Chapter-1

INTRODUCTION TO PROJECT

INTRODUCTION:

Lot of people suffering from serious visual illnesses, that prevents them from travelling. Accordingly, they have to use a large vary of tools and techniques to need them in their travel. One in all these techniques is orientation and quality specialist who helps the visually impaired and blind people and trains them to move and on their own independently and safely counting on their different remaining senses.

Recently, several techniques are developed to boost the quality of blind people who accept signal processing and sensor technology. These known as "Electronic Travel Aid (ETA)". Devices help the blind to move freely in an atmosphere regardless of its systems inform the blind of the presence of an object at a selected distance ahead or near to him/her. These details allow the user alter his/her approach. Info about the object characteristics in produce extra knowledge to boost space manifestation and memory of the blind to overcome the previous limitations this work offers an easy efficient.configurable electronic system for the blind and visually impaired persons to succour them in their mobility regardless of whenever they are outdoor or indoor the originality of a proposed system is that it utilizes an embedded version system of 3 straight forward IR sensors and brings a long all reflective signals in order to codify an obstacle through PIC microcontroller.

Chapter-2

MICROCONTROLLERS

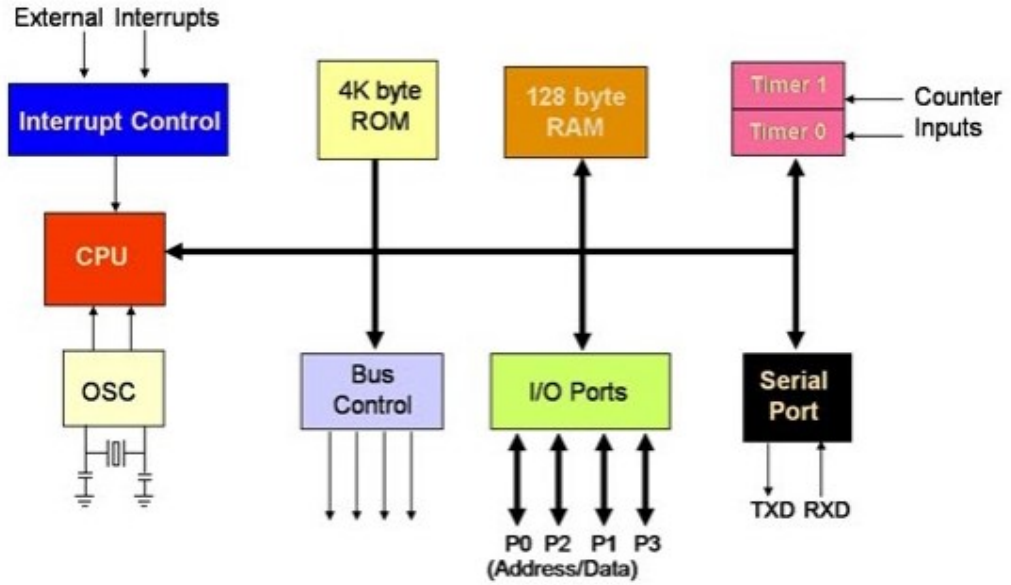
A device that controls the transfer of data from a computer to a peripheral device and vice versa. For example, disk drives, display screens, keyboards, and printers all require controllers.

Introduction to Microcontrollers

A computer-on-chip is also a variation of a small chip which mixes the processor core (CPU), some memory, and I/O (input/output) lines, all on one chip. The laptop-on-a- chip is named the PC who's correct that means may be a computer employing a number of microprocessors as its CPUs, whereas the idea of the PC is thought to be a microcontroller. A microcontroller is often viewed as a collection of digital logic circuits integrated on one microchip. This chip is employed for less than specific applications.

Most microcontrollers don't need a considerable quantity of time to be told the thanks to efficiently program , although many, though several of, that have quirks, that may got to perceive before conceive to develop 1st application.

Along with microcontrollers obtaining quicker, smaller and a lot of power economical there are additionally obtaining a lot of options. Often, the primary version of microcontroller can simply have memory and digital I/O, however because the device family matures, a lot of and a lot of pat numbers with variable options are going to be out there.



Chapter-3

SOFTWARE REQUIREMENTS

Software Requirements

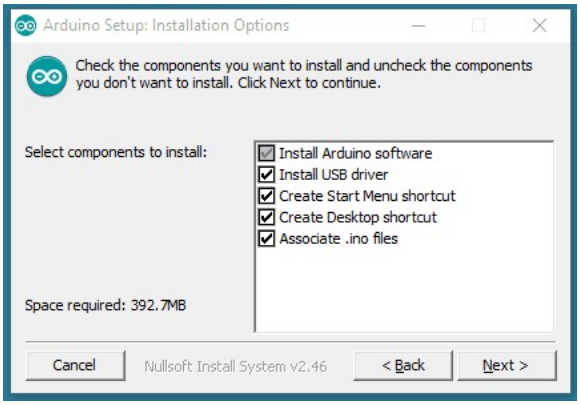
- Arduino Software (IDE)
- Tinkercad https://www.tinkercad.com/

Arduino IDE (2.0.0-rc7):

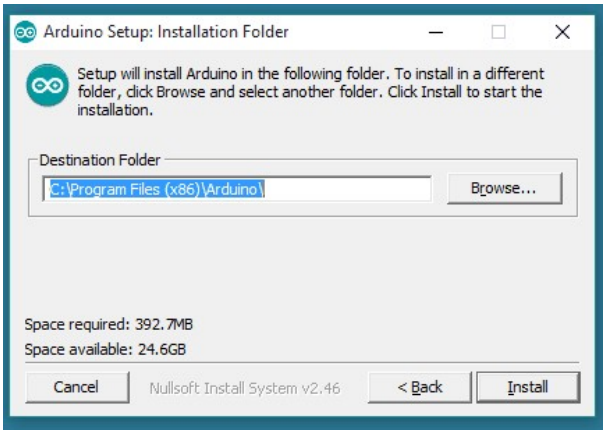
Download the Arduino Software (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

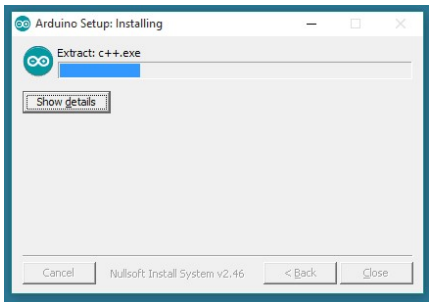
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.



Choose the components to install.



Choose the installation directory.



Installation in progress.

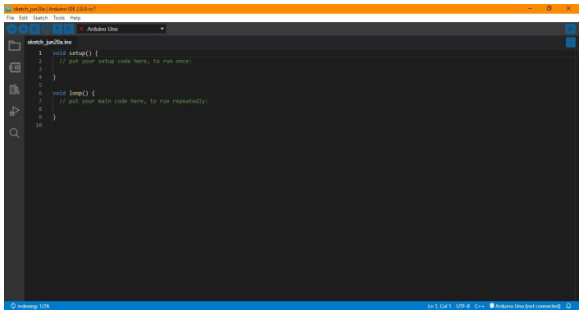
The process will extract and install all the required files to execute properly the Arduino Software (IDE)

The text of the Arduino getting started guide is licensed under a Creative Commons Attribution-ShareAlike 3.0 License. Code samples in the guide are released into the public domain.

Arduino language is not actually C/C++ Language . It uses wiring language frame work which is simplified version of C/C++ Language. When you type your Arduino code it is not real C/C++ code , for converting into C/C++ language program a number of changes have to happen for your Arduino code.

Arduino Family of boards use AVR series of micro controllers developed by the ATMEL Corporation. Arduino IDE uses avr-gcc Compiler and related tools to build the binary executable code for an AVR device .when you click on the option of verify and upload your code in the Arduino IDE passes through different stages.

The first step is Pre-Processing. After pre-processing your code converted into C/C++ code and then avr - gcc compiler compile this code into the binary executable code for Arduino board.



After installation, this is how the IDE looks like. Here we can write, verify, upload the code according to our mode of operation.

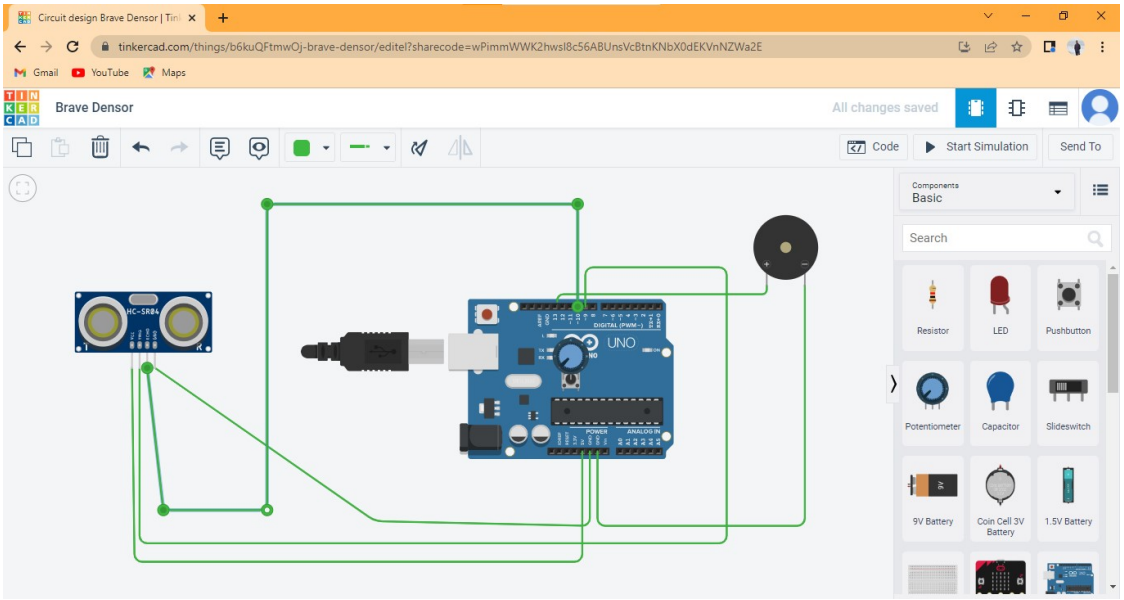
Tinkercad

Tinkercad is a free-of-charge, online 3D modeling program that runs in a web browser. Since it became available in 2011 it has become a popular platform for creating models for 3D printing as well as an entry-level introduction to constructive solid geometry in schools.

Tinkercad was founded by former Google engineer Kai Backman and his cofounder Mikko Mononen, with a goal to make 3D modeling, especially the design of physical items, accessible to the general public, and allow users to publish their designs under a Creative Commons license. In 2011, the tinkercad.com website was launched as a web-based 3D modelling tool for WebGL-enabled browsers, and in 2012 the company moved its headquarters to San Francisco.[5] By 2012 over 100,000 3D designs had been published by users.

In May 2013, Autodesk announced at a Maker Faire that they would acquire Tinkercad. In March 2017, Autodesk recommended users of the soon to be retired 123D Sculpt migrate to Tinkercad (or Maya LT). In May, Autodesk discontinued its 123D Circuits (Circuits.io) "Electronics Lab". The program's features were merged into Tinkercad.

Tinkercad uses a simplified constructive solid geometry method of constructing models. A design is made up of primitive shapes that are either "solid" or "hole". Combining solids and holes together, new shapes can be created, which in turn can be assigned the property of solid or hole. In addition to the standard library of primitive shapes, a user can create custom shape generators using a built-in JavaScript editor.



Chapter-4

HARDWARE REQUIREMENTS

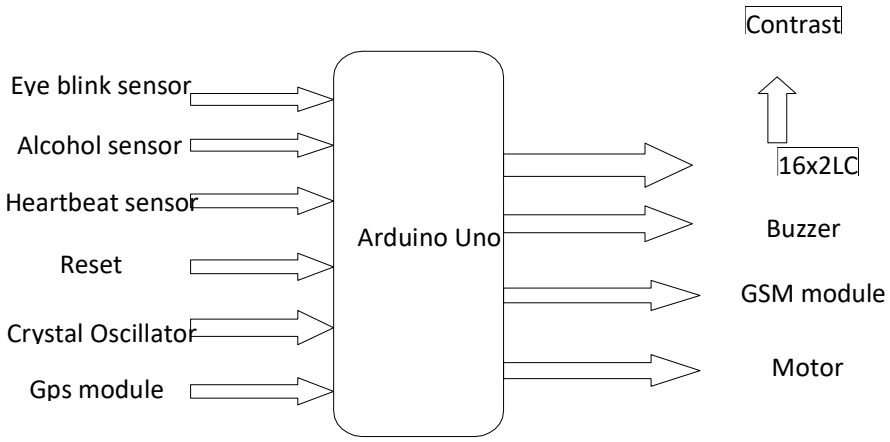
Hardware Requirements

- Arduino uno ATMEGA 328P microcontroller
- Ultrasonic Sensor
- Buzzer
- Power Supply

ARDUINO UNO

In this project the tendency to used ARDUINO UNO ATMEGA 328 microcontroller. For many applications, going to be able to realize a tool at intervals the family that meets this specifications with a minimum of external devices, or an external however which can create attaching external devices easier, each in terms of wiring and programming for several microcontrollers.

Programmers will design terribly cheaply or maybe in-built to the ultimate application circuit eliminating the necessity for a separate circuit.



Arduino Uno ATMEGA 328P:

Arduino is an open –source hardware and Software Company, project and user communitythat designs and manufactures single-board microcontrollers andmicrocontroller kits for building digital devices and interactive objects that can sense and control both physically and digitally.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus(USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C+ +. In addition to using traditional compiler tool chains, the arduino project provides an integrated development environment (IDE) based on the processing language project.

Arduino 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: simple connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Can tinker with UNO without worrying too much about doing something wrong, worst case scenario 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 USB Arduino 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.

ARDUINO FAMILY

1. Arduino UNO

2. Arduino NANO

3. Arduino MINI

4. Arduino MEGA (2560)

5. Arduino MEGA (1280)

6. Node MCU ESI12E

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. Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Arduino board has the following new features:

1. 1.0 pin out: 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.
2. Stronger RESET circuit.
3. 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.

Figure: ARDUINO UNO

Arduino UNO Specifications

1. Microcontroller ATmega328
2. Operating Voltage 5V
3. Input Voltage (recommended) 7-12V
4. Input Voltage (limits) 6-20V
5. Digital I/O Pins 14 (of which 6 provide PWM output)
6. Analog input Pins 6
7. DC Current per I/O Pin 40 m A
8. DC Current for 3.3V Pin 50 m A
9. Flash Memory 32 KB (ATmega328) of which 0.5 KB used by boot loader
10. SRAM 2 KB (ATmega328)
11. EEPROM 1 KB (ATmega328)
12. Clock Speed 16 MHz

Power:

The power pins are as follows:

1. VIN. 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.
2. 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.
3. 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
4. GND. Ground pins.

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).

Pin Description

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digitalWrite (), 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 k ohms. In addition, some pins have specialized functions:

1. 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.
2. 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.
3. PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write () function.
4. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
5. 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, labelled 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:

There are a couple of other pins on the board:

a) AREF. Reference voltage for the analog inputs. Used with analog Reference ().
b) 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.

Programming:

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino 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 Uno comes preburned with a boot loader 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).

You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

a. On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

b. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user-contributed tutorial for more information.

USB Jack:

Connecting a male USB A to male USB B cable is how you upload programs from your computer to your Arduino board. This also powers your Arduino.

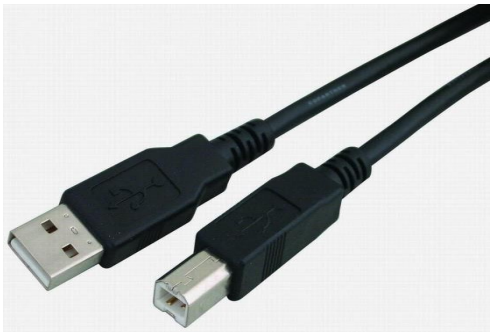


Figure : Male USB A to male USB B cable

Power Jack:

The power jack is where you connect a component to power up your Arduino (recommended voltage is 5V). There are several ways to power up your Arduino: rechargeable batteries, disposable batteries, wall-warts and solar panel, for example. For more information about this subject you can read this blog post on Random Nerd Tutorials

Arduino – http://randomnerdtutorials.com/arduino-5-ways-to-power-up-your-arduino/5 Ways to Power Up your Arduinohttp://randomnerdtutorials.com/arduino-5-ways-to-power-up-your-arduino/.

Figure : Examples of ways to power up

Software Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno 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 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half- second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET- EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Over current Protection:

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and over current. 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.

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.

UART (Universal Asynchronous Receiver/Transmitter) FEATURES:

a) 16 byte Receive and Transmit FIFOs

b) Register locations conform to ‘550 industry standard.

c) Receiver FIFO triggers points at 1, 4, 8, and 14 bytes.

d) Built-in fractional baud rate generator with auto bauding capabilities.

e) Mechanism that enables software and hardware flow control implementation

PIN DESCRIPTION

Pin	Type	Description
RXD0	Input	Serial Input. Serial receive data.
TXD0	Output	Serial Output. Serial transmit data.

Register Description

UART0 contains registers organized as shown in Table. The Divisor Latch Access Bit (DLAB) is contained in U0LCR [7] and enables access to the Divisor Latches.

RS232 CABLE

To allow compatibility among data communication equipment, an interfacing standard called RS232 is used. Since the standard was set long before the advent of the TTL logic family, its input and output voltage levels are not TTL compatible. For this reason, to connect

any RS232 to a microcontroller system, voltage converters such as MAX232 are used to convert the TTL logic levels to the RS232 voltage levels and vice versa.

MAX232 IC

Max232 IC is a specialized circuit which makes standard voltages as required by RS232 standards. This IC provides best noise rejection and very reliable against discharges and short circuits. MAX232 IC chips are commonly referred to as line drivers.To ensure data transfer between PC and microcontroller, the baud rate and

voltage levels of Microcontroller 0;l;land PC should be the same. The voltage levels of microcontroller are logic1 and logic 0 i.e., logic 1 is +5V and logic 0 is 0V. But for PC, RS232 voltage levels are considered and they are: logic 1 is taken as -3V to -25V and logic 0 as +3V to +25V. So, in order to equal these voltage levels, MAX232 IC is used. Thus this IC converts RS232 voltage levels to microcontroller voltage levels and vice versa.

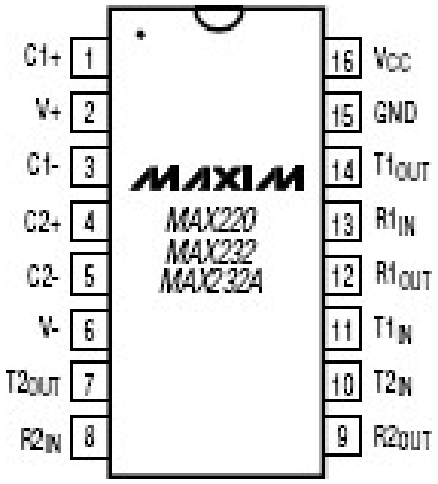
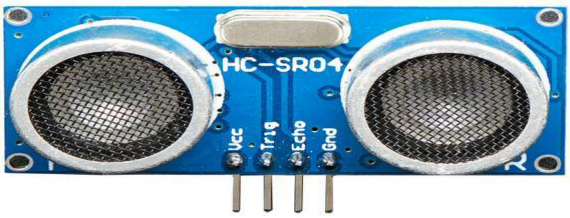


Figure:3.2 Pin Diagram of MAX232 IC

ULTRASOIC SENSOR

The ultrasonic sensor (or transducer) works on the same principles as a radar system. An ultrasonic sensor can convert electrical energy into acoustic waves and vice versa. The acoustic wave signal is an ultrasonic wave travelling at a frequency above 18 kHz. The famous HC SR04 ultrasonic sensor generates ultrasonic waves at 40 kHz frequency.

Typically, a microcontroller is used for communication with an ultrasonic sensor. To begin measuring the distance, the microcontroller sends a trigger signal to the ultrasonic sensor. The duty cycle of this trigger signal is 10µS for the HC-SR04 ultrasonic sensor. When triggered, the ultrasonic sensor generates eight acoustic (ultrasonic) wave bursts and initiates a time counter. As soon as the reflected (echo) signal is received, the timer stops. The output of the ultrasonic sensor is a high pulse with the same duration as the time difference between transmitted ultrasonic bursts and the received echo signal.



The microcontroller interprets the time signal into distance using the following functions:

$$Distance\ (cm) = \frac{echo\ pulse\ width\ (uS)}{58}$$

$$Distance\ (inch) = \frac{echo\ pulse\ width\ (uS)}{148}$$

Theoretically, the distance can be calculated using the TRD (time/rate/distance) measurement formula. Since the calculated distance is the distance traveled from the ultrasonic transducer to the object—and back to the transducer—it is a two-way trip. By dividing this distance by 2, you can determine the actual distance from the transducer to the object. Ultrasonic waves travel at the speed of sound (343 m/s at 20°C). The distance between the object and the sensor is half of the distance traveled by the sound wave.[iv] The following equation calculates the distance to an object placed in front of an ultrasonic sensor:

$$distance = \frac{time\ taken\ x\ speed\ of\ sound}{2}$$

Applications

1. Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
2. Used to measure the distance within a wide range of 2cm to 400cm
3. Can be used to map the objects surrounding the sensor by rotating it
4. Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

Fig: 2D model of the component

Multiple areas of engineering use ultrasonic sensors. “No-contact” distance measuring is very useful in automation, robotics, and instrumentation. Below, we investigate the applications of ultrasonic sensors:

Limitations of ultrasonic sensors

Ultrasonic sensors such as the HC-SR04 can efficiently measure distances up to 400 cm with a slight tolerance of 3 mm. [xiii] However, if a target object is positioned such that the ultrasonic signal is deflected away rather than reflected back to the ultrasonic sensor, the calculated distance can be incorrect. In some cases, the target object is so small that the reflected ultrasonic signal is insufficient for detection, and the distance cannot be measured correctly.

Furthermore, objects like fabric and carpet can absorb acoustic signals. If the signal is absorbed in the target object's end, it cannot reflect back to the sensor , and hence, the distance cannot be measured.

Figure : Ultrasonic sensor measuring distance from object during drone's flight. (Source: Radio Link

BUZZER

A buzzer or beeper is an audio signalling 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.

FIG14: BUZZER

History

Electromechanical

The electric buzzer was invented in 1831 by Joseph Henry. This are mainly used in early doorbells until were phased out in the early 1930s in favour of musical chimes, which had a softer tone.

Piezoelectric

Piezoelectric buzzers, or piezo buzzers, as sometimes called, were invented by Japanese manufactures and fitted into wide array of products during the 1970s to 1980s. This advancement mainly came about because of cooperative efforts by Japanese manufacturing companies. In 1951, established the Barium Titanate Application Research Committee, which allowed the companies to be "competitively cooperative" and bring about several piezoelectric innovations and inventions.

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 requires drivers. Other examples are doorbells.

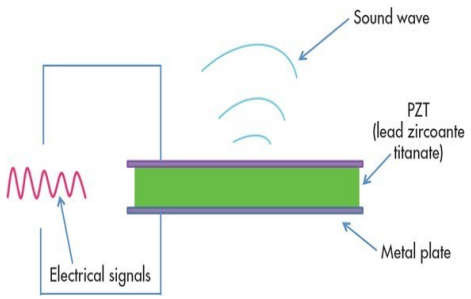


Fig14.2: Piezoelectric

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.

Fig14.2.1 Piezoelectric Beeper

Interior of a readymade loudspeaker, showing a piezoelectric-disk-beeper (with 3 electrodes…including 1 feedback electrode (the central, small, electrode joined with red wire in this photo), and an oscillator to self-drive the buzzer.)

Modern 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
- Enunciator 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 and used for pranks)

POWER SUPPLY

The Arduino Uno can be powered via the USB connection or with an external power supply/External Battery. 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 centre-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 board can operate on an external supply / external Battery of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

CHAPTER-5

WORKING PROCEDURE

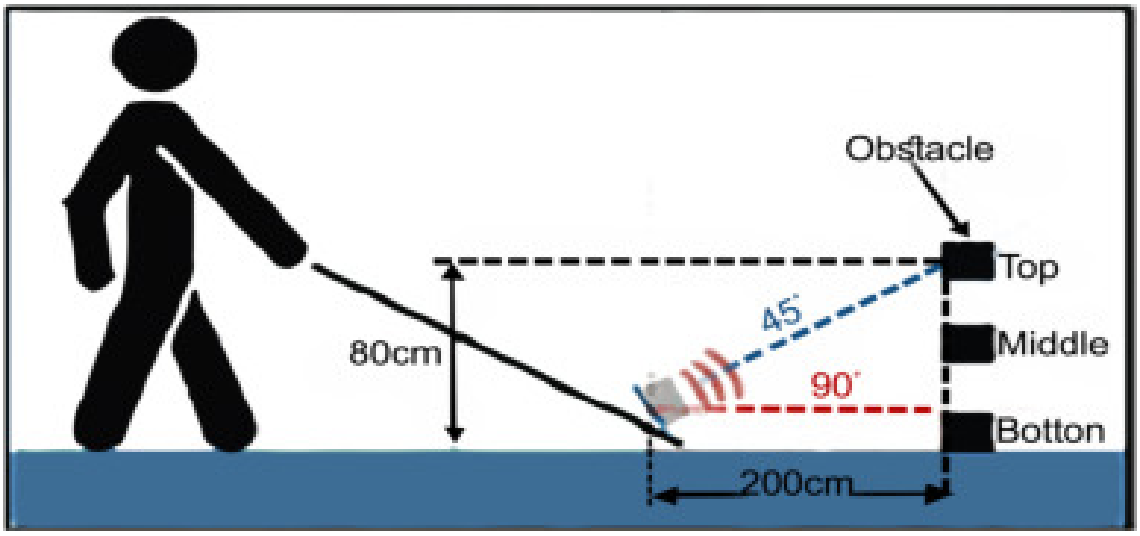
In this project the components mainly used are:

Working

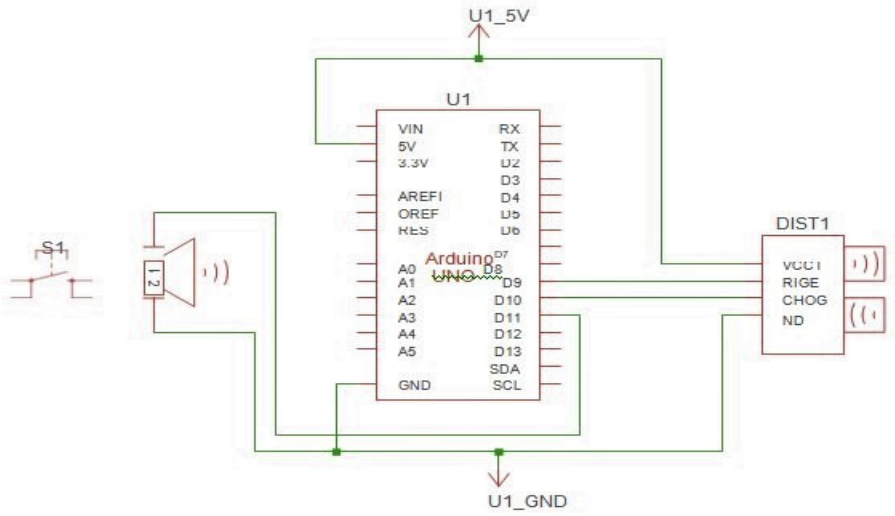
In this project the components mainly used are:

- Arduino UNO
- Ultrasonic sensor
- DC Buzzer
- Jumper Wires
- PVC Pipe
- 9V Battery
- Switch

The main part in the system is the microcontroller that controls the other components in the system. When the ultrasonic sensors detect any objects or obstacle in 180 degree path it will activate the buzzer and the vibration motor. In addition to that, when the GSM modem receive a message it will be sent to the microcontroller which will get the location of the stick from the GPS modem and transmit the location to the GSM modem in response to the sender. In the areas with low signals cameras can be use, this system works by fitting a camera on the persons head, it will use certain algorithm to identify the highs and obstacles in front the blind person. In case of an emergency, the user of the stick will press the emergency button and the signal from the button will go to the microcontroller which will get the location from the GPS modem and transmit the location to the GSM modem which will send a SMS messages to the all saved numbers in the system.



Schematic Diagram



The above diagram is the schematic representation of this project where,

- Take Arduino UNO ,Buzzer, Ultrasonic Sensor,9V Battery, PVC pipe, Switch.
- Take pipe and place ultrasonic sensor on it.
- Connect the Jumper wires on it.
- Then place the Buzzer on the pipe.
- Do connections from the circuit diagram.
- Place the Arduino UNO ,Battery, Switch place on the stick.
- Upload the code.
- After uploading the code , now check for the blind stick.

Chapter-6
RESULT

Artificial Eye For Blind People:

HC-SR04 is utilized to recognize the presence of an obstruction. At the point when the ultrasonic sensor recognizes the presence of a deterrent, the flag will send to vibrator to vibrate with various vibration qualities.

A signal of 40 kHz is sent by the ultrasonic sensor. In this project three ultrasonic sensor is used. Two sensor is used for obstacle detection and one sensor is used for pit detection. For obstacle detection, we set a threshold distance which is 50cm. If detected distance is less than the threshold distance then vibratory motor will vibrate. The vibration strength of motor is the function of detected distance. As any obstacle is coming towards the sensor then vibration strength of motor is increasing and vice-versa. One sensor is used for pit detection, for pit detection we set a threshold distance of 5cm. if distance is greater than 5cm then it gives the signal to the sensor and sense for an obstacle.

Artificial Eye For Blind People

Chapter-7
CONCLUSION

This device is used to extend the user's range of which used to extend the user's range of touch sensation. It is usually swung in a low sweeping motion, across the intended path of travel, to detect obstacles. However using walking stick may not solve the problem & and It is well known that visually impaired people use their hearing sense to compensate for their reduced eyesight. For instance, they can recognize sound sources. This project focuses on the Electronic Blind Mobility Aid helps Blind people travelling by themselves using ULTRASONIC sensors mounted on a walking stick.

This system offers a low-cost, reliable, portable, low power consumption and robust solution for navigation with obvious short response time. Though the system is hard-wired with sensors and other components, it's light in weight. Further aspects of this system can be improved via wireless connectivity between the system components, thus, increasing the range of the ultrasonic sensor and implementing a technology for determining the speed of approaching obstacles.

Reference:

Project Idea:

- https://create.arduino.cc/projecthub/mohammadsohail0008/smart-stick-for-blind-peoples-5ee884
- https://techatronic.com/smart-blind-stick-using-arduino-and-ultrasonic-sensor/
- https://www.tinkercad.com/

Arduino:

- https://en.wikipedia.org/wiki/Arduino_Uno
- https://docs.arduino.cc/hardware/uno-rev3
- http://docs.arduino.cc/software/ide-v1/tutorials/Windows
- https://www.seeedstudio.com/blog/2019/12/04/introduction-to-the-arduino-what-is-arduino/

Ultrasonic Sensor:

- https://components101.com/sensors/ultrasonic-sensor-working-pinout-datasheet

Buzzer:

- https://www.elprocus.com/buzzer-working-applications/

APPENDIX -1
SOURCE CODE

```
int trigger_pin=9;
int encho_pin=10;
int buzzer_pin=11;
int time;
int distance;

void setup()
{
  Serial.begin(9600);
  pinMode(trigger_pin,OUTPUT);
  pinMode(encho_pin,INPUT);
  pinMode(buzzer_pin,OUTPUT);
}

void loop()
{
  digitalWrite(trigger_pin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger_pin,LOW);
  time=pulseIn(encho_pin,HIGH);
  distance=(time*0.034)/2;
 if(distance<=300)
 {
   Serial.println("Object is near");
   Serial.print("Distance=");
   Serial.println(distance);



   digitalWrite(buzzer_pin,HIGH);
```

```
  delay(500);
 }
 else
 {
   Serial.println("object is far");
   Serial.print("Distance=");
   Serial.println(distance);
   digitalWrite(buzzer_pin,LOW);
   delay(500);
 }
}
```