

RAPPORT DE MINI-PROJET SYSTÈMES À MICROPROCESSEURS

Par :

Roua Ben Emna et Mouna Chtourou

Réalisation d'un détecteur de mensonge



Année universitaire : 2022/2023

TABLE DES MATIÈRES

Introduction générale	1
1 Introduction	2
1 C'est quoi un détecteur de mensonge	3
2 Idée de notre projet	3
3 Fonctionnalités	3
4 Les avantages de détecteur de mensonge	4
2 Environnement Logicielle	6
1 Arduino IDE	7
2 Proteus 8 professional	7
3 Environnement matériel	8
1 Carte Adruino UNO	9
2 Capteur de rythme cardiaque	9
3 Plaque d'essai 170 points	10
4 Bouton poussoir	10
5 Résistance	11
6 Fil De Connexion	12
7 Afficheur LCD I2C 16*2	12
4 SIMULATION ET CÂBLAGE	13
1 Simulation par l'outil Proteus 8 Professional	14

2	Câblage des matériels	15
5	Réalisation	16
1	Code du projet	17
1.1	Importation du bibliothèques et creation du variable	17
1.2	Initialisation de la communication série, configuration des broches d'entrée et de l'écran LCD	18
1.3	Fonctionnement de détecteur de mensonge	19
2	Assemblage des matériels et construction	23
	Conclusion générale	25
	Références bibliographiques	26

TABLE DES FIGURES

2.1	Arduino IDE	7
2.2	Proteus	7
3.1	Carte Adruino UNO	9
3.2	Pulse Sensor	10
3.3	Plaque d'essai	10
3.4	Bouton poussoir	11
3.5	Résistance	11
3.6	Fils De Connexion	12
3.7	Afficheur LCD I2C 16*2	12
4.1	Simulation Proteus	14
4.2	Câblage	15
5.1	Code - Importation et creation	17
5.2	Code - Setup	18
5.3	Code - Loop	19
5.4	Lecture du capteur analogique et du l'état du bouton	19
5.5	Affichage sur l'écran LCD	20
5.6	Détection de l'appui sur le bouton	20
5.7	Calcul de la fréquence cardiaque	21
5.8	Vérifier l'honnêteté de la personne	22

5.9 Matériels	23
5.10 Matériels détecteur de mensonge	23
5.11 Aff-detector	24
5.12 Aff-liar	24
5.13 Aff-truth	24
5.14 détecteur de mensonge	24

INTRODUCTION GÉNÉRALE

CE mini projet intitulé Réalisation d'un détecteur de mensonge s'inscrit dans le cadre de la matière Système Microprocesseur, élaboré par : Roua Ben emna et Mouna Chtourou, étudiantes en 1ère ING 1 à l'Institut Supérieur d'Informatique à Ariana.

Durant ce rapport, nous allons élaborer une présentation totale du projet comportant la description, les fonctionnalités de notre mini projet et les avantages de détecteur de mensonge. Ensuite, nous spécifions les matériels utilisés et le langage que nous avons opté. Puis, nous aborderons la partie simulation et câblage.

Arrivant à la partie réalisation, où nous exposons des captures sur le code et quelques photos réelles de notre mini projet.

Finalement nous clôturons par une conclusion.

CHAPITRE 1

INTRODUCTION

1	C'est quoi un détecteur de monsange	3
2	Idée de notre projet	3
3	Fonctionnalités	3
4	Les avantages de détecteur de monsange	4

1 C'est quoi un détecteur de mensonge

Un détecteur de mensonge est un outil utilisé pour déterminer si une personne dit la vérité ou non. Il existe différentes méthodes pour détecter les mensonges, dont certaines reposent sur l'analyse du langage corporel, des réactions physiologiques ou des réponses verbales.

2 Idée de notre projet

L'objectif de notre projet est de concevoir et de réaliser un détecteur de mensonge basé sur la détection des battements de cœur. L'idée est de développer un outil capable de détecter les mensonges en mesurant les variations de la fréquence cardiaque qui peuvent être provoquées par le stress ou l'anxiété.

3 Fonctionnalités

Voici les étapes de fonctionnement de notre détecteur de mensonge basé sur la fréquence cardiaque :

- Placer le capteur Pulse Sensor pour mesurer la fréquence cardiaque.
- Attendre que les données soient collectées.
- Analyser les variations de la fréquence cardiaque pour détecter le stress ou l'anxiété.
- Poser une question à la personne et attendre sa réponse.
- Appuyer sur un bouton pour lancer l'analyse de la fréquence cardiaque pendant la réponse.
- Analyser la fréquence cardiaque après chaque réponse pour détecter les variations significatives.
- Afficher le résultat (ment ou nn) sur l'écran LCD.
- Répéter les étapes 4 à 7 pour chaque question posée.

4 Les avantages de détecteur de mensonge



Aide à résoudre les enquêtes criminelles : les détecteurs de mensonge peuvent aider à déterminer si une personne impliquée dans une enquête criminelle est coupable ou innocente. Cela peut être particulièrement utile lorsque les preuves sont insuffisantes ou lorsque les témoignages des suspects sont contradictoires.



Contribue à une prise de décision équitable : dans les litiges judiciaires, les détecteurs de mensonge peuvent aider à établir la vérité et à garantir une prise de décision équitable.



Améliore la sécurité : les détecteurs de mensonge peuvent être utilisés dans les enquêtes de sécurité pour évaluer les risques potentiels et détecter les activités suspectes.



Facilite la recherche scientifique : les détecteurs de mensonge peuvent aider les chercheurs à collecter des données précises sur les émotions et les comportements humains, ce qui peut être utile dans divers domaines tels que la psychologie, la neuroscience et la communication.



Renforce la confiance dans les relations interpersonnelles : l'utilisation de détecteurs de mensonge peut aider à renforcer la confiance dans les relations interpersonnelles en détectant les mensonges et en favorisant la communication honnête.

CHAPITRE 2

ENVIRONNEMENT LOGICIELLE

1	Arduino IDE	7
2	Proteus 8 professional	7

Introduction

Au cours de cette section, nous présentons les outils et logiciels utilisés tout au long de notre projet.

1 Arduino IDE

Le logiciel libre Arduino (IDE) permet d'écrire facilement du code et de le télécharger sur la carte. Ce logiciel peut être utilisé avec n'importe quelle carte Arduino.[1]



FIGURE 2.1 : Arduino IDE

2 Proteus 8 professional

La CAO électronique Proteus est une suite logicielle, éditée par la société Labcenter Electronics et revendue en France exclusivement par Multipower.[2]



FIGURE 2.2 : Proteus

CHAPITRE 3

ENVIRONNEMENT MATÉRIEL

1	Carte Adruino UNO	9
2	Capteur de rythme cardiaque	9
3	Plaque d'essai 170 points	10
4	Bouton poussoir	10
5	Résistance	11
6	Fil De Connexion	12
7	Afficheur LCD I2C 16*2	12

Introduction

Au cours de cette section, nous présentons les matériels utilisés tout au long de notre projet.

1 Carte Arduino UNO

L'Arduino UNO est une carte microcontrôleur basée sur le microcontrôleur ATmega328P. C'est une plateforme matérielle open-source qui vous permet de créer des projets électroniques interactifs en la connectant à un ordinateur, une source d'alimentation et divers composants électroniques tels que des capteurs, des actionneurs et des afficheurs.

La carte dispose de 14 broches d'entrée/sortie numériques, 6 entrées analogiques, un cristal quartz de 16 MHz, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation. Elle dispose également d'une large gamme de bibliothèques et d'outils logiciels qui facilitent la programmation et le contrôle de la carte à l'aide de l'environnement de développement intégré Arduino (IDE). L'UNO est l'une des cartes Arduino les plus populaires et largement utilisées, connue pour sa simplicité et sa polyvalence. [3]

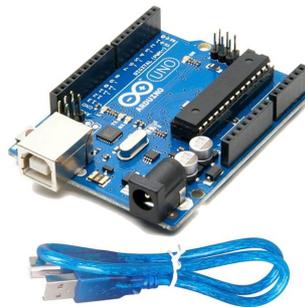


FIGURE 3.1 : Carte Arduino UNO

2 Capteur de rythme cardiaque

Le capteur de rythme cardiaque Pulse Sensor Amped est un capteur électronique qui mesure la fréquence cardiaque en détectant les variations de la lumière transmise à travers un tissu vasculaire, tel que le bout d'un doigt. Il est facile à utiliser et à intégrer dans des projets électroniques et est compatible avec les cartes Arduino, Raspberry Pi, et d'autres plateformes électroniques courantes.[4]



FIGURE 3.2 : Pulse Sensor

3 Plaque d'essai 170 points

Une plaque d'essai 170 points est une plaque de circuit imprimé qui permet de prototyper des circuits électroniques à petite échelle en utilisant des fils de raccordement ou des composants enfichables. Elle est économique et pratique pour les projets électroniques de bricolage.

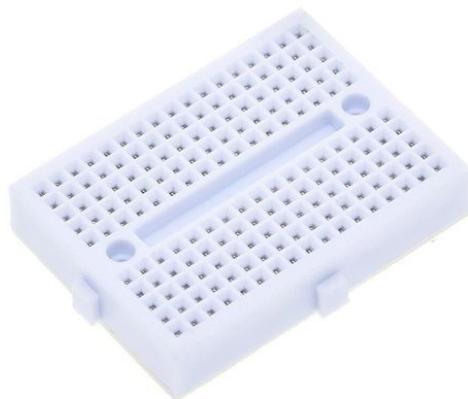


FIGURE 3.3 : Plaque d'essai

4 Bouton poussoir

Un bouton poussoir est un interrupteur momentané qui ferme ou ouvre un circuit électrique temporairement en appuyant sur un bouton. Il est utilisé pour activer ou désactiver une fonction

ou pour lancer une action précise dans des projets électroniques.



FIGURE 3.4 : Bouton poussoir

5 Résistance

La résistance de 1 kohm est une valeur de résistance standard qui est couramment utilisée pour protéger un bouton-poussoir. Lorsqu'un bouton-poussoir est enfoncé, le circuit électrique est fermé, ce qui permet au courant de circuler. Cependant, cela peut entraîner une surcharge de courant qui peut endommager les broches d'entrée de l'Arduino.

En plaçant une résistance de 1 kohm en série avec le bouton-poussoir, le courant est limité, ce qui réduit l'usure du bouton-poussoir et prolonge sa durée de vie. Cette résistance agit comme une barrière pour limiter le courant qui circule à travers le bouton-poussoir, ce qui protège les broches d'entrée de l'Arduino contre une surcharge de courant.



FIGURE 3.5 : Résistance

6 Fil De Connexion

Un fil de connexion est un fil électrique utilisé pour connecter des composants électroniques entre eux ou avec une source d'alimentation. Il est souvent utilisé pour relier les composants électroniques à une plaque d'essai, à une carte de circuit imprimé ou à une breadboard, et est disponible dans différentes longueurs, couleurs et calibres.



FIGURE 3.6 : Fils De Connexion

7 Afficheur LCD I2C 16*2

Un afficheur LCD I2C 16x2 est un type d'afficheur qui utilise le protocole de communication I2C et peut afficher jusqu'à 16 caractères sur deux lignes de texte. Il ne nécessite que deux fils pour la communication et est facile à intégrer dans les projets électroniques.[5]



FIGURE 3.7 : Afficheur LCD I2C 16*2

CHAPITRE 4

SIMULATION ET CÂBLAGE

1	Simulation par l'outil Proteus 8 Professional	14
2	Câblage des matériels	15

Introduction

Durant cette section nous allons illustrer la simulation faite par l'outil Proteus 8 Professional et l'Environnement de développement Arduino définis dans la section précédente et le câblage des matériels .

1 Simulation par l'outil Proteus 8 Professional

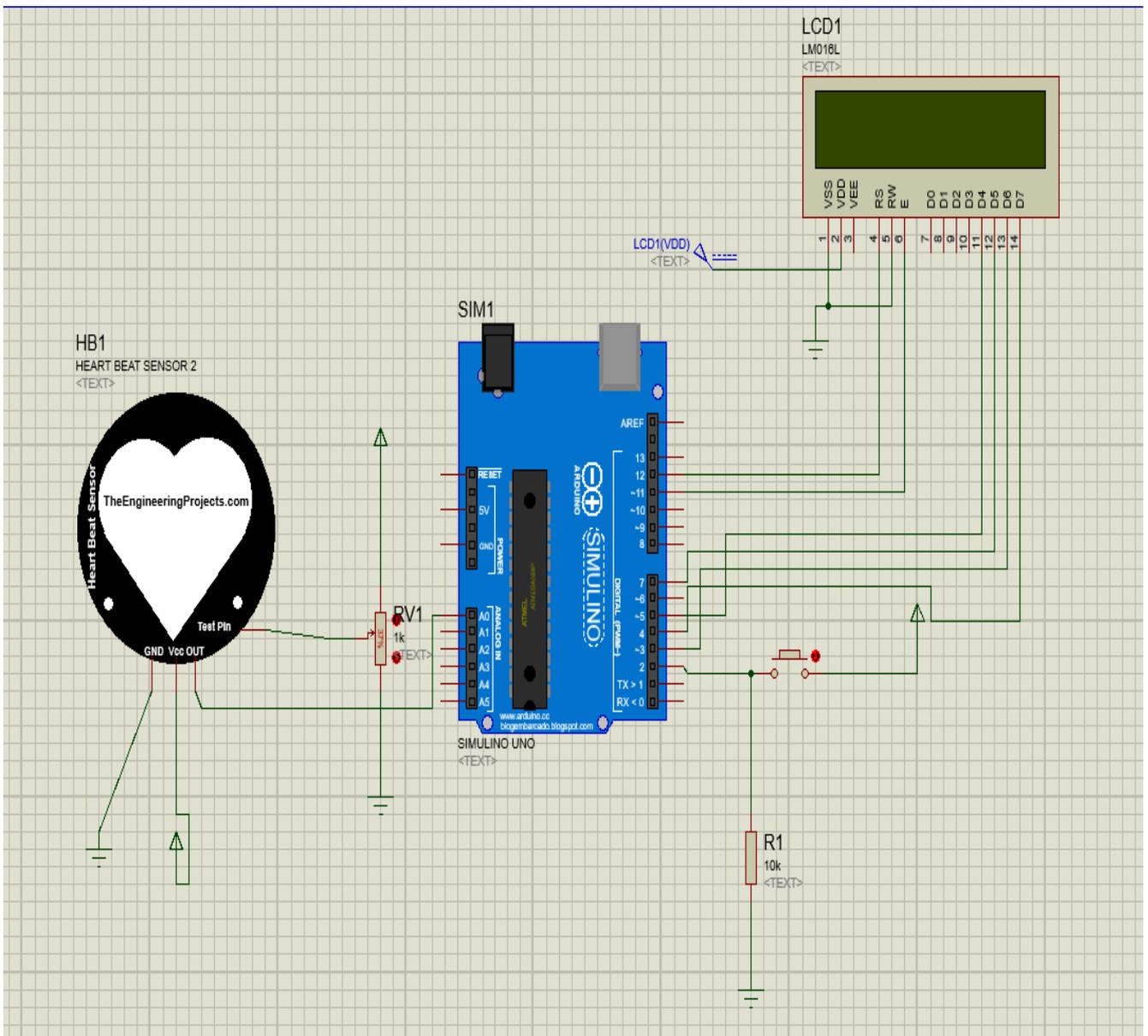


FIGURE 4.1 : Simulation Proteus

2 Câblage des matériels

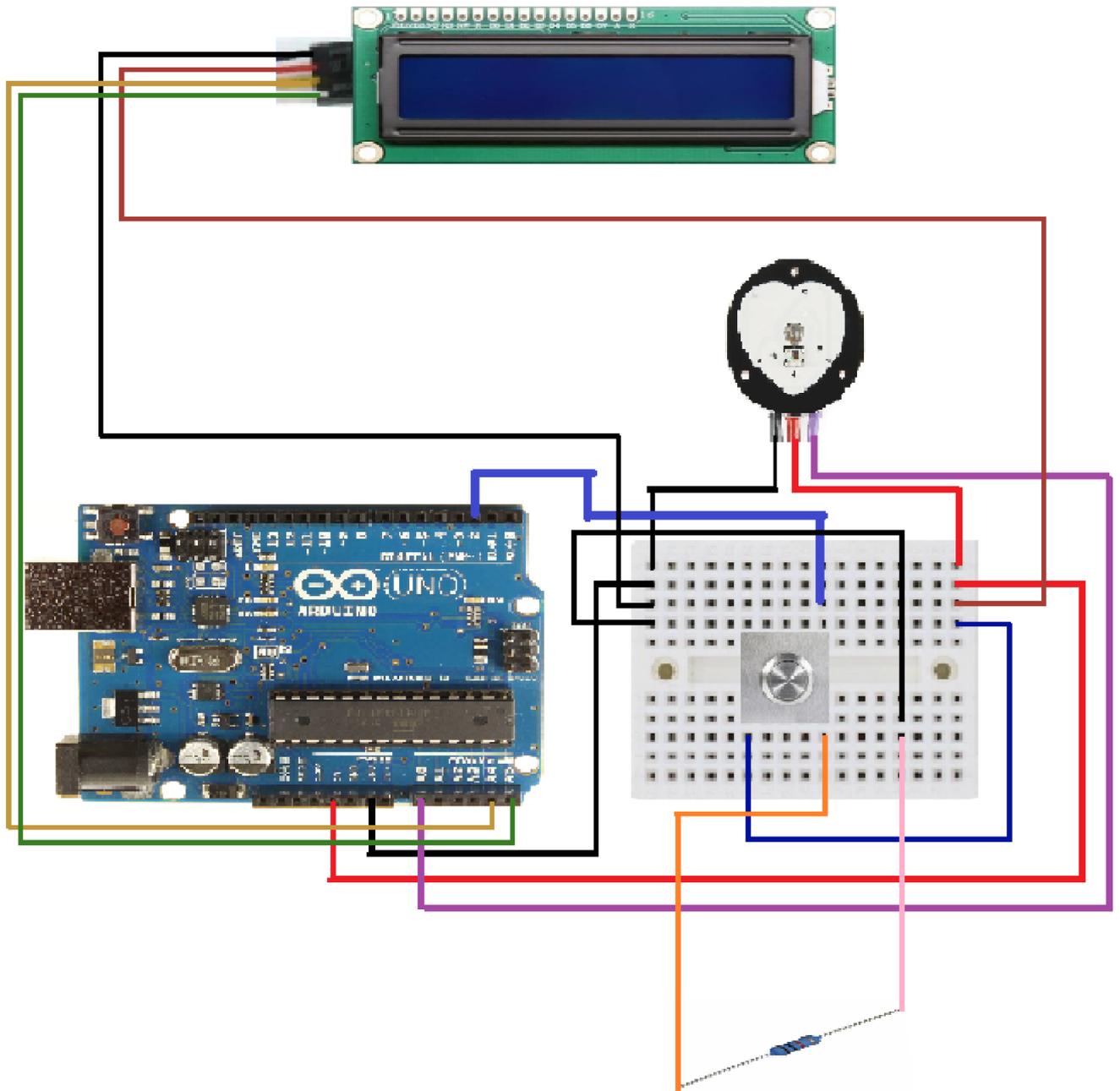


FIGURE 4.2 : Câblage

CHAPITRE 5

RÉALISATION

1	Code du projet	17
2	Assemblage des matériels et construction	23

Introduction

Cette partie va être consacrée à la mise en œuvre concrète des éléments planifiés de notre projet et à la réalisation du code.

1 Code du projet

1.1 Importation du bibliothèques et creation du variable

```
1  #include <LiquidCrystal_I2C.h>
2  LiquidCrystal_I2C lcd(0x27, 16, 2);
3  boolean countStatus;
4  int beat, bpm;
5  unsigned long millisBefore;
6  int firstBPM[10];
7  int index = 0;
8  int maxBPM = 0;
9  const int buttonPin = 2;
10 int buttonState = 0;
11 bool Button = false;
12 unsigned long tempsDebut;
13 const unsigned long tempsAttente = 15000;
14 byte Heart[] = {
15     B00000,
16     B01010,
17     B11111,
18     B11111,
19     B01110,
20     B00100,
21     B00000,
22     B00000
23 };
```

FIGURE 5.1 : Code - Importation et creation

1.2 Initialisation de la communication série, configuration des broches d'entrée et de l'écran LCD

```
25 void setup() {  
26     Serial.begin(9600);  
27     pinMode(buttonPin, INPUT);  
28     lcd.init();  
29     lcd.backlight();  
30     lcd.createChar(0, Heart);  
31 }  
32 }
```

FIGURE 5.2 : Code - Setup

1.3 Fonctionnement de détecteur de mensonge

```
void loop() {

  int sensorValue = analogRead(A0);
  buttonState = digitalRead(buttonPin);

  lcd.setCursor(0, 0);
  lcd.write(byte(0));
  lcd.setCursor(2, 0);
  lcd.print("Lie detector");
  lcd.setCursor(15, 0);
  lcd.write(byte(0));

  if (buttonState == HIGH && !Button) {
    Button = true;
    tempsDebut = millis();
  }
  if (Button && millis()-tempsDebut > tempsAttente) {
    Button = false;
  }

  if (countStatus == 0) {
    if (sensorValue > 516) {
      countStatus = 1;
      beat++;
    }
  } else {
    if (sensorValue < 510) {
      countStatus = 0;
    }
  }
}
```

(a) Loop 1

```
if (millis()-millisBefore > 15000) {
  bpm = beat*4;
  beat = 0;
  if (bpm >= 60 && bpm <= 100) {
    Serial.print("BPM : ");
    Serial.println(bpm);

    if (index < 5) {
      firstBPM[index] = bpm;
      if (bpm > maxBPM) {
        maxBPM = bpm;
      }
      index++;
    } else {
      if (Button) {
        boolean isLying = (bpm > maxBPM);
        if (isLying) {
          lcd.clear();
          lcd.print("Oups you're lying");
        } else {
          lcd.clear();
          lcd.print("Great you're telling the truth");
        }
      }
    }
  } else {
    Serial.println("Invalid BPM value");
    Serial.println(bpm);
  }
  millisBefore=millis();
}
delay(1);
}
```

(b) Loop 2

FIGURE 5.3 : Code - Loop

1.3.1 Explication

Lecture du capteur analogique et du l'état du bouton :

```
int sensorValue = analogRead(A0);
buttonState = digitalRead(buttonPin);
```

FIGURE 5.4 : Lecture du capteur analogique et du l'état du bouton

Ce bout de code lit la valeur analogique de la broche A0 à l'aide de la fonction `analogRead()` et la stocke dans la variable `sensorValue`. Il lit également l'état numérique du bouton connecté à la broche `buttonPin` à l'aide de la fonction `digitalRead()` et le stocke dans la variable `buttonState`.

Affichage sur l'écran LCD :

```
lcd.setCursor(0, 0);  
lcd.write(byte(0));  
lcd.setCursor(2, 0);  
lcd.print("Lie detector");  
lcd.setCursor(15, 0);  
lcd.write(byte(0));
```

FIGURE 5.5 : Affichage sur l'écran LCD

Ces lignes définissent la position du curseur sur l'écran LCD et affichent le titre "Détecteur de mensonge" avec les symboles spéciaux sur les côtés.

Détection de l'appui sur le bouton :

```
if (buttonState == HIGH && !Button) {  
    Button = true;  
    tempsDebut = millis();  
}  
if (Button && millis()-tempsDebut > tempsAttente) {  
    Button = false;  
}
```

FIGURE 5.6 : Détection de l'appui sur le bouton

Ces lignes vérifient si le bouton a été pressé et si le temps d'attente minimum est écoulé depuis le dernier appui sur le bouton. Si c'est le cas, la variable "Button" est "true" et le temps de début est stocké dans la variable "tempsDebut". Si le temps d'attente minimum est écoulé, la variable "Button" est "false".

Calcul de la fréquence cardiaque :

```
if (countStatus == 0) {  
    if (sensorValue > 516) {  
        countStatus = 1;  
        beat++;  
    }  
} else {  
    if (sensorValue < 510) {  
        countStatus = 0;  
    }  
}  
  
if (millis()-millisBefore > 15000) {  
    bpm = beat*4;  
    beat = 0;  
}
```

FIGURE 5.7 : Calcul de la fréquence cardiaque

Ce code est pour mesurer le rythme cardiaque de l'utilisateur. Il compte le nombre de battements de cœur en fonction des changements de la valeur du capteur. La fréquence cardiaque est ensuite calculée en nombre de battements par minute (BPM) et stockée dans la variable bpm.

Vérifier l'honnêteté de la personne :

```
    if (bpm >= 60 && bpm <= 100) {
        Serial.print("BPM : ");
        Serial.println(bpm);

        if (index < 5) {
            firstBPM[index] = bpm;
            if (bpm > maxBPM) {
                maxBPM = bpm;
            }
            index++;
        } else {
            if (Button) {
                boolean isLying = (bpm > maxBPM);
                if (isLying) {
                    lcd.clear();
                    lcd.print("Oups you're lying");
                } else {
                    lcd.clear();
                    lcd.print("Great you're telling the truth");
                }
            }
        }
    } else {
        Serial.println("Invalid BPM value");
        Serial.println(bpm);
    }
}
```

FIGURE 5.8 : Vérifier l'honnêteté de la personne

Dans ce bout de code on vérifie si la personne ment ou pas

- Si la valeur de BPM est comprise entre 60 et 100, il y a deux cas possibles :
 - ✓ Si l'index est inférieur à 5, alors la valeur de BPM est stockée dans un tableau nommé "firstBPM" à l'index correspondant. Si la valeur de BPM est supérieure à la valeur maximale enregistrée jusqu'à présent (stockée dans la variable "maxBPM"), la valeur maximale est mise à jour. L'index est également incrémenté pour le prochain enregistrement.
 - ✓ Si l'index est supérieur ou égal à 5, cela signifie que les 5 premières valeurs de BPM ont été enregistrées et la détection de mensonge commence. Si le bouton est enfoncé (Button == true), alors le programme compare la valeur de BPM actuelle à la valeur

maximale enregistrée. Si la valeur actuelle est supérieure à la valeur maximale, cela signifie que la personne ment. Le programme efface alors l'affichage LCD et affiche le message "Oups you're lying". Sinon, le programme affiche le message "Great you're telling the truth".

- Si la valeur de BPM n'est pas comprise entre 60 et 100, le programme affiche un message d'erreur "Invalid BPM value" suivi de la valeur de BPM.

2 Assemblage des matériels et construction



FIGURE 5.9 : Matériels

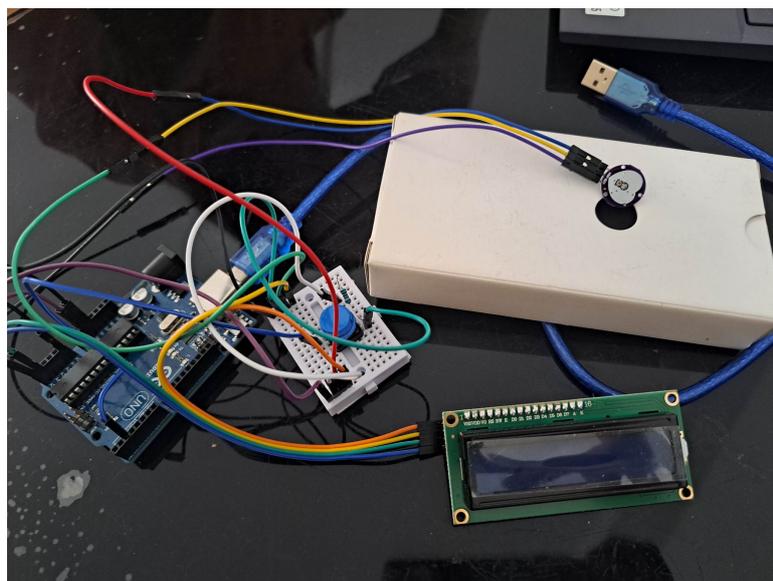


FIGURE 5.10 : Matériels détecteur de mousange

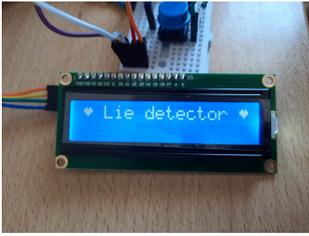


FIGURE 5.11 : Aff-detector



FIGURE 5.12 : Aff-liar



FIGURE 5.13 : Aff-truth

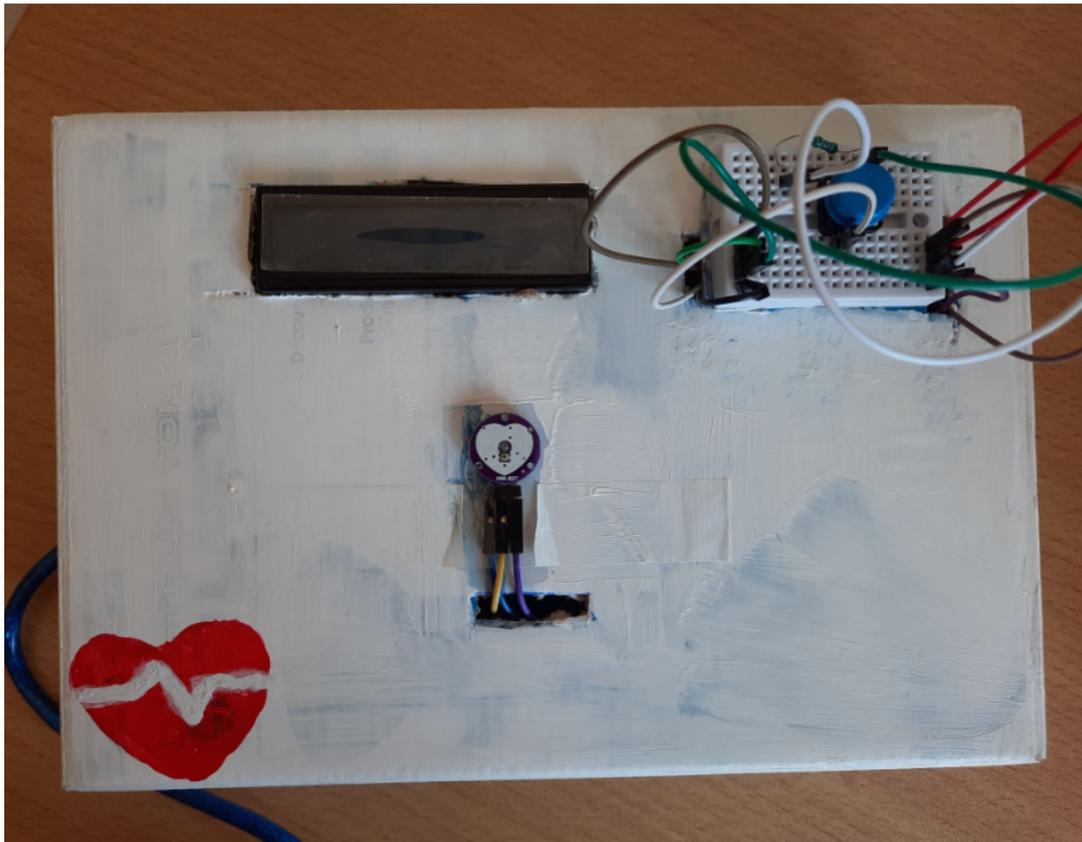


FIGURE 5.14 : détecteur de mensonge

CONCLUSION GÉNÉRALE

Pour conclure, ce projet de construction d'un détecteur de mensonge dans le cadre de la matière "Système de microprocesseur" nous a permis de mettre en pratique les connaissances acquises tout au long de notre formation. Nous avons commencé par une introduction pour présenter le contexte et les objectifs de ce projet, puis nous avons identifié les technologies et langages adaptés, ainsi que les matériaux nécessaires. La simulation nous a permis de valider notre conception avant de passer à la phase de réalisation, où nous avons mis en œuvre nos compétences pour construire un détecteur de mensonge fonctionnel.

Ce projet a ainsi été une opportunité pour nous d'approfondir notre compréhension du fonctionnement des systèmes de microprocesseurs et de développer nos compétences en matière de conception et de réalisation de projets électroniques.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Positron libre. Qu'est-ce que arduino? <https://www.positron-libre.com/electronique/arduino/arduino.php>, 2020.
- [2] elektronique. Proteus (isis et ares). <http://www.elektronique.fr/logiciels/proteus.php>, 2019.
- [3] collegegujan. Fonction traitement de l'information : carte de prototypage rapide arduino uno. https://www.collegegujan.fr/sites/technopc/img/2016_2017/4eme/sequence1/activite14/4emeactivite14/fonction_traitement_de_linformation__carte_de_prototypage_rapide_arduino_uno.html, 2019.
- [4] robocraze. what is pulse sensor. <https://robocraze.com/blogs/post/what-is-pulse-sensor>, 2020.
- [5] robotique tech. Afficheur i2c lcd 1602a. https://www.robotique.tech/tutoriel/afficheur-i2c_lcd-1602a/, 2020.