Internet of Things

Project Report: The Connected Pillbox

HERVE François

BONY Alexandre

ROBICHON Charles

EFREI

2019-2020

Summary:

Introduction

As a project of Internet of things, we were asked to create a product that would answer a need for potential customers. In order to carry out this project, we were given an Arduino starter pack box.

For our project, we decided to create a connected pillbox providing help to remind users when they need to take their medicines. This report describe our work on the project and its features.

Concept and utility

For our project, we decided to look at the issue some people may encounter when they have to take medicines quite often or with a large amount of medicines: it is always easy to forget, and it becomes an issue when you know how important these medicines are.

So our idea was to create a connected pillbox, which will help the customers and remind them when to take their pills. This pillbox will alert the users when they need to take their pills and check if the pills were taken or not. We also wanted to link the pillbox with the phone of the user in order to send him notifications and reminders.

For this project, our plan for the main features of our connected pillbox were the following:

-An alarm that would start when the user need to take his pills

-A verification system, to check if the user toke his pills

-A notification system with the phone of the user

The idea of a sophisticated pillbox was already implemented in the market, but we mostly found out that most of already existing products were really expensive and with less features. This is why our project was to create a cheaper connected pillbox that could still please the need of the customers.

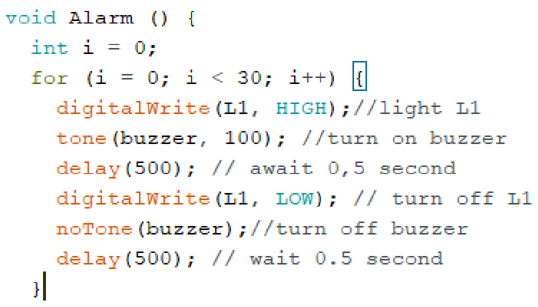
Technical description

Alarm System:

For the alarm, that is a light signal and a sound signal, we use a LED and a piezo buzzer that we branch with a parallel connection.

We created a function (Alarm()) that simply turns on and off the LED AND buzzer every half seconds.

As we repeat the action 30 times, the Alarm lasts 30 seconds

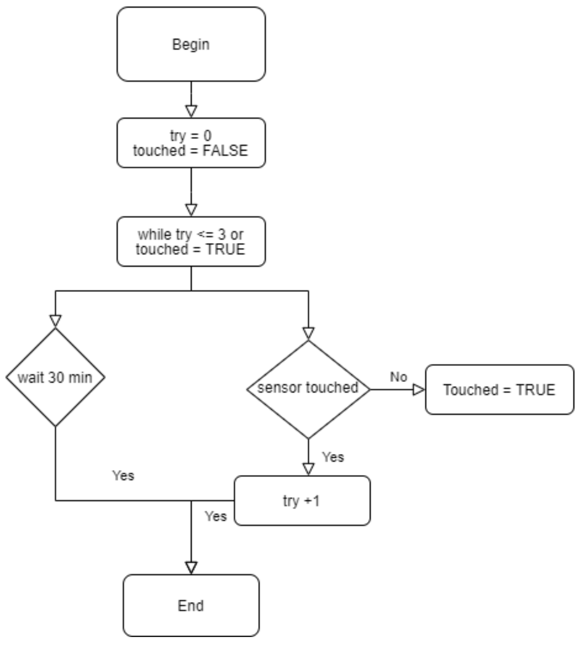


```
void Alarm () {
  int i = 0;
  for (i = 0; i < 30; i++) {
    digitalWrite(L1, HIGH);//light L1
    tone(buzzer, 100); //turn on buzzer
    delay(500); // await 0,5 second
    digitalWrite(L1, LOW); // turn off L1
    noTone(buzzer);//turn off buzzer
    delay(500); // wait 0.5 second
  }
```

Verification system:

To make sure the user took his pill, we created the function Snooze() that uses a touch sensor. The goal is that the user touches the sensor for two seconds to let the system know that he took the pill.

Activity diagram of snooze function:



There are actually no wait function for 30 min but we check that the sensor is touched every 2 seconds (this is why the user must touch it for two seconds) for 900 times (900*2=1800 seconds that is 30 minutes)

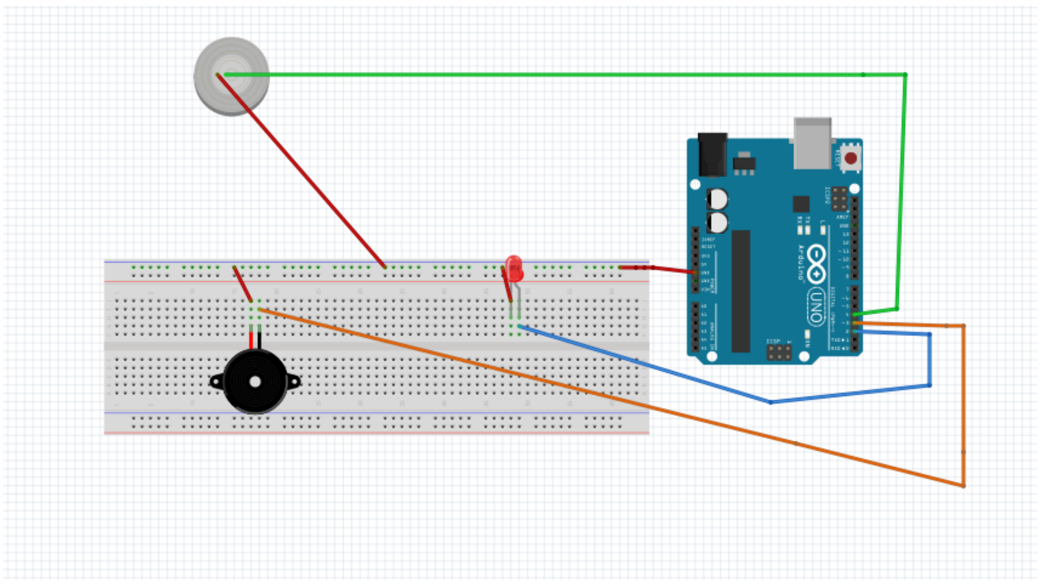
The returning of retard is used to calculate exactly the time to wait after the function to have a perfect day of waiting until the next alarm:

```
int Snooze() {
  int i = 0, j = 0;
  int retard = 0;
  bool touched = 0;
  for (j = 0; j < 3; j++) {
    for (i = 0; i < 900; i++) {
      if (digitalRead(finger) == HIGH) {
        Serial.println("Sensor is touched");
        touched = 1;
      }
      else {

      }
      delay(2000);
    }
    if (!touched) {
      Alarm();
      retard = retard+ 600*30; // retard du au temps de fonctionnement de l'alarme
    }
    Serial.println(j);
  }
  return retard;
}
```

```
void loop() //fonction principale, elle se répète (s'exécute) à l'infini
{
  Alarm();
  int retard= Snooze();
  delay(84570000-retard);
}
```

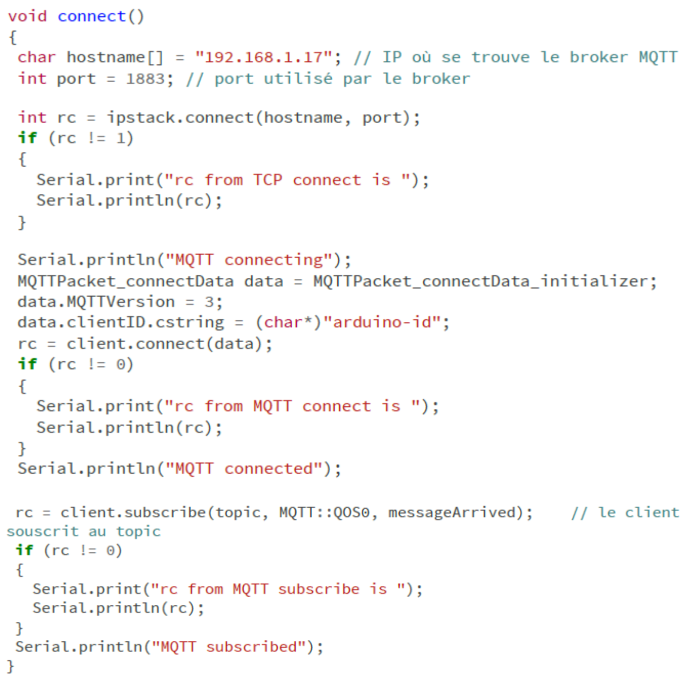
The setup is as follows:



Notification system:

Our idea was to implement a notification system that would send a message or a notification by WIFI to the user. We wanted use the ESP 32 to make this connection between the pillbox and the user's phone. However we only managed to link from the phone to the card, which is not the way we wanted.

For communication between the phone and the Arduino card, we used the MQTT protocol. For this, we have the broker installed on the computer with the code we can manually activate the alarm with phone. However the code must be personalized for each Arduino card so that the application can make the interface between the telephone and the card.



```
void connect()
{
 char hostname[] = "192.168.1.17"; // IP où se trouve le broker MQTT
 int port = 1883; // port utilisé par le broker

 int rc = ipstack.connect(hostname, port);
 if (rc != 1)
 {
   Serial.print("rc from TCP connect is ");
   Serial.println(rc);
 }

 Serial.println("MQTT connecting");
 MQTTPacket_connectData data = MQTTPacket_connectData_initializer;
 data.MQTTVersion = 3;
 data.clientID.cstring = (char*)"arduino-id";
 rc = client.connect(data);
 if (rc != 0)
 {
   Serial.print("rc from MQTT connect is ");
   Serial.println(rc);
 }
 Serial.println("MQTT connected");

 rc = client.subscribe(topic, MQTT::QOS0, messageArrived);    // le client
souscrit au topic
 if (rc != 0)
 {
   Serial.print("rc from MQTT subscribe is ");
   Serial.println(rc);
 }
 Serial.println("MQTT subscribed");
}
```

Conclusion

To conclude our project, our connected pillbox made with Arduino provide a useful and cheap reminder for customers that need help to take their medicines. With the help of an alarm and a verification, the pillbox will make sure that the user did not forget to take his pills.

This project was also a good way for us to learn and practice with the Arduino equipment, its features but also its constraints.