CHAPTER 1

INTRODUCTION

In the recent decades, Urbanization has increased tremendously. At the same phase there is an increase in waste production. Waste management has been a crucial issue to be considered. This paper is a way to achieve this good cause. In this system, smart bin is built on a microcontroller with Ultrasonic sensor, GAS sensor, Temperature and light automation. Ultrasonic sensor is placed at the top of the dustbin which will measure the stature of the dustbin. The threshold stature is set as 10cm. Project will be programmed in such a way that when the dustbin is being filled, the remaining height from the threshold height will be displayed. Once the garbage reaches the threshold level ultrasonic sensor will trigger the IOT network which will continuously alert the required authority until the garbage in the dustbin is squashed. Once the dustbin is squashed, people can reuse the dustbin. At regular intervals dustbin will be squashed. Once these smart bins are implemented on a large scale, by replacing our traditional bins present today, waste can be managed efficiently as it avoids unnecessary lumping of wastes on roadside. Foul smell from these rotten wastes that remain untreated for a long time, due to negligence of authorities and carelessness of public may lead to long term problems. Breeding of insects and mosquitoes can create nuisance around promoting unclean environment. This may even cause dreadful diseases.

Though the world is in a stage of up gradation, there is yet another problem that has to be dealt with. Garbage! Pictures of garbage bins being overfull and the garbage being spilled out from the bins can be seen all around. This leads to various diseases as large number of insects and mosquitoes breed on it. A big challenge in the urban cities is solid waste management. Hence, smart dustbin is a system which can eradicate this problem or at least reduce it to the minimum level.

1.1 EXISTING SYSTEM:

With day by day increase in population, number of start-up units, the average waste generated by an individual is increasing every day which leads to severe waste management issues. Nowadays sanitariness and cleanliness are one of the agitated issues by any state all over the world. This is mainly caused due to improper waste monitoring and management processes. To overcome these problems a garbage monitoring system is designed which periodically measures the level of trash in the containers and alerts the concerned authorities. This is done by positioning a sensor network to collect the garbage levels in the containers and remotely triggering the required controls using IoT. IoT is one well-known technology which helps in solving many real-world problems. Internet of Things (IoT) strongly supports the notion of interfacing and observing the real world objects (things) through the Internet. The detection of bin parameters in the container is carried out using the NodeMCU ESP8266 microcontroller board. The main objective of this paper is to trigger an alert message to the people concerned when the container is filled thereby avoiding the over spilling of garbage.

Problem identifications

- Monitoring and communication process has been made
- There is temperature, gas monitering unit and light automation

1.2 PROPOSED SYSTEM:

In this system, smart bin is built on a microcontroller based platform which is interfaced with ESP8266 wifi module and sensor network (Ultrasonic/MQ6/LDR/LM35). Ultrasonic sensor is fixed at the top of the dustbin which will measure the stature of the dustbin. The threshold stature is set as equal distance according size of the dust bin. The controller will be programmed in such a way that when the dustbin is being filled, the remaining height from the threshold height will be displayed. Once the garbage reaches the threshold level sensor will trigger the Microcontroller. The microcontroller sends the information to Wi-Fi module which will continuously alert the required authority until the garbage in the dustbin is squashed. The light automation process also present with this system for enabling the light is ON at night time and disabling the light at day time. This system also includes GAS sensor (MQ6), Temperature sensor and LDR for measuring the various gas level, temperature and light intensity level from dust bin as well as environment.

CHAPTER 2

LITERATURE REVIEW

1. **Paavan Lakshmana Chowdary S, Sai Teja G, Naga Mahesh K Department of Electronics and Communication Engineering National Institute of Technology, Andhra Pradesh An IoT based Smart Garbage Alert System IEEE-2019**

With day by day increase in population, number of start-up units, the average waste generated by an individual is increasing every day which leads to severe waste management issues. Nowadays cleanliness are one of the agitated issues by any state all over the world. This is mainly caused due to improper waste monitoring and management processes. To overcome these problems a garbage monitoring system is designed which periodically measures the level of trash in the containers and alerts the concerned authorities. This is done by positioning a sensor network to collect the garbage levels in the containers and remotely triggering the required controls using IoT. IoT is one well-known technology which helps in solving many real-world problems. Internet of Things (IoT) strongly supports the notion of interfacing and observing the real world objects (things) through the Internet. The detection of bin parameters in the container is carried out using the NodeMCU ESP8266 microcontroller board. The main objective of this paper is to trigger an alert message to the people concerned when the container is filled thereby avoiding the over spilling of garbage.

2. **Pablo Velásquez Department of Power and machinery University of Concepción Chillán, Chile A low-cost IoT based Environmental Monitoring System. A citizen approach to pollution awareness IEEE-2018**

The concept of quality of life is used as a measure of the welfare of a society, highlighting the relationship between the environment and health, mainly associated with pollution. Hence the importance of monitoring and communicating its results, in a timely manner, to the community. But often the last part of the process fail, there is a lack of communication between data, information and community. Using open hardware and open source tools, and based on the Internet of things concept, a lowcost citizen monitoring network, that cost less than US $ 150 was implemented. This network measures Carbon Monoxide (CO), Temperature, Relative Humidity, Particulate Matter 2.5, Noise and UV radiation, with a reading frequency of every 40 s and a hibernation period of 15 min. The environmental monitoring network gather information from the environment, through sensors, and then store it in a MySQL database, to finally be deployed in a web site hosted on a local web server Apache, implemented ex professor. In effect, the system is able to deliver geotagged information and the state of environmental pollution to the citizens.

3. **Eveneet Johar, Rahul mishra Atharva college of engineering, Mumbai IoT based intelligent garbage monitoring system IEEE 2019**

Waste management is simple yet effective ways of reducing the amount of waste dumped into our landfills. But there are people who are unaware or even choose to ignore the fact that waste segregation and recycling are environment friendly solutions to the problem of wastes management and disposal. In the Philippines, there are recycling centers but the process is tedious and done manually. There are guidelines implemented by the government with regards to recycling but these efforts have yet to touch the mindset of the people. Escalating amounts of recyclables that are not maximized and indifference in proper waste segregation has led to the group in developing a solution to this. This project IOT Based Intelligent Garbage Monitoring system is a newfangled system which will keep the cities clean. This system monitors the garbage bins and informs about the level of garbage collected in the garbage bins via a web page. This system uses two ultrasonic HC-SR04 sensors placed over the bins to track down the garbage level and correlate it with the garbage bin's depth. The system makes use of 8051 microcontroller to control every process and Wi-Fi modem for sending data to server. Dustbins are provided with economical implanted components which helps in tracking the level of the garbage bins and an unparallel ID will be provided for every dustbin so that it is easy to find which garbage bin is full. When the level reaches the threshold limit, the device will transmit the level along with the unique ID provided. This data can be retrieved by the person who has authority to access the data from their place with the help of Internet and an immediate action can be made to empty the dustbins.

4. **Parveen Sultana VIT University IoT garbage monitoring system IEEE 2017**

The aim is to cover all the rural and urban areas of the country to present this country as an ideal country before the world. With the proliferation of the Internet of Things (IoT) modules such as smartphones, sensors, cameras. It is possible to collect massive mount of garbage. In the metropolitan cities it is not possible to check each and every place where the garbage dump yard is full or not. So we have introduced a new concept using load cell. This is a sensor which intimates about the load placed on it. So that the garbage can also be checked in this way. Here we are using AT89S52 as our controller. A threshold value is set in the controller. Controller will monitor the status load cell. When that value is met then an intimation will be sent to the officials through IoT about the over load and also to clear the garbage as soon as possible.

5. **A Anitha Garbage monitoring system using IoT IEEE 2017**

Nowadays certain actions are taken to improve the level of cleanliness in the country. People are getting more active in doing all the things possible to clean their surroundings. Various movements are also started by the government to increase cleanliness. We will try to build a system which will notify the corporations to empty the bin on time. In this system, we will put a sensor on top of the garbage bin which will detect the total level of garbage inside it according to the total size of the bin. When the garbage will reach the maximum level, a notification will be sent to the corporation's office, and then the employees can take further actions to empty the bin. This system will help in cleaning the city in a better way. By using this system people do not have to check all the systems manually but they will get a notification when the bin will get filled.

6. **Prof. Dr. Sandeep M. Chaware1 , Shriram Dighe2 , Akshay Joshi3 , Namrata Bajare4 , Rohini Korke5 Faculty, Computer Engineering Dept, TSSM'S BSCOER, Narhe, Pune Smart Garbage Monitoring System using Internet of Things (IOT)**

The Internet of Things (IoT) shall be able to incorporate transparently and seamlessly a large number of different and heterogeneous end systems, while providing open access to selected subsets of data for the development of a plethora of digital services. Building a general architecture for the IoT is hence a very complex task, mainly because of the extremely large variety of devices, link layer technologies, and services that may be involved in such a system. One of the main concerns with our environment has been solid waste management which in addition to disturbing the balance of the environment also has adverse effects on the health of the society. The detection, monitoring and management of wastes is one of the primary problems of the present era. The traditional way of manually monitoring the wastes in waste bins is a complex, cumbersome process and utilizes more human effort, time and cost which is not compatible with the present day technologies in any way. This an advanced method in which waste management is automated. This project IoT Garbage Monitoring system is a very innovative system which will help to keep the cities clean. This system monitors the garbage bins and informs about the level of garbage collected in the garbage bins via a web page. This web page also sends all information to garbage collection vehicles.

CHAPTER 3

PROPOSED SYSTEM FUNCTION

3.1 FUNCTIONAL BLOCKS

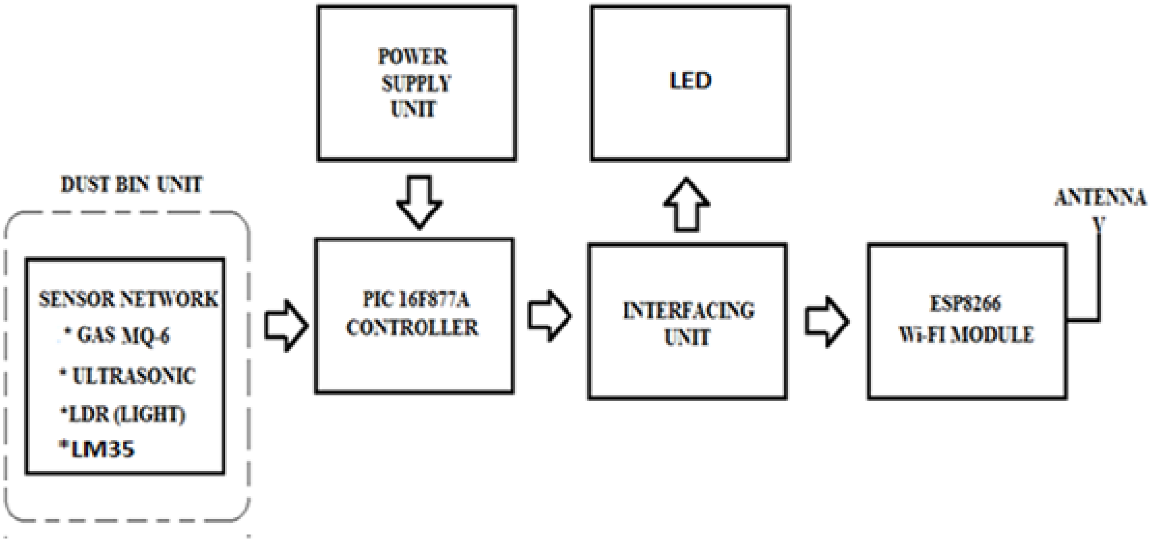


Fig: 3.1 Block diagram of the System

The system consists of gas sensor (MQ-6), ultrasonic sensor (HC-SR04), Temperature sensor (LM-35), LDR (Light Dependent Resistor), PIC16F877A microcontroller, LED Lamp, transistor (BC 548) driver, Wifi module ESP8266 and +5V/+12V power supply unit. Ultrasonic sensor is fixed at the top of the dustbin which will measure the stature of the dustbin. The threshold stature is set as equal distance according size of the dust bin. The controller will be programmed in such a way that when the dustbin is being filled, the remaining height from the threshold height will be displayed. Once the garbage reaches the threshold level sensor will trigger the Microcontroller. The microcontroller sends the information to Wi-Fi module which will continuously alert the required authority until the garbage in the dustbin is squashed. The wifi network is connected with server for accessing the data's about the dust bin throughout the world.

The light automation process also present with this system for enabling the light is ON at night time and disabling the light at day time by using LDR. The resistance value of the LDR is varied in accordance with the instantaneous value of the environment value. The LDR is connected with transistor as an input of the transistor base. The negative terminal of the LED lamp is connected as an output (Collector) of the transistor. The positive terminal of the LED lamp is connected directly with +12V power supply terminal. The High resistance (20K) value is occurred when the environment light is dark (night) condition. In this condition, the output of the transistor act as a virtual ground then LED lamp is glow. The resistance value of the LDR is decreased when the environment light is bright. In this condition, the output of the transistor is in high potential and act as a open circuit. The LED lamp is off.

The gas sensor is used to sensing the environment as well as dust bin gas level. The LM35 transistor is used to measure the temperature of the dust bin as well as environment. All the sensor output is applied to the input of the microcontroller. The controller is used to calculate the received data's with predefined universal formula. The calculated values are transmitting from the controller to web server through Wi-Fi module (ESP8266).

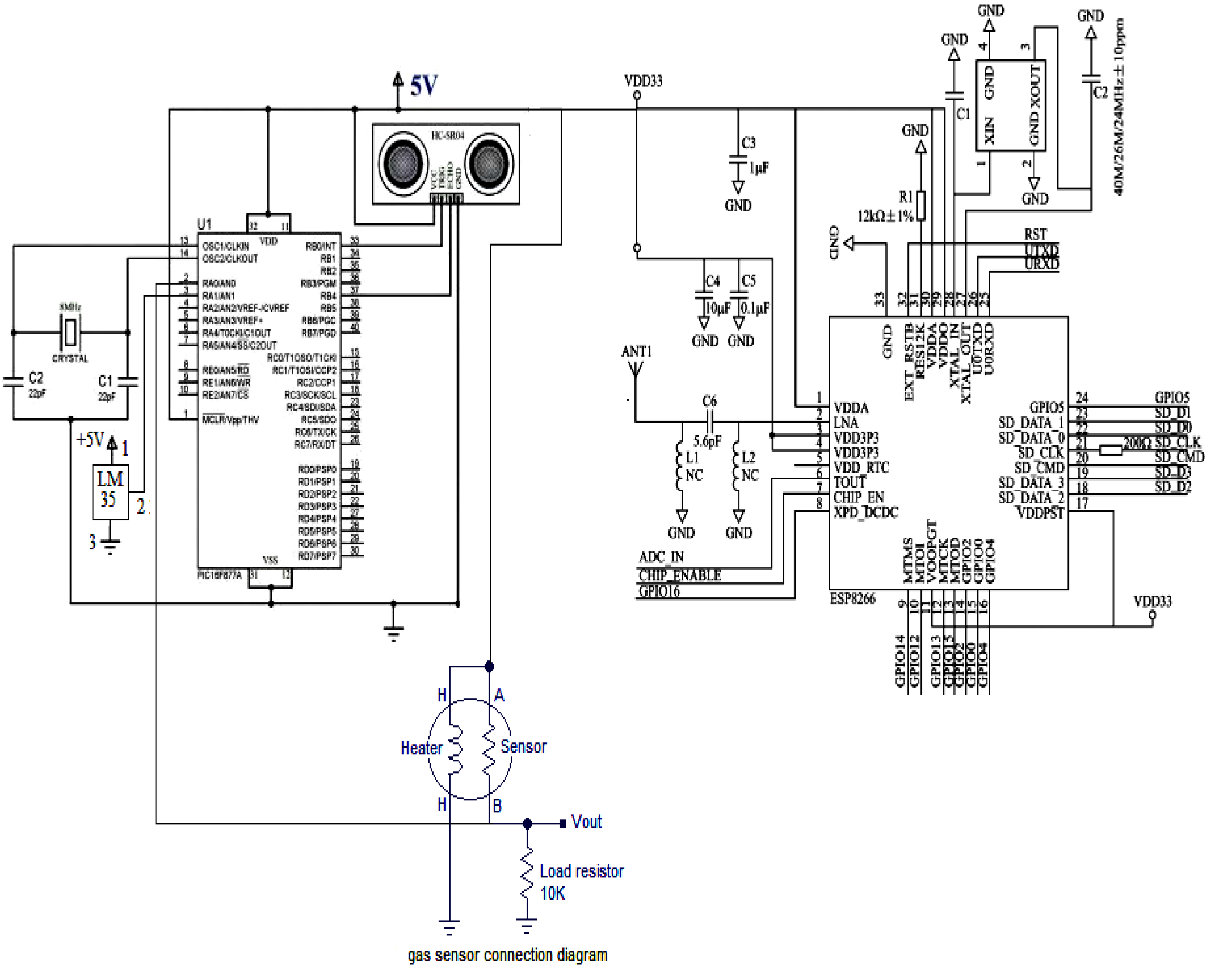


Fig: 3.2 Circuit diagram of the dust bin monitoring unit

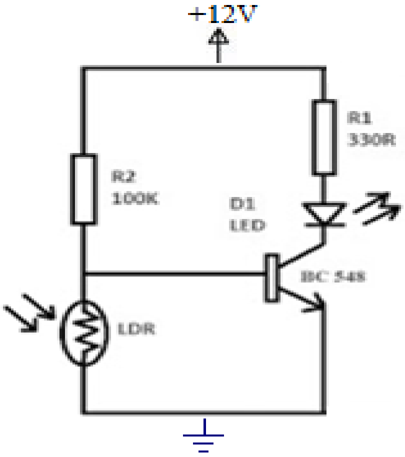


Fig: 3.3 Circuit diagram of the Light automation

3.1.1 Power supply unit:

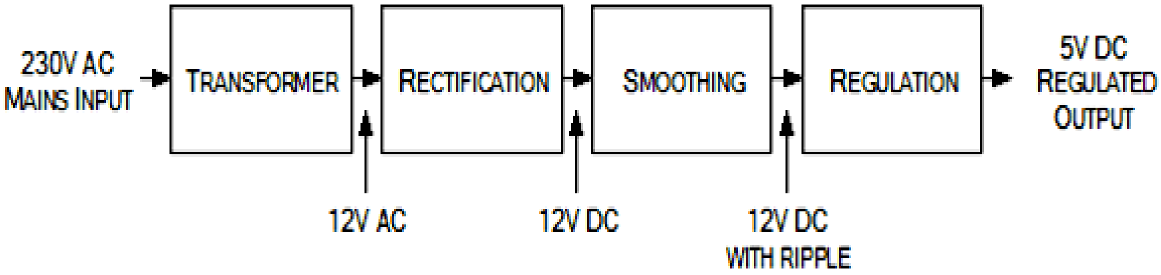


Fig. 3.4: Block diagram of the power supply unit system

The transformer used here is a step-down transformer which converts 230v AC into 12V AC. A full wave rectifier made around the diodes converts the ac supply into a pulsating dc supply. Here the rectifier consists of two 1N4001 silicon diodes which are capable of delivering current up to 1 amp.

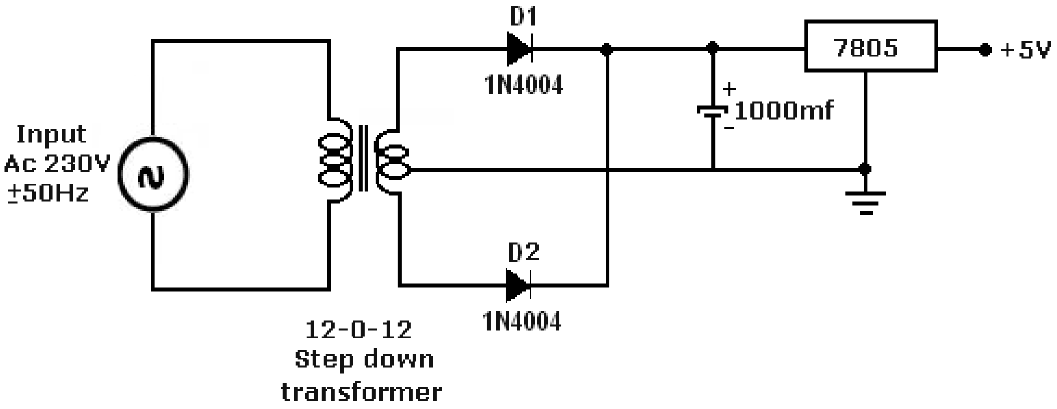


Fig. 3.5 Functional circuit diagram of power supply unit

The ripple content in the rectifier output is smoothened by adding a capacitor filter in parallel to the output. The value of capacitor may be from 1000 to 4700 microfarads. Higher the chosen value more is the filtering. The 12V dc is regulated to 5V dc using a 3-terminal series pass regulator with the input pin (pin1) to output of rectifier, output pin (pin3) to the supply output. The common pin (pin2) is connected to the supply ground. The output of the regulator will be 5volts.

CHAPTER-4
HARDWARE REQUIREMENTS

4.1 PIC MICROCONTROLLER

The PIC controller used in our project is PIC16F877A, the pin diagram of which is shown in figure. It is used to energize and de-energize the contactors during the weld and non-weld periods. The internal timer of the PIC microcontroller is used to set time delay between non-weld period and power cut off to the primary of the welding transformer.

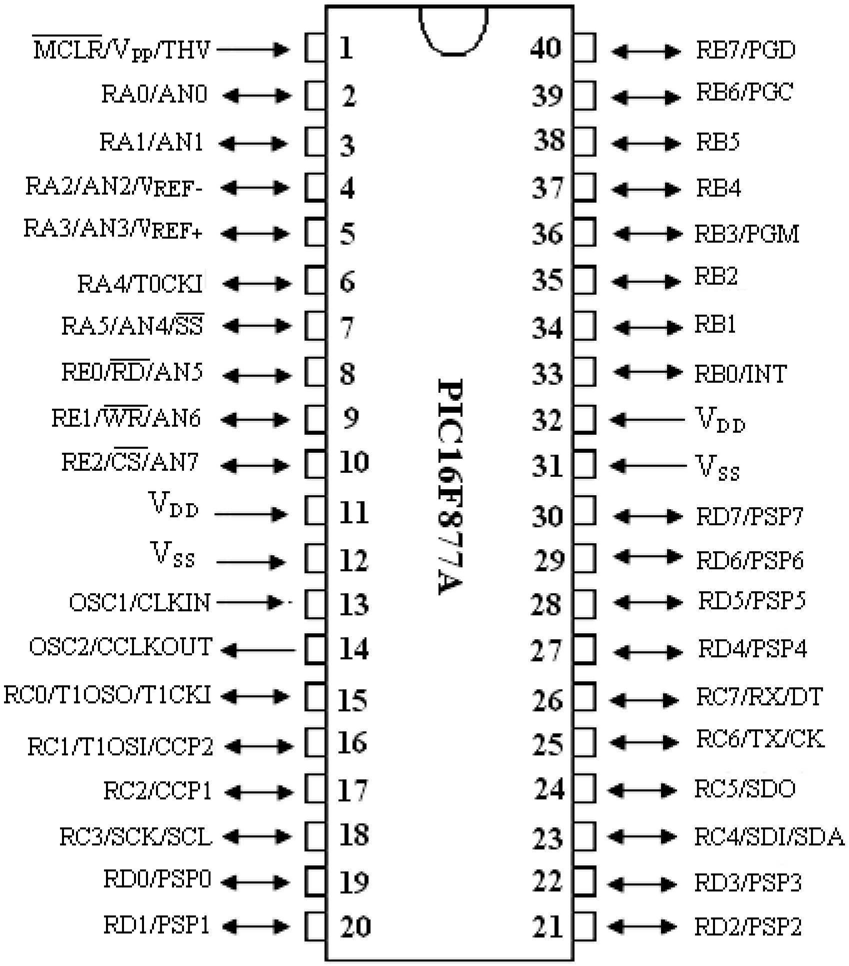


Fig: 4.1 Pin Diagram of PIC16F877A

The advantages of PIC microcontroller are as follows

i. Increased reliability through a small part count.
ii. Reduced stock levels, as one microcontroller replaces several parts.
iii. Simplified product assembly
iv. Greater product flexibility and adaptability
v. Rapid product changes or development by changing the program and not hardware.
vi. Some practical controllers

This powerful (200 nanosecond instruction execution) yet easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 40- or 44-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices. The PIC16F877A features 256 bytes of EEPROM data memory, self programming, an ICD, 2 Comparators, 8 channels of 10-bit Analog-to-Digital (A/D) converter, 2 capture/compare/PWM functions, the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I^2C™) bus and a Universal Asynchronous Receiver Transmitter (USART). All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications.

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 prescaler
- imer1: 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
- Synchronous Serial Port (SSP) with SPI
- (Master mode) and I^{2}C™ (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

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical

- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

.CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

4.2 ESP8266 Wifi Module

ESP-12E WiFi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi MAC/BB/RF/PA/LNA, on-board antenna. The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. Users can use the add modules to an existing device networking, or building a separate network controller. ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface).

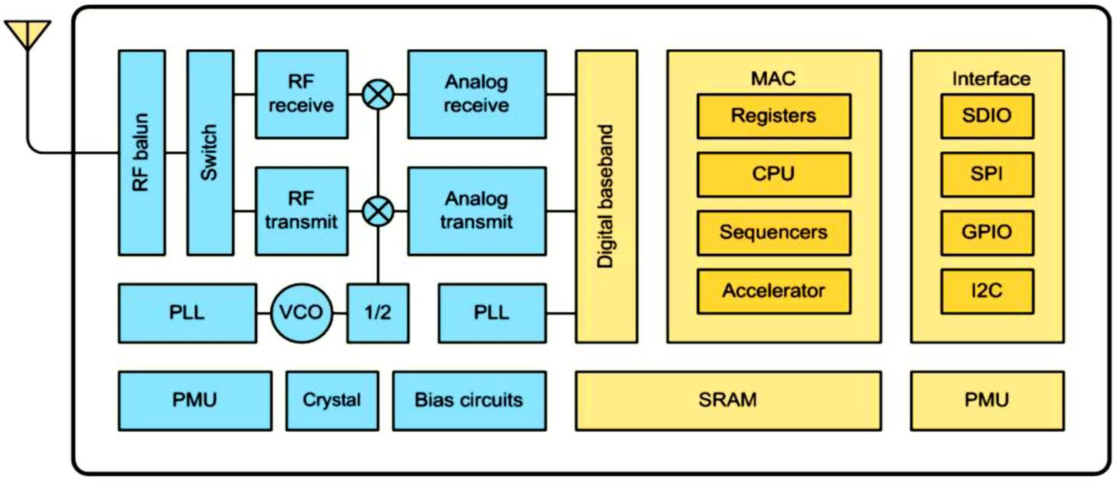


Fig: 4.2 ESP8266EX Block Diagram

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area. ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; codes for such applications are provided in examples in the SDK. Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing. for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

.

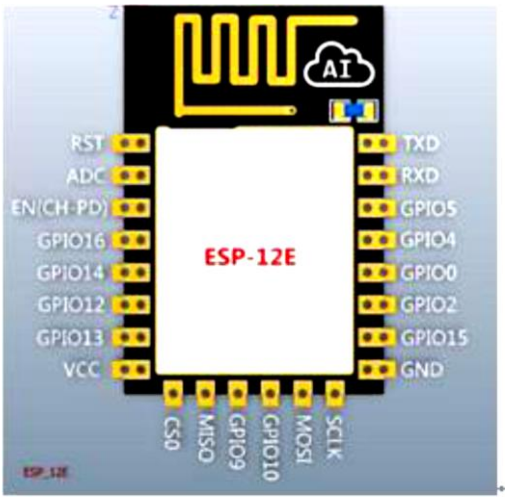


Fig: 4.3 ESP8266 Module

Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU, 10-bit ADC, TCP/IP protocol stack
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO

- A-MPDU & A-MSDU aggregation and 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C

Functional Descriptions

MCU

ESP8266EX is embedded with Tensilica L106 32-bit micro controller (MCU), which features extra low power consumption and 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. ESP8266EX is often integrated with external sensors and other specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

Memory Organization

Internal SRAM and ROM

ESP8266EX WiFiSoC is embedded with memory controller, including SRAM and ROM. MCU can visit the memory units through iBus, dBus, and AHB interfaces. All memory units can be visited upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor. According to our current version of SDK provided, SRAM space that is available to users is assigned as below:

- RAM size < 36kB, that is to say, when ESP8266EX is working under the station mode and is connected to the router, programmable space accessible to user in heap and data section is around 36kB.)

- There is no programmable ROM in the SoC, therefore, user program must be stored in an external SPI flash.

External SPI Flash

This module is mounted with an 4 MB external SPI flash to store user programs. If larger definable storage space is required, a SPI flash with larger memory size is preferred. Theoretically speaking, up to 16 MB memory capacity can be supported.

Suggested SPI Flash memory capacity:

- OTA is disabled: the minimum flash memory that can be supported is 512 kB;
- OTA is enabled: the minimum flash memory that can be supported is 1 MB.
- Several SPI modes can be supported, including Standard SPI, Dual SPI, and Quad SPI.

4.3 MQ-6 SENSOR

Fig: 4.4 Schematic of MQ-6

TABLE: I Pin details of the MQ-6

1	VCC	This pin powers the module, typically the operating voltage is +5V
2	Ground	Used to connect the module to system ground
3	Digital Out	get digital output from this pin, by setting a threshold value using the potentiometer
4	Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas

Features of MQ6 Gas sensor

- Operating Voltage is +5V
- Can be used to detect LPG or Butane gas
- Analog output voltage: 0V to 5V
- Digital Output Voltage: 0V or 5V (TTL Logic)
- Preheat duration 20 seconds
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

4.4 LM 35

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy.

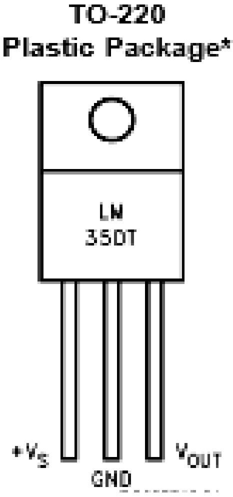


Fig: 4.5 Pin Diagram of LM35

It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features:

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming

- Operates from 4 to 30 volts
- Less than 60 µA current drain

4.5 ULTRASONIC SENSOR

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear)

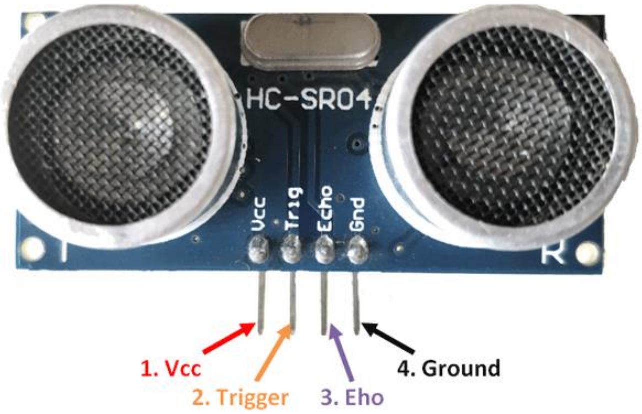


Fig: 4.6 Pin details of HC-SR04

TABLE- II pin details of HC-SR04

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.

3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

4.6 LIGHT DEPENDENT RESISTOR (LDR)

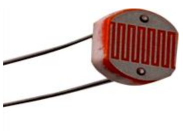
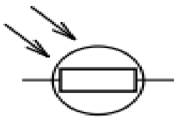


Fig.4.7: Schematic of LDR



Symbol of LDR

A photo resistor, light dependent resistor (LDR) or cadmium sulfide (CdS) cell is a resistor whose resistance decreases with increasing incident light intensity. It can also be referred to as a photoconductor.

A photo resistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance. A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, e.g. silicon. In intrinsic devices the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire band gap. Extrinsic devices have impurities, also called do pants, added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (i.e., longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconductor.

4.7TRANSFORMER

A **transformer** is a static device that transfers electrical energy from one circuit to another through inductively coupled conductors—the transformer's coils. 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.

Fig: 4.8 schematic of transformer

In an ideal transformer, the induced voltage in the secondary winding (V_s) is in proportion to the primary voltage (V_P), and is given by the ratio of the number of turns in the secondary (N_s) to the number of turns in the primary (N_P) as follows:

$$\frac{V_s}{V_P} = \frac{N_s}{N_P}$$

BASIC PRINCIPLES

The transformer is based on two principles: first, that an electric current can produce a magnetic field (electromagnetism), and, second that a changing magnetic field within a coil of wire induces a voltage across the ends of the coil (electromagnetic induction). Changing the current in the primary coil changes the magnetic flux that is developed. The changing magnetic flux induces a voltage in the secondary coil.

An ideal transformer

An ideal transformer is shown in the adjacent figure. Current passing through the primary coil creates a magnetic field. The primary and secondary coils are wrapped around a core of very high magnetic permeability, such as iron, so that most of the magnetic flux passes through both the primary and secondary coils.

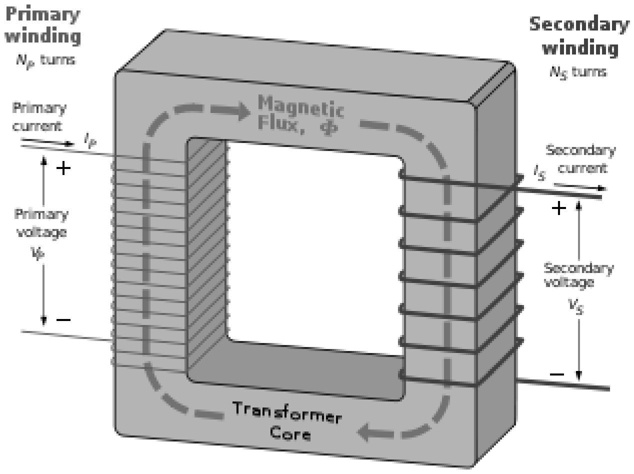


Fig: 4.9 transformer principles

Induction law

The voltage induced across the secondary coil may be calculated from Faraday's law of induction, which states that:

$$V_s = N_s \frac{\mathrm{d}\Phi}{\mathrm{d}t},$$

Where V_s is the instantaneous voltage, N_s is the number of turns in the secondary coil and Φ is the magnetic flux through one turn of the coil. If the turns of the coil are oriented perpendicular to the magnetic field lines, the flux is the product of the magnetic flux density *B* and the area *A* through which it cuts. The area is constant, being equal to the cross-sectional area of the transformer core, whereas the magnetic field varies with time according to the excitation of the primary. Taking the ratio of the two equations for V_s and V_p gives the basic equation for stepping up or stepping down the voltage

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}.$$

N_p/N_s is known as the *turns ratio*, and is the primary functional characteristic of any transformer. In the case of step-up transformers, this may sometimes be stated as the reciprocal, N_s/N_p. *Turns ratio* is commonly expressed as an irreducible fraction or ratio: for example, a transformer with primary and secondary windings of, respectively, 100 and 150 turns is said to have a turns ratio of 2:3 rather than 0.667 or 100:150.

CHAPTER-5
SOFTWARE DETAILS

5.1 EMBEDDED ‘C’

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C use most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch. case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, unions ,etc. As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the **embedded programming language** of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.

Initially C was developed by Kernighan and Ritchie to fit into the space of 8K and to write (portable) operating systems. Originally it was implemented on UNIX operating systems. As it was intended for operating systems development, it can manipulate memory addresses. Also, it allowed programmers to write very compact codes. This has given it the reputation as the language of choice for hackers too. As assembly language programs are specific to a processor, assembly language didn't offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn't find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Even though C might have lost its sheen as mainstream language for general purpose applications, it still is having a strong-hold in embedded programming. Due to the wide acceptance of **C in the embedded systems**, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of **embedded systems using C**. Subsequent sections will discuss **what is Embedded C**, **features of C language**, similarities and **difference between C and embedded C**, and **features of embedded C programming**.

5.1.1 Embedded Systems Programming

- Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows:
- Embedded devices have resource constraints (limited ROM, limited RAM, and limited stack space, less processing power).
- Components used in embedded system and PCs are different; embedded systems typically uses smaller, less power consuming components.
- Embedded systems are more tied to the hardware.

Two salient **features of Embedded Programming** are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of languages:

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++, Java, Ada, etc.
- Application level language like Visual Basic, scripts, Access, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

Use of **C in embedded systems** is driven by following advantages

- It is small and reasonably simpler to learn, understand, program and debug.
- C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
- Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.
- As C combines functionality of assembly language and features of high level languages, C is treated as a ‘middle-level computer language’ or ‘high level assembly language’
- It is fairly efficient
- It supports access to I/O and provides ease of management of large embedded projects.

Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high level languages. Compared to other high level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation.

Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms (with some changes). Moreover, programs developed in C are much easier to understand, maintain and debug. Also, as they can be developed more quickly, codes written in C offers better productivity. C is based on the philosophy 'programmers know what they are doing'; only the intentions are to be stated explicitly. It is easier to write good code in C & convert it to an efficient assembly code (using high quality compilers) rather than writing an efficient code in assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.

Dynamic C and B# are some proprietary languages which are also being used in embedded applications. Efficient embedded C programs must be kept small and efficient; they must be optimized for code speed and code size. Good understanding of processor architecture embedded C programming and debugging tools facilitate this.

5.1.2 Differentiation C and Embedded C

Though **C and embedded C** appear different and are used in different contexts, they have more similarities than the differences. Most of the constructs are same; the difference lies in their applications. C is used for desktop computers, while **embedded C** is for microcontroller based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory. However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code exceeds the limit, the system is likely to crash.

Compilers for C (ANSI C) typically generate OS dependant executables. **Embedded C** requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications. Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications.

Embedded systems often do not have a console, which is available in case of desktop applications. So, what basically is different while programming with **embedded C** is the mindset; for embedded applications, we need to optimally use the resources, make the program code efficient, and satisfy real time constraints, if any. All this is done using the basic constructs, syntaxes, and function libraries of ‘C’.

5.2 PIC C Compiler

This C compiler, is fully optimised for use with PIC microcontrollers. Built in functions make coding the software very easy. Based on original K&R, the integrated C development environment gives developers a fast method to produce efficient code from an easily maintainable high level language.

CAPABILITIES

- Arrays up to 5 subscripts
- Structures and Unions may be nested.
- Custom bit fields (1-8 bits) within structures.
- ENUMurated types,
- CONSTant variables, arrays and strings.
- Full function parameter support (any number).
- Some support for C++ reference parameters.

This integrated C development environment gives developers the capability to quickly produce very efficient code from an easily maintainable high level language.

The compiler includes built in functions to access the PIC hardware such as READ_ADC to read a value from the A/D converter. Discrete I/O is handled by describing the port characteristics in a PRAGMA. Functions such as INPUT and OUTPUT_HIGH will properly maintain the tri-state registers. Variables including structures may be directly mapped to memory such as I/O ports to best represent the hardware structure in C. The microcontroller clock speed may be specified in a PRAGMA to permit built in functions to delay for a given number of microseconds or milliseconds. Serial I/O functions allow standard functions such as GETC and PRINTF to be used for RS-232 like I/O. The hardware serial transceiver is used for applicable parts when possible. For all other cases a software serial transceiver is generated by the compiler. The standard C operators and the special built in functions are optimized to produce very efficient code for the bit and I/O functions. Functions may be implemented inline or separate.

Function parameters are passed in reusable registers. Inline functions with reference parameters are implemented efficiently with no memory overhead. During the linking process the program structure including the call tree is analyzed. Functions that call one another frequently are grouped together in the same page. Calls across pages are handled automatically by the tool transparent to the user. Functions may be implemented inline or separate. RAM is allocated efficiently by using the call tree to determine how locations can be re-used. Constant strings and tables are saved in the device ROM. The output hex and debug files are selectable and compatible with popular emulators & programmers including MPLAB for source level debugging. The Professional Package (PCW) provides both compilers in a powerful Windows environment.

5.3 DESIGN IMPLEMENTATION

Implementation refers to the final process of moving the solution from development status to production status. Depending on your project, this process is often called deployment, go-live, rollout or installation. For the purposes of life cycle step, all of these terms are synonymous with "implementation." There is no single way to implement an application. It depends on the characteristics of the project and the solution. Some implementations are as easy as saying “we are now live.” This type of implementation can work when the solution is brand new and you are developing and testing in what will become the production environment. In these cases, implementation is just a state of mind. One day the solution is in development, and the next day it is in production. At the other extreme are implementations that might be projects within themselves. For instance, there may be a software application that needs to be deployed to your division offices all around the world. This could take months to accomplish and require a full lifecycle of planning, analysis, design, etc. In this case, the implementation can be structured as a separate project. If the implementation is relatively straightforward, then there is no reason for elaborate implementation processes. However, most projects have a number of implementation events to plan for and execute successfully. Often, a smoothly run project gets a black eye because of problems in implementation. Planning for implementation cannot start at the same time the system implementation gets started.

5.4 SOFTWARE REQUIREMENTS

Operating System	:	Windows Family
Tool	:	Wamp Server
Testing Tool	:	Selenium
Language (Front End)	:	PHP
Language (Back End)	:	My SQL

5.5 SOFTWARE SPECIFICATION

5.5.1 Wamp Server:

Wamp Server refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language.

The acronym WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL handles the database components, while PHP, Python, or PERL represents the dynamic scripting languages. The word WAMP is the abbreviation of the package: Apache, MySQL, and one of Perl, PHP, or Python in Microsoft Windows operating system. It is a kind of open source software that can be used to build dynamic websites or servers. They are all independently-created programs, however, it has increasingly high compatibility degree because they often be put together to use. Therefore, they formed a powerful web applications platform. Furthermore, what is the package of Apache, MySQL and Perl, PHP or Python? Apache HTTP server is web server software which ranked the first over the world. The pronunciation of “Apache” is from the word “A patchy server”, which means that the program has patch.

There are continuously people to develop its new function and modify the original defects by its open community. Initially, Apache is just used in small or text internet work. Then, it gradually expanded to all sort of UNIX system, especially for Linux system, the Apache support it pretty perfect. Apache has a variety of products. It can support both SSL technologies and multiple virtual hosts. PHP stands for Hypertext Preprocessor. It is a kind of HTML embedded language which executed on the server. PHP mixed with C language, java, Perl and PHP self-created language. And the language style is similar to C language. PHP is embedded into the procedure in HTML documents to execute. It can also perform the compiled code. This kind of compiler can achieve encryption and optimizing code running in order to make code running faster. PHP has very powerful function and support almost all popular databases and operating systems.

5.5.2 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by RasmusLerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP: Hypertext Preprocessor. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

5.5.3 MY SQL

MySQL , in July 2013, it was the world's second most[a] widely used RDBMS, and the most widely used open-source client–server model RDBMS. The SQL acronym stands for Structured Query Language. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used WAMP open-source web application software stack.

5.5.4 SQLBuddy

SQLBuddy is an open source web based application written in PHP intended to handle the administration of MySQL and SQLite with the use of a Web browser. The project places an emphasis on ease of installation and a simple user interface.

SQL Buddy offers a comprehensive feature set, and supports SQLite as well as MySQL databases. The tool allows you to create, modify, or drop tables, indexes, foreign key relationships, and records. Backups are easy and you can run ad hoc SQL queries.SQL Buddy has a fast and attractive Ajax-powered interface with multiple languages and themes. The download is only 320kB (1.1MB extracted) and requires no installation—simply copy the files to your server and log in with a database user ID and password. Overall, SQL Buddy beats phpMyAdmin on many levels. It’s more than an alternative; it may even be a better option.

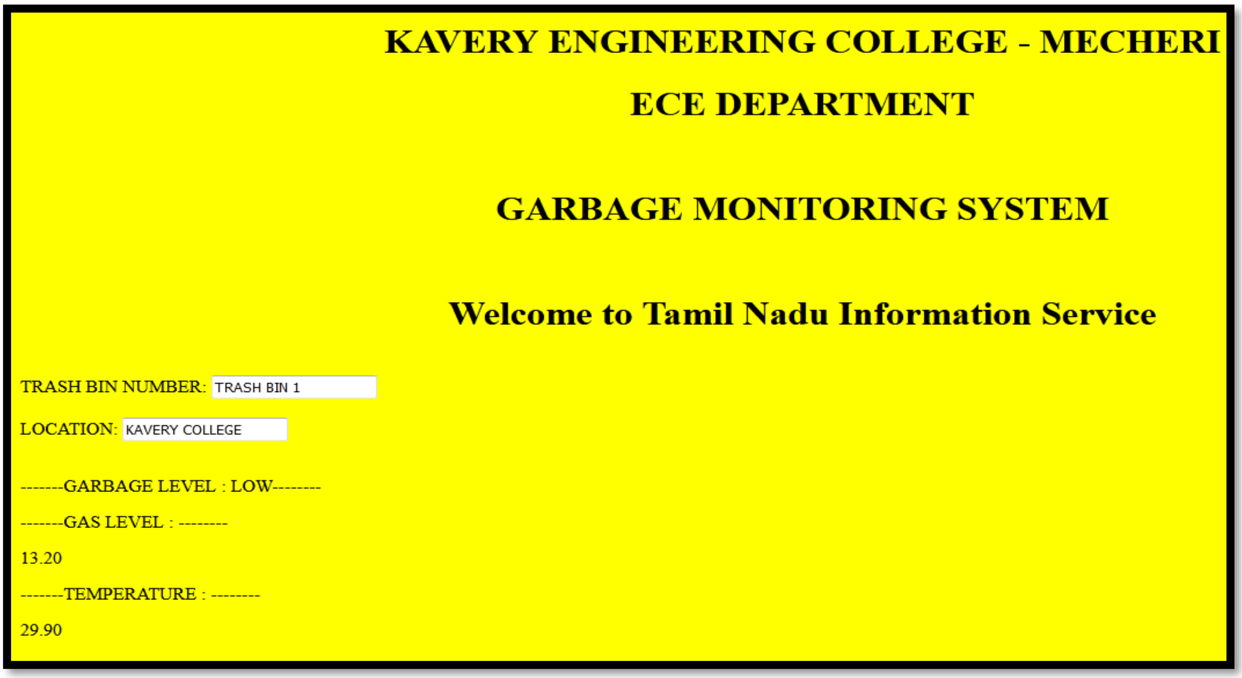
5.6 SOURCE CODING

```
#include <16F877A.h>
#device adc = 10  // Configura ADC para 10 Bits
#FUSES NOWDT                    //No Watch Dog Timer
#FUSES XT                       //Crystal osc <= 4mhz for PCM/PCH , 3mhz to 10 mhz for
PCD
#FUSES NOPUT                    //No Power Up Timer
#FUSES NOPROTECT                //Code not protected from reading
#FUSES NODEBUG                  //No Debug mode for ICD
#FUSES NOBROWNOUT               //No brownout reset
#FUSES NOLVP                    //No low voltage prgming, B3(PIC16) or B5(PIC18)
used for I/O
#FUSES NOCPD                    //No EE protection
#FUSES NOWRT                    //Program memory not write protected

#use delay(clock = 4000000) // Configura clock do PIC
// Configura comunicação serial para 9600,8,N,1
#use rs232(baud = 9600, parity = N, xmit = PIN_C6, rcv = PIN_C7, bits = 8)

#define trig pin_C0
#define echo pin_C1

#define Buzzer PIN_B2

float digital_reading, digital_reading2 ;
int16 distance, time;          // Defining variables

void main() {

  output_low(Buzzer);
  setup_adc(ADC_CLOCK_INTERNAL); // initialize ADC with a sampling rate of Crystal/4 MHz
  setup_adc_ports(ALL_ANALOG);   //set pins AN0-AN7 to analog
  setup_psp(PSP_DISABLED);
  setup_spi(SPI_SS_DISABLED);
  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
  setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);     // initiating timer
  while(TRUE) {
  set_adc_channel(1); //temp
  delay_ms(2);    // ADC module is slow, needs some time to adjust.
  digital_reading = read_adc();
  delay_us(100);                  // 0.1ms delay for ADC stabilization
  digital_reading = digital_reading/10;
  delay_us(100);
  set_adc_channel(2); //temp
  delay_ms(2);    // ADC module is slow, needs some time to adjust.
  digital_reading2 = read_adc();
  delay_us(100);                  // 0.1ms delay for ADC stabilization
  digital_reading2 = digital_reading2/10;
  delay_us(100);
  output_high(trig);                       // ping the sonar
  delay_us(20);                         // sending 20us pulse
  output_low(trig);
  while(!input(ECHO))                       // wait for high state of echo pin
  {}
  set_timer1(0);                          // setting timer zero
  while(input(ECHO))                       // Wait for high state of echo pin
 {}
```

```
  time=get_timer1();                    // Getting the time
  distance=time*0.028 + 1.093 ;            // Calculating the distance
  delay_ms(2000);
  if(distance>=4)
{
    output_low(Buzzer);
    printf("%2.2f", digital_reading2);
    printf("\n");
    delay_ms(1000);
    printf("%2.2f", digital_reading);
    printf("\t");
    delay_ms(1000);
    printf("*1111\r"); // Low
}
if(distance==3)
{
    output_low(Buzzer);
    printf("%2.2f", digital_reading2);
    printf("\n");
    delay_ms(1000);
    printf("%2.2f", digital_reading);
    printf("\t");
    delay_ms(1000);
    printf("*2222\r"); // medium
}
if(distance==2)
{
   output_low(Buzzer);
   printf("%2.2f", digital_reading2);
   printf("\n");
   delay_ms(1000);
   printf("%2.2f", digital_reading);
   printf("\t");
   delay_ms(1000);
   printf("*3333\r"); // almost full
}
if(distance<=1)
{
    output_high(Buzzer);
    printf("%2.2f", digital_reading2);
    printf("\n");
    delay_ms(1000);
    printf("%2.2f", digital_reading);
    printf("\t");
    delay_ms(1000);
    printf("*4444\r"); // overflow
}  }}
```

CHAPTER-6

EXPERIMENTAL RESULTS



KAVERY ENGINEERING COLLEGE - MECHERI

ECE DEPARTMENT

GARBAGE MONITORING SYSTEM

Welcome to Tamil Nadu Information Service

TRASH BIN NUMBER: TRASH BIN 1

LOCATION: KAVERY COLLEGE

-------GARBAGE LEVEL : LOW--------

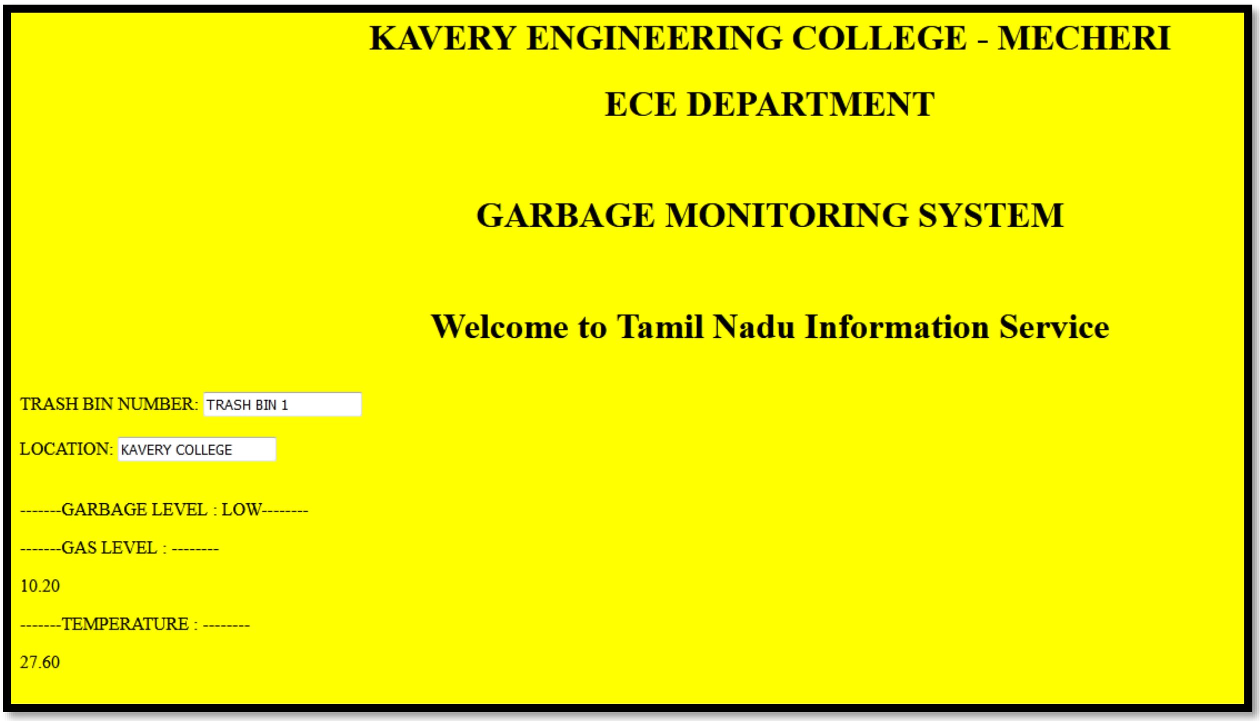
-------GAS LEVEL : --------

13.20

-------TEMPERATURE : --------

29.90

Fig: 6.1 Web page design



KAVERY ENGINEERING COLLEGE - MECHERI

ECE DEPARTMENT

GARBAGE MONITORING SYSTEM

Welcome to Tamil Nadu Information Service

TRASH BIN NUMBER: TRASH BIN 1

LOCATION: KAVERY COLLEGE

-------GARBAGE LEVEL : LOW--------

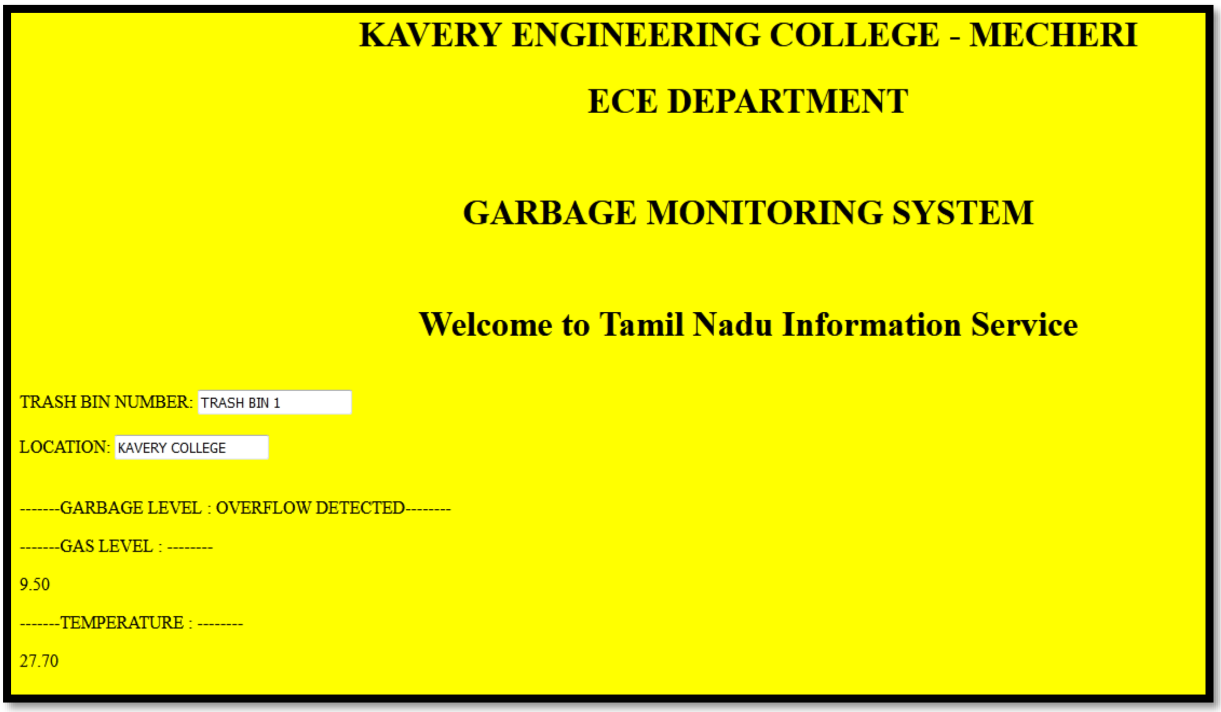
-------GAS LEVEL : --------

10.20

-------TEMPERATURE : --------

27.60

Fig: 6.2 dust bin low level



KAVERY ENGINEERING COLLEGE - MECHERI

ECE DEPARTMENT

GARBAGE MONITORING SYSTEM

Welcome to Tamil Nadu Information Service

TRASH BIN NUMBER: TRASH BIN 1

LOCATION: KAVERY COLLEGE

-------GARBAGE LEVEL : OVERFLOW DETECTED--------

-------GAS LEVEL : --------

9.50

-------TEMPERATURE : --------

27.70

Fig: 6.3 dust bin overflow detected

CHAPTER 7

CONCLUSION

The proposed system uses a more powerful and efficient microcontroller and the data is uploaded to the cloud only when leftover space and garbage weight in the garbage container cross thresholds. The collected data is stored in the cloud and can be used for predicting the waste generation. The automated execution of commands and triggering helps in providing exact values of the garbage levels resulting in efficient garbage collection and monitor.

7.1 FUTURE ENHANCEMENT

The system will develop the recycling (burning process for converting the waste into heat energy i.e., power generator or fertilizer) the wastage of the dust within the unit.

CHAPTER 8

REFERENCES

1. Paavan Lakshmana Chowdary S, Sai Teja G, Naga Mahesh K Department of Electronics and Communication Engineering National Institute of Technology, Andhra Pradesh An IoT based Smart Garbage Alert System IEEE-2019
2. Pablo Velásquez Department of Power and machinery University of Concepción Chillán, Chile A low-cost IoT based Environmental Monitoring System. A citizen approach to pollution awareness IEEE-2018
3. Eveneet Johar, Rahul mishra Atharva college of engineering, Mumbai IoT based intelligent garbage monitoring system IEEE 2019

4. Parveen Sultana VIT University IoT garbage monitoring system IEEE 2017

5. A Anitha Garbage monitoring system using IoT IEEE 2017

6. Prof. Dr. Sandeep M. Chaware1 , Shriram Dighe2 , Akshay Joshi3 , Namrata Bajare4 , Rohini Korke5 Faculty, Computer Engineering Dept, TSSM’S BSCOER, Narhe, Pune Smart Garbage Monitoring System using Internet of Things (IOT)

7. G. K. Shyam, S. S. Manvi and P. Bharti, "Smart waste management using Internet-of-Things (IoT)," 2017 2nd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 2017, pp. 199-203

8. Gopal Kirshna Shyam, Sunilkumar S. Manvi, Priyanka Bharti, “Smart waste management using Internet-of-Things (IoT),” 2017 2nd International Conference on Computing and Communications Technologies (ICCCT), pp.199–203, 2017.