

Les objets connectés (MQTT)



ТΡ

LE CREN Anthony

Les logins et les mots de passe décrits dans ce document sont donnés comme exemple.

1 Qu'est-ce que MQTT ?

MQTT (Message Queuing Telemetry Transport) est une messagerie publish-subscribe basé sur le protocole TCP/IP. MQTT est utilisé pour les IOT (Internet of Things / objects connectés). Il est conçu comme un transport de messagerie de publication / abonnement extrêmement léger en termes de ressources.

Mise en situation :

Vous avez réalisé un système à base d'ESP8266 permettant de mesurer la température d'une pièce. Vous voulez connaitre cette température quand vous êtes à l'extérieur de votre maison. A première vue, une solution serait de concevoir une page WEB afin de pouvoir y accéder depuis un navigateur. MQTT va vous permettre de remplir cet objectif plus rapidement en utilisant une bande passante très réduite. Il est aussi possible de déclencher un mécanisme à distance comme une des volets roulants etc... La communication peut ainsi être bidirectionnelle.



Communication en 2 étapes :

- Un objet connecté va publier (publish) son message vers le Broker (Serveur)
- Pour recevoir des messages le client va souscrire (subscribe) leurs réceptions auprès du Broker.

Pour que les messages ne se mélangent pas, ils sont publiés sur une chaine (topic). Par exemple /capteurs/temperature. Voir la <u>vidéo</u> de présentation.

1 Publication de température

Q1 A l'aide de l'annexe 1, configurer votre broker sur le site https://www.shiftr.io/cloud/

Noter l'adresse, le login et le mot de passe utilisé lors de l'enregistrement dans le tableau cidessous :

| Adresse du serveur (URL) | touchard.cloud.shiftr.io |
|-----------------------------|------------------------------|
| Client id (myMqttClient) | Python ou esp8266 ou arduino |
| Login (USER_ID) | touchard |
| Mot de passe (MQTT_API_KEY) | MFmD747Blp8YlJYl |
| Topic ou token | capteurs |

Q2 Vérifier la connexion sur le point d'accès internet en utilisant le programme

«wifi_connexion.py»

Configurer et **tester** le programme ci-dessous avec vos propres identifiants conformément à la question **Q1.** (Q2-mqtt-publish.py)

```
from umqtt.robust import MQTTClient
import network
import sys
import time
from time import sleep ms
from machine import Pin
from onewire import OneWire
from ds18x20 import DS18X20
sta_if = network.WLAN(network.STA_IF)
print(sta_if.active())
print(sta if.ifconfig())
myMqttClient = b"micropython"
URL = b"touchard.cloud.shiftr.io"
USER ID = b'touchard'
MQTT API KEY = b'MFmD747BIp8YIJYI'
client = MQTTClient(client id=myMqttClient,
                    server=URL,
                    user=USER ID,
                    password=MQTT_API_KEY,
                    ssl=False)
try:
   client.connect()
   print("connection ok");
except Exception as e:
   print('could not connect to MQTT server {}{}'.format(type(e).___name___, e))
   svs.exit()
bus = OneWire(Pin(12))
ds = DS18X20 (bus)
capteur temperature = ds.scan()
PUBLISH PERIOD IN SEC = 10
while True:
    try:
        ds.convert_temp()
        sleep ms( 750 )
        temp celsius = ds.read temp(capteur temperature[0])
        print("Température : ", temp celsius )
        client.publish("/capteurs/temperature", str(int(temp_celsius)))
        print("publish ok");
        time.sleep(PUBLISH_PERIOD_IN_SEC)
    except KeyboardInterrupt:
        print('Ctrl-C pressed...exiting')
        client.disconnect()
        sys.exit()
        print("exit")
```

Q3 Vérifier l'envoi de données au broker. (<u>https://touchard.cloud.shiftr.io/</u>)

| Last Messages | | | Active Connections | True |
|--|-------------------------|--------------|--------------------|---|
| 0 msgs/s capteurs/temperature 16 micropython 13:59:41 NR Q0 | touchard.cloud.shiftnio | 1 connection | Recent Errors | ('192.168.1.104', '255.255.255.0', '192.168.1.1', '192.168.1.1') |
| mi | | | | connection ok |
| | | | | Température : 16.25 |
| | | | | publish ok |

Q4 A l'aide de l'annexe 2,3 ou 4, **configurer** le client MQTT sur un smartphone. Préférence pour **MQTT panel** : annexe 3)

Noter à nouveau la clé et le mot de passe en lien avec le broker (Q1) dans le tableau ci-dessous :

| Adresse du serveur | touchard.cloud.shiftr.io |
|--------------------|--------------------------|
| Client id | smartphone |
| login | touchard |
| Mot de passe | MFmD747Blp8YlJYl |
| Topic ou token | capteurs |

Q5 Vérifier l'envoi de données du broker vers le smartphone.



Q6 Identifier le subscriber et le publisher en complétant le tableau ci-dessous par oui ou non

| | Subscriber | Publisher | |
|----------------------|------------|-----------|--|
| L'esp8266 est le | Non | Oui | |
| Le smartphone est le | Oui | Non | |

La qualité de service (QDS) ou quality of service (QoS) est la capacité à véhiculer dans de bonnes conditions un type de trafic donné.

| Other settings | QoS0. Le message envoyé n'est pas stocké par le Broker. Il n'y a pas d'accusé de réception. Le message sera perdu en cas d'arrêt du serveur ou du client. C'est le mode par défaut |
|----------------|--|
| QoS(1) | QoS1. Le message sera livré au moins une fois. Le client renvoie le message jusqu'à ce que le broker envoi en retour un accusé de réception. |
| QoS(2) | QoS2. Le broker sauvegarde le message et le transmettra jusqu'à ce qu'il ait été réceptionné par tous les souscripteurs connectés |

2 Commande de la Led de l'IOT avec le smartphone

Q7 Configurer et tester le programme ci-dessous avec vos propres identifiants conformément à la question **Q1** (Q7-mqtt-subscribe-led.py)

```
from umqtt.robust import MQTTClient
import network
import sys
from machine import Pin
led = Pin(0, Pin.OUT)
def cb(topic, msg):
   print((topic, msg))
    valeur=int(msg.decode("ascii"))
   if valeur==1:
        led.on()
    elif valeur==0:
       led.off()
sta if = network.WLAN(network.STA IF)
print(sta_if.active())
print(sta if.ifconfig())
myMqttClient = b"micropython"
URL = b"touchard.cloud.shiftr.io"
USER_ID = b'touchard'
MQTT API KEY = b'MFmD747BIp8YIJYI'
client = MQTTClient(client_id=myMqttClient,
                    server=URL,
                    user=USER ID,
                    password=MQTT API KEY,
                    ssl=False)
client.set callback(cb)
try:
   client.connect()
   print("connection ok");
   client.subscribe("/actionneurs/led")
except Exception as e:
   print('could not connect to MQTT server {}{}'.format(type(e).__name__, e))
   sys.exit()
while True:
    try:
        client.check msg()
    except KeyboardInterrupt:
       print('Ctrl-C pressed...exiting')
        client.disconnect()
        sys.exit()
        print("exit")
```

Q8 Ajouter un panel appelé « **Switch** »de commande comme le montre l'exemple ci-dessous, puis valider la commande de la led sur le site shiftr.io



Q9 Identifier le subscriber et le publisher en complétant le tableau ci-dessous par oui ou non

| | Subscriber | Publisher |
|----------------------|------------|-----------|
| L'esp8266 est le | Oui | Non |
| Le smartphone est le | Non | Oui |



http://mqtt.org/documentation

| Connect Command (client to server) | | | | |
|--|--|--|--|--|
| http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718028 | | | | |
| MQ Telemetry Transport Protocol Connect Command 0001 0000 = Header Flags: 0x10 (Connect Command) Msg Len: 53 Protocol Name: MQTT Version: 4 1100 0010 = Connect Flags: 0xc2 Keep Alive: 60 Client ID: processing User Name: weatherSensors Password: bme280Sensors | | | | |
| 0000 90 4d 4a a3 0e 00 25 22 8a 86 a0 08 00 45 00 0010 00 6b 93 a5 40 00 40 6 96 d9 co a8 01 15 36 4c .k.@.@. .k.@.@. .k.@.@. 6L 0020 18 05 b7 ca 07 5b 79 8f ed 67 52 30 38 88 80 18 | | | | |
| Connect Command | | | | |
| 0001 0000 = Header Flags: 0x10 (Connect Command) | | | | |
| Msg Len: 53 | | | | |
| Protocol Name: MQTT | | | | |
| Version: 4 | | | | |
| Keen Alive: 60 | | | | |
| Client ID: processing | | | | |
| User Name: weatherSensors | | | | |
| Password: bme280Sensors | | | | |
| | | | | |
| 0000 <mark>10 35</mark> 00 04 <mark>4d 51 54 54 <mark>04</mark> c2 <mark>00 3c</mark> 00 0a 70 72 .5MQTT<pr< mark=""></pr<></mark> | | | | |
| 0010 6f 63 65 73 73 69 6e 67 00 0e 77 65 61 74 68 65 ocessingweathe | | | | |
| 0020 72 53 65 6e 73 6f 72 73 00 0d 62 6d 65 32 38 30 rSensorsbme280 | | | | |
| 0030 53 65 6e 73 6f 72 73 Sensors | | | | |



Connect Ack (server to client) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718033 Transport Protocol Telemetr Connect Ack 0010 0000 = Header Flags: 0x20 (Connect Ack) Msg Len: 2 0000 0000 = Connection Ack: Connection Accepted (0) 4a a3 0e 00 08 00 45 00 .%"....M J....E. 0000 00 25 22 8a 86 a0 90 4d 7e 22 36 4c 18 05 c0 a8 .8..@... ~"6L.... 0010 00 38 02 8f 40 00 ea 06 0020 01 15 07 5b b7 ca 52 30 38 88 79 8f ed 9e 80 18 ...[..R0 8.y.... 0030 00 72 ab 7f 00 00 01 01 08 0a 07 17 3a 0e d4 85 .r..... 0040 d3 f7 20 02 00 00 Connect Ack 0010 0000 = Header Flags: 0x20 (Connect Ack) Msg Len: 2 0000 0000 = Connection Ack: Connection Accepted (0)

0000 20 02 00 00

Subscribe Request (client to server) http://docs.oasis-open.org/mgtt/mgtt/v3.1.1/os/mgtt-v3.1.1-os.html# Toc398718063 MQ Telemetry Transport Protocol Connect Ack 0010 0000 = Header Flags: 0x20 (Connect Ack) Msg Len: 2 0000 0000 = Connection Ack: Connection Accepted (0) .%"....M J.....E. 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 0010 00 38 02 8f 40 00 ea 06 7e 22 36 4c 18 05 c0 a8 .8..@... ~"6L.... 0020 01 15 07 5b b7 ca 52 30 38 88 79 8f ed 9e 80 18 ...[..R0 8.y.... 00 72 ab 7f 00 00 01 01 08 0a 07 17 3a 0e d4 85 .r.... 0040 d3 f7 20 02 00 00 Subscribe Request 1000 0010 = Header Flags: 0x82 (Subscribe Request) Msg Len: 25 Message Identifier: 1 Topic: /sensors/temperature 0000 82 19 00 01 00 14 2f 73 65 6e 73 6f 72 73 2f 74/sensors/t 0010 65 6d 70 65 72 61 74 75 72 65 00 emperature.

Subscribe Ack (server to client) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718068 Subscribe Ack 1001 0000 = Header Flags: 0x90 (Subscribe Ack) Msg Len: 3 Message Identifier: 100 = Granted Qos: Fire and Forget (0)

A PUBLISH Control Packet is sent from a Client to a Server or from Server to a Client to transport an Application Message.

| http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718037 • W0 Telemetry Transport Protocol • W011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 • 0010 002 52 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 0010 002 52 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 0010 002 52 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 0011 0000 el 29 3d 90 00 a1 01 08 a0 71 72 ca dd 48 5 co a8 .0 0020 01 15 07 5b b7 ca 52 30 38 9b 79 8f ed 98 00 15 0020 02 24 40 00 00 101 08 a0 71 72 ca dd 48 5 0030 07 24 40 00 00 11 08 00 71 72 cb 32 35 2e 38 30 Publish Message incoming 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 00000 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 65 6d 0/sensors/tem 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html# Toc398718037 • W0 Telemetry Transport Protocol • No Telemetry Transport Protocol < |
|---|
| You Telemetry Transport Protocol Publish Message 9011 0000 = Header Flags: 0x30 (Publish Message) Mg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 00 25 22 68 68 a0 90 4d 4a a3 0e 90 08 0d 45 00, M JE. 0010 00 25 22 68 06 a0 90 ea 06 7e 05 36 4c 18 05 c0 a8, M JE. 0011 000 2 2 e4 00 00 e1 10 00 a0 17 3c ad d4 85, I.M 0 s Publish Message incoming 0011 0000 = Header Flags: 0x30 (Publish Message) Mg Len: 27 Topic: /sensors/temperature Message: 25.80 Publish Message (client to server) http://docs.oasis-open.org/mgtt//mgtt/v3.1.1/os/mgtt-v3.1.1-os.html# Toc398718037 Mg Len: 27 Topic: /sensors/temperature Message (client to server) http://docs.oasis-open.org/mgtt/mgtt/v3.1.1/os/mgtt-v3.1.1-os.html# Toc398718037 Mg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 65 6d 0/sensors/tem 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 perature25.80 Publish Message (client to server) http://docs.oasis-open.org/mgtt/mgtt/v3.1.1/os/mgtt-v3.1.1-os.html# Toc398718037 Mg Len: 27 Topic: /sensors/temperature Mg Len: 27 Ng Len: 27 Sensors/temperature Mg Len: 27 Sensors/temperature Mitol 0000 = Header Flags: 0x30 (Publish Message) Mg Len: 27 Mg Len: 27 Sensors/temperature Mg Len: 27 Mg Len: 27 Topic: /sensors/temperature Mg Len: 27 Mg Len: 27 Mg Len: 27 Mg Len: 27 Mg Len: 24 |
| 0000 00 25 22 88 68 99 44 aa 30 60 80 45 00 <td< td=""></td<> |
| Publish Message incoming 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 65 6d 0/sensors/tem 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 perature25.80 Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718037 • M0 Telemetry Transport Protocol • Publish Message • 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 • 9000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00(m) • 0011 0000 = Header Flags: 0x30 (Publish Message) • Msg Len: 27 Topic: /sensors/temperature Message: 25.80 • 9010 00 51 02 93 40 00 ea 06 7e 05 36 4c 18 05 co a8(, 0, 0, 0) • 072 2e 40 00 00 01 10 80 a0 71 73 ca d4 45, 1(N 8,,,,,,, |
| 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 65 6d 0/sensors/tem 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 perature25.80 Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718037 ✓ M0 Telemetry Transport Protocol ◆ Publish Message ♦ 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0010 00 51 02 93 40 00 ea 06 7e 05 36 4c 18 05 co a8 0011 00 72 2e 40 00 00 01 01 08 00 71 73 cat d4 48 5. (r.@ |
| Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 65 6d 0/sensors/tem 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 perature25.80 Publish Message (client to server) http://docs.oasis-open.org/mqt/mqt/v3.1.1/os/mqtt-v3.1.1-os.html# Toc398718037 ▼ W0 Telemetry Transport Protocol ▼ Publish Message ▶ 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: / sensors/temperature Message: 25.80 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 0010 00 51 02 93 40 00 ea 06 7e 05 36 4c 18 05 c0 a8 .Q.@~.6L 0030 00 72 2e 40 00 00 01 01 08 0a 07 17 3c ad d4 85 .r.[.00 8.y 0040 d4 3e 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 .>@/s ensors/t 0050 05 6f 70 65 72 61 74 75 72 65 32 35 2e 38 30 emperatu re25.80 Publish Message outcoming .0011 0000 = Header Flags: 0x30 (Publish Message) /s ensors/t 0041 043 e 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 .>@/s ensors/t 0050 05 6f 07 0 65 72 61 74 75 72 65 32 35 2e 38 30 |
| Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718037 • M0 Telemetry Transport Protocol • 0011 0000 = Header Flags: 0x30 (Publish Message) • Mg sage: 25.80 |
| 0000 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 65 6d 0/sensors/tem 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 perature25.80 Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html# Toc398718037 • M0 Telemetry Transport Protocol • Publish Message • 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 %"M JE. 0011 0000 = Header Flags: 0x30 (Publish Message) |
| 0000 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 65 6d 0/sensors/tem 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 perature25.80 Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718037 ✓ NO Telemetry Transport Protocol ◆ Publish Message ▶ 0611 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00,, M JE. 0010 00 51 02 93 40 00 ea 06 7e 05 36 4c 18 05 co a8,,,,,,, . |
| 0010 70 65 72 61 74 75 72 65 32 35 2e 38 30 perature25.80 Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718037 • MQ Telemetry Transport Protocol • Publish Message • 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 0010 00 51 02 93 40 00 ea 06 7e 05 36 4c 18 05 co a8 .Q0 ~.6L 0011 15 07 5b b7 ca 52 30 38 9b 79 8f ed e9 80 18 [.R0 8.y 0000 07 22 e4 40 00 00 01 01 08 0a 07 17 3c ad 44 85 [.R0 8.y 0040 d4 3e 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 .>0/s ensors/t 0050 65 6d 70 65 72 61 74 75 72 65 32 35 2e 38 30 emperatu re25.80 Publish Message outcoming 0011 0000 = Header Flags: 0x30 (Publish Message) [sensors/t Msg Len: 24 24 |
| Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html# Toc398718037 ✓ MQ Telemetry Transport Protocol ✓ Publish Message ▶ 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 0010 00 25 102 93 40 00 ea 06 7e 05 36 4c 18 05 co a8 .Q@ 0020 01 15 07 5b b7 ca 52 30 38 9b 79 8f ed e9 80 18 0030 00 72 2e 40 00 00 01 01 08 0a 07 17 3c ad d4 85 .r.@ |
| Publish Message (client to server) http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718037 ▼ MQ Telemetry Transport Protocol ▼ Publish Message ▶ 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0001 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 00010 00 25 102 93 40 00 ea 06 7e 05 36 4c 18 05 c0 a8 .00 0020 01 15 07 5b b7 ca 52 30 38 9b 79 8f ed e9 80 18 [R0 8.y 0030 00 72 2e 40 00 00 01 01 08 0a 07 17 3c ad d4 85 .r.@ 0040 d4 3e 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 .>0/s ensors/t emperatu re25.80 Publish Message outcoming 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 24 |
| http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html# Toc398718037 • MQ Telemetry Transport Protocol • Publish Message • 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. 0010 00 51 02 93 40 00 ea 06 7e 05 36 4c 18 05 c0 a8 .Q0 ~.6L 0020 01 15 07 5b b7 ca 52 30 38 9b 79 8f ed e9 80 18 [.R0 8.y 0030 00 72 2e 40 00 00 01 01 08 0a 07 17 3c ad d4 85 .r.@/s ensors/t 0040 d4 3e 30 1b 00 14 2f 73 65 6e 73 6f 72 73 2f 74 .>0/s ensors/t 0050 65 6d 70 65 72 61 74 75 72 65 32 35 2e 38 30 Publish Message Publish Message outcoming 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 24 |
| MQ Telemetry Transport Protocol Publish Message 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 27 Topic: /sensors/temperature Message: 25.80 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 08 00 45 00 .%"M JE. Message: 25.80 0000 00 51 02 93 40 00 ea 06 7e 05 36 4c 18 05 c0 a8 .0 |
| 0000 00 25 22 8a 86 a0 90 4d 4a a3 0e 00 86 00 45 00 < |
| Publish Message outcoming 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 24 |
| 0011 0000 = Header Flags: 0x30 (Publish Message) Msg Len: 24 |
| Msg Len: 24 |
| |
| Topic: /sensors/temperature |
| Message: 20 |
| 0000 301800142f73656e736f72732f74656d0 /sensors/tem |
| |

Annexe 1 : Configuration d'un broker (serveur MQTT) en ligne



Page 9/27

| shiftr.io Cloud Welcome nsi.touchard. Please choose a password and create your shiftr.io Cloud | shiftr.io Cloud Dashboard | Deploy | Instances Support | N nsi.touchard |
|--|--|--|---|--|
| account. nsi.touchard@gmail.com ••••••• Create Account | With shift instances. W | Welcome to io Cloud you can lau lith the Basic plan yo Deploy | shiftr.io Cloud! unch and manage multiple shiftr.io ou can get connected for free today y Instance | Dashboard Manage Groups Account Settings Sign Out |
| Setup You need to create Instances are managed in groups to o share them with other shift | a group first. consolidate billing and soon ftr.io Cloud users. | | Créer un groupe | e |
| Setup The group "nsi" has You can now continue with d Deploy Insta To change the group settings you can go account drop down in the | been created. eploying an instance. ance o to "Manage Groups" from your | | Cliquer sur « De Instance » | eploy |

| Summary Review the configuration and deploy your instance. Plan Basic \$0.00 Total per month \$0.00 | |
|--|---|
| Review the configuration and deploy your instance. Plan Basic 50.00 Total per month 50.00 Deploy Instance Instances | Cliquer sur l'instance (lien en couleur verte) |
| touchard touchard.cloud.shiftr.io | |



| | Settings | Tokens | Créer un « token » ou topic. |
|---|---|---|--|
| [| General Tokens Webhooks Embed Admin To change the instance configuration, visit the shiftrio Cloud console. | Tokens grant access to your instance. Create Token No tokens created. To connect to your instance, use as the host touchard.cloud.shiftr.io and the MQTT port 1883. The username is always touchard while the password is the above configured token secret. | |
| | Settings General Tokens Webhooks Embed Admin To change the instance configuration, visit the shiftr.io Cloud console. | New Token Create a new token to grant access to your instance. Description capteurs Secret MFmD747BIPSYIJYI The secret can only be set during token creation. Permission Full Access Tcreate Token Cancel | Renseigner la description du nouveau Token. Par exemple « capteurs » |
| | Settings General Tokens Webhooks Embed Admin To change the instance configuration, visit the shiftr.io Cloud console. | Tokens Create Token capteurs mqtt://touchard:MFmD747BIp8YIJYI@touchard.cloud.shi Full Access To connect to your instance, use as the host touchard.cloud.shiftr.io and the MQTT port 1883. The username is always touchard while the password is the above configured token secret. | La création est maintenant terminée |
| | | | |

| BTS CIEL - TP | | | | TP |
|---|--------------------------------------|-----------------|-----------------|----|
| Aller sur la page accueil de l'ir | nstance <u>https://to</u> | uchard.cloud.sh | <u>iftr.io/</u> | |
| \leftrightarrow \rightarrow C \textcircled{O} \textcircled{O} $https://touchard.clou$ | d.shiftr.io 👓 🖂 | ✿ Rechercher | III\ 🗊 🔹 | ≡ |
| Last Mossager | | Activ | o Connections | |
| 0 msgs/s | touchard touchard.cloud.shiftr.io | 0 connections | e connections | |
| | | | | |
| | | | Recent Errors | |
| | 0 | 0 errors/s | | |
| | | | | |
| | | | | |
| shiftr.io Cloud Docs Blog | | m | 2 🌣 💽 | |
| | | | | |

La configuration du Broker (Serveur MQTT) est terminée.

Exemple de configuration

| Adresse du serveur | touchard.cloud.shiftr.io |
|--------------------|----------------------------------|
| Client id | python ou micropython ou arduino |
| login | touchard |
| Mot de passe | MFmD747Blp8YlJYl |
| Topic ou token | capteurs |

Remarque :

Il est possible d'utiliser le <u>broker</u> shitr.io hors ligne et pouvant être utilise sur un réseau local sans avoir de connexion Internet. (Installation du programme shiftr-io-desktop.exe)

Il existe d'autres brokers gratuits comme <u>https://www.maqiatto.com/</u>, mais il ne dispose pas comme shiftr.io d'une interface graphique indiquant les connexions en cours.

Annexe 2 : Configuration du client sur Android (MQTT Dash)

Installer le programme MQTT dash



https://play.google.com/store/apps/details?id=net.routix.mgttdash

Configuration :



Ajouter une connexion vers le broker shiftr.io

| Free | 🗟 .ய 93 % 🗩 20:21 | Free 🤶 🔐 93 % 🗩 20:21 | | | | |
|---|-------------------|---|--|--|--|--|
| MQTT Dash | Ľ | MQTT Dash | | | | |
| Default (automatically connect on start up). Note: this option is useful if you have just one connection configured. If you have more than one connection, you can create home screen shortcut for every connection. To create shortcut long press on any connection in connections list. Keep screen on when connected to this broker Allow metrics management. If disabled, you can't add, edit, delete or rearrange metrics. This serves as protection from | | Enable connection encryption (SSL/TLS). Note: if server certificate is self-signed, you need to install it to your device or enable option below, otherwise connection will fail. If server certificate issued by a known Certificate Authority (CA), it will work out of box, without installing to you device. Also don't forget, that MQTT servers have different ports for plain and SSL/TLS connections. This broker uses self-signed SSL/TLS certificate. I trust this certificate at my own risk. User name | | | | |
| Name | | touchard | | | | |
| capteurTest | | User password | | | | |
| Address | | <u></u> | | | | |
| touchard.cloud.shiftr.io | | Client ID (must be unique) | | | | |
| Port | | mqttdash | | | | |
| 1883 | | Tile size Small | | | | |
| \triangleleft O | | \triangleleft O \Box | | | | |

| | | ((r | 10:12 |
|-------------|---|-----|----------|
| MQTT Dash | ! | Ô | \oplus |
| sensorsTest | | | |

Puis ajouter le(s) topics nécessaires.



Exemple pour le Topic température (Ne pas oublier de décocher « Enable publishing »)

| Free | 🤶 ,ຟ 93 % 🗩 20:21 | | | | |
|---|--|--|--|--|--|
| MQTT Dash | | | | | |
| This metric is inter text (e.g. temperat expected to be stri Name | nded for displaying payload ure displaying). Payload is ng. | | | | |
| temperature | | | | | |
| Topic (sub) | | | | | |
| /capteurs/temperature | | | | | |
| Extract from JSON JSON format), e.g. documentation at <u>https://github.com</u> /master/README. | path (if payload is in : \$.level.value. JSON path the URL below: /jayway/JsonPath/blob md | | | | |
| Enable publist | hing | | | | |
| Prefix | Postfix | | | | |
| Main text size | | | | | |
| \bigtriangledown | 0 | | | | |

Annexe 3 : Configuration du client sur Android (MQTT panel)

Installer le programme IoT MQTT Panel



https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod

Ce programme un plus convivial que MQTT dash et possède des widgets. (Élément de base de l'interface graphique d'un logiciel : fenêtre, barre d'outils, par exemple). Une fois installé, cliquer sur

ajouter une connexion icone 🔍, puis configurer le broker shiftr.io

| Configuration de la connexion au l | Broker Sélectionner MqttPanel |
|---|---|
| Free 😤 all 93 % ← Edit Connection | [∞] ■ 20:23 Free <a> II 93 % ■ 20:23 |
| Connection name * MqttPanel | |
| Client ID 1 smartphone | MqttPanel |
| Broker Web/IP address * touchard.cloud.shiftr.io | (2) |
| Port numb Network protocol 1883 TCP ~ | · (2) |
| Dashboard list | Ð |
| Nucleo | |
| Additional options 2 | ✓ |
| Connection time Keep alive 30 60 | |
| Username Password touchard •••••• | 3 |
| Add will message | • |
| Notify on disconnect | |
| Connect automatically 4 | 4 6 3 |
| CANCEL | |
| | |



| Ajouter un panel en cliquant sur l'icône rouge + | Sélectionner Gauge | | Configu | rer le topic | | |
|---|--------------------|-----------------------------------|---------|--------------------------------|--------------------|-------------------------------|
| Free | Free Sel | ≈88% — 1 ect panel type to add | 11:06 | Free ← Ed | lit panel | ରି _{ଥା} ∥93 % 🗩 20:: |
| | - 9a | Node Status | • | Panel nan Tempér Topic * | ne* rature | |
| | =,∕ | Combo Box | 2 | Payload m | ni Pa | yload ma |
| | ۲ | Radio Buttons | ? | Sho | w received tim | estamp |
| | . | LED Indicator | ? | Unit ° C | Q. 0 | oS 👻 |
| | <u>}</u> | Linear Progress | • | Color se | ectors |) — #ff5722 |
| | ა | Circular Progress | ? | 0 t | to <u>33,33</u> to | 66,67 to 50 |
| | i | Vertical Meter | ? | 🗌 Enat | ole notification | 0 |
| | \sim | Gauge | 2 | 🗌 Payl | oad is JSON D | ata |
| \triangleleft 0 \Box | | | | | CANC | |

TP

La configuration du logiciel sur le smartphone est terminée. L'utilisateur est prêt à recevoir l'information de température en provenance de la carte IOT.



Exemple de configuration en lien avec le broker :

| Name | MqttPanel |
|--------------------------|--------------------------|
| Adresse du serveur | touchard.cloud.shiftr.io |
| Client id | Smartphone |
| User name (key) | touchard |
| User password (password) | MFmD747Blp8YlJYl |
| Topic name | capteurs/temperature |

Port 1883 : communication non sécurisée (données en clair sur le réseau) par défaut dans ce document.

Port 8881 : communication non sécurisé SSL (Secure Socket Layer) / TLS (Transport Layer Security)

Annexe 4 : Configuration du client sur IPhone (EasyMQTT)

Installer le programme EasyMQTTsur le téléphone



EasyMQTT (a) Luca Kaufmann Conçu pour iPad ***** 5,0 + 1 nate Gratuit - Inclut des achats intégrés

https://apps.apple.com/fr/app/easymgtt/id1523099606

Ajouter une connexion vers le broker shiftr.io, puis tester la connexion



2021-01-17

2021-01-17

*

Topic: ca Message: 20 15:15:26

15:15:15

*

 \star

Add to Siri (unlock with EasyMQTT Plus)

Publish

4

Annexe 5 Utilisation de Node-red

Node-RED est un outil de développement basé sur une programmation visuelle pour connecter un ensemble de périphériques matériels, des API et des services en ligne dans le cadre de l'Internet des objets.

3.1 Installation de Node-red sous Windows :

- Commencer par installer node js

https://nodejs.org/en/



https://nodered.org/docs/platforms/windows

- Puis dans la console saisir les commandes suivantes :

node --version && npm -version

```
npm install -g --unsafe-perm node-red
```

node-red

Ne pas fermer la console

| 📼 node-red | < Node-RED: Running on Wind: X 🌒 Nodejs X 🗧 Node-RED X 🛨 |
|---|---|
| (berunt 1 Success: "C:\Users\anthonu\AnnData\Roaming\num\node_modules\node_red\no | ← → C [*] û (C 127.0.0.1:1880) flow/1798c2ef.b17bdd |
| de Marte de sécurité Windows | Node-RED |
| add C:N 12 1 We lu We lu C:N 12 1 We lu C:N C:N C:N C:N C:N C:N C:N C:N | 4 filter nodes Flow 1 V input 1 |
| Le Pare-feu Windows a bloqué certaines fonctionnalités de Node.js: Server-side JavaScript sur tous les réseaux publics et privés. 2 Non : <u>Node.js: Serve</u> 2 Éditgur : Node.js C:\program files\nodejs\node.exe | inject inject status i status i e |
| 12 Ghemin C: program files/nodejs/node.exe | ink O |
| 12 1 Autoriser Node.js: Server-side JavaScript à communiquer sur ces réseaux : Réseaux privés, tels qu'un réseau domestique ou un réseau d'entreprise |) męt 0 |
| 12 1 12 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | top 0 websocket 0 |
| 12 Siles programmes sont autorisés via un pare-feu, quels sont les risques encourus ? | 6 tep 0 (1) udp 0 |
| You: Te Autoriser l'accès Annuler | ~ output |
| file will not be recoverable, you will have to delete it and re-enter your credentials. | debug |
| You should set your own key using the 'credentialSecret' option in your settings file. Node-RED will then re-encrypt your credentials file using your chosen key the next time you deploy a change. | C Inde C |
| 12 Feb 10:05:06 - [info] Starting flows 12 Feb 10:05:06 - [info] Started flows 12 Feb 10:05:06 - [info] Server now running at http://127.0.0.1:1880/ | Chtp response C websocket |

Ouvrir un navigateur puis saisir l'adresse suivante :

http://127.0.0.1:1880

Ne pas fermer la console.

3.2 Add-ons

Ajouts d'outils de gestion graphique.

| User Settings | | | | Deploy - | Ξ |
|---------------|--|------------------|---|---|---|
| | | Close | 4 | View | 1 |
| View | Nodes | | 4 | Import | |
| Keyboard | | sort: a-z recent | • | Export | |
| Palotto | ۹ node-red-dashboard 4 | 2 / 1859 🗰 | | Search flows | |
| Falette | node-red-contrib-ui_list Node-RED UI widget node for simple list 0.1.3 3 months ago | install | • | Configuration nodes Flows Subflows | |
| | node-red-dashboard A set of dashboard nodes for Node-RED 2.13.2 # 1 week ago | 5 install | | Manage palette Settings | 2 |
| | | | | Keyboard shortcuts Node-RED website v0.19.5 | |

Il en va de même pour d'autres outils en connaissant le nom du fichier à installer.

BTS CIEL - TP 3.2 Installation sous linux

curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash sudo apt-get install -y nodejs
node -v
sudo npm install -g --unsafe-perm node-red
node-red

3.3 1er pas

Tracer le diagramme suivant :

| Node-RED | | | | | | Deploy 🔻 |
|-----------------------|----------------------|-------|-------------|----------|------------------|-------------------|
| Q filter nodes | Test 1 | | | + | i info | i 🕀 🔝 |
| ✓ input | | | | ^ | ✓ Information | on |
| ⇒ iniect 2 | | | | | Flow | "1798c2ef.b17bdd" |
| | 2 | | 3 | E | Name | Test 1 |
| catch 📮 🗐 | | ro:20 | temperature | | Status | Enabled |
| status | - /sensors/temperatu | ie.20 | connected | | Flow Designation | cription |
| ≱ link | | | | | None | |
|) mqtt <mark>4</mark> | 4 | | | | | |
| http |)) tempera | ature | msg.payload | | | |
| websocket | connected | | 5 | | | |
|)) tcp | | | | | | |
| 🕴 udp | | | | | | |
| ∽ output | | | | | | |
| debug 5 | | | | | | |
| link 🔅 | | | | | | |
| mqtt)) 3 | | | | | | |

Configuration du broker shiftr.io :

| Edit mqtt out node | Edit mqtt out node > Edit mqtt-broker node |
|---|--|
| Delete Cancel Done | Delete Cancel Update |
| v node properties | Name shiftr.io |
| Server shiftr.io | Connection Security Messages |
| Topic Topic | Server broker.shiftr.io Port 1883 |
| ⊕ QoS O ▼ ⑦ Retain | Enable secure (SSL/TLS) connection |
| Name temperature | Client ID Node-red |
| Tip: Leave topic, gos or retain blank if you want to set them via msg | ⊘ Keep alive time (s) 60 |
| properties. | Use legacy MQTT 3.1 support |
| temperature | Connection Security Messages |
| Connected | Lusername weatherSensors |
| | Password |

Ici la valeur est fixe, 20° pour tester le fonctionnement.

| Edit inject no | de | Edit mqtt in no | ode |
|---|---|-----------------|-------------------------|
| Delete | Cancel Done | Delete | Cancel Done |
| ✓ node prop | erties | v node prop | erties |
| Payload | ▼ ^a z 20 | Server | shiftr.io 👻 |
| Topic 📰 | /sensors/temperature | 📰 Торіс | /sensors/temperature |
| | Inject once after 0.1 seconds, then | 🕀 QoS | 0 |
| C Repeat | none 💌 | Name | temperature |
| Name Name | Name | | |
| Note: "inter "interval" sh See info box | val between times" and "at a specific time" will use cron. nould be less than 596 hours. x for details. | |) temperature connected |

Test du diagramme :

| 1 | 3: clique /sensors/temperature:20 temperature) publish |
|--|---|
| 13/02/2019 à 14:13:54 node: 6a55b4e9.278984 /sensors/temperature : msg.payload : string[2] "2.0" |) temperature msg.payload I subscribe |
| 13/02/2019 à 14:29:43 node: 6a55b4e9.278984 /sensors/temperature : msg.payload : string[2] "20" | 20 Node-red = 14:33:47 - Q0 NR Node-red humidite framesors temperature pression |

- 1- Cliquer sur Deploy
- 2- Puis sur debug pour voir les messages de debug
- 3- Enfin sur le bouton envoi (symbole inject)

L'information de température 20 ° test envoyé vers le broker, puis revient.

3.4 Gestion graphique

Utilisation d'un vu mètre (gauge)

| dast | nboard | slid | er 🔶 (| colour p | vicker | cha | t 🗹 🖻 | |
|--------------------------|---------------|-------------------|-----------------|------------------------|--------|----------------------------|------------------------------|----------------|
| € ℃ | button | 0 <u>123</u> nume | eric | form | n | e audio | out | |
| | dropdown | abc text in | nput 🔶 🛛 | text | abc | notifica | ation 🖂 | |
| | switch | date p | icker 🖕 🛛 | gauge | 0 | ui cor | itrol 🜓 🕨 | |
| Edit gauge node | • | | Edit gauge node | > Edit dashboard group | node | | -/ | eploy 👻 🗧 |
| Delete | | Cancel Done | Delete | | Cancel | Update | dashboard | i 🛞 🔟 |
| v node proper | ties | | Name | Tomp | | | Layout Site | Theme 🖸 |
| ⊞ Group ■ | [Home] Temp | • | Tab Width | Home | | • • | Tabs & Links ♠ ✓ ☑ Home ■ | × + tab + link |
| 🔳 Туре | Gauge | | | Display group name | | | v iii lemp ∎ Iii gauge ∎ | |
| 1 Label | gauge | J | | Allow group to be coll | lapsed | | | |
| | {{value}} | | | | | | | |
| £ Units | units | | Delete | | Cancel | Update | | |
| Range | min 0 max 50 | | Name | Home | | | | Ŧ |
| Colour gradient | | | 🗈 lcon | dashboard | | | | |
| Sectors | 0 optional op | tional 50 | Ø State | C Enabled | |) temperature connected | msg.paylo | bad |
| Name | | | 🌮 Nav. Menu | C Visible | | _ | gauge (| ີ |

Ou se trouve le vu mètre une fois que l'on clique sur **Deploy** ?

| Deploy - | \leftarrow \rightarrow C \textcircled{a} (i) 127.0.0.1:1880/ui/#!/0 \bigtriangledown \textcircled{b} \bigcirc Rechercher \checkmark \checkmark | > ≡ |
|--|--|----------|
| i i i i | Home | |
| Layout Site Theme Tabs & Links Image: Site of the site | Тетр | ^ |
| ✓ ☑ Home | gauge | |
| ∽ ⊞ Temp ⊠ gauge | 20 0 units 50 | E |

Une nouvelle fenêtre dans le navigateur s'ouvre avec le vu mètre.

Inscription sur le site https://www.twilio.com/



Trial version : 200 sms gratuits, puis payant : 7,6 cts /sms

Exemple de dashboard une fois l'inscription terminée

Twilio vous fourni un numéro de téléphone : ici Trial number

Il est possible d'envoyer des sms que sur des numéros de téléphones vérifiés : (le vôtre lors de l'inscription).

En vert le crédit sms restant

| Dashboard | alecren@orange.fr's Account Dashboard | | | | |
|-------------------|---|--|--|--|--|
| Billing | Project Info | | | | |
| Osage Settings | Go to billing | Here's how your Twilio Trial account works: | | | |
| Upgrade | \$13.196 +33644643297 • Need more numbers? | You can send messages and make calls to <u>verified numbers</u>. Messages and calls include a note about this coming from a "Twilio trial account." | | | |
| | ACCOUNT SID | Learn more about $\underline{your\ trial}\ \pi\ or\ \underline{upgrade}\ to\ remove\ restrictions.$ | | | |
| | AC1a7667786a1e516477d61e796dbebdd1 | PROJECT NAME | | | |
| | AUTH TOKEN | alecren@orange.fr's Account | | | |
| | Show | | | | |

Installation dans Node-red

| User Settin | js | Deploy 👻 |
|---------------------|--|---|
| | Clos | e View |
| View | Nodes Install sort: a-z recent | Import ♂ Export |
| Reyboard Palette | Q twilio 4/18 | 59 × Search flows |
| | install install install | Configuration nodes Flows Subflows |
| | № node-red-node-twilio ∞ A Node-RED node to send SMS messages via the Twilio service. 0.1.0 <u>●</u> 11 months ago | Manage palette |
| ~ m | twilio (3) | Settings Keyboard shortcuts Node-RED website |
| | twilio 🕀 | Node-RED website v0.19.5 |

|)) temp connec | ted ale | | |
|-------------------|--------------------------------|---------------------|------------------------------------|
| Edit twilio out n | ode | Edit twilio out noo | de > Edit twilio-api node |
| Delete | Cancel Done | Delete | Cancel |
| ✓ node proper | rties | Account SID | AC1a7667786a1e516477d61e796dbebdd1 |
| La Twilio | anthony 🔹 🖉 | From 🗹 | +33644643297 trial number |
| i≡ Output | SMS | 🔒 Token | ••••• |
| ⊠ То | +33 Votre N°de portable | Name | anthony |
| Name | ale | | |
| 1 -⁄_ De | eploy - 2 - /sensors/temperat | ure:20 | 3 Arrivée du sms |

Il existe de nombreuses possibilités avec Node-red

Ce document n'a pas pour but d'expliquer toutes les possibilités mais de démarrer rapidement

Liens :

 $\underline{http://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/8054/8054-objets-communicants.pdf$

https://www.youtube.com/results?search_query=node+red