

Tractian desafio hardware IoT operation

A ideia dessa solução é utilizar o modelo publish, subscribe do protocolo MQTT, comum para soluções IoT, para controlar o envio e recebimento de informações dos sensores de campo para o servidor.

O modelo inicial segue o seguinte esquema:

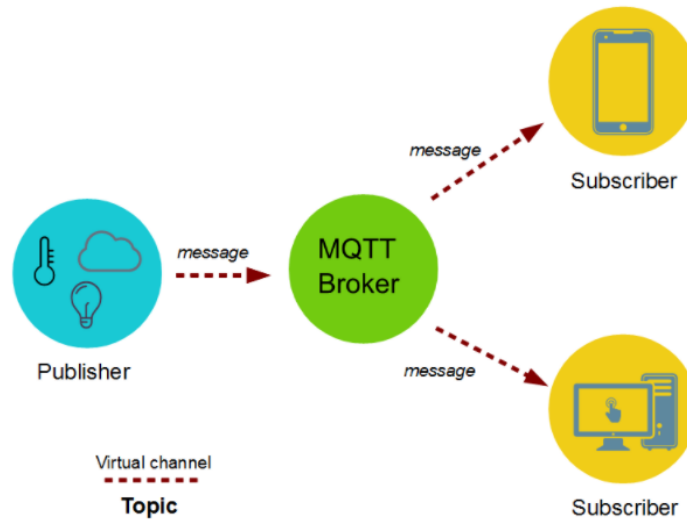


Imagem 1: Esquema básico protocolo mqtt

Onde consideramos Publisher os sensores de interesse em campo e subscriber aqueles que desejam monitorar os equipamentos de alguma forma. O broker funciona como um intermediador das mensagens e fica alocado no servidor.

Para desenvolver minha solução e para melhor entendimento do fluxo de funcionamento optei por uma ferramenta de programação gráfica, o node-red.

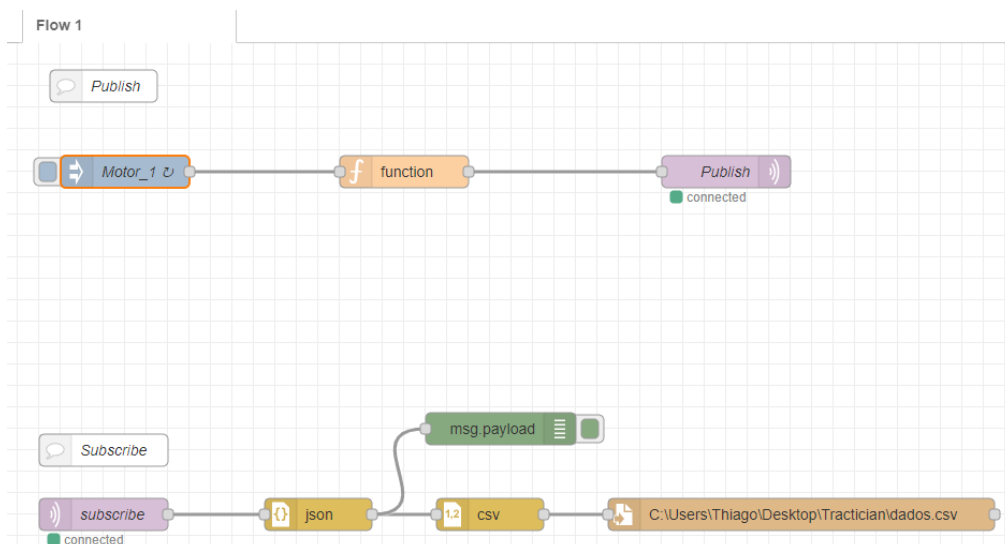


Imagem 2 : Fluxo de funcionamento do programa

Publish:

Para fins de teste publish e subscribe estão alocados no localhost

A primeira etapa é definir os tópicos de interesse no processo, nessa solução consideramos “Motor_1” como um equipamento monitorado por alguns sensores

Edit inject node

Delete Cancel Done

Properties

Name Motor_1

msg. topic = sensores/motor1

msg. payload.tempe = 100

msg. payload.vibrac = 10

msg. payload.ligado = true

+ add inject now

☐ Inject once after 0.1 seconds, then

Repeat interval

every 5 seconds

☒ Enabled

Imagem 3: Representação dos sensores monitorados

Uma solução real oferecida pela Tractian o tópico deveria ser formado pelo, Número do Chip, Código do Sensor e Código do Módulo.

Exemplo: sensor: numerodochip/codigosensor/codigodomodulo

Além de camadas de segurança que são disponibilizadas pelo protocolo

Exemplo: mqtt://broker.io/identificação usuario/ sensor/tipo de informação

Function:

O bloco Function foi utilizado para simulação do funcionamento de um sensor com valores aleatórios e um exemplo de como alertas poderiam ser criados e monitorados a partir de um worker programado pelo desenvolvedor da solução

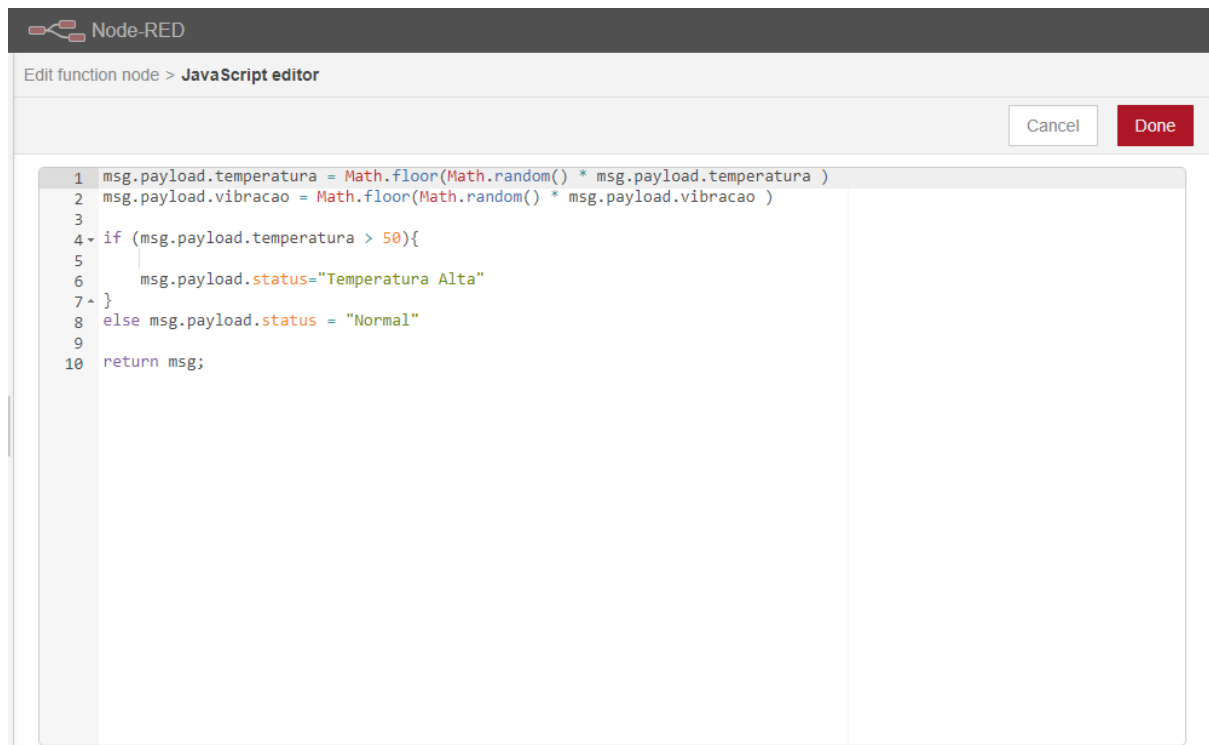


Imagem 4: Linhas de código do bloco Function

Após essa etapa as informações são enviadas do publish para o broker, para utilizar essa solução foi instalado o broker mosquitto no servidor local e recebidas pelo subscribe que é assinante do tópico de interesse.

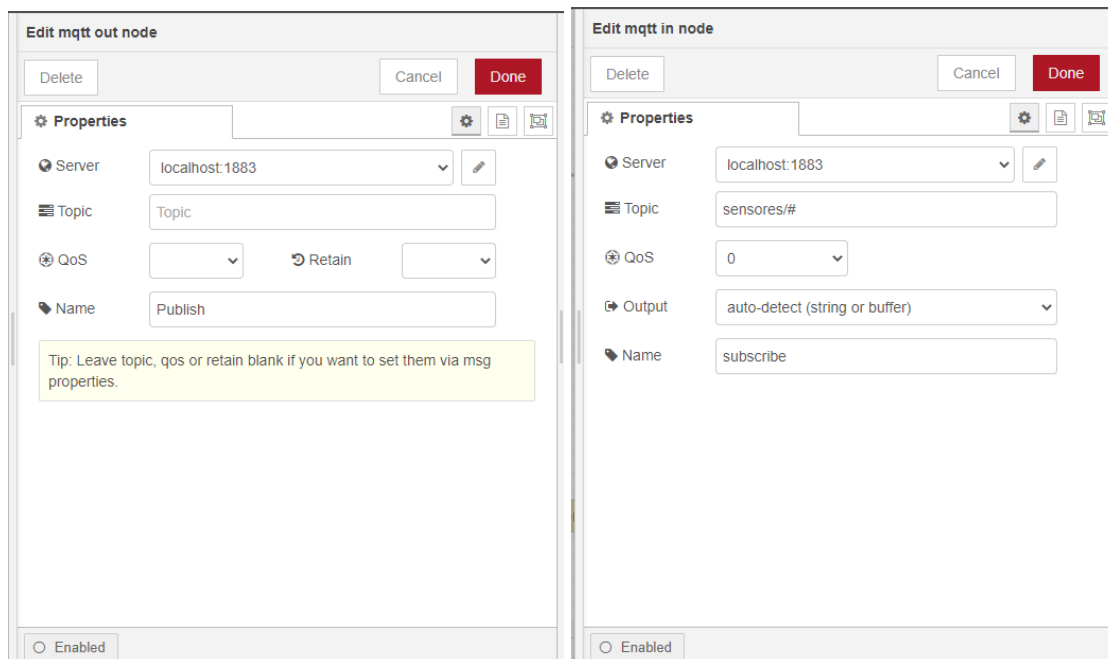


Imagem 5: Publish e Subscribe

Tratamento de dados:

Após as etapas de comunicação, os dados poderão ser armazenados num servidor local ou na nuvem dependendo das características do processo.

Nessa solução os dados disponibilizados pelos sensores que monitoram o “Motor_1”, são armazenados em um arquivo CSV a cada novo ciclo de envio de mensagem.

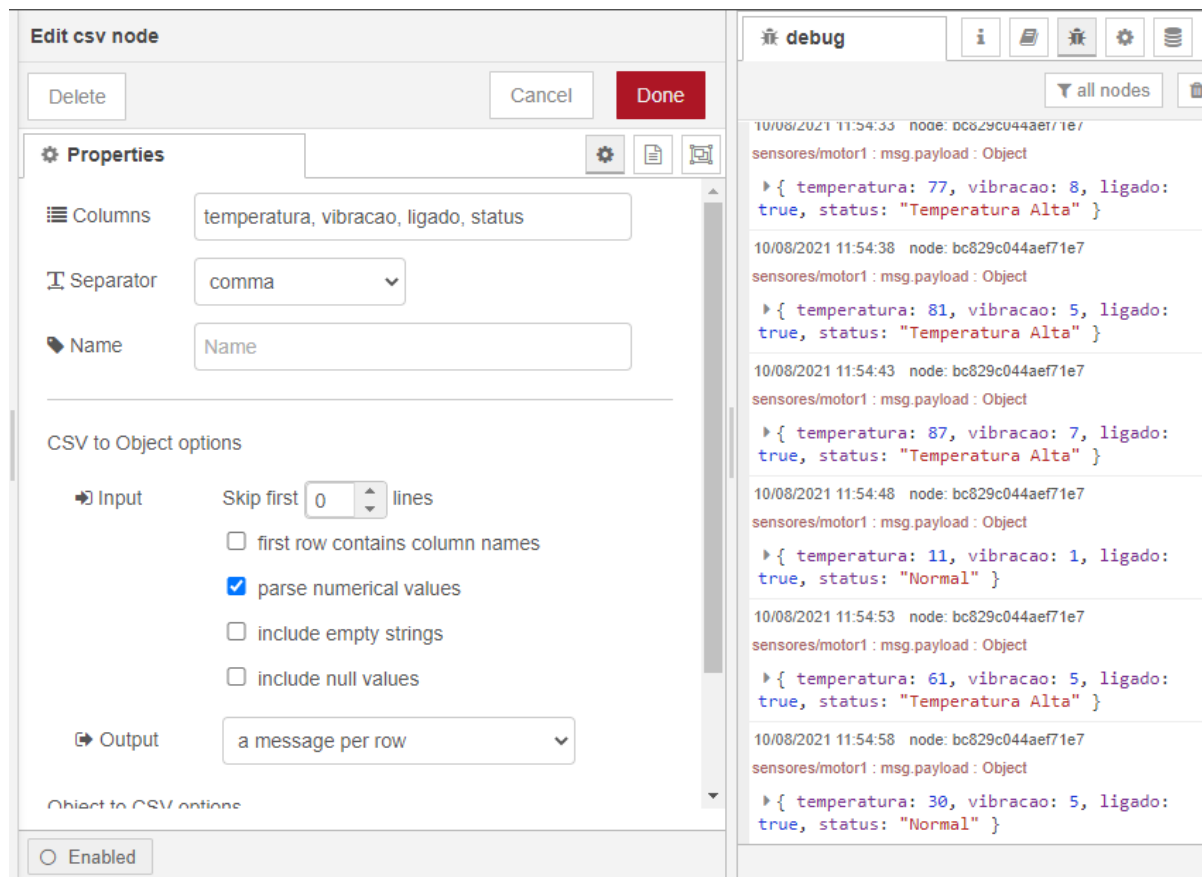


Imagem 6: Dados enviados para arquivo CSV

dados

Arquivo Editar Ver Inserir Formatar Dados Ferramentas Complementos Ajuda A última edição foi há alguns segundos

100% RS % .0 .00 123 Arial 10 B I S A

	A	B	C	D	E	F
1	temperatura	vibracao	ligado	status		
2						
3						
4	80	8	TRUE	Temperatura Alta		
5						
6	7	2	TRUE	Normal		
7						
8	69	8	TRUE	Temperatura Alta		
9						
10	18	0	TRUE	Normal		
11						
12	79	2	TRUE	Temperatura Alta		
13						
14	71	1	TRUE	Temperatura Alta		
15						
16	12	9	TRUE	Normal		
17						
18	91	9	TRUE	Temperatura Alta		
19						

+ dados

Imagem 7: Dados gerados pelos sensores

A partir desse momento o valor pode ser gerado a partir dos dados, poderíamos utilizar inteligência artificial para descobrir melhor momento para troca de componentes, descobrir melhor eficiência do motor e tirar o melhor aproveitamento possível do processo, e também gerar ferramentas gráficas para utilizar no monitoramento do processo como um sistema supervisorio.

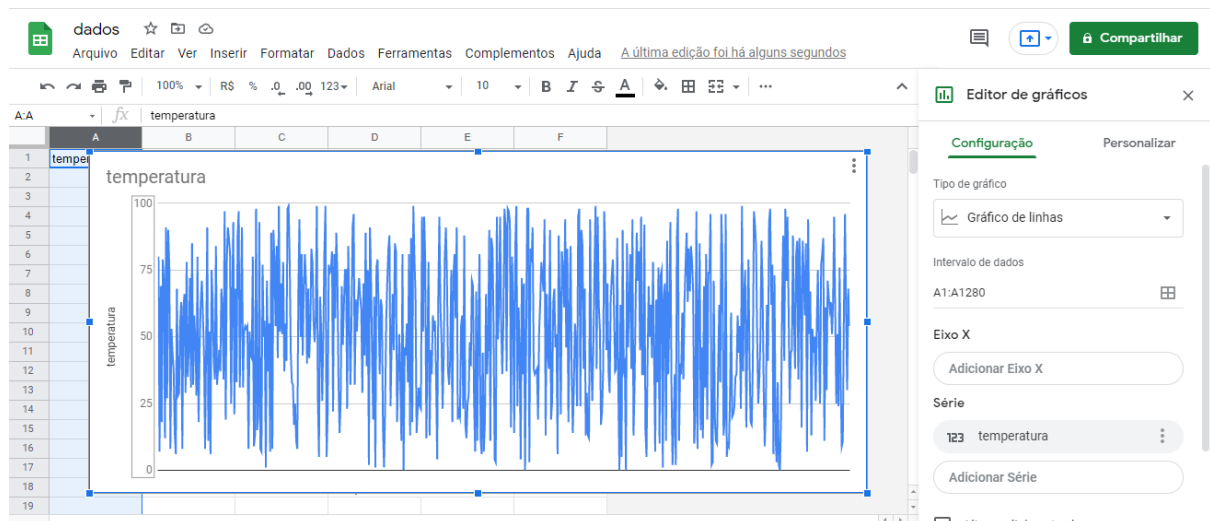


Imagem 8: Gráfico básico gerado a partir dos dados do processo

Espero que consigam avaliar plenamente a solução, qualquer dúvida estou disponível para contato nos números (11) 97324-3010 whatsapp e (11) 2552-8881.

Os arquivos e o flow estão disponíveis:

https://github.com/ThiagoSiSa/Desafio_IoT_Traction