

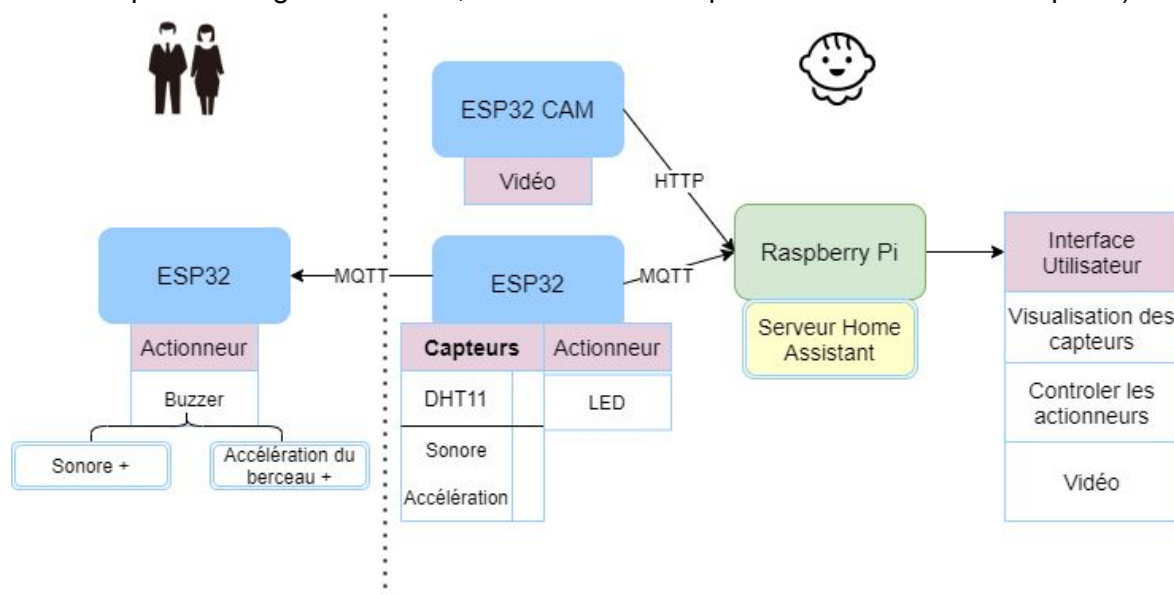
Projet IOT: Crazy Baby

- Version 1 : basée sur Home Assistant

Généralité du projet

Notre projet vise à réaliser la fonction de surveillance des bébés, à l'aide de **caméra** et de certains **capteurs** et **actionneurs**.

Les parents peuvent visualiser la surveillance de la vidéo du bébé en temps réel, ainsi que les paramètres environnementaux (tels que la température et l'humidité) où se trouve dans la chambre du bébé, et l'accélération du berceau. Lorsque les pleurs continus du bébé ou le mouvement anormal du berceau sont détectés, les parents reçoivent un rappel sonore à distance (via un buzzer). De plus, les parents peuvent également contrôler à distance les lumières de la chambre de bébé via la plateforme (parce que nous n'avons pas pu acheter un interrupteur intelligent/connecté, nous utilisons temporairement une LED à la place).



A. Choix du serveur / plateforme

Home Assistant est un logiciel d'automatisation et de gestion d'objets connectés. Il est gratuit, récent et populaire et est simple à mettre en œuvre.



Home Assistant

<https://www.home-assistant.io/>

Il nous a fourni un contrôleur central en tant que serveur pour traiter et envoyer des données. Il suffit d'utiliser le script `.yaml` pour ajouter le périphérique ou l'entité correspondante. Il est équipé d'une interface utilisateur Web, et nous pouvons également y ajouter des appareils directement.

Les actions pour les appareils et les entités peuvent être contrôlées via l'interface utilisateur Web de Home Assistant à distance.

MQTT support:

<https://www.home-assistant.io/integrations/mqtt/>

<https://home-assistant.cc/component/mqtt/>

Home Assistant prend en charge le protocole MQTT et divers brokers MQTT. Home Assistant a un MQTT broker intégré (HBMQTT), ce qui suit est les informations du broker:

<https://hbmqtt.readthedocs.io/en/latest/>

Setting	Value
Broker	localhost
Port	1883
Username	homeassistant
Password	HA API pwd(123456789)
Websocket	8080

B. Installation et configuration du home assistant

Premièrement, on a installé un Raspbian(version Lite) OS sur le Raspberry Pi.

Ensuite, on a installé le Home Assistant Core, les étapes comprises:

- Installation des indépendances
- Création d'un compte de Home Assistant
- Création d'un environnement virtuel

Les instructions détaillées se trouvent dans le lien ci-dessous.

<https://www.home-assistant.io/installation/raspberrypi#install-home-assistant-core>

Démarrer le serveur Home Assistant

```
pi@raspberrypi:~ $ sudo -u homeassistant -H -s
homeassistant@raspberrypi:/home/pi $ cd /srv/homeassistant
homeassistant@raspberrypi:/srv/homeassistant $ source bin/activate
(homeassistant) homeassistant@raspberrypi:/srv/homeassistant $ hass
2021-02-13 18:25:49 ERROR (MainThread) [metno] https://aa015h6buqvi86i1.api.met
.no/weatherapi/locationforecast/2.0/complete returned
2021-02-13 18:25:49 WARNING (MainThread) [homeassistant.bootstrap] Support for t
he running Python version 3.7.3 is deprecated and will be removed in the first r
elease after December 7, 2020. Please upgrade Python to 3.8.0 or higher.
```

Ensuite, on visite l'interface d'utilisateur via l' IP et le port du serveur(Raspberry Pi).

<http://192.168.43.62:8123/>

Dans le fichier de configuration configuration.yaml , on peut faire les configuration de MQTT, y compris, les paramètres à publier ou subscriber, ainsi que la configuration du broker. Ici on utilise le broker intégré HBMQTT.

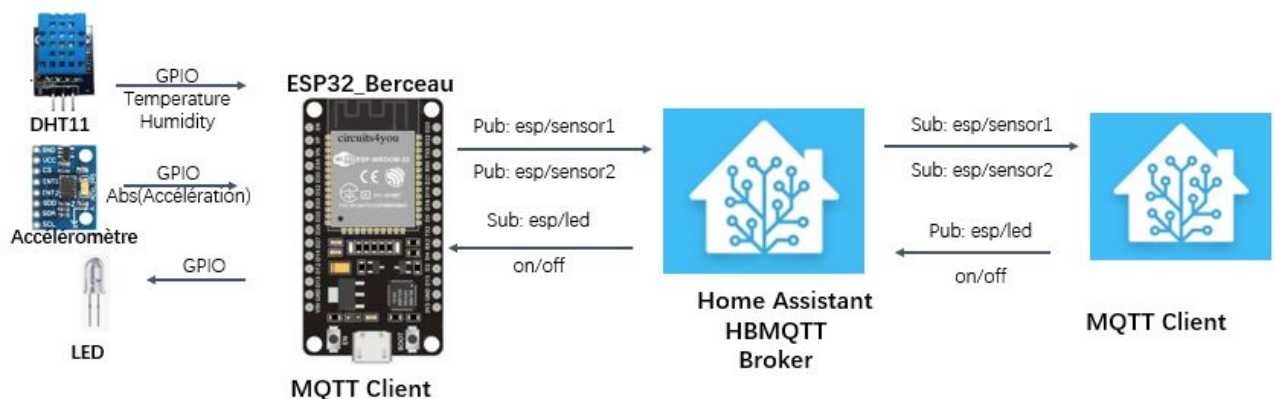
```
mqtt:
  broker: 192.168.43.62
  port: 1883
  client_id: homeassistant
  keepalive: 60
  username: homeassistant #mqtt username
  password: 123456789 #mqtt password
  discovery: true
  tls_insecure: false
```

C. Communication entre les appareils/entités

Nous avons écrit un serveur de streaming vidéo pour le ESP32 CAM en utilisant le protocole HTTP. Nous pouvons accéder au serveur par une adresse IP.

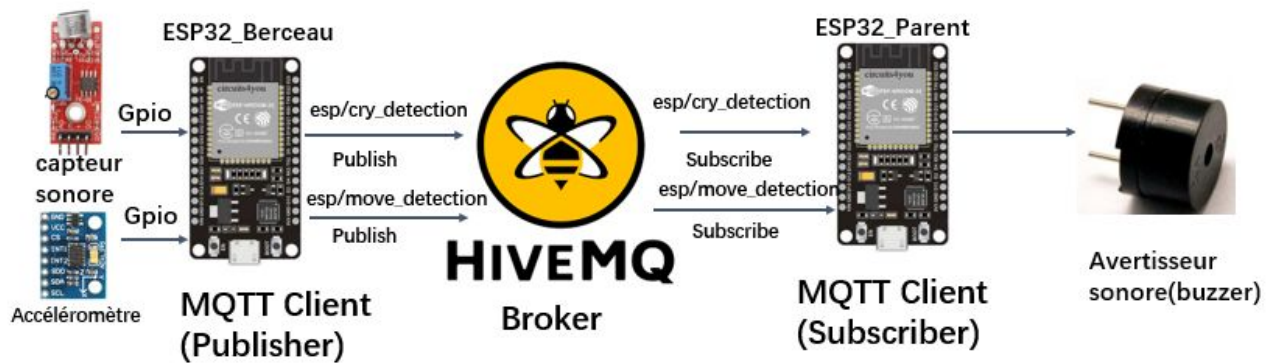
On transfère tous les autres paramètres par le protocole MQTT.

Les figures ci-dessous démontrent le structure du protocole MQTT entre l'ESP32_berceau et Platform Home Assistant, l'ESP32_berceau et l'ESP32_parent .



Communication entre l'ESP32_berceau et Platform HA

Les capteurs DHT11 et accéléromètre mesurent la température , l'humidité et l'accélération du berceau et les transfèrent au ESP_berceau. L'ESP_berceau publie les topics qui contiennent les paramètres correspondants au broker HBMQTT, intégré au Home Assistant. Ensuite, par notre configuration, notre compte Home Assistant sert comme un MQTT client, qui peut subscribe les topic, et les affiche sur le page web, on peut y accéder par l'adresse IP du Raspberry Pi.



Communication entre l'ESP32_berceau et l'ESP32_Parent

Le détection du pleurs et le tremblement du berceau sont réalisés par le ESP32_Berceau. Lorsque les valeurs dépassent certaines valeurs (seuil), l'ESP32 publie les topics correspondants au Broker HiveMQ. L'ESP32_Parent se trouve dans le part du parent, il subscribe ces topics. Lorsqu'il reçoit un message, l' avertisseur sonore est activé.

D. Réalisation des fonctions en détail

Les capteurs utilisés:

Capteur des Température et Humidité(DHT11):

<https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf>

Accéléromètre (GY521):

<https://xtcomp.co.za/image/catalog/GY-521.pdf>

Détecteur Sonore (GT1146):

<https://www.gotronic.fr/pj2-35240-2301.pdf>

Uploader vers	Nom du code	Fonction réalisée	Protocole	
ESP32 CAM	homeassista nt_video.ino	Initialisation du module vidéo: <i>ESP32 CAM</i> Pour <i>Home assistant</i>	http	Accès via adresse IP
ESP32_berceau	homeassista nt_sensor.in o	Dans le sens montant vers home assistant: <i>DHT11, accéléromètre, détecteur sonore</i> Dans le sens descendant: activer <i>led</i> depuis une interface web	mqtt	Broker: homeassistant
ESP32_parent	parent_recei ve.ino	Réception du signal <i>buzzer</i> du côté parent	mqtt	Broker: hivemq

Description des codes des ESP32s






On configure les topics à publish ou subscribe dans le fichier configuration.yaml dans le part de Home Assistant.

```
sensor:
  - platform: mqtt
    state_topic: "esp/sensor1"
    name: 'Temperature'
    unit_of_measurement: 'C'
    value_template: '{{ value_json.Temperature }}'
    device_class: temperature
  - platform: mqtt
    state_topic: "esp/sensor1"
    name: 'Humidity'
    unit_of_measurement: '%'
    value_template: '{{ value_json.Humidity }}'
    device_class: humidity
  - platform: mqtt
    state_topic: "esp/sensor2"
    name: 'Accelerate'
    unit_of_measurement: 'g'
    value_template: '{{ value_json.Accelerate }}'
```

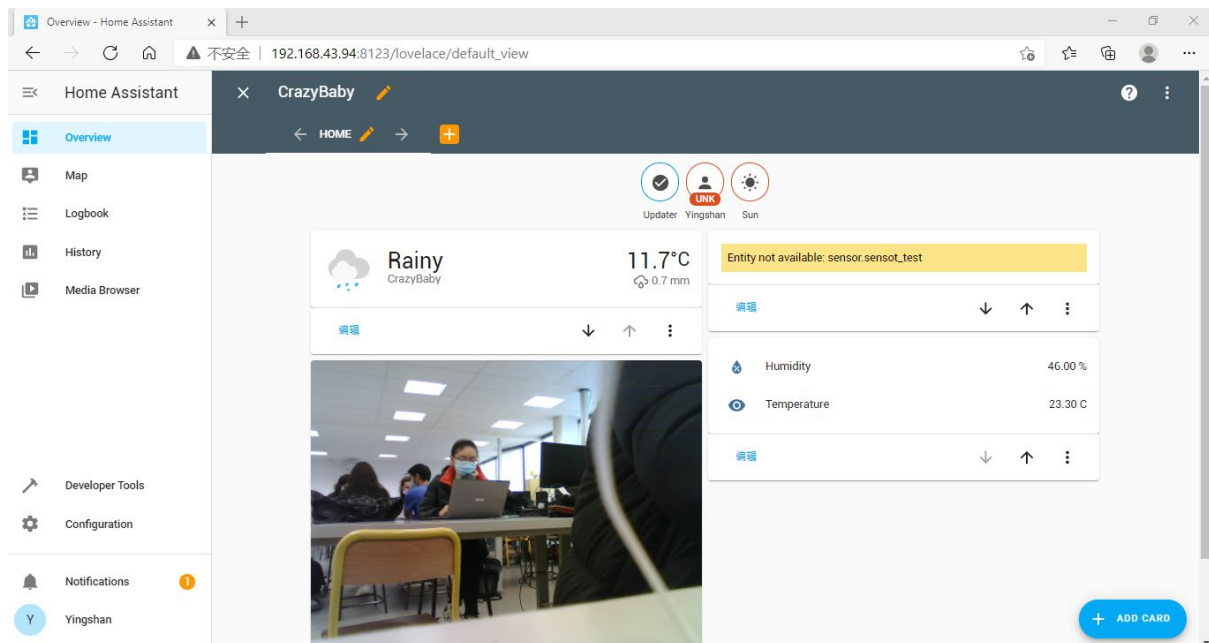
Les topics à subscribe

```
switch:
  - platform: mqtt
    command_topic: "esp/led"
    name: "Bedroom Light"
    payload_on: "ON"
    payload_off: "OFF"
```

Les topics à publish

	Humidity	24.00 %
	Temperature	23.60 C
	Accelerate	0.12 g
	Bedroom Light	

Ensuite, dans l'interface utilisateur, on peut voir les paramètres et contrôler le Led dans la chambre.



Interface d'Utilisateur de Home Assistant

A la fin, l'Interface d'Utilisateur l'aire comme ça, on connecte le fenêtre de vidéo avec l'IP du serveur de streaming vidéo.

- Version2 : basée sur le serveur écrit en python

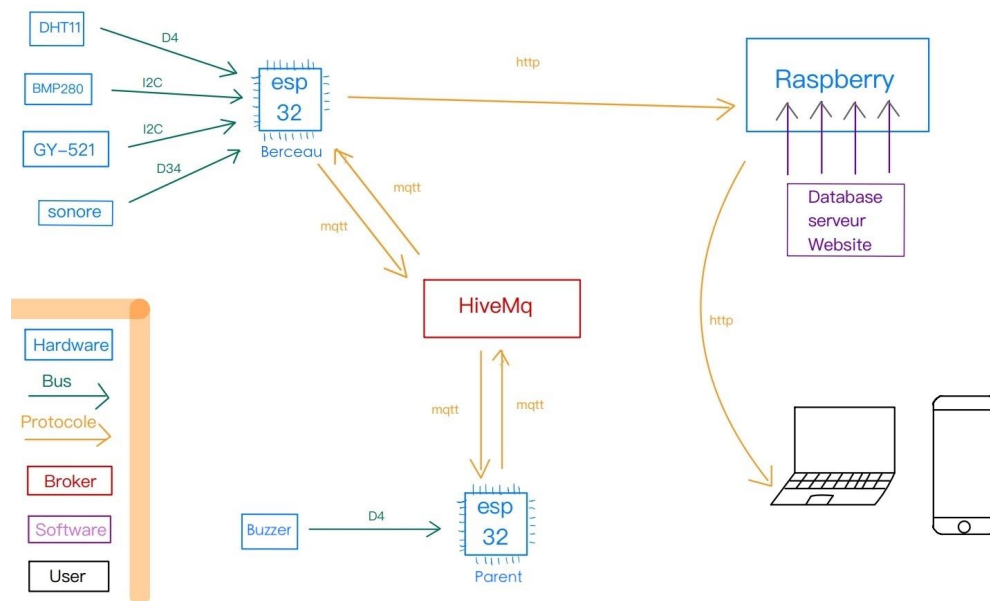
La première version a implémenté toutes les fonctions. Par ailleurs, nous avons également implémenté une deuxième solution basée sur le serveur écrit en nous-mêmes. C'est une pratique du contenu appris dans la classe IOT. Ces deux versions sont indépendantes l'une de l'autre. La partie matérielle reste inchangée dans la deuxième méthode de réalisation de notre projet. Seulement le serveur et le choix du protocole dans certaines parties de communication ont changé.

● Spécification des codes

Hardware	Software	Fonction	Protocol	
ESP32 berceau	Version2.ino	Transmission des données du capteur vers serveur, indiquer l'autre esp32 quand le bébé pleure ou bouge: <ul style="list-style-type: none"> • DHT11, • GY-521 • BMP280, • Sonore • led 	http(pour les capteur) et mqtt(pour l'alarme)	Accès via adresse IP Broker: HiveMQ

ESP32 du parent	test.ino	Recevoir le signal d'alarme <ul style="list-style-type: none"> Buzzer 	mqtt	Broker: HiveMQ
Raspberry Pi	serveur.py CrazyBaby.db	Recevoir les données du capteur, les sauvegarder dans database, comprendre les website	http	Accès via adresse IP

● Schéma du système



Référence

Introduction de MQTT de Homeassistant: <https://home-assistant.cc/component/mqtt/>
ESP32-CAM: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf
Capteur des Température et Humidité(DHT11): <https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf>
Capteur d'accéléromètre (GY521): <https://xtcomp.co.za/image/catalog/GY-521.pdf>
Capteur Sonore (GT1146): <https://www.gotronic.fr/pj2-35240-2301.pdf>