

DOCUMENTAÇÃO DO SISTEMA DE IOT DE CONTAGEM DE PASSAGEIROS

Faculdade Impacta — São Paulo, SP, Brasil, 2023

INTEGRANTES:

Rafael Belmonte Izukawa
Noemi Cavalcanti Almeida
Miguel Vinicius Santos Albiach
Victor Neri Kokado
Guilherme Valverde Trindade
Wesley Camilo da Silva Nascimento

Link do Github para acesso ao código: <<https://github.com/RafaIzu/TCC>>.

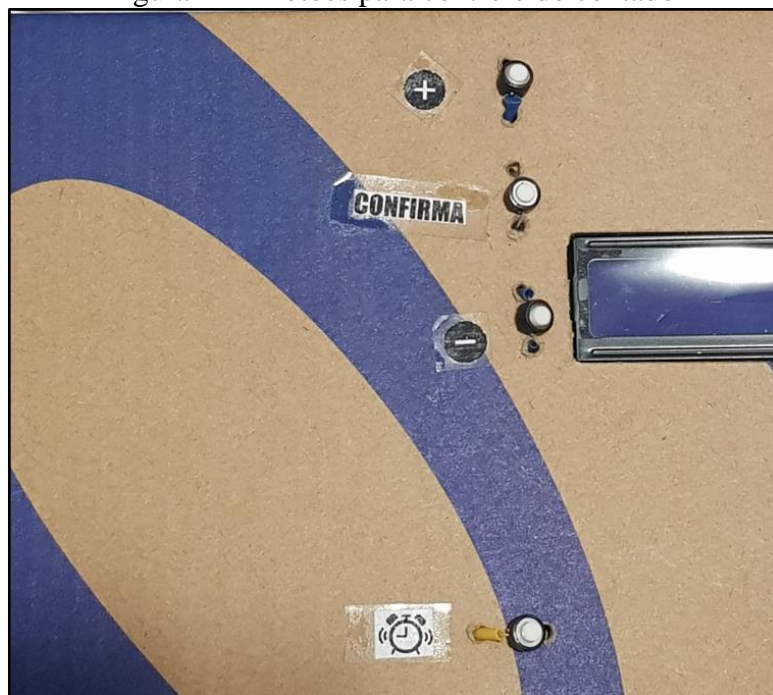
1. INTRODUÇÃO

O seguinte documento consiste na documentação do sistema de IoT para a contagem de passageiros da tese de conclusão do curso de Sistemas de informação da faculdade Impacta®. Neste documento é feito a descrição da operação do sistema, a descrição do código e o diagrama do circuito eletrônico do contador.

2. OPERAÇÃO DO SISTEMA

O sistema é ativado ligando-se o cabo universal serial bus (USB) do contador a um carregador portátil. Logo em seguida, através da *liquid crystal display* (LCD), o sistema pede para informar qual é o limite de pessoas que poderão embarcar no veículo. Para isso, pressiona-se os botões de incremento **+** e decremento **-** e, após definir o número limite de passageiros, deve-se pressionar um terceiro botão de confirmação (**CONFIRMA**) conforme mostrado na figura 1. A figura 2 mostra a mensagem da tela de LCD para informar o número limite de passageiros.

Figura 1 — Botões para controle do contador



Fonte: Os autores

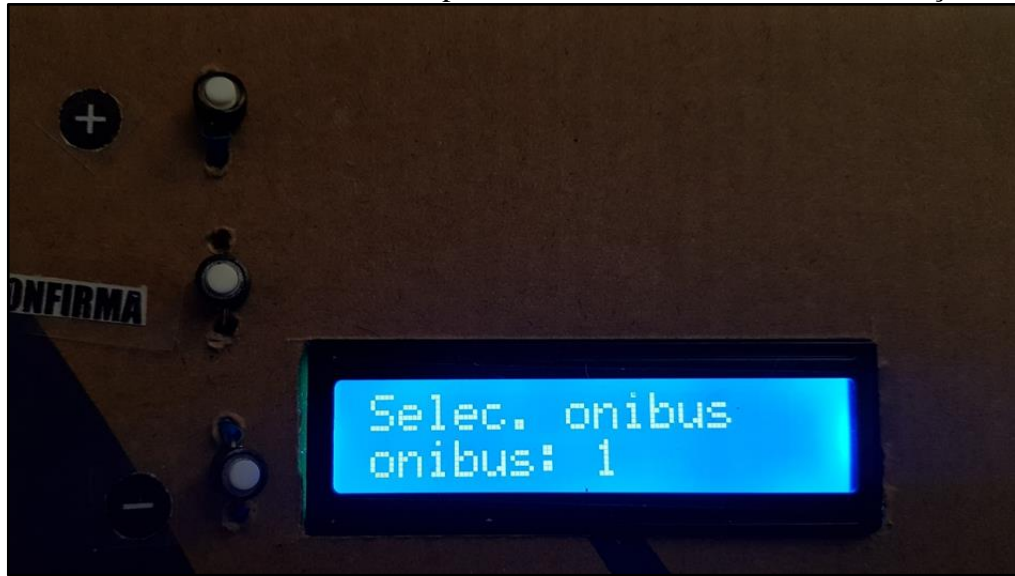
Figura 2 — LCD informando para selecionar o número limite de passageiros



Fonte: Os autores

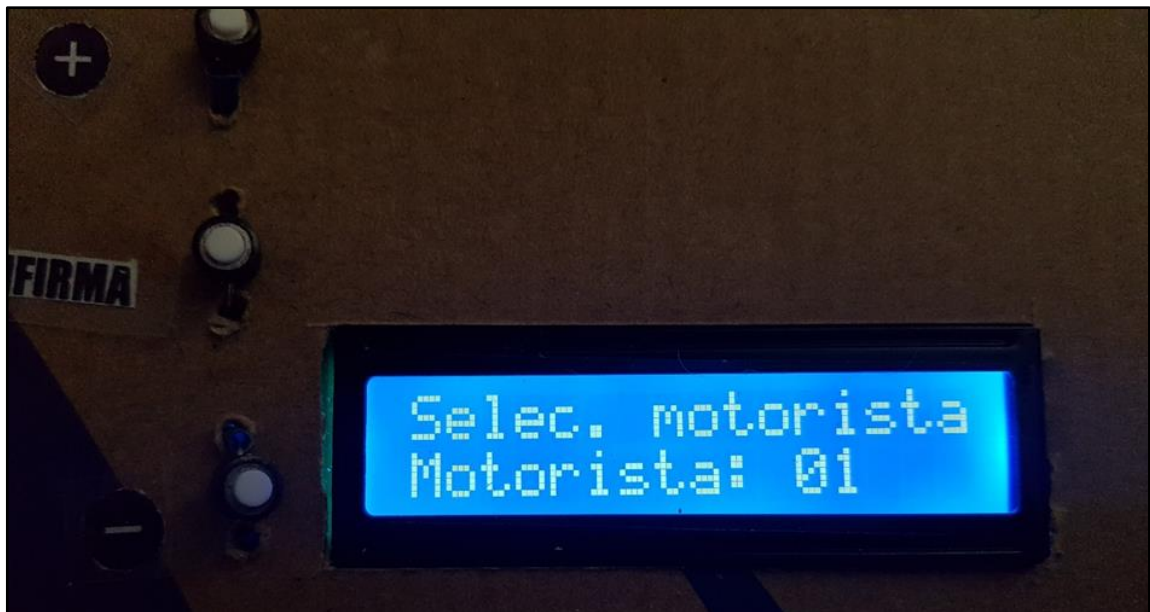
Logo, deverá ser informado o código do veículo no qual o contador está localizado e o código do motorista, para isso, deve-se utilizar os mesmos botões como mostrado nas figuras 3 e 4.

Figura 3 — LCD informando ao usuário para selecionar o número de identificação do ônibus



Fonte: Os autores

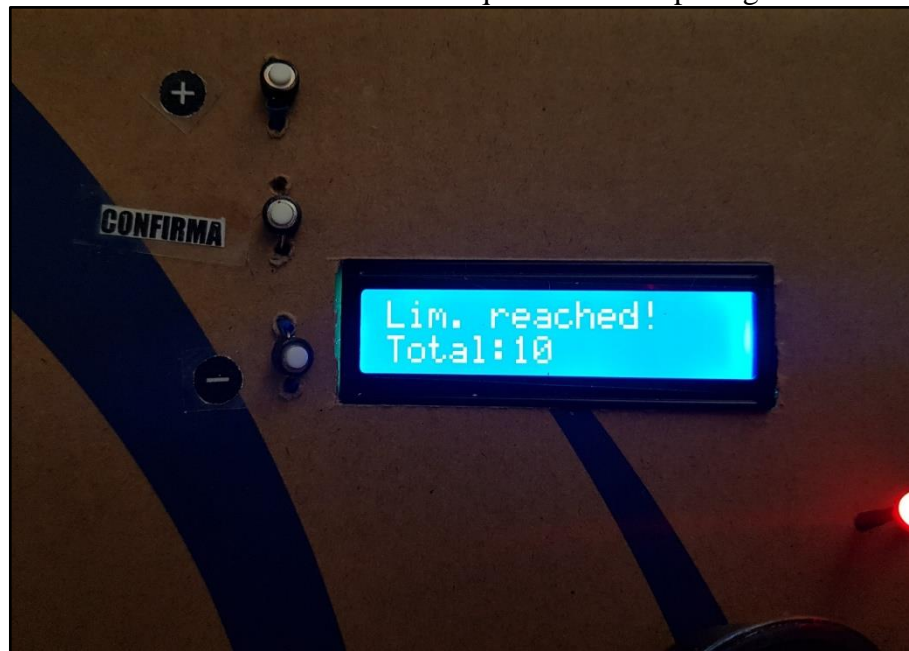
Figura 4 — LCD informando ao usuário para selecionar o número de identificação do motorista



Fonte: Os autores

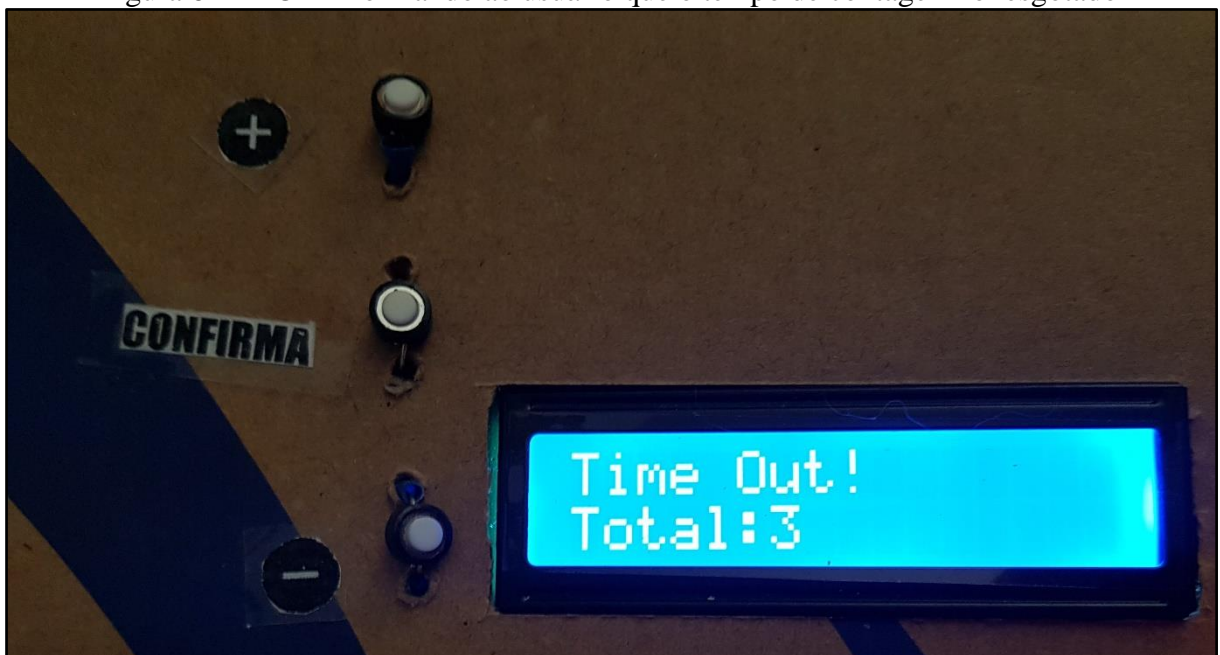
Após fornecido essas informações, será iniciado a contabilização de pessoas que passam pelo sensor, ou seja, será contabilizado o número de pessoas que entram ou saem de um veículo ou sala por um minuto enquanto o LCD mostra o número de passageiros contabilizados. Depois que esse tempo é atingido ou o total de pessoas que podem entrar dentro de uma sala ou veículo atinge o seu limite, um sinal sonoro será emitido pelo *speaker* junto com um sinal luminoso pelo *light emitting diode* (LED). A figura 5 mostra o LCD informando ao usuário que o limite de contagem de passageiros foi atingido, e a figura 6 mostra o LCD informando ao usuário que o tempo de contagem de passageiros foi esgotado.

Figura 5 — LCD informando ao usuário que o limite de passageiros foi atingido




Fonte: Os autores

Figura 6 — LCD informando ao usuário que o tempo de contagem foi esgotado



Fonte: Os autores

Logo em seguida, os dados serão enviados para um *Application Programming Interface* (API), chamado de BusPrime, no qual irá armazená-los no banco de dados e, finalmente, o sistema entra em modo *deep sleep mode* para economia de energia e pode ser “acordado” pressionando o botão de reativação .

3. DESCRIÇÃO DO CÓDIGO DO SISTEMA DE CONTAGEM DE PESSOAS

Quando o sistema é ativado, primeiramente são definidos quais são as principais bibliotecas que serão utilizadas na aplicação. Para esse projeto, foram utilizadas as seguintes bibliotecas:

- Wifi: Biblioteca nativa do ESP32®. Utilizada para testar e conectar o aplicativo em uma rede *Wireless Fidelity* (Wi-Fi).
- HTTPClient: Biblioteca nativa do ESP32®. Utilizada para enviar requisições do tipo *Hypertext Transfer Protocol* (HTTP) por meio de *strings*.
- LiquidCrystal_I2C: Utilizada para controlar a tela de LCD com módulo adaptador *Inter-Integrated Circuit* (I2C). Esta biblioteca é de código livre e pode ser obtida em: <<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>>

Depois, são definidas as principais variáveis que serão usadas globalmente no código pelas rotinas nas quais são: os números dos terminais do ESP32® e quais componentes estarão ligados a elas como os sensores, tela de LCD, *speaker*, botões e LED. Também são passados outros valores como o limite do tempo que o sistema deve realizar a contagem, tempos de *delay* e informações para se conectar à rede Wi-Fi e a *Uniform Resource Locators* (URL) da API.

A primeira função a ser chamada é a *setup*. Essa função é executada somente uma vez logo após que o microcontrolador é ativado ou reiniciado. Essa função é normalmente usada para definir as variáveis e os *pin modes*.

No caso do sistema descrito neste documento, a primeira ação que ocorre nesta função é a chamada da função que faz o ESP32® “acordar” caso o microcontrolador esteja no modo *sleep mode* e depois é definido os terminais do microcontrolador. Após a definição dos terminais do microcontrolador são chamadas as funções que definem as seguintes variáveis: número limite de passageiros que devem ser contabilizados, o código do ônibus e do motorista.

Depois, é passado a função *Loop*. Essa função é executada logo após quando a função *setup* finaliza a sua execução. Quando essa função é executada, ela se repete infinitamente até que o microcontrolador seja desligado ou reiniciado. Normalmente nesta função, é passado a principal tarefa da aplicação. Assim, é nessa função onde ocorre a contagem dos transeuntes e onde ocorre, também, a chamada de outras funções.

A contagem de pessoas é realizada dentro de um *loop while* no qual só é finalizado quando o tempo é esgotado, sendo este definido pela variável constante global “timeOut” junto com a função embutida “millis”, ou se o número limite de pessoas é atingido, sendo este definido pela variável constante global “passagerLimit”. Deve-se observar que a contagem do tempo utiliza a mesma função “millis”.

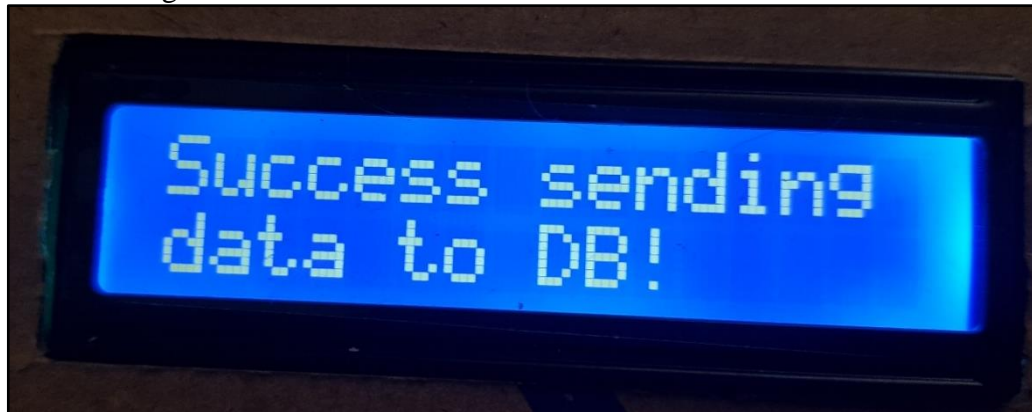
Verifica-se primeiramente se o primeiro sensor infravermelho (“sensor de entrada”) foi ativado ou se o segundo sensor foi ativado (“sensor de saída”). Se o sensor de entrada foi ativado primeiro, soma-se o valor 1 no qual é acumulado na variável “counter”. Depois, inicia-se um breve período de “tolerância” para verificar se o transeunte não recuou que ocorre dentro de um *loop while* aninhado. Se o transeunte recua, o sensor de entrada é ativado novamente e subtrai o valor 1 do total contabilizado. Porém, se o tempo de tolerância é atingido, é considerado que o transeunte ou passageiro passou pelo sensor e não voltou. Essa funcionalidade foi empregada para detectar as “passagens falsas”.

Já se o sensor de saída foi ativado primeiro, ocorre o inverso, o valor 1 é subtraído do total contabilizado e um período de tolerância é iniciado para detectar as “passagens falsas”. Se o transeunte passa pelo mesmo sensor de saída, o valor é somado ao total contabilizado.

Quando o tempo é esgotado ou o limite de pessoas é atingido, o *loop while* é encerrado. A tela de LCD informa que a contagem foi finalizada e informa o motivo do término, ou seja, se foi atingido o limite de pessoas contabilizadas ou se o tempo foi esgotado. Logo, é informado pela tela de LCD qual foi o total de pessoas contabilizadas. Nesse mesmo momento, a função “alert” é chamada. Essa função tem a principal função de ativar o “speaker” e o LED. O “speaker” emite um sinal sonoro dez vezes, assim como o LED, que pisca dez vezes.

Depois que se finaliza a função de contabilização de passageiros, o sistema chama a função “sendToDB” no qual são passados três parâmetros: o número total de pessoas contabilizada, o número de identificação do ônibus e do motorista. Dentro dessa função, primeiro é feita a tentativa de conexão com o roteador Wi-Fi. Se a conexão for bem-sucedida, os dados que foram passados para essa rotina serão utilizados para formar uma *string* concatenada. Essa *string* consiste em uma URL que, por meio do método POST do protocolo HTTP, é enviada para a API localizada em um servidor em nuvem do Azure. Logo em seguida, a tela de LCD informa que o envio dos dados ao servidor foi feito com sucesso.

Figura 7 — Sistema informando fracasso no envio de dados



Fonte: Os autores

Se a conexão com o roteador for malsucedida ou se a requisição for rejeitada pela API, uma mensagem de falha é mostrada pela tela de LCD como mostra a figura 8.

Figura 8 — Sistema informando fracasso no envio de dados



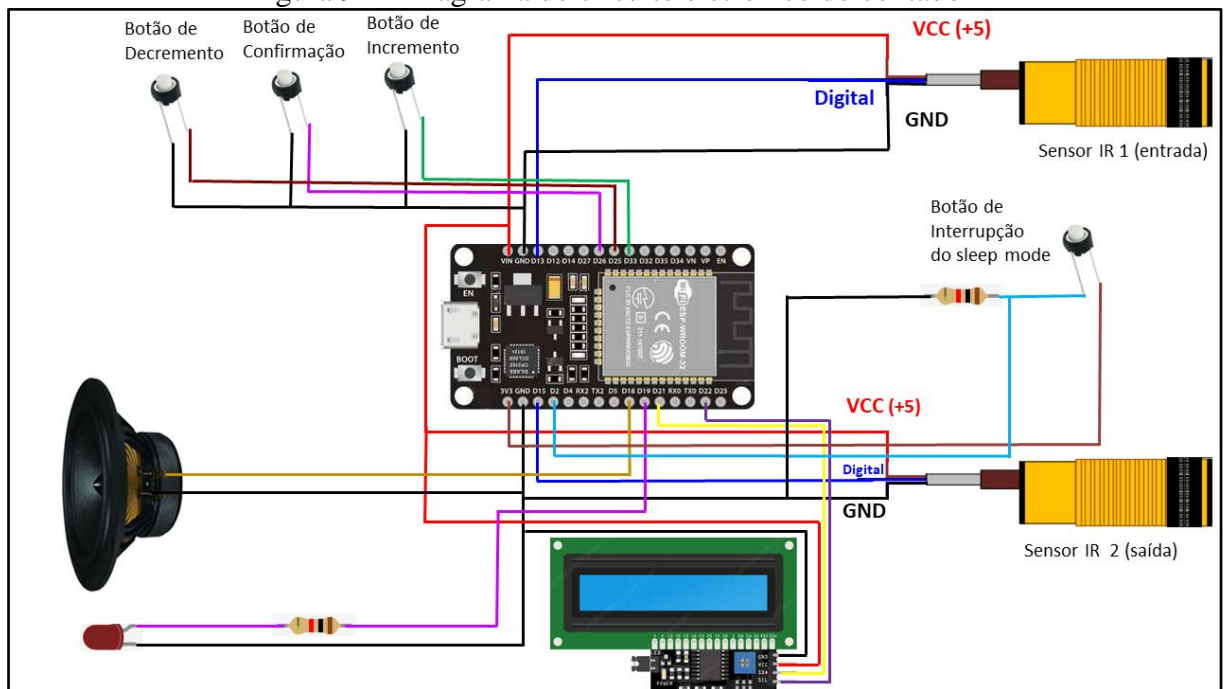
Fonte: Os autores

Após o término da rotina “sendToDB”, a luz da tela de LCD é apagada e o microcontrolador entra no modo *sleep mode*.

4. Diagrama do circuito eletrônico do contador

A figura 9 ilustra o diagrama do circuito eletrônico do contador. Aqui são apresentados a relação dos sensores e atuadores com o microcontrolador.

Figura 9 — Diagrama do circuito eletrônico do contador



Fonte: Os autores