T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

PROJECT FINAL REPORT
MORSE INTERPRETER

EEE1002 Course Project

Esat Canberk Özçelik

İSTANBUL,May 2019

Contents

1 Overview

1.1 Problem Statement and Objectives

For learners and beginners it may be hard to analyse the Morse code at first glance. Aim of this project is to assemble a design where one can convert the Morse code entered through key to LCD screen as ASCII characters. One can also give inputs via light where it can be useful for direct interpretation of code via long distances. In addition, a secondary two-part circuitry has added where the purpose of these circuitry is as follows:

- Convert the sound waves from a sound source to light waves.
- Convert the light wave back to sound.

With this, one can send information through great distances in field where it can have a helpful use. This sound can be Morse code, where the user will receive the signals from headphones and use the interpreter to translate the message or it can also be used to send listen-able audio directly.

1.2 Background Information

The word "telegraph" comes from Ancient Greek, where "têle" means "at distance" and "gráphein" means "to write". Telegraphy dates back thousands of years ago when people first started to communicate over distances by the drum sounds or the smoke and light of the fire. throughout time many techniques have been used such as Semaphore Telegraphs and Optical Telegraphs(Heliographs), where in Semaphore two flags in two hands are changed places according to message, a man could hold this flags or for more conventional use a tower would be used with wooden arms; and for optical telegraphy beams of sun would be directed to receiving tower according to the message as on-off signals.

As the time progressed electrical telegraph started to be seen and in 1847, Samuel Morse received the patent for his telegraph at the Beylerbeyi Palace, İstanbul from Sultan Abdülmecid,who personally tested the new invention[1]. With the conference held in Vienna, 1851 it is decided to adopt the Morse telegraph as the system for international communications[2].

Samuel Morse (1791-1872)

As technology advanced the use of telegraphy decreased and disbanded but the Morse Code and the basis ideas of its techniques are still being used by the Military, Scouts and Amateur Radio Operators.

2 Methodology

2.1 Project Design

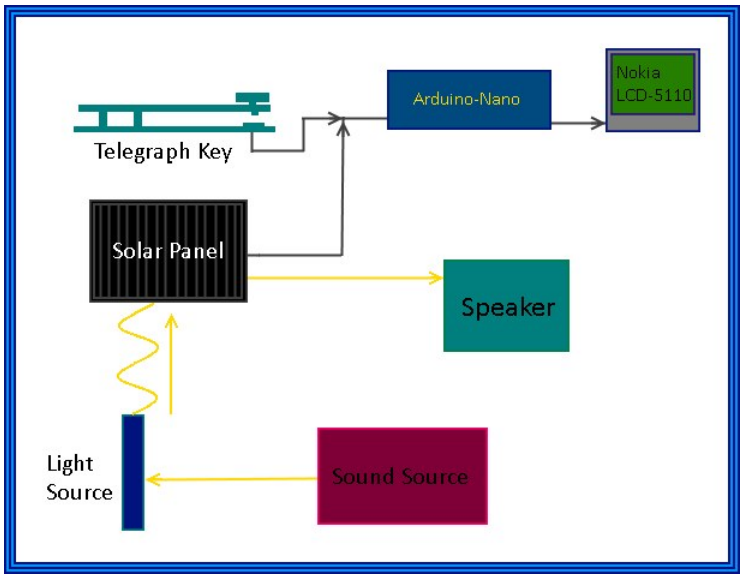


Figure 1

The design consists of two sections. One can see these sections in *Figure1* shewn with *Grey* and *Yellow*.
In the section *Grey*, one can give inputs of Morse code by the telegraph key or by a light source directed to the solar panel and by the Arduino, it will be interpreted and shewn on the LCD as ASCII characters.
In the *Yellow* section one can transmit data by laser from vast distances and transmit the message. This message can be a Morse code or direct sound. In this procedure we are transmitting data by the light waves. After the transmission, the solar panel, acting as a photosensor, will convert the light waves to electric current and the sound will be played through the speakers.

2.2 Project Components

- Arduino Nano
- Nokia 5110 LCD
- Light Emitting Diode(LED)
- Buzzer

2.2.1 Hardware

For the controller I have selected Arduino Nano for its small size, which uses ATmega328P micro controller. For the screen I used Nokia 5110 LCD screen and wrote a code from the data sheet. As the key turns on and off, a buzzer with and LED wired series to key indicates the situation of the key.

For the secondary circuit, a light source(LED or Laser) is connected to a sound source and 5V power supply.

***Detailed schematic of the circuit is given in the attachment.**

2.2.2 Software

The software consists of two different function amalgamated into one. Briefly these functions effectuate the following tasks:

- Running the LCD
- Interpreting the Morse Signal

After investigation of Data Sheet of the Nokia, 5110-LCD and online research I assembled the code for the LCD. This task required the HEX decimal interpretation of the desired actions to code. With the table given in the next page, I have adjusted the and wrote library and code. To apply a required task, I wrote the binary commands according to table then convert it to hexadecimal to give the commands to Arduino.And here I shall give credit where the educational videos of Julian Ilet on *YouTube* has helped me for the analysing of the data sheet and the font library[4].

Later the need was to have a font to print therefore a latter code was added as a library for this task. Now one can print anything of one desires, adjust the screen brightness and so on. Now the next task is to convert digital signals in to *dits(".")* and *(dahs("-")*. We over come this problem with the use of *"Millis"* function.

Table 1 Instruction set

INSTRUCTION	D/$\overline{C}$	COMMAND BYTE								DESCRIPTION
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(H = 0 or 1)										
NOP	0	0	0	0	0	0	0	0	0	no operation
Function set	0	0	0	1	0	0	PD	V	H	power down control; entry mode; extended instruction set control (H)
Write data	1	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	writes data to display RAM
(H = 0)										
Reserved	0	0	0	0	0	0	1	X	X	do not use
Display control	0	0	0	0	0	1	D	0	E	sets display configuration
Reserved	0	0	0	0	1	X	X	X	X	do not use
Set Y address of RAM	0	0	1	0	0	0	Y_2	Y_1	Y_0	sets Y-address of RAM; $0 \le Y \le 5$
Set X address of RAM	0	1	X_6	X_5	X_4	X_3	X_2	X_1	X_0	sets X-address part of RAM; $0 \le X \le 83$
(H = 1)										
Reserved	0	0	0	0	0	0	0	0	1	do not use
	0	0	0	0	0	0	0	1	X	do not use
Temperature control	0	0	0	0	0	0	1	TC_1	TC_0	set Temperature Coefficient (TC_x)
Reserved	0	0	0	0	0	1	X	X	X	do not use
Bias system	0	0	0	0	1	0	BS_2	BS_1	BS_0	set Bias System (BS_x)
Reserved	0	0	1	X	X	X	X	X	X	do not use
Set V_{OP}	0	1	V_{OP6}	V_{OP5}	V_{OP4}	V_{OP3}	V_{OP2}	V_{OP1}	V_{OP0}	write V_{OP} to register

Figure 2: A table from Nokia LCD 5110 Data Sheet[3]

```
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 > 30)                        //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
}
```

***Complete code is given in the appendix 4.1**

2.3 Final Product and Results

Altough I have designed a 3D model for the exterior body, due to limitations I could not print it but the design can be found at the appendix. Further on, I have proceed with a cardboard design for the main body and used a 3D printed Morse key which I have manufactured in the past. For the secondary, after I soldered the stable components, I have used crocodile clips to be able to change between light sources such as LED and Laser with ease.

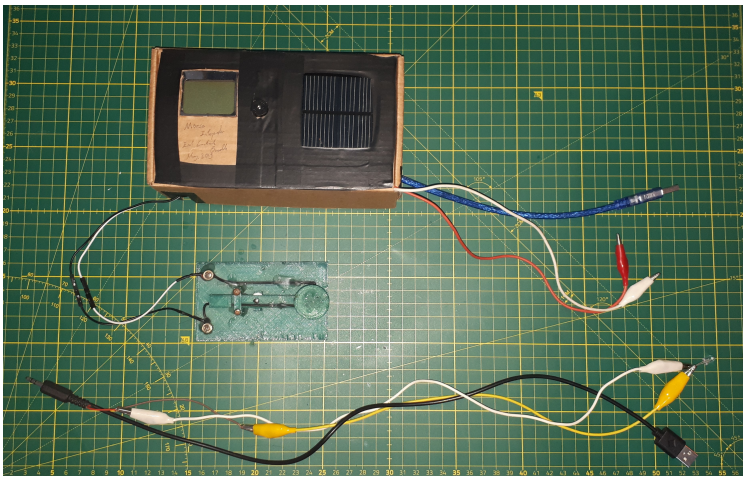


Figure 3: All the Components

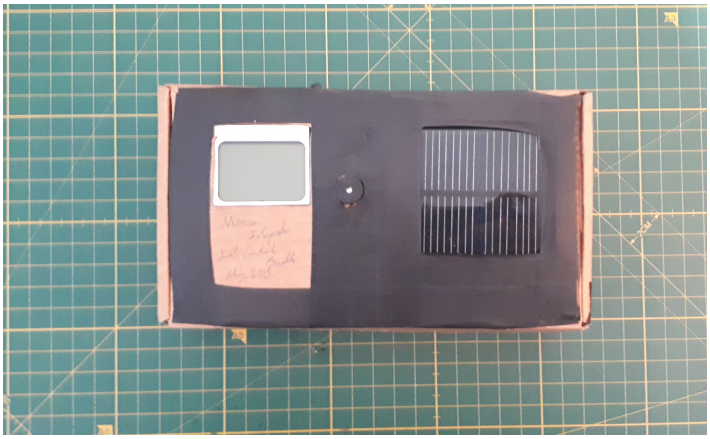


Figure 4: Main Processor

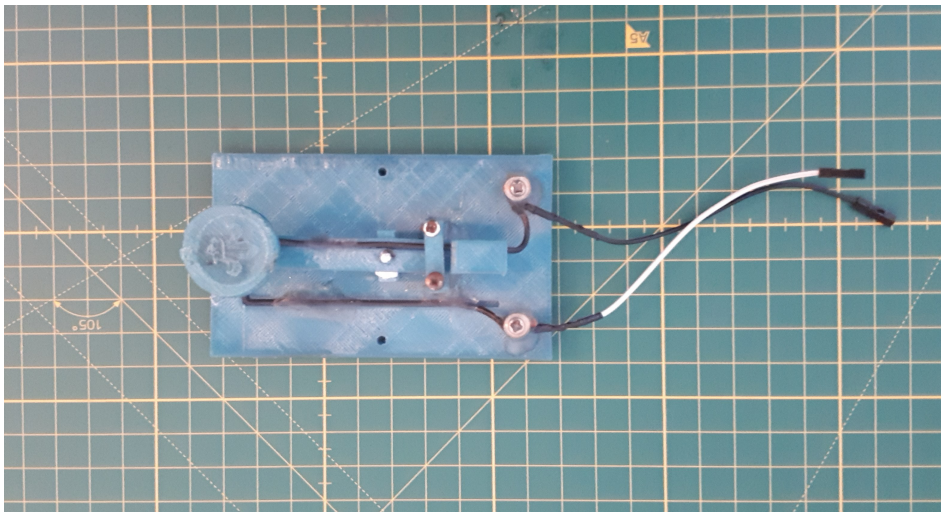


Figure 5: Morse Key

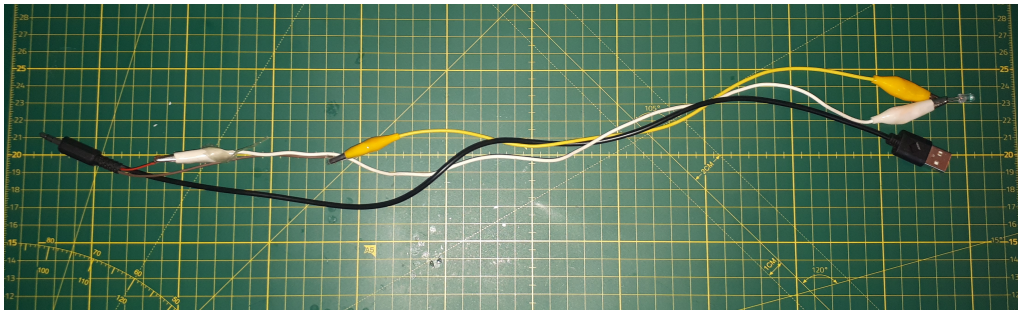


Figure 6: Sound to Wave Interpreter

3 Work Plan

3.1 Tasks and Time Line

The major concern in the project was the code for analysis of the input signals and assigning them as *dits(".")* or *dahs("-")*. After some research I have found the Arduino function *"Millis"* and managed to over come this problem.

The second problem was to run the LCD. For this task, I could have used some library for simplicity but foreseeing that it would be running to ease and earn me nothing in the end, I wanted to run the LCD by my own code or perhaps write a library. I found the *Data Sheet* for the *Nokia-LCD 5110* and after some reading and additional research I managed to figure out how I could run the LCD and from that I wrote few lines of code and wrote functions according to my needs.

The project Gantt chart.

	WEEKS										
TASK LIST	1	2	3	4	5	6	7	8	9	10	11
1. Research and Design											
1.1 Define the goals, and perform a literature search.											
1.2 Prepare an overall design and order parts.											
2. Construction and Coding											
2.1 Assemble system A											
2.2 Assemble system B											
2.3 Prepare Software											
2.4 Integrate all systems											
3. Testing and Documentation											
3.1 Prepare the proposal											
3.2 Verification											
3.3 Prepare the report											
3.4 Presentation											

Figure 7: Gannt Chart

3.2 Project Costs

Component	**Cost(TL)**
Arduino-Nano	33,00
Nokia 5110 LCD	20,00
Buzzer	1,32
Light Emitting Diode	1,90
Jumper Cables	17,70
Bread Board	9,50
Solder Paste-50g	15,68
Soldering Wire-200g	44,99
Soldering Station	134,50
Sum	**260,49**

4 Conclusion

As conclusion we have managed to take input values from mechanical and light-varied keys, interpret them to Morse Code and print them on the LCD screen. With addition of the secondary, we also managed to convert sound to light waves and back to sound waves thus enabling us to transmit any information over vast distances.

By the course of the project I have learned the function of Arduino *"Millis"* and how to work with timers. I have learned the basics of the operations of an LCD by programming it by analysing its data sheet and where which I also gained change to use my skills of binary and hexadecimal conversions which we have gained in the *MAT1006 Engineering Mathematics* course.

5 References

1. https://web.archive.org/web/20071010112702/http://www.istanbulcityguide.com/history/body_mansions_palaces.htm
2. https://mysite.du.edu/~jcalvert/tel/morse/morse.htm
3. https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf
4. https://www.youtube.com/user/julius256

6 Appendix

6.1 Code

6.1.1 Main

```
#define RST 12
#define CE  11
#define DC  10
#define DIN  9
#define CLK  8
#include "font.h";

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

void LcdWriteString(char *characters){
  while(*characters) LcdWriteCharacter(*characters++);
}

void LcdWriteCharacter(char character){
for(int i=0;i<5;i++) LcdWriteData(ASCII[character-0x20][i]);
LcdWriteData(0x00);
}
void LcdWriteData(byte dat)
{
  digitalWrite(DC, HIGH); //DC pin is low for commands
  digitalWrite(CE, LOW);
  shiftOut(DIN, CLK, MSBFIRST, dat); //transmit serial data
  digitalWrite(CE, HIGH);
}
void LcdWriteCmd(byte cmd)
{
  digitalWrite(DC, LOW); //DC pin is low for commands
  digitalWrite(CE, LOW);
  shiftOut(DIN, CLK, MSBFIRST, cmd); //transmit serial data
  digitalWrite(CE, HIGH);


}
void setup()

{
  Serial.begin(9600);
```

```
  pinMode(inputPin, INPUT_PULLUP); //internal pullup resistor is used to
  //simplify the circuit
  pinMode(ledPin,OUTPUT);

  pinMode(RST, OUTPUT);
  pinMode(CE, OUTPUT);
  pinMode(DC, OUTPUT);
  pinMode(DIN, OUTPUT);
  pinMode(CLK, OUTPUT);
  digitalWrite(RST, LOW);
  digitalWrite(RST, HIGH);

  LcdWriteCmd(0x21);  // LCD extended commands
  LcdWriteCmd(0xA9);  // set LCD Vop (contrast)
  LcdWriteCmd(0x04);  // set temp coefficent
  LcdWriteCmd(0x14);  // LCD bias mode 1:40
  LcdWriteCmd(0x20);  // LCD basic commands
  LcdWriteCmd(0x0C);  // LCD normal video


  for(int i=0;i<504;i++)LcdWriteData(0x00);
}

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 > 30)                        //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
```

```
}
char readio()
{
  if (signal_len < 150 && signal_len > 20)
  {
    return '.';                          //if button press less than 0.6sec, it is a dot
  }
  else if (signal_len > 150)
  {
    return '-';                          //if button press more than 0.6sec, it is a dash
  }
}
void convertor()
{
  static String letters[] = {".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....",
"..", ".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-",
".-.", "...", "-", "..-", "...-", ".--",  "-..-", "-.--", "--..", "E" };
  int i = 0;
  if (code == ".-.-.-")
  {
    Serial.print(".");        //for break
    LcdWriteCharacter(".");
  }
  else
  {
    while (letters[i] != "E")  //loop for comparing input code with letters array
    {
      if (letters[i] == code)
      {
        Serial.print(char('A' + i));
        LcdWriteCharacter(char('A' + i));
        break;
      }
      i++;
    }
    if (letters[i] == "E")
    {
      Serial.println("");  //if input code doesn't match any letter, error
      LcdWriteCharacter("");
    }
  }
  code = "";                             //reset code to blank string
}
```

6.1.2 Font Library

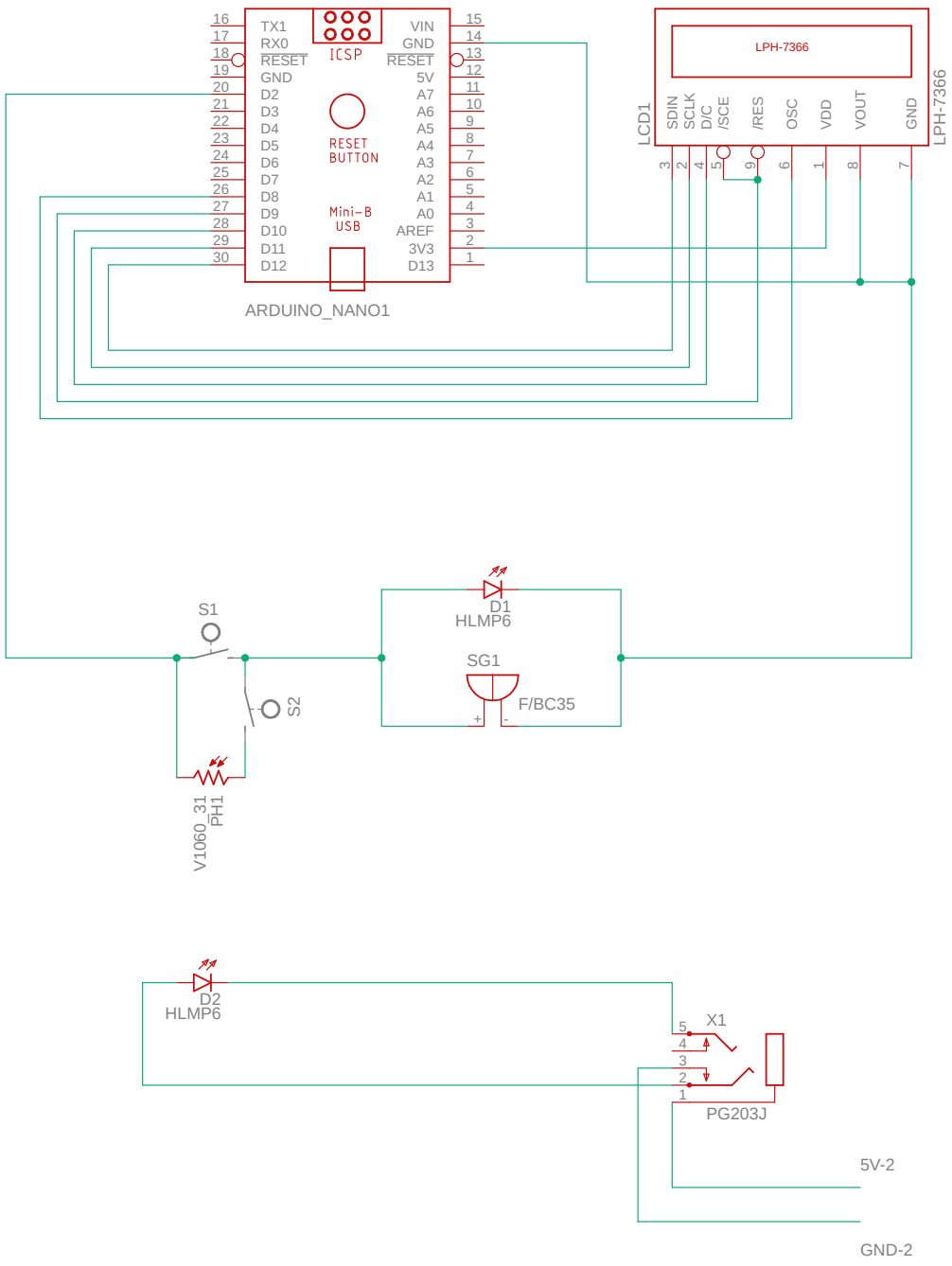
```
#include <Arduino.h>

static const byte ASCII[][5] =
{
 {0x00, 0x00, 0x00, 0x00, 0x00} // 20
,{0x00, 0x00, 0x5f, 0x00, 0x00} // 21 !
,{0x00, 0x07, 0x00, 0x07, 0x00} // 22 "
,{0x14, 0x7f, 0x14, 0x7f, 0x14} // 23 #
,{0x24, 0x2a, 0x7f, 0x2a, 0x12} // 24 $
,{0x23, 0x13, 0x08, 0x64, 0x62} // 25 %
,{0x36, 0x49, 0x55, 0x22, 0x50} // 26 &
,{0x00, 0x05, 0x03, 0x00, 0x00} // 27 '
,{0x00, 0x1c, 0x22, 0x41, 0x00} // 28 (
,{0x00, 0x41, 0x22, 0x1c, 0x00} // 29 )
,{0x14, 0x08, 0x3e, 0x08, 0x14} // 2a *
,{0x08, 0x08, 0x3e, 0x08, 0x08} // 2b +
,{0x00, 0x50, 0x30, 0x00, 0x00} // 2c ,
,{0x08, 0x08, 0x08, 0x08, 0x08} // 2d -
,{0x00, 0x60, 0x60, 0x00, 0x00} // 2e .
,{0x20, 0x10, 0x08, 0x04, 0x02} // 2f /
,{0x3e, 0x51, 0x49, 0x45, 0x3e} // 30 0
,{0x00, 0x42, 0x7f, 0x40, 0x00} // 31 1
,{0x42, 0x61, 0x51, 0x49, 0x46} // 32 2
,{0x21, 0x41, 0x45, 0x4b, 0x31} // 33 3
,{0x18, 0x14, 0x12, 0x7f, 0x10} // 34 4
,{0x27, 0x45, 0x45, 0x45, 0x39} // 35 5
,{0x3c, 0x4a, 0x49, 0x49, 0x30} // 36 6
,{0x01, 0x71, 0x09, 0x05, 0x03} // 37 7
,{0x36, 0x49, 0x49, 0x49, 0x36} // 38 8
,{0x06, 0x49, 0x49, 0x29, 0x1e} // 39 9
,{0x00, 0x36, 0x36, 0x00, 0x00} // 3a :
,{0x00, 0x56, 0x36, 0x00, 0x00} // 3b ;
,{0x08, 0x14, 0x22, 0x41, 0x00} // 3c <
,{0x14, 0x14, 0x14, 0x14, 0x14} // 3d =
,{0x00, 0x41, 0x22, 0x14, 0x08} // 3e >
,{0x02, 0x01, 0x51, 0x09, 0x06} // 3f ?
,{0x32, 0x49, 0x79, 0x41, 0x3e} // 40 @
,{0x7e, 0x11, 0x11, 0x11, 0x7e} // 41 A
,{0x7f, 0x49, 0x49, 0x49, 0x36} // 42 B
,{0x3e, 0x41, 0x41, 0x41, 0x22} // 43 C
,{0x7f, 0x41, 0x41, 0x22, 0x1c} // 44 D
,{0x7f, 0x49, 0x49, 0x49, 0x41} // 45 E
,{0x7f, 0x09, 0x09, 0x09, 0x01} // 46 F
,{0x3e, 0x41, 0x49, 0x49, 0x7a} // 47 G
```

```
,{0x7f, 0x08, 0x08, 0x08, 0x7f} // 48 H
,{0x00, 0x41, 0x7f, 0x41, 0x00} // 49 I
,{0x20, 0x40, 0x41, 0x3f, 0x01} // 4a J
,{0x7f, 0x08, 0x14, 0x22, 0x41} // 4b K
,{0x7f, 0x40, 0x40, 0x40, 0x40} // 4c L
,{0x7f, 0x02, 0x0c, 0x02, 0x7f} // 4d M
,{0x7f, 0x04, 0x08, 0x10, 0x7f} // 4e N
,{0x3e, 0x41, 0x41, 0x41, 0x3e} // 4f O
,{0x7f, 0x09, 0x09, 0x09, 0x06} // 50 P
,{0x3e, 0x41, 0x51, 0x21, 0x5e} // 51 Q
,{0x7f, 0x09, 0x19, 0x29, 0x46} // 52 R
,{0x46, 0x49, 0x49, 0x49, 0x31} // 53 S
,{0x01, 0x01, 0x7f, 0x01, 0x01} // 54 T
,{0x3f, 0x40, 0x40, 0x40, 0x3f} // 55 U
,{0x1f, 0x20, 0x40, 0x20, 0x1f} // 56 V
,{0x3f, 0x40, 0x38, 0x40, 0x3f} // 57 W
,{0x63, 0x14, 0x08, 0x14, 0x63} // 58 X
,{0x07, 0x08, 0x70, 0x08, 0x07} // 59 Y
,{0x61, 0x51, 0x49, 0x45, 0x43} // 5a Z
,{0x00, 0x7f, 0x41, 0x41, 0x00} // 5b [
,{0x02, 0x04, 0x08, 0x10, 0x20} // 5c \yen
,{0x00, 0x41, 0x41, 0x7f, 0x00} // 5d ]
,{0x04, 0x02, 0x01, 0x02, 0x04} // 5e ^
,{0x40, 0x40, 0x40, 0x40, 0x40} // 5f _
,{0x00, 0x01, 0x02, 0x04, 0x00} // 60 ‘
,{0x20, 0x54, 0x54, 0x54, 0x78} // 61 a
,{0x7f, 0x48, 0x44, 0x44, 0x38} // 62 b
,{0x38, 0x44, 0x44, 0x44, 0x20} // 63 c
,{0x38, 0x44, 0x44, 0x48, 0x7f} // 64 d
,{0x38, 0x54, 0x54, 0x54, 0x18} // 65 e
,{0x08, 0x7e, 0x09, 0x01, 0x02} // 66 f
,{0x0c, 0x52, 0x52, 0x52, 0x3e} // 67 g
,{0x7f, 0x08, 0x04, 0x04, 0x78} // 68 h
,{0x00, 0x44, 0x7d, 0x40, 0x00} // 69 i
,{0x20, 0x40, 0x44, 0x3d, 0x00} // 6a j
,{0x7f, 0x10, 0x28, 0x44, 0x00} // 6b k
,{0x00, 0x41, 0x7f, 0x40, 0x00} // 6c l
,{0x7c, 0x04, 0x18, 0x04, 0x78} // 6d m
,{0x7c, 0x08, 0x04, 0x04, 0x78} // 6e n
,{0x38, 0x44, 0x44, 0x44, 0x38} // 6f o
,{0x7c, 0x14, 0x14, 0x14, 0x08} // 70 p
,{0x08, 0x14, 0x14, 0x18, 0x7c} // 71 q
,{0x7c, 0x08, 0x04, 0x04, 0x08} // 72 r
,{0x48, 0x54, 0x54, 0x54, 0x20} // 73 s
,{0x04, 0x3f, 0x44, 0x40, 0x20} // 74 t
,{0x3c, 0x40, 0x40, 0x20, 0x7c} // 75 u
```

```
,{0x1c, 0x20, 0x40, 0x20, 0x1c} // 76 v
,{0x3c, 0x40, 0x30, 0x40, 0x3c} // 77 w
,{0x44, 0x28, 0x10, 0x28, 0x44} // 78 x
,{0x0c, 0x50, 0x50, 0x50, 0x3c} // 79 y
,{0x44, 0x64, 0x54, 0x4c, 0x44} // 7a z
,{0x00, 0x08, 0x36, 0x41, 0x00} // 7b {
,{0x00, 0x00, 0x7f, 0x00, 0x00} // 7c |
,{0x00, 0x41, 0x36, 0x08, 0x00} // 7d }
,{0x10, 0x08, 0x08, 0x10, 0x08} // 7e <
,{0x78, 0x46, 0x41, 0x46, 0x78} // 7f >
};
```

6.2 Schematic

ESAT CANBERK ÖZÇELİK	NAME: BAU EEE1002 PROJECT MORSE INTERPRETER	DATE: 28.05.2001	SHEET: 1/1
		Devices See (2.2)	



6.3 3D Exterior Design

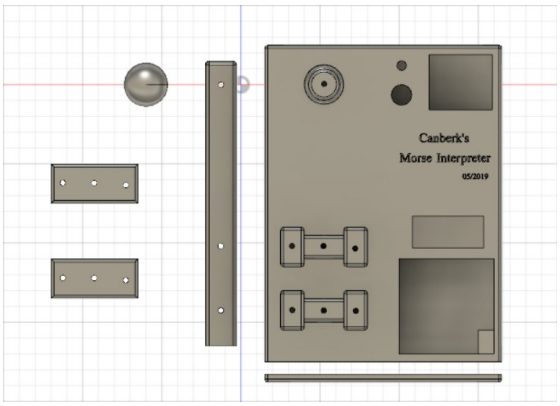


Figure 8: Top View

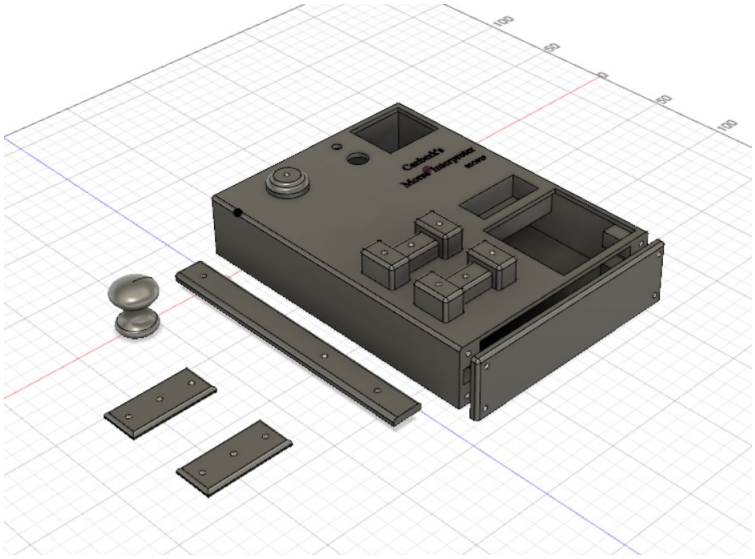


Figure 9: Corner View

6.4 International Morse Code Table

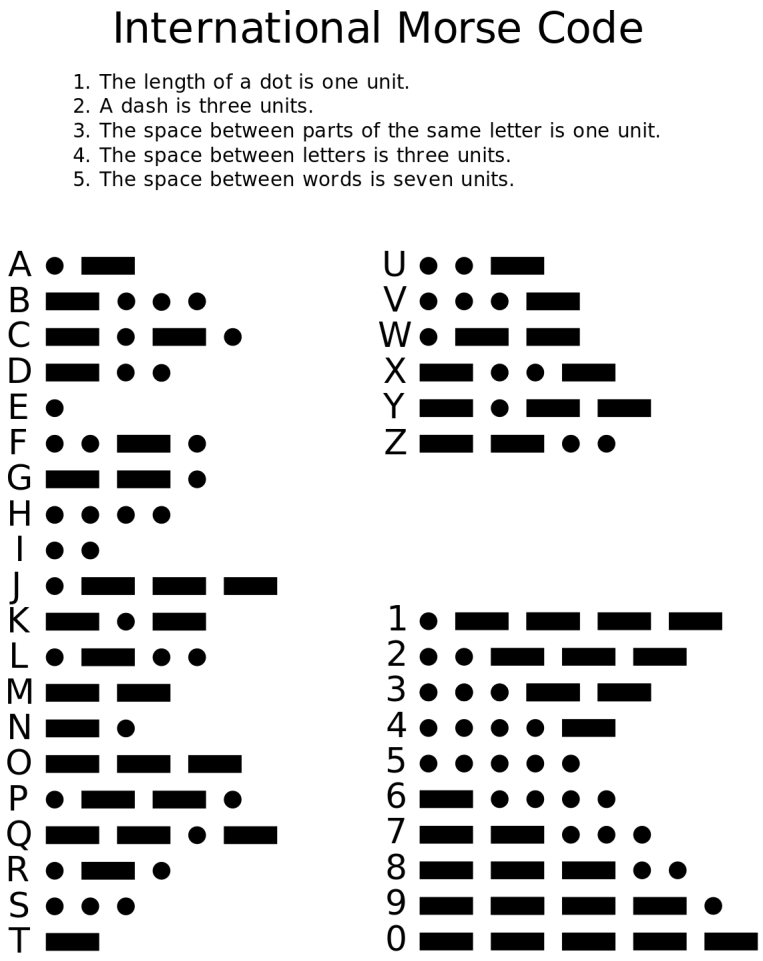


Figure 10

Esat Canberk OZCELIK
Student ID:1805100
Address: Zümrütevler mah. Tekcan sok.
23/1 D:1 Maltepe/İstanbul-TURKEY
e-mail: esatcanberk.ozcelik@bahcesehir.edu.tr
Phone: +905078467183