



# Projeto 07

## Coisas Remotas – Prática

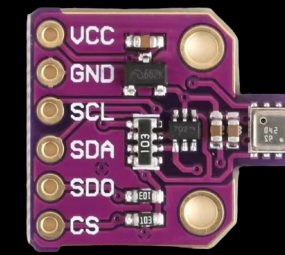
Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG4051 – Projeto Internet das Coisas

# Testes Iniciais

1

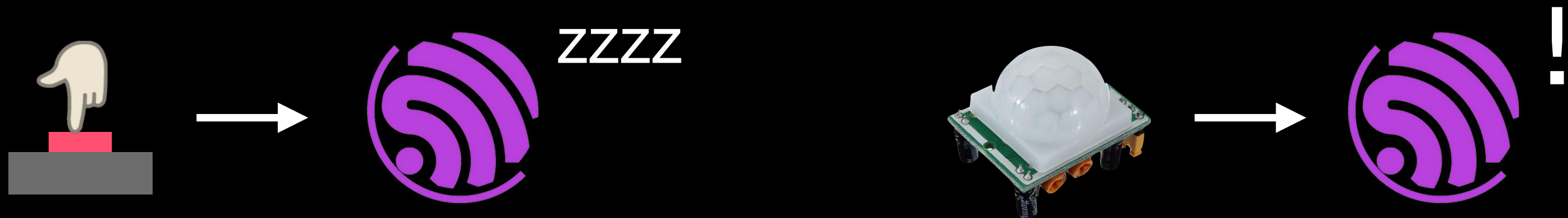
A cada 5s, leia a temperatura, umidade, pressão e resistência do gás e imprima os valores na Serial.



25.4 °C  
70% de umidade  
1020 hPa  
30 kOhm

2

Ao apertar o botão cima (pino 1), coloque o ESP32 para hibernar. Ao receber um HIGH do pino 21 (sensor de movimento), acorde o ESP32. Veja a diferença de consumo de corrente.



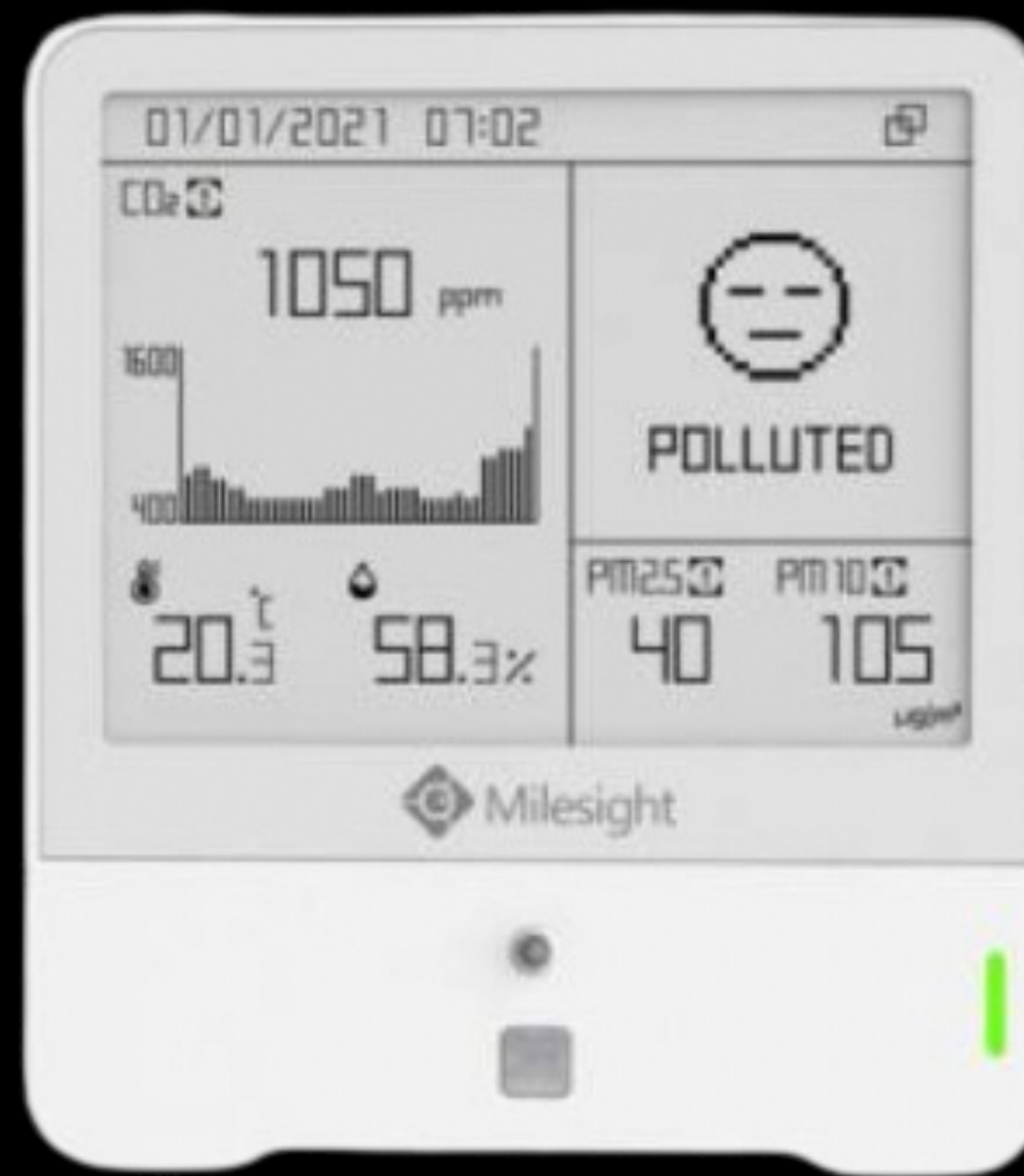
3

Ao apertar o botão central, envie o seu nome pelo LoRa com o AT+SEND=. Observe o payload recebido no MQTTX. Acesse [base64decode.org](https://base64decode.org) para decodificar o payload e verificar se ele está correto.

v3/iot/devices/**ID\_DO\_DISPOSITIVO**/up

Testes Iniciais

Protótipo

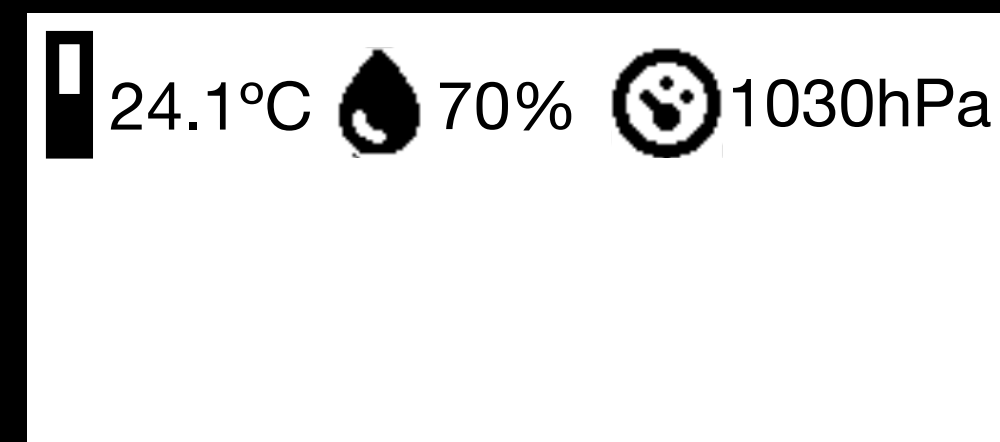


Monitoramento das Condições Ambientais Remotas

1

Após passar 5 segundos do começo do programa, leia a temperatura, umidade, pressão e resistência do gás e mostre-os no display ePaper no formato abaixo.

↳ DICA: use a fonte `u8g2_font_open_iconic_all_4x_t` e desenhe um termômetro com dois retângulos.



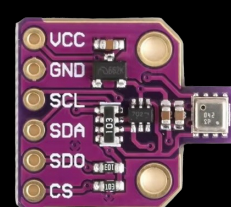
2

Após atualizar o display, codifique os dados lidos usando o CayenneLPP

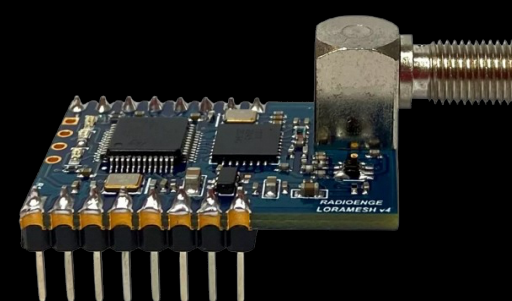
↳ DICA: no caso da resistência do gás, use o `addAnalogInput`.

Envie esses dados binários para o LoRa com o `AT+SENDB=`.

Verifique o resultado no MQTTX.



`AT+SENDB=...`



**MQTT X**

3

No Node-RED, ao receber o tópico do MQTT do seu dispositivo, insira os dados na tabela `medicoes` do PostgreSQL.

Verifique o resultado no Beekeeper Studio.



Protótipo: Coleta de Dados via LoRaWAN

4

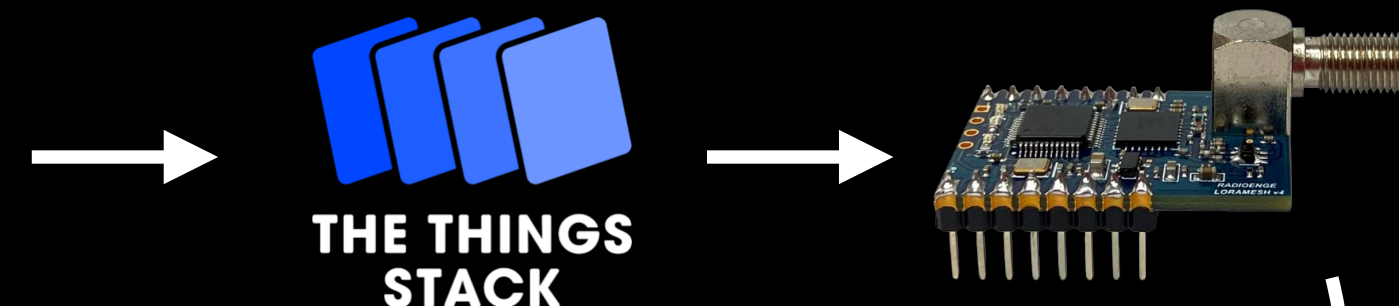
Ao receber uma mensagem do módulo começando com "RX:", pegue a substring com o hexadecimal, converta-o para texto e exiba-o na parte inferior do display.

↳ DICA: use a função `indexOf` do Arduino para encontrar o 2º ":" na string.

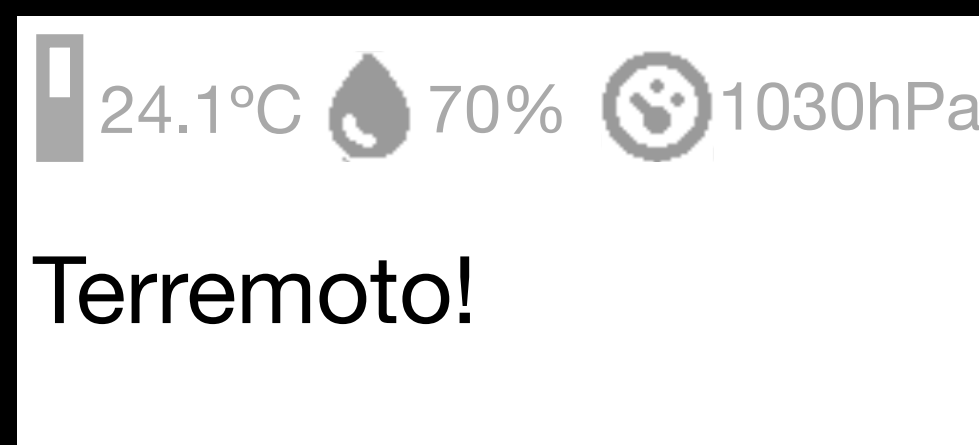
Teste o resultado publicando um tópico de downlink no MQTTX.

v3/iot/devices/**ID\_DO\_DISPOSITIVO**/down/push

```
{
  "downlinks": [{
    "f_port": 5,
    "frm_payload": "VGVycmVtb3RvIQ==",
    "priority": "HIGH"
  }]
}
```

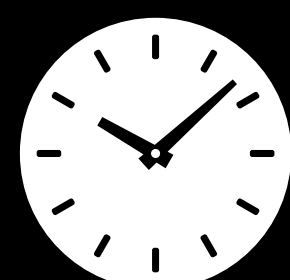


RX:54657272656D6F746F21:2:-70:28



5

Após 10 segundos do começo do programa, coloque o ESP32 em repouso. Acorde 5 segundos depois disso.



10s



5s



Protótipo: Downlink e Repouso

# Integração



1

Crie um novo dashboard no Grafana com os dados da sua bancada.

Crie uma visualização tipo Stat para mostrar o valor atual do gás em k $\Omega$ .

Crie visualizações tipo Gauge para mostrar o dado mais recente de umidade (em %) e temperatura (em °C).

Coloque valores mínimo e máximo + faixas de cores para a temperatura.



2

Crie uma variável no dashboard para escolher o número da bancada.

Filtre os dados da visualização pela bancada escolhida.

Número da Bancada	4
-------------------	---

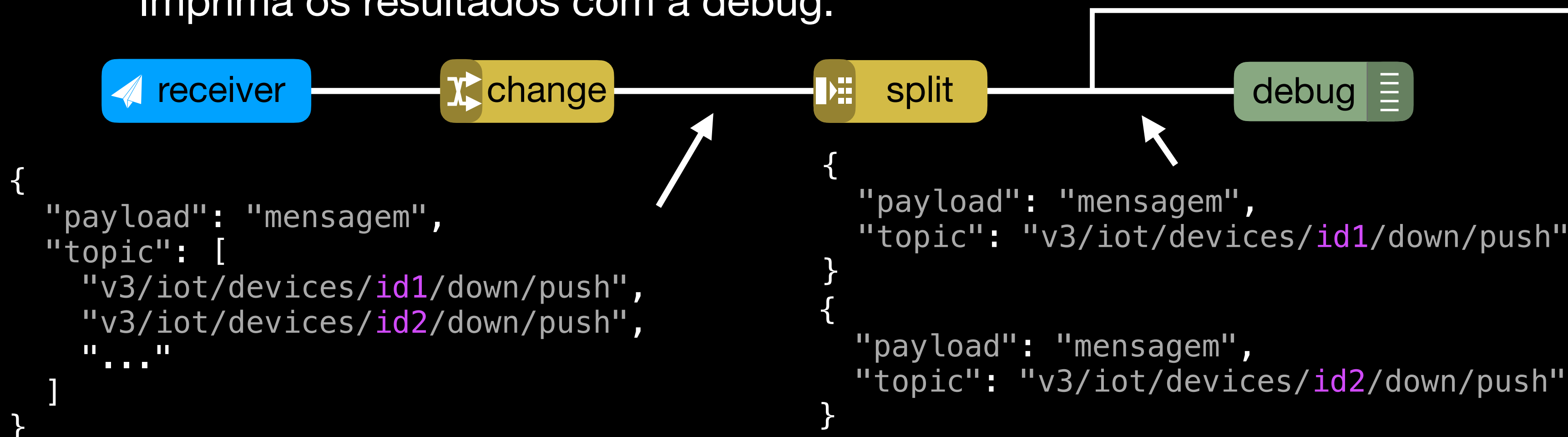
3

No Node-RED, monitore as mensagens recebidas do Telegram.

Use o bloco Change para colocar a mensagem no payload e criar um JSON de lista de tópicos downlink para cada dispositivo.

Use o bloco Split para gerar mensagens separadas para cada tópico.

Imprima os resultados com a debug.



4

Use o bloco Base64 para converter o payload para base64.

Use o bloco Template para montar o JSON da mensagem de downlink.

↪ DICA: use 3 chaves em vez de 2 ( {{{payload}}} ).

Publique o resultado no MQTT.



Integração: Envio de Alerta do Telegram para Todos os Dispositivos

Desafio

# CayenneLPP

canal 1	temperatura	272 (27.2°C)	canal 2	temperatura	255 (25.5°C)	canal 1	movimento	true
01	67	01 10	02	67	00 FF	01	66	01

Eu podia enviar só os dados puros,  
para diminuir ainda mais o payload...



Diminuição do Tamanho do Payload

1

Crie uma função que receba um dado inteiro e um total de bytes e retorne uma String com dígitos hexa.

↳ DICA: use a divisão por 256 ou deslocamento de bits dentro de um for e converta gere a string hexa de modo similar ao que fizemos no Cayenne.

```
String gerarHexa(int dado, int totalBytes)
    gerarHexa(31, 1) → "1F"
    gerarHexa(31, 2) → "001F"
    gerarHexa(1000, 2) → "03E8"
```

2

Chame a função para a temperatura multiplicada por 10 (com 2 bytes), a humidade multiplicada por 2 (com 1 byte),

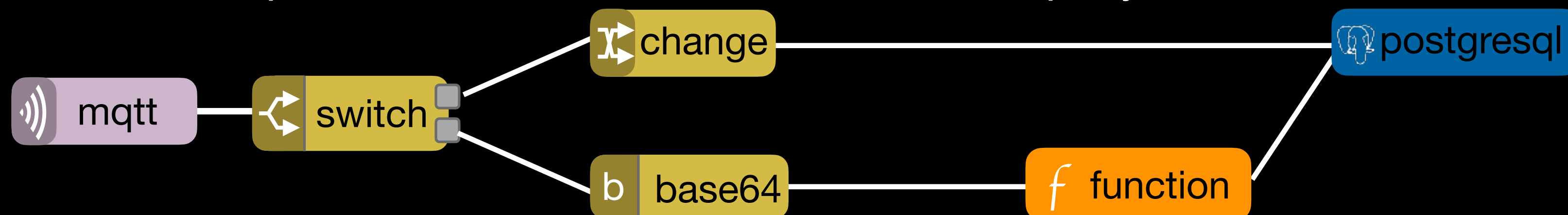
Envie a string com tudo isso para o módulo no lugar do Cayenne.

3

No Node-RED, verifique se o decoded\_payload é diferente de nulo.

Se for, faça o mesmo caminho de antes.

Se não for, use o bloco base64 para converter o frm\_payload para um array de bytes. Use então o bloco Function para combinar os elementos de volta para números e colocá-los dentro do queryParameters.



Desafio



[janks.link/iot/projeto07.zip](https://janks.link/iot/projeto07.zip)

Material do Projeto 07