

Einfacher Morsetrainer für Arduino

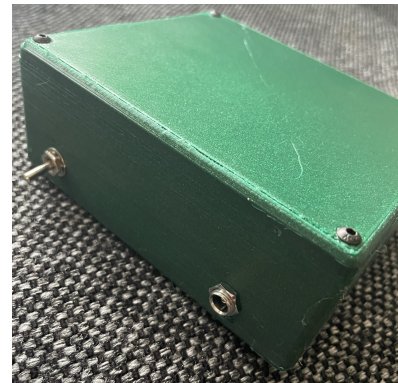
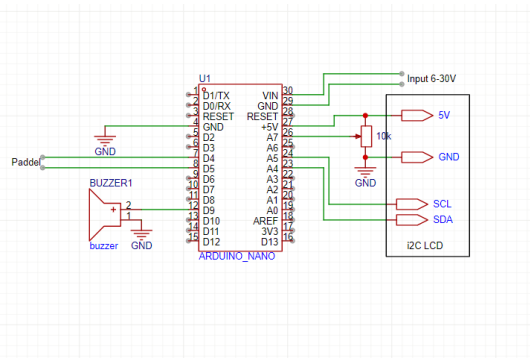
Es handelt sich um ein kleines Projekt für einen Morsetrainer für Paddel. Ich habe ein Gerät gesucht auf dem Netz, aber die meisten sind entweder grosse Projekte wie «morserino» oder aber nicht geeignet für Paddles, sondern nur für einfache Keyer. Auf Youtube habe ich ein kleines Projekt für den Arduino Nano gefunden, den Trainer aufgebaut und er hat so gut funktioniert, dass ich beschlossen habe, ein kleines Gehäuse zu entwerfen.

Der Code ist sehr einfach gehalten und kann sehr einfach erweitert werden um Sonderzeichen, Memory Funktionen etc.

Auf der Vorderseite befinden sich das 16x2 LCD, ein 10k Potentiometer sowie die 3.5mm Stereo-Klinkenbuchse für das Paddle. Der Mittelabgriff des Potis geht zum Eingang A7, die äusseren Pins an 5V und Masse. Das LCD ist ein 16x2 I2C Modell und benötigt lediglich 4 Verbindungen vom Arduino.

Auf der Rückseite habe ich einen Jack, um die Akkus zu laden, sowie den Ein/Ausschalter. Als Akku habe ich 2 Stück 18650 benutzt, in Serie geschaltet. Der Stromverbrauch liegt bei 45mA

Schema:



Stückliste:

- 1 Arduino Nano(gekauft bei Bastelgarage in Lyss)
- 1 i2C LCD 16x2 (gekauft bei Bastelgarage in Lyss)
- 1 Potentiometer 10k (Aliexpress oder Reichelt)
- 1 Piezosummer (Aliexpress oder Reichelt)
- 1 3.5mm Stereo Klinke Einbaubuchse (Aliexpress oder Reichelt)
- 1 DC Jack mit 2.5mm Pin (Aliexpress oder Reichelt)
- 1 Kippschalter für ein/aus (Aliexpress oder Reichelt)

<https://www.youtube.com/watch?v=Qx3kiWBnbvQ>

Die .stl Dateien zum Drucken sowie das Fusion-360 File mit der Zeichnung (für Änderungen) finden sich hier:

<https://www.thingiverse.com/thing:6818707>

Der Code für den Morsetrainer befindet sich hier

<https://github.com/arnongold/Arduino/blob/main/MorseKeyer/main/main.ino>

Anhang: Inhalt des .ino Files für den Arduino

/ Verwendete Pins: A7 für Poti, D4 und D5 für das Paddel *, D9 für den Buzzer plus A4(SDA) und A5(SCL)*/*

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,20,2);
```

```
int delay_ms = 5;
```

```
int buzzer_pin = 9;
```

```
int analogValue;
```

```
int previousValue;
```

```
int WPM;
```

```
int unit;
```

```
const int dit = 1;
```

```
const int da = 2;
```

```
bool transmittingDit = false;
```

```
bool transmittingDa = false;
```

```
bool transmittingPause = false;
```

```
unsigned long transmissionStartedAt;
```

```
unsigned long transmissionEndedAt;
```

```
unsigned long lastClickAt;
```

```
int transmitNext;
```

```
String letter = "";
```

```
String line = "";
```

```
bool spacePrinted = false;
```

```
void setup() {
```

```
    pinMode(4,INPUT_PULLUP);
```

```
    pinMode(5,INPUT_PULLUP);
```

```
    pinMode(7,INPUT_PULLUP);
```

```
    pinMode(buzzer_pin, OUTPUT);
```

```
    Serial.begin(9600);
```

```
    lcd.init();
```

```
    lcd.backlight();
```

```
    lcd.setCursor(1,0);
```

```
    lcd.print("Morse Keyer");
```

```
    lcd.setCursor(3,1);
```

```
    lcd.print("by 4X4RD"); /* Hier kannst Du Dein Callsign eingeben */
```

```
    delay(2000);
```

```
    lcd.clear();
```

```
    lastClickAt = millis();
```

```
    letter = "";
```

```
    transmitNext = 0;
```

```
}
```

```
void loop() {
```

```

    getWPM();
    readKey();
    TransmitMorse();

    checkClear();
}

void TransmitMorse()
{
    if (transmittingPause) {
        if (millis() > transmissionStartedAt + unit) {
            transmittingPause = false;
            transmissionEndedAt = millis();
        }
    } else if (transmittingDit){
        if (millis() > transmissionStartedAt + unit)
        {
            analogWrite(buzzer_pin, 0);
            transmittingDit = false;
            transmittingPause = true;
            transmissionStartedAt = millis();
            transmitNext = 0;
        }
    } else if (transmittingDa) {
        if (millis() > transmissionStartedAt + (unit * 3))
        {
            analogWrite(buzzer_pin, 0);
            transmittingDa = false;
            transmittingPause = true;
            transmissionStartedAt = millis();
            transmitNext = 0;
        }
    } else {
        if (transmitNext != 0) {
            switch (transmitNext){
                case dit:
                    analogWrite(buzzer_pin, 50);
                    transmittingDit = true;
                    transmissionStartedAt = millis();
                    break;
                case da:
                    analogWrite(buzzer_pin, 50);
                    transmittingDa = true;
                    transmissionStartedAt = millis();
                    break;
            }
        }
    }
}
}

```

```

void getWPM()
{
  analogValue = analogRead(A7);
  WPM = 5 + (analogValue / 33);
  if (previousValue != WPM)
  {
    lcd.setCursor(0,1);
    lcd.print("WPM:");
    lcd.setCursor(4,1);
    lcd.print("  ");
    lcd.setCursor(4,1);
    lcd.print(WPM);
    previousValue = WPM;
    unit = 1.2/WPM*1000;
    Serial.print(analogValue);
    Serial.print(" Unit." );
    Serial.print(unit);
    Serial.print(" WPM:" );
    Serial.print(WPM);
    Serial.print("\n");
  }
}

void readKey()
{
  if (transmitNext != 0)
    return;
  if (digitalRead(4) == LOW)
  {
    if (millis() > lastClickAt + unit)
    {
      transmitNext = dit;
      lastClickAt = millis();
      letter = letter + dit;
    }
  }

  if (digitalRead(5) == LOW)
  {
    if (millis() > lastClickAt + unit)
    {
      transmitNext = da;
      lastClickAt = millis();
      letter = letter + da;
    }
  }

  if ((millis() > lastClickAt + (unit * 2)) && letter != "") {

```

```

    decodeLetter();
}

if (millis() > max(transmissionEndedAt,transmissionStartedAt) + (unit * 6)) {
    printSpace();
}
}

```

```

void decodeLetter()
{
    Serial.print(letter + "\n");

    switch(letter.length()){
    case 1:
        if (letter == "1") printLetter("E");
        else printLetter("T");
        break;
    case 2:
        if (letter == "12") printLetter("A");
        else if (letter == "21") printLetter("N");
        else if (letter == "11") printLetter("I");
        else if (letter == "22") printLetter("M");
        break;
    case 3:
        if (letter == "111") printLetter("S");
        else if (letter == "112") printLetter("U");
        else if (letter == "121") printLetter("R");
        else if (letter == "122") printLetter("W");
        else if (letter == "211") printLetter("D");
        else if (letter == "212") printLetter("K");
        else if (letter == "221") printLetter("G");
        else if (letter == "222") printLetter("O");
        break;
    case 4:
        if (letter == "1111") printLetter("H");
        else if (letter == "1112") printLetter("V");
        else if (letter == "1121") printLetter("F");
        else if (letter == "1211") printLetter("L");
        else if (letter == "1221") printLetter("P");
        else if (letter == "1222") printLetter("J");
        else if (letter == "2111") printLetter("B");
        else if (letter == "2112") printLetter("X");
        else if (letter == "2121") printLetter("C");
        else if (letter == "2122") printLetter("Y");
        else if (letter == "2211") printLetter("Z");
        else if (letter == "2212") printLetter("Q");
        else if (letter == "2221") printLetter(".");
        else if (letter == "2222") printLetter("-");
        break;
    }
}

```

```

case 5:
  if (letter == "12222") printLetter("1");
  else if (letter == "11222") printLetter("2");
  else if (letter == "11122") printLetter("3");
  else if (letter == "11112") printLetter("4");
  else if (letter == "11111") printLetter("5");
  else if (letter == "21111") printLetter("6");
  else if (letter == "22111") printLetter("7");
  else if (letter == "22211") printLetter("8");
  else if (letter == "22221") printLetter("9");
  else if (letter == "22222") printLetter("0");
  break;
}

letter = "";
spacePrinted = false;
}

void printSpace()
{
  if (!spacePrinted) {
    printLetter(" ");
    spacePrinted = true;
  }
}

void checkClear()
{
  if (digitalRead(7) == LOW)
    clearLine();
}

void clearLine()
{
  line = "";
  lcd.setCursor(0,0);
  lcd.print(" ");
}

void printLetter(String newLetter)
{
  line = line + newLetter;
  if (line.length() > 16)
    line = line.substring(1);
  lcd.setCursor(0,0);
  lcd.print(line);
}

```