

SISTEMAS OPERACIONAIS EMBARCADOS - ESTACIONAMENTO INTELIGENTE

Guilherme Richard Alves de Oliveira, Hallana Rayssa Alves da Silva

Graduação em Engenharia Eletrônica, Faculdade do Gama

Universidade de Brasília

Gama, DF, Brasil

email: guilhermerychard04@gmail.com, hallanarayssa@gmail.com

1. JUSTIFICATIVA

Existem diversos centros comerciais, condomínios e faculdades que possuem um estacionamento para seus usuários. Dados do IBGE (2018) mostram que a frota de veículos no Distrito Federal ultrapassam 1,8 milhões.

Segundo dados do DETRAN (2018) a frota dos veículos emplacados aumentou 3,3% em 2018 em relação a 2017. Destes, 55,8% já estão licenciados. Por isso, é necessário que haja um sistema de segurança, controle associados a um baixo custo e potência e que possa ser implementado através de um sistema embarcado.

Neste projeto será implementado um sistema para estacionamentos privativo que foque na segurança do usuário, através de um cadastramento prévio da placa do veículo. O sistema a ser implementado também fará o controle de lotação do estacionamento, registrando entrada, saída e tempo de permanência dos usuários.

2. OBJETIVOS

2.1. Objetivo Geral

Implementar um sistema de automação para estacionamentos rotativos que efetue a leitura e identificação do veículo dos usuários, efetuando o controle da entrada, saída e permanência do veículo.

2.2. Objetivos Específicos

De forma a complementar o objetivo geral, os objetivos específicos deste trabalho são:

- Efetuar o cadastramento do usuário, através de uma interface mobile;
- Efetuar uma comunicação com o Banco de Dados do projeto;

- Fazer a leitura e o reconhecimento da placa do carro do usuário;
- Liberar o acesso caso haja confirmação de cadastro;
- Armazenar no Banco de Dados informações referentes à entrada, saída e permanência do veículo;
- Informar a quantidade de vagas disponíveis;

3. REQUISITOS

Para implementar o sistema de automação serão necessários os seguintes requisitos:

- Cadastramento dos usuários.
- Aquisição de imagem da placa do veículo.
- Controle de Entrada.
- Controle de Saída.
- Disponibilidade de vagas.

3.1. Cadastramento dos usuários

- O usuário deverá efetuar o cadastramento prévio para utilizar o sistema a ser descrito no escopo desse projeto;
- Para o cadastramento, será necessário a inserção dos seguintes dados: CPF do usuário, placa e modelo do veículo.

3.2. Controle de Entrada

A entrada dar-se-á da seguinte forma:

- Haverá uma câmera para reconhecimento da placa do veículo na entrada do estacionamento. A câmera permanecerá em Low-Power Mode (LPM), até que um sensor de movimento acione o sistema.

- Realizado o reconhecimento do veículo, a cancela será liberada para a entrada do carro.
- O usuário será notificado, via aplicativo, o horário entrada no estacionamento.
- Após a passagem do veículo a cancela será fechada e haverá um decremento do número de vagas do estacionamento.

3.3. Controle de Saída

- A câmera é acionada pela movimentação do veículo e fará o reconhecimento da placa, abrindo a cancela para saída do carro.
- O usuário será informado do horário de saída do estabelecimento, via aplicativo. Mostrando também o tempo de permanência do mesmo no local.
- O sistema incrementa a quantidade de vagas disponíveis.

3.4. Disponibilidade de vagas

- Através do controle de saída e entrada dos veículos, o sistema disponibilizará a quantidade de vagas disponíveis no estacionamento.

4. BENEFÍCIOS

- Controle de entrada e saída de usuários de estacionamentos, condicionado cadastro e identificação prévia, tornando o ambiente mais seguro.
- Possível monitoramento do tempo de permanência dos usuários.
- Maior confiabilidade da identificação do usuários se comparado aos sistemas que utilizam cartões.
- Prevenção de furtos de veículo, uma vez que o usuário recebe a notificação em seu aparelho, referentes ao acesso do mesmo ao local.

5. MATERIAIS

Para a implementação do projeto serão utilizados os seguintes materiais:

- 01 - Raspberry Pi Model A+.
- 01 - Câmera VGA OV7670 CMOS
- 01 - Servo Motor.
- 01 - Módulo Sensor de Presença PIR HC-SR501.
- 01 - Display de LCD.
- 01 - Cancela.

5.1. Raspberry Pi Model A+

Para desenvolvimento do projeto, será utilizada a Raspberry Pi. A Raspberry Pi é um pequeno computador de bordo do tamanho de um cartão de crédito, desenvolvido no Reino Unido pela Raspberry Pi Foundation, com a intenção de promover o ensino básico da ciência da computação nas escolas.

Embora tenha capacidade para expansão de sua RAM (Random Access Memory), e não tenha dispositivos como CD, DVD e HD, possui tudo o que é necessário para tornar-se um computador pessoal.[1]



Fig. 1. Raspberry Model A+.

5.2. Câmera

O módulo câmera VGA utiliza o sensor CMOS OV7670. Existem duas versões dessa câmera no mercado. A primeira utiliza apenas o sensor OV7670 e apresenta os pinos para conexão entre o sensor e o outro dispositivo. A segunda versão, por sua vez, possui um buffer (FIFO - First In First Out) e armazena dados do sensor CMOS. Para este projeto, será utilizada a primeira opção, uma vez que ela apresenta entre suas características:

- Resolução máxima de 640 x 480;
- Taxa de transferência máxima de 30 fps (frames por segundo);
- Tensão de operação entre 2,5 à 3,0 V, entre outras.

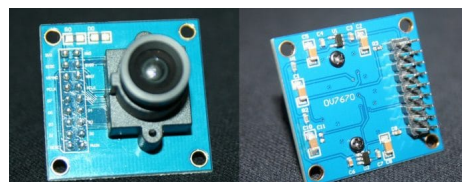


Fig. 2. Módulo Câmera OV7670

5.3. Servo Motor

Os servomotores, também conhecidos como servos, são componentes elaborados com o uso de engrenagens, capazes de gerar movimento em sentido horário e anti horário. É um motor em que podemos controlar a sua posição angular com o uso de um sinal PWM, do inglês Pulse Width Modulation.

Características do servo modelo: Servo 9g Tower Pro SG90:

- Tensão de Operação: 3 a 7,2V.
- Faixa de Rotação: 180°
- Modulação: PWM.
- Velocidade (4.8V): 0.12 s / 60°
- Torque(4.8V): 1,5kg/cm.
- Dimensões: 22x12x19mm.
- Peso: 9g;
- Tamanho do cabo: 24cm.



Fig. 3. Servo Motor.

5.4. Módulo Sensor de Presença PIR HC-SR501

Os sensores de presença atuam gerando nível alto em uma de suas saídas no momento em que detectam o movimento de algum corpo que emita radiação infravermelha. Neste projeto, será utilizado o sensor de presença PIR HC-SR501, um módulo que usa um sensor PIR (piroelétrico), capaz de detectar movimento de um corpo numa área de detecção de até 7 metros.

5.5. Display de LCD

O display será utilizado para informar o número de vagas disponíveis.



Fig. 4. Sensor de distância Ultrassônico.

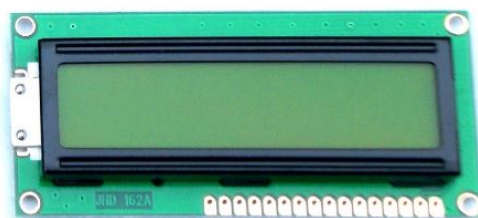


Fig. 5. Display LCD.

6. DESENVOLVIMENTO

6.1. Desenvolvimento do Ponto de Controle 2 - Protótipo funcional utilizando ferramentas mais básicas da placa de desenvolvimento e bibliotecas.

6.1.1. Diagrama de Blocos do Sistema

Para compreender o sistema foi elaborado um diagrama de blocos e os processos que ocorrerão que pode ser visualizado na Figura 6.

Considerando que o cadastro do usuário ocorreu previamente o sistema se inicia após a detecção veicular e consequente acionamento da câmera. A partir daí a câmera fará a aquisição de imagem da placa do carro e o sistema realizará o reconhecimento dos caracteres da placa do veículo.

O dado captado será verificado no banco de dados e o caso o usuário esteja cadastrado ele será notificado por aplicativo e terá o acesso ao estacionamento liberado. Caso não haja cadastro o acesso será negado.

6.1.2. Aquisição da Imagem

A aquisição da imagem ocorrerá através da câmera posicionada na entrada/saída do estacionamento capturando a placa do veículo. Para a implementação dessa parte do projeto será utilizado os comandos da biblioteca fswebcam.

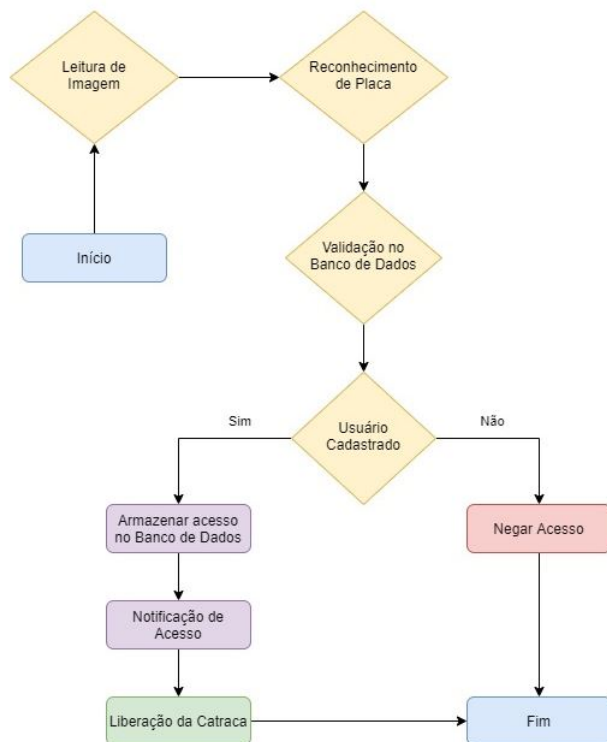


Fig. 6. Diagrama de Blocos do Sistema.

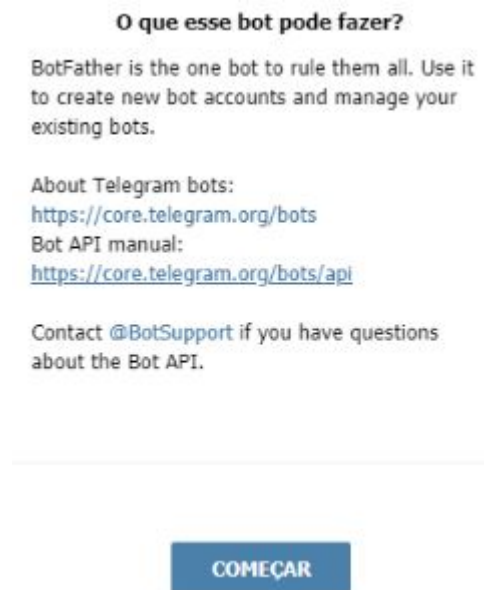


Fig. 7. Criação de um novo bot.

A biblioteca fswebcam é capaz de realizar a autoconfiguração relativa à qualidade e dimensão de qualquer imagem obtida através da entrada USB da Raspberry.

6.1.3. Comunicação entre a Raspberry Pi e o Telegram Bot

- Telegram Bot

O Telegram Bot é um aplicativo executado dentro do Telegram. Através dele é possível interagir com os bots enviando-lhes mensagens comandos e solicitações on-line. Com o aplicativo é possível acionar relés, LEDs e monitorar ambientes em qualquer lugar e a qualquer hora.

No projeto utilizaremos o Telegram para o cadastramento dos usuários. Para testar a comunicação da Raspberry com o Telegram Bot e verificar se os comandos enviados estão acionando os pinos GPIO da Raspberry, serão realizados os seguintes passos nesta etapa do projeto:

- Configuração do Telegram Bot.

Nesta etapa seleciona-se o BotFather no Telegram, cria-se um novo Bot designando nome e username e em seguida o aplicativo gera um token que permite o acesso HTTP API para controle do Bot.



Fig. 8. Token gerado após criação do bot.

- Configuração da Raspberry Pi para rede e SSH.

Para execução do software Telegram Bot na Raspberry Pi é necessário a configuração para rede e SSH.

- Implementação e execução do Código para comunicação entre Raspberry Pi e o Telegram.
- Acionamento do LED conectado a Raspberry através de comandos inseridos no Telegram Bot.

6.1.4. Banco de Dados

O banco de dados é de fundamental importância neste projeto, uma vez que o sistema a ser desenvolvido liberará a passagem do veículo, caso identifique o cadastro do veículo em seu sistema. Desta forma, foi escolhido o MySQL como banco de dados, uma vez que há na internet um fácil acesso ao material e tutoriais.

A Língua de Consulta Estruturada, do inglês Structured Query Language, ou chamada pela abreviação SQL é conhecida como uma linguagem de consulta padrão, utilizada para manipular bases de dados relacionais. É utilizada em inúmeros sistemas, como por exemplo MySQL, SQL Server, Oracle, Sybase, etc. [2]

Ainda segundo dados do Dev Media, o SQL possui muitos outros recursos além da consulta ao banco de dados. Esses recursos estão divididos em cinco partes:

- Data Definition Language (Linguagem de Definição de Dados), conhecida como DDL;
- Data Manipulation Language (Linguagem de Manipulação de Dados), comumente chamada de DML;
- Data Query Language (Linguagem de Consulta de Dados), ou DQL;
- Data Control Language (Linguagem de Controle de Dados), ou DCL;
- Data Transaction Language (Linguagem de Transação de Dados), conhecida como DTL.

7. REVISÃO BIBLIOGRÁFICA

Durante o processo de levantamento bibliográfico, encontrou-se diversos trabalhos relacionados a estacionamentos inteligentes. Dentre eles, destacam-se alguns trabalhos teóricos e produtos já vendidos para o mercado.

Exemplos de grandes clientes dos sistemas de estacionamentos inteligentes são shoppings e supermercados: ambos

com a finalidade de reduzir o tempo gasto pelo cliente na procura de vagas, reduzir o congestionamento e consequentemente, melhorar a experiência do mesmo em seu estabelecimento.

Em alguns desses modelos é possível encontrar, ainda na entrada do estabelecimento, painéis que indicam a quantidade de vagas disponíveis em cada andar, bem como sistemas de LEDs, posicionados sobre a vaga e que indicam o status dela.



Fig. 9. Smart Parking - House Solutions (2017)

Em maio de 2019 a companhia Novas Software lançou no mercado o aplicativo Smart Parking. O aplicativo encontra-se disponível nas principais plataformas de e-commerce voltadas para a compra de aplicativos. Com o objetivo de encontrar no mapa estacionamentos cadastrados, o usuário pode acessar o horário de funcionamento, localização e quantidade de vagas disponíveis. É possível, também, realizar reservas de vagas e fazer check-in e check-out da vaga, com pagamento pelo uso do estacionamento. [3]



Fig. 10. Aplicativo Smart Parking, Novas Software (2019)

Um dos trabalhos teóricos estudados, utiliza sensores instalados no estacionamento e que enviam dados a serem armazenados no banco de dados MySQL. A interface implementada nesse projeto consistia em uma página web onde era permitido ao usuário efetuar a reserva de vagas e o pagamento de acordo com a utilização do espaço. O projeto permitia também visualizar no site a quantidade de vagas livres, reservadas e ocupadas. [4]

8. REFERENCIAS

- [1] Raspberry Pi. (2020) About us. Raspberry Pi. Access date: 17 set. 2020. [Online]. Available: <https://www.raspberrypi.org/about/>
- [2] Bruno Nascimento. (2015) Mysql tutorial. Dev Media. Access date: 20 out. 2020. [Online]. Available: <http://www.devmedia.com.br/mysql-tutorial/33309>
- [3] Novas Software. Smart parking. Novas Software. Access date: 21 out. 2020. [Online]. Available: <https://play.google.com/store/apps/details?id=com.novas.smartparking>
- [4] Thiago P.R. Vicente. (2020) Controle inteligente de vagas para estacionamento utilizando o conceito de internet das coisas. USP São Carlos. Access date: 20 out. 2020. [Online]. Available: <http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-31012017-162548/?lang=br>