**LOW-COST WIRELESS SENSOR NETWORK IN FIELD OPERATION MONITORING OF AC MOTOR USING IOT**

***ARCOT DEEPAK KUMAR REVANTH(20951A0439)***

***SAMBHANA HARI KRISHNA (20951A0457)***

1. ***RAHUL CHAND(20951A04C7)***

**LOW-COST WIRELESS SENSOR NETWORK IN FIELD OPERATION MONITORING OF AC MOTOR USING IOT**

***A Project Report submitted in partial fulfilment of the requirements***

***for the award of the degree of***

**BACHELOR OF TECHNOLOGY IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

***by***

**SAMBHANA HARI KRISHNA (20951A0457)**

**ARCOT DEEPAK KUMAR REVANTH(20951A0439)**

**G.RAHUL CHAND(20951A04C7)**

Under the Esteemed Guidance of

**Mr. B. Brahmaiah**

**Assistant Professor**

**Department Of Electronics And Communication Engineering**

**INSTITUTE OF AERONAUTICAL ENGINEERING**

**(Autonomous)**

**Dundigal , Hyderabad – 500 043, Telangana**

**January 2024**

© 2024, All rights reserved.

**DECLARATION**

I certify that

1. The work contained in this report is original and has been done by me under the guidance of my supervisor(s).
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have followed the guidelines provided by the Institute in preparing the report.
4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

**Place: Signature of the Student:**

**Date: Roll Number:**

**Arcot Deepak.Kumar Revanth(20951a0439)**

**Sambhana Hari Krishna (20951a0457)**

**G.Rahul Chand(20951a04c7)**

**CERTIFICATE**

This is to certify that the project report entitled **LOW-COST WIRELESS SENSOR NETWORK IN FIELD OPERATION MONITORING OF AC MOTOR USING IOT** submitted by **S. Hari Krishna** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in ELECTRONICS AND COMMUNICATION Engineering is a bonafide record of work carried out by him/her under my/our guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

**Supervisor Head of the Department**

**Date:**

**APPROVAL SHEET**

This project report entitled Low-Cost Wireless Sensor Network In Field Operation Monitoring Of Ac Motor Using Iot by SAMBHANA HARI KRISHNA (20951A0457) is approved for the award of the Degree Bachelor of Technology in Electronics and Communication Engineering.

**Examiners Supervisor(s)**

**Principal**

1. **V. Narasimha Prasad)**

**Date:**

**Place:**

**ACKNOWLEDGEMENT**

I am greatly indebted to my project guide, Mr. B. Brahmaiah , Assistant Professor, Department of Electronics and Communication Engineering, for his invaluable guidance and inspiration which have sustained me to accomplish my work successfully.

I have great pleasure in expressing my sincere thanks to Dr. P Munaswamy, Head of the department, who ignited my hidden potential, built career, in calculated self- confidence, sincerity and discipline within me and gave of success.

It is my pleasure to acknowledge gratefully to the Management and Principal, for the inspiration, valuable suggestions and keen interest during my work.

I am grateful to the teaching and non-teaching faculty members of the Department of Electronics and Communication Engineering, for their encouragement and the facilities provided during my project work.

I appreciate the arduous tasks of my friends, near and dear who injected patience and fortitude to overcome the challenges that have come my way. I perceive this opportunity as a big milestone in my career development.

I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, to attain desired career objectives. Hope to continue cooperation with all of you in the future.

With Gratitude,

SAMBHANA HARI KRISHNA

20951A0457

**ABSTRACT**

This research project focuses on the development of an efficient and automated power survey system tailored for real-time monitoring of AC motors in industrial environments. Leveraging Wi-Fi technology for wireless data transmission, the system incorporates sensors for voltage, current, temperature, and frequency, interfaced with a central microcontroller serving as the main control unit. The Wi-Fi module utilizes radio frequency transmission in the 2.4 GHz band, exhibiting initial speeds ranging from 1 Mbps to 2 Mbps, with an operational range of 40-300 feet. The microcontroller processes inputs from all sensors, executing an embedded 'C' language program to analyze and transmit the data to the Thingspeak cloud. The collected parameters are subsequently accessible on mobile devices, providing users with real-time insights. Additionally, the system is equipped with a buzzer to alert operators in case of abnormal parameter values, enhancing its utility for prompt response to potential issues. This research contributes to the advancement of smart monitoring systems for industrial applications, showcasing the integration of wireless technologies and intelligent programming in ensuring the reliability and efficiency of AC motor operations.

## CONTENTS

|  |  |  |
| --- | --- | --- |
| Title Page |  | I |
| Certificate by the Supervisor | | II |
| Declaration |  | III |
| Acknowledgement | | IV |
| Abstract |  | V |
| Contents |  | VI |
| List of Figures | | VII |
| List of Tables | | VII |
| List of Symbols | | IX |
| Abbreviations | | X |
| Abstract |  | XI |
| Chapter 1 | Introduction | 1 |
|  | 1.1 Section | 1 |
|  | 1.2 Section | … |
|  | 1.3 Section | … |
| Chapter 2 | Review of Relevant Literature | … |
| Chapter 3 | Methodology | … |
| Chapter 4 | Results and Discussions | … |
| Chapter 5 | Conclusions and Future Scope of Study | … |
| References |  | … |
| List of Publications | | … |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page** |
| 3.1 | Microscopic images of nano particles | 11 |
| 3.2 | Injection system specifications | 13 |
| 4.1 | Atomic percentage of elements on the stainless steel pin | 18 |
| 4.2 | Image of Zirconia doped cerium oxide nano particles | 19 |
| 4.3 | Capillary rise method | 21 |
| 4.7 | Capillary rise measurement by travelling microscope | 25 |
| 4.8 | Set up for contact angle measurement | 27 |
| 5.3 | Schematic of DLS System | 30 |
| 5.4 | Variation of density with volume %ge of surfactant | 32 |

**Chapter 1**

**INTRODUCTION**

In the realm of industrial automation, the real-time monitoring of AC motors is pivotal for ensuring operational efficiency and preventing potential faults. The system integrates Wi-Fi technology, a microcontroller, and an array of sensors to collect, process, and transmit electrical parameters in real-time. The utilization of Wi-Fi technology serves as a cornerstone for wireless data transmission, providing the project with the means to transmit data seamlessly through the air. This technological choice forms the backbone of the project's wireless communication infrastructure, enabling the seamless integration of data into the broader industrial monitoring framework. At the core of the proposed system is a microcontroller, serving as the central control unit. This intelligent device interfaces with various sensors, including voltage, current, temperature, and a frequency counter, all essential for monitoring the health and performance of AC motors. The key highlight of the system lies in its ability to alert operators in case of abnormal parameter values, adding a layer of proactive monitoring and fault prevention. To execute these functions, the microcontroller is loaded with an intelligent program, written in embedded 'C' language, showcasing the synergy of hardware and software in achieving the project's objectives. The major features of this project underscore its relevance and innovation in the domain of industrial monitoring.

Firstly, the system facilitates automatic parameter monitoring of AC motors, streamlining the data collection process. Secondly, it employs wireless technology for data reception and display, eliminating the need for cumbersome wired setups and enhancing the system's adaptability to diverse industrial environments. Thirdly, the incorporation of alert mechanisms for abnormal parameter values positions the system as a proactive solution for fault prevention. Lastly, the integration of Thingspeak technology for wireless data monitoring ensures a scalable and accessible platform for industrial operations. In summary, this research project pioneers a low-cost wireless sensor network for the field operation monitoring of AC motors, underscoring the transformative potential of IoT in industrial settings.

**Chapter 2**

**REVIEW OF LITERATURE**

**1. "Critical Survey on IoT Based Monitoring and Control of Induction Motor" by Sudharani Potturi, Dr. Rajashekar, P. Mandi (2022):**

This paper explores the implementation of IoT for monitoring and controlling induction motors in applications like electric vehicles, industries, and agriculture. However, the study is marked by limitations, including limited experimental validation, a lack of mature design tools, outdated references, and an incomplete comparison or evaluation of proposed solutions.

**2. "IoT-Based Wireless Induction Motor Monitoring" by Mehmet Şen, Basri Kul (2017):**

The research likely involves the development and implementation of a wireless IoT system for monitoring induction motors. While addressing sensor integration and data transmission, the study lacks detailed experimental validation, a comprehensive comparison with existing solutions, and potential insights into scalability or practical implementation issues.

**3. "IoT-Based Traction Motor Drive Condition Monitoring in Electric Vehicles: Part 1" by Jakkrit Kunthong et al. (2017):**

This study focuses on developing an IoT-based system for monitoring the condition of traction motors in electric vehicles. However, it suffers from limitations such as limited validation under real-world electric vehicle conditions, a lack of a comprehensive comparison with alternative monitoring systems, and potential challenges related to scalability or practical implementation.

**4. "Real-Time Condition Monitoring System for Industrial Motors" by S. S. Goundar et al. (2015):**

This study likely involves the development of a real-time monitoring system for industrial motors, utilizing sensor integration and advanced algorithms. Despite its potential, the research exhibits limitations, including limited validation across diverse industrial motor conditions and insufficient information on scalability and practical implementation challenges in industrial settings.

**5. "IoT Platform for Condition Monitoring of Industrial Motors" by A. Ajitha et al. (2017):**

This research aims to develop an IoT platform for monitoring industrial motor conditions. However, it faces limitations such as limited validation with diverse industrial motor conditions, a lack of comprehensive comparisons with alternative monitoring platforms, and potential challenges related to scalability and real-world implementation.

**6. "Submersible Pump Control for Agriculture Irrigation Using IoT Technique" by S. Vaseela et al. (2017):**

The study likely involves the development of an IoT-based system for controlling submersible pumps in agricultural irrigation. Despite its potential applications, the research faces limitations including limited validation with diverse agricultural conditions and a lack of comprehensive comparisons with alternative irrigation control systems. Practical implementation challenges in agricultural settings are also not extensively discussed.

**7. "Design of IoT-Based Intelligent AC Drive System Using Space Vector Algorithm" by Amey C. Thombre et al. (2017):**

This research focuses on designing an intelligent AC drive system using the Space Vector Algorithm with IoT integration. However, the study has limitations, including limited validation with diverse AC drive system conditions, a lack of comprehensive comparisons with alternative intelligent drive systems, and potential challenges related to scalability and real-world implementation across various applications.

**8. "Motor Control Used To Be Boring" by Alexander Tessarolo (2016):**

This work discusses advancements or innovations in motor control without providing detailed practical implementation details or experimental validation. The study also has a limited focus on specific applications or use cases for the discussed motor control advancements, with potential challenges related to scalability or real-world integration.

**9. "Online Parameter Monitoring of Induction Motor Using Wireless Network" by Geetha P, V. Saravanan (2012):**

The research likely involves the development and implementation of a system for monitoring induction motor parameters using a wireless network. Despite its potential, the study faces limitations such as limited validation with diverse induction motor conditions, a lack of comprehensive comparisons with alternative monitoring systems, and potential challenges related to scalability and real-world implementation in industrial settings.

**10. "IoT Based Health Monitoring System for Electrical Motors" by Naveed Khan et al. (2019):**

This research involves designing an IoT-based system for health monitoring of electrical motors, measuring parameters like vibration, current, and temperature. However, the study has limitations including a lack of discussion on specific fault identification algorithms, scalability challenges, and limited validation across various motor types and operational scenarios.

**11. "IoT Application for Fault Diagnosis and Prediction in Elevators" by Omid Salim, E. A. (2017):**

The methodology likely includes sensor integration within elevators to monitor various parameters for proactive fault diagnosis and prediction using IoT. Despite its focus, the study has limitations, such as a lack of detailed information on sensor types and placement, scalability challenges, and limited validation across diverse elevator models and fault scenarios.

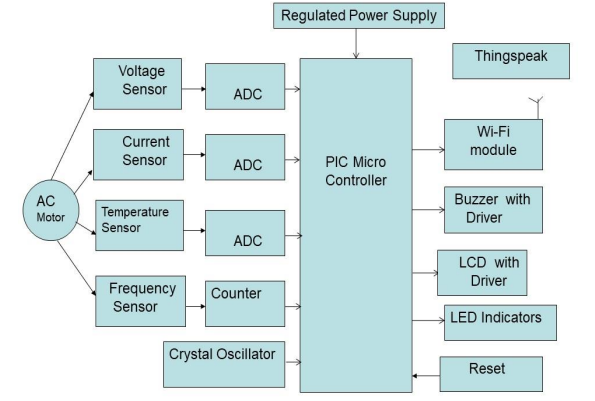
**CHAPTER 3**

**METHODOLOGY**

**HARDWARE DESCRIPTION**

**3.1 Introduction:**

In this chapter the block diagram of the project and design aspect of independent modules are considered. Block diagram is shown in fig: 3.1:



**FIG 3.1: Block diagram of Low-cost wireless sensor network in field operation monitoring of AC motor using IOT**

**The main blocks of this project are:**

1. Regulated power supply.

2. PIC Microcontroller.

3. ESP8266 Wi-Fi module.

4. Voltage sensor.

5. Current Sensor.

6. Frequency Sensor.

7. Temperature Sensor.

8. Reset.

9. Crystal oscillators.

10. LED indicators.

11. Buzzer with driver.

12. LCD display.

**3.2 Microcontroller PIC16F73**



Fig: 3.2 Microcontrollers

**3.2.1 Introduction**

The PIC16F73 CMOS FLASH-based 8-bit microcontroller is upward compatible with the PIC16C73B/74B/76/77, PIC16F873/874/876/877devices. It features 200 ns instruction execution, self programming, an ICD, 2 Comparators, 8 channels of 8-bit Analog-to-Digital (A/D) converter, 2 capture/compare/PWM functions, a synchronous serial port that can be configured as either 3-wire SPI or 2-wire I2C bus, a USART, and a Parallel Slave Port.

**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)

• Pin out compatible to the PIC16C73B/74B/76/77

• Pin out compatible to the PIC16F873/874/876/877

• Interrupt capability (up to 12 sources)

• Eight level deep hardware stack

**Special Microcontroller Features**

• Power-up Timer (PWRT) and oscillator Start-up Timer (OST)

• Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation

• Programmable code protection

• Power saving SLEEP mode

• In-Circuit Serial Programming (ICSP ) via two Pins

**Peripheral Features**

• Timer0: 8-bit timer/counter with 8-bit prescaler

• Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock

• Timer2: 8-bit timer/counter with 8-bit period register, prescaler 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

• 8-bit, up to 8-channel Analog-to-Digital converter

• Synchronous Serial Port (SSP) with SPI (Master mode) and I2C (Slave)

• Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)

**Analog Comparator module**

• 2 analog comparators

• Programmable on-chip voltage reference module

• Programmable input multiplexing from device inputs and internal VREF

• Comparator outputs are externally accessible

**CMOS Technology:**

Low power, high speed CMOS FLASH technology

• Fully static design

• Wide operating voltage range: 2.0V to 5.5V

• High Sink/Source Current: 25 mA

• Industrial temperature range

• Low power consumption: - < 2 mA typical @ 5V, 4 MHz – 20 μA typical @ 3V, 32 kHz - < 1 μA typical standby current

Following are the major blocks of PIC Microcontroller.

**Program memory (FLASH)** is used for storing a written program. Since memory made in FLASH technology can be programmed and cleared more than once, it makes this microcontroller suitable for device development.

**EEPROM** - data memory that needs to be saved when there is no supply. It is usually used for storing important data that must not be lost if power supply suddenly stops. For instance, one such data is an assigned temperature in temperature regulators. If during a loss of power supply this data was lost, we would have to make the adjustment once again upon return of supply. Thus our device looses on self-reliance.

**RAM** - Data memory used by a program during its execution. In RAM are stored all inter-results or temporary data during run-time.

**PORTS** are physical connections between the microcontroller and the outside world. PIC16F73 has 22 I/O.

**FREE-RUN TIMER** is an 8-bit register inside a microcontroller that works independently of the program. On every fourth clock of the oscillator it increments its value until it reaches the maximum (255), and then it starts counting over again from zero. As we know the exact timing between each two increments of the timer contents, timer can be used for measuring time which is very useful with some devices.

**CENTRAL PROCESSING UNIT** has a role of connective element between other blocks in the microcontroller. It coordinates the work of other blocks and executes the user program.



Harvard vs von Nueman Block Architectures

**CISC, RISC**

It has already been said that PIC16F73 has RISC architecture. This term is often found in computer literature, and it needs to be explained here in more detail. Harvard architecture is a newer concept than von-Neumann. It rose out of the need to speed up the work of a microcontroller. In Harvard architecture, data bus and address bus are separate. Thus a greater flow of data is possible through the central processing unit, and of course, a greater speed of work. Separating a program from data memory makes it further possible for instructions not to have to be 8-bit words. PIC16F73 uses 14 bits for instructions which allows for all instructions to be one word instructions. It is also typical for Harvard architecture to have fewer instructions than von-Neumann's, and to have instructions usually executed in one cycle.

Microcontrollers with Harvard architecture are also called "RISC microcontrollers". RISC stands for Reduced Instruction Set Computer. Microcontrollers with von Neumann's architecture are called 'CISC microcontrollers'. Title CISC stands for Complex Instruction Set Computer.

Since PIC16F73 is a RISC microcontroller, that means that it has a reduced set of instructions, more precisely 35 instructions. (Ex. Intel's and Motorola's microcontrollers have over hundred instructions) All of these instructions are executed in one cycle except for jump and branch instructions. According to what its maker says, PIC16F73 usually reaches results of 2:1 in code compression and 4:1 in speed in relation to other 8-bit microcontrollers in its class.

**Crystal oscillator:**

The crystal oscillator speed that can be connected to the PIC microcontroller range from DC to 20Mhz. Using the CCS C compiler normally 20Mhz oscillator will be used and the price is very cheap. The 20 MHz crystal oscillator should be connected with about 22pF capacitor. Please refer to my circuit schematic. There are 5 input/output ports on PIC microcontroller namely port A, port B, port C, port D and port E. Each port has different function. Most of them can be used as I/O port.

**Applications**

PIC16F73 perfectly fits 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 PIC16F73 applicable even in areas where microcontrollers had not previously been considered (example: timer functions, interface replacement in larger systems, coprocessor applications, etc.).

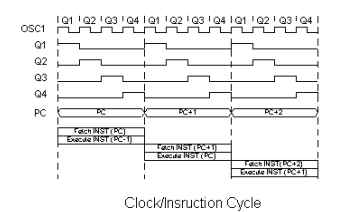
In System Programmability of this chip (along with using only two pins in data transfer) makes possible the flexibility of a product, after assembling and testing have been completed. This capability can be used to create assembly-line production, to store calibration data available only after final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

**Clock / instruction cycle**

Clock is microcontroller's main starter, and is obtained from an external component called an "oscillator". If we want to compare a microcontroller with a time clock, our "clock" would then be a ticking sound we hear from the time clock. In that case, oscillator could be compared to a spring that is wound so time clock can run. Also, force used to wind the time clock can be compared to an electrical supply

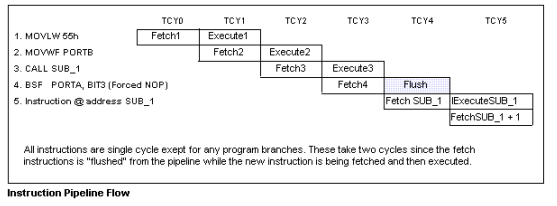
Clock from the oscillator enters a microcontroller via OSC1 pin where internal circuit of a microcontroller divides the clock into four even clocks Q1, Q2, Q3, and Q4 which do not overlap. These four clocks make up one instruction cycle (also called machine cycle) during which one instruction is executed.

Execution of instruction starts by calling an instruction that is next in string. Instruction is called from program memory on every Q1 and is written in instruction register on Q4. Decoding and execution of instruction are done between the next Q1 and Q4 cycles. On the following diagram we can see the relationship between instruction cycle and clock of the oscillator (OSC1) as well as that of internal clocks Q1-Q4. Program counter (PC) holds information about the address of the next instruction.



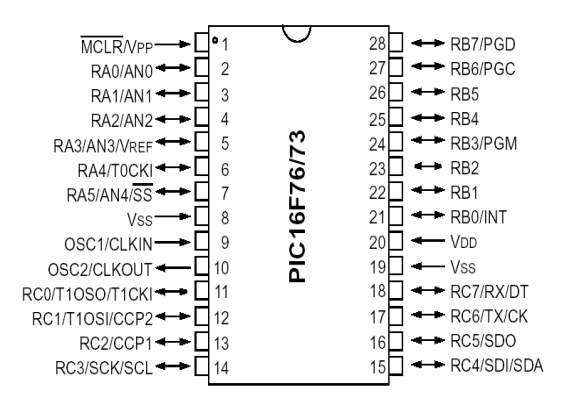
**Pipelining**

Instruction cycle consists of cycles Q1, Q2, Q3 and Q4. Cycles of calling and executing instructions are connected in such a way that in order to make a call, one instruction cycle is needed, and one more is needed for decoding and execution. However, due to pipelining, each instruction is effectively executed in one cycle. If instruction causes a change on program counter, and PC doesn't point to the following but to some other address (which can be the case with jumps or with calling subprograms), two cycles are needed for executing an instruction. This is so because instruction must be processed again, but this time from the right address. Cycle of calling begins with Q1 clock, by writing into instruction register (IR). Decoding and executing begins with Q2, Q3 and Q4 clocks.

****

**Pin description**

PIC16F73 has a total of 28 pins. It is most frequently found in a DIP28 type of case but can also be found in SMD case which is smaller from a DIP. DIP is an abbreviation for Dual In Package. SMD is an abbreviation for Surface Mount Devices suggesting that holes for pins to go through when mounting aren't necessary in soldering this type of a component.

****

Pins on PIC16F73 microcontroller have the following meaning:  
  
There are 28 pins on PIC16F73. Most of them can be used as an IO pin. Others are already for specific functions. These are the pin functions.  
1. MCLR – to reset the PIC  
2. RA0 – port A pin 0  
3. RA1 – port A pin 1  
4. RA2 – port A pin 2  
5. RA3 – port A pin 3  
6. RA4 – port A pin 4  
7. RA5 – port A pin 5  
8. VSS – ground  
9. OSC1 – connect to oscillator  
10. OSC2 – connect to oscillator  
11. RC0 – port C pin 0 VDD – power supply  
12. RC1 – port C pin 1  
13. RC2 – port C pin 2  
14. RC3 – port C pin 3  
15. RC4 - port C pin 4  
16. RC5 - port C pin 5  
17. RC6 - port C pin 6  
18. RC7 - port C pin 7  
19. VSS - ground  
20. VDD – power supply  
21. RB0 - port B pin 0  
22. RB1 - port B pin 1  
23. RB2 - port B pin 2  
24. RB3 - port B pin 3  
25. RB4 - port B pin 4  
26. RB5 - port B pin 5  
27. RB6 - port B pin 6  
28. RB7 - port B pin 7  
  
By utilizing all of this pin so many application can be done such as:  
1. LCD – connect to Port B pin.  
2. LED – connect to any pin declared as output.  
3. Relay and Motor - connect to any pin declared as output.  
4. External EEPROM – connect to I2C interface pin – RC3 and RC4 (SCL and SDA)   
5. LDR, Potentiometer and sensor – connect to analogue input pin such as RA0.  
6. GSM modem dial up modem – connect to RC6 and RC7 – the serial communication interface using RS232 protocol..  
  
For more detail function for each specific pin please refer to the device datasheet from Microchip.

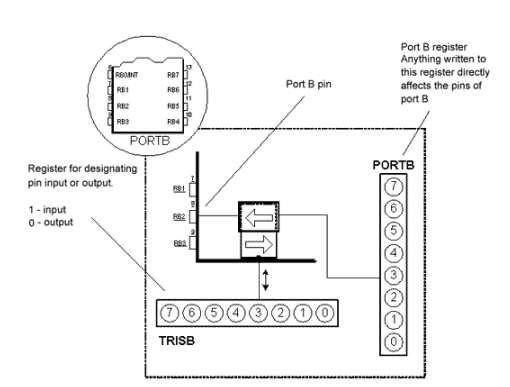
**Ports**

Term "port" refers to a group of pins on a microcontroller which can be accessed simultaneously, or on which we can set the desired combination of zeros and ones, or read from them an existing status. Physically, port is a register inside a microcontroller which is connected by wires to the pins of a microcontroller. Ports represent physical connection of Central Processing Unit with an outside world. Microcontroller uses them in order to monitor or control other components or devices. Due to functionality, some pins have twofold roles like PA4/TOCKI for instance, which is in the same time the fourth bit of port A and an external input for free-run counter. Selection of one of these two pin functions is done in one of the configuration registers. An illustration of this is the fifth bit T0CS in OPTION register. By selecting one of the functions the other one is disabled.

All port pins can be designated as input or output, according to the needs of a device that's being developed. In order to define a pin as input or output pin, the right combination of zeros and ones must be written in TRIS register. If the appropriate bit of TRIS register contains logical "1", then that pin is an input pin, and if the opposite is true, it's an output pin. Every port has its proper TRIS register. Thus, port A has TRISA, and port B has TRISB. Pin direction can be changed during the course of work which is particularly fitting for one-line communication where data flow constantly changes direction. PORTA and PORTB state registers are located in bank 0, while TRISA and TRISB pin direction registers are located in bank.

**PORTB&TRISB**  
PORTB has adjoined 8 pins. The appropriate register for data direction is TRISB. Setting a bit in TRISB register defines the corresponding port pin as input, and resetting a bit in TRISB register defines the corresponding port pin as output.

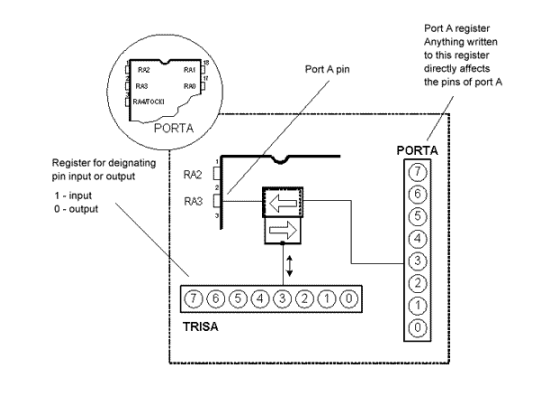
Each PORTB pin has a weak internal pull-up resistor (resistor which defines a line to logic one) which can be activated by resetting the seventh bit RBPU in OPTION register. These 'pull-up' resistors are automatically being turned off when port pin is configured as an output. When a microcontroller is started, pull-ups are disabled.

****

Four pins PORTB, RB7:RB4 can cause an interrupt which occurs when their status changes from logical one into logical zero and opposite. Only pins configured as input can cause this interrupt to occur (if any RB7:RB4 pin is configured as an output, an interrupt won't be generated at the change of status.) This interrupt option along with internal pull-up resistors makes it easier to solve common problems we find in practice like for instance that of matrix keyboard. If rows on the keyboard are connected to these pins, each push on a key will then cause an interrupt. A microcontroller will determine which key is at hand while processing an interrupt It is not recommended to refer to port B at the same time that interrupt is being processed.

**PORTA and TRISA**

PORTA has 5 adjoining pins. The corresponding register for data direction is TRISA at address 85h. Like with port B, setting a bit in TRISA register defines also the corresponding port pin as input, and clearing a bit in TRISA register defines the corresponding port pin as output.



It is important to note that PORTA pin RA4 can be input only. On that pin is also situated an external input for timer TMR0. Whether RA4 will be a standard input or an input for a counter depends on T0CS bit (*TMR0 Clock Source Select bit*). This pin enables the timer TMR0 to increment either from internal oscillator or via external impulses on RA4/T0CKI pin.

**Memory organization**

PIC16F73 has two separate memory blocks, one for data and the other for program. EEPROM memory with GPR and SFR registers in RAM memory make up the data block, while FLASH memory makes up the program block.

**Program memory**

Program memory has been carried out in FLASH technology which makes it possible to program a microcontroller many times before it's installed into a device, and even after its installment if eventual changes in program or process parameters should occur. The size of program memory is 1024 locations with 14 bits width where locations zero and four are reserved for reset and Interrupt vector.

**Datamemory**  
Data memory consists of EEPROM and RAM memories. EEPROM memory consists of 256 eight bit locations whose contents are not lost during loosing of power supply. EEPROM is not directly addressable, but is accessed indirectly through EEADR and EEDATA registers. As EEPROM memory usually serves for storing important parameters (for example, of a given temperature in temperature regulators), there is a strict procedure for writing in EEPROM which must be followed in order to avoid accidental writing. RAM memory for data occupies space on a memory map from location 0x0C to 0x4F which comes to 68 locations. Locations of RAM memory are also called GPR registers which is an abbreviation for *General Purpose Registers*. GPR registers can be accessed regardless of which bank is selected at the moment.

**3.3 REGULATED POWER SUPPLY:**

**3.3.1 Introduction:**

Power supply is a supply of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

A power supply may include a power distribution system as well as primary or secondary sources of energy such as

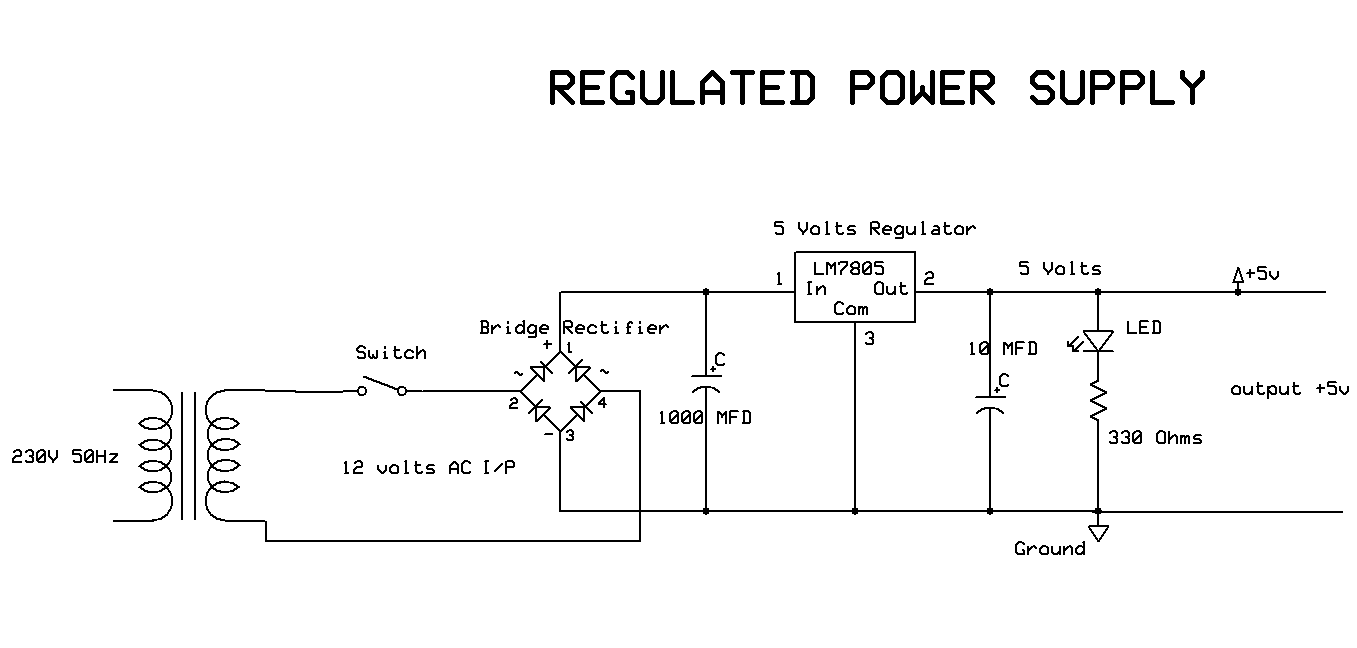
* Conversion of one form of electrical power to another desired form and voltage, typically involving converting AC line voltage to a well-regulated lower-voltage DC for electronic devices. Low voltage, low power DC power supply units are commonly integrated with the devices they supply, such as computers and household electronics.
* Batteries.
* Chemical fuel cells and other forms of energy storage systems.
* Solar power.
* Generators or alternators.

**3.3.2 Block Diagram:**

****

**Fig 3.3.2 Regulated Power Supply**

The basic circuit diagram of a regulated power supply (DC O/P) with led connected as load is shown in fig: 3.3.3.



**Fig 3.3.3 Circuit diagram of Regulated Power Supply with Led connection**

The components mainly used in above figure are

* 230V AC MAINS
* TRANSFORMER
* BRIDGE RECTIFIER(DIODES)
* CAPACITOR
* VOLTAGE REGULATOR(IC 7805)
* RESISTOR
* LED(LIGHT EMITTING DIODE)

The detailed explanation of each and every component mentioned above is as follows:

**Transformation:** The process of transforming energy from one device to another is called transformation. For transforming energy we use transformers.

**Transformers:**

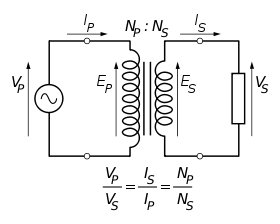
A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors without changing its frequency. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction.

If a load is connected to the secondary, an electric current will flow in the secondary winding and electrical energy will be transferred from the primary circuit through the transformer to the load. This field is made up from lines of force and has the same shape as a bar magnet.

If the current is increased, the lines of force move outwards from the coil. If the current is reduced, the lines of force move inwards.

If another coil is placed adjacent to the first coil then, as the field moves out or in, the moving lines of force will "cut" the turns of the second coil. As it does this, a voltage is induced in the second coil. With the 50 Hz AC mains supply, this will happen 50 times a second. This is called MUTUAL INDUCTION and forms the basis of the transformer.

The input coil is called the PRIMARY WINDING; the output coil is the SECONDARY WINDING. Fig: 3.3.4 shows step-down transformer.

****

**Fig 3.3.4: Step-Down Transformer**

The voltage induced in the secondary is determined by the TURNS RATIO.



For example, if the secondary has half the primary turns; the secondary will have half the primary voltage.

Another example is if the primary has 5000 turns and the secondary has 500 turns, then the turn’s ratio is 10:1.

If the primary voltage is 240 volts then the secondary voltage will be x 10 smaller = 24 volts. Assuming a perfect transformer, the power provided by the primary must equal the power taken by a load on the secondary. If a 24-watt lamp is connected across a 24 volt secondary, then the primary must supply 24 watts.

To aid magnetic coupling between primary and secondary, the coils are wound on a metal CORE. Since the primary would induce power, called EDDY CURRENTS, into this core, the core is LAMINATED. This means that it is made up from metal sheets insulated from each other. Transformers to work at higher frequencies have an iron dust core or no core at all.

Note that the transformer only works on AC, which has a constantly changing current and moving field. DC has a steady current and therefore a steady field and there would be no induction.

Some transformers have an electrostatic screen between primary and secondary. This is to prevent some types of interference being fed from the equipment down into the mains supply, or in the other direction. Transformers are sometimes used for IMPEDANCE MATCHING.

We can use the transformers as step up or step down.

**Step Up transformer:**

In case of step up transformer, primary windings are every less compared to secondary winding.

Because of having more turns secondary winding accepts more energy, and it releases more voltage at the output side.

**Step down transformer:**

Incase of step down transformer, Primary winding induces more flux than the secondary winding, and secondary winding is having less number of turns because of that it accepts less number of flux, and releases less amount of voltage.

**Rectification:**

The process of converting an alternating current to a pulsating direct current is called as rectification. For rectification purpose we use rectifiers.

**Rectifiers:**

A rectifier is an electrical device that converts alternating current (AC) to direct current (DC), a process known as rectification. Rectifiers have many uses including as components of power supplies and as detectors of radio signals. Rectifiers may be made of solid-state diodes, vacuum tube diodes, mercury arc valves, and other components.

A device that it can perform the opposite function (converting DC to AC) is known as an inverter.

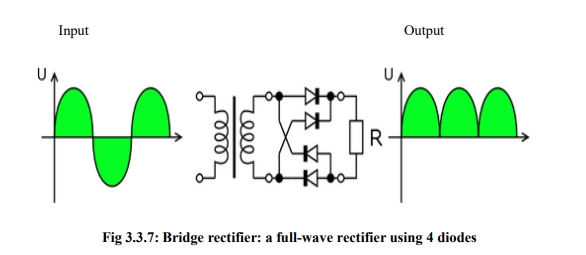
When only one diode is used to rectify AC (by blocking the negative or positive portion of the waveform), the difference between the term diode and the term rectifier is merely one of usage, i.e., the term rectifier describes a diode that is being used to convert AC to DC. Almost all rectifiers comprise a number of diodes in a specific arrangement for more efficiently converting AC to DC than is possible with only one diode. Before the development of silicon semiconductor rectifiers, vacuum tube diodes and copper (I) oxide or selenium rectifier stacks were used.

**Bridge full wave rectifier:**

The Bridge rectifier circuit is shown in fig:3.8, which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the figure. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge.

For the positive half cycle of the input ac voltage, diodes D1 and D3 conduct, whereas diodes D2 and D4 remain in the OFF state. The conducting diodes will be in series with the load resistance RL and hence the load current flows through RL.

For the negative half cycle of the input ac voltage, diodes D2 and D4 conduct whereas, D1 and D3 remain OFF. The conducting diodes D2 and D4 will be in series with the load resistance RL and hence the current flows through RL in the same direction as in the previous half cycle. Thus a bi-directional wave is converted into a unidirectional wave.

****

**DB107:**

Now -a -days Bridge rectifier is available in IC with a number of DB107. In our project we are using an IC in place of bridge rectifier. The picture of DB 107 is shown in fig: 3.3.8.

**Features:**

* Good for automation insertion
* Surge overload rating - 30 amperes peak
* Ideal for printed circuit board
* Reliable low cost construction utilizing molded
* Glass passivated device
* Polarity symbols molded on body
* Mounting position: Any
* Weight: 1.0 gram



**Fig 3.3.8: DB107**

**Filtration:**

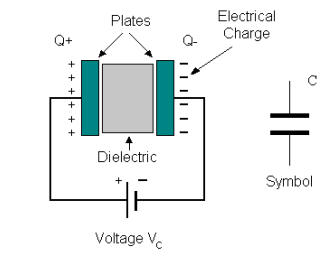
The process of converting a pulsating direct current to a pure direct current using filters is called as filtration.

**Filters:**

Electronic filters are electronic circuits, which perform signal-processing functions, specifically to remove unwanted frequency components from the signal, to enhance wanted ones.

**Introduction to Capacitors:**

The **Capacitor** or sometimes referred to as a Condenser is a passive device, and one which stores energy in the form of an electrostatic field which produces a potential (static voltage) across its plates. In its basic form a capacitor consists of two parallel conductive plates that are not connected but are electrically separated either by air or by an insulating material called the Dielectric. When a voltage is applied to these plates, a current flows charging up the plates with electrons giving one plate a positive charge and the other plate an equal and opposite negative charge This flow of electrons to the plates is known as the Charging Current and continues to flow until the voltage across the plates (and hence the capacitor) is equal to the applied voltage Vcc. At this point the capacitor is said to be fully charged and this is illustrated below. The construction of capacitor and an electrolytic capacitor are shown in figures 3.3.9 and 3.3.10 respectively.

**Fig 3.3.9:Construction Of a Capacitor Fig 3.3.10:Electrolytic Capaticor**

Units of Capacitance:

Microfarad  (μF) 1μF = 1/1,000,000 = 0.000001 = 10-6 F

 Nanofarad  (nF) 1nF = 1/1,000,000,000 = 0.000000001 = 10-9 F

 Pico farad  (pF) 1pF = 1/1,000,000,000,000 = 0.000000000001 = 10-12 F

**Regulation:**

The process of converting a varying voltage to a constant regulated voltage is called as regulation. For the process of regulation we use voltage regulators.

**Voltage Regulator:**

A voltage regulator (also called a ‘regulator’) with only three terminals appears to be a simple device, but it is in fact a very complex integrated circuit. It converts a varying input voltage into a constant ‘regulated’ output voltage. Voltage Regulators are available in a variety of outputs like 5V, 6V, 9V, 12V and 15V. The LM78XX series of voltage regulators are designed for positive input. For applications requiring negative input, the LM79XX series is used. Using a pair of ‘voltage-divider’ resistors can increase the output voltage of a regulator circuit. It is not possible to obtain a voltage lower than the stated rating. You cannot use a 12V regulator to make a 5V power supply. Voltage regulators are very robust. These can withstand over-current draw due to short circuits and also over-heating. In both cases, the regulator will cut off before any damage occurs. The only way to destroy a regulator is to apply reverse voltage to its input. Reverse polarity destroys the regulator almost instantly. Fig: 3.3.11 shows voltage regulator.



**Fig 3.3.11: Voltage Regulator**

**Resistors:**

A resistor is a two-terminal electronic component that produces a voltage across its terminals that is proportional to the electric current passing through it in accordance with Ohm's law:

***V* = *IR***

Resistors are elements of electrical networks and electronic circuits and are ubiquitous in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel/chrome).

The primary characteristics of a resistor are the resistance, the tolerance, maximum working voltage and the power rating. Other characteristics include temperature coefficient, noise, and inductance. Less well-known is critical resistance, the value below which power dissipation limits the maximum permitted current flow, and above which the limit is applied voltage. Critical resistance is determined by the design, materials and dimensions of the resistor.

Resistors can be made to control the flow of current, to work as Voltage dividers, to dissipate power and it can shape electrical waves when used in combination of other components. Basic unit is ohms.

**Theory of operation:**

Ohm's law:

The behavior of an ideal resistor is dictated by the relationship specified in Ohm's law:

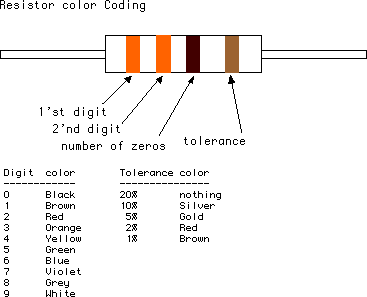
***V = IR***

Ohm's law states that the voltage (V) across a resistor is proportional to the current (I) through it where the constant of proportionality is the resistance (R).

**Power dissipation:**

The power dissipated by a resistor (or the equivalent resistance of a resistor network) is calculated using the following:

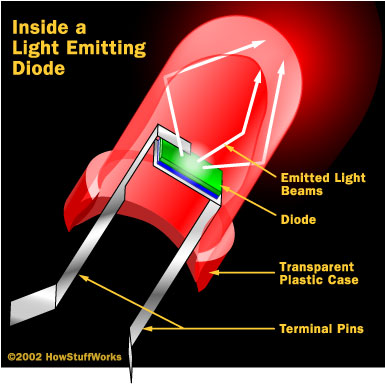
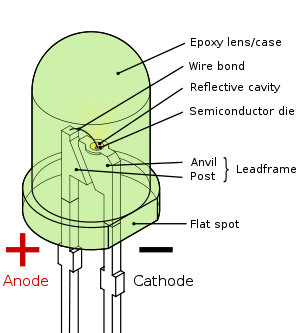
P = I^2 R = I V = \frac{V^2}{R}

** **

**Fig 3.3.12: Resistor Fig 3.3.13: Color Bands In Resistor**

**3.4. LED:**

A light-emitting diode (LED) is a semiconductor light source. LED’s are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LED’s emitted low-intensity red light, but modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness. The internal structure and parts of a led are shown in figures 3.4.1 and 3.4.2 respectively.

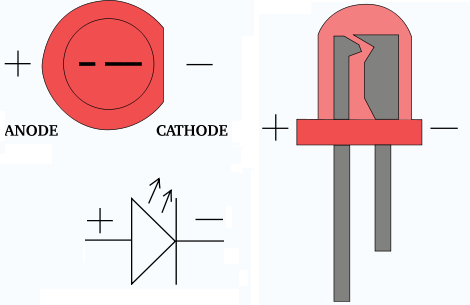
 

**Fig 3.4.1: Inside a LED Fig 3.4.2: Parts of a LED**

**Working:**

The structure of the LED light is completely different than that of the light bulb. Amazingly, the LED has a simple and strong structure. The light-emitting semiconductor material is what determines the LED's color. The LED is based on the semiconductor diode.

When a diode is forward biased (switched on), electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the colour of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor. An LED is usually small in area (less than 1 mm2), and integrated optical components are used to shape its radiation pattern and assist in reflection. LED’s present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, faster switching, and greater durability and reliability. However, they are relatively expensive and require more precise current and heat management than traditional light sources. Current LED products for general lighting are more expensive to buy than fluorescent lamp sources of comparable output. They also enjoy use in applications as diverse as replacements for traditional light sources in automotive lighting (particularly indicators) and in traffic signals. The compact size of LED’s has allowed new text and video displays and sensors to be developed, while their high switching rates are useful in advanced communications technology. The electrical symbol and polarities of led are shown in fig: 3.4.3.



**Fig 3.4.3: Electrical Symbol & Polarities of LED**

LED lights have a variety of advantages over other light sources:

* High-levels of brightness and intensity
* High-efficiency
* Low-voltage and current requirements
* Low radiated heat
* High reliability (resistant to shock and vibration)
* No UV Rays
* Long source life
* Can be easily controlled and programmed

Applications of LED fall into three major categories:

* Visual signal application where the light goes more or less directly from the LED to the human eye, to convey a message or meaning.
* Illumination where LED light is reflected from object to give visual response of these objects.
* Generate light for measuring and interacting with processes that do not involve the human visual system.

**3.5 ESP8266 WI-FI Module:**

**ESP 8266 Description:**

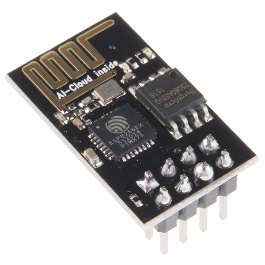
The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers (and that’s just out of the box)! The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces; it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution!

**Note:** The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic. Please do not power it directly from your 5V dev board.

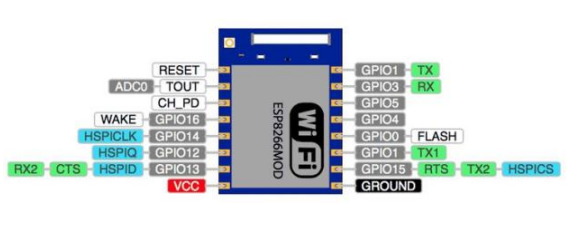
**Note:** This new version of the ESP8266 Wi-Fi Module has increased the flash disk size from 512k to 1MB.



**Fig 3.5.1. ESP8266 MODULE**

**Features:**

* 802.11 b/g/n
* Wi-Fi Direct (P2P), soft-AP
* Integrated TCP/IP protocol stack
* Integrated TR switch, balun, LNA, power amplifier and matching network
* Integrated PLLs, regulators, DCXO and power management units
* +19.5dBm output power in 802.11b mode
* Power down leakage current of <10uA
* 1MB Flash Memory
* Integrated low power 32-bit CPU could be used as application processor
* SDIO 1.1 / 2.0, SPI, UART
* STBC, 1×1 MIMO, 2×1 MIMO
* A-MPDU & A-MSDU aggregation & 0.4ms guard interval
* Wake up and transmit packets in < 2ms
* Standby power consumption of < 1.0mW (DTIM3)



**Fig.3.5.2. ESP8266 Architecture**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Function** |
| VCC | P | Power 3.0 ~ 3.6V |
| GND | P | Ground |
| RESET | I | External reset signal (Low voltage level: Active) |
| ADC(TOUT) | I | ADC Pin Analog Input 0 ~ 1V |
| CH\_PD | I | Chip Enable. High: On, chip works properly; Low: Off, small current |
| GPIO0(FLASH) | I/O | General purpose IO, If low while reset/power on takes chip into serial programming mode |
| GPIO1(TX) | I/O | General purpose IO and Serial TXd |
| GPIO3(RX) | I/O | General purpose IO and Serial RXd |
| GPIO4 | I/O | General purpose IO |
| GPIO5 | I/O | General purpose IO |
| GPIO12 | I/O | General purpose IO |
| GPIO13 | I/O | General purpose IO |
| GPIO14 | I/O | General purpose IO |
| GPIO15(HSPI\_CS) | I/O | General purpose IO, Connect this pin to ground through 1KOhm resistor to boot from internal flash. |

Note: TX1 and TX GPIO1 are internally connected, On board blue LED is also connected to this pin

**Electrical Characteristics**

Working Voltage: 3.3V

Maximum IO Driving Power IMAX: 12 mA

Maximum IO Voltage Level VMAX: 3.6V

Current Consumption: 100mAmp

**3.6 Temperature sensor:**

The digital temperature sensor like DS18B20 follows single wire protocol and it can be used to measure temperature in the range of -67oF to +257oF or -55oC to +125oC with +-5% accuracy. The range of received data from the 1-wire can range from 9-bit to 12-bit. Because, this sensor follows the single wire protocol, and the controlling of this can be done through an only pin of Microcontroller. This is an advanced level protocol, where each sensor can be set with a 64-bit serial code which aids to control numerous sensors using a single pin of the microcontroller. This article discusses an overview of a DS18B20 temperature sensor.

The DS18B20 is one type of temperature sensor and it supplies 9-bit to 12-bit readings of temperature. These values show the temperature of a particular device. The communication of this sensor can be done through a one-wire bus protocol which uses one data line to communicate with an inner microprocessor. Additionally, this sensor gets the power supply directly from the data line so that the need for an external power supply can be eliminated. The applications of the DS18B20 temperature sensor include industrial systems, consumer products, systems which are sensitive thermally, thermostatic controls, and thermometers.

DS18B20 Pin Configuration

The pin configuration of DS18B20 discussed below.

• Pin1 (Ground): This pin is used to connect to the GND terminal of the circuit

• Pin2 (Vcc): This pin is used to give the power to the sensor which ranges from 3.3V or 5V

• Pin3 (Data): The data pin supplies the temperature value, which can communicate with the help of 1-wire method.

****

**Fig. 3.6.1 DS16820 Temperature Sensor**

**Specifications**

* The specifications of this sensor include the following.
* This sensor is a programmable and digital temperature sensor
* The range of power supply is 3.0V – 5.5V
* Fahrenheit equal s to -67°F to +257°F
* The accuracy of this sensor is ±0.5°C
* The o/p resolution will range from 9-bit to 12-bit
* It changes the 12-bit temperature to digital word within 750 ms time
* This sensor can be power-driven from the data line
* Alarm options are programmable
* The multiplexing can be enabled by Unique 64-bit address
* The temperature can be calculated from -55°C to +125°C.
* These are obtainable like SOP, To-92, and also as a waterproof sensor

**Working Principle**

The working principle of this DS18B20 temperature sensor is like a temperature sensor. The resolution of this sensor ranges from 9-bits to 12-bits. But the default resolution which is used to power-up is 12-bit. This sensor gets power within a low-power inactive condition. The temperature measurement, as well as the conversion of A-to-D, can be done with a convert-T command. The resulting temperature information can be stored within the 2-byte register in the sensor, and after that, this sensor returns to its inactive state.

If the sensor is power-driven by an exterior power supply, then the master can provide read time slots next to the Convert T command. The sensor will react by supplying 0 though the temperature change is in the improvement and reacts by supplying 1 though the temperature change is done.

**DS18B20 Temperature Sensor Applications**

* The applications of DS18B20 include the following.
* This sensor is extensively used to calculate temperature within rigid environments which includes mines, chemical solutions, otherwise soil, etc.
* This sensor is used to measure the liquid temperature.
* We can use it in the thermostat controls system.
* It can be used in industries as a temperature measuring device.
* This sensor is used as a thermometer.
* It can be used in devices like which are sensitive to thermal.
* These are used in HVAC systems.

**3.7 AC Motor**

**3.7.1. Introduction:**

AC motors, or alternating current motors, are a type of electric motor that operates on the principle of alternating current. They are widely used in various applications, from industrial machinery to household appliances. AC motors are popular due to their simplicity, reliability, and ability to provide a constant and smooth operation. There are two main types of AC motors: synchronous and asynchronous (also known as induction motors). In this explanation, we'll focus on asynchronous AC motors, as they are more commonly used.

**3.7.2. Working Principle:**

The basic working principle of an AC motor involves the interaction between magnetic fields and electric currents. An asynchronous AC motor consists of two main parts: a stator and a rotor.

1. Stator: The stator is the stationary part of the motor and contains a set of coils that are connected to the AC power supply. When an AC voltage is applied to these coils, it produces a rotating magnetic field.

2. Rotor: The rotor is the rotating part of the motor, typically located inside the stator. It can be constructed with conductive bars or coils. As the magnetic field generated by the stator rotates, it induces an electromotive force (EMF) in the rotor.

3. Principle of Induction: The rotor, being in the presence of the rotating magnetic field, experiences a changing magnetic flux. According to Faraday's law of electromagnetic induction, an EMF is induced in the rotor conductors. This induced EMF causes a current to flow in the rotor conductors.

4. Production of Rotating Magnetic Field: The current in the rotor conductors, in turn, generates its own magnetic field. This magnetic field interacts with the stator's rotating magnetic field, creating a torque that causes the rotor to turn. This torque allows the motor to do mechanical work.

5. Continuous Rotation: The rotor continually follows the changing magnetic field of the stator, maintaining rotation as long as the AC power is applied. The speed of rotation is influenced by the frequency of the AC power supply and the number of poles in the motor.

**Types of AC Motors:**

* **Synchronous Motors**: These motors rotate at a constant speed synchronized with the frequency of the AC power supply. They find applications where precise speed control is required.
* **Induction Motors**: The most common type, where the rotor speed is always less than the synchronous speed. Induction motors are self-starting and robust, making them suitable for a wide range of applications.

AC motors are versatile and are used in various devices, ranging from small fans to large industrial machinery. Their simplicity and reliability make them essential in many aspects of modern life.

**3.8 Current sensor:**

In practice a current transformer can be used as a current sensor. The current sensor is to be connected in series with the transmission lines. When current in a circuit is too high to directly apply to measuring instruments, a current transformer produces a reduced current accurately proportional to the current in the circuit, which can be conveniently connected to measuring and recording instruments. A current transformer also isolates the measuring instruments from what may be very high voltage in the monitored circuit.

The primary of the current transformer is to be connected to the transmission lines and the secondary is to be connected to the microcontroller.

**3.9 LCD display:**

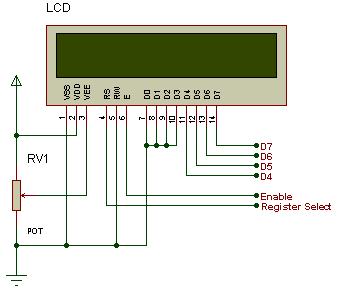
**LCD DISPLAY**

**LCD Background:**

One of the most common devices attached to a micro controller is an LCD display. Some of the most common LCD’s connected to the many microcontrollers are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

**Basic 16x 2 Characters LCD**

**Figure 1: LCD Pin diagram**

****

**Pin description:**

|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Name** | **Description** |
| Pin no. 1 | **VSS** | Power supply (GND) |
| Pin no. 2 | **VCC** | Power supply (+5V) |
| Pin no. 3 | **VEE** | Contrast adjust |
| Pin no. 4 | **RS** | 0 = Instruction input 1 = Data input |
| Pin no. 5 | **R/W** | 0 = Write to LCD module 1 = Read from LCD module |
| Pin no. 6 | **EN** | Enable signal |
| Pin no. 7 | **D0** | Data bus line 0 (LSB) |
| Pin no. 8 | **D1** | Data bus line 1 |
| Pin no. 9 | **D2** | Data bus line 2 |
| Pin no. 10 | **D3** | Data bus line 3 |
| Pin no. 11 | **D4** | Data bus line 4 |
| Pin no. 12 | **D5** | Data bus line 5 |
| Pin no. 13 | **D6** | Data bus line 6 |
| Pin no. 14 | **D7** | Data bus line 7 (MSB) |

**Table 1: Character LCD pins with Microcontroller**

The LCD requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

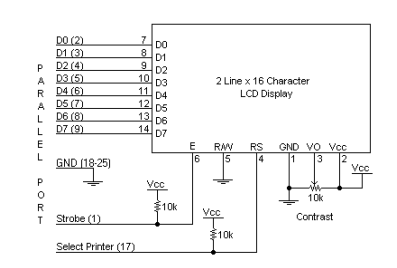
The **EN** line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring **EN** high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The **RS** line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen we would set RS high.

The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

**Schematic:**

****

**Circuit Description:**

Above is the quite simple schematic. The LCD panel's Enable and RegisterSelect is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. We can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if we have trouble with the circuit working properly.

**SETB RW**

Handling the EN control line:

As we mentioned above, the EN line is used to tell the LCD that we are ready for it to execute an instruction that we've prepared on the data bus and on the other control lines. Note that the EN line must be raised/ lowered before/after each instruction sent to the LCD regardless of whether that instruction is read or write text or instruction. In short, we must always manipulate EN when communicating with the LCD. EN is the LCD's way of knowing that we are talking to it. If we don't raise/lower EN, the LCD doesn't know we're talking to it on the other lines.

Thus, before we interact in any way with the LCD we will always bring the **EN** line low with the following instruction:

**CLR EN**

And once we've finished setting up our instruction with the other control lines and data bus lines, we'll always bring this line high:

**SETB EN**

The line must be left high for the amount of time required by the LCD as specified in its datasheet. This is normally on the order of about 250 nanoseconds, but checks the datasheet. In the case of a typical microcontroller running at 12 MHz, an instruction requires 1.08 microseconds to execute so the EN line can be brought low the very next instruction. However, faster microcontrollers (such as the DS89C420 which executes an instruction in 90 nanoseconds given an 11.0592 MHz crystal) will require a number of NOPs to create a delay while EN is held high. The number of NOPs that must be inserted depends on the microcontroller we are using and the crystal we have selected. The instruction is executed by the LCD at the moment the EN line is brought low with a final CLR EN instruction.

**Checking the busy status of the LCD:**

As previously mentioned, it takes a certain amount of time for each instruction to be executed by the LCD. The delay varies depending on the frequency of the crystal attached to the oscillator input of the LCD as well as the instruction which is being executed.

While it is possible to write code that waits for a specific amount of time to allow the LCD to execute instructions, this method of "waiting" is not very flexible. If the crystal frequency is changed, the software will need to be modified. A more robust method of programming is to use the "Get LCD Status" command to determine whether the LCD is still busy executing the last instruction received.

The "Get LCD Status" command will return to us two tidbits of information; the information that is useful to us right now is found in DB7. In summary, when we issue the "Get LCD Status" command the LCD will immediately raise DB7 if it's still busy executing a command or lower DB7 to indicate that the LCD is no longer occupied. Thus our program can query the LCD until DB7 goes low, indicating the LCD is no longer busy. At that point we are free to continue and send the next command.

**3.10 Buzzer:**

Basically, the sound source of a piezoelectric sound component is a piezoelectric diaphragm. A piezoelectric diaphragm consists of a piezoelectric ceramic plate which has electrodes on both sides and a metal plate (brass or stainless steel, etc.). A piezoelectric ceramic plate is attached to a metal plate with adhesives. Applying D.C. voltage between electrodes of a piezoelectric diaphragm causes mechanical distortion due to the piezoelectric effect. For a misshaped piezoelectric element, the distortion of the piezoelectric element expands in a radial direction. And the piezoelectric diaphragm bends toward the direction. The metal plate bonded to the piezoelectric element does not expand. Conversely, when the piezoelectric element shrinks, the piezoelectric diaphragm bends in the direction Thus, when AC voltage is applied across electrodes, the bending is repeated, producing sound waves in the air.

To interface a buzzer the standard transistor interfacing circuit is used. Note that if a different power supply is used for the buzzer, the 0V rails of each power supply must be connected to provide a common reference.

If a battery is used as the power supply, it is worth remembering that piezo sounders  
draw much less current than buzzers. Buzzers also just have one ‘tone’, whereas a  
piezo sounder is able to create sounds of many different tones.

To switch on buzzer -high 1

To switch off buzzer -low 1

Notice (Handling) In Using Self Drive Method

1) When the piezoelectric buzzer is set to produce intermittent sounds, sound may be heard continuously even when the self drive circuit is turned ON / OFF at the "X" point shown in Fig. 9. This is because of the failure of turning off the feedback voltage.

2) Build a circuit of the piezoelectric sounder exactly as per the recommended circuit shown in the catalog. Hfe of the transistor and circuit constants are designed to ensure stable oscillation of the piezoelectric sounder.

3) Design switching which ensures direct power switching.

4) The self drive circuit is already contained in the piezoelectric buzzer. So there is no need to prepare another circuit to drive the piezoelectric buzzer.

5) Rated voltage (3.0 to 20Vdc) must be maintained. Products which can operate with voltage higher than 20Vdc are also available.

6) Do not place resistors in series with the power source, as this may cause abnormal oscillation. If a resistor is essential to adjust sound pressure, place a capacitor (about 1μF) in parallel with the piezo buzzer.

7) Do not close the sound emitting hole on the front side of casing.

8) Carefully install the piezo buzzer so that no obstacle is placed within 15mm from the sound release hole on the front side of the casing.

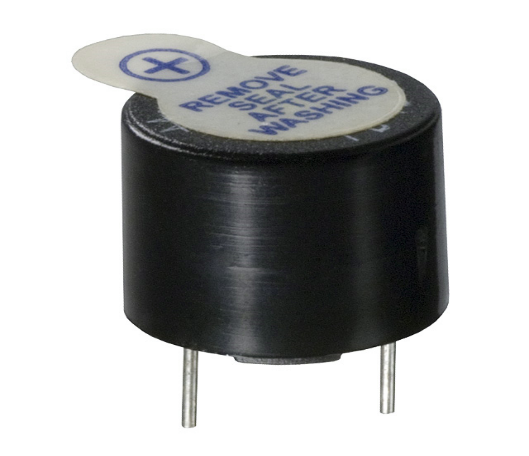


Fig: Picture of buzzer

**SOFTWARE DESCRIPTION**

This project is implemented using following software’s:

* Express PCB – for designing circuit
* PIC C compiler - for compilation part

**3.11 Express PCB:**

Breadboards are great for prototyping equipment as it allows great flexibility to modify a design when needed; however the final product of a project, ideally should have a neat PCB, few cables, and survive a shake test. Not only is a proper PCB neater but it is also more durable as there are no cables which can yank loose.

Express PCB is a software tool to design PCBs specifically for manufacture by the company Express PCB (no other PCB maker accepts Express PCB files). It is very easy to use, but it does have several limitations.

It can be likened to more of a toy then a professional CAD program.

It has a poor part library (which we can work around)

It cannot import or export files in different formats

It cannot be used to make prepare boards for DIY production

Express PCB has been used to design many PCBs (some layered and with surface-mount parts. Print out PCB patterns and use the toner transfer method with an Etch Resistant Pen to make boards. However, Express PCB does not have a nice print layout. Here is the procedure to design in Express PCB and clean up the patterns so they print nicely.

**3.11.1 Preparing Express PCB for First Use:**

Express PCB comes with a less then exciting list of parts. So before any project is started head over to Audio logic and grab the additional parts by morsel, ppl, and tangent, and extract them into your Express PCB directory. At this point start the program and get ready to setup the workspace to suit your style.

Click View -> Options. In this menu, setup the units for “mm” or “in” depending on how you think, and click “see through the top copper layer” at the bottom. The standard color scheme of red and green is generally used but it is not as pleasing as red and blue.

**3.11.2 The Interface:**

When a project is first started you will be greeted with a yellow outline. This yellow outline is the dimension of the PCB. Typically after positioning of parts and traces, move them to their final position and then crop the PCB to the correct size. However, in designing a board with a certain size constraint, crop the PCB to the correct size before starting.

Fig: 3.11.1 show the toolbar in which the each button has the following functions:



**Fig 3.11.1 : Tool bar necessary for the interface**

* The select tool: It is fairly obvious what this does. It allows you to move and manipulate parts. When this tool is selected the top toolbar will show buttons to move traces to the top / bottom copper layer, and rotate buttons.
* The zoom to selection tool: does just that.
* The place pad: button allows you to place small soldier pads which are useful for board connections or if a part is not in the part library but the part dimensions are available. When this tool is selected the top toolbar will give you a large selection of round holes, square holes and surface mount pads.
* The place component: tool allows you to select a component from the top toolbar and then by clicking in the workspace places that component in the orientation chosen using the buttons next to the component list. The components can always be rotated afterwards with the select tool if the orientation is wrong.
* The place trace: tool allows you to place a solid trace on the board of varying thicknesses. The top toolbar allows you to select the top or bottom layer to place the trace on.
* The Insert Corner in trace: button does exactly what it says. When this tool is selected, clicking on a trace will insert a corner which can be moved to route around components and other traces.
* The remove a trace button is not very important since the delete key will achieve the same result.

**3.11.3 Design Considerations:**

Before starting a project there are several ways to design a PCB and one must be chosen to suit the project’s needs.

Single sided, or double sided?

When making a PCB you have the option of making a single sided board, or a double sided board. Single sided boards are cheaper to produce and easier to etch, but much harder to design for large projects. If a lot of parts are being used in a small space it may be difficult to make a single sided board without jumper over traces with a cable. While there’s technically nothing wrong with this, it should be avoided if the signal travelling over the traces is sensitive (e.g. audio signals).

A double sided board is more expensive to produce professionally, more difficult to etch on a DIY board, but makes the layout of components a lot smaller and easier. It should be noted that if a trace is running on the top layer, check with the components to make sure you can get to its pins with a soldering iron. Large capacitors, relays, and similar parts which don’t have axial leads can NOT have traces on top unless boards are plated professionally.

When using a double sided board you must consider which traces should be on what side of the board. Generally, put power traces on the top of the board, jumping only to the bottom if a part cannot be soldiered onto the top plane (like a relay), and vice- versa.

Some projects like power supplies or amps can benefit from having a solid plane to use for ground. In power supplies this can reduce noise, and in amps it minimizes the distance between parts and their ground connections, and keeps the ground signal as simple as possible. However, care must be taken with stubborn chips such as the TPA6120 amplifier from TI. The TPA6120 datasheet specifies not to run a ground plane under the pins or signal traces of this chip as the capacitance generated could effect performance negatively.

**3.12 PIC Compiler:**

PIC compiler is software used where the machine language code is written and compiled. After compilation, the machine source code is converted into hex code which is to be dumped into the microcontroller for further processing. PIC compiler also supports C language code.

It’s important that you know C language for microcontroller which is commonly known as Embedded C. As we are going to use PIC Compiler, hence we also call it PIC C. The PCB, PCM, and PCH are separate compilers. PCB is for 12-bit opcodes, PCM is for 14-bitopcodes, and PCH is for 16-bit opcode PIC microcontrollers. Due to many similarities, all three compilers are covered in this reference manual. Features and limitations that apply to only specific microcontrollers are indicated within. These compilers are specifically designed to meet the unique needs of the PIC microcontroller. This allows developers to quickly design applications software in a more readable, high-level language. When compared to a more traditional C compiler, PCB, PCM, and PCH have some limitations. As an example of the limitations, function recursion is not allowed.

This is due to the fact that the PIC has no stack to push variables onto, and also because of the way the compilers optimize the code. The compilers can efficiently implement normal C constructs, input/output operations, and bit twiddling operations. All normal C data types are supported along with pointers to constant arrays, fixed point decimal, and arrays of bits.

PIC C is not much different from a normal C program. If you know assembly, writing a C program is not a crisis. In PIC, we will have a main function, in which all your application specific work will be defined. In case of embedded C, you do not have any operating system running in there. So you have to make sure that your program or main file should never exit. This can be done with the help of simple while (1) or for (;;) loop as they are going to run infinitely.

We have to add header file for controller you are using, otherwise you will not be able to access registers related to peripherals.

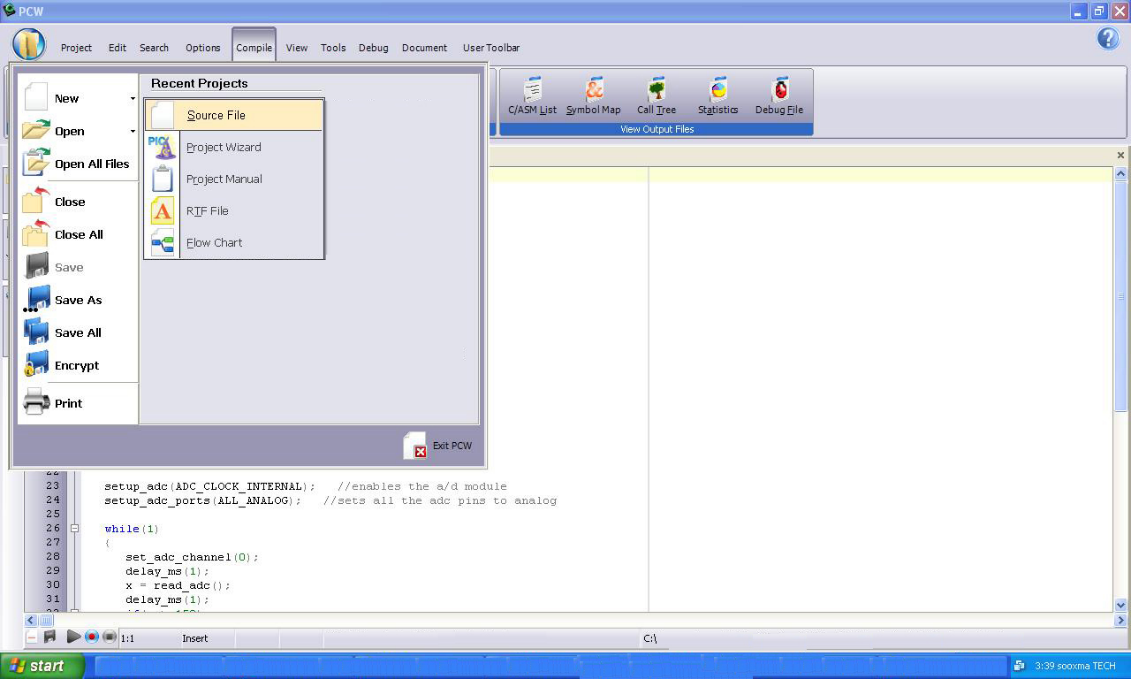
#include <16F876.h> // header file for PIC 16F876**//**

**3.13 Compilation and simulation steps:**

For PIC microcontroller, PIC C compiler is used for compilation. The compilation steps are as follows:

1.Open PIC C compiler.

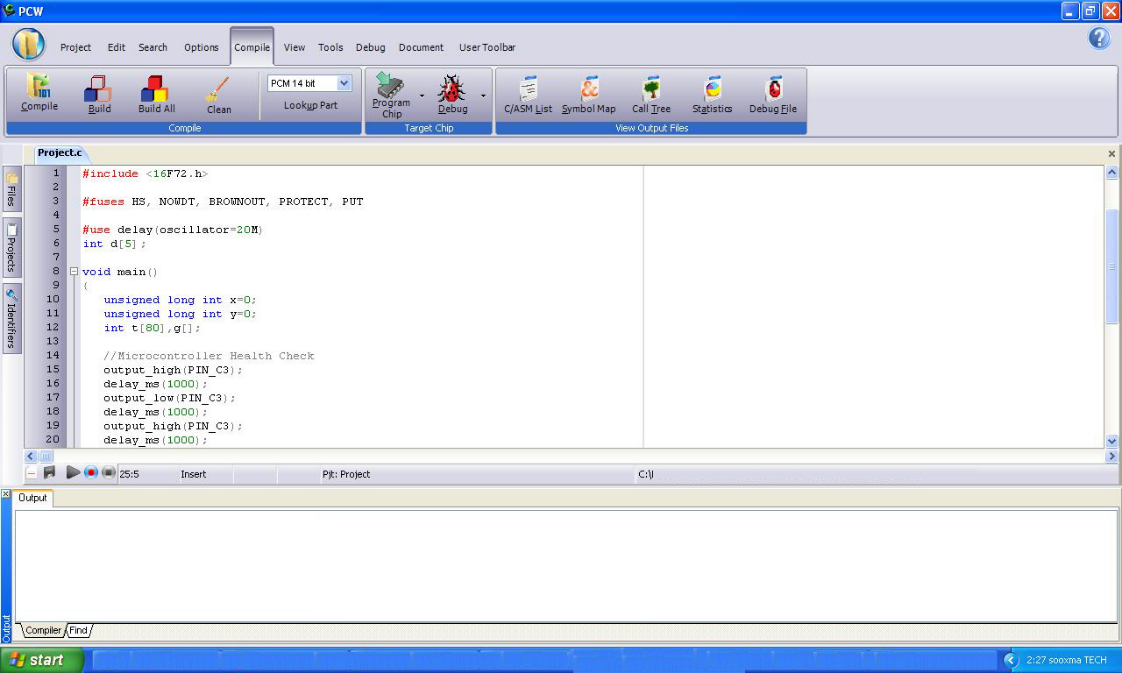
2.You will be prompted to choose a name for the new project, so create a separate folder where all the files of your project will be stored, choose a name and click save.



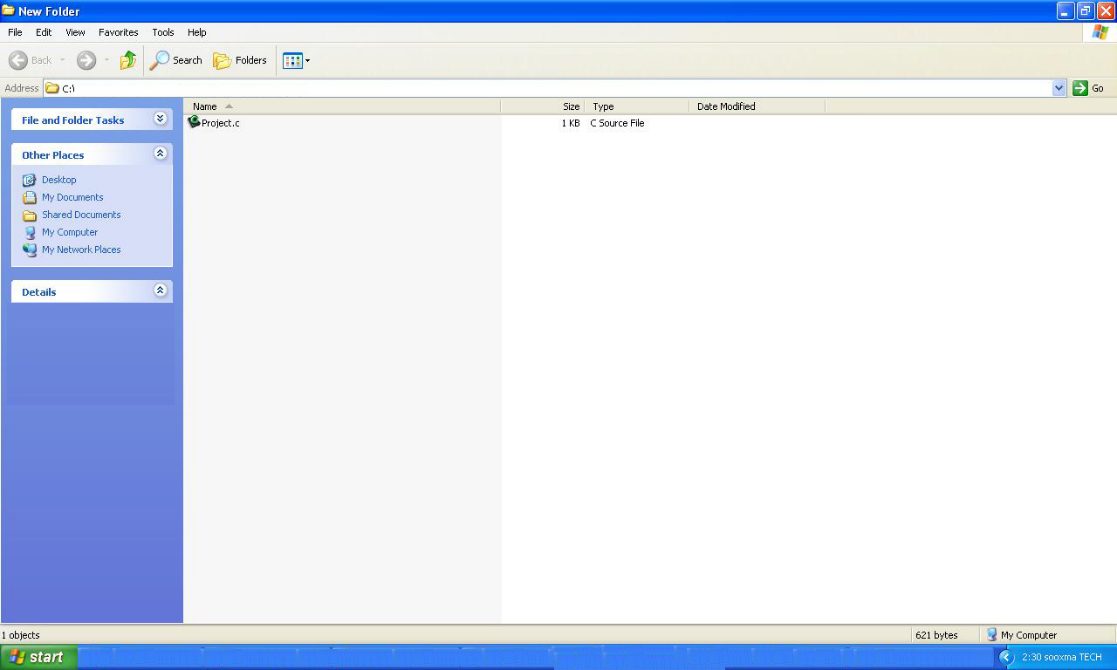
**Fig 3.13.1: Picture of opening a new file using PIC C compiler**

3.Click **Project, New**, and something the box named 'Text1' is where your code should be written later.

4.Now you have to click 'File, Save as' and choose a file name for your source code ending with the letter '.c'. You can name as 'project.c' for example and click save. Then you have to add this file to your project work.

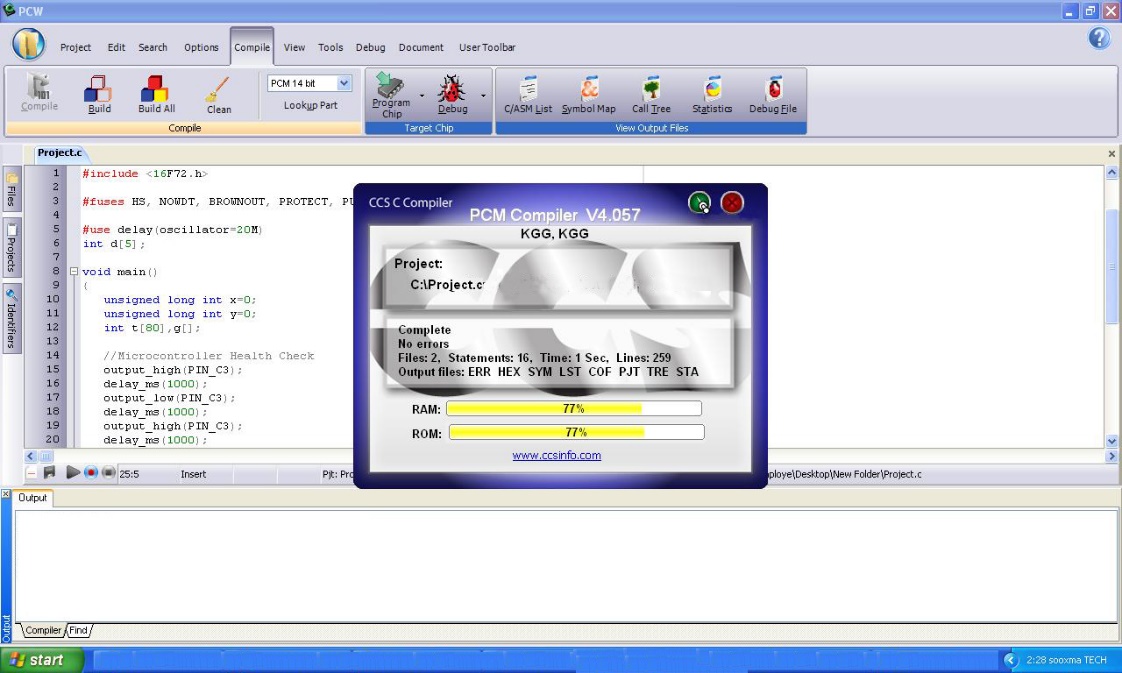


**Fig 3.13.2: Picture of compiling a new file using PIC C compiler**



**Fig 3.13.3: Picture of compiling a project.c file using PIC C compiler**

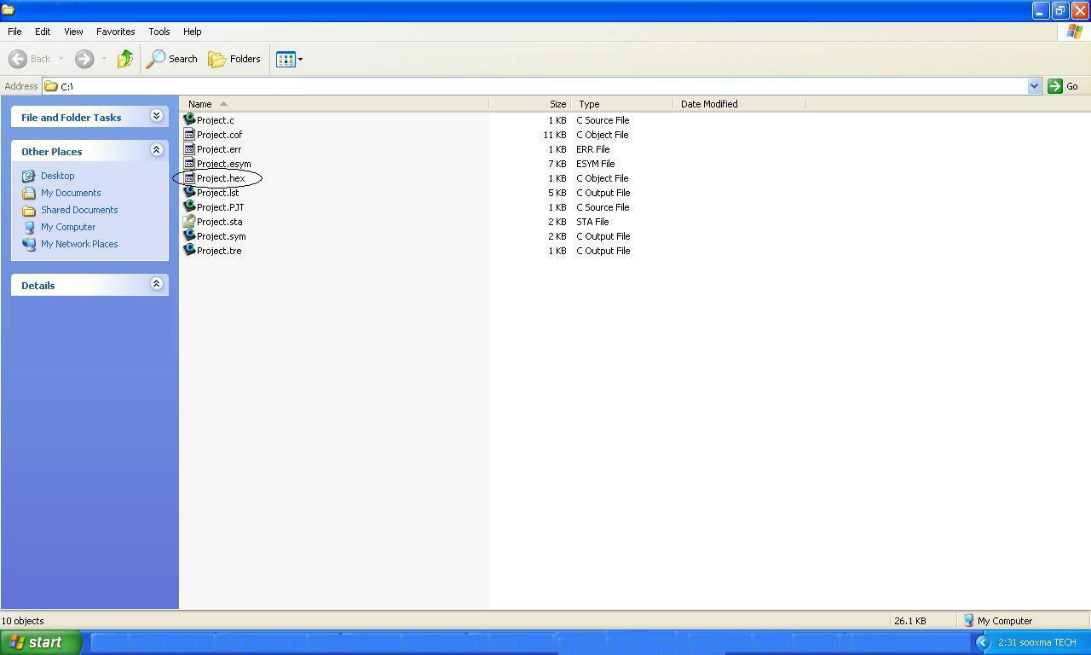
5.You can then start to write the source code in the window titled 'project.c' then before testing your source code; you have to compile your source code, and correct eventual syntax errors.



**Fig 3.13.4: Picture of checking errors and warnings using PIC C compiler**

6.By clicking on compile option .hex file is generated automatically.

7.This is how we compile a program for checking errors and hence the compiled program is saved in the file where we initiated the program.



**Fig 3.13.5: Picture of .hex file existing using PIC C compiler**

After compilation, next step is simulation. Here first circuit is designed in Express PCB using Proteus 7 software and then simulation takes place followed by dumping. The simulation steps are as follows:

8.Open Proteus 7 and click on IS1S6.

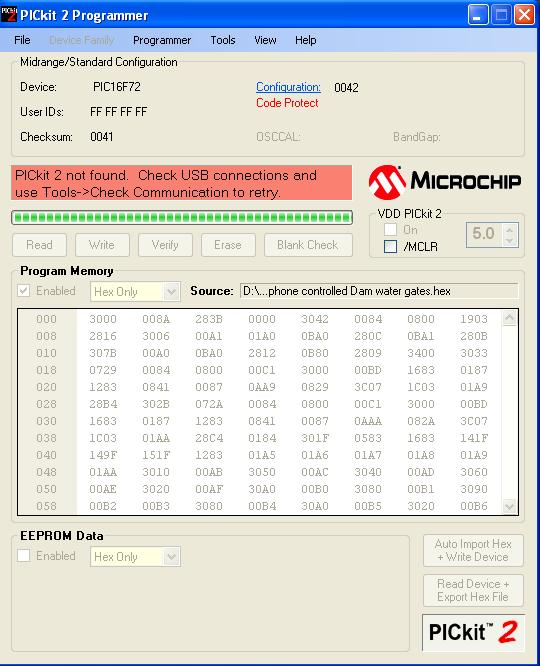
9.Now it displays PCB where circuit is designed using microcontroller. To design circuit components are required. So click on component option.

10. Now click on letter ’p’, then under that select PIC16F73 ,other components related to the project and click OK. The PIC 16F73 will be called your “*'*Target device*”,* which is the final destination of your source code.

**3.14 Dumping steps:**

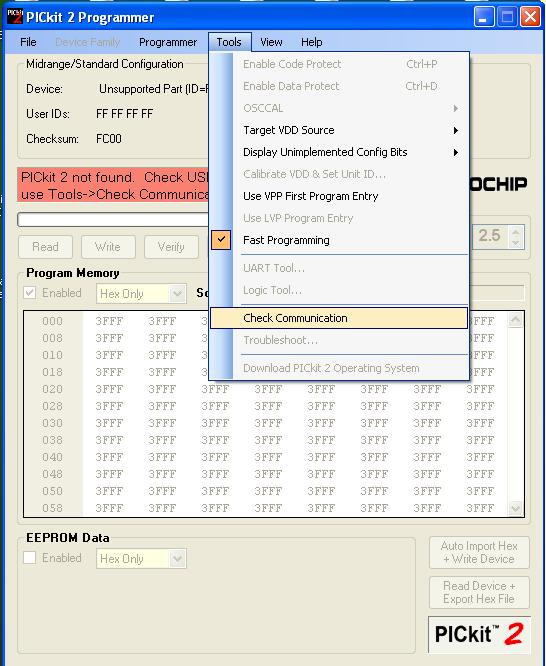
The steps involved in dumping the program edited in proteus 7 to microcontroller are shown below:

1.Initially before connecting the program dumper to the microcontroller kit the window is appeared as shown below.



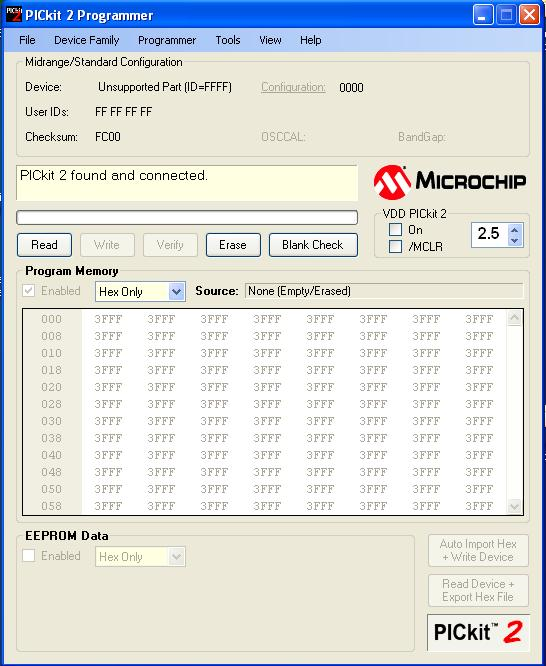
**Fig 3.14.1: Picture of program dumper window**

2.Select Tools option and click on Check Communication for establishing a connection as shown in below window



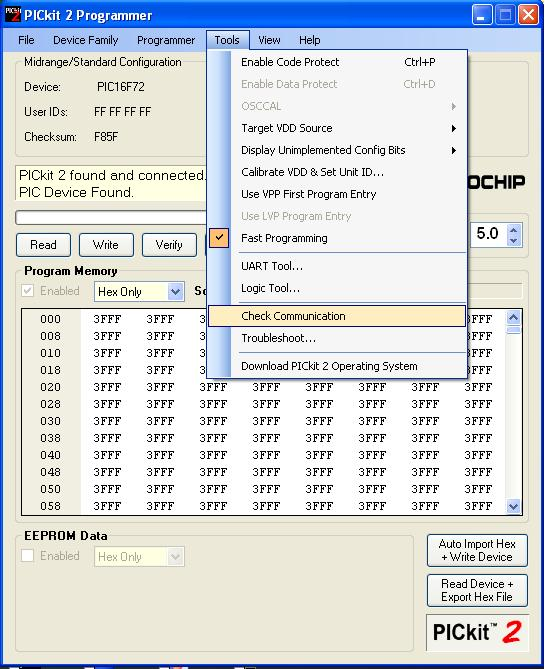
**Fig 3.14.2: Picture of checking communications before dumping program into microcontroller**

3. After connecting the dumper properly to the microcontroller kit the window is appeared as shown below.



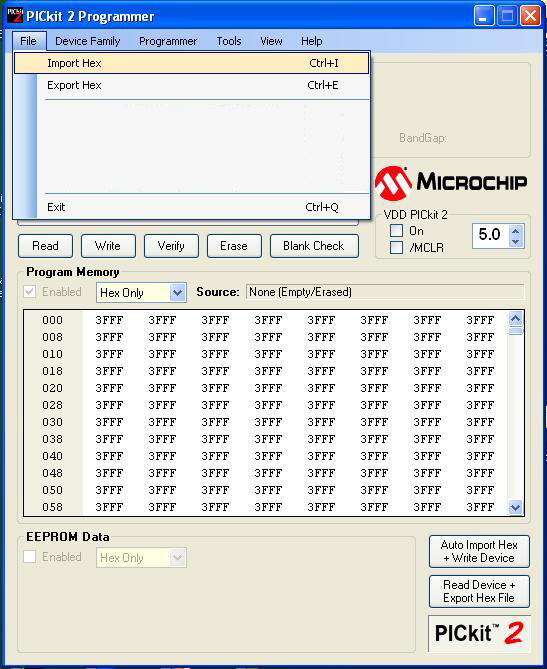
**Fig 3.14.3: Picture after connecting the dumper to microcontroller**

4. Again by selecting the Tools option and clicking on Check Communication the microcontroller gets recognized by the dumper and hence the window is as shown below.



**Fig 4.9: Picture of dumper recognition to microcontroller**

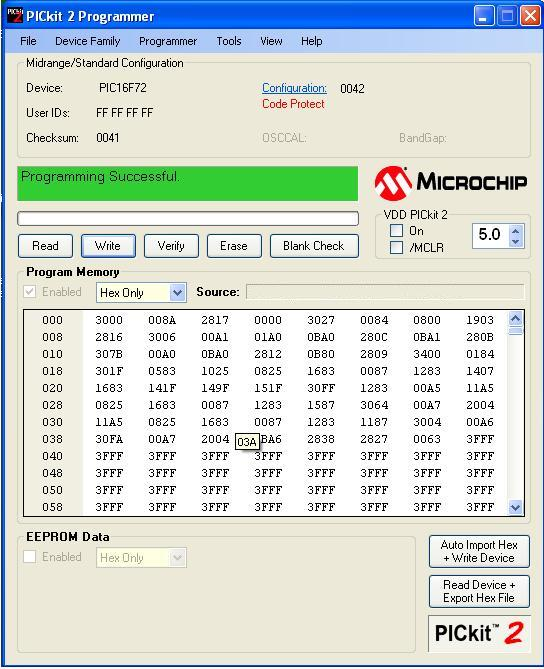
5. Import the program which is ‘.hex’ file from the saved location by selecting File option and clicking on ‘Import Hex’ as shown in below window.

****

**Fig 4.10: Picture of program importing into the microcontroller**

6. After clicking on ‘Import Hex’ option we need to browse the location of our program and click the ‘prog.hex’ and click on ‘open’ for dumping the program into the microcontroller.

7. After the successful dumping of program the window is as shown below.

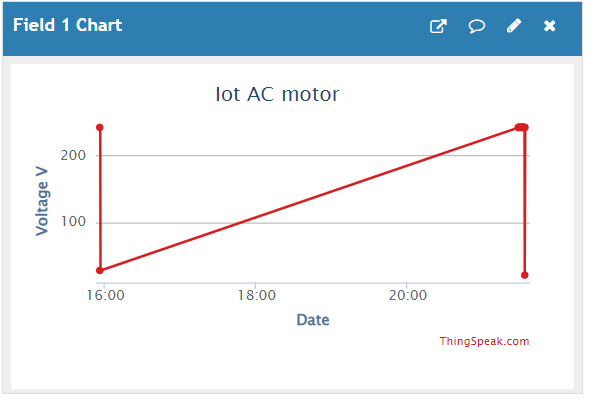
****

**Fig 3.14: Picture after program dumped into the microcontroller**

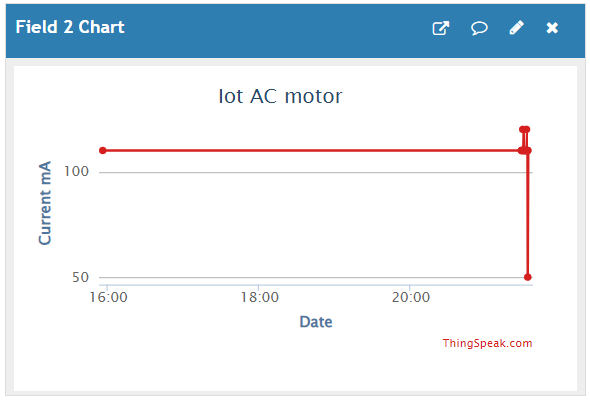
**CHAPTER 4 : RESULTS**

**4.1 RESULT**

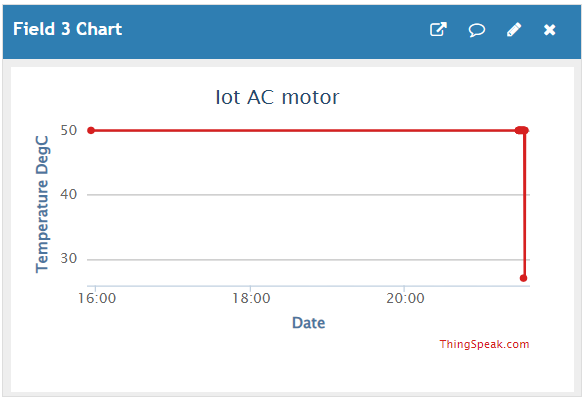
The main controlling device of the whole system is a Microcontroller. AC motor through Voltage sensor, current sensor, temperature sensor and frequency counter are interfaced to Microcontroller. The Microcontroller gets input from all the sensors, which will be processed and transmitted into to the thingspeak cloud over WI-FI module. The parameters are displayed on the mobile phone. The system alerts through BUZZER, if any abnormality is detected in the parameter values. To perform the task, Microcontroller is loaded with an intelligent program written in embedded ‘C’ language. The Microcontroller is programmed using Embedded C language.



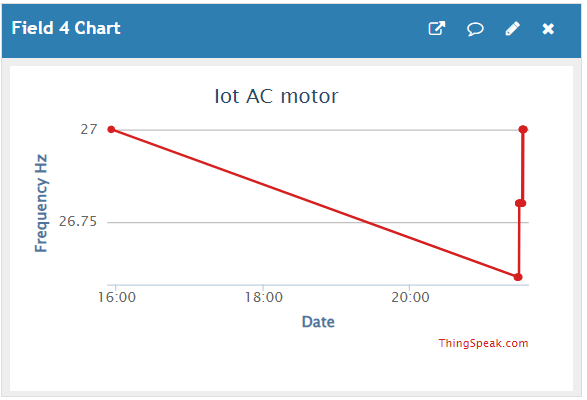
**Fig 4.1.1: Representation of Voltage graph.**

****

**Fig 4.1.2: Representation of Current graph.**

****

**Fig 4.1.3: Representation of Temperature graph.**



**Fig 4.1.4: Representation of Frequency graph.**

**CHAPTER 5: CONCLUSION AND FUTURE SCOPE**

**5.1 Conclusion :**

In conclusion, implementing AC motor monitoring systems using IoT (Internet of Things) technology offers a myriad of benefits for industries and applications reliant on electric motors. The integration of IoT enables real-time data acquisition, analysis, and remote management, leading to enhanced efficiency, reduced downtime, and proactive maintenance strategies. By leveraging sensors, connectivity, and advanced analytics, organizations can gain valuable insights into motor performance, detect potential issues before they escalate, and optimize energy consumption.Furthermore, AC motor monitoring through IoT facilitates predictive maintenance, ultimately extending the lifespan of equipment and reducing overall operational costs. The ability to remotely monitor and control motors not only improves operational efficiency but also contributes to a more sustainable and environmentally friendly approach by optimizing energy usage.

Integrating features of all the hardware components used have been developed in it. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Secondly, using highly advanced IC’s with the help of growing technology, the project has been successfully implemented. Thus the project has been successfully designed and tested.

**Future Scope:**

The future scope for AC motor monitoring systems using IoT (Internet of Things) is promising and involves several advancements and opportunities. Here are some key aspects of the future scope for AC motor monitoring systems with IoT:

**Real-time Monitoring and Diagnostics:**

IoT-enabled AC motor monitoring systems can provide real-time data on motor performance, temperature, vibration, and other critical parameters. Advanced analytics and machine learning algorithms can be employed to analyze this data in real-time, enabling predictive maintenance and early fault detection.

**Predictive Maintenance:**

By leveraging historical data and machine learning algorithms, predictive maintenance models can be developed to anticipate potential motor failures before they occur. This approach minimizes downtime, reduces maintenance costs, and extends the lifespan of AC motors.

**Energy Efficiency Optimization:**

IoT-enabled systems can monitor energy consumption patterns of AC motors and optimize their performance for energy efficiency. Smart algorithms can adjust motor operation parameters based on real-time energy demand, contributing to overall energy savings in industrial processes.

**Remote Monitoring and Control:**

With IoT, AC motor monitoring can be conducted remotely, allowing for centralized monitoring of multiple motors across different locations. Remote control features enable adjustments to motor settings, operation modes, and troubleshooting without physical presence.

**Cost Reduction and Scalability:**

As technology matures, the cost of IoT sensors and devices is expected to decrease, making AC motor monitoring systems more affordable and scalable for a wide range of industries.

**Environmental Monitoring:**

Integration of environmental sensors with AC motor monitoring systems can provide insights into the working conditions, helping industries comply with environmental regulations and reduce their ecological footprint.

The future of AC motor monitoring systems using IoT holds great potential for increased efficiency, reduced downtime, and improved overall performance in industrial settings. Ongoing advancements in technology and a focus on innovation will likely shape the evolution of these systems in the coming years.