ABSTRACT

The uncollected waste material when the waste bin is full is a common problem nowadays. Thus, an efficient waste management for the waste material is essential in ensuring a clean and green surrounding environment. This paper presents an Internet of Things (IoT) based Smart Waste Collection Monitoring and Alert System to monitor the waste material at the selected site of garbage collection area. The system is implemented using an ultrasonic sensor which is connected to Arduino UNO as to monitor waste bin garbage level. In this system, waste bin depth level will be sent via Arduino Ethernet Shield with an Internet connection to the Ubidots IoT Cloud. The Ubidots store the collected waste bin level data into IoT database and display the waste bin depth level on online dashboard for real-time visualization. The Ubidots Event manager invoke a notification alert to garbage collector mobile phone via a SMS when the waste bin is nearly filled for immediate waste collection. Therefore, the waste collection became more effective and systematic.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATION

S.NO	ABBREVATION	EXPANSION
1.	**IOT**	Internet Of Things
2.	**IDE**	Integrated Development Environment
3.	**RFID**	Radio Frequency Identification
4.	**IP**	Internet Protocol
5.	**ICSP**	In Circuit Serial Programming
6	**USB**	Universal Serial Bus
7	**OTP**	One Time Programmable
8	**PCB**	Pitch Circuit Board
9	**Wi-Fi**	Wireless Fidelity
10	**TTL**	Transistor-Transistor Logic
11	**ESP**	Espressif Modules
12	**LED**	Light Emitting Diode
13	**WPA**	Wi-Fi Protected Access
14	**COM**	Common Port
15	**MAC**	Media Access Control
16	**APIs**	Application Program Interface’s
17	**LCD**	Liquid Crystal Display

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

At present, most of the cities around the world require challenging solutions for solid waste management, as there is rapid growth in residential areas and the economy. Municipal authorities have inadequate resources for waste management institutions to effectively collect the waste generated. It becomes an excessive wastage of resources when bins are collected that are filled up partially. The IoT based garbage monitoring system is a very innovative system which will help to keep the environment and cities clean. This system monitors the garbage bins throughout the city and informs about the level of garbage collected in the garbage bins to a person in the administrative department. For number of times we have seen that the dustbins are being overflown with the waste materials and the concern person don't have any information about it within the time, due to which unsanitary conditions are formed in the surroundings environment and living area. Bad smell is out due to waste in dustbin at the same time. Also, the bad look of the city which leads to air and environmental pollution and to some harmful infections and diseases around the locality which is spreadable easily. There are number of unwanted manual checks of garbage bin's level by municipal corporations which is less effective and time consuming. Trucks are sent to empty the dustbins whether they are full or not. And the trucks need fuel which is costly. Several sensing methods have been integrated and have combined their verdicts that offer the detection of bin condition and its parameter measurement. Though results and developed algorithms are efficient for automatic bin status monitoring work lacks remote monitoring of bin. So, in this paper we have proposed system which can be deployed in general purpose dust bins placed at public places and which allows us to monitor its status remotely over web browser for efficient waste management. Due to drastic increase in economic and population growth in the nation there is huge development in the generation of the solid waste. Solid waste management is a main problem of surroundings in the whole globe. SWM is a huge problem not only in urban cities of India but in most of the nations in the globe. There is a requirement to evolve an effective system which will resolve this issue or decrease it to some level. It will support them to maintain their surroundings green and clean in an effective way.

1.2 PROBLEM SPECIFICATION

EXISTING SYSTEM:

- In existing method, the dustbin is not monitored continuously
- There is no wireless technology is available for monitoring.

DISADVANTAGES OF EXISTING SYSTEM

- Difficult to find the dustbin is full or not remotely.

PROPOSED SYSTEM

- In our proposed system we are going to monitor the dustbin in real time and update the status of the dustbin

ADVANTAGES OF PROPOSED SYSTEM

- Fast response
- Person nearby is detected.
- One time installation

1.3 METHODOLOGIES

This section describes the conceptual framework and the methodology adopted for this work including the systematically organized different stages of the research in conjunction with the detailed implementation features of the proposed system. In addition, it clarifies the structural components of the proposed system and their integration to achieve the research aim. The flowchart inThis section describes the conceptual framework and the methodology adopted for this work including the systematically organized different stages of the research in conjunction with the detailed implementation features of the proposed system. In addition, it clarifies the structural components of the proposed system and their integration to achieve the research aim. The flowchart in This section describes the methodology adopted for this project IOT is responsible for connecting each and every network with a common controller.. The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network After that, the wiring and hardware implementation is carried out. Then the programming phase of Arduino Software is accomplished. For design enhancement and optimization, any problem that occurs during building the project, is identified and solved during testing phase. Some improvement is also done to avoid the same error. Finally, the complete design of the project is evaluated.

1.4 CONTRIBUTIONS

Smart Waste collection is very popular due to its numerous benefits in promising area, these techniques will controls all the electronic devices which will reduce the human involvement to get minimize. It will providevarious benefits such as greater safety, comfort, and security, a more rational use of energy and other resources thus contributing to a significant savings. This research application domain is very important and it will implement in future as it offers very powerful means for supporting and helping special needs of the elderly and people with disabilities for monitoring and control of home appliances. There are a number of factors that needs to be considered when designing a smart smart waste collection monitoring and alert system via IOT.

1.5 LAYOUT OF THE THESIS

In the chapter 1, we came to know the objectives of our project, problem specification, methodologies, Modules and Contribution. We got the idea of our project and came to know the entire description of the project.
A literature survey or a literature review is a project report is that section which shows the various analysis and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the intent of the project.
In the third chapter i.e., Requirement Specification, we describe about existing system along with its drawbacks and also the proposed system with its advantages.
In System Design, our deep explanation is about System architecture, UML diagram, Class Diagram, Object Diagram, State Diagram, Activity Diagram, Sequence Diagram, Collaboration Diagram, Data Flow Diagram, E-R diagrams and Modules along with their brief description.
In chapter 5, Implementation, we have discussed about Development Tools, given the Sample of Code that helps to work our project. I.e., for facial expression recognition, Technique for deep neural network models, Snapshots, Various Types of Testing and finally Test Cases.
In the last chapter, we discuss about the Conclusion and Future Enhancements.

CHAPTER – 2

LITERATURE REVIEW

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 paper, smart bin is built on a microcontroller-based platform Arduino Mega board which is interfaced with GSM modem and Ultrasonic sensor. Now a day's Automatic systems are being preferred over manual system to make life simpler and easier in all aspects. The number of users of internet has grown so rapidly that it has become a necessary part of our daily life. Dust bins placed across cities set at open placesare flooding because of increment in the waste each day andmaking unhygienic condition for the citizens, we have proposedwaste management system for smart cities whichallows municipal corporations to monitor status of dustbinsremotely over web server and keep cities clean very efficiently byoptimizing cost and time required for it. As soon as dustbin hasreached its maximum level, waste management department getsalert via SMS via GSM module placed at dustbin, so department cansend waste collector vehicle to respective location to collectgarbage.[1]

In most of the cities the overflowed garbage dumpsters are creating an obnoxious smell and making an unhygienic environment. The Collection of garbage is a very much needed municipal service that requires huge expenditures and execution of this operation is high-priced. The high pricing is due to the various factors such as man power, navigation of vehicles, fuel, maintenances and environmental costs. The above factor necessitates the design, implementation and execution of the new Smart Intelligent Garbage Alert System (SIGAS) for the smart cities. This paper focuses on the implementation of an IoT based embedded system which integrates various Sensors & controllers with RF transmitter and receiver for dumpster and vehicle monitoring system with their performance measured in real time environment.[2]

The uncollected waste material when the waste bin is full is a common problem nowadays. Thus, an efficient waste management for the waste material is essential in ensuring a clean and green surrounding environment. This paper presents an Internet of Things (IoT) based Smart Waste Collection Monitoring and Alert System to monitor the waste material at the selected site of garbage collection area. The system is implemented using an ultrasonic sensor which is connected to Arduino UNO as to monitor waste bin garbage level. In this system, waste bin depth level will be sent via Arduino Ethernet Shield with an Internet connection to the Ubidots IoT Cloud. The Ubidots store the collected waste bin level data into IoT database and display the waste bin depth level on online dashboard for real-time visualization. The Ubidots Event manager invoke a notification alert to garbage collector mobile phone via a

SMS when the waste bin is nearly filled for immediate waste collection. Therefore, the waste collection became more effective and systematic.[3]

Indiscriminate disposal of solid waste is a major issue in urban centers and it poses a serious threat to healthy living of the citizens. Timely access to reliable information on the level of solid waste at different locations within the city will help both the local authorities and the citizens to effectively manage this menace. In this paper, an intelligent solid waste monitoring system is developed using Internet of Things (IoT) and cloud computing technologies. Waste containers are strategically situated within the communities and the fill level of solid waste in each of the containers is detected using ultrasonic sensors. The sensor data is transmitted to an IoT cloud platform, ThingSpeak, via a Wireless Fidelity (Wi-Fi) communication link. At different fill levels, the system is designed to send appropriate notification message (in form of tweet) to alert relevant authorities and concerned citizen(s) for necessary action. Also, the fill level is monitored on ThingSpeak in real-time. The system performance shows that the proposed solution may be found useful for efficient waste management in smart and connected communities.[4]

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 GENERAL INTRODUCTION TO EMBEDDED SYSTEM

Embedded systems are designed to do some specific task rather than be a general purpose computer for multiple tasks. Some also have real time performance constraints that must met, for reason such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

An embedded system is not always a separate block very often it is physically built in to the device it is controlling. The software written for embedded systems is often called firmware, and is stored in read only memory or flash convector chips rather than a disk drive. It often runs with limited computer hardware resources: small or no keyboard, screen and little memory.

To perform any application in the embedded system we require microprocessor and microcontroller. In the microprocessor an external memory is connected which increases the size of the microprocessor and multiple operations are being performed by the microprocessor but whereas in the microprocessor the memory is inbuilt and also we can use this controller only for the specific applications where the speed is increased so most probably microcontrollers are used in the different applications in the embedded systems rather than microprocessor.

An embedded system can be defined as a computing device that does a specific focused job. Appliances such as the air-conditioner, VCD player, DVD player, printer, fax machine, mobile phone etc. are examples of embedded systems. Each of these appliances will have a processor and special hardware to meet the specific requirement of the application along with the embedded software that is executed by the processor for meeting that specific requirement. The embedded software is also called “firm ware”. The desktop/laptop computer is a general purpose computer. You can use it for a variety of applications such as playing games, word processing, accounting, software development and so on. In contrast, the software in the embedded systems is always fixed listed below.Embedded systems do a very specific task; they cannot be programmed to do different things. . Embedded systems have very limited resources, particularly the memory. Generally, they do not have secondary storage devices such as the CDROM or the floppy disk. Embedded systems have to work against some deadlines. A specific job has to be completed within a specific time. In some embedded systems, called real-time systems, the deadlines are stringent. Missing a deadline may cause a catastrophe-loss of life or damage to property. Embedded systems are constrained for power. As many embedded systems operate through a battery, the power consumption has to be very low. Some embedded systems have to operate in extreme environmental conditions such as very high temperatures and humidity.

3.2HARDWARE REQURIEMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by engineers as the starting point for the system design. It should what the system do and not how it should be implemented. Based on this IOT project there are many hardware components. They are mentioned below

Microcontroller

A Microcontroller (or MCU) is a computer-on-a-chip used to control electronicdevices. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor (the kind used in a PC). A typical microcontroller contains all the memory and interfaces needed for a simple application, whereas a general purpose microprocessor requires additional chips to provide these functions.

A microcontroller is a single integrated circuit with the following key features:

- central processing unit - ranging from small and simple 8-bit processors to sophisticated 32- or 64-bit processors
- input/outputinterfaces such as serial ports
- RAM for data storage
- ROM, EEPROM or Flash memory for program storage
- clock generator - often an oscillator for a quartz timing crystal, resonator or RC circuit

Microcontrollers are inside many kinds of electronic equipment (see embedded system). They are the vast majority of all processor chips sold. Over 50% are "simple" controllers, and another 20% are more specialized digital signal processors (DSPs) (ref?). A typical home in a developed country is likely to have only one or two general-purpose microprocessors but somewhere between one and two dozen microcontrollers. A typical mid range vehicle has as many as 50 or more microcontrollers. They can also be found in almost any electricaldevice: washing machines, microwave ovens, telephones etc.

Arduino uno

Arduino/genuino uno is a microcontroller board based on the atmega328p (datasheet). It has 14 digital input/output pins (of which 6 can be used as pwm outputs), 6 analog inputs, a 16 mhz quartz crystal, a usb connection, a power jack, an icsp header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a usb cable or power it with a ac-to-dc adapter or battery to get started.. You can tinker with your uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in italian and was chosen to mark the release of arduino software (ide) 1.0. The uno board and version 1.0 of arduino software (ide) were the reference versions of arduino, now evolved to newer releases. The uno board is the first in a series of usbarduino boards, and the reference model for the arduino platform; for an extensive list of current, past or outdated boards see the arduino index of boards.

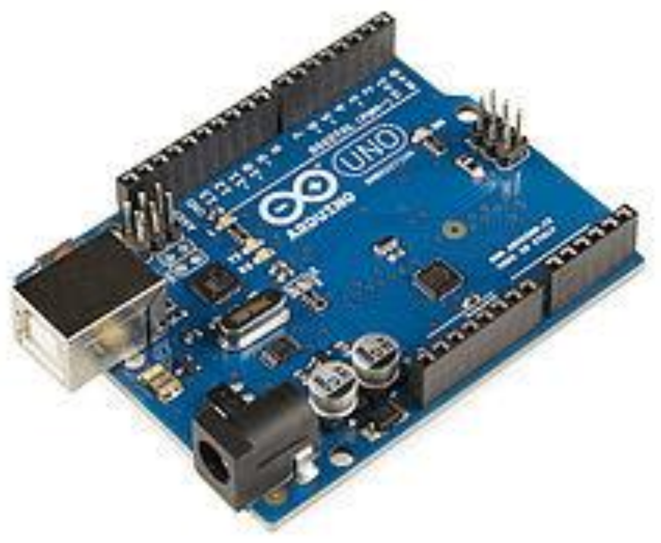
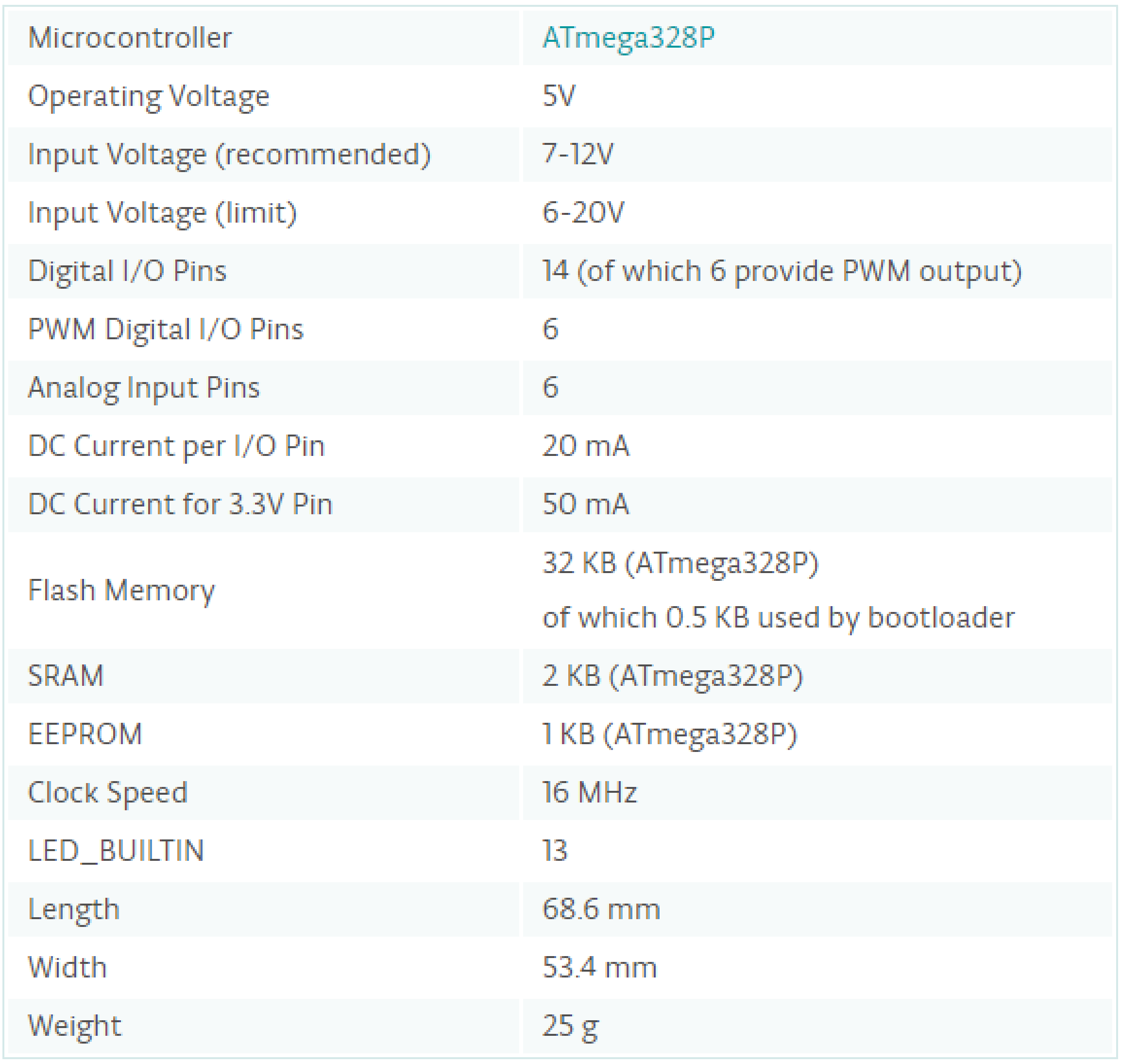


Figure 3.1 Arudino Board

Technical specifications



Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table3.1 ardunio features

Programming

The arduino/genuino uno can be programmed with the (arduino software (ide)). Select "arduino/genuino uno from the tools > board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The atmega328 on the arduino/genuinouno comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original stk500 protocol (reference, c header files).

Waring
The arduino/genuinouno has a resettable polyfuse that protects your computer's usb ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 ma is applied to the usb port, the fuse will automatically break the connection until the short or overload is removed.

DIfferences with other boards
The uno differs from all preceding boards in that it does not use the ftdiusb-to-serial driver chip. Instead, it features the atmega16u2 (atmega8u2 up to version r2) programmed as ausb-to-serial converter.

Power
The arduino/genuinouno board can be powered via the usb connection or with an external power supply. The power source is selected automatically.
External (non-usb) power can come either from an ac-to-dc adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the gnd and vin pin headers of the power connector.

The power pins are as follows:

Vin. The input voltage to the arduino/genuino board when it's using an external power source (as opposed to 5 volts from the usb connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5v.this pin outputs a regulated 5v from the regulator on the board. The board can be supplied with power either from the dc power jack (7 - 12v), the usb connector (5v), or the vin pin of the board (7-12v). Supplying voltage via the 5v or 3.3v pins bypasses the regulator, and can damage your board. We don't advise it.

3v3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 ma.

Gnd. Ground pins.

Ioref. This pin on the arduino/genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the ioref pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5v or 3.3v.

Memory
The atmega328 has 32 kb (with 0.5 kb occupied by the bootloader). It also has 2 kb of sram and 1 kb of eeprom (which can be read and written with the eeprom library).

input and output
See the mapping between arduino pins and atmega328p ports. The mapping for the atmega8, 168, and 328 is identical.

Circuit pin diagram

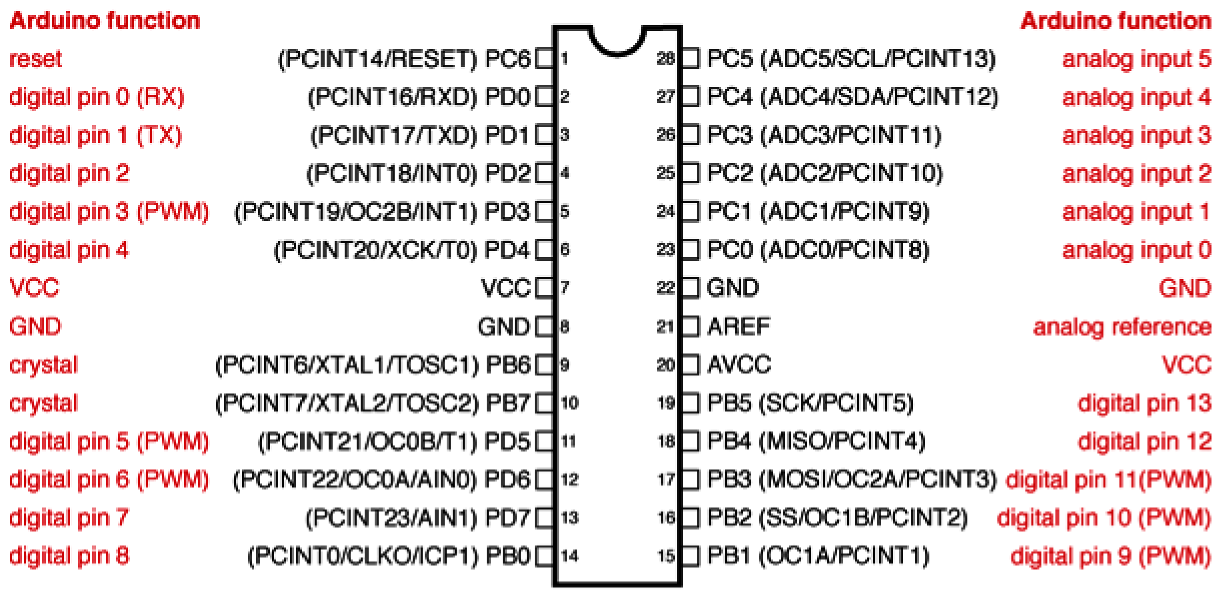


Figure 3.2 Ardunio pin diagram

Each of the 14 digital pins on the uno can be used as an input or output, using pinmode(), digitalwrite (), and digitalread () functions. They operate at 5 volts. Each pin can provide or receive 20 ma as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40ma is the value that must not be exceeded on any i/o pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

Serial: 0 (rx) and 1 (tx). Used to receive (rx) and transmit (tx) ttl serial data. These pins are connected to the corresponding pins of the atmega8u2 usb-to-ttl serial chip.

External interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachinterrupt() function for details.

Pwm: 3, 5, 6, 9, 10, and 11. Provide 8-bit pwm output with the analog write() function.

Spi: 10 (ss), 11 (mosi), 12 (miso), 13 (sck). These pins support spi communication using the spi library.

Led: 13. There is a built-in led driven by digital pin 13. When the pin is high value, the led is on, when the pin is low, it's off.

Twi: a4 or sda pin and a5 or scl pin. Support twi communication using the wire library.

The uno has 6 analog inputs, labeled a0 through a5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the aref pin and the analogreference() function.

There are a couple of other pins on the board:

Aref.Reference voltage for the analog inputs. Used with analogreference().

Reset. Bring this line low to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

Arduino/genuinouno has a number of facilities for communicating with a computer, another arduino/genuino board, or other microcontrollers. The atmega328 provides uarttl (5v) serial communication, which is available on digital pins 0 (rx) and 1 (tx). An atmega16u2 on the board channels this serial communication over usb and appears as a virtual com port to software on the computer. The 16u2 firmware uses the standard usb com drivers, and no external driver is needed. However, on windows, a .inf file is required. The arduino software (ide) includes a serial monitor which allows simple textual data to be sent to and from the board. The rx and txleds on the board will flash when data is being transmitted via the usb-to-serial chip and usb connection to the computer (but not for serial communication on pins 0 and 1).A softwareserial library allows serial communication on any of the uno's digital pins.

Automatic (software) reset

Rather than requiring a physical press of the reset button before an upload, the arduino/genuinouno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (dtr) of the atmega8u2/16u2 is connected to the reset line of the atmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The arduino software (ide) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of dtr can be well-coordinated with the start of the upload.

MAX232 IC:

The MAX232 is an IC, first created by Maxim Integrated Products, that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ± 7.5 V) from a single + 5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to + 5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case.

The receivers reduce RS-232 inputs (which may be as high as ± 25 V), to standard 5 VTTL levels. These receivers have a typical threshold of 1.3 V, and a typical hysteresis of 0.5 V.

Voltage Level

RS232 line type and logic level	RS232 voltage	TTL voltage to/from MAX232
Data transmission (Rx/Tx) logic 0	+3 V to +15 V	0 V
Data transmission (Rx/Tx) logic 1	-3 V to -15 V	5 V
Control signals (RTS/CTS/DTR/DSR) logic 0	-3 V to -15 V	5 V
Control signals kjnic 1	+3 V to +15 V	0 V

Table3.2 showing voltage of RS232 & MAX232

The later MAX232A is backwards compatible with the original MAX232 but may operate at higher baud rates and can use smaller external capacitors – 0.1 μF in place of the 1.0 μF capacitors used with the original device. The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply RS 232 voltage levels from a single 5v supply. Each receiver converts RS-232 to 5v TTL/CMOS levels. Each driver converts TLL/CMOS input levels into EIA-232 levels. The P3_0 (RX) and P3_1 (TX) pin of controller is connected to the max 232 driver and the TX and RX pin of max 232 is connected to the GSM modem or PC.

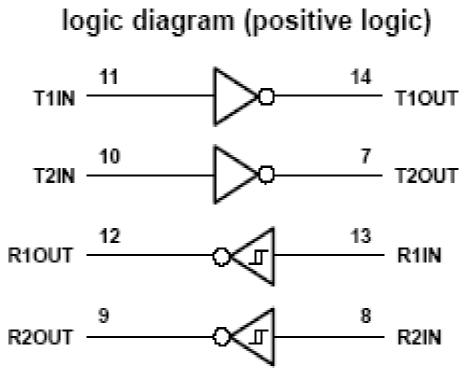


Fig3.3. Logic Diagram

Logic Diagram

In this circuit the microcontroller transmitter pin is connected in the MAX232 T2IN pin which converts input 5v TTL/CMOS level to RS232 level. Then T2OUT pin is connected to reviver pin of 9 pin D type serial connector which is directly connected to PC.

In PC the transmitting data is given to R2IN of MAX232 through transmitting pin of 9 pin D type connector which converts the RS232 level to 5v TTL/CMOS level. The R2OUT pin is connected to receiver pin of the microcontroller. Likewise the data is transmitted and received between the microcontroller and PC or other device vice versa.

Liquid crystal display

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

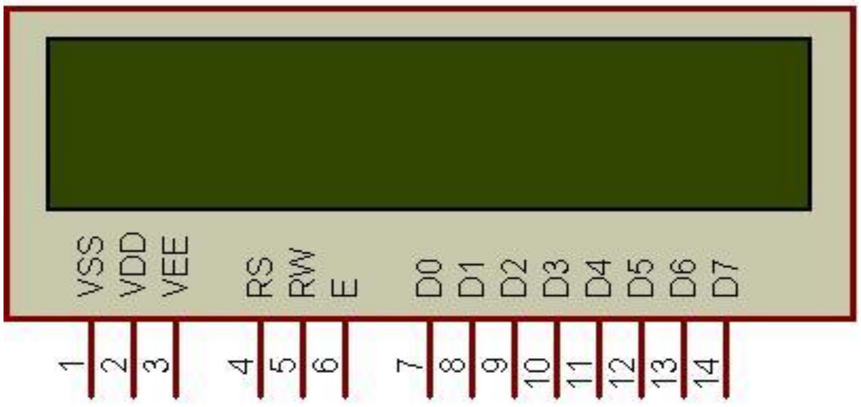


Figure 3.4 16x2 LCD

Pin Description

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections). Pin description is shown in the table below.

Pin Configuration table for a 16X2 LCD character display:-

Pin Number	Symbol	Function
1	Vss	Ground Terminal
2	Vcc	Positive Supply
3	Vdd	Contrast adjustment
4	RS	Register Select; 0→Instruction Register, 1→Data Register
5	R/W	Read/write Signal; 1→Read, 0→ Write
6	E	Enable; Falling edge
7	DB0	Bi-directional data bus, data transfer is performed once, thru DB0 to DB7, in the case of interface data length is 8-bits; and twice, through DB4 to DB7 in the case of interface data length is 4-bits. Upper four bits first then lower four bits.
8	DB1	
9	DB2	
10	DB3	

11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED-(K)	Back light LED cathode terminal
16	LED+(A)	Back Light LED anode terminal

Table.3.3: Pin Configuration of LCD

Data/Signals/Execution of LCD

LCD accepts two types of signals, one is data, and another is control. These signals are recognized by the LCD module from status of the RS pin. Now data can be read also from the LCD display, by pulling the R/W pin high. As soon as the E pin is pulsed, LCD display reads data at the falling edge of the pulse and executes it, same for the case of transmission.

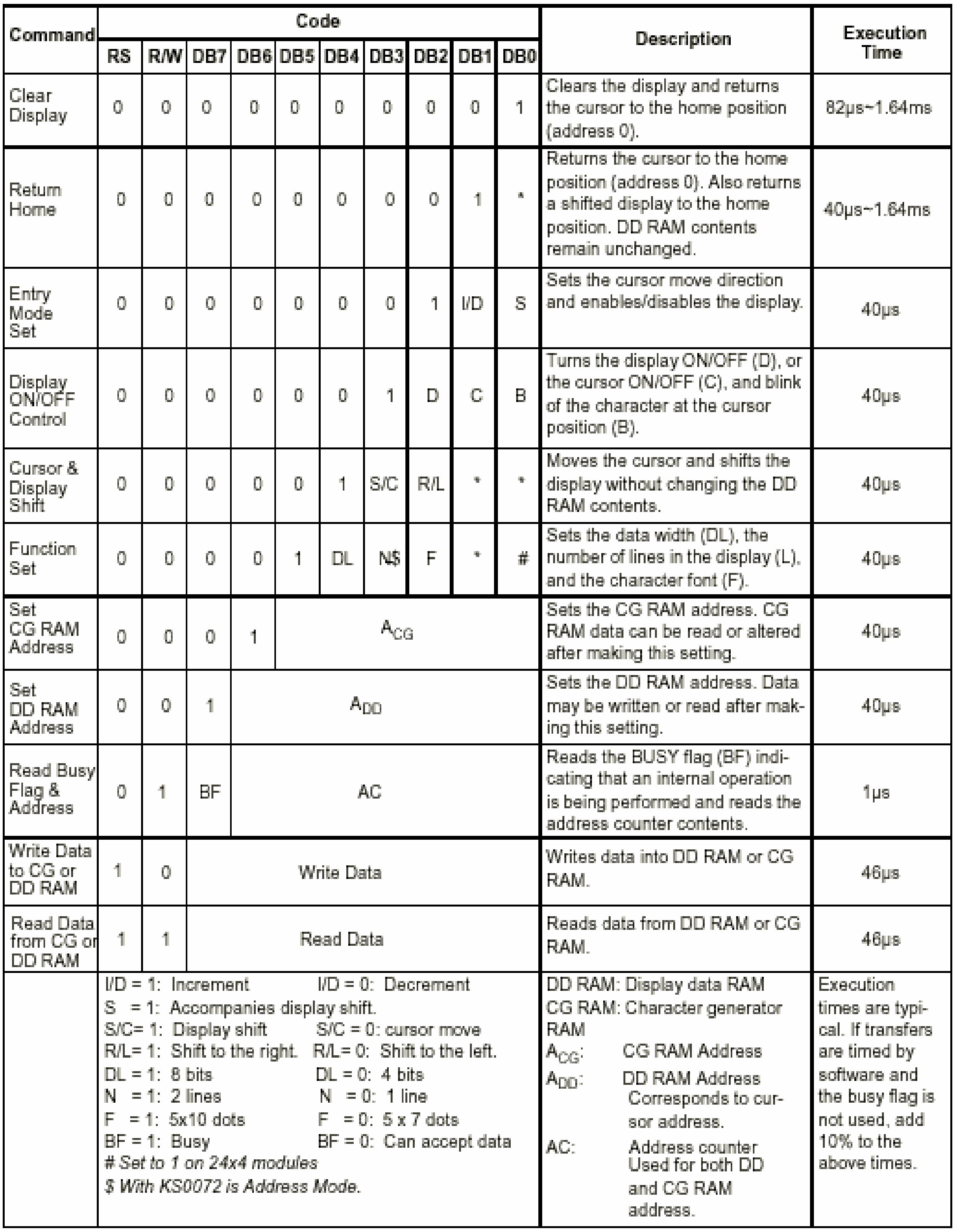
LCD display takes a time of 39-43µS to place a character or execute a command. Except for clearing display and to seek cursor to home position it takes 1.53ms to 1.64ms. Any attempt to send any data before this interval may lead to failure to read data or execution of the current data in some devices. Some devices compensate the speed by storing the incoming data to some temporary registers.

Instruction Register (IR) and Data Register (DR)There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. when send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD. We will discuss more on LCD instruction set further in this tutorial.

Commands and Instruction set

Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU. These signals, which include register selection signal (RS), read/write signal (R/W), and the data bus (DB0 to DB7), make up the LCD instructions (Table 3). There are four categories of instructions that:

- Designate LCD functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions



Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	F	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0	0	0	1	A_{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs
Set DD RAM Address	0	0	1	A_{DD}							Sets the DD RAM address. Data may be written or read after making this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5x10 dots F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*										DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

Table 3.4: Code for LCD

Although looking at the table you can make your own commands and test them. Below is a brief list of useful commands which are used frequently while working on the LCD.

List of Command

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add*	128+add*
17	Set CGRAM address or set pointer to CGRAM location	0x40+add**	64+add**

Table3.5: Frequently used commands and instructions for LCD

* DDRAM address given in LCD basics section see Figure 2,3,4
** CGRAM address from 0x00 to 0x3F, 0x00 to 0x07 for char1 and so on.

Liquid crystal displays interfacing with Controller

The LCD standard requires 3 control lines and 8 I/O lines for the data bus.

• 8 data pins D7:D0

Bi-directional data/command pins.

Alphanumeric characters are sent in ASCII format.

• RS: Register Select

RS = 0 -> Command Register is selected

RS = 1 -> Data Register is selected

• R/W: Read or Write

0 -> Write, 1 -> Read

• E: Enable (Latch data)
Used to latch the data present on the data pins.
A high-to-low edge is needed to latch the data.

Ultrasonic sensor

Ultrasonic sensors are industrial control devices that use sound waves above 20,000 Hz, beyond the range of human hearing, to measure and calculate distance from the sensor to a specified target object.

Features of ultrasonic sensors:

- Devices with TEACH-IN functionality for fast and simple installation
- ULTRA 3000 software for improved adaptation of sensors to applications
- Adjustable sensitivity to the sound beam width for optimized adjustment of the sensor characteristics according to the application
- Temperature compensation - compensates for sound velocity due to varying air temperatures
- Synchronization input to prevent cross-talk interference when sensors are mounted within close proximity of each other
- Sensors with digital and/ or analog outputs

Description:

Ultrasonic sensors use electrical energy and a ceramic transducer to emit and receive mechanical energy in the form of sound waves. Sound waves are essentially pressure waves that travel through solids, liquids and gases and can be used in industrial applications to measure distance or detect the presence or absence of targets

Ultrasonic sensors (also known as tranceivers when they both send and receive) work on a principle similar to radar or sonar which evaluate attributes of a target by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object.

This technology can be used for measuring: wind speed and direction (anemometer), fullness of a tank, and speed through air or water. For measuring speed or direction a device uses multiple detectors and calculates the speed from the relative distances to particulates in the air or water. To measure the amount of liquid in a tank, the sensor measures the distance to the surface of the fluid. Further applications include: humidifiers, sonar, medical ultrasonography, burglar alarms, and non-destructive testing.

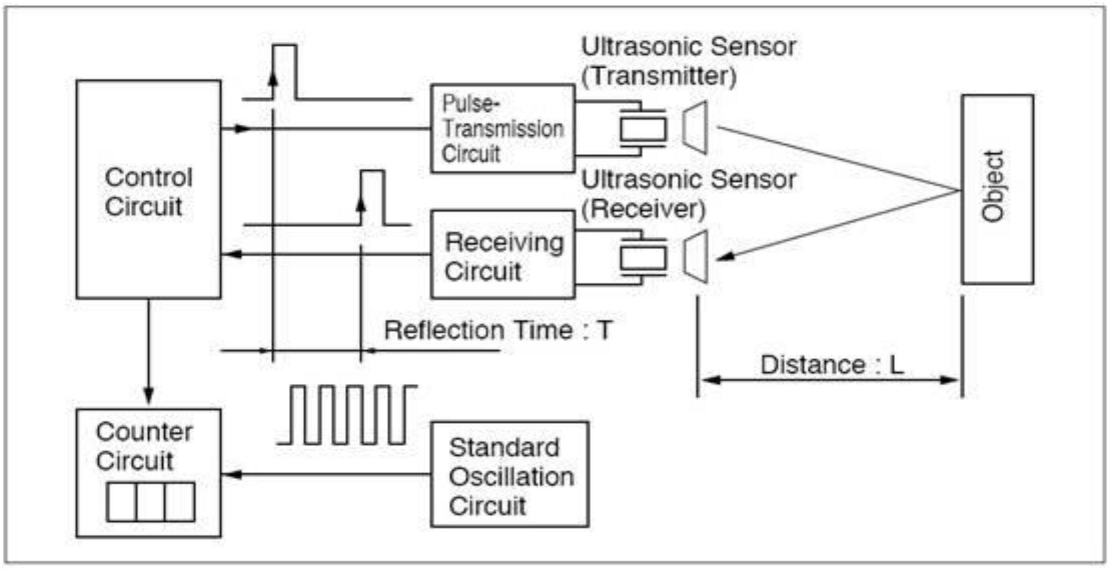

Ultrasonic Sensor (Transmitter)
Pulse-Transmission Circuit
Control Circuit
Ultrasonic Sensor (Receiver)
Receiving Circuit
Object
Reflection Time : T
Distance : L
Counter Circuit
Standard Oscillation Circuit

Figure 3.5: Block Diagram of Ultrasonic Sensor

Systems typically use a transducer which generates sound waves in the ultrasonic range, above 20,000 hertz, by turning electrical energy into sound, then upon receiving the echo turn the sound waves into electrical energy which can be measured and displayed.

The technology is limited by the shapes of surfaces and the density or consistency of the material. For example foam on the surface of a fluid in a tank could distort a reading.

Transducers

Sound field of a non focusing 4MHz ultrasonic transducer with a near field length of N=67mm in water. The plot shows the sound pressure at a logarithmic db-scale. Sound pressure field of the same ultrasonic transducer (4MHz, N=67mm) with the transducer surface having a spherical curvature with the curvature radius R=30mm

An ultrasonic transducer is a device that converts energy into ultrasound, or sound waves above the normal range of human hearing. While technically a dog whistle is an ultrasonic transducer that converts mechanical energy in the form of air pressure into ultrasonic sound waves, the term is more apt to be used to refer to piezoelectric transducers that convert electrical energy into sound. Piezoelectric crystals have the property of changing size when a voltage is applied, thus applying an alternating current (AC) across them causes them to oscillate at very high frequencies, thus producing very high frequency sound waves.

The location, at which a transducer focuses the sound, can be determined by the active transducer area and shape, the ultrasound frequency and the sound velocity of the propagation medium. The example shows the sound fields of an unfocused and a focusing ultrasonic transducer in water.

Range

This ultrasonic rangefinder can measure distances up to 2.5 meters at accuracy of 1 centi-meter.

Working

The sensor has a ceramic transducer that vibrates when electrical energy is applied to it. The vibrations compress and expand air molecules in waves from the sensor face to a target object. A transducer both transmits and receives sound. The ultrasonic sensor will measure distance by emitting a sound wave and then "listening" for a set period of time, allowing for the return echo of the sound wave bouncing off the target, before retransmitting.
Microcontroller and the ultrasonic transducer module HC-SR04 forms the basis of this circuit. The ultrasonic module sends a signal to the object, then picks up its echo and outputs a wave form whose time period is proportional to the distance. The microcontroller accepts this signal, performs necessary processing and displays the corresponding distance on the 3 digit seven segment display. This circuit finds a lot of application in projects like automotive parking sensors, obstacle warning systems, terrain monitoring robots, industrial distance measurements etc.
It has a resolution of 0.3cm and the ranging distance is from 2cm to 500cm. It operates from a 5V DC supply and the standby current is less than 2mA. The module transmits an ultrasonic signal, picks up its echo, measures the time elapsed between the two events and outputs a waveform whose high time is modulated by the measured time which is proportional to the distance.
The supporting circuits fabricated on the module makes it almost stand alone and what the programmer need to do is to send a trigger signal to it for initiating transmission and receive the echo signal from it for distance calculation.
The HR-SR04 has four pins namely Vcc, Trigger, Echo, GND and they are explained in detail below.

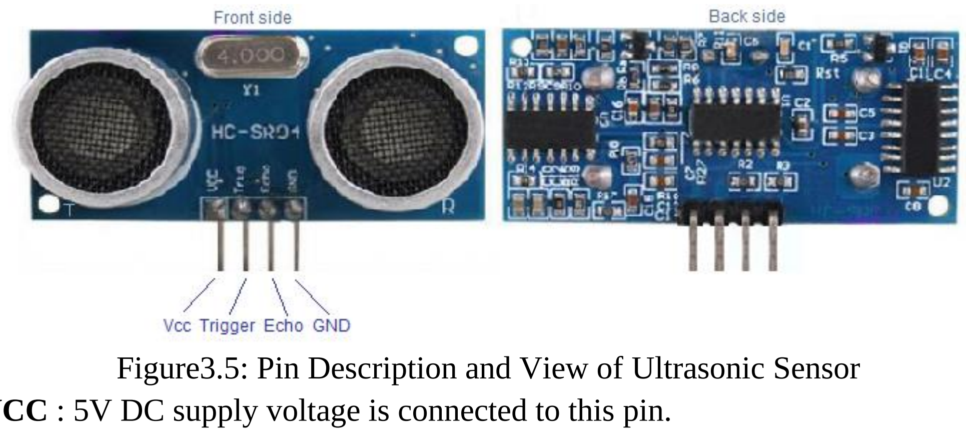


Figure3.5: Pin Description and View of Ultrasonic Sensor

1) **VCC** : 5V DC supply voltage is connected to this pin.

2) **Trigger**: The trigger signal for starting the transmission is given to this pin. The trigger signal must be a pulse with 10uS high time. When the module receives a valid trigger signal it issues 8 pulses of 40KHz ultrasonic sound from the transmitter. The echo of this sound is picked by the receiver.

3)**Echo**: At this pin, the module outputs a waveform with high time proportional to the distance.

4) **GND**: Ground is connected to this pin.

The transmitter part of the circuit is build around IC1(NE 555).The IC1 is wired as an astable multi vibrator operating at 40KHz.The output of IC1 is amplifier the complementary pair of transistors (Q1 & Q2) and transmitted by the ultrasonic transmitter K1.The push button switch S1 is used the activate the transmitter.

The receiver uses an ultrasonic sensor transducer (K2) to sense the ultrasonic signals. When an ultrasonic signal is falling on the sensor, it produces a proportional voltage signal at its output. This weak signal is amplified by the two stage amplifier circuit comprising of transistors Q3 and Q4.The output of the amplifier is rectified by the diodes D3 & D4.The rectified signal is given to the inverting input of the opamp which is wired as a comparator. Whenever there is an ultrasonic signal falling on the receiver, the output of the comparator activates the transistors Q5 & Q6 to drive the relay. In this way the load connected via the relay can be switched. The diode D5 is used as a free-wheeling diode.

Detectors

Since piezoelectric crystal generate a voltage when force is applied to them, the same crystal can be used as an ultrasonic detector. Some systems use separate transmitter and receiver components while others combine both in a single piezoelectric transceiver.

Alternative methods for creating and detecting ultrasound include magnetostriction and capacitive actuation.

Application

Ultrasonic sensors use sound waves rather than light, making them ideal for stable detection of uneven surfaces, liquids, clear objects, and objects in dirty environments. These sensors work well for applications that require precise measurements between stationary and moving objects.

Ultrasonic sensor provides a very low-cost and easy method of distance measurement. This sensor is perfect for any number of applications that require you to perform measurements between moving or stationary objects. Naturally, robotics applications are very popular but it is also find in product which is useful in security systems or as an infrared replacement if so desired.

Use in medicine

Medical ultrasonic transducers (probes) come in a variety of different shapes and sizes for use in making pictures of different parts of the body. The transducer may be passed over the surface of the body or inserted into an body opening such as the rectum or vagina. Clinicians who perform ultrasound-guided procedures often use a probe positioning system to hold the ultrasonic transducer.

Use in industry

Ultrasonic sensors are used to detect the presence of targets and to measure the distance to targets in many automated factories and process plants. Sensors with an on or off digital output are available for detecting the presence of objects, and sensors with an analog output

which varies proportionally to the sensor to target separation distance are commercially available.
Other types of transducers are used in commercially available ultrasonic cleaning devices. An ultrasonic transducer is affixed to a stainless steel pan which is filled with a solvent (frequently water or isopropanol) and a square wave is applied to it, imparting vibrational energy on the liquid.

Application In Industries

- Measurement of dynamically changing diameters
- Measurement of dynamically changing distances
- Measurement of dynamically changing heights
- Measurement of dynamically changing depths
- Counting number of units.

ULN2003:-

The ULN2003 is a monolithic IC consists of seven NPN Darlington transistor pairs with high voltage and current capability. It is commonly used for applications such as relay drivers, motor, display drivers, led lamp drivers, logic buffers, line drivers, hammer drivers and other high voltage current applications. It consists of common cathode clamp diodes for each NPN Darlington pair which makes this driver IC useful for switching inductive loads.

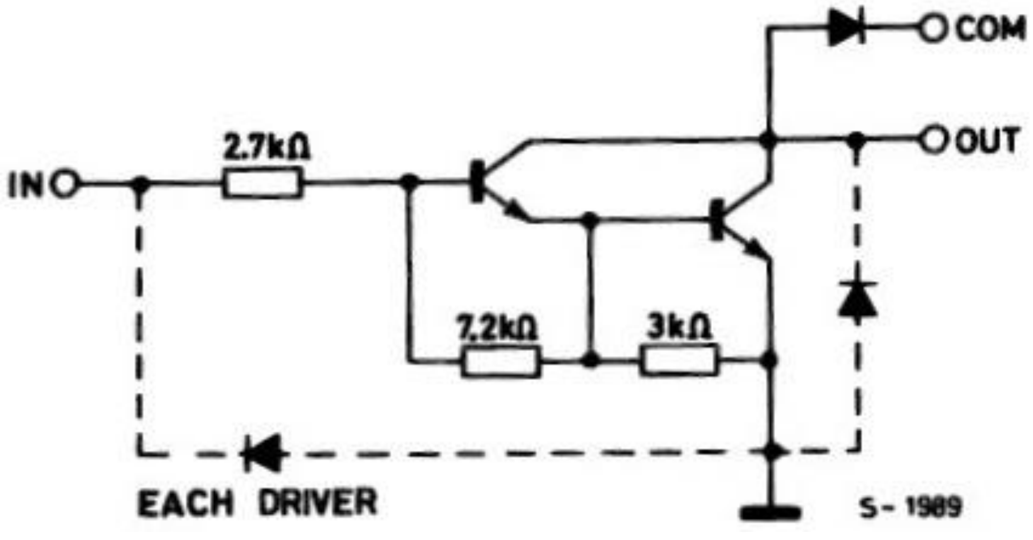


Figure 3.7 ULN 2003

The output of the driver is open collector and the collector current rating of each darlington pair is 500mA. Darlington pairs may be paralleled if higher current is required. The driver IC also consists of a 2.7KΩ base resistor for each darlington pair. Thus each darlington pair can be operated directly with TTL or 5V CMOS devices. This driver IC can be used for high voltage applications up to 50V.

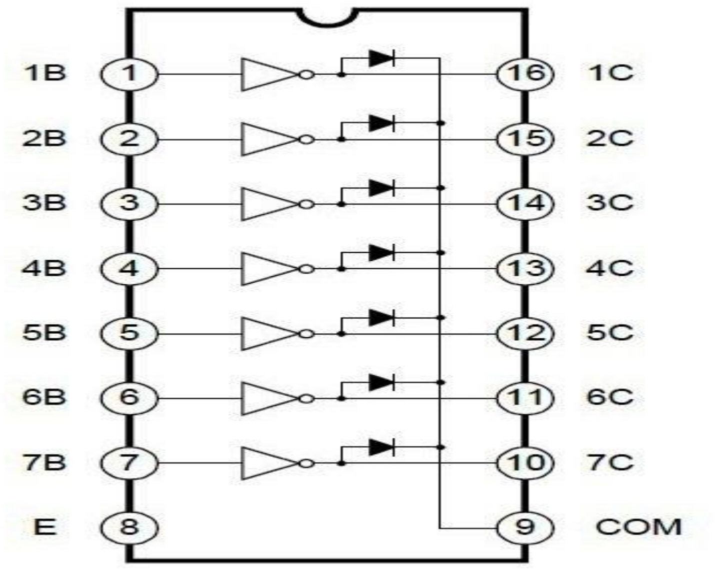


Figure3.8 Logic Diagram of ULN2003

Note that the driver provides open collector output, so it can only sink current, cannot source. Thus when a 5V is given to 1B terminal, 1C terminal will be connected to ground via darlington pair and the maximum current that it can handle is 500A. From the above logic diagram we can see that cathode of protection diodes are shorted to 9th pin called COM. So for driving inductive loads, it must connect to the supply voltage.

ULN2003 is widely used in relay driving and stepper motor driving applications.

FEATURES
* 500mA rated collector current (Single output)
* High-voltage outputs: 50V
* Inputs compatible with various types of logic.
* Relay driver application

DC Motor

A DC motor in simple words is a device that converts direct current (electrical energy) into mechanical energy. It’s of vital importance for the industry today.

A DC motor is designed to run on DC electric power. Two examples of pure DC designs are Michael Faraday's homo-polar motor (which is uncommon), and the ball bearing motor, which is (so far) a novelty.
By far the most common DC motor types are the brushed and brushless types, which use internal and external commutation respectively to create an oscillating AC current from the DC source—so they are not purely DC machines in a strict sense.
We in our project are using brushed DC Motor, which will operate in the ratings of 12v DC 0.6A. The speed of a DC motor can be controlled by changing the voltage applied to the armature or by changing the field current. The introduction of variable resistance in the armature circuit or field circuit allowed speed control. Modern DC motors are often controlled by power electronics systems called DC drives.

Figure3.9. DC Motor

Usage
The DC motor or Direct Current Motor to give it its full title, is the most commonly used actuator for producing continuous movement and whose speed of rotation can easily be controlled, making them ideal for use in applications were speed control, servo type control, and/or positioning is required. A DC motor consists of two parts, a "Stator" which is the stationary part and a "Rotor" which is the rotating part. The result is that there are basically three types of DC Motor available.

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

SOFTWARE REQUIREMENTS

FRONT END	:	C,C++
BACK END	:	ARDUNIO

OPERATING SYSTEM	:	WINDOWS 8
IDE	:	ARDUNIO IDE

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

3.5 NON-FUNCTIONAL REQUIREMENTS

Efficiency

Our multi-modal event tracking and evolution framework is suitable for multimedia documents from various social media platforms, which can not only effectively capture their multi-modal topics, but also obtain the evolutionary trends of social events and generate effective event summary details over time. Our proposed mmETM model can exploit the multi-modal property of social event, which can effectively model social media documents including long text with related images and learn the correlations between textual and visual modalities to separate the visual-representative topics and non-visual-representative topics.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

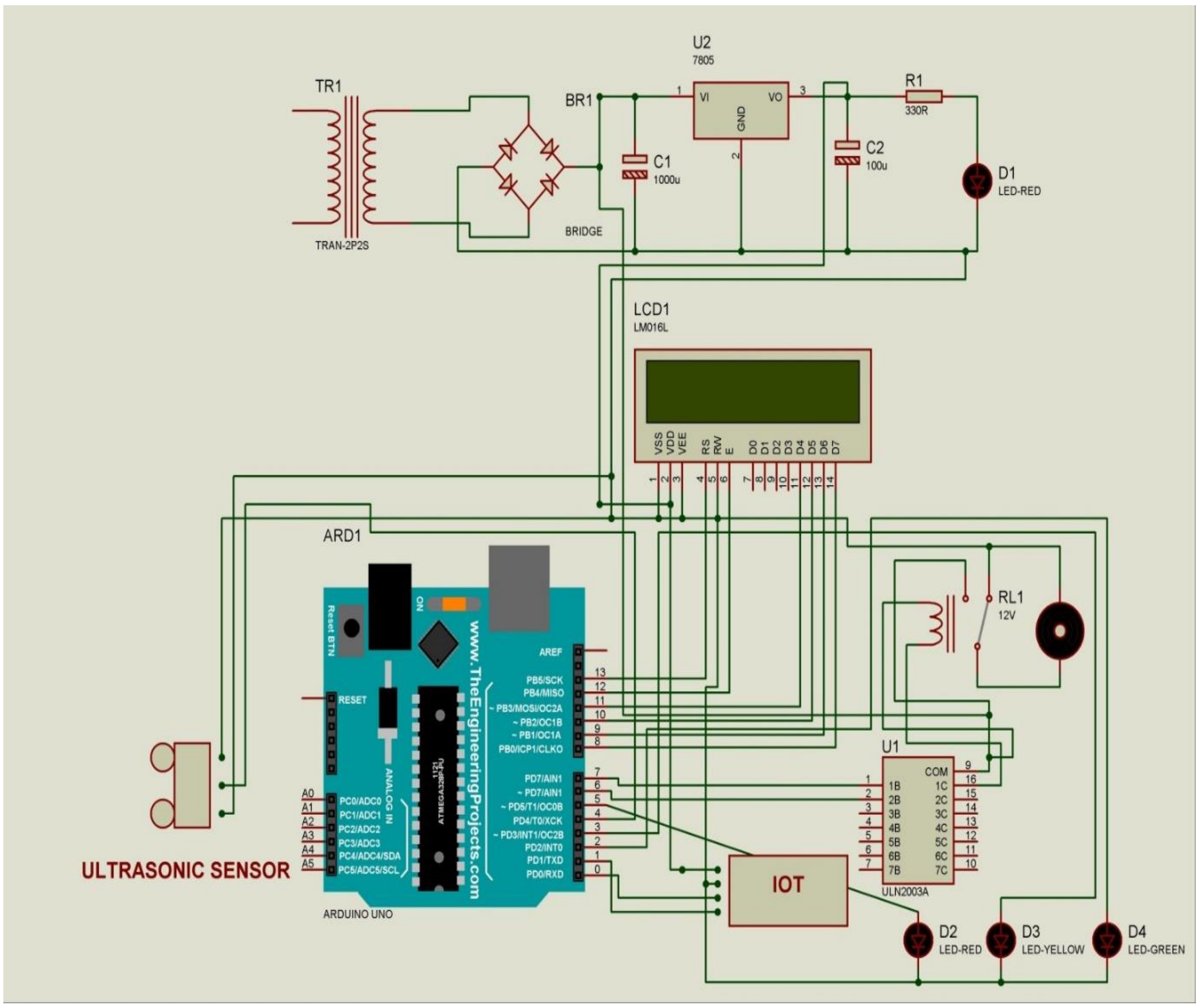


Figure4.1 System Architecture

BLOCK DIAGRAM

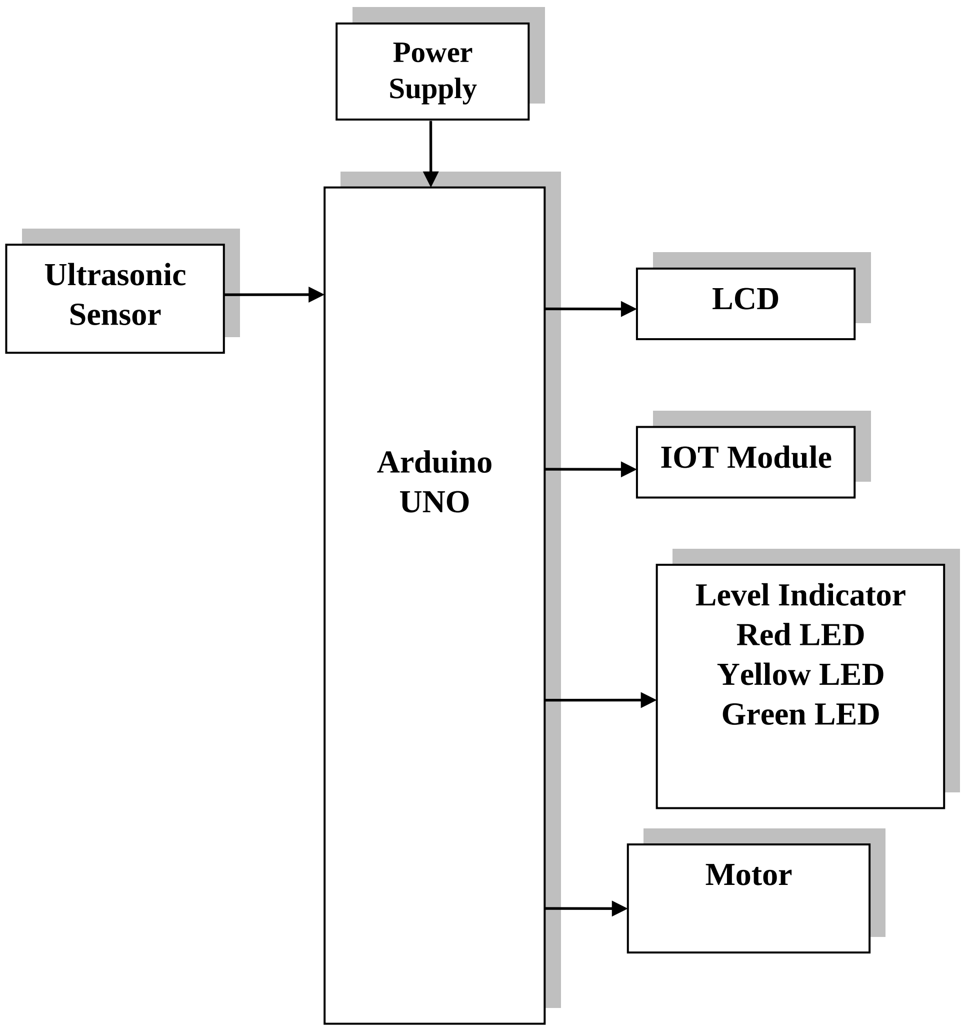


Figure4.2 Block diagram

Explaination

In Smart waste collection architectures, it generally common that all components From this h gateway, all other appliances and components are controlled by the users and all protocols for operating the home equipment's and appliances are defined in this gateway. Fi outlines a traditional setup for a smart home architecture wherein the home gateway controls other home appliances and also connects with other user devices through the Internet and the block diagram shows how all the devices are connected.

4.2 MODULES

Power supply

The power supply section is the section which provide +5V for the components to work. IC LM7805 is used for providing a constant power of +5V.
The ac voltage, typically 220V, is connected to a transformer, which steps down that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.
A regulator circuit removes the ripples and also retains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

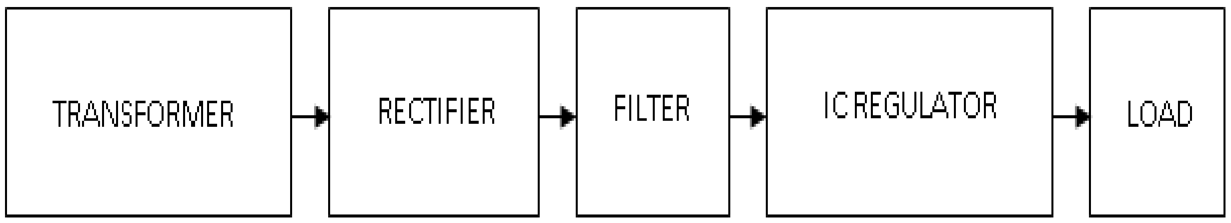


Figure4.3. Block Diagram of Power Supply

Transformer

Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC.
Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in India) to a safer low voltage.
The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up. The transformer will step down the power supply voltage (0-230V) to (0- 6V) level. Then the secondary of the potential transformer will be connected to the bridge rectifier, which is constructed with the help of PN junction diodes. The advantages of using bridge rectifier are it will give peak voltage output as DC.

Rectifier

There are several ways of connecting diodes to make a rectifier to convert AC to DC. The bridge rectifier is the most important and it produces full-wave varying DC. A full-wave rectifier can also be made from just two diodes if a centre-tap transformer is used, but this method is rarely used now that diodes are cheaper. A single diode can be used as a rectifier but it only uses the positive (+) parts of the AC wave to produce half-wave varying DC

Bridge Rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners. Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

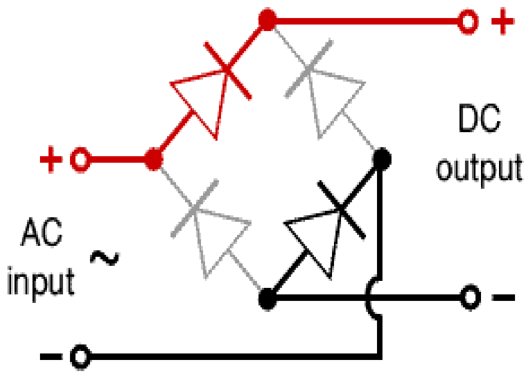


Figure4.4. Bridge Rectifier

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit.

i. The main advantage of this bridge circuit is that it does not require a special centre tapped transformer, thereby reducing its size and cost.

ii. The single secondary winding is connected to one side of the diode bridge network and the load to the other side as shown below.

iii. The result is still a pulsating direct current but with double the frequency.

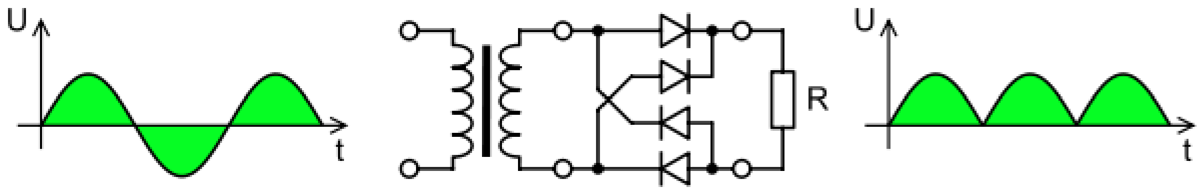


Figure4.5. Output Waveform of DC

Smoothing

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.

Voltage Regulators

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

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts. Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection').

Many of the fixed voltage regulator ICs has 3 leads and look like power transistors, such as the 7805 +5V 1Amp regulator. They include a hole for attaching a heat sink if necessary.

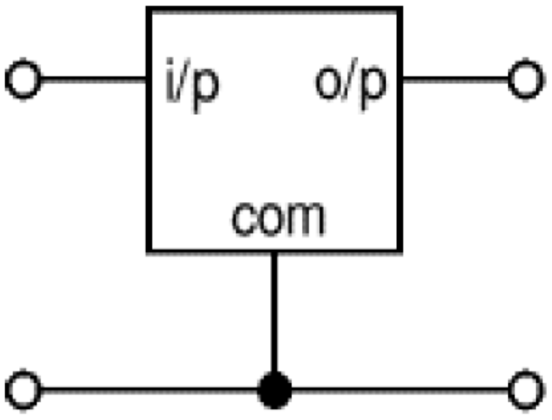


Figure4.6. Regulator

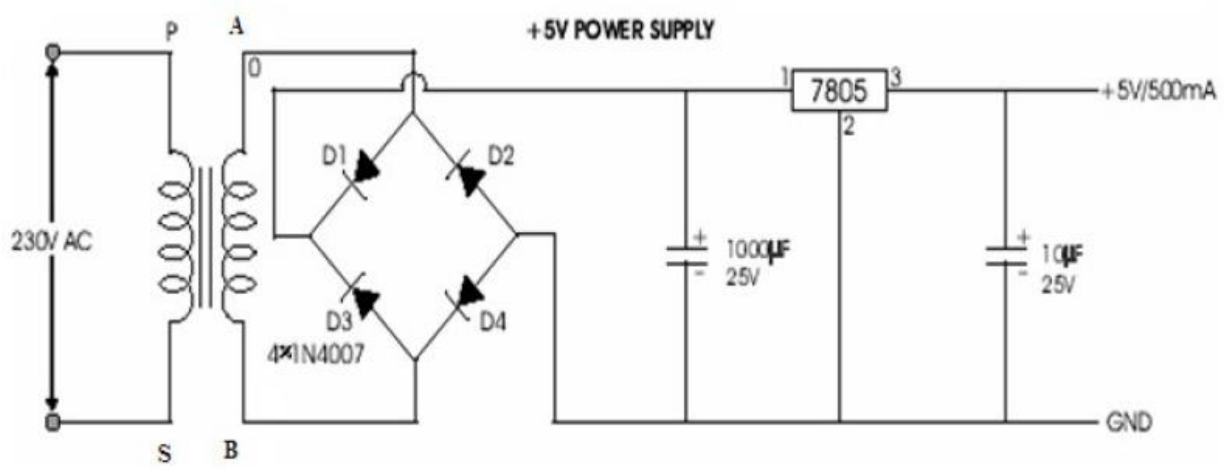


Figure4.7. Circuit Diagram of Power Supply

4.3UML DIAGRAMS

Use-case diagram:

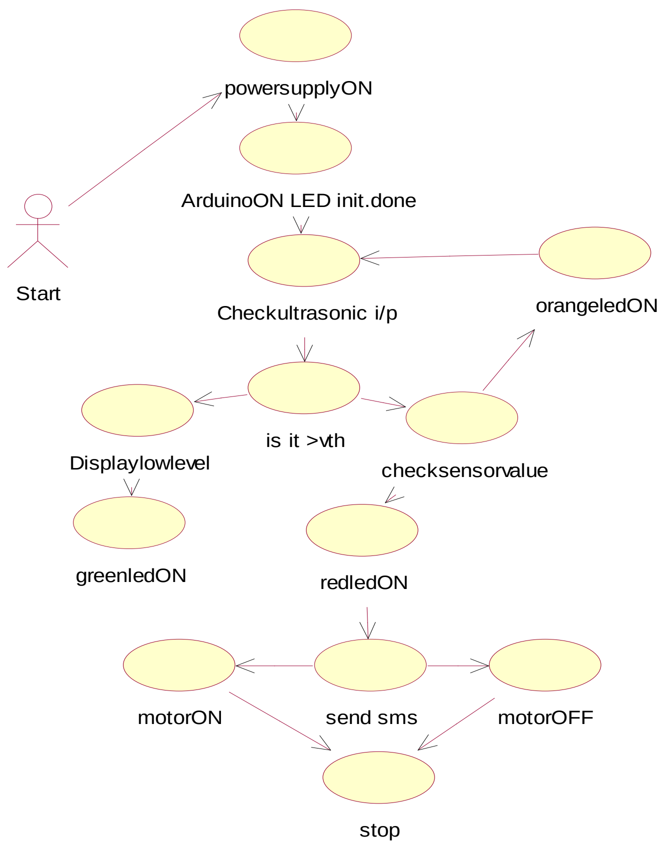


Figure 4.8: use case diagram

Explanation

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

Component Diagram:

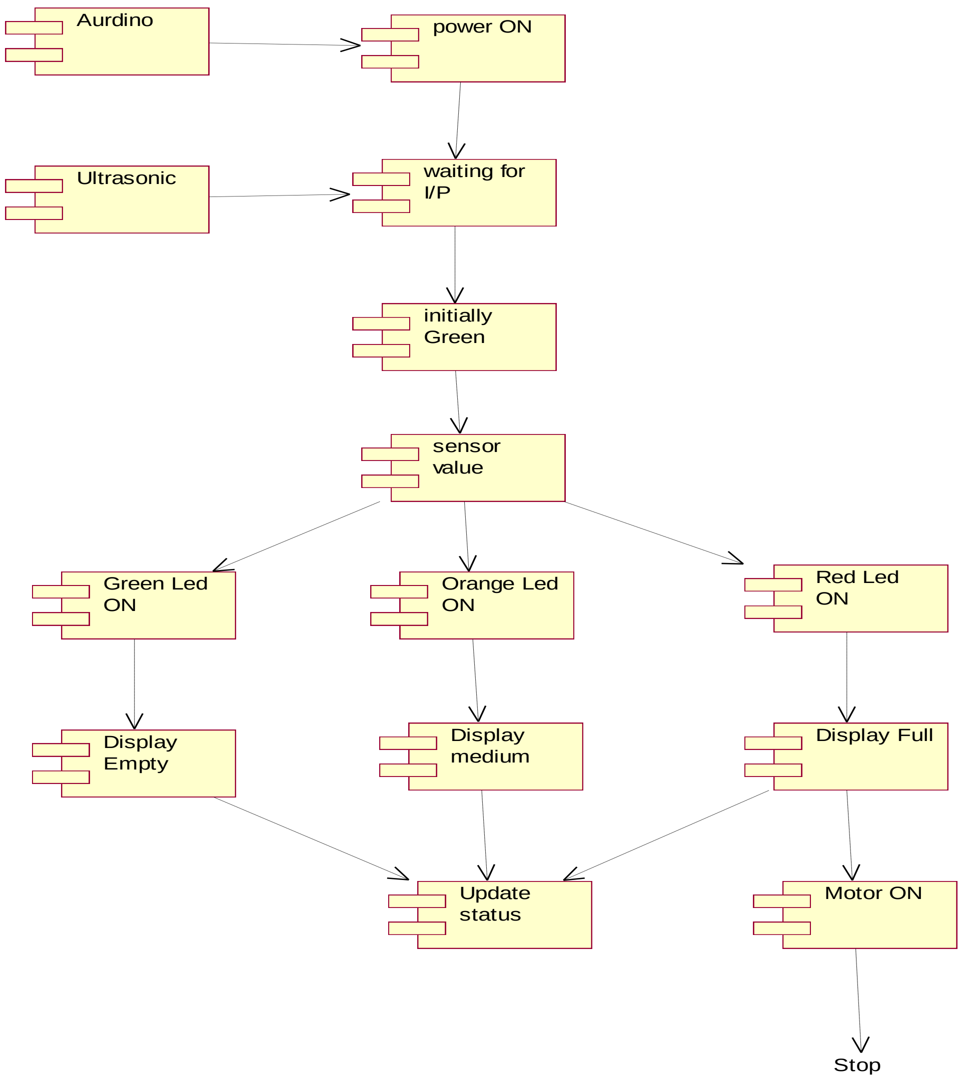


Figure 4.9 Component diagram

Explanation

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

ER-Diagram:

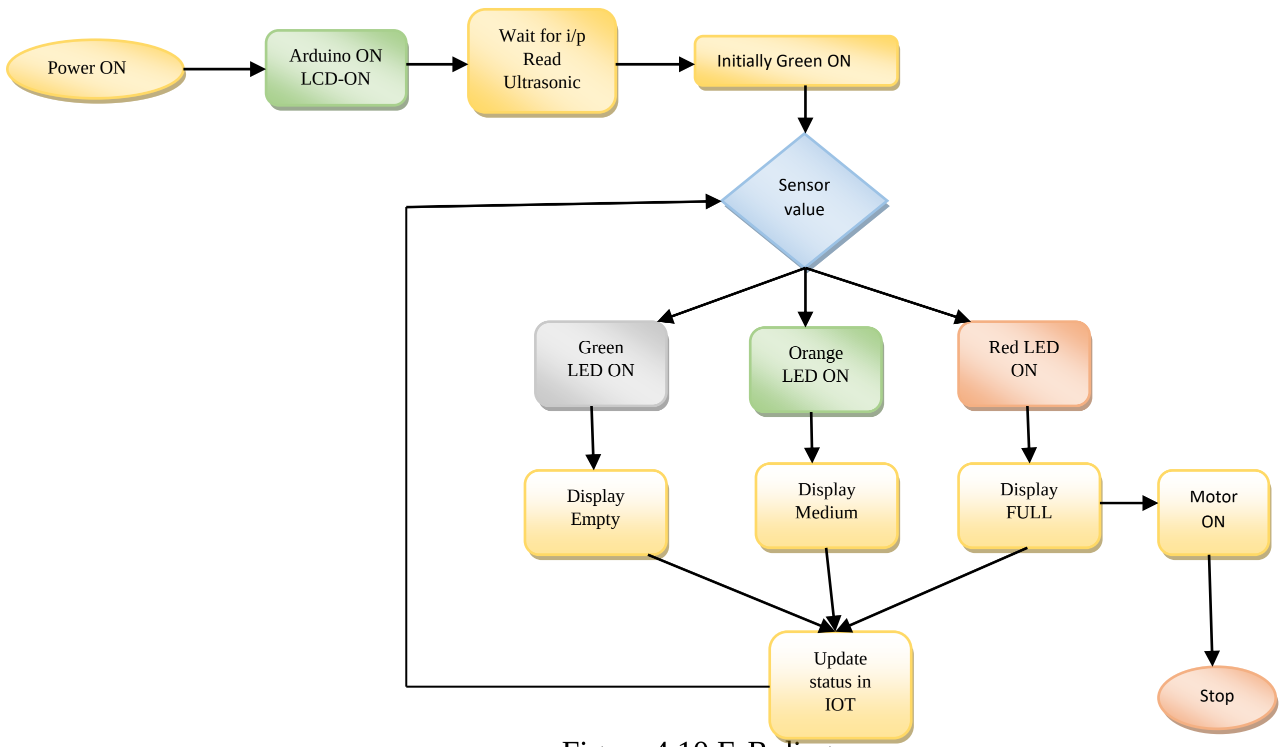


Figure 4.10 E-R diagram

Explanation

In software engineering, an entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators.

State Diagram:

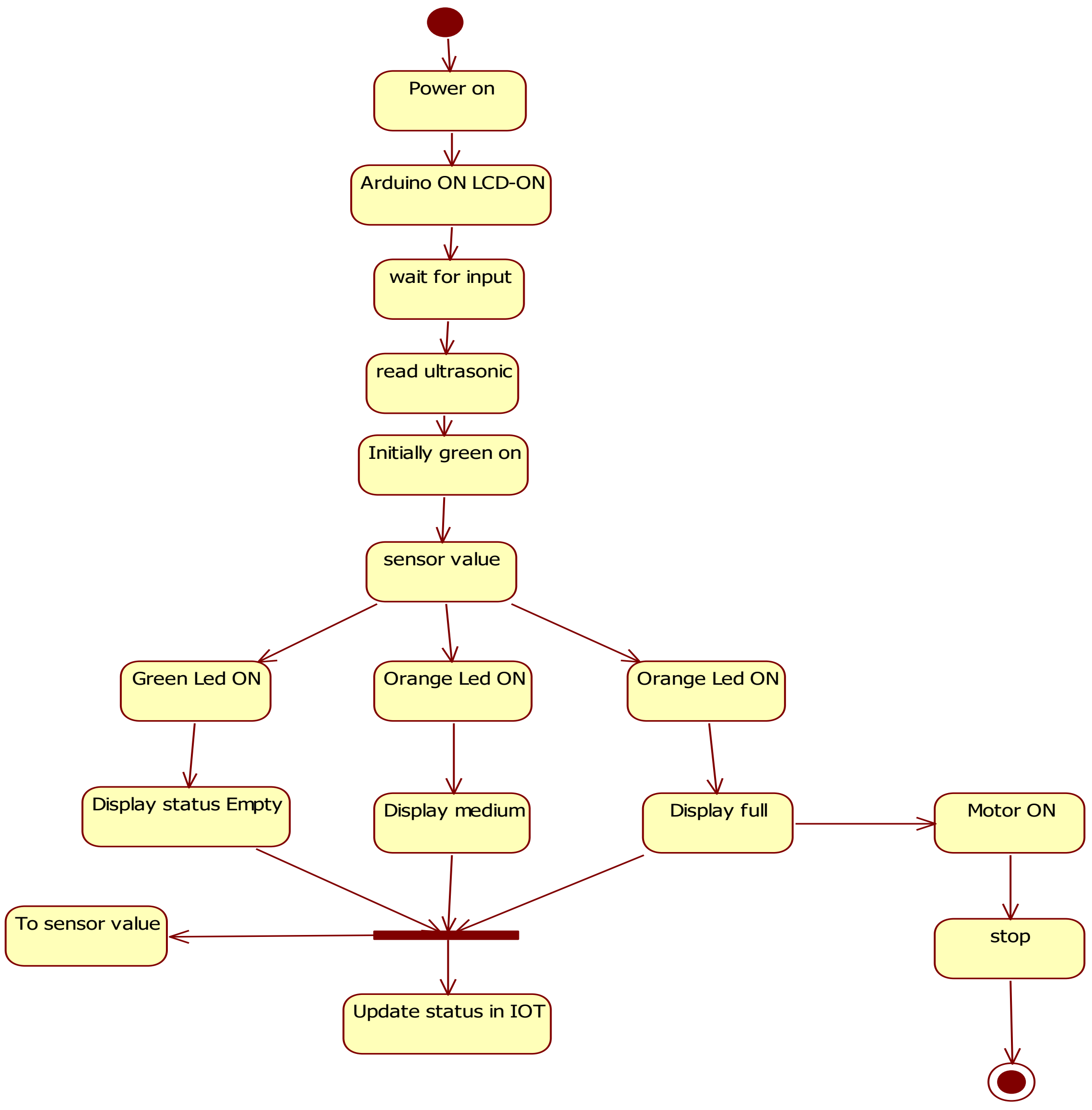


Figure 4.11 State diagram

Explanation

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist,which differ slightly and have different semantics.

Activity Diagram:

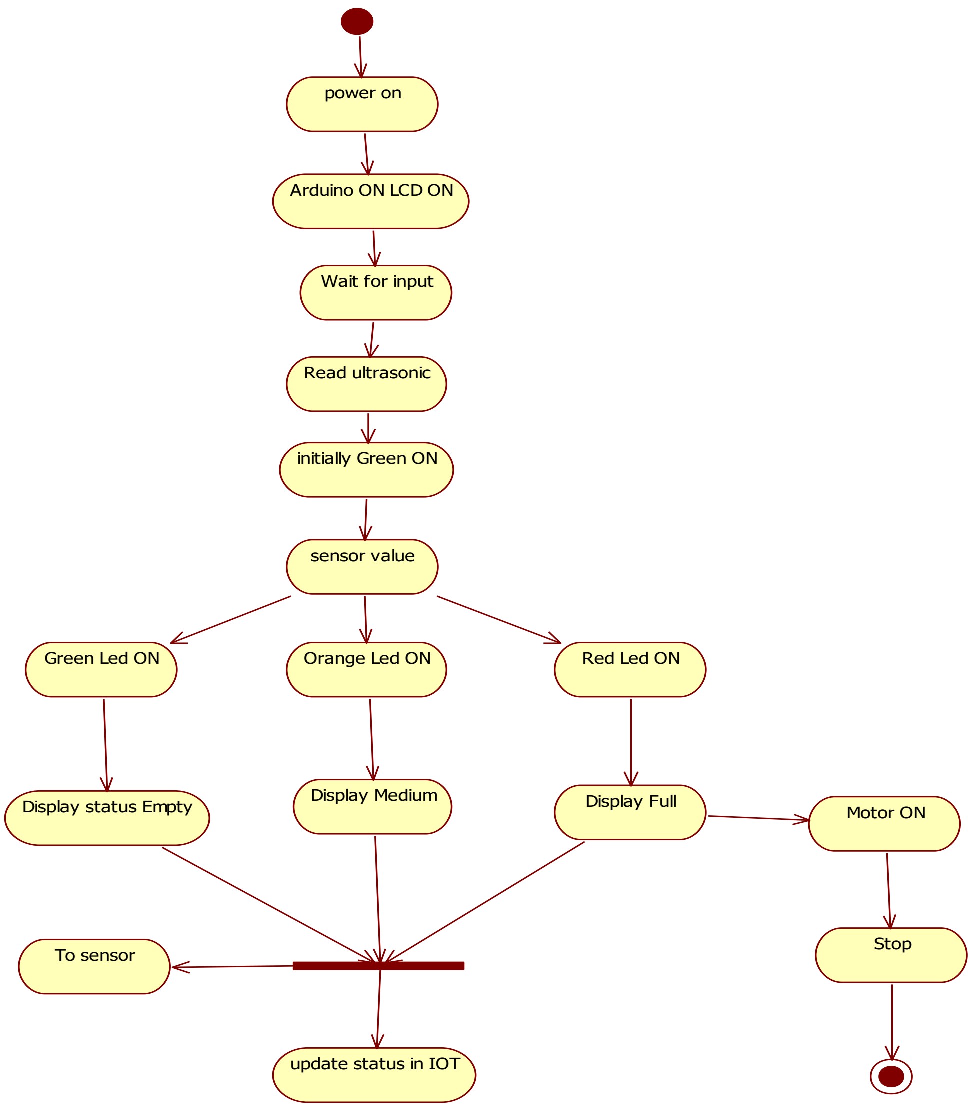


Figure 4.12 Activity diagram

Explanation

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

CHAPTER 5

SOFTWARE SPECIFICATION

5.1 TECHNOLOGIES USED:

The platform used here is IOT. The Primary languages are used in IOT are C,C++, . In this project “C” is choosen for implementation.

Internet of things (IoT)

Internet of things (IoT), is another advance technology in IT sector, provides internetworking for numerous of devices such as sensors, actuators, PLCs and other electronic embedded smart devices and controls, and various software’s’ and provides systems network configuration and connectivity, which enables communication between these numerous devices for information exchanging.

Nowadays, IoT is one of the most advanced, efficient, and cost less technological solution which encompasses various hardware and software resources; and allows remotely connected sensing devices to sense with more capabilities, provides efficiency and can be monitored and controlled through deployed of existing systems or infrastructures, resulting the physical World integration with computer controllers (or systems). As IoT provides interconnectivity among various real-time sensing sensors and PLC and other intelligent devices, therefore this technology will be an entity indicated for the more advance cyber-systems encircling the significant developments, “such as smart grid, smart vehicle systems, smart medical systems, smart cities, and others smart systems.” In early future, IoT has striven to provide advance or smart connectivity for variety of electronic and intelligent equipment’s or devices, IT-based systems and the more advanced services through deploying of various traditional and real-time protocols, networks domains, and system software/hardware applications, which will be an work followed by machine-to-machine technological concept. Through interconnection of various devices and managing ofThe internet of things (IoT) is the network of physical devices, vehicles, buildings and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "the infrastructure of the information society. The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020. IoT is a system defines an environment that encompasses numerous of objects; sensors that connected with these objects are accessible over the Internet through employing of various Networks connections, such wired or wireless.

5.2 Development Tool

Here we use ardunio ide to develop the code (or) to implement the code to our project. The code must be uploaded my using the ardunio ide through ARDUNIO via PC's, Laptops etc.

The Arduino Integrated Development Environment
Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing sketches
Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.
NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Verify
Checks your code for errors compiling it.

Upload
Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New
Creates a new sketch.

Open
Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbookmenu instead.

Save
Saves your sketch.

Serial Monitor
Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- New
  Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open
  Allows to load a sketch file browsing through the computer drives and folders.
- Open-Recent Recent
  Provides a short list of the most recent sketches, ready to be opened.
-
- Sketchbook
  Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- Examples
  Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- Close
  Closes the instance of the Arduino Software from which it is clicked.
- Save
  Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- Save-as.. as...
  Allows to save the current sketch with a different name.
- Page-Setup Setup
  It shows the Page Setup window for printing.
- Print
  Sends the current sketch to the printer according to the settings defined in Page Setup.
- Preferences
  Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.

- Quit
  Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

  Edit

- Undo/Redo
  Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- Cut
  Removes the selected text from the editor and places it into the clipboard.
- Copy
  Duplicates the selected text in the editor and places it into the clipboard.
- Copy for Forum
  Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- Copy as HTML
  Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- Paste
  Puts the contents of the clipboard at the cursor position, in the editor.
- Select All
  Selects and highlights the whole content of the editor.
- Comment/Uncomment
  Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent
  Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- Find
  Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next
  Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- Find Previous
  Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

  Sketch

- Verify/Compile
  Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

•

- Upload
  Compiles and loads the binary file onto the configured board through the configured Port.
-
- Upload Using Programmer
  This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.
- Export Compiled Binary
  Saves a .hex file that may be kept as archive or sent to the board using other tools.
- Show Sketch Folder
  Opens the current sketch folder.
- Include Library
  Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- Add File...
  Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

Tools:

- Auto-Format
  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- Archive-Sketch
  Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- FixEncoding&Reload
  Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- Serial-Monitor
  Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- Board
  Select the board that you're using. See below for descriptions of the various boards.
- Port
  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- Programmer
  For selecting a harware programmer when programming a board or chip and not using the

onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.

- Burn-Bootloader
  The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

Help
Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

- Find in Reference
  This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook
The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File >Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Uploading
Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board.

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #includestatements from the top of your code. There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library. To write your own library, see this tutorial.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory. For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

Serial monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the serial monitor. You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

Preferences

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

Language support

Since version 1.0.1 , the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

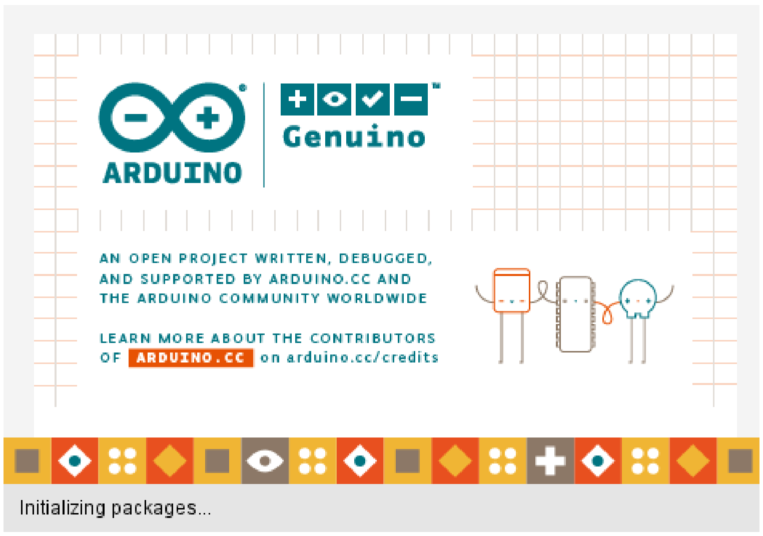
Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader.

5.3 SAMPLE CODE:

```
unsigned int i=0,a,b,c,d;
const int uPin = 13;
const int rPin = 10;
const int yPin = 9;
const int gPin = 8;
const int m1Pin = 11;
const int m2Pin = 12;
int uState = 0;
#define trig1 A0
#define echo1 A1
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>

long duration , cm, inches, inches1;
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
//SoftwareSerial mySerial(10, 11); // RX, TX
void sendwifi(String chr,unsigned int len)
{
  String temp[20];
  Serial.print("AT+CIPSEND=0,");
  Serial.println(len);
  delay(1000);
  Serial.println(chr);
  delay(2000);

}
void setup() {
  pinMode(gPin, OUTPUT);
pinMode(yPin, OUTPUT);
pinMode(m1Pin, OUTPUT);
pinMode(m2Pin, OUTPUT);
pinMode(uPin, INPUT);
pinMode(rPin, OUTPUT);
pinMode(trig1,OUTPUT);
pinMode(echo1, INPUT);
  lcd.begin(16,2);
 lcd.clear();lcd.setCursor(0, 0);lcd.print("welcome");
//mySerial.begin(9600);
Serial.begin(9600);

delay(1000);
lcd.clear();lcd.setCursor(0, 0);lcd.print("WIFI INIT");
Serial.print("AT\r\n");
delay(1000);
Serial.print("ATE0\r\n");
delay(1000);

Serial.print("AT+CIPMUX=1\r\n");
delay(1000);
Serial.print("AT+CIPSERVER=1,23\r\n");
delay(1000);
  lcd.clear();lcd.setCursor(0, 0);lcd.print("192.168.4.1");
  lcd.setCursor(0, 1);lcd.print("Port: 23");
   delay(1000);
     lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTED");
  sendwifi("WELCOME\r\n",9);
  delay(1000);



}
void distance()
{
  digitalWrite(trig1,LOW);
delayMicroseconds(5);
  digitalWrite(trig1,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig1,LOW);
  delayMicroseconds(5);
  duration = pulseIn(echo1,HIGH);
  inches=(duration/2)/74;

  Serial.print("Distance In1:");
  Serial.println(inches);
  delay(2000);
}
void loop() {
  loop1:
  if (uState == LOW)
{
  lcd.begin(16,2);
     lcd.print("IR DETECTED");
     sendwifi("IR DETECTED\r\n",strlen("IR DETECTED\r\n"));
     delay(2000);
  distance();
  if(inches < 50){
    lcd.begin(16,2);
     lcd.print("LOAD:");
    i++;
    a=i/100;
     b=i%100;
     c=b/10;
     d=b%10;
     a=a|0x30;
     c=c|0x30;
     d=d|0x30;
lcd.print(a);
lcd.print(c);
lcd.print(d);
  delay(500);
  }
if(i<5 && i>0)
{
  lcd.clear();
   digitalWrite(gPin, HIGH);
   digitalWrite(yPin, LOW);
   digitalWrite(rPin, LOW);
  lcd.setCursor(0, 1);
  lcd.print("NORMAL LEVEL");
  delay(1000);
  sendwifi("NORMAL LEVEL\r\n",strlen("NORMAL LEVEL\r\n"));
  delay(1000);
}
if(i<10 && i>5)
{
  lcd.clear();
   digitalWrite(gPin, LOW);
   digitalWrite(yPin, HIGH);
   digitalWrite(rPin, LOW);
  lcd.setCursor(0, 1);
  lcd.print("MEDIUM LEVEL");
  delay(1000);
  sendwifi("MEDIUM LEVEL\r\n",strlen("MEDIUM LEVEL\r\n"));
  delay(1000);
}
if(i<16 && i>14)
{
  lcd.clear();
   digitalWrite(gPin, LOW);
   digitalWrite(yPin, LOW);
   digitalWrite(rPin, HIGH);
  lcd.setCursor(0, 1);
  lcd.print("HIGH LEVEL");
  delay(1000);
  sendwifi("HIGH LEVEL\r\n",strlen("HIGH LEVEL\r\n"));
  delay(1000);
  digitalWrite(m1Pin, HIGH);
   digitalWrite(m2Pin, LOW);
   delay(500);
   digitalWrite(m1Pin, LOW);
   digitalWrite(m2Pin, LOW);
   delay(500);
   digitalWrite(m1Pin, LOW);
   digitalWrite(m2Pin, HIGH);
   delay(500);
   digitalWrite(m1Pin, LOW);
   digitalWrite(m2Pin, LOW);
   delay(500);
}
}
else
{
  lcd.begin(16,2);
     lcd.print("WELCOME");delay(2000);
}
}
```

5.4 SNAPSHOTS:

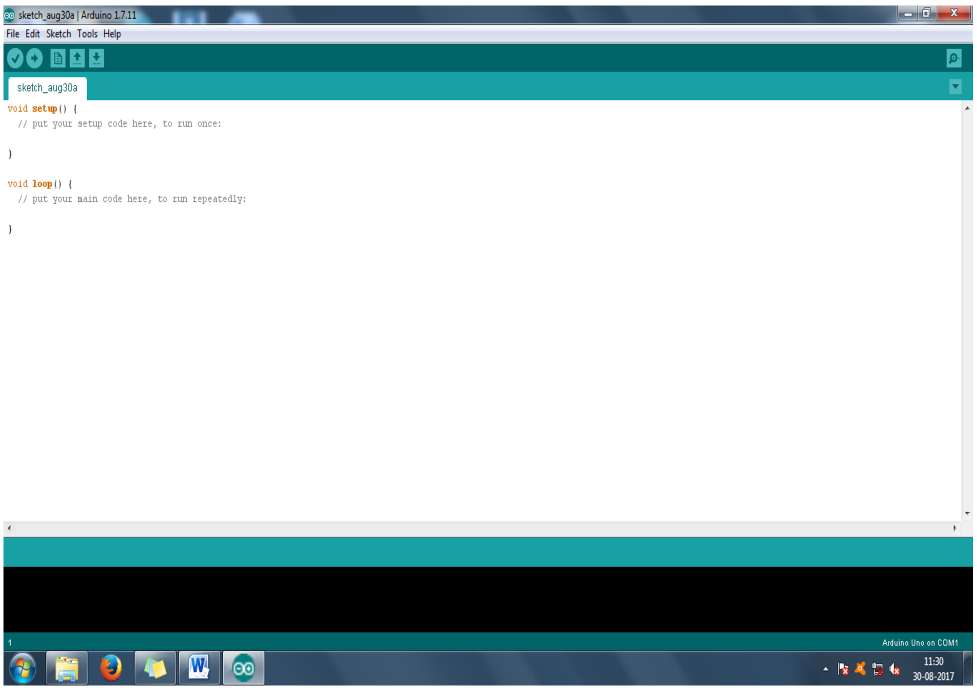


AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

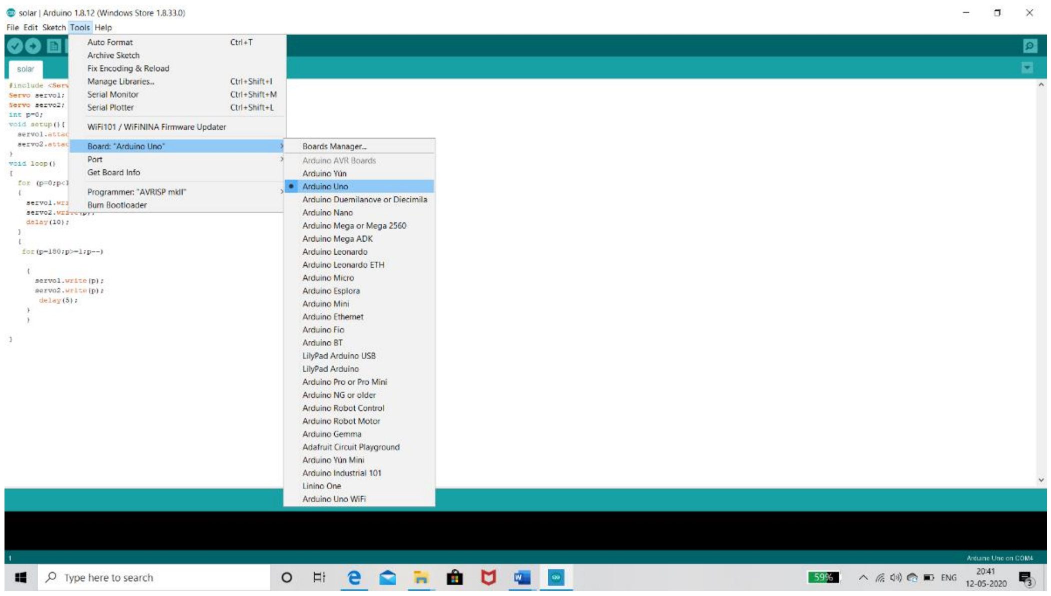
Initializing packages...

Snapshot 1 of opening the Ardunio software



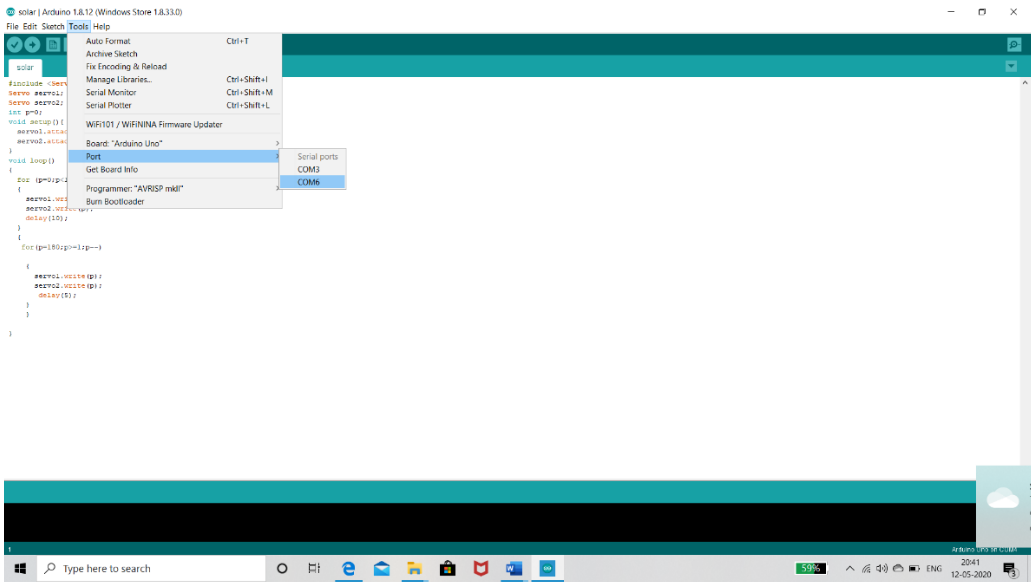
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Snapshot 2 showing the Ardunio IDE file



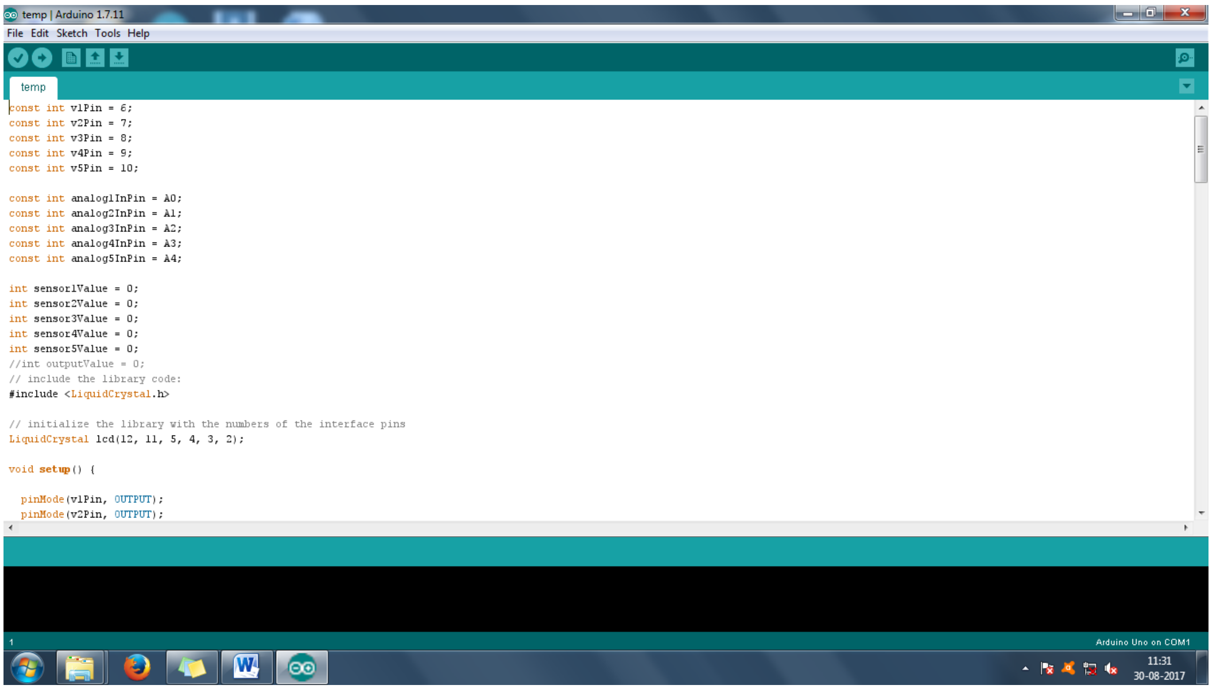
```
#include <Serv
Servo servo1;
Servo servo2;
int p=0;
void setup(){
  servo1.attac
  servo2.attac
}
void loop()
{
  for (p=0;p<1
  {
   servo1.wri
   servo2.write(p);
   delay(10);
  }
  {
   for(p=180;p>=1;p--)

    {
     servo1.write(p);
     servo2.write(p);
      delay(5);
    }
    }

}
```

Snapshot 3 adjusting the board


solar | Arduino 1.8.12 (Windows Store 1.8.33.0)
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM3
COM6
Type here to search
ENG
20:41
12-05-2020

```
#include <Serv
Servo servo1;
Servo servo2;
int p=0;
void setup(){
  servo1.atta
  servo2.atta
}
void loop()
{
  for (p=0;p<1
  {
   servo1.wr
   servo2.write(p);
   delay(10);
  }
  {
   for(p=180;p>=1;p--)

    {
     servo1.write(p);
     servo2.write(p);
      delay(5);
    }
    }

}
```

Snapshot 4 adjusting the port



```
const int v1Pin = 6;
const int v2Pin = 7;
const int v3Pin = 8;
const int v4Pin = 9;
const int v5Pin = 10;

const int analog1InPin = A0;
const int analog2InPin = A1;
const int analog3InPin = A2;
const int analog4InPin = A3;
const int analog5InPin = A4;

int sensor1Value = 0;
int sensor2Value = 0;
int sensor3Value = 0;
int sensor4Value = 0;
int sensor5Value = 0;
//int outputValue = 0;
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

  pinMode(v1Pin, OUTPUT);
  pinMode(v2Pin, OUTPUT);
```

Snapshot 5 of code compiling

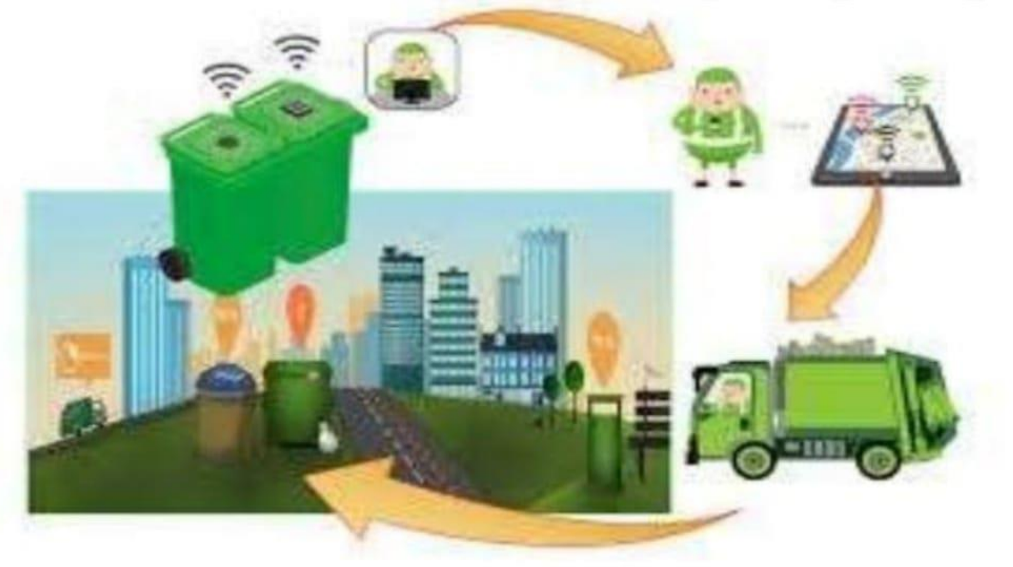
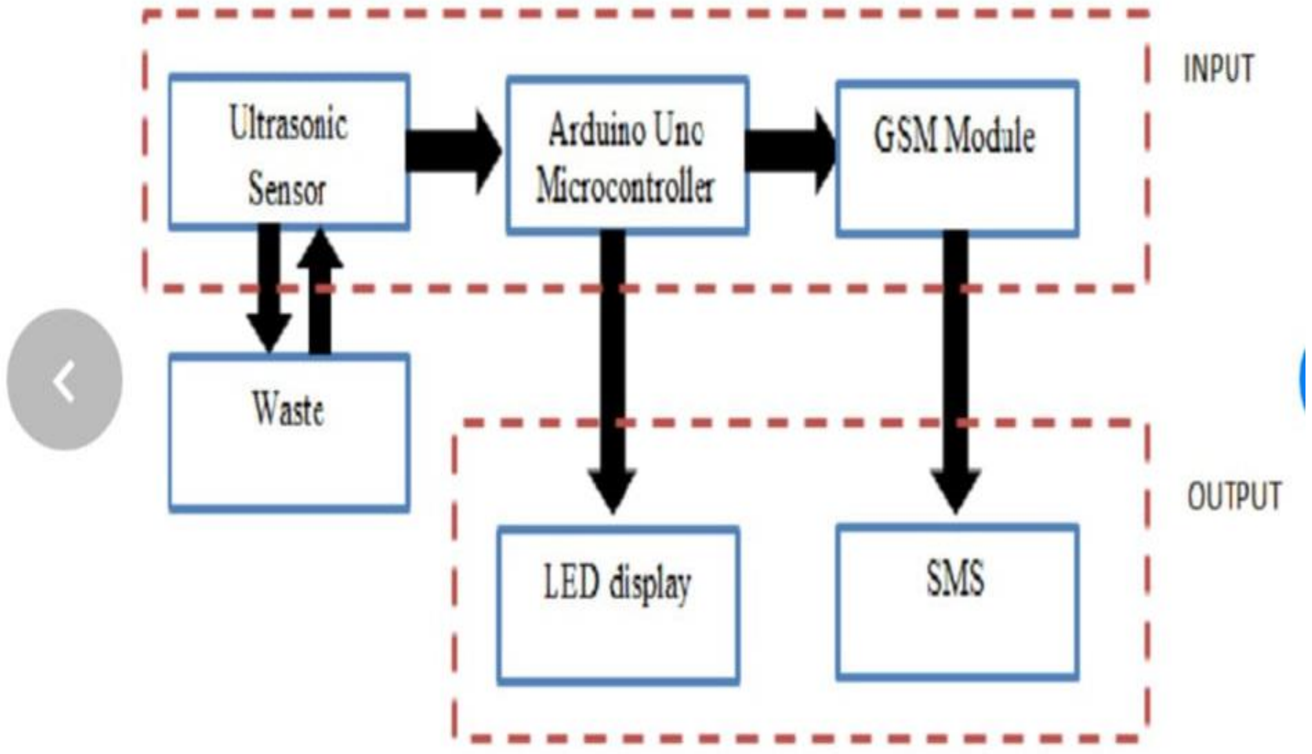


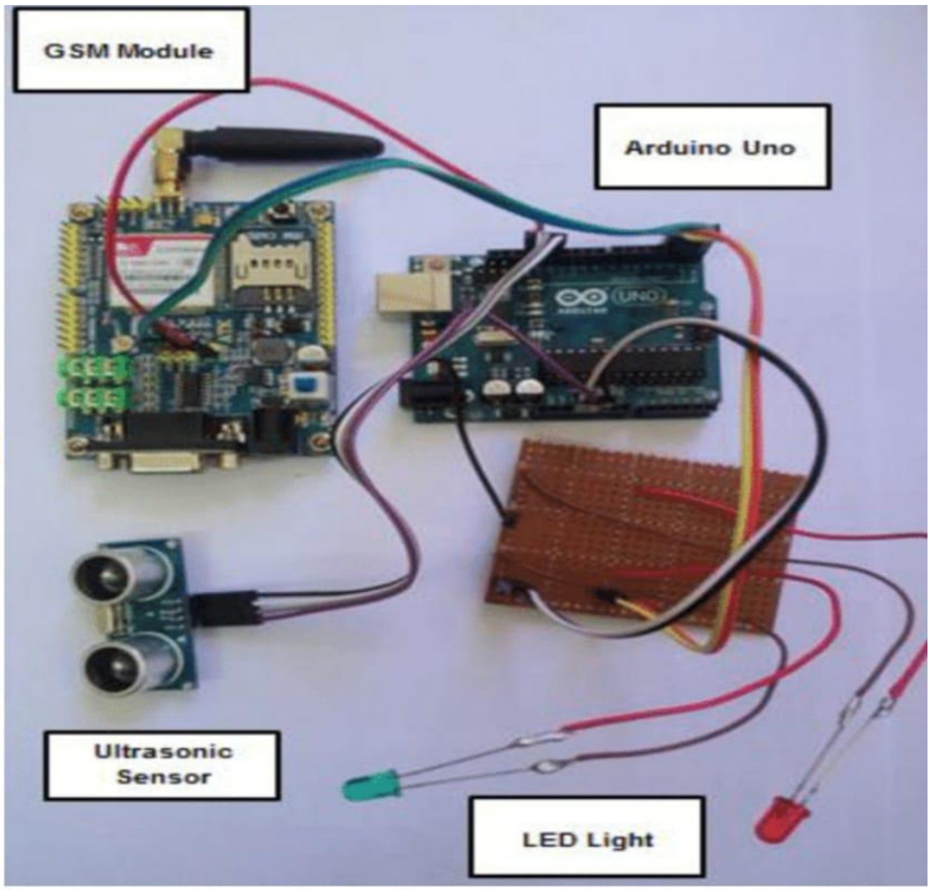
Fig. Concept of Smart Dustbin System

Snapshot 6 of Concept of Smart Dustbin System

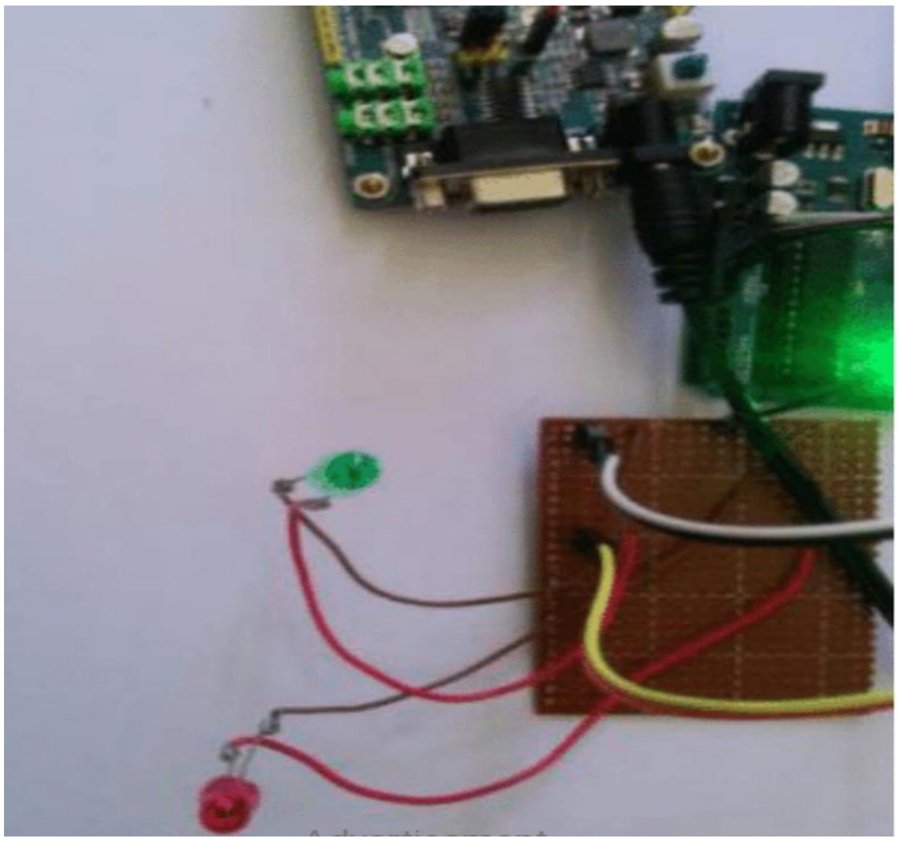


Waste Management Monitoring System Block Diagram

Snapshot 7 of Waste Management Monitoring Syatem Block Diagram



Snapshot 8 of FINAL MODULE OF IOT


Snapshot 9 of MODULE REPRESENTING THE CONNECTON OF WIRES

CHAPTER 6

SOFTWARE TESTING

6.1 GENERAL

IOT is a connection of identifiable embedded devices with the existing Internet infrastructure. IOT testing is a type of testing to check IOT devices. Software testing in IOT is used for testing the code for functioning of all the IOT devices. Today there is increasing need to deliver better and faster services. There is a huge demand to access, create, use and share data from any device. The thrust is to provide greater insight and control, over various interconnected IOT devices. Hence, IOT testing framework is important. Gray Box testing should be used with IOT testing as it allows to design effective test case. This allows you to know the OS, the architecture, third-party hardware, new connectivity and hardware device limitation. Real Time Operating System is vital to delivering the scalability, modularity, connectivity, security, which is important for IOT and it should be automated.

6.2 TYPES OF TESTING

Usability Testing:

There are so many devices of different shape and form factors are used by the users. Moreover, the perception also varies from one user to other. That's why checking usability of the system is very important in IoT testing.

Compatibility Testing:

There are lots of devices which can be connected though IOT system. These devices have varied software and hardware configuration. Therefore, the possible combination are huge. As a result, checking the compatibility in IOT system is important.

Reliability and Scalability Testing:

Reliability and Scalability is important for building an IOT test environment which involves simulation of sensors by utilizing virtualization tools and technologies.

Data Integrity Testing:

It's important to check the Data integrity in IOT testing as it involves large amount of data and its application. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

IOT elements Testing Types	**Sensor**	**Application**	**Network**	**Backend (Data Center)**
Reliability Testing	True	True	False	False
Usability Testing	True	True	False	False
Security Testing	True	True	True	True
Performance Testing	False	True	True	True
Compatibility Testing	True	True	False	False
Data integrity Testing	False	True	True	True

Table 6.1 test cases

6.3 TEST CASES

- Verify that IoT gadget is able to register to the network and data connection is made successfully.
- Verify that IoT gadget transmits data with IoT device application in the form of encrypted data.
- Verify that the code used in IOT projects is error free or not.
- Verify that function of IOT gadgets in projects are working when the code is ready.

6.4 TEST RESULTS

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Performance testing helps to identify the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors. Security testing helps to identify the theft in our smart home system project.

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

In this paper, we propose a new solution to enhance waste collection efficiently using the Arduino Uno with Arduino Ethernet Shield technology and ultrasonic sensor systems. In this proposed system, the garbage overflow of garbage can be avoided and managed efficiently. This will intimate or send SMS or email to the authorized person through Ubidots platform. The garbage managing system and the facility of collecting the garbage presently doesn't fit tot he current requirement. Hence better facility of collecting garbage and transportation should be provided. Since, this system provides the information when the bin gets completely filled with garbage, it reduces the number of times the arrival of vehicle which collects the garbage. This method finally helps in keeping the environment clean. Thus ,the waste collection is made

7.2 FUTURE ENHANCEMENTS

In future work, the application developed for this solution can be evolved by adding new facilities that can bring to the end user more significant interactions with the management system besides integration with a platform, to calculate the best path in collection routes, seeking efficiency with a lower cost of operating the fleet of trucks. In addition, the investment and operation costs of this solution will be a very interesting study and can be performed as future work.

REFERENCES:

[1] J. Sreenivasan, M. Govindan, M. Chinnasami, and I. Kadiresu,"Solid Waste Management in Malaysia – A Move TowardsSustainability," Waste Manag. An Integr. Visions, vol. 2005, no.April 2005, pp. 55–70, 2012.

[2] M. Adam, M. E. Okasha, O. M. Tawfeeq, M. A. Margan and B.Nasreldeen, "Waste Management System Using IoT," 2018International Conference on Computer, Control, Electrical, andElectronics Engineering (ICCCEEE), Khartoum, 2018, pp. 1-4

[3] S. Aleyadeh and A. M. Taha, "An IoT-Based Architecture for WasteManagement," 2018 IEEE International Conference onCommunications Workshops (ICC Workshops), Kansas City, MO,2018, pp. 1-4.

[4] G. K. Shyam, S. S. Manvi and P. Bharti, "Smart waste managementusing Internet-of-Things (IoT)," 2017 2nd International Conferenceon Computing and Communications Technologies (ICCCT),Chennai, India, 2017, pp. 199-203.

[5] W. Chen, Y. Wang, P. Huang, Y. Huang and M. Tsai, "A Smart IoTSystem for Waste Management," 2018 1st International CognitiveCities Conference (IC3), Okinawa, 2018, pp. 202-203.

[6] S. S. Chaudhari and V. Y. Bhole, "Solid Waste Collection as aService using IoT-Solution for Smart Cities," 2018 InternationalConference on Smart City and Emerging Technology (ICSCET),Mumbai, 2018, pp. 1-5.

[7] E. Al-Masri, I. Diabate, R. Jain, M. H. Lam and S. Reddy Nathala,"Recycle.io: An IoT-Enabled Framework for Urban WasteManagement," 2018 IEEE International Conference on Big Data(Big Data), Seattle, WA, USA, 2018, pp. 5285-5287

[8] S. Paul, S. Banerjee and S. Biswas, "Smart Garbage MonitoringUsing IoT," 2018 IEEE 9th Annual Information Technology,Electronics and Mobile Communication Conference (IEMCON),Vancouver, BC, 2018, pp. 1181-1185.

[9] H. N. Saha, S. Gon, A. Nayak, S. kundu and S. Moitra, "Iot BasedGarbage Monitoring and Clearance Alert System," 2018 IEEE 9thAnnual Information Technology, Electronics and MobileCommunication Conference (IEMCON), Vancouver, BC, 2018, pp.204-208.

[10] S. K. Memon, F. Karim Shaikh, N. A. Mahoto and A. Aziz Memon,"IoT based smart garbage monitoring & collection system usingWeMos & Ultrasonic sensors," 2019 2nd International Conferenceon Computing, Mathematics and Engineering Technologies(iCoMET), Sukkur, Pakistan, 2019, pp. 1-6.

[11] R. N. Subrahmanian, S. K. S, T. Suvidharbabu, and B. Manikandan,“Smart Bin using IoT with Fog Computing,” International Journalof Emerging Technology in Computer Science & Electronics vol.24, no. 6, pp. 13–16, 2017.

[12] S. Murugaanandam, V. Ganapathy and R. Balaji, "Efficient IOTBased Smart Bin for Clean Environment," 2018 InternationalConference on Communication and Signal Processing (ICCSP),Chennai, 2018, pp. 0715-0720.

[13] N. Alsbou, M. A. Samad, M. Alhashem and A. S. A. Abuabed,"Developing a Self-Powered Enlarging Smart Waste Bin," 201814th International Wireless Communications & Mobile Computing
Conference (IWCMC), Limassol, 2018, pp. 683-689.

[14] S. Mdukaza, B. Isong, N. Dladlu and A. M. Abu-Mahfouz,"Analysis of IoT-Enabled Solutions in Smart Waste Management,"IECON 2018 - 44th Annual Conference of the IEEE IndustrialElectronics Society, Washington, DC, 2018, pp. 4639-4644.

[15] D. Evans, “The Internet of Things How the Next Evolution of theInternet” Cisco Internet Business Solutions Group (IBSG) no. April,2011.http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf