IOT BASED SMART HOME DESIGN USING POWER AND SECURITY MANAGEMENT

A PROJECT WORK REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SUMITTED BY

G.VINAY KUMAR	**Reg.No.13KQ1A0549**
K.NAGALAKSHMI	**Reg.No.13KQ1A0518**
K.SAI KARUNAKAR	**Reg.No.13KQ1A0552**
B.SIREESHA	**Reg.No.13KQ1A0507**

Under the Esteemed Guidance of

Mrs. E.V.N.JYOTHI, M.Tech.,(Ph.D)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PACE INSTITUTE OF TECHNOLOGY & SCIENCES

(Approved by A.I.C.T.E., New Delhi, Affiliated to JNTU Kakinada)

ACCREDITED BY NAAC WITH'A' GRADE,(An ISO 9001:2008 Certified institution)

NH4, Near Vallurama Temple-ONGOLE 523272,Contact No:08592-201167,www.pace.ac.in

PACE INSTITUTE OF TECHNOLOGY & SCIENCES

(Approved by A.I.C.T.E., New Delhi, Affiliated to JNTU Kakinada)

ACCREDITED BY NAAC WITH'A' GRADE,(An ISO 9001:2008 Certified institution)

NH4, Near Vallurama Temple-ONGOLE 523272,Contact No:08592-201167,www.pace.ac.in

CERTIFICATE

This is to certify that the project entitled **"IOT BASED SMART HOME DESIGN USING POWER AND SECURITY MANAGEMENT"** is a bona fide work done by

G.VINAY KUMAR	**Reg.No.13KQ1A0549**
K.NAGALAKSHMI	**Reg.No.13KQ1A0518**
K.SAI KARUNAKAR	**Reg.No.13KQ1A0552**
B.SIREESHA	**Reg.No.13KQ1A0507**

Submitted in the partial fulfillment of the requirements for the Degree of Bachelor of Technology in Computer Science and Engineering during the academic years 2013-2017.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

Internal Guide	**Head of Department**
Mrs. E.V.N.Jyothi, M.Tech.,(Ph.D)	**Dr. R.Veeranjaneyulu M.Tech .,Ph.D**
Associate Professor	**Associate Professor & HOD**

External Examiner

PACE INSTITUTE OF TECHNOLOGY & SCIENCES

(Approved by A.I.C.T.E., New Delhi, Affiliated to JNTU Kakinada)

ACCREDITED BY NAAC WITH'A' GRADE,(An ISO 9001:2008 Certified institution)

NH4, Near Vallurama Temple-ONGOLE 523272,Contact No:08592-201167,www.pace.ac.in

2013-2017

DECLARATION

We hereby declare that the project report titled **"IOT BASED SMART HOME DESIGN USING POWER AND SECURITY MANAGEMENT"** under the guidance of **Mrs. E.V.N.Jyothi, M.Tech.,(Ph.D)Computer Science & Engineering,** is submitted in partial fulfillment of the requirements for the award of the Degree of Master of technology in Computer Science and Engineering

This is a record of bona fide work carried out by me and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

ACKNOWLEDGEMENT

I take this opportunity to express my deep respect and profound gratitude to our esteemed guide **Mrs. E.V.N.Jyothi, M.Tech.,(Ph.D) ,Computer Science & Engineering , PACE INSTITUTE OF TECHNOLOGY & SCIENCES Vallur,** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project.

I am ineffably indebted to our honorable Secretary & Correspondent **Er.M.Sridhar B.E** and our beloved Principal **Dr C.V.Subba Rao B.tech, M.E, Ph.D, MISTE** for their kind support and help.

I would like to extend my sincere and heartfelt thanks to **Mr. P.ANIL KUMAR M.Tech (Ph.D)** Head of Computer Science & Engineering Department, who has given valuable advice and suggestions and has encouraged us by providing all the facilities necessary to complete this project successfully.

I am thankful to all the **Professors and Faculty Members** in the department for their teachings and academic support and thanks to **Technical Staff and Non-teaching staff** in the department for their support.

I also acknowledge with a deep sense of reverence, my gratitude towards my parents, members of my family and friends for their constant support and encouragement.

G.VINAY KUMAR	**Reg.No.13KQ1A0549**
K.NAGALAKSHMI	**Reg.No.13KQ1A0518**
K.SAI KARUNAKAR	**Reg.No.13KQ1A0552**
B.SIREESHA	**Reg.No.13KQ1A0507**

ABSTRACT

The paper presents the design and implementation of an Ethernet-based Smart Home intelligent system for monitoring the electrical energy consumption based upon the real time tracking of the devices at home an “PIC MICRO CONTROLLER” board, which can be used in homes and societies. The proposed system works on real time monitoring and voice control, so that the electrical devices and switches can be remotely controlled and monitored with or without an android based app. It uses various sensors to not only monitor the real time device tracking but also maintaining the security of your house. It is monitored and controlled remotely from an android app using theInternet or the Intranet connectivity. The proposed outcome of the project aims as multiple benefits of saving on electricity bills of the home as well as keep the users updated about their home security with an option of controlling the switching of the devices by using their voice or simple toggle touch on their smart phone, and last but most importantly, monitor the usage in order to conserve the precious natural resources by reducing electrical energy consumption.

Keywords-- Internet of thing (IoT), Power consumption, Smart devices, Home automation, Security, Android etc.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

IOT	INTERNET OF THINGS
AWS	AMAZONE WEB SERVICES
PIC	PERIPHERAL INTERFACE CONTROLLER
LCD	LIQUID CRYSTAL DISPLAY
IC	INTERGRATED CIRCUIT
VREF	VOLTAGE REFERENCE
PSP	PARALLEL SLAVE PORT
RTOS	REAL TIME OPERATING SYSTEM

INTRODUCTION

The Internet of Things is the network of "things" which are connected to a common network path in order to communicate, exchange data or control each other. The network path can be interconnected or interconnected with the "things" being either embedded software, hardware or any sensor. It refers to the state where the things will have more and more data and information associated with them and has a ability to communicate, produce new information and become the integral part of the free world wide web. It not only features internet connectivity but also features cloud and data management, security management and all other fields concerned with the era of internet.

Nowadays, there is a growing demand of automation and intelligent systems so that it leaves us with less human intervention and smart decision making devices. With the growing demand, comes the growing competition which has forced the competitors to come out with more intelligent, efficient as well as user friendly models. This has made our lives easier from making our intelligent travel arrangements to our personal medical care. With a tap of your finger you can control your lights, with a single tap you can book your flight tickets, monitor traffic and weather and so on. It will refine our work flows, prioritizing tasks and projects based on on going assessments in real time of what is happening throughout our organization. The Internet of Things will maintain our appliances and vehicles, determining when they are next due for service, cleaning, or – in the case of our refrigerators –restocking (and making appropriate arrangements, such as repair appointments and grocery orders). It will enable our cars to communicate with other cars on the road as they self-drive us toand fro. It will regulate our lights, heat, AC, and other home appliances and devices, turning them on and off as we enter andexit rooms and as they "learn" our schedule. And that's not all.

Save money on energy use, while keeping your office or building comfortable. The cost of simply forgetting to turn offyour classroom lights and electric appliances can really add up over time. Controlling temperature and lighting based on time of day or occupancy can really reduce energy costs.

This paper proposes an IoT based smart and intelligent energy and security management system to autonomous power control system in a user friendly and a mobile way so that a user can manage the power management as well as security of their house even when not at the house itself, minimizing the power consumption and maximize utilization of resource by smart real2016 1st International Conference on Innovation and Challenges in Cyber Security (ICICCS 2016)time tracking and monitoring of the electrical devices and security of the house.

1.1 Issues around IoT

Internet of Things immediately triggers questions around the privacy of personal data. Whether real-time information about our physical location or updates about our weight and blood pressure that may be accessible by our health care providers, having new kinds and more detailed data about ourselves streaming over wireless networks and potentially around the world is an obvious concern.

Supplying power to this new proliferation of IoT devices and their network connections can be expensive and logistically difficult. Portable devices require batteries that someday must be replaced. Although many mobile devices are optimized for lower power usage, energy costs to keep potentially billions of them running remains high.

Numerous corporations and start-up ventures have latched onto the Internet of Things concept looking to take advantage of whatever business opportunities are available. While competition in the market helps lower prices of consumer products, in the worst case it also leads to confusing and inflated claims about what the products do.

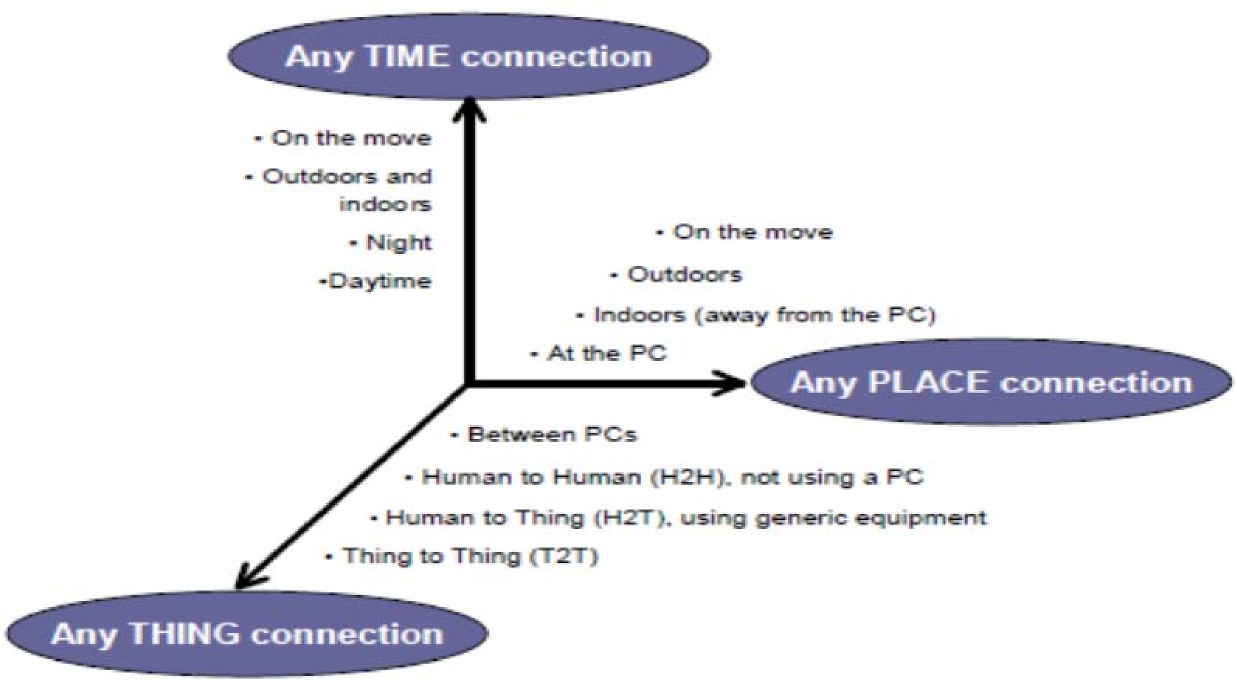


Fig 1.1: How IoT things are connecting

1.2 IoT Devices

If you keep track of the Sillicon Vally news then you are sure to know that IoT is the latest buzz. So many of the startups now are focussing towards IoT. Therefore when we are talking about IoT, we are basically talking about infinite possibilities, hardware and software platform. So when you try to find a list of IoT devices in internet, you might end up getting frustrated purely because it's is difficult to provide such list.

Therefore research of this particular subsection has taken significant time. Finally I have put togather somemost common and popular technologies in IoT to give you an overview of what devices are we really talking about.

We divide the IoT devices into two braod categories: The wearable ones and Microcontroller/Microprocessor driven embedded IoT devices. Some of the Embedded devices like Arduino Lillypad are minisque and you can further utilize them to make your own wearable solution. But in wearable I have included hardware which are pretty standard and IoT has only software scope for the developer.

I have also put some list oof common peripheral hardware that you might have to learn while working with IoT hardware in embedded level.

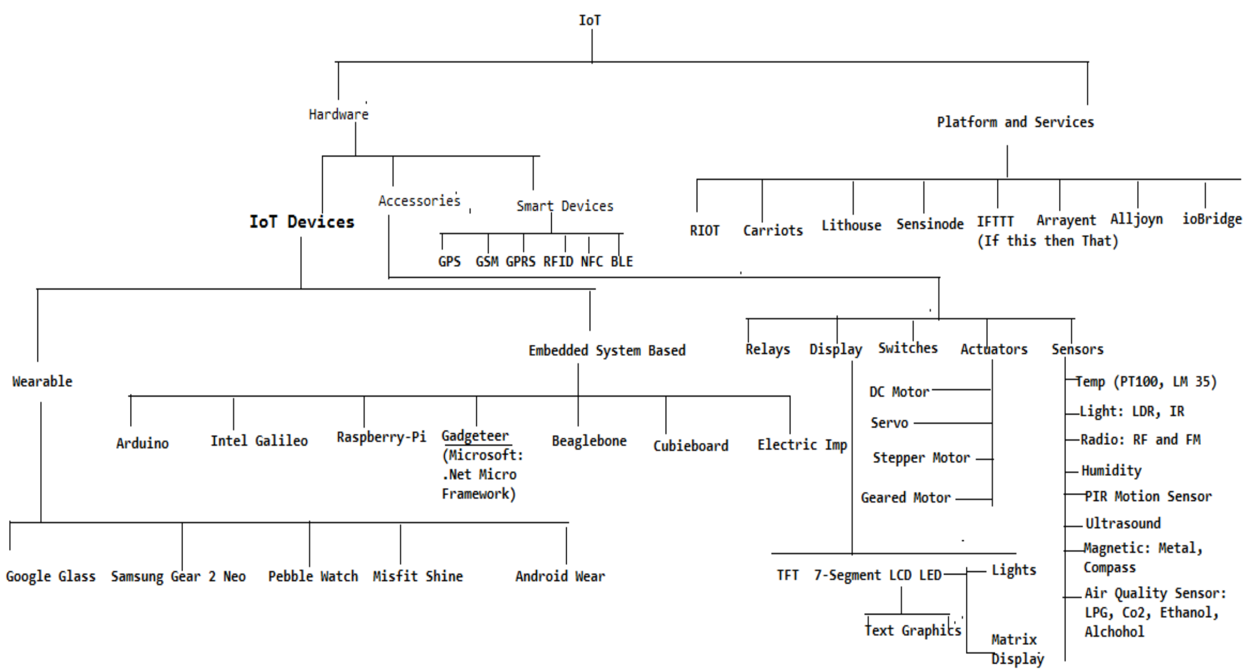


Fig 1.2: Common IoT devices and Technologies

1.3 IoT Platforms

At this stage we divide our IoT development into two parallel technologies: Wearable and Embedded. Developers can build apps for custom wearable devices like Peeble, Samsung Gear or can opt to create their own platform using Embedded solution and then can develop app for that platform.

Wearable Platform

Tizen is fast becoming one of the most popular platform for Mobile and wearable devices. **Tizen SDK** comes ported with wearable emulator which makes it easier to develop wearable solutions for Tizen platform.

As figure 1.1 suggests, a large Android Wear devices are now being made and marketed. Smart watches are getting popular by every day. Android Wear apps can be developed and tested in Eclipse. **This Android Developer Guide helps** you in setting up Android Wear development environment in Eclipse.

Salesforce is another platform which is coming up with awesome development environment, APIs in wearable technologies. Their solution is extended from Peeble to Google glass. Salesforce is really worth a try if you are planning to have a serious go at wearable technology as a career option. Checkout **Salesforce Wear page.**

Cloud Platform for IoT

Let's rediscuss the possibilities to beverage vending maching once more in terms of sheer possibilities. In the conventional vending machine you need to press a button or put a coin to trigger the process of liquid flow, which stops after certain quantity. Now how about integrating paypal or Google money with the vending machine? How about a customer discovering the vending machine as "website" along with it's location and then pays online for a glass ofbeverage. Once payment is successfull he gets an access token. He can pass the token to the machine through NFC and bingo he gets his drink.

Now this logical possibility is very important for understanding IoT and IoT really can bring several services (like online payment gateway), several hardware platform (like embedded board of the vending machine) and smart objects and data like NFC, GPS into a seamless environment.

Now if you can integrate online payment into beverage vending machine, why not in for a community washing machine? If you are using location service for beverage machine, then whey not utilize the location and payment service for the toll gate? Why not get the data of a medical diagnesis like ECG (acquired through another embedded board partaining to medical electronics) into cloud such that several doctors can view it and form a comprehensive opinion about the patient's state?

Well, infact all of them are possible. A little understanding of web and software design would take your mind towards cloud. Just like Web of Machines, in a Machine to Macine (M2M) or Machine to Objects (M2O) or any similar communication several modules will be common and several modules demands data to be available for sharing. Cloud APIs comes in handy in this regrad.

For instance when you have to make a device discoverable in web, you have to assign a fixed IP address, maintain a router and follow several networking skills. You might not have the knowledge and infrastructure needed for maintaining a commercial sophisticated network for IoT.

Yaler is a great example of what services and cloud can bring to table. This provides connection as a service such that your device is easily discoverable and communicable over the web without much hassle and takes care of underneath security.

Axeda Provides infrastructure for M2M architecture.

OpenIoT is an open source IoT platform that provides out of other services a unique Sensing as a Service.

Google has already integrated location services with it's cloud. Location extracted from your devices is silently put in your status updates in facebook and twitter and are also used for more personalized searches.

So cloud APIs has a great potential in IoT in all levels of architecture starting from firmware to hardware to more top level architecture.

PROBLEM DOMAIN

In this below paper we have identified a problem that is enhances in this project for the future use.

EXISTED MODEL

In this model we have proposed an Ethernet based system that let users monitor real time switching information of the electrical devices and controlling them through an android appas well as monitoring the security of their homes in case of unwanted entry or fire. Our model uses temperature sensor and smoke sensors to check for fire at the users home, PIR motion sensors to check for the unwanted presence at their homes and also monitor and control the real time tracking and switching ofall their electrical devices through an android based mobile app. The system is connected to this android app using internet connectivity for better and fast communication. The model has an option of controlling devices by either sending voice commands or by simple tap-to-toggle system, making the overall system user friendly and easy to manage.

So getting in detail about the model, we have temperature sensor which works along with smoke sensor to check the presence of fire at home, PIR motion sensor to detect the human movement in the house, and relay connected devices so that they can be easily toggled by the microcontroller. The brain of our model is an Ethernet based Intel Galileo 2nd Generation Board which let our devices and sensors connected to the internet. The2nd generation Intel Galileo board provides a single board which is based on the Intel Quark SoCX1000, a 32-bit Intel Pentium processor- class system on a chip (SoC).It is Arduino certificated and designed to be hardware, software, and pin compatible with large range of Arduino Uno R3 shields.

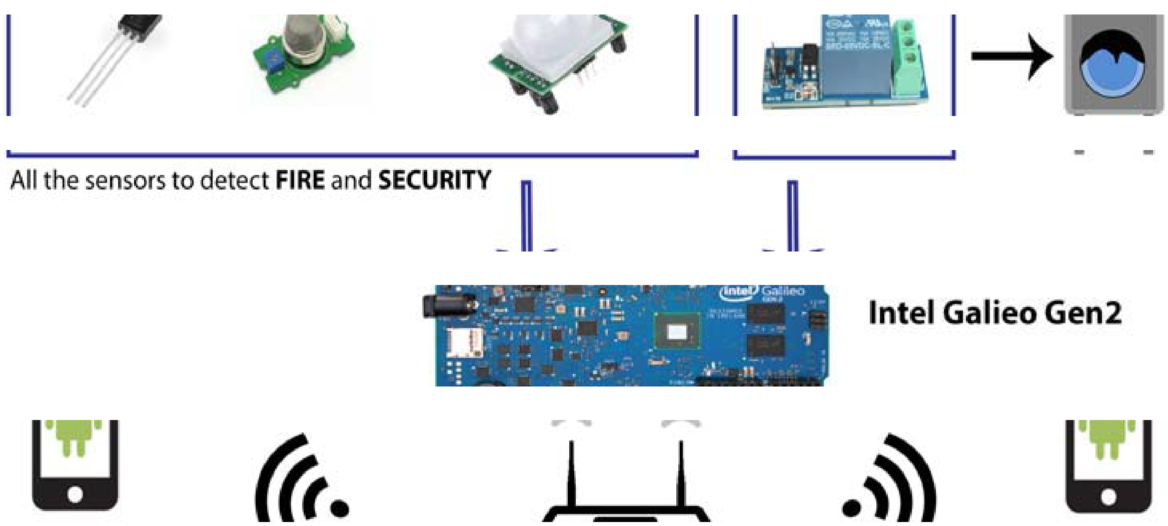


Fig 2.1: Layout of the existed model

On the other side of our model, we have an android based mobile app that has options to track the switching time of the devices, controlling the switching of devices either throughtouch mode or voice mode and also generates alerts in case of security breach or fire. The app is android based which is connected to the internet thorough either Wi-Fi or mobile dat. It connects to the Intel Galileo based server over the internet and lets the users to monitor with the help of an internal mobile timer and toggles the switching by tap-to-touch or voice using GoogleAPI speech recognition tool. User can manually switch on or off the PIR sensor or the fire tracking system and even get alerts in case they do detect a change. The alert is sent real time to the user app and shown in the alert tab. Thus, an energy monitoring security system is being set up in the home with a user-friendly mobile app to make your home a smart and an intelligent home

The above model which was proposed is very cost effectiviness and it design is so complicated than our project,So this existed model is rectified and we have proposed a new model.

Overview of Technologies

3.1 EMBEDED SYSTEMS

System

A system is an arrangement in which all its unit assemble work together according to a set of rules. It can also be defined as a way of working, organizing or doing one or many tasks according to a fixed plan. For example, a watch is a time displaying system. Its components follow a set of rules to show time. If one of its parts fails, the watch will stop working. So we can say, in a system, all its subcomponents depend on each other.

Embedded System

As its name suggests, Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke.

An embedded system has three components −

- It has hardware.
- It has application software.
- It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small scale embedded system may not have RTOS.

So we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

Characteristics of an Embedded System

- **Single-functioned** − An embedded system usually performs a specialized operation and does the same repeatedly. For example: A pager always functions as a pager.
- **Tightly constrained** − All computing systems have constraints on design metrics, but those on an embedded system can be especially tight. Design metrics is a measure of an implementation's features such as its cost, size, power, and performance. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.

- **Reactive and Real time** − Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay. Consider an example of a car cruise controller; it continually monitors and reacts to speed and brake sensors. It must compute acceleration or de-accelerations repeatedly within a limited time; a delayed computation can result in failure to control of the car.
- **Microprocessors based** − It must be microprocessor or microcontroller based.
- **Memory** − It must have a memory, as its software usually embeds in ROM. It does not need any secondary memories in the computer.
- **Connected** − It must have connected peripherals to connect input and output devices.
- **HW-SW systems** − Software is used for more features and flexibility. Hardware is used for performance and security.

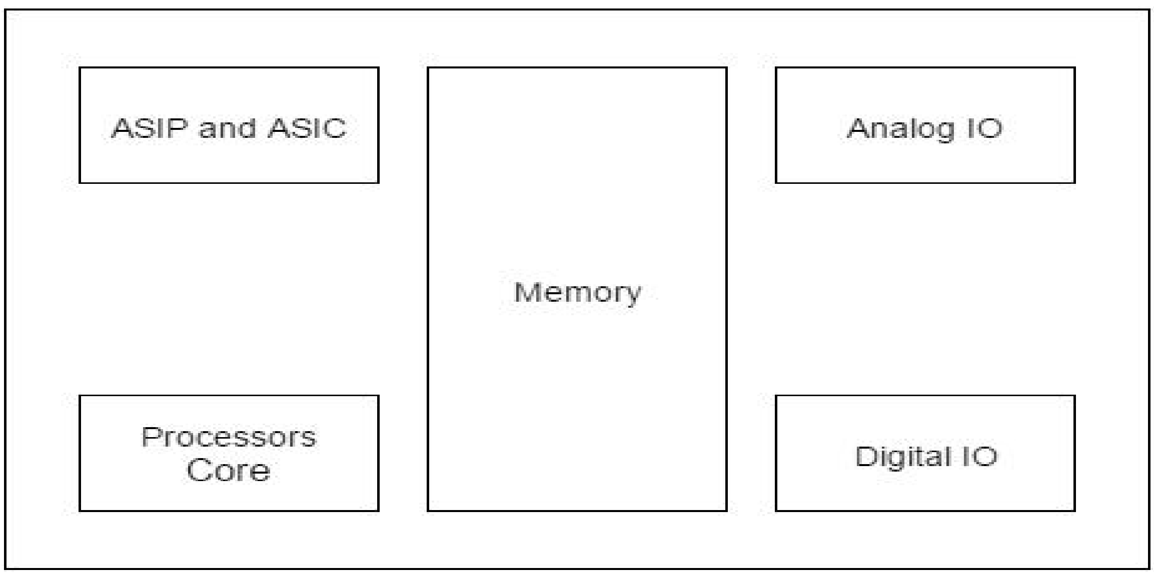


Fig 3.1: Overview Embedded System

Advantages

- Easily Customizable
- Low power consumption
- Low cost
- Enhanced performance

Disadvantages

- High development effort
- Larger time to market

Basic Structure of an Embedded System

The following illustration shows the basic structure of an embedded system −

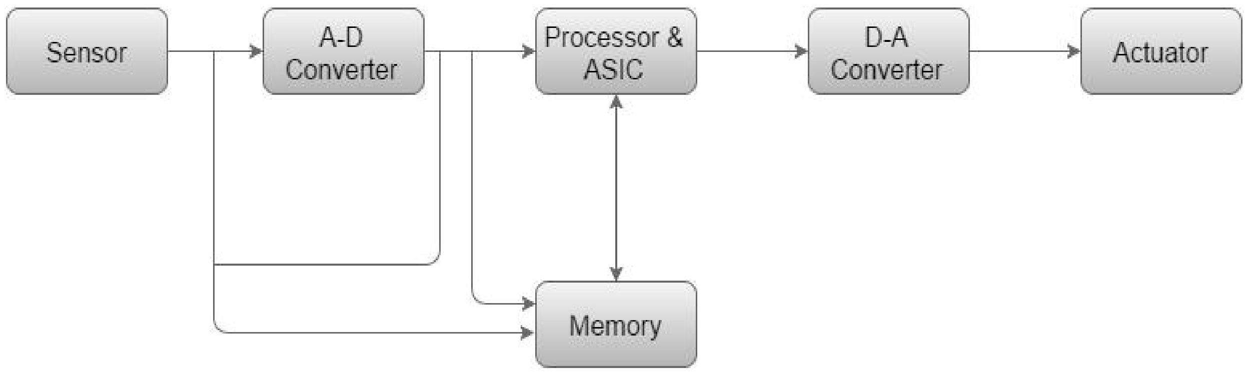


Fig 3.2: Basic Structure of an Embedded System

- **Sensor** − It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** − An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
- **Processor & ASICs** − Processors process the data to measure the output and store it to the memory.
- **D-A Converter** − A digital-to-analog converter converts the digital data fed by the processor to analog data
- **Actuator** − An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

3.2 IOT TECHNOLOGY

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with **unique identifiers** and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

IoT has evolved from the convergence of **wireless technologies**, micro-electromechanical systems (MEMS), microservices and the internet. The convergence has helped tear down the silo walls between operational technology (OT) and information technology (IT), allowing unstructured **machine-generated data** to be analyzed for insights that will drive improvements.

Kevin Ashton, cofounder and executive director of the Auto-ID Center at MIT, first mentioned the Internet of Things in a presentation he made to Procter & Gamble in 1999. Here's how Ashton explains the potential of the Internet of Things:Today computers -- and, therefore, the internet -- are almost wholly dependent on human beings for information. Nearly all of the roughly 50 petabytes (a peta byte is 1,024 **terabytes**) of data available on the internet were first captured and created by human beings by typing, pressing a record button, taking a digital picture or scanning a bar code. The problem is, people have limited time, attention and accuracy -- all of which means they are not very good at capturing data about things in the real world. If we had computers that knew everything there was to know about things -- using data they gathered without any help from us -- we would be able to track and count everything and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling and whether they were fresh or past their best."

IPv6's huge increase in address space is an important factor in the development of the Internet of Things. According to Steve Leibson, who identifies himself as "occasional docent at the Computer History Museum," the address space expansion means that we could "assign an IPV6 address to every atom on the surface of the earth, and still have enough addresses left to do another 100+ earths." In other words, humans could easily assign an IP address to every "thing" on the planet. An increase in the number of smart nodes, as well as the amount of upstream data the nodes generate, is expected to raise new concerns about **data privacy, data sovereignty** and security.

Practical applications of IoT technology can be found in many industries today, including **precision agriculture, building management, healthcare, energy and transportation.** Connectivity options for electronics engineers and application developers working on products and systems for the Internet of Things include:

Although the concept wasn't named until 1999, the Internet of Things has been in development for decades. The first **internet appliance,** for example, was a Coke machine at Carnegie Melon University in the early 1980s. The programmers could connect to the machine over the internet, check the status of the machine and determine whether or not there would be a cold drink awaiting them, should they decide to make the trip down to the machine.

The Future of IoT

As far as the reach of the Internet of Things, there are more than **12 billion devices** that can currently connect to the Internet, and researchers at IDC estimate that by 2020 there will be 26 times more connected things than people.

Gartner, Inc. forecasts that 4.9 billion connected things will be in use in 2015, up 30 percent from 2014, and will reach 25 billion by 2020. The Internet of Things (IoT)* has become a powerful force for business transformation, and its disruptive impact will be felt across all industries and all areas of society.

Category	2013	2014	2015	2020
Automotive	96.0	189.6	372.3	3,511.1
Consumer	1,842.1	2,244.5	2.874.9	13,172.5
Generic Business	395.2	479.4	623.9	5,158.6
Vertical Business	698.7	836.5	1,009.4	3,164.4
Grand Total	3,032.0	3,750.0	4,880.6	25,006.6

Table 3.1: Internet of Things Units Installed Base by Category

3.3 GPRS TECHNOLOGY

General Packet Radio System is also known as GPRS is a third-generation step toward internet access. GPRS is also known as GSM-IP that is a Global-System Mobile Communications Internet Protocol as it keeps the users of this system online, allows to make voice calls, and access internet on-the-go. Even Time-Division Multiple Access (TDMA) users get benefit from this system, as it provides packet radio access. GPRS also permits the network operators to execute an Internet Protocol (IP) based core architecture for integrated voice and data applications, whichcontinuesto be used and expanded for 3G services. GPRS supersedes the wired connections, as this system has simplified access to the packet data networks like the internet.

The packet radio principle is employed by GPRS to transport user data packets in a structuralway between GSM mobile stations and external packet data networks. These packets can be directly routed to the packet switched networks from the GPRS mobile stations.

Who Owns GPRS?

The GPRS specifications are written by the European Telecommunications Standard Institute (ETSI), the European counterpart of the American National Standard Institute
(ANSI).

Key Features

Following three key features describe wireless packet data:

- Always online feature -Removes the dial-up process, making applications only one click away.
- Upgrade to existing systems -Operators do not need to replace their equipment; rather, GPRS is added on top of the existing infrastructure.
- An integral part of future 3G systems -GPRS is the packet data core network for 3G systems EDGEand WCDMA.

Goals of GPRS

GPRS is the first step toward an end-to-end wireless infrastructure and has the following goals:

•Open architecture
•Consistent IP services
•Same infrastructure for different air interfaces
•Integrated telephony and Internet infrastructure
•Leverage industry investment in IP
•Service innovation independent of infrastructure

Benefits of GPRS

- **Higher Data Rate**

GPRS benefits the users in many ways, one of which is higher data rates in turn of shorter access times. In the typical GSM mobile, setup alone is a lengthy process, and equally rates for data permission are restrained to 9.6 kbps. The session establishment time offered while GPRS is in practice is lower than one second and ISDN-line data rates are up to many 10 kbps.

- **Easy Billing**

GPRS packet transmission offers a more user-friendly billing than that of circuit switched services. In circuit switched services, billing is based on the duration of the connection. This is unsuitable for applications with bursty traffic. The user must pay for the entire airtime, even for the idle periods when no packet has been sent (e.g., when the user reads a Web page).In contrast to this, with packet switched services, billing can be based on the amount of transmitted data. The advantage for the user is that he or she can be "online" over a long period of time but will be billed based on the transmitted data volume only GPRS has opened a wide range of unique services to the mobile wireless subscribers.

Characteristics of GPRS

Following are some of the characteristics that have opened a market full of enhanced value services to the users:
•**Mobility -**The ability to maintain constant voice and data communications while on the move.
•**Immediacy** -Allows subscribers to obtain connectivity when needed, regardless of location and without a lengthy login session.
•**Localization -**Allows subscribers to obtain information relevant to their current location.

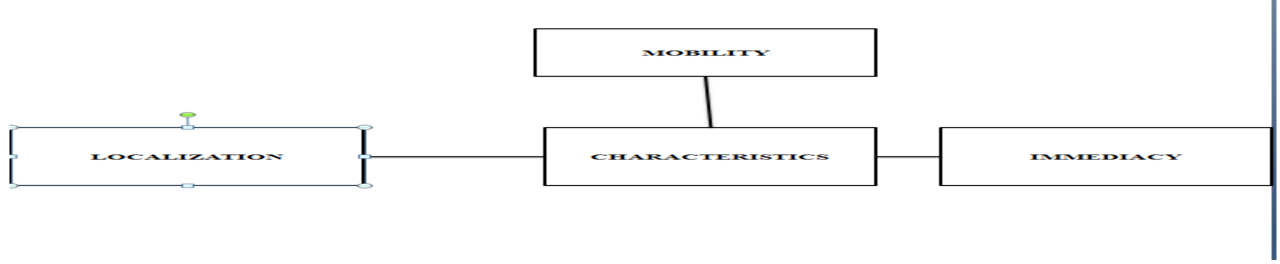


Fig 3.2: GPRS Characteristics Overview

Using the above three characteristics, varied possible applications are being developed for the mobile subscribers. These applications, in general, can be divided into two high-level categories:

- **Corporation**
- **Consumer**

These two levels further include:

- **Communications -** E-mail, fax, unified messaging, and intranet/internet access, etc.
- **Value-added services -** Information services, games, etc.
- **E-commerce -** Retail, ticket purchasing, banking and financial trading, etc.
- **Location-based applications -** Navigation, traffic conditions, airline/rail schedules, location finder, etc.
- **Vertical applications -** Freight delivery, fleet management, and sales-force automation.
- **Advertising -** Advertising may be location sensitive. For example, a user entering a mall can receive advertisements specific to the stores in that mall.

Along with the above applications, non-voice services such as SMS, MMS, and voice calls are also possible with GPRS. Closed User Group (CUG) is a common term used after GPRS is in the market. In addition, it is planned to implement supplementary services, such as Call Forwarding Unconditional (CFU), and Call Forwarding on Mobile subscriber Not Reachable (CFNRc), and Closed User Group (CUG).

GPRS Architecture Diagram:

GPRS attempts to reuse the existing GSM network elements as much as possible, but to effectively build a packet-based mobile cellular network, some new network elements, interfaces, and protocols for handling packet traffic are required. The architecture diagram of GPRS is as follows:

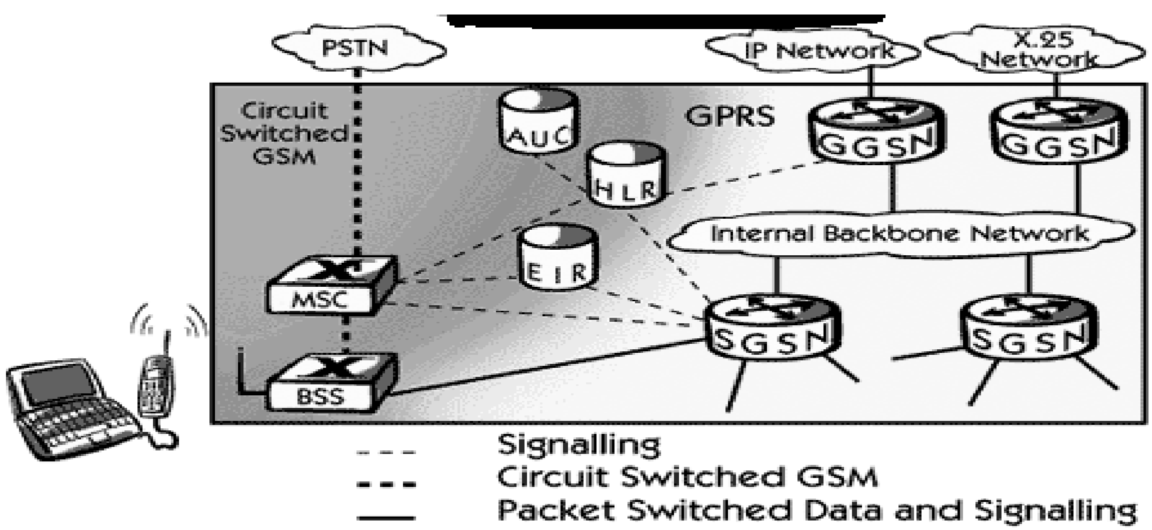


Fig 3.3: GPRS Architecture

GSM Network Element	Modification or Upgrade Required for GPRS
Mobile Station (MS)	New Mobile Station is required to access GPRS services. These new terminals will be backward compatible with GSM for voice calls.
BTS	A software upgrade is required in the existing Base Transceiver Station (BTS).
BSC	The Base Station Controller (BSC) requires a software upgrade and the installation of a new hardware called the 'packet control unit (PCU).' The PCU directs the data traffic to the GPRS network and can be a separate hardware element associated with the BSC.
GPRS Support Nodes (GSNs)	The deployment of GPRS requires the installation of new core network elements called the 'serving GPRS support node (SGSN)' and 'gateway GPRS support node (GGSN).'
Databases (HLR, VLR, etc.)	All the databases involved in the network will require software upgrades to handle the new call models and functions introduced by GPRS.

Table 3.2: GPRS Architecture Elements

Hardware Implementation

This chapter briefly explains about the Hardware Implementation of the project. It discusses the design and working of the design with the help of block diagram and circuit diagram and explanation of circuit diagram in detail. It explains the features, timer programming, serial communication, interrupts of pic microcontroller. It also explains the various modules used in this project.

Project Design

The implementation of the project design can be divided in two parts.

- Hardware implementation
- Firmware implementation

Hardware implementation deals in drawing the schematic on the plane paper according to the application, testing the schematic design over the bread board using the various IC'S to find if the design meets the objective, carrying out the PCB layout of the schematic tested on breadboard, finally preparing the board and testing the design hardware.

The firmware part deals in programming the microcontroller so that it can control the operation of the IC's used in the implementation. In the present work, we have used the proteous design software for PCB circuit design, the MPLAB ide software development tool to write and compile the source code each has been written in the c language.

The project design and principles are explained in this chapter using the block diagram and circuit diagram. The block diagram discusses about the required components of working condition is explained using circuit diagram and system writing diagram.

Block Diagram of the Project:

The block diagram of the design is as shown in Fig 3.1.the transmitter consists of power supply unit, pic microcontroller, Gas Sensor, Temperature Sensor, Humidity Sensor, IOT Module and MAX232.

The IOT Module can be inserted a sim that sim can transmits the analysed data of the Gas Sensor, Temperature Sensor, Humidity Sensor data to the cloud environment, then we can check and access from any point of the world, as per the user requirement.

4.1 Block Diagram

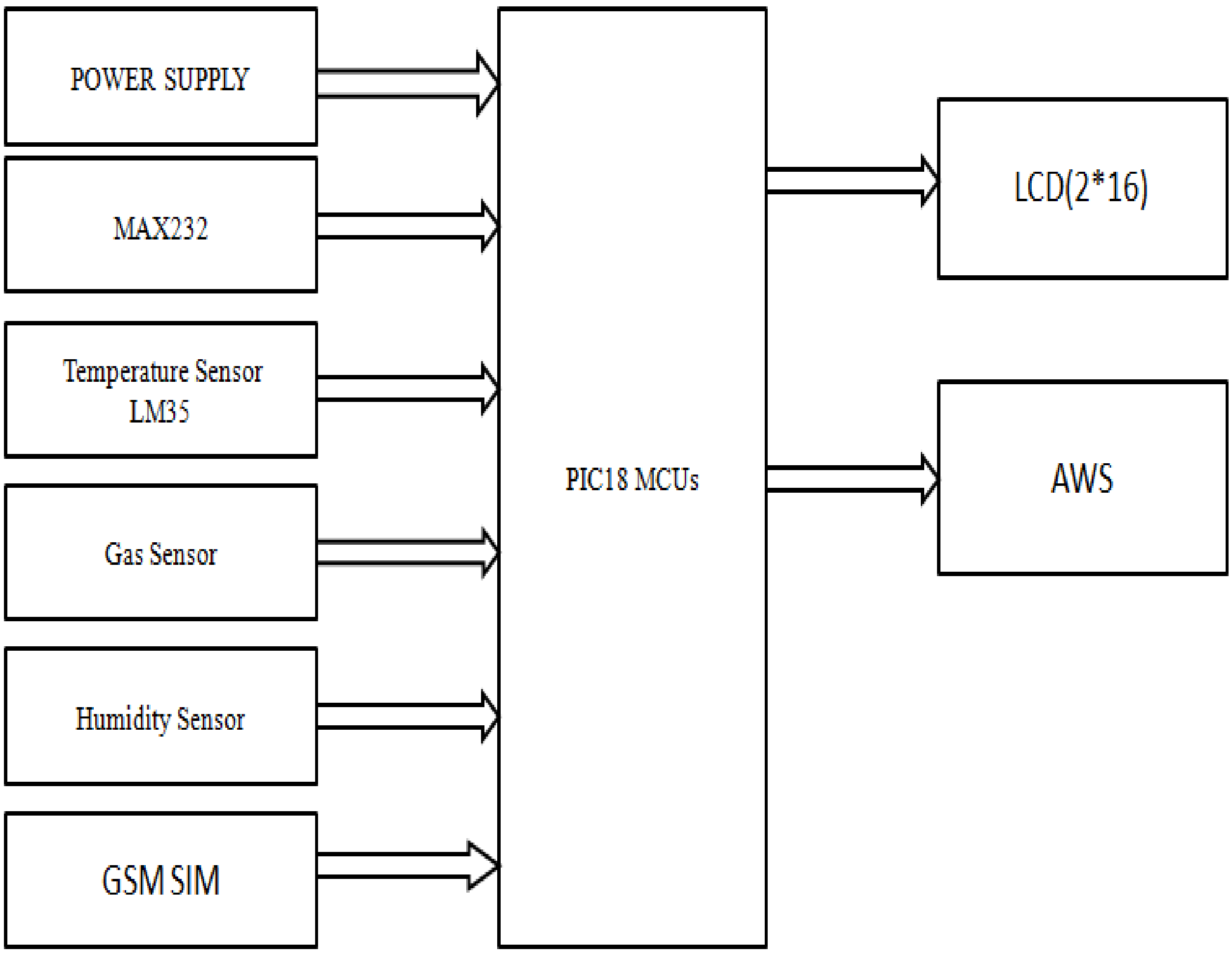


Fig 4.1: Block diagram of project

4.2 POWER SUPPLY

The ac voltage, typically 220V rms, is connected to a transformer, which step that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting DC voltage, usually has some ripple or AC voltage variation. A regulator circuit removes the ripples and also remains the same DC value even if the input DC voltage varies, or the load connected to the output DC voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units. The power supply block diagram is shown in Fig 3.2

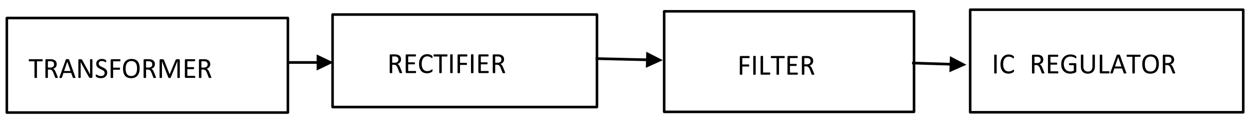


Fig 4.2: Power supply Block diagram

Transformer

The transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op–amp. The advantages of using a precision rectifier are it will give a peak voltage output as DC; the rest of the circuits will give only RMS output.

Rectifier

The rectifier is a device which converts Alternating current into some DC form. The output of the rectifier is in pulsated DC form. There are some semiconductor devices used to convert AC to pulsating DC named diodes. Normally we are using PN junction diodes in rectifiers. These diodes allow the current flow to it when the incoming voltage is higher than or equal to 0.7 V. Otherwise, it will act as an open circuit.

Filter

The next section of the power supply is filtered section. That is nothing but capacitor. The capacitor is used as a filter to convert Pulsated DC to pure DC. We can utilize the capacitor’s charging, discharging characteristics and convert the pulsated DC to pure DC.

Voltage Regulator

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for referral source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustable set voltage. The regulators can be selected for operation with load currents from hundreds of mille amperes to tens of amperes, corresponding to power ratings from mille watts to tens of watts.

4.3 LIQUID CRYSTAL DISPLAY (LCD)

Liquid Crystal Displays (LCD's) have materials, which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal.

An LCD consists of two glass panels, with the liquid crystal material sand witched in between them. The inner surface of the glass plates is coated with transparent electrodes which define the character, symbols or patterns to be displayed polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle.

One each polarizes are posted outside the two glass panels. These polarizes would rotate the light rays passing through them to a definite angle, in a particular direction. When the LCD is in the off state, light rays are rotated by the two polarizes and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarizes, which would result in activating / highlighting the desired characters. The LCD's are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations.

The LCD does not generate light and so light is needed to read the display. By using backlighting, reading is possible in the dark. The LCD's have long life and a wide operating temperature range. Changing the display size or the layout size is relatively simple which makes the LCD's more customers friendly.

The LCDs used exclusively in watches, calculators and measuring instruments are the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCDs being extensively used in telecommunications and entertainment electronics. The LCDs have even started replacing the Cathode Ray Tubes (CRTs) used for the display of text and graphics, and also in small TV applications.

Crystalloids dot–matrix (alphanumeric) liquid crystal displays are available in TN, STN types, with or without backlight. The use of C-MOS LCD controller and driver ICs result in low power consumption. These modules can be interfaced with a 4-bit or 8-bit microprocessor /Micro controller.

- The built-in controller IC has the following features:
- Correspond to high speed MPU interface (2MHz)
- 80 x 8 bit display RAM (80 Characters max)
- 9,920-bit character generator ROM for a total of 240 character fonts. 208 character fonts (5 x 8 dots) 32 character fonts (5 x 10 dots)
- 64 x 8 bit character generator RAM, 8 character generator RAM, 8 character fonts (5 x 8 dots) 4 character fonts (5 x 10 dots)
- Programmable duty cycles
- 1/8 – for one line of 5 x 8 dots with cursor
- 1/11 – for one line of 5 x 10 dots with cursor
- 1/16 – for one line of 5 x 8 dots with cursor
- Wide range of instruction functions display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift.
- Automatic reset circuit, which initializes the controller / driver ICs after power on.

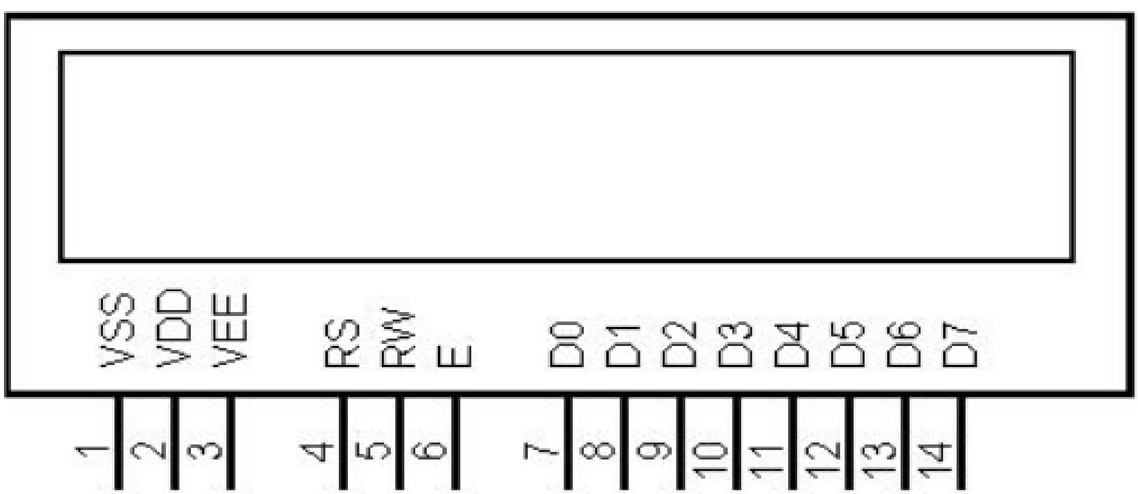


Fig 4.3: Block diagram of LCD Display

Pin No.	Symbol	Level	Description
1	VSS	0V	Ground
2	VDD	5.0V	Supply Voltage for logic
3	NC	—	
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	NC	—	
16	NC	—	

Table 4.2: Interface Pin Function

4.4 MICROCONTROLLER

Introduction to Microcontroller

Microcontrollers are destined to play an increasingly important role in revolutionizing various industries and influencing our day to day life stronger than one can imagine. Since its emergence in the early 1980's the microcontroller has been recognized as a general purpose building block for intelligent digital systems. It is found using diverse area, starting from simple children's toys to highly complex spacecraft. Because of its versatility and many advantages, the application domain has spread in all conceivable directions, making it ubiquitous. As a consequence, it has generated a great deal of interest and enthusiasm among students, teachers and practicing engineers, creating an acute education need for imparting the knowledge of microcontroller based system design and development. It identifies the vital features responsible for their tremendous impact, the acute educational need created by them and provides a glimpse of the major application area.

Microcontroller

A microcontroller is a complete microprocessor system built on a single IC. Microcontrollers were developed to meet a need for microprocessors to be put into low cost products. Building a complete microprocessor system on a single chip substantially reduces the cost of building simple products, which use the microprocessor's power to implement their function, because the microprocessor is a natural way to implement many products. This means the idea of using a microprocessor for low cost products comes up often. But the typical 8-bit microprocessor based system, such as one using a Z80 and 8085 is expensive. Both 8085 and Z80 system need some additional circuits to make a microprocessor system. Each part carries the costs of money. Even though a product design may require only very simple system, the parts needed to make this system as a low cost product.

To solve this problem microprocessor system is implemented with a single chip microcontroller. This could be called microcomputers, as all the major parts are in the IC. Most frequently they are called microcontroller because they are used they are used to perform control functions.

The microcontroller contains a full implementation of a standard MICROPROCESSOR, ROM, RAM, I/0, CLOCK, TIMERS, and also SERIAL PORTS. Microcontroller also called "system on a chip" or "single chip microprocessor system" or "computer on a chip".

A microcontroller is a Computer-On-A-Chip, or, if you prefer, a single-chip computer. Micro suggests that the device is small, and the controller tells you that the device' might be used to control objects, processes, or events. Another term to describe a microcontroller is embedded controller, because the microcontroller and its support circuits are often built into, or embedded in, the devices they control.

Today microcontrollers are very commonly used in a wide variety of intelligent products. For example, most personal computer keyboards and implemented with a microcontroller. It replaces Scanning, Debounce, Matrix Decoding, and Serial transmission circuits. Many low cost products, such as Toys, Electric Drills, Microwave Ovens, VCR and a host of other consumer and industrial products are based on microcontrollers. The typical block diagram of microcontroller is shown in Fig 3.3

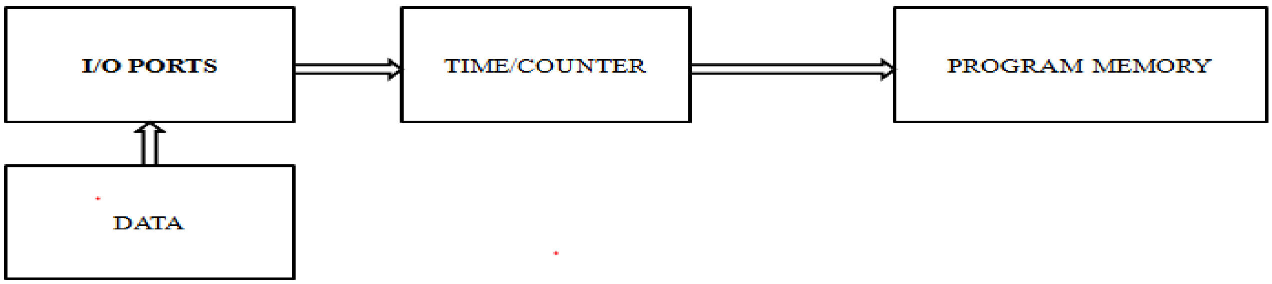


Fig 4.4: Block Diagram of Typical Microcontroller

4.5 PIC MICROCONTROLLER

PIC is abbreviated as **Peripheral Interface Controller**

Microcontroller PIC16F877A is one of the PIC Micro Family microcontrollers which is popular at this moment, start from beginner until all professionals. Because of its user friendly nature and use of FLASH memory technology where programs can be read and written thousand times.

PIC16F877A has 40 pins in which 33 are I/O pins. Hence PIC16F877A perfectly fits in to many uses, from automotive industries and controlling home appliances to industrial instruments, remote sensors, electrical door locks and safety devices. It is also ideal for smart cards as well as for battery supplied devices because of its low consumption.

EEPROM memory makes it easier to apply microcontrollers to devices where permanent storage of various parameters is needed (codes for transmitters, motor speed, receiver frequencies, etc.). Low cost, low consumption, easy handling and flexibility make PIC16F877A applicable even in areas where microcontrollers had not previously been considered (example: timer functions, interface replacement in larger systems, coprocessor applications, etc.).

In System Programmability of this chip (along with using only two pins in data transfer) makes possible the flexibility of a product, after assembling and testing have been completed. This capability can be used to create assembly-line production, to store calibration data available only after final testing, or it can be used to improve programs on finished products.

4.5.1 FEATURES

HIGH-PERFORMANCE RISC CPU

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, up to 368 x 8 bytes of Data Memory (RAM), up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers.

PERIPHERAL FEATURES

- Timer0: 8-bit timer/counter with 8-bit prescalar
- Timer1: 16-bit timer/counter with prescalar,can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescalar and postscaler
- Two Capture, Compare, PWM modules
- Capture is 16-bit, max. resolution is 12.5 ns
- Compare is 16-bit, max. resolution is 200 ns
- PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI(Master mode) and I2C (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR).

ANALOG FEATURES

- 10-bit, up to 8-channel Analog-to-Digital
- Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with Two analog comparators
- Programmable on-chip voltage reference
- (VREF) module
- Programmable input multiplexing from device
- Inputs and internal voltage reference
- Comparator outputs are externally accessible

4.5.2 CHARACTERISTICS OF THE RISC SYSTEM

- Instruction set of 35 commands
- Instructions are carried out in 2 clock cycles
- You can calculate the execution time of the command in accordance with the frequency clock (if the frequency clock is 20Mhz, execution time of a command is 200ns)
- The ports of 28/40/44 pins microcontrollers are the same

3.5.3 PIC FAMILY:

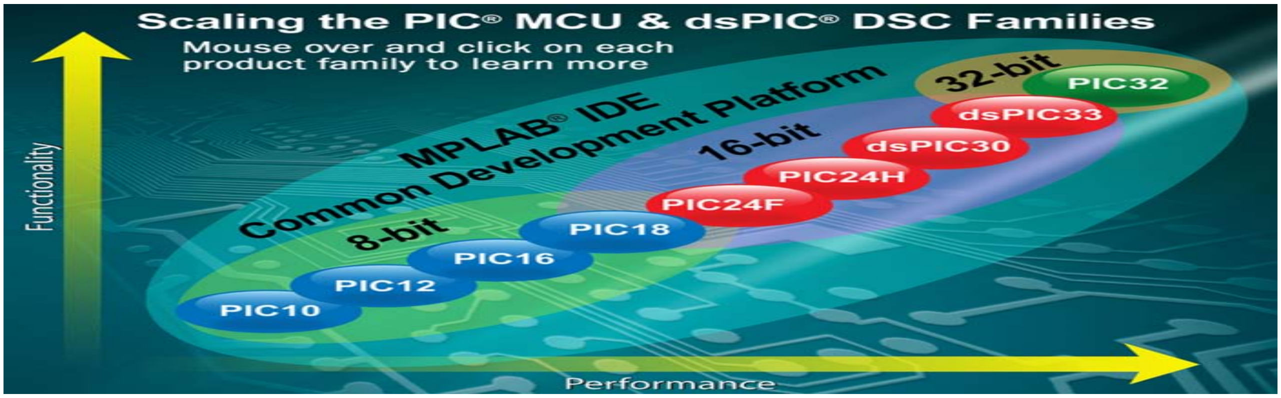


Fig 4.5: Graph of pic family

PIC Microcontrollers MCU from Microchip Company divided into 4 large families. Each family has a variety of components that provide built-in special features

The first family, PIC10 (10FXXX) - is called Low End

The PIC10FXXX devices from Microchip Technology are low-cost, high-performance, 8-bit, fully static, Flash-based CMOS microcontrollers. They employ RISC architecture with only 33 single-word/ single-cycle instructions. The 12-bit wide instructions are highly symmetrical. The easy-to-use and easy to remember instruction set reduces development time significantly. The PIC10FXXX devices contain an 8-bit ALU and working register.

The second family, PIC12 (PIC12FXXX) – is called Mid-Range

Mid-Range devices feature 14-bit program word architecture and are available in 8 to 64-pin packages that offer an operating voltage range of 1.8-5.5V, small package footprints, interrupt handling, an 8-level hardware stack, multiple A/D channels and EEPROM data memory. Mid-range devices offer a wide range of package options and a wide range of peripheral integration. These devices feature various serial analog and digital peripherals, such as: SPI, I2C™, USART, LCD and A/D converters.

The third family is PIC16 (16FXXX)

With six variants ranging from 3.5K-14 Kbytes of Flash memory, up to 256 bytes of RAM and a mix of peripherals including EUSART, CCP and onboard and comparators. These devices are well suited for designers with applications that need more code space or I/O than 14-pin variants supply, and are looking to increase system performance and code efficiency by employing hardware motor control and communications capability.

The fourth family is PIC 17/18(18FXXX)

The PIC18 family utilizes 16-bit program word architecture and incorporates an advanced RISC architecture with 32 level-deep stack, 8x8 hardware multiplier, and multiple internal and external interrupts. With the highest performance in Microchips 8-bit portfolio, the PIC18 family provides up to 16 MIPS and linear memory. PIC18 is the most popular architecture for new 8-bit designs where customers want to program in C language.

4.5.4 PIN DIAGRAM FOR PIC MICRCONTROLLER

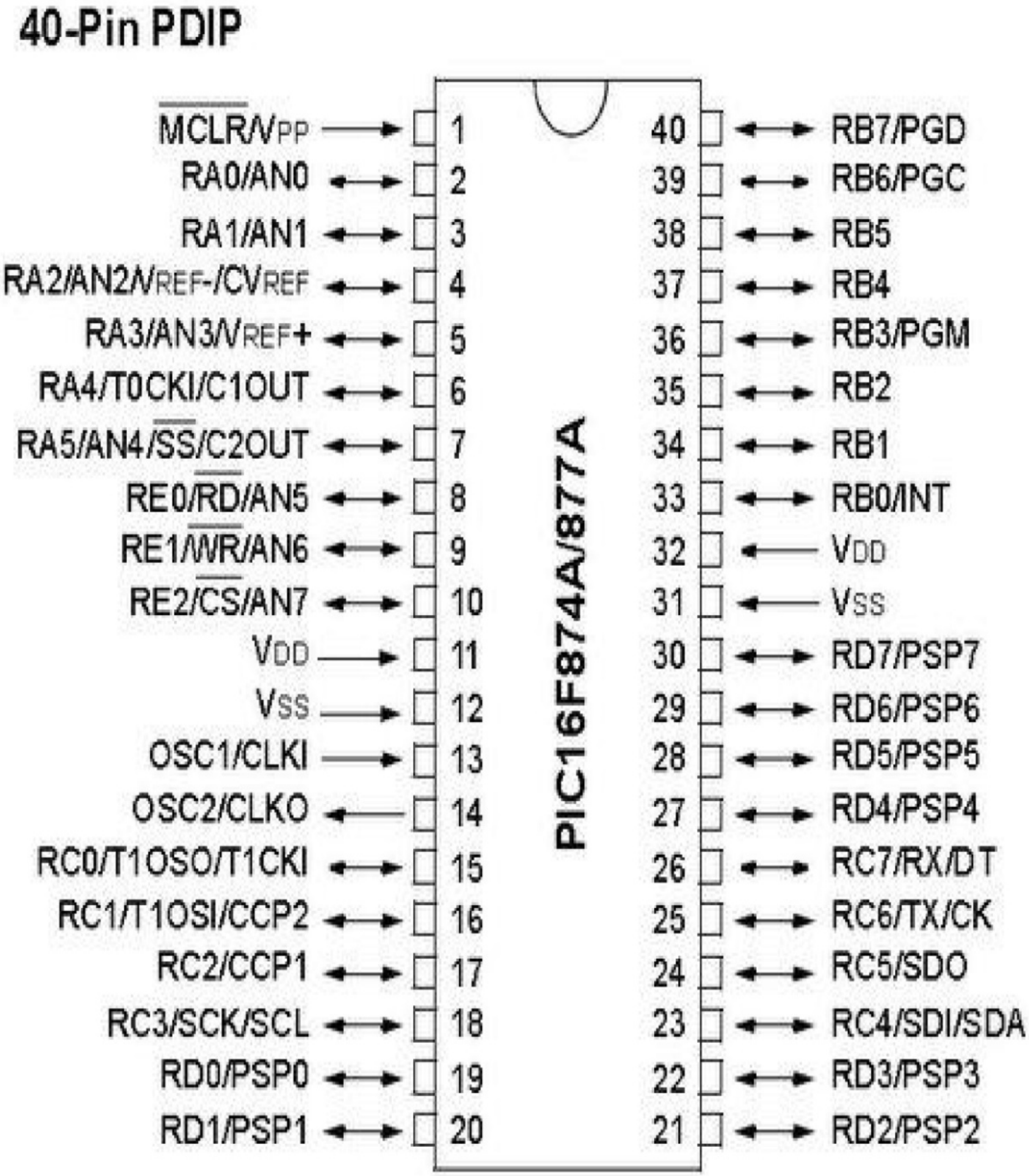


Fig 4.6: PIN DIAGRAM FOR PIC MICRCONTROLLER

4.5.5 PIC16F877A ARCHITECTURE

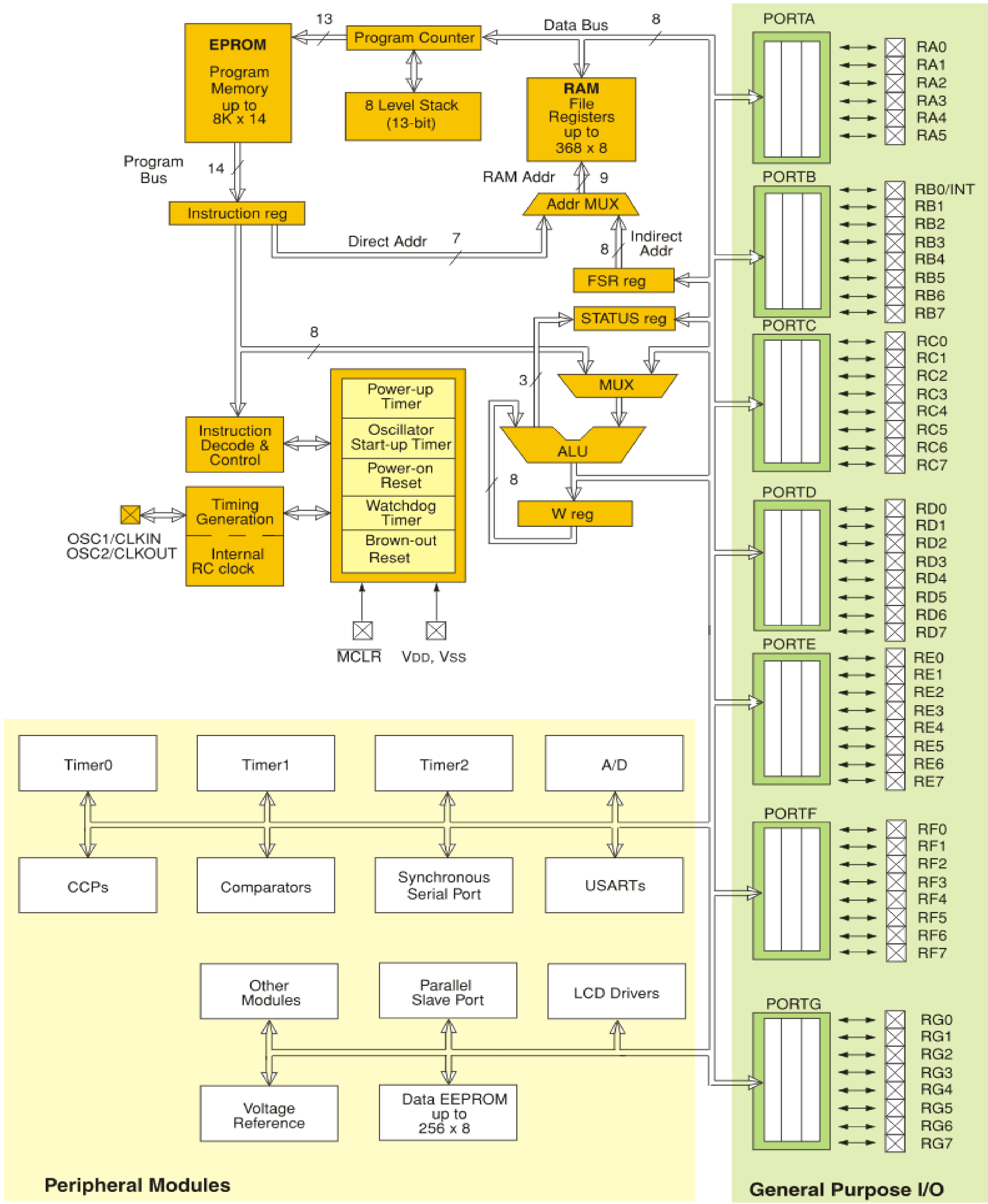


Fig 4.7: PIC16F877A architecture

4.5.6 MEMORY ORGANIZATION

Memory of the PIC16F877 divided into 3 types of memories:

4.5.6.1 Program Memory

A memory that contains the program (which we had written), after we've burned it. As a reminder, Program Counter executes commands stored in the program memory, one after the other.

The PIC16F877A devices have a 13-bit program counter capable of addressing an 8K word x 14 bit program memory space. This memory is used to store the program after we burn it to the microcontroller. The PIC16F876A/877A devices have 8K words x 14 bits of Flash program memory that can be electrically erased and reprogrammed. Each time we burn program into the micro, we erase an old program and write a new one.

Program Counter (PC) keeps track of the program execution by holding the address of the current instruction. It is automatically incremented to the next instruction during the current instruction execution.

The PIC16F877A family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer are not readable or writable. In the PIC microcontrollers, this is a special block of RAM memory used only for this purpose.

4.5.6.2 Data Memory

This is RAM memory type, which contains a special registers like SFR (Special Faction Register) and GPR (General Purpose Register). The variables that we store in the Data Memory during the program are deleted after we turn off the micro. These two memories have separated data buses, which makes the access to each one of them very easy.

The data memory is partitioned into multiple banks which contain the General purpose registers and special function registers. Number of banks may vary depending on the microcontroller; for example, micro PIC16F84 has only two banks. Each bank extends up to 7Fh (128 bytes).

The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. While program is being executed, it is working with the particular bank. The default bank is BANK0.

To access a register that is located in another bank, one should access it inside the program. There are special registers which can be accessed from any bank, such as STATUS register.

The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. While program is being executed, it is working with the particular bank. The default bank is BANK0.

SENSORS

In the broadest definition, a **sensor** is an electronic component, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. A sensor is always used with other electronics, whether as simple as a light or as complex as a computer.

Sensors are used in everyday objects such as touch-sensitive elevator buttons (tactile sensor) and lamps which dim or brighten by touching the base, besides innumerable applications of which most people are never aware. With advances in micro machinery and easy-to-use microcontroller platforms, the uses of sensors have expanded beyond the traditional fields of temperature, pressure or flow measurement, for example into MARG sensors. Moreover, analog sensors such as potentiometers and force-sensing resistors are still widely used. Applications include manufacturing and machinery, airplanes and aerospace, cars, medicine, robotics and many other aspects of our day-to-day life.

A sensor's sensitivity indicates how much the sensor's output changes when the input quantity being measured changes. For instance, if the mercury in a thermometer moves 1 cm when the temperature changes by 1 °C, the sensitivity is 1 cm/°C (it is basically the slope Dy/Dx assuming a linear characteristic). Some sensors can also affect what they measure; for instance, a room temperature thermometer inserted into a hot cup of liquid cools the liquid while the liquid heats the thermometer. Sensors are usually designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages.

A good sensor obeys the following rules

- It is sensitive to the measured property
- It is insensitive to any other property likely to be encountered in its application, and
- It does not influence the measured property.

Here are a few examples of the many different types of sensors

In a mercury-based glass thermometer, the input is temperature. The liquid contained expands and contracts in response, causing the level to be higher or lower on the marked gauge, which is human-readable.

An oxygen sensor in a car's emission control system detects the gasoline/oxygen ratio, usually through a chemical reaction that generates a voltage. A computer in the engine reads the voltage and, if the mixture is not optimal, readjusts the balance.

Motion sensors in various systems including home security lights, automatic doors and bathroom fixtures typically send out some type of energy, such as microwaves, ultrasonic waves or light beams and detect when the flow of energy is interrupted by something entering its path.

A photosensor detects the presence of visible light, infrared transmission (IR), and/or ultraviolet (UV) energy.

5.1 TEMPERATURE SENSOR

Description

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temp erature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy). The LM35 -series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60-µA Current Drain

- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

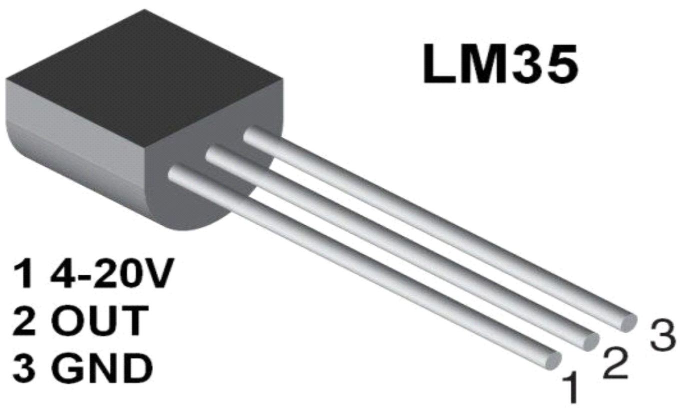


Fig 5.1: Temperature Sensor

4.2 GAS SENSOR

In current technology scenario, monitoring of gases produced is very important. From home appliances such as air conditioners to electric chimneys and safety systems at industries monitoring of gases is very crucial. **Gas sensors** are very important part of such systems. Small like a nose, gas sensors spontaneously react to the gas present, thus keeping the system updated about any alterations that occur in the concentration of molecules at gaseous state.

MQ-2 Semiconductor Sensor for Combustible Gas Sensitive material of MQ-2 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, The sensor’s conductivity is more higher along with the gas concentration rising. Please use simple electro circuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensitity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

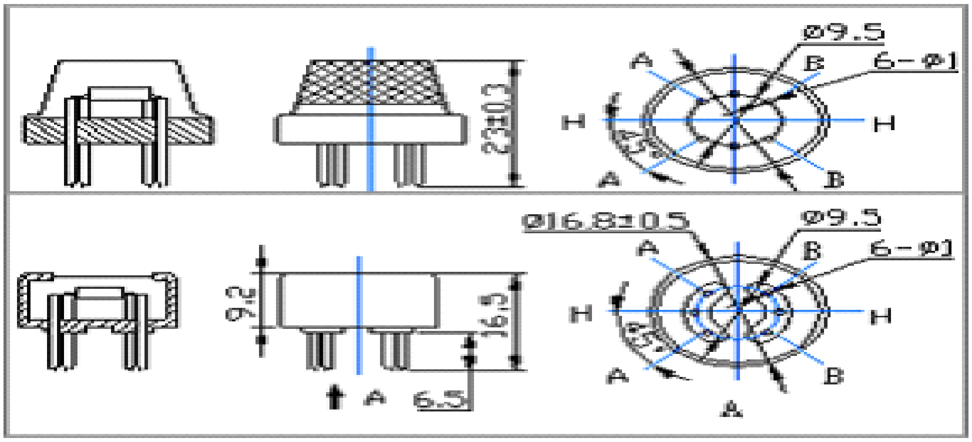


Fig 5.2: Block Diagram of Gas Sensor

Character Configuration

- Good sensitivity to Combustible gas in wide range
- High sensitivity to LPG, Propane and Hydrogen
- Long life and low cost
- Simple drive circuit

Application

- Domestic gas leakage detector
- Industrial Combustible gas detector
- Portable gas detector
- Technical Data Basic test loop

The above is basic test circuit of the sensor.

Resistance of sensor (Rs): Rs= (Vc/VRL-1)×RL Sensitivity Characteristics Influence of Temperature/Humidity the MQ-2, ordinate means resistance ratio of the sensor characteristics. Ordinate means resistance ratio (Rs/Ro), abscissa is concentration of gases. Rs means of the sensor (Rs/Ro), Rs means resistance of sensor resistance in different gases, Ro means resistance of in 1000ppm Butane under different tem. and humidity.

Sensor in 1000ppm Hydrogen. All test are under standard Ro means resistance of the sensor in environment of test conditions. 1000ppm Methane, 20°C65%RH Structure and configuration Structure and configuration of MQ-2 gas sensor is shown as Fig., sensor composed by micro AL2O3 ceramic tube, Tin Dioxide (SnO2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin, 4of them are used to fetch signals, and other 2 are used for providing heating current.

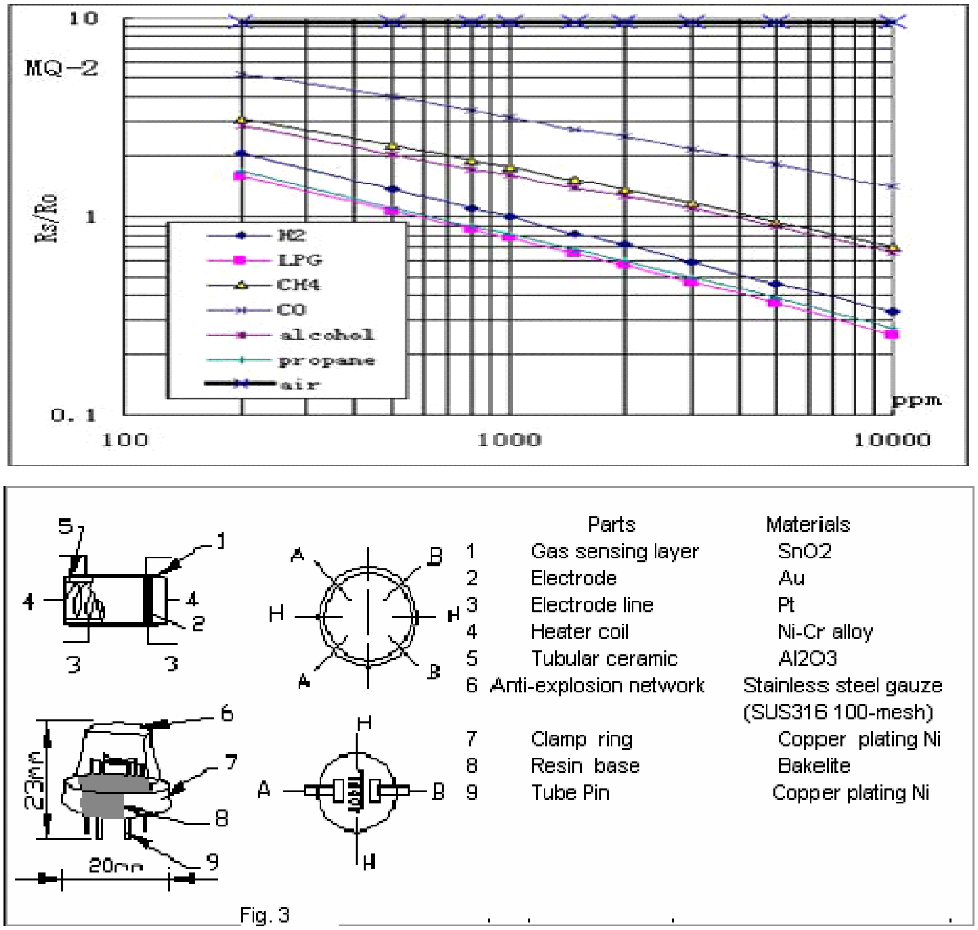


Fig 5.3: Graph of various gases detecting in gas sensor

Notification

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixature, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H2Sz, SOX，Cl2，HCletc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

Do avoid icing on sensor'surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1、3 pins or 4、6 pins, it will make lead broken, and without signal when apply on 2、4 pins

2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, if will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof without silicon gel bag with clean air.

For the sensors with long time storage but no electrify, they need long aging time for stability before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then repture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine

2.7.2 Speed: 1-2 Meter/ Minute

2.7.3 Warm-up temperature：100±20℃

2.7.4 Welding temperature：250±10℃

2.7.5 1 time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will be reduced

5.3 HUMIDITY SENSOR

DHT22 capacitive humidity sensing digital temperature and humidity module is one that contains the compound has been calibrated digital signal output of the temperature and humidity sensors. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability.

The sensor includes a capacitive sensor wet components and a high-precision temperature measurement devices, and connected with a high-performance 8-bit microcontroller. The product has excellent quality, fast response, strong anti-jamming capability, and high cost. Standard single-bus interface, system integration quick and easy. Small size, low power consumption, signal transmission distance up to 20 meters, making it the best choice of all kinds of applications.

DHT22 has higher precision and can replace the expensive imported SHT10 temperature and humidity sensor. It can measure the environment temperature and humidity to meet the high demand.The product has high reliability and good stability.If it's used and combined with special sensor.

Caution: DHT22 digital temperature and humidity sensor is designed for analog sensor interfaces. The analog port will be used as the digital which will not occupy the original digital port of the Arduino.The lines of the sensor which can transform the analog function to digital that can be use on digital port.

Specifications

- Supply voltage: 5V
- Temperature range:-40-80℃resolution0.1℃error <±0.5℃
- Humidity range:0-100%RH resolution0.1%RH error±2%RH
- Sequence of the line:VCC,GND,S

- Size: 38 x 20mm

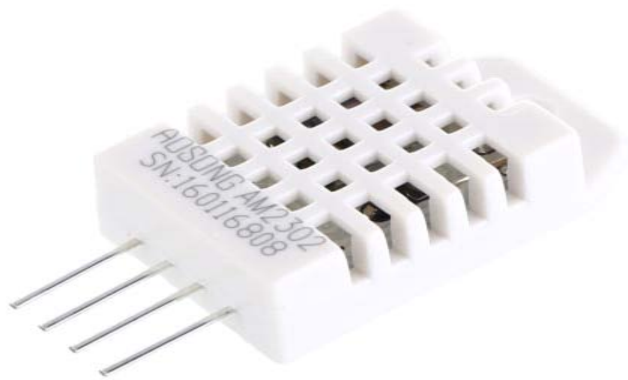


Fig 5.3: Humidity Sensor.

SERIAL COMMUNICATION

Unlike parallel communication, where several bits are send at one time, Serial Communication is a process of transmitting data bit by bit. In this tutorial, you will learn how to serially communicate a PC or any other device with a PIC microcontroller. You will also study the use of a communication component – UART (Universal Asynchronous Receiver Transmitter) present within the microcontroller.

6.1 ASYNCHRONOUS AND SYNCHRONOUS COMMUNICATION

Synchronous communication

When using the synchronous communication – the information is transmitted from the transmitter to the receiver:

- In sequence
- Bit after bit
- With fixed baud rate
- And the clock frequency is transmitted along with the bits

That means that the transmitter and the receiver are synchronized between them by the same clock frequency. The clock frequency can be transmitted along with the information, while it is encoded in the information itself, or in many cases there is an additional wire for the clock.

This type of communication is faster compare to the asynchronous communication since it is "constantly transmitting" the information, with no stops.

Asynchronous communication

When using the asynchronous communication - the transmitter and the receiver refraining to transmit long sequences of bits because **there isn't a full synchronization** between the transmitter, that sends the data, and the receiver, that receives the data.

In this case, the information is divided into frames, in the size of byte. Each one of the frame has:

- “Start” bit marks the beginning of a new frame.
- “Stop” bit marks the end of the frame.

Frames of information must not necessarily be transmitted at equal time space, since they are independent of the clock.

6.2 PIC MICROCONTROLLER SERIAL COMMUNICATION

Serial Communication using pic16F877A microcontroller, In this chapter you are going to explain about the what is serial communication? What are applications of serial communication pins of PIC16F877A microcontroller is used to transmit and receive data to and from other devices.

For example we are performing a task for the serial communication in the microcontroller

Serial Communication is the process of sending data one bit at a time. It is achieved by using the UART feature within the pic microcontroller**.** UART (Universal Asynchronous Receiver Transmitter) is a serial communication interface which is used for transmitting and receiving data. The UART feature is first initialized and then it can be used for transferring data. One of the serial communication architecture includes RS232 standard. It is commonly used in computer serial ports. Using RS232 interface, the communication between a microcontroller and a PC or two microcontrollers can be made possible.

CIRCUIT DIAGRAM

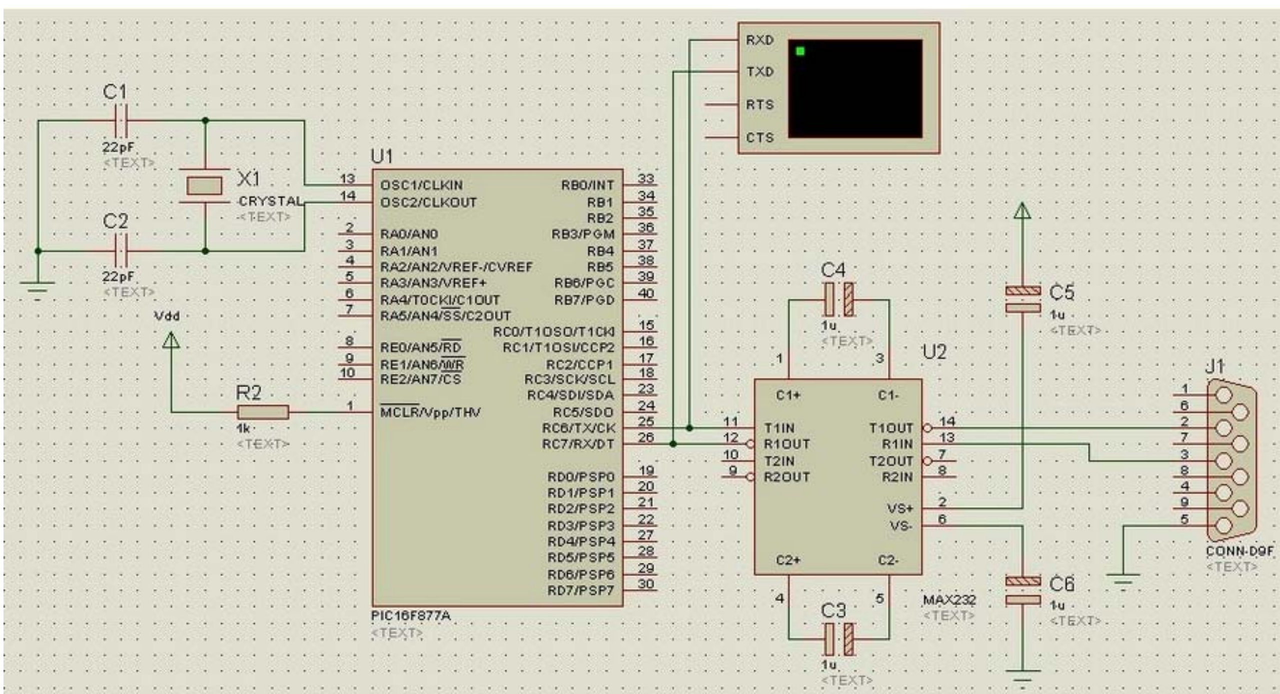


Fig 6.1: Serial Communication using pic microcontroller

DESCRIPTION

In order to make the communication possible between PIC and a PC, a DB9 connector (female) is used as shown in the schematic, whose male part is connected to the PC. The voltage levels of RS232 protocol are different from that of UART. The voltage level generally used in MCU is HIGH=5V and LOW=0V. Whereas the RS232 working voltages are HIGH=+12V and LOW=-12V.

In order to use the RS232 protocol, we will need a level converter. A level converter will convert the ±12V coming from PC into 5V which can be fed to the controller and vice versa. The level converter used here is called ICMAX232. All the connections of IC with the controller and also with the DB9 connector are shown in the circuit diagram above. RC6/TX and RC7/RX pins of the microcontroller are connected to the T1IN and R1OUT pins of the ICMAX232 respectively. Similarly, the T1OUT and R1IN pins of MAX232 are connected to the DB9 pins 2 (RX) and 3(TX) respectively. Pin 5 of the DB9 connector is grounded. A virtual terminal is connected to the RX and TX pics of the microcontroller to display the transmitting and receiving data.

APPLICATIONS OF SERIAL COMMUNICATION

In this day and age, serial communication is used in various industrial projects where data needs to be collected from different devices. HMI used in banks, marts, industries etc are also serially interconnected.

6.3 MAX232

The MAX-232 IC is an integrated circuit which consists of 16 pins and it is a resourceful IC mostly used in the voltage level signal problems. Generally, the MAX-232 IC is used in the RS232 communication system for the conversion of voltage levels on TTL devices that are interfaced with the PC serial port and the Microcontroller. This IC is used as a hardware layer converter like to communicate two systems simultaneously. The image of MAX232 is shown below.

MAX232 IC

Pin Description of MAX-232 IC

The MAX-232 IC is used to exchange the TTL/CMOS logic to RS232 logic levels through the serial communication of Microcontroller with the personal computers. The microcontroller operates at the TTL logic levels with the voltage of 0-5v, and the serial communication with PC works on the RS232 with the volts of -25V to +25V. The difference of voltage will become the makes difficult to establish the communication with each other.

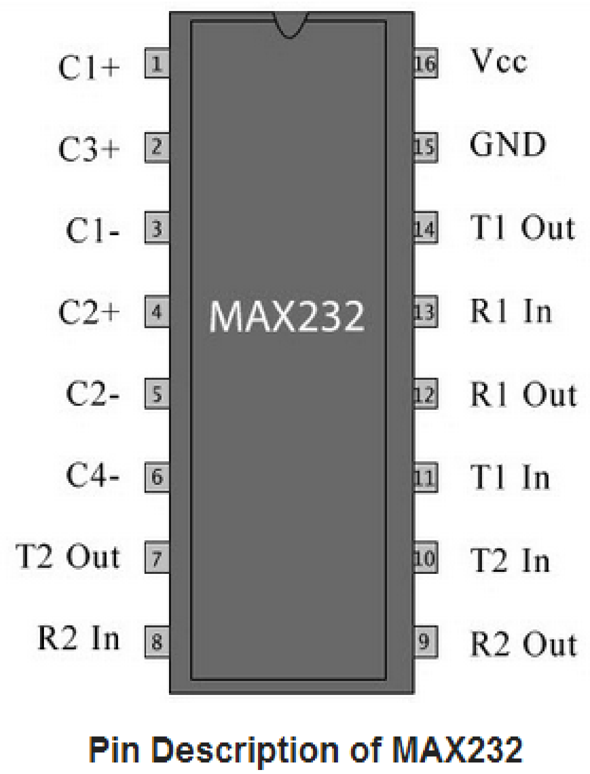


Fig 6.2: pin description of max232

The in-between links are provided through the MAX232 interface, it is a transmitter/receiver which includes the voltage generator to the supply of RS232 voltage levels through the 5V supply. The RS232 input converts to 5V TTL/COMS level through these receivers, i.e. R1 and R2 do not exclude the +-30V. The transmitters T1 and T2 are used to convert the TTL/COMS input levels to RS232 level.

The transmitter obtains input from the microcontrollers and microprocessors through serial transmission pins and the output is given to the receiver of RS232. The receiver on the other side obtains the input from transmission pin of RS232 serial port and the output is given to the receiver pin of the microcontroller. The external capacitor is added to the MAX232 with a range of 1µF to 22µF.

The pin configuration of MAX232 and the description of the each pin of is discussed in the tabular form

S. No	Name	Function
1	C1 +	Positive Voltage Multiplier Unit of External Capacitor
2	C3 +	Positive Voltage Multiplier Unit of External Capacitor
3	C1-	Negative Voltage Multiplier unit of External capacitor
4	C2 +	Positive Voltage Multiplier Unit of External Capacitor
5	C2 –	Negative Voltage Multiplier unit of External capacitor
6	C4-	Negative Voltage Multiplier unit of External capacitor
7	T2 Out	Transmitter data Output from RS232
8	R2 In	Receiver Data Input from RS232
9	R2 Out	Receiver Data Output from TTL logic level
10	T2 In	Transmitter Data Input from TTL logic level
11	T1 In	Transmitter Data Input from TTL logic level
12	R1 Out	Output of Receiver Data from TTL logic level
13	R1 In	Input of Receiver Data from RS232
14	T1 Out	Transmitter Output from RS232
15	GND	Ground
16	Vcc	Power supply voltage

Table 6.1: Pin configurations

MAX 232 INTERFACING WITH MICROCONTROLLER

The Max-232 IC is started by the maxim integrated products in 1987. It is an integrated circuit which converts the signals from the RS232 serial port to the proper signal which are used in the TTL compatible digital logic circuits. The MAX232 can convert the signals like RX, TX, CTS, and RTS and it is a dual driver/receiver.

The driver increases the output voltage levels of TIA232 from a 5 volt supply to 7.5 volts by using the external capacitor and on chip charge pumps. The receiver reduces the input levels of the TIA232 from 25 volts to the standard voltage level, i.e. 5volts of TTL levels and there is a threshold of 1.3 volts and hysteresis of 0.5 volts for the receiver. Further the max232 IC is extended by the four receivers and transmitters simultaneously with eight receivers and transmitters which are MAX238 and MAX248 and there are many combinations of receivers and transmitters.

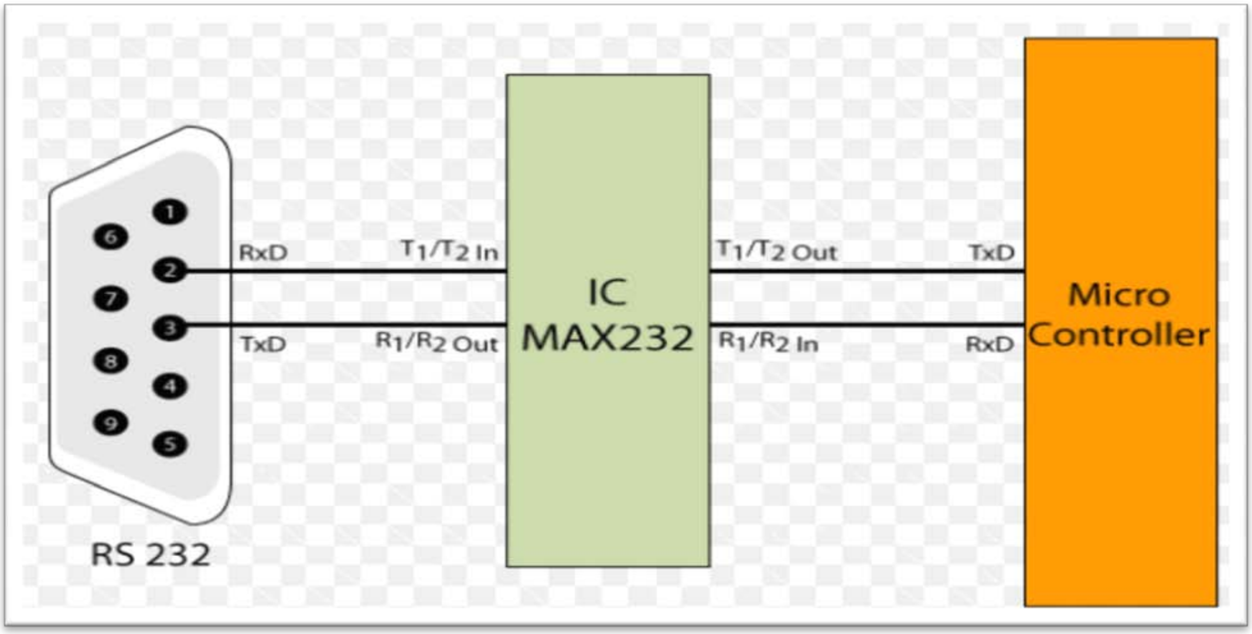


Fig 6.3:Block Diagram of MAX232 Interfacing with microcontroller

FIRMWARE IMPLEMENTATION OF PROJECT

7.1 MPLAB IDE

Microchip has a large suite of software and hardware development tools integrated within one software package called MPLAB Integrated Development Environment (IDE). MPLAB IDE is a free, integrated toolset for the development of embedded applications on Microchip's PIC and ds PIC microcontrollers. It is called an Integrated Development Environment, or IDE, because it provides a single integrated environment to develop code for embedded microcontrollers.

MPLAB IDE runs as a 32-bit application on MS Windows, is easy to use and includes a host of free software components for fast application development and super-charged debugging. MPLAB IDE also serves as a single, unified graphical user interface for additional Microchip and third party software and hardware development tools. Moving between tools is a snap, and upgrading from the free software simulator to hardware debug and programming tools is done in a flash because MPLAB IDE has the same user interface for all tools.

COMPONENTS OF MPLAB IDE

The MPLAB IDE has both built-in components and plug-in modules to configure the system for a variety of software and hardware tools.

Project Manager: The project manager provides integration and communication between the IDE and the language tools.

Editor: The editor is a full-featured programmer's text editor that also serves as a window into the debugger.

Assembler/Linker and Language Tools: The assembler can be used stand-alone to assemble a single file, or can be used with the linker to build a project from separate source files, libraries and recompiled objects. The linker is responsible for positioning the compiled code into memory areas of the target microcontroller.

Debugger: The Microchip debugger allows breakpoints, single stepping, watch windows and all the features of a modern debugger for the MPLAB IDE. It works in conjunction with the editor to reference information from the target being debugged back to the source code.

Execution Engines: There are software simulators in MPLAB IDE for all PICmicro MCU and dsPIC DSC devices. These simulators use the PC to simulate the instructions and some peripheral functions of the PICmicro MCU and dsPIC DSC devices. Optional in-circuit emulators and in-circuit debuggers are also available to test code as it runs in the applications hardware.

Key Features

MPLAB IDE is a Windows® Operating System (OS) based Integrated Development Environment for the PIC MCU families and the ds PIC Digital Signal Controllers.

The MPLAB IDE provides the ability to:

- Create and edit source code using the built-in editor.
- Assemble, compile and link source code.
- Debug the executable logic by watching program flow with the built-in simulator or in real time with in-circuit emulators or in-circuit debuggers.
- Make timing measurements with the simulator or emulator.
- View variables in Watch windows.
- Program firmware into devices with device programmers

Language Tools

Choose MPLAB C Compilers, the highly optimized compilers for the PIC18 series microcontrollers, high performance PIC24 MCUs, dsPIC digital signal controllers and PIC32MX MCUs. Or, use one of the many products from third party language tools vendors. Most integrate into MPLAB IDE to function transparently from the MPLAB project manager, editor and debugger. Microchip's MPLAB C compilers are full-featured, ANSI compliant high-performance tools tightly integrated with MPLAB IDE. Source level debugging allows single stepping through C source code and inspecting variables and structures at critical points in the code.

Microchip MPLAB C Compliers

- MPLAB C Compiler for PIC18 MCUs (C compiler for PIC18)
- MPLAB C Compiler for PIC24 MCUs (C compiler for PIC24)
- MPLAB C Compiler for dsPIC DSCs (C compiler for dsPIC)
- MPLAB C for PIC32 MCUs (C compiler for PIC32)

Third Party Compilers

- HI-TECH
- IAR
- CCS
- Micrium
- Flowcode

Flowcode 4 is one of the worlds most advanced graphical programming languages for microcontrollers. The great advantage of Flowcode is that it allows those with little to no programming experience to create complex electronic systems in minutes. Flowcode supports code generation for the PIC (Flowcode for PIC12, PIC16, and PIC18 series), PIC24 and dsPIC series of microcontrollers. The professional edition includes drivers for a wide range of subsystems like LCDs, keypads, 7-segment displays, ADC, and PWM, as well as communication protocols including I2C, SPI, RS-232, Zigbee, TCP/IP, etc. Flowcode is compatible with Microchips PICkit programmer as well as third party programmers, and is also compatible with the HI-TECH C compiler.

Debuggers and Programmers

MPLAB IDE offers full control over programming bothcode and data, as well as the Configuration bits to set the various operating modes of the target microcontrollers or digital signal controllers.

MPLAB SIM: High-speed software simulator features peripheral simulation, complex stimulus injection and register logging. MPLAB SIM executes your code and can be exercised with stimulus signals from fi les, from mouse clicks and from easily set up waveforms. The contents of variables and special function registers can be logged to a file for analysis.

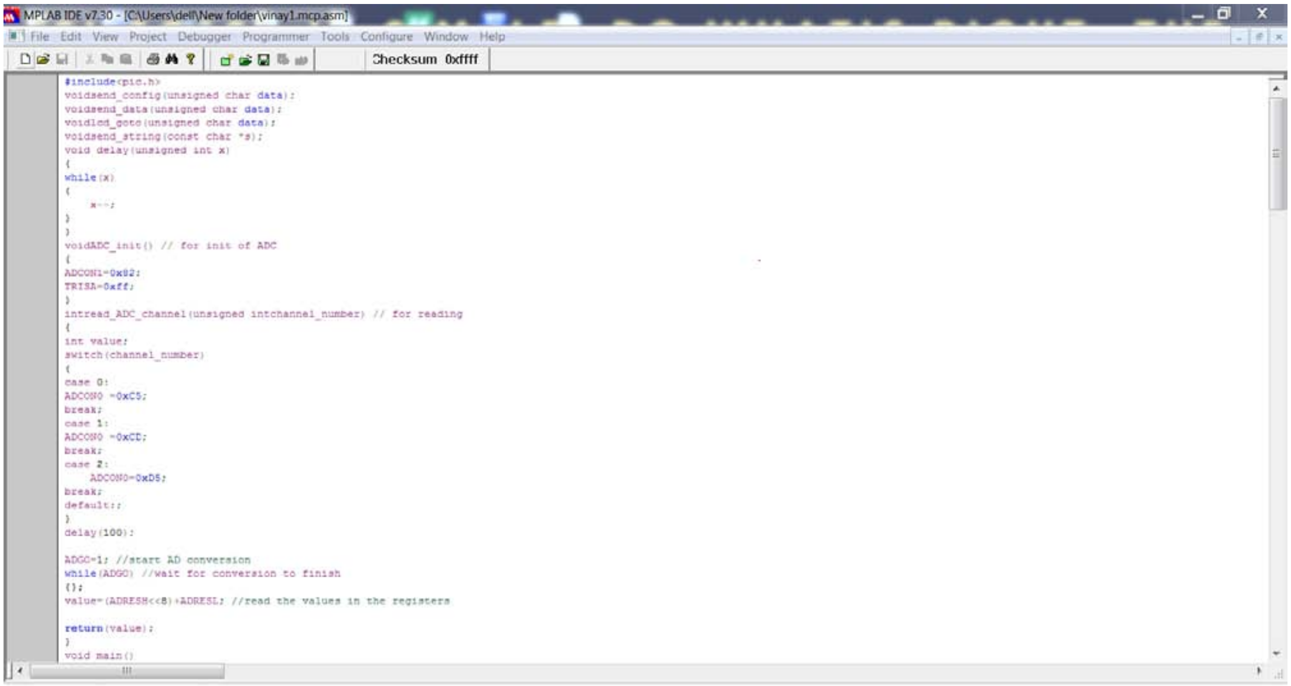
PICKit 3 ICD: The PICkit 3 Debug Express allows debugging and programming of PIC Flash microcontrollers and dsPIC DSCs using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE).

MPLAB ICD 3: MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE). The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB REAL ICE systems (RJ-11). It provides an economic alternative to an emulator. By using some of the on-chip resources, MPLAB ICD 3 can download code into a target microcontroller inserted in the application, set breakpoints, single step and monitor registers and variables.

MPLAB REAL ICE: MPLAB REAL ICE In-Circuit Emulator System is Microchip’s next generation high speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC and dsPIC Flash microcontrollers with the easy-to-use but powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 3 system (RJ11) or with the high speed, noise tolerant, low voltage differential signal (LVDS) interconnection (CAT5). MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE.

CODE EXECUTION



SAMPLE CODE

```
#include<pic.h>

voidsend_config(unsigned char data);

voidsend_data(unsigned char data);

voidlcd_goto(unsigned char data);

voidsend_string(const char *s);

void delay(unsigned int x)

{

while(x)

{

        x--;

}
```

```
}
voidADC_init() // for init of ADC
{
ADCON1=0x82;
TRISA=0xff;
}
intread_ADC_channel(unsigned intchannel_number) // for reading
{
int value;
switch(channel_number)
{
case 0:
ADCON0 =0xC5;
break;
case 1:
ADCON0 =0xCD;
break;
case 2:
        ADCON0=0xD5;
break;
default:;
}
delay(100);
ADGO=1; //start AD conversion
while(ADGO) //wait for conversion to finish
```

```
{};
value=(ADRESH<<8)+ADRESL; //read the values in the registers
return(value);
}
void main()
{
inta,b,c,d=0,d0,d1,d2,d3,f,e=0,d4,d5,d6,d7,d8,d9,d10,d11;
TRISA0=1;
TRISA1=1;
TRISA2=1;
TRISC=0x00;
TRISB=0x00;
TRISE=0x00;
PORTC=0x00;
PORTA=0x00;
PORTB=0x00;
While(1)
{
send_config(0x01);
send_config(0x06);
send_config(0x0c);
send_config(0x38);
send_config(0x80);
delay(100);
ADC_init();
```

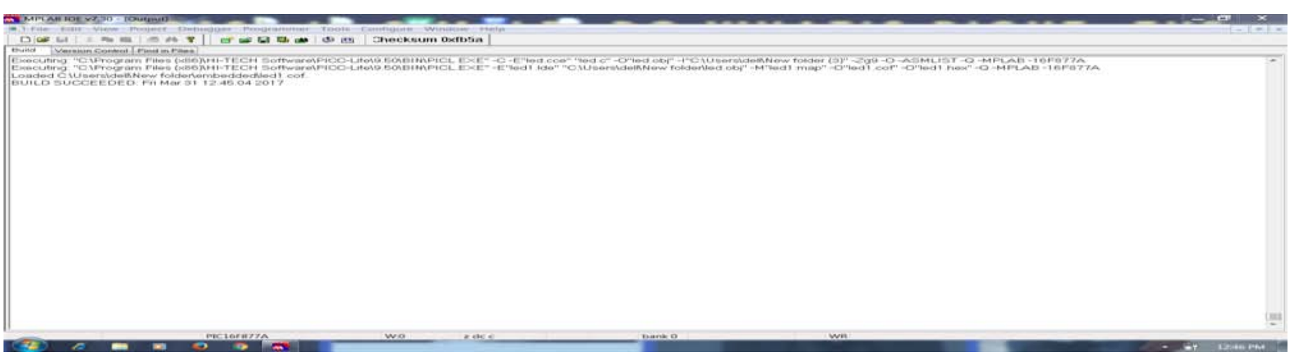
```
c=read_ADC_channel(0);
e=read_ADC_channel(1);
f=read_ADC_channel(2);
c=c/2.05;
e=e/7.11;
f=f/3.08;
d0=c/1000;
d=c%1000;
d1=d/100;
d=d%100;
d2=d/10;
d3=d%10;
d4=e/1000;
e=e%1000;
d5=e/100;
e=e%100;
d6=e/10;
d7=e%10;
d8=f/1000;
f=f%1000;
d9=f/100;
f=f%100;
d10=f/10;
d11=f%10;
if(d2>=4)
```

```
{
	PORTC=0x01;
}
//PORTC=0x00;
d0=d0+0x30;
d1=d1+0x30;
d2=d2+0x30;
d3=d3+0x30;
d4=d4+0x30;
d5=d5+0x30;
d6=d6+0x30;
d7=d7+0x30;
d8=d8+0x30;
d9=d9+0x30;
d10=d10+0x30;
d11=d11+0x30;
lcd_goto(0);
send_config(0x80);
send_string("Temp SEN  GAS");
lcd_goto(20);
send_data(d0);
send_data(d1);
send_data(d2);
send_data(d3);
send_string(" ");
```

```
send_data(d4);
send_data(d5);
send_data(d6);
send_data(d7);
send_string(" ");
send_data(d8);
send_data(d9);
send_data(d10);
send_data(d11);
send_string(" ");
send_config(0x02);
}
}
voidsend_config(char data)
{
PORTE=0X04;
delay(100);
PORTB=data;
delay(100);
RE2=0;
}
voidsend_data(char data)
{
PORTE=0X05;
delay(100);
```

```
PORTB=data;
delay(100);
RE2=0;
}
voidlcd_goto(unsigned char data)
{
if(data<16)
send_config(0x80+data);
else
{
data=data-20;
send_config(0xc0+data);
}
}
voidsend_string(const char*s)
{
while(s&&*s)
send_data(*s++);
}
```

BULDING CODE:



7.2 PROTEUS SOFTWARE

About Proteus

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for **DOS** in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode **SPICE Simulation** was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based autorouting was added in 2002 and 2006 saw another major product update with 3D Board Visualisation. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Feature led product releases are typically biannual, while maintenance based service packs are released as required.

It is a software suite containing **schematic**, **simulation** as well as PCB designing.

- **ISIS** is the software used to draw schematics and simulate the circuits in real time.The simulation allows human access during run time,thus providing real time simulation.
- **ARES** is used for PCB designing.It has the feature of viewing output in 3D view of the designed PCB along with components.
- The designer can also develop 2D drawings for the product.

Features:

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like **oscilloscope**, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, **switches**, **displays**, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

ARES offers PCB designing up to 14 inner layers, with surface mount and through hole packages. It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

Microcontroller Simulation

The micro-controller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables it's used in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design.It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as a training or teaching tool. Support is available for co-simulation of:

- Microchip Technologies PIC10, PIC12, PIC16,PIC18,PIC24,dsPIC33 Microcontrollers.
- Atmel AVR (and Arduino), 8051 and ARM Cortex-M3 Microcontrollers
- NXP 8051, ARM7, ARM Cortex-M0 and ARM Cortex-M3 Microcontrollers.
- Texas Instruments MSP430, PICCOLO DSP and ARM Cortex-M3 Microcontrollers.
- Parallax Basic Stamp, Freescale HC11, 8086 Microcontrollers.

PROTEUS DESIGN OF HARDWARE

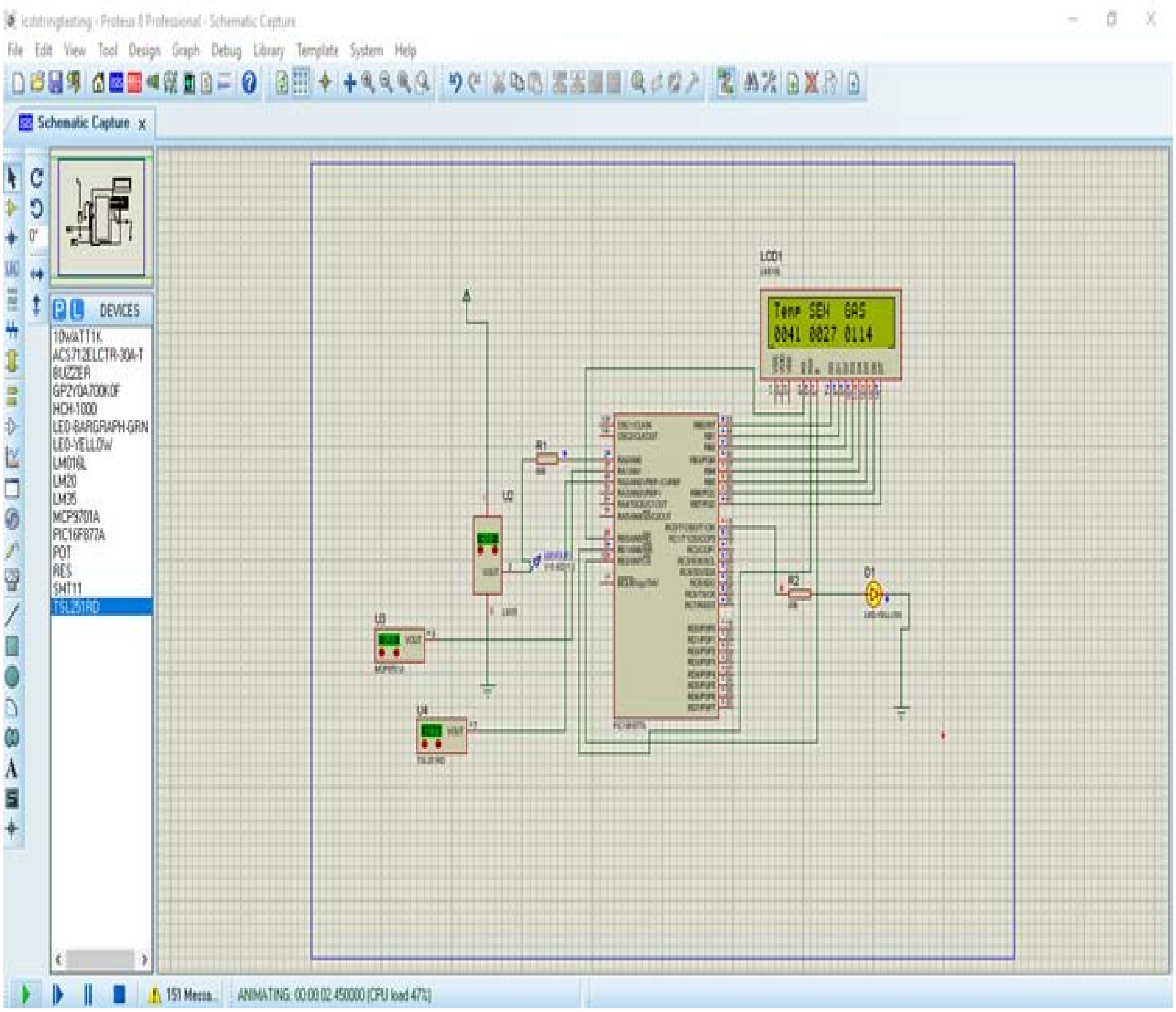


Fig 7.1:Hardware Designing

HARDWARE IMPLEMENTATION:

In this below figure indicates the hardware implementation of the project, In this hardware we can indentifies the sensors ,pic microcontroller and iot module which are interconnected together to become a complete iot project.

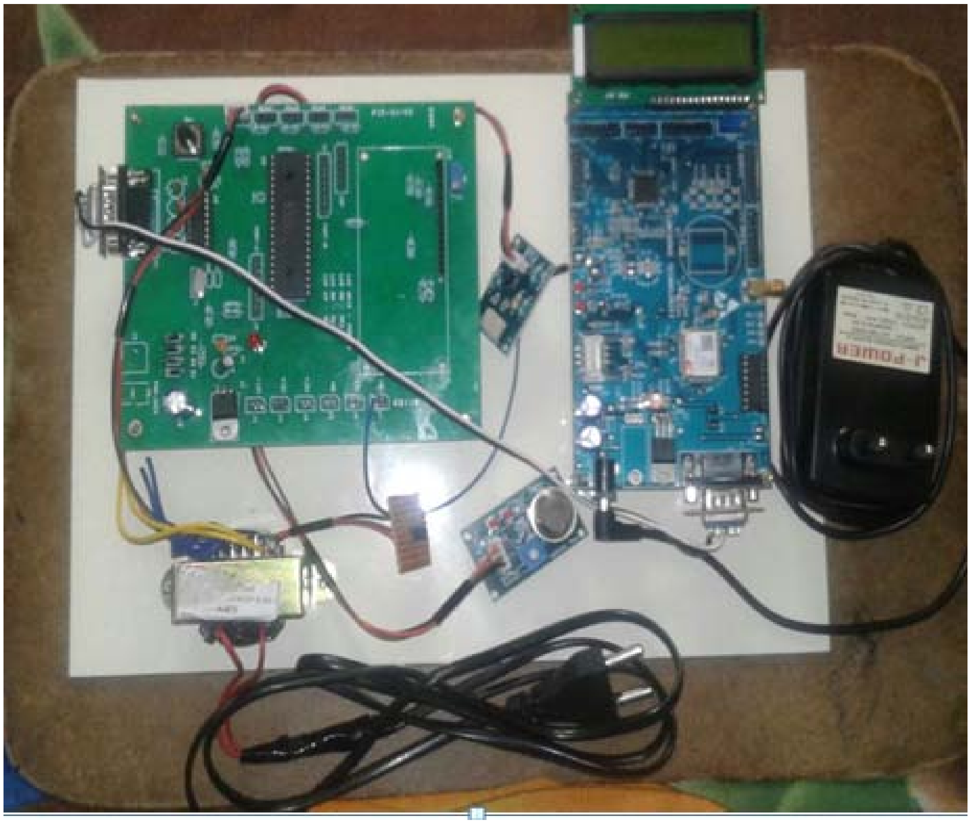


Fig 7.2: Hardware Implementation

CLOUD CONNECTIVITY

AWS IoT

AWS IoT provides secure, bi-directional communication between Internet-connected things (such as sensors, actuators, embedded devices, or smart appliances) and the AWS cloud. This enables you to collect telemetry data from multiple devices and store and analyze the data. You can also create applications that enable your users to control these devices from their phones or tablets.

AWS IoT Components

AWS IoT consists of the following components:

Device gateway

Enables devices to securely and efficiently communicate with AWS IoT.

Message broker

Provides a secure mechanism for things and AWS IoT applications to publish and receive messages from each other. You can use either the MQTT protocol directly or MQTT over WebSocket to publish and subscribe. You can use the HTTP REST interface to publish.

Rules engine

Provides message processing and integration with other AWS services. You can use a SQL-based language to select data from message payloads, process and send the data to other services, such as Amazon S3, Amazon DynamoDB, and AWS Lambda. You can also use the message broker to republish messages to other subscribers.

Security and Identity service

Provides shared responsibility for security in the AWS cloud. Your things must keep their credentials safe in order to securely send data to the message broker. The message broker and rules engine use AWS security features to send data securely to devices or other AWS services.

Thing registry

Some times referred to as the *device r*egistry. Organizes the resources associated with each thing. You register your things and associate up to three custom attributes with each thing. You can also associate certificates and MQTT client IDs with each thing to improve your ability to manage and troubleshoot your things.

Thing shadow

Some times referred to as a device shadow. A JSON document used to store and retrieve current state information for a thing (device, app, and so on).

Thing Shadows service

Provides persistent representations of your things in the AWS cloud. You can publish updated state information to a thing shadow, and your thing can synchronize its state when it connects. Your things can also publish their current state to a thing shadow for use by applications or devices.

How to Get Started with AWS IoT

- To learn more about AWS IoT, see How AWS IoT Works.
- To learn how to connect a thing to AWS IoT, see Getting Started with AWS IoT.

Accessing AWS IoT

AWS IoT provides the following interfaces to create and interact with your things:

AWS Command Line Interface (AWS CLI)—Run commands for AWS IoT on Windows, OS X, and Linux. These commands allow you to create and manage things, certificates, rules, and policies. To get started, see the AWS Command Line Interface User Guide. For more information about the commands for AWS IoT, see iot in the *AWS Command Line Interface Reference*.

AWS IoT API—Build your IoT applications using HTTP or HTTPS requests. These API allow you to programmatically create and manage things, certificates, rules, and policies. For more information about the API actions for AWS IoT, see Actions in the *AWS IoT API Reference*.

AWS SDKs—Build your IoT applications using language-specific APIs. These SDKs wrap the HTTP/HTTPS API and allow you to program in any of the supported languages. For more information, see AWS SDKs and Tools.

AWS IoT Device SDKs—Build applications that run on your devices that send messages to and receive messages from AWS IoT. For more information see, AWS IoT SDKs .

Related Services

AWS IoT integrates directly with the following AWS services:

Amazon Simple Storage Service—Provides scalable storage in the AWS cloud. For more information, see Amazon S3.

Amazon DynamoDB—Provides managed NoSQL databases. For more information, see Amazon DynamoDB.

Amazon Kinesis—Enables real-time processing of streaming data at a massive scale. For more information, see Amazon Kinesis.

AWS Lambda—Runs your code on virtual servers from Amazon EC2 in response to events. For more information, see AWS Lambda.

Amazon Simple Notification Service—Sends or receives notifications. For more information, see Amazon SNS.

Amazon Simple Queue Service—Stores data in a queue to be retrieved by applications. For more information, see Amazon SQS.

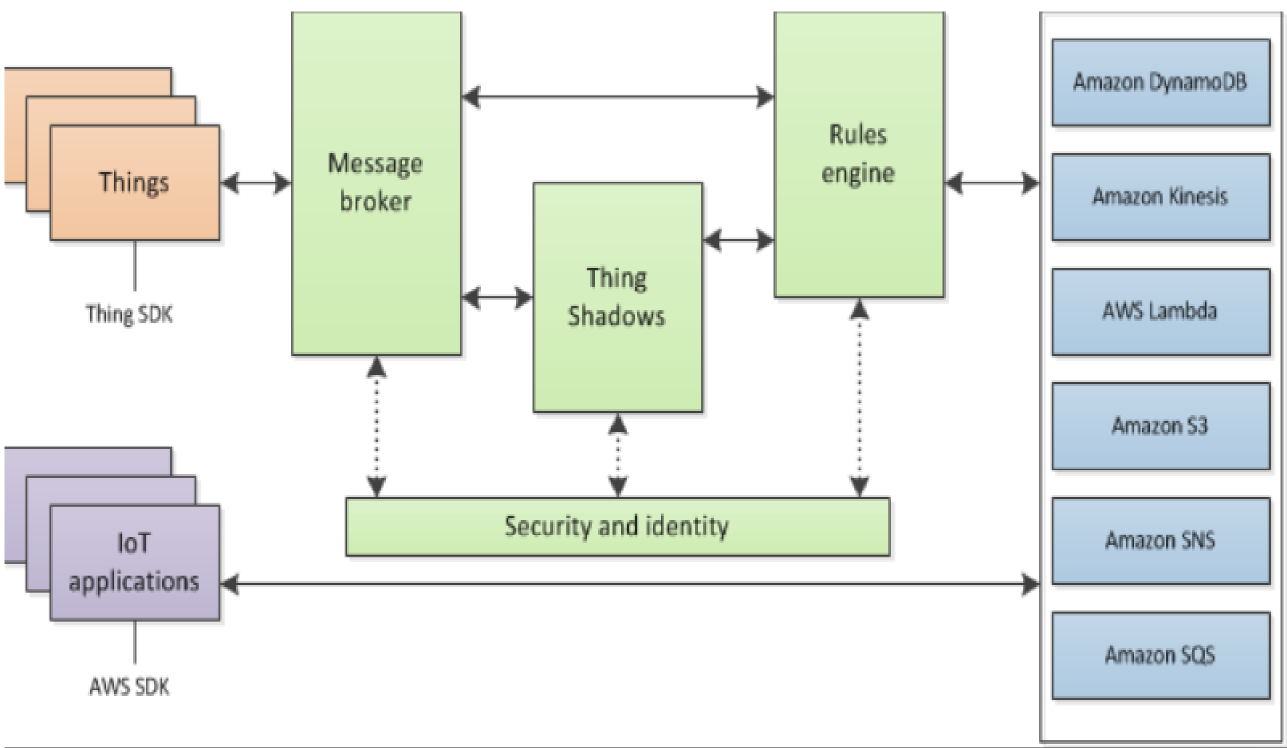


Fig 8.1 : Connecting things to aws IoT cloud

AWS IoT Console

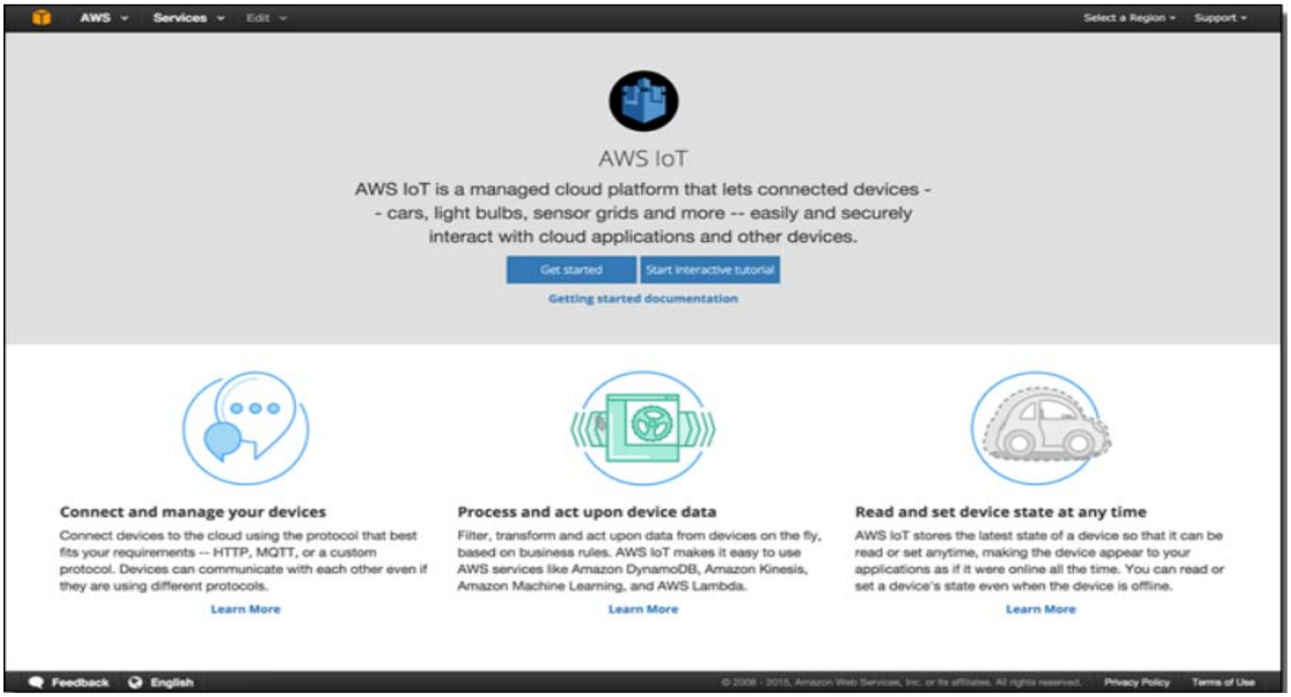


Fig 8.2: IoT console

MONITORING SENSORS DATA IN AWS IOT CLOUD

<u>Website Link:</u> http://www.iotclouddata.com/project/378/iot16view.php

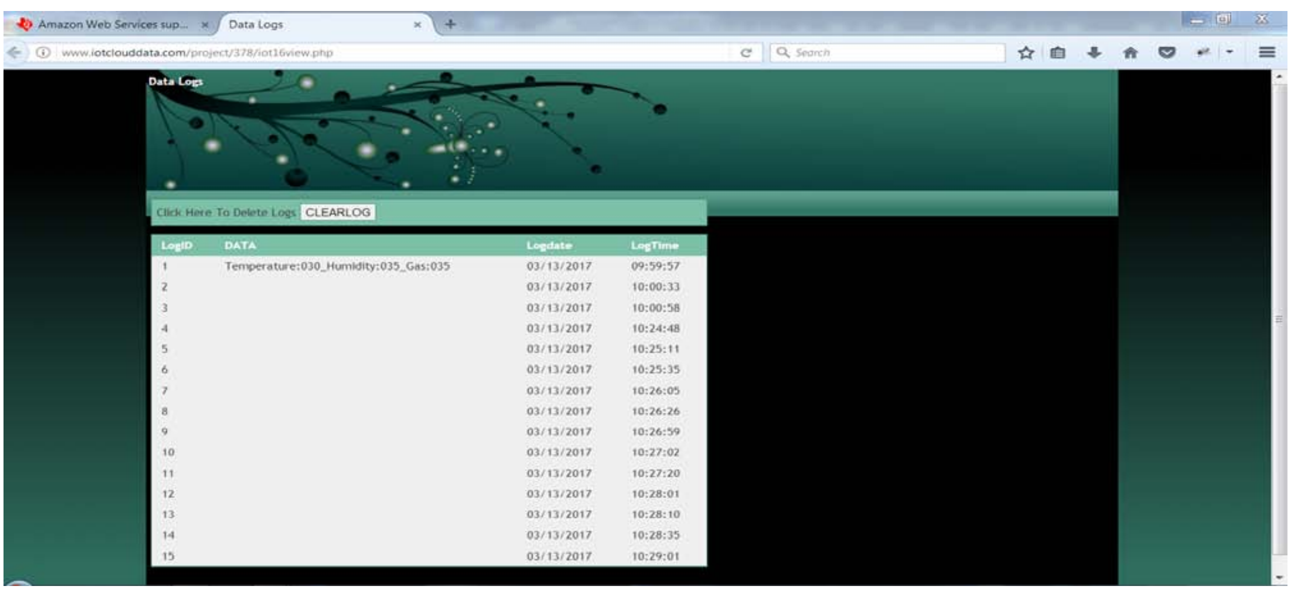


Fig 8.3 : Storing sensors data into the cloud

CONCLUSION AND FUTURE ENHANCEMENT

Internet of Things (IoT) is somehow a leading path to the smart world with ubiquitous computing and networking to ease different tasks around users and provide other tasks, such as easy monitoring of different phenomena surrounding us. Inthe IoT, environmental and items from daily life, termed “things”, “objects”, or “machines” are enhanced with computing and communication technologies. They join the communication framework, meeting a variety of services based onperson-to-person, person-to-machine, machine-to-person and machine-to-machine interactions using wired and wireless communication. These connected machines or objects/things will be the new Internet or network users and will generate data traffic of the emerging IoT. They will perform new services to be carried out by the current or future Interne

The digital era revolutionized human society during the last century. In fact, information digitization processes have led to the design of computers, phones and other machines offering a plethora of applications running on standalone computing machines. Then digitized information transport developed. This has introduced digital communication and networking where machinesare connected to form very large networks and offer remote applications. These machines connected to these networks created the opportunity to deploy different services, either in voice communication, data transfer or entertainment, such as TV, and has led to this digital societ.

From experimental result section, it is clear that proposed system given has proved to be a better way of energy and security management. The main idea of this system is to monitor the energy usage and security of the house in a user friendly and a mobile way so that a user can manage the power management as well as security of their house even when not at the house itself.

BIBLOGRAPHY

1.http://www.microcontrollerboard.com/pic_serial_communication.html#SynandAsynComm

2.http://www.cs.ucr.edu/~eblock/pages/pictools/install.html

3.https://aws.amazon.com/blogs/aws/category/aws-iot/

4.http://www.engpaper.com/iot-2016.htm

5.http://www.edgefxkits.com/blog/max232ic-and-interfacing-needs/

6. Honeywell Automation: https://homesecurity.honeywell.co

7. Rintala, Mikko, Jussi Sormunen, Petri Kuisma, and Matti Rahkala."Automation System Products and Research."(2014).

8.Sandeep Patel, Punit Gupta, Mayank Kumar Goyal, "Low Cost HardwarDesign of a Web Server for Home Automation Systems", Conference onAdvances in Communication and Control Systems(CAC2S), 2013

9. Golzar, M.G. ; AsanPardazan Co. ; Tajozzakerin, H.R., "A New IntelligentRemote Control System for Home Automation and Reduce EnergyConsumption", Mathematical/Analytical Modelling and ComputerSimulation (AMS), 2010, IE