



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

José Eduardo Garcez

Retrofit de Instrumentos de Veículo Antigo Utilizando ESP8266-ESP12

Araranguá
2022

José Eduardo Garcez

Retrofit de Instrumentos de Veículo Antigo Utilizando ESP8266-ESP12

Trabalho de Conclusão de Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro de Computação.

Orientador: Prof. Orientador, Dr. Fábio Rodrigues De La Rocha

Araranguá
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Garcez, José Eduardo

Retrofit de Instrumentos de Veículo Antigo Utilizando
ESP8266-ESP12 / José Eduardo Garcez ; orientador, Fábio
Rodrigues De La Rocha , 2022.

50 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. Sistemas Embarcados. 3.
Aquisição de Sinais. 4. Microcontroladores. 5. App. I. De
La Rocha , Fábio Rodrigues . II. Universidade Federal de
Santa Catarina. Graduação em Engenharia de Computação. III.
Título.

José Eduardo Garcez

Retrofit de Instrumentos de Veículo Antigo Utilizando ESP8266-ESP12

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Engenheiro de Computação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 26 de julho de 2022.

Banca Examinadora:

Prof. Roderval Marcelino, Dr.
Instituição UFSC

Prof. Jim Lau, Dr.
Instituição UFSC

RESUMO

Este trabalho apresenta o projeto e implementação de um protótipo embarcado veicular que é capaz de capturar dados de sensores ligados ao funcionamento do veículo (níveis de fluidos, temperaturas, e pressões) e atuar sobre alguns componentes (temperatura do ar condicionado e de um refrigerador de alimentos). Com o sistema é possível realizar verificações dos sensores e disparar alertas de falhas. O protótipo foi idealizado para uma realidade de um veículo modelo antigo onde a eletrônica embarcada é pequena ou nula e serve para propor uma solução de um sistema de instrumentação, que ofereça opções de monitoramento e controle de sistemas, para veículos automotores antigos. A implementação é realizada utilizando microcontroladores ESP8266-ESP12 em função da existência de uma interface de rádio padrão 802.11 interna e sua capacidade de atuar como um *access point* (AP) e servidor de conteúdo. A aplicação do protótipo foi realizada através de um aplicativo móvel Android e páginas desenvolvidas com html e javascript que conectam-se ao servidor de conteúdo do microcontrolador e obtém dos dados/alarmes e controla a ativação de motores, solenoides, entre outros.

Palavras-chave: sistemas embarcados; aquisição de dados; microcontroladores; App.

ABSTRACT

This work presents the design and implementation of an embedded vehicle prototype that is capable of capturing data from sensors linked to the vehicle's operation (fluid levels, temperatures, and pressures) and acting on some components (air conditioning and a refrigerator temperature). With the prototype it is possible to perform sensor checks and trigger fault alerts. The prototype was designed for a reality of an old model vehicle where the on-board electronics is small or null and serves to propose a solution for an instrumentation system, that offers monitoring and control systems options for motor vehicles old. The implementation is performed using ESP8266-ESP12 microcontrollers due to the existence of an internal standard 802.11 radio interface and its ability to act as an access point and content server. The application of the prototype was carried out through an Android mobile application and pages developed with html and JavaScript that connect to the microcontroller's content server and obtain data/alarms and control the activation of motors, solenoids, among others.

Keywords: embedded system; data acquisition; microcontrollers; App.

LISTA DE FIGURAS

Figura 1 – Primeiro automóvel.	18
Figura 2 – Diagrama elétrico do Ford T 1919.	19
Figura 3 – Aquisição de dados.	25
Figura 4 – Microprocessador 8080 - ligações dos pinos.	26
Figura 5 – MCU 8051 - Ligações dos pinos.	27
Figura 6 – Arquitetura de projeto embarcado.	28
Figura 7 – Arquitetura ESP8266EX.	29
Figura 8 – Comunicação UART.	30
Figura 9 – Topologia de conexões USB.	32
Figura 10 – Conexões I2C.	33
Figura 11 – Protocolo I2C.	33
Figura 12 – Conexões SPI.	34
Figura 13 – Protocolo SPI.	34
Figura 14 – Conexões 1-Wire.	35
Figura 15 – Modelos OSI e TCP/IP.	36
Figura 16 – Especificações 802.11.	37
Figura 17 – Kit de desenvolvimento NodeMCU V1.0.	40
Figura 18 – Descrição pinos NodeMCU.	40
Figura 19 – Sensor de temperatura DS18B20 a prova d'água.	42
Figura 20 – Módulo sensor de temperatura e pressão BMP280.	43
Figura 21 – Módulo ZS-042 RTC.	44
Figura 22 – Sensor de nível de fluído de arrefecimento- ES7510.	45
Figura 23 – Sensor de nível de combustível.	45
Figura 24 – Mostrador nível de combustível.	46
Figura 25 – Pastilha de efeito termoelétrico.	47
Figura 26 – Adaptador microSD Catalex.	48
Figura 27 – PCF8574 - Expansor 8 bits para barramento I2C.	48
Figura 28 – Módulo 8 relés.	49
Figura 29 – Diagrama elétrico do protótipo.	50
Figura 30 – Layout placa de circuito impresso (PCB) do protótipo.	51
Figura 31 – Arquitetura cliente servidor.	52
Figura 32 – Interface gráfica do aplicativo frontend.	54
Figura 33 – Página inicial do app.	55
Figura 34 – Página principal do app.	56
Figura 35 – Programação do aplicativo frontend app.	57
Figura 36 – Página html servidor http.	58
Figura 37 – Página html/javascript <i>websocket</i>	59

Figura 38 – Formato do arquivo de armazenamento das informações.	64
Figura 39 – Visão Geral do Sistema.	70

LISTA DE QUADROS

LISTA DE TABELAS

Tabela 1 – Aferição do medidor de combustível.	46
--	----

LISTA DE ABREVIATURAS E SIGLAS

ABS	<i>Antilock Braking System</i>
ADC	<i>Analog-to-Digital Converter</i>
ANFAVEA	Associação Nacional de Fabricantes de Veículos Automotores
AP	<i>Access Point</i>
CI	Círculo Integrado
CPU	<i>central Processing Unit</i>
CSMA/CA	<i>Carrier Sense Multiple Access / Collision Avoidance</i>
DAC	<i>Digital-to-Analog Converter</i>
DIP	<i>Dual In-line Package</i>
ECU	<i>Engine Control Unit</i>
FTP	<i>File Transfer Protocol</i>
GM	General Motors
GND	<i>Ground</i>
GPS	<i>Global Positioning System</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I2C	<i>Inter-Integrated Circuit</i>
ICMP	<i>Internet Control Message Protocol</i>
IDE	<i>Integrated development Environment</i>
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IP	<i>Internet Protocol</i>
ISO	<i>International Standards Organization</i>
LAN	<i>Local Area Network</i>
LLC	<i>Logical Link Control</i>
MAC	<i>Medium Access Control</i>
MCU	<i>Microcontroller Unit</i>
MISO	<i>Master In Slave Out</i>
MOS	<i>Metal-Oxide Semiconductor</i>
MOSI	<i>Master Out Slave In</i>
NRZI	<i>Non Return to Zero Inverted</i>
OSI	<i>Open Systems Interconnection</i>
PROM	<i>Programmable Read-Only Memory</i>
RAM	<i>Random Access Memory</i>
ROM	<i>Read-Only Memory</i>
RTC	<i>Real Time Clock</i>
Rx	recepção de dados
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>

SMTP	<i>Simple Mail Transfer Protocol</i>
SOC	<i>System On a Chip</i>
SPI	<i>Serial Peripheral Interface</i>
SS	<i>Slave Select</i>
STA	<i>Station</i>
TCP	<i>Transmission Control Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
Tx	Transmissão de dados
UART	<i>Universal Asynchronous Receiver Transmitter</i>
UDP	<i>User Datagram Protocol</i>
URI	<i>Uniform Resource Identifier</i>
USB	<i>Universal Serial Bus</i>
VW	VolksWagen
WEP	<i>Wired Equivalent Privacy</i>
WLAN	<i>Wireless Local Area Network</i>
WPA	<i>WiFi Protected Access</i>
WS	<i>Web Socket</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
1.1.2.1	<i>Compreender o funcionamento básico do veículo teste</i>	15
1.1.2.2	<i>Projetar o sistema</i>	15
1.1.2.3	<i>Desenvolver o sistema</i>	15
1.1.2.4	<i>Realizar bateria de testes</i>	15
1.1.2.5	<i>Analisar os dados obtidos</i>	16
1.2	METODOLOGIA	16
2	REVISÃO DE LITERATURA	17
2.1	CENÁRIO DA ELETRÔNICA EM AUTOMÓVEIS	17
2.2	AQUISIÇÃO DE DADOS	24
2.3	MICROPROCESSADORES E MICROCONTROLADORES	25
2.4	PROTOCOLOS DE COMUNICAÇÃO	30
2.4.1	Transmissor/receptor universal assíncrono (UART)	30
2.4.2	Barramento serial universal (USB)	31
2.4.3	Circuito inter integrado (I2C)	32
2.4.4	Interface periférica serial (SPI)	34
2.4.5	1-Wire	35
2.4.6	Protocolo de Controle de transmissão e internet (TCP/IP) .	35
2.4.7	Redes WiFi 802.11	36
3	PROJETO E DESENVOLVIMENTO	38
3.1	PROTÓTIPO HARDWARE	39
3.1.1	Microcontrolador	39
3.1.2	Periféricos externos - sensores, módulos e atuadores	41
3.1.2.1	<i>Sensor de temperatura DS18B20</i>	42
3.1.2.2	<i>Sensor de temperatura e pressão BMP280</i>	42
3.1.2.3	<i>Módulo ZS-042 RTC</i>	43
3.1.2.4	<i>Sensor de nível de fluido de arrefecimento ES7510</i>	44
3.1.2.5	<i>Sensor de combustível</i>	45
3.1.2.6	<i>Pastilha de efeito Peltier TEC1-12706</i>	47
3.1.2.7	<i>Módulo microSD</i>	48
3.1.2.8	<i>Expansor 8 bits para barramento I2C - PCF8574</i>	48
3.1.2.9	<i>Módulo de relés</i>	49
3.1.2.10	<i>Circuitos auxiliares</i>	49
3.1.3	Integração do hardware	50

3.2	PROTÓTIPOS <i>SOFTWARE</i>	51
3.2.1	Arquitetura cliente servidor	52
3.2.2	Software do servidor <i>web</i>	52
3.2.3	<i>Software</i> do cliente <i>web</i> - app Android	54
3.2.4	Software do cliente <i>web</i> - <i>browser http</i>	57
3.2.5	<i>Software</i> do cliente <i>web</i> - <i>browser websocket</i>	58
3.2.6	<i>Software</i> do cliente <i>web</i> - outras páginas	59
4	CONCLUSÃO E TRABALHOS FUTUROS	61
	REFERÊNCIAS	65
	ANEXO A – Visão geral do sistema	70

1 INTRODUÇÃO

Conforme a pesquisa realizada, mesmo os primeiros veículos, assim como, o Ford Modelo T, produzidos a mais de cem anos atrás, já utilizavam dispositivos elétricos. Componentes, tais como, baterias, dispositivos atuadores, assim como, lâmpadas e buzinas, e transdutores, assim como, os dínamos, já equipavam um Ford T em 1919. Observando-se, a evolução histórica dos dispositivos eletroeletrônicos, em paralelo, com a evolução dos veículos automotores. Percebe-se, desde os primórdios, que existe uma ligação estreita entre esses dois setores. No princípio, integram-se aos automóveis, componentes elétricos que poderiam melhorar aspectos básicos, assim como, dar partida em um motor ou iluminar um caminho escuro. Em um segundo momento, torna-se o tempo de permanência no veículo, mais agradável ou informativo, por meio da inclusão de rádios e reprodutores de mídias. Evoluindo-se, para um maior estreitamento dessa relação, empregam-se, dispositivos eletrônicos mais complexos e especialmente projetados para veículos. Com isso, busca-se melhorar, o desempenho e a eficiência, dos motores, câmbios, suspensões, freios, sistemas de informação e sistemas de entretenimento. Dispositivos complexos, assim como, ABS, ECU e sistemas multimídia, acompanham grande parte dos veículos produzidos na atualidade. Com a aceleração, do desenvolvimento da indústria de componentes eletrônicos, à partir da década de 1980, impulsionada de acordo com previsões de Moore, onde, o número de componentes eletrônicos, que constituem um circuito integrado, dobraria a cada dezoito meses. Aliado a isso, o surgimento e a rápida popularização, dos microprocessadores e microcontroladores. Passa a existir uma situação, na qual, os automotores, projetados como bens de consumo duráveis, que permanecem funcionais por décadas, trazem consigo, equipamentos eletrônicos, que se tornam rapidamente defasados. Percebe-se que essa defasagem, ocorre principalmente, em itens de conforto e conveniência e sistemas de informação e entretenimento. Visando-se, diminuir essa defasagem tecnológica, desenvolve-se este trabalho. Propondo-se, a elaboração de um sistema embarcado, com a finalidade de adquirir dados, assim como, níveis de fluidos, temperaturas, e pressões, provenientes de sensores alocados em um veículo. Com isso, pretende-se implementar, dispositivos de alarmes e controles, baseados nas leituras obtidas desses sensores. Desse modo, pretende-se sugerir uma solução, para viabilizar um sistema de instrumentação, que ofereça opções de monitoramento e controle de sinais, para veículos automotores antigos, que não disponham desse tipo de recurso. Ou promover melhorias, quanto ao funcionamento, de dispositivos existentes. Para tanto, disponibiliza-se o acesso às medidas realizadas, através de um sistema telemático sem fio, utilizando-se o microcontrolador ESP8266-ESP12, o qual, possui uma interface de rádio, com o protocolo de comunicação 802.11 já integrados, e que também, fornece a possibilidade de funcionar como um servidor de conteúdo. Para realizar-se as interações com o servidor, opta-se por desenvolver um app para Android e páginas para *browsers*, onde podem ser visualizado os dados cole-

tados, alarmes de funcionamento e ligar ou desligar dispositivos como motores elétricos, solenoides, e outros.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral do trabalho é identificar um problema que pode ser resolvido com eletrônica embarcada e sobre esse aplicar os conceitos de amostragem de dados, controle e ainda visualização de dados utilizando aplicativos móveis. Como aplicação alvo, foi escolhido o retrofit de um automóvel, isto é, ser capaz de desenvolver a eletrônica embarcada para sensoriamento e controle de alguns elementos de um automóvel antigo que originalmente dispõe de pouca ou nenhuma eletrônica embarcada.

1.1.2 Objetivos Específicos

1.1.2.1 Compreender o funcionamento básico do veículo teste

Verificar aspectos de funcionamento que podem ser melhorados com um sistema embarcado, determinar os locais e formas de instalação de sensores e atuadores sem comprometer ou interferir no funcionamento dos dispositivos existentes.

1.1.2.2 Projetar o sistema

Levantar os materiais necessários, microcontroladores, sensores de temperatura, sensores de pressão atmosférica, sensores de nível de fluidos, sensores de rotação, sistema de relógio e sistema de armazenamento de informações, acionadores de cargas elétricas e projetar os circuitos auxiliares necessários para atender os requisitos funcionais e não funcionais listados na relação de requisitos apresentada no capítulo de projeto e desenvolvimento.

1.1.2.3 Desenvolver o sistema

Implementar uma solução embarcada (*software/hardware*), utilizando sensores eletrônicos para substituir ou complementar os elementos mecânicos/eletromecânicos existentes no veículo. Realizar a integração do hardware através de placas de placas de circuito impresso e conectores, implementar aplicações para visualização e controle dos elementos veiculares e ainda identificar as melhores alternativas de prover comunicação entre o veículo e os aplicativos.

1.1.2.4 Realizar bateria de testes

Realizar testes para verificar o funcionamento e desempenho do sistema.

1.1.2.5 Analisar os dados obtidos

Verificar se os dados obtidos na bateria de testes são condizentes com os valores esperados.

1.2 METODOLOGIA

Para perseguir os objetivos propostos, realiza-se uma pesquisa de cunho exploratório, com caráter qualitativo e quantitativo, buscando determinar quais seriam os requisitos obrigatórios e os desejáveis, no que tange ao funcionamento do sistema. Com isso, busca-se na bibliografia existente os trabalhos similares, a fim de detectar o estado da arte, avaliando-se quais as técnicas, componentes de *hardware* e plataformas de desenvolvimento de *software* estão sendo utilizadas para desenvolver aplicações semelhantes. Em um segundo momento, a pesquisa desenvolve-se de maneira experimental, buscando-se avaliar qualitativamente, aspectos como o *design* e usabilidade, do sistema desenvolvido e quantitativamente, avaliando-se quais procedimentos e funções apresentam melhor desempenho, assim como, tempo de resposta, quantidade de memória necessária, aferição de medidas entre outros.

2 REVISÃO DE LITERATURA

2.1 CENÁRIO DA ELETRÔNICA EM AUTOMÓVEIS

Conforme a pesquisa realizada, os primeiros registros, do que conhecemos hoje como automóvel, surgiram dos arquivos de Leonardo Da Vinci, através de ideias concebidas por volta do século XVI, as quais, vislumbravam um tipo de carroça, feita principalmente de madeira, que se movimentaria sobre três rodas e que não possuía artefatos de ligação para tração proveniente de animais, ou seja, conotava o uso de algum tipo de auto propulsão, tal como, um motor, aparato que era indisponível nessa época. (KENNEDY, 2004).

Com o advento dos motores movidos à vapor, no início do século XVIII, e após algumas décadas de seu desenvolvimento, surge a invenção da carroça a vapor em 1769, atribuída ao engenheiro francês Nicolas Joseph Cugnot, a qual, é considerada por muitos, como o primeiro meio de transporte não movido por animais, sendo assim, autopropulsionado. Tal invento, configurou-se como um marco evolutivo do automóvel, e que materializou a ideia inicial de Da Vinci. (GODINHO, 2020).

Contemporaneamente, porém no início do século XIX, desenvolvia-se em paralelo, o estudo do físico italiano Alessandro Volta. Volta, dedicava-se a entender os fenômenos elétricos. Suas descobertas deram origem ao acumulador de energia conhecido como Pilha de Volta, a qual, seria a precursora, da bateria de chumbo, criada pelo físico francês, Gaston Planté em 1859. Essa nova tecnologia, proposta por Planté, que após sucessivas etapas de desenvolvimento, é utilizada até os dias atuais em uma variedade de aplicações, tais como: sistemas de alarmes, inversores de energia, sistemas de telecomunicações, bem como, em acumuladores de energia para veículos automotores dos mais variados, quais sejam, motocicletas, tratores, caminhões e também automóveis. Além disso, no último quarto desse século, foi patenteado o telefone, por Alexander Graham Bell, a válvula e a lâmpada elétrica por Thomas Edison e Guglielmo Marconi realizou a primeira transmissão de sinal de rádio, fato que, possibilitaria a transmissão de mensagens entre locais distantes, através do telegrafo sem fio.(FONSECA FILHO, 2007).

Por volta do final do século XIX, surgem os primeiros motores de combustão interna, com destaque para invenção do motor de combustão Otto, desenvolvido pelo engenheiro alemão Nikolaus Otto e cujos princípios, também são utilizados, em muitos automóveis fabricados nos dias de hoje. (GODINHO, 2020).

Assim, com a imaginação de Da Vinci, vislumbrando um meio transporte, movido por uma força motriz, não disponível em sua época. Conceito qual, materializou-se aproximadamente 300 anos após sua morte, com a carroça a vapor de Cugnot. A qual, daria lugar, à invenção de Karl Benz, que em 1886, patenteou o primeiro automóvel com motor a combustão interna e com tecnologia Otto. Lado a lado, com os avanços na área da eletricidade e magnetismo, proporcionam um rápido desenvolvimento dos veículos com motores de combustão interna. (CABRAL, 2020).

Figura 1 – Primeiro automóvel.



Fonte: Mercedez-Benz (2022)

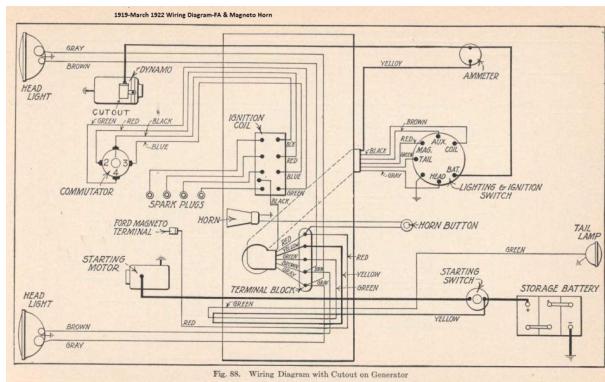
Mesmo que rudimentar, o modelo de Bens, apresentado na Figura 1, seria, através de uma visão sistêmica, uma versão próxima dos modelos de veículos atuais. Tal veículo, possuía um sistema de sustentação ou carroceria, que servia para acomodar os ocupantes do veículo e os demais sistemas. Nesse modelo de automóvel, a carroceria era constituída de uma estrutura de tubos metálicos, estribos e bancos de madeira. Existia também, um sistema de propulsão Otto, o qual podia ser alimentado por gasolina, e um sistema de transmissão de energia do motor para o solo formado por um conjunto de polias correias e rodas. Existiam também, um sistema de direção com eixos e alavancas, e um sistema de freios composto de alavancas e abrasivos ligados às rodas. (MERCEDEZ-BENZ, 2022).

Com uma década de evolução, e com o aumento da utilização dos automóveis, em 1908, o industrial Henry Ford, passou a produzir carros padronizados em massa, sendo que, o primeiro modelo Ford T, já possuía um sistema de iluminação, que utilizava uma bateria de chumbo, nos moldes da inventada por Gaston Planté e lâmpadas elétricas, inventadas por Thomas Edison 1879. Pode-se perceber, observando-se a Figura 2, que representa o esquema de ligações elétricas de um Ford Modelo T de 1919, que o sistema elétrico desse modelo, contava com transdutores, assim como, dínamos e atuadores, assim como, buzina e motor de partida.

Certamente, o sistema elétrico de um Ford T, não possuía mais que alguns poucos metros de fios, ligando algumas lâmpadas e chaves. Conforme dados sugeridos por Coutinho (2021), em meados do século XX, a quantidade média de fios, contidos em um veículo de produção em série, já era de 45 metros, significando um aumento considerável na complexidade do sistema elétrico dos automóveis. Esse autor, ainda no diz, que na atualidade, um carro normal, com motor a combustão, carrega aproximadamente 1 Km de fios, sendo que, os com eletrônica mais sofisticada, possuem 4 Km de fios instalados, ressaltando-se, que os carros com propulsão elétrica, podem ter uma quantidade de condutores, quatro vezes maior do que isso.

Para Anjos (2011), a história do automóvel no Brasil, inicia-se, no momento da

Figura 2 – Diagrama elétrico do Ford T 1919.



Fonte: MFCA (2017)

primeira importação de um carro Peugeot Type 3, em 1893, adquirido pela família Santos Dumont. Ainda conforme esse autor, em 1903, existiriam seis automóveis circulando na cidade de São Paulo, sendo que, no ano seguinte, o número de veículos circulando, já era de 83. Com o crescimento acelerado, do mercado nacional de veículos automotores, em 1919, a Ford Motors Company, passa a comercializar no Brasil o modelo Ford T. Dado o sucesso desse empreendimento, outras montadoras foram atraídas pelo mercado brasileiro. Em 1925 a General Motors (GM) passou a comercializar a marca Chevrolet, e também obteve, um grande sucesso de vendas no mercado nacional.

Além disso, nesta época, compreendida entre a Primeira Guerra Mundial em 1914 e o fim da Segunda Guerra em 1945, conforme Mateus e Loureiro (1998), ocorreu um rápido desenvolvimento dos sistemas de comunicações, que teria sido impulsionado, pelas demandas das grandes guerras. Nesse momento, os centros de estratégia militar, encontravam-se distantes dos frontes de batalha. Assim, os sistemas de rádio, comunicação e criptografia, ganharam grande destaque. Na medida em que, interceptar uma mensagem do inimigo, sobre o movimento de tropas ou frotas de navios, constituía-se numa vantagem competitiva muito valorosa, podendo até determinar qual seria o lado vencedor.

Como exemplos destes avanços tecnológicos, em 1921, entra em funcionamento o sistema comercial de Radiodifusão nos Estados Unidos. Já em 1927, é estabelecida, uma linha de radiotelefonia entre Londres e Nova York. Em 1936, Alan Turing desenvolve conceitos, que dariam origem aos primeiros computadores. A máquina, proposta idealmente por Turing, seria a precursora de computadores, assim como, o Colossus, utilizados para decodificar mensagens de guerra, criptografadas pelo Eixo, que era formado pelos países da Alemanha, Itália e Japão. Esse tipo de tecnologia, teria sido, de grande relevância, na determinação do fim da Segunda Guerra, vencida pelos Aliados, Estados Unidos, União Soviética e Inglaterra. Essas tecnologias, permitiu que os Aliados, conseguissem interceptar e decodificar, mensagens do inimigo. As quais, forneciam as posições das frotas de guerra

e as ordens de movimentação de tropas do Eixo.(MATEUS; LOUREIRO, 1998).

Durante o período em que se desenvolveu a Segunda Guerra Mundial, na década de 1940, houve uma escassez de peças de reposição para automóveis no mercado nacional, isso forçou, o desenvolvimento de pequenas oficinas, voltadas a produzir peças, para manutenção da frota de veículos em circulação. Com o fim desse período guerra, onde, grande parte dos esforços produtivos internacionais, estavam voltados para atender as demandas urgentes das batalhas. Na década de 1950, a Volkswagen (VW), trouxe para o Brasil, as primeiras trinta unidades do Fusca e algumas Kombis. Nesse momento, surge a ideia, de nacionalização da indústria automobilística brasileira, como medida, para impulsionar a produção local de motores. Assim, governo brasileiro estabeleceu, que somente poderiam entrar em território nacional, veículos totalmente desmontados e sem peças componentes que já fossem fabricadas localmente.(ANJOS, 2011).

Conforme Mehl (2013), nesta época pós guerra, já eram bem conhecidas as válvulas eletrônicas e os relés eletromecânicos. E que esses dispositivos, seriam amplamente utilizados, em sistemas de comunicações de rede física e de radiofrequência, os quais, foram muito empregados e desenvolvidos durante a Segunda Guerra. Também nesse período, algumas empresas, estavam desenvolvendo o dispositivos semicondutores, tais como, o diodo e o transistor de silício, os quais, viriam mais tarde, substituir as frágeis válvulas em muitas aplicações. Desse modo, permitindo-se, que aparelhos como rádios, televisores, carregadores de baterias, etc., se tornassem mais resistentes, menores, com menor consumo de energia e mais baratos. Assim, em decorrência, do grande crescimento da indústria automobilística mundial e ao grande interesse do público norte-americano por programas de rádio, teria sido fabricado, pelos irmãos Galvin, o primeiro rádio para automóveis, denominado Motorola. Esse tipo de aparelho, passou a ser oferecidos pela indústria automobilística, como um acessório, para veículos produzidos à partir da década de 1950.

Com o avanço da eletrônica, na década de 1960, as válvulas a vácuo, foram rapidamente substituídas, pelos transistores e diodos de silício. Isso permitiu, que a fabricação de equipamentos, assim como, rádios, televisores, calculadoras, entre outros, fossem produzidos, com um custo mais baixo, tamanho menor e com maior confiabilidade dos circuitos eletrônicos. Buscando-se, ainda maior estabilidade e redução de tamanho dos circuitos, surgem os primeiros circuitos integrados (CI). Um CI, poderia reunir, em uma única pastilha de silício, diversos transistores, e até mesmo, circuitos eletrônicos inteiros. Conforme Mehl (2013), a Texas Instruments e a Fairchild Semiconductor, foram as primeiras empresas a produzir esse tipo de dispositivo semi condutor, sendo que, o primeiro integrado fabricado pela Fairchild, em 1960, era um flip-flop, que continha, quatro transistores produzidos em uma única pastilha de silício de 3 x 3 mm, o qual, seria utilizado em equipamentos militares.

Segundo dados, fornecidos pela Associação Nacional de Fabricantes de Veículos Automotores (ANFAVEA), a produção total de veículos automotores no Brasil, no ano

de 1957, teria sido de 30.542 unidades, constituindo-se em 34,21 % de automóveis, 5,20 % de veículos comerciais leves, 53,23 % de caminhões e 7,35 % ônibus. No ano seguinte, a produção total passaria para 60.938, significando um aumento anual de aproximadamente 100 % na quantidade produzida.(ANFAVEA, 2022).

Aproximadamente, dez anos após, a produção dos primeiros CI's, foi fundada, a empresa Intel Corporation, por dois ex-funcionários da Fairchild, que trabalharam no projeto do primeiro circuito integrado, Robert Noyce e Gordon Moore. Em 1971, a Intel passou a produzir, os primeiros microprocessadores Intel 4004 de 4 bits, o qual, possuía uma capacidade de cálculo, superior aos gigantescos computadores do fim da Segunda Guerra, tais como, o ENIAC e o Colossus. O sistema, idealizado pela Intel, utilizava-se de um conceito, que é empregado até os dias atuais. O paradigma proposto por Von Neumann, da década de 1940. Esse modo de pensar um sistema, baseava-se, na divisão do sistema computacional, em unidades de controle, aritmética, memória e interfaces de entrada e saída. Tal arquitetura, possibilitava uma certa flexibilidade ao conjunto, com o qual, poderia gerar vários tipos de aplicações, alterando-se apenas, o programa armazenado na memória permanente, ROM (*Read-Only Memory*). Assim, para viabilizar a arquitetura Von Neumann, também foram projetados e produzidos, o CI 4001, com a funcionalidade de memória de programa ROM, e que contava com 2k bits de capacidade; o 4002, que era uma memória de execução (RAM), com 320 bits, sendo constituída de 80 endereços com 4 bits de dados cada; e o 4003, que era um dispositivo de entradas e saídas de 10 bits. Esse novo sistema, obteve grande sucesso de vendas, estimulando a Intel, a investir em processadores mais poderosos e com mais memória. Em 1975, por ocasião do lançamento comercial, de um chip de memória, que já era composto por cerca de 64.000 transistores, Gordon Moore afirmou, que o número de componentes, que fariam parte de um circuito integrado, dobraria a cada dezoito meses. (MEHL, 2013).

Para ilustração de custos, no mercado automotivo, conforme anúncio no jornal Folha de São Paulo. Em 1970, um VW Fusca novo, poderia ser comprado no Brasil, por aproximadamente Cr\$ 13.200,00. Considerando-se a taxa de câmbio, do Dólar Americano frente ao Cruzeiro, que estava por volta de 4,80 Cr\$/Us\$, significava que esse carro, custava na época, aproximadamente Us\$ 2.750,00. (FOLHA, 1970). Esse valor, seria um pouco maior, do que o valor de uma calculadora, que utilizava a família de circuitos integrados de primeira geração da Intel, a qual, conforme Mehl (2013), poderia ser comprada por mais de Us\$2.000,00, em 1971. Esse comparativo de valores, indicaria que nesse momento, ainda não seria viável, utilizar esse tipo de tecnologia, em automóveis destinados ao mercado popular, devido ao seu alto custo.

De acordo com ANFAVEA (2022), no Brasil, a produção de veículos em 1969 foi de pouco mais de 350.000 unidades, sendo que, os principais fabricantes eram a General Motors (GM), a VolksWagen (VW) e a Ford, que juntas, dominavam o mercado. Este período, compreendido entre 1968 e 1973, no Brasil, experimentou-se grandes taxas de

crescimento da produção. Esse momento ficou conhecido, como o período do milagre econômico brasileiro. Tal prosperidade, atraiu a atenção para o mercado nacional de automotores. A Agrale, amplia sua produção, com a inauguração de sua fábrica própria em Caxias do Sul, em 1975. No ano seguinte, a Mercedes-Benz, inicia a construção de uma fabrica em Campinas e a FIAT, inaugura sua fábrica em Betim. Dentre os carros mais populares, nessa década, estavam o Ford Corcel, o Fusca e o Passat da VW, o GM Chevete e o Fiat 147, todos eles com quase nenhuma tecnologia eletrônica, se comparados com veículos atuais.

Cabe-se retomar, que no Brasil, durante o regime da ditadura militar, de 1964 a 1985, existia uma ideia de protecionismo de mercado, a qual, visava fomentar o desenvolvimento da indústria nacional em vários segmentos. Para tanto, muitos insumos, equipamentos e produtos eram sobretaxados ou até mesmo proibidos para importação. Tal política, causou um atraso tecnológico, nos veículos fabricados domesticamente e a industria eletrônica local, desenvolveu-se numa velocidade muito menor do que os concorrentes externos. Com isso, somente no final da década de 1980, é que surgem, as primeiras aplicações comerciais, utilizando microprocessadores em veículos. Como por exemplo, a Engine Control Unit (ECU), ou unidade de controle do motor, que eram dispositivos que controlavam a injeção de combustível nos motores. Esse tipo de sistema, foi desenvolvido na Europa, pela empresa alemã Bosch, na década de 1960, e um dos primeiros veículos nacionais a contar com esse tipo de tecnologia foi VW Gol-GTI em 1988. (ANFAVEA, 2022).

Conforme Kume, Piani e Souza (2003), até o final da década de 1980, a industria nacional, era baseada em um processo de substituições de importações, o qual, impunha barreiras a entrada de produtos estrangeiros, que tivessem um similar produzido em território nacional. Essas barreiras poderiam ser tanto tarifárias, com a imposição de taxas de importação muito elevadas, quanto não tarifárias, impondo limitação de quantidades importadas ou através do uso de listas de produtos com importação proibida. Em 1990, começou uma política de abertura econômica, onde, uma das medidas adotadas, foi a de abolir as barreiras não tarifárias. Isso resultou, no fim da proibição de importar, cerca de mil e duzentos produtos, que vigorava desde 1975.

Com essa abertura do mercado nacional, as montadoras instaladas no país, começaram a produzir para o mercado nacional, veículos com tecnologias, já utilizadas por elas, na produção de veículos em outros países. E até mesmo, comercializar localmente, modelos de automóveis, que foram produzidos em suas montadoras no estrangeiro. Esse movimento, também acabou atraindo para o país, novas montadoras, assim como, a Peugeot, Renault e Citroen, as quais, passaram a comercializar domesticamente, os seus produtos, que haviam sido desenvolvidos para outros mercados mais exigentes.(ANJOS, 2011).

Juntamente com a liberação de importações, cresceu também, a quantidade de eletrônica embarcada nos veículos. Atualmente, conforme ANFAVEA (2022), a produção

nacional de veículos total, em 2018, aproximou-se de 2,7 milhões de unidades ao ano. Em 2017 foram produzidos pouco mais de 2,2 milhões de automóveis, significando um aumento de aproximadamente 20 % no total produzido, além disso, os automóveis populares atuais, de acordo com Coutinho (2021), podem carregar consigo, aproximadamente 1 Km de fiação interligando vários relês, fusíveis, lâmpadas, atuadores elétricos, alguns motores elétricos, sensores de grandezas físicas, assim como temperatura e pressão, bem como, sensores de emissão de gases, sistemas de controle de injeção de combustível, aparelhos telefônicos, dispositivos de reprodução de áudio e vídeo, sistemas de alarmes, sistemas de localização global (GPS) entre outros.

Dado a essa grande quantidade de tipos de dispositivos, Papaioannou (2005), sugere uma taxonomia interessante, e a qual teria emprego nos modelos produção atualmente empregados pela industria automotiva, qual seja:

- a) Os dispositivos tipo trem de força, onde enquadram-se os equipamentos eletrônicos que atuam no sistema de propulsão dos automóveis, assim como, motor, câmbio e diferencial e como exemplo pode-se citar as unidades de controle de injeção de combustível ou ECU;
- b) Os dispositivos de segurança, que seriam os circuitos que controlam sistemas como freio ABS e acionamento de Air-bag;
- c) Os dispositivos de conforto e conveniência, categoria onde encaixam-se os sistemas de controle de ar-condicionado, iluminação interna, entre outros;
- d) Os sistemas de informação e entretenimento, que abrangem o painel de instrumentos, reprodutores de áudio e vídeo, sistemas de telemática e navegação GPS;
- e) Os sistemas de alimentação, que controlam o uso e produção de energia elétrica, onde situam-se o alternador, bateria, motor de partida e outros;
- f) Os sistemas x-by-wire, tais como aceleradores eletrônicos.

Com o que foi exposto, percebe-se que a indústria de automotiva, desde seus primórdios, teve uma ligação estreita com a indústria eletroeletrônica. No principio, integrando aos seus produtos, componentes elétricos que poderiam melhorar aspectos básicos, como dar partida em um motor, ou iluminar um caminho escuro. Em um segundo momento, tornando o tempo de permanência no veículo mais agradável ou informativo, através da inclusão, de rádios e reprodutores de mídias. Evoluindo-se, para um maior estreitamento dessa relação, inicia-se o emprego de dispositivos eletrônicos mais complexos e especialmente projetados para veículos, com a função de melhorar o desempenho e eficiência dos motores, câmbios, suspensões, freios, sistema de informação e entretenimento que acompanham grande parte dos veículos produzidos na atualidade.

2.2 AQUISIÇÃO DE DADOS

Conforme Tocci, Widmer e Moss (2011), estaríamos constantemente lidando com quantidades, as quais, nos sistemas físicos, seriam, medidas, monitoradas, guardadas e manipuladas aritmeticamente. Desse modo, existiriam duas formas para representar-se essas medidas, o modo analógico e o modo digital.

Na representação analógica, uma quantidade pode ser representada por um indicador proporcional e continuamente variável. Como por exemplo, um voltímetro analógico apresenta em seu mostrador, através da deflexão de um ponteiro sobre uma escala pré-definida, o valor proporcional a uma tensão medida. Dessa maneira, uma quantidade elétrica, seria convertida em um movimento mecânico proporcional. "Em um sistema analógico, as quantidades físicas podem variar ao longo de uma faixa contínua de valores."(TOCCI; WIDMER; MOSS, 2011, p. 6).

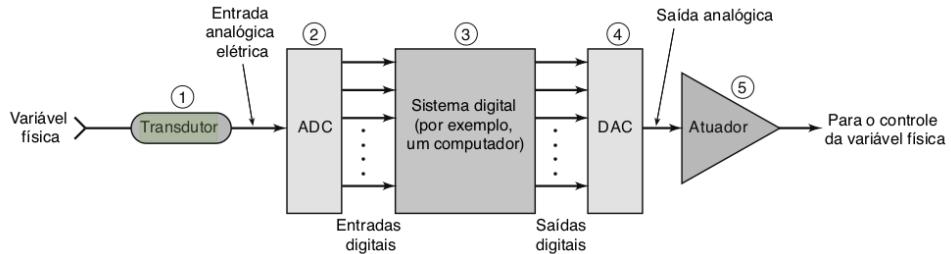
Na representação digital, as quantidades não podem assumir um número contínuo, mas um número bem definido em uma escala discreta de valores. Como por exemplo, em um multímetro digital com precisão de uma casa decimal, poderíamos obter a medida de tensão de 2,3 Volts, a qual, é apresentada no mostrador de forma não ambígua. "Na representação digital, as quantidades são representadas não por indicadores continuamente variáveis, mas por símbolos chamados dígitos."(TOCCI; WIDMER; MOSS, 2011, p. 4).

Ainda conforme esses autores, a maioria das grandezas físicas, pode ser medida em uma faixa contínua de valores, portanto, seriam naturalmente representadas por valores analógicos. Como exemplos de variáveis físicas estão a temperatura, pressão, velocidade rotacional, posição entre outros. Para converter-se, esses tipos de grandezas, em valores elétricos. Seriam necessários, componentes denominados, transdutores. "Um transdutor converte a variável física em elétrica."(TOCCI; WIDMER; MOSS, 2011, p. 629).

A variável física é geralmente uma grandeza não elétrica. Um transdutor converte a variável física em elétrica. Alguns transdutores comuns são sensores de temperatura, fotocélulas, fotodiodos, medidores de vazão, transdutores de pressão e tacômetros. A saída elétrica do transdutor é uma tensão ou corrente analógica proporcional à variável física monitorada. (TOCCI; WIDMER; MOSS, 2011, p. 629).

Além disso, para que esses sinais elétricos, obtidos pelos transdutores, possam ser processados por computadores, seria necessário a sua discretização ou digitalização através de sistemas conversores analógico-digital (ADC). "O processo pelo qual o computador adquire esses dados analógicos digitalizados é conhecido como aquisição de dados."(TOCCI; WIDMER; MOSS, 2011, p. 649). Também seria possível, o processo de conversão de valores digitais para analógicos, através de componentes denominados DAC (*Digital-to-Analog Converter*). Assim, esses dois tipos de dispositivos funcionariam como interfaces entre um sistema digital e o mundo analógico. (TOCCI; WIDMER; MOSS, 2011).

Figura 3 – Aquisição de dados.



Fonte: Tocci, Widmer e Moss (2011)

Na Figura 3, ilustra-se o processo de aquisição de dados de uma variável física, a qual, é convertida em um sinal elétrico analógico, utilizando-se um transdutor, esse sinal é discretizado e convertido para valores digitais por intermédio de um ADC. A próxima etapa, seria o tratamento lógico e aritmético desse sinal por um sistema microprocessado. Como por exemplo, deseja-se equalizar digitalmente, a voz de uma pessoa, que foi captada por um microfone. Assim, a grandeza física envolvida na entrada do sistema seria a pressão sonora emitida pela voz. O microfone funciona como o transdutor de pressão sonora para impulsos elétricos analógicos. Esse sinal, é digitalizado pelo ADC. Com o sinal digitalizado, e utilizando-se filtros digitais, pode-se ressaltar ou diminuir certas faixas de frequências, ou seja, equalizar a informação sonora recebida do ADC. Assim, para reproduzir-se esse som equalizado, poderia-se converter o som digitalizado em analógico através de um DAC (*Digital-to-Analog Converter*). Por fim, esse sinal analógico, precisaria de um atuador, como um alto-falante, que converte impulsos elétricos em ondas sonoras, as quais, podem ser percebidas por outras pessoas ou sensores de som.

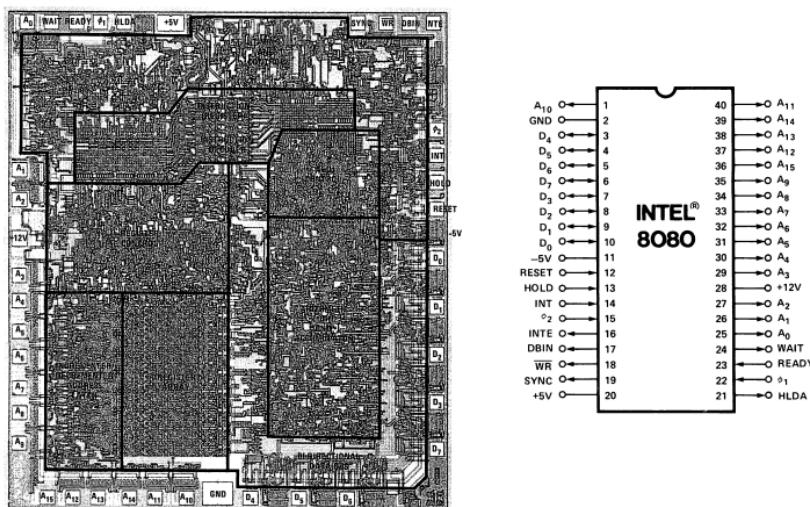
2.3 MICROPROCESSADORES E MICROCONTROLADORES

Como mencionado anteriormente, o termo microprocessador (CPU), teve origem na empresa Norte Americana Intel, no início da década de 1970, quando foi desenvolvido o circuito integrado Intel 4004. Um processador com largura do barramento de dados de 4 bits, projetado para a produção de uma calculadora eletrônica de mesa. Esse projeto, baseava-se na divisão do sistema computacional, em unidades de controle, aritmética, memória e interfaces de entrada e saída I/O. Assim, como periféricos, passou-se a produzir: uma memória de programa ROM, materializada no CI Intel 4001; uma memória de dados RAM, CI 4002; e um dispositivo de entrada e saída de dados, CI 4003; os quais, em conjunto com o processador Intel 4004, tinham maior capacidade de efetuar cálculos do que computadores tais como o ENIAC de 1946. Poucos anos após ao lançamento dessa família de componentes, a Intel passaria a produzir, uma nova geração de componentes, para compor a família do processador de 8 bits de dados, o Intel 8080. Baseado nesses

componentes, a empresa Norte Americana MITS, desenvolveu o primeiro microcomputador denominado Altair.(MEHL, 2013).

Em dezembro de 1971, a Intel lança o primeiro microprocessador de 8 bits de dados, construído com tecnologia MOS de canal N, que contava com um barramento de endereçamento de 16 bits e um barramento de controle de 10 bits, encapsulados no padrão DIP-40. A Figura 4, ilustra a distribuição dos pinos e a litografia utilizada para produzir-se o componente. (INTEL, 1975).

Figura 4 – Microprocessador 8080 - ligações dos pinos.



Fonte: Intel (1975).

Para compor essa família de componentes, a Intel também projetou e produziu, vários outros componentes periféricos, utilizados para formar, um sistema microcomputadorizado. Sejam eles: Vários tipos de memórias de programa, com diferentes capacidades e tecnologias, tais como a Intel 8708; uma variedade de memórias de dados, tais como a Intel 8101; controladores de memórias; Interfaces de I/O paralelas, tais como o Intel 8255; Interfaces de I/O seriais, tais como a Intel 8251; timers, tais como o 8253; entre outros. (INTEL, 1975).

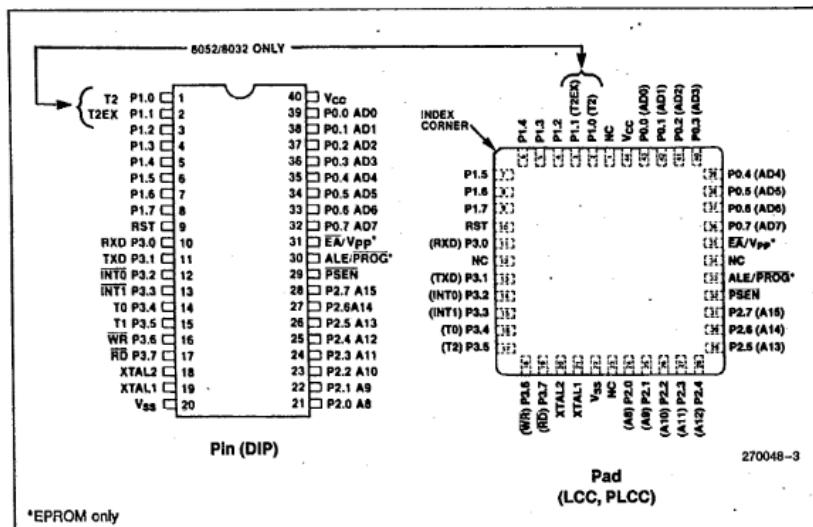
Conforme Intel (2022), a primeira geração de microcontroladores (MCU), foi MCS-48, o qual foi projetado para ter uma vida curta, e logo foi substituído pelo MCS-51, projetado para ter uma arquitetura duradoura e amplamente adaptável. Tal componente, obteve grande aceitação no mercado, e teve 100 milhões de unidades vendidas na década de 1980. Para tanto, teve aplicações das mais variadas, seja compondo sistemas de freios ABS, aplicações na aviação, instrumentos musicais ou até brinquedos.

O MCS-51 foi disponibilizado em várias versões, formando uma classe de componentes, que eram diferenciados, conforme os tipos e capacidades, dos componentes periféricos internos. Tais como, as memórias RAM e ROM utilizadas, e outras características cons-

trutivas como a velocidade de operação. Nesse contexto, o Intel 8051, seria o componente padrão, e possuía internamente, um processador de 8 bits, com capacidade de endereçamento de 64k de memória de dados e de programa. Também contava internamente, com uma memória PROM 4K bytes, 128 bytes de memória RAM, 32 portas bi-direcionais de I/O, 2 temporizadores/contadores de 16 bits, porta de comunicação UART, sistema de interrupções e oscilador de clock. (INTEL, 1994).

Na Figura 5, ilustra-se as conexões dos pinos de um CI 8051, em dois tipos de encapsulamento. Percebe-se, que diferentemente dos microprocessadores, tal como o Intel 8080, grande parte dos terminais tem uma dupla função, indicando a presença de componentes periféricos internos. Essa multiplexação de sinais em um mesmo pino, promove a redução da quantidade de pinos externos e constitui-se em uma característica que acompanha os microcontroladores, os quais, foram projetados para ser sistemas completos em um único chip (SOC). (PERIM; NASCIMENTO, 2017).

Figura 5 – MCU 8051 - Ligações dos pinos.



Assim, um microcontrolador como o 8051, poderia desempenhar uma mesma funcionalidade, para a qual seriam necessários um conjunto de componentes. Como por exemplo: um microprocessador 8080; adicionado de uma memória ROM, assim como um CI 8708; de um conjunto de memória RAM, formados por CI's 8101; um temporizador/contador, como o 8253; uma interface de I/O paralela, como o 8255, uma interface de I/O serial, como o 8251; controladores de memórias e barramentos; entre outros componentes.

Conforme Perim e Nascimento (2017), existiria uma tênue diferença entre microprocessadores e microcontroladores. Onde, microcontroladores possuem em sua composição um microprocessador e dispositivos periféricos, assim como, memórias, interfaces de I/O, conversores AD, timers e outros, integrados em um mesmo componente. Sendo assim, mi-

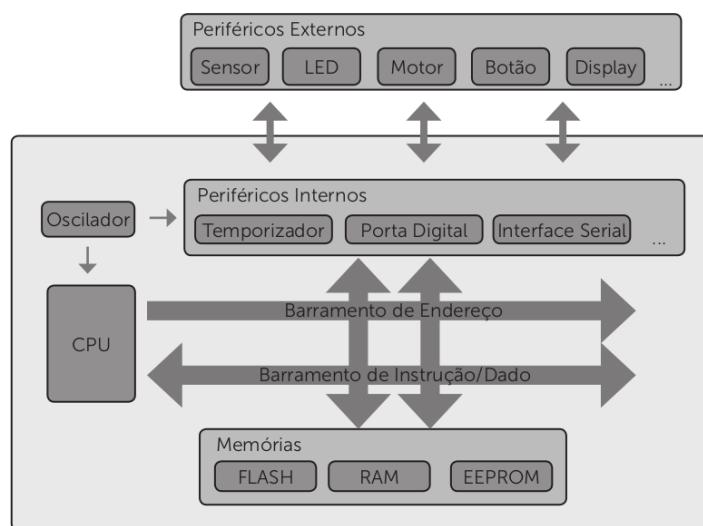
crocontroladores, configuram-se como uma evolução dos microprocessadores, com foco em tarefas específicas, e visando a simplificação de circuitos e a redução de custos. Enquanto a evolução dos microprocessadores, tomaria a direção de realizar tarefas mais generalizadas, com o aumento da largura de barramentos de endereços e dados, e maiores frequências de trabalho.

Para esses autores, componentes assim como, microprocessadores, microcontroladores e sistemas digitais sem software, seriam partes fundamentais, para produzir-se um sistema embarcado, e como exemplos, de sistemas embarcados, pode-se citar: as impressoras, aparelhos celulares, máquinas de lavar, televisores digitais, tocadores de mp3 e outros. Além disso, ressalta-se a ideia, de que atualmente, praticamente todos os sistemas embarcados seriam produzidos a partir de um microcontrolador, devido à sua praticidade, facilidade de implementação, robustez e custo.

Um sistema embarcado é um dispositivo eletrônico e computacional capaz de controlar periféricos a partir da execução de um programa, que não seja um computador. É projetado para realizar tarefas bem particulares, com grande restrição de recursos, memória e processamento. (PERIM; NASCIMENTO, 2017, p. 14).

Atualmente, existem vários modelos de MCU, o que permitiria uma escolha adequada para cada projeto, considerando-se aspectos como a quantidade de memória necessária, quais periféricos internos de entrada e saída, velocidade de trabalho, e outros elementos que compõem a arquitetura de um microcontrolador. Na Figura 6, ilustra-se a um modelo de arquitetura genérico, para um projeto de sistema embarcado, sugerido por Perim e Nascimento (2017).

Figura 6 – Arquitetura de projeto embarcado.

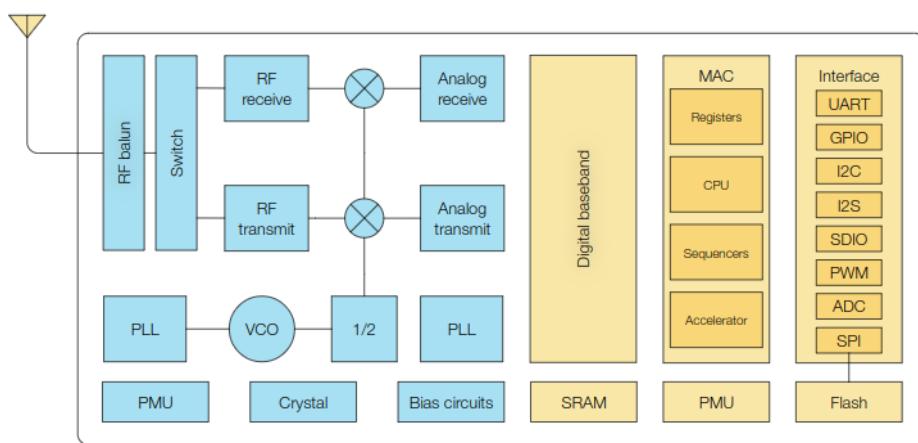


Fonte: Perim e Nascimento (2017).

Oliveira (2017), comenta, que esse processo de desenvolvimento SOC, continua até os dias atuais. Sendo que, na década de 1990, tornou-se popular, os microcontroladores da família PIC da empresa Microchip, e na última década, os microcontroladores da empresa Atmel, tornaram-se populares, com sua introdução e uso, na plataforma Arduino. Mais recentemente, a introdução de interfaces de rede, tais como, a Ethernet, WiFi e Bluetooth, aliados a introdução de tecnologias de gerencia de consumo de energia, passaram a compor a arquitetura de novos microcontroladores, tais como, os da família ESP, da empresa chinesa Espressif. Para esse autor, os microcontroladores, assim como o ESP8266, oferecem um conjunto completo de componentes para processamento. Contando com processador, memória e interfaces de entrada e saída para diversos tipos de protocolos e são oferecidos com baixo custo.

No diagrama de blocos de funções do ESP8266EX, apresentado na Figura 7, pode-se observar, localizados mais a esquerda na ilustração, os blocos encarregados das funções de rádio que são utilizados no sistema WiFi. Esse MCU, atende os protocolos, 802.11 b/g/n WLAN MAC, e opera na frequência de 2.4 GHz. Desse modo, seria possível, que esse MCU opere em redes WiFi, em modo ponto de acesso (AP), permitindo que outros dispositivos conectem-se diretamente nessa interface, ou em modo station (STA), conectando-se em algum outro AP, ou ainda, permite-se operar nos dois modos simultaneamente. Mais a direita na ilustração, apresentam-se os blocos de controle, que além da CPU e memórias, conta com uma unidade de gerenciamento de energia (PMU) e uma interface de entrada e saída de informação com suporte a oito protocolos de comunicação.

Figura 7 – Arquitetura ESP8266EX.



Fonte: Espressif (2020)

2.4 PROTOCOLOS DE COMUNICAÇÃO

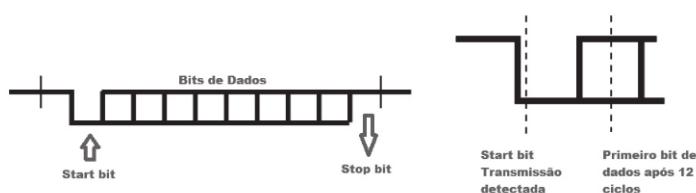
A necessidade comunicar-se a distância sempre foi uma demanda da sociedade moderna, onde, pessoas buscam interagir com outras pessoas em locais distantes. Desse modo, busca-se transmitir mensagens de maneira a economizar tempo, recursos e esforços. A partir dessa ideia, surgem nos primórdios, oferecendo-se como uma alternativa ao deslocamento com fins de comunicação, os sinais de fumaça, os sinais sonoros, a sinalização por gestos e por iluminação. Para que a comunicação decorra de forma satisfatória, as partes envolvidas deveriam ter um prévio acordo sobre o significado de cada sinalização. Nesse contexto, define-se que "um protocolo é um acordo entre as partes que se comunicam, estabelecendo como se dará a comunicação". (TANENBAUM; WETHERALL, 2011, p. 18).

Com a popularização de sistemas microprocessados, à partir da década de 1980, surge a necessidade da interligação desses tipos de dispositivos, assim, diversas especificações e protocolos de comunicação são criados. Perim e Nascimento (2017), afirmam que a grande maioria dos microcontroladores, possuem internamente, módulos periféricos que são responsáveis pela comunicação de dados e a qual normalmente é estabelecida de forma serial, de maneira síncrona ou assíncrona. Entre os protocolos de comunicação serial, mais comumente empregados nesse tipo de dispositivo, mencionam-se, o protocolo UART, o SPI e o I2C.

2.4.1 Transmissor/receptor universal assíncrono (UART)

Conforme Perim e Nascimento (2017), em uma transmissão assíncrona, não existe a necessidade de um sinal específico para sincronismo, tal como, um sinal de clock. Para tanto, as partes envolvidas nesse tipo de troca de dados, devem estar previamente configuradas, especificando-se alguns parâmetros, tais como, a velocidade em que se estabelece o enlace, ou seja, quantos bits serão transmitidos ou recebidos em um segundo (bps), o tamanho ou quantidade de informação útil que se está enviando e alguns bits de controle, para garantir a entrega adequada da informação.

Figura 8 – Comunicação UART.



Fonte: Perim e Nascimento (2017).

Para Oliveira (2017), a interface UART, seria a forma mais simples de estabelecer

comunicação digital, necessitando-se apenas três conexões de sinais, uma referência GND, uma via de transmissão de dados (Tx) e outra para recepção de dados (Rx). Em casos onde a comunicação se dá unidirecionalmente, onde somente um dispositivo transmite e o outro recebe informações, como no caso de alguns sensores, a necessidade de conexões reduz-se para apenas duas vias.

A forma de comunicação mais simples e comum presente em quase todos os microcontroladores é a UART (Universal Asynchronous Receiver Transmitter), usada nas portas RS-232 (com tensão de 12 volts), muito comuns nos PCs antigos. É assíncrona e utiliza apenas dois pinos para comunicação, TX e RX, além da alimentação. No caso de sensores e outros dispositivos que não recebem nenhuma informação, apenas enviam, é possível usar somente um pino TX. (OLIVEIRA, 2017, p. 152).

Ainda conforme esse autor, esse tipo comunicação, seria normalmente utilizada da depuração dos programas dos microcontroladores, onde, esses programas enviam mensagens através da UART, as quais podem ser recebidas em programas próprios para computadores, assim como o Monitor Serial do Arduino, através de interfaces USB.

2.4.2 Barramento serial universal (USB)

De acordo com Perim e Nascimento (2017), o protocolo USB, seria amplamente utilizado, na maioria dos computadores e dispositivos eletrônicos mais modernos, e possibilitaria uma comunicação de dados de forma serial e com altas taxas de transferências. Ideali (2021), concorda com esses autores, ressaltando que a padronização USB, seria um dos mais bem sucedidos projetos de interconexão entre computadores e periféricos, sendo que, na especificação USB 1.0, proposta em 1996, havia uma capacidade de transferência de 12 Mbps. Na nova versão, a USB 2.0, disponibilizada em 2001, era compatível com os dispositivos da versão anterior, e passou a oferecer taxas de transferência de até 480 Mbps. Em 2008, o estabeleceu-se o padrão USB 3.0, com taxas de 4,8 Gbps, também compatível com os dispositivos fabricados para as especificações anteriores e otimizada para oferecer um baixo consumo de energia.

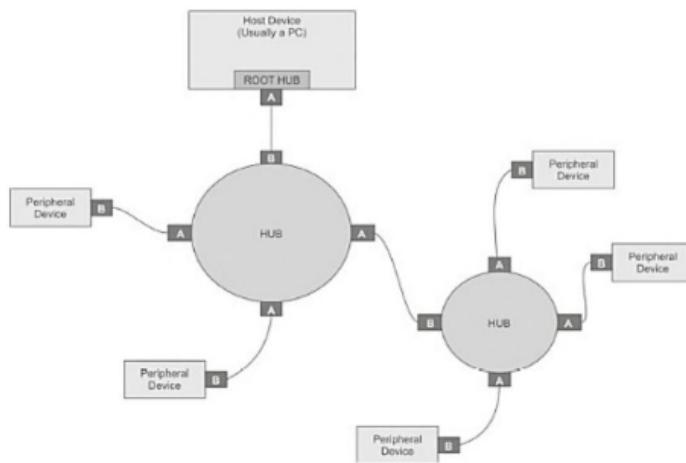
Conforme Ideali (2021), o USB (*Universal Serial Bus*), seria um conjunto especificações, originalmente desenvolvido em 1995 por um grupo de empresas, tais como, Intel, Compac, Microsoft, Digital, IBM e Northen Telecom, com o objetivo de definir um barramento de expansão para adicionar-se periféricos a um sistema microprocessado. Esse padrão, especificaria características mecânicas, assim como, o formato de conectores, características elétricas, assim como tensões e correntes suportadas e quantidades de condutores utilizados, bem como, a especificação das características e lógica de funcionamento.

Assim, para implementar-se o barramento USB, seria empregada uma arquitetura mestre-escravo, onde o mestre, normalmente denominado Host, controla todas as atividades de comunicação. Os escravos do barramento, normalmente denominados dispositivos USB, respondem aos comandos do Host. Com a finalidade de expandir o número de

portas USB disponíveis, são especificados os dispositivos hub USB, os quais, possuem uma porta *upstream*, utilizada para conectar-se ao Host ou a outro hub USB de uma camada superior e portas *downstream*, utilizadas para conexão de dispositivos USB. Esse componente, possibilitaria a implementação de uma topologia estrela com camadas, onde até 127 dispositivos, poderiam ser ligados a um Host. (IDEALI, 2021).

Na Figura 9, ilustra-se a topologia empregada nas interfaces USB, que consistiria, em redes com topologia estrela, onde todos os integrantes se conectam a um dispositivo centralizador, com a possibilidades de estabelecer-se camadas através de portas *upstream*.

Figura 9 – Topologia de conexões USB.



Fonte: Ideali (2021)

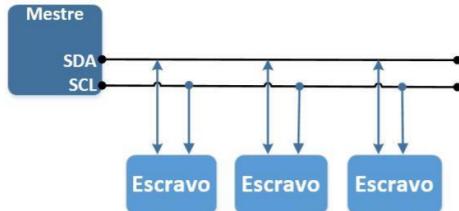
Tanenbaum e Wetherall (2011), concordam com esses autores, e complementam, informando que para esse tipo de comunicação, utiliza-se uma técnica de modulação de sinais denominada NRZI (*Non Return to Zero Inverted*), com a função de aumentar-se a taxa de transferência.

2.4.3 Circuito inter integrado (I2C)

Conforme Oliveira (2017), o *Inter-Integrated Circuit* (I2C), seria um protocolo de comunicação serial síncrono, com baixa taxa de transferência. Foi inicialmente desenvolvido pela empresa Philips e posteriormente pela empresa Intel, e que teria como objetivo, promover a comunicação entre, microprocessadores e microcontroladores, e seus dispositivos periféricos. Essa interface seria implementada utilizando-se duas vias: uma via denominada serial data (SDA), responsável por transmitir e receber dados e outra via denominada serial clock (SCL), responsável pelo sincronismo da comunicação. Além disso, esse protocolo utilizaria uma arquitetura mestre-escravo, onde o MCU ou a CPU assumem o papel de mestre do barramento, controlando o processo de comunicação. Os dispositivos

periféricos assumem o papel de escravo do barramento, fornecendo as informações requisitadas pelo mestre. Na Figura 10, apresenta-se o diagrama de ligações entre um mestre e seus escravos em um barramento I2C.

Figura 10 – Conexões I2C.

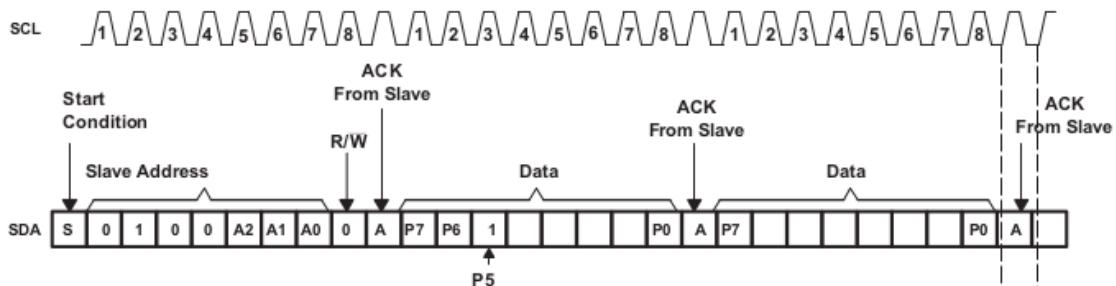


Fonte: Mário Jorge Garcia de Oliveira (2017).

Como vantagem, para utilização do protocolo I2C, estaria a capacidade de coexistência de vários dispositivos em um mesmo barramento, uma vez que os escravos são diferenciados pelo endereço que assumem. Como exemplos de dispositivos I2C, pode-se mencionar, algumas interfaces de cristal líquido, relógios de tempo real (RTC), entre outros variados sensores e módulos de sensores.

Na Figura 11, apresenta-se um diagrama de temporização para uma operação de escrita em um periférico I2C. Percebe-se, que o primeiro byte útil enviado pelo mestre, refere-se ao endereço do dispositivo, com sete bits, mais um bit que corresponde ao tipo de operação. Se a operação for de leitura (1), ou se for de escrita (0). Se existir algum periférico com esse endereço, uma mensagem ACK será enviada pelo escravo. Com a confirmação ACK, o mestre envia o próximo byte. Nesse exemplo, o segundo byte enviado pelo mestre, seria o endereço interno do periférico. Após receber um novo ACK do escravo, o mestre envia o último byte, que representa os dados que serão gravados nesse dispositivo I2C.

Figura 11 – Protocolo I2C.

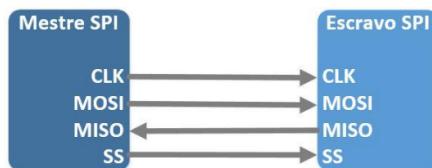


Fonte: Texas-Instruments (2015)

2.4.4 Interface periférica serial (SPI)

Para Mário Jorge Garcia de Oliveira (2017), a interface serial periférica (SPI), é um protocolo de comunicação, serial, síncrono e bi-direcional, envolvendo um microcontrolador e um periférico, interligados em uma arquitetura mestre-escravo, composta por quatro sinais. O sinal de clock (CLK), gerado pelo mestre e responsável pelo sincronismo da comunicação; o sinal MOSI (*Master Out Slave In*), representam os dados que saem do mestre em direção ao escravo; o sinal MISO (*Master In Slave Out*), representam os dados que vão do periférico escravo em direção ao mestre; e por fim o sinal SS (*Slave Select*), que define qual dispositivo escravo está habilitado para comunicar-se. Na Figura 12, ilustra-se como ocorre a interligação entre um periférico SPI e seu Mestre.

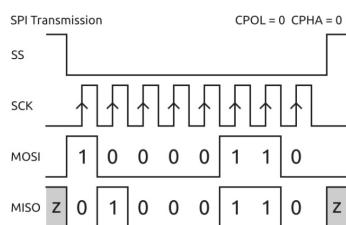
Figura 12 – Conexões SPI.



Fonte: Mário Jorge Garcia de Oliveira (2017).

Nesse tipo de transmissão, o clock sinaliza o fim do primeiro bit e o início do próximo. Quando a linha de clock muda seu valor (0 para 1 ou vice-versa), um bit é transferido do mestre para o escravo (MOSI) e um bit é transferido do escravo para o mestre (MISO). Na Figura 13, apresenta-se um exemplo, da transferência de 1 byte com os valores 100001100, considerando-se o valor de borda de subida do sinal de clock, utilizando um periférico SPI. A transmissão inicia com o mestre setando uma transição do pino SS, de nível 1 para nível lógico 0. Na próxima borda de subida, do sinal de clock SCK, um bit é transferido do mestre para o escravo através do pino MOSI, que nesse exemplo, seria a informação com valor 1, ou verdadeiro. Simultaneamente, 1 bit é transferido do escravo para o mestre, através do pino MISO, que nesse caso seria o valor 0 ou falso.

Figura 13 – Protocolo SPI.

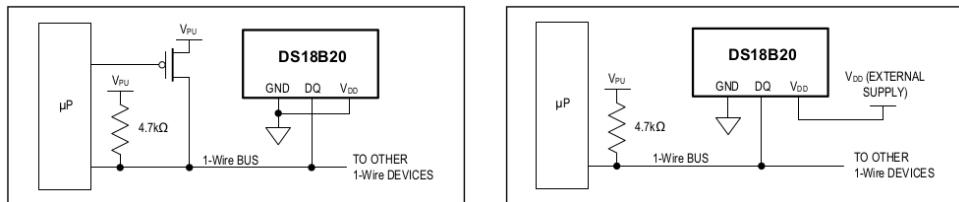


Fonte: Oliveira (2017).

2.4.5 1-Wire

O *one wire* ou 1-Wire, Maxim-Integrated (2019), é um protocolo de comunicação serial, que utiliza a arquitetura mestre-escravo, onde um mestre controla um ou mais escravos. Para implementar-se esse tipo comunicação, seria necessário uma via para trafegar os dados de forma bi-direcional, e um condutor para referência GND. Nesse tipo de configuração, a alimentação do dispositivo escravo, pode ser feita através da linha de dados, através da ligação de um resistor *pull-up* de aproximadamente 5 K ohms, utilizando-se uma técnica de tensão parasita. Porém, por conveniência, pode-se utilizar uma via extra, para fornecer alimentação para o circuito escravo. Com isso, pode-se obter taxas de transferência de dados mais elevadas. Para conseguir localizar os dispositivos no barramento 1-Wire, cada componente escravo tem um numero de registro único (ROMID), formado com 64 bits. O processo de comunicação inicia-se com o mestre enviando um sinal reset, fazendo a via de dados permanecer com nível lógico zero por pelo menos 480 micro segundos. Quando a linha de dados passa para nível 1, o mestre aguarda o pulso de resposta de pelo menos um escravo do barramento, indicando que esse está presente e pronto para receber instruções.

Figura 14 – Conexões 1-Wire.



Fonte: Maxim-Integrated (2019)

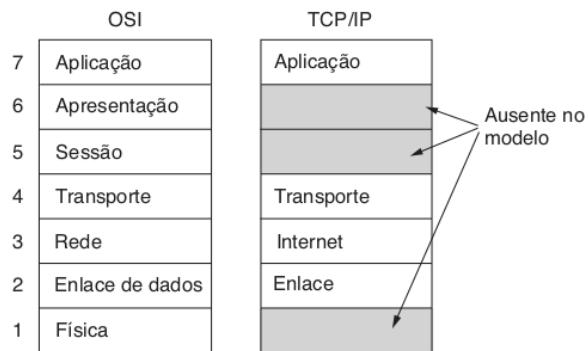
Na Figura 14, ilustra-se o diagrama de ligações dos dispositivos 1-wire, com e sem a utilização de alimentação parasita. Para o dispositivo 1-wire DS18b20, conforme informações do fabricante, a alimentação deve ser aplicada com no mínimo 3 volts e no máximo com 5,5 volts e esse componente consome aproximadamente 1 mA durante o funcionamento.

2.4.6 Protocolo de Controle de transmissão e internet (TCP/IP)

Para Tanenbaum e Wetherall (2011), o TCP/IP seria um modelo de referência de arquitetura de redes de computadores, originalmente implementada como uma rede de pesquisa, patrocinada pelo Departamento de Defesa dos Estados Unidos, denominada ARPANET, a qual foi sucedida pela Internet mundial. Existiria também, o modelo de referência *Open Systems Interconnection* (OSI), o qual teria sido desenvolvido pela *International Standards Organization* (ISO). Esse autor ressalta, que os protocolos do modelo

TCP/IP seriam os mais utilizados, sendo que, o Protocolo de Controle de Transmissão (TCP) e o Protocolo de Internet (IP), seriam os mais importantes e os quais emprestam o nome a esse modelo de arquitetura de rede.

Figura 15 – Modelos OSI e TCP/IP.



Fonte: Tanenbaum e Wetherall (2011)

Esses dois modelos de referência, seriam estabelecidos em camadas, formando uma pilha de protocolos independentes e com funções bem definidas. Na Figura 15, pode-se observar a estrutura de camadas dos dois referidos modelos, percebe-se que no modelo OSI existiriam mais camadas. Algumas funções, presentes em camadas específicas do modelo OSI, assim como a camada de apresentação, sessão e física, são aglutinadas em outra camadas do modelo TPC/IP.

Na camada de aplicação, situam-se os protocolos, assim como, o HTTP (*Hypertext Transfer Protocol*), utilizado para transferência de textos, o FTP (*File Transfer Protocol*), utilizado para transferência de arquivos e o SMTP (*Simple Mail Transfer Protocol*), utilizado para email. Na camada de transporte, aplicam-se os protocolos, assim como, o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*). Na camada Internet, localizam-se os protocolos IP (*Internet Protocol*) e ICMP (*Internet Control Message Protocol*). Na camada de enlace, implementam-se os padrões de conexão de redes, assim como o Ethernet 802.3 e o WiFi 802.11.(TANENBAUM; WETHERALL, 2011).

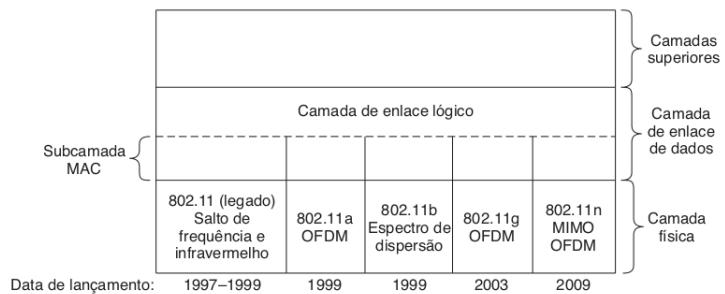
2.4.7 Redes WiFi 802.11

Em 1997 o IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos), propôs as primeiras normas para o padrão de comunicação 802.11, o qual faz parte da especificação IEEE 802, que oferece recomendações para padronização técnica das redes locais (LAN). Assim, seria sugerido um conjunto de protocolos, para a utilização do canal de comunicação, a fim de implementar-se uma rede local que opera com a camada física via radio frequência ou uma WLAN (*Wireless Local Area Network*). Para tanto utilizam-se rádios de curto

alcance para transmitir sinais nas bandas de frequência não licenciadas de 2,4 GHz ou 5 GHz. (TANENBAUM; WETHERALL, 2011).

A Figura 16, ilustra a localização dos protocolos 802.11 no modelo OSI, bem como uma linha de tempo, que permite visualizar o momento em que surgem algumas das variações desse tipo de rede.

Figura 16 – Especificações 802.11.



Fonte: Tanenbaum e Wetherall (2011)

Na subcamada MAC (*Medium Access Control*), determina-se qual participante terá a oportunidade de transmitir, uma vez que nesse tipo de rede, o canal de comunicação é compartilhado. Desse modo, na medida em que aumenta-se o número de participantes ativos simultaneamente, reduz-se as taxas de transferências experimentadas por cada usuário. Para regular como serão transmitidos os quadros de dados, utiliza-se o protocolo CSMA/CA (*Carrier Sense Multiple Access / Collision Avoidance*), o qual oferece métodos para acessar e transmitir dados no canal de radio da camada física. A subcamada LLC (*Logical Link Control*), teria como função, ocultar as diferenças entre as diversas variações das redes 802, assim, os pacotes de dados que saem ou chegam a camada de rede tem o mesmo conteúdo. (TANENBAUM; WETHERALL, 2011).

Ainda conforme esse autor, existiriam duas topologias de rede que poderiam ser estabelecidas utilizando-se o padrão 802.11. Nas redes *ad hoc*, uma variedade de estações, assim como *laptops*, poderiam conectar-se uns ao outros sem a presença de pontos de acesso. Nas redes do modo Infraestrutura, cada estação está associada a um ponto de acesso (AP), o qual pode estar conectado a outras redes, assim como a Internet e por esse motivo seriam muito mais utilizados.

Entre os serviços oferecidos por esse tipo de rede, estão os de associação, desassociação e reassociação. Para associar-se a um AP, pode-se utilizar ou não, um método de autenticação. Para um AP configurado como rede aberta, qualquer usuário está apto a solicitar uma associação e transmitir ou receber pacotes de dados. Também pode-se utilizar um esquema de autenticação com chave, tais como o WEP (*Wired Equivalent Privacy*), o WPA (*WiFi Protected Access*), ou WPA2. (TANENBAUM; WETHERALL, 2011).

3 PROJETO E DESENVOLVIMENTO

Após observar-se o veículo, no qual aplicaria-se a solução embarcada, definiu-se os requisitos funcionais e não funcionais deste trabalho. Também elencou-se, quais seriam obrigatórios e quais seriam desejáveis. Essa relação de requisitos é apresentada no Quadro 1, onde RFO refere-se aos requisitos funcionais obrigatórios e RFD os requisitos funcionais desejáveis, no que refere-se as características de operação do sistema embarcado. RNFO, representa os requisitos não funcionais obrigatórios e RNFD os requisitos não funcionais desejáveis. Ambos referem-se ao meios necessários para atender-se os requisitos funcionais.

Quadro 1 – Requisitos funcionais e não funcionais.

RFO e RFD	Requisitos funcionais obrigatórios e desejáveis.
RFO-1	Medição de temperatura do fluido de arrefecimento do motor;
RFO-2	Alerta de temperatura do fluido de arrefecimento do motor;
RFO-3	Alerta de nível do fluido de arrefecimento no radiador;
RFO-4	Medição de nível de combustível;
RFO-5	Alerta de nível de combustível;
RFO-6	Medição de temperatura da geladeira do veículo;
RFO-7	Acelerar o processo de refrigeração da geladeira do veículo;
RFO-8	Medição de temperatura interna da cabine;
RFO-9	Fornecer um sistema digital de controle da temperatura interna da cabine;
RFO-10	Fornecer uma interface para leitura dos dados coletados e envio de informações para os sistemas de controle;
RFD-1	Medição de pressão atmosférica e altitude;
RFD-2	Fornecer um alerta sonoro de falhas;
RNFO e RNFD	Requisitos não funcionais obrigatórios e desejáveis.
RNFO-1	Fornecer um sistema de armazenamento para os dados coletados nas medições;
RNFO-2	Fornecer um sistema de coleta dos dados armazenados através de um PC ou rede;
RNFO-3	Fornecer Medida temperatura do dissipador de calor da pastilha Peltier;
RNFO-4	Fornecimento de uma interface atuadora de potência para controle de motores e trocadores de calor;
RNFO-5	Projeto e construção de protótipos de <i>software</i> e <i>hardware</i> ;
RNFD-1	Fixação dos sensores e atuadores e passagem de fiação no veículo.
RNFD-2	Acondicionamento do sistema no protótipo.

Fonte: Autor.

3.1 PROTÓTIPO *HARDWARE*

No projeto e desenvolvimento do *hardware*, desenvolve-se um protótipo funcional, baseado em kits de desenvolvimento, encontrados no varejo do mercado de componentes eletrônicos e segue-se as seguintes etapas:

- a) Pesquisa e Escolha do sistema microcontrolador;
- b) Pesquisa e escolha dos sensores utilizados no projeto;
- c) Pesquisa e escolha das interfaces de aquisição de sinais e atuadores;
- d) Pesquisa e escolha do sistema de armazenamento de dados;
- e) Aquisição de matéria-prima e testes de bancada;
- f) Elaboração do diagrama elétrico das ligações necessárias;
- g) Elaboração da placa de circuito impresso para a integração do *hardware*;
- h) Construção da primeira versão do protótipo projetado;

3.1.1 Microcontrolador

O microcontrolador escolhido para este projeto foi o ESP8266, o qual, conforme o fabricante, Espressif (2022), seria um componente, que integra, um processador Cadence-Tensilica L106 de 32 bits, que utiliza tecnologia RISC com velocidade de até 160 MHz, associado com várias interfaces, assim como: interface periférica padrão digital, chaves de antena, adaptador de RF, amplificador de potência, amplificador de recepção de baixo ruído, filtros e gerenciamento de energia.

Algumas versões desse chip são disponibilizados, dependendo-se a quantidade de pinos I/O que deseja-se. Para elaborar-se este projeto, utilizou-se o ESP8266EX de modelo ESP12E, qual seria, uma das configurações disponibilizadas pelo fabricante, e que conta com uma antena integrada no circuito impresso, aliados a 22 pinos de conexão externa. Além disso, existe também, alguns tipos de kit de desenvolvimento que utilizam esse módulo. Sendo assim, foi utilizado, o kit de desenvolvimento conhecido como NodeMCU V1.0, o qual, fornece alimentação ao ESP12E, disponibiliza uma interface de comunicação, que converte o padrão USB para UART, que é utilizada para carregar os programas e firmwares, além de fornecer acesso facilitado aos pinos de I/O do módulo ESP12E.

Dentre os motivos da escolha desse kit de desenvolvimento, pode-se mencionar, a existência de um *software development kit* (SDK) disponibilizado pelo fabricante e também, a compatibilidade com plataforma Arduino, onde, para esse ambiente de desenvolvimento, existe um grande número de desenvolvedores de *software*, os quais, disponibilizam códigos que podem ser reaproveitados neste projeto. Além disso, existe uma grande variedade de módulos disponíveis no mercado, tais como, sensores de pressão atmosférica, acelerômetros,

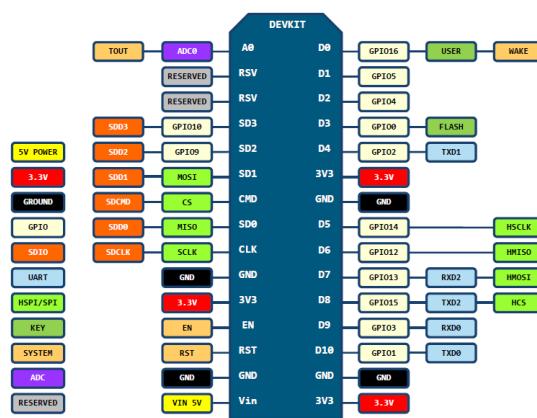
Figura 17 – Kit de desenvolvimento NodeMCU V1.0.



Fonte: AiThinker (2022).

gps, sensores de temperatura, entre outros, que podem ser facilmente integrados através de biblioteca de funções para Arduino ou utilizando-se os SDK's fornecidos pela Espressif. Pode-se elencar, como um outro motivador para o uso desse microcontrolador, a existência integrada, de um sistema de rádio frequência padrão WiFi 802.11, com alcance suficiente para a finalidade deste projeto, e que também, permite a operação como servidor de requisições *web*. Outro fator determinante para a escolha, seria a quantidades de pinos de IO, que é suficiente ou podem facilmente ser expandidas através de outros dispositivos, integrados através de protocolos de comunicação, tais como, o SPI, I²C, RS232 ou 1-Wire. Além disso, Também possui um conversor AD integrado que possibilita a aquisição de sinais analógicos. Somando-se ao que foi dito, outra vantagem encontrada, seria que, essas placas de desenvolvimento são facilmente encontradas no mercado e tem baixo custo de aquisição. Na Figura 17, apresenta-se a imagem do kit de desenvolvimento utilizado.

Figura 18 – Descrição pinos NodeMCU.

PIN DEFINITION

Fonte: NodeMCU (2022).

Na Figura 18, ilustra-se, a descrição dos pinos do kit de desenvolvimento, fornecendo

a descrição e a identificação de como são chamados os pinos de *Input/Output* (I/O). Como por exemplo, na plataforma Arduino, normalmente, pode-se designar uma entrada ou saída de sinal através dos apelidos, D0 à D9 e uma leitura de sinal analógico através de A0, mas também pode-se acessar essas portas através do número da *General Purpose Input Output* (GPIO). Para o SDK Espressif testado, deve-se utilizar o número da GPIO ou declarar-se algum *alias*. Além disso, outros documentos, firmwares, esquemas elétricos, ferramentas de software e demais informações sobre esse módulo de desenvolvimento podem ser encontrados no github e são disponibilizados gratuitamente.(NODEMCU, 2022).

3.1.2 Periféricos externos - sensores, módulos e atuadores

De acordo com Perim e Nascimento (2017), os periféricos internos de um microcontrolador, seriam os componentes com funções específicas, que formam a arquitetura de um microcontrolador. Assim, seriam periféricos internos, as memórias voláteis e permanentes, conversores AD, portas de I/O, portas de comunicação e outros, dependendo-se de cada tipo de microcontrolador. Para esse autor, muitas vezes seriam necessários, para implementar uma determinada solução, a incorporação de periféricos externos ao chip do microcontrolador. "Até mesmo os projetos mais primários precisarão de periféricos externos".(PERIM; NASCIMENTO, 2017, p.20).

Conforme Papaioannou (2005), sensores seriam componentes, que convertem grandezas físicas ou químicas em um sinal elétrico. Esse autor, sugere uma taxonomia para esse tipo de componente, onde, haveriam duas classes principais, diferenciadas pelo aspecto construtivo do componente. Baseado nesse critério, existiriam os sensores semicondutores ou ativos e os sensores passivos. Porém, uma mesma medida, assim como a temperatura, pode ser obtida utilizando-se, tanto sensores ativos, quanto passivos. Em vista disso, seria mais conveniente, classificar esses dispositivos, de acordo com sua função. Para esse autor, os principais sensores encontrados em veículos, e classificados por sua função, seriam: Sensores de posição, de velocidade, de rotação, aceleração, vibração, sensores de força e torque, sensores de vazão, de concentração, de temperatura, sensor de sujeira, de chuva e por fim, sensor de imagem.

Muitas vezes, em projetos embarcados, deseja-se acionar cargas incompatíveis com os níveis de tensão e corrente disponibilizados nos microcontroladores e seus periféricos. Para adaptar os sinais de controle das portas I/O, aos níveis de tensão e corrente demandados por motores, compressores, lâmpadas e outros, faz-se necessária a utilização de componentes, assim como os triacs, transistores ou dispositivos atuadores, tais como os relés.

Entre os módulos atuadores, os mais comuns são os módulos de relé, que podem aparecer em diversas configurações de número de canais, de um a oito canais. Esses módulos são ótimos para controlar diversos tipos de dispositivos com função de liga/desliga, como lâmpadas, motores e aparelhos eletrônicos.(OLIVEIRA, 2017, p. 49).

3.1.2.1 Sensor de temperatura DS18B20

Para as medidas de temperatura externa do automóvel, da pastilha Peltier, nos lados quente e frio e da temperatura do fluido de arrefecimento do motor, utiliza-se sensores modelo DS18B20. Entre as vantagens encontradas nesse sensor, pode-se mencionar, que conforme Maxim-Integrated (2019), esse dispositivo é digital, e sendo assim, minimiza a interferência de ruído na leitura do sinal de temperatura. Dessa forma, o dado de temperatura seria coletado e processado localmente e transmitido via barramento de comunicação digital. Para fazer isso, o fabricante utilizou o protocolo 1-Wire, que permite a um microcontrolador, consultar a temperatura de vários sensores desse tipo, utilizando-se apenas um pino de I/O. Isso torna-se possível, porque cada sensor DS18B20 possui um endereço serial exclusivo de 64 bits, o que possibilita, conectar vários sensores em um mesmo barramento, utilizando-se apenas, uma única porta do microcontrolador, para obter-se os valores de temperatura de cada sensor separadamente. Além disso, esse sensor, possui um encapsulamento em aço inoxidável, que permite operar em ambientes secos ou submerso em líquidos, podendo-se medir temperaturas entre -55 e 125 graus centígrados, o que seria suficiente para este projeto.

Figura 19 – Sensor de temperatura DS18B20 a prova d’água.



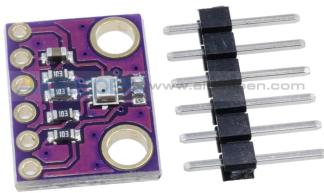
Fonte: Aideepen (2022a).

3.1.2.2 Sensor de temperatura e pressão BMP280

Para obter-se as medidas de temperatura e pressão atmosférica, e com isso, estimar-se a altitude em que o sistema se encontra, optou-se por utilizar, o módulo fornecido por Aideepen (2022b), o qual faz uso do sensor BMP280. De acordo com Bosh (2021), esse seria, um sensor digital de precisão e configura-se como uma solução de baixo custo para medir pressão barométrica e temperatura. No módulo utilizado, esse sensor é montado sob uma placa de circuito impresso, que conta com os circuitos auxiliares, para a utilização juntamente com o protocolo I2C no endereço 0x76h, que é utilizado para transmitir os

valores lidos para o microcontrolador. Com a relação existente, entre a pressão atmosférica e a altitude, pode-se utilizar esse dispositivo, como altímetro. A faixa de leitura de pressão desse sensor, vai de 300 hPa até 1100 hPa que corresponderia a altitudes de -500 à 9000 m, relativos ao nível do mar, com resolução de medida 0,25 m. A faixa de medida da temperatura, vai de -55 até 85 graus centígrados com precisão de 0,01 graus centígrados, conforme dados fornecidos pelo fabricante.

Figura 20 – Módulo sensor de temperatura e pressão BMP280.



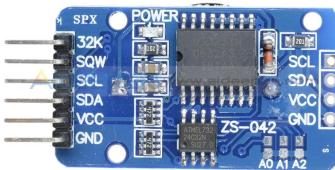
Fonte: Aideepen (2022b).

Na Figura 20, percebe-se que o módulo possui um circuito simples, cuja principal vantagem, seria facilitar ou possibilitar o acesso aos pinos do sensor, bem como, permitir a sua rápida inserção ou retirada do restante do circuito para execução de testes. Desse modo, considerando-se que as características desse módulo atende os requisitos para medir a temperatura interna da cabine do veículo e que apresenta vantagens, assim como, o baixo custo e facilidade de aquisição, ser um dispositivo popular que conta com bibliotecas para a plataforma de *software* escolhida, e o fato de poder usar o protocolo de comunicação I2C, que permite adicionar outros sensores sem a necessidade de consumir novas portas de I/O, adicionou-se esse sensor ao projeto.

3.1.2.3 Módulo ZS-042 RTC

O módulo ZS-042 é composto por dois circuitos integrados, o DS3231 que é um dispositivo Real Time Clock (RTC), um relógio de tempo real, que permite a conexão de uma bateria externa, que mantém o relógio do sistema funcionando, mesmo quando a energia principal do dispositivo é interrompida. O RTC tem a capacidade de fornecer dados como segundos, minutos, horas, dia, data, mês e ano. Esses são transferidos em série para o microcontrolador, através de um barramento bidirecional I2C. Tais informações são úteis para gerar um indexador das demais medidas realizadas. Além disso, esse componente possui um sensor de temperatura que pode medir entre 0 e 85 graus centígrados que irá monitorar a temperatura de operação do protótipo. Também integrado a esse módulo, existe uma memória EEPROM 24c32 com 32K bits de capacidade que fornece opção para armazenamento de dados de forma permanente.(AIDEEPEN, 2022c).

Figura 21 – Módulo ZS-042 RTC.



Fonte: Aideepen (2022c).

Na Figura 21, nota-se o circuito integrado DS3231, com 16 pinos, que conforme Maxim-Integrated (2015), pode funcionar com tensões de 2,3 Volts até 5,5 Volts, permitindo o seu uso em microcontroladores de 3,3 Volts, como seria, no caso deste projeto. Para acessar esse componente, utiliza-se o protocolo I2C, configurado para o endereço hexadecimal 68h. Após o endereçamento I2C, deve-se enviar ao barramento, um bit indicando a direção da comunicação, isto é, se será uma operação de leitura ou de escrita no componente. Sequencialmente, deve ser enviado o endereço hexadecimal, variando-se entre 00h e 12h, onde, para cada endereço, está associado um registrador de 8 bits, que armazena uma das informações contidas nesse dispositivo. O endereço 00h armazena a informação segundos do relógio. O endereço 01h armazena a informação hora e assim por diante. O CI menor, com 8 pinos, constitui-se da memória flash 24c32, que conforme (MICROSHIP, 2004), possui 4.000 endereços com 8 bits de dados cada e os quais podem ser acessados através da interface I2C no endereço configurado através dos *jumpers* A0, A1 e A2, que neste projeto corresponde ao endereço hexadecimal 57h.

3.1.2.4 Sensor de nível de fluído de arrefecimento ES7510

Para monitorar o nível de fluído de arrefecimento, contido no radiador do motor, optou-se por utilizar, uma chave magnética a prova d'água. Na Figura 22, pode-se observar os aspectos construtivos desse sensor, onde, um flutuador magnético, tem a capacidade deslizar por um eixo central.

Quando o nível de fluído está normal, a chave mantém-se aberta, implicando em um sinal com nível lógico verdadeiro no pino D0 do kit de desenvolvimento NodeMCU, através de um resistor externo de *pull-up* para a tensão de 3,3 Volts. Quando o nível do fluído de arrefecimento baixar, o flutuador desce até o batente inferior, acionando a chave, o que implica na condução de um sinal lógico falso para porta I/O responsável pelo monitoramento desse dispositivo.

Figura 22 – Sensor de nível de fluído de arrefecimento- ES7510.

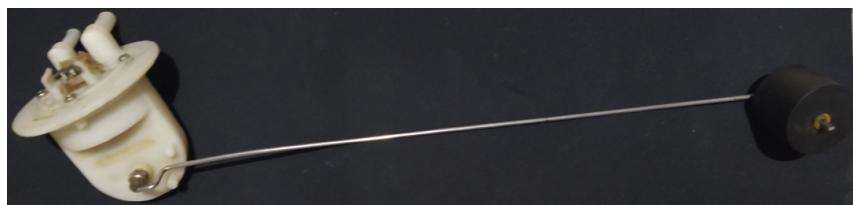


Fonte: Elecall (2018).

3.1.2.5 Sensor de combustível

A medição do volume de combustível do veículo de teste, é feita através de um medidor específico, que consiste de uma boia, ligada ao eixo de um potenciômetro através de uma haste e instalada dentro de um tanque com volume nominal de 110 litros. Observando-se a Figura 23, a esquerda, na parte inferior, tem-se o potenciômetro com a haste ligada ao eixo. Na parte superior situa-se a entrada e saída de combustível e os terminais elétricos desse componente, os quais ficam para fora do tanque. À direita situa-se o flutuador, na ponta da haste, que gera o movimento do eixo conforme o nível de fluído no tanque.

Figura 23 – Sensor de nível de combustível.



Fonte: Autor.

Para avaliar-se o funcionamento desse sistema, realizou-se um ensaio, no qual, com o tanque vazio, adicionou-se combustível de dois em dois litros e mediu-se a resistência elétrica fornecida na saída do dispositivo. Como resultado, para cada volume adicionado, obteve-se os dados apresentados na Tabela 1.

Observando-se os dados, percebe-se uma relação inversa entre o volume de combustível e a resistência medida, onde maiores volumes correspondem a menores resistividades. Também percebe-se, que a variação não apresenta comportamento linear em toda faixa de valores, aparentemente, por volta de 30 litros e 60 litros existe uma mudança da taxa de crescimento dos valores de resistência. Ligado em serie com o medidor de combustível, no painel do veículo, existe um galvanômetro ligado a um regulador de tensão de 5 Volts. Desse modo, quanto mais combustível, menor a resistividade do sensor e maior será a corrente elétrica que fluirá no circuito, sendo assim, o ponteiro do instrumento deflexiona no sentido horário. Na Figura 24, ilustra-se a relação, entre a resistividade do sensor e a

Tabela 1 – Aferição do medidor de combustível.

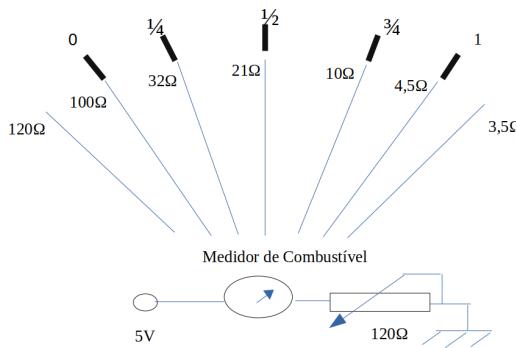
Litros	Ohms	Litros	Ohms	Litros	Ohms	Litros	Ohms
2	118	32	40,0	62	17,0	92	7,2
4	118	34	38,0	64	14,0	94	7,2
6	118	36	36,0	66	13,0	96	6,6
8	112	38	34,0	68	13,0	98	6,0
10	105	40	34,0	70	13,0	100	5,4
12	98	42	32,0	72	12,0	102	4,9
14	98	44	29,0	74	11,5	104	4,8
16	89	46	29,0	76	11,5	106	4,5
18	89	48	27,0	78	10,9	108	4,3
20	78	50	25,0	80	10,3	110	4,0
22	69	52	23,0	82	9,6	112	3,5
24	59	54	23,0	84	9,6	113	3,5
26	59	56	21,0	86	9,1		
28	49	58	19,0	88	8,4		
30	40	60	19,0	90	7,9		

Fonte: Autor.

deflexão do ponteiro do marcador de combustível, bem como, o circuito elétrico formado por esses dois componentes.

Por simplificação e para auxiliar-se no desenvolvimento do *software* do protótipo, utiliza-se para simular o sensor de combustível, um potenciômetro multivoltas, com valor de 220 Ohms, associado em paralelo com um resistor de 220 Ohms. Assim, existe a possibilidade de variar-se valores de resistência elétrica, entre 0 e 110 Ohms. Considera-se para esta etapa de desenvolvimento, essa seria uma aproximação suficientemente boa, comparando-se com os valores obtidos com o medidor de combustível instalado no veículo teste, entre 3,5 e 118 Ohms.

Figura 24 – Mostrador nível de combustível.



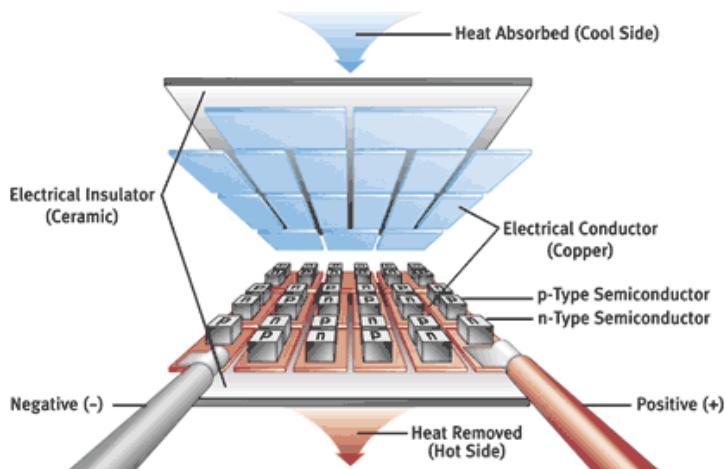
Fonte: Autor.

Para realizar-se a medida de combustível simulada, utiliza-se a porta analógica A0 do kit NodeMCU, com resolução de 10 bits, para valores de tensão de entrada entre 0 e 3 Volts. Para isso, estabeleceu-se um divisor de tensão, entre potenciômetro ligado a massa e um resistor de 66 Ohms ligado a 5 volts. Com isso, consegue-se variar a entrada de A0, entre 0 e 3 volts, excursionando-se o eixo do potenciômetro.

3.1.2.6 Pastilha de efeito Peltier TEC1-12706

Este tipo de dispositivo, conforme HB-Corporation (2022), utiliza-se do efeito termoelétrico ou efeito Peltier-Seebeck, que consistiria, na geração de um gradiente de temperatura, quando, uma corrente elétrica percorre por dois metais diferentes. Inversamente, poderia-se obter, uma diferença de potencial elétrico, através de uma diferença de temperatura entre os lados desse tipo de dispositivo. Desse modo, utilizando-se uma configuração de materiais apropriada, e fazendo-se circular uma corrente elétrica, pode-se extrair calor de um dos lados do dispositivo, denominado lado frio, e dissipar esse calor absorvido no outro lado do componente, lado quente, como ilustrado na Figura 25.

Figura 25 – Pastilha de efeito termoelétrico.



Fonte: HB-Corporation (2022).

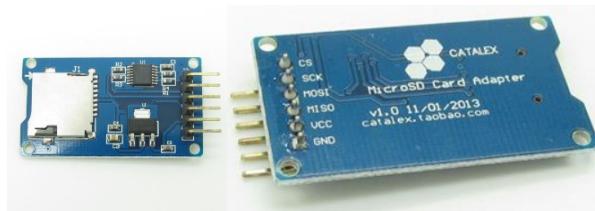
Para este trabalho, optou-se pelo dispositivo TEC1-12706, que conforme HB-Corporation (2022), consome uma corrente máxima de 6,4 Amperes e suporta uma tensão máxima de 16,4 Volts, valores que são suportados pelos demais componentes envolvidos neste projeto, assim como baterias, relés e fiação. Além disso, existe uma boa oferta desses dispositivos, que encontram aplicações comerciais, em bebedouros d'água e mini geladeiras. A parte fria desse componente é fixada na parte metálica da geladeira do veículo. No lado quente, foi acoplado um dissipador de calor de alumínio com um ventilador que força o ar quente para fora do veículo.

Para facilitar o desenvolvimento dos *softwares*, montou-se uma pastilha TEC1 entre dois *cooler's* de microprocessadores, com dois sensores DS18b20, um para o lado frio e outro para medir a temperatura do lado quente do sistema. Esse componente foi adicionado ao projeto, com a intenção de acelerar o resfriamento da geladeira quando o compressor está ligado, também, para auxiliar na obtenção de temperaturas menores do sistema. Além disso, também permitiria, manter-se o resfriamento quando o motor do veículo está desligado.

3.1.2.7 Módulo microSD

Para integrar-se ao *hardware*, uma memória micro SD, utilizou-se o módulo micro SD Catalex, o qual possui, um regulador de tensão para alimentar a memória sd, na qual, utiliza-se o formato FAT32 e que irá armazenar as informações coletadas pelo sistema em um arquivo de texto com extensão .log. Esses dados armazenados podem ser lidos pelo sistema ou em outros dispositivos compatíveis com essa tecnologia. Para tanto esse módulo disponibiliza um soquete adaptador, que permite facilmente a inserção e retirada do cartão de memória e uma interface de comunicação serial SPI que permite a leitura e gravação de dados através do microcontrolador ESP-12.

Figura 26 – Adaptador microSD Catalex.

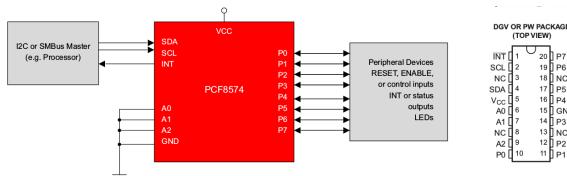


Fonte: Catalex (2013).

3.1.2.8 Expansor 8 bits para barramento I2C - PCF8574

Com a necessidade de interagir com motores, compressores e a pastilha TEC1, fez-se necessário a utilização de um expansor de portas IO, sendo que, optou-se pelo CI PCF8574, o qual, conforme Texas-Instruments (2015), funciona a partir de um barramento de comunicação serial que utiliza o protocolo I2C, que pode ser compartilhado com outros dispositivo, e portanto, não necessitaria de portas adicionais do microcontrolador. Assim, a inclusão desse dispositivo, permite a criação de um subsistema com 8 portas de IO adicionais. Além disso, esse componente oferece a possibilidade de realizar-se leituras nas portas, permitindo a inclusão de sensores tais como chaves mecânicas, *reed switches*, sensores de presença, entre outros.

Figura 27 – PCF8574 - Expansor 8 bits para barramento I2C.



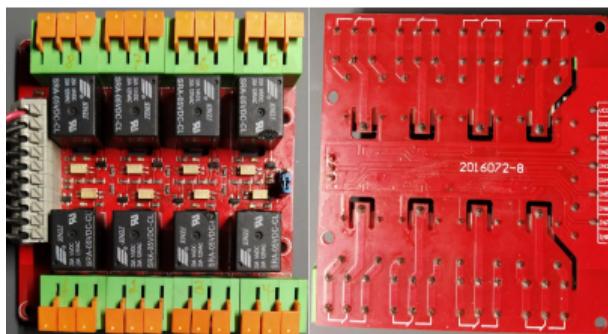
Fonte: Texas-Instruments (2015).

Como pode-se notar na Figura 27, o endereço I2C desse dispositivo, pode ser configurado através dos pinos A0, A1 e A2, que em nosso caso, foram ligados diretamente a tensão de referência, implicando que o endereço hexadecimal correspondente seria o 20h.

3.1.2.9 Módulo de relés

Para o acionamento de cargas, foi necessário a inclusão de um dispositivo para adaptar os sinais de controle das portas do CI PCF8574 aos níveis de tensão e corrente demandados pelos motores e compressores e a pastilha TEC1. Para isso, adicionou-se ao protótipo um módulo com 8 relés com bobinas de 5 V, que são acionadas através de fotoacopladores, os quais promovem isolação elétrica e a compatibilidade entre as portas do PCF8574 e o circuito de potência constituídos de bobinas de relés e motores.

Figura 28 – Módulo 8 relés.



Fonte: Autor.

Como pode-se observar na Figura 28, o conector no centro mais à esquerda, seria o responsável por conduzir os sinais de controle de acionamento dos relés através de fotoacopladores e transistores, e também, responsável por fornecer ao módulo, uma alimentação de 5 Volts necessária para o acionamento das bobinas dos relés. No caso deste projeto, um nível lógico falso recebido pelo módulo, significa o acionamento do relé. Essa lógica, poderia ser invertida através da configuração da chave, situada no centro entre as duas linhas de relés. Além disso, podem ser acionadas, cargas que consumam até 20 Amperes em cada saída, através dos terminais normalmente abertos (NA) ou normalmente fechados (NF), disponíveis nos conectores apresentados na porção esquerda desta imagem, na parte superior e também inferior. A parte direita da ilustração representa a porção esquerda da imagem vista pelo ângulo inferior.

3.1.2.10 Circuitos auxiliares

Para detectar-se o sinal de giro do motor, e poder-se inferir se o veículo está ligado ou não, foi necessário uma adaptação de níveis de tensão. O sinal de rotação do motor, proveniente do alternador do carro, tem uma amplitude de aproximadamente 12 Volts, e portanto, incompatível com o nível suportado pelo NodeMCU (D2). Assim, para promover essa adaptação, optou-se por utilizar um componente fotoacoplador. Assim, os pinos do led emissor, do CI fotoacoplador, foram ligados ao sinal proveniente do alternador, através de um resistor 1K Ohm. O coletor do transistor desse componente, foi ligado para a alimentação de 3V3 através de um resistor de 4K7, e ao pino D2 do MCU. Assim um

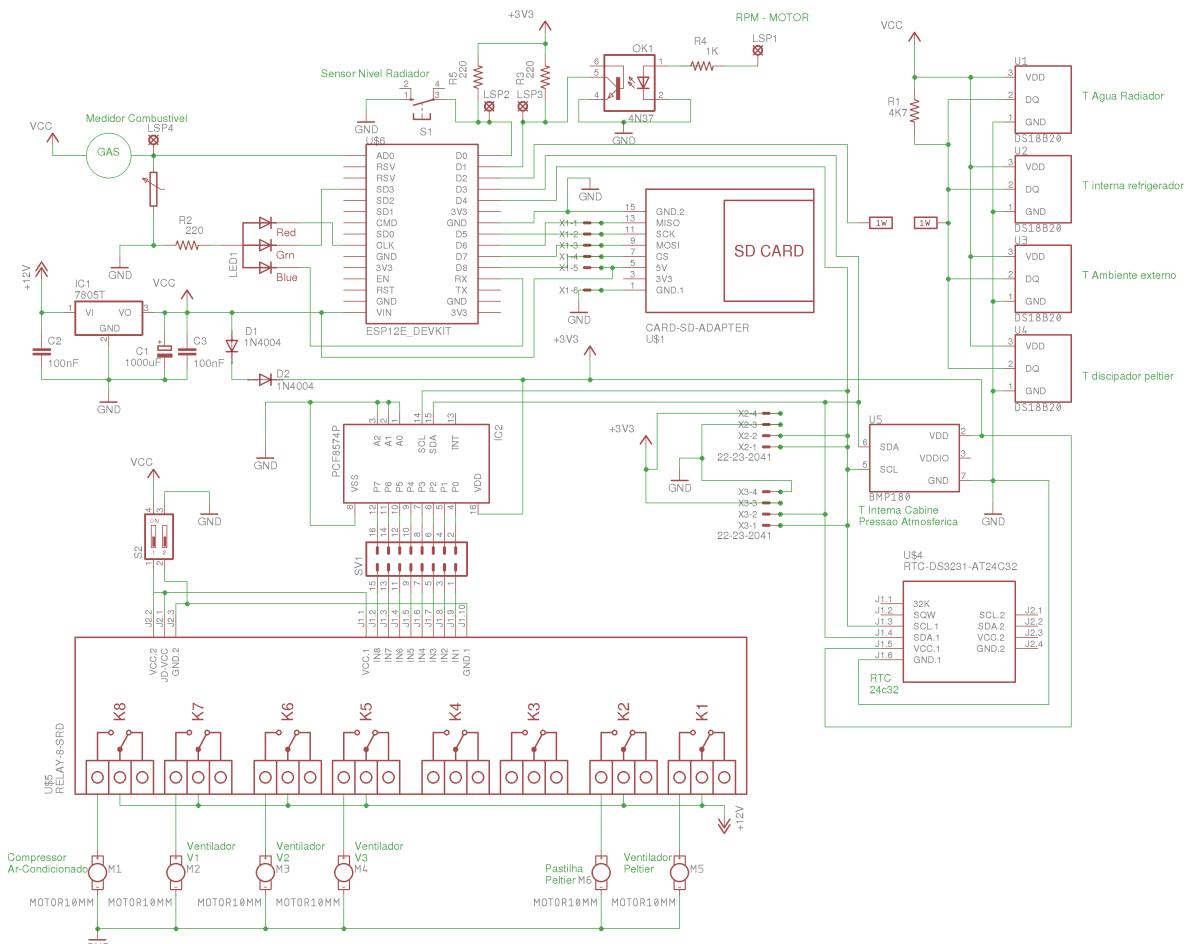
pulso nível 1, proveniente do alternador, é recebido como um pulso negativo no pino D2. O diagrama elétrico desse circuito está representado na Figura 29.

3.1.3 Integração do hardware

Com o fim de integrar todos os componentes do sistema, faz-se necessário a construção de um projeto elétrico, que especifique as portas do microcontrolador, onde serão implementadas as interfaces de comunicação I2C, SPI e OneWire, necessárias para a troca de informações entre o ESP-12 e os demais módulos e sensores, bem como promover a alimentação adequada para esses circuitos.

Assim desenvolveu-se o esquema de ligações representado pelo diagrama elétrico apresentado na Figura 29. Para tanto, especificou-se que: O pino D0, da placa de desenvolvimento NodeMCU V1.0, verificará o nível de fluído no radiador do automóvel através de um interruptor de boia magnética ES7510. O pino D1 foi conectado ao sensor de rotação do motor, permitindo medir a quantidade de rotações por minuto (RPM) que o motor está desenvolvendo, assim possibilitando inferir se o motor está ligado.

Figura 29 – Diagrama elétrico do protótipo.

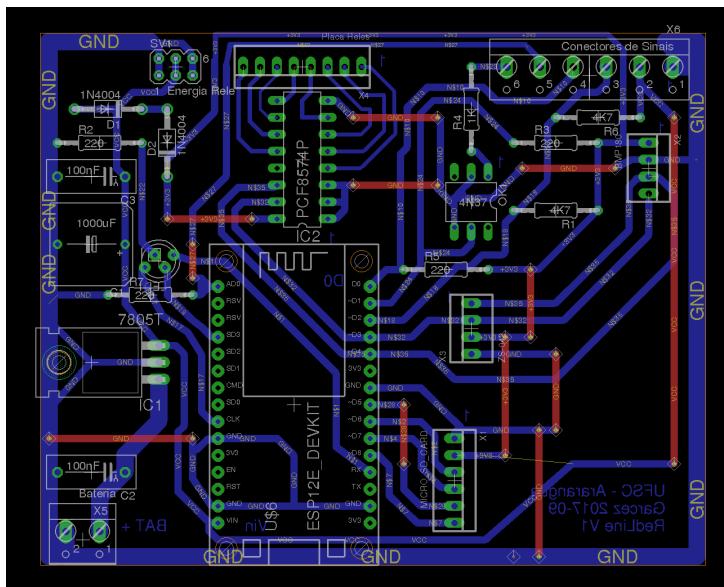


Fonte: Autor.

O pino D2 foi definido como sinal de dados do barramento OneWire, o qual é responsável pela comunicação do microcontrolador com os sensores DS18B20. O pino D3 foi designado como Dados (SDA) e o Pino D4 como clock (SCL), que juntos implementam o protocolo I2C que fará a comunicação entre o PCF8574, o módulo ZS-042 e o módulo BMP280 e o microcontrolador. Os pinos D5, D6, D7 e D8 implementam o protocolo SPI utilizado para gravar os dados coletados no cartão de memória SD, sendo que D5 implementa o clock (SCK), o D6 conecta-se ao pino MISO do SD, o D7 ao pino MOSI e o D8 ao pino seletor de chip (CS) e o pino A0 fará a digitalização da leitura de combustível. Além do diagrama elétrico pode-se visualizar o mapeamento das portas e os protocolos utilizados, na Figura 39 do Anexo A.

Para implementar o esquema de ligações foi necessário construir uma placa de circuito impresso (PCB), responsável pela interligação física das ligações propostas no projeto elétrico e cujo layout, pode se observado na Figura 30. Essa PCB, foi elaborada juntamente com o diagrama elétrico, utilizando-se o *software* Eagle. Para a confecção da placa, utilizou-se a prototipadora cnc, disponibilizado pela UFSC campus Araranguá.

Figura 30 – Layout placa de circuito impresso (PCB) do protótipo.



Fonte: Autor.

3.2 PROTÓTIPOS SOFTWARE

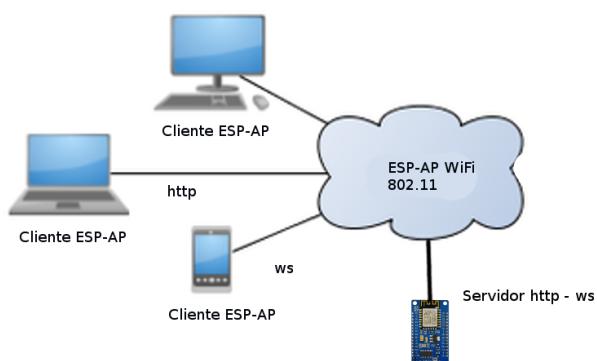
No projeto e desenvolvimento dos *softwares* protótipos, produz-se algumas soluções de *softwares* funcionais, que classificam-se em dois segmentos principais. O *software backend*, que será executado pelo microcontrolador escolhido e os protótipos de *frontend*, de uma arquitetura cliente-servidor *web*, que será executado em um browser ou em um app para Android. Para tanto, segue-se as seguintes etapas:

- a) Pesquisa e escolha dos ambientes de desenvolvimento de *software*;
- b) Pesquisa e escolha de bibliotecas de funções para interagir com *hardware* de módulos e sensores;
- c) Desenvolvimento do *software backend* que disponibiliza os dados coletados para clientes de uma rede WiFi;
- d) Desenvolvimento dos *softwares frontend*, que disponibilizam algumas opções de interfaces. Tanto para que o usuário possa interagir com o sistema, quanto para facilitar o desenvolvimento e a manutenção do sistema;

3.2.1 Arquitetura cliente servidor

Um servidor, em uma arquitetura cliente/servidor, é um *software*, que mantém uma porta de comunicação aberta, esperando a conexão de um cliente. Os requisitos para acesso, tais como, o endereço de sua localização na rede, senha para acesso e o formato das requisições, devem ser de conhecimento do cliente. Por outro lado, o cliente, que também é um *software*, normalmente é acionado por um usuário. *Softwares*, tais como, os *browsers* são clientes de servidores da *web*. No modelo de comunicação, cliente e servidor, o cliente é sempre responsável pelo inicio da comunicação. Esse conceito, é utilizado pelos *softwares* desenvolvidos para o protótipo. Onde, o MCU ESP-12, trabalha como um servidor de requisições *Hypertext Transfer Protocol* (HTTP) ou *websocket* (WS). (OLIVEIRA, 2017).

Figura 31 – Arquitetura cliente servidor.



Fonte: Oliveira (2017).

3.2.2 Software do servidor web

Para a elaboração do *software backend*, utiliza-se a plataforma Arduino. Conforme os desenvolvedores desse ambiente, foi concebido por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis, utilizando-se o framework Wiring, que consiste em uma plataforma de prototipagem eletrônica *open-source* e que se baseia, em *hardware*

e *software* flexíveis e fáceis de usar. Assim, esse ambiente de desenvolvimento integrado (IDE), é disponibilizado para sistemas operacionais Windows, MacOS, Linux 32 e 64 bits e também para Linux ARM. Além disso, nessa IDE, para o desenvolvimento de programas e bibliotecas, utiliza-se, as linguagens de programação C ou C++. Cabe ressaltar, que nessa plataforma, para compilar um programa que foi desenvolvido, pode-se escolher-se um alvo, dentre os vários tipos de kit's de desenvolvimento compatíveis. Sendo que, o kit NodeMCU, que utiliza-se neste protótipo, é um dos dispositivos compatíveis.(Arduino, 2018).

Fazendo-se uso dessa plataforma, produz-se o *software backend* para o protótipo. Visando-se atender, os requisitos funcionais obrigatórios e os requisitos funcionais desejáveis. Para isso, coleta-se todas as informações relevantes, e que foram obtidas através dos sensores. Armazenam-se as informações, em memória volátil e em arquivo de texto com extensão .log. Assim, o servidor de requisições *web*, pode entrega essas informações aos clientes. As características e detalhes de funcionamento dos *softwares*, podem ser melhor observadas, consultando-se o código fonte, disponibilizado em Garcez (2022).

Para atender-se, as requisições do tipo http GET. Emprega-se, o modelo de servidor, ESP8266WebServer. O qual, é disponibilizado, na forma de biblioteca funções, no ambiente Arduino. Para o funcionamento desse servidor, uma função deve ser chamada, para verificar se existe alguma solicitação http, na porta de comunicação configurada para o serviço.

Para atribuir-se funcionalidade a esse serviço, um conjunto de *Uniform Resource Identifier* (URI), necessitou ser configurado. Neste trabalho, determinou-se que a URI raiz (/), fornece ao cliente, uma página html, contendo a relação das outras URI's, que podem ser acessadas nesse servidor.

Como por exemplo, existe uma requisição, de um cliente, ao endereço do servidor, com número IP, 192.168.4.1, na porta, 80 e com URI, (/leituraDados). Ou seja, (<http://192.168.4.1/leituraDados>). Neste protótipo em desenvolvimento, uma URI (/leituraDados), foi discriminada, e executa uma função, para atualizar os dados dos sensores instalados. De posse das informações atualizadas, é formada uma *string*, denominada bilhete. No bilhete, os dados são separados por vírgulas e espaços em branco. O servidor, como resposta à essa requisição, entrega ao cliente, na forma de texto plano, o bilhete e encerra-se a conexão. Essa URI específica, (/leituraDados), é empregada no funcionamento de partes dos aplicativos app Android e browser http.

Desse modo, outras URI's foram implementadas para atender os requisitos estabelecidos. Assim como, a URI (/index), que entrega como resposta, a página browser http, a URI (/index2), que envia como resposta a página browser *websocket*. Além dessas, em conjunto com as demais URI's elaboradas, disponibilizam-se, todas as funcionalidades, oferecidas pelo *backend* de requisições http.

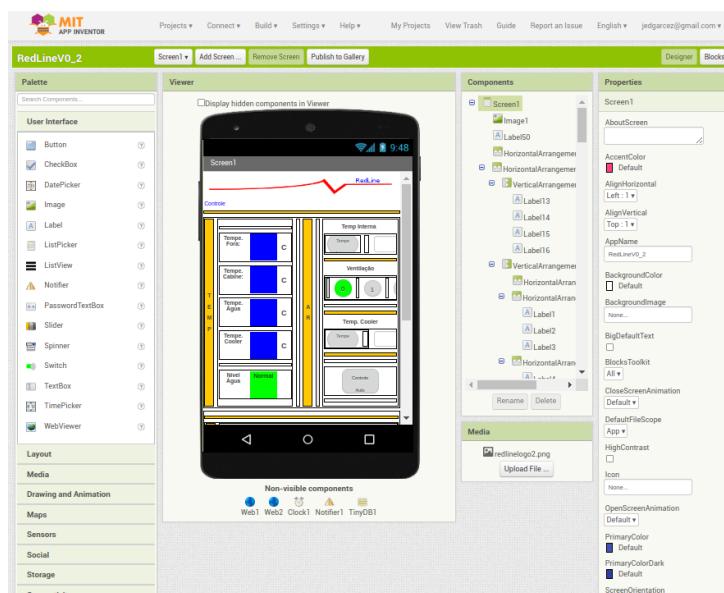
Para estabelecer-se comunicação via *websocket*, utiliza-se a biblioteca arduinoWebSockets, disponibilizada por Venter (2022). Esse tipo de conexão é iniciada pelo cliente,

mas, uma vez estabelecida, cria-se um canal permanente, onde tanto o cliente como o servidor podem enviar mensagens a qualquer momento. Para isso, uma função no laço principal, verifica constantemente, se existe alguma mensagem na porta de comunicação 81, que foi configurada para esse serviço. Neste projeto desenvolvido, uma mensagem válida, é identificada por um prefixo, seguido do conteúdo útil. Por exemplo, uma mensagem contendo a informação de volume de combustível, é formada como (#COMB+valor). Dessa forma, quando a função do *software* do MCU, verifica que o volume de combustível sofreu uma variação significativa, pode enviar uma mensagem ao outro lado. Utilizando-se a convenção pré-estabelecida, o recebedor da mensagem, pode executar alguma funcionalidade programada.

3.2.3 Software do cliente web - app Android

Para a construção da aplicação *frontend*, destinada para sistemas Android, optou-se por utilizar a plataforma MIT App Inventor. Esse ambiente, utiliza conceitos de programação visual, que permite o desenvolvimento de aplicativos, sem o uso de códigos escritos. Para isso, essa IDE de programação, disponibiliza uma interface *web*, onde pode-se desenvolver o desenho da interface gráfica. Permitindo-se adicionar, áreas de texto, botões, imagens, listas e outros elementos. Associado a esses elementos gráficos, pode-se desenvolver, com um conjunto de funções, elementos que promovem funcionalidades e dinamicidade, a interface gráfica produzida. Como por exemplo, ao encostar-se o dedo em um botão, pode-se trocar a cor desse botão, indicando que ele foi ativado e pode-se também, enviar essa informação para outros dispositivos conectados via WiFi.

Figura 32 – Interface gráfica do aplicativo frontend.



Fonte: Autor.

Observando-se a Figura 32, percebe-se, mais à esquerda, uma coluna de elementos,

que são utilizados na construção da interface do aplicativo. Nessa área, encontram-se, os botões, notificadores, *textbox*, *labels*, tabelas, elementos de conectividade, entre outros, utilizados na elaboração da parte visual e na programação do *software*. Na área central, dessa mesma ilustração, é mostrado uma simulação da aparência do app, utilizando-se os elementos mencionados anteriormente. Logo abaixo do telefone, aparecem os itens, que tem função de programação e não interferem no visual do app. Tais como, os de conectividade *web*, banco de dados, entre outros. Deslocando-se para a direita, encontra-se, a coluna de propriedades dos elementos já adicionados a interface. Nessa área, pode-se alterar as condições iniciais dos elementos, assim como, tamanho de texto, cor de fundo, valores, e outros.

O app desenvolvido, inicia em uma tela auxiliar, onde é oferecida a opção de conectar-se ao servidor, através do endereço da rede do AP NodeMCU ou se deseja inserir o endereço de cliente, na rede onde o NodeMCU conectou-se. Essa tela, é mostrada na Figura 33. Após essa etapa, a página principal é exibida. A Figura 34, ilustra a página principal do app protótipo que foi desenvolvido.

Figura 33 – Página inicial do app.



Fonte: Autor.

Considerando-se a representação do telefone, ilustrado na Figura 34. Mais a esquerda, situa-se a área de exibição das temperaturas enviadas pelo servidor. O primeiro valor mostrado nessa coluna é a temperatura externa ou ambiente, logo abaixo, é mostrado a temperatura da cabine. Com base nesses dois valores, implementa-se no microcontrolador, uma estratégia para um controle automático de temperatura do ar condicionado. Um pouco mais abaixo, é mostrado a temperatura da água do radiador. Continuando a descer nessa coluna, é mostrado a temperatura interna da geladeira do automóvel. Por último, nessa coluna da área TEMP, é mostrado uma sinalização de alarme de nível d'água.

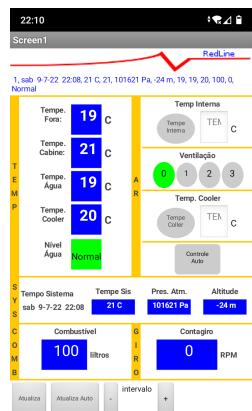
Ainda, considerando-se essa ilustração, na parte mais a direita, situam-se os controles do ar condicionado. São oferecidos campos, para inserção de valores de temperaturas,

que serão enviadas ao servidor. Botões, para acionar o ventilador do ar condicionado e um botão, para acionar-se a função de ar condicionado automático.

Mais abaixo, na área SYS, são apresentadas, as informações do relógio do servidor, fornecidas através dos dados providos pelo módulo ZS-042. O próximo dado recebido e apresentado nessa área, é a temperatura do ambiente do hardware. Essa medida, é utilizada para monitorar, a temperatura de acondicionamento do sistema, no veículo. Ao lado direito, é apresentado a pressão atmosférica, em Pascais e a altitude relativa estimada. Na área COMB, é apresentado o volume de combustível no tanque e na área GIRO, a rotação do motor em RPM.

Na parte mais inferior, da tela principal, é oferecido um botão, de atualização dos dados. Se desejado, pode-se configurar, uma atualização automática.

Figura 34 – Página principal do app.

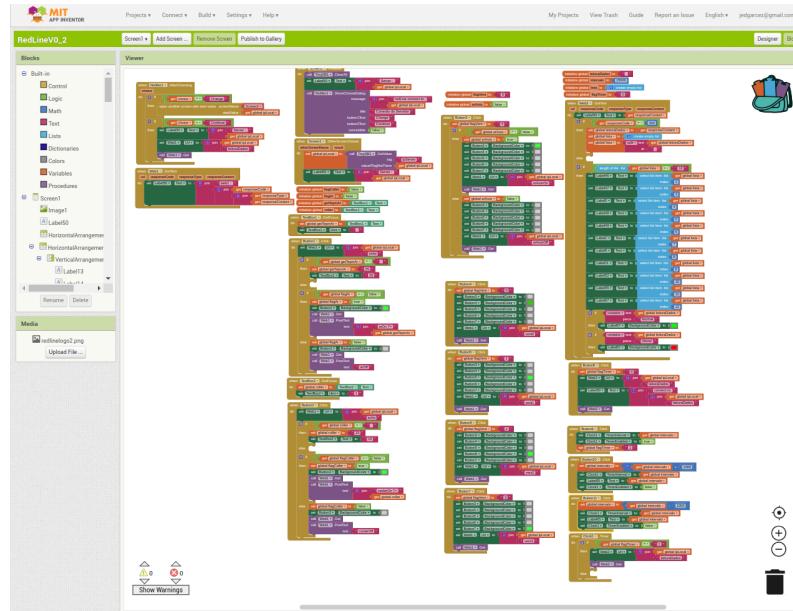


Fonte: Autor.

Na área de programação do App Inventor, são disponibilizados blocos, que realizam funções, como declaração de variáveis, tratamento de eventos, operações lógicas e matemáticas, ações sobre a interface gráfica, tais como, mudanças de cor, substituição de textos, e outros. Com isso, é possível fazer pequenas aplicações, de forma rápida com uma boa quantidade de recursos. Na Figura 35, é apresentado a programação da tela principal do aplicativo desenvolvido.

O princípio de funcionamento do aplicativo, baseia-se em requisições http GET, que são enviadas ao servidor com o *software backend*. A resposta recebida, é uma sequência pré-ordenada, dos dados apurados pelo microcontrolador no momento da requisição para a URI (/leituraDados). Utilizando-se, um bloco do tipo *when web.GotText*, é construída, uma lista, onde, cada elemento, é separado e destinado, para suas respectivas posições, nos elementos gráficos da tela. Como esse aplicativo, é um protótipo, para fins, de teste de comportamento, e para facilitar a análise do funcionamento, foram adicionados botões, para atualização. Cada vez, que o botão é acionado, o app envia uma requisição e trata os

Figura 35 – Programação do aplicativo frontend app.



Fonte: Autor.

dados recebidos. Também foi adicionado um botão, Atualiza Auto. Quando esse botão é acionado, o app envia consecutivas requisições, determinadas pelo tempo, em milisegundos, definido no campo ao lado.

Quando acionado, os botões relativos ao ventilador, o app envia uma requisição GET, onde, uma URI programada no *software backend*, reconhece o comando e executa as ações. Nesse tipo de requisição, o servidor, envia ao app, apenas a confirmação de recebimento, sem dados de retorno. Além disso, foram programadas, ações para alterar a cor do botão que está ativo. As demais funcionalidades, relativas ao ar condicionado e geladeira, ainda estão sendo desenvolvidas e não possuem funções definidas.

3.2.4 Software do cliente web - browser http

Para funcionar, de forma semelhante ao cliente app, ou seja, recebendo um conjunto de informações pré-ordenadas em um bilhete. Desenvolveu-se, uma página html, que é atualizada pelo *software backend*, a cada nova requisição recebida, em uma determinada URI. Assim, a cada requisição feita pelo cliente, obtém-se como resposta, a página html, ilustrada na Figura 36. Desse modo, pode-se utilizar, um computador de mesa com um navegador *web*. Desse modo, pode-se testar, o desenvolvimento das funcionalidades destinadas ao app, utilizando-se um PC. Se for desejado, também pode-se utilizar essa forma, para receber, as informações dos sensores, disponibilizadas pelo sistema. Além da parte visual, existem diferenças no processamento das informações. No app, a mensagem é processada pelo telefone. Na página html, a mensagem é gerada e processada, internamente no MCU, e enviada para o cliente.

A Figura 36, ilustra a pagina html enviada pelo servidor após executar-se a requisição para a URI devida.

Figura 36 – Página html servidor http.



Fonte: Autor.

3.2.5 Software do cliente web - browser websocket

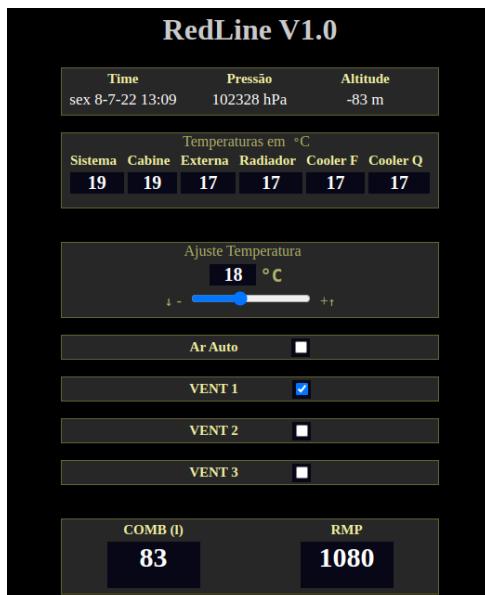
Também foi desenvolvido, para trocar informações com o *software backend*, uma página para clientes *browsers*, elaborada com html associado com javascript. Para isso, faz-se uso, de uma conexão *websocket*. Essa página, pode ser aberta, através de uma requisição http, para a URI encarregada. Uma vez que essa página é recebida pelo cliente, uma conexão socket é iniciada na porta 81. Quando estabelecida, esse tipo de conexão persiste aberta. Ao contrário, das requisições http, utilizadas neste trabalho. Desse modo, mensagens podem ser trocadas, a qualquer momento e em qualquer sentido. Na Figura 37, ilustra-se, a interface desenvolvida com javascript, associado ao html. Percebe-se, que o visual é semelhante ao da página html. Possuindo os mesmos campos de entrega de informação. Porém, a forma de funcionamento é diferente. Na página html, os campos são preenchidos no *backend*, que envia a página completa e fecha a conexão. O cliente browser recebe a página, e apenas, renderiza a informação. Na página ws, a conexão é permanente, e mensagens, são trocadas a todo momento.

Os dados, apurados pelo *hardware* e *software* do MCU, tais como, a informação de rotação do motor, podem modificar-se rapidamente. Como o uso do WS, essa informação, pode ser enviada de imediato, sem a necessidade de gastar-se tempo, com protocolos de inicialização de comunicação. Além disso, diferentemente do servidor http, o lado do

servidor WS, pode enviar mensagens, independentemente de solicitação do cliente. Assim, uma mensagem recebida pelo navegador *web*, é tratada pelo javascript, e a atualização das informações mostradas, não consome recursos do MCU.

Semelhantemente ao cliente html, mostrado na Figura 36, na parte superior, da ilustração da Figura 37, é mostrado uma mensagem de alarme, quando for necessário. Logo abaixo, formando um grupo horizontal de informações, apresentam-se, a data e hora do MCU, seguido pela leitura de pressão atmosférica e finalizado com a altitude estimada. No segundo grupo horizontal, todas as informações de temperatura são apresentadas. No terceiro grupo, apresenta-se um botão deslizante, com a função, de enviar a temperatura alvo, do controle automático, do ar e da geladeira. Quando esse botão é acionado, o valor da temperatura, é apresentado no campo logo acima. Essa ação, dispara uma mensagem, com o formato "#ART+valor do botão", que é enviada ao MCU. Logo abaixo desse botão, localiza-se um grupo de *checkbox*, com função auto exclusiva. O acionamento de uma, implica no desligamento de outro, que esteja ligado. Desse modo, quando um desses campos é verificado, funções do javascript, atualizam o html e enviam a informação ao MCU. No último grupo horizontal, na parte mais inferior, é mostrado os valores de volume de combustível e de rotação do motor.

Figura 37 – Página html/javascript *websocket*.



Fonte: Autor.

3.2.6 Software do cliente *web* - outras páginas

Também produziu-se, outras páginas, para a utilização do servidor http bem como para o servidor *websocket*. Exemplificando-se, quando tenta-se acessar a URI raiz, com uma requisição similar a, <http://192.168.4.1/>, originada em um browser ou outro tipo de aplicativo. Ou ainda, utilizando-se comandos como o curl. Obtém-se como resposta, o

conjunto de URI's que podem ser acessadas nesse servidor. Também desenvolveu-se, uma página de teste, que estabelece uma conexão ws e sobrepones as regras estabelecidas para as demais páginas. Com isso, pode-se acionar ou desligar, dispositivos sujeitos a regras, no funcionamento normal. Como por exemplo, o motor do ventilador possui 3 fios positivos, os quais, não podem ser ligados a 12 Volts simultaneamente. Estando em funcionamento normal, o sistema de *software*, impede que se acione, mais do que um relé responsável por ligar o ventilador. Utilizando-se a página teste, todos os relés podem ser acionados.

4 CONCLUSÃO E TRABALHOS FUTUROS

O *hardware* que foi desenvolvido para o protótipo, apresenta boa flexibilidade. Podendo servir, como um kit base, para o desenvolvimento de outros tipos de aplicações, tais como, automação residencial, dispositivos de controle de sistemas mecânicos, estudos acadêmicos, e outros. Para isso, bastaria incluir novos módulos e sensores ou substituir os empregados neste trabalho e desenvolver novos *softwares*. Além disso, pode-se também, adicionar diversas novas funcionalidades ao sistema desenvolvido até o momento. Como por exemplo, com a instalação de um modulo gps, poderia-se desenvolver um controle automático de velocidade, indicadores de aceleração e frenagem, estimar-se o consumo de combustível por Km rodado, entre outros. Com um sensor de vibrações, associado com as coordenadas de um gps, poderia-se alimentar um banco de dados, contendo informações sobre o estado de conservação de estradas, por meio da detecção impactos em buracos na pista.

Os *softwares* desenvolvidos apresentam-se em fase intermediária, porém satisfatória, pois somente um requisito funcional desejável, o alarme sonoro, não foi pelo menos parcialmente atendido. Caberia continuar o desenvolvimento de *software*, tanto no lado do servidor, quanto das aplicações cliente, melhorando as funções já desenvolvidas e organizando melhor os códigos através da criação de novas bibliotecas e adicionando-se novas funcionalidades, que contribuam para a evolução desse projeto.

Quanto aos requisitos apresentados no Quadro 1, da sessão de projeto e desenvolvimento. avalia-se que:

RFO-1, medição de temperatura do fluido de arrefecimento do motor; foi atendido completamente utilizando-se um sensor DS18b20, que suportou as temperaturas e a imersão no fluido de arrefecimento. Os valores medidos são apresentados nas interfaces gráficas.

RFO-2, alerta de temperatura do fluido de arrefecimento do motor; foi atendido completamente, oferecendo-se na interface do app a mudança da cor de fundo, do campo que exibe a temperatura da água, de azul para vermelho. E nas interfaces para navegadores, oferecendo uma mensagem de texto logo abaixo do título da tela.

RFO-3, alerta de nível do fluido de arrefecimento no radiador; foi atendido completamente, oferecendo-se na interface do app a mudança da cor de fundo, do campo que exibe o nível da água, de verde para vermelho. E nas interfaces para navegadores, oferecendo uma mensagem de texto logo abaixo do título da tela.

RFO-4, medição de nível de combustível; foi atendido parcialmente, pois ainda não foi desenvolvido, o método para estimar o volume de combustível com o sensor instalado no veículo. Utilizou-se um circuito com um potenciômetro para simular o sensor.

RFO-5, alerta de nível de combustível; foi atendido parcialmente, pois ainda não foi desenvolvido, o método para estimar o volume de combustível com o sensor instalado

no veículo. Utilizou-se um circuito com um potenciômetro para simular o sensor. Quando o nível simulado, for menor do que oito litros, acontece uma mudança na cor de fundo, de azul para amarelo, no campo que apresenta essa informação no app Android. Nos demais aplicativos *frontend*, uma mensagem de texto é apresentada.

RFO-6, medição de temperatura da geladeira do veículo; foi atendido completamente, utilizando-se um sensor DS18b20 fixado na parte metálica do refrigerador. Os valores medidos são apresentados nas interfaces gráficas.

RFO-7, acelerar o processo de refrigeração da geladeira do veículo; foi atendido parcialmente, utilizando-se uma pastilha TEC1 fixado na parte metálica do refrigerador. Ainda não foram feitos ensaios para quantificar a redução da temperatura que esse dispositivo é capaz de promover na geladeira do veículo. Foram realizados testes simulados em bancada, para auxiliar o desenvolvimento dos programas.

RFO-8, medição de temperatura interna da cabine; foi atendido completamente, utilizando-se um sensor BMP280, instalado no ambiente da cabine do veículo. Os valores medidos são apresentados nas interfaces gráficas.

RFO-9, fornecer um sistema digital de controle da temperatura interna da cabine; foi atendido parcialmente, utilizando-se um sensor DS18b20 fixado na parte externa traseira do veículo e um sensor BMP280, no ambiente interno da cabine. Foi desenvolvido um sistema básico de controle, baseado na diferença, dessas duas temperaturas. Caberia testar a eficiência desse controle, no ambiente real de uso.

RFO-10, fornecer uma interface para leitura dos dados coletados e envio de informações para os sistemas de controle; foi atendido parcialmente, não foram programadas todas as funcionalidades para controle do ar condicionado.

RFD-1, medição de pressão atmosférica e altitude; foi atendido completamente, utilizando-se um sensor BMP280, já instalado no ambiente da cabine do veículo. Os valores medidos são apresentados nas interfaces gráficas.

RFD-2, fornecer um alerta sonoro de falhas; não foi atendido.

RNFO-1, fornecer um sistema de armazenamento para os dados coletados nas medições; foi atendido completamente, utilizando-se strings pré formadas, e um conjunto de convenções para envio de mensagens entre os servidores de conteúdo e as aplicações móveis.

RNFO-2, fornecer um sistema de coleta dos dados armazenados através de um PC ou rede; foi atendido completamente, utilizando-se um cartão microSD, mas para evitar-se a retirada do cartão para posterior leitura em um computador, planeja-se utilizar um sistema de transferência de arquivos via rede WiFi.

RNFO-3, fornecer Medida temperatura do dissipador de calor da pastilha Peltier; foi atendido completamente, utilizando-se um sensor DS18b20 fixado na parte metálica do dissipador de calor da pastilha TEC1. Os valores medidos são apresentados nas interfaces gráficas, excetuando-se o app Android. Essa aferição, serve para o sistema de proteção de

sobretemperatura da TEC1.

RNFO-4, fornecimento de uma interface atuadora de potência para controle de motores e trocadores de calor; foi atendido completamente, utilizando-se um módulo de 8 relés. Porém planeja-se substituir-se esse componente, devido a falhas de fixação dos fios, nos conectores da parte de potência.

RNFO-5, projeto e construção de protótipos de software e hardware; considera-se atendido completamente, considerando-se uma primeira aproximação para a solução pretendida.

RNFD-1, fixação dos sensores e atuadores e passagem de fiação no veículo; foi atendido parcialmente. Os sensores DS18B20 foram instalados e testes básicos de funcionamento foram realizados. O sensor de nível de água foi instalado e testado. A pastilha TEC1 foi instalada e foi realizado um teste básico de funcionamento. Como não dispomos, de um modulo de 8 relés e um módulo BMP280 sobressalente, esses dispositivos não foram instalados no veículo.

RNFD-2 acondicionamento do sistema no protótipo. considera-se atendido parcialmente, definiu-se o local de fixação no veículo, porém, a caixa de acondicionamento do protótipo, necessita ser avaliada.

A plataforma Arduino permitiu um desenvolvimento rápido do *software backend*, por conta das bibliotecas de funções para uso dos módulos e sensores empregados nesse protótipo. Além disso, a linguagem empregada possui boa documentação e muitos exemplos de uso disponíveis online. Outro fator importante, foi a compatibilidade com o kit NodeMCU, que funcionou adequadamente. A compilação e transferência do firmware é razoavelmente rápida. Assim, para o seguimento desse projeto e para a implementação de novas funcionalidades incorporadas, será mantido o uso dessa ide.

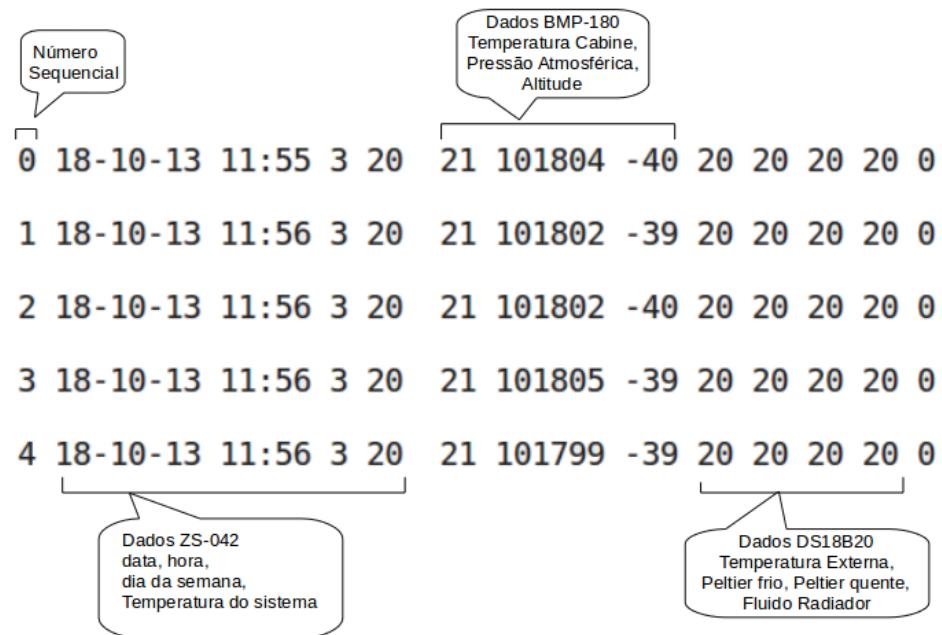
O ambiente App Inventor, é capaz de permitir, uma evolução rápida do desenvolvimento do *software frontend*, principalmente, para quem não tem nenhuma familiaridade com o desenvolvimento de aplicativos para Android. Além disso, nessa plataforma, não é necessário a instalação local de ide's ou compiladores, pois configura-se, como um ambiente de desenvolvimento na nuvem. Como desvantagem, o processo de baixar o app compilado e instalar no telefone Android, pode se tornar massante e um pouco demorado.

A elaboração da interface com o usuário, desenvolvida utilizando-se a linguagem javascript e html, mostrou-se satisfatória. Através de navegadores web, pode-se fornecer as mesmas funcionalidades desenvolvidas no app, porém, com a vantagem do uso multiplataforma. Assim, o usuário do sistema poderia acessar e enviar dados ao servidor através de um navegador, instalado em um PC, MAC, tablet, celular ou outro equipamento, que disponha desse tipo de recurso.

As informações coletadas juntamente aos sensores, são armazenadas em arquivos de texto gravados em um cartão de memória micro SD. No arquivo tipo texto, as informações das leituras dos módulos e sensores são organizadas em tuplas que são formadas por um

número identificador sequencial que inicia em zero e é incrementado de uma unidade a cada leitura gravada pelo sistema. Sendo que, esse contador, retorna para zero diariamente e a cada vez que o sistema é reiniciado. Seguindo-se a composição da linha, conforme a ilustração da Figura 38, são armazenadas as informações fornecidas pelo módulo ZS-042, que são: data no formato aa-mm-dd, a hora no formato hh:mm, a temperatura de funcionamento do ambiente de acondicionamento do protótipo. Seguindo-se, são apresentados os valores informados pelo módulo BMP-180, que são a temperatura da cabine, a pressão atmosférica em Pa e a altitude estimada. Seguindo-se apresentam-se as informações de temperatura dos sensores DS18B20 instalados juntamente ao ambiente externo da cabine, no lados quente e frio da pastilha Peltier e no fluido de arrefecimento do motor.

Figura 38 – Formato do arquivo de armazenamento das informações.



Fonte: Autor.

As informações contidas nas tupla, são separadas por espaço em branco e podem ser facilmente decodificadas por outros *softwares* como planilhas eletrônicas. Cabe ressaltar, que o protótipo foi apenas testado em bancada, cabendo a realização de testes de funcionamento, robustez, confiabilidade e demais testes, os quais devem ser executados em uma etapa futura do desenvolvimento.

REFERÊNCIAS

- AIDEEPEN. **DS18B20**: Digital Temperature Sensor Thermometer Waterproof Sensor Probe Stainless Steel. [S.l.: s.n.], 2022. Disponível em: https://www.aideepen.com/products/1m-2m-2-5m-3m-waterproof-ds18b20-digital-temperature-sensor-thermometer-waterproof-sensor-probe-stainless-steel?_pos=6&_sid=bd50bb204&_ss=r. Acesso em: 20 mai. 2022.
- AIDEEPEN. **Module BMP280 Digital Pressure Sensor**. [S.l.: s.n.], 2022. Disponível em: <https://www.aideepen.com/collections/pressure-sensor/products/bmp280-pressure-sensor-module-arduino-high-precision-atmospheric>. Acesso em: 20 mai. 2022.
- AIDEEPEN. **ZS-042**: DS3231 AT24C32 IIC Module. [S.l.: s.n.], 2022. Disponível em: https://www.aideepen.com/products/ds3231-at24c32-iic-module-precision-rtc-real-time-clock-memory-for-arduino?_pos=2&_sid=699819a57&_ss=r. Acesso em: 20 mai. 2022.
- AITHINKER. **NodeMCU-8266**. [S.l.: s.n.], 2022. Disponível em: http://www.ai-thinker.com/pro_view-67.html. Acesso em: 20 mai. 2022.
- ANFAVEA. **Séries mensais, a partir de janeiro/1957, de automóveis por segmento (automóveis, comerciais leves, caminhões, ônibus, total) de produção; licenciamento de nacionais, importados e total; exportações em unidades**. São Paulo, SP.: ANFAVEA, 2022. Disponível em: <https://anfavea.com.br/site/edicoes-em-excel/>. Acesso em: 2 jun. 2022.
- ANJOS, Eduardo Giovannetti Pereira dos. **A evolução da eletrônica embarcada na indústria automobilística brasileira**. São Caetano do Sul, SP.: CEUN-EEM, 2011. P. 126. Disponível em: <https://maua.br/files/monografias/a-evolucao-da-eletronica-embarcada-na-industria-automobilistica-brasileira.pdf>. Acesso em: 2 jun. 2022.
- BOSH. **BMP280 Digital Pressure Sensor**: Revision 1.26. [S.l.: s.n.], out. 2021. Disponível em: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp280-ds001.pdf>. Acesso em: 20 mai. 2022.
- CABRAL, Danilo Cezar. **Qual foi o primeiro carro a combustão?** [S.l.]: Abril, 2020. Disponível em: <https://super.abril.com.br/mundo-estranho/qual-foi-o-primeiro-carro-a-combustao/>. Acesso em: 31 mai. 2022.

- CATALEX. **Micro SD Card Micro SDHC Mini TF Card Adapter Reader Module for Arduino.** [S.l.: s.n.], 2013. Disponível em: <http://datalogger.pbworks.com/w/file/fetch/89507207/Datalogger%5C%20-%5C%20SD%5C%20Memory%5C%20Reader%5C%20Datasheet.pdf>. Acesso em: 20 mai. 2022.
- HB-CORPORATION. **Peltier Modules: Thermoelectric Coolers.** [S.l.: s.n.]. Disponível em: <https://peltiermodules.com>. Acesso em: 14 mai. 2022.
- COUTINHO, Roberto. **Quantos metros de fiação existe em um carro?** [S.l.]: Abril, 2021. Disponível em: <https://quattrorodas.abril.com.br/auto-servico/quantos-metros-de-fiacao-existe-em-um-carro/>. Acesso em: 31 mai. 2022.
- ELECALL. **ES7510: Small Stainless Steel Float Switch.** [S.l.: s.n.], 2018. Disponível em: https://www.elecall.cn/en/pd.jsp?id=407#pfc=%5C%7B%5C%22groupIds%5C%22%5C%3A%5C%5B63%5C%5D%5C%2C%5C%22lid%5C%22%5C%3A1%5C%2C%5C%22sc%5C%22%5C%3A%5C%7B%5C%22key%5C%22%5C%3A%5C%22name%5C%22%5C%2C%5C%22desc%5C%22%5C%3Afalse%5C%7D%5C%7D&_jcp=3_63. Acesso em: 25 mai. 2022.
- ESPRESSIF. **ESP8266.** [S.l.: s.n.], mai. 2022. Disponível em: <https://www.espressif.com/en/products/socs/esp8266>. Acesso em: 20 mai. 2022.
- ESPRESSIF. **ESP8266EX Datasheet.** [S.l.], 2020. Disponível em: https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. Acesso em: 12 jun. 2022.
- FOLHA, de São Paulo. **Cotação das Moedas.** São Paulo, SP.: Folha, nov. 1970. Disponível em: <https://acervo.folha.com.br/>. Acesso em: 14 nov. 2018.
- FONSECA FILHO, Cléuzio. **História da computação: O Caminho do Pensamento e da Tecnologia.** Porto Alegre, RS.: EDIPUCRS, 2007. P. 205.
- GARCEZ, José Eduardo. **Códigos do sistema embarcado.** [S.l.: s.n.], 2022. Disponível em: <https://github.com/jgarcez/REDLINE>. Acesso em: 11 jul. 2022.
- GODINHO, Renato Domith. **Como foi inventado o automóvel?** [S.l.]: Abril, 2020. Disponível em:

<https://super.abril.com.br/mundo-estranho/como-foi-inventado-o-automovel/>. Acesso em: 31 mai. 2022.

IDEALI, Wagner. **Conectividade em Automação IoT**: Protocolos I2C, SPI, USB, TCP-IP entre outros. 1. ed. Rio de Janeiro, RJ.: Alta Books, 2021. P. 256.

INTEL. **Intel 8080**. [S.l.: s.n.], set. 1975. Disponível em:
<https://datasheetspdf.com/pdf/1328908/Intel/8080/1>. Acesso em: 7 jun. 2022.

INTEL. **Intel MCS-51**. [S.l.: s.n.], out. 1988. Disponível em:
<https://pdf1.alldatasheet.com/datasheet-pdf/view/107780/INTEL/8051.html>. Acesso em: 7 jun. 2022.

INTEL. **Intel MCS51 Microcontroller**. [S.l.: s.n.], fev. 1994. Disponível em:
<http://web.mit.edu/6.115/www/document/8051.pdf>. Acesso em: 7 jun. 2022.

INTEL. **Wrinkles and the MCS-51**. [S.l.: s.n.], 2022. Disponível em:
<https://www.intel.com/content/www/us/en/history/virtual-vault/articles/wrinkles-and-the-mcs-51.html>. Acesso em: 7 jun. 2022.

KENNEDY, Frances. **Cientistas italianos recriam carro de Leonardo da Vinci**. [S.l.]: BBC-Brasil, 2004. Disponível em:
https://www.bbc.com/portuguese/ciencia/story/2004/04/040426_davinci. Acesso em: 31 mai. 2022.

KUME, Honorio; PIANI, Guida; SOUZA, Carlos Frederico Bráz de. **A Política Brasileira de Importação no Período de 1987-1998**: Descrição e Avaliação. Brasília, DF.: IPEA, 2003. P. 30. Disponível em:
https://www.ipea.gov.br/portal/images/stories/PDFs/livros/Capitulo_1_politica.pdf. Acesso em: 2 jun. 2022.

MATEUS, Geraldo Robson; LOUREIRO, Antonio Alfredo F. **Introdução à Computação Móvel**. Belo Horizonte, MG.: DCC/UFMG, 1998. P. 199.

MAXIM-INTEGRATED. **DS18B20 Programmable Resolution 1-Wire Digital Thermometer**. [S.l.: s.n.], jul. 2019. Disponível em:
<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. Acesso em: 14 mai. 2022.

MAXIM-INTEGRATED. **DS3231**: Extremely Accurate I2C-IntegratedRTC/TCXO/CrystalTypical Operating Circuit. [S.l.: s.n.], 2015. Disponível em: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>. Acesso em: 23 mai. 2022.

MEHL, Edvaldo Luiz de Matos. **Do Transistor ao Microprocessador**. Curitiba, PR.: DELT/UFPR, 2013. P. 17. Disponível em:
http://www.eletr.ufpr.br/mehl/historia_transistor.pdf. Acesso em: 2 jun. 2022.

MERCEDEZ-BENZ. **1885–1886. The first automobile**. [S.l.: s.n.], 2022. Disponível em: <https://group.mercedes-benz.com/company/tradition/company-history/1885-1886.html?r=dai>. Acesso em: 31 mai. 2022.

MFCA, Model T Ford Club of America. **Model T Ford Forum**. [S.l.: s.n.], 2017. Disponível em:
<http://www.mtfca.com/discus/messages/708324/756175.html?1495407992>. Acesso em: 20 jun. 2022.

MICROSHIP. **24c32**: 32K 5.0V I2C™ Smart Serial™ EEPROM I2C. [S.l.: s.n.], 2004. Disponível em: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/21061H.pdf>. Acesso em: 23 mai. 2022.

NODEMCU. **NodeMCU Github**. [S.l.: s.n.], mai. 2022. Disponível em:
<https://github.com/nodemcu>. Acesso em: 13 mai. 2022.

OLIVEIRA, givenSérgio de. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. 1. ed. São Paulo, SP: Novatec Editora, 2017. P. 257.

OLIVEIRA, Mário Jorge Garcia de. **Desenvolvimento de uma plataforma para Internet das Coisas baseada em Wi-Fi**. Braga: EENG-Uminho, 2017. Disponível em: <https://hdl.handle.net/1822/46068>. Acesso em: 3 jun. 2022.

PAPAIOANNOU, Iannis Nicolaos. **Estudo da eletrônica embarcada automotiva e sua situação atual no Brasil**. São Paulo, SP: POLI-USP, 2005. P. 89. Disponível em: <https://www.livrosgratis.com.br/ler-livro-online-31709/estudo-da-eletronica-embarcada-automotiva-e-sua-situacao-atual-no-brasil>. Acesso em: 2 jun. 2022.

PERIM, Victor Gonçalves de Carvalho Feitosa;
NASCIMENTO, Jefferson Nataline Rosa do. **Microcontroladores e**

microprocessadores. Londrina, PR.: Editora e Distribuidora Educacional S.A, 2017. P. 232.

TANENBAUM, Andrew S.; WETHERALL, David J. **Redes de Computadores.** 5. ed. São Paulo: Pearson, 2011. P. 583.

TEXAS-INSTRUMENTS. **PCF8574:** Remote8-Bit I/O Expanderfor I2C Bus. [S.l.: s.n.], mar. 2015. Disponível em:

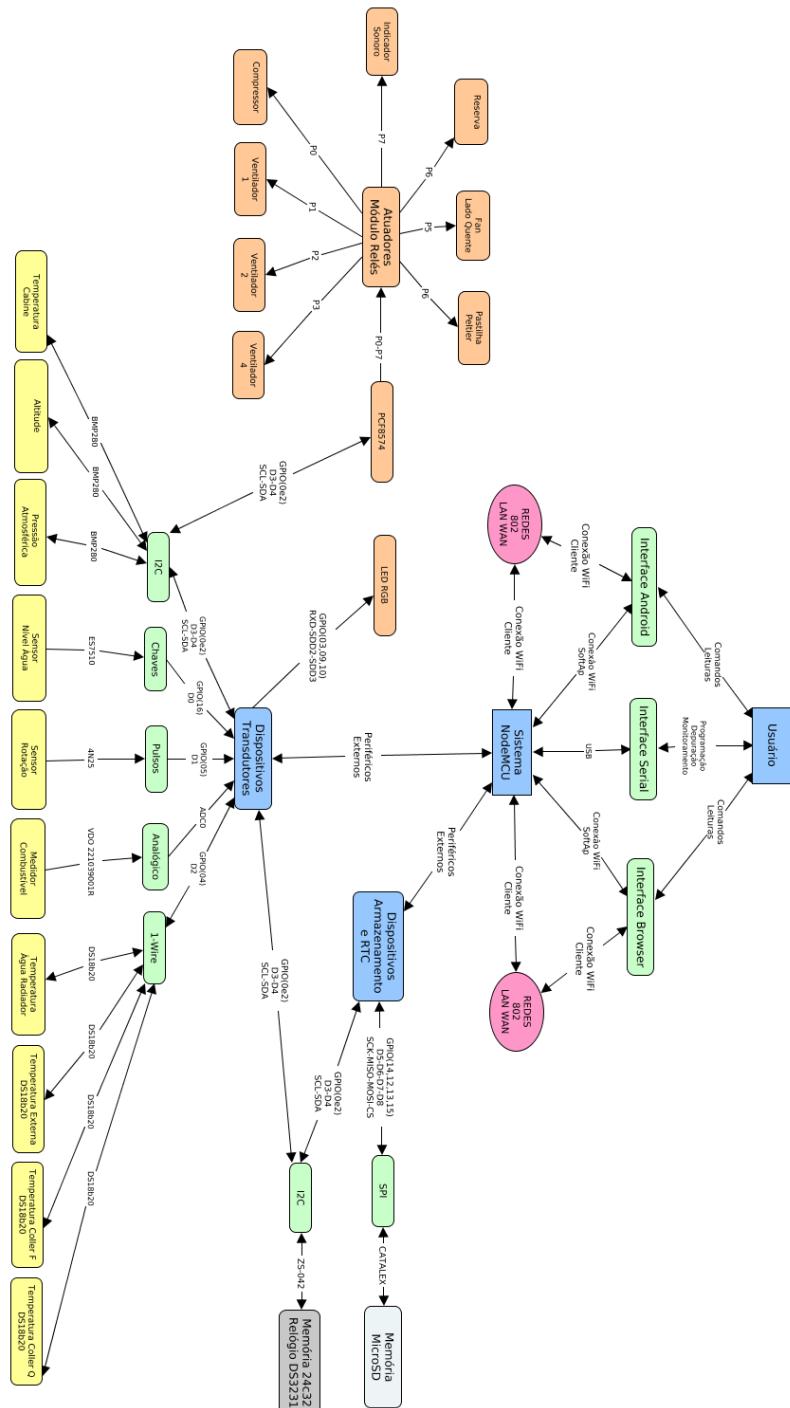
https://www.ti.com/lit/ds/symlink/pcf8574.pdf?ts=1653074501862&ref_url=https%5C%253A%5C%252F%5C%252Fwww.ti.com%5C%252Fproduct%5C%252FPCF8574%5C%253FDCM%5C%253Dyes%5C%2526utm_source%5C%253Dsupplyframe%5C%2526utm_medium%5C%253DSEP%5C2526utm_campaign%5C%253Dnot_alldatasheet%5C%2526dclid%5C%253DCLvy9Y_m7vcCFQ7qEQgd4OcNBg. Acesso em: 20 mai. 2022.

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas digitais:** princípios e aplicações. 11. ed. São Paulo, SP.: Pearson Prentice Hall, 2011. P. 842.

VENTER, Chris. **arduinoWebSockets.** [S.l.: s.n.], 2022. Disponível em: <https://github.com/Links2004/arduinoWebSockets/>. Acesso em: 20 mai. 2022.

ANEXO A – Visão geral do sistema

Figura 39 – Visão Geral do Sistema.



Fonte: Autor.