



PROPOSTA DE PROGRAMAÇÃO EM PINHEIRINHO USANDO PROTOCOLO ESP-NOW

DOI: 10.37702/2175-957X.COBENGE.2023.4448

Rubens Chiullo Silva - rubenschiullo@gmail.com
IFRO

Mariane de Barros Andrade - marianebarrosandrade@gmail.com
IFRO

Fabício Lopes Coelho Júnior - fabriciojr_net@hotmail.com
IFRO

Larissa Samara Paula de França - larissa.oks@gmail.com
Fundação Universidade Federal de Rondônia

Artur Vitório Andrade Santos - artur.santos@ifro.edu.br
Instituto Federal de Rondônia

José Diogo Forte de Oliveira Luna - jose.luna@ifro.edu.br
Instituto Federal de Rondonia

Resumo: *The present work proposed a development for programming a Full Tree, a light-signal panel for racing light-powered racing cars. The proposal involved concepts such as wireless communication through the protocol ESP-Now and hotspot development for smartphone connection for viewing results. In addition, a programming was carried out in order specific for activating light signals do Pinheirinho, timing reaction times, arrival and total journey of each trolley. The project proved to be effective with stability in communication without wire held and display of race results via remote via smartphone*

Palavras-chave: ESP-NOW, wireless, Protocols, HTML

PROPOSTA DE PROGRAMAÇÃO EM PINHEIRINHO USANDO PROTOCOLO ESP-NOW

1 INTRODUÇÃO

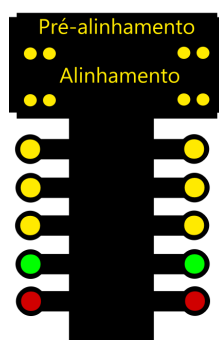
Com a chegada da quarta revolução industrial (ou Indústria 4.0), conquistou-se a presença disruptiva da transmissão e tratamento de dados industriais em diversas áreas como a Tecnologia de Informação e Comunicação (TIC), *cloud computing* (CC) e Sistemas Ciberfísicos (CPS). Essa ação juntamente com as das outras revoluções industriais elevaram o trabalho humano na indústria para deixar as atividades no chão de fábrica executadas por máquinas e focar nas atividades de alto nível como o gerenciamento e supervisão de processos (LUGLI; ALEXANDRE, 2019, p. 17).

Dentre as principais tecnologias proporcionadas pela Indústria 4.0, a troca de dados pré-processados e disponibilizados pela Internet entre dispositivos industriais interconectados configura o ramo mais amplo da Internet das Coisas (IoT) (CORTEZ et al., [s.d.]). Os dispositivos IoT se incluem em uma área da tecnologia dotada de diversos protocolos tais como o Bluetooth, MQTT, ESP-NOW e outros.

Dentre os protocolos de comunicação IoT, existem aqueles que tem foco na aplicação para ambientes automatizados, como o protocolo ESP-NOW. ESP-NOW é um protocolo de comunicação sem fio desenvolvido pela *Espressif* e é aplicado em diversos casos como iluminação inteligente, controle remoto, sensores, atuadores e outros dispositivos IoT (BRITO et al., 2019).

O conceito IoT pôde ser aplicado em uma experiência com uma competição de carrinhos de brinquedo ativados pela luz. A ação foi executada nas dependências do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), Campus Calama, na cidade de Porto Velho / RO, no primeiro semestre de 2022, por um grupo de acadêmicos do curso de Engenharia de Controle e Automação do mesmo Instituto. A competição consistia na disputa de tempo entre dois carrinhos em duas pistas independentes e exclusivas. O contexto de corrida se mostrou oportuno para a criação de um painel de sinais luminosos em forma de Pinheirinho (Figura 1). O painel Pinheirinho é formado por luzes que, mediante microcontroladores e sensores posicionados na pista para identificar a posição dos carrinhos, indicam numa corrida o alinhamento simétrico dos carrinhos no início da pista, no ato da largada, identificação de carrinho na chegada e possível antecipação de largada ("queimar" a largada). (FUELTECH, 2020)

Figura 1 – Painel Pinheirinho.



Fonte: Autoria própria.

Esse projeto teve motivação por criar a programação implantada no microcontrolador do painel Pinheirinho usando protocolo ESP-NOW para comunicar a rede entre os microcontroladores do Pinheirinho criando um ponto de acesso wireless para visualizar informações da corrida via smartphone.

2 REFERENCIAL TEÓRICO

Para basear os aspectos teóricos deste trabalho também foram considerados as publicações de Secco (2019), Souza (2016) e Beirão et al. (2021), que demonstraram ser possível estabelecer diversos tipos de conexão sem fio estáveis através do microcontrolador ESP8266, inclusive com protocolo ESP-NOW.

O protocolo ESP-NOW é um protocolo de comunicação direta e sem fio desenvolvido pela *Espressif* e opera na função ponto-a-ponto no padrão IEEE 802.11 e faixa de frequência 2,4 GHz (CORTEZ et al., [s.d.]). O protocolo ESP-NOW encapsula e envia os dados por uma mensagem do transmissor ao dispositivo receptor. Esse é comumente usado para controles de luzes inteligentes, controle remoto, sensores e outros ("*Espressif Systems*", [s.d.]). Uma particularidade desse protocolo é que necessita que os dispositivos estejam emparelhados antes da comunicação por um endereço do protocolo CBC-MAC (Cipher Block Chaining Message Authentication Code Protocol). ("*ESP-IDF Programming Guide - ESP32*", [s.d.])

Nesse caso, o protocolo foi usado para transferir os dados de leitura dos sensores LDR de um ESP32 no final da corrida para um ESP32 no início da corrida devido à longa distância do trajeto. Isso facilitou a implantação da estrutura física de todo o trabalho e evitou o uso de cabos de dados extensos.

Para implantar o protocolo ESP-NOW, foram utilizados microcontroladores ESP32, selecionados no lugar de microcontroladores ESP8266 mencionados na literatura principalmente por ser uma versão otimizada das placas de baixo custo da Espressif e por estar em disponibilidade de acesso na execução desse trabalho.

Os microcontroladores ESP32 possuem 32 pinos de entrada GPIO (General Purpose Input/Output) em relação ao ESP8266, que possui apenas 17 pinos (FAGUNDES, [s.d.]). Essa diferença de pinos foi importante devido à grande quantidade de componentes eletroeletrônicos que seriam controlados, dentre os quais estão sensores LDR (Resistores dependentes de luz) e LEDs. Além disso, a maior variedade de opções de rede do ESP32 como o *Bluetooth* e *Wireless* permite esse microcontrolador ser testado com vários tipos de protocolo de comunicação, além do ESP-NOW.

Visto que não há literatura específica para o painel Pinheirinho e seu funcionamento, esse trabalho orientou-se pelas regras da competição realizada no IFRO em comparação a uma corrida de arrancada realizada em contexto real. Segundo Sergers (2014), a cronometragem é uma ferramenta importante para as corridas, pois ela mensura o desempenho em geral do veículo, do motorista e dos pneus. Além disso, os resultados obtidos podem ser usados para inspecionar o desempenho do veículo como método para investigar melhorias aplicáveis.

O projeto de Beirão et al. (2021) prova que é possível aplicar a linguagem HTML na programação de um site para que o ESP8266 possa ser um ponto de acesso de conexão com smartphone, podendo transmitir e exibir informações diretamente no dispositivo *mobile*.

O software Arduino IDE foi utilizado para criar os códigos para os dois ESP32 utilizados. Segundo Gomes (2020), um código produzido no Arduino IDE através de sketches "é traduzido para a linguagem C e transmitido para o compilador avr-gcc, que traduz os comandos para uma linguagem que pode ser compreendida pelo microcontrolador da placa".

3 METODOLOGIA

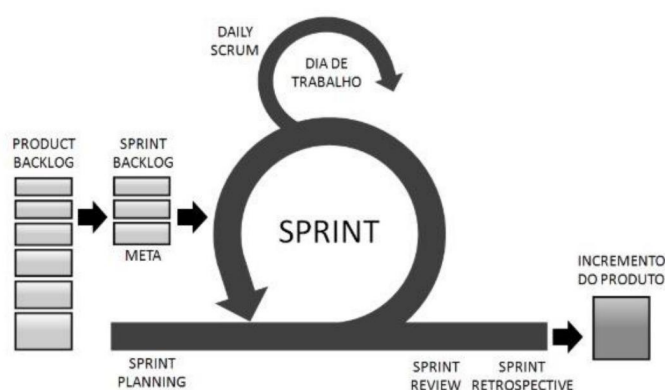
O método abordado nesse projeto tem como base a lógica da programação implantada no microcontrolador do Pinheirinho que nesse caso será o ESP32.

3.1 Metodologia Scrum para implementação de programação

A fim de orientar a metodologia de execução do trabalho, utilizou-se o *framework Scrum*. *Scrum* indicado para projetos que não se sabe certamente como serão executados, no entanto, esse conceito de trabalho focaliza os objetivos finais do processo (GOMES, 2020). O *Scrum* é composto por três partes principais: transparência, onde se deve deixar claro os objetivos de cada etapa; inspeção, parte de verificação dos artefatos do *Scrum* a fim de que se consiga correções e melhorias rápidas; Adaptação, parte em que as inspeções realizadas sejam devidamente aplicadas e executadas para atingir os resultados esperados. (SILVA; LOVATO, 2016)

A Figura 2 mostra uma forma resumida do ciclo do *Scrum*. O *framework Scrum* funciona em um ciclo que aqui pode ser definido em: (1) Fazer um *Product backlog*, ou seja, elencar quais são os objetivos que o produto final deve atender; (2) Transferir objetivos do *Product backlog* para o *sprint backlog*, esse que é o processo de iniciação do; (3) *sprint*, que por si é o tempo de execução para atingir o mínimo de objetivos do *product backlog* e; no fim, (4) obter o mínimo do produto viável para uso ou retornar para o *sprint backlog* a fim de que se possa atingir mais objetivos do produto final.

Figura 2 - Ciclo do Scrum.



Fonte: (GOMES, 2020)

Nessa corrida, dois carrinhos disputam para saber quem chega mais rápido da largada ao ponto de chegada. Todo o processamento da corrida deve ser exibido através dos sinais luminosos do Pinheirinho. Antes de se iniciar a largada, os carrinhos devem passar por um processo de alinhamento para evitar que iniciem em distâncias diferentes.

Product Backlog e Sprints: Para realização desse trabalho, foram necessários cinco ciclos de *sprints*, definidos pela tabela 1. O conjunto de *sprints backlog* forma o *product backlog*.

Tabela 1 – Tabela de ciclos de sprint.

Ciclo de Sprint	sprint backlog
1	Projeto eletroeletrônico do sistema
2	Projeto mecânico do sistema
3	Criar programação no ESP32 no início da corrida para que seja capaz de realizar o acionamento dos sinais luminosos do Pinheirinho e monitorar os sensores de alinhamento e largada.
4	Incrementar protocolo ESP-NOW na programação a fim de que o ESP32 posicionado ao início da corrida consiga receber os dados dos sensores no fim da corrida
5	Incrementar programação para que o ESP32 no início da corrida crie um ponto de acesso para smartphone a fim de visualizar os resultados em um site estruturado na linguagem HTML

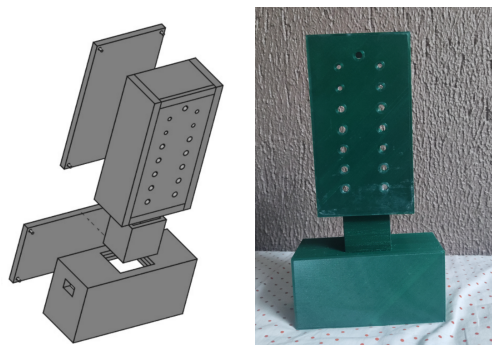
Fonte: Autoria própria.

Para atingir os objetivos dos sprints, o projeto contou com a colaboração da equipe que estava desenvolvendo o Pinheirinho. Esse artigo tem foco nas áreas da programação dos sprints, ou seja, nos ciclos 3, 4 e 5 da Tabela 1. Os ciclos de construção mecânica e eletroeletrônica foram realizados por membros da equipe de desenvolvimento.

3.2 Projeto Mecânico

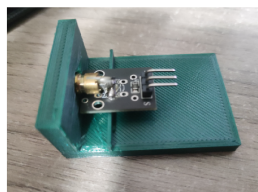
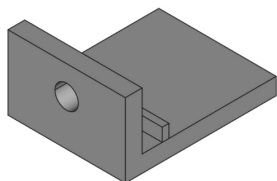
A estrutura mecânica da corrida foi impressa utilizando material ABS de impressão 3D modelado no software *open source Freecad* e teve dimensões adaptadas para que acomodasse os aparelhos de sensoramento, além de desenvolver de uma estrutura maior para que se fosse possível conectar os LEDs que funcionariam como os sinais luminosos do Pinheirinho. As estruturas podem ser observadas nas figuras 3 e 4 abaixo:

Figura 3 – Estrutura do Pinheirinho impressa em 3D.



Fonte: Autor em conjunto com a equipe de desenvolvimento do Pinheirinho

Figura 4 – Estrutura de apoio em modelo 3D (à esquerda) e estrutura de apoio impressa (à direita) do emissor de laser infravermelho, usado no sensoramento.



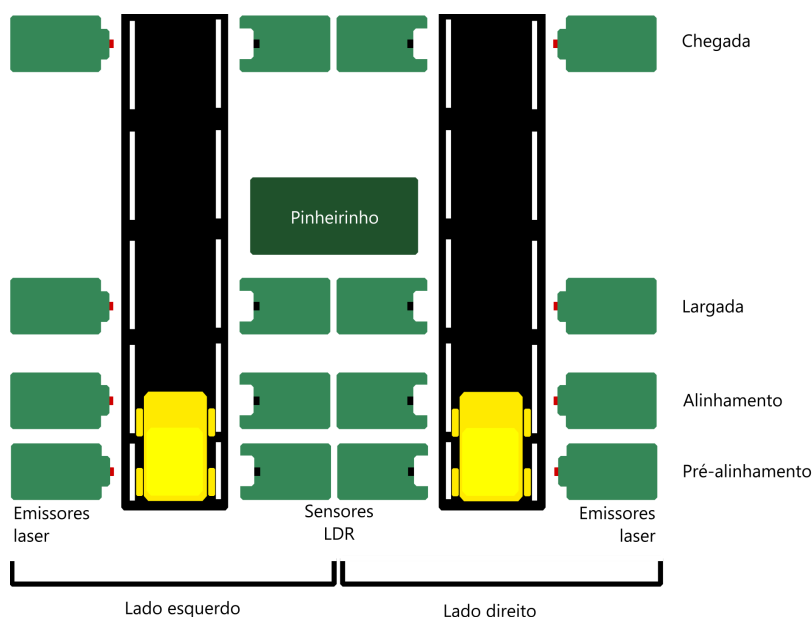
Fonte: Autor em conjunto com a equipe de desenvolvimento do Pinheirinho

3.3 Projeto Eletroeletrônico

O circuito eletroeletrônico do projeto possui tanto os microcontroladores ESP32, quanto emissores de luz laser e sensores LDR para sensoriamento, baterias para alimentação dos circuitos elétricos e resistores para controlar a corrente elétrica.

O monitoramento para saber a localização dos carrinhos por meio de sensores de resistores dependentes de luz (LDR) ativados por emissores de laser é mostrado conforme a figura 5. Além de sensores para a largada e chegada, também foram necessários sensores para monitorar o alinhamento dos carrinhos antes de começar a contagem do Pinheirinho.

Figura 5 – Esquemático da estrutura da corrida.



Fonte: Autoria própria

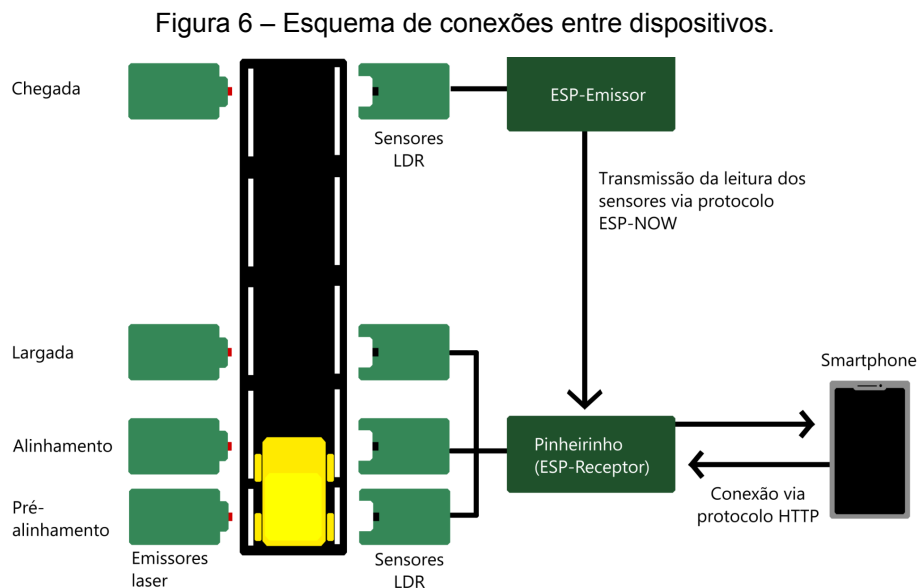
3.4 Projeto Computacional

3.4.1 Esquema de conexão dos dispositivos

Para controlar o acionamento dos sinais luminosos do Pinheirinho, foi utilizado o microcontrolador ESP32 devido ao seu baixo custo de obtenção e disponibilidade de acesso ao autor durante a execução do projeto.

Em razão da distância alta entre a largada e a chegada, foi necessário o uso de dois microcontroladores, sendo o primeiro chamado de ESP-Receptor e selecionado para os acionamentos das luzes do Pinheirinho no início da corrida próximo tanto dos sensores

de pré-alinhamento quanto dos sensores de alinhamento e largada. Já o segundo microcontrolador foi posicionado no final da corrida próximo aos sensores de chegada, chamado de ESP-Emissor, e seria responsável pelo envio das informações dos sensores de chegada ao ESP-Receptor utilizando o protocolo ESP-NOW conectado a uma rede *wireless*. O esquema pode ser observado na figura 6.



Fonte: Autoria própria

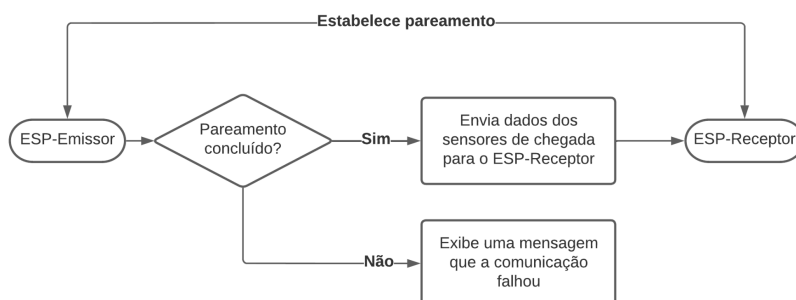
Além dessas funções, o ESP-Receptor também é responsável por gerar um ponto de acesso *wireless* a fim de que se possa conectar um smartphone para exibir um site que mostre os resultados da corrida, podendo atualizar os resultados e reiniciar a corrida de forma remota.

3.4.2 Descrição do algoritmo de programação

Para a realização desse trabalho foram necessários o desenvolvimento de dois arquivos de códigos produzidos no software Arduino IDE com uso das bibliotecas WiFi e ESP-NOW. A biblioteca WiFi é nativa do Arduino IDE e permite usar várias funções para conexões entre vários dispositivos wireless. Já a biblioteca ESP-NOW é responsável pelo uso das funções utilizadas no protocolo ESP-NOW.

A fim de realizar a conexão do protocolo ESP-NOW entre os ESP32, foi usado como base as práticas de *Dronebot Workshop* (2022) na sua publicação "*ESP NOW - Peer to Peer ESP32 Communications*". Nessa publicação, assim como exigido pelo ESP-NOW, é dito necessário obter um endereço MAC (Media Access Control) dos dispositivos que vão se comunicar. O endereço MAC é um código hexadecimal formado por seis dígitos e único para qualquer dispositivo de conexão de rede (DRONEBOT WORKSHOP, 2022). Nesse caso, o endereço obtido foi: 08, 3A, F2, AC, 87, B4. Tendo obtido o endereço MAC, basta inseri-lo na programação para executar o pareamento (Figura 7).

Figura 7 – Fluxograma do pareamento entre dispositivos no protocolo ESP-NOW.

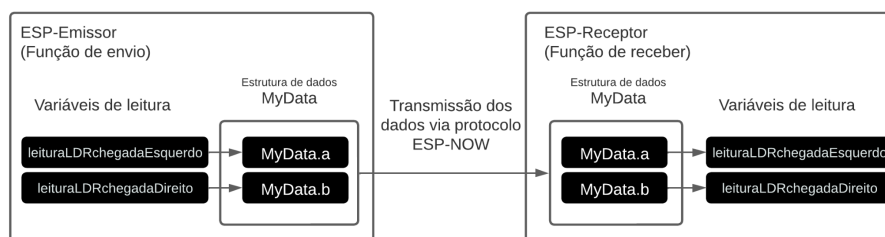


Fonte: Autoria própria

Quando os dispositivos são acionados, é realizada uma tentativa de comunicação. Se a tentativa falha, é exibida uma mensagem no Monitor Serial do Arduino IDE que a comunicação falhou. Se a tentativa teve êxito, os dados registrados na leitura dos sensores de chegada pelo ESP-Emissor são enviados para o ESP-Receptor. Após ajustar o pareamento entre os dispositivos, também é necessário criar uma estrutura de dados que seja igual tanto para o ESP-Emissor quanto para o ESP-Receptor. A estrutura de dados pode ser entendida como a cápsula que armazena os dados que vão ser transmitidos. Os dados que serão enviados são as variáveis de leitura dos sensores LDR na chegada da pista: `leituraLDRchegadaEsquerdo`, e `leituraLDRchegadaDireito`.

Um modelo esquemático da transferência de dados do ESP-NOW pode ser observado na figura 8. Tanto ESP-Emissor, quanto ESP-Receptor possuem as variáveis `leituraLDRchegadaEsquerdo` e `leituraLDRchegadaDireito`. A leitura das variáveis pelo ESP-Emissor transfere essas informações para a estrutura de dados `MyData`, que encapsula e transmite-as para o ESP-Receptor. O ESP-Receptor recebe esses dados na estrutura de dados `MyData` e os transfere para as suas variáveis `leituraLDRchegadaEsquerdo` e `leituraLDRchegadaDireito`.

Figura 8 – Esquemático da transferência de dados do ESP-NOW.

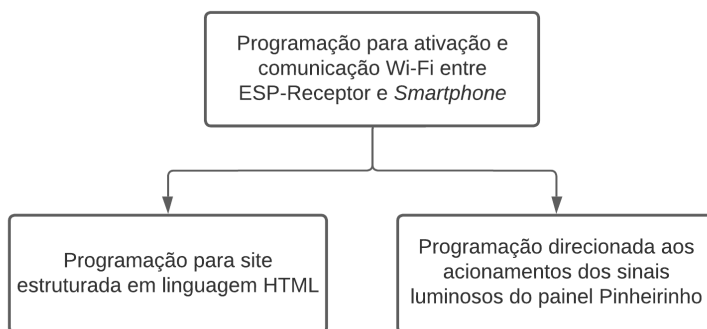


Fonte: Autoria própria

Em paralelo ao protocolo ESP-NOW, também foi desenvolvido o código de comunicação Wi-Fi do ESP32, adaptada a partir da publicação "Esp32 WiFi: Comunicação com a internet" de Teixeira (2019). A comunicação Wi-Fi foi implantada somente no ESP-Receptor a fim de que ele possa gerar um ponto de acesso para conexão com smartphone e exibir informações da corrida mediante um site estruturado em linguagem HTML.

Na programação, a rede Wi-Fi é ativada e se mostra disponível para uso através de um login com SSID (*service set identifier*, que é o nome de identificação da rede) e senha para se conectar com um smartphone. Conforme a figura 9, dentro da conexão Wi-Fi estabelecida, é inserida a programação do site estruturada em linguagem HTML e a listagem de código responsável pelos acionamentos das luzes do painel Pinheirinho.

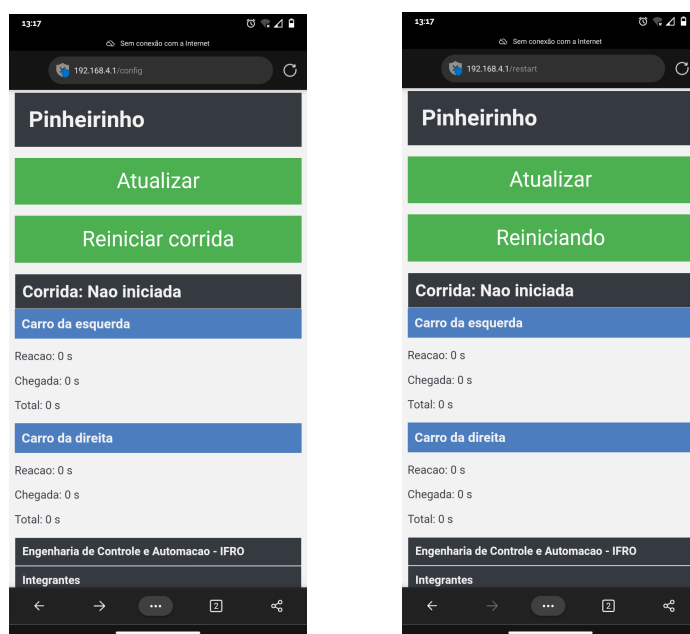
Figura 9 – Fluxograma de hierarquia dos códigos para comunicação WiFi.



Fonte: Autoria própria

Assim que um dispositivo smartphone se conecta à rede WiFi, pode-se acessar o site programado digitando o endereço IP (Protocolo de Internet) do ESP-Receptor como uma URL de site no navegador de internet (Figura 10). O site desenvolvido contém os tempos cronometrados da corrida para cada carrinho, situação da corrida ("Não iniciada", "Em andamento" ou "Concluída") e dois botões: Um botão para atualizar as informações da corrida exibidas na tela e outro botão para reiniciar a contagem dos tempos e desligar as luzes que estiverem acesas no painel Pinheirinho.

Figuras 10 e 11 – Interface de visualização de página web desenvolvida na linguagem HTML (Esquerda) durante uso normal e interface para reiniciar variáveis (Direita).



Fonte: Autoria própria

Além disso, a programação realizada permite que seja feita a leitura da referência final do URL do site, permitindo adaptar a tela conforme a referência utilizada. Na Figura 10, com referência terminada em "/config" é mostrada uma tela com o botão para reiniciar corrida. Já na Figura 11, com referência "/restart", o mesmo botão mostra que se está reiniciando a corrida. Quando um dispositivo conectado estabelece uma conexão ESP-NOW entre o ESP-Emissor e ESP-Receptor, o Pinheirinho estará apto para monitorar a corrida. A programação foi estruturada em casos da função switch case (figura 12), onde foi possível separar as ações por etapas a fim de analisar todo o comportamento da corrida. Todos os casos podem ser analisados na tabela 2.

Tabela 2 – Tabela de casos usados no algoritmo do ESP32
posicionado no início da corrida

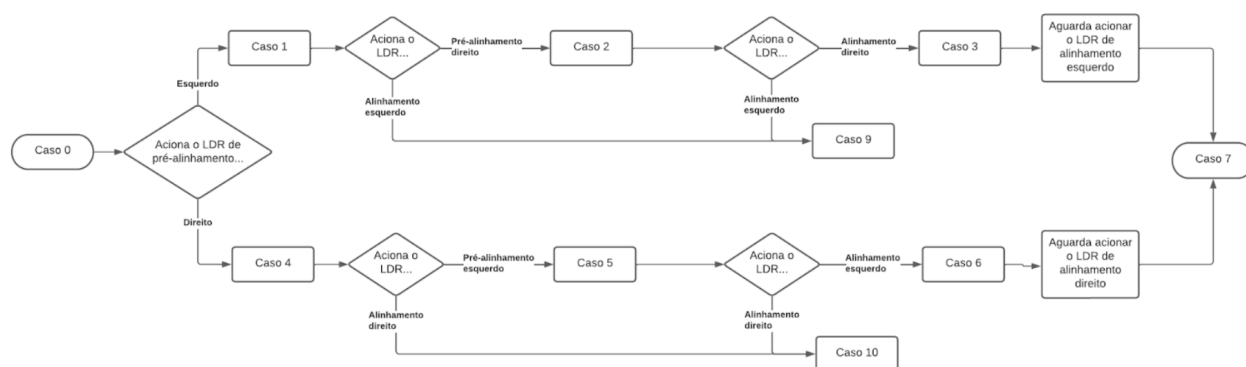
Casos	Grupo de função
Casos 0 ao 6	Pré-alinhamento e alinhamento
Caso 7	Largada
Caso 8	Chegada
Casos 9 e 10	Queima de alinhamento ou queima de largada
Caso 11	Exibição de resultados

Fonte: Autoria própria.

Para realizar todo o processo, foram necessários utilizar casos numerados de 0 a 11, divididos em cinco grupos: Casos de alinhamento (Numerados de 0 a 6), caso de largada (Caso 7), caso de chegada (Caso 8), casos de queima de alinhamento ou queima de largada (9 e 10) e caso para exibição de resultados (Caso 10), conforme mostra a Tabela 2.

Conforme supracitado e mostrado na figura 12, os alinhamentos foram programados para serem executados em uma ordem específica. Durante o alinhamento, os carros têm que sincronizar as passagens pelos sensores em uma ordem dependente de qual carro atravessa primeiro. Caso o carro da esquerda seja o primeiro a cruzar o sensor de pré-alinhamento, ele deve esperar o carro da direita atravessar os sensores de pré-alinhamento e alinhamento da direita para depois avançar até o alinhamento. O caso se inverte caso o carro da direita cruze o pré-alinhamento primeiro. Junto à isso, o Pinheirinho deverá acender os sinais luminosos de pré-alinhamento e alinhamento (amarelos) conforme os carros avançam e acender os sinais de falha (vermelhos) quando um carro avança de forma diferente dessa ordem.

Figura 12 – Casos de alinhamento do Pinheirinho.



Fonte: Autoria própria

Assim que o alinhamento estiver concluído, o caso atuante irá trocar para o caso 7, onde será iniciada a contagem para largada da corrida. Ela será exibida no Pinheirinho de cima para baixo na sequência dos pares de luzes amarelas abaixo do alinhamento. As luzes deverão acender depois de um intervalo de tempo e quando todas estiverem acesas, aciona o par de luzes verdes, dando início a corrida. No tempo em que as luzes amarelas estão acendendo para largada, os sensores deverão dar sinais caso algum carro inicie a corrida antecipadamente, fazendo uma troca do caso atual para um caso de falha (Caso 9 ou 10).

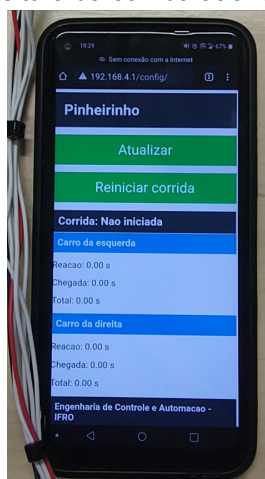
Nessa corrida, será avaliado três tempos para cada carro: O tempo de reação, que é o tempo entre o acionamento do sinal verde para largada e o carro avançar o sensor de largada; o tempo de chegada, sendo o tempo entre o cruzamento do carro entre o sensor de largada e o sensor de chegada, e; o tempo total, o qual é o tempo desde o acionamento do sinal verde de largada até a chegada do carro no fim da corrida.

O microcontrolador receptor também é responsável por criar um ponto de acesso para que se possa conectar um smartphone a fim de se possa conectar a um site para monitorar a corrida. Para criar o layout do site para smartphone, também foi necessária a aplicação de linguagem HTML no código de programação. Os dados serão lidos em duas respostas diferentes: A primeira será pela resposta de sinal luminoso no Pinheirinho e a segunda será pelo smartphone, por comunicação wireless com o ESP32 receptor.

4 RESULTADOS

A conexão estabelecida por meio do protocolo ESP-NOW entre os microcontroladores foi estável e com rápida transferência de informações. Assim, o ESP-Receptor conseguiu utilizar todos os dados dos sensores para monitorar a corrida por meio dos casos do algoritmo implementado no microcontrolador. Além disso, foi realizado com sucesso a exibição dos resultados através da conexão entre smartphone e o ESP-Receptor (Figura 13).

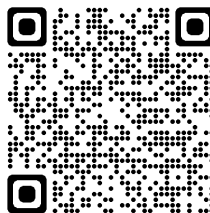
Figura 13 – Imagem da leitura da corrida através do smartphone.



Fonte: Autoria própria

O seguinte QR Code dá acesso a uma pasta de fotos e vídeos do desenvolvimento do projeto, conforme Figura 14:

Figura 14 – QR Code para acesso a fotos e vídeos do trabalho.



Fonte: Autoria própria

5 CONCLUSÃO

O protocolo de conexão ESP-Now é muito útil para ser trabalhado em associação com o módulo *wireless* do microcontrolador ESP32. Através dele, foi possível estabelecer uma comunicação sem fio e estável entre os dispositivos, além de permitir que o algoritmo implantado no ESP-Receptor pudesse acionar os sinais luminosos do Pinheirinho visando monitorar uma corrida de carrinhos em andamento. Em paralelo à isso, usar a conexão do módulo *wireless* do microcontrolador permitiu que fosse possível a conexão entre o ESP-Receptor e um smartphone, permitindo o acompanhando remoto da corrida para exibição dos resultados processados no algoritmo implantado no ESP-Receptor em uma interface programada na linguagem HTML.

REFERÊNCIAS

Arduino Docs. Disponível em: <<https://docs.arduino.cc/>>. Acesso em: 26 maio. 2023.

BEIRÃO, L. J. et al. MONITAGRO – MONITORAMENTO DE ATIVIDADES AGROPECUÁRIAS UTILIZANDO SENSORES E INTERNET DAS COISAS. Anais da Mostra Nacional de Iniciação Científica e Tecnológica Interdisciplinar (MICTI) - e-ISSN 2316-7165, v. 1, n. 12, 2021.

BONDÍA, J. L.. Notas sobre a experiência e o saber de experiência. Revista Brasileira de Educação, n. 19, p. 20–28, jan. 2002.

BRITO, L. L. F. et al. PROTOCOLOS DE COMUNICAÇÃO PARA INTERNET OF THINGS (IOT). Intercursos Revista Científica, v. 17, n. 1, 2019.

CORTEZ, H. B. DE C. et al. Arquitetura IoT para monitoramento de planta industrial: uma comparação dos protocolos MQTT e ESP-Now. [s.d.].

DRONEBOT WORKSHOP. ESP NOW - peer to peer ESP32 communications. Disponível em: <<https://dronebotworkshop.com/esp-now/>>. Acesso em: 28 maio. 2023.

ESP-IDF Programming Guide - ESP32. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/index.html>>. Acesso em: 28 maio. 2023.

Espressif Systems. Disponível em: <<https://www.espressif.com/>>. Acesso em: 28 maio. 2023.

FAGUNDES, F. D. Comparativo entre plataformas de desenvolvimento de sistemas microcontrolados para aplicações em internet das coisas. [s.d.].

FreeCAD. Disponível em: <<https://www.freecadweb.org/>>. Acesso em: 28 maio. 2023.

FUELTECH. Tudo que você sempre quis saber sobre o pinheirinho! Disponível em: <<https://www.youtube.com/watch?v=JiMuah0zJSY>>. Acesso em: 28 maio. 2023.

GOMES, R. Relatório de Projeto - Projeto Rc Car. Escola Superior de Tecnologia e Gestão, dez. 2020.

LUGLI, A. B.; SANTOS, M. M. D. Redes industriais para automação industrial. [s.l.] Saraiva Educação S.A, [s.d.].

SECCO, R. A. M. Desenvolvimento de uma rede de sensores sem fio em topologia mesh utilizando dispositivos IOT. [s.l.] Universidade Tecnológica Federal do Paraná, 4 jul. 2019.

SERGERS, J. Analysis techniques for racecar data acquisition. Warrendale: SAE International, 2014.

SILVA, E. C. DA; LOVATO, L. A. Framework Scrum: Eficiência em Projetos de Software. Revista de Gestão e Projetos, v. 07, n. 02, p. 01–15, 2016.

SOUZA, M. V. DE. Domótica de baixo custo usando princípios de IoT. [s.l.] Brasil, 11 ago. 2016.

TEIXEIRA, G. ESP32 WiFi: Comunicação com a Internet. Blog UsinainfoUsinainfo, , 20 maio 2019. Disponível em: <<https://www.usinainfo.com.br/blog/esp32-wifi-comunicacao-com-a-internet/>>. Acesso em: 28 maio. 2023

PROGRAMMING PROPOSAL IN PINHEIRINHO USING ESP-NOW PROTOCOL

Abstract: *The present work proposed a development for programming a Full Tree, a light-signal panel for racing light-powered racing cars. The proposal involved concepts such as wireless communication through the protocol ESP-Now and hotspot development for smartphone connection for viewing results. In addition, a programming was carried out in order specific for activating light signals do Pinheirinho, timing reaction times, arrival and total journey of each trolley. The project proved to be effective with stability in communication without wire held and display of race results via remote via smartphone.*

Keywords: *ESP-NOW, wireless, Protocols, HTML.*