

Integrantes do grupo:

Cassio Gaspar Valezzi – RM 551059

Rafael Perussi Caczan – RM 93092

Octávio Hernandez Chiste Cordeiro – RM 97894

Italo de Souza – RM 551500

Semáforo Inteligente: Don't Stop!

Nosso sistema consiste em uma série de semáforos interconectados e equipados com sensores para possibilitar uma tomada de decisões eficiente, reduzindo engarrafamentos e frustrações no trânsito.

Isso será atingido através de sensores, como um sensor PIR ou ultrassônico, para que seja possível detectar a presença de veículos ou pedestres. Além disso, as informações que forem usadas durante a tomada de decisões serão armazenadas e enviadas para a nuvem em um banco de dados, sintetizando-as em informações úteis para órgãos de mobilidade urbana.

Um exemplo de como isso pode ser feito é através da plataforma TagoIO, capaz de receber mensagens através de um tópico MQTT.

Como componentes, utilizamos uma placa Arduino UNO, leds e um sensor ultrassônico.

Segue um exemplo de uso do TagoIO conectado com Node-Red através de um simulador para fazer o display das informações em um dashboard.

Dependências:

Node.JS

Node-Red [Pallette: node-red-node-serialport]

Tago.IO

SimulIDE

com0com

Biblioteca ArduinoJson

Componentes:

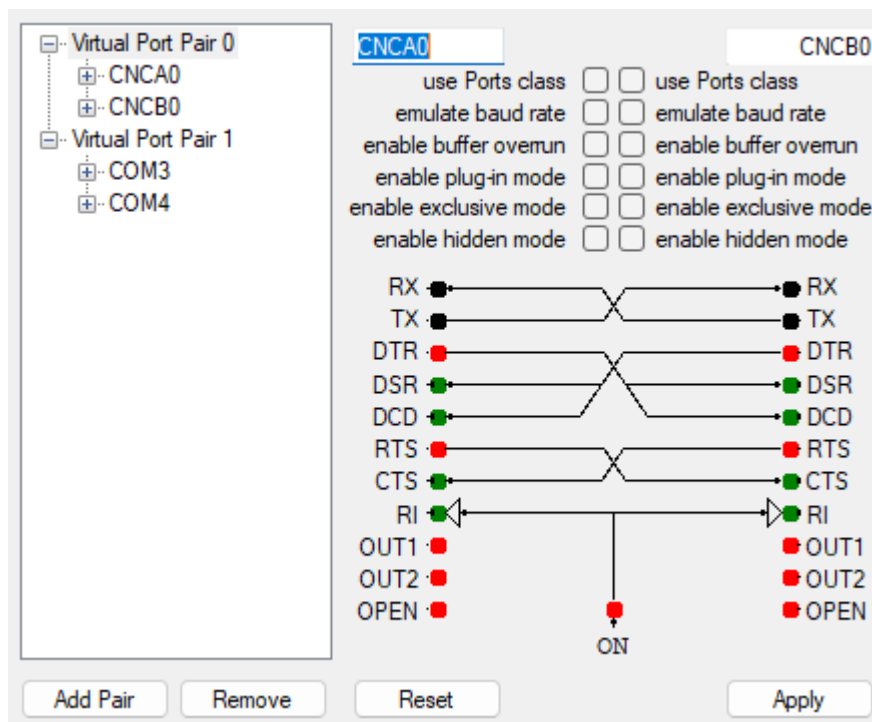
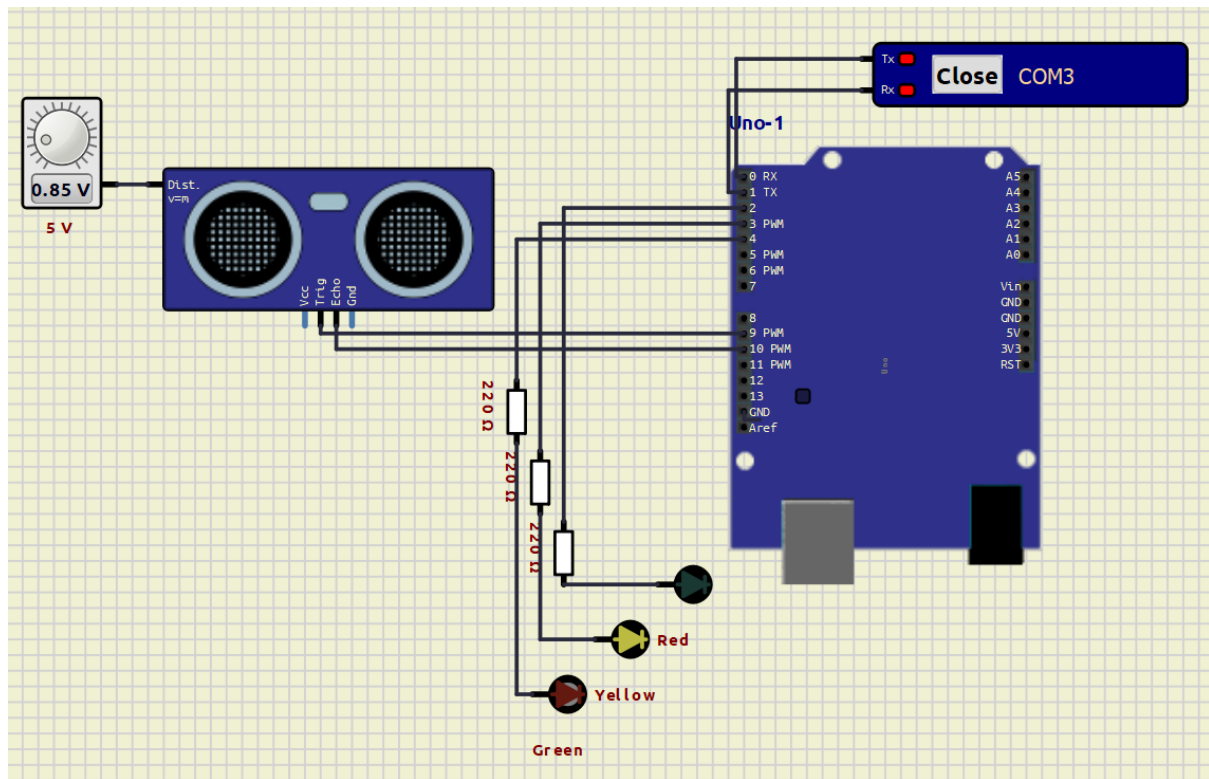
Arduino UNO

Sensor Ultrassônico

3x LEDs

3x Resistores 220 ohms

Para começar, abra o arquivo “DontStop.sim1” no SimulIDE e o com0com.



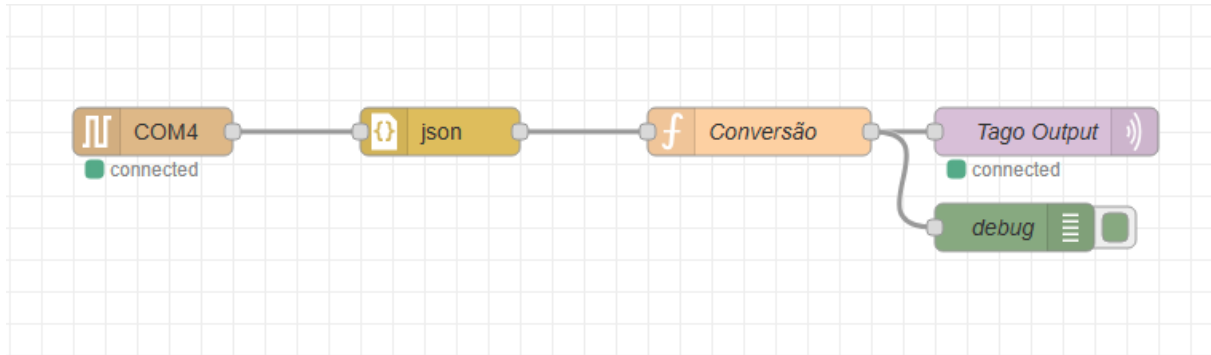
Abra uma de suas portas no serialport do simulador.

E tenha em mente o par da mesma.



Após isso, inicie o simulador e abra o Node-Red com o comando “node-red” no CMD e abrindo o endereço localhost:1880 ou 127.0.0.1:1880.

Inicie o flow contido no arquivo “nodered.json”



Configure o node “Serial IN” para que conecte-se à porta pareada a que foi utilizada no SimulIDE. Após isso, acesse o Tago.io e crie um device MQTT Custom.

Name
Don't Stop

Network
MQTT

Connector
Custom MQTT

Token & Serial Number
Token Name: Token #2
Generate
Default

Acesse o token “Default” e copie o código. Esta é sua senha para o Node-Red.

Token information



Token

741fb1f0-1b6a-44b1-8e7f-e2431caec53b

Name

Default



Expires at


Never



Close


Copy Token

Retorne agora ao Node-Red e configure o nó “MQTT Out” da seguinte forma:


 Server 

 Topic

 QoS  Retain


 Name


O tópico deve ser “tago/data/post” enquanto o servidor deve ser mqtt.tago.io, na porta 1883.


 Server Port


☒ Connect automatically

☐ Use TLS

 Protocol

 Client ID

 Keep Alive

 Session ☒ Use clean session

Na aba “Security”, escolha qualquer username e utilize o código do token como senha:

 Username

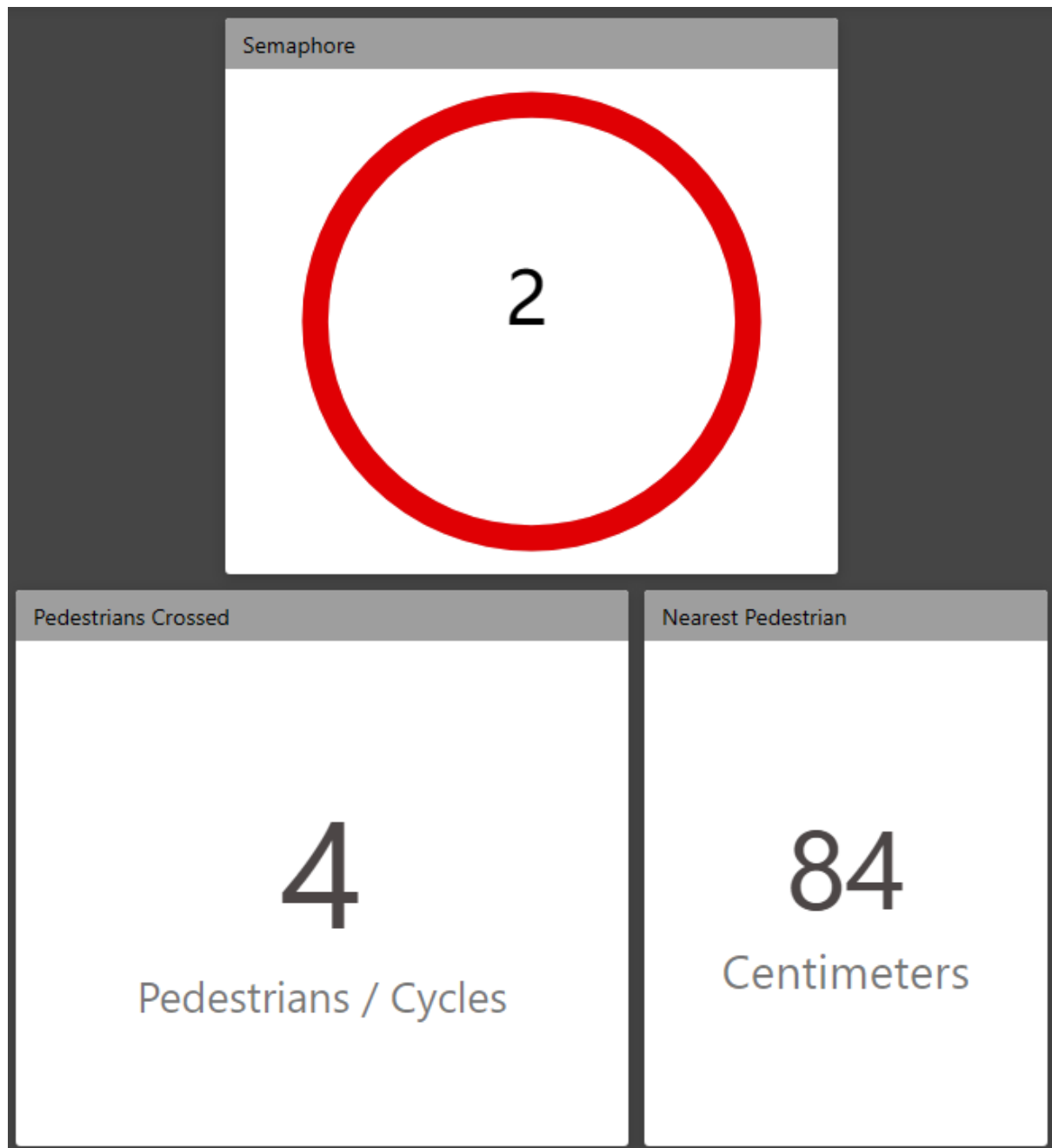
 Password

Após isso, selecione deploy e o sistema estará funcional!

Para checar que o circuito está funcionando, observe o comportamento dos LEDs e cheque se respondem ao sensor ultrassônico.

Para checar que o Node-Red está recebendo e processando as informações, cheque a área de debug.

Por fim, instale o template de dashboard encontrado neste [link](#) e configure-o para visualizar os dados de forma legível.



A aba "Semaphore" indica a posição do semáforo e o quão distante está de permitir a passagem de um pedestre. Ao atingir 2, o semáforo está vermelho e um pedestre pode atravessar, aumentando a contagem "Pedestrians Crossed" em 1.

A aba "Nearest Pedestrian" representa a leitura do sensor ultrassônico e mostra a distância do pedestre mais próximo, em centímetros. Para alterar este valor no simulador, é necessário conectar um controlador de voltagem:

A partir deste ponto, o sistema já está ativo e quaisquer testes podem ser realizados através do dashboard e do simulador.



Seguem imagens do sistema físico para referência:

