DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY
KUDLU GATE
BANGALORE - 560068

MINI PROJECT REPORT

ON

"MORSE CODE DECODER"

SUBMITTED TO THE 3RD SEMESTER DIGITAL CIRCUITS AND LOGIC DESIGN LABORATORY- 2019

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

SUBHAM MITTAL - ENG18CS0285
SYED MUSADDIQ - ENG18CS0293
TRIDIB BHATTACHARJEE - ENG18CS0301
VAISHAALI KONDAPALLI - ENG18CS0305
VAISHNAVI A PUNAGIN - ENG18CS0307

Under the supervision of

PROF. SHIVAPRASAD A C

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068

CERTIFICATE

This is to certify that Mr./Ms. **SUBHAM MITTAL (ENG18CS0283), SYED MUSADDIQ (ENG18CS0293), TRIDIB BHATTACHARJEE (ENG18CS0301), VAISHAALI K (ENG18CS0305) and VAISHNAVI A PUNAGIN (ENG18CS0307)** *have satisfactorily completed their Mini Project as prescribed by the University for the* **3rd** *semester B.Tech. programme in Computer Science & Engineering during the year* **2019-2020** *at the School of Engineering, Dayananda Sagar University., Bangalore.*

Date: _____

_____

Signature of the faculty in-charge

Max Marks	Marks Obtained

_____

Signature of Chairman

Department of Computer Science & Engineering

DECLARATION

We hereby declare that the work presented in this mini project entitled- "**Morse Code Decoder**", has been carried out by us and it has not been submitted for the award of any degree, diploma or the mini project of any other college or university.

SUBHAM MITTAL - ENG18CS0285
SYED MUSADDIQ -ENG18CS0293
TRIDIB BHATTACHARJEE -ENG18CS0301
VAISHAALI KONDAPALLI - ENG18CS0305
VAISHNAVI A PUNAGIN - ENG18CS0307

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to our **Chairman, Dr. M K Banga**, for providing necessary departmental facilities, moral support and encouragement.

We are very much thankful to **Prof. Shivaprasad A C,** for providing help and suggestions in completion of this mini project successfully.

We have received a great deal of guidance and co-operation from our friends and we wish to thank all that have directly or indirectly helped us in the successful completion of this project work.

SUBHAM MITTAL - ENG18CS0285
SYED MUSADDIQ -ENG18CS0293
TRIDIB BHATTACHARJEE -ENG18CS0301
VAISHAALI KONDAPALLI - ENG18CS0305
VAISHNAVI A PUNAGIN - ENG18CS0307

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

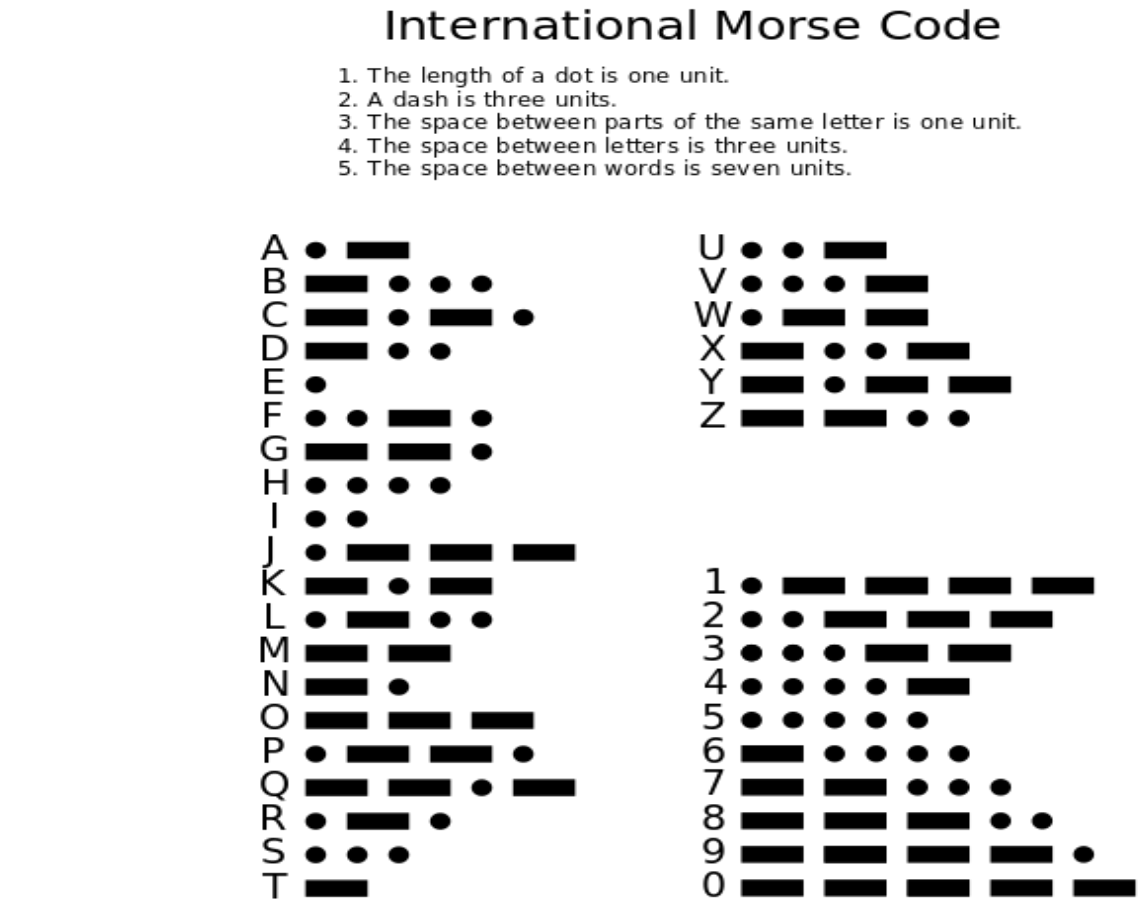
This project uses an Arduino UNO board along with a push button switch and serial monitor to decode Morse code.

Morse code is used in telecommunication; it is a method of transmitting and receiving coded information. Each character (letter or numeral) is coded/represented by a unique sequence of dots and dashes. Compared to voice, Morse code is less sensitive to poor signal conditions, yet still comprehensible to humans without a decoding device, therefore, a useful alternative to synthesized speech for sending automated data to skilled listeners (radio operator) on a voice channel.

The project's first part is composed of an electric microphone followed by a first level band pass filter. Its band edges are determined by the size of the coupling capacitors, and the feedback capacitor between the transistor's base and collector terminals.

Morse code is usually transmitted by on-off keying of an information-carrying medium such as electric current, radio waves, visible light, or sound waves. The current or wave is present during the time period of the dot or dash and absent during the time between dots and dashes.

In an emergency, Morse code can be generated by improvised methods such as turning a light on and off, tapping on an object or sounding a horn or whistle, making it one of the simplest and most versatile methods of telecommunication. The most common distress signal is SOS – three dots, three dashes, and three dots – internationally recognized by treaty.


International Morse Code
1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
1
2
3
4
5
6
7
8
9
0

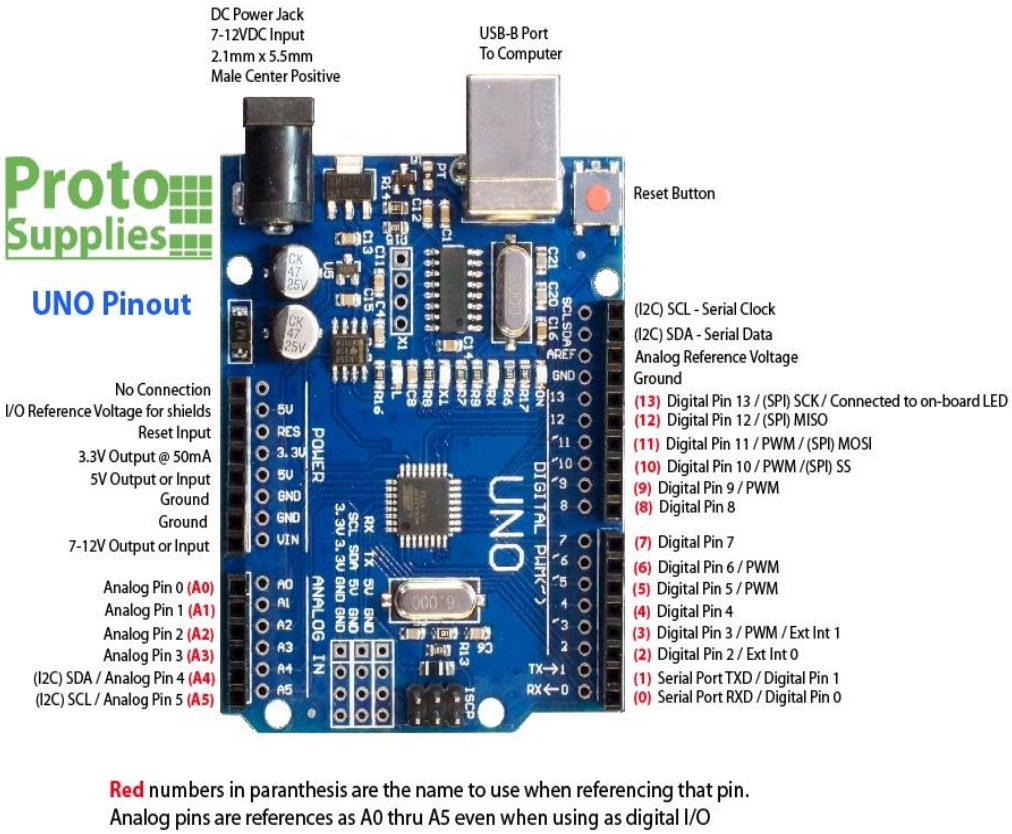

- dot
- dash

CHAPTER 2

HARDWARE REQUIREMENTS

1. ArduinoUNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuitshe board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.



2. Breadboard

A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused.

3. Tactileswitch(Pushbutton)

In electrical engineering, a switch is an electrical component that can "make" or "break" an electrical circuit, interrupting the current or diverting it from one conductor to another. The mechanism of a switch removes or restores the conducting path in a circuit when it is operated. It may be operated manually, for example, a light switch or a keyboard button, may be operated by a moving object such as a door, or may be operated by some sensing element for pressure, temperature or flow. A switch will have one or more sets of contacts, which may operate simultaneously, sequentially, or alternately. Switches in high-powered circuits must operate rapidly to prevent destructive arcing, and may include special features to assist in rapidly interrupting a heavy current. Multiple forms of actuators are used for operation by hand or to sense position, level, temperature or flow. Special types are used, for

example, for control of machinery, to reverse electric motors, or to sense liquid level.

4. LED(5mm)

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor.[5] White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device. LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. LEDs are used in applications as diverse as aviation lighting, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper, plant growing light, and medical devices.

5. Resistor(220Ω)

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can

dissipate many watts of electrical power as heat, may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity. Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in electronic equipment. Practical resistors as discrete components can be composed of various compounds and forms. Resistors are also implemented within integrated circuits. The electrical function of a resistor is specified by its resistance: common commercial resistors are manufactured over a range of more than nine orders of magnitude. The nominal value of the resistance falls within the manufacturing tolerance, indicated on the component.

6. Jumper wires Male-to-Male (4 nos.)

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their

"end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

7. 512MBRAM(1GBRecommended)

Random-access memory (RAM /ræm/) is a form of computer memory that can be read and changed in any order, typically used to store working data and machine code. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In contrast, with other direct-access data storage media such as hard disks, CD-RWs, DVD-RWs and the older magnetic tapes and drum memory, the time required to read and write data items varies significantly depending on their physical locations on the recording medium, due to mechanical limitations such as media rotation speeds and arm movement. RAM contains multiplexing and demultiplexing circuitry, to connect the data lines to the addressed storage for reading or writing the entry. Usually more than one bit of storage is accessed by the same address, and RAM devices often have multiple data lines and are said to be "8-bit" or "16-bit", etc. devices cells.

8. Laptop/Desktop computer for Serial Monitor

SOFTWARE REQUIREMENTS

1. Microsoft Windows 7, Windows 8/8.1 or Windows 10 operating system. An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources. For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

2. Microsoft,.NET,Framework3(or)higher.

   .NET Framework (pronounced as "dot net") is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named as Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named the Common Language Runtime (CLR). The CLR is an application virtual machine that provides services such as security, memory management, and exception handling. As such, computer code written using .NET Framework is called "managed code". FCL and CLR together constitute the .NET Framework.

3. Intel Pentium / AMD processor or equivalent running at 1 GHz or more. In computing, a processor or processing unit is an electronic circuit which performs operations on some external data source, usually memory or some other data stream. The term is frequently used to refer to the central processor (central processing unit) in a system, but typical computer systems (especially SoCs) combine a number of specialised "processors".

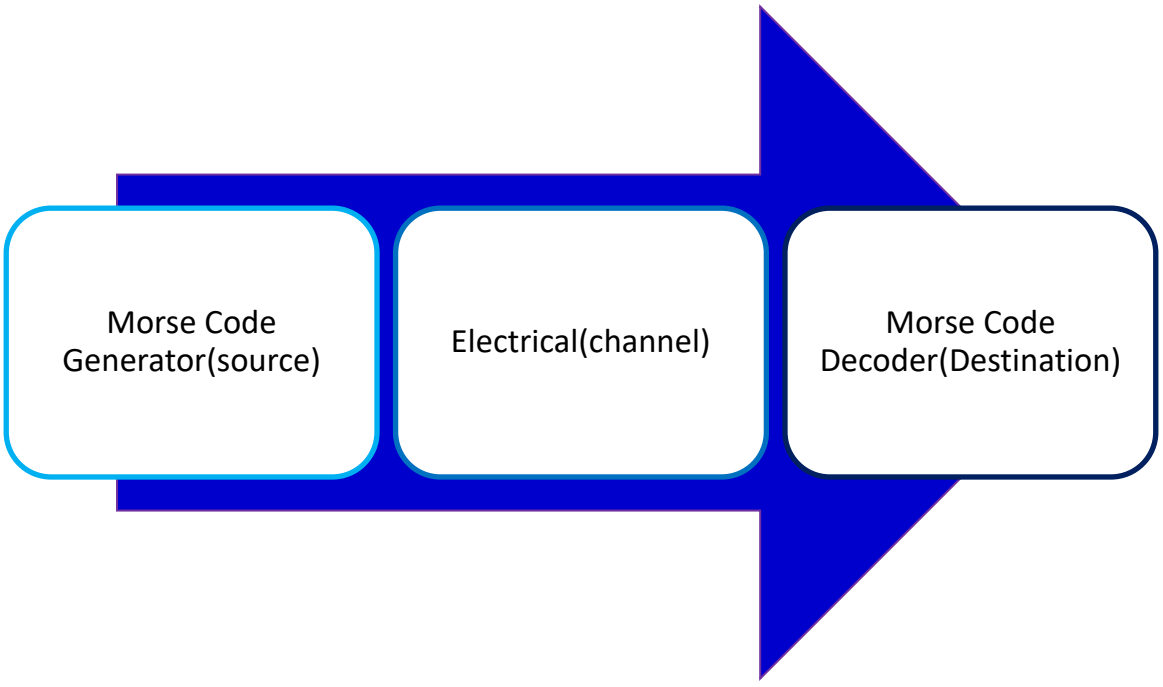
4. AurduinoIDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2.[4] 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 toolchain, also included with the IDE distribution. he 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.

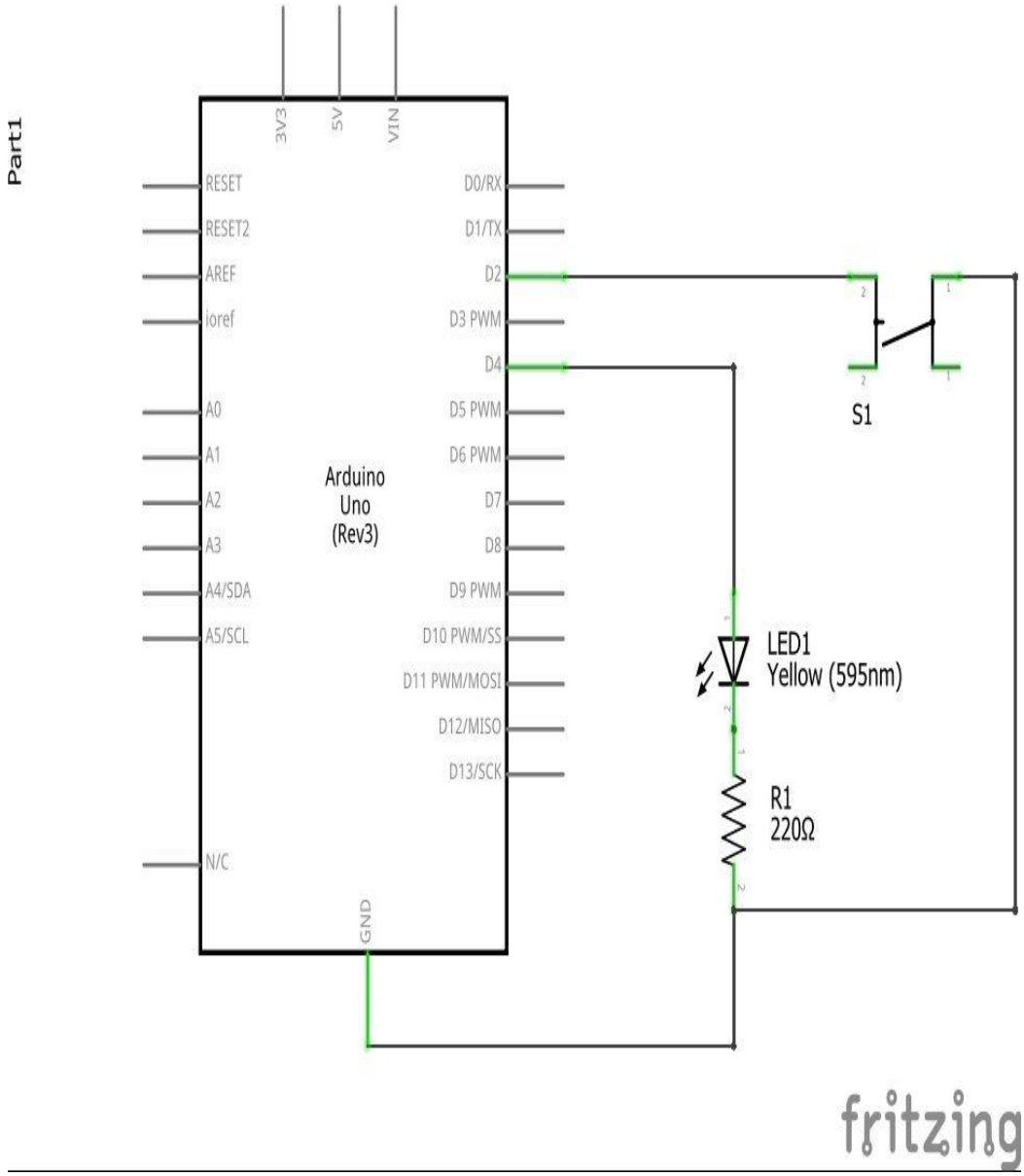
CHAPTER 3

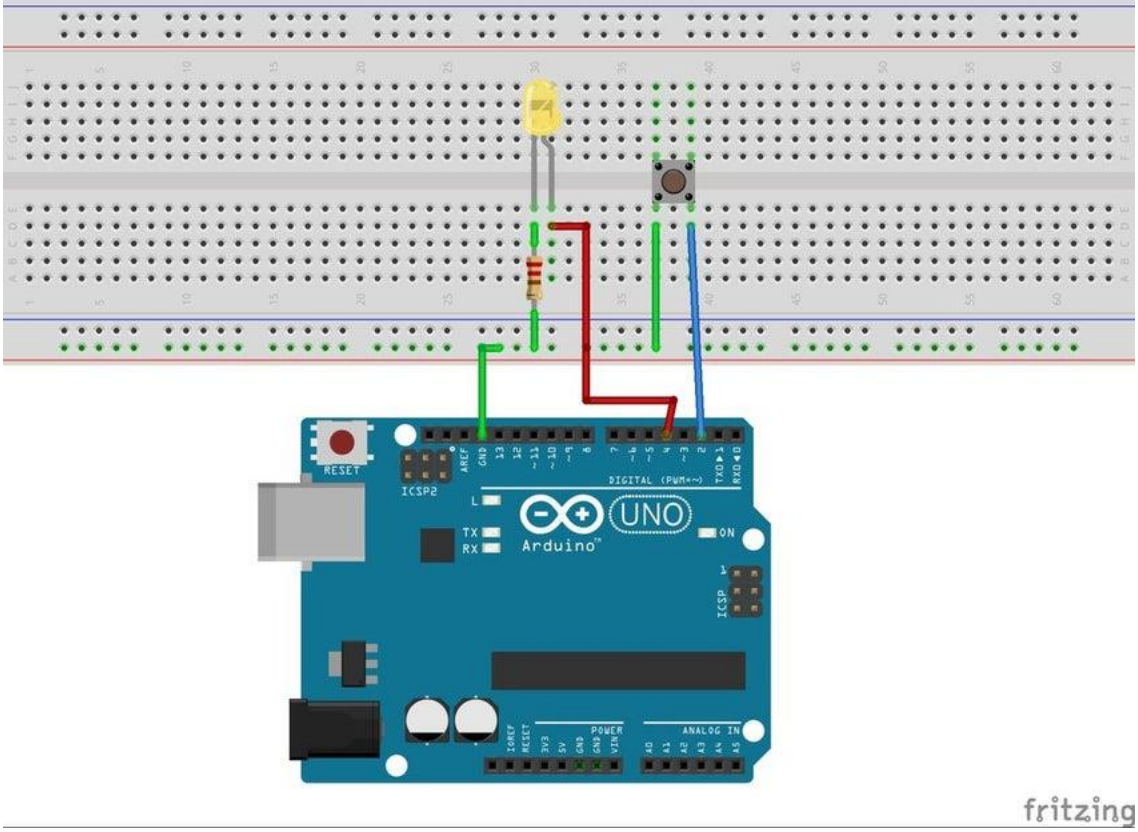
DESIGN METHOD

The basic design for the decoder pipeline will be as follows:



1. Block diagram of the project





1. Connect one end of the push button to the digital I/O pin no. 2 of the Arduino UNO board and the other end to the ground.

2. Connect the anode of the LED to the digital I/O pin no. 4 and the cathode to the ground through a 220 Ω resistor.

3. Connect the Arduino UNO to a PC using the serial communication USB cable.

4. After completion, your circuit should look similar to the given picture.

CHAPTER 4

MODULE DESCRIPTION

1. Once you have uploaded the program to the Arduino board, you are ready to use your Morse code decoder.

2. You can refer the attached Morse code chart to try out your decoder.

3. The LED helps to know for how long the push button is pressed, so that you can differentiate between a dot and a dash.

4. You will have to open the serial monitor from your Arduino app on the PC. All decoded Morse codes will appear on the serial monitor.

- Coding of functional modules

```
unsigned long signal_len,t1,t2;   //time for which button is pressed
int inputPin = 2;                 //input pin for push button

int ledPin = 4;                  //output pin for LED

String code = "";                 //string in which one alphabet is stored
```

Setup the serial connection and I/O pins:

```
void setup()

{
  Serial.begin(9600);

  pinMode(inputPin, INPUT_PULLUP); //internal pullup resistor is used t
o simplify the circuit
```

```
  pinMode(ledPin,OUTPUT);
}
```

- Main Loop for running program and decoding:

```
void loop()
{
NextDotDash:
  while (digitalRead(inputPin) == HIGH) {}
  t1 = millis();                    //time at button press
  digitalWrite(ledPin, HIGH);       //LED on while button pressed
  while (digitalRead(inputPin) == LOW) {}
  t2 = millis();                    //time at button release
  digitalWrite(ledPin, LOW);        //LED off on button release
  signal_len = t2 - t1;             //time for which button is pressed
  if (signal_len > 50)              //to account for switch debouncing
  {
    code += readio();               //function to read dot or dash
  }
  while ((millis() - t2) < 500)     //if time between button press greater
than 0.5sec, skip loop and go to next alphabet
  {
```

```
   if (digitalRead(inputPin) == LOW)
   {
     goto NextDotDash;
   }
 }
 convertor();              //function to decipher code into alphabet
}
```

- Function for reading dot or dash:

```
char readio()
{
 if (signal_len < 600 && signal_len > 50)
 {
 return '.';                  //if button press less than 0.6sec, it is a dot
 }
 else if (signal_len > 600)
 {
 return '-';                  //if button press more than 0.6sec, it is a dash
 }
}
```

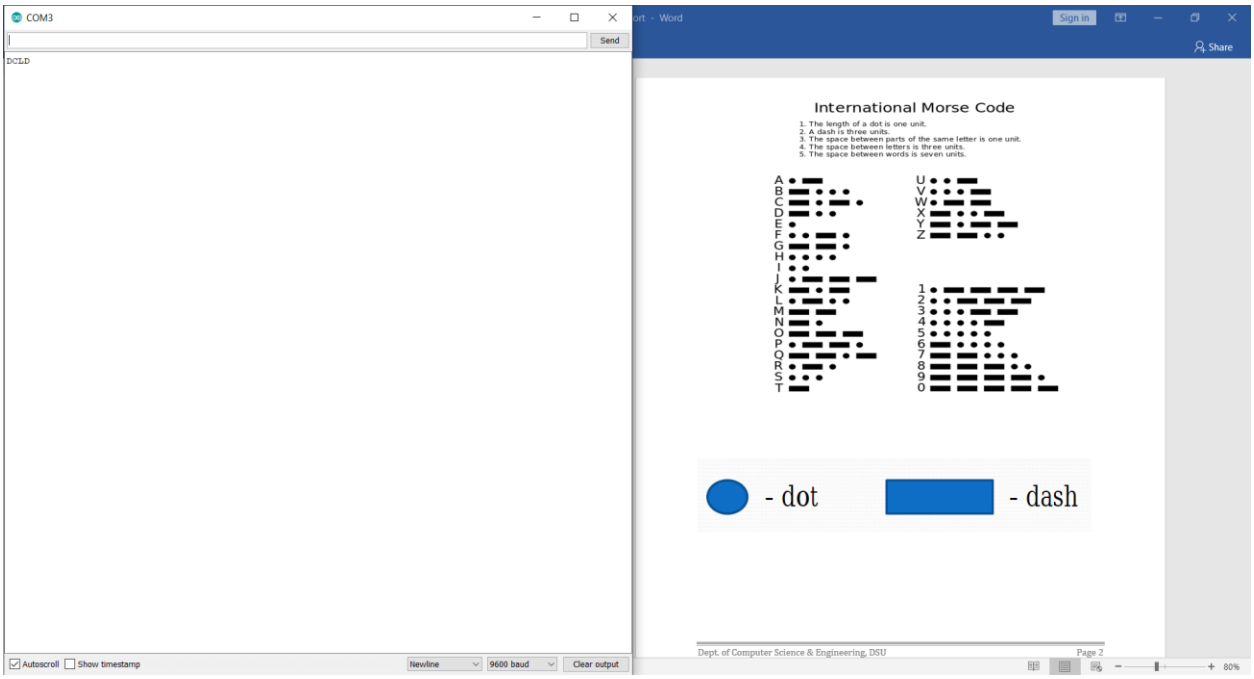
Function for converting dots and dashes into alphabet:

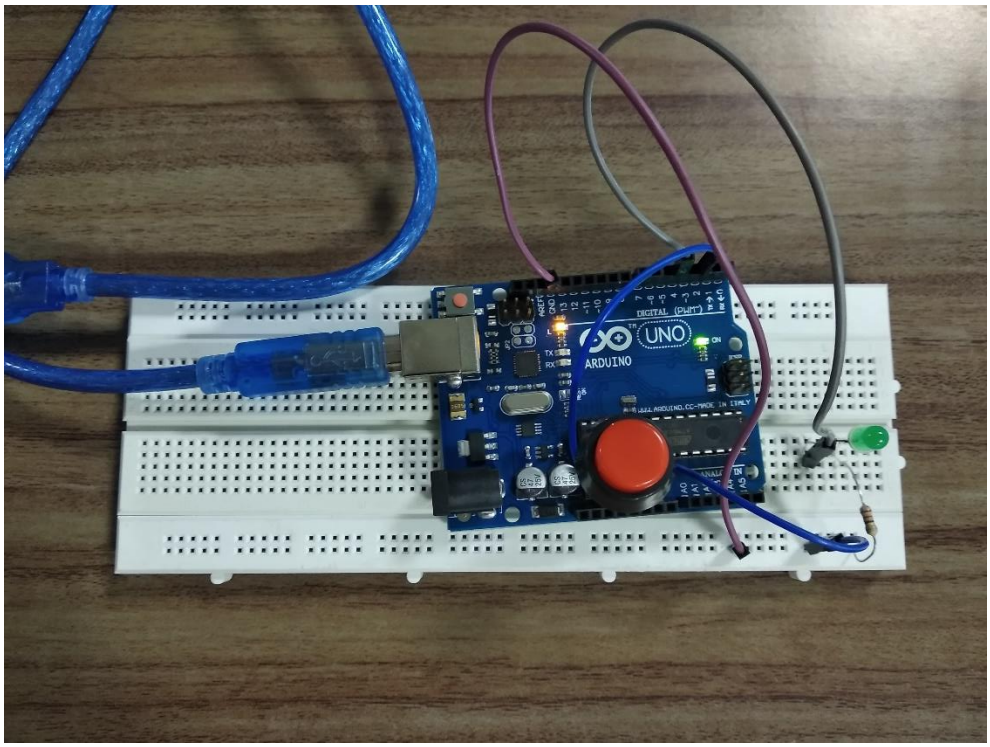
```
void convertor()
{

  static String letters[] = {".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..",
".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-",

                          ".-.", "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", "E"

                         };

  int i = 0;

  if (code == ".-.-.-")

  {

    Serial.print(".");        //for break

  }

  else

  {

while (letters[i] != "E")      //loop for comparing input code with letters ar
ray

    {

      if (letters[i] == code)

      {

        Serial.print(char('A' + i));

        break;

      }

      i++;
```

```
    }
    if (letters[i] == "E")
    {
      Serial.println("");  //if input code doesn't match any letter, error
    }
  }
  code = "";                          //reset code to blank string
}
```

CHAPTER 5

RESULTS





CHAPTER 6

CONCLUSION

We achieved our main goals for this project. Audio could have been added to make the project more “realistic”, but we believe this is a minor issue; the decoder functions in the way we had intended. What we have learned isn’t about hardware technicalities, but overarching themes when it comes to hardware design and the analog-to-digital interface.

We spent a lot of time thinking about how to best convert an analog input into a digital representation. This raised many design-related questions that we realize are always faced when doing some sort of analog-digital conversion, like in all of our modern day electronic devices. Of course, our conversion task was relatively simple. But at the same time it made us think about the kind of design decisions engineers and designers make when, for example, deciding how to interpret the pressure applied on a smart phone screen as some discrete value.

Aeronautical navigational aids, such as VORs and NDBs, constantly identify in Morse code. Compared to voice, Morse code is less sensitive to poor signal conditions, yet still comprehensible to humans without a decoding device. Morse is therefore a useful alternative to synthesized speech for sending automated data to skilled listeners on voice channels. Many amateur radio repeaters, for example, identify with Morse, even though they are used for voice communications. For emergency signals, Morse code can be sent by way of improvised sources that can be easily "keyed" on and off, making it one of the simplest and most versatile

methods of telecommunication. The most common distress signal is SOS or three dots, three dashes and three dots, internationally recognized by treaty.

Thus we have successfully completed a basic Morse code Decoder using an Arduino UNO and a tactile switch.

CHAPTER 7

FUTURE WORK

Although the decoder was successful, there are areas that could be improved on. The decoder could possibly be made more tolerant by tracking the average dash length as well as the average dot length. It would also be interesting to collect data from “real” Morse code that exists, and see how closely it fits the specification that was used in this project.

This would allow the decoder to be tuned for real world use, and so be even more accurate when presented with real Morse code. The most interesting addition would have been a frequency scanner and demodulator which would have allowed the systems to listen to the actual beeps being transmitted over the air, or listen to a speaker the is beeping. This would have made for a more impressive demo and been of slightly more practical value in the real world.

REFERENCES

1) https://www.instructables.com/id/Morse-Code-Decoder/

2) https://en.wikipedia.org/wiki/Morse_code

3) https://www.arduino.cc/en/main/software