



Conception et réalisation d'un système de télésurveillance médical

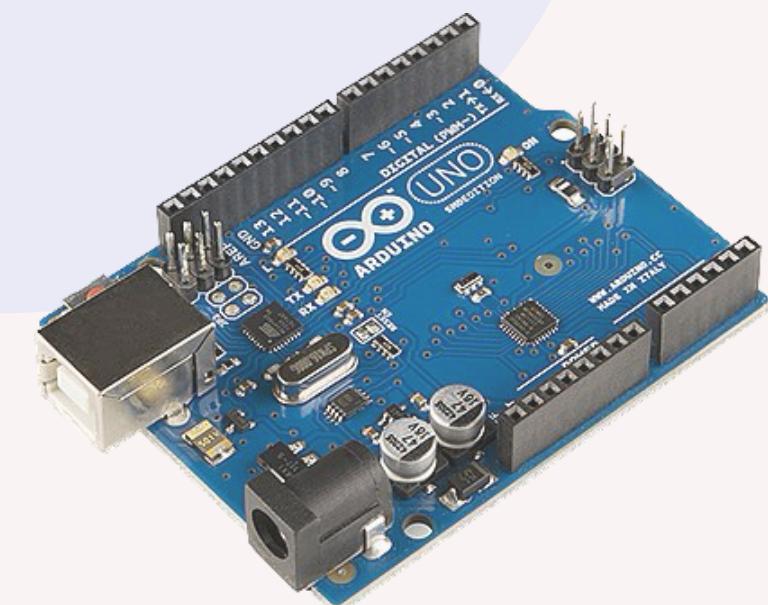


RÉALISÉE PAR :

NIAMATELLAH LAHKIM, SARA EL BARI, MORGANE CARBILLET

L3 BCMP PLURI IA

2023-2024



Plan

Introduction

Cahier des charges

Description du système physique

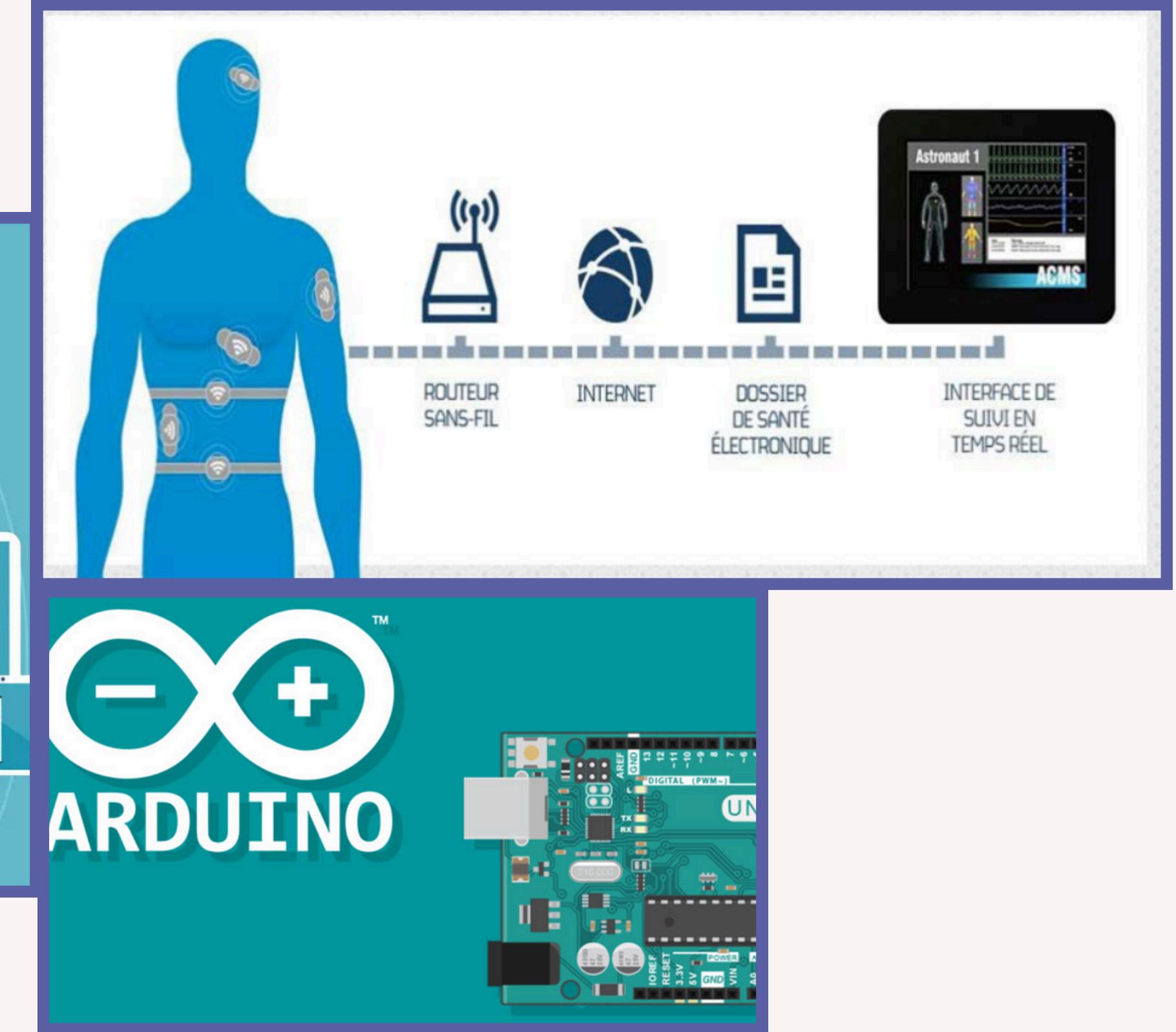
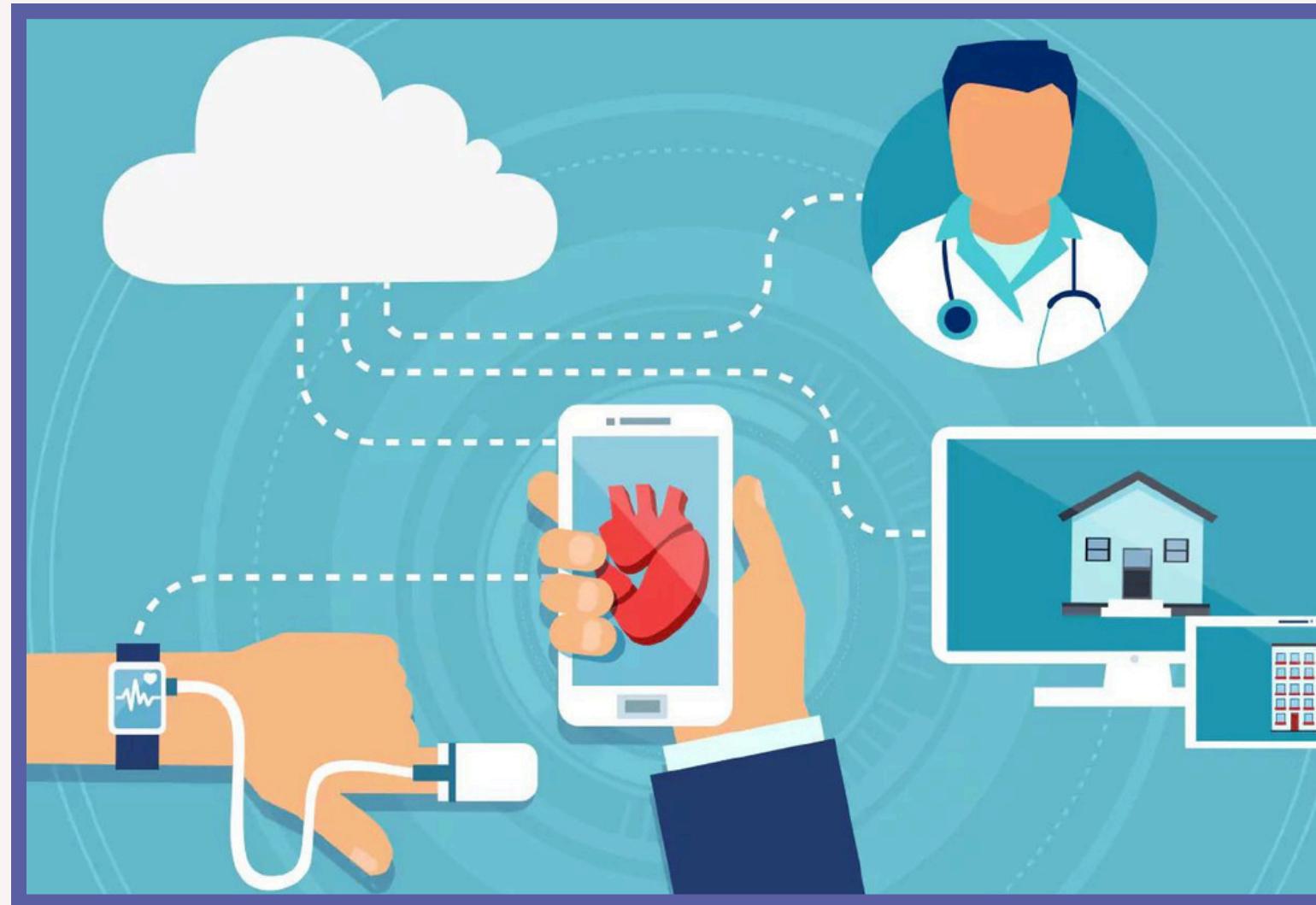
Description du système électronique

Descriptif du programme informatique

Conclusion



Introduction

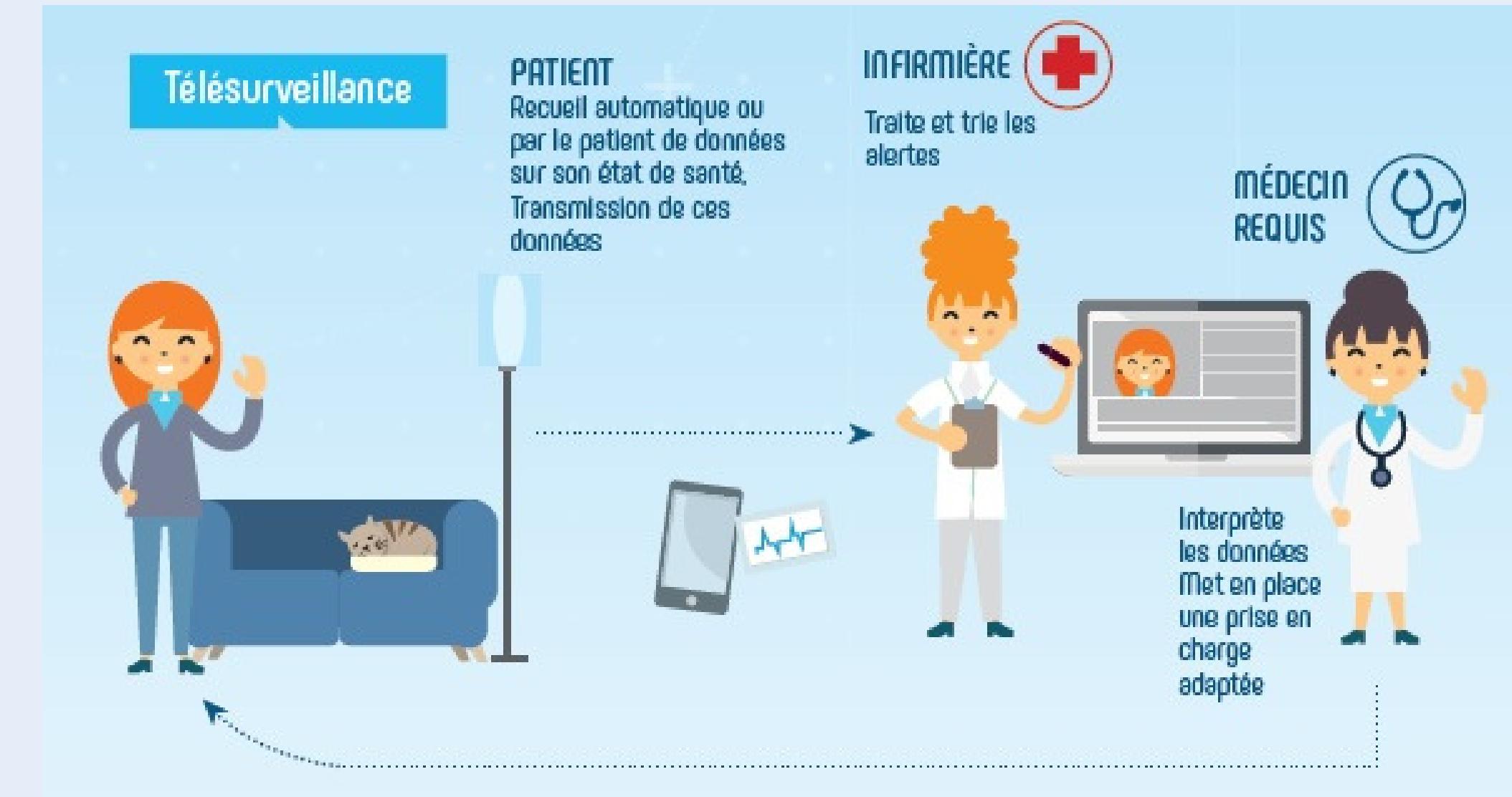


Objet: Concevoir et construire un système de télésurveillance médical permettant le suivi continu à distance de paramètres de santé vitaux grâce à un microcontrôleur Arduino Uno

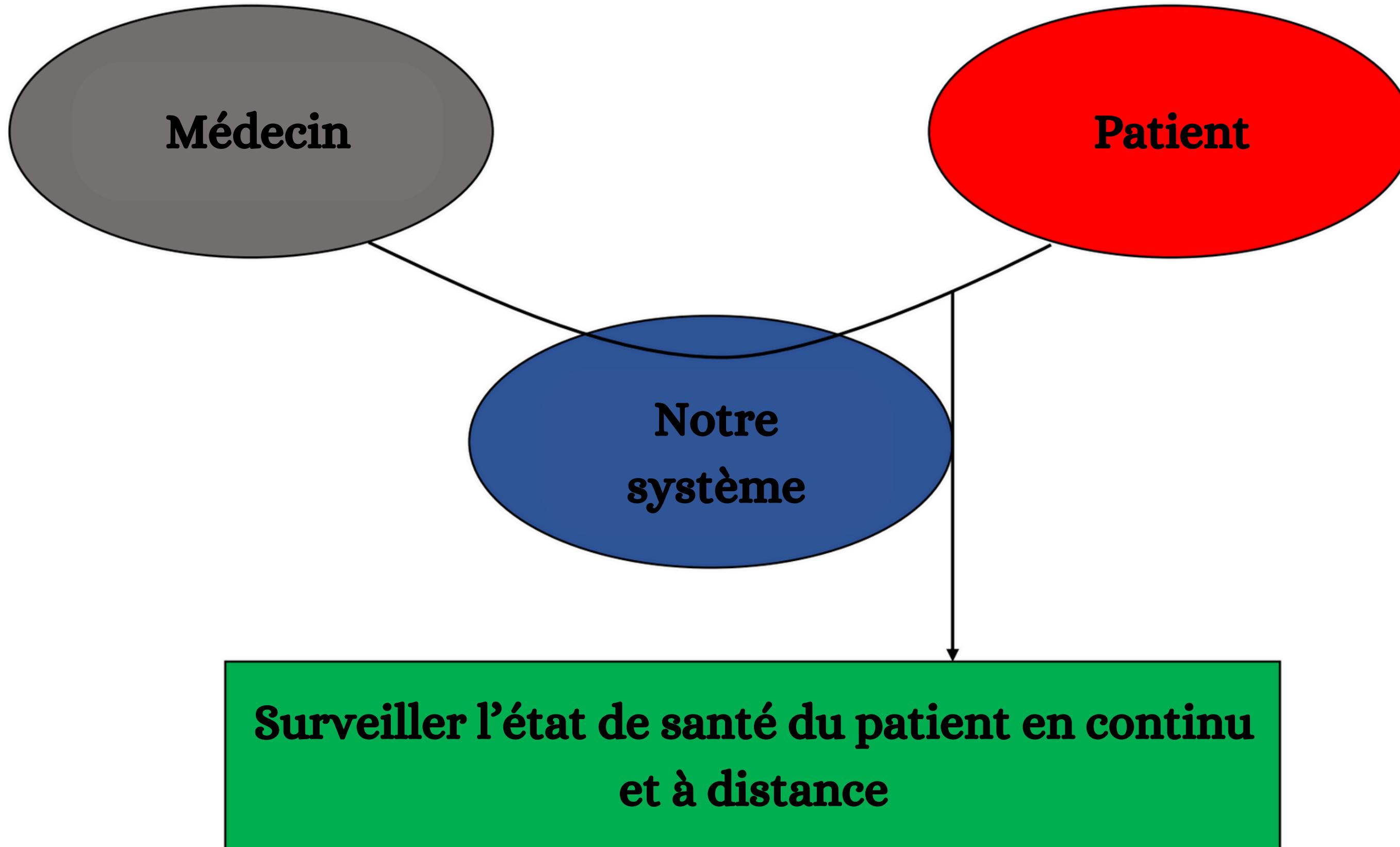
Cahier de charge

ATTENDU:

- Mesurer la température corporelle, la fréquence cardiaque et la saturation en oxygène du sang du patient.
- Afficher sur un écran les valeurs de ces mesures.
- Alerter par un signal sonore lorsque les valeurs sont en dehors des normes.
- Envoyer les données à distance vers un logiciel ThingSpeak.



❖ Digramme Bête à corne

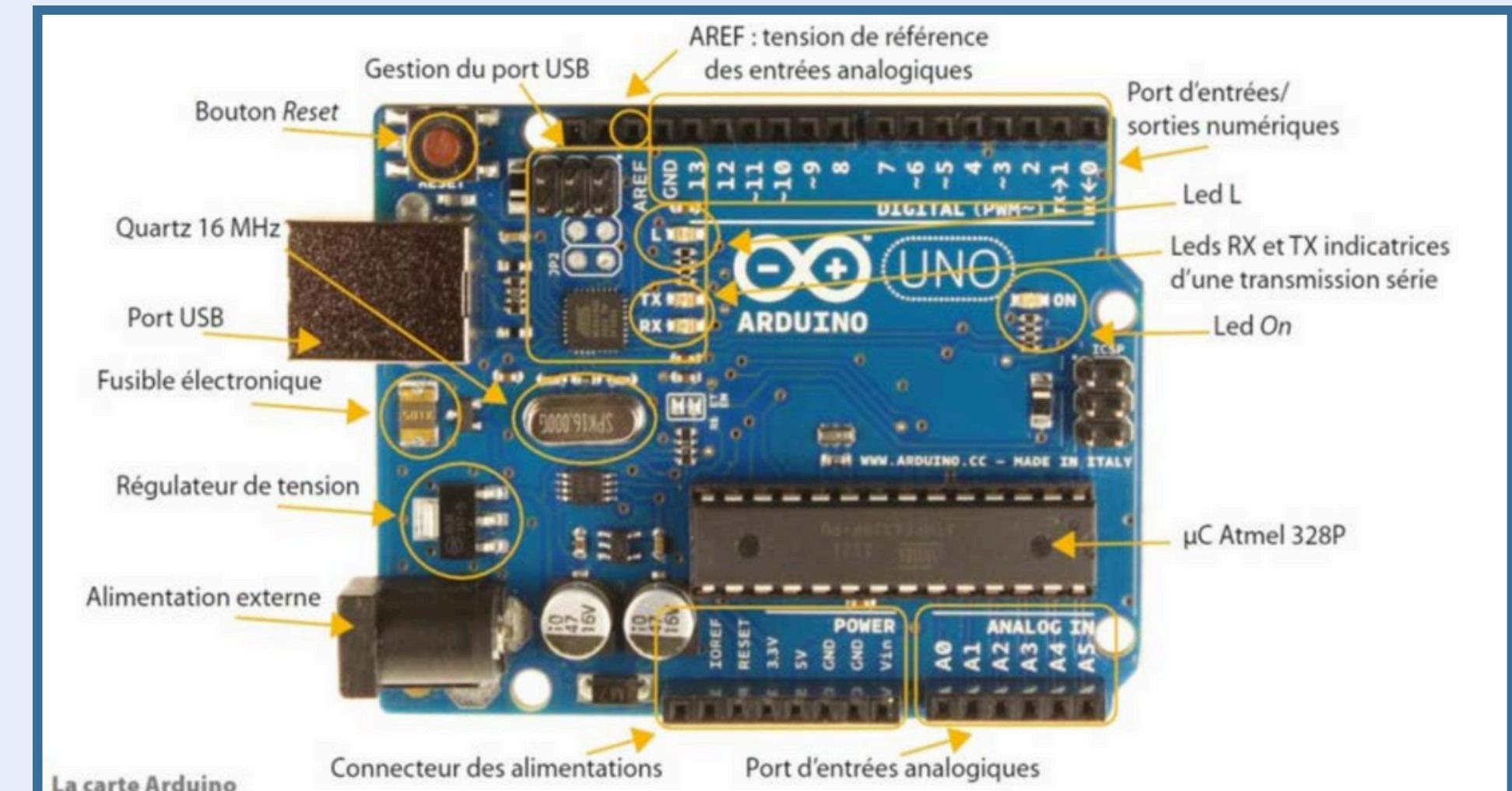


Cahier de charge

BESOINS:

1) La carte Arduino UNO :

- Carte équipée d'un microcontrôleur ATmega 328.
- Langage de programmation: C ou C++ simplifié.
- Présence d'un nombre suffisant de broches d'entrée – sorties pour des projets élémentaires .
- Tension d'alimentation recommandée de 7V à 12V.

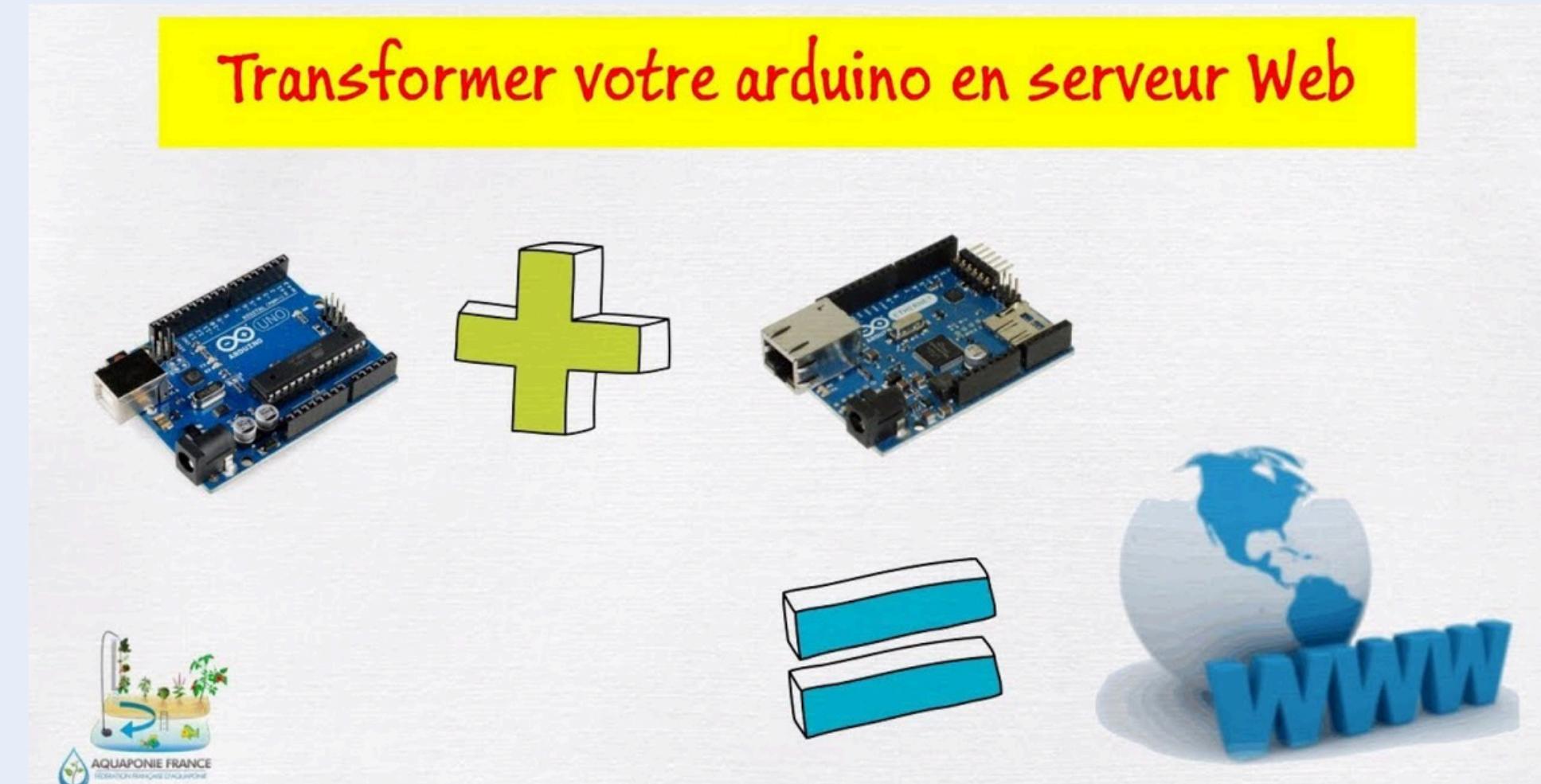


Cahier de charge

BESOINS:

2) La carte Ethernet Shield:

- Module d'extension pour les cartes Arduino.
- Permet de se connecter à un réseau Ethernet ou Internet.
- Elle fonctionne avec une tension d'entrée de 5V.
- Possibilité d'insérer une carte Micro SD afin de stocker des fichiers.
- Communique avec la carte Arduino via SPI (Serial Peripheral Interface)

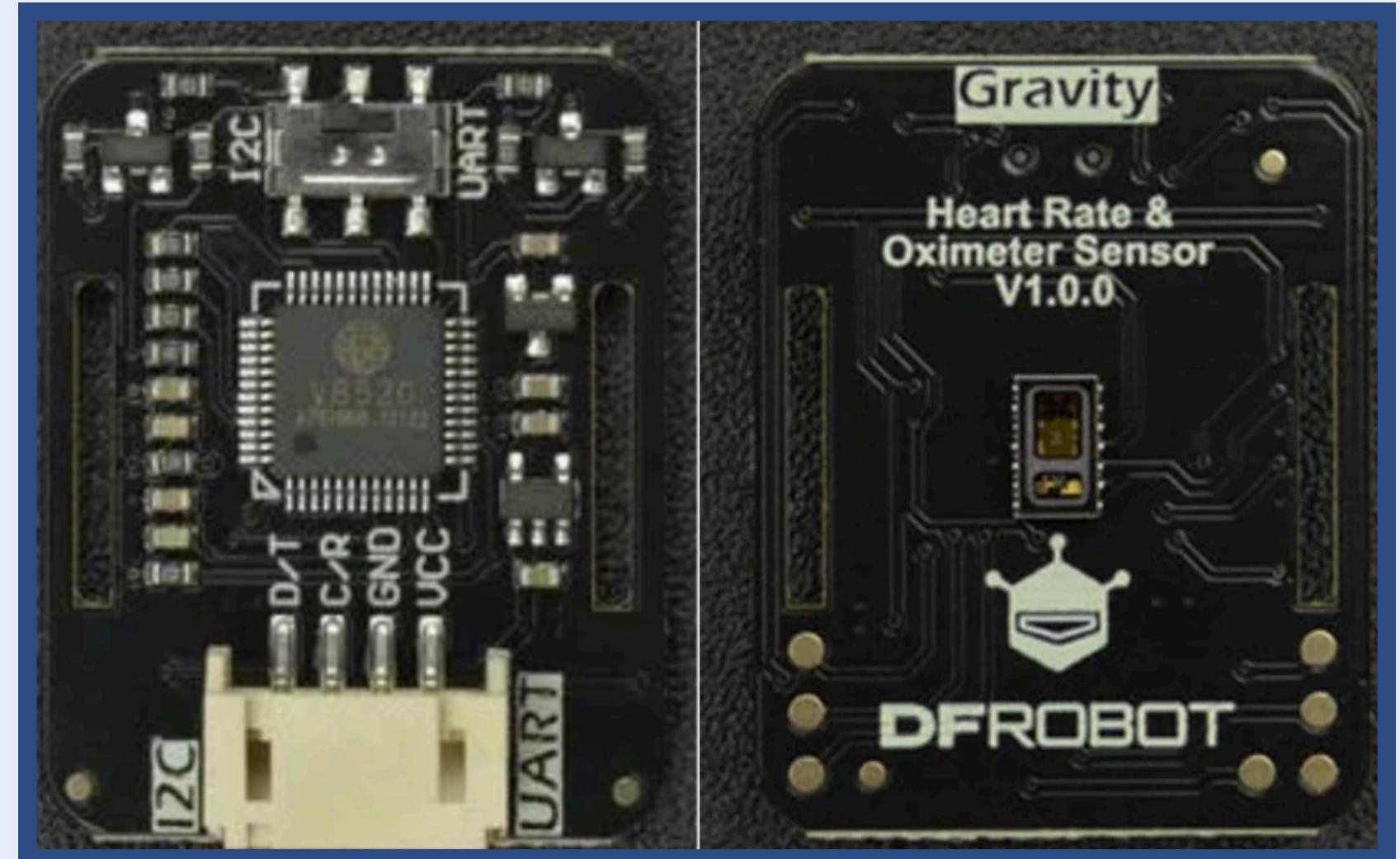


Cahier de charge

BESOINS:

3) Capteur MAX30102:

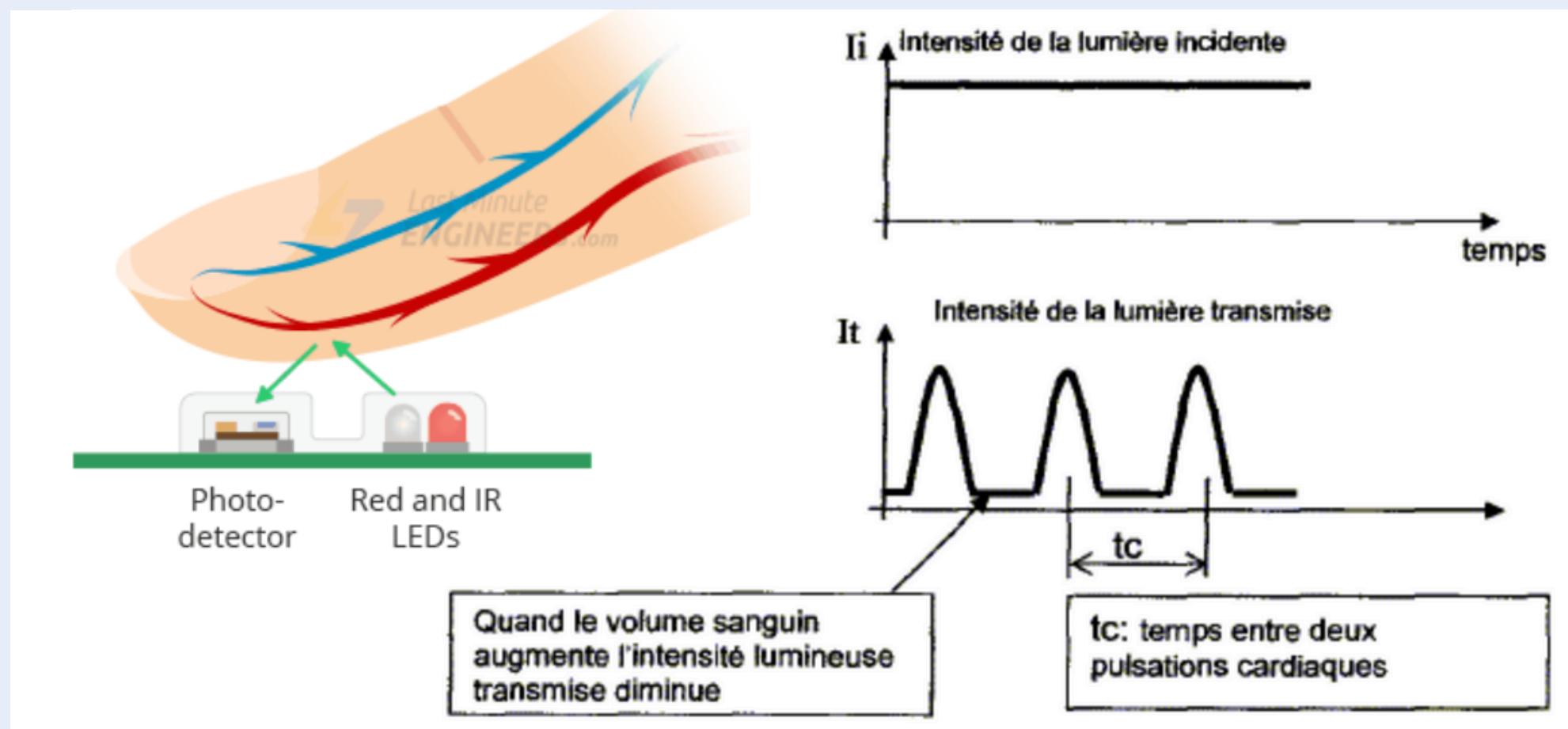
- Moniteur de fréquence cardiaque et oxymètre de pouls (niveau d'oxygène dans le sang).
- Mesure la température corporelle.
- Faible consommation d'énergie (**600 μ A** en mode **mesure** vs **0,7 μ A** en mode **veille**).
- Taux d'échantillonage élevé



Cahier de charge

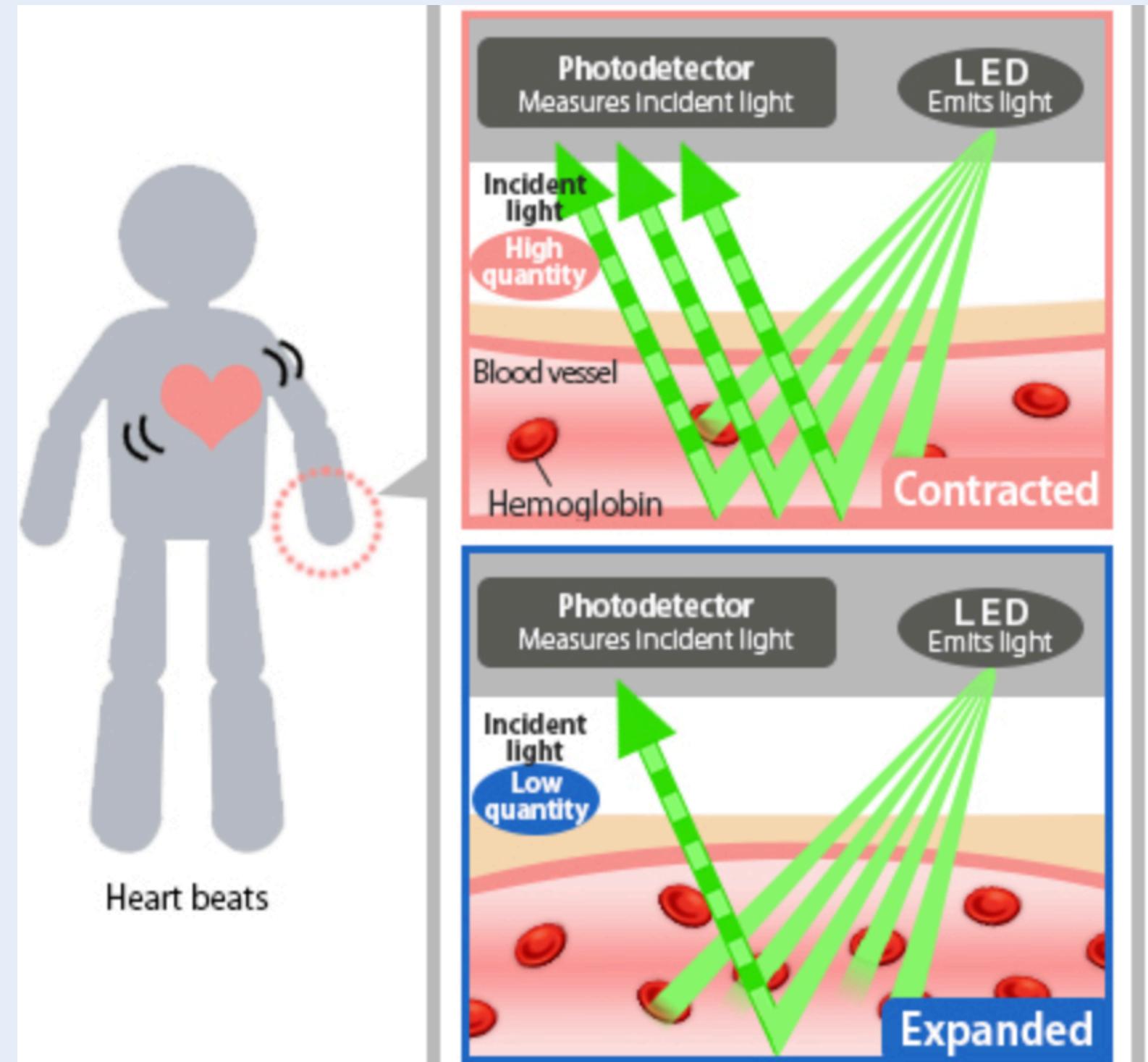
3) Capteur MAX30102:

Mesure de la fréquence cardiaque:



Principe de l'oxymétrie de Pouls:

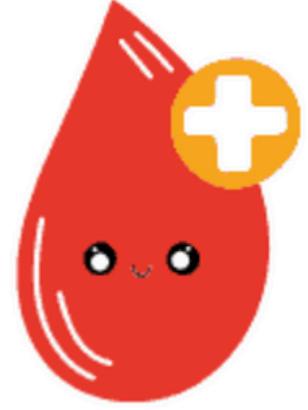
$$SpO_2 = \frac{CHbO_2}{CHb} \rightarrow \text{IR}$$
$$\rightarrow \text{R}$$



Cahier de charge

3) Capteur MAX30102:

Taux de saturation d'oxygène dans le sang



valeur normale :
95 à 100%

valeur critique :
< 90%

Battements par minute



fréquence normale :
55 à 85 bpm

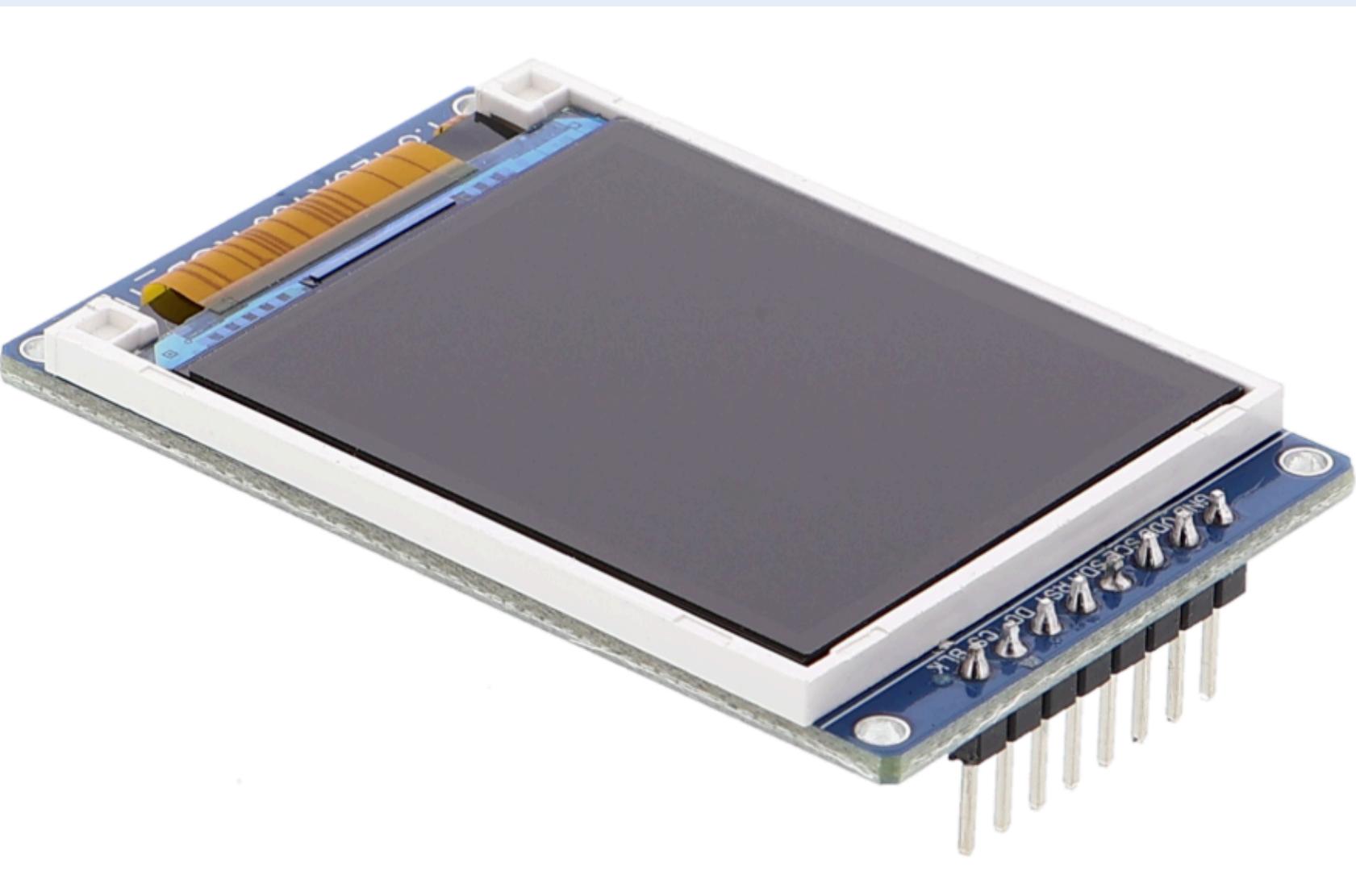
fréquence inquiétante :
< 55 bpm ou > 100 bpm

Cahier de charge

BESOINS:

5) Ecran LCD TFT (thin film transistor) 1,8 :

- Afficheur coloré 120 x 160 pixels de couleur.
- intègre un lecteur de carte SD.
- Chargement d'images à partir d'une carte SD.
- Dimension total: 56 x 34 x 5mm
- Compatible avec la carte ARDUINO.



Cahier de charge

BESOINS:

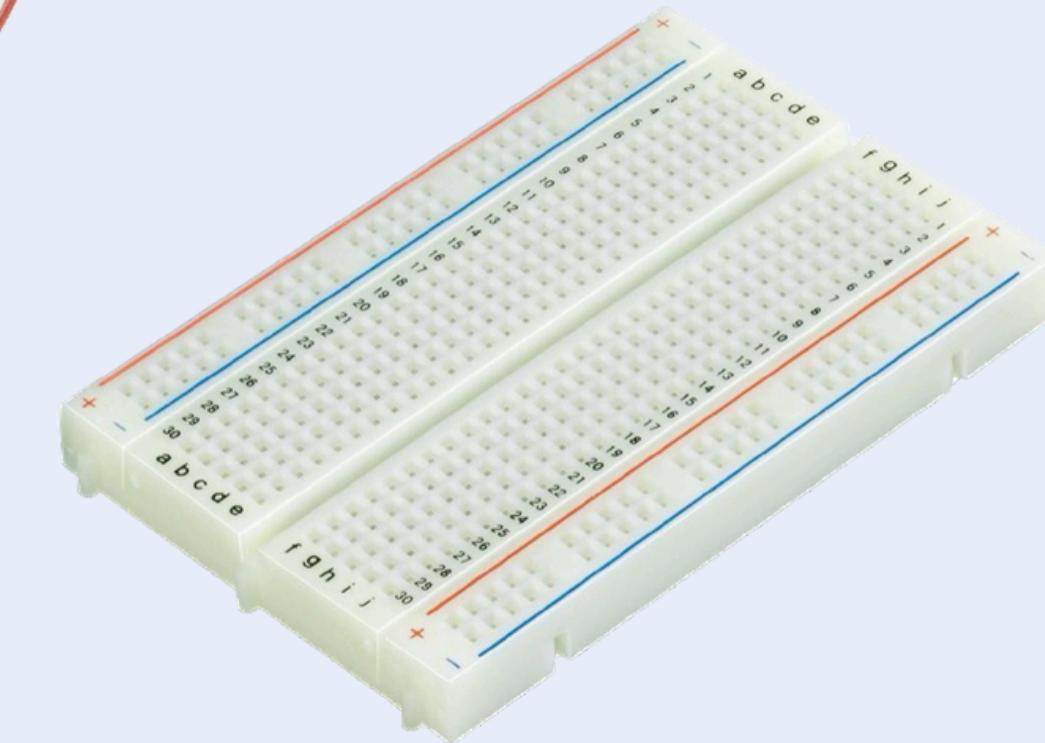
6) Buzzer:

- Production d'un son aigu.
- Rôle d'alarme.

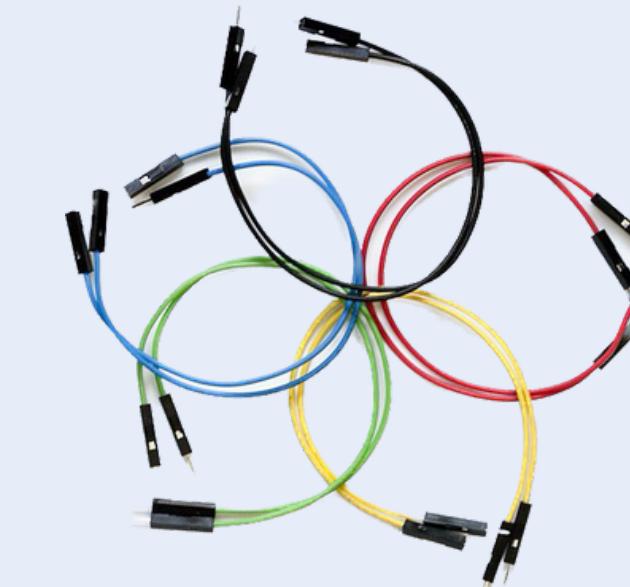


7) Platine d'essai ou Breadboard:

- Relie les composants électroniques.



8) Fils de connexion (jumper wire)



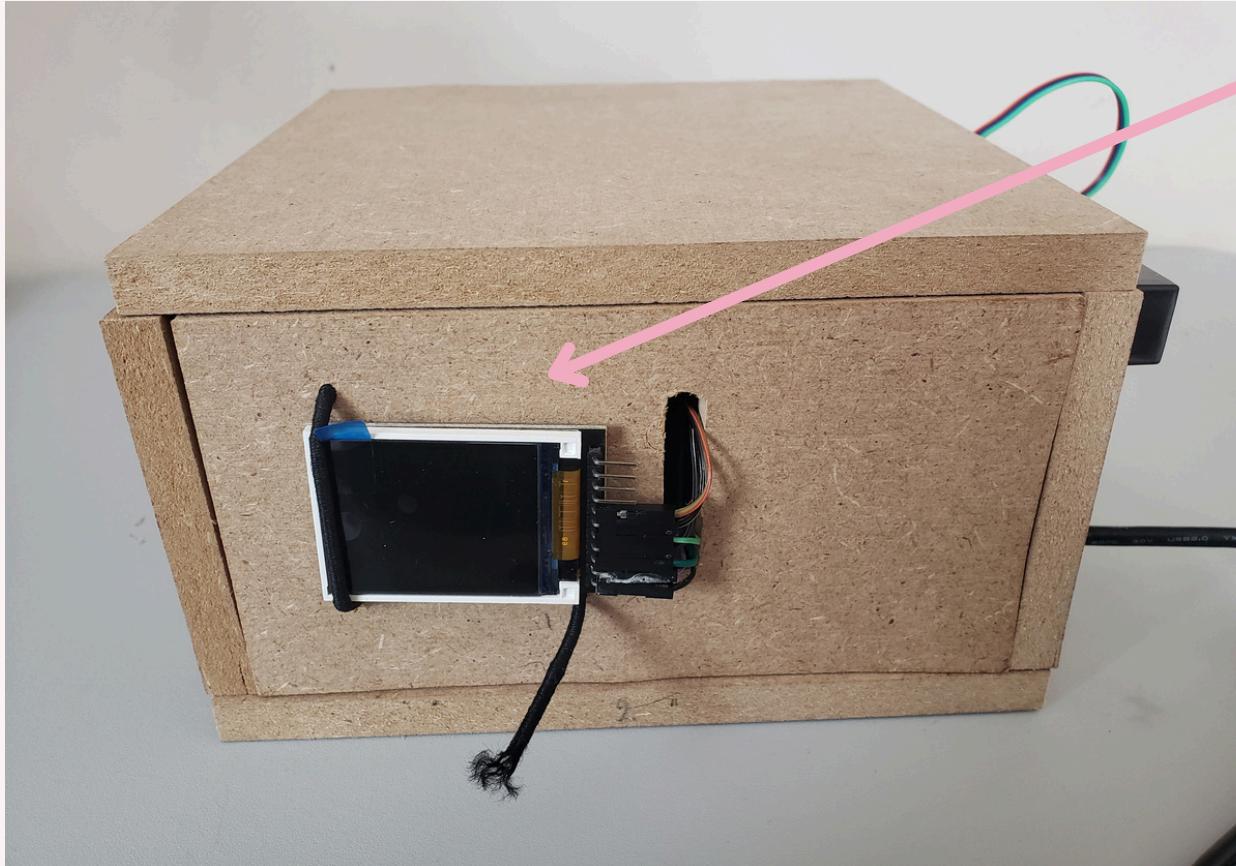
Cahier de charge

CÔTÉ FINANCIER:

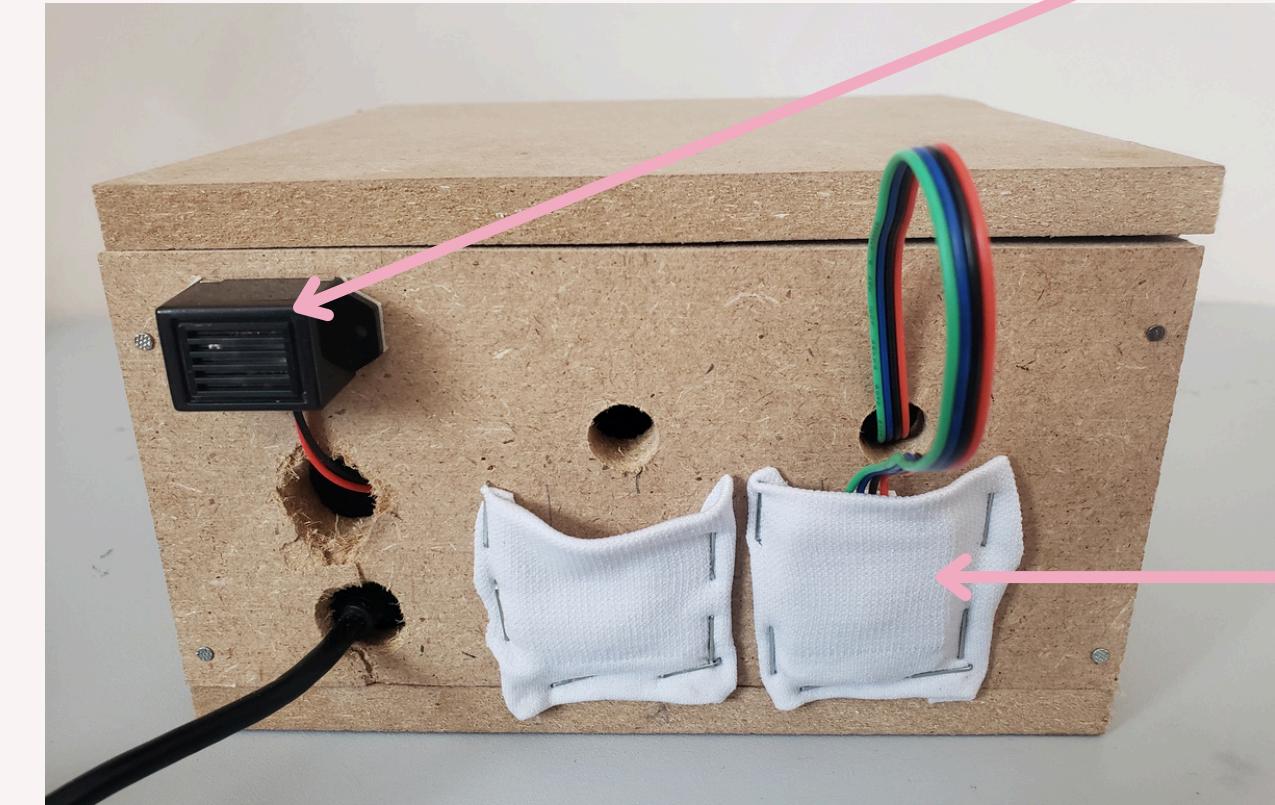
Carte Arduino UNO	28 €
carte Ethernet Shield	17 €
Capteur MAX30102	16 €
Ecran LCD TFT 1,8	12,80 €
Buzzer	1,68 €
Fils de connexion	2,24 €
Boîte	1,90 €

**Total:
79,62 €**

Description du système physique

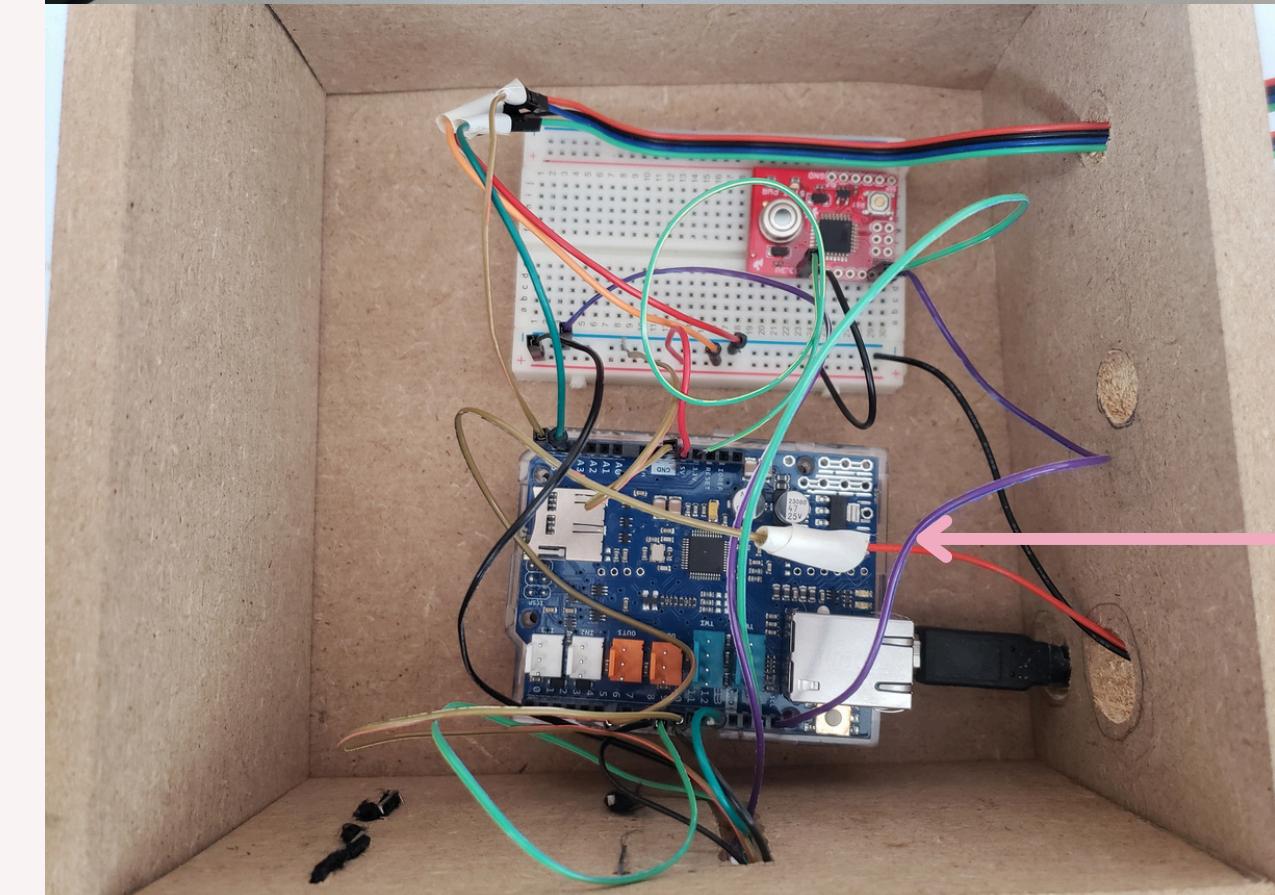


Ecran LCD



Buzzer

Capteur
MAX30102



Dimension:
17 x 17 x 10 cm

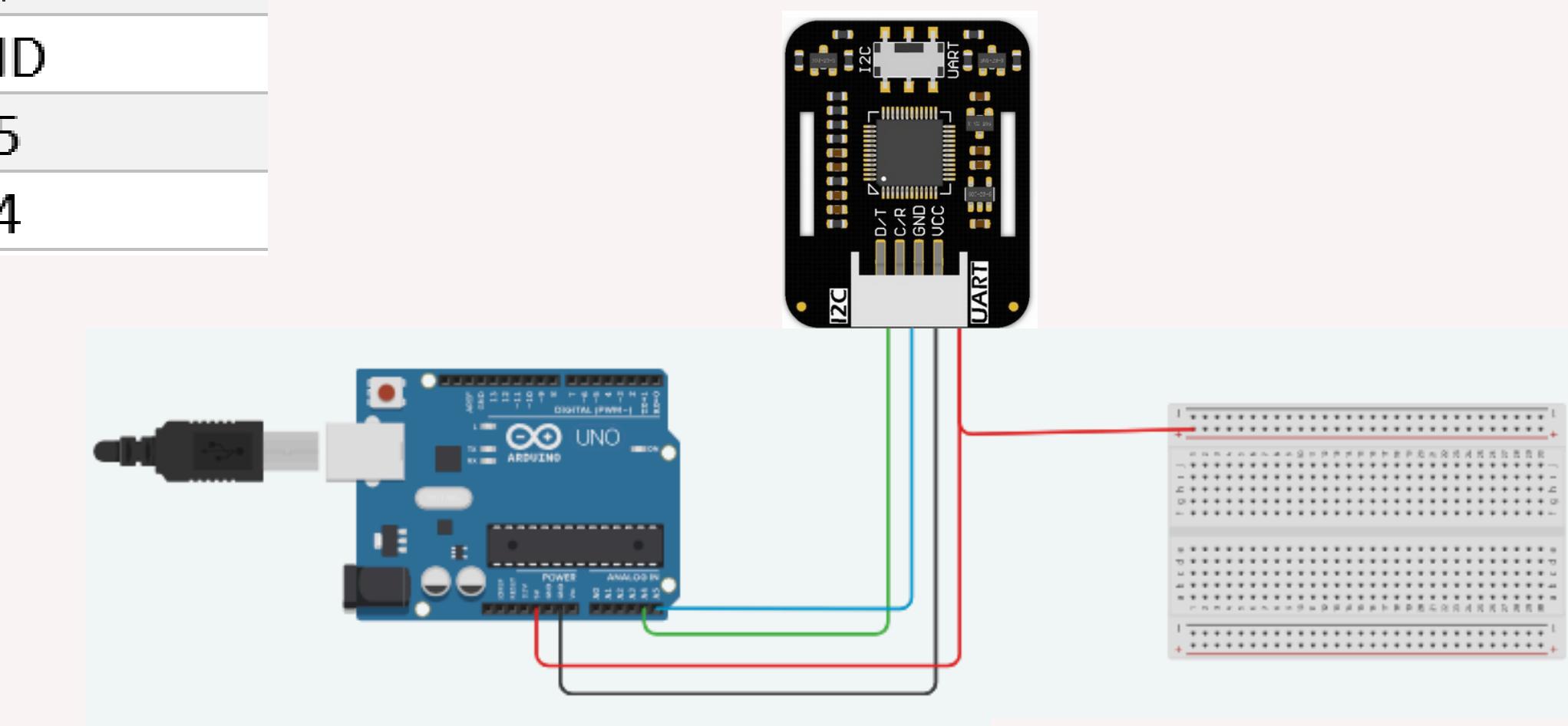
Carte Arduino
+
Carte Ethernet

Description du système électronique

MAX30102

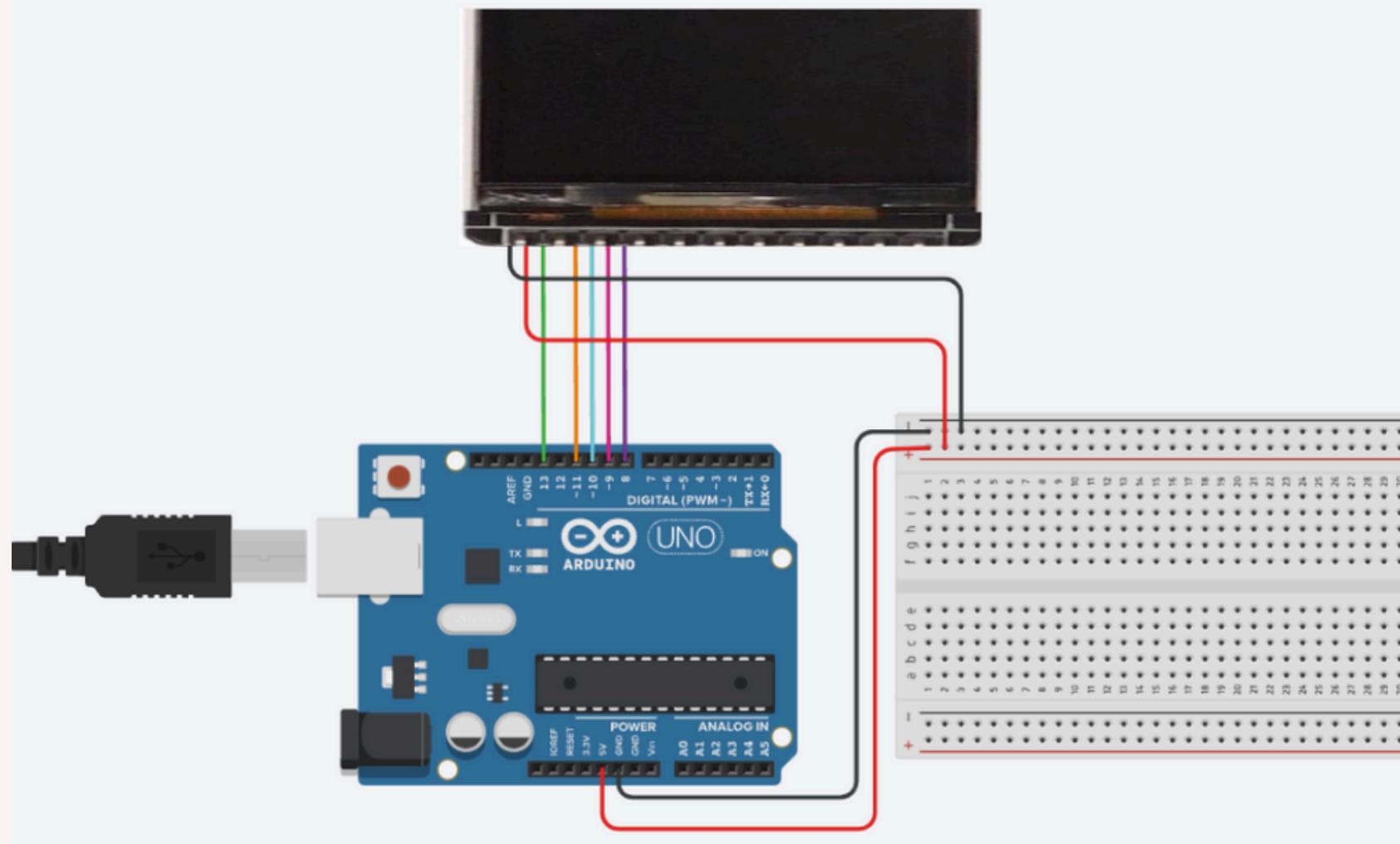
Branchements :

Carte UNO	MAX30102
VCC	5V
GND	GND
C/R (fil bleu)	A5
D/T (fil vert)	A4



Description du système électronique

ECRAN LCD TFT 1,8:



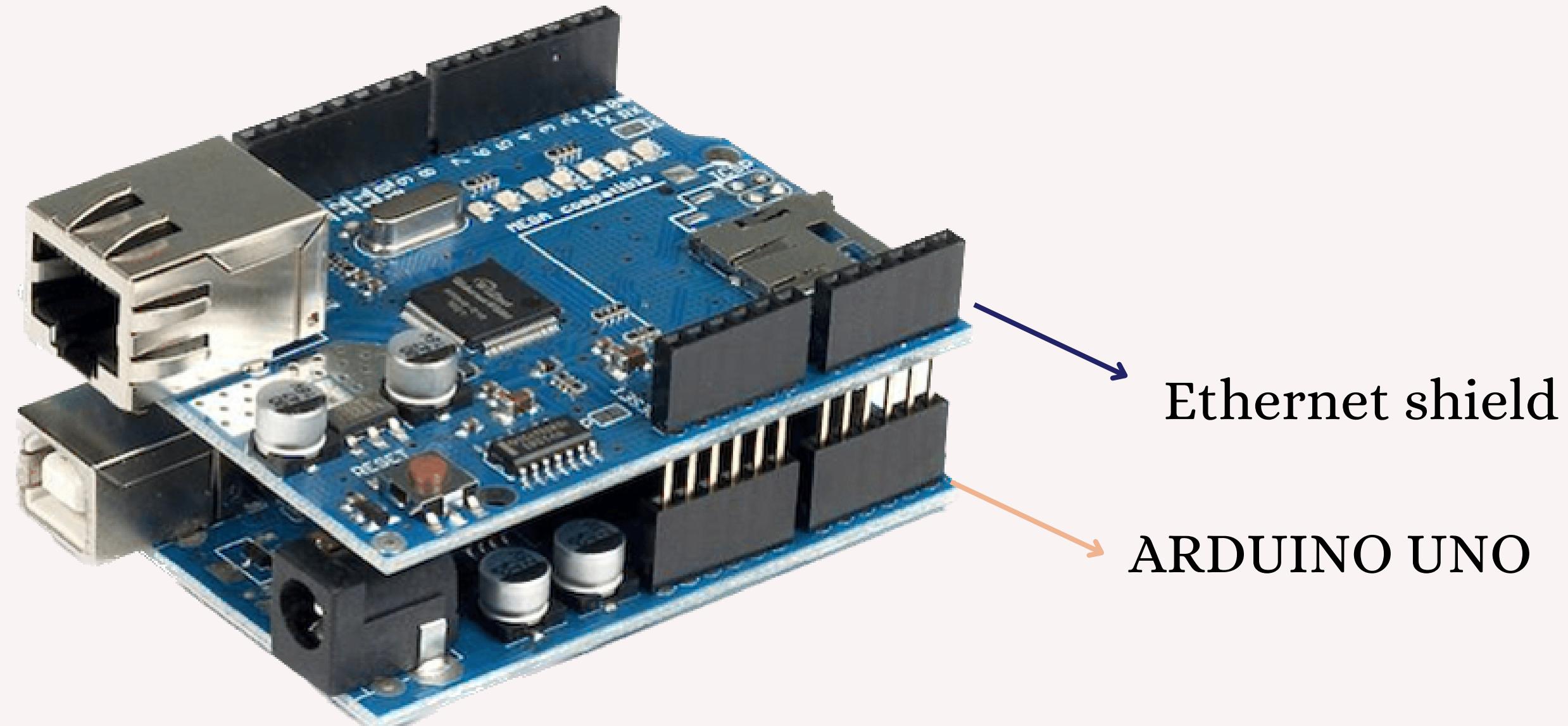
Branchements:

Carte UNO	Ecran
+5V	+5V
GND	GND
13	SCL
11	SDA
9	RS/DC
8	RES
10	CS

GND de l'Arduino avec la borne - de la platine et 5V de l'Arduino avec la borne + de la platine.

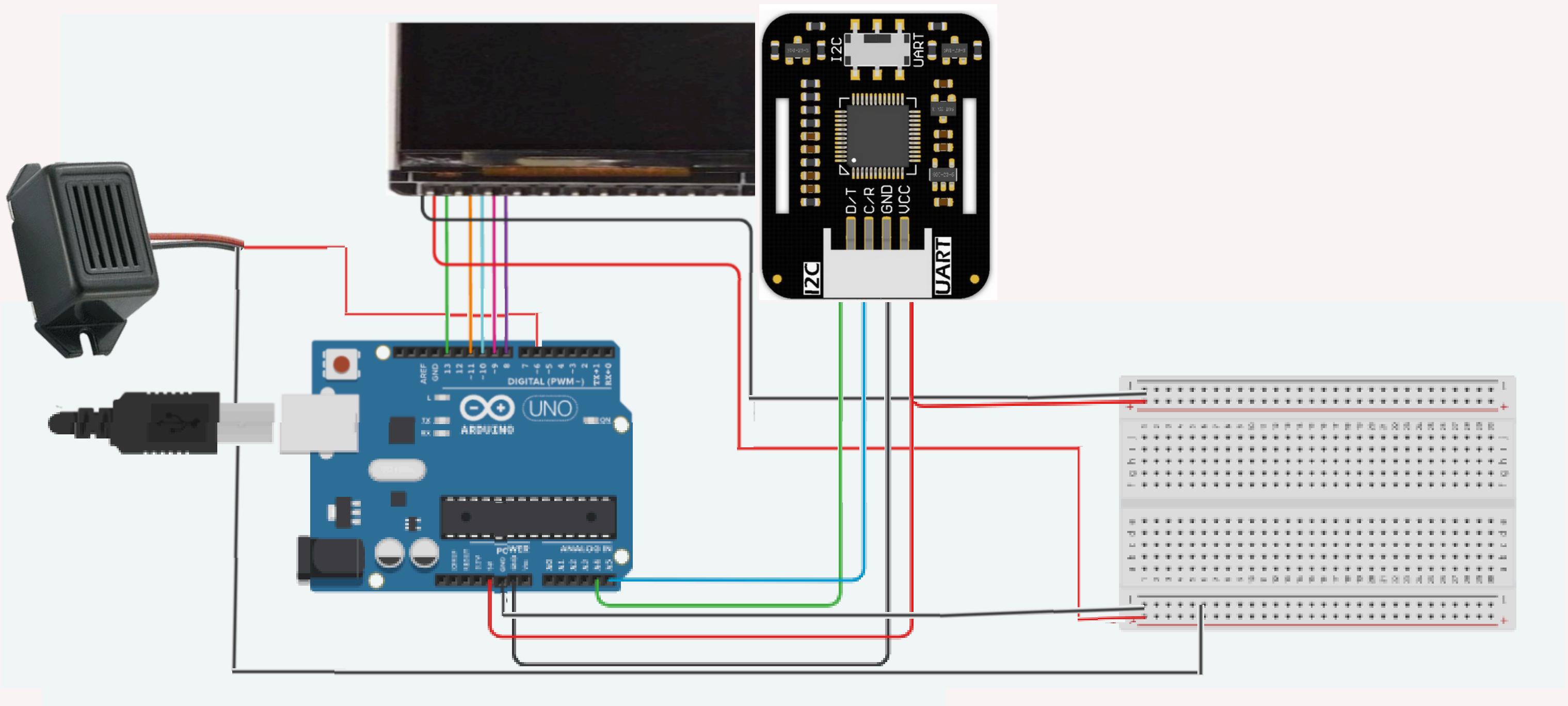
Description du système électronique

ETHERNET SHIELD



Description du système électronique

SCHEMA GLOBAL DU MONTAGE:



>>> Logiciel Tinkercad

Description du logiciel ThingSpeak:

The screenshot shows the homepage of the ThingSpeak website. The header includes a navigation bar with links for Internet Of Things - This, Internet of Things (IoT) Plat, Xively by LogMeIn: IoT Plat, Carriots - Internet of Things, nimbots platform, Azure IoT Hub | Microsoft A, Google Cloud Computing, Sign In, and Sign Up. Below the header is a large banner with the text "Billions and Billions" and "The open data platform for the Internet of Things". It features two buttons: "Get Started" (green) and "Contact Us" (white). A green notification bar at the bottom left says "Signed out successfully." Below the banner, there are three main sections: "Collect" (Send sensor data to the cloud), "Analyze" (Analyze and visualize your data), and "Act" (Trigger a reaction). At the bottom, there are sections for "ThingSpeak Features" (Real-time data collection and storage, MATLAB® analytics and visualizations, Alerts, Device communication, Open API, Geolocation data) and "Works With" (Arduino®, Particle Photon and Core, Raspberry Pi™, Mobile and web apps, Twitter®, Twilio®).

Signed out successfully.

 Collect

Send sensor data to the cloud.

 Analyze

Analyze and visualize your data.

 Act

Trigger a reaction.

ThingSpeak Features

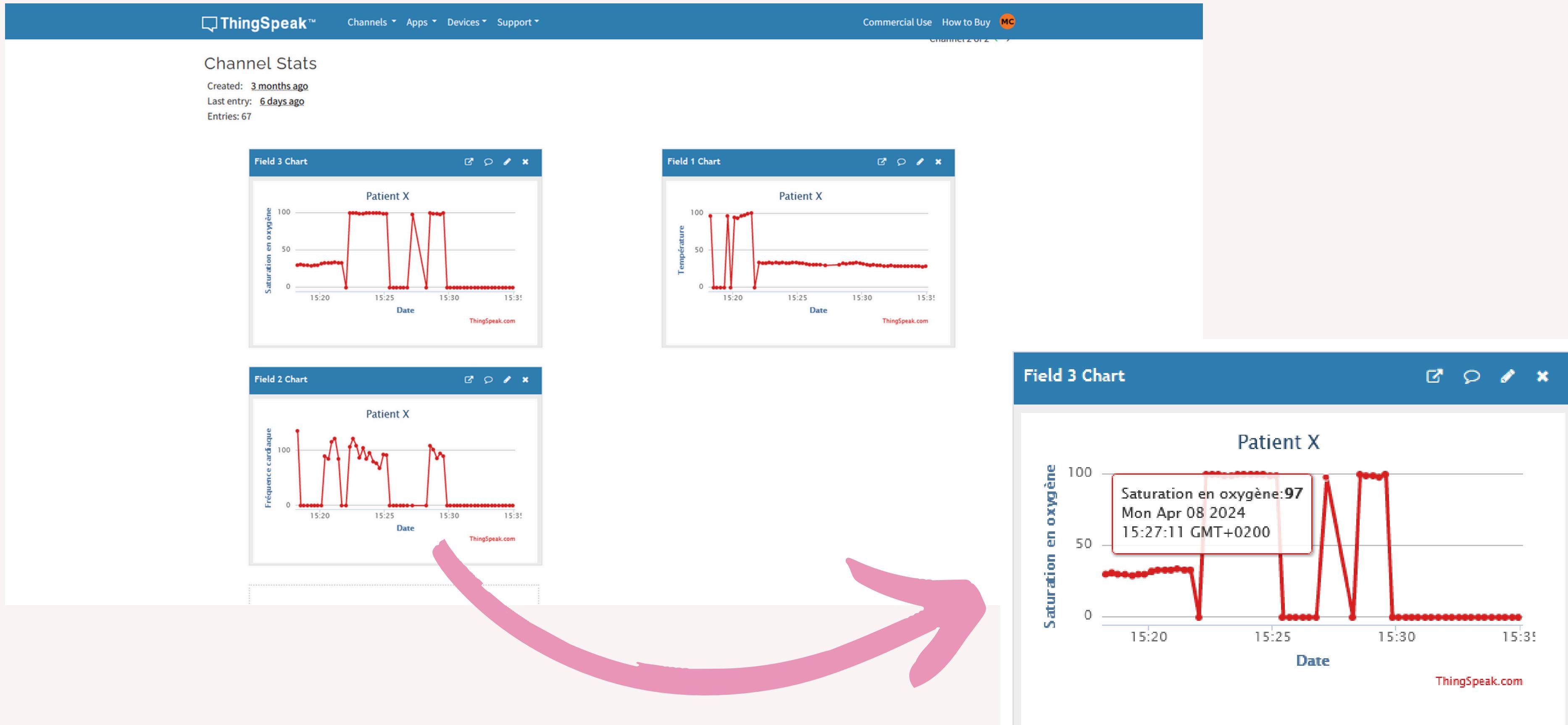
- Real-time data collection and storage
- MATLAB® analytics and visualizations
- Alerts
- Device communication
- Open API
- Geolocation data

Works With

- Arduino®
- Particle Photon and Core
- Raspberry Pi™
- Mobile and web apps
- Twitter®
- Twilio®

Description du logiciel ThingSpeak:

Visualisation de graphiques + valeurs numériques.



Descriptif du programme informatique

```
#include <SPI.h>
#include <Ethernet.h>
#include <TFT.h>
#include "DFRobot_BloodOxygen_S.h"
#include <ThingSpeak.h>

// Définition des broches pour l'écran TFT
#define cs 10
#define dc 9
#define rst 8

// Création de l'objet TFTscreen pour contrôler l'écran TFT
TFT TFTscreen = TFT(cs, dc, rst);

// Définition de la constante pour la communication I2C
#define I2C_COMMUNICATION
```

- Définition des bibliothèques
- Définition des broches des différents composants

```
#ifdef I2C_COMMUNICATION
#define I2C_ADDRESS 0x57
DFRobot_BloodOxygen_S_I2C MAX30102(&Wire, I2C_ADDRESS);
#else
// Définition des broches pour le capteur de fréquence cardiaque et de
saturation en oxygène
#if defined(ARDUINO_AVR_UNO) || defined(ESP8266)
SoftwareSerial mySerial(4, 5);
DFRobot_BloodOxygen_S_SoftWareUart MAX30102(&mySerial, 9600);
#else
DFRobot_BloodOxygen_S_HardWareUart MAX30102(&Serial1, 9600);
#endif
#endif
```

Descriptif du programme informatique

```
// Paramètres ThingSpeak
char ssid[] = "iPhone de Niama"; // SSID de votre réseau WiFi
char pass[] = "niamal234"; // Mot de passe de votre réseau WiFi
unsigned long channelID = 2402209; // ID de votre canal ThingSpeak
const char * writeAPIKey = "8SLRNOU7BL1XR69V"; // Clé API d'écriture
ThingSpeak

// Initialise la bibliothèque Ethernet avec l'adresse MAC spécifiée
byte mac[] = { 0xA8, 0x61, 0x0A, 0xAE, 0xDF, 0x56 };
EthernetClient client;
```

- Configuration des paramètres du réseau internet et du canal ThingSpeak
- Initialisation des variables

```
void setup() {
    // Initialisation de l'écran TFT
    TFTscreen.begin();
    // Définition de la couleur de fond : blanc
    TFTscreen.background(255, 255, 255);
    // Définition de la couleur d'écriture : bleue
    TFTscreen.stroke(255, 0, 0);
    // Initialisation de la communication série
    Serial.begin(115200);
    // Initialisation du capteur de fréquence cardiaque et de saturation en
oxygène
    while (false == MAX30102.begin()) {
        Serial.println("init fail!");
        delay(1000);
    }
    Serial.println("init success!");
    Serial.println("start measuring...");
    MAX30102.sensorStartCollect();
    pinMode(6, OUTPUT); //initialisation de la broche pour le buzzer

    // Initialise la connexion Ethernet et ThingSpeak
    Ethernet.begin(mac);
    ThingSpeak.begin(client);
}
```

Descriptif du programme informatique

Numéro du graphique
Variable

```
void loop() {
    // Effacer l'écran en le remplissant avec la couleur blanche (0xFFFF)
    TFTscreen.fillRect(0xFFFF);
    // Lecture des données du capteur de fréquence cardiaque et de saturation
    en oxygène
    MAX30102.getHeartbeatSPO2();
    // Affichage des données sur le moniteur série
    Serial.print("SPO2 est : ");
    Serial.print(MAX30102._sHeartbeatSPO2.SPO2);
    Serial.println("%");
    Serial.print("fréquence cardiaque est : ");
    Serial.print(MAX30102._sHeartbeatSPO2.Heartbeat);
    Serial.println(" bpm");
    Serial.print("Temperature est : ");
    Serial.print(MAX30102.getTemperature_C());
    Serial.println(" °C");

    // Envoi des données à ThingSpeak
    ThingSpeak.setField(1, MAX30102.getTemperature_C());
    ThingSpeak.setField(2, MAX30102._sHeartbeatSPO2.Heartbeat);
    ThingSpeak.setField(3, MAX30102._sHeartbeatSPO2.SPO2);

    int statusCode = ThingSpeak.writeFields(channelID, writeAPIKey);

    if (statusCode == 200) {
        Serial.println("ThingSpeak update successful.");
    } else {
        Serial.println("Problem updating ThingSpeak. HTTP error code " +
String(statusCode));
    }
}
```

Affichage des valeurs dans le moniteur série

Descriptif du programme informatique

Fréquence cardiaque :

- inférieure à 60 bpm :
 >>> signal sonore
- supérieure à 120 bpm :
 >>> signal sonore



Température :

- inférieure à 30 °C :
 >>> signal sonore
- supérieure à 35 °C :
 >>> signal sonore



```
// Vérification de la saturation en oxygène (spo2)
int spo2 = MAX30102._sHeartbeatSPO2.SPO2;
if (spo2 <= 90) {
    tone(6, 1000);
    delay(500);
    noTone(6);
} else {
    noTone(6);
}

// Vérification de la fréquence cardiaque
int beatAvg = MAX30102._sHeartbeatSPO2.Heartbeat;
if (beatAvg < 60 || beatAvg > 120) {
    tone(6, 1000);
    delay(500);
    noTone(6);
} else {
    noTone(6);
}

// Vérification de la température
int celsius = MAX30102.getTemperature_C();
if (celsius < 30 || celsius > 35) {
    tone(6, 1000);
    delay(500);
    noTone(6);
} else {
    noTone(6);
}

// Affichage des données sur l'écran TFT
TFTscreen.setTextSize(1);
TFTscreen.text("SPO2 : " + String(MAX30102._sHeartbeatSPO2.SPO2) +
" %").c_str(), 2, 20;
TFTscreen.text("Fréquence cardiaque : " +
String(MAX30102._sHeartbeatSPO2.Heartbeat) + " bpm").c_str(), 2, 40;
TFTscreen.text("Temperature : " + String(MAX30102.getTemperature_C()) +
" °C").c_str(), 2, 60;

// Pause de 15 secondes avant la prochaine lecture et envoi des données
delay(15000);
}
```

Saturation en oxygène :

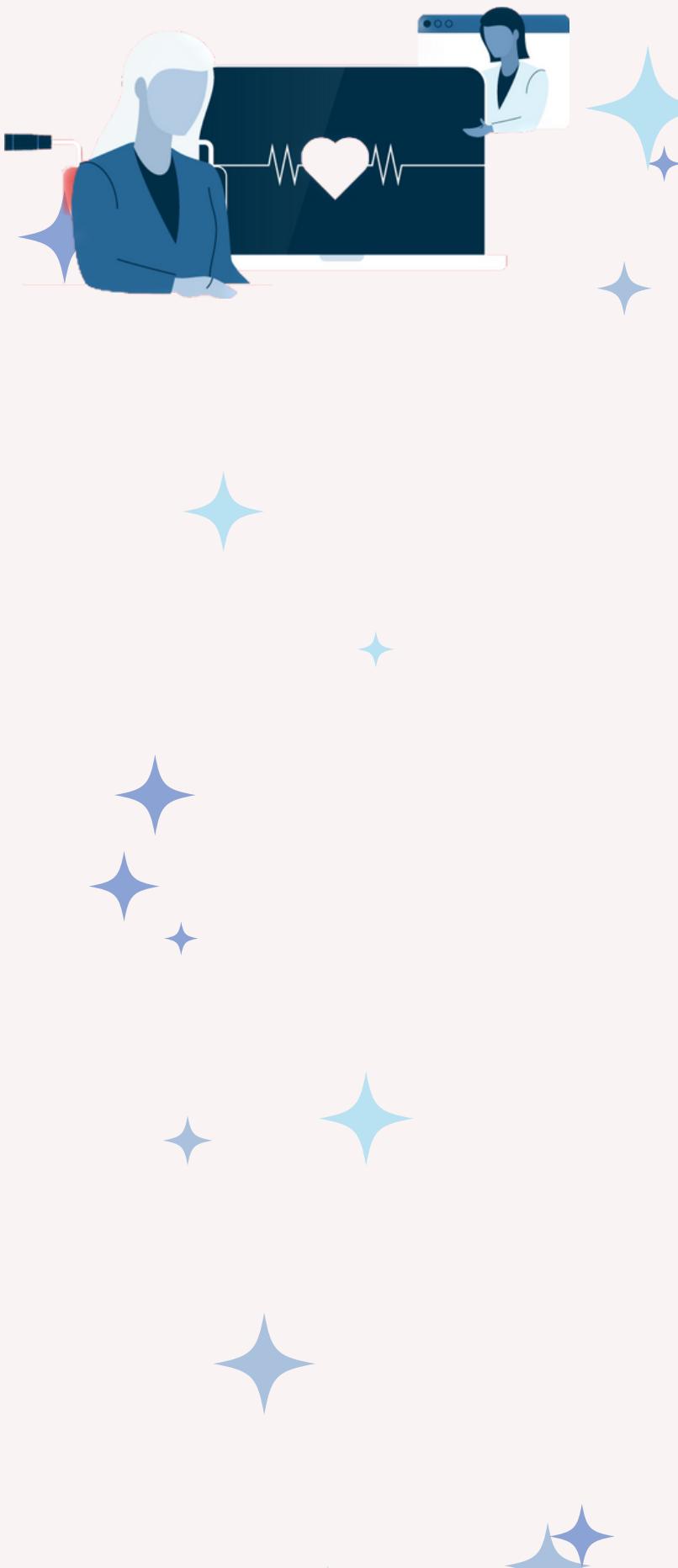
- inférieure à 90% :
 >>> signal sonore



Conclusion

Difficultés rencontrées:

- Problèmes avec la réalisation de la boîte
- Manque de capteurs médicaux
 - Code informatique trop volumineux pour ARDUINO
- Erreurs de lecture de données





Thank you for your attention
Do you have any questions ?



Composants utilisés:

<https://www.gotronic.fr/pj2-rb-tft1-8-fr-1553.pdf>

<https://www.gotronic.fr/art-capteur-de-pouls-pim438-31600.htm>

<https://www.gotronic.fr/art-ethernet-shield-2-a000024-23299.htm>

<https://www.gotronic.fr/art-buzzer-sv1407-21014.htm>

<https://www.lextronic.fr/carte-arduino-uno-dip-rev3-a000066-2474.html>