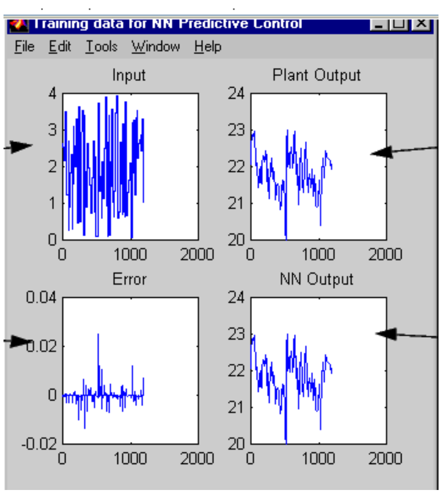



**Figure 4.18e: Signal of the NNC Training Data**

### 4.20 Firing Rule of Neural Network

In training the ANN for classification a range of values is selected and passed through the network. Any value that does not give correct result is logic error. Then reset the software. A table of the parameter is built each time the software is fine tuned. The neural network is used as classification medium to determine which one should fire. Figure 4.19 shows the firing arrangement. To determine the one that will fire

Send input X, (X can be discrete or continuous value)

X passes through synaptic weight W

The network multiplies X and W resulting to XW

The sum of XW is stored in S

S passes through output through the activation function F(S)

The activation function can be a step or threshold type that passes through logical 1

If S>0 or S<0, A positive or negative bias can be introduced to alter threshold value. If the threshold is strong enough, it fires

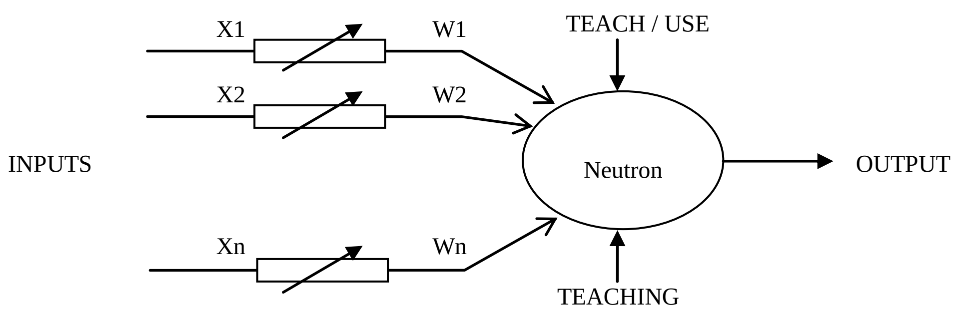



**Fig 4.19: Firing rule of Neural Network**

#### 4.20.1 Learning in ANNs .

There are two types of learning algorithm: Supervised and unsupervised. Learning used here is unsupervised learning. In unsupervised learning, only input stimuli are shown in the network.

The network is self-organizing (organizes itself internally so that each hidden processing elements responds strategically to a different set of input stimuli or groups of stimuli. No knowledge is supplied about which classification (outputs) are correct and those that the network derived may or may not be meaningful to the network developer.

However, setting model parameter can control the number of categories into which the network classifies the inputs. Regardless, a human must examine the final categories to assign meaning and determine the usefulness of the results. Examples of this type of learning are adaptive Reasoning theory (ART) of Kohonem self-organizing feature maps.

#### 4.20.2 LEARNING PROCESS

1. Compute temporary output
2. Compare output with desired result
3. Adjust the weight or repeat the power

When existing outputs, are available for comparison, the learning process starts by setting the connection weights, either by rules or randomly. The difference between the actual output Y or $Y_T$ and the desired out Z for a giving set of inputs is an error delta. The objective is to minimize the delta and this is done by adjusting the network's weight. The key is change the weight in the right direction. The process of adjusting the weight is called learning or training.

#### 4.20.3 Result of Neural Network Training & Testing

The feedback neural network is a back propagation, unsupervised-learning network. The network for each process unit selected for the classification neural network had one hidden layers of 7 neurons, in addition to one input layers of three neurons and one output layer. Network training was performed on 10 data sets and testing with 5 data sets. For the neural network to have the desired accuracy, the network was trained with a training tolerance of 5% of the set point of the desired process variable. The learning rate was close to the expected set point of the temperature, pressure, flow rate and level. The expected results from the ANN classification are presented below. Ideally, once the ANN has been trained and experiences the variables, someone may expect it to identify each one of them accurately.

### 4.20.5 Training Data

Tables 4.6a, 4.6b, 4.6c and 4.6d show the training data for the Artificial Neural Network. The experimental data are the different reading the plant. Since the nine sub-units are the same and each of them has either two or three variables, we decided to use two sub-units. One with three variables and the other with two variables

Table 4.6a: Well stream heater (U1)

Temperature

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 115.0 | 122 | 158 | 166 | 150 |
| 115.5 | 122.5 | 158.5 | 166 | 150 |
| 116.0 | 122.4 | 159 | 166.5 | 150 |
| 116.5 | 122.5 | 159 | 166.5 | 150 |
| 115.0 | 123.0 | 159.5 | 166.5 | 150 |
| 114.0 | 123.0 | 159.5 | 166.5 | 150 |
| 114.0 | 123.5 | 160 | 167 | 150 |
| 114.5 | 122.2 | 160 | 167.5 | 150 |
| 112.0 | 122.6 | 160.5 | 167.8 | 150 |
| 112.0 | 123.5 | 160.5 | 168 | 150 |
| 116.0 | 123.5 | 160.7 | 168 | 150 |
| 117.0 | 124 | 160.8 | 168.5 | 150 |

PRESSURE

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 84.0 | 85 | 95 | 96 | 90 |
| 84.0 | 85 | 95 | 96 | 90 |
| 84.0 | 85.5 | 95.2 | 96.2 | 90 |
| 84.5 | 85.8 | 95.2 | 96.2 | 90 |
| 84.6 | 88.0 | 95.5 | 96.4 | 90 |
| 84.8 | 86.0 | 95.5 | 95.4 | 90 |
| 83.0 | 86.4 | 95.6 | 96.8 | 90 |

| | | | | |
|---|---|---|---|---|
| 83.5 | 86.6 | 95.6 | 97.0 | 90 |
| 83.5 | 86.8 | 95.8 | 97.0 | 90 |
| 82.0 | 86.8 | 95.8 | 97.5 | 90 |
| 82.0 | 87.0 | 95.8 | 97.5 | 90 |

LEVEL

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 806 | 807 | 893 | 895 | 850 |
| 806 | 807.5 | 893 | 895 | 850 |
| 805 | 807.5 | 893.2 | 895.5 | 850 |
| 805.5 | 807.8 | 893.2 | 895.8 | 850 |
| 805.5 | 807.8 | 893.4 | 896 | 850 |
| 804.0 | 808.0 | 893.5 | 896.5 | 850 |
| 804.4 | 808.0 | 893.5 | 896.5 | 850 |
| 803.0 | 808.8 | 894.0 | 896.5 | 850 |
| 802.0 | 809.0 | 894.5 | 897 | 850 |

-

**Table 4.6b: (U2) DUAL INLET MONIFOD TEMPERATURE**

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 42.0 | 42.5 | 48 | 49 | 45 |
| 42.0 | 42.5 | 48 | 49 | 45 |
| 41.5 | 42.8 | 48.2 | 49.5 | 45 |
| 41.5 | 42.8 | 48.2 | 49.5 | 45 |
| 41.8 | 42.2 | 48.4 | 49.8 | 45 |
| 41.8 | 42.2 | 48.4 | 49.8 | 45 |
| 40.0 | 41.8 | 48.8 | 50.0 | 45 |
| 40.0 | 41.8 | 48.8 | 51.0 | 45 |
| 40.5 | 42.0 | 48.9 | 51.5 | 45 |
| 40.5 | 42.2 | 48.9 | 51.5 | 45 |

PREASURE

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 96 | 97 | 108 | 109 | 102.8 |
| 95.5 | 97 | 108.2 | 109 | 102.8 |
| 95.5 | 96.4 | 108.2 | 109.5 | 102.8 |
| 95.8 | 96.4 | 108.5 | 109.5 | 102.8 |
| 95.8 | 96.6 | 108.5 | 109.8 | 102.8 |
| 94.0 | 96.8 | 108.8 | 109.8 | 102.8 |
| 94.0 | 96.8 | 108.8 | 109.8 | 102.8 |
| 93.5 | 98 | 109.0 | 110 | 102.8 |
| 93.5 | 98 | 109.0 | 110 | 102.8 |
| 93.0 | 97.5 | 109.0 | 110 | 102.8 |
| 93.0 | 97.5 | 109.2 | 110.5 | 102.8 |

**Table 4.6c: INLET SEPERATOR (U3)**

TEMPERATURE

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 42 | 42.5 | 48 | 49 | 45 |
| 41.9 | 42.5 | 48.2 | 49.2 | 45 |
| 41.8 | 42.4 | 48.2 | 49.2 | 45 |
| 41.6 | 42.4 | 48.4 | 49.4 | 45 |
| 41.6 | 42.2 | 48.4 | 49.4 | 45 |
| 41.4 | 42.2 | 48.6 | 49.8 | 45 |
| 41.4 | 42.0 | 48.6 | 49.8 | 45 |
| 40.0 | 42.0 | 48.8 | 49.8 | 45 |
| 40.0 | 41.8 | 48.8 | 50.0 | 45 |
| 39.0 | 41.6 | 49.0 | 50.0 | 45 |
| 39.0 | 41.4 | 49.0 | 50.5 | 45 |

PREASURE

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 96 | 97 | 108 | 109 | 102 |
| 95.8 | 96.8 | 108.2 | 109.2 | 102 |
| 98.8 | 96.8 | 108.2 | 109.2 | 102 |
| 95.6 | 96.6 | 108.4 | 109.4 | 102 |
| 95.6 | 96.6 | 108.4 | 109.4 | 102 |
| 95.4 | 96.4 | 108.6 | 109.6 | 102 |
| 95.4 | 96.4 | 108.6 | 109.6 | 102 |
| 95.2 | 96.2 | 108.8 | 109.8 | 102 |
| 95.2 | 96.1 | 108.9 | 108.8 | 102 |
| 95.0 | 96.1 | 108.9 | 110 | 102 |
| 95.0 | 96.0 | 109 | 110 | 102 |

(U3)

LEVEL

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 246 | 247 | 273 | 274 | 260 |
| 246 | 247 | 273.2 | 274.1 | 260 |
| 245.8 | 246.8 | 273.2 | 274.2 | 260 |
| 245.8 | 246.8 | 273.4 | 274.2 | 260 |
| 245.6 | 246.6 | 273.4 | 274.4 | 260 |
| 245.6 | 246.6 | 273.6 | 274.4 | 260 |
| 245.4 | 246.4 | 273.6 | 273.8 | 260 |
| 245.4 | 246.4 | 2738 | 274.8 | 260 |
| 245.2 | 246.2 | 273.8 | 275 | 260 |
| 245.2 | 246.2 | 273.8 | 275 | 260 |
| 245.0 | 2462 | 274.0 | 275 | 260 |
| 245.0 | 246.0 | 274.0 | 275 | 260 |

**Table 4.6d: GAS COOLER (U4)**

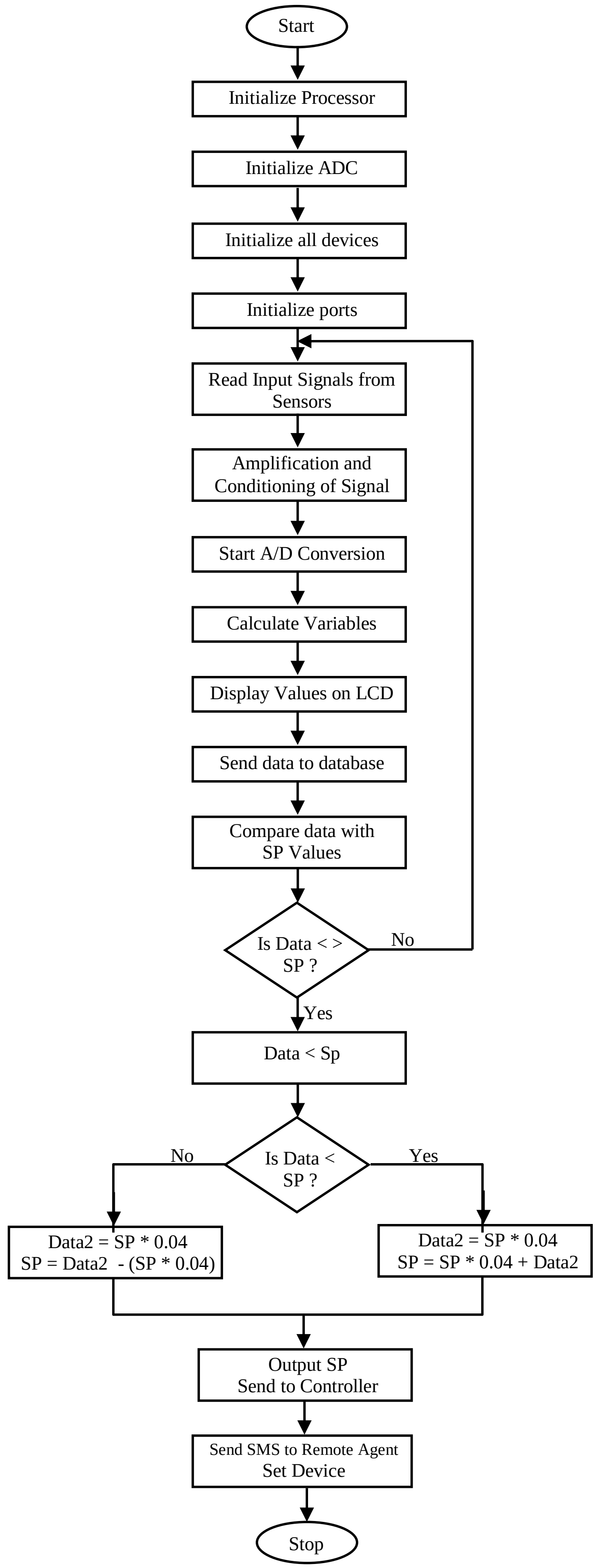
TEMPERATURE

| X1 | X2 | X3 | X4 | SP |
|---|---|---|---|---|
| 22 | 23 | 27 | 28 | 25 |
| 21.8 | 23 | 27.1 | 28.1 | 25 |
| 21.8 | 22.8 | 27.2 | 28.1 | 25 |
| 21.6 | 22.8 | 27.2 | 28.2 | 25 |
| 21.6 | 22.6 | 27.4 | 28.2 | 25 |
| 21.4 | 22.6 | 27.4 | 28.4 | 25 |
| 21.4 | 22.4 | 27.6 | 28.4 | 25 |
| 21.2 | 22.2 | 27.6 | 28.6 | 25 |
| 21.2 | 22.2 | 27.8 | 28.6 | 25 |
| 21.0 | 22.1 | 27.8 | 28.8 | 25 |
| 21.0 | 22.1 | 27.9 | 29 | 25 |

[Source SPDC Log Book]

**4.21 The Control Flow diagram of Entire System**

Figure 4.20a shows the monitoring and control flowchart of the entire system. The steps involved include initialization of the processors and all the ports. Signal amplification and condition. The variable is read converted and displayed on the LCD,




# Hint

If Data < SP,

Compute 4% of SP

Increase SP by 4%,

wait for Data = SP

If Data > SP,

Compute 4% of SP

Decrease SP by 4%

wait for Data = SP

**Figure 4.20a: The Monitoring and Control Flowchart**

Figure 4.20b is the block diagram form how the program for the microcontroller with three loops is structured to service each of the cases or loops. For each of the case/loop, you select the variable, condition the analogue signal value; convert the signal to its digital value using ADC. The processor reads the value, then determines the K value and performs action corresponding to K value. Let's assume that the micro controller services every loop in 20ms.

That means every 20millisecond a new loop is updated. Each loop sets an update after 60seconds for 3 variables and after 80seconds for 4 variables. The microcontroller services each loop at regular intervals. Each loop is controlled by a different procedure. For instance the control of temperature in the stream well heater will be different from control of pressure in the well head. Note that the executive program sits on the loop waiting for user command from either the keyboard or the GSM in the remote part. The GSM and the keyboard are connected to the interrupt inputs. When the microcontroller receives an interrupt, it goes to a procedure which determines whether it is time to service the next loop by counting the interrupt.

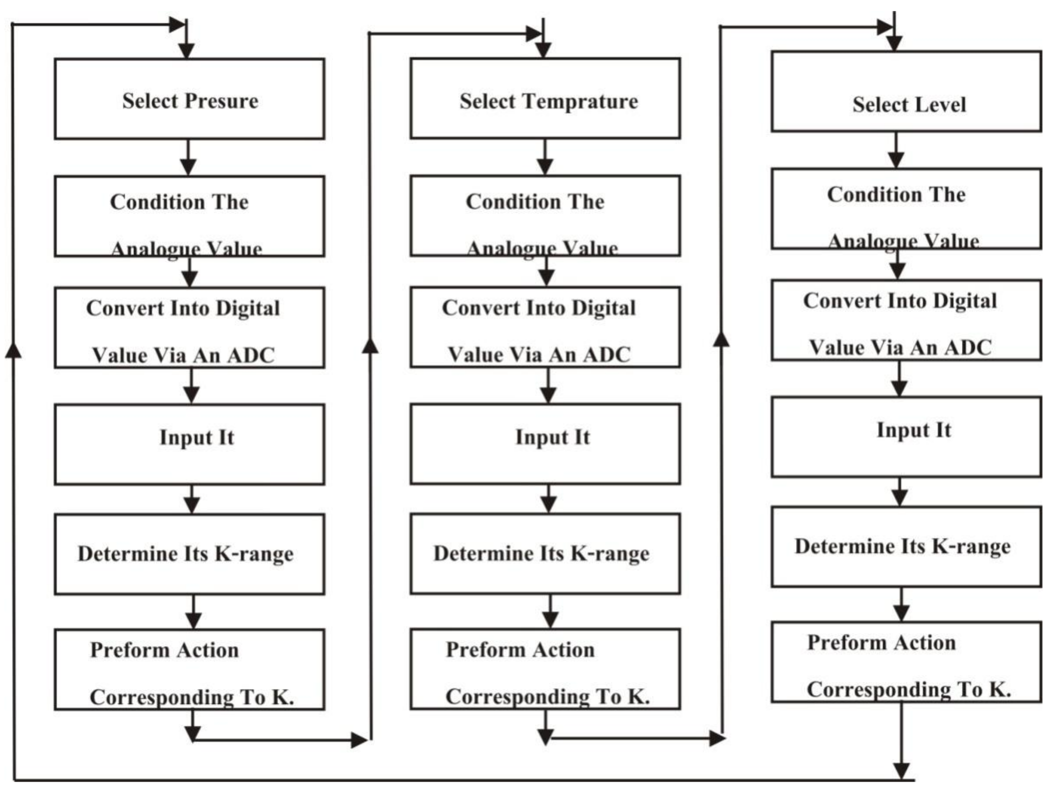



Figure 4.20b Block Diagram Of ASM Control Chart.[Author]

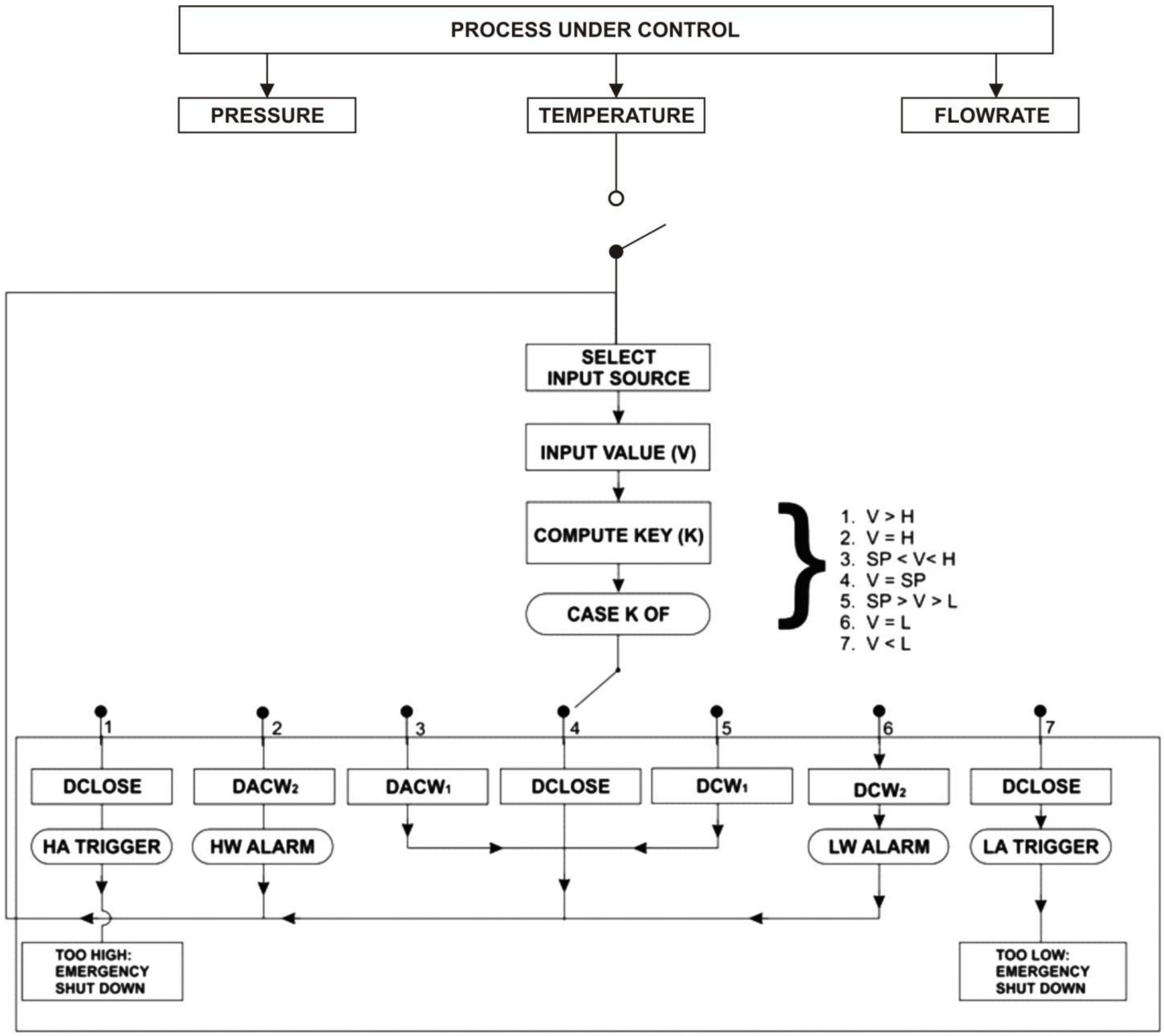



**Fig 4.21: ASM Chart for the Process controlled by each of the nine microprocessors.**

### 4.21.1 The process control Algorithm

Figure 4.21 is the Algorithmic state machine chart showing the control of the system. The process control begins with the initialization of all the ports, time and unmasking and enabling of interrupt. The executive program waits for user command from either the keyboard or the GSM line remote part.

The ASM control chart is the procedure that the processor uses to monitor and control multiple process variables. An input source is selected which may be, Temperature, Pressure, or Level value and is read by the processor and the key is computed.

They are seven keys as indicated in ASM control chart namely:

1. Value (V) > High (H)
2. Value (V) = High.
3. SP < Value < H
4. V = SP
5. SP  > V > L
6. V = L
7. V < L

The case that will be selected will depend on the key.

If V > H, case 1 is selected, and the device will be closed and the high Alarm device will be triggered and there will be Emergency shutdown.

If Value = High, case 2 is selected, the Device will be set Anti clockwise twice low warning Alarm

If SP < Value < H case 3 is selected and device is set anti clockwise once.

If V = SP, case 4is selected, the Device is close.

If SP > V > L, case 5 is selected, the Device is turned Clockwise once and returned back to selected input source.

If V = L, case 6 is selected, that is value is equal to low. The device is turned Clockwise twice; a low alarm working is sounded

If V< L case 7 is selected (V>L) the DLA Trigger is sounded and there will be an emergency short down of the plant.

For any of the cases selected, the block diagram in figure 4.20b will be executed. Each of the case is a loop.

### 4.22 The Complete Circuit Diagram

The complete circuit diagram of the proposed Remote- based industrial process monitoring and control system is given in figure 4.22. The hardware of the system mainly includes a 8-bit microcontroller chip, a GSM module, and RS232 interfaces. The microcontroller is interfaced with different sensors for sensing different variables. Temperature sensor detects the temperature, Pressure sensor sense pressure Level sensor and flow rate sensor sense flow. The analog data from sensors are amplified, conditioned and converted to digital value using A/D converter. The data is provides to the Neural microcontroller for analysis, classification and any other necessary actions. In case of emergency condition an alarm is raised depending upon this data and an SMS is send to the user's mobile.

Real time clock/calendar helps in proper day-to-day recording of data. The measured values and the state of the devices are displayed on the LCD. The GSM module which is the most important part of this system is interfaced with the Microcontroller using a RS232 interfaces. The modules act like an interface between the controller and GSM network. The GSM module must have a SIM (Subscriber Identity Module) card to make the network identify the user. The microcontroller communicates with the GSM module using the AT commands.

These AT commands are used to send and receive SMS. The programming code for the microcontroller is written in high level language. The use of a personal computer provides for the storage of archival data. The system informs user about any abnormal conditions via SMS from the GSM Module to the users mobile and actions are taken accordingly by the user.

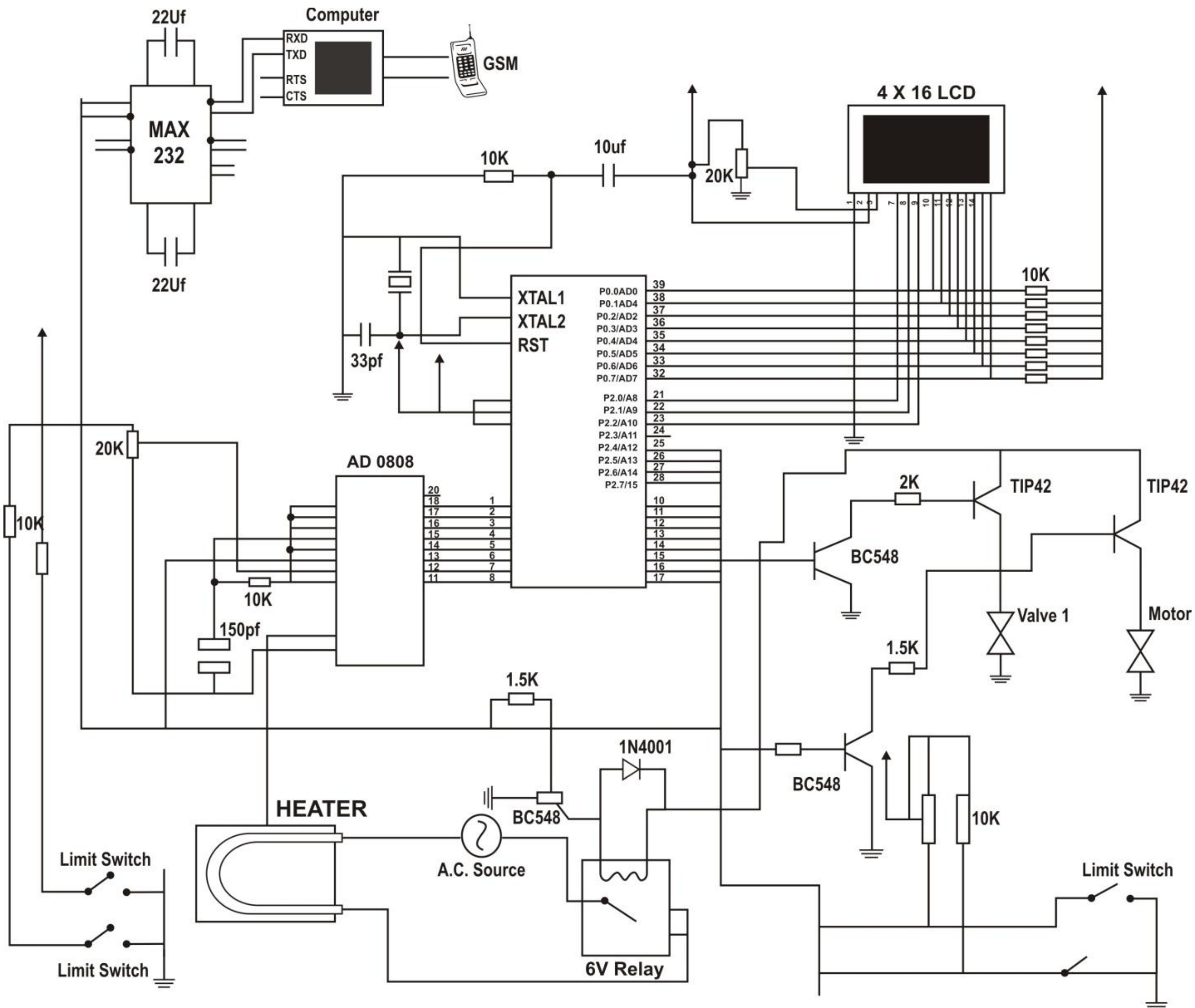



**Figure 4.22: Diagram of the Temperature Control Subsystem.**

### 4.23. The Operation of Microcontroller System

The micro-controller used in this work is the AT89C52 with provisions for input and output embedded in it. It consists of timers, Analog to Digital Converters (ADCs), Universal Synchronous Asynchronous Receiver Transmitter (USART), etc. It is an 8-bit microcontroller with flash program memory and Electrically Erasable Programmable Read Only Memory (EEPROM). It obeys 83-instructions which include byte operations, bits operations and branching. It has four 8-bits ports which can be bit-wise or byte-wise addressed.

#### 4.23.1 Microcontroller Interfacing

The AT89C52 which is a member of Intel 8051 family is Harvard architecture, single chip microcontroller which was developed by Intel in 1980 for use in embedded systems. Intel's original versions were popular in the 1980s and early l990s, but has today largely been superseded by vast range of faster and/or functionally enhanced 8051 compatible devices manufactured by more than 20 independent manufacturers including Atmel, Infineon Technologies (formerly Siemens AG), Maxim Integrated Products (via its Dallas Semiconductor subsidiary), NXP (formally Philips Semiconductor), Nuvoton (formerly Winbond), ST Microelectronics, Silicon Laboratories (formerly Cygnal), Texas Instruments and Cypress Semiconductor. Intel's official designation for the 8051 family of microcontrollers is MCS 51. Intel's original 8051 family was developed using NMOS technology but later versions, identified by a letter C in their name (e.g.80C51) used CMOS technology and were less power-hungry than their NMOS predecessors. This made them more suitable for battery-powered devices.

#### 4.23.3 Overview of the 8051 Microcontroller

The features of the 8051 microcontroller are summarized below:

- 128 bytes Ram
- 4k bytes of on-chip Rom
- Two timers
- One serial port
- 64kbytes external code memory space
- 210 bit addressable locations
- Four ports (each 8 bit wide)

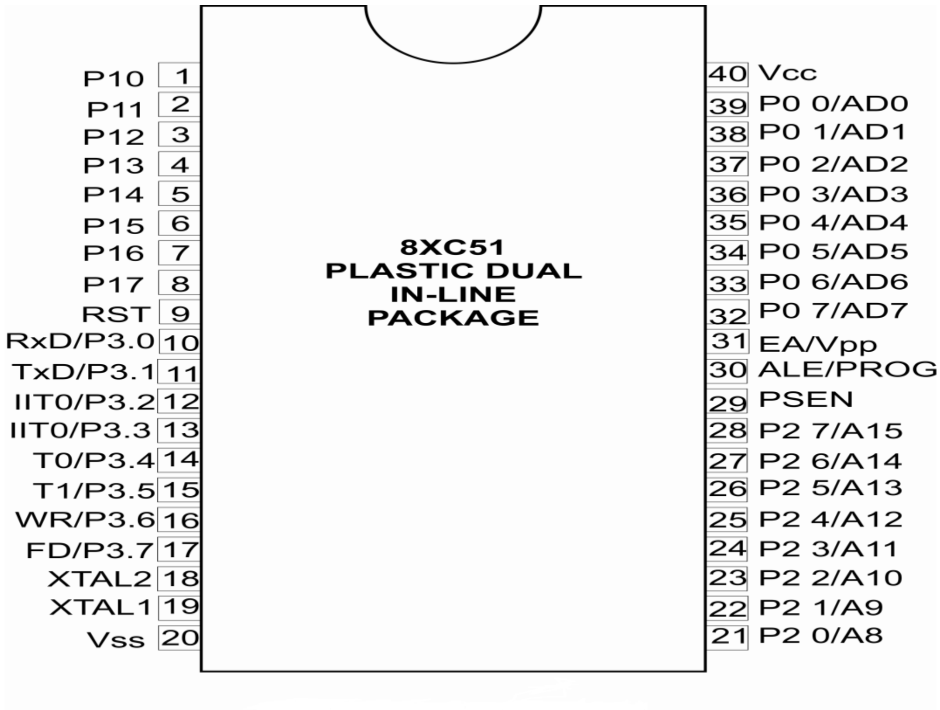



Figure 4.22b: Pin Diagram of 8051(http://www.atmel.com)

Examining figure 4.22b, note that 8051 have total of 40 pins. Out of the 40 pins a total of 32 pins are set aside for the four ports P0, P1, P2 and P3 where each port takes 8 pins. The rest of the pins are designated as Vcc, GND, XTAL1, XTAL2, RST, FA, PSEN and ALE.

### 4.22.4 Port 0:

Port 0 occupies a total of 8 pins (pins 32-39) it can be used for input or output. To use the pins of port 0 as both input and output ports each pin must be connected to a 10k-ohm pull up resistor.

Port 0 is also designated as AD0-AD7, allowing it to be used for both address and data. When connecting 8051 to an external memory port 0 provides both address and data.

### 4.22.5 Port 1:

Port 1 occupies a total of 8 pins (pins 1 through 8). It can be used as input or output. This port does not need any pull up resistors since it already has pull up resistors internally. No alternate functions are designed to port 1 pins.

### 4.22.6 Port 2:

Port 2 occupies a total of 8 pins (pins 21 through 28) it can be used as input or output. Like port 1, port 2 does not need any pull up resistor since it had already pull up resistor internally.

The dual role of port 2 is that it provides 16 bit address for external memory along with port 0. As shown in figure 4.22a, port 2 is also designated as A8-A15, indicating its dual function.

### 4.22.7 Port 3:

Port 3 occupies a total of 8 pins, pins 10 through 17. It can be used as input or output. Port 3 does not need any pull up resistor just as P1 and P2 did not. Port 3 has the additional function of providing some extremely important signals such as interrupts. Table 4.7 provides these alternate functions of port 3.

**Table 4.7: Port 3 Alternate Function**

| P3 Bit | Function | Pin |
|---|---|---|
| P3.0 | RxD | 10 |
| P3.1 | TxD | 11 |
| P3.2 | $\overline{\text{INT0}}$ | 12 |
| P3.3 | INT1 | 13 |
| P3.4 | T0 | 14 |
| P3.5 | T1 | 15 |
| P3.6 | $\overline{\text{WR}}$ | 16 |
| P3.7 | RD | 17 |

#### 4.22.8. PSEN (PROGRAM STORE ENABLE):

Program store enable is an output pin. It is a control signal that enables external (program memory. It is usually connected to EPROM's output enable (OE).

#### 4.22.9. ALE (ADDRESS LATCH ENABLE):

ALE is an output pin and is active high. The port multiplexes the address and data. ALE is used to de-multiplexing the data and address. It is designated pin 30.

#### 4.22.10. EA (EXTERNAL ACCESS):

EA is pin 31 in DIP packages. It is an input pin and must be connected to either Vcc or GND. EA pin must be connected to GND to indicate that the code is stored externally. And if it is connected to Vcc it indicates that code to be executed is stored on the on-chip rom.

#### 4.22.11. RST:

Pin 9 is RESET pin. It is an input and is active high (normally low). Upon applying a high pulse to this pin, the Microcontroller will reset and terminate all activities. This is often referred as power-on-reset. It will set program counter to 0 and data in all the registers will be lost. In order for the reset input to be effective, it must have minimum duration of two machine cycles. Figure 4.22c shows how to connect RST pin to the power on reset circuitry.

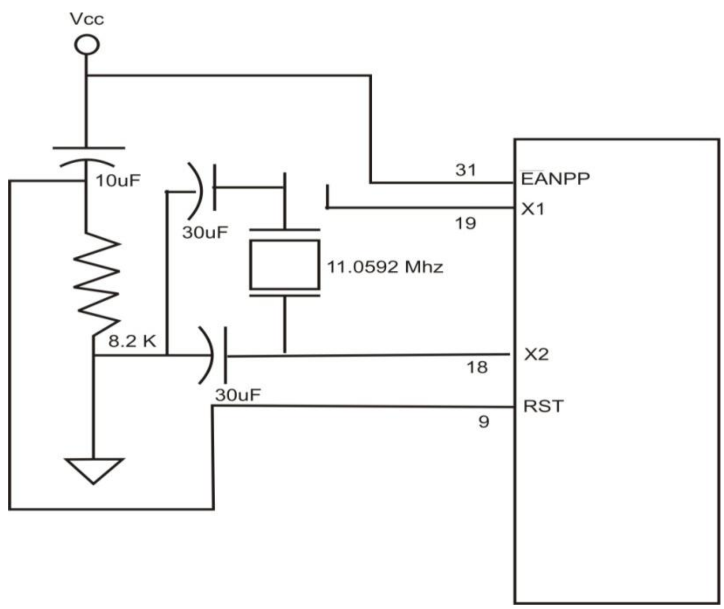



**Figure 4.22c: Power on reset circuit.**

### 4.22.12 Vcc:

Pin number 40 provides supply voltage to the chip. The voltage source is +5V.

### 4.22.13 GND:

Pin number 20 is ground.

### 4.22.15 XTAL1 AND XTAL2:

The 8051 has on-chip oscillator but requires an external clock to run it. Most often a quartz crystal oscillator is connected to inputsXTAL1 (pin 19) and XTAL2 (pin 18), the crystal oscillator connected also needs two 30 pF capacitors. One side of each capacitor is connected to ground, the normal crystal frequency is 12 MHZ.

## 4.23 Memory Organization

Most microcontrollers are rarely used as the CPU in the computer systems; instead they are employed as the central component in control oriented design. There is limited memory and there is no disk drive or disk operating system. The control program must reside in ROM. For this reason, 8051 implements a separate memory space for program and data. The details are as under.

### 4.23.1 RAM Memory Space

There are I28 bytes of RAM in the 8051. The 128 bytes of RAM inside the 8051 are assigned addresses 00 to 7FH. These 128 bytes are divided into three different groups as follows.

1. A total of 32 bytes from locations 00 to 1F hex are set aside for register banks and the stack.
2. A total of 16 bytes from locations 20H to 2FH are set aside for bit addressable read/write memory.
3. A total of 80 bytes from locations 30H to 7FH are used for read and write storage. They are widely used for the purpose of storing the data and parameters by 8051 programmers.

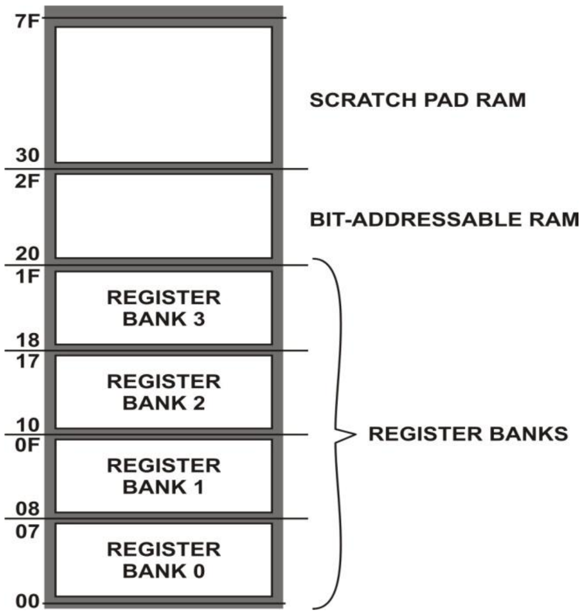



**Fig 4.23: General Purpose RAM**

### 4.23.2 General Purpose RAM:

As shown in figure 4.23 the genera1 purpose RAM has addresses from 30H to 7FH, the bottom 32 bytes from 00H to 2FH can be used similarly. Any location in the general RAM can be accessed freely using the direct and indirect addressing modes.

### 4.23.3 Register Banks:

A total of 32 bytes are set aside for register banks and stacks. These 32 bytes are divided into 4 banks of registers in which each bank has 8 registers, R0-R7. RAM locations from 0 to 7 are set aside for bank 0 of R0-R7, where R0 is RAM location 0, R1 is RAM location 1 and so on. The second bank of registers R0-R7 starts at RAM location 08 and goes to location 0FH. The third bank of R0-R7 starts at memory location 10H and goes to location 10H. Finally, RAM locations 18H to 1FH are set aside for the fourth bank of R0-R7. The following shows how the 32 bytes are allocated into 4 banks. By default these registers are at addresses 00H-07H. The register bank one is not used (generally) as it also serves purpose of stack. This part of RAM can be accessed by name as well as the address.

### 4.23.4 Bit-Addressable RAM

Of the 128 byte internal RAM of the 8051, only 16 bytes are bit addressable. The bit-addressable RAM locations are 20H to 2FH. These 16 bytes provide 128 bits of RAM. They are addressed as 0 to 127 (in decimal) or 00 to 7FH. Therefore the bit addresses 0 to 7 are for the first byte of internal RAM location 20H, and 8 to 0FH are bit addresses of the second byte of RAM location 21H and so on. The last byte of 2FH has bit addresses of 78H to 7FH.

The idea of individually accessing bits through software is a powerful feature of most microcontrollers. Bits can be set, cleared, ANDed, ORed etc with a single instruction.

This part of RAM is bit addressable. This means that we can use the bits address directly. A bit can be set with SETB & cleared with CLR. We can only access this directly by address. There are no special names for this part of RAM. The address of a bit is independent of the byte address.

Table 4.7b provides a list of single bit instructions. Notice that the single bit instructions use only direct addressing mode. There is no indirect addressing mode for single bit instructions.

**Table4.7b: Single Bit Instructions**

| **Instruction** | **Function** |
|---|---|
| Set B | Set the bit (Bit=1) |
| CLR | Clear the bit (bit = 0) |
| CPL | Compliment the bit |
| JNB | Jump to target bit (bit = 0) |
| JBC | Jump to target bit (bit = 1) Clear the bit |

## 4.24 MAX 232 CHIP

A line driver such as the MAX232 chip is required to convert RS232 voltage levels to TTL levels, and vice versa. 8051 has two pins that are used specifically for transferring and receiving data serially. These two pins are called TxD and RxD and are part of the port 3 group (P3.0 and P3.1). These pins are TTL compatible; therefore, they require a line driver to make them RS232 compatible.

We need a line driver (voltage converter) to convert the R232's signal from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ±7.5V) from a single +5V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0V to +5V range, as power supply design does not need to be more complicated just for driving the RS-232 in this case. The receivers reduce RS-232 inputs (which may be as high as ±25V), to standard 5V TTL levels. These receivers have a typical threshold of 1.3V and a typical hysteresis of 0.5V. Figure 24a is the PIN diagram of MAX232 while figure 4.24b is the circuit diagram. Figure 4.24c is the connection with serial port.

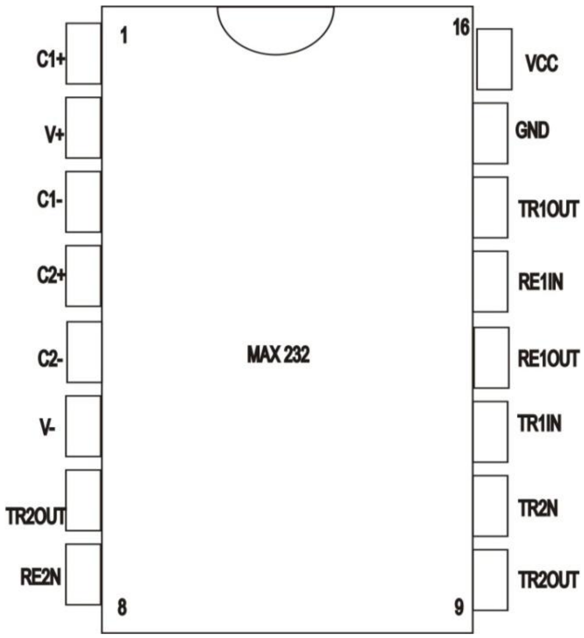



Figure 4.24a: Pin Diagram of  Max232[http://freecircuitdiagram.com/2009/04/24/ttl-rs232-level-converter-using-max232-ic/]

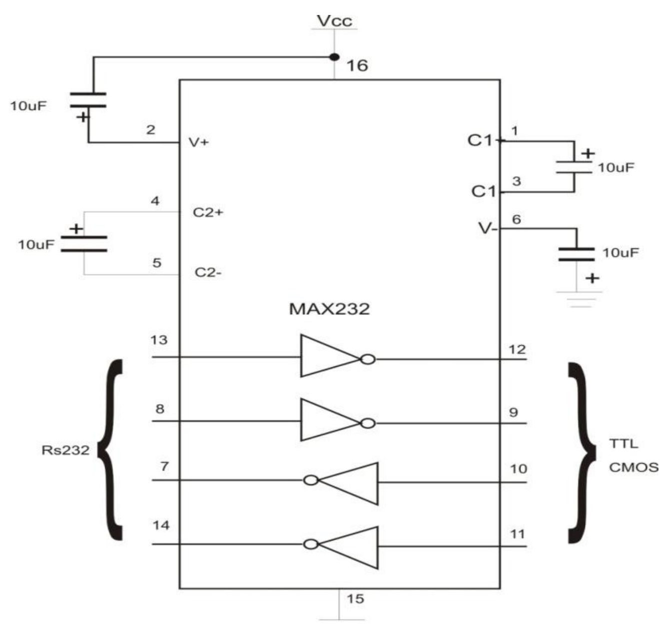



**Fig 4.24b: Circuit Diagram of Max**

**[**https://www.google.com/search?q=max232+pin+diagram&ie=]

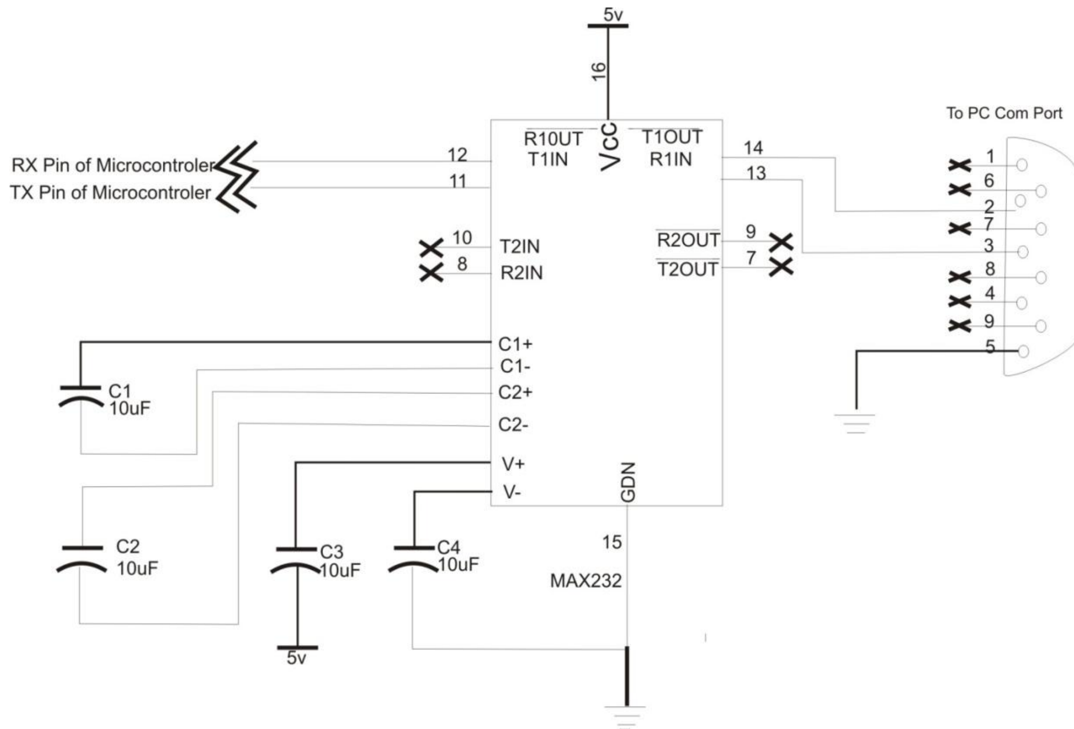



**Figure 4.24c: Connection with Serial Port.**

**[**http://microcontroller.circuitlab.org/2012/02/schematic-max-232-interfacing-with.html]

### 4.24.1 Voltage Levels

It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15V, and changes TTL Logic 1 to between -3 to -15V, and vice versa for converting from RS232 to TTL. This can be confusing when you realize that the RS232 data transmission voltages at a certain logic state are opposite from the RS232 control line voltages at the same logic state. To clarify the matter, see the table 4.7.

**Table 4.7: Voltage Level**

| **Rs 232 line type and voltage level** | **RS 232 Voltage** | **TTL Voltage to/from MAX 232** |
|---|---|---|
| Data transmission (RX/TX) logic 0 | +3v to +15v | 0v |
| Data transmission (RX/TX) logic i | -3v to -15v | 5v |
| Control signal (RTS/CTS/DTR/DSR) Logic 0 | -3v to -15v | 5v |
| Control signal (RTS/CTS/DTR/DSR) Logic 1 | +3v to +15v | 0v |

Max 232 uses four capacitors and has two set of line driver. Capacitors can have value from 1μF - 10 μF.

To communicate over UART or USART, we just need three basic signals which are namely, RxD (receive), TxD (transmit), GND (common ground). So to interface MAX 232 with any microcontroller (AVR, ARM, 8051, PIC etc) we just need the basic signals. A simple schematic diagram of connections between a microcontroller and MAX232 is shown figure 4.25

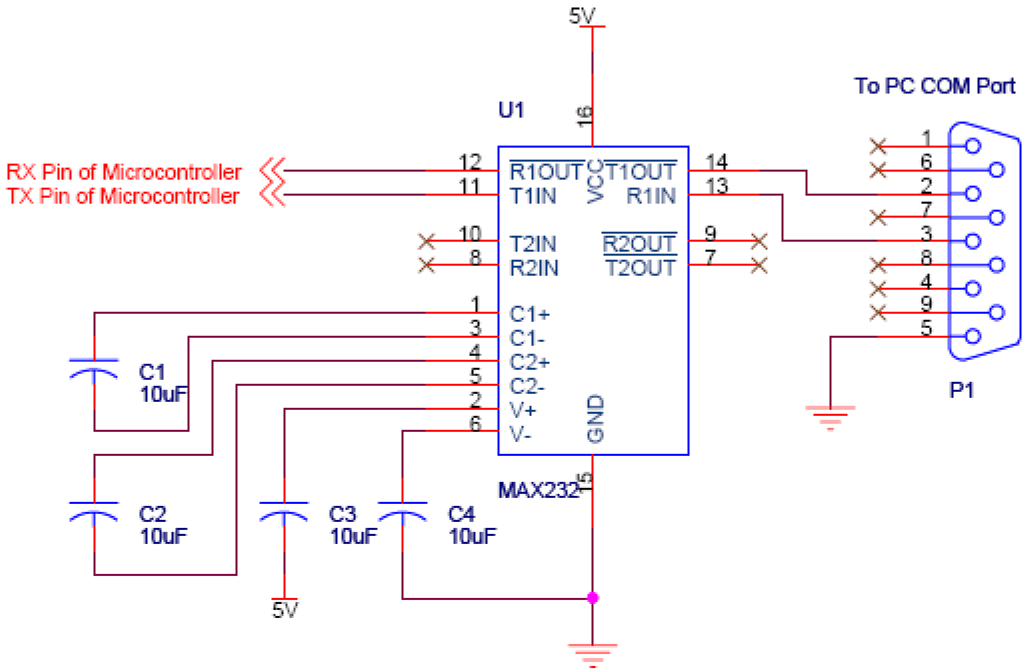



**Fig 4.25: Schematic Diagram of Connections between Microcontroller and MAX 232**

[https://www.google.com/search?q=max232+pin+diagram&ie=utf-8&oe=utf-8]

## 4.25. The Output Parameter

*The output parameters are:*

• opening/closing the valves/pumps for water and/or Gas/Oil, and adjusting their volume in combination;

• Turning energy systems on/off (lights, heating,);

### 4.25.1 The Output Interface

The output interface has to do with the device and circuitry that uses the processed signals from the controller to activate any of the input devices, in other to effect the final control of the plant/field device to achieved the desired value. They include: The LCD, The DAC  Transistor, Multiplexer and the necessary switch .Figure 4.26a is an output circuit that controls the level controller (LC) and the temperature controller (TC)

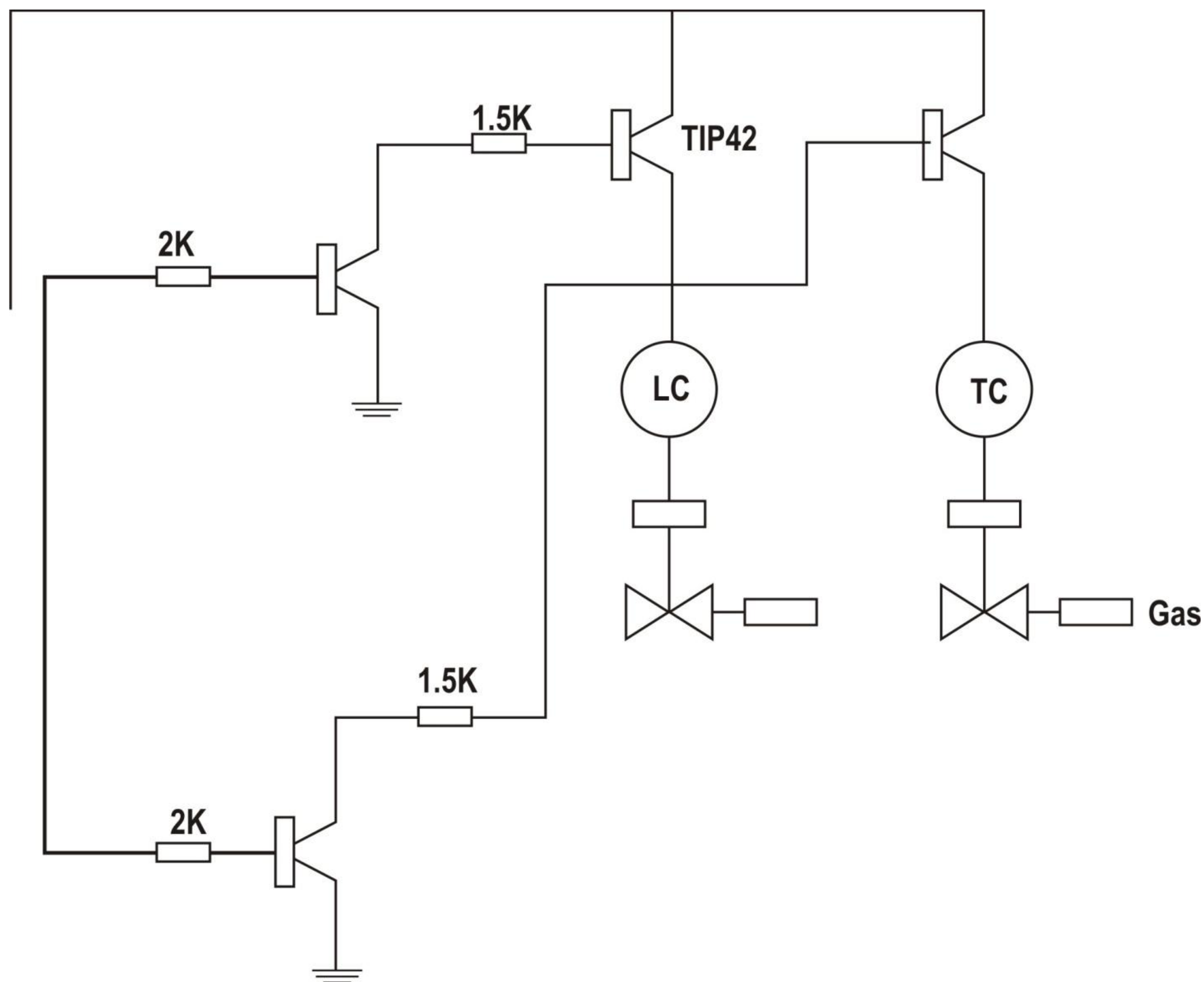



**Fig 4.26a: Output Interface.**

### 4.26 The LCD Interface to the Microcontroller

The display is a standard LM020a which displays 4 lines of 20 characters (20×4). Each character is of 5×10 pixels. The display receives ASCII codes for each character at the data inputs (D0–D7). The data is presented to the display inputs by the MCU, and latched in by the pulsation of the E (Enable) input. The RW (Read/Write) line can be tied low (write mode), as the LCD is receiving data only. The RS (Register Select) input allows commands to be sent to the display. RS = 0 selects command mode, RS = 1 data mode. The display itself contains a microcontroller; the standard chip in this type of display is the Hitachi HD44780. It must be initialized according to the data and display options required. Figure 4.26b shows the Display Unit. The LCD which is an output device to the microcontroller displays the various states of the process being monitored and other necessary information related to the plant and its environment.

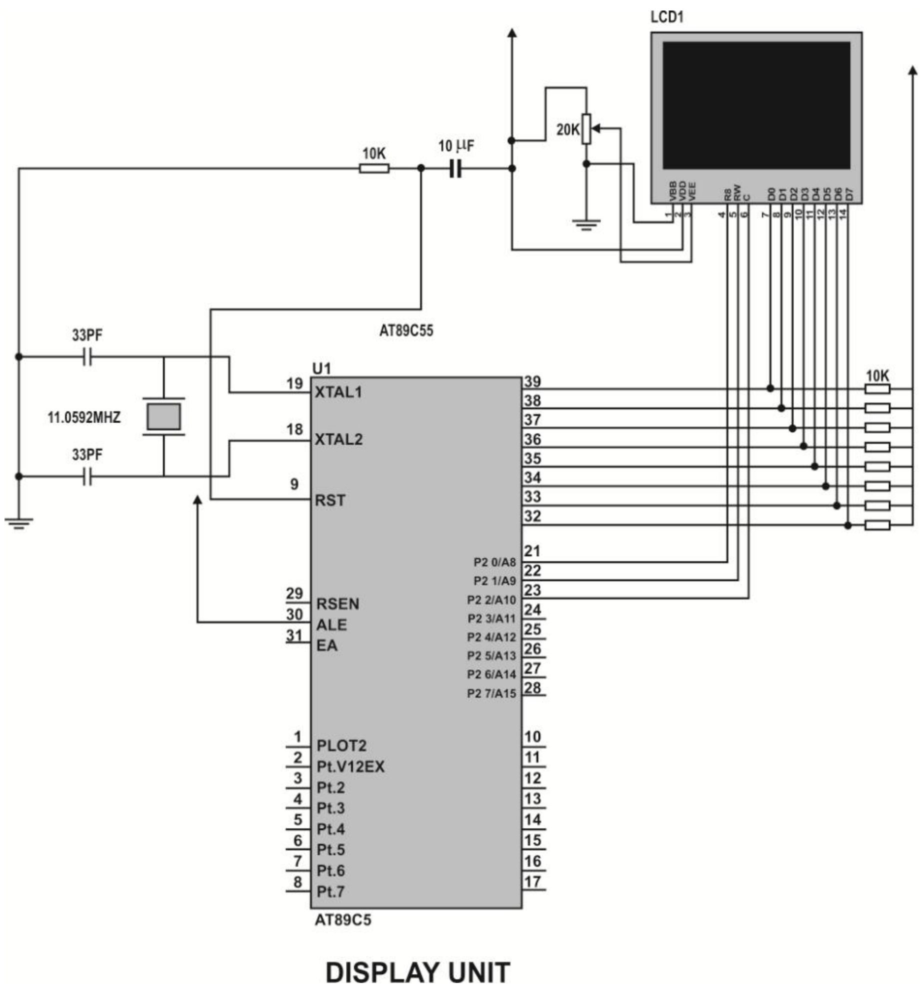



**Figure 4.26b: Display Unit**

### 4.27 The PC Interface

The PC serves as SMS gateway and also hosts a database to store various diagnostic information and archival data for decision making.  The computer method was adopted in this work. This method involves the use of level converter to interface between the microcontroller and the computer. The computer has the application program that gets data from the controller and in turn send appropriate SMS message using Attention Command (AT) via the phone. Figure 4.27 is the Computer Method.

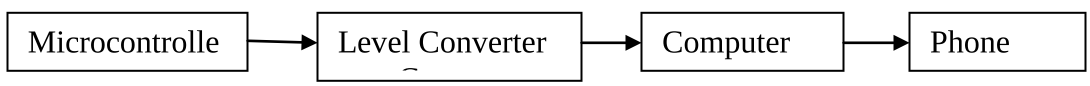



**Figure 4.27: The Computer Method**

**LIMITATION OF COMPUTER METHOD**

1. Although this method enjoys the ease of high level of language programming, it is not cost effective.
2. Besides, it cannot be easily deployed to remote areas.

### 4.28 SMS Interface

The widely available networks are based on GSM. The network provides a wide area of coverage and can be utilized more cost-effectively for this system and communication protocols that are DTMF (Dual Tone Multi Frequencies) Sharma and Kumar (2006), In this work the MTN network was used because of it wide coverage..

The use of SMS in remote application is becoming common in our modern society. This is because SMS is not limited by distance as it makes use of global system for mobile (GSM) communication network. Sending SMS from a computer using application program via mobile phone might be easier compared to sending the same from a microcontroller, yet in most remote applications, it may not be feasible to mount a computer unit at every point data is required to be taken and sent to host computer. In sensor networks for example, one may want to monitor the temperature, level, pressure and other physical measurable quantities of a process located in a remote place and send the result wirelessly to a database in a host computer. In such cases, one will definitely not use a computer at the remote location for obvious reasons. Wireless transmitter like radio frequency transmitters can be used but their coverage is limited by distance. One way to send such data is by interfacing the sensors to microcontrollers (which consume little) and than sending the data from the controller to the database of the host computer via a mobile phone. A major advantage of this approach is that even our old abandoned phones can do the work perfectly. Inyiama et al (2011).

**Design of SMS Interface**

Figure 4.28 shows the GSM subsystem. The GSM subsystem is interfaced with the Microcontroller using a RS232 interface. Since the RS232 is not compatible with today's Microprocessors and Microcontrollers, we need a voltage converter to convert RS232"s signals to TTL voltage levels. MAX 232 was used as a voltage converter. The MAX232 converter converts from RS232 voltage levels to TTL voltage levels and vice versa. One advantage of the MAX232 chip is that it uses a +5v power source, which is the same as the source voltage for the microcontroller. In other words, with a single +5v power supply we can power both the microcontroller and MAX232, with no need for the dual power supplies that are common in many older systems. The MAX 232 has two sets of line drivers for transferring and receiving data.

The subunit acts like an interface between the controller and GSM network. The GSM has a SIM (Subscriber Identity Module) card to make the network identify the user. A Nokia N90 series GSM modem was also interfaced to the PC to facilitate interaction with system/maintenance engineers anywhere they maybe, anytime it becomes necessary. The microcontroller communicates with the GSM module using the AT commands and other high level language,. These AT commands are used to send and receive SMS. The programming code for the microcontroller is written in some high level language. The GSM module has an RS232 interface for serial communication with an external peripheral. In this case, the transmit pin (Tx) of the computer's Serial port is connected with the receive pin (Rx) of the GSM module's RS-232 interface. The transmit pin (Tx) of the RS-232 of GSM module is connected to receive pin (Rx) of microcontroller's serial transmission pin. And the serial transmit pin of the microcontroller is connected to the receive pin of the computer's Serial port. Therefore the commands and their results are transmitted and received in a triangular. Appendix D is a Program to Interface GSM Module with 8051 microcontroller.

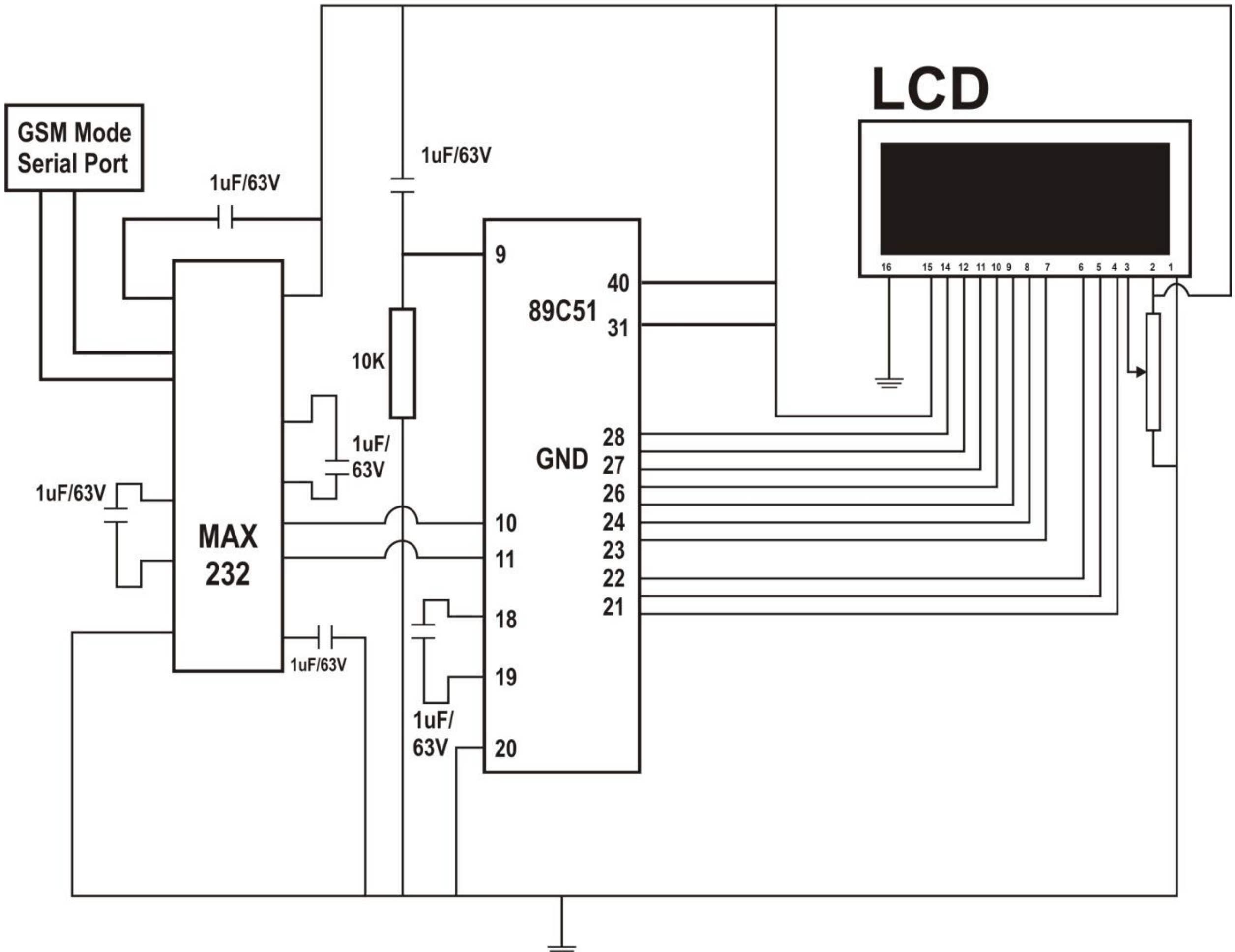



**Figure 4.28: GSM Interface to each Microcontroller**

### 4.29. System Architecture for SBMC System

In using the GSM for remote process control system using SMS, the Mobile Phone is integrated with the microcontroller through a Personal Computer, The remote personnel receives the status report of the process through SMS message from the user Mobile Phone, The user sends a command to controller to control the process variable by sending a set command. When action is taken, a feedback report is sent to the user. In a situation where the user cannot effect control, the attention of the control room can be drawn through the same means. Figure 4.29 is a flow chart showing the steps involved in controlling the process by the user.

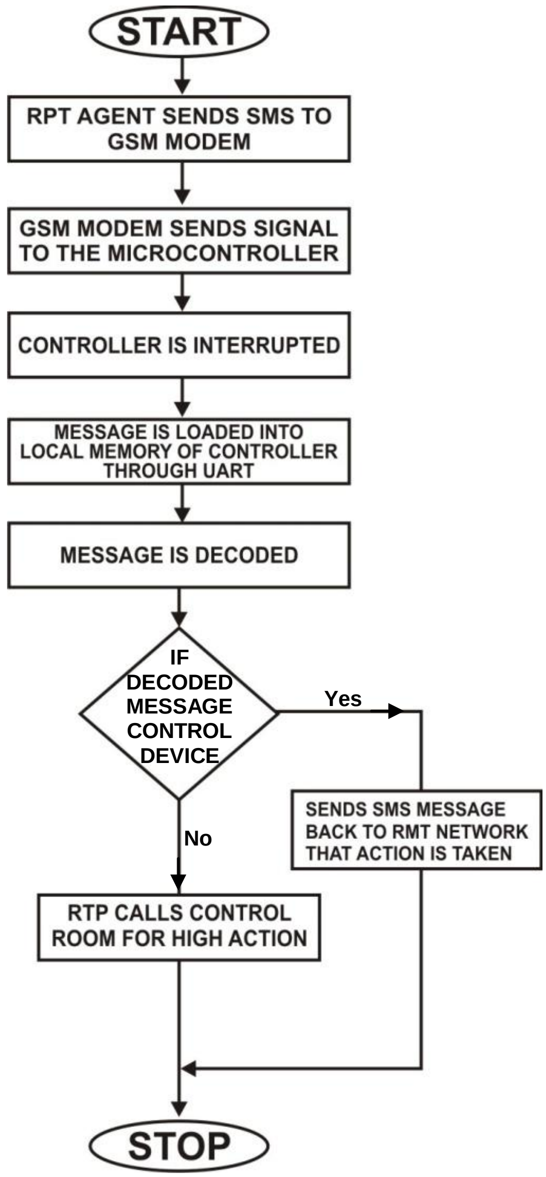



**Figure 4.29: Development Process of GSM Sub-unit.**

The structure of the system is working with the following steps:

Remote user receiver receives messages from user cell phone or mobile phone of process statue.

i) The remote user sends text messages (SMS) commands to the MODEM

ii) The MODEM sends message to the controller.

iii) GSM receiver decodes the sent message and sends the commands to the microcontroller.

iv) Microcontroller issues commands to the devices to control the process.

v) The Microcontroller checks for completion status and apply operation on field Devices.

vi) The remote user is informed of the outcome of their request by sending a completion status message back to remote user in the form of another SMS message.

The user can send a message to the control room

**4.30 Message Sending & Receiving**

The AT (Attention) commands and Visual Basic are the basic commands that communicate with the GSM mobile phone. Table 4.8 indicates some common AT commands necessary for SMS transmission and reception. Detailed format of AT commands are available in mobile phone, T68i developers guidelines

**Table 4.8: Some Common AT Commands for SMS Transmission and Reception**

| | |
|---|---|
| AT(Attention Command) | Checks the communication between the phone and any accessory. |
| AT+CPMS (Preferred Message Storage) | Selects memory storage spaces to be used for reading, writing, etc. |
| AT+CNMI (New Message Indication to TE) | Selects the procedure how the reception of new messages from the network is indicated to the TE when TE is active. |
| AT+CMGR (Read Message) | Returns messages with location value <index> from preferred message storage <mem1> to the TE. If the status of the message is „received unread", the status in the storage changes to „received read". |
| AT+CMGS (Send Message) | Sends message from a TE to the network. |
| AT+CMSS (Send From Storage) | Sends message with location value <index> from message storage <mem2> to the network |
| AT+CMGW (Write Message To Memory) | Stores a message to message storage <mem2>. The memory location <index> of the stored message is returned. |
| AT+CMGD (Delete Message) | Deletes message from preferred message <mem1> storage location |

### 4.30.1 Design of the Feedback (Device STATUS)

The system is programmed to give feedback message when the control has been effected .

**Algorithm**

**Step 1**: Start

**Step 2**: Phone initialization

**Step 3**: Get Hardware Software

**Step 4**: Poll SMS from mobile phone

**Step 5:** Receive SMS

**Step 7**: Check measured value of variable

**Step 8**: Control the device based on status by sending SET (Value)

**Step 9**: Notify end user

**Step 10**: Go to step1

### 4.30.2 Advantages of the System

The system offers several attractive features like**:**

**1. Convenience** – SMS technology is easy to use and learn and can be accessed easily when needed.

**2. Accessibility** – instructions can be sent to the microcontroller to be controlled and monitored from any location provided there is the existence of an active GSM network or control from anywhere in world if cellular coverage is available.

**3. Portability** – a microcontroller can be controlled and monitored from any GSM phone that supports SMS. Considering the fact that most GSM phones support SMS, the system is therefore highly portable.

**4. Saves Time** – an SMS based remote monitoring and computer control system saves time as the user is not required to gain access to an internet connection or make a dedicated connection to the computer to be controlled as opposed to a Bluetooth-based system or an Internet based system.

**5. Cheaper** – SMS services are generally cheap and are sometimes provided for free (at least for certain periods) by service providers. Furthermore, most service providers do not charge users for receiving SMS.

**6. Mobility** – User and/or system administrators are more likely to have their phones with them at all times than they are likely to physically be in front of their computers. An SMS based system therefore enables them have ubiquitous access to the computer to be controlled and monitored.

**7.** Acknowledgement about execution of command from system to user.

**8.** To alerts user on occurrence of any abnormal conditions like power failure, parameters.

**9.** The ease of implementation and cost effective approach.

### 4.31 Design of Fault Diagnosis Sub-System

The objective of this sub-system is to monitor the plant during normal operation and detect the occurrence of faults, time of occurrence and the remedy. Most of the faults in a chemical process plant is associated with relieve valves and sensor failures. A database of all the likely faults and their remedies were created. Whenever any fault is detected and remedied, the database is usually updated. Figure 4.2 is the flow diagram of the self diagnostic features. The data acquisition system acquires data from the plant using the sensors.. The acquired data is initially processed at the data classification module, which uses neural networks and statistics to perform data pre-processing, data compression and data format transfer. It has condition monitoring module, coupled with forward chaining inference, which alerts the user to abnormal process conditions. Then, the incident report module uses the case-based reasoning technique to determine whether the current conditions had previously occurred. If they had, the fault diagnosis module will generate a solution. Otherwise backward chaining will be invoked to
detect faults.

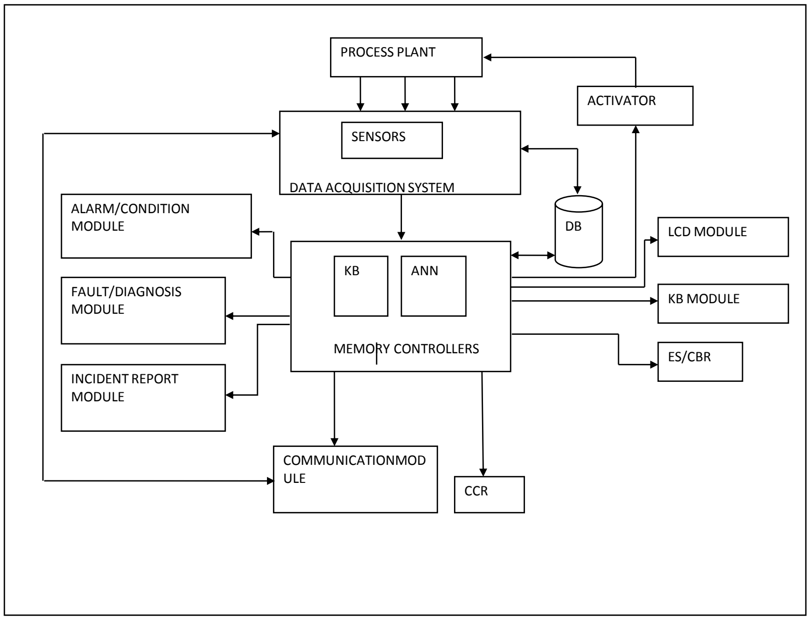



**Figure 4.30: Complete Block Diagram of the System**

# CHAPTER FIVE
# SYSTEM SIMULATION AND TESTING

## 5.1 System Simulation

System Simulation is the mimicking of the operation of a real system, such as the day-to-day operation of a bank, or the value of a stock portfolio over a time period, or the running of an assembly line in a factory, or the staff assignment of a hospital or a security company, in a computer (Hossein1995)

Peter (2003) stated that instead of building extensive mathematical models by experts, the readily available simulation software has made it possible to model and analyze the operation of a real system by non-experts, who are managers but not programmers. Or put in other words system simulation is a technique of solving problems by observation of performance over time of a dynamic model of the system

Peter went on to describe simulation as the execution of a model, represented by a computer program that gives information about the system being investigated. The simulation approach of analyzing a model is opposed to the analytical approach, where the method of analyzing the system is purely theoretical. As this approach is more reliable, the simulation approach gives more flexibility and convenience. The activities of the model consist of events, which are activated at certain points in time and in this way affect the overall state of the system. The points in time that an event is activated are randomized, so no input from outside the system is required. Events exist autonomously and they are discrete so between the executions of two events nothing happens. The SIMSCRIPT provides a process-based approach of writing a simulation program. With this approach, the components of the program consist of entities, which combine several related events into one process.

## 5.2 Simulation Process of the System

The model of the remote industrial process monitoring and control system using artificial neural network was developed using the proteus simulation software. The various training data was fed into the system. The schematic capturing was used for the design entry. Figure 5.1 is the diagram of the functional simulation of the entire units of the process plant for the processing of oil and gas in shell. It is made up of the eight sub-plants representing each sub unit of the plant. The eight different plants have 27 input variables. Sub-units such as well stream heater, inlet separator and Liquid cooler have three variables each. The rest of the subunits have two or four variables each. When you simulate the plant by using the play button, the fluid will start flowing and the different variables namely temperature, pressure, level and flow rate are monitored and regulated. Experimental data from SHELL was used to test the plant. The simulation was done using proteus ISIS schematic capturing tool. The source code was developed using a variant of assembly language called C-Sharp. The source code is attached as appendix A. Figures 5.2 to 5.4 are the stream well, Inlet- separator and dehydration sub- units of the process plant.

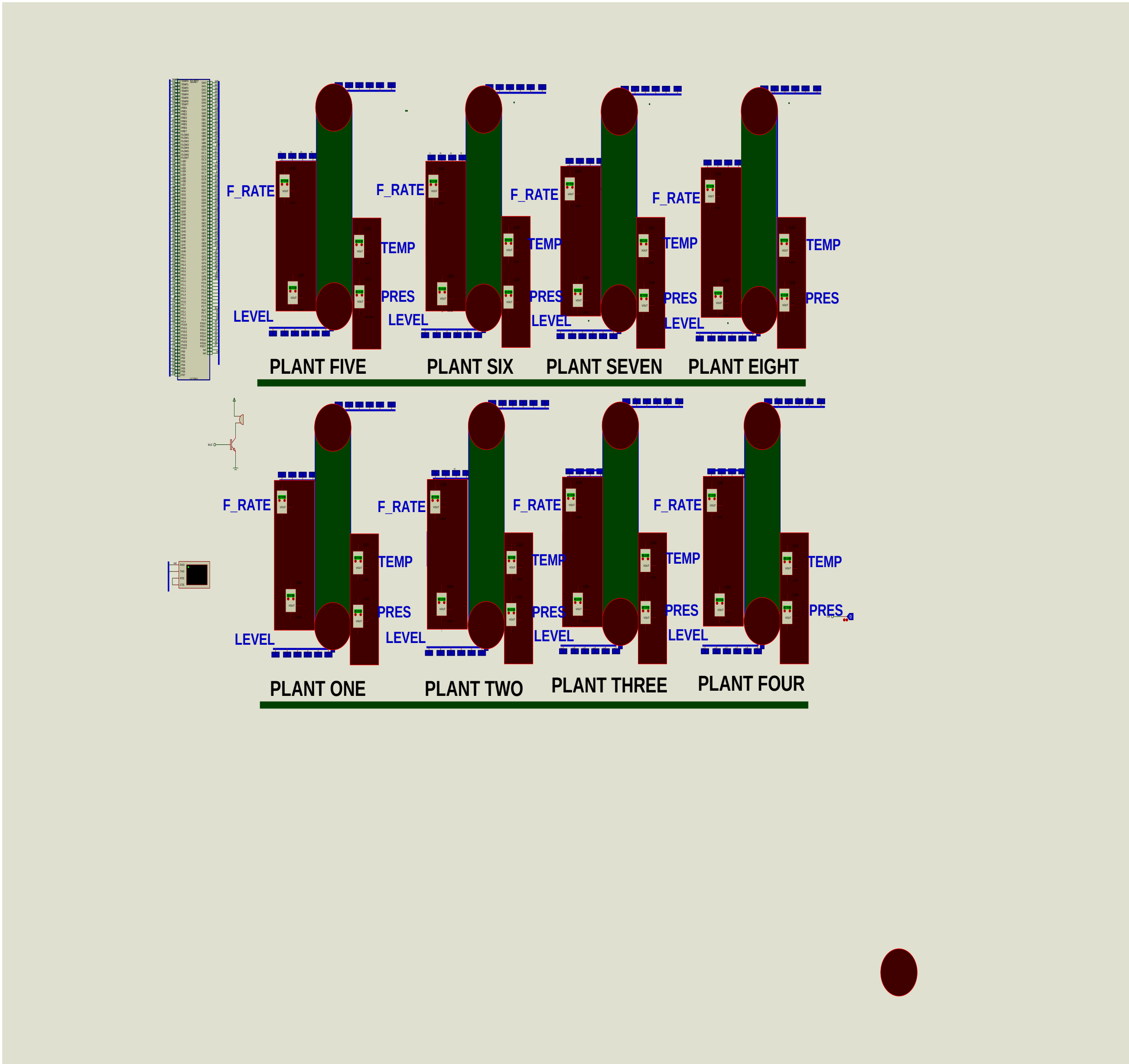



**Figure 5.1: Simulation of Process Control System using Proteus Capturing Tool**

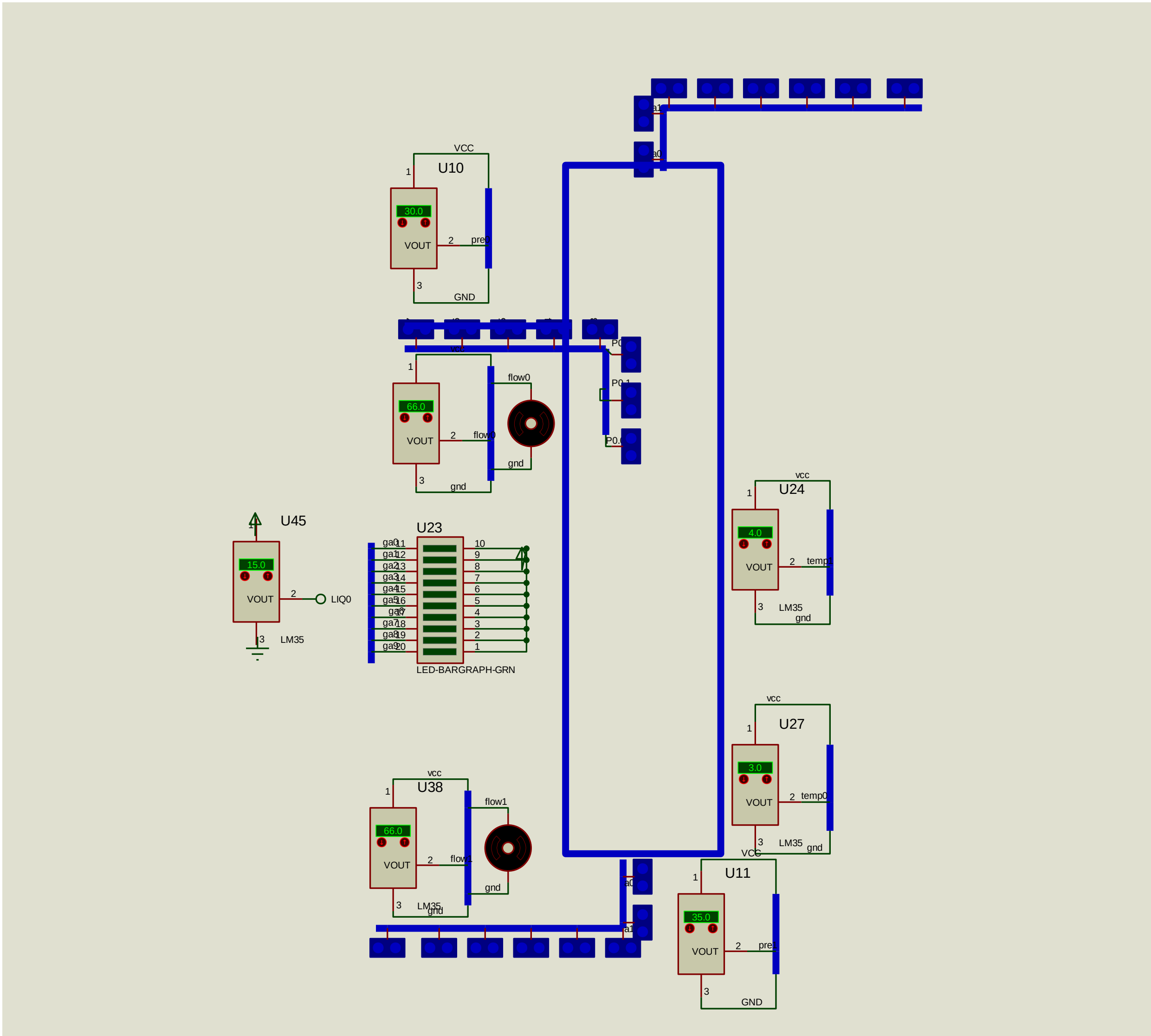


**Figure 5.2: A Sub- unit of the Process Plant (Stream well).**

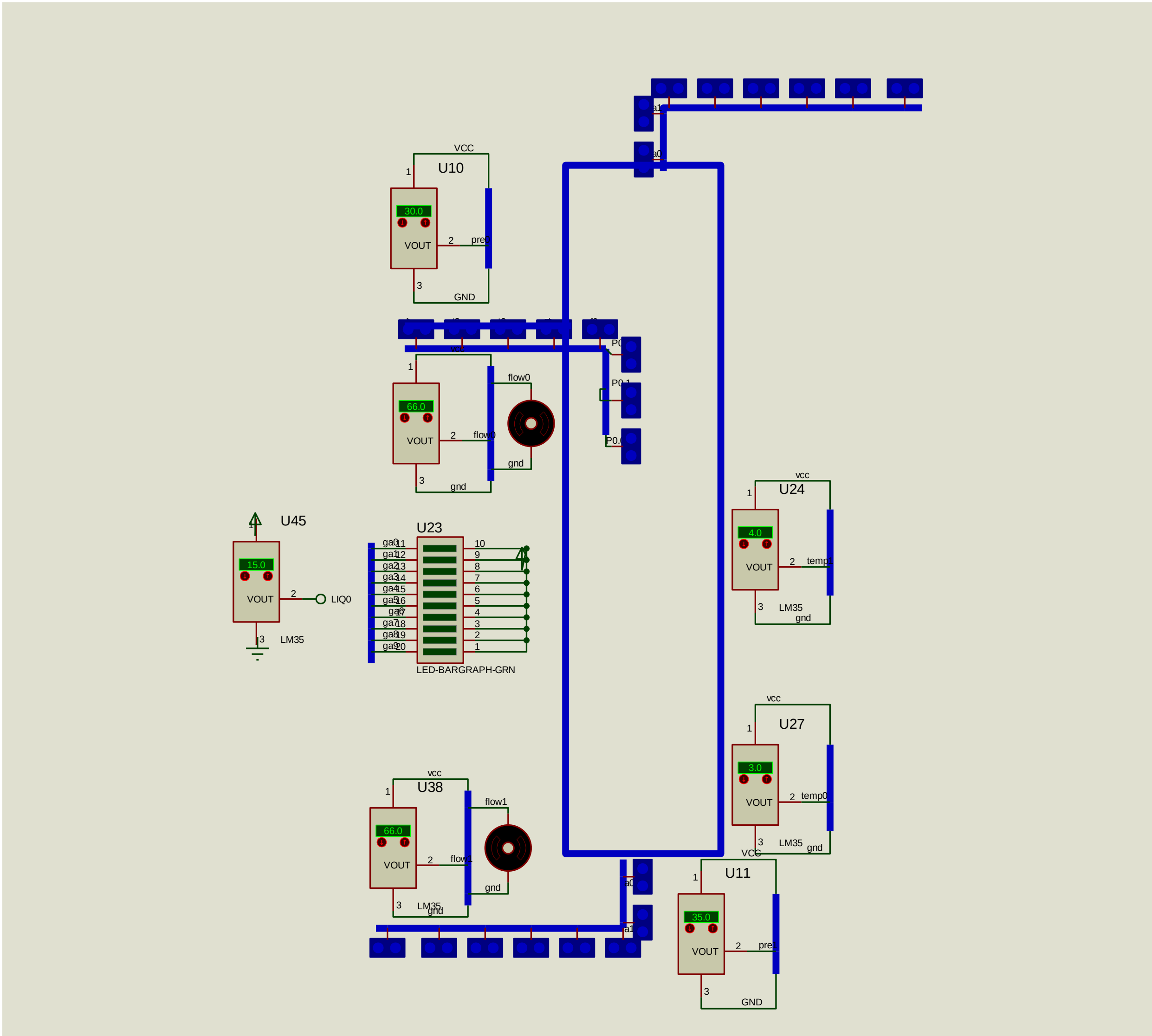



**Figure 5.3: A sub- unit of the Process Plant (Inlet Separator).**

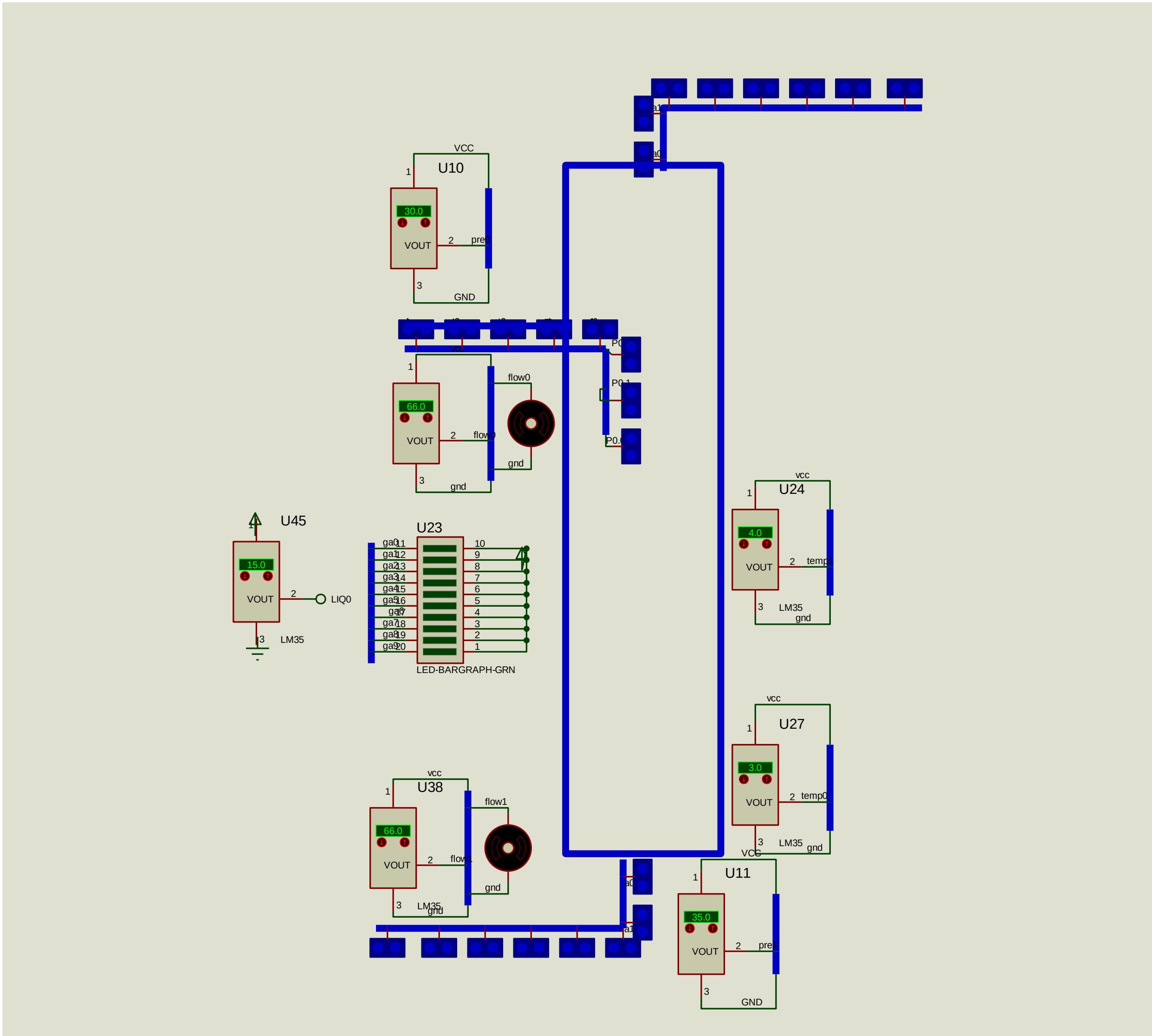



**Figure 5.4: A sub- unit of the Process Plant.(Dehydration unit)**

## 5.3 Schematic Capturing and Simulation

The aim of prototyping is to develop a prototype that will show the practicability of this work. In developing a prototype one took into cognizance that in oil/Gas process industries there are four principal variables to control namely pressure, Temperature, Level and flow-rate. None of the nine units involved in processing oil and gas has all the four parameters. From data collected from SHELL, the different sub-units consist of mostly three and two of the process variables. Table 5.1 is the values of the variables in the different units.

**Table 5.1: values of the Variables in the different units**

Well steam Heater (U1)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <115 | 122 | 150 | 158 | >166 |
| Pressure | <85 | 85 | 90 | 95 | >95 |
| Level | <807 | 807 | 850 | 893 | 805 |

Dual Inlet Monitored (U2)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <42.5 | 42.5 | 45 | 48 | >48 |
| Pressure | >97 | 97 | 102.8 | 108 | >108 |

Inlet Separator (U3)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <42.5 | 42.5 | 45 | 48 | >48 |
| Pressure | <97 | 97 | 102 | 108 | >108 |
| Level | <247 | 247 | 260 | 273 | >273 |

Gas Cooler (U4) Inlet Variable

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <23 | 23 | 25 | 27 | >27 |
| Pressure | <47 | 47 | 50 | 53 | >53 |

Output Variable (U4)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <32 | 32 | 35 | 37 | >37 |
| Pressure | <89 | 89 | 93 | 98 | >98 |

Gas Compression (U5)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <33 | 33 | 35 | 37 | >37 |
| Pressure | <93 | 93 | 98 | 103 | >103 |
| Flow rate | <59 | 59 | 65.9 | 70 | >70 |

Condensate processing (U6)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <25 | 21 | 23 | 25 | >25 |
| Pressure | <60 | 60 | 65 | 70 | >70 |

Pump (U7)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Tempe ration | <40 | 40 | 42 | 44 | >44 |
| Pressure | <46 | 46 | 52 | 58 | >58 |

Liquid Cooler(U8).

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <15 | 15 | 20 | 25 | >25 |
| Pressure | <45 | 45 | 50 | 55 | >55 |
| Level | <60 | 60 | 65 | 70 | >70 |

Dehydration Tag (U9)

| | LL | L | N | H | HH |
|---|---|---|---|---|---|
| Temperature | <40 | 40 | 45 | 50 | >50 |
| Pressure | <90 | 90 | 102 | 110 | >110 |

NOTE

LL = Low Low

L = Low

N= Normal

H= High

HH= High High

The test carried out on the selected units (two of the units with three variables and two of the units with two variables) and the prototypes designed and simulated show that the control of the final elements is stable and recorded negligible offset. The data recorded above is from the prototype sets.

The variable used to measure the performance of the control system is the error e (t), which is the differences between the constant set point or reference value r, and the controlled variable,

e(t) = r – c(t)……………………………………………………………………………….(35)

In principle the objective of a control system is to make the error e (t) exactly Zero

**5.4 Integration and Simulation Testing**

In line with the objective of this work, models of the control units, the input and output units and GSM sub unit were designed, tested and integrated. The test result shows that the various units are performing its specific role in the entire system properly. The integration result shows that the various unit of the system performing a specific function or role in the whole system.

After the integration the result is tested to ensure that it produces the right output. If this stage fails the code is debugged and the faulty unit is isolated, and undergoes unit testing, after which it is re-integrated and the final test performed again until success is achieved.

The variable used to measure the performance of the control system is the error e (t), which is the difference between the constant set point or reference value r, and the controlled variable c(t):

e(t)= r – c(t) as shown in equation (35)

In principle the objective of a control system is to make the error e(t) exactly zero

From the temperature, pressure and Level reading, the system is providing accurate control actions. Figures 5.4a, 5.4b, 5.4c, 5.4d, 5.4e and 5.4f illustrate the temperature, pressure and level control of the stream well of the system from the data in table 5.1 using MATLAB 7.1.

**Temperature Control**

```
temp =[100 102 104 106 110 112 116 118 120 122 124 126 135 140 145 148 150 151 152 154
148 149 150 152 154 150 155 160 158 156];
>> time = [0:2:58];
```

```
plot(time,temp)
xlabel('Time(sec)')
ylabel('Temp(oC)')
grid on
```

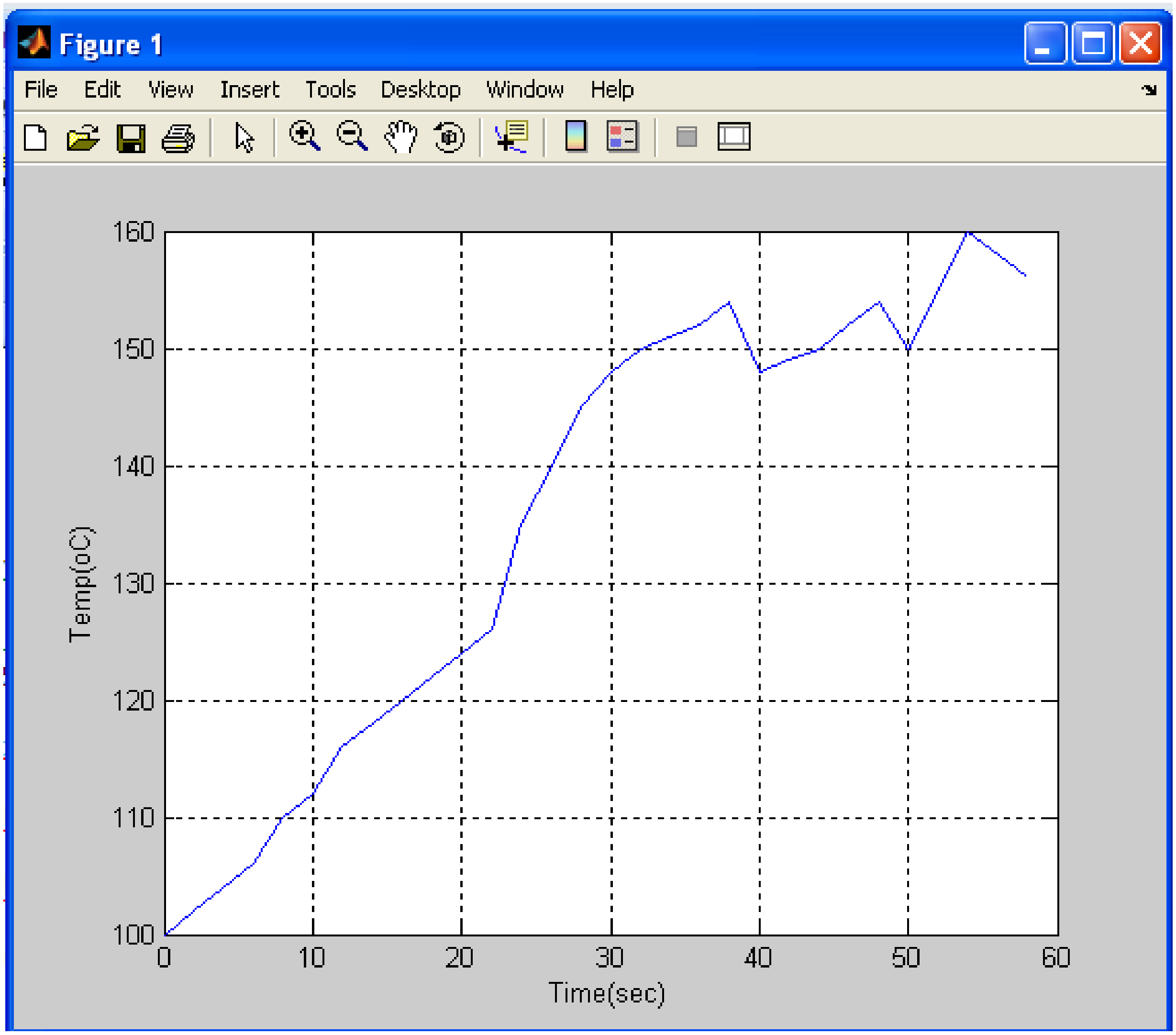



**Figure 5.4a: Temperature Control curve of set point of 150**

The curve in fig 5.4a shows the temperature of a process in gas tank. The temperature was taken after five seconds interval .The set point is 150 $^{o}$C with 4% either way. The curve shows a negligible offshoot. Figure 5.4b is a comparison of neural controller and on/off controller. The green curve is for the neural controller (NC) while the blue is for the On/Off controller. .Neural controller achieved a better control and stability at the set point of 150 $^{o}$C.

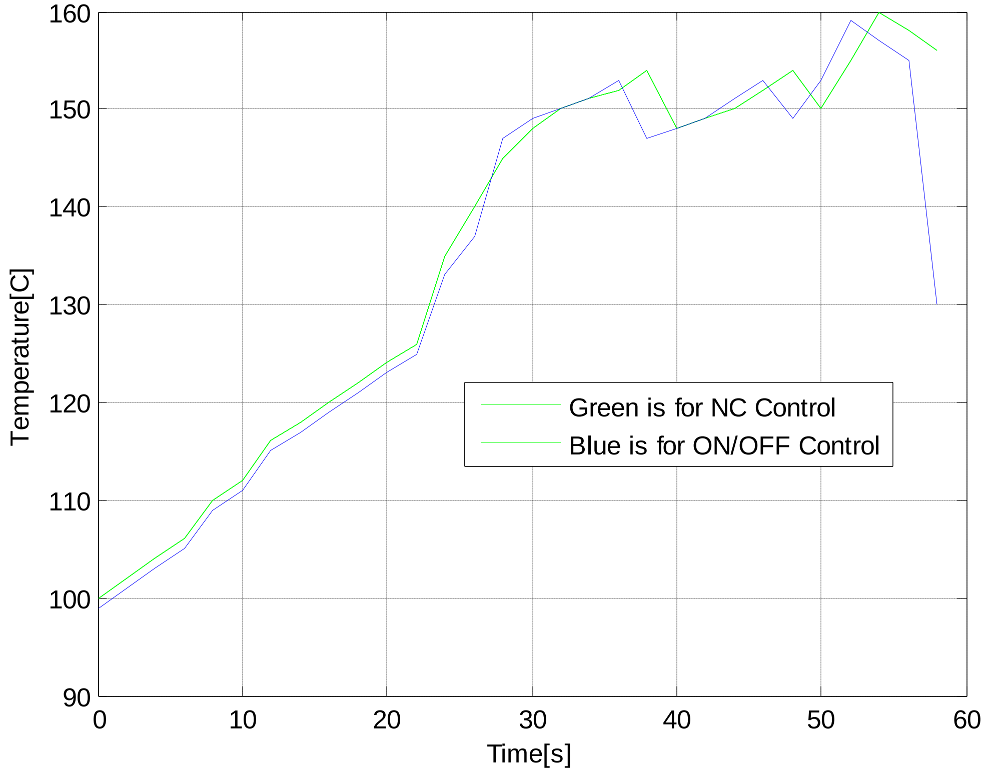



**Figure 5.4b: Temperature Control curve of set point of 150 for NC & ON/OF Controller**

**Level Control**

level = [770 780 800 810 850 851 852 855 848 848 850 850 855 860 900];

time = [0:14];

plot (time, level)

xlabel ('Time(sec)')

ylabel (‘Level (m)')

grid on

The Figure 5.4c shows a Level Control curve of set point of 85**0** while **f**igure 5.4d is a comparison of the curve generated by NNC and ON/OFF controller. The curve generated by NNC is on blue colour while that of ON/OFF Controller is on red. The curve generated by the NNC demonstrated better result in terms of stability.

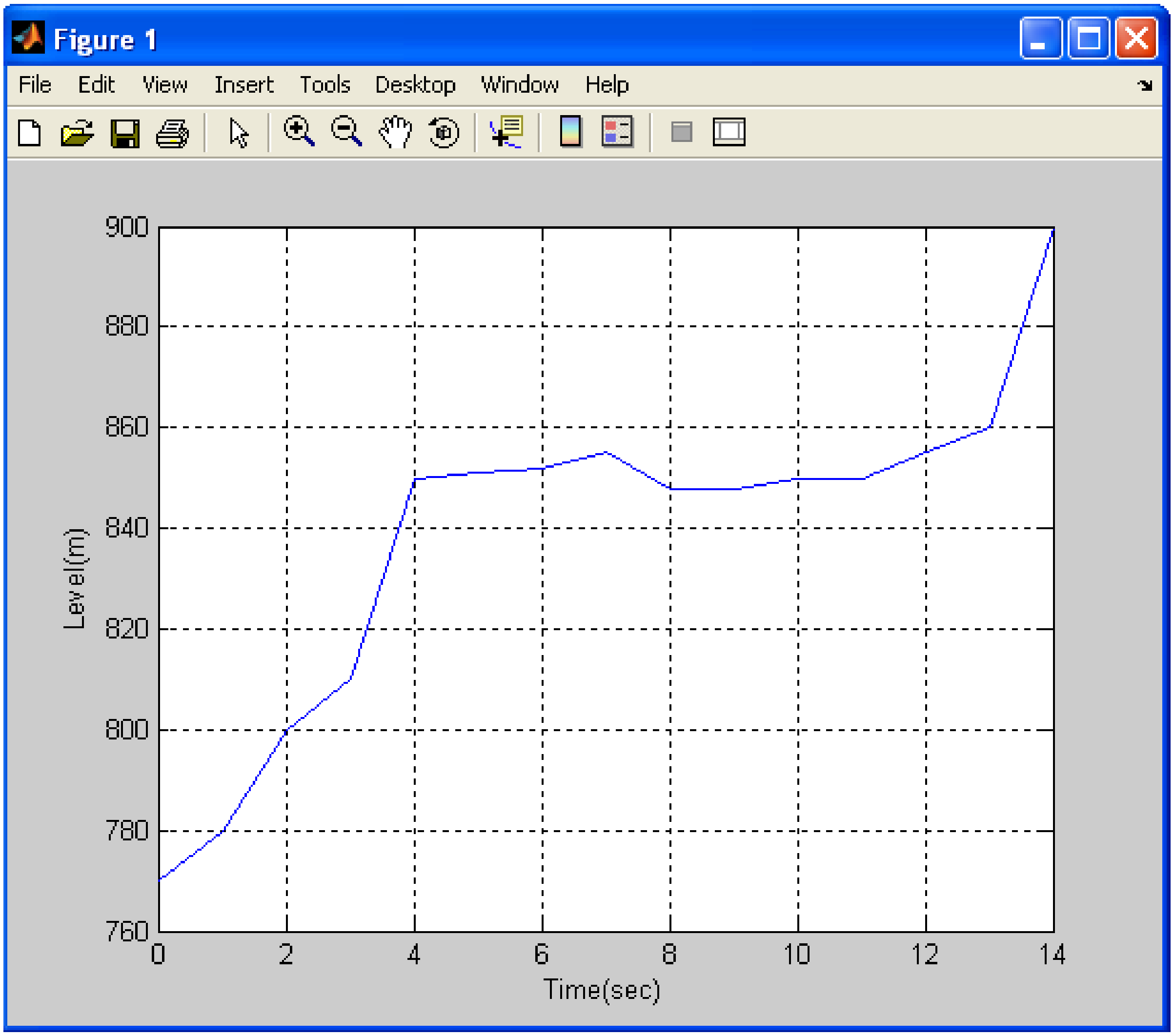



**Figure 5.4c: Level Control Curve with set point of 850**

```
LEVEL CONTROL CODE FOR DOUBLE GRAPH
Level=[770 780 800 810 850 851 852 855 848 848 850 850 855 860 900];
Lev=[760 770 790 800 840 850 850 830 840 840 830 840 840 850 890];
time=[0:14];
plot(time,Level, 'g')%first function
hold on
grid on
xlabel('time(sec)')
ylabel('Level(m)')
plot(time,Lev, 'r')%second function
```

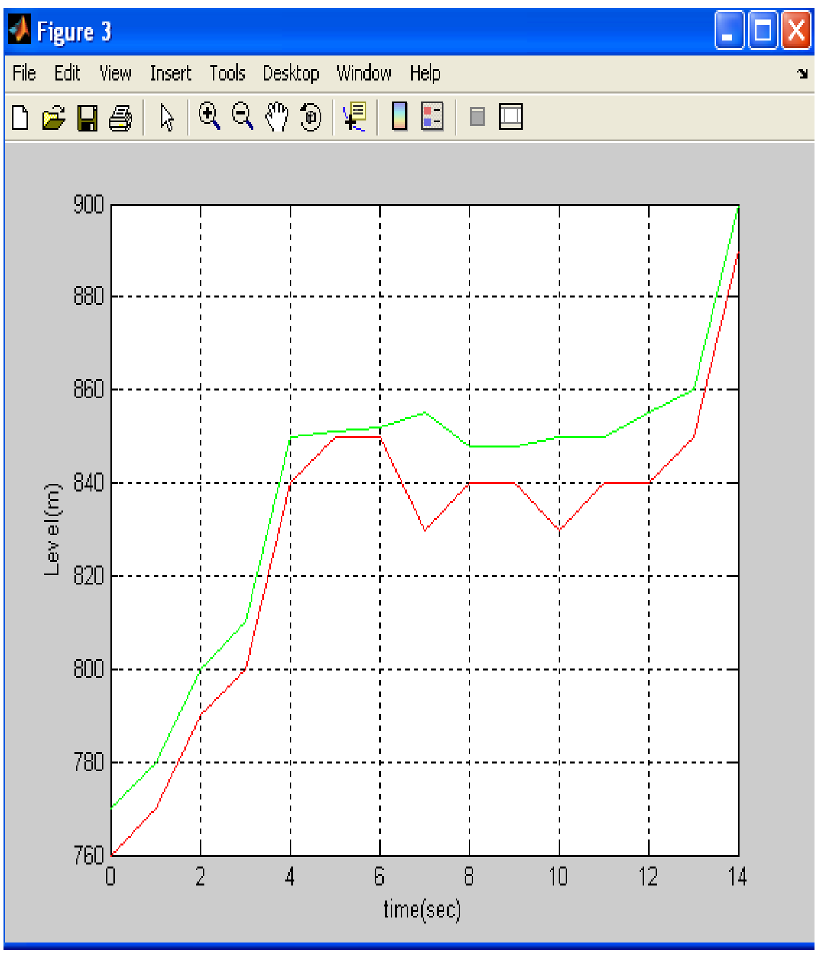



**Figure 5.4d: Level Control Curve with set point of 850 for NC & ON/OFF**

**Pressure Control**

pressure =[60 65 70 75 80 85 90 95 90 85 87 89 90 90 90 90 90];

plot(time, pressure)

> time = [5:5:85];

>> xlabel ('Time(sec)')

>> ylabel('Psi')

>> ylabel('Pressure(psi)')

>> grid on

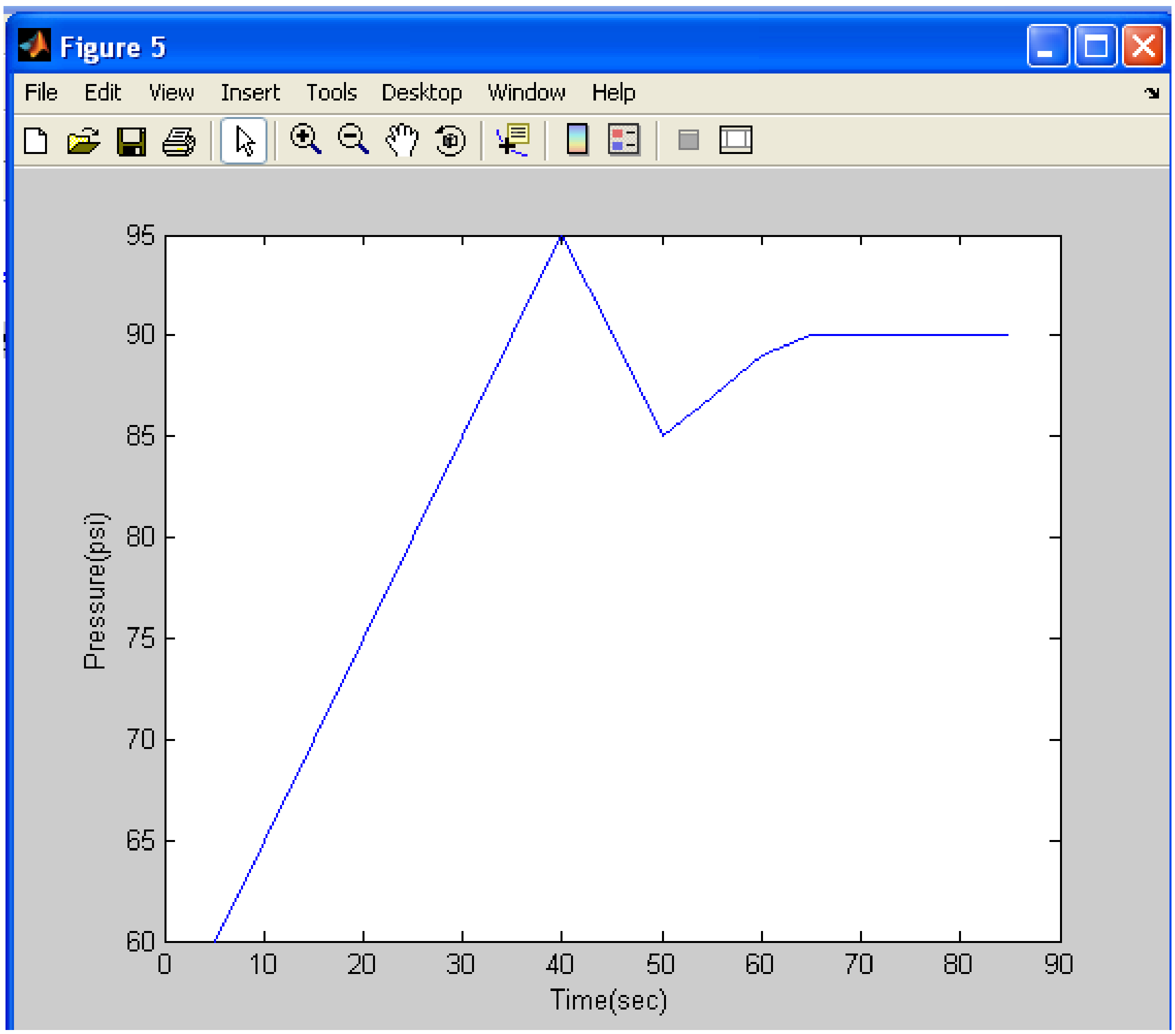



**Figure 5.4e: Graph of pressure Control curve with a set point of 90** KN/$m^2$

The curve in 5.4e shows the control operation of the pressure variable in the stream well with a set point of 90 KN/m$^2$. The curve demonstrated that the overshoot is within limits.

The curves in 5.4f show the comparison between neural controller and on/off controller. The blue curve is generated using neural controller while the red curve is generated using on/off controller. The neural controller demonstrated better control and stability.

```
PRESSURE CONTROL FOR DOUBLE GRAPH
Level=[770 780 800 810 850 851 852 855 848 848 850 850 855 860 900];
Lev=[760 770 790 800 840 850 850 830 840 840 830 840 840 850 890];
time=[0:14];
plot(time,Level, 'g')%first function
hold on
grid on
xlabel('time(sec)')
ylabel('Level(m)')
plot(time,Lev, 'r')%second function
```

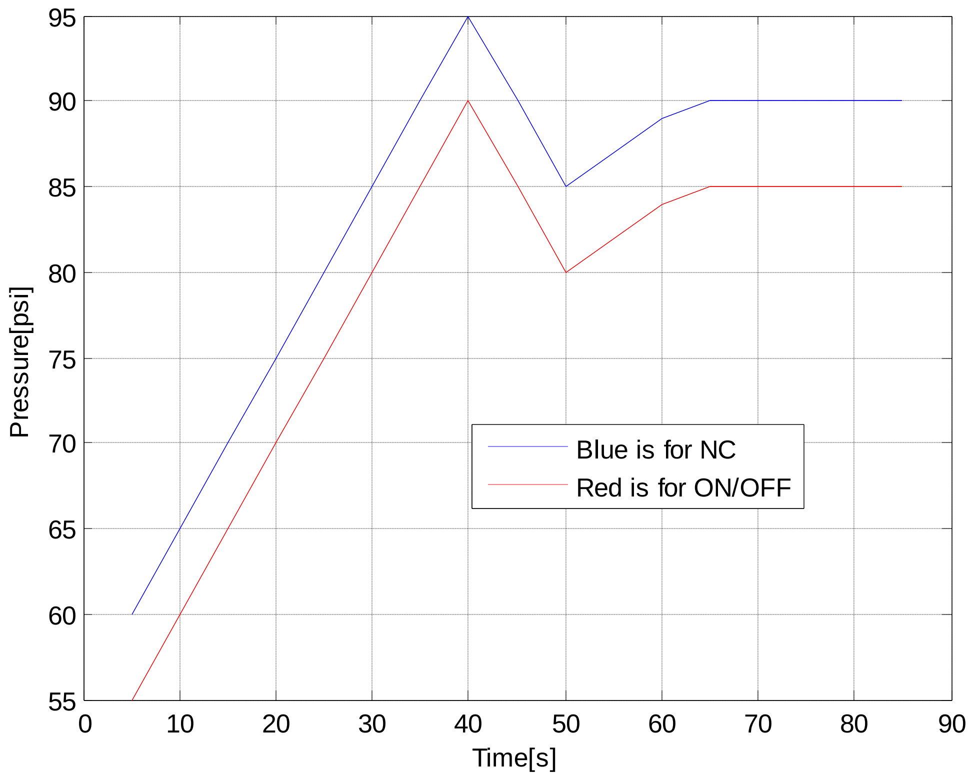



**Figure 5.4f Graph of Pressure Control curves with a set point of 90 for NC & ON/OFF Controller**

### 5.6 Performance Evaluation

Performance evaluation of a work of this type is based on the objectives. Performance evaluation of this system should be based on the functionality of the system in line with the required specification

The design objective is the ability to monitor and control the variables associated with industrial process such as temperature, pressure, level and flow rate with acceptable limit or set point in order to ensure quality and proper running of the devices associated with this variables and to communicate engineering and management information through GSM to remote users. Functional simulation carried out on this system show that effective monitoring, controlling and setting of control parameters as well as fault diagnoses can be achieved. The experimental result obtained from the test carried out on the prototype and the resultant graph confirms that the system can be used to effectively monitor and control industrial system. The new system is compared with

ON/OFF controller and it is shown that ON/OFF Controller based system fails miserably because of its limitations. Such limitations include inability to handle nonlinear and Complex systems as well as multivariable systems.

On the other hand ANN based approach has resulted in possible implementation of better and more efficient control. Artificial neural networks do not require a prior knowledge of system and have inherent ability to adapt to the changing conditions unlike conventional methods. The results also show that remote user is able to monitor and control process variables.

### 5.6.1Performance Evaluation of System Availability

The system availability was improved from 70% in the old system to 90% in the new. The current system has a system availability of seventy percent (70%) but new system has a system availability of ninety percent (90%) calculated as follows

Total available time ($T_A$) per month = 30 days *24 hrs given 720 hrs

Available Time ($A_V$) = 70% of Total available time $T_A$ = 504 hrs

Maintenance time ($M_T$) = $T_A$ -$A_V$ (720 – 504) = 216

To calculate the difference in terms of hrs = twenty percent of total hrs

20 percent of the total time= twenty 20 of 720 = 144

Add $A_V$ to 144 (504 to 144) = 648

Therefore the plant availability 648/720 * 100/1 = 90%

From the calculation above the system availability is now to 90%

# CHAPTER SIX
# SUMMARY AND CONCLUSION

## 6.1 Summary of Achievements

This work has been able to achieve the following:

- The study of the design methods and architectures developed so far in SHELL for industrial process monitoring and control.
- The designing and development of an SMS-based industrial process monitoring and control systems using artificial neural network. This system monitors and controls industrial variables namely pressure, temperature, level and flow rate in a SHELL plant and alert remote personnel in case of departure from set point who can remotely effect a change.
- The work has also demonstrated the use of artificial neural network and rule based expert system to closely monitor and control industrial process for better plant operation and maximize plant availability.
- The use of nine dedicated microprocessors one per plant unit has provided more computing and programming power to handle real time operations and complex situations
- The introduction of rule based expert system and artificial neural networks have helped in designing a system that is self monitoring and self diagnosing.
- Design and development of a system that is self learning and self adaptive using.
- The introduction of SMS-based interface which is the back bone of the remote control has been achieved thereby making it possible for personnel/operators to know what is happening in the plant anytime anywhere.
- The short message service based interface alerts industrial personnel when and if any of these variables goes out of bounds in spite of the control systems effort.
- Functional simulation of the control system was done using proteus ISIS
- Virtual demonstration of the control process using laptop, MTN modem,, N90 Nokia series and Remote manager was carried out

## 6.2 Problems Encountered and Solutions

- Books and reference materials on the subject matter are very scarce.
- Information gathering and data collection was not easy.

- Sourcing data from shell was very difficult as most information were classified.
- There was not enough information on this area on the web.
- Refusal by Shell to release information didn't help matters as most Information on their operations was classified.
- Frequent power failure was the major constraint at the time of programming.
- A lot of money was spent on transportation

## 6.3 Suggestions for Further Improvement/study

It is suggest that further work should be done on comprehensive web-based industrial process monitoring and control system which is fully interactive.

With advancement in GSM technology it is suggested that work should be done on the development of audio or voice based remote monitoring and control system. With the help of the audio or voice system, monitoring and control can be effected especially when physically handicapped persons are involved.

## 6.4 Contribution to Knowledge

The contribution of this dissertation to the body of knowledge includes among others:

**a.** The work provides an ANN-based automatic means of monitoring in real time the critical parameters in the oil and gas industries and takes preventive measures to forestall plant shut down when a critical parameter deviate from the norm.

**b.** The work provided an expert system based means of communicating with the plant personnel about any situation that requires human intervention

**c.** The dissertation also provided a means of effecting a control action by the personnel from anywhere to redress any adverse condition before it gets out of hand. The work also developed a knowledge Based system and software interface for the industrial remote control and monitoring.

**d.** The intensive and continuous monitoring that disrupt operation has bee taken care by this system thereby cutting down maintenance time.

**e.** The work as a result of the use of artificial neural network and rule based expert system has introduced a system that is self learning, self adaptive & self diagnosing and has the ability to document faults and their remedies for future repair.

### 6.5 Conclusion

This work has described an approach to remote industrial process monitoring and control using Artificial Neural Networks. The work has explored and demonstrated the utility of artificial neural networks technology for developing effective controller for dynamic industrial processes.

ANN models, approach have been found as effective ways to model complex process due to their non-linear character structures. This approach can enhance the regulator and advanced control capabilities of various industrial processes variables.

The work demonstrated that it is possible to design a system that will monitor industrial variables such as pressure, temperature, level and flow rate in an Oil and Gas plant and alert remote personnel (in case of departure from set points) who can possibly make a change. As result of the use of multi-processors it was possible to keep each of the monitored variables within set limits using Artificial Neural Network approach. The design of an SMS-Based software to alert industrial personnel when and if any of these variables goes out of bounds in spite of the control systems effort has been shown. The system is self learning; self adaptive and self diagnosing using Rule based expert system.

The new system is compared with ON/OFF controller and it is shown that ON/OFF

Controller based system fails miserably because of its limitations. On the other hand ANN based approach has resulted in possible implementation of better and more efficient control. Artificial neural networks do not require a prior knowledge of system and have inherent ability to adapt to the changing conditions unlike conventional methods. It is noteworthy that ANN based systems can save lot of resources (energy and power) and can provide optimized results to all type of engineering areas.

The final result shows that remote industrial process monitoring and control system using artificial neural network and rule based has a better performance than conventional approaches. It also increases the availability of management information through real-time data capture and is cost effective.

### 6.6 Recommendation

The use of ANN is recommended to meet the requirements with respect to environment, health and safety of plant personnel as well for minimal energy consumption.

The problem of software and hardware use in Nigeria is a heart rendering one. There should be full introduction and adoption of software and hardware automation system in Nigeria. Skill acquisition centers should be affiliated to our universities where the students can acquire practical and thorough skill in programming especially assembly language, microprocessors and object oriented programming

The universities should assist students by making some of the facilities required for this type of research projects readily available to them. Such facilities like computers, textbooks, software, hardware and others materials for research development should be adequately provided and possibly, Internet services should be provided at subsidized rates for the researchers and their lecturers.

## References

Adams D. R Wagner G. E(1986). “Computer Information Systems: An Introduction” South Western Publishing Company Cincinnati, Ohio. Pp 523-526.

Akpado Keneth (2010) ‘Re-Engineering Industrial Process Using Intelligent Agent”.(Ph.D Dissertation).Department of Electrical and Electronics, Nnamdi Azikiwe University Awka, pp64-67

Aimel “8-bit microcontroller with 20k byte flash datasheet” http://www.atmel.com

Alfred, J.M, Van Oorschot P.C. and Scott, (2001)”.Handbook of Applied Cryptography”. 5th Edn. CRC Press. Kohonen, T, Self-Organizing Maps. Springler-Verlag, Berlin.pp 1-6

Ahmad A, Mohd K and Abd H(2001). “Neural Networks for Process Monitoring, Control and Fault Detection: Application to Tennessee Eastman Plant “pp 2-4

Atmel corporation “Microcontroller MICS-S1 Family Of Products Literature ”
Author Publication, Singapore Pg. 562 – 565 ITU (1999)

Atmel Corporation’ Motorola mcs-51 family of product literature’ 2325 orchard parkway, San Jose CA 95131, USA

Atmel 8-bit Microcontroller with 20k byte flash datasheet’ http://www.atmel.com

Aykut. S, Demetgul, M. & Tansel, I. N. (2010). “Selection of Optimum Cutting Condition of Cobalt Based Alloy with GONN”. The International Journal of Advanced Manufacturing Technology, Vol. 46, No. 9-12, pp. 957-967.

Beale, M. H., Hagan, M. T. & Demuth, H. B. (2010). “Neural Network Toolbox 7 User Guide, Mathworks”.

John H. (1984) “Guidelines for preparing the research proposal”, University Press of America Inc, New York.

Belforte. G, Mauro, S.& Mattiazzo, G. (2004). “A Method for Increasing The Dynamic Performance Of Pneumatic Servo Systems With Digital Valves,” Mechatronics, Vol.14, pp. 1105–1120.

Brophy, B., Kelly, K., & Byrne, G. (2002). AI-based condition monitoring of the drilling process. Journal of Materials Processing Technology, 124, 305-310. pp. 115–130

Carpenter, G. A & Grossberg, S. (1987). “ART-2: Self-Organization of Stable Category Recognition Codes for Analog Input Pattern”, Applied Optics, Vol. 26, pp. 4919-4930.

Carpenter, G. A. et al. (1992). “Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning Of Analog Multidimensional Maps”, IEEE Transactions On Neural Networks, Vol. 3, pp. 698-713.

Carpenter, G. A., Grossberg S., & Reynolds J. H. (1991). “ARTMAP: Supervised Real-Time Learning And Classification of Nonstationary Data By A Self-Organizing Neural Networks” Vol. 4, pp.565-588.

Carpenter, G. A., Grossberg, S. & Rosen, D. B. (1991a). “ART 2-A: An Adaptive Resonance Algorithm For Rapid Category Learning And Recognition, Neural Networks,” Vol. 4, pp.493-504.

Carpenter, G. A., Grossberg, S. & Rosen, D. B. (1991b). “Fuzzy ART: Fast Stable Learning And Categorization Of Analog Patterns By An Adaptive Resonance System, Neural Networks” , Vol. 4, pp. 759-771.

Casini. M. Prattichizzo., D Vicino A.(2004),”The Automatic Control Teleb” IEEE Control System Magazine, 2004.

Castells, M. (2001), “Lessons from the History of Internet”, in “The Internet Galaxy”, Ch. 1, pp 9-35. Oxford University Press

Chen, C. & Mo, C. (2004). “ A Method for Intelligent Fault Diagnosis of Rotating Machinery, Digital Signal Processing,” Vol. 14, pp. 203–217.

Christos S and Dimitrion N (2001) “Introduction to ANN” “Research Report”

Clancey, J.W., Shortliffe, H.E. (Eds.), (1984).” Readings in Medical Artificial Intelligence”: The First Decease. Addison-Wesley, Reading, MA.

Curtis D. J (2006),” Process instrumentation technology,” New Delhi- 110001, Prentice Hall of India, 8th Edition.pp 21-25

Demetgul M.; Tansel IN. & Taskin S. (2009).” Fault Diagnosis of Pneumatic Systems with Artificial Neural Network Algorithms. Expert Systems with Applications,” Vol. 36, No. 7, pp. 10512-10519. Festo Didactic GmbH & Co. PneuPos Manuel, 2010.

Dorf R C. and Bishop R. H (2004) “Modern Control System” India. Pearson Publication P.L.C. India.

Efraim T, Jay E. Aronson J .E and Ting D L (2007), “Artificial Intelligence & Expert system.”New Delhi India.Prentice Hall Publication pp 200-205

Egwin W, Leena Y and Pasi J..(2003) "Web Based Monitoring And Control Of Industrial Processes". University of Oulu, Control Engineering Laboratory University of Oulu. pp 2-6

Eliane R and Kevin K (2002) 'Artificial Intelligence" Second Edition, ed. Mc Graw-Hill, pp-478-479

Eliana Z. (2000) **"**Process Monitoring And Control Using Artificial Neural Networks And Other Advanced Techniques**"** Ph. D. Dissertation, Dipartimento di Principi e Impianti di Ingegneria Chimica "I. Sorgato"Università di Padova Via Marzolo, 9 I-5131 Padova PD (Italy)

Furuya M., Kato H., Sekozawa T (2000) "Secure Web-Based Monitoring And Control System" Industrial Electronics Society, IECON 2000, 26th Annual Conference of the IEEE vol. 4, , pp. 2443-2448.

Gang, Z (2010) "Wireless Sensor Networks for Industrial Process Monitoring and Control: A Survey Digital Technology Laboratory"pp 2-4

Garrett, A. (2003). "Fuzzy ART and Fuzzy ARTMAP Neural Networks, Artificial Neural Networks - Industrial and Control Engineering Applications"

Giarratano, J & Riley, G (1993) "Expert Systems: Principal And Programming", Boston, MA: PWS Publishing (1993).

Grossberg S. (1987). "Competitive Learning: From Interactive Activation to Adaptive Resonance, Cognitive Science", Vol. 11, pp. 23-63.

Hagan M,. Demuth H, and Beale M (1996)." Neural Network Design'. PWS Publishing, Boston, MA,Center for Process Simulation, Department of Chemical Engineering, Faculty of Chemical & Natural Resources Engineering, Universiti Teknologi Malaysia, 80990 Johor Bahru, Malaysia

Hall D. (1986)" Microprocessors and Interfacing. Programming and Hardware" 2nd Edition, Glen Macmillian/Mcraw-Hill, Lake Forest, Columbus Illinois. pp. 340

Harish Ramamurthy, B. S. Prabhu and Rajit Gadh(2007) "Wireless Internet for the Mobile Enterprise Consortium" University of California, Los Angeles Los Angeles, California, pp 1-3.

Haydary J, Pavlik T,(2009) "Steady-State and Dynamic Simulation of Crude Oil Distillation using ASPEN Plus and ASPEN Dynamics, Petroleum and Coal," 51(2), 100-109.

Hossein , A(1995) "System Simulation: The Shortest Root to Application" University of Baltimor, [Internet]. pp. 1-3

Huang, P. T., & Chen, J. C. (2000). Neural network-based tool breakage monitoring system for end milling operations.Journal of Industrial Technology, 16 (2), 2-7.

Huang, R, Xi, L, Li, X. Liu, C.R, Qiu, H. and Lee, J. (2007). "Residual Life Predictions for Ball Bearings Based On Self-Organizing Map And Back Propagation Neural Network Methods", Mechanical Systems And Signal Processing, Vol. 21, pp.193–207.

Hung, S. Y., Liang, T. P., & Liu, V. W. (1996). Integrating arbitrage pricing theory and artificial neural networks to support portfolio management. Decision Support Systems, 18 , 301-316.

Hunt, K.J., Sbarbaro, D., Zbikowki, R., & Gawthrop, P.J.(1992), "Neural Networks for Control Systems - A Survey", *IEEE Transactions on Neural Networks*, Vol. 28, , pp. 1083-1112.

Hunt, K.J., & Sbarbaro, D.(1991), "Neural Networks for Nonlinear Internal Model Control", *IEE Proceedings -D.*, Vol. 138, , pp. 431-438.

Industrial Electronics Society, IECON (2000), 26th Annual Conference of the IEEE vol. 4, 22-28 October, pp. 2443-2448.

Inyiama H. C and Okezie C. C. (2007), "Designing Microcontroller- Based Universal Process Control Systems", Electroscope, volume Number 2,

Inyiama H.C and Mbonu E. S (2010), "A Cost-Effective Approach to Microcontroller-based SMS Application: A Case for MyX-SAGEM PHONE SERIES". Department of Electronics and Computer Engineering, Nnamdi Azikiwe University, Awka. INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH Vol. 3 No. 9.

Kaparthi, S., and Sure N.C. ( 1991) "A Neural Network System for Shape-based Classification and 193Coding of Rotational parts," International Journal of Production Research, 29 (9), ,pp.1771-1784.

Kaparthi, S., and Suresh, N.C (1992)., "Machine-component Cell Formation in Group Technology: A Neural Network Approach," International Journal of Production Research, Vol. 30, No. 6, pp. 1353-1367.

Karpenko, M.; Sepehri, N. & Scuse, D.(2003)." Diagnosis of Process Valve Actuator Faults Using a Multilayer Neural Network. Control Engineering Practice," Vol. 11, pp.12891299.

Kirubashankar, R Krishnemunthy K and India J (2009) "Remote Monitoring system for distributed Control of Industrial Plant Process" Journal of Scientific and Industrial Research Vol 68, PP –SS6- 860

L1 H, V, (2006) Web-based Remote Monitoring and Control for Process Plants" Preventing of International conference on Machine Learning and Cybernaties.

Lee, C.H, YU, T.M and chen J.C (2011) "Negative Neural Network Controller Design For A Class Of Non Linear Systems Using Simultaneous Per- Stochastic Approximation (SPSA)" processing of the international multi conference of Engineers and computer scientists (Vol. 11)

Lee, H., Malave, C. O., and Ramachandran, S. (1992) "A Selforganizing Neural Network Approach For The Design Of Cellular Manufacturing Systems," Journal of Intelligent Manufacturing (1992)3,325-332.

Lee, I. S.; Kim, J. T.; Lee, J. W.; Lee, D. Y. & Kim, K. Y. (2003)." Model-Based Fault Detection and Isolation Method Using ART2 Neural Network", International Journal of Intelligent Systems, Vol.18, pp. 1087–1100.

Lekan T. P , Gutti B & Alfred A. S (2013) "Expert System Design And Control Of Crude Oil Distillation Column Of A Nigerian Refinery Using Artificial Neural Network Model". IJRRAS volume15 issue 3. June 2013

Li, X. & Kao, I. (2005). "Analytical Fault Detection and Diagnosis (FDD) for Pneumatic Systems in Robotics and Manufacturing Automation. Intelligent Robots and Systems," IEEE/RSJ International Conference, pp.2517-2522.

Lu, P. J.; Hsu, T. C.; Zhang, M. C. & Zhang, J. (2000), "An Evaluation of Engine Fault Diagnostics Using Artificial Neural Networks", ASME J. Eng. Gas Turbines Power, Vol. 123, pp. 240–246.

McCarthy, J (1984)" Some Expert System Need Common Sense", Stanford University, Stanford, California.

Mashudi, M. R (2001). Forecasting Water Level Using Neural Network In The Operation Of Reservoir In Citarym Cascade, West Java Indonesia. Dissertation George Washington university.

Mendonça, L. F.; Sousa, J. M. C. & Costa, J. M. G. (2009)." An Architecture for Fault Detection And Isolation Based on Fuzzy Methods, Expert Systems With Applications." Vol. 36 pp. 1092–1104.

Mitchel, M.A Lopez, Fidaligo J.N and Nccalley J.A (2000) "Neural Network to predict the Dynamic Frequency Response of a Power System to an under-Frequency Load Shuddering" I.E.E.E Journal PP 346 – 351.

McCulloch, W. & Pitts W. (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133.

Mustafa Demetgul, Sezai Taskin and Ibrahim Nur Tansel (2007) "Conditioning Monitoring and Fault Diagnosis for a Servo-Pneumatic System with Artificial Neural Network Algorithms"

Na, C. & Lizhuang, M. (2008).“Pattern Recognition Based on Weighted and Supervised ART2”. International Conference on Intelligent System and Knowledge Engineering,Xiamen, China, Nov. 17-18, 2008,Vol. 2, pp. 98-102.

Nakutis, Ž. & Kaškonas, P. (2007. “Pneumatic Cylinder Diagnostics Using Classification Methods”. Instrumentation And Measurement Technology Conference Proceedings, Warsaw, 1-3 May 2007, pp.1-4.

Nakutis, Ž., Kaškonas, P. (2005).”Application of ANN for Pneumatic Cylinder Leakage Diagnostics”. Matavimai, Vol. 36, No.4, pp.16-21.

Nedovic, L. and V. Devedzic (2002) . “Expert Systems in Finance: A Cross-Section of the Field. ” Expert Systems with Applications, Vol 23. No. 1. Pp. 49-66. New Delhi India. Prentice Hall Publication

Nogami, T. et al. (1995). “Failure Diagnosis System On Pneumatic Control Valves By Neural Networks”. IEEE International Conference on Neural Networks, Vol. 2, pp. 724-729.

Norvilas, A., Negiz, A., DeCicco, J., Cinar, A., (2000).” Intelligent Process Monitoring By Interfacing Knowledge-Based System And Multivariate Statistical Monitoring”. Journal of Process Control 10, 341–350.

Nurminen, J., et al (2003) “What makes Expert Systems Survive Over 10 Years: Empirical Evaluation of Several Engineering Applications.” Expert Systems with Applications, Vol. 24, No. 1 pp. 199-211.

Ogunaike, B. A & Bay W. H., (1994) “Process Dynamics, Modeling and Control”, Oxford University Press.

Olabiyiosi etal (2004) “Web Intelligence Technologies: Challenges and Trends in New Information Age”- web conference proceedings vol. 15,

Omiyi B (2008) “Managing Change and Innovation in the Oil and Gas industry journal of Nigerian Institute of Management”, ISSN 0025-178X volume 44; number.

Peter S (2003) Introduction to Simulation and Modeling[Internet]

Power, Y.& Bahri, A.P., (2004).” Integration Techniques in Intelligent Operational Management: A Review”. Knowledge-Based Systems 20, 1–9.

Power, Y. & Bahri, A.P., (2004).” Integration Techniques In Intelligent Operational Management: A Review”. Knowledge-Based Systems 20, 1–9.

Prasad, R.P., Davis, F.J., Jirapinyo, Y., Josephson, R.J.& Bhalodia, M. (1998). "Structuring Diagnostic Knowledge for large-Scale Process Systems." Computers & Chemical Engineering 22 (12), 1897–1905.

Qingcang. Y., Chen . B, Cheng H. H. (2004)," Web-Based Control System Design and Analysis" IEEEE Control System Magazine,.

Qui b., Gooi H.B.: "Web-based SCADA display systems (WSDS) for Access via Internet,", Future Generation Computer Systems, Volume 18(5), pp. 595-611.

Ramamurthy B, Bhargavi S & ShashiKumar R (2010) "Development of a Low-Cost GSM SMS-Based Humidity Remote Monitoring and Control System for Industrial Applications" pp 1-4

Rajakarunakaran, S. Venkumar, P. Devaraj, D & Rao, K.S.P. (2008). "Artificial Neural Network Approach for Fault Detection in Rotary System", Applied Soft Computing, Vol. 8, No. 1, pp.740–748.

Rao, M., Xia, Q. (1994)." Integrated Distributed Intelligent System for Online Monitoring and Control of Pulp Process". Canadian Artificial Intelligence 33, 5–10.

Rao, M., Sun, X., Feng, J., (2000). "Intelligent System Architecture for Process Operation Support". Expert Systems with Applications 19, 279–288.

Rengaswamy, R., Venkatasubramanian, V., (1992). "An Integrated Framework for Process Monitoring, Diagnosis, and Control using Knowledge-based Systems and Neural Networks". IFAC, Delaware, USA, pp. 49–54.

Rumelhart, D. E. and McClelland, J. L.,(1986) , "Parallel Distributed Processing: Explorations in the Microstructure of Cognition". Volume 1: Foundations, MIT Press, Cambridge, MA. pp 318-362.

Saipem (2005) Soku LGSP De- Dottlenecking Project
Process Control Philosophy of SPDC,F10044-100-pcs-1360-01029

Seema C, Mitra, R and Vijah K (2007) "Neural Network/Fuzzy Controller for Multiple Input Multiple Output (MIMO)" World Academy of science Engineering and Technology PP. 485 – 491

Sepasi, M. & Sassani, F. (2010).' On-line Fault Diagnosis of Hydraulic Systems Using Unscented Kalman Fitler", International Journal of Control and Automation Systems, Vol. 8, No. 1, pp. 149-156.

Shi, L. & Sepehri, N. (2004). "Adaptive Fuzzy-Neural-Based Multiple Models for Fault Diagnosis of a Pneumatic Actuator," Proceeding of the 2004 American Control Conference, Boston. Massachusetts, June 30 -July 2, 2004.

Smith, C.A., & Carripio, A. B (1985}, "Principle and Practice of Automatic Process Control," John Wiley and Sons

Sommerville I (2011). "Software Engineering" 9th edition, pp 538, Pearson Publication New York, USA.

Song, S. O.; Lee, G. & Yoon, E.N. (2004). "Process Monitoring of an Electro-Pneumatic Valve Actuator Using Kernel Principal Component Analysis", Proceedings of the International Symposium on Advanced Control of Chemical Processes. Hong Kong,China,

Stephanopolous G (1984) "Chemical Process Control- An Introduction to Theory and Practice", prentice-hall of india private limited.pp 55-60

Stuart R & Norving P, (1995) "Intelligent Agents (Artificial Intelligence)", A modern Approach "Practice Hall Series in intelligence , Prentice Hall incorporated PP 31-52

Sun, C.L., Uraikul, V., Chan, W.C., Tontiwachwuthikul, P., (2000). "An Integrated Expert System/Operation Research Approach for Automation Of Natural Gas Pipeline Operations." Engineering Applications Artificial Intelligence 13 (4), 465–475.

Takosoglu, J. E.; Dindorf, R. F. & Laski, P. A. (2009). "Rapid Prototyping Of Fuzzy Controller Pneumatic Servo-System". International Journal of Advanced Manufacturing Technology, Vol. 40, No.3/4, pp. 349-361.

Tansel I. N.; Demetgul M. & Sierakowski R. L. (2009)." Detemining Initial Design Parameters By Using Genetically Optimized Neural Network Systems." Composite Materials Technology: Neural Network Applications. Taylor & Francis(CRC Press).

Tansel, I. N.; Demetgul, M.; Leon, R. A.; Yenilmez, A. & Yapici A.(2009). "Design of Energy Scavengers of Structural Health Monitoring Systems" Using Genetically Optimized Neural Network Systems, Sensors and Materials, Vol. 21, No. 3, pp. 141-153.

Taylor, J. H. & Sayda, A. F (2005). "An Intelligent Architecture for Integrated Control and Asset Management for Industrial Processes". Joint International Symposium on Intelligent Control & 13th Mediterranean Conference on Control and Automation (ISIC-MED).

Teo, J & Darwin, R (2004)." Evolutionary Robotics: Challenges in Automatic Robot Synthesis". 2nd International Conference on Artificial Intelligence in Engineering and Technology (ICAET 2004). Kota Kinabablu, Sabah, Malaysia, August 2004, Vol 1, pp 7-13.

Teo, J (2005) "Evolutionary Multi-Objective Optimization for Automatic Synthesis of Artificial Neural Network Robot Controllers". Malaysian Journal of Computer Science, volume 18, number 2, pp 54-62.

Timothy J, S, Julie K.S & Tomás V (2010) "The Application of Artificial Neural Networks to Monitoring and Control of an Induction Hardening Process". (IJACSA) International Journal of Advanced Computer Science and Applications,

Tonnang Z.E.H.(2010) "Distillation Column Control Using Artificial Neural Networks", M. Sc Thesis, Microprocessors and Control Engineering, Department of Electrical and Electronics Engineering, Faculty of Technology, University of Ibadan,

Umair M. S & Usman R. (2007)" Automation of Irrigation System Using ANN based Controller. Research Centre for Modeling & Simulation," National University of Science and Technology, Pakistan. International Journal of Electrical & Computer Sciences IJECS-IJENS Vol:10 No:02 45

Uppal, F.J.; Patton, R.J. & Palade, V. (2002). "Neuro-Fuzzy Based Fault Diagnosis Applied to an Electro-Pneumatic Valve", Proceedings of the 15th IFAC World Congress, Barcelona, Spain, pp. 2483-2488.

Uraikul, V., Chan, W.C., Tontiwachwuthikul, P.,(2000)." Development Of An Expert System For Optimizing Natural Gas Pipeline Operations". Expert System with Applications 18, 271–282.

Wang, J. et al. (2004)." Identification of Pneumatic Cylinder Friction Parameters Using Genetic Algorithms". IEEE/ASME Transactions on Mechatronics, Vol. 9, No.1, pp. 100-104. Artificial Neural Networks - Industrial and Control Engineering Applications

Wang L and Tan K.C (2006)"Modern Industrial Automation Software Design" Institute of Electrical and Electronics Engineers, Imc

Wang L and Tan K.C (2006)" Design Principles of Modern Industrial Automation Systems": Institute Of Electrical And Electronics Engineers, Imc

Wang, Z.X., Liu, Y., Griffin, J.P., (1998). "A combined ANN and Expert System Tool For Transformer Fault Diagnosis." IEEE Transactions on Power Delivery 13, 1224–1229.

Ward S. H (2007) "Using Neural Networks To Forecast Flood Events: A Proof Of Concept". A Dissertation Submitted to the H. Wayne Huizenga School of Business and Entrepreneurship Nova Southeastern University. Pp 4-6

Warnier E, Yliniemi L & Joensuu P(2003), "Web Based Monitoring And Control Of Industrial Processes",. 15 p.. ISBN 951-42-7173-4.

Wolf. W (2010). http://www.embedded.com/design/prototyping-and development/ 4008904/3/Picking-the-right-system-design-methodology-for-your-embedded-apps-Part-1-3

Xia, Q., Rao, M., (1999b)." Knowledge Architecture And System Design For Intelligent Operation Support Systems". Expert Systems with Applications 17, 115–127.

Yang S.H., Chen X., &Alty J.L.(2003) "Design Issues And Implementation Of Internet-Based Process Control Systems," Control Engineering Practice, Volume II (6), pp. 709-720.

Yang, B. S.; Han, T. & An J. L. (2004). "ART–KOHONEN Neural Network for Fault Diagnosis of Rotating Machinery", Mechanical Systems and Signal Processing, Vol. 18, pp. 645–657.

Zalizawati, B. A (2008) "Development of Multiple-Input Multiple-Output and Multiple-Input Single-Output Neural Network Models for Continuous Distillation Column", M. Sc. Thesis, School of Chemical Engineering, Malaysia University, Malaysia (2008).

Zhao, G (2010). Wireless Sensor Networks for Industrial Process Monitoring and Control: A Survey Digital Technology Laboratory Davis, CA, 95618, United States. pp 1-3

http://www.disagroup.com 2011-06-11

www.neurosolutions.com/products/ns/whatsNNhtml.

www.Fucpubstjohns.edu/-rolferm/4322/chapter 12

www.neurosolutions.com/products/ns/whatsnn.html

www.embededd.com/design/prototyping&development

Wikepedia.org/software_developmet_process.

userpages.umbc.edu/~khoo/survey1.html.

userpages.umbc.edu/~khoo/survey2.html.

http://freecircuitdiagram.com/2009/04/24/ttl-rs232-level-converter-using-max232-ic/

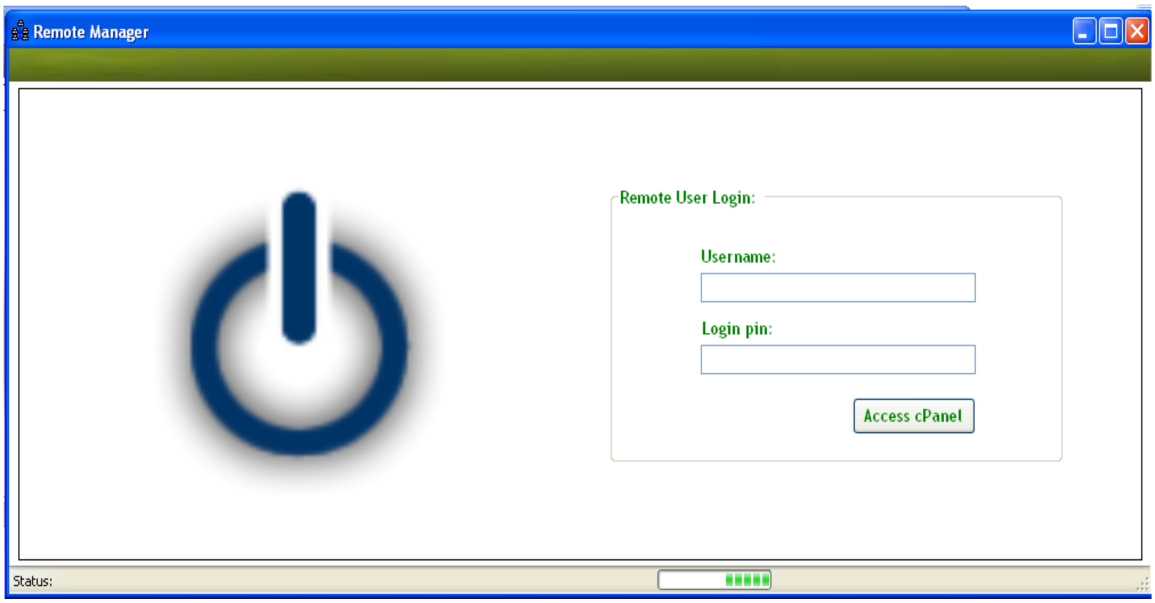
http://www.engineersgarage.com/electronic-components/at89c51-microcontroller-datasheet

http://microcontroller.circuitlab.org/2012/02/schematic-max-232-interfacing-with.html

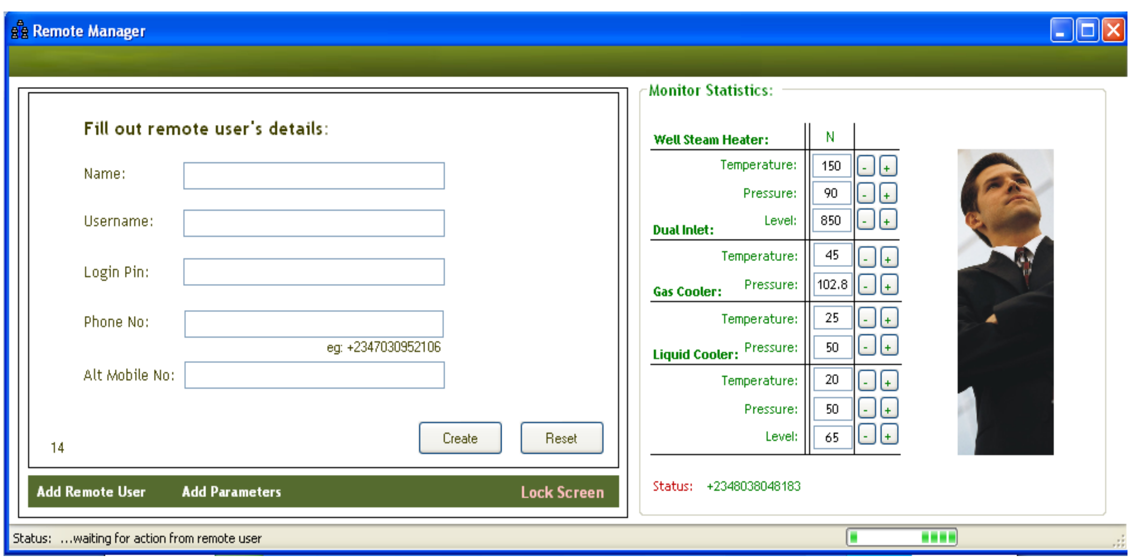
https://www.google.com/search?q=max232+pin+diagram&ie=utf-8&oe=utf-8

# Appendix A
# (Source Code for SMS Interface)

## User Interface




Login Form




Remote Contro Panel

```
Imports System.IO
Imports MySql.Data.MySqlClient
Module Module1


'Mysql Database connection
Public conn As MySqlConnection
Public query, action, path As String 'dim individual query
Public connStr As String = "Database=remotemanager;" & "Data Source=localhost;" & _
                  "User Id=root;Password=explore;" '& "Connection Timeout=20"
End Module

Imports Vb6 = Microsoft.VisualBasic : Imports MySql.Data.MySqlClient
Imports System.Drawing.Drawing2D

Public Class Form1
   Public pina As Integer : Public msg As String
   Public mNo As String = "07030952106" : Public curUser As String
   Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Integer)

   Sub HandleFrame()
      'Handle Frame display modals.
      Panel1.Visible = False
      Panel2.Visible = False
      'pane3.Visible = False
      'pane4.Visible = False

      Select Case pina
         Case 1
            Panel1.Visible = True
         Case 2
            Panel2.Visible = True
         Case 3
            'Panel3.Visible = True
         Case 4
            'pane4.Visible = True
      End Select

      pina = 0
   End Sub

   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
      LoginScreen.Visible = True : T1.Select()
      defualtValue()
   End Sub
```

```
    Sub defualtValue()
        ' Normal value for all the variable

        'Well steam heater:
        m1.Text = 150      'Temperature
        m2.Text = 90       'Pressure
        m3.Text = 850      'Level

        'Dual Inlet Monitored:
        m4.Text = 45       'Temperature
        m5.Text = 102.8    'Pressure

        'Gas Cooler:
        m6.Text = 25       'Temperature
        m7.Text = 50       'Pressure

        'Liquid Cooler:
        m8.Text = 20       'Temperature
        m9.Text = 50       'Pressure
        m10.Text = 65      'Level

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        If T2.Text = "" Then
            MsgBox("Please enter login pin", MsgBoxStyle.Information)
            T2.Select() : Exit Sub
        End If

        If T2.Text = "admin" Then
            LoginScreen.Visible = False
            Base.Visible = True : T2.Text = ""
            pina = 0 : HandleFrame()
        Else
            MsgBox("Invalid login pin, try again", MsgBoxStyle.Information)
            T2.Text = "" : T2.Select() : Exit Sub
        End If
    End Sub

    Private Sub cmdReset_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdReset.Click
        TextBox1.Text = "" : TextBox2.Text = "" : TextBox3.Text = ""
        TextBox4.Text = "" : TextBox5.Text = ""
    End Sub
```

```
    Private Sub lblAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblAdd.Click
        pina = 1 : HandleFrame()
    End Sub

    Private Sub lblLogout_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblLogout.Click
        LoginScreen.Visible = True
        'Base.Visible = False
    End Sub

    Private Sub cmdCreate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCreate.Click

        If TextBox1.Text = "" Then
            MsgBox("Remote user's name is required.", MsgBoxStyle.Information)
            TextBox1.Select() : Exit Sub
        ElseIf TextBox2.Text = "" Then
            MsgBox("Username is required.", MsgBoxStyle.Information)
            TextBox2.Select() : Exit Sub
        ElseIf TextBox3.Text = "" Then
            MsgBox("A login pin is required.", MsgBoxStyle.Information)
            TextBox3.Select() : Exit Sub
        ElseIf TextBox4.Text = "" Then
            MsgBox("Remote user mobile no is required.", MsgBoxStyle.Information)
            TextBox4.Select() : Exit Sub
        ElseIf IsNumeric(TextBox4.Text) = False Then
            MsgBox("Remote mobile no must be number.", MsgBoxStyle.Information)
            TextBox4.Select() : Exit Sub
        End If


        Try
            Dim query As String = _
            "INSERT INTO remoteuser(name, username, password, phone, mobile) VALUES('" & _
            TextBox1.Text & "', '" & TextBox2.Text & "', '" & TextBox3.Text & "', '" & _
            TextBox4.Text & "', '" & TextBox5.Text & "')"

            Dim connection As New MySqlConnection(connStr)
            Dim cmd As New MySqlCommand(query, connection)
            connection.Open()
            cmd.ExecuteNonQuery()
            connection.Close()

            Dim msg As String
```

```
            msg = MsgBox("Remote user account created successfully." & vbNewLine & _
            "Send remote user details vis SMS", 64 + 4)
            If msg = vbYes Then
                'Call smsSender()
                TextBox1.Text = "" : TextBox2.Text = "" : TextBox3.Text = ""
                TextBox4.Text = "" : TextBox5.Text = ""
            End If
            Return
        Catch ex As Exception
            MsgBox("Saving Problem! " + ex.Message)
        End Try
    End Sub

    Private Sub C1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CB1.SelectedIndexChanged

    End Sub

    Private Sub lblAdd1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblAdd1.Click
        pina = 2 : HandleFrame()
    End Sub

    Private Sub cmdRecall_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdRecall.Click
        TB1.Text = "" : TB2.Text = "" : CB1.ResetText()
    End Sub

    Private Sub addPara_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles addPara.Click
        If CB1.Text = "" Then
            MsgBox("Please Select Monitoring  Parameter.", MsgBoxStyle.Information)
            Exit Sub
        ElseIf TB1.Text = "" Then
            MsgBox("Please enter " & CB1.Text & "'s Min. Value", MsgBoxStyle.Information)
            TB1.Select() : Exit Sub
        ElseIf IsNumeric(TB1.Text) = False Then
            MsgBox("Min. Value must be number.", MsgBoxStyle.Information)
            TB1.Text = "" : TB1.Select() : Exit Sub
        ElseIf TB2.Text = "" Then
            MsgBox("Please enter " & CB1.Text & "'s Max. Value", MsgBoxStyle.Information)
            TB2.Select() : Exit Sub
        ElseIf IsNumeric(TB2.Text) = False Then
            MsgBox("Max. Value must be number.", MsgBoxStyle.Information)
            TB2.Text = "" : TB2.Select() : Exit Sub
        End If
```

```
        Try
            Dim query As String = _
            "INSERT INTO parameter(monitor, min, max) VALUES('" & _
            CB1.Text & "', '" & TB1.Text & "', '" & TB2.Text & "')"

            Dim connection As New MySqlConnection(connStr)
            Dim cmd As New MySqlCommand(query, connection)
            connection.Open()
            cmd.ExecuteNonQuery()
            connection.Close()

            MsgBox("Parameter added successfully.", MsgBoxStyle.Information)

            TB1.Text = "" : TB2.Text = ""

            Return
        Catch ex As Exception
            MsgBox("Problem! " + ex.Message)
        End Try
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
        Label13.Text = Now
        spbar.Value += 10

        Label15.Text = Label15.Text + 1

        If Label15.Text = 2 Then
            Timer3.Enabled = True
            'Label7.Text = "am back"
        End If

        If Label15.Text > 200 Then
            Label15.Text = 0
            Timer3.Enabled = False
            'Label7.Text = "am off"
        End If

        If spbar.Value = 90 Then
            spbar.Value = 0
        End If

    End Sub
```

```
    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer2.Tick
        On Error Resume Next
        Call checkIrregularity()
        'Label14.Text = Now

    End Sub

    Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer3.Tick
        Call searchAction()
    End Sub

    Sub checkIrregularity()
        ' Well Steam Heater =============
        If m1.Text <= 122 Then
            msg = "Well Steam Heater's temperature is getting too low, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Well Steam Heater's temperature is getting too low"
            Call nextAct()
        ElseIf m1.Text >= 158 Then
            msg = "Well Steam Heater's temperature is getting too high, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Well Steam Heater's temperature is getting too high"
            Call nextAct()

        ElseIf m2.Text <= 85 Then
            msg = "Well Steam Heater's pressure is getting too low, send SET to " & mNo & " to
normalize pressure."
            sbar.Text = "Well Steam Heater's pressure is getting too low"
            Call nextAct()
        ElseIf m2.Text >= 95 Then
            msg = "Well Steam Heater's pressure is getting too high, send SET to " & mNo & " to
normalize pressure."
            sbar.Text = "Well Steam Heater's pressure is getting too high"
            Call nextAct()

        ElseIf m3.Text <= 807 Then
            msg = "Well Steam Heater's level is getting too low, send SET to " & mNo & " to
normalize level."
            sbar.Text = "Well Steam Heater's level is getting too low"
            Call nextAct()
        ElseIf m3.Text >= 893 Then
            msg = "Well Steam Heater's level is getting too high, send SET to " & mNo & " to
normalize level."
            sbar.Text = "Well Steam Heater's level is getting too high"
```

```
            Call nextAct()

            'Dual Inlet Monitor ================
        ElseIf m4.Text <= 42.5 Then
            msg = "Dual Inlet Monitor's temperature is getting too low, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Dual Inlet Monitor's temperature is getting too low"
            Call nextAct()
        ElseIf m4.Text >= 48 Then
            msg = "Dual Inlet Monitor's temperature is getting too high, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Dual Inlet Monitor's temperature is getting too high"
            Call nextAct()

        ElseIf m5.Text <= 97 Then
            msg = "Dual Inlet Monitor's level is getting too low, send SET to " & mNo & " to
normalize level."
            sbar.Text = "Dual Inlet Monitor's level is getting too low"
            Call nextAct()
        ElseIf m5.Text >= 108 Then
            msg = "Dual Inlet Monitor's level is getting too high, send SET to " & mNo & " to
normalize level."
            sbar.Text = "Dual Inlet Monitor's level is getting too high"
            Call nextAct()

            'Gas Cooler ================
        ElseIf m6.Text <= 23 Then
            msg = "Gas Cooler's temperature is getting too low, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Gas Cooler's temperature is getting too low"
            Call nextAct()
        ElseIf m6.Text >= 27 Then
            msg = "Gas Cooler's temperature is getting too high, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Gas Cooler's temperature is getting too high"
            Call nextAct()

        ElseIf m7.Text <= 47 Then
            msg = "Gas Cooler's level is getting too low, send SET to " & mNo & " to normalize
level."
            sbar.Text = "Gas Cooler's level is getting too low"
            Call nextAct()
        ElseIf m7.Text >= 53 Then
            msg = "Gas Cooler's level is getting too high, send SET to " & mNo & " to normalize
level."
            sbar.Text = "Gas Cooler's level is getting too high"
```

```
            Call nextAct()

            ' Liquid Cooler =============
        ElseIf m8.Text <= 15 Then
            msg = "Liquid Cooler's temperature is getting too low, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Liquid Cooler's temperature is getting too low"
            Call nextAct()
        ElseIf m8.Text >= 25 Then
            msg = "Liquid Cooler's temperature is getting too high, send SET to " & mNo & " to
normalize temperature."
            sbar.Text = "Liquid Cooler's temperature is getting too high"
            Call nextAct()

        ElseIf m9.Text <= 45 Then
            msg = "Liquid Cooler's pressure is getting too low, send SET to " & mNo & " to
normalize pressure."
            sbar.Text = "Liquid Cooler's pressure is getting too low"
            Call nextAct()
        ElseIf m9.Text >= 55 Then
            msg = "Liquid Cooler's pressure is getting too high, send SET to " & mNo & " to
normalize pressure."
            sbar.Text = "Liquid Cooler's pressure is getting too high"
            Call nextAct()

        ElseIf m10.Text <= 60 Then
            msg = "Liquid Cooler's level is getting too low, send SET to " & mNo & " to normalize
level."
            sbar.Text = "Liquid Cooler's level is getting too low"
            Call nextAct()
        ElseIf m10.Text >= 70 Then
            msg = "Liquid Cooler's level is getting too high, send SET to " & mNo & " to normalize
level."
            sbar.Text = "Liquid Cooler's level is getting too high"
            Call nextAct()
        End If
    End Sub

    Sub nextAct()
        ' Send SMS and Insert " " into the database
        Timer2.Enabled = False ' checkIrregularity
        Call smsSender() : Call DBReset()
    End Sub

    Sub smsSender()
        Try
```

```
        With MSC1

            .CommPort = 4
            .Settings = "9600,N,8,1"
            .Handshaking = MSCommLib.HandshakeConstants.comRTS
            .RTSEnable = True
            .DTREnable = True
            .RThreshold = 1
            .SThreshold = 1
            .InputMode = MSCommLib.InputModeConstants.comInputModeText
            .InputLen = 0
            .PortOpen = True 'must be the last

            ' Send an 'AT' command to the phone
            MSC1.Output = "AT" & vbCrLf
            Sleep(500)
            MSC1.Output = "AT+CMGF=1" & vbCrLf 'This line can be removed if your modem
will always be in Text Mode...
            Sleep(500)
            MSC1.Output = "AT+CMGS=" & Chr(34) & curUser & Chr(34) & vbCrLf 'Replace
this with your mobile Phone's No.
            Sleep(1000)
            MSC1.Output = "Remote Server: " & vbNewLine & msg & Chr(26)
            Sleep(2000)
            .PortOpen = False
        End With

      Catch ex As Exception
        MsgBox("Phone or Modem not found " & vbNewLine & ex.Message)
        defualtValue() : Timer2.Enabled = True

      End Try

    End Sub

    Sub DBReset()
      Dim updet As String = "UPDATE parameter SET action = '" & "" & _
                  "' WHERE monitor  = '" & "problem" & "'"

      Dim connection As New MySqlConnection(connStr)
      Dim cmd As New MySqlCommand(updet, connection)
      connection.Open()

      Try
        cmd.ExecuteNonQuery()
        Return
```

```
      Catch ex As Exception
         MsgBox("Error:! " + ex.Message)
      End Try

      connection.Close()
   End Sub

   Sub searchAction()
      Dim chk As Boolean
      Try
         Dim query As String = "SELECT * FROM  parameter"
         Dim connection As New MySqlConnection(connStr)
         Dim cmd As New MySqlCommand(query, connection)

         connection.Open()

         Dim reader As MySqlDataReader
         reader = cmd.ExecuteReader()

         While reader.Read()
            If (reader.GetString("action")) = "SET" Then
               chk = True
               Call defualtValue()
               Timer2.Enabled = True
            Else
               sbar.Text = "...waiting for action from remote user "
            End If
         End While

         reader.Close()
         connection.Close()

         Call DBReset()  ' Reset Database

      Catch ex As Exception
         MsgBox("Searching wahala " & ex.Message)
      End Try
   End Sub

   Private Sub b1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b1.Click
      m1.Text = Val(m1.Text) - 1
   End Sub

   Private Sub b2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b2.Click
```

```
        m1.Text = Val(m1.Text) + 1
    End Sub

    Private Sub b3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b3.Click
        m2.Text = Val(m2.Text) - 1
    End Sub

    Private Sub b4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b4.Click
        m2.Text = Val(m2.Text) + 1
    End Sub

    Private Sub b5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b5.Click
        m3.Text = Val(m3.Text) - 1
    End Sub

    Private Sub b6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b6.Click
        m3.Text = Val(m1.Text) + 1
    End Sub

    Private Sub b7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b7.Click
        m4.Text = Val(m4.Text) - 1
    End Sub

    Private Sub b8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b8.Click
        m4.Text = Val(m4.Text) + 1
    End Sub

    Private Sub b9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b9.Click
        m5.Text -= 1
    End Sub

    Private Sub b_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
b.Click
        m5.Text += 1
    End Sub

    Private Sub b11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b11.Click
        m6.Text -= 1
```

```
    End Sub

    Private Sub b12_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b12.Click
        m6.Text += 1
    End Sub

    Private Sub b13_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b13.Click
        m7.Text -= 1
    End Sub

    Private Sub b14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b14.Click
        m7.Text += 1
    End Sub

    Private Sub b15_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b15.Click
        m8.Text -= 1
    End Sub

    Private Sub b16_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b16.Click
        m8.Text += 1
    End Sub

    Private Sub b17_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b17.Click
        m9.Text -= 1
    End Sub

    Private Sub b18_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b18.Click
        m9.Text += 1
    End Sub

    Private Sub b20_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b20.Click
        m10.Text -= 1
    End Sub

    Private Sub b21_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles b21.Click
        m10.Text += 1
    End Sub
```

```
   Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnLogin.Click
        If T1.Text = "" Or T2.Text = "" Then
            MsgBox("Please enter your full login details", MsgBoxStyle.Information)
            Exit Sub
        End If
        Dim chk As Boolean
        Try
            Dim query As String = "SELECT * FROM  remoteuser"
            Dim connection As New MySqlConnection(connStr)
            Dim cmd As New MySqlCommand(query, connection)

            connection.Open()
            Dim reader As MySqlDataReader
            reader = cmd.ExecuteReader()

            While reader.Read()
                If (T1.Text = reader.GetString("username")) And (T2.Text =
reader.GetString("password")) Then
                    chk = True
                    LoginScreen.Visible = False
                    txtUser.Text = reader.GetString("phone")
                    curUser = reader.GetString("phone")
                    T1.Text = "" : T2.Text = ""
                End If
            End While
            reader.Close()
            connection.Close()

            Call DBReset()  ' Reset Database

        Catch ex As Exception
            MsgBox("Login wahala " & ex.Message)
        End Try

        If chk = False Then
            MsgBox("Invalid Login Details", MsgBoxStyle.Information)
            T1.Text = "" : T2.Text = ""
        End If

    End Sub
End Class
```

**Appendix B**
**(Source Code of Inlet Seperator)**

```
Line  I  Addr  Code        Source

  1:             B      A0    adc_a          bit p2.0
  2:             B      A1    adc_b          bit p2.1
  3:             B      A2    adc_c          bit p2.2
  4:             B      A3    adc_start      bit p2.3
  5:             B      A4    adc_ale        bit p2.4
  6:             B      A5    adc_clk        bit P2.5
  7:             B      B2    latch_1 bit p3.2
  8:             B      B3    latch_2 bit p3.3
  9:             B      B4    latch_3 bit p3.4
 10:             B      B5    latch_4 bit p3.5
 11:             B      B6    buzzer  bit p3.6
 12:             N    0030     phase_1 equ 30h
 13:             N    0031     phase_2 equ 31h
 14:             N    0032     phase_3 equ 32h
 15:             N    0033     ph1_hund equ 33h
 16:             N    0034     ph1_ten equ 34h
 17:             N    0035     ph1_unit equ 35h
 18:             N    0036     ph2_hund equ 36h
 19:             N    0037     ph2_ten equ 37h
 20:             N    0038     ph2_unit equ 38h
 21:             N    0039     ph3_hund equ 39h
 22:             N    003A     ph3_ten equ 3ah
 23:             N    003B     ph3_unit equ 3bh
 24:             N    003C      ph4_hund equ 3ch
 25:             N    003D     ph4_ten equ 3dh
 26:             N    003E     ph4_unit equ 3eh
 27:             N    003F      ph5_hund equ 3fh
 28:             N    0040     ph5_ten equ 40h
 29:             N    0041     ph5_unit equ 41h
 30:             N    0042      ph6_hund equ 42h
 31:             N    0043     ph6_ten equ 43h
 32:             N    0044     ph6_unit equ 44h
 33:             N    0045      ph7_hund equ 45h
 34:             N    0046     ph7_ten equ 46h
 35:             N    0047     ph7_unit equ 47h
 36:             N    0048      ph8_hund equ 48h
 37:             N    0049     ph8_ten equ 49h
 38:             N    0050     ph8_unit equ 50h
 39:             N    0051     keep_data equ 51h
 40:             N    0000    Org 0000h
 41:      0000  C3            clr c
```

```
0001  C2 B6         clr buzzer
0003  75 90 FF      mov p1,#0ffh
Line  I  Addr  Code        Source
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0006  75 89 20      MOV TMOD,#20H ;timer 1, mode 2
0009  75 8D FD      MOV TH1,#-3H ;4800 baud rate
000C  75 98 50      MOV SCON,#50H ;8-bit, 1 stop, REN enabled
000F  D2 8E         SETB TR1 ;sta
0011  90 05 B9      mov dptr ,# humid_display
0014  12 09 89              CALL TRANS


                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0017                start:
                    ;call read_sensor_0
0017  12 00 54      call adc_process
                      ;mov phase_1,a
                    ;call convert1

                      ;call  display1
                      ;call delay
001A 80 FB          jmp start
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


001C                read_sensor_0:
001C C2 A0                  clr adc_a                   ;
001E  C2 A1                 clr adc_b                   ;Select Channel 0
0020  C2 A2                 clr adc_c
0022  22                    ret

                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0023                read_sensor_1:
0023  D2 A0                 setb adc_a                  ;
0025  C2 A1                 clr adc_b                   ;Select Channel 0
0027  C2 A2                 clr adc_c
0029  22                    ret
                    ;;;;;;;;;;;;;;;;;;;;;;;
002A                read_sensor_2:
002A C2 A0                  clr adc_a                   ;
002C D2 A1                  setb adc_b                  ;Select Channel 0
002E  C2 A2                 clr adc_c
0030  22                    ret
                    ;;;;;;;;;;;;;;;;;;;;;;;
0031                read_sensor_3:
0031  D2 A0                 setb adc_a                  ;
```

```
0033  D2 A1                 setb adc_b                    ;Select Channel 0
0035  C2 A2                 clr adc_c
0037  22                    ret
                       ;;;;;;;;;;;;;;;;;;;;;;
0038                   read_sensor_4:
0038  C2 A0                 clr adc_a                    ;
003A  C2 A1                 clr adc_b                    ;Select Channel 0
003C  D2 A2                 setb adc_c
003E  22                    ret
                       ;;;;;;;;;;;;;;;;;;;;;;
003F                   read_sensor_5:
003F  D2 A0                 setb adc_a                    ;
0041  C2 A1                 clr adc_b                    ;Select Channel 0

Line  I  Addr  Code        Source
0043  D2 A2                 setb adc_c
0045  22                    ret
                       ;;;;;;;;;;;;;;;;;;;;;;
0046                   read_sensor_6:
0046  C2 A0                 clr adc_a                    ;
0048  D2 A1                 setb adc_b                    ;Select Channel 0
004A  D2 A2                 setb adc_c
004C  22                    ret
                       ;;;;;;;;;;;;;;;;;;;;;;
004D                   read_sensor_7:
004D  D2 A0                 setb adc_a                    ;
004F  D2 A1                 setb adc_b                    ;Select Channel 0
0051  D2 A2                 setb adc_c
0053  22                    ret


0054                   adc_process:
0054  75 90 FF         mov p1,#0ffh
0057  C2 B2             clr latch_1        ;;;;;;;;;;;;;;;;;;;;;;   this for temperature
0059  D2 B3            setb latch_2        ;;;;;;;;;;;;;;;;;;;;;;
005B  D2 B4            setb latch_3
005D  D2 B5            setb latch_4

005F  11 1C                call read_sensor_0
0061  C2 A4                clr adc_ale
0063  C2 A3                clr adc_start

                                                ;
0065  12 04 99             call delay_small
0068  D2 A4                setb adc_ale                  ;ale pin high
006A  12 04 99             call delay_small
```

```
006D D2 A3        setb adc_start          ;start pin high
006F  12 04 99    call delay_small
0072  C2 A4       clr adc_ale               ;ale pin low
0074  12 04 99    call delay_small
0077  C2 A3       clr adc_start           ;start pin low
0079  12 04 A6    call delay_long
007C E5 90        mov a,P1
007E  24 42       add a,#66    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;lllll
0080  F5 51       mov keep_data,a
0082  12 04 99    call delay_small
0085  90 06 95    mov dptr ,#temp_0
0088  12 09 89    call trans
008B  E5 51       mov a,keep_data
008D 12 04 01     call sensor_1
0090  90 09 6D    mov dptr ,#temp
0093  12 09 89    call trans
0096  12 09 9A    call temp_check ;;;;check temperature
0099  C3          clr c
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
009A 11 23        call read_sensor_1
009C C2 A4        clr adc_ale
009E  C2 A3       clr adc_start
                              ;
00A0 12 04 99     call delay_small
Line  I  Addr  Code        Source

00A3 D2 A4        setb adc_ale             ;ale pin high
00A5 12 04 99     call delay_small
00A8 D2 A3        setb adc_start          ;start pin high
00AA12 04 99      call delay_small
00ADC2 A4         clr adc_ale               ;ale pin low
00AF 12 04 99     call delay_small
00B2 C2 A3        clr adc_start           ;start pin low
00B4 12 04 A6     call delay_long
00B7 E5 90        mov a,P1
00B9 24 42        add a,#66
00BB F5 51        mov keep_data,a
00BD12 04 99      call delay_small
00C0 90 06 AB     mov dptr ,#temp_1
00C3 12 09 89     call trans
00C6 E5 51        mov a,keep_data
00C8 12 04 14     call sensor_2
00CB 90 09 6D     mov dptr ,#temp
00CE 12 09 89     call trans
00D1 12 09 9A     call temp_check ;;;;check temperature
```

```
00D4 C3             clr c
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
00D5 11 2A          call read_sensor_2
00D7 C2 A4          clr adc_ale
00D9 C2 A3          clr adc_start
                                   ;
00DB 12 04 99       call delay_small
00DE D2 A4          setb adc_ale               ;ale pin high
00E0 12 04 99       call delay_small
00E3 D2 A3          setb adc_start             ;start pin high
00E5 12 04 99       call delay_small
00E8 C2 A4          clr adc_ale                ;ale pin low
00EA 12 04 99       call delay_small
00ED C2 A3          clr adc_start              ;start pin low
00EF 12 04 A6       call delay_long
00F2 E5 90          mov a,P1
00F4 24 42          add a,#66
00F6 F5 51          mov keep_data,a
00F8 12 04 99       call delay_small
00FB 90 06 BF       mov dptr ,#temp_2
00FE 12 09 89       call trans
0101 E5 51          mov a,keep_data
0103 12 04 27       call sensor_3
0106 90 09 6D       mov dptr ,#temp
0109 12 09 89       call trans
010C 12 09 9A       call temp_check ;;;;check temperature
010F C3             clr c
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0110 11 31          call read_sensor_3
0112 C2 A4          clr adc_ale
0114 C2 A3          clr adc_start
                                   ;
0116 12 04 99       call delay_small
0119 D2 A4          setb adc_ale               ;ale pin high
011B 12 04 99       call delay_small
Line  I  Addr  Code        Source

011E D2 A3          setb adc_start             ;start pin high
0120 12 04 99       call delay_small
0123 C2 A4          clr adc_ale                ;ale pin low
0125 12 04 99       call delay_small
0128 C2 A3          clr adc_start              ;start pin low
012A 12 04 A6       call delay_long
012D E5 90          mov a,P1
```

```
012F  24 42      add a,#66
0131  F5 51      mov keep_data,a
0133  12 04 99   call delay_small
0136  90 06 D3   mov dptr ,#temp_3
0139  12 09 89   call trans
013C  E5 51      mov a,keep_data
013E  12 04 3A   call sensor_4
0141  90 09 6D   mov dptr ,#temp
0144  12 09 89   call trans
0147  12 09 9A   call temp_check ;;;;check temperature
014A  C3         clr c
                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
014B  11 38      call read_sensor_4
014D  C2 A4      clr adc_ale
014F  C2 A3      clr adc_start

                                      ;
0151  12 04 99   call delay_small
0154  D2 A4      setb adc_ale                    ;ale pin high
0156  12 04 99   call delay_small
0159  D2 A3      setb adc_start                  ;start pin high
015B  12 04 99   call delay_small
015E  C2 A4      clr adc_ale                     ;ale pin low
0160  12 04 99   call delay_small
0163  C2 A3      clr adc_start                   ;start pin low
0165  12 04 A6   call delay_long
0168  E5 90      mov a,P1
016A  24 42      add a,#66
016C  F5 51      mov keep_data,a
016E  12 04 99   call delay_small
0171  90 06 E7   mov dptr ,#temp_4
0174  12 09 89   call trans
0177  E5 51      mov a,keep_data
0179  12 04 4D   call sensor_5
017C  90 09 6D   mov dptr ,#temp
017F  12 09 89   call trans
0182  12 09 9A   call temp_check ;;;;check temperature
0185  C3         clr c
                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0186  11 3F      call read_sensor_5
0188  C2 A4      clr adc_ale
018A  C2 A3      clr adc_start

                                      ;
018C  12 04 99   call delay_small
018F  D2 A4      setb adc_ale                    ;ale pin high
```

```
 265:    0191  12 04 99          call delay_small
 266:    0194  D2 A3             setb adc_start               ;start pin high
 267:    0196  12 04 99          call delay_small

Line  I  Addr  Code          Source
 268:    0199  C2 A4             clr adc_ale                   ;ale pin low
 269:    019B  12 04 99          call delay_small
 270:    019E  C2 A3             clr adc_start                ;start pin low
 271:    01A0  12 04 A6          call delay_long
 272:    01A3  E5 90             mov a,P1
 273:    01A5  24 42             add a,#66
 274:    01A7  F5 51             mov keep_data,a
 275:    01A9  12 04 99          call delay_small
 276:    01AC  90 06 FB          mov dptr ,#temp_5
 277:    01AF  12 09 89          call trans
 278:    01B2  E5 51             mov a,keep_data
 279:    01B4  12 04 60          call sensor_6
 280:    01B7  90 09 6D          mov dptr ,#temp
 281:    01BA  12 09 89          call trans
 282:    01BD  12 09 9A          call temp_check ;;;;check temperature
 283:    01C0  C3                clr c
 284:                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 285:    01C1  11 46             call read_sensor_6
 286:    01C3  C2 A4             clr adc_ale
 287:    01C5  C2 A3             clr adc_start
 288:
 289:                                                   ;
 290:    01C7  12 04 99          call delay_small
 291:    01CA  D2 A4             setb adc_ale                  ;ale pin high
 292:    01CC  12 04 99          call delay_small
 293:    01CF  D2 A3             setb adc_start               ;start pin high
 294:    01D1  12 04 99          call delay_small
 295:    01D4  C2 A4             clr adc_ale                   ;ale pin low
 296:    01D6  12 04 99          call delay_small
 297:    01D9  C2 A3             clr adc_start                ;start pin low
 298:    01DB  12 04 A6          call delay_long
 299:    01DE  E5 90             mov a,P1
 300:    01E0  24 42             add a,#66
 301:    01E2  F5 51             mov keep_data,a
 302:    01E4  12 04 99          call delay_small
 303:    01E7  90 07 0F          mov dptr ,#temp_6
 304:    01EA  12 09 89          call trans
 305:    01ED  E5 51             mov a,keep_data
 306:    01EF  12 04 73          call sensor_7
 307:    01F2  90 09 6D          mov dptr ,#temp
 308:    01F5  12 09 89          call trans
```

```
01F8  12 09 9A          call temp_check ;;;;check temperature
01FB C3                 clr c
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01FC 11 4D              call read_sensor_7
01FE  C2 A4             clr adc_ale
0200  C2 A3             clr adc_start

                                                ;
0202  12 04 99          call delay_small
0205  D2 A4             setb adc_ale                    ;ale pin high
0207  12 04 99          call delay_small
020A D2 A3              setb adc_start                  ;start pin high
020C 12 04 99           call delay_small
020F  C2 A4             clr adc_ale                     ;ale pin low
0211  12 04 99          call delay_small

Line  I  Addr  Code        Source
0214  C2 A3             clr adc_start                   ;start pin low
0216  12 04 A6          call delay_long
0219  E5 90             mov a,P1
021B 24 42              add a,#66
021D F5 51              mov keep_data,a
021F  12 04 99          call delay_small
0222  90 07 23          mov dptr ,#temp_7
0225  12 09 89          call trans
0228  E5 51             mov a,keep_data
022A 12 04 86           call sensor_8
022D 90 09 6D           mov dptr ,#temp
0230  12 09 89          call trans
0233  12 09 9A          call temp_check ;;;;check temperature
0236  C3                clr c
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;; this for presure
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0237  D2 B2             setb latch_1
0239  C2 B3             clr  latch_2
023B D2 B4              setb latch_3
023D D2 B5              setb latch_4
                    ;mov p1,#0ffh
023F  11 1C             call read_sensor_0
0241  C2 A4             clr adc_ale
0243  C2 A3             clr adc_start

                                                ;
0245  12 04 99          call delay_small
0248  D2 A4             setb adc_ale                    ;ale pin high
024A 12 04 99           call delay_small
```

```
024D D2 A3        setb adc_start          ;start pin high
024F 12 04 99     call delay_small
0252 C2 A4        clr adc_ale             ;ale pin low
0254 12 04 99     call delay_small
0257 C2 A3        clr adc_start           ;start pin low
0259 12 04 A6     call delay_long
025C E5 90        mov a,P1
                  ;add a,#8
025E F5 51        mov keep_data,a
0260 12 04 99     call delay_small
0263 90 07 37     mov dptr ,#presu_0
0266 12 09 89     call trans
0269 E5 51        mov a,keep_data
026B 12 04 01     call sensor_1
026E 90 09 72     mov dptr ,#presu
0271 12 09 89     call trans
0274 12 09 CF     call pres_check
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0277 11 23        call read_sensor_1
0279 C2 A4        clr adc_ale
027B C2 A3        clr adc_start

                                ;
027D 12 04 99     call delay_small
0280 D2 A4        setb adc_ale            ;ale pin high
0282 12 04 99     call delay_small
0285 D2 A3        setb adc_start          ;start pin high
Line I Addr Code     Source
0287 12 04 99     call delay_small
028A C2 A4        clr adc_ale             ;ale pin low
028C 12 04 99     call delay_small
028F C2 A3        clr adc_start           ;start pin low
0291 12 04 A6     call delay_long
0294 E5 90        mov a,P1
                   ;add a,#8
0296 F5 51        mov keep_data,a
0298 12 04 99     call delay_small
029B 90 07 51     mov dptr ,#presu_1
029E 12 09 89     call trans
02A1 E5 51        mov a,keep_data
02A3 12 04 14     call sensor_2
02A6 90 09 72     mov dptr ,#presu
02A9 12 09 89     call trans
02AC 12 09 CF     call pres_check
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
02AF 11 2A        call read_sensor_2
```

```
398:    02B1 C2 A4          clr adc_ale
399:    02B3 C2 A3          clr adc_start
400:
401:                                                  ;
402:    02B5 12 04 99       call delay_small
403:    02B8 D2 A4          setb adc_ale                      ;ale pin high
404:    02BA 12 04 99       call delay_small
405:    02BD D2 A3          setb adc_start                    ;start pin high
406:    02BF 12 04 99       call delay_small
407:    02C2 C2 A4          clr adc_ale                       ;ale pin low
408:    02C4 12 04 99       call delay_small
409:    02C7 C2 A3          clr adc_start                     ;start pin low
410:    02C9 12 04 A6       call delay_long
411:    02CC E5 90          mov a,P1
412:                         ;add a,#8
413:    02CE F5 51          mov keep_data,a
414:    02D0 12 04 99       call delay_small
415:    02D3 90 07 69       mov dptr ,#presu_2
416:    02D6 12 09 89       call trans
417:    02D9 E5 51          mov a,keep_data
418:    02DB 12 04 27       call sensor_3
419:    02DE 90 09 72       mov dptr ,#presu
420:    02E1 12 09 89       call trans
421:    02E4 12 09 CF       call pres_check
422:                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
423:    02E7 11 31          call read_sensor_3
424:    02E9 C2 A4          clr adc_ale
425:    02EB C2 A3          clr adc_start
426:
427:                                                  ;
428:    02ED 12 04 99       call delay_small
429:    02F0 D2 A4          setb adc_ale                      ;ale pin high
430:    02F2 12 04 99       call delay_small
431:    02F5 D2 A3          setb adc_start                    ;start pin high
432:    02F7 12 04 99       call delay_small
433:    02FA C2 A4          clr adc_ale                       ;ale pin low
434:    02FC 12 04 99       call delay_small
435:    02FF C2 A3          clr adc_start                     ;start pin low
Line  I  Addr  Code         Source
436:    0301 12 04 A6       call delay_long
437:    0304 E5 90          mov a,P1
438:                        ; add a,#8
439:    0306 F5 51          mov keep_data,a
440:    0308 12 04 99       call delay_small
441:    030B 90 07 81       mov dptr ,#presu_3
442:    030E 12 09 89       call trans
```

```
0311  E5 51          mov a,keep_data
0313  12 04 3A       call sensor_4
0316  90 09 72       mov dptr ,#presu
0319  12 09 89       call trans
031C  12 09 CF       call pres_check
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
031F  11 38          call read_sensor_4
0321  C2 A4          clr adc_ale
0323  C2 A3          clr adc_start
                                                ;
0325  12 04 99       call delay_small
0328  D2 A4          setb adc_ale               ;ale pin high
032A  12 04 99       call delay_small
032D  D2 A3          setb adc_start             ;start pin high
032F  12 04 99       call delay_small
0332  C2 A4          clr adc_ale                ;ale pin low
0334  12 04 99       call delay_small
0337  C2 A3          clr adc_start              ;start pin low
0339  12 04 A6       call delay_long
033C  E5 90          mov a,P1
                      ;add a,#8
033E  F5 51          mov keep_data,a
0340  12 04 99       call delay_small
0343  90 07 99       mov dptr ,#presu_4
0346  12 09 89       call trans
0349  E5 51          mov a,keep_data
034B  12 04 4D       call sensor_5
034E  90 09 72       mov dptr ,#presu
0351  12 09 89       call trans
0354  12 09 CF       call pres_check
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0357  11 3F          call read_sensor_5
0359  C2 A4          clr adc_ale
035B  C2 A3          clr adc_start
                                                ;
035D  12 04 99       call delay_small
0360  D2 A4          setb adc_ale               ;ale pin high
0362  12 04 99       call delay_small
0365  D2 A3          setb adc_start             ;start pin high
0367  12 04 99       call delay_small
036A  C2 A4          clr adc_ale                ;ale pin low
036C  12 04 99       call delay_small
036F  C2 A3          clr adc_start              ;start pin low
0371  12 04 A6       call delay_long
```

```
0374  E5 90        mov a,P1
                    ;add a,#8
0376  F5 51        mov keep_data,a
Line  I  Addr  Code        Source
0378  12 04 99     call delay_small
037B  90 07 B1     mov dptr ,#presu_5
037E  12 09 89     call trans
0381  E5 51        mov a,keep_data
0383  12 04 60     call sensor_6
0386  90 09 72     mov dptr ,#presu
0389  12 09 89     call trans
038C  12 09 CF     call pres_check
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
038F  11 46        call read_sensor_6
0391  C2 A4        clr adc_ale
0393  C2 A3        clr adc_start
                                            ;
0395  12 04 99     call delay_small
0398  D2 A4        setb adc_ale                       ;ale pin high
039A  12 04 99     call delay_small
039D  D2 A3        setb adc_start                     ;start pin high
039F  12 04 99     call delay_small
03A2  C2 A4        clr adc_ale                        ;ale pin low
03A4  12 04 99     call delay_small
03A7  C2 A3        clr adc_start                      ;start pin low
03A9  12 04 A6     call delay_long
03AC  E5 90        mov a,P1
                    ;add a,#8
03AE  F5 51        mov keep_data,a
03B0  12 04 99     call delay_small
03B3  90 07 C9     mov dptr ,#presu_6
03B6  12 09 89     call trans
03B9  E5 51        mov a,keep_data
03BB  12 04 73     call sensor_7
03BE  90 09 72     mov dptr ,#presu
03C1  12 09 89     call trans
03C4  12 09 CF     call pres_check
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
03C7  11 4D        call read_sensor_7
03C9  C2 A4        clr adc_ale
03CB  C2 A3        clr adc_start
                                            ;
03CD  12 04 99     call delay_small
03D0  D2 A4        setb adc_ale                       ;ale pin high
```

```
 534:    03D2 12 04 99            call delay_small
 535:    03D5 D2 A3               setb adc_start              ;start pin high
 536:    03D7 12 04 99            call delay_small
 537:    03DA C2 A4               clr adc_ale                   ;ale pin low
 538:    03DC 12 04 99            call delay_small
 539:    03DF C2 A3               clr adc_start               ;start pin low
 540:    03E1 12 04 A6            call delay_long
 541:    03E4 E5 90               mov a,P1
 542:                              ;add a,#8
 543:    03E6 F5 51               mov keep_data,a
 544:    03E8 12 04 99            call delay_small
 545:    03EB 90 07 E1            mov dptr ,#presu_7
 546:    03EE 12 09 89            call trans
 547:    03F1 E5 51               mov a,keep_data
Line  I  Addr  Code          Source
 548:    03F3 12 04 86            call sensor_8
 549:    03F6 90 09 72            mov dptr ,#presu
 550:    03F9 12 09 89            call trans
 551:    03FC 12 09 CF            call pres_check
 552:                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;; this for flow rate
 553:                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 554:
 555:    03FF 01 54               jmp adc_process
 556:
 557:
 558:
 559:
 560:
 561:
 562:
 563:
 564:
 565:
 566:
 567:    0401 12 04 BB        sensor_1:call convert1
 568:    0404 E5 33                    mov a,ph1_hund
 569:    0406 12 09 92                 call send
 570:    0409 E5 34                    mov a,ph1_ten
 571:    040B 12 09 92                 call send
 572:    040E E5 35                    mov a,ph1_unit
 573:    0410 12 09 92                 call send
 574:                                  ;mov a,#0dh
 575:                                  ;call send
 576:    0413 22                       ret
 577:                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 578:    0414 12 04 D7        sensor_2:call convert2
```

```
0417  E5 36                 mov a,ph2_hund
0419  12 09 92              call send
041C  E5 37                 mov a,ph2_ten
041E  12 09 92              call send
0421  E5 38                 mov a,ph2_unit
0423  12 09 92              call send
                            ;mov a,#0dh
                            ;call send
0426  22                    ret
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0427  12 04 F3      sensor_3:call convert3
042A  E5 39                 mov a,ph3_hund
042C  12 09 92              call send
042F  E5 3A                 mov a,ph3_ten
0431  12 09 92              call send
0434  E5 3B                 mov a,ph3_unit
0436  12 09 92              call send
                            ;mov a,#0dh
                            ;call send
0439  22                    ret
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
043A  12 05 0F      sensor_4:call convert4
043D  E5 3C                 mov a,ph4_hund
043F  12 09 92              call send
0442  E5 3D                 mov a,ph4_ten
Line  I  Addr  Code         Source
0444  12 09 92              call send
0447  E5 3E                 mov a,ph4_unit
0449  12 09 92              call send
                            ;mov a,#0dh
                            ;call send
044C  22                    ret
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
044D  12 05 2B      sensor_5:call convert5
0450  E5 3F                 mov a,ph5_hund
0452  12 09 92              call send
0455  E5 40                 mov a,ph5_ten
0457  12 09 92              call send
045A  E5 41                 mov a,ph5_unit
045C  12 09 92              call send
                            ;mov a,#0dh
                            ;call send
045F  22                    ret
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0460  12 05 47      sensor_6:call convert6
0463  E5 42                 mov a,ph6_hund
```

```
0465  12 09 92            call send
0468  E5 43               mov a,ph6_ten
046A  12 09 92            call send
046D  E5 44               mov a,ph6_unit
046F  12 09 92            call send
                          ;mov a,#0dh
                          ;call send
0472  22                  ret
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0473  12 05 63      sensor_7:call convert7
0476  E5 45               mov a,ph7_hund
0478  12 09 92            call send
047B  E5 46               mov a,ph7_ten
047D  12 09 92            call send
0480  E5 47               mov a,ph7_unit
0482  12 09 92            call send
                          ;mov a,#0dh
                          ;call send
0485  22                  ret
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0486  12 05 7F      sensor_8:call convert8
0489  E5 48               mov a,ph8_hund
048B  12 09 92            call send
048E  E5 49               mov a,ph8_ten
0490  12 09 92            call send
0493  E5 50               mov a,ph8_unit
0495  12 09 92            call send
                          ;mov a,#0dh
                          ;call send
0498  22                  ret

0499                delay_small:
0499  78 46              mov r0,#70
049B                l1_delay_small:
049B  B2 A5              cpl adc_clk
049D  00                 nop
Line  I  Addr  Code       Source
049E  00                 nop
049F  00                 nop
04A0  00                 nop
04A1  00                 nop
04A2  00                 nop
04A3  D8 F6         djnz r0,l1_delay_small
04A5  22                 ret
```

```
04A6                delay_long:
04A6 78 8C              mov r0,#140
04A8                l1_delay_long:
04A8 B2 A5              cpl adc_clk
04AA 00                 nop
04AB 00                 nop
04AC 00                 nop
04AD 00                 nop
04AE 00                 nop
04AF D8 F7              djnz r0,l1_delay_long
04B1 22                 ret
04B2                delay:
04B2 7A 19             mov r2,#25
04B4 79 FF             mov r1,#255
04B6                delay1:
04B6 D9 FE             djnz r1,delay1
04B8 DA FC             djnz r2,delay1
04BA 22             ret
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
04BB                convert1:;mov a,phase_1
04BB 75 F0 64               mov b,#100
04BE 84                     div ab
04BF 12 05 9B               call decode
04C2 F5 33                  mov  ph1_hund,a
04C4 E5 F0                  mov a,b
04C6 75 F0 0A               mov b,#10
04C9 84                     div ab
04CA 12 05 9B               call decode
04CD F5 34                  mov ph1_ten,a
04CF E5 F0                  mov a,b
04D1 12 05 9B               call decode
04D4 F5 35                  mov ph1_unit,a
                            ;call display1
04D6 22                     ret
                    ;;;;;;;;;;;;;;;;;;;;;;;;;
04D7                convert2:;mov a,phase_1
04D7 75 F0 64               mov b,#100
04DA 84                     div ab
04DB 12 05 9B               call decode
04DE F5 36                  mov  ph2_hund,a
04E0 E5 F0                  mov a,b
04E2 75 F0 0A               mov b,#10
04E5 84                     div ab
04E6 12 05 9B               call decode
04E9 F5 37                  mov ph2_ten,a
```

```
 715:    04EB E5 F0                 mov a,b
Line  I  Addr  Code          Source
 716:    04ED 12 05 9B              call decode
 717:    04F0  F5 38                mov ph2_unit,a
 718:                               ;call display1
 719:    04F2  22                   ret
 720:                       ;;;;;;;;;;;;;;;;;;;;;;;;;;
 721:    04F3               convert3:;mov a,phase_1
 722:    04F3  75 F0 64             mov b,#100
 723:    04F6  84                   div ab
 724:    04F7  12 05 9B             call decode
 725:    04FA F5 39                 mov  ph3_hund,a
 726:    04FC E5 F0                 mov a,b
 727:    04FE  75 F0 0A             mov b,#10
 728:    0501  84                   div ab
 729:    0502  12 05 9B             call decode
 730:    0505  F5 3A                mov ph3_ten,a
 731:    0507  E5 F0                mov a,b
 732:    0509  12 05 9B             call decode
 733:    050C  F5 3B                mov ph3_unit,a
 734:                               ;call display1
 735:    050E  22                   ret
 736:                       ;;;;;;;;;;;;;;;;;;;;;;;;;;
 737:    050F               convert4:;mov a,phase_1
 738:    050F  75 F0 64             mov b,#100
 739:    0512  84                   div ab
 740:    0513  12 05 9B             call decode
 741:    0516  F5 3C                mov  ph4_hund,a
 742:    0518  E5 F0                mov a,b
 743:    051A 75 F0 0A              mov b,#10
 744:    051D 84                    div ab
 745:    051E  12 05 9B             call decode
 746:    0521  F5 3D                mov ph4_ten,a
 747:    0523  E5 F0                mov a,b
 748:    0525  12 05 9B             call decode
 749:    0528  F5 3E                mov ph4_unit,a
 750:                               ;call display1
 751:    052A 22                    ret
 752:                       ;;;;;;;;;;;;;;;;;;;;;;;;;;
 753:    052B               convert5:;mov a,phase_1
 754:    052B 75 F0 64              mov b,#100
 755:    052E 84                    div ab
 756:    052F  12 05 9B             call decode
 757:    0532  F5 3F                mov  ph5_hund,a
 758:    0534  E5 F0                mov a,b
 759:    0536  75 F0 0A             mov b,#10
```

```
0539  84            div ab
053A 12 05 9B       call decode
053D F5 40          mov ph5_ten,a
053F  E5 F0         mov a,b
0541  12 05 9B      call decode
0544  F5 41         mov ph5_unit,a
                    ;call display1
0546  22            ret
               ;;;;;;;;;;;;;;;;;;;;;;;;;;
0547           convert6:;mov a,phase_1
0547  75 F0 64      mov b,#100
054A 84             div ab
Line  I  Addr  Code        Source
054B 12 05 9B       call decode
054E  F5 42         mov  ph6_hund,a
0550  E5 F0         mov a,b
0552  75 F0 0A      mov b,#10
0555  84            div ab
0556  12 05 9B      call decode
0559  F5 43         mov ph6_ten,a
055B E5 F0          mov a,b
055D 12 05 9B       call decode
0560  F5 44         mov ph6_unit,a
                    ;call display1
0562  22            ret
               ;;;;;;;;;;;;;;;;;;;;;;;;;;
0563           convert7:;mov a,phase_1
0563  75 F0 64      mov b,#100
0566  84            div ab
0567  12 05 9B      call decode
056A F5 45          mov  ph7_hund,a
056C E5 F0          mov a,b
056E  75 F0 0A      mov b,#10
0571  84            div ab
0572  12 05 9B      call decode
0575  F5 46         mov ph7_ten,a
0577  E5 F0         mov a,b
0579  12 05 9B      call decode
057C F5 47          mov ph7_unit,a
                    ;call display1
057E  22            ret
               ;;;;;;;;;;;;;;;;;;;;;;;;;;
057F           convert8:;mov a,phase_1
057F  75 F0 64      mov b,#100
0582  84            div ab
0583  12 05 9B      call decode
```

```
0586  F5 48                 mov  ph8_hund,a
0588  E5 F0                 mov a,b
058A 75 F0 0A               mov b,#10
058D 84                     div ab
058E  12 05 9B              call decode
0591  F5 49                 mov ph8_ten,a
0593  E5 F0                 mov a,b
0595  12 05 9B              call decode
0598  F5 50                 mov ph8_unit,a
                            ;call display1
059A 22                     ret
                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
059B 44 30                  decode :orl a,#30h
059D 22                   ret
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
059E  90 06 7A         display1: mov dptr,#humid_display2
05A1 12 09 89               call trans
05A4 E5 33                  mov a,ph1_hund
05A6 12 09 92               call send
05A9 E5 34                  mov a,ph1_ten
05AB 12 09 92               call send
05AE E5 35                  mov a,ph1_unit

Line  I  Addr  Code           Source
05B0 12 09 92               call send
05B3 74 0D                  mov a,#0dh
05B5 12 09 92               call send
05B8 22                     ret
                       ;;;;;;;;;;;;;;;;;;;;;;
05B9 20 20 20 20       humid_display  : DB "     REMOTE INDUSTRIAL PROCESS "
,0dh
05BD 20 52 45 4D
05C1 4F 54 45 20
05C5 49 4E 44 55
05C9 53 54 52 49
05CD 41 4C 20 50
05D1 52 4F 43 45
05D5 53 53 20 0D
05D9 20 20 20 20                DB "       MONITORING AND CONTROL  " ,0dh
05DD 20 20 20 4D
05E1 4F 4E 49 54
05E5 4F 52 49 4E
05E9 47 20 41 4E
```

```
        05ED 44 20 43 4F
        05F1 4E 54 52 4F
        05F5 4C 20 20 0D
 837:   05F9 20 20 20 20              DB "      USING ANN AND RULE BASED " ,0dh
        05FD 20 20 55 53
        0601 49 4E 47 20
        0605 41 4E 4E 20
        0609 41 4E 44 20
        060D 52 55 4C 45
        0611 20 42 41 53
        0615 45 44 20 0D
 838:   0619 20 20 20 20              db "           EXPERT SYSTEM      " ,0DH
        061D 20 20 20 20
        0621 20 20 20 20
        0625 45 58 50 45
        0629 52 54 20 53
        062D 59 53 54 45
        0631 4D 20 20 20
        0635 20 20 20 0D
 839:   0639 20 20 20 20              DB "     -------------------------" ,0dh
        063D 20 2D 2D 2D
        0641 2D 2D 2D 2D
        0645 2D 2D 2D 2D
        0649 2D 2D 2D 2D
        064D 2D 2D 2D 2D
        0651 2D 2D 2D 2D
        0655 2D 2D 2D 0D
 840:   0659 20 20 20 20              DB "     -------------------------"
        065D 20 2D 2D 2D
        0661 2D 2D 2D 2D
        0665 2D 2D 2D 2D
        0669 2D 2D 2D 2D
        066D 2D 2D 2D 2D
        0671 2D 2D 2D 2D
        0675 2D 2D 2D
 841:   0678 0D 00                    db 0dh,0
Line  I  Addr  Code            Source
 842:   067A 52 65 6C 61    humid_display2  : DB "Relative Humidity Value : "
        067E 74 69 76 65
        0682 20 48 75 6D
        0686 69 64 69 74
        068A 79 20 56 61
        068E 6C 75 65 20
        0692 3A 20
 843:   0694 00                       db 0
 844:                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
0695  0D 0D          temp_0 : db 0dh,0dh
0697  50 6C 61 6E            db "Plant one temp   : ",0
069B  74 20 6F 6E
069F  65 20 74 65
06A3  6D 70 20 20
06A7  20 3A 20 00
06AB  50 6C 61 6E    temp_1 : db "Plant two temp   : ",0
06AF  74 20 74 77
06B3  6F 20 74 65
06B7  6D 70 20 20
06BB  20 3A 20 00
06BF  50 6C 61 6E    temp_2 : db "Plant three temp : ",0
06C3  74 20 74 68
06C7  72 65 65 20
06CB  74 65 6D 70
06CF  20 3A 20 00
06D3  50 6C 61 6E    temp_3 : db "Plant four temp  : ",0
06D7  74 20 66 6F
06DB  75 72 20 74
06DF  65 6D 70 20
06E3  20 3A 20 00
06E7  50 6C 61 6E    temp_4 : db "Plant five temp  : ",0
06EB  74 20 66 69
06EF  76 65 20 74
06F3  65 6D 70 20
06F7  20 3A 20 00
06FB  50 6C 61 6E    temp_5 : db "Plant six temp   : ",0
06FF  74 20 73 69
0703  78 20 74 65
0707  6D 70 20 20
070B  20 3A 20 00
070F  50 6C 61 6E    temp_6 : db "Plant seven temp : ",0
0713  74 20 73 65
0717  76 65 6E 20
071B  74 65 6D 70
071F  20 3A 20 00
0723  50 6C 61 6E    temp_7 : db "Plant eight temp : ",0
0727  74 20 65 69
072B  67 68 74 20
072F  74 65 6D 70
0733  20 3A 20 00
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

0737  0D 0D          presu_0: db 0dh,0dh
0739  50 6C 61 6E            db "Plant one pressure   : ",0
073D  74 20 6F 6E
```

```
          0741  65 20 70 72
Line  I  Addr  Code           Source
          0745  65 73 73 75
          0749  72 65 20 20
          074D 20 3A 20 00
 858:     0751  50 6C 61 6E    presu_1: db "Plant two pressure   : ",0
          0755  74 20 74 77
          0759  6F 20 70 72
          075D 65 73 73 75
          0761  72 65 20 20
          0765  20 3A 20 00
 859:     0769  50 6C 61 6E    presu_2: db "Plant three pressure : ",0
          076D 74 20 74 68
          0771  72 65 65 20
          0775  70 72 65 73
          0779  73 75 72 65
          077D 20 3A 20 00
 860:     0781  50 6C 61 6E    presu_3: db "Plant four pressure  : ",0
          0785  74 20 66 6F
          0789  75 72 20 70
          078D 72 65 73 73
          0791  75 72 65 20
          0795  20 3A 20 00
 861:     0799  50 6C 61 6E    presu_4: db "Plant five pressure  : ",0
          079D 74 20 66 69
          07A1 76 65 20 70
          07A5 72 65 73 73
          07A9 75 72 65 20
          07AD20 3A 20 00
 862:     07B1 50 6C 61 6E    presu_5: db "Plant six pressure   : ",0
          07B5 74 20 73 69
          07B9 78 20 70 72
          07BD65 73 73 75
          07C1 72 65 20 20
          07C5 20 3A 20 00
 863:     07C9 50 6C 61 6E    presu_6: db "Plant seven pressure : ",0
          07CD74 20 73 65
          07D1 76 65 6E 20
          07D5 70 72 65 73
          07D9 73 75 72 65
          07DD20 3A 20 00
 864:     07E1 50 6C 61 6E    presu_7: db "Plant eight pressure : ",0
          07E5 74 20 65 69
          07E9 67 68 74 20
          07ED 70 72 65 73
          07F1 73 75 72 65
```

```
         07F5  20 3A 20 00
 865:                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 866:
 867:    07F9  0D 0D          level_0: db 0dh,0dh
 868:    07FB 50 6C 61 6E            db "Plant one level   : ",0
         07FF  74 20 6F 6E
         0803  65 20 6C 65
         0807  76 65 6C 20
         080B 20 20 3A 20
         080F  00
 869:    0810  50 6C 61 6E    level_1: db "Plant two level   : ",0
         0814  74 20 74 77
Line  I  Addr  Code           Source
         0818  6F 20 6C 65
         081C 76 65 6C 20
         0820  20 20 3A 20
         0824  00
 870:    0825  50 6C 61 6E    level_2: db "Plant three level : ",0
         0829  74 20 74 68
         082D 72 65 65 20
         0831  6C 65 76 65
         0835  6C 20 3A 20
         0839  00
 871:    083A 50 6C 61 6E     level_3: db "Plant four level  : ",0
         083E  74 20 66 6F
         0842  75 72 20 6C
         0846  65 76 65 6C
         084A 20 20 3A 20
         084E  00
 872:    084F  50 6C 61 6E    level_4: db "Plant five level  : ",0
         0853  74 20 66 69
         0857  76 65 20 6C
         085B 65 76 65 6C
         085F  20 20 3A 20
         0863  00
 873:    0864  50 6C 61 6E    level_5: db "Plant six level   : ",0
         0868  74 20 73 69
         086C 78 20 6C 65
         0870  76 65 6C 20
         0874  20 20 3A 20
         0878  00
 874:    0879  50 6C 61 6E    level_6: db "Plant seven level : ",0
         087D 74 20 73 65
         0881  76 65 6E 20
         0885  6C 65 76 65
         0889  6C 20 3A 20
```

```
        088D 00
 875:   088E  50 6C 61 6E    level_7: db "Plant eight level : ",0
        0892  74 20 65 69
        0896  67 68 74 20
        089A  6C 65 76 65
        089E  6C 20 3A 20
        08A2  00
 876:                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 877:   08A3 0D 0D           flow_0 : db 0dh,0dh
 878:   08A5 50 6C 61 6E             db "Plant one flow rate   : ",0
        08A9 74 20 6F 6E
        08AD 65 20 66 6C
        08B1 6F 77 20 72
        08B5 61 74 65 20
        08B9 20 20 3A 20
        08BD 00
 879:   08BE 50 6C 61 6E     flow_1 : db "Plant two flow rate   : ",0
        08C2 74 20 74 77
        08C6 6F 20 66 6C
        08CA 6F 77 20 72
        08CE 61 74 65 20
        08D2 20 20 3A 20
        08D6 00
Line  I  Addr  Code          Source
 880:   08D7 50 6C 61 6E     flow_2 : db "Plant three flow rate : ",0
        08DB 74 20 74 68
        08DF 72 65 65 20
        08E3 66 6C 6F 77
        08E7 20 72 61 74
        08EB 65 20 3A 20
        08EF 00
 881:   08F0  50 6C 61 6E    flow_3 : db "Plant four flow rate  : ",0
        08F4  74 20 66 6F
        08F8  75 72 20 66
        08FC 6C 6F 77 20
        0900  72 61 74 65
        0904  20 20 3A 20
        0908  00
 882:   0909  50 6C 61 6E    flow_4 : db "Plant five flow rate  : ",0
        090D 74 20 66 69
        0911  76 65 20 66
        0915  6C 6F 77 20
        0919  72 61 74 65
        091D 20 20 3A 20
        0921  00
 883:   0922  50 6C 61 6E    flow_5 : db "Plant six flow rate   : ",0
```

```
         0926  74 20 73 69
         092A  78 20 66 6C
         092E  6F 77 20 72
         0932  61 74 65 20
         0936  20 20 3A 20
         093A  00
 884:    093B  50 6C 61 6E    flow_6 : db "Plant seven flow rate : ",0
         093F  74 20 73 65
         0943  76 65 6E 20
         0947  66 6C 6F 77
         094B  20 72 61 74
         094F  65 20 3A 20
         0953  00
 885:    0954  50 6C 61 6E    flow_7 : db "Plant eight flow rate : ",0
         0958  74 20 65 69
         095C  67 68 74 20
         0960  66 6C 6F 77
         0964  20 72 61 74
         0968  65 20 3A 20
         096C  00
 886:    096D  20 27 43 0D    temp : db " 'C" ,0dh,0
         0971  00
 887:    0972  20 70 61 73    presu : db " pascal",0dh,0
         0976  63 61 6C 0D
         097A  00
 888:    097B  20 43 6D 5E    flow : db " Cm^3/s" ,0dh,0
         097F  33 2F 73 0D
         0983  00
 889:    0984  20 43 6D 0D    level : db " Cm" ,0dh,0
         0988  00
 890:                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 891:    0989                 TRANS:
 892:    0989  E4              H_1: CLR A
 893:    098A  93                   movc a,@a+dptr;get the character
Line  I  Addr  Code           Source
 894:    098B  A3                   inc dptr
 895:    098C  12 09 92             call send
 896:    098F  70 F8                jnz h_1
 897:                               ;cjne a,#"@",h_1
 898:    0991  22                   RET
 899:
 900:
 901:
 902:    0992  F5 99          SEND: MOV SBUF,A ;load the data
 903:    0994  30 99 FD        H_2: JNB TI,H_2 ;stay here until last bit
 904:    0997  C2 99                CLR TI ;get ready for next char
```

```
0999  22                    RET ;return to caller

                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
099A C3               temp_check:    clr c
099B E5 51                      mov a, keep_data
099D B4 2D 06                   cjne a,#45,t_go
09A0 D2 B6                      setb buzzer
09A2 12 04 B2                   call delay
09A5 22                         ret
09A6                  t_go:
09A6 94 2D                      subb a,#45
09A8 40 08                      jc t_grt_tan0
09AA 50 00                      jnc t_less_tan0

09AC                  t_less_tan0:
09AC D2 B6                  setb buzzer
09AE 12 04 B2               call delay
09B1 22                     ret
09B2                  t_grt_tan0:
09B2 E5 51                      mov a,keep_data
09B4 B4 2A 06                        cjne a,#42,t_go2
09B7 D2 B6                           setb buzzer
09B9 12 04 B2                        call delay
09BC 22                              ret
09BD 94 2A            t_go2:         subb a,#42
09BF 50 02                      jnc t_grt_tan1
09C1 40 06                      jc t_less_tan1
09C3 C2 B6            t_grt_tan1:   clr buzzer
09C5 12 04 B2                        call delay
09C8 22                         ret
09C9 D2 B6            t_less_tan1:   setb buzzer
09CB 12 04 B2                        call delay
09CE 22                         ret
                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
09CF                  pres_check:
09CF C3                         clr c
09D0 E5 51                      mov a, keep_data
09D2 B4 6C 06                   cjne a,#108,p_go
09D5 D2 B6                      setb buzzer
09D7 12 04 B2                   call delay
09DA 22                         ret
09DB                  p_go:
09DB 94 6C                      subb a,#108
09DD 40 08                      jc p_grt_tan0
09DF 50 00                      jnc p_less_tan0
```

Line I Addr Code Source

```
09E1              p_less_tan0:
09E1  D2 B6               setb buzzer
09E3  12 04 B2            call delay
09E6  22                  ret
09E7              p_grt_tan0:
09E7  E5 51                   mov a,keep_data
09E9  B4 61 06                    cjne a,#97,p_go2
09EC  D2 B6                       setb buzzer
09EE  12 04 B2                    call delay
09F1  22                          ret
09F2  94 61       p_go2:          subb a,#97
09F4  50 02                   jnc p_grt_tan1
09F6  40 06                   jc p_less_tan1
09F8  C2 B6       p_grt_tan1:     clr buzzer
09FA  12 04 B2                    call delay
09FD  22                      ret
09FE  D2 B6       p_less_tan1:    setb buzzer
0A00  12 04 B2                    call delay
0A03  22                      ret

                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0A04              level_check:
0A04  C3                      clr c
0A05  E5 51                   mov a, keep_data
0A07  B4 FF 06                cjne a,#255,l_go
0A0A  D2 B6                   setb buzzer
0A0C  12 04 B2                call delay
0A0F  22                      ret
0A10              l_go:
0A10  94 FF                   subb a,#255
0A12  40 08                   jc l_grt_tan0
0A14  50 00                   jnc l_less_tan0

0A16              l_less_tan0:
0A16  D2 B6               setb buzzer
0A18  12 04 B2            call delay
0A1B  22                  ret
0A1C              l_grt_tan0:
0A1C  E5 51                   mov a,keep_data
0A1E  B4 F7 06                    cjne a,#247,l_go2
0A21  D2 B6                       setb buzzer
0A23  12 04 B2                    call delay
0A26  22                          ret
0A27  94 F7       l_go2:          subb a,#247
0A29  50 02                   jnc l_grt_tan1
```

```
0A2B 40 06                    jc l_less_tan1
0A2D C2 B6          l_grt_tan1:     clr buzzer
0A2F 12 04 B2                         call delay
0A32 22                       ret
0A33 D2 B6          l_less_tan1:    setb buzzer
0A35 12 04 B2                         call delay
0A38 22                       ret
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0A39                flow_check:

Line  I  Addr  Code          Source
0A39 C3                       clr c
0A3A E5 51                    mov a, keep_data
0A3C B4 73 06                 cjne a,#115,f_go
0A3F D2 B6                    setb buzzer
0A41 12 04 B2                 call delay
0A44 22                       ret
0A45                f_go:
0A45 94 55                    subb a,#85
0A47 40 08                    jc f_grt_tan0
0A49 50 00                    jnc f_less_tan0

0A4B                f_less_tan0:
0A4B D2 B6                  setb buzzer
0A4D 12 04 B2               call delay
0A50 22                     ret
0A51                f_grt_tan0:
0A51 E5 51                    mov a,keep_data
0A53 B4 5F 06                         cjne a,#95,f_go2
0A56 D2 B6                            setb buzzer
0A58 12 04 B2                         call delay
0A5B 22                               ret
0A5C 94 5F          f_go2:            subb a,#95
0A5E 50 02                    jnc f_grt_tan1
0A60 40 06                    jc f_less_tan1
0A62 C2 B6          f_grt_tan1:     clr buzzer
0A64 12 04 B2                         call delay
0A67 22                       ret
0A68 D2 B6          f_less_tan1:    setb buzzer
0A6A 12 04 B2                         call delay
0A6D 22                       ret

                        end
```

register banks used:  ---

no errors

L I S T   O F   S Y M B O L S
=============================

| SYMBOL | TYPE | VALUE | LINE |
|---|---|---|---|
| ??ASEM_51 | NUMBER | 8051 | |
| ??VERSION | NUMBER | 0130 | |
| AC | BIT | D6 | |
| ACC | DATA | E0 | |
| ADC_A | BIT | A0 | 1 |
| ADC_ALE | BIT | A4 | 5 |
| ADC_B | BIT | A1 | 2 |
| ADC_C | BIT | A2 | 3 |
| ADC_CLK | BIT | A5 | 6 |
| ADC_PROCESS | CODE | 0054 | 116 |
| ADC_START | BIT | A3 | 4 |
| B | DATA | F0 | |
| BUZZER | BIT | B6 | 11 |
| CONVERT1 | CODE | 04BB | 689 |
| CONVERT2 | CODE | 04D7 | 705 |
| CONVERT3 | CODE | 04F3 | 721 |
| CONVERT4 | CODE | 050F | 737 |
| CONVERT5 | CODE | 052B | 753 |
| CONVERT6 | CODE | 0547 | 769 |
| CONVERT7 | CODE | 0563 | 785 |
| CONVERT8 | CODE | 057F | 801 |
| CY | BIT | D7 | |
| DECODE | CODE | 059B | 817 |
| DELAY | CODE | 04B2 | 680 |
| DELAY1 | CODE | 04B6 | 683 |
| DELAY_LONG | CODE | 04A6 | 669 |
| DELAY_SMALL | CODE | 0499 | 655 |
| DISPLAY1 | CODE | 059E | 821 |
| DPH | DATA | 83 | |
| DPL | DATA | 82 | |
| EA | BIT | AF | |

| | | | |
|---|---|---|---|
| ES | BIT | AC | |
| ET0 | BIT | A9 | |
| ET1 | BIT | AB | |
| EX0 | BIT | A8 | |
| EX1 | BIT | AA | |
| EXTI0 | CODE | 0003 | |
| EXTI1 | CODE | 0013 | |
| F0 | BIT | D5 | |
| FLOW | CODE | 097B | 888 |
| FLOW_0 | CODE | 08A3 | 877 |
| FLOW_1 | CODE | 08BE | 879 |
| FLOW_2 | CODE | 08D7 | 880 |
| FLOW_3 | CODE | 08F0 | 881 |
| FLOW_4 | CODE | 0909 | 882 |
| FLOW_5 | CODE | 0922 | 883 |
| FLOW_6 | CODE | 093B | 884 |
| FLOW_7 | CODE | 0954 | 885 |
| FLOW_CHECK | CODE | 0A39 | 1005 |
| F_GO | CODE | 0A45 | 1012 |
| F_GO2 | CODE | 0A5C | 1027 |
| F_GRT_TAN0 | CODE | 0A51 | 1021 |
| F_GRT_TAN1 | CODE | 0A62 | 1030 |
| F_LESS_TAN0 | CODE | 0A4B | 1017 |
| F_LESS_TAN1 | CODE | 0A68 | 1033 |
| HUMID_DISPLAY | CODE | 05B9 | 835 |
| HUMID_DISPLAY2 | CODE | 067A | 842 |
| H_1 | CODE | 0989 | 892 |
| H_2 | CODE | 0994 | 903 |
| IE | DATA | A8 | |
| IE0 | BIT | 89 | |
| IE1 | BIT | 8B | |
| INT0 | BIT | B2 | |
| INT1 | BIT | B3 | |
| IP | DATA | B8 | |
| IT0 | BIT | 88 | |
| IT1 | BIT | 8A | |
| KEEP_DATA | NUMBER | 0051 | 39 |
| L1_DELAY_LONG | CODE | 04A8 | 671 |
| L1_DELAY_SMALL | CODE | 049B | 657 |
| LATCH_1 | BIT | B2 | 7 |
| LATCH_2 | BIT | B3 | 8 |
| LATCH_3 | BIT | B4 | 9 |
| LATCH_4 | BIT | B5 | 10 |
| LEVEL | CODE | 0984 | 889 |
| LEVEL_0 | CODE | 07F9 | 867 |
| LEVEL_1 | CODE | 0810 | 869 |

| | | | |
|---|---|---|---|
| LEVEL_2 | CODE | 0825 | 870 |
| LEVEL_3 | CODE | 083A | 871 |
| LEVEL_4 | CODE | 084F | 872 |
| LEVEL_5 | CODE | 0864 | 873 |
| LEVEL_6 | CODE | 0879 | 874 |
| LEVEL_7 | CODE | 088E | 875 |
| LEVEL_CHECK | CODE | 0A04 | 972 |
| L_GO | CODE | 0A10 | 979 |
| L_GO2 | CODE | 0A27 | 994 |
| L_GRT_TAN0 | CODE | 0A1C | 988 |
| L_GRT_TAN1 | CODE | 0A2D | 997 |
| L_LESS_TAN0 | CODE | 0A16 | 984 |
| L_LESS_TAN1 | CODE | 0A33 | 1000 |
| OV | BIT | D2 | |
| P | BIT | D0 | |
| P0 | DATA | 80 | |
| P1 | DATA | 90 | |
| P2 | DATA | A0 | |
| P3 | DATA | B0 | |
| PCON | DATA | 87 | |
| PH1_HUND | NUMBER | 0033 | 15 |
| PH1_TEN | NUMBER | 0034 | 16 |
| PH1_UNIT | NUMBER | 0035 | 17 |
| PH2_HUND | NUMBER | 0036 | 18 |
| PH2_TEN | NUMBER | 0037 | 19 |
| PH2_UNIT | NUMBER | 0038 | 20 |
| PH3_HUND | NUMBER | 0039 | 21 |
| PH3_TEN | NUMBER | 003A | 22 |
| PH3_UNIT | NUMBER | 003B | 23 |
| PH4_HUND | NUMBER | 003C | 24 |
| PH4_TEN | NUMBER | 003D | 25 |
| PH4_UNIT | NUMBER | 003E | 26 |
| PH5_HUND | NUMBER | 003F | 27 |
| PH5_TEN | NUMBER | 0040 | 28 |
| PH5_UNIT | NUMBER | 0041 | 29 |
| PH6_HUND | NUMBER | 0042 | 30 |
| PH6_TEN | NUMBER | 0043 | 31 |
| PH6_UNIT | NUMBER | 0044 | 32 |
| PH7_HUND | NUMBER | 0045 | 33 |
| PH7_TEN | NUMBER | 0046 | 34 |
| PH7_UNIT | NUMBER | 0047 | 35 |
| PH8_HUND | NUMBER | 0048 | 36 |
| PH8_TEN | NUMBER | 0049 | 37 |
| PH8_UNIT | NUMBER | 0050 | 38 |
| PHASE_1 | NUMBER | 0030 | 12 |
| PHASE_2 | NUMBER | 0031 | 13 |

| | | | |
|---|---|---|---|
| PHASE_3 | NUMBER | 0032 | 14 |
| PRESU | CODE | 0972 | 887 |
| PRESU_0 | CODE | 0737 | 856 |
| PRESU_1 | CODE | 0751 | 858 |
| PRESU_2 | CODE | 0769 | 859 |
| PRESU_3 | CODE | 0781 | 860 |
| PRESU_4 | CODE | 0799 | 861 |
| PRESU_5 | CODE | 07B1 | 862 |
| PRESU_6 | CODE | 07C9 | 863 |
| PRESU_7 | CODE | 07E1 | 864 |
| PRES_CHECK | CODE | 09CF | 939 |
| PS | BIT | BC | |
| PSW | DATA | D0 | |
| PT0 | BIT | B9 | |
| PT1 | BIT | BB | |
| PX0 | BIT | B8 | |
| PX1 | BIT | BA | |
| P_GO | CODE | 09DB | 946 |
| P_GO2 | CODE | 09F2 | 961 |
| P_GRT_TAN0 | CODE | 09E7 | 955 |
| P_GRT_TAN1 | CODE | 09F8 | 964 |
| P_LESS_TAN0 | CODE | 09E1 | 951 |
| P_LESS_TAN1 | CODE | 09FE | 967 |
| RB8 | BIT | 9A | |
| RD | BIT | B7 | |
| READ_SENSOR_0 | CODE | 001C | 66 |
| READ_SENSOR_1 | CODE | 0023 | 73 |
| READ_SENSOR_2 | CODE | 002A | 79 |
| READ_SENSOR_3 | CODE | 0031 | 85 |
| READ_SENSOR_4 | CODE | 0038 | 91 |
| READ_SENSOR_5 | CODE | 003F | 97 |
| READ_SENSOR_6 | CODE | 0046 | 103 |
| READ_SENSOR_7 | CODE | 004D | 109 |
| REN | BIT | 9C | |
| RESET | CODE | 0000 | |
| RI | BIT | 98 | |
| RS0 | BIT | D3 | |
| RS1 | BIT | D4 | |
| RXD | BIT | B0 | |
| SBUF | DATA | 99 | |
| SCON | DATA | 98 | |
| SEND | CODE | 0992 | 902 |
| SENSOR_1 | CODE | 0401 | 567 |
| SENSOR_2 | CODE | 0414 | 578 |
| SENSOR_3 | CODE | 0427 | 589 |
| SENSOR_4 | CODE | 043A | 600 |

| | | | |
|---|---|---|---|
| SENSOR_5 | CODE | 044D | 611 |
| SENSOR_6 | CODE | 0460 | 622 |
| SENSOR_7 | CODE | 0473 | 633 |
| SENSOR_8 | CODE | 0486 | 644 |
| SINT | CODE | 0023 | |
| SM0 | BIT | 9F | |
| SM1 | BIT | 9E | |
| SM2 | BIT | 9D | |
| SP | DATA | 81 | |
| START | CODE | 0017 | 54 |
| T0 | BIT | B4 | |
| T1 | BIT | B5 | |
| TB8 | BIT | 9B | |
| TCON | DATA | 88 | |
| TEMP | CODE | 096D | 886 |
| TEMP_0 | CODE | 0695 | 845 |
| TEMP_1 | CODE | 06AB | 847 |
| TEMP_2 | CODE | 06BF | 848 |
| TEMP_3 | CODE | 06D3 | 849 |
| TEMP_4 | CODE | 06E7 | 850 |
| TEMP_5 | CODE | 06FB | 851 |
| TEMP_6 | CODE | 070F | 852 |
| TEMP_7 | CODE | 0723 | 853 |
| TEMP_CHECK | CODE | 099A | 908 |
| TF0 | BIT | 8D | |
| TF1 | BIT | 8F | |
| TH0 | DATA | 8C | |
| TH1 | DATA | 8D | |
| TI | BIT | 99 | |
| TIMER0 | CODE | 000B | |
| TIMER1 | CODE | 001B | |
| TL0 | DATA | 8A | |
| TL1 | DATA | 8B | |
| TMOD | DATA | 89 | |
| TR0 | BIT | 8C | |
| TR1 | BIT | 8E | |
| TRANS | CODE | 0989 | 891 |
| TXD | BIT | B1 | |
| T_GO | CODE | 09A6 | 914 |
| T_GO2 | CODE | 09BD | 929 |
| T_GRT_TAN0 | CODE | 09B2 | 923 |
| T_GRT_TAN1 | CODE | 09C3 | 932 |
| T_LESS_TAN0 | CODE | 09AC | 919 |
| T_LESS_TAN1 | CODE | 09C9 | 935 |
| WR | BIT | B6 | |

## Appendix C

// Program to Interface GSM Module with 8051 microcontroller (AT89C51) without using PC

```
#include<reg51.h>
#define port P1
#define dataport P2                         // Data port for LCD
sbit rs = port^2;
sbit rw = port^3;
sbit en = port^4;
int count,i;
unsigned char check,str[15];
bit check_space;

void init_serial()                      // Initialize serial port
{
        TMOD=0x20;                      // Mode2
        TH1=0xfd;                       // 9600 baud
        SCON=0x50;                      // Serial mode=1 ,8-Bit data,1 Stop bit ,1 Start bit, Receiving
on
        TR1=1;                              // Start timer
}
void delay(unsigned int msec)               // Function for delay
{
        int i,j;
        for(i=0;i<msec;i++)
           for(j=0; j<1275; j++);
}

void lcd_cmd(unsigned char item)            // Function to send command on LCD
{
        dataport = item;
        rs= 0;
        rw=0;
        en=1;
        delay(1);
        en=0;
        return;
}

void lcd_data(unsigned char item)           // Function to display character on LCD
{
        dataport = item;
        rs= 1;
        rw=0;
        en=1;
        delay(1);
```

```
        en=0;
        return;
}

void lcd_data_string(unsigned char *str)      // Function to display string on LCD
{
        int i=0;
        while(str[i]!='\0')
        {
          lcd_data(str[i]);
          i++;
          delay(10);
        }
        return;
}
void lcd()
{
        lcd_cmd(0x38);                                      // For using 8-bit 2 row LCD
        delay(5);
        lcd_cmd(0x0F);                              // For display on cursor blinking
        delay(5);
        lcd_cmd(0x80);                                      // Set the cursor on first
position of LCD
        delay(5);
}

void transmit_data(unsigned char str)// Function to transmit data through serial port
{
        SBUF=str;                           //Store data in SBUF
        while(TI==0);                       //Wait till data transmits
        TI=0;
}

 void receive_data()  interrupt 4       // Function to recieve data serialy from RS232 into
microcontroller
 {
        RI=0;
        str[++count]=SBUF;                  //Read SBUF

 }

unsigned char byte_check()              // Function to check carraige return and new line character
{
        switch(str[0])
        {
                case 0x0a:
```

```
		{				// Return 0x00 for new line
		return 0x00;
		break ;
		}
		case 0x0d:
		{				// Return 0x01 for carriage return
		return 0x01;
		break ;
		}
		default :return 0x02 ;		// Return 0x02 for characters except new line and
carriage return
	}
}

void main()
{
	lcd();							// Initialize LCD
	init_serial();						// Initialize serial port
	count=(-1);
	delay(500);
	lcd_data_string("Ready");
	delay(10);
	lcd_cmd(0x01);
	IE=0x94;
	transmit_data('A');					// Transmit 'A' to serial port
	delay(1);
	transmit_data('T');					// Transmit 'T' to serial port
	delay(1);
	transmit_data(0x0d);					// Transmit carriage return to serial
port
	delay(50);
	while(1)
	{
		if(count>=0)
		{
			check=byte_check();			// Check the character
			if(check!=0x00)
			{
				if(check==0x01)
				{
					if(check_space==1)	// Check previous character
					{
						lcd_data(0x20);
						check_space=0;
					}
				}
```

```
				else
				{
					lcd_data(str[0]);
					check_space=1;
				}
			}
			count--;
			for(i=0;i<count;i++)			// Shift the whole array to one left
			{
				str[i]=str[i+1];
			}
		}
	}
}
```

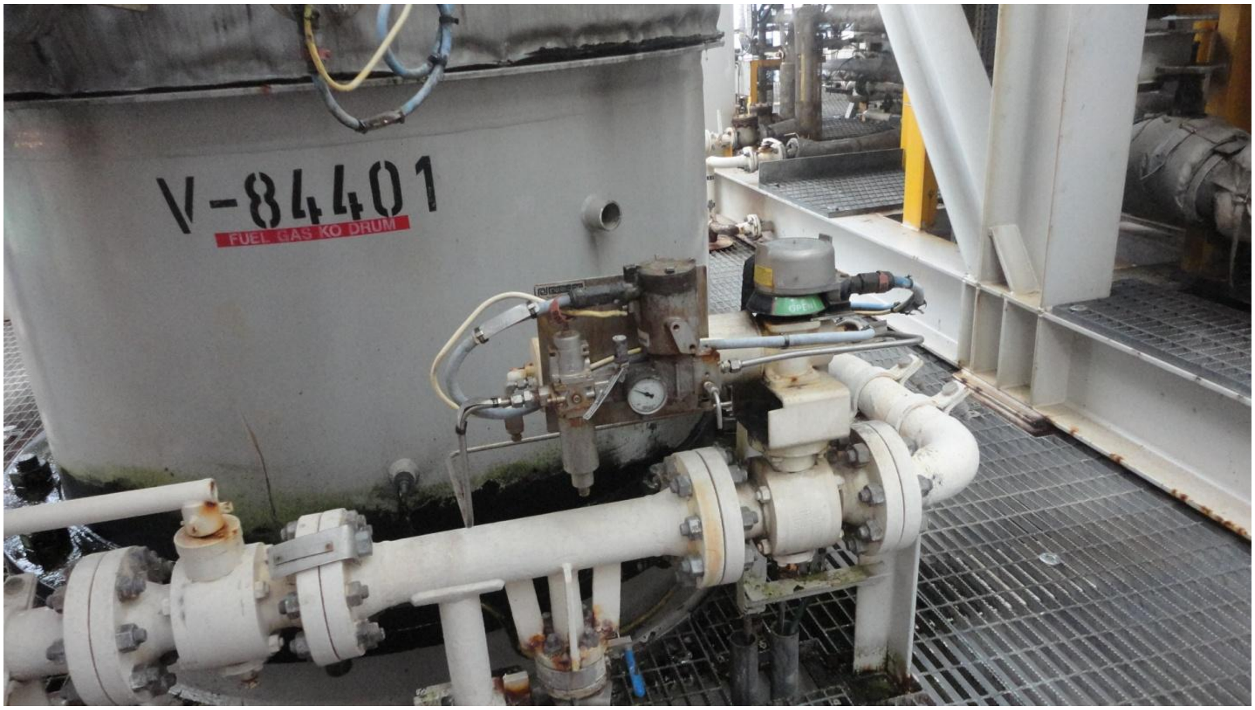

SPDC Equipment/ Control Room 3

SPDC Equipment/ Control Room 4


SPDC Equipment/ Control Room 5

SPDC Equipment/ Control Room 6

SPDC Equipment/ Control Room 7



SPDC Equipment/ Control Room 8


SPDC Equipment/ Control Room 9