

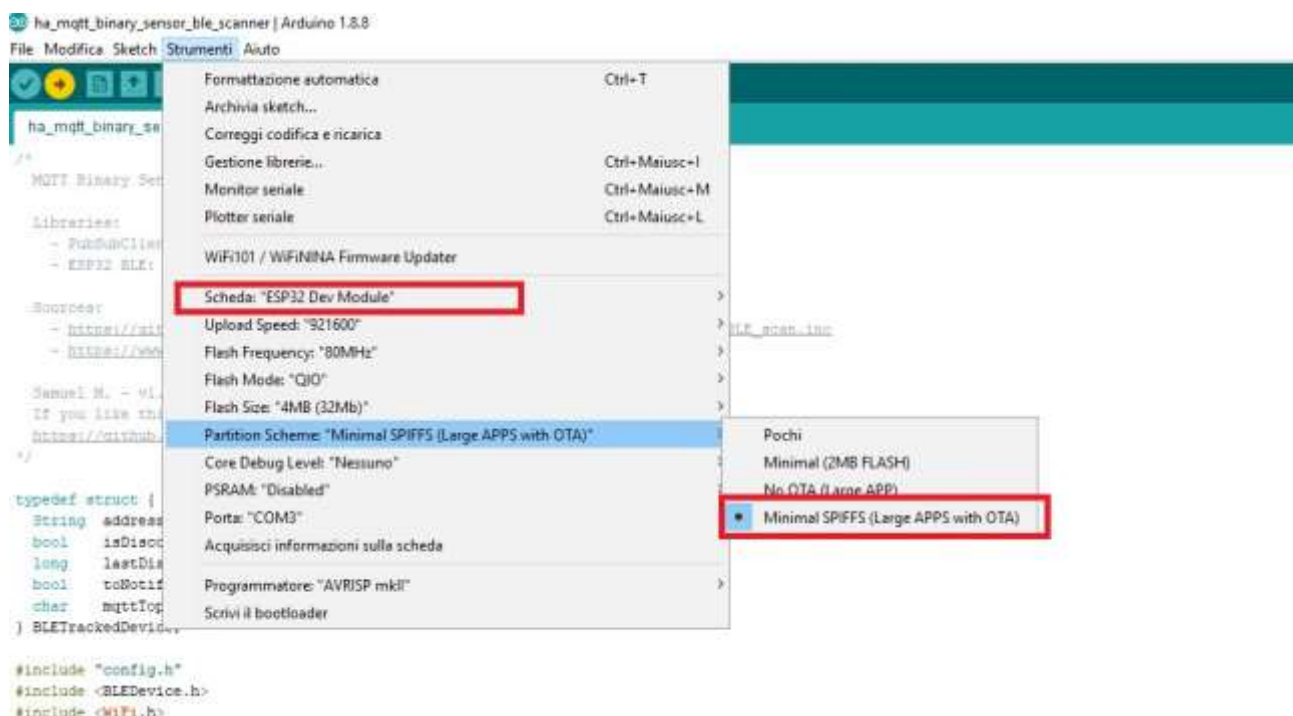
## Implementare rilevamento presenze tramite ESP32 e Tag economici BTLe by SDeSalve

Prerequisiti: arduino IDE, Broker MQTT preconfigurato e relativi dati di accesso, Scheda ESP32

1) aggiungere scheda ESP32 ad Arduino IDE (seguire questa guida <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/> )

assicurarsi di avere i Driver corretti. Altrimenti scaricarli ed installarli da <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

2) Collegare la scheda ESP32 al computer. Aprire Arduino IDE ed impostare la scheda a ESP32 Dev Module e selezionare la Porta COM corretta. Seleziona le partizioni da utilizzare come nell'immagine seguente:



### 3) Testare il funzionamento della scheda ESP32 caricando questo sketch:

```
#include <BLEDevice.h>

#include <BLEUtils.h>

#include <BLEScan.h>

#include <BLEAdvertisedDevice.h>

int scanTime = 30; //In seconds

class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {

    void onResult(BLEAdvertisedDevice advertisedDevice) {

        Serial.printf("Advertised Device: %s \n", advertisedDevice.toString().c_str());

    }

};

void setup() {

    Serial.begin(115200);

}

void loop() {

    // put your main code here, to run repeatedly:

    delay(5000);

    Serial.println("Scanning...");

    BLEDevice::init("");

    BLEScan* pBLEScan = BLEDevice::getScan(); //create new scan

    pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());

    pBLEScan->setActiveScan(true); //active scan uses more power, but get results faster

    BLEScanResults foundDevices = pBLEScan->start(scanTime);

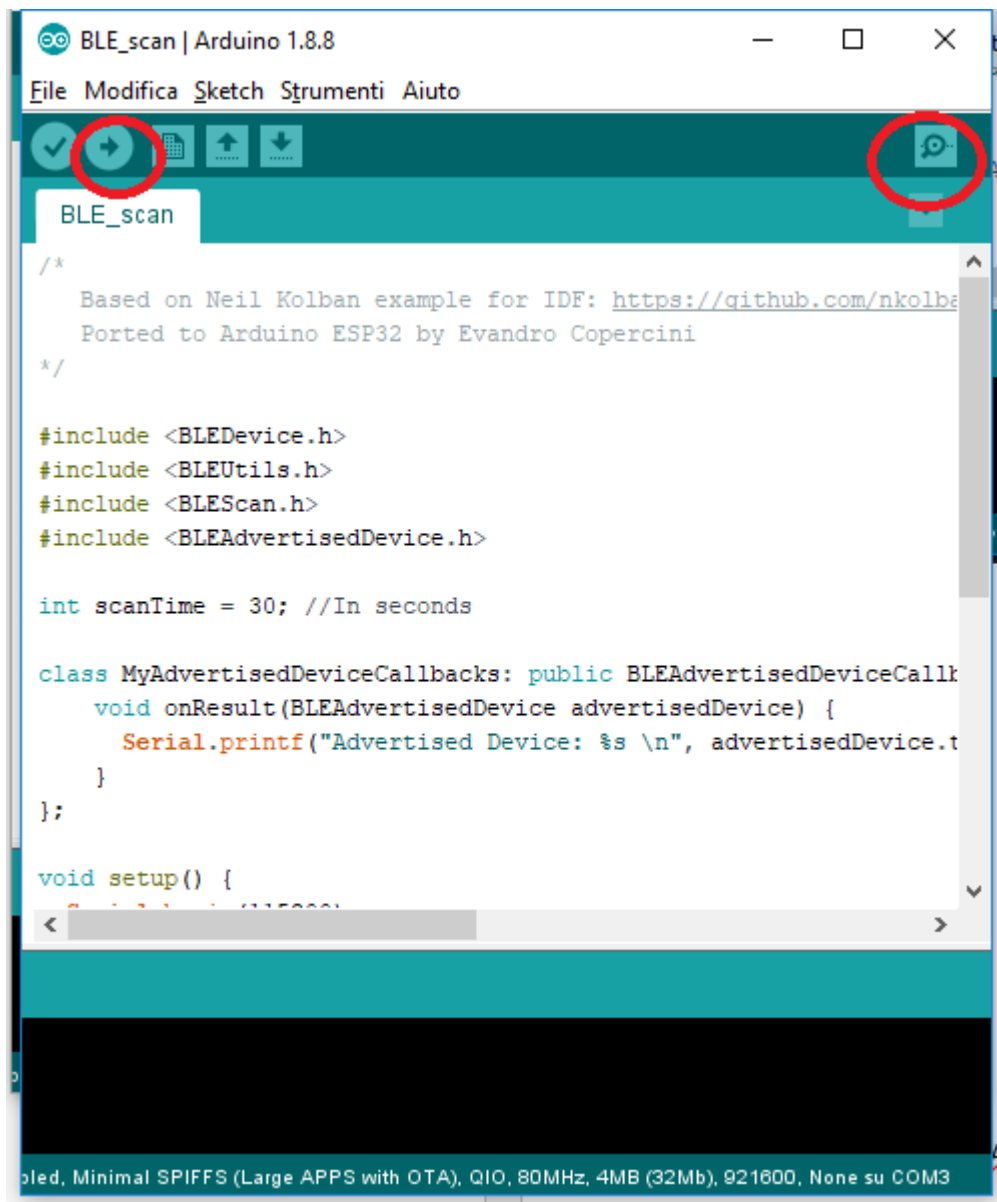
    Serial.print("Devices found: ");

    Serial.println(foundDevices.getCount());

    Serial.println("Scan done!");

}
```

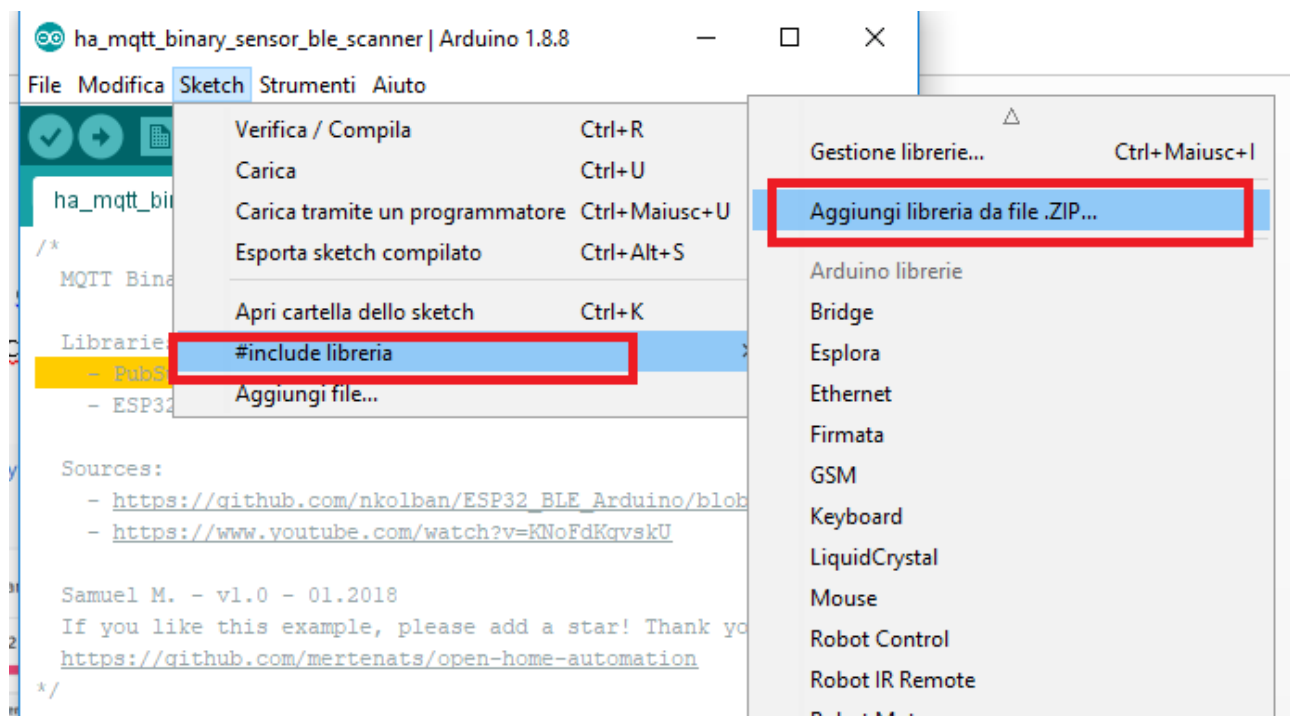
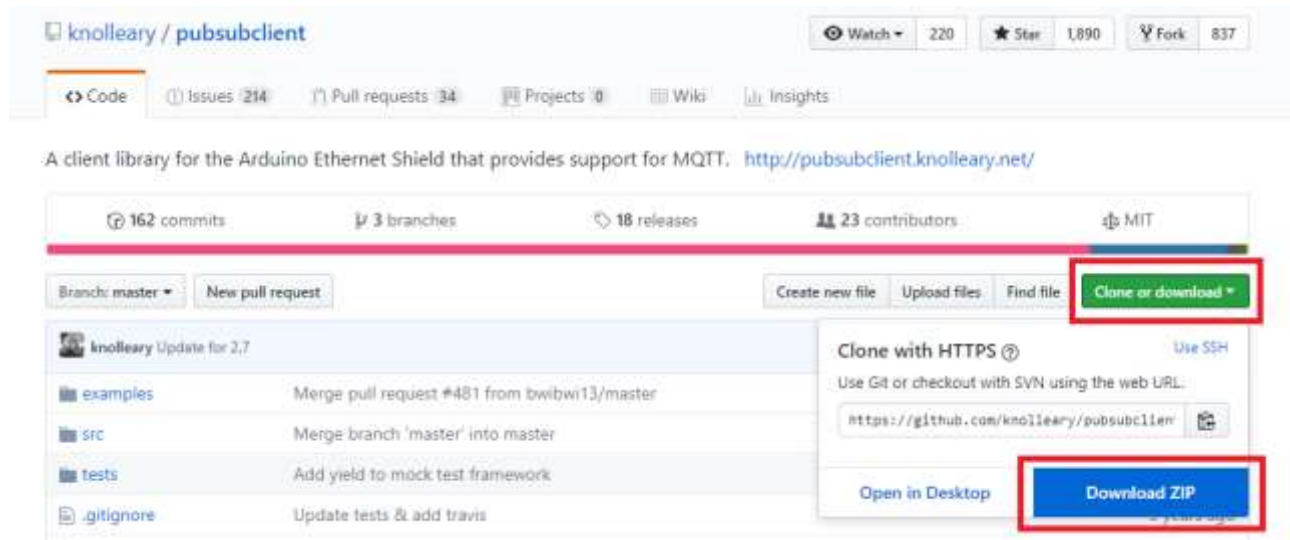
4) Inviare lo sketch alla ESP32 con la freccia a DX e premere la lente per vedere l'output della console:



Prendere nota dei MAC ADDRESS dei dispositivi BTL e da tracciare nella finestra della console

5) Scaricare ed installare in Arduno IDE la seguente libreria:

- PubSubClient: <https://github.com/knolleary/pubsubclient>



6) scaricare i file

- ha\_mqtt\_binary\_sensor\_ble\_scanner.ino
- example.config.h

[https://github.com/sdesalve/Open-Home-Automation/tree/master/ha\\_mqtt\\_binary\\_sensor\\_ble\\_scanner](https://github.com/sdesalve/Open-Home-Automation/tree/master/ha_mqtt_binary_sensor_ble_scanner)

7) rinominare example.config.h in config.h e personalizzarlo con i dati della propria rete Wifi, i Mac Address dei dispositivi da tracciare, quelli del Broker MQTT e decommentare la linea seguente(togliendo "//")

```
// #define DEBUG_SERIAL
```

8) Installare un client MQTT per collegarsi al Broker e vedere i topic.

Consiglio per Chrome questa APP:

<https://chrome.google.com/webstore/detail/mqttbox/kaajoficamnjihkeomgfljpicifbkaf>

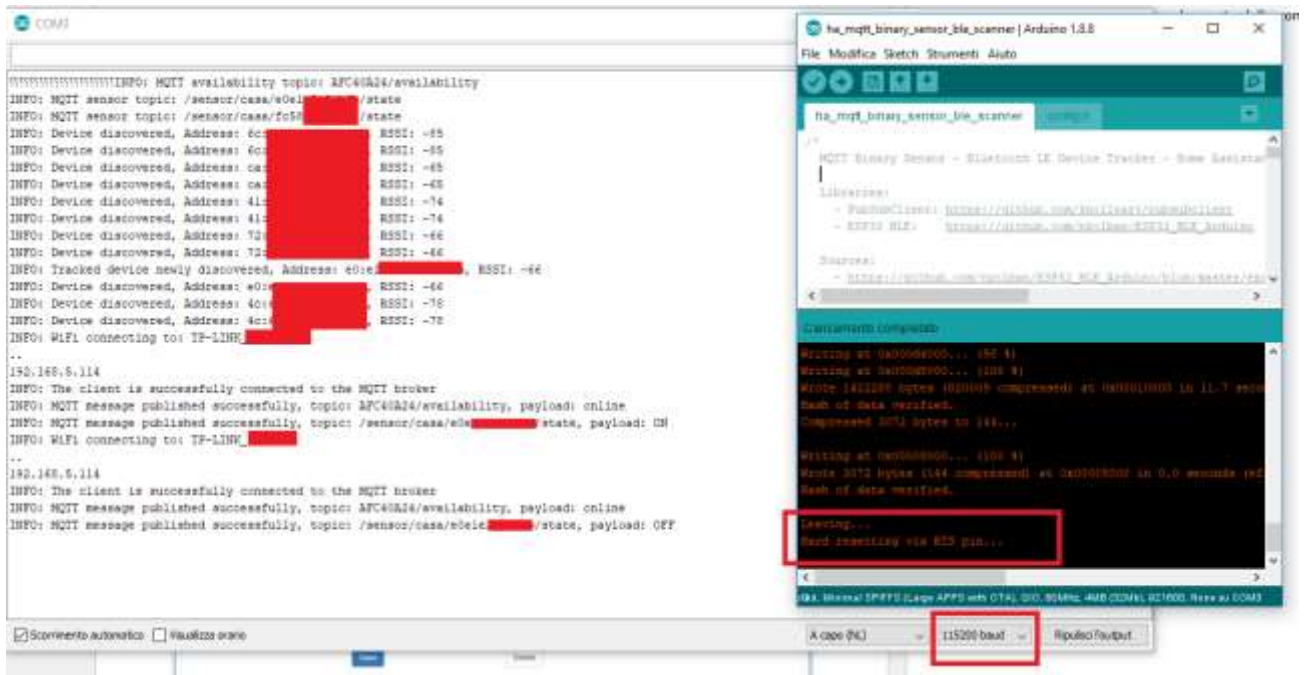
Configurarla per farla collegare al broker di Hassio inserendo i propri parametri nei campi evidenziati:

The screenshot shows the 'MQTT CLIENT SETTINGS' window. The 'Protocol' dropdown is set to 'mqtt / tcp'. The 'Host' field contains '192.168.5.149'. The 'Username' field contains 'mqtt'. The 'Password' field is masked with dots. Other visible settings include 'MQTT Client Name' as 'Hassio', 'MQTT Client Id' as 'dfca50eb-60b7-49c5-a82f-2c...', 'Reconnect Period' as '1000', 'Connect Timeout' as '30000', 'KeepAlive' as '10', and 'Will - Topic' as 'Will - Topic'. The 'Save' button is highlighted in blue.

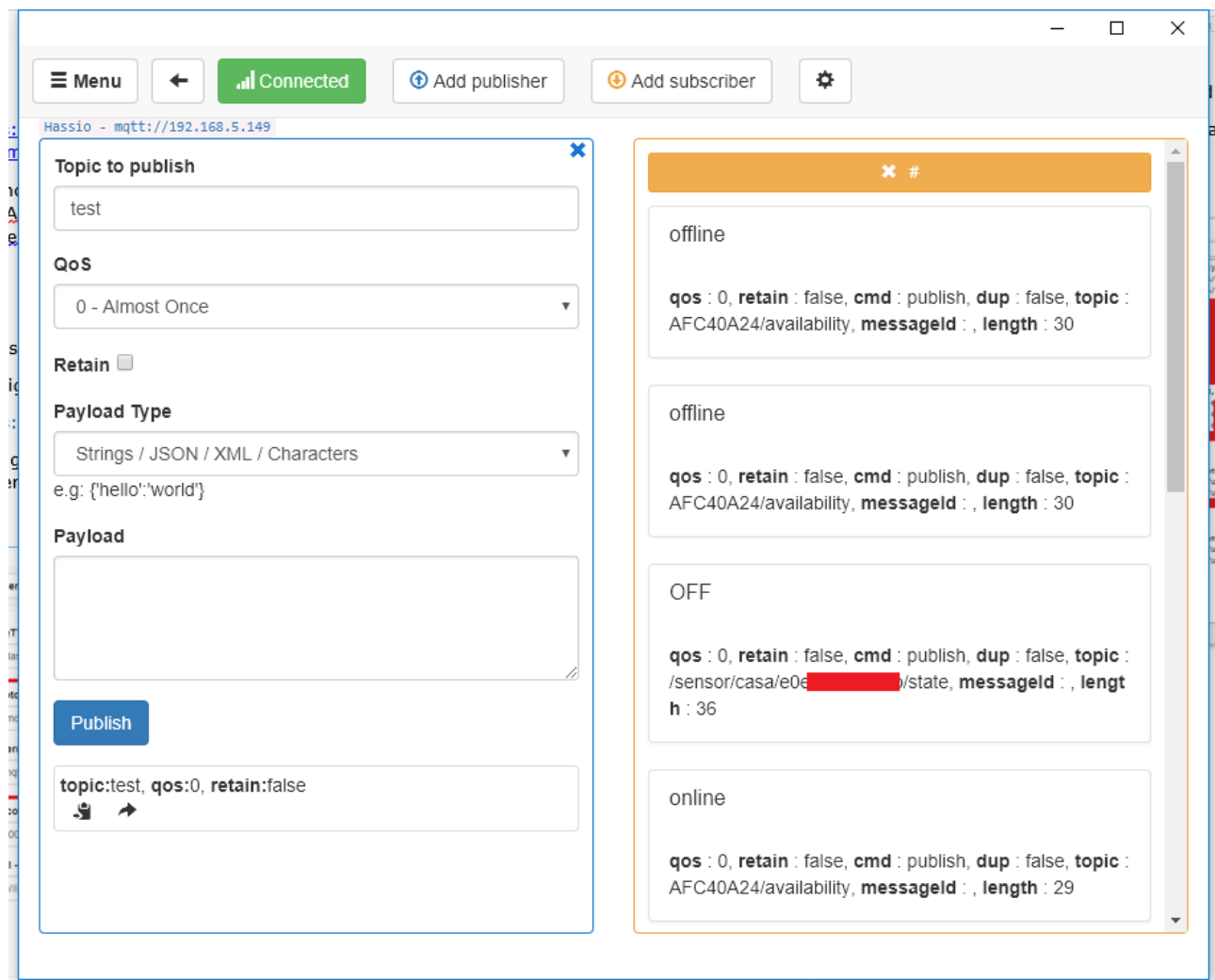
Sottoscrivere il topic # per vedere tutti i topic MQTT che verranno ricevuti dal Broker MQTT

9) compila, invia ed apri la schermata della console come indicato al punto 4 di questa guida.

Se tutto ha funzionato correttamente verrà visualizzato qualcosa del genere nella console di Arduino IDE:



E qualcosa del genere nel client MQTT



10) Aggiungi ad Home Assistant uno o più sensor MQTT per tracciare la presenza dei tag BTL e sensor:

```
- platform: mqtt
  name: "Tracker iTag Utente"
  state_topic: "XXXXXXXXXX/sensor/casa/XXXXXXXXXX/state"
  availability_topic: "XXXXXXXXXX/availability"
  initial_state: 'OFF'
  value_template: >-
    {% if value == 'ON' %}
      home
    {% elif value == 'OFF' %}
      away
    {% else %}
      offline
    {% endif %}
```

Scritta velocemente e probabilmente con qualche errore da @SDeSalve per  
<https://t.me/HomeAssistant> il 08/12/2018 22:30:11

Aggiornata il 13/12/2018 22:26:49