

# BOITE AUX LETTRES CONNECTEE

M2M 2016/2017

**Clément BROCARD**  
**Oswaldo MORENO**





# SOMMAIRE

<b>INTRODUCTION</b>	<b>3</b>
<b>MATÉRIEL UTILISÉ</b>	<b>3</b>
<b>TOOLCHAIN</b>	<b>7</b>
<b>DIFFICULTÉS RENCONTRÉES</b>	<b>8</b>
<b>ARCHITECTURE</b>	<b>8</b>
<b>CODE UTILISÉ</b>	<b>9</b>
<b>PROTOTYPE</b>	<b>10</b>
<b>NODE-RED RASPBERRY</b>	<b>11</b>
<b>NODE-RED CLOUD</b>	<b>11</b>
<b>GRAPHANA</b>	<b>11</b>
<b>CONCLUSION</b>	<b>12</b>



# INTRODUCTION

Dans le cadre d'une tendance croissante, les systèmes embarqués sont de plus en plus nombreux. C'est la raison pour laquelle les domaines d'applications sont très vastes (Télécommunications, transport, Multimédia,...)

De nos jours, l'offre matérielle et logicielle proposée par les entreprises à travers le monde permet aux utilisateurs d'obtenir d'une façon abordable et flexible toutes les outils nécessaires pour réaliser ses propres développements.

L'objectif de ce projet est de mettre en place une infrastructure minimale qui permet de collecter des données à travers des capteurs distribués. Notamment via l'utilisation des cartes de développement STM32 Nucleo, Raspberry Pi et l'émission/réception sur le réseau Lora RF.

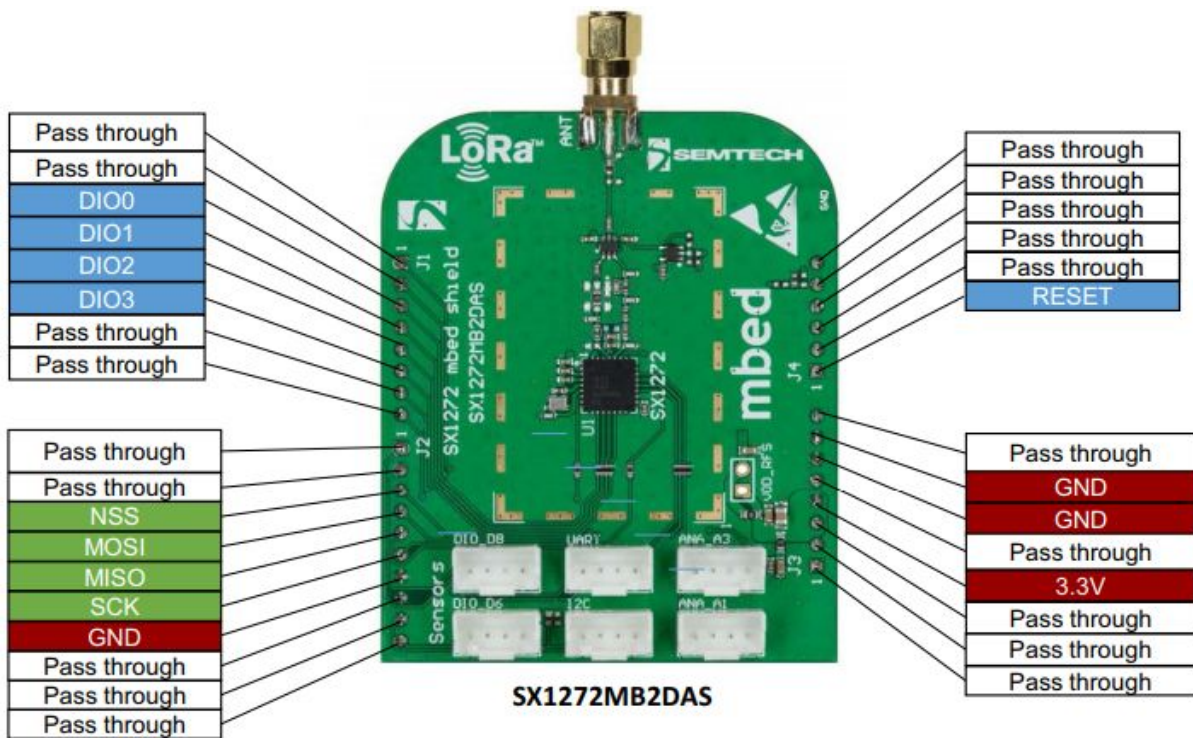
Ce projet a été réalisé en séance du cours M2M (Machine to Machine) sous la direction du professeur Didier Donsez.

## MATÉRIEL UTILISÉ

L'émetteur/récepteur SX1272 Lora modem fournit une portée de communication ultra longue, tout en minimisant la consommation d'énergie.

Nous avons utilisé un modèle monté sur un "shield" prêt à être couplé avec la carte STM32.





La carte de développement et programmation embarquée ST Microelectronics NUCLEO-L073RZ , qui possède une bonne puissance de calcul ainsi que des nombreux ports d'entrées/sorties.



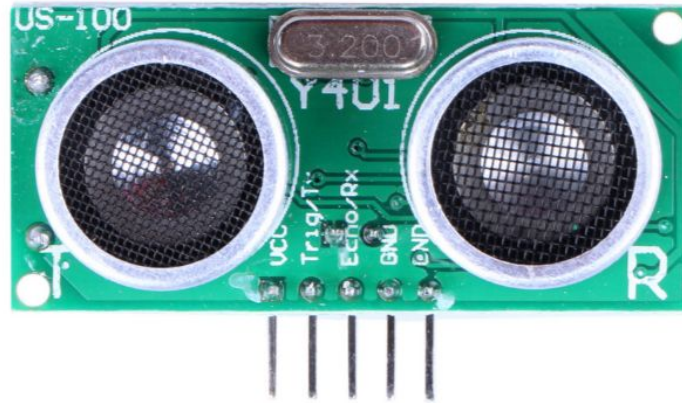
NUCLEO-L073RZ





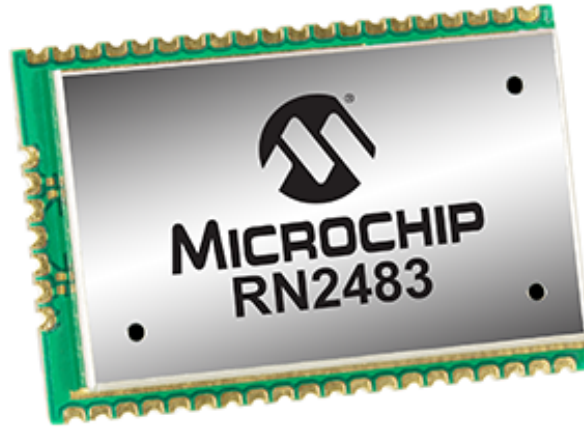


US-100 Module Carte Capteur Détecteur Ultrason Mesure Distance Ultrasonique, permet de mesurer la distance sans contact de 2cm ~ 4,5m, avec large gamme de tension d'entrée de 2,4 à 4,5, la fonction GPIO statique.





La puce microchip RN2483, couplée en série avec le Raspberry PI. Cette puce a pour avantage d'être certifiée grâce à un module intégré. Elle est contrôlée avec des commandes via un terminal série.



Le Raspberry PI avec une distribution Raspbian.





# TOOLCHAIN

Nous avons utilisé les outils suivant au long du projet:

## 1 - Keil (MDK-ARM)

Uniquement compatible avec les systèmes windows. Keil est un kit de développement logiciel complet pour microcontrôleurs, repose sur  $\mu$ Vision et prend en charge les dispositifs Cortex-M.

**A noter que l'outil STLink ne marche pas sur Windows.**

## 2 - Tera-term

Logiciel libre qui permet émuler les différents types de terminaux informatiques. Il prend en charge telnet, SSH 1 et 2 et les connexions de port série. Utilisé pour afficher les données envoyés ou recus.

## 3 - MBED compiler

<https://developer.mbed.org/compiler>

Espace de travail pour réaliser du développement en ligne pour les dispositifs embarqués, Il permet d'utiliser des microcontrôleurs ARM.



Dans une machine physique (Linux ou Mac) une machine virtuelle (VirtualBox ou VmWare) tourne. Une système d'exploitation Windows 7 exécute le kit de développement KEIL, uVision et Teraterm.



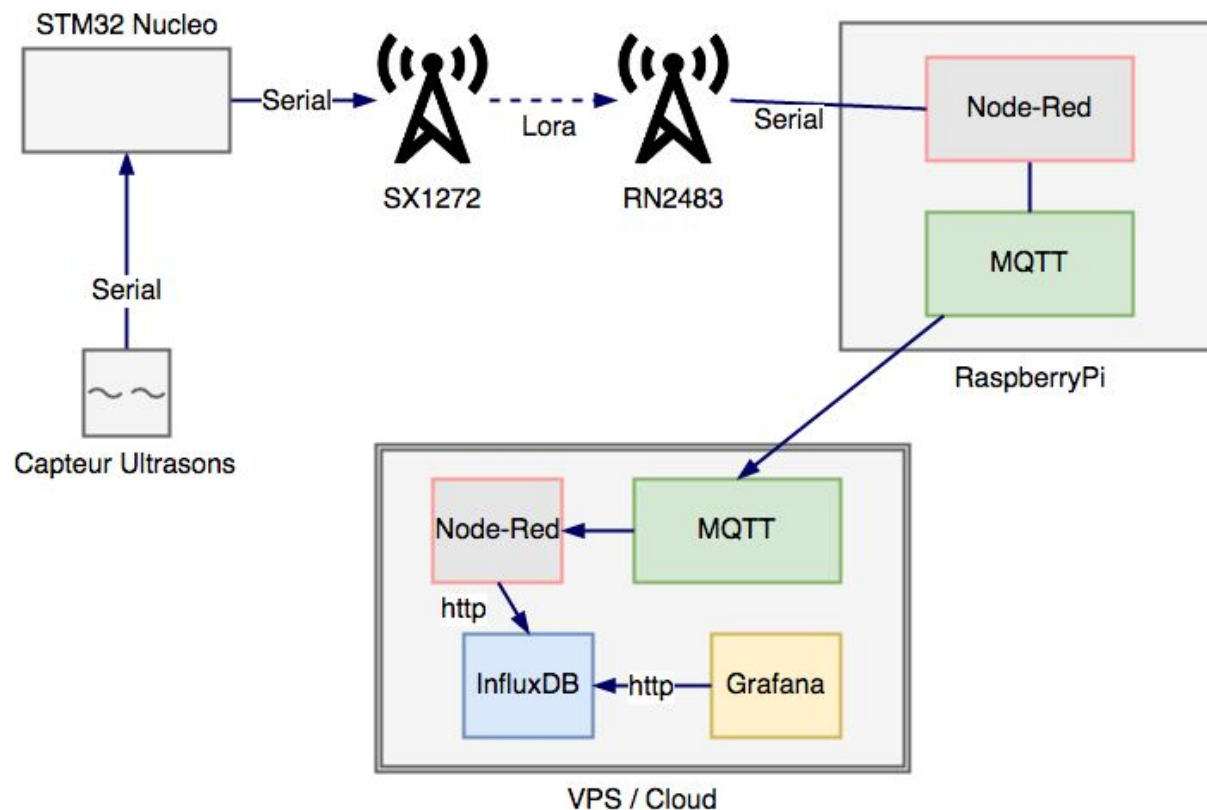
## DIFFICULTÉS RENCONTRÉES

D'abord, le premier capteur ultrason utilisé était en panne, un des pins ne marchait pas, nous avons alors passé environ 1 semaine là-dessous.

Un autre point important est l'utilisation des technologies Lora, nous n'avons pas trouvé sur Internet assez d'exemples en comparaison à d'autres technologies pour mettre en pratique et pour réaliser des tests.

Le kit de développement STM32 ne fonction que sous Windows, un contrainst pour nous car nous utilisons des systèmes d'exploitation basées sous UNIX.

## ARCHITECTURE



Ce schéma représente l'architecture du système, avec un flot de données allant du capteur ultrasons et de la carte STM32 vers le Raspberrypi en réseau sans fil Lora, puis la transmission en MQTT vers la plateforme cloud (ici un serveur virtuel hébergé chez PHPNET France).

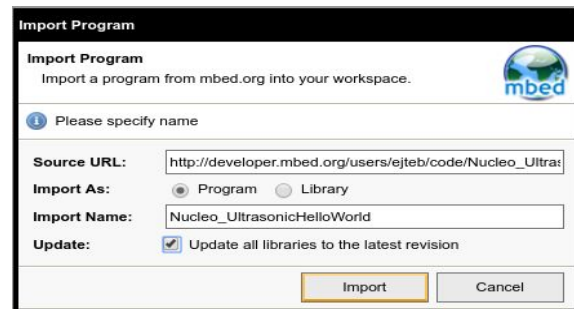




# CODE UTILISÉ

Pour la partie Lora (capture des données) du code existant dans le lien ci-dessous est utilisé. <https://developer.mbed.org/components/HC-SR04/>

Note : pour bien compiler le projet il faut d'abord importer toutes les librairies. Cliquer cage **"Update all libraries to the latest revision"**



Le main est défini comme suit:  
main.cpp

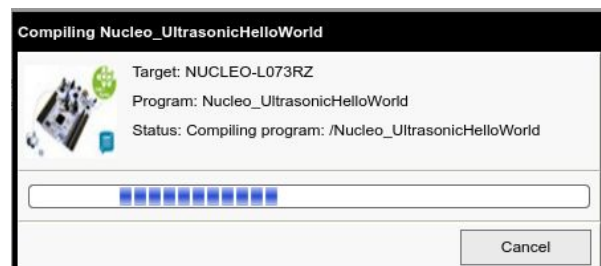
```
#include "mbed.h"
#include "ultrasonic.h"

void dist(int distance)
{
    //put code here to happen when the distance is changed
    printf("Distance changed to %dmm\r\n", distance);
}

ultrasonic mu(D8, D9, .1, 1, &dist);    //Set the trigger pin to D8 and the echo pin to D9
                                        //have updates every .1 seconds and a timeout after 1
                                        //second, and call dist when the distance changes

int main()
{
    mu.startUpdates();//start mesuring the distance
    while(1)
    {
        //Do something else here
        mu.checkDistance();    //call checkDistance() as much as possible, as this is where
                               //the class checks if dist needs to be called.
    }
}
```

Finalement il faut compiler (CTRL+D) un fichier .BIN est créé, celui-ci doit être copié directement dans la carte STM32 connecté dans un port serial spécifique.





# PROTOTYPE

La réalisation de ce projet a abouti à la création d'un prototype d'une boîte aux lettres qui détecte avec un sensor de mouvement lorsqu'une personne s'approche, puis cette donnée est envoyée vers un dispositif remote en utilisant le réseau lorawan.

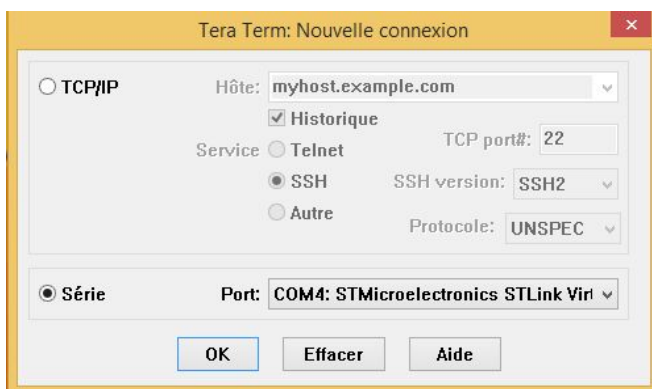
Toutes les données sont remontées dans un serveur web Node-red pour créer un scénario adapté à notre projet. Toutefois les données peuvent être stockées dans un serveur dédié virtuel VPS et visualisées avec Grafana via le connecteur MQTT.

La première étape consiste à exécuter la détection de mouvement:

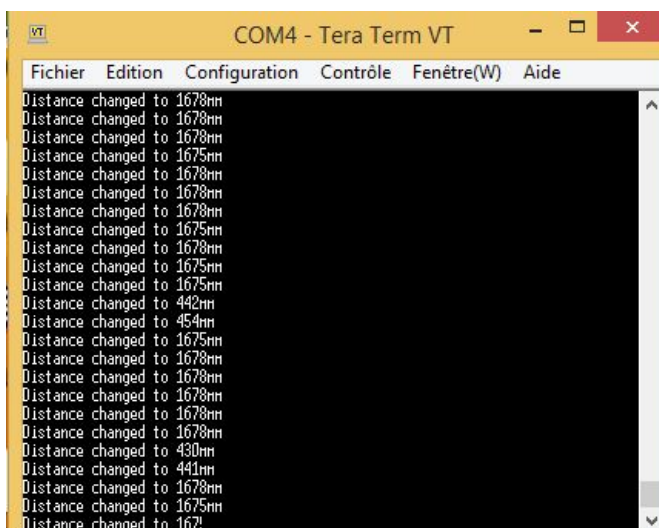
Le fichier .BIN doit être copié dans la carte STM32 mentionnée précédemment.

Lancer TeraTerm 

Sélectionner le port serial > OK

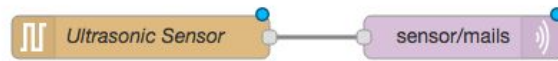


La détection de mouvement s'affiche comme suit

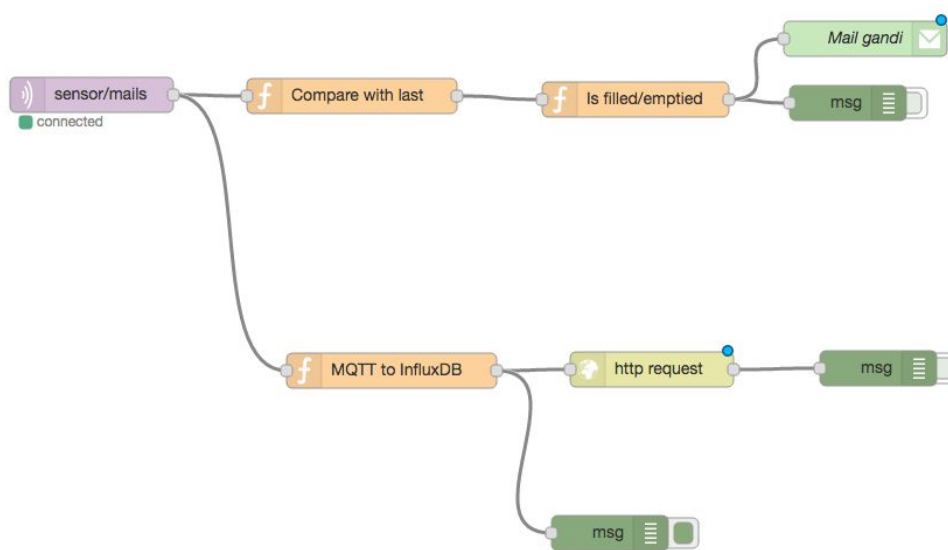




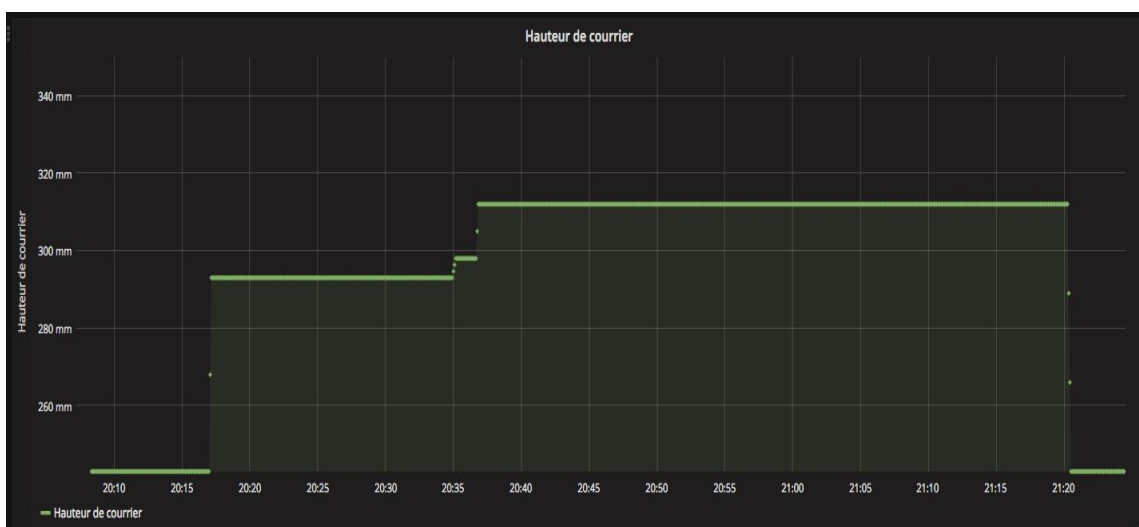
# NODE-RED RASPBERRY



# NODE-RED CLOUD



# GRAPHANA





## CONCLUSION

Dans un premier temps nous avons découvert le panorama actuel des différentes technologies et cartes utilisés dans le domaine des systèmes embarqués... etc

Dans un second temps, l'utilisation des technologies très répandues comme docker, mosquito, etc... nous a permis de maîtriser et améliorer nos compétences pour monter un projet de but à but assez rapidement.

De plus, ce projet nous a permis aussi de mettre en pratique tous les concepts appris en cours, notamment les aspects de génie logiciel, la gestion des projets, IOT, et l'agilité.

Finalement, même si le temps pour réaliser ce projet est court, le résultat est accord à nos atteints.

