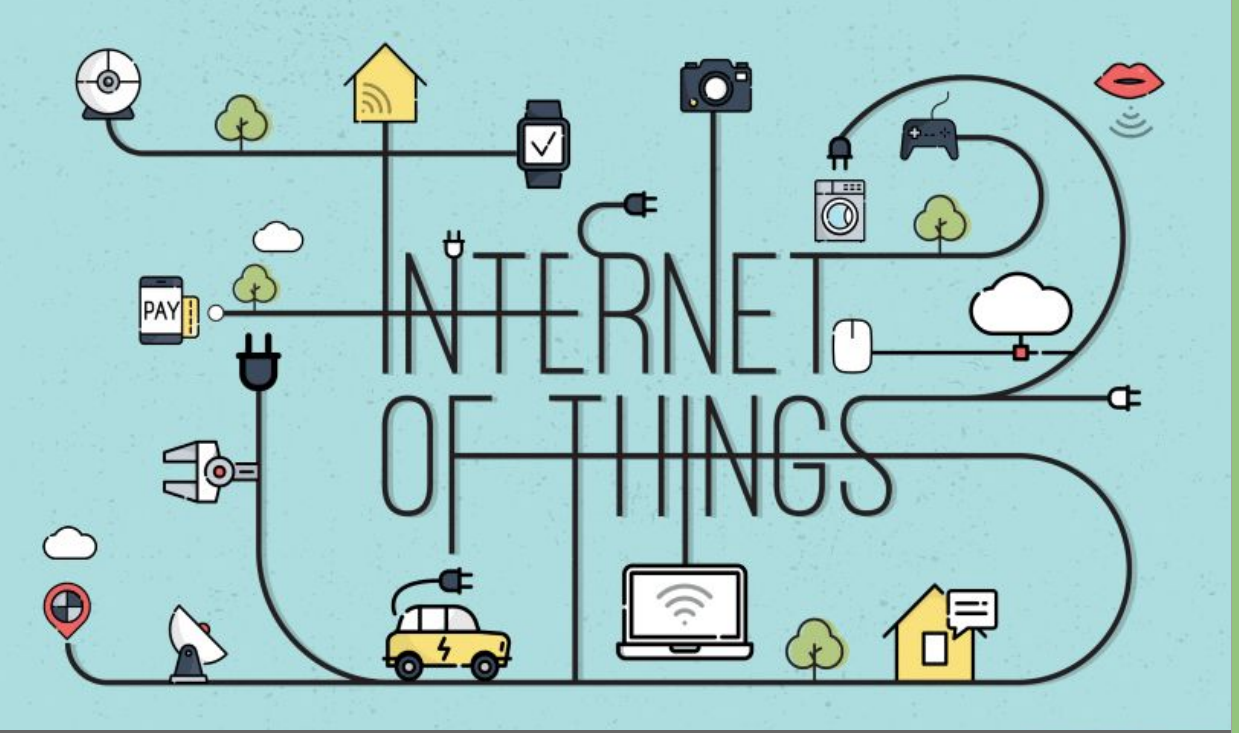


Publicando na Nuvem com ESP32 e MQTT

ARDUINO

Professor: Luiz Felipe Oliveira

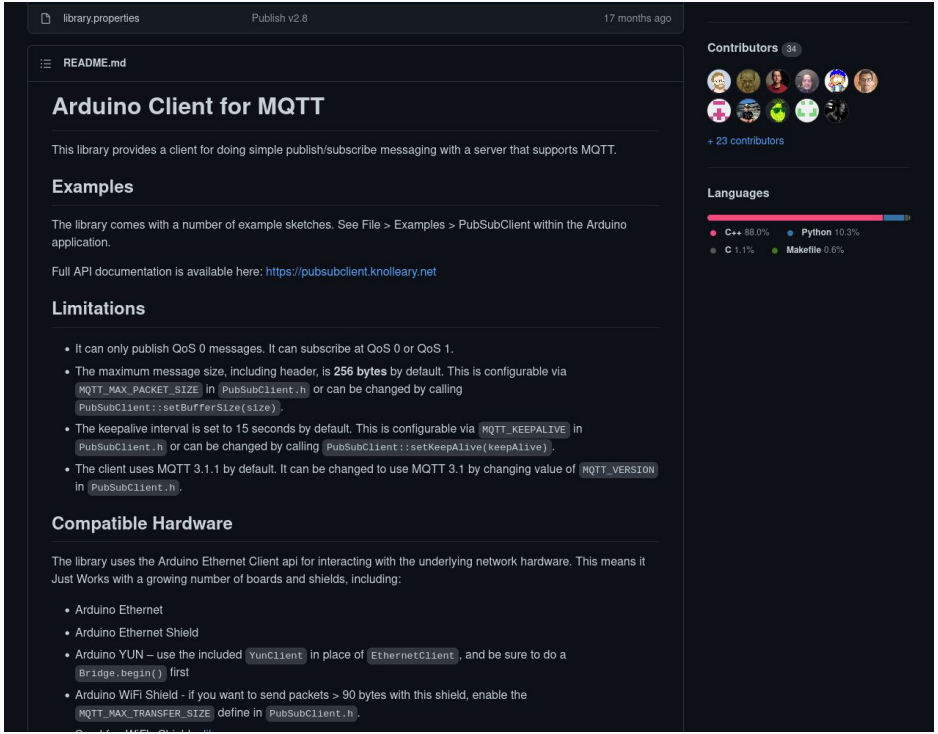
Parte 0: Explicando o MQTT

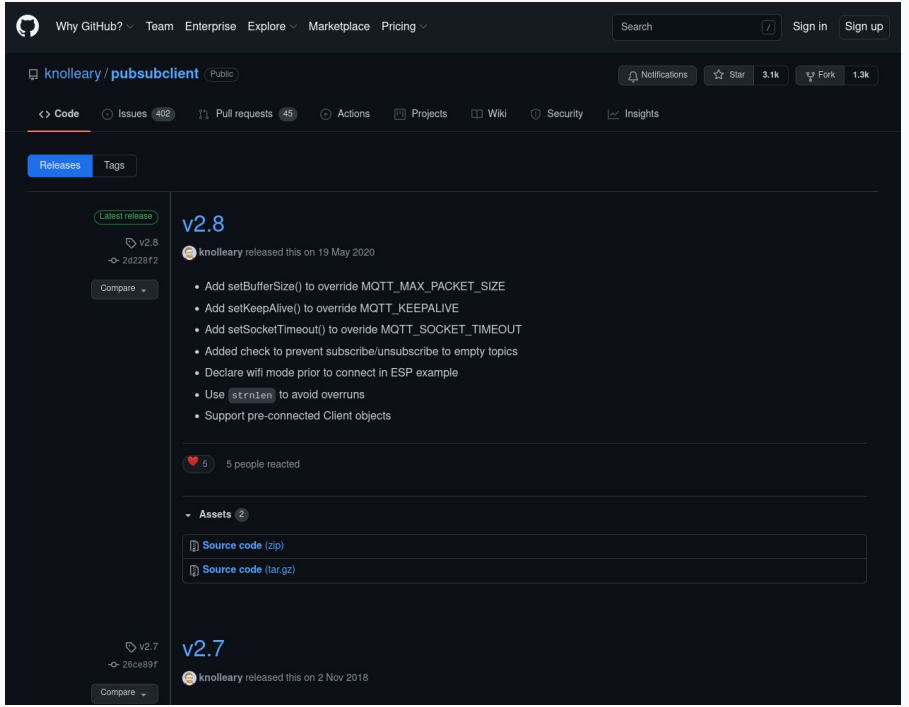
Parte 1: Publicação dos Dados

Parte 2: Visualização dos Dados

Instalando a Biblioteca

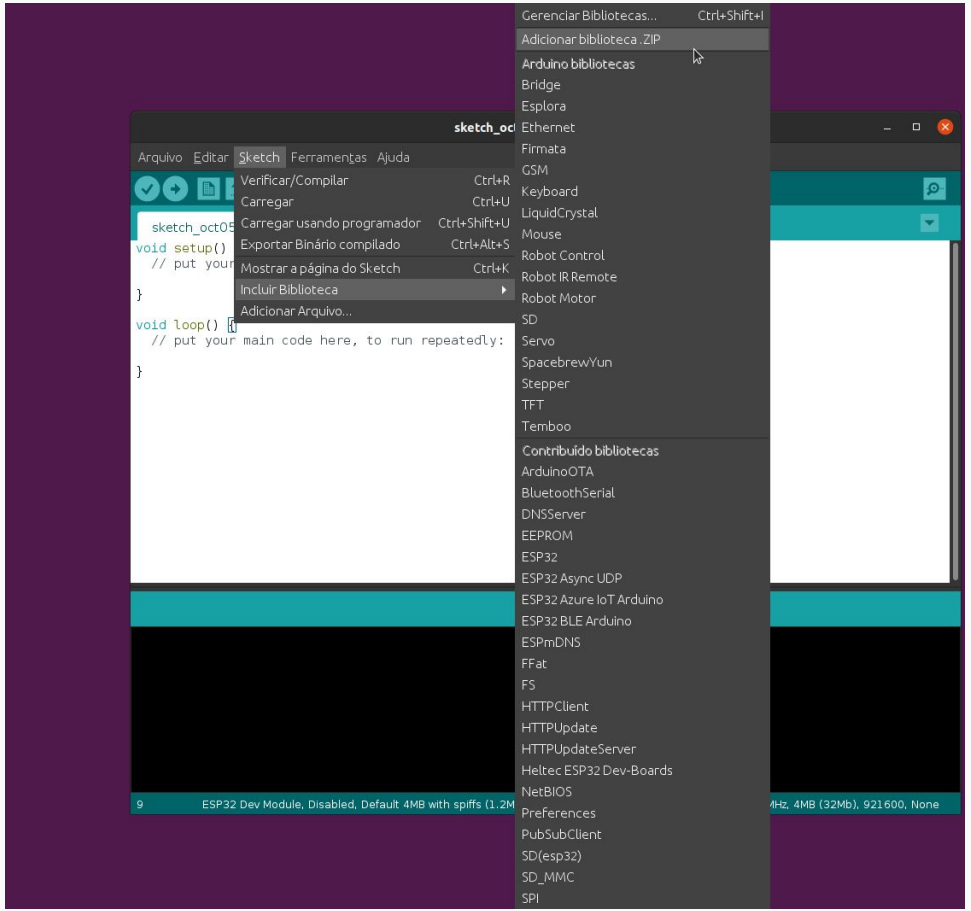
https://github.com/knolleary/pubsubclient





Instalando a Biblioteca

https://github.com/knolleary/pubsubclient



Modo Station + Cloud + **App**

WIFI MODE STATION

CLOUD

MinhaRede

192.168.0.1

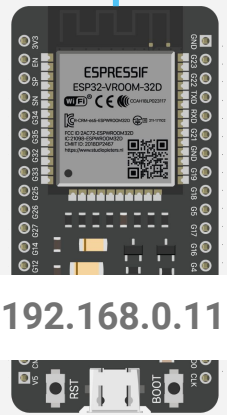
CLOUD

MQTT Broker

MQTT

DISPOSITIVO

ARDUINO IDE



MQTT Client

MQTT

publisher

millis();

192.168.0.11

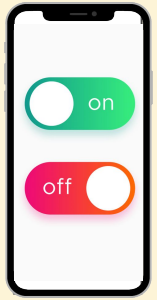
CLIENTE

MQTT

subscriber

MQTT Dash

Control the Things via MQTT protocol



Bibliotecas e definições iniciais


```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ESPmDNS.h>

//Net Setup
#define NET_SSID "LabIoT"
#define NET_PASSWORD "XXXXXXX"

//MQTT Setup
#define MQTT_ID "luiz-esp32-0800313131"
#define MQTT_BROKER "broker.hivemq.com"
#define MQTT_PORT 1883
#define MQTT_MILLIS_TOPIC "ifrj_cnit_luizfelipe_millis"
WiFiClient espClient; //Cliente de rede
PubSubClient MQTT(espClient); //Cliente MQTT

char millis_str[10] = "";
```

Configuração WiFi


```
void setupWifi() {
 //Configura a conexão à rede sem fio
  if (WiFi.status() == WL_CONNECTED)
       return;
  Serial.println();
 Serial.print("Connecting to ");
 Serial.println(NET_SSID);

 WiFi.begin(NET_SSID, NET_PASSWORD);

 while (WiFi.status() != WL_CONNECTED) {
     delay(500);
     Serial.print(".");
 }

 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());
}
```

Configuração MQTT


```
void setupMQTT() {
  MQTT.setServer(MQTT_BROKER, MQTT_PORT);//informa qual broker e porta deve ser conectado

  while (!MQTT.connected())
   {
       Serial.print("*Tentando se conectar ao Broker MQTT: " );
       Serial.println(MQTT_BROKER);
       if (MQTT.connect(MQTT_ID))
       {
           Serial.println("Conectado com sucesso ao broker MQTT!" );
       }
       else
       {
           Serial.println("Falha ao reconectar no broker.");
           Serial.println("Havera nova tentativa de conexao em 2s" );
           delay(2000);
       }
   }
}
```

Função SETUP


```
void setup(void) {
 //Configura o baudrate da comunicação serial
 Serial.begin(115200);

 setupWifi();

 setupMQTT();

}
```

Função LOOP


```
void loop(void) {

 sprintf(millis_str, "%d", millis());

 MQTT.publish(MQTT_MILLIS_TOPIC, millis_str);

 setupWifi();
 setupMQTT();
 delay(2000);
}
```

```
int sprintf ( char * str, const char * format, values );
```

corresponding argument:

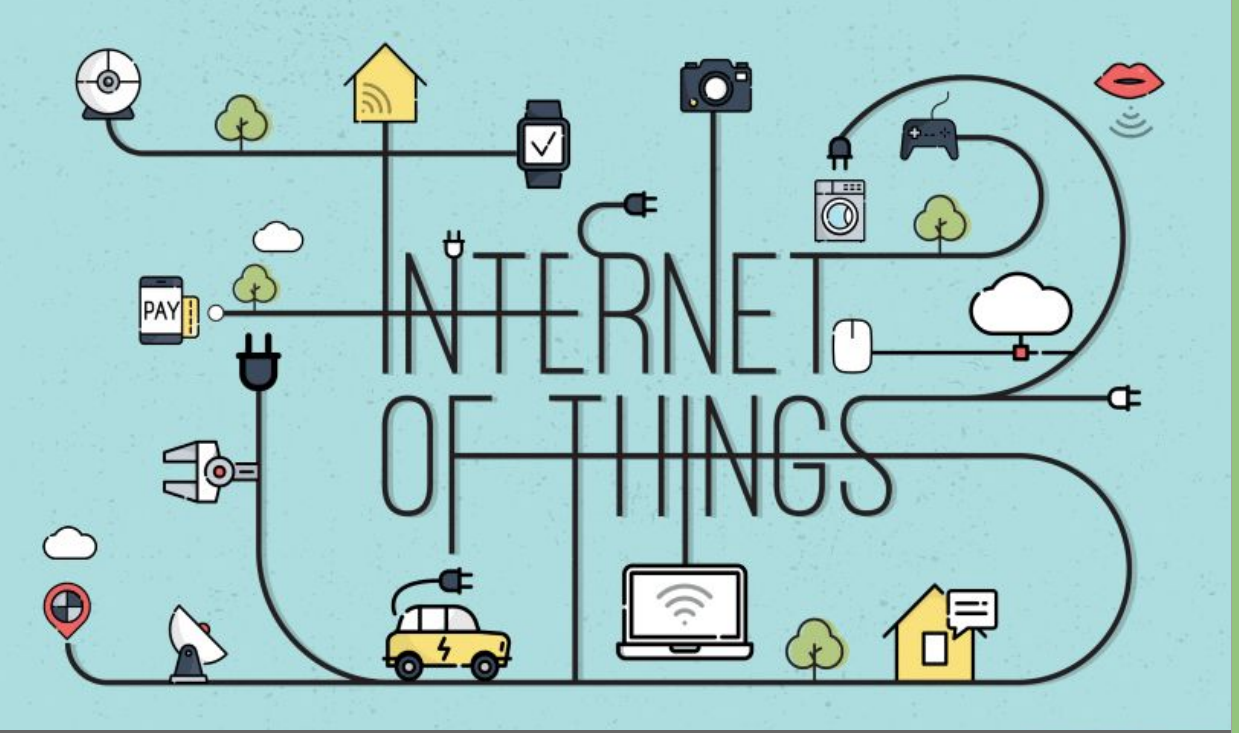
specifier	Output	Example
d *or* i	Signed decimal integer	392
u	Unsigned decimal integer	7235
o	Unsigned octal	610
x	Unsigned hexadecimal integer	7fa
X	Unsigned hexadecimal integer (uppercase)	7FA
f	Decimal floating point, lowercase	392.65
F	Decimal floating point, uppercase	392.65
e	Scientific notation (mantissa/exponent), lowercase	3.9265e+2

Composes a string with the same text that would be printed if *format* was used on printf, but instead of being printed, the content is stored as a *C string* in the buffer pointed by *str*.

Format

corresponding argument:

specifier	Output	Example
d *or* i	Signed decimal integer	392
u	Unsigned decimal integer	7235
o	Unsigned octal	610
x	Unsigned hexadecimal integer	7fa
X	Unsigned hexadecimal integer (uppercase)	7FA
f	Decimal floating point, lowercase	392.65
F	Decimal floating point, uppercase	392.65
e	Scientific notation (mantissa/exponent), lowercase	3.9265e+2
E	Scientific notation (mantissa/exponent), uppercase	3.9265E+2
g	Use the shortest representation: %e or %f	392.65
G	Use the shortest representation: %E or %F	392.65
a	Hexadecimal floating point, lowercase	-0xc.90fep-2
A	Hexadecimal floating point, uppercase	-0XC.90FEP-2
c	Character	a
s	String of characters	sample
p	Pointer address	b8000000
n	Nothing printed. The corresponding argument must be a pointer to a `signed int`. The number of characters written so far is stored in the pointed location.	
%	A % followed by another % character will write a single % to the stream.	%



Dúvidas

ARDUINO™

Professor: Luiz Felipe Oliveira