Expérience n°3:

Buts de la manipulation:

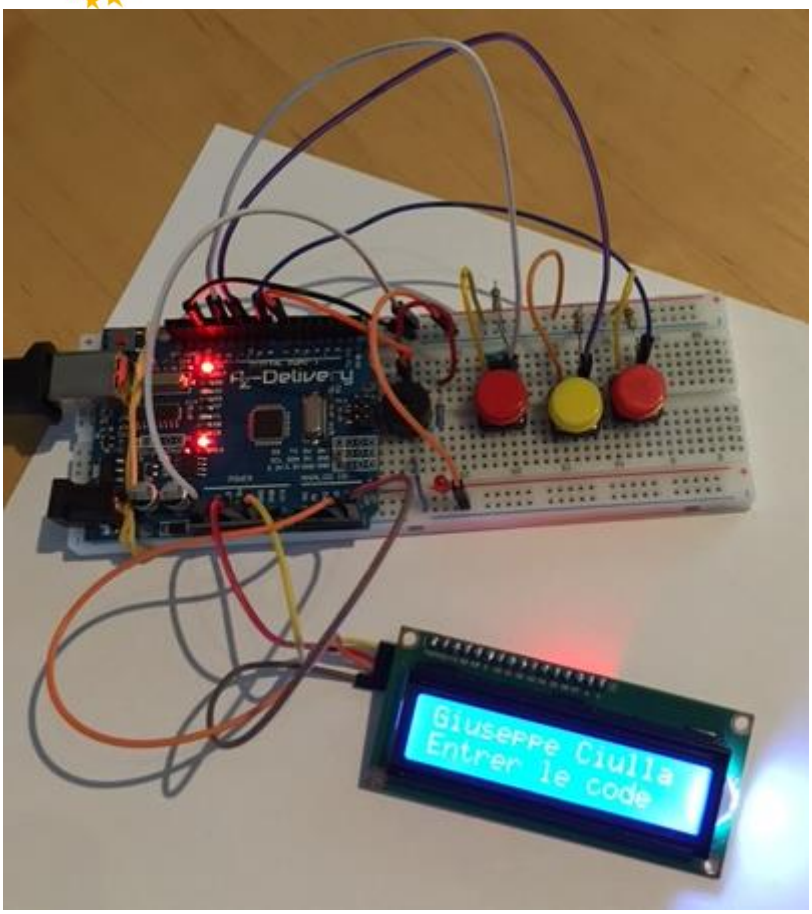
Décodage d'un code en Morse à partir d'un montage Arduino

Cet appareil lit le code Morse saisi manuellement à partir de trois touches de signal et le traduit en texte brut et affiche le code Morse que vous saisissez sur un écran LCD.
Le décodeur s'adapte automatiquement à votre vitesse de frappe.

Matériels:

1x Arduino UNO
1x Buzzer
1x LCD avec connecteur I2C et 2 lignes de texte
3x Interrupteurs , une planche à pain et des fils.
1x Diode Led
1x Résistances 220 Ω
3x Résistances 1k Ω

Schéma du montage avec l'arduino et la plaquette d'essai:



Mode opératoire:

On introduit préalablement le code programme du décodeur Morse. On connecte le buzzer, la Led en série avec une résistance de 220 Ω entre GND et la broche 7 de l'Arduino et la clé Morse (j'utilise trois interrupteurs tactiles et trois résistances de 1kΩ) entre GND , le +5V et les broches 12 , 8 et 10.
Outre les broches GND et +5V, l'écran LCD dispose d'une connexion SDA et SCL que vous connectez aux broches portant les mêmes noms ou A4 et A5 sur l'Arduino.
Le message apparaît et on introduit les caractères du code Morse avec les boutons poussoirs. Celui de gauche pour le (-) et celui de droite pour le (.) un signal buzzer est produit pour chaque pression. Chaque lettre du code doit être activé par le bouton poussoir du milieu.

Conclusion : Ce système permet de décoder un code morse à partir de trois boutons poussoirs et un signal sonore est produit pour chaque pression exercée. L'écran LCD visualise le décodage du code Morse.

Code programme:

```
// PGM Morse

#include <Wire.h> //  I2C
#include <LiquidCrystal_I2C.h>
```


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


int BUZZERpin = 7; //  BUZZER pin
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"}
};

LiquidCrystal_I2C lcd(0x27, DISPLAY_NUMOFCOLUMNS, 2);
char getToneFromButtonStates()
{
        //(reviendra si le bouton est relâché)
        //(tedy, l'état actuel est 0, le précédent est 1)

        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 == "------")// 6caractères
                text = "";
```


```
}


void setup() {

        lcd.init();
        lcd.backlight();
        lcd.print("Giuseppe Ciulla");
        lcd.setCursor(0, 1);
        lcd.print("Entrer le code");

pinMode(BUZZERpin,OUTPUT);//initialize the BUZZER pin as output

        pinMode(CARKABUTTONPIN, INPUT);



        pinMode(TECKABUTTONPIN, INPUT);



        pinMode(ODDELBUTTONPIN, INPUT);
}

void loop() {


        carkaButtonState = digitalRead(CARKABUTTONPIN);
 if(digitalRead(CARKABUTTONPIN) ==HIGH )
{
digitalWrite(BUZZERpin,HIGH);//turn on the led
}
else
{
digitalWrite(BUZZERpin,LOW);//turn off the led
}

  teckaButtonState = digitalRead(TECKABUTTONPIN);
if(digitalRead(TECKABUTTONPIN) ==HIGH )
{
digitalWrite(BUZZERpin,HIGH);//turn on the led
}



        oddelButtonState = 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;
                        }
                }

                //(L'écriture sur l'écran s'effectue uniquement lorsqu'un bouton a été effacé)

                lcd.clear();
                lcd.print(text);
                lcd.setCursor(0, 1);
                lcd.print(tonesBuffer);

        }

        //(Le statut précédent est mis à jour)
        carkaButtonLastState = carkaButtonState;
        teckaButtonLastState = teckaButtonState;
        oddelButtonLastState = oddelButtonState;

}
```