

İSİM-SOYİSİM: Elif İrem KAYA

ÖĞRENCİ NO.: 02225076056 (İ.Ö.)

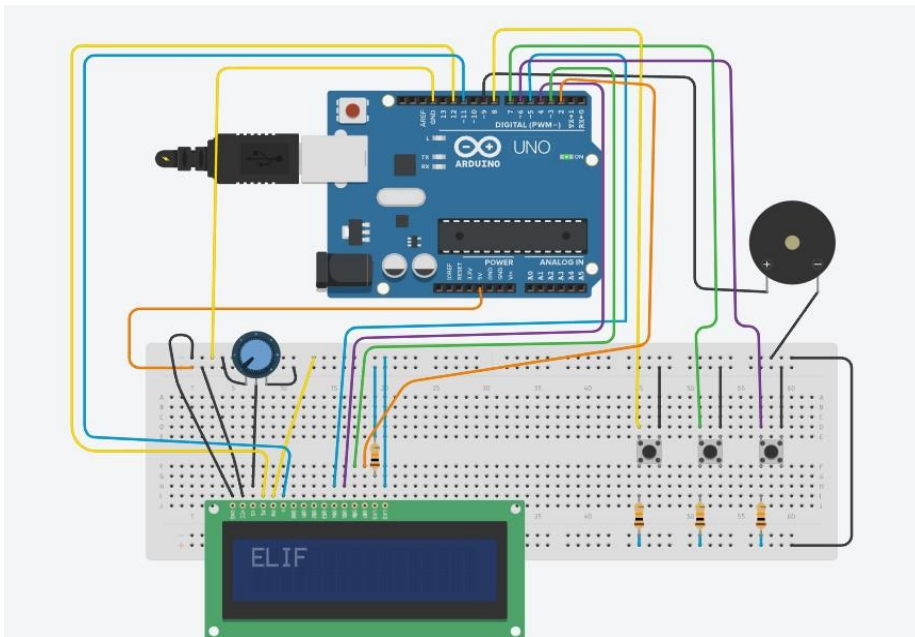
1)Proje Adı: MORSEKOD: ACİL DURUM EĞİTİM ARACI

2)Proje Amacı: Çocuklara morse kodunu öğretmek iletişim becerilerini geliştirmeyi ve afet bilincini artırmayı hedefler. Arduino tabanlı bu cihaz, morse koduyla temel iletişim becerilerini pratik yapma imkanı sunar ve acil durum senaryolarında ihtiyaçları olacak ifadeleri kodlayarak afet öncesi hazırlık yapar. Bu sayede çocuklar, hem eğlenirken hem de afet durumlarında etkili iletişim kurma yetenekleri kazanırlar.

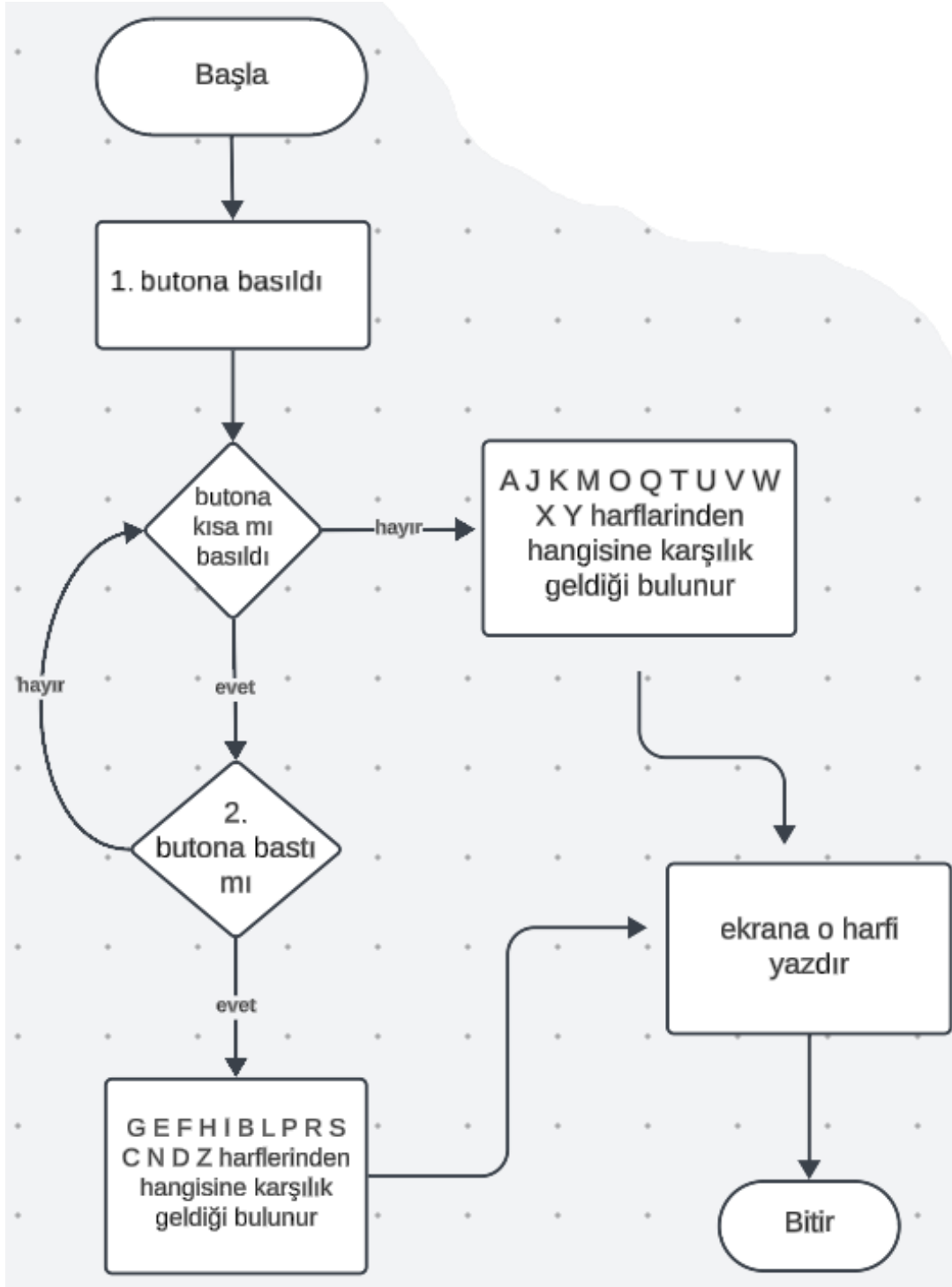
3)Projede Kullanılan Donanımsal Bileşenler ve Açıklamaları:

- * Arduino UNO : Mikrodenetleyici
- * Push buton : Mors kodun algılanmasını, ekrana yazılmasını ve sese dönüşümünde kullanılacak
- * Lcd ekran: Mors kodla ne yazdığımızı görmemize yardım eder
- * Buzzer: Algılanan mors kodun sese dönüşümünü sağlar
- * Breadboard: Devre elemanlarının bir araya getirilmesini sağlar
- * Potansiyometre: Ekranın ışık şiddetini ayarlamamızı sağlar
- * Direnç: Ekrana ve butonlara fazla akım gelmesini önler

4) Devre Çizimi (Tinkercad) :



5) Yazılım Akış Şeması ve Kodları:



```

#include <LiquidCrystal.h>

#define CARKABUTTONPIN 12
#define TECKABUTTONPIN 8
#define ODDELBUTTONPIN 10
#define DISPLAY_NUMOFCOLUMNS 16

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int carkaButtonState = 0; int
carkaButtonLastState = 0; int
teckaButtonState = 0; int
teckaButtonLastState = 0; int
oddelButtonState = 0; int
oddelButtonLastState = 0;
String tonesBuffer;
String text;
String expectedText;

String symbolsAlphabet[][2] =
{
    { ".-", "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" }
    };

    char getToneFromButtonStates()
    {
        if (!carkaButtonState &&
carkaButtonLastState)          return '-';          if
(!teckaButtonState && teckaButtonLastState)
return '.';          if (!oddelButtonState &&
oddelButtonLastState)          return ' ';
        return
(char)0;

    }

    char
getSymbolFromBuffer()
    {
        if (tonesBuffer ==
"")          return ' ';
        for (int i = 0; i < sizeof symbolsAlphabet / sizeof
symbolsAlphabet[0]; i++)          if (tonesBuffer == symbolsAlphabet[i][0])
return symbolsAlphabet[i][1][0];
        return
(char)0;
    }

    void
extractActionFromTonesBuffer()
    {
        if (tonesBuffer == ".....")
text.remove(text.length() - 1, 1);          if
(tonesBuffer == "-----")          text =
"";
    }

    void setup() {
        lcd.begin(DISPLAY_NUMOFCOLUMNS, 2); // LCD başlatılır
        lcd.print("Morseovkovnitko");          lcd.setCursor(0, 1);
        lcd.print("6x.Smaze1 6x-Vse");
        pinMode(CARKABUTTONPIN,
INPUT);          pinMode(TECKABUTTONPIN,
INPUT);          pinMode(ODDELBUTTONPIN,
INPUT);
    }

    void loop() {          carkaButtonState =
digitalRead(CARKABUTTONPIN);          teckaButtonState =
digitalRead(TECKABUTTONPIN);          odde1ButtonState =
digitalRead(ODDELBUTTONPIN);
        char tone =
getToneFromButtonStates();
        if (tone !=
(char)0)
        {
            if (tone
== ' ')

```

```

        {
            char symbol =
getSymbolFromBuffer();
            if (symbol !=
(char)0)
            {
                text += symbol;
if (text.length() > DISPLAY_NUMOFCOLUMNS)
            {
                text
= (String)symbol;
            }
        }
        else
        {
            extractActionFromTonesBuffer();
            tonesBuffer = "";
        }
        else { tonesBuffer +=
tone;
            if (tonesBuffer.length() >
DISPLAY_NUMOFCOLUMNS)
            {
                tonesBuffer =
(String)tone;
            }
        }
        lcd.clear();
        lcd.print(text);
        lcd.setCursor(0, 1);
        lcd.print(tonesBuffer);

    }
    carkaButtonLastState =
carkaButtonState;
    teckaButtonLastState =
teckaButtonState;
    oddelButtonLastState =
oddelButtonState;

}

```