

HackatH2On IoT

La telelectura al servei de la ciutadania, la ciutat, i el medi ambient.



Aigües de
Barcelona



Innovació

Amb la col·laboració de: **CETAQUA**
CENTRO TECNOLÓGICO DEL AGUA

AllWize

HackatH₂O_n IoT

Taller amb AllWize

Xose Pérez



**Aigües de
Barcelona**



Innovació

Amb la col·laboració de:

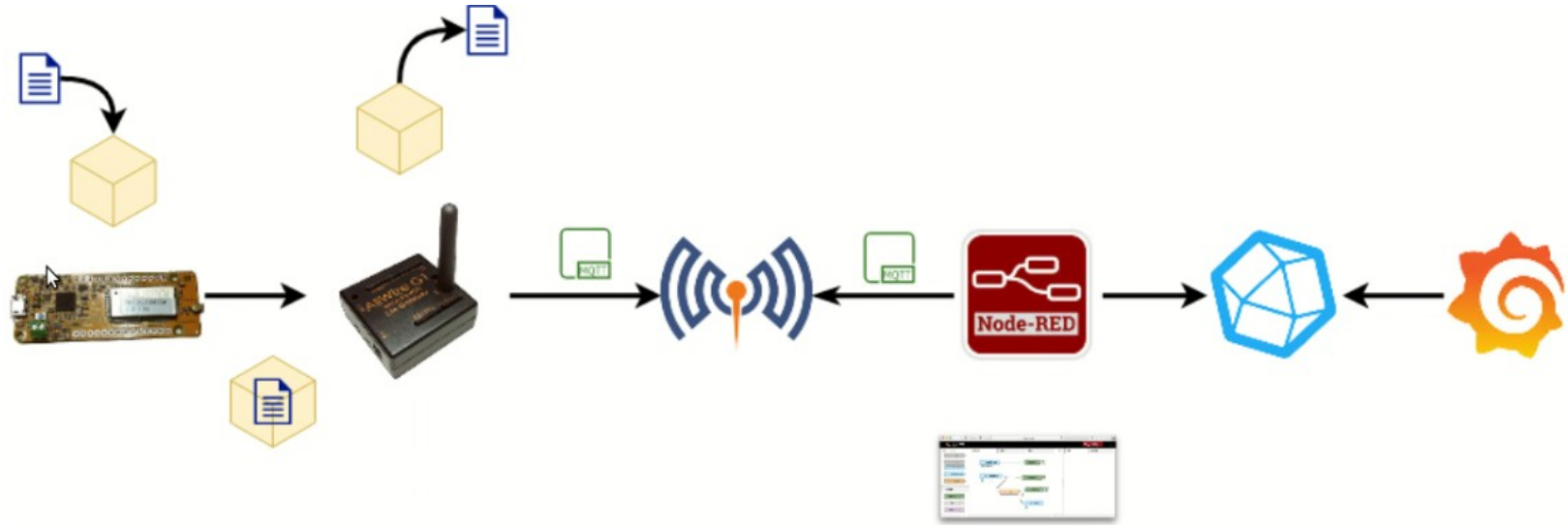
CETAQUA
CENTRO TECNOLÓGICO DEL AGUA

AllWize

Demostració

Demostració de com muntar una petita xarxa privada connectada a un servidor fent servir MQTT.

Arquitectura de la solució



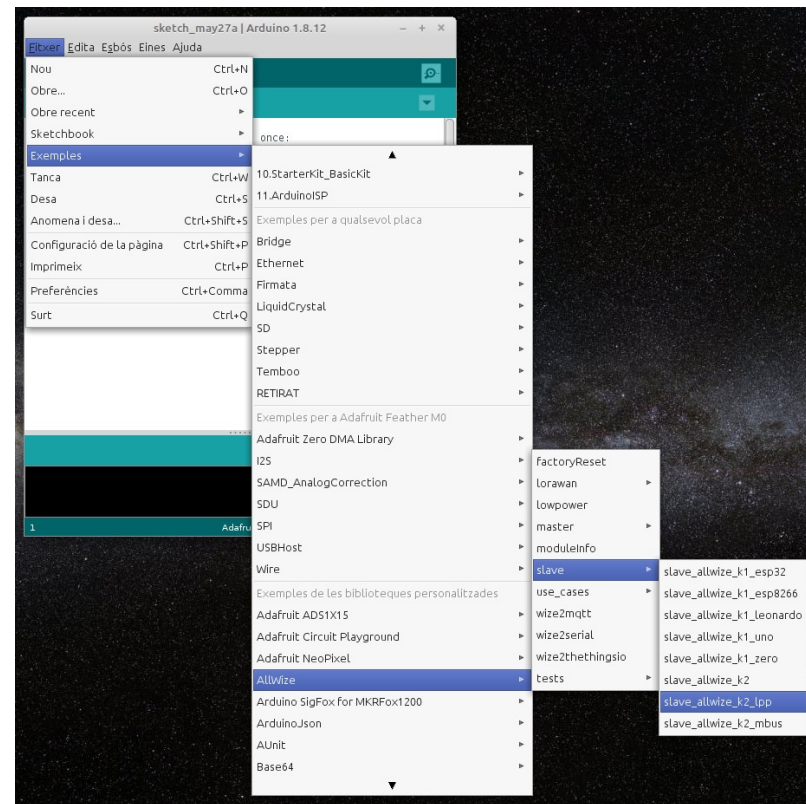
Repositori de codi

Aquest repositori conté el codi per el taller. Podem fer un «clone» amb GIT o bé descarregar-lo en un ZIP i descomprimir-lo en el nostre ordinador.

<https://github.com/AllWize/allwize-training>

Exemples

A més del codi del taller, podem accedir a altres exemples de codi fent servir les plaques de AllWize des del menú de l'IDE d'Arduino.



Dependències i configuració de la placa

Dependències

```
#include "AllWize.h"  
#include "CayenneLPP.h"
```

Configuració de la placa

```
#if defined(ARDUINO_AVR_LEONARDO)  
    #define RESET_PIN      7  
    #define MODULE_SERIAL  Serial1  
    #define DEBUG_SERIAL   Serial  
#endif // ARDUINO_AVR_LEONARDO  
  
#if defined(ARDUINO_ALLWIZE_K2)  
    #define RESET_PIN      7  
    #define MODULE_SERIAL  SerialWize  
    #define DEBUG_SERIAL   SerialUSB  
#endif // ARDUINO_ALLWIZE_K2
```

Configuració de la radio i objectes globals

Configuració de la ràdio

```
#define WIZE_CHANNEL      CHANNEL_04
#define WIZE_POWER        POWER_20dBm
#define WIZE_DATARATE     DATARATE_2400bps
#define WIZE_UID           0x20212223
```

Instàncies globals

```
AllWize allwize(&MODULE_SERIAL, RESET_PIN);
CayenneLPP lpp(16);
```


Inicialització de la ràdio

Inicilitació de la ràdio

```
// Init AllWize object
allwize.begin();
if (!allwize.waitForReady()) {
    DEBUG_SERIAL.println("[WIZE] Error connecting to the module, check your wiring!");
    while (true);
}

// WIZE radio settings
allwize.slave();
allwize.setChannel(WIZE_CHANNEL, true);
allwize.setPower(WIZE_POWER);
allwize.setDataRate(WIZE_DATARATE);
allwize.setUID(WIZE_UID);
```

Enviament

Enviament de dades

```
// Payload
lpp.reset();
lpp.addTemperature(1, getTemperature());
lpp.addRelativeHumidity(2, getHumidity());
lpp.addBarometricPressure(3, getPressure());

// Send the payload
if (!allwize.send(lpp.getBuffer(), lpp.getSize(), LORAWAN_PORT)) {
    DEBUG_SERIAL.println("[WIZE] Error sending message");
}
```

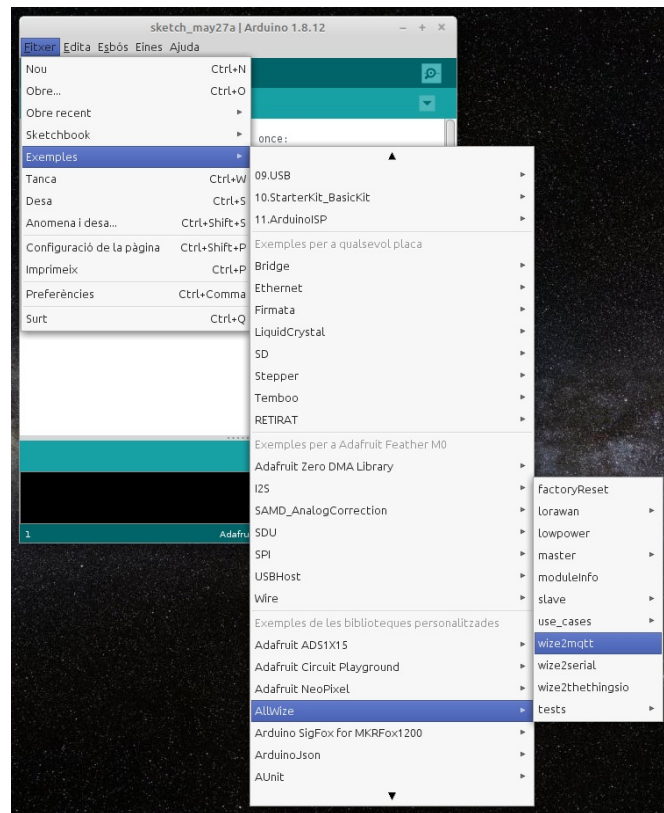
Passarel·la (Wize a MQTT)

Ara passarem a programar el dispositiu que farà de passarel·la. El codi implementa una passarel·la entre Wize i MQTT fent servir una placa basada en un ESP8266.

Primer haurem d'afegir suport per ESP8266 amb el gestor de plaques.

A continuació obrim el projecte sota la carpeta «gateway» del repositori.

Reanomenem l'arxiu «configuration.h.sample» a «configuration.h» i editem els continguts per adaptar-los a les nostres necessitats.



Passarel·la Wize a MQTT

El codi de la passarel·la gestiona:

- La connexió WiFi
- La ràdio Wize en mode «master» (receptor)
- L'enrutament cap a un servidor MQTT, produint dos tipus de missatges en format JSON:
 - PING

```
[MQTT] Sending: gateway/482412345678/ping => {"gateway":  
"mid":"4824","uid":"12345678","sn":"0000000000000000"}}
```

- DATA

```
[WIZE] ADDR: 0x20212223, RSSI: -42, DATA: 026700A00368A6047326A205880651F70053340003E8  
[MQTT] Sending: gateway/482412345678/uplink => {"app":254,"net":16,"uid":"20212223","cpt":13,"metadata":  
{"ch":4,"freq":169.4438,"dr":2400,"toa":73.33334},"gateway":  
{"mid":"4824","uid":"12345678","sn":"0000000000000000","rssi":-  
42},"payload":"026700A00368A6047326A205880651F70053340003E8","fields":  
{"temperature":16,"humidity":83,"pressure":989,"latitude":41.4199,"longitude":2.13,"altitude":10}}
```

Missatges MQTT

```
gateway/482412345678/ping
{
  "gateway": {
    "mid": "4824",
    "uid": "12345678",
    "sn": "000000000000000000"
  }
}
```

```
gateway/482412345678/uplink
{
  "app": 254,
  "net": 16,
  "uid": "20212223",
  "cpt": 15,
  "metadata": {
    "ch": 4,
    "freq": 169.4438,
    "dr": 2400,
    "toa": 73.33334
  },
  "gateway": {
    "mid": "4824",
    "uid": "12345678",
    "sn": "000000000000000000",
    "rssi": -42
  },
  "payload": "026700B90368620473266405880651F70053340003E8",
  "fields": {
    "temperature": 18.5,
    "humidity": 49,
    "pressure": 982.8,
    "latitude": 41.4199,
    "longitude": 2.13,
    "altitude": 10
  }
}
```

Limitacions

Aquesta demostració té algunes limitacions:

- Fem servir una passarel·la monocanal
- Només suporta missatges de pujada (uplinks)

Això significa que:

- Hi haurà més col·lisions / suporta menys dispositius degut a que no hi ha ortogonalitat
- Sense missatges de baixada (downlinks) no hi ha comprovació de recepció (ACK)
- Sense missatges de baixada tampoc podem implementar actuadors

Gràcies



Aigües de
Barcelona



Innovació