Digital Transformation & Cloud Services 2019
M1

Project Report

Internet of Things

eFrei
PARIS
Ecole d'ingénieurs du numérique

DEFFRADAS Thomas
TCHEUPI Xavier
WONG Kevin

Summary

I. Team Description

1. Team name
   - Team of Things
2. Team Members
   - DEFFRADAS Thomas
   - TCHEUPI KAMGA Xavier
   - WONG Kevin
3. GitHub usernames
   - Domoviye
   - KevinKyou
   - Tcheup

II. Product Description

A. Description

Today's world where IoT product are making progress through many field, we wanted to innovate a product which we be really useful for the cold chain. That's why we came up with our connected temperature sensor sends informations and data with LoRa technology.
It will be easier and faster to send and retrieve all the data from the sensor. It could be used in a cluster to allow several sensors to send the data.
The data will be send to a server where the data will be treated to have a better display.
This temperature sensor will ensure consistency, compliance, and quality control in your retail food establishment from the time products are received through storage, handling, and purchase.

Stores may use a wide variety of equipment to store goods, such as display cases, walk-in refrigerators, and freezers. As a result, there is significant opportunity to improve older monitoring techniques (such as manual data logging) to increase operational efficiency, but also to maintain better quality control over perishable and temperature-sensitive products.

Adopting a comprehensive food safety management system gives visibility into an otherwise unconnected grocery store. Remote monitoring and alerting systems provide robust quality maintenance measures, and prevent spoilage during transit and storage in-store.

B. Existing solutions

There is already some similar functionalities that already exist but the format is not exactly the same, there are big box for retail, but it uses Wi-Fi for an amount of data too low for the power consumptions of the object.

C. IoT Solution

List of components :

- LoRa module
  - LINK : (1) 15,99 € https://www.amazon.fr/MakerHawk-Transmission-Communication-Performance-Anti-interf%C3%A9rence/dp/B07GQQ5Q4W/ref=sr_1_5?__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=3V251GF0XG4U1&keywords=lora+arduino&qid=1574259399&sprefix=lora+ardui%2Caps%2C153&sr=8-5
  - ESP32 can be used if it's available in the starter kit from EFREI
- Temperature sensor DS18B20
  - LINK : (2) 7.49 € https://www.amazon.fr/AZDelivery-DS18B20-Capteur-Temp%C3%A9rature-num%C3%A9rique/dp/B01MZG48OE/ref=sr_1_6?keywords=DS18B20&qid=1574257848&sr=8-6&th=1

- Arduino board (Available in the Arduino Starter Kit)
- Display (Available in the Arduino Starter Kit)
- Led/Buzzer and small pieces (Available in the Arduino Starter Kit)
- ESP 32 (Available in the Arduino Starter Kit)

Our solution has a low power consumption and uses the right network to send data. That way, it is easily possible to add more devices to work in a cluster. Possible use in retail and cold chain industry.

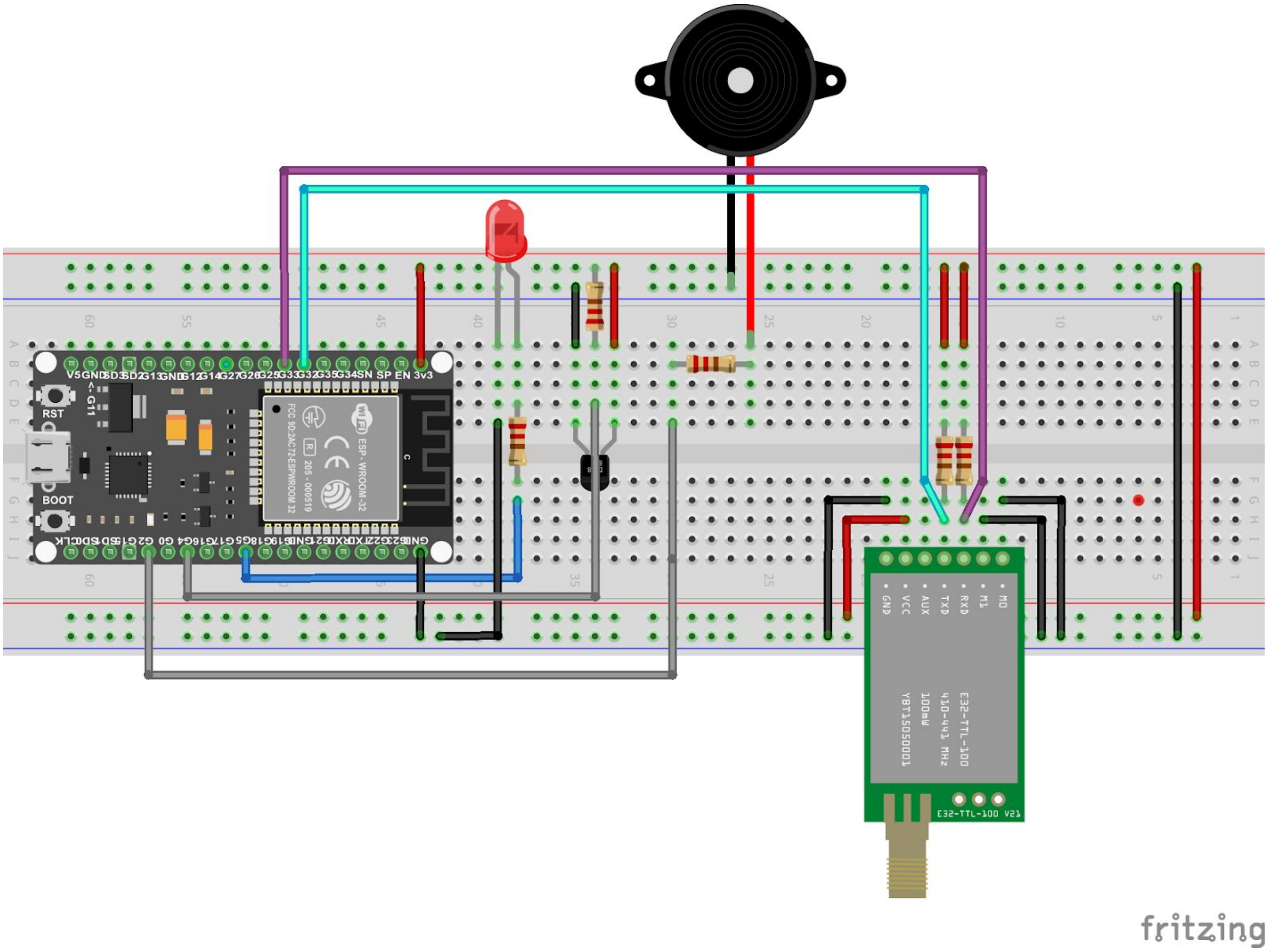
D. Norms and regulations

There is a norm linked to temperature sensor implied in food retail or retail involved in consumable products by the World Health Organization
LINK :
https://www.who.int/medicines/areas/quality_safety/quality_assurance/TS-temp-monitoring-final-sign-off-a.pdf

“1.1.1 Temperature monitoring systems Air temperature monitoring systems and devices should be installed in all temperature-controlled rooms, cold rooms, freezer rooms, refrigerators and freezers used to store TTSPPs. Electronic sensors should be accurate to ± 0.5°C or better. Sensors should be located in areas where the greatest variability in temperature is expected to occur within the qualified storage volume and they should be positioned so as to be minimally affected by transient events such as door opening.”

The temperature sensor DS18B20 is perfect for our project since it has the requirements from the WHO guidance.

E. Fritzing schema



F. Server side

- ThingsBorard solution

ThingsBoard is an open-source IoT platform, thanks to standard IoT protocols (MQTT, CoAP and HTTP), it provides IoT devices management, data collection, processing and virtualization.

- First step :
  In ThingBorard demo we can create assets, and devices to test and manage the temperature data sending, reception and the treatment.
  - Sending
    The virtual device can generate random telemetry data which are sent in a message by the POST method.
  - Reception and treatment
    The root rule chain receive messages, treat and can display them in several widgets

- Second step :
  This part allows us to connect to our temperature sensor with the LoRa module to send messages via a Post method the current temperature telemetry every 5 or 10 seconds. ThingsBoard provides tokens for data reception on devices.

G. Problems

- Material issues
  - At the beginning of the project, the temperature sensor had a connection problem.
  - Near the end of the project the temperature sensor had a malfunction

- Code issues
  - For the buzzer, we realize that the sound didn't came out (we were using tone() to change the frequency of the buzzer).
  - The LoRa part has some problems with the librairies

- Servers issues
  - The connection between the LoRa module and ThingsBoard is not implemented. We were planning to use it with a token generated for the ESP32 code. This way it can act as a socket with our ThingsBoard implementation to send data for the different widgets.

H. Problems solutions

- Material solutions
  - For the temperature sensor problem part, we change the wires places so it would work properly then when it started to malfunction we checked the DS18B20 DataSheets to compare the errors codes. The sensor has an alarm signaling feature. The second code was different (-127°C) so it wasn't a wiring problem. Our unit was working properly in the beginning and we were able to get the temperature from the sensor even in the end.

- Code solutions
  - For the buzzer we searched and we found that because we were using the ESP32 it didn't recognize the function tone(), so instead we used ledc…() functions in order to vary the frequency, which will affect the type of sound produced.
    Details :
    - We started by declaring some global variables needed to control the PWM functionality of the ESP32.
    - We set the initial frequency, the channel of the PWM and the resolution of the duty cycle specification.
    - We callled the ledcWriteTone() function, passing as inputs the PWM channel and the frequency to set
    - To set the duty cycle, we just need to call the ledcWrite() function, passing as input both the channel and the duty cycle value to set.

- Servers solutions :

  For the server, we didn’t test the process to connect Thingboard and the device. The complexity was to sending data every 5 or 10 secondes. This process block the other application and might block the LoRa module resource and can create some malfunction.