Morse Code Trainer

A Lab Project Presented to the Faculty
of
Computer Science and Automation
Ilmenau University of Technology, Germany

Submitted To
M.Sc. Thomas Dietrich

Prepared By
Amr Mahmoud Asad Audi
Md. Romel Hasnat

July 14, 2016

Morse Code Trainer

Task:

To connect a PC keyboard and a beeper to an Arduino. Develop a morse code trainer that will play corresponding Morse code sounds when one of the keys A-Z is pressed.

General Idea:

1) When arduino is active it will give a well come sound e.g, “Hello”
2) After pressing “Enter” key arduino will be transferred signal to the beeper
3) There will be a unique sound for every key (A-Z)
4) There will be a unique sound for the unique word
5) There will be a unique sound for the unique sentences as well
6) “Page Up” and “Page Down” keys to select the letter, word and sentence mode
7) “Shift” key will be define the upper and lower case
8) A error tone will be given pressing the invalid keys

Tools:

1) 1 PS/2 Keyboard
2) 1 Arduino UNO
3) 1 Beeper
4) 1 Power supply (5V)
5) Connecting wires

- **Design:**

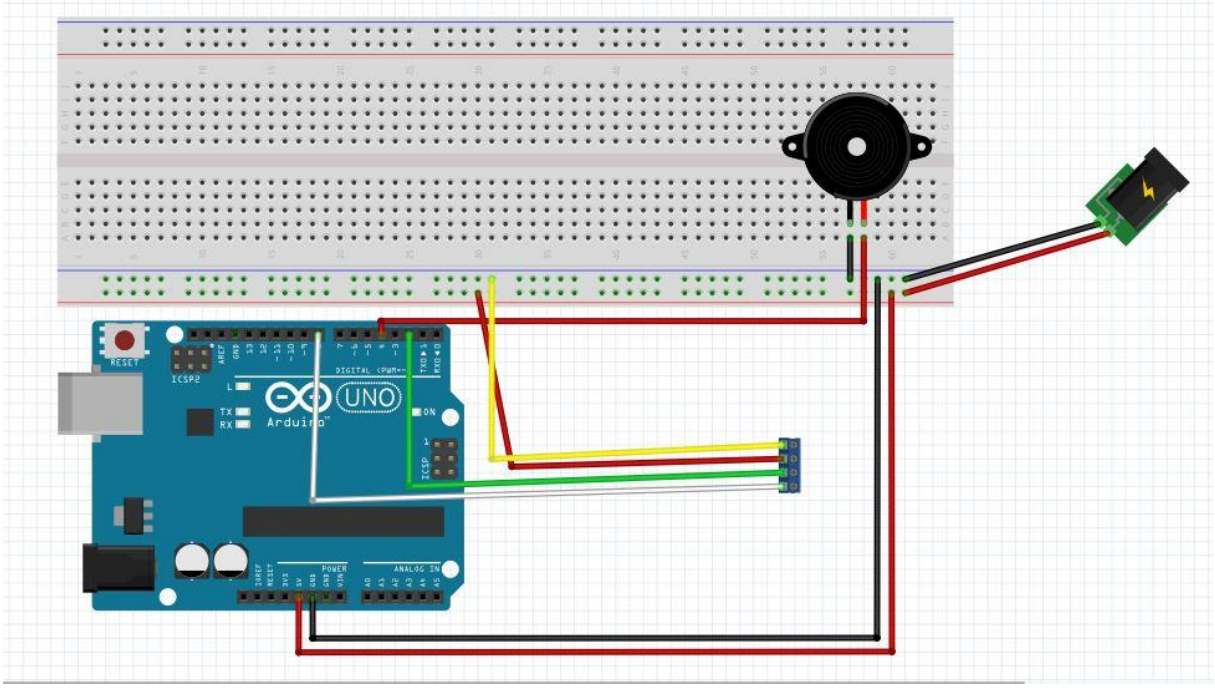

Fig 01: Morse code project design in Fritzing

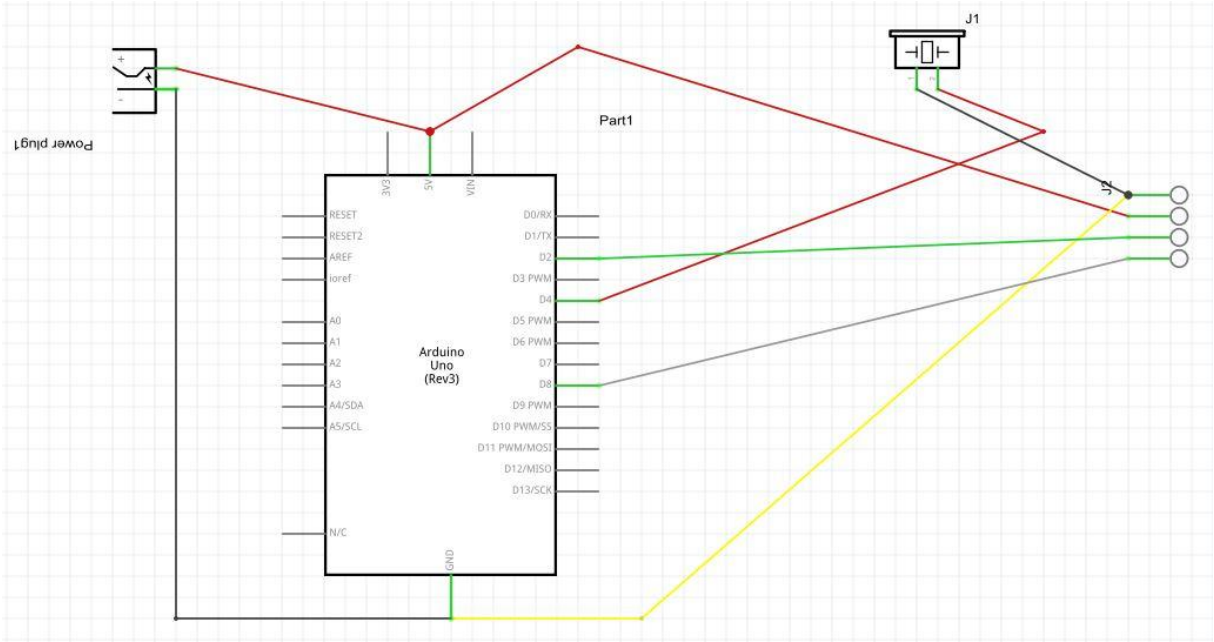

Fig 02: Circuit diagram

Description:

In the above figure (1) the Morse code project design is shown in Fritzing designing tool. Here two cables red and black are connected with the power supply and with the bread board. Red cable is for 5V power supply and black cable is for ground. For supplying power to the arduino board another red cable is connected with the power socket of breadboard to the 5V pin of arduino board and a black cable is connected from the ground socket of bread board to the GND pin of arduino.

A beeper also attached with the bread board. The red cable of beeper is connected with the pin.4 of the arduino board. This is transferring signal from arduino to the beeper. And the black cable is for grounding. We used a PS/2 keyboard to input command. There are four cables in PS/2 keyboard. Red colored one is for power, yellow one is for ground, green one is for clock and white one is for the data. The yellow and red cable is connected with the bread board for power and grounding. Green cable is connected to the pin.2 of arduino board and white cable is connected with pin.8 of the arduino board.

- **PS/2 Keyboard Cable Color Combination:**

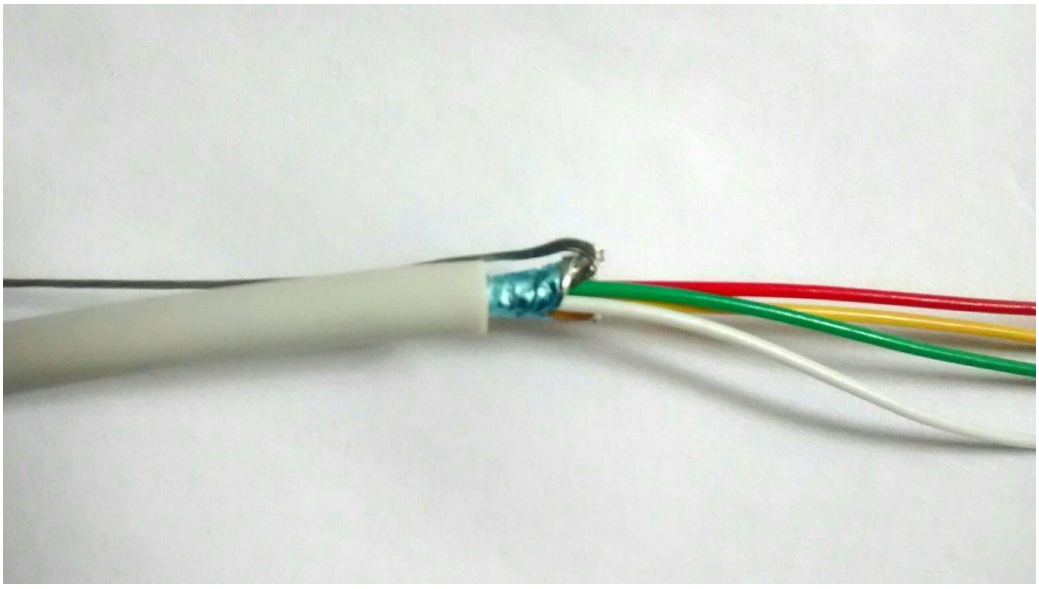

Fig 03: Keyboard cable color combination

Color	Purpose
Red	Power
Yellow	Ground
Green	Clock
White	Data

Table 1: Color combination of PS/2 keyboard

- **Arduino Pin Combination:**

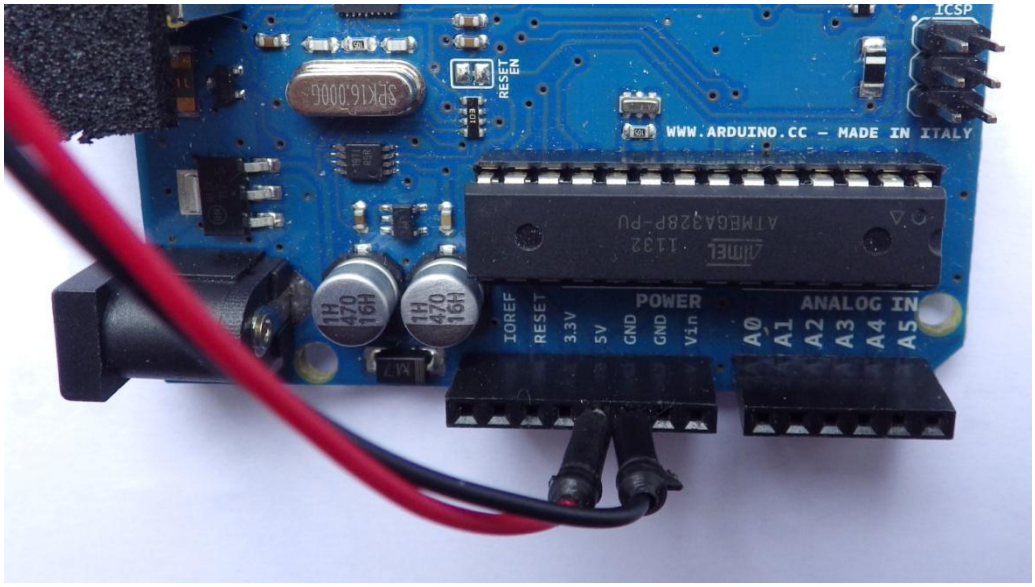

Fig 04: Power connection of with arduino

Arduino Pin	Purpose
5V (+Ve):	Red
GND (-Ve):	Black

Fig 05: Data, Clock and Signal connection in arduino

Arduino Pin	Purpose
Pin 2	Clock (White)
Pin 4	Signal to beeper
Pin 8	Data

- **Connections:**

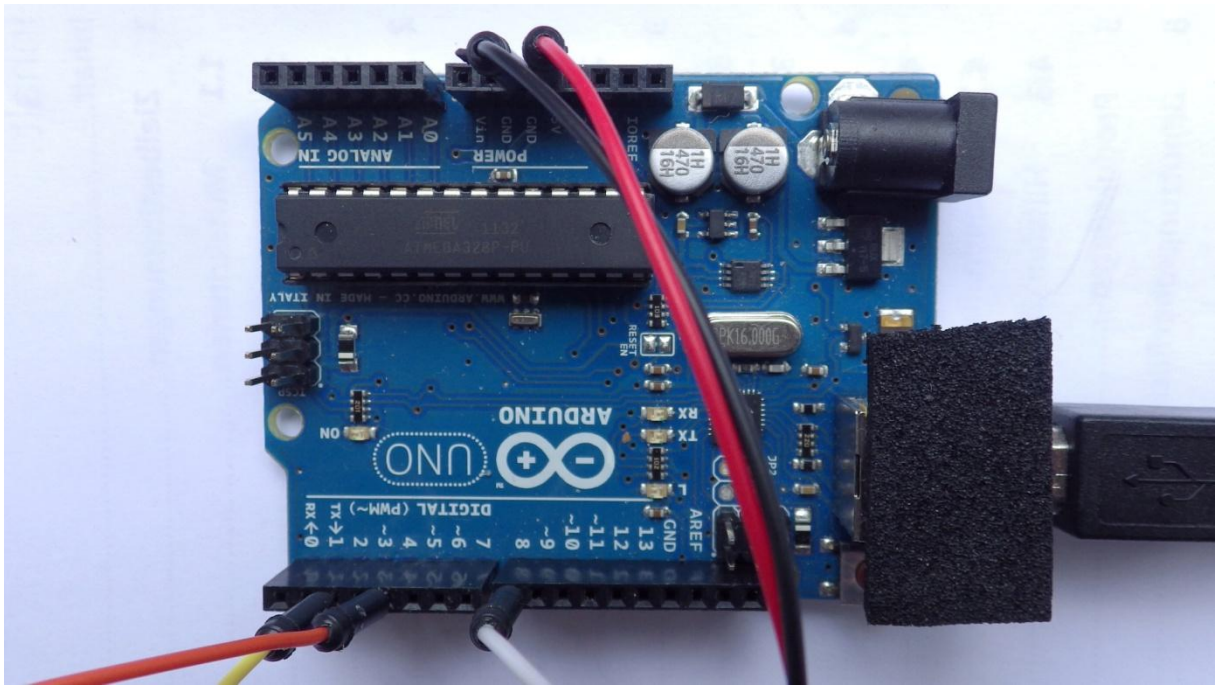

Fig 06: Full arduino pin connections

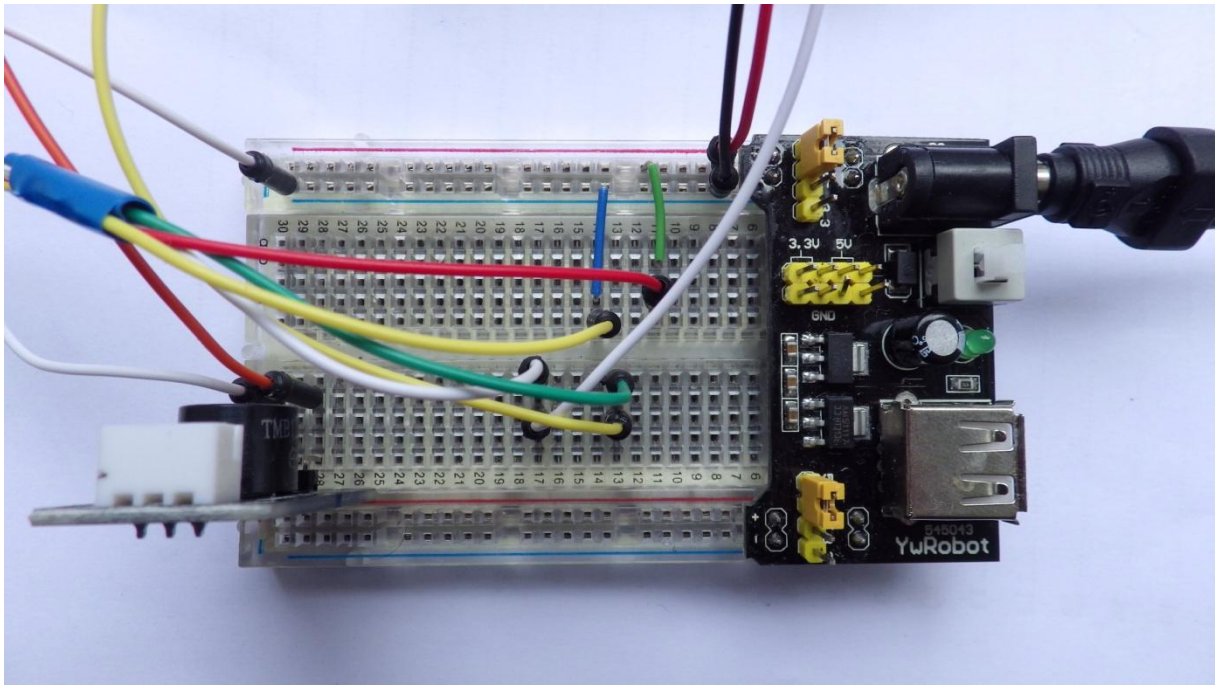

Fig 07: Full breadboard pin connections

Code:

```
#include <PS2Keyboard.h>

const int DataPin = 8;

const int IRQpin = 2;

static int CurrentMode=1 ;

static String Cache = "";

PS2Keyboard keyboard;

#define PS2_PAGEUP      25

#define PS2_PAGEDOWN    26




void setup() {


  delay(1000);

  keyboard.begin(DataPin, IRQpin);

  Serial.begin(9600);

  SayHello();


}



void loop() {

  if (keyboard.available()) {

    char c = keyboard.read();

    Serial.write(c);

    if (c == PS2_PAGEUP || c == PS2_PAGEDOWN)

    {


      if (c == PS2_PAGEUP)
```

```
{
  if (CurrentMode == 1)
  {
    CurrentMode = 2;
    tone(4, 100, 200);
    tone(4, 100, 200);

  }
  else if (CurrentMode == 2)
  {
    CurrentMode = 3;
    tone(4, 100, 200);
    tone(4, 100, 200);
    tone(4, 100, 200);

  }
  else if (CurrentMode == 3)
  {
    CurrentMode = 1;
    tone(4, 100, 200);

  }
}
else
{
  if (CurrentMode == 1)
  {
    CurrentMode = 3;
    tone(4, 100, 200);
    tone(4, 100, 200);
    tone(4, 100, 200);
```

```
    }
    else if (CurrentMode == 2)
    {
      CurrentMode = 1;
      tone(4, 100, 200);


    }
    else if (CurrentMode == 3)
    {
      CurrentMode = 2;
      tone(4, 100, 200);
      tone(4, 100, 200);

    }
   }

   return;
  }

  KeyPress(c);
}

if (Serial.available() != 0) {
  char c = Serial.read();
  Serial.write(c);

  if (c == PS2_PAGEUP || c == PS2_PAGEDOWN)
  {
```

```
if (c == PS2_PAGEUP)
{
  if (CurrentMode = 1)
  {
    CurrentMode = 2;
    tone(4, 100, 200);
    tone(4, 100, 200);

  }
  else if (CurrentMode = 2)
  {
    CurrentMode = 3;
    tone(4, 100, 200);
    tone(4, 100, 200);
    tone(4, 100, 200);

  }
  else if (CurrentMode = 3)
  {
    CurrentMode = 1;
    tone(4, 100, 200);

  }
}
else
{
  if (CurrentMode = 1)
  {
    CurrentMode = 3;
    tone(4, 100, 200);
```

```
        tone(4, 100, 200);
        tone(4, 100, 200);

      }
      else if (CurrentMode = 2)
      {
        CurrentMode = 1;
        tone(4, 100, 200);

      }
      else if (CurrentMode = 3)
      {
        CurrentMode = 2;
        tone(4, 100, 200);
        tone(4, 100, 200);

      }
    }

    return;
  }

  KeyPress(c);
 }

}



void KeyPress(char c)
{
 if (CurrentMode == 1)
```

```
{
  SendTone(c);


}
if (CurrentMode == 2)
{
  if (c == ' ')
  {
    for (int i = 0; i < Cache.length(); i++)
    {
      SendTone(Cache[i]);
    }
    SendTone(c);
    Cache = "";


  }
  else {
    Cache = Cache + c;


  }
}
if (CurrentMode == 3)
{
  if (c == PS2_ENTER)
  {
    for (int i = 0; i < Cache.length(); i++)
    {
      SendTone(Cache[i]);
    }
    //SendTone(c);
    Cache = "";
```

```
    }
    else {
      Cache = Cache + c;

    }
  }
}




void SayHello()
{
  SendTone('h');
  SendTone('e');
  SendTone('l');
  SendTone('l');
  SendTone('o');
}

void SendTone( char c)
{
  c=tolower(c);
  switch (c)
  {
    case 'a':
      p(1); l(1);
      break;
    case 'b':
      l(1); p(3);
      break;
    case 'c':
```

```
  l(1); p(1); l(1); p(1);
  break;
case 'd':
  l(1); p(2);
  break;
case 'e':
  p(1);
  break;
case 'f':
  p(2); l(1); p(1);
  break;
case 'g':
  l(2); p(1);
  break;
case 'h':
  p(4);
  break;
case 'i':
  p(2);
  break;
case 'j':
  p(1); l(3);
  break;
case 'k':
  l(1); p(1); l(1);
  break;
case 'l':
  p(1); l(1); p(2);
  break;
case 'm':
  l(2);
```

```
  break;
case 'n':
  l(1); p(1);
  break;
case 'o':
  l(3);
  break;
case 'p':
  p(1); l(2); p(1);
  break;
case 'q':
  l(2); p(1); l(1);
  break;
case 'r':
  p(1); l(1); p(1);
  break;
case 's':
  p(3);
  break;
case 't':
  l(1);
  break;
case 'u':
  p(2); l(1);
  break;
case 'v':
  p(3);  l(1);
  break;
case 'w':
  p(1); l(2);
  break;
```

```
case 'x':
  l(1); p(2); l(1);
  break;
case 'y':
  l(1); p(1); l(2);
  break;
case 'z':
  l(2); p(2);
  break;
case '1':
  p(1); l(4);
  break;
case '2':
  p(2); l(3);
  break;
case '3':
  p(3);  l(2);
  break;
case '4':
  p(4);  l(1);
  break;
case '5':
  p(5);
  break;
case '6':
  l(1); p(4);
  break;
case '7':
  l(2); p(3);
  break;
case '8':
```

```
      l(3);  p(2);
      break;
    case '9':
      l(4);  p(1);
      break;
    case '0':
      l(5);
      break;
    case ' ':
      pausa(800);
      break;
    case PS2_ENTER:
      l(6);
      break;
    default:
      tone(4, 300, 550); //error tone
      break;

  }
  pausa(200);
}

void p(int repetetions) { //dot
  for (int i = 0; i < repetetions; i++)
  {
    digitalWrite(13, HIGH);
    tone(4, 1000);
    delay(66);
    digitalWrite(13, LOW);
    noTone(4);
    delay(66);
```

```
  }
}
void l(int repetetions) { //dash
  for (int i = 0; i < repetetions; i++)
  {
    digitalWrite(13, HIGH);
    tone(4, 1000);
    delay(198);
    digitalWrite(13, LOW);
    noTone(4);
    delay(68);
  }
}
void SendHelloMessage()
{
  SendTone('h');
  SendTone('e');
  SendTone('l');
  SendTone('l');
  SendTone('o');

}
void pausa(int delayLength) { //pause between words for 1000 and letters for 198
  delay(delayLength);
}
```