IoT BASED PORTABLE SMART CANE FOR VISUALLY IMPAIRED PEOPLE

A MINI PROJECT REPORT

Submitted by

DHANUSH.L	**211420105019**
GOPINATH.S	**211420105027**
HARISH.S.S	**211420105030**
JANESH.M.S	**211420105037**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MAY 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institute, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this Mini project report **"IoT BASED PORTABLE SMART CANE FOR VISUALLY IMPAIRED PEOPLE"** is the bonafide work of **"DHANUSH.L (211420105019), GOPINATH.S (211420105027), HARISH.S.S (211420105030), JANESH.M.S (211420105037)** who carried out the project work under my supervision.

SIGNATURE
Dr. S. SELVI, M.E, Ph.D.
HEAD OF THE DEPARTMENT
PROFESSOR
Department of Electrical and
Electronics Engineering,
Panimalar Engineering College,
Chennai - 600 123

SIGNATURE
Mr. K. P. DINAKARAN, M.E
SUPERVISOR
ASSISTANT PROFESSOR
Department of Electrical and
Electronics Engineering,
Panimalar Engineering College,
Chennai - 600 123

Submitted for End Semester Mini Project Viva Voce held
on ……………. …at Panimalar Engineering College, Chennai.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Our sincere thanks to our Honourable Founder and Chairman **Dr. JEPPIAR, M.A., B.L., Ph.D.,** for his sincere endeavor us in his premier institution.

We would like to express our deep gratitude to our beloved Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., M.Phil., Ph.D.,** for his enthusiastic motivation which inspired us a lot in completing this project and our sincere thanks to our Directors **Mrs. C. VIJAYA RAJESWARI, Dr. C. SHAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for providing us with necessary facilities for the completion of this project.

We would like to express thanks to our Principal**, Dr. K. MANI, M.E., Ph.D.,** for extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr. S. Selvi, M.E., Ph.D.,** Professor and Head of Department of Electrical and Electronics Engineering.

Personally, we thank our guide **K. P. DINAKARAN, M.E.,** Assistant Professor in Department of Electrical and Electronics Engineering for the persistent motivation and support for this project, who at all times was the mentor of germination of this project from a small idea.

We express our sincere thanks to the project coordinators **Dr. S. DEEPA, M.E., Ph.D., & Dr. N. MANOJ KUMAR, M.E., Ph.D.,** in Department of Electrical and Electronics Engineering for the valuable suggestions from time to time at every stage of our project.

Finally, we would like to take the opportunity to thank our family members, faculty and non-teaching staff members of our department, friends, well-wishers who have helped us for the successful completion of our project.

ABSTRACT

Blind peoples are very difficult to move one place to another place. So we introduce the smart stick to help the visually impaired people without the help of others. We use Ultrasonic sensor is used to detect the object obstacles, alert through the buzzer and then Heartbeat sensor is used to monitor the heartbeat values. We use Touch sensor (Panic switch button) if blind people are in panic situation just press the Touch sensor, the helper of blind person gets alert through GSM Module. Water level sensor is used to detect the liquid obstacles present in the stick. IoT (ESP8266) for information about the cane through the web server.

Walking with confidence is one of them, which may have different challenges in different environments or countries. We have considered the Indian context, where outdoor environment are often crowded and noisy. Keeping these challenges in mind, a new smart stick has been developed which is capable of detecting obstacles of any height in front or slightly sideways of the person. The stick gives a fair idea about the distance and the location of obstacles through vibration in the hand. Different frequencies of the generated vibration about the distance of the obstacle.

Real-time experiments have been conducted in different environments by different environments by different people to observe the accuracy of the stick, and the results are quite encouraging. The Smart Cane is an electronic device to assist in obstacle detection for the visually impaired. The Smart Cane is an electronic device that fits as a handle on walking sticks used by visually impaired people. The main aim of our project visually impaired person travel alone without the help of others.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1	**INTRODUCTION**	1
2	**LITERATURE SURVEY**	3
3	**EXISTING METHOD**	5
	3.1 Block diagram	5
	3.2 Description	5
4	**PROPOSED METHOD**	6
	4.1 Block diagram	6
	4.2 Description	6
5	**HARDWARE REQUIREMENTS**	7
	5.1 Arduino UNO	7
	5.1.1 Power	7
	5.1.2 Memory	8
	5.1.3 Pin configuration	10
	5.1.4 USB overcurrent protection	11
	5.1.5 Physical characteristics	11
	5.2 Power supply	11
	5.3 ESP8266 Module	14
	5.3.1 Features	15
	5.4 GSM Module	16
	5.4.1 LED status indicator	18
	5.5 Internet of Things(IoT)	19
	5.5.1 Benefits of IoT	21

5.6 Heartbeat Sensor 22

5.6.1 Specifications 24

5.7 Buzzer 24

5.7.1 Types 25

5.7.2 26

5.8 Ultrasonic Sensor 26

5.8.1 Operation 27

5.8.2 Applications 27

5.8.3 Ultrasonic distance measurement 27

5.8.4 Features of Ultrasonic Sensor 28

5.9 Water level Sensor 28

5.9.1 Specifications 28

5.10 Touch Sensor 29

5.10.1 Working principle 29

5.10.2 Applications 30

6 SOFTWARE REQUIREMTS 31

6.1 Arduino UNO 31

6.2 Embedded C 33

6.2.1 Salient features of language 33

6.2.2 Basic additional features 34

6.2.3 The structure of an Embedded C 36

7 HARDWARE KIT 37

8 CONCLUSION 38

REFERENCES 39

APPENDIX 41

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO
3.1	Block Diagram	5
4.1	Prototype Model	6
5.1	Arduino UNO	8
5.2	Pin configuration of Arduino UNO	10
5.3	Block diagram of basic power supply	11
5.4	Circuit diagram of power supply	13
5.5	EP8266 Module	15
5.6	GSM Module	17
5.7	Parts of GSM Module	18
5.8	Cloud computing	20
5.9	Example of an IoT system	21
5.10	Heartbeat Sensor	22
5.11	Circuit diagram of Heartbeat Sensor	24
5.12	Buzzer	25
5.13	Circuit diagram of Buzzer	25
5.14	Ultrasonic Sensor	26
5.15	Water level Sensor	28
5.16	Touch Sensor	29
6.1	Arduino IDE software	31
6.2	Table of keyword	34

LIST OF ABBREVIATIONS

WHO	World Health Organization
ICT	Information and Communication Technologies
PCB	Printed Circuit Board
PVC	Poly Vinyl Chloride
GPS	Global Positioning System
GSM	Global System for Mobile Communication
IoT	Internet of Things
USB	Universal Serial Bus
FTDI	Future Technology Devices International Limited
DFU	Device Firmware Update
DC	Direct Current
RAM	Random Access Memory
LED	Light Emitting Diode
UART	Universal Asynchronous Receiver - Transmitter
CPU	Central Processing Unit
SoC	System on Chip
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
VLSI	Very Large Scale Integration

CHAPTER 1
INTRODUCTION

According to World Health Organization (WHO), internationally the number of visually impaired people of all ages is projected to be 285 million, of whom 39 million are sightless or blind. People 50 years and older are 82% of all sightless or blind.

The main reasons for visual impairment are uncorrected refractive errors (43%) and cataracts (33%); the first reason for blindness is cataract (51%). Those who are visually impaired have to face problems in education, transportation, traveling, employment, and autonomous living daily. But moving around freely is probably the hardest challenge for them. As they are not depending on their own eyes.

Hence, we leveraged technology to come up with a portable, cost-effective, and reliable device to overcome challenges for visually impaired people, allowing them to walk confidently with maximizing control of their mobile device by wireless communication that performs some key functions by pressing buttons from their mounted device on the white cane.

In this research paper, we have developed an android application that is linked with the portable smart device mounted on a white cane through a Bluetooth module and some other sensors and modules controlled by Arduino UNO.

The device is tested with blind people and gives better results for reliability, user friendly, portability, less weight, and economical so that everyone can easily purchase, mount, and configure to walk more confidently.

And perform a necessary operation and obstacle detection in the range of 5 feet with varying buzzer frequency after every 12 inches to give better understanding of distance to obstacle also the facility to operate mobile from the mounted device such as sending a message to caretakers, dialing a call, help message, SMS read and open Google maps to navigate by a single click on the mounted button on a white cane that wireless communication through Bluetooth.

More importantly, the blind person start an Android application and connect with the device by just shaking the mobile horizontally, the application will be automatically started and connected. Furthermore, it generates a confirmation voice after successful connectivity with the device on the white cane.

The Smart cane is an electronic device to assist in obstacle detection for the visually impaired. The Smart cane is a device that fits as a handle on walking sticks used by visually impaired people. The main aim of our project visually impaired person travel alone without the help of others.

CHAPTER 2

LITERATURE SURVEY

An android-based Portable Smart Cane for Visually Impaired People
Adhil Khan; Kashif Nisar; Sana Nisar; BS Chowdhry; Ehtisham Lodhi; Jawad Khan; Muhammad Reazul Haque 2018.
In today’s encouraging world of technology, mobile applications are a speedily increasing segment of the world wide mobile market. An android operating system is the highest accepted and extremely developing open platform for mobile application development.

Due to the rise of the impaired people population and there are limited technological-based facilities, we want to leverage technology to develop an Information & Communication Technologies (ICT) based smart portable cane for visually impaired people using android application.

Design and Implementation of Smart Cane for Visually Impaired People
Do Ngoc Hung Vo Minh-ThanhQuoc Luong Huy Nguyen Triet
This paper, a smart cane can help visually impaired people to detect the obstacles around them is designed and implemented. A PCB board uses a microcontroller to get data om ultrasonic sensors to determine the distance from the cane to the nearest obstacles.

This distance information will be transferred to the Android smart phone via Bluetooth protocol. In this system blind person will navigate through a cane interfaced with an android application. A blind person can establish voice call or SMS to a predefined number just by pressing the emergency button will get notified as Facebook status updated with emergency alert.

Smart Cane for Blind Person Assisted with Android Application and Save Our Souls Transmission
Pratibha Pawaskar
Today technology is improving daily in different aspects in order to provide flexible proposes Smart Cane (stick) for blind person. But there is no such kind of good system to navigate a blind person and help in emergency situation. In this paper, user friendly device is proposed that can identify the obstacles in the path using ultrasonic sensor.

Design and Implementation of a Solar Powered Navigation Technology for the Visually Impaired

Nisha Sharma, Somen Archarya, Kaushal Kumar, Narendra Singh and O.P. Chaurasia

The Blind Navigation System using Arduino is a system that intends to enhance blind peoples access to the environment, particularly in Ghana, Africa. This research aimed at designing a safe navigation system to allow seamless transitions for visually impaired people from one location to another, as well as a tool to assist them in communicating with their surroundings and guardians when in a difficult situation.

The design uses PVC pipe as the cane, Arduino UNO, ultrasonic and water sensors for processing and monitoring, a buzzer and vibration motor to offer an alarm system via vibration and sound, housed within a circuit box and the handle, and finally powered by a portable mini solar panel with a rechargeable battery.

Smart Blind Helping Stick Using IOT and Android

Srishti Sakahi, Sakshi Vaval, Varsha Chawla, Ms Nupur Agarwal, Ms Kavita Namdev

The term blindness is used for complete or nearly complete vision loss. Blind people need some help while travelling to feel safe. In this paper, we have developed a smart stick which increases the accessibility of blind person to move around and voice output is given when obstacle is or pothole is detected. This smart stick is cheap, lightweight and fast. The stick consists of 3 ultrasonic sensors for obstacle detection and 1 ultrasonic sensor for pothole detection.

An IOT Machine Learning-Based Mobile Sensors Unit for Visually Impaired People

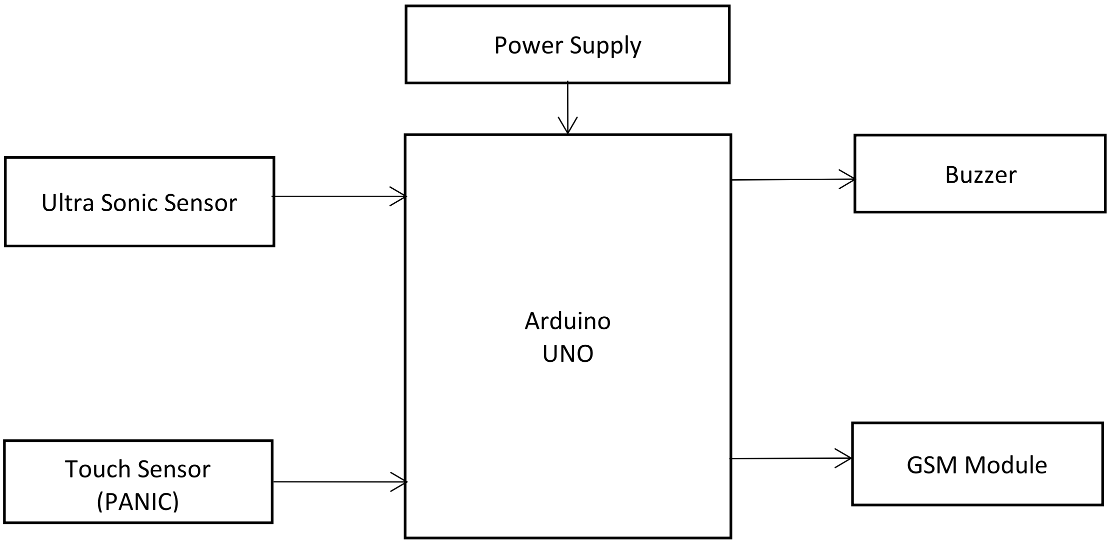
Salam Dhou Ahmed Alnabulsi, A.R. Al-Ali Mariam Arshi, Fatima Darwish, Sara Almaazmi, Reem Alameeri

Visually impaired people face many challenges that limit their ability to perform daily tasks and interact with the surrounding world. Navigating around places is one of the biggest challenges that face visually impaired people, especially those with complete loss of vision. As the Internet of Things (IoT) concept starts to play a major role in smart cities applications, visually impaired people can be one of the benefited clients. In this paper, we propose a smart IoT-based mobile sensors unit that can be attached to an off-the-shelf cane, hereafter a smart cane, to facilitate independent movement for visually impaired people.

CHAPTER 3

EXISTING METHOD

3.1 BLOCK DIAGRAM



3.2 DESCRIPTION

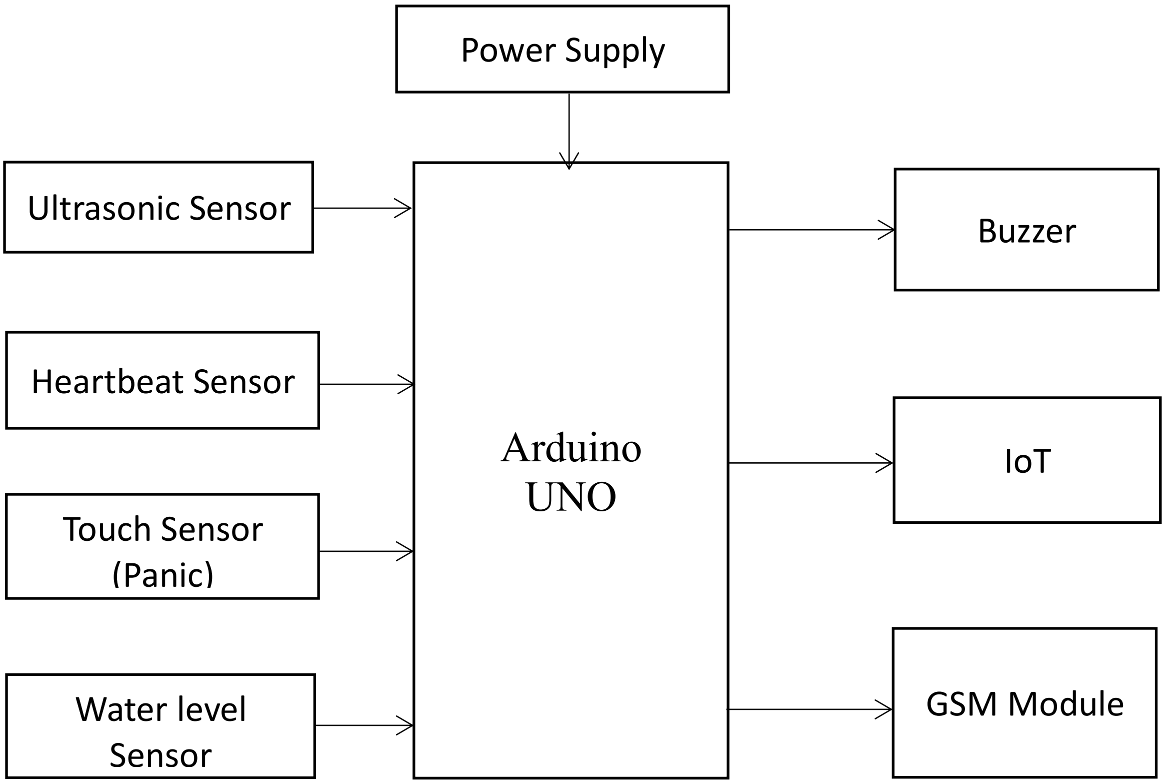
In this project, our major work is on the android application, where we are using existing sensors of a smartphone instead of add additional sensors in the device that causes reduction of cost and decrease the weight and size of the device to make it more smart and economical. The aforementioned sensors are an accelerometer, GSM and speaker. These features make this device unique compare to other existing smart cane.

The Smart cane is used for Blind people to detect obstacle in that smart cane we use ultrasonic sensor, ESP8266(IoT), touch sensor, buzzer, GSM, IOT for information about the cane through the web server using GSM we can sent panic message through phone solar panel battery for power consumption.

CHAPTER 4

PROPOSED METHOD

4.1 BLOCK DIAGRAM



4.2 DESCRIPTION

The Smart cane is used for blind people to detect obstacle in that Smart cane we use Ultrasonic Sensor, ESP8266 (IoT), Touch Sensor, Heartbeat Sensor, Buzzer, GSM Module. We use Ultrasonic Sensor is used to detect the object obstacles, if any object obstacles present automatically buzzer will be ON condition. Heartbeat Sensor is used to detect the Heart beat values. We use Touch Sensor (panic switch button) if blind people are in panic situation just press the panic switch, the helper of blind person gets alert through GSM Module.

GSM Module is used for sending and receiving SMS. Water level Sensor is used to detect the liquid obstacles, if blind people walking in the road there is a drainage pit, it can able to detect the liquid obstacles present in the stick, if any liquid obstacles present automatically buzzer will be ON condition. IoT (ESP8266) for information about the cane through the web server and also we can able to monitor.

CHAPTER 5
HARDWARE REQUIREMENTS

5.1 ARDUINO UNO

The Arduino UNO is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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. The UNO differs from all preceding boards in that it does not use the FTDI USB to serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB to serial converter. Revision 2 of the UNO board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- Added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

- Stronger RESET circuit.

- Atmega 16U2 replace the 8U2. "UNO" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

5.1.1 POWER

The input voltage to the Arduino 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.

- 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.

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

- Ground pins.

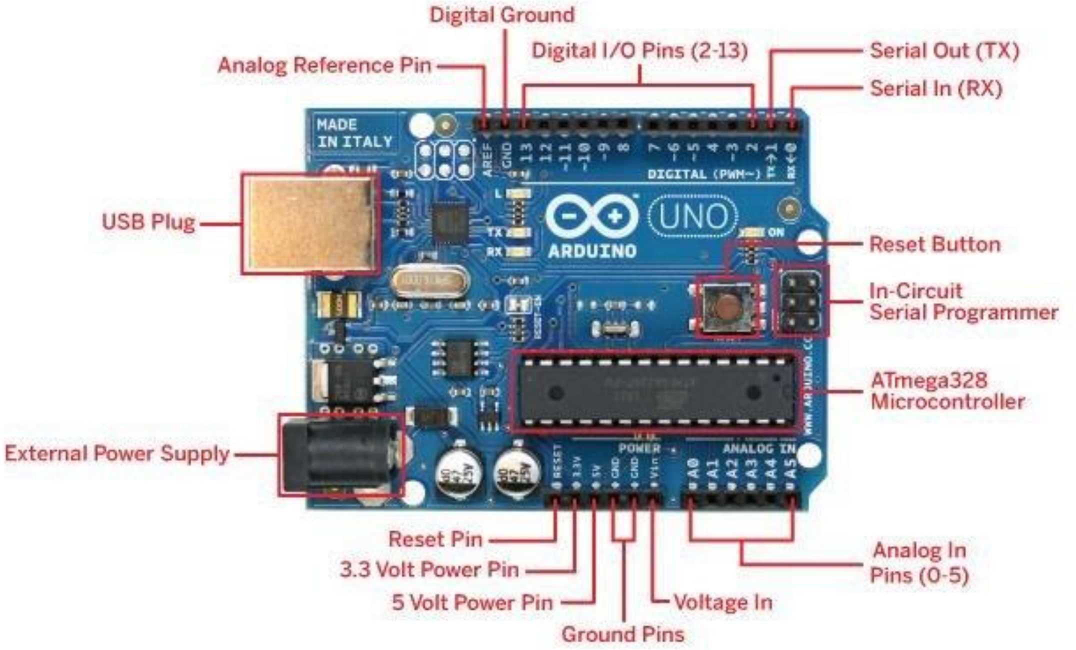


Figure 5.1 Diagram of Arduino UNO

5.1.2 MEMORY

The ATmega328 has 32 KB (with 0.5 KB used for the boot loader). 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 Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts.

Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kilo Ohms.

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 attach Interrupt () 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 connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. 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 analog Reference () function. Additionally, some pins have specialized functionality.

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board.

- AREF: Reference voltage for the analog inputs. Used with analog Reference ().

- 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. See also the mapping between Arduino pins and ATmega328 ports.

The mapping for the Atmega8, 168, and 328 is identical. Communication The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers.

The ATmega328 provides UART TTL (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 includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs 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 Software Serial library allows for serial communication on any of the UNO's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication.

The Arduino software includes a Wire library to simplify use of the I2C bus, see the documentation for details. For SPI communication, use the SPI library.

5.1.3 PIN CONFIGURATION

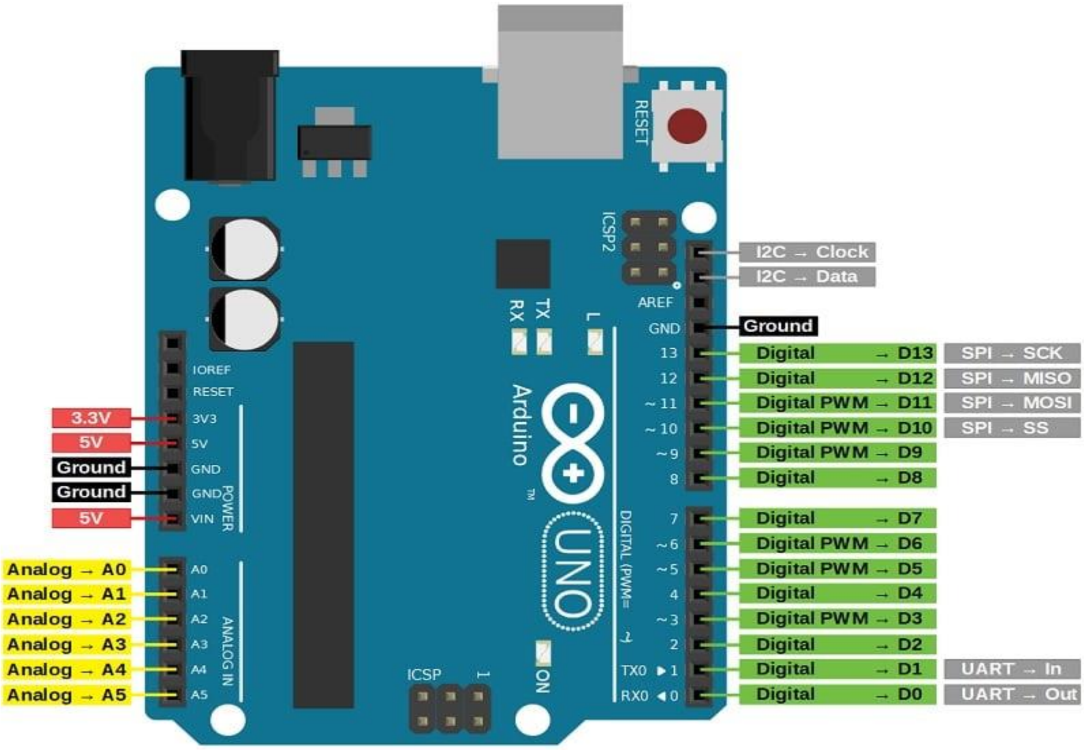


Figure 5.2 Pin configuration of Arduino UNO

5.1.4 USB OVERCURRENT PROTECTION

The Arduino UNO has a re-settable poly fuse 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.

5.1.5 PHYSICAL CHARACTERISTICS

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16), not an even multiple of the 100-mil spacing of the other pins.

5.2 POWER SUPPLY

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

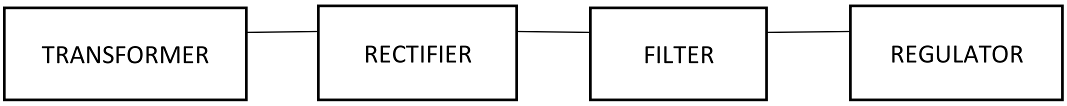


Figure 5.3 Block diagram of basic power supply

The transformer steps up or steps down the input line voltage and isolates the power supply from the power line. The rectifier section converts the alternating current input signal to a pulsating direct current. However, as you proceed in this chapter you will learn that pulsating dc is not desirable. For this reason, a filter section is used to convert pulsating dc to a purer, more desirable form of dc voltage.

The final section, the regulator, does just what the name implies. It maintains the output of the power supply at a constant level in spite of large changes in load current or input line voltages. Now that you know what each section does, let's trace an AC signal through the power supply. At this point you need to see how this signal is altered within each section of the power supply. Later on, in the chapter you will see how these changes take place. In view B of figure 4-1, an input signal of 115 volts AC is applied to the primary of the transformer. The transformer is a step-up transformer with a turn's ratio of 1:3. We can calculate the output for this transformer by multiplying the input voltage by the ratio of turns in the primary to the ratio of turns in the secondary therefore, 115 volts AC´ 3 = 345 volts ac (peak-to- peak) at the output.

Because each diode in the rectifier section conducts for 180 degrees of the 360-degree input, the output of the rectifier will be one-half, or approximately 173 volts of pulsating DC. The filter section, a network of resistors, capacitors, or inductors, controls the rise and fall time of the varying signal.

Consequently, the signal remains at a more constant DC level. We will see the filter process more clearly in the discussion of the actual filter circuits. The output of the filter is a signal of 110 volts dc, with ac ripple riding on the dc. The reason for the lower voltage (average voltage) will be explained. The regulator maintains its output at a constant 110-volt dc level, which is used by the electronic equipment (more commonly called the load).

Simple 5v supply for digital circuits

- Brief description of operation: Gives out well-regulated +5V output, output current capability of 100 mA.
- Circuit protection: Built-in overheating protection shuts down output when regulator IC gets too hot.
- Circuit complexity: Very simple and easy to build.
- Circuit performance: Very stable +5V output voltage, reliable operation.
- Availability of components: Easy to get, uses only very common basic components.
- Applications: Part of electronics devices, small laboratory power supply.
- Power supply voltage: Unregulated DC 8-18V power supply.
- Power supply current: Needed output current +5 mA.

CIRCUIT DESCRIPTION

This circuit is a small +5V power supply, which is useful when experimenting with digital electronics. Those transformers are easily available, but usually their voltage regulation is very poor, which makes then not very usable for digital circuit experimenter unless a better regulation can be achieved in some way. This circuit can give +5V output at about 150mA current, but it can be increased to 1A when good cooling is added to 7805 regulator chip. The circuit has over overload and terminal protection.

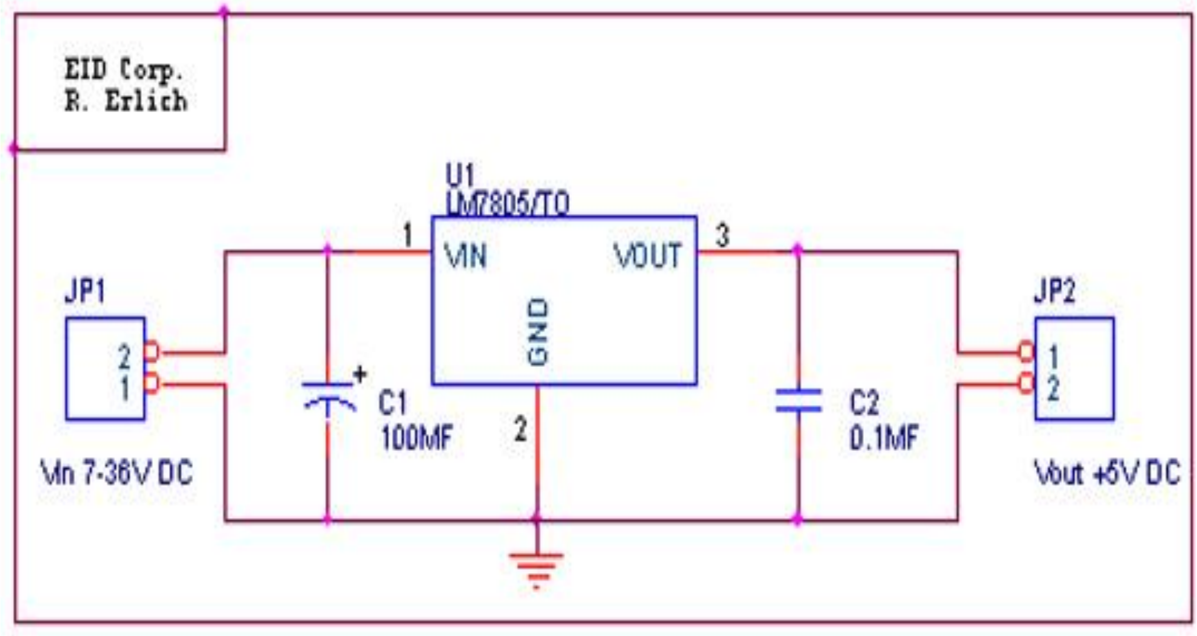


Figure 5.4 Circuit diagram of power supply

CIRCUIT DIAGRAM OF THE POWER SUPPLY

The capacitors must have enough high voltage rating to safely handle the input voltage feed to circuit. The circuit is very easy to build into a piece of Vero board. Pin out of the 7805 regulator IC.

1. Unregulated voltage in
2. Ground
3. Regulated voltage out

COMPONENT LIST

- 7805 regulator IC.
- 100µF electrolytic capacitor, at least 25V voltage rating.
- 10µF electrolytic capacitor, at least 6V voltage rating.
- 100µF ceramic or polyester capacitor.

5.2.1 MORE OUTPUT CURRENT

If we need more than 150mA of output current, we can update the output current up to 1A doing the following modifications:

- Change the transformer from where we take the power to the circuit to a model which can give as much current as we need from output.
- Put a heat sink to the 7805 regulators (so big that it does not overheat because of the extra losses in the regulator).

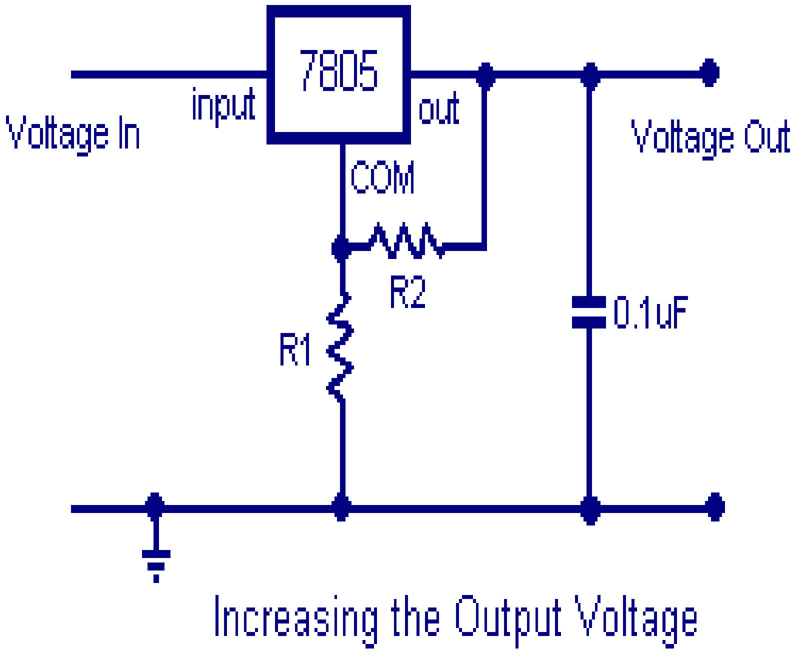


Figure 5.4 Circuit diagram of power supply

OTHER OUTPUT VOLTAGES

If we need other voltages than +5V, we can modify the circuit by replacing the 7805 chips with another regulator with different output voltage from regulator 78xx chip family. The last numbers in the chip code tells the output voltage. The input voltage must be at least 3V greater than regulator output voltage to otherwise the regulator does not work well.

5.3 ESP8266 MODULE

The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by manufacturer Espressif Systems in Shanghai, China. The chip first came to the attention of western makers in August 2014 with the ESP-01 module, made by a third-party manufacturer AI-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands.

However, at first there was almost no English language documentation on the chip and the commands it accepted. The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, chip, and the software on it, as well as to translate the Chinese documentation.

5.3.1 Features:

- Memory
- 32 KB instruction RAM
- 32 KB instruction cache RAM
- 80 KB user-data RAM
- 16 KB ETS system-data RAM
- External QSPI flash: up to 16 MB is supported (512 KB to 4 MB typically included)
- IEEE 802.11 b/g/n Wi-Fi Integrated TR switch, balun, LNA, power amplifier and matching network
- WPA/WPA2 authentication, or open networks
- 16 GPIO pins
- SPI
- I^{2}C (software implementation)
- I^{2}S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2

- 10-bit ADC (successive approximation ADC) ESP8266 is Wi-Fi enabled system on chip (SoC) module developed by Espress if system. It is mostly used for development of IoT (Internet of Things) embedded applications.

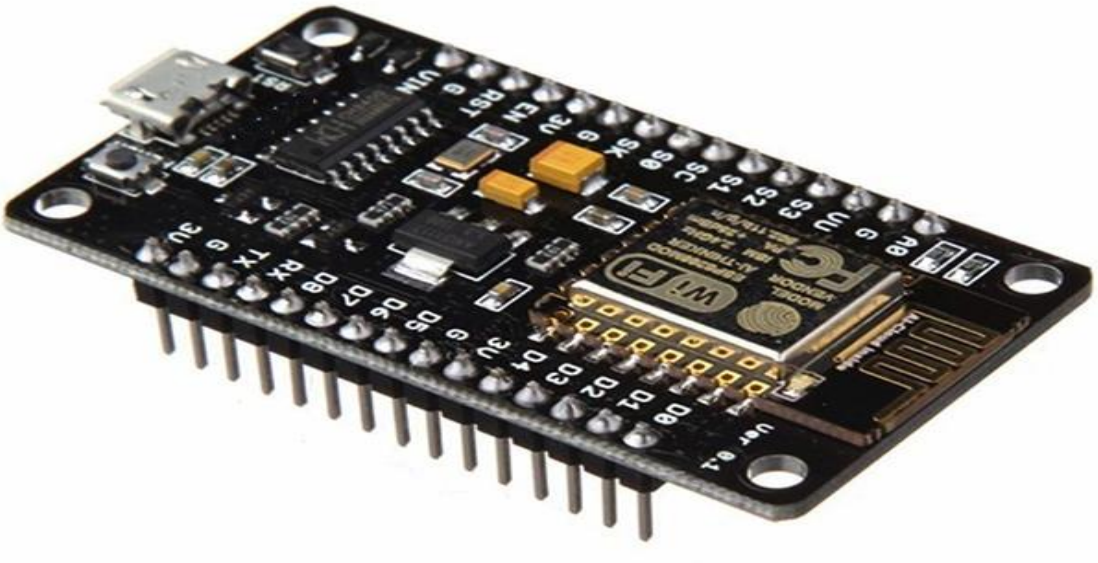


Figure 5.5 ESP8266 module

ESP8266 comes with capabilities of
- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2)
- general-purpose input/output (16 GPIO)
- Inter-Integrated Circuit (I²C) serial communication protocol
- analog-to-digital conversion (10-bit ADC)
- Serial Peripheral Interface (SPI) serial communication protocol
- I²S (Inter-IC Sound) interfaces with DMA (Direct Memory Access) (sharing pins with GPIO)
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2)
- pulse-width modulation (PWM). It employs a 32-bit RISC CPU based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed through SPI.

ESP8266 module is low-cost standalone wireless transceiver that can be used for end-point IoT developments.

To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate.

There are many third-party manufacturers that produce different modules based on this chip. So, the module comes with different pin availability options like

- ESP-01 comes with 8 pins (2 GPIO pins) – PCB trace antenna.
- ESP-02 comes with 8 pins, (3 GPIO pins) – U-FL antenna connector.
- ESP-03 comes with 14 pins, (7 GPIO pins) – Ceramic antenna.
- ESP-04 comes with 14 pins, (7 GPIO pins) – No ant. etc.

For example, below figure shows ESP-01 module pins.

5.4 GSM MODULE

SIM800L is a very compact GSM/GPRS module that can be integrated into a lot of IoT projects due to its compact size and flexibility. This module can do anything that a normal cell phone can do like text message, make and receive phone calls and connect to the internet through GPRS and on top of that, it supports quad-band GSM/GPRS means it can work anywhere in the world.

The SIM800L GSM/GPRS module has 12 pins which are NET, VCC, RST, RXD, TXD, GND, SPK-, SPK+, MIC-, MIC+, DTR, RING.

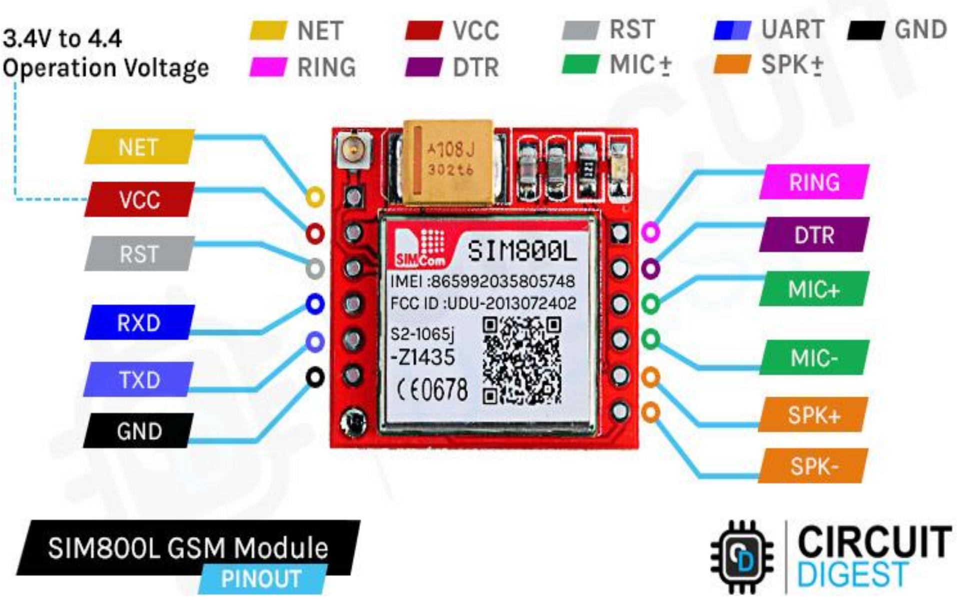


Figure 5.6 GSM module

NET is a pin where you can solder the helical antenna that comes with the module.

VCC is the Power supply pin of the module and it needs to be powered anywhere from 3.4V to 4.4 volts. Connecting this module to a 5V supply will most likely destroy it and if you connect it to 3.3V it won't even run. A lithium battery or a buck converter with 2A current capacity is recommended for this module.

RST is the hard reset pin of the sim800L module. If you are having trouble communicating with this, pull the pin low for 100ms.

RXD is the RX pin for the module used in serial communication, TXD is the TX pin for this module used in Serial communication.

GND is the Ground pin for this module; connect this pin to the Ground pin of the ESP32.

RING is the ring indicator pin of the module. This pin generally is active high. It will go low for 120ms to indicate incoming calls and can also be configured to pulse when an SMS is received.

DTR this pin can be used to put the module in sleep mode. Pulling the pin high puts the module in sleep mode and disables the serial. Pulling it low will wake the module up.

MIC+- These two pins can be used to connect an external microphone to the module.

SPK+- these two pins can be used to connect an external speaker to the module.

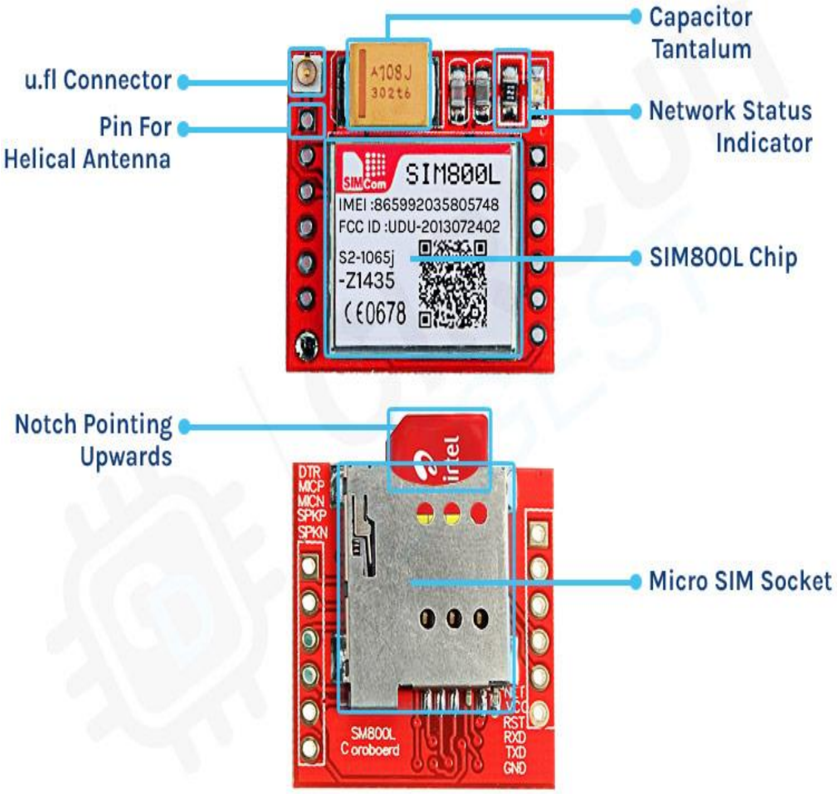


Figure 5.7 parts of GSM module

The SIM800L module is a compact, versatile, and easy-to-use module for GSM and GPRS.

If you take a closer look at the SIM800L module there is not much on the PCB. On the front side of the PCB, we have the UFL connector and the SIM800L module itself. We also have some capacitors for decoupling and we have a 1K current limiting resistor for the LED. Finally, we have a big 100µF,16V tantalum capacitor on board. On the backside of the board we have the sim holder that is a push-to-lock type connector. This means you just need to insert a SIM card and push it for the card to work.

5.4.1 LED Status Indicator:

As we have mentioned earlier, there is an LED indicator on the top of the SIM800L module. It will blink at various rates to indicate network conditions.

- Blink every one second:

If the LED on the module is blinking every second, it indicates that the module is running but it’s unable to connect to the cellular network right now.

- Blink every two seconds:

When the onboard LED on the monitor blinks every two seconds this means the GPRS data connection you requested is active and ready to accept requests on demand.

- Blink every three seconds:

When the LED on the module is blinking every three seconds, the module is connected to a network and can send/receive voice and SMS.

5.5 INTERNET OF THINGS (IoT)

• The term Internet of Things generally refers to scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate, exchange and consume data with minimal human intervention.

• Enabling Technologies: The concept of combining computers, sensors, and networks to monitor and control devices has existed for decades. The recent confluence of several technology market trends, however, is bringing the Internet of Things closer to widespread reality. These include Ubiquitous Connectivity, Widespread Adoption of IP-based Networking, Computing Economics, Miniaturization, Advances in Data.

• Connectivity Models: IoT implementations use different technical communications models, each with its own characteristics. Four common communications models described by the Internet Architecture Board include: Device-to-Device, Device-to-Cloud, Device-to-Gateway, and Back-End Data-Sharing. These models highlight the flexibility in the ways that IoT devices can connect and provide value to the user.

IoT devices are implemented using both hardware and software components. Dedicated hardware components are used to implement the interface with the physical world, and to perform tasks which are more computationally complex. Microcontrollers are used to execute software that interprets inputs and controls the system. This module discusses the roles of both the hardware and software components in the system.

The functions of common hardware components are described and the interface between the software and hardware through the microcontroller is explained. IoT devices often use an operating system to support the interaction between the software and the microcontroller. We will define the role of an operating system in an IoT device and how an IoT operating system differs from a standard one.

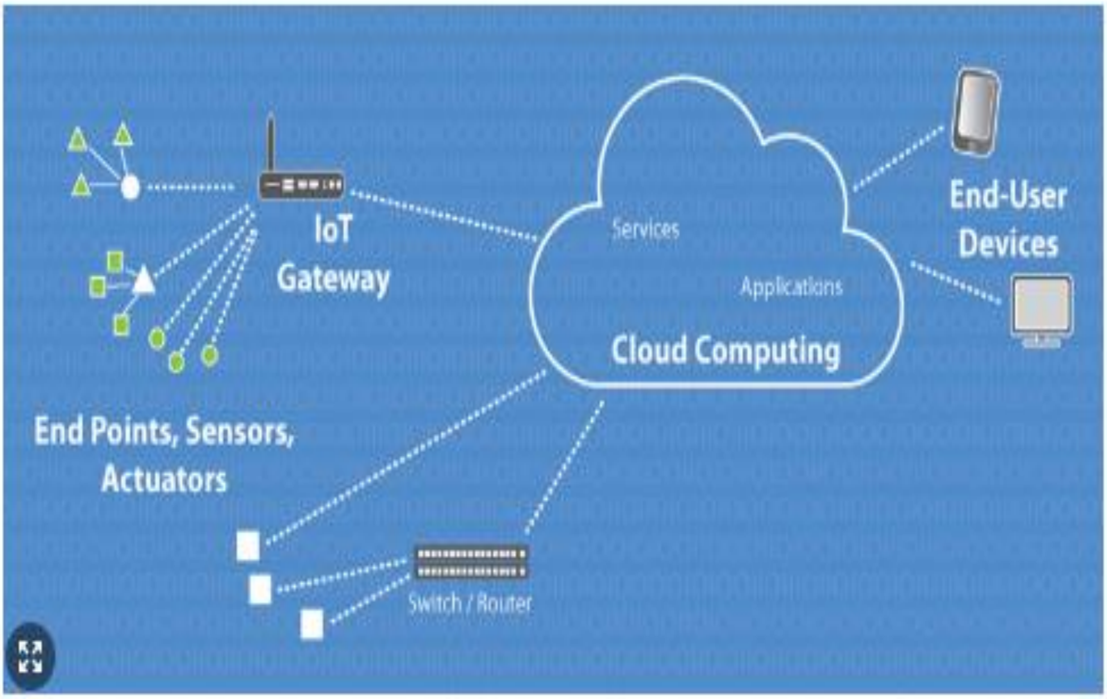


Figure 5.8 Cloud computing

How IoT works:

An IoT ecosystem consists of web-enabled smart devices that use embedded processors, sensors and communication hardware to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally.

Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.

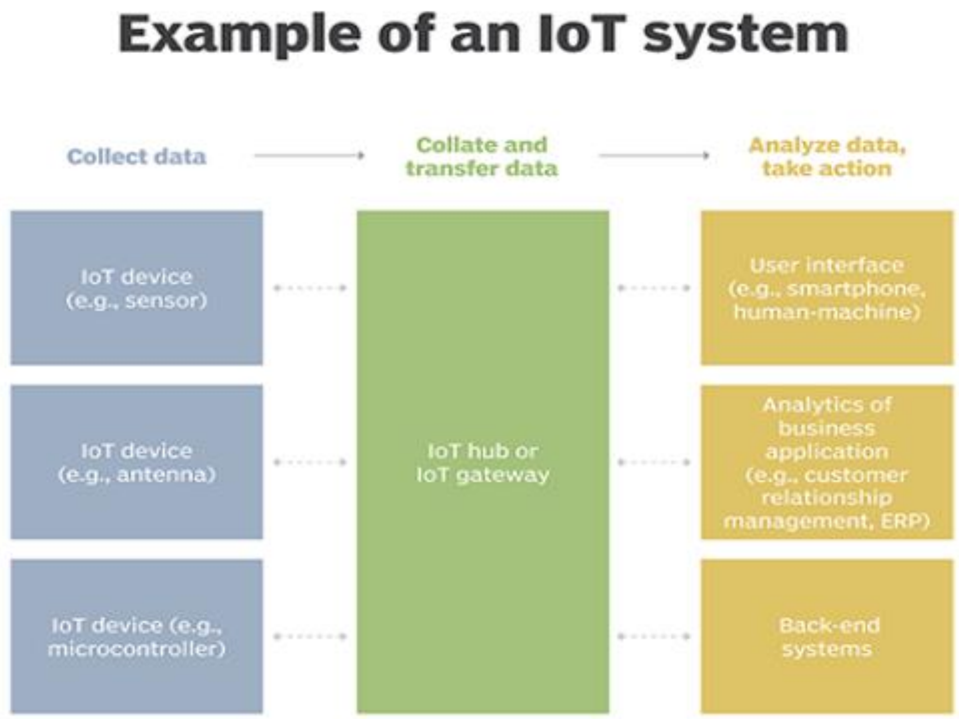

Example of an IoT system
Collect data
Collate and transfer data
Analyze data, take action
IoT device (e.g., sensor)
IoT device (e.g., antenna)
IoT device (e.g., microcontroller)
IoT hub or IoT gateway
User interface (e.g., smartphone, human-machine)
Analytics of business application (e.g., customer relationship management, ERP)
Back-end systems

Figure 5.8 Example of an IoT system

5.5.1 Benefits of IoT:

The internet of things offers a number of benefits to organizations, enabling them to:

- monitor their overall business processes
- improve the customer experience
- save time and money
- enhance employee productivity
- integrate and adapt business models
- make better business decisions and
- generate more revenue.

IoT encourages companies to rethink the ways they approach their businesses, industries and markets and gives them the tools to improve their business strategies.

Consumer and enterprise IoT applications

There are numerous real-world applications of the internet of things, ranging from consumer IoT and enterprise IoT to manufacturing and industrial IoT (IoT). IoT applications span numerous verticals, including automotive, telecommunication and more.

In the consumer segment, for example, smart homes that are equipped with smart thermostats, smart appliances and connected heating, lighting and electronic devices can be controlled remotely via computers, smartphones or other mobile devices.

Wearable devices with sensors and software can collect and analyze user data, sending messages to other technologies about the users with the aim of making users' lives easier and more comfortable. Wearable devices are also used for public safety -- for example, improving first responder’s response times during emergencies by providing optimized routes to a location or by tracking construction workers' or firefighters' vital signs at life-threatening sites.

5.6 HEARTBEAT SENSOR

Heartbeat Sensor is an electronic device that is used to measure the heart rate i.e. speed of the heartbeat. Monitoring body temperature, heart beat and blood pressure are the basic things that we do in order to keep us healthy.

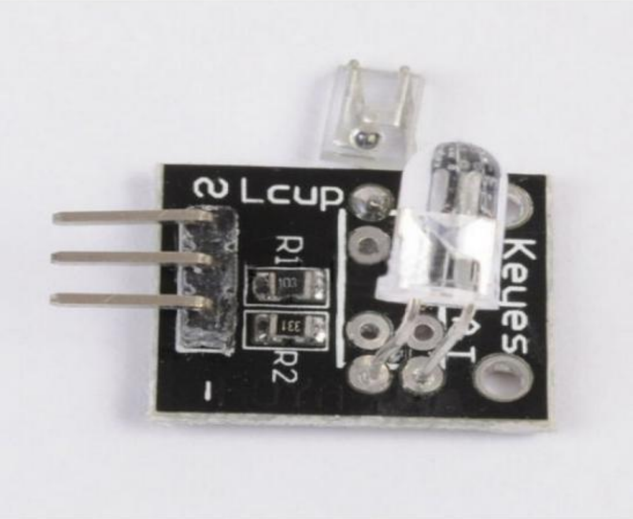


Figure 5.10 Heartbeat sensor

In order to measure the body temperature, we use thermometers and a sphygmomanometer to monitor the Arterial Pressure or Blood Pressure.

Heart rate can be monitored in two ways:

- One way is to manually check the pulse either at wrists or neck and
- Other way is to use a Heartbeat Sensor.

Principle of Heartbeat Sensor

The heartbeat sensor is based on the principle of photoplethysmography. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (avascular region). in the case of applications where the heart pulse rate is to be monitored, the timing of the pulses is more important. The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by the blood, the signal pulses are equivalent to the heartbeat pulses.

There are two types of photoplethysmography:

- Transmission: Light emitted from the light-emitting device is transmitted through any vascular region of the body like earlobe and received by the detector.

- Reflection: Light emitted from the light-emitting device is reflected by the regions.

Working of a Heartbeat Sensor

The basic heartbeat sensor consists of a light-emitting diode and a detector like a light detecting resistor or a photodiode. The heartbeat pulses cause a variation in the flow of blood to different regions of the body.

When tissue is illuminated with the light source, i.e. light emitted by the led, it either reflects (a finger tissue) or transmits the light (earlobe). some of the light is absorbed depends on the blood volume in that tissue. The detector output is in the form of the electrical signal and is proportional to the heartbeat rate.

This signal is a DC signal relating to the tissues and the blood volume and the AC component synchronous with the heartbeat and caused by pulsatile changes in arterial blood volume is superimposed on the DC signal.

Thus the major requirement is to isolate that AC component as it is of prime importance.

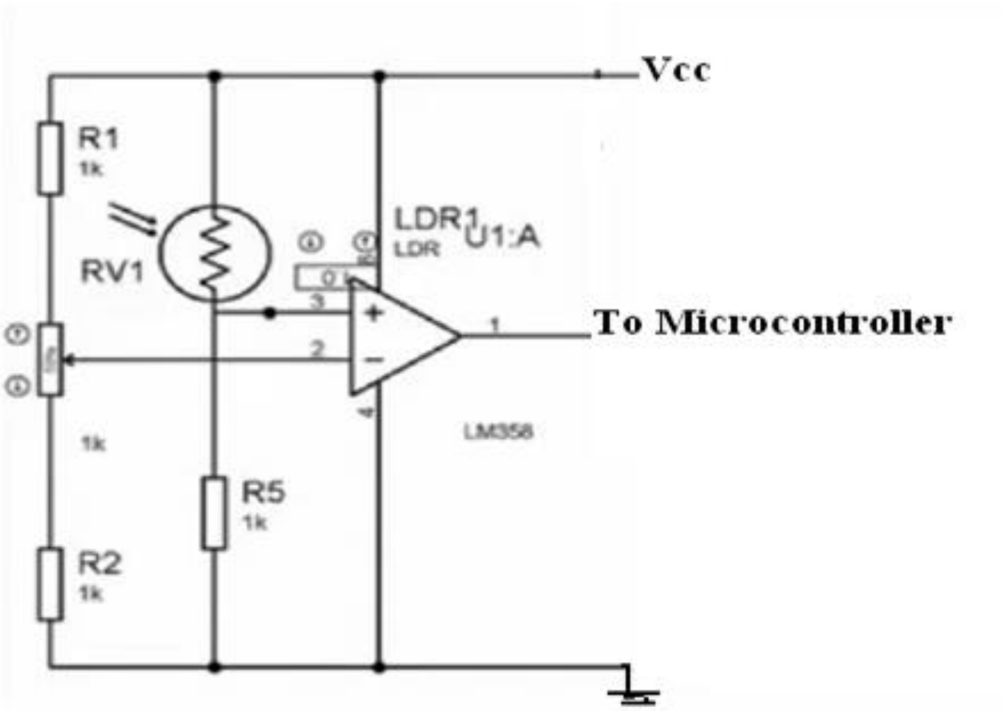


Figure 5.11 Circuit diagram of Heartbeat sensor

To achieve the task of getting the AC signal, the output from the detector is first filtered using a 2 stage HP-LP circuit and is then converted to digital pulses using a comparator circuit or using simple ADC.
The digital pulses are given to a microcontroller for calculating the

Heartbeat rate, given by the formula

BPM(Beats Per Minute) = 60*f
Where f is the pulse frequency

5.6.1 Specifications:

Voltage : 5V
Diameter : 16mm
LED Wavelength : 609nm
Overall Thickness : 0.125 inch (-3mm)
Working Current : -4mA

5.7 BUZZER

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Figure 5.12 Buzzer

5.7.1 TYPES

Electromechanical

Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

Mechanical

A joy buzzer is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.

Piezoelectric

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep. A piezoelectric buzzer/beeper also depends on acoustic cavity resonance or Helmholtz resonance to produce an audible beep.

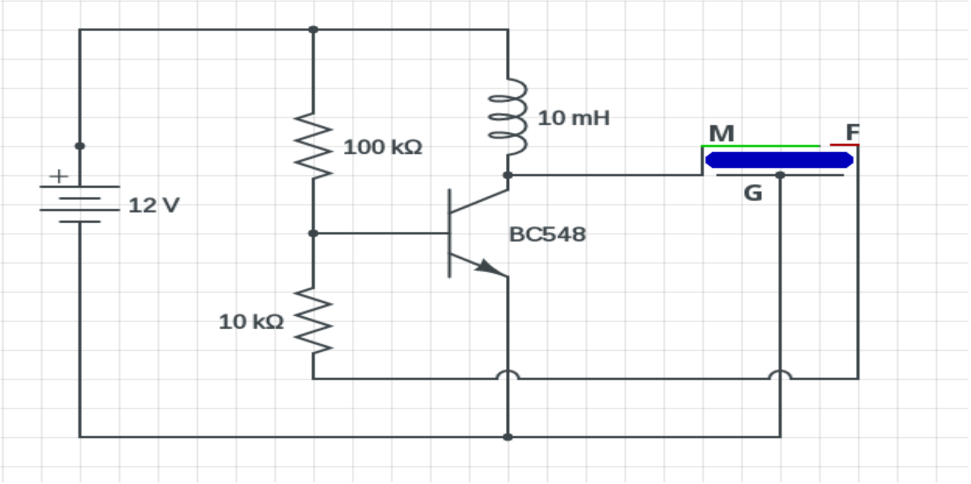


Figure 5.13 Circuit diagram of Buzzer

5.7.2 APPLICATIONS

While technological advancements have caused buzzers to be impractical and undesirable, there are still instances in which buzzers and similar circuits may be used.

Present day applications include:

- Novelty uses
- Judging panels
- Educational purposes
- Annunciator panels
- Electronic metronomes
- Game show lock-out device
- Microwave ovens and other household appliances
- Sporting events such as basketball games
- Electrical alarms
- Joy buzzer (mechanical buzzer used for pranks)

5.8 ULTRASONIC SENSORS

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object’s proximity. High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

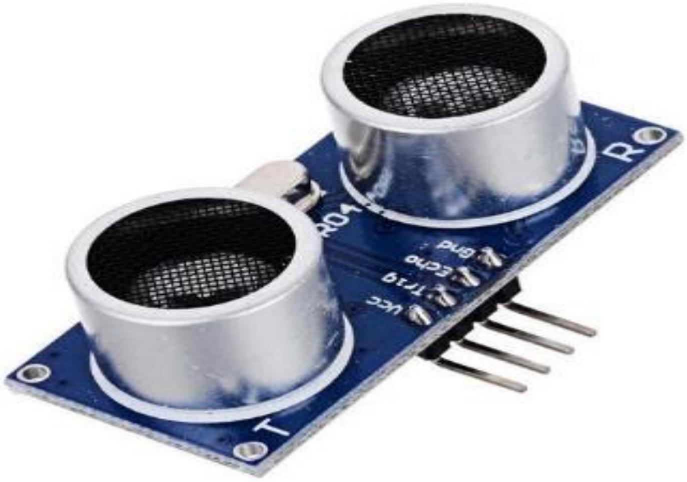


Figure 5.14 Ultrasonic sensor

Ultrasonic sound vibrates at a frequency above the range of human hearing. Transducers are the microphones used to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a tsonic pulse.

Ultrasound is reliable in any lighting environment and can be used inside or outside. Ultrasonic sensors can handle collision avoidance for a robot, and being moved often, as long as it isn't too fast.

Ultrasonics are so widely used, they can be reliably implemented in grain bin sensing applications, water level sensing, drone applications and sensing cars at your local drive-thru restaurant or bank.

Ultrasonic rangefinders are commonly used as devices to detect a collision.

Ultrasonic Sensors are best used in the non-contact detection of:

- Presence
- Level

5.8.1 Operation of ultrasonic sensors:

When an electrical pulse of high voltage is applied to the ultrasonic transducer it vibrates across a specific spectrum of frequencies and generates a burst of sound waves.

Whenever any obstacle comes ahead of the ultrasonic sensor the sound waves will reflect back in the form of echo and generates an electric pulse. It calculates the time taken between sending sound waves and receiving echo. The echo patterns will be compared with the patterns of sound waves to determine detected signal's condition.

5.8.2 Applications involving Ultrasonic detection:

The distance of obstacle or discontinuities in metals is related to velocity of sound waves in a medium through which waves are passed and the time taken for echo reception. Hence the ultrasonic detection can be used for finding the distances between particles, for detecting the discontinuities in metals and for indicating the liquid level.

5.8.3 Ultrasonic Distance Measurement

Ultrasonic sensors are used for distance measuring applications. These gadgets regularly transmit a short burst of ultrasonic sound to a target, which reflects the sound back to the sensor. The system then measures the time for the echo to return to the sensor and computes the distance to the target using the speed of sound within the medium.

Different sorts of transducers are utilized within industrially accessible ultrasonic cleaning devices. An ultrasonic transducer is affixed to a stainless steel pan which is filled with a solvent and a square wave is applied to it, conferring vibration energy on the liquid.

The ultrasonic distance sensors measures distance using sonar; an ultrasonic (well above human hearing) beat is transmitted from the unit and distance-to-target is determined by measuring the time required for the echo return. Output from the ultrasonic sensor is a variable-width beat that compares to the distance to the target.

5.8.4 8 Features of Ultrasonic Distance Sensor:

1. Supply voltage: 5V (DC)
2. Supply current: 15mA
3. Modulation frequency: 40Hz
4. Output: 0 - 5V (Output high when obstacle detected in range)
5. Beam Angle: Max 15 degree
6. Distance: 2cm - 400cm
7. Accuracy: 0.3cm
8. Communication: Positive TTL pulse.

5.9 WATER LEVEL SENSOR

Level sensors are used to detect the level of substances that can flow. Such substances include liquids, slurries, granular material and powders. Level measurements can be done inside containers or it can be the level of a river or lake.

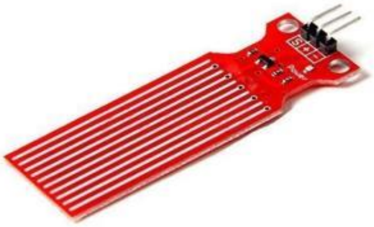


Figure 5.15 Water level sensor

5.9.1 Specifications:

Operating voltage : DC3-5V

Operating current : less than 20mA

Sensor Type : Analog

Detection Area : 40mm x 16mm

Operating temperature : 10°C to -30°C

Humidity : 10% to -90% non-condensing

5.10 TOUCH SENSOR

Touch Sensors are the electronic sensors that can detect touch. They operate as a switch when touched. These sensors are used in lamps, touch screens of the mobile, etc… Touch sensors offer an intuitive user interface.

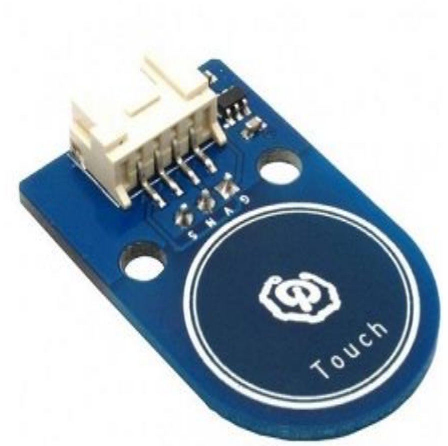


Figure 5.16 Touch sensor

Touch sensors are also known as Tactile sensors. These are simple to design, low cost and are produced in large scale. With the advance in technology, these sensors are rapidly replacing the mechanical switches. Based on their functions there are two types of touch sensors Capacitive sensor and Resistive sensor.

Capacitive sensors work by measuring capacitance and are seen in portable devices. These are durable, robust and attractive with low cost. Resistive sensors don't depend on any electrical properties for operation. These sensors work by measuring the pressure applied to their surface.

5.10.1 Working Principle of Touch Sensor:

Touch sensors work similar to a switch. When they are subjected to touch, pressure or force they get activated and acts as a closed switch. When the pressure or contact is removed they act as an open switch.

Capacitive touch sensor contains two parallel conductors with an insulator between them. These conductors plates act as a capacitor with a capacitance value C0.

When these conductor plates come in contact with our fingers, our finger acts as a conductive object. Due to this, there will be an uncertain increase in the capacitance.

A capacitance measuring circuit continuously measures the capacitance C0 of the sensor. When this circuit detects a change in capacitance it generates a signal.

The resistive touch sensors calculate the pressure applied on the surface to sense the touch. These sensors contain two conductive films coated with indium tin oxide, which is a good conductor of electricity, separated by a very small distance.

Across the surface of the films, a constant voltage is applied. When pressure is applied to the top film, it touches the bottom film. This generates a voltage drop which is detected by a controller circuit and signal is generated thereby detecting the touch.

5.10.2 Applications:

Capacitor sensors are easily available and are of very low cost. These sensors are highly used in mobile phones, iPods, automotive, small home appliances, etc… These are also used for measuring pressure, distance, etc… A drawback of these sensors is that they can give a false alarm.

Resistive touch sensors only work when sufficient pressure is applied. Hence, these sensors are not useful for detecting small contact or pressure. These are used in applications such as musical instruments, keypads, touch-pads, etc.. where a large amount of pressure is applied.

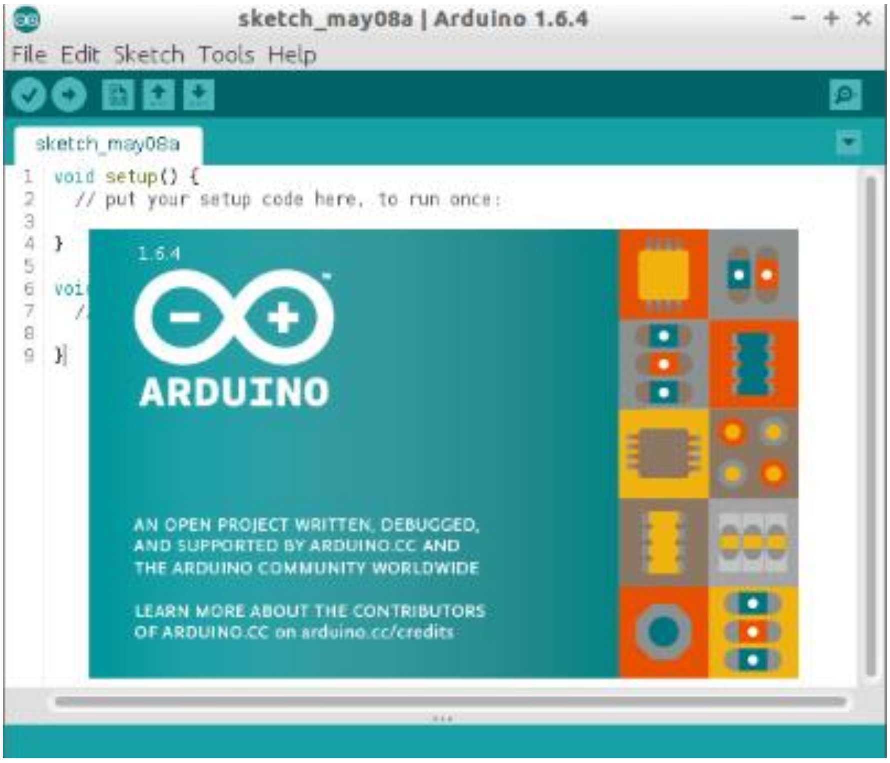
Some of the examples of touch sensors available in the market are TTP22301, TTP229, etc…

CHAPTER 6

SOFTWARE REQUIREMENT

6.1 ARDUINO IDE

The Arduino IDE is incredibly minimalistic, yet it provides a nearcomplete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.



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

}

voi
  /

}
```

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

Figure 6.1 Arduino IDE software

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an “Arduino language.”

However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device. While more advanced projects will take advantage of the built-in tools in the IDE, most projects will rely on the six buttons found below the menu bar.

1. The check mark is used to verify your code. Click this once you have written your code.
2. The arrow uploads your code to the Arduino to run.
3. The dotted paper will create a new file.
4. The upward arrow is used to open an existing Arduino project.
5. The downward arrow is used to save the current file.
6. The far right button is a serial monitor, which is useful for sending data from the Arduino to the PC for debugging purposes.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution.The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an “Arduino language.” However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution.

The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

6.2 EMBEDDED C

Embedded C Programming is the soul of the processor functioning inside each and every embedded system we come across in our daily life, such as mobile phone, washing machine, and digital camera. Each processor is associated with an embedded software. The first and foremost thing is the embedded software that decides functioning of the embedded system. Embedded C language is most frequently used to program the microcontroller.

Earlier, many embedded applications were developed using assembly level programming. However, they did not provide portability. This disadvantage was overcome by the advent of various high-level languages like C, Pascal, and COBOL. However, it was the C language that got extensive acceptance for embedded systems, and it continues to do so. The C code written is more reliable, scalable, and portable; and in fact, much easier to understand.

C language was developed by Dennis Ritchie in 1969. It is a collection of one or more functions, and every function is a collection of statements performing a specific task.

C language is a middle-level language as it supports high-level applications and low-level applications. Before going into the details of embedded C programming, we should know about RAM memory organization.

6.2.1 Salient features of the language

• C language is a software designed with different keywords, data types, variables, constants, etc.
• Embedded C is a generic term given to a programming language written in C, which is associated with a particular hardware architecture.
• Embedded C is an extension to the C language with some additional header files. These header files may change from controller to controller.
• The microcontroller 8051 #include is used. The embedded system

designers must know about the hardware architecture to write programs. These programs play prominent role in monitoring and controlling external devices. They also directly operate and use the internal architecture of the microcontroller, such as interrupt handling, timers, serial communication and other available features.

6.2.2 The basic additional features of the embedded software

Data types:

The data type refers to an extensive system for declaring variables of different types like integer, character, float, etc. The embedded C software uses four data types that are used to store data in the memory. The ‘char’ is used to store any single character; ‘int’ is used to store integer value, and ‘float’ is used to store any precision floating point value.

The size and range of different data types on a 32-bit machine is given in the following table. The size and range may vary on machines with different word sizes.

Keywords:

There are certain words that are reserved for doing specific tasks. These words are known as keywords. They are standard and predefined in the Embedded C.

Keywords are always written in lowercase. These keywords must be defined before writing the main program. The basic keywords of an embedded software are given below:

NAME	FUNCTIONS
sbit	Accessing of single bit
bit	Accessing of bit addressable memory of RAM
sfr	Accessing of sfr register by another name

Figure 6.2 Tables of keywords

sbit: This data type is used in case of accessing a single bit of SFR register.

- Syntax: sbit variable name = SFR bit
- Ex: sbit a=P2^1
- Explanation: If we assign p2.1 as ‘a’ variable, then we can use ‘a’ instead of p2.1 anywhere in the program, which reduces the complexity of the program.

Bit: This data type is used for accessing the bit addressable memory of RAM (20h-2fh).

- Syntax: bit variable name
- Ex: bit c
- Explanation: It is a bit sequence setting in a small data area that is used by a program to remember something.

SFR: This data type is used for accessing a SFR register by another name. All the SFR registers must be declared with capital letters.

- Syntax: SFR variable name = SFR address of SFR register
- Ex: SFR port0=0x80
- Explanation: If we assign 0x80 as ‘port0’, then we can use 0x80 instead of port0 anywhere in the program, which reduces the complexity of the program.

SFR Register: The SFR stands for ‘Special Function Register’. Microcontroller 8051 has 256 bytes of RAM memory. This RAM is divided into two parts: the first part of 128 bytes is used for data storage, and the other of 128 bytes is used for SFR registers.

All peripheral devices like I/O ports, timers and counters are stored in the SFR register, and each element has a unique address.

6.2.3 The Structure of an Embedded C Program

- comments
- preprocessor directives
- global variables
- main() function

{

- local variables
- statements
- …………..
- …………..

}

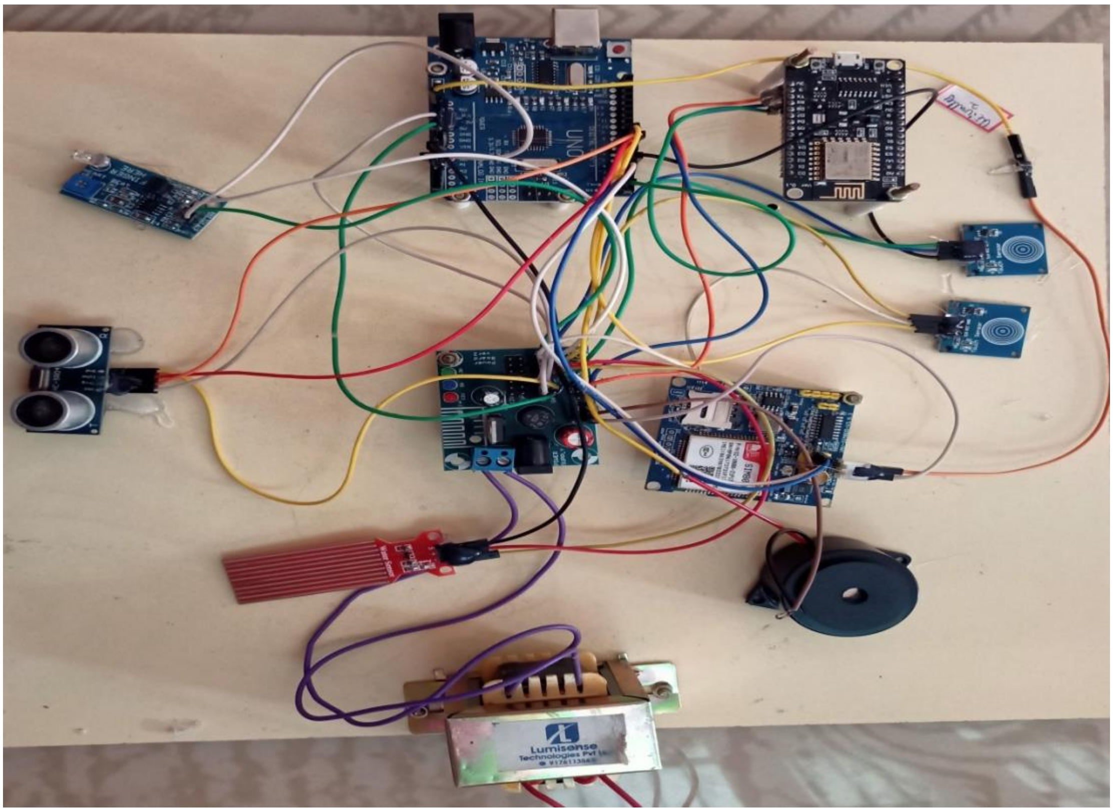
- fun(1)

{

- local variables
- statements
- …………..
- …………..

}

CHAPTER 7
HARDWARE KIT



The Smart cane is used by blind individuals to detect obstacles, and it includes sensors for touch, heart rate, buzzer, GSM Module, and ultrasonic sensor. We employ an ultrasonic sensor to look for objects and impediments; if any are found, the buzzer will immediately turn on.

Heartbeat readings are discovered using a heartbeat sensor. When a blind person is in a panic, we employ a touch sensor to activate the panic switch, which alerts the blind person’s helper via the GSM module. SMS is sent and received using a GSM module.

If there is a drainage pit and there are blind persons going down the road, a water level sensor is utilized to detect liquid barriers. If any liquid obstacles are present, the buzzer will immediately turn on. Information about the cane is sent by IoT (ESP8266) via a web server, and we are also able to monitor.

CHAPTER 8
CONCLUSION

In this project, our major work , where we are using existing sensors add additional sensors in the device that causes reduction of cost and decrease the weight and size of the device to make it more smart and economical. The aforementioned sensors are an accelerometer, GSM.

These features make this device unique compare to other existing smart cane. The smart cane is used for Blind people to detect obstacle in that smart cane we use ultrasonic sensor, ESP8266 (IoT), Touch sensor, buzzer, GSM. IoT for information about the cane through the web server. Using GSM we can sent panic message through phone Solar panel battery for power consumption.

The goal of the smart cane is to bring the white cane up to technological modernity while maintaining its affordable price. The Smart Cane is geared towards an elderly, less affluent demographic group that would demand comfort, accessibility, and affordability from the product.

Using the ultrasonic sensor, Arduino board, and vibration motor, the Smart Cane greatly increases the object detection range of the white cane, thereby improving the lives of the visually impaired users. Also the GSM-GPS technology used in the system provides security during emergency situations. Overall, the use of technology provides much more benefits than the white cane, and has taken a great leap towards improving the lives of the visually impaired.

REFERENCES

[1] M. Adil Khan; Kashif Nisar; Sana Nisar; BS Chowdhry; Ethisham Lodhi; Jawad Khan; Muhammad Reazul Haque "An Android-based Portable Smart Cane for Visually Impaired People".

[2] Do Ngoc HungVo Minh-ThanhQuoc Luong Huy Nguyen Triet "Design and Implementation of Smart Cane for Visually Impaired People".

[3] Pratibha Pawaskar "Smart Cane for Blind Person Assisted with Android Application and Save Our Souls Transmission".

[4] Nisha Sharma, Somen Acharya, Kaushal Kumar, Narendra Singh, and O.P. Chaurasia "Design and Implementation of a Solar Powered navigation Technology for the Visually Impaired".

[5] Sristi Agarwal, Sakshi Vaval, Varsha Chawla, Ms Nupur Agarwal, Ms Kavita Namdev "Smart Blind helping stick using IoT and Android".

[6] Nisha Sharma, Somen Acharya, Kaushal Kumar, Narendra Singh, and O.P. Chaurasia "Smart Walking System based on Artificial Intelligence".

[7] Salam Dhou Ahmad Alnabulsi, A.R. Al-Ali Mariam Arshi, Fatima Darwish, Sara Almaazmi, Reem Alameeri "An IoT Machine Learning-based Mobile Sensors Unit for Visually Imapired People".

[8] Klaus Finkenzeller. RFID Handbook: Fundamentals and Application in Contactless Smart Cards and Identification, Second Edition. John Wiley & Sons, Inc., 2003.

[9] Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach, Second Edition. Pearson Hall, pp. 97-110, 2003.

[10] Iwan Ulrich, Johann Borenstein. "The Guide Cane-Applying Mobile Robot Technologies to Assist Visually Impaired.", IEEE Transactions on Systems, Man, and Cybernetics, —Part A: Systems and Humans, Vol. 31, No. 2, pp. 131-136, March 2001.

[11] Vladimir Kulyukin, Chaitanya Gharpure, Nathan De Graw, John Nicholson, Sachin Pavithran. “A Robotic Guide for the Visually Impaired in Indoor Environments.” Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI04), San Jose, CA, 2004.

[12] Tomohiro Amemiya, Jun Yamashita, Koichi Hirota, Michitaka Hirose, "Development of Virtual Leading Blocks for the Blind", In Proc. of VRSJ 8th Annual Conf., pp. 359-362, VRSJ 8th Annual Conf., Gifu, 2003.9.

[13] Singh Vaibhav, Paul Rohan, “Smart cane for the visually impaired: Design and controlled field testing of an affordable obstacle detection system”, IIT Delhi.

APPENDIX

I.Code for the prototype

```
#define triggerpin 7
#define echopin 6
#define touch1 4
#define touch2 8
#define buzzer 5
#define waterlevel A0
#define SECONDS 15000
unsigned int long Time = 0;
int flag = 0;
int count;
int value;
#include <SoftwareSerial.h>
SoftwareSerial gsm(2,3);
String a;
void setup()
{
    Serial.begin(9600);
    gsm.begin(9600);
    pinMode(triggerpin,OUTPUT);
    pinMode(echopin,INPUT);
    pinMode(buzzer,OUTPUT);
    pinMode(touch1,INPUT);
    pinMode(touch2,INPUT);
    pinMode(waterlevel,INPUT);
}

void loop()
{
    long distance;
    long duration;
    digitalWrite(triggerpin,HIGH);
    delay(10);
    digitalWrite(triggerpin,LOW);
    duration=pulseIn(echopin,HIGH);
```

```
distance=duration*0.0343/2;
Serial.print("distance in cm");
Serial.println(distance);
int x=digitalRead(touch1);
Serial.println(x);
if(x==1 && distance<=30)
{
    digitalWrite(buzzer,HIGH);
    Serial.print("danger");
    delay(5000);
}
else
{
    digitalWrite(buzzer,LOW);
    Serial.print("danger is avorted");
}
int z=analogRead(waterlevel);
Serial.println(z);
if(z>400)
{
    digitalWrite(buzzer,HIGH);
    Serial.println("danger");
}
else
{
    digitalWrite(buzzer,LOW);
    Serial.println("danger is avorted");
    delay(1000);
}
int y=digitalRead(touch2);
Serial.println(y);
if(y==1
{
    Time = millis();
    count = 0;
```

```
      while (Time + SECONDS > millis()) //it will only read after 1
      second delay
      {
              value = analogRead(A1);

              if (value > 300 && flag == 0)
              {
                      count++;
                      Serial.print(".");
                      flag = 1;
              }
              else if (value < 300 && flag == 1){
              flag = 0;
              }
      }
      count = count * 4;
      Serial.print("heart rate= ");
      Serial.println(count);
      delay(1000);
      if(count<44)
      {
          gsm.println("AT+CMGF=1");    //Sets the GSM Module in Text
      Mode
          delay(1000);  // Delay of 1000 milli seconds or 1 second
          gsm.println("AT+CMGS=\"+919025817262\"\r"); // Replace x
  with mobile number
          delay(1000);
          gsm.println("I Am in DANGER MY HEART BEAT IS  ");
          // The SMS text you want to send
          delay(1000);
          gsm.println((char)26);// ASCII code of CTRL+Z
          delay(1000);
          Serial.print("I am going to die");
      }
  }
}
```