

**FACULDADE DE TECNOLOGIA SENAI “MARIANO FERRAZ”
CURSO DE PÓS-GRADUAÇÃO EM INTERNET DAS COISAS**

RAFAEL GOMES DE PAULA

TÍTULO DO TRABALHO: e subtítulo se houver

**SÃO PAULO
2021**

RAFAEL GOMES DE PAULA

TÍTULO DO TRABALHO: e subtítulo se houver

Trabalho de Conclusão de Curso apresentado à
Faculdade de Tecnologia Senai “Mariano
Ferraz” como requisito parcial para obtenção do
título de Especialista em Internet das Coisas
Orientador: Prof. Esp. CAIO VINICIUS RIBEIRO
DA SILVA

SÃO PAULO

ESTA PAGINA DEVERÁ CONTER A FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA
DA INSTITUIÇÃO.
ESTE ITEM DEVERÁ SER IMPRESSO NO VERSO DA FOLHA DE ROSTO

RAFAEL GOMES DE PAULA

TÍTULO DO TRABALHO: e subtítulo se houver

Trabalho de Conclusão de Curso apresentado à FACULDADE DE TECNOLOGIA SENAI “MARIANO FERRAZ” como requisito parcial para obtenção do título de Especialista em Internet das Coisas

Banca examinadora

Mestre André
Faculdade de Tecnologia SENAI “Mariano Ferraz”

Especialista Alexandro
Faculdade de Tecnologia SENAI “Mariano Ferraz”

Especialista Caio
Faculdade de Tecnologia SENAI “Mariano Ferraz”

São Paulo, _____ de _____ 20

DEDICATÓRIA

Ao meu pai, exemplo que sempre levarei
comigo..

AGRADECIMENTOS

Aos professores e colegas de curso, que contribuíram para a realização deste trabalho com dedicação e conhecimento.

Agradecimentos especiais à minha esposa e filha, pela paciência e carinho.

A toda equipe da Faculdade de Tecnologia Senai Mariano Ferraz.

“Pense globalmente, aja localmente”.

John Lennon

RESUMO

O resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento; deve ser composto de uma sequência de frases concisas e objetivas e não de enumeração de tópicos em um parágrafo único. A primeira frase deve ser significativa, explicando o tema principal do documento, a seguir, deve-se indicar a informação sobre a categoria do trabalho (estudo de caso, análise de situação etc.). Não se deve colocar nos resumos: símbolos, fórmulas, equações, citações de outros autores. Descrever utilizando a terceira pessoa do plural ou singular, em sua extensão o resumo deve ter aproximadamente 150 palavras.

Palavras-chave: as palavras-chave devem ser em número de 3 a 5 palavras, devem ser sugeridas e, em seguida, deve ser verificada, junto à biblioteca, a possibilidade de sua utilização.

Exemplo:

Palavras chave: Resumo. Normatização. Pesquisa.

ABSTRACT

Tradução do Resumo para o Inglês

Keywords: tradução das palavras-chave para o inglês.

LISTA DE ILUSTRAÇÕES

Figura 1 – Raspberry Pi 4 – Model B.....	XX
Figura 2 – LilyGo TTGO T-Call.....	XX
Figura 3 – Módulo SN65HVD23 CAN Bus.....	XX
Figura 4 – Módulo OLED SSD1306.....	XX
Figura 5 – Módulo RTC DS3231.....	XX

LISTA DE TABELAS

Tabela 1 – Cronograma	22
Tabela 2 – Recursos	22

LISTA DE ABREVIATURAS E SIGLAS

IoT	<i>Internet of Things</i> - Internet das Coisas
IP	<i>Internet Protocol</i> - Protocolo de Internet
TCP	<i>Transmission Control Protocol</i> - Protocolo de Controle de Transmissão
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> - Protocolo de Controle de Transmissão/Protocolo de Internet
IPv4	<i>Internet Protocol version 4</i> - Protocolo de Internet versão 4
IPv6	<i>Internet Protocol version 6</i> - Protocolo de Internet versão 6
CI	<i>Circuito Integrado</i>
SO	Sistema Operacional
ARM	<i>Advanced RISC Machine</i>
API	<i>Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
MQTT	<i>Message Queue Telemetry Transport</i>
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>

LISTA DE SÍMBOLOS

1 INTRODUÇÃO

Desde 1886, primeira patente de veículo automotor registrada, os veículos automotores vêm revolucionando a forma como nós nos locomovemos e executamos as tarefas diárias. Por muito tempo, a indústria automobilística não priorizou a implementação de tecnologia embarcada nos carros, pois os esforços eram direcionados principalmente aos motores e ao design. Em 1996, 100 anos após a primeira patente, a indústria automobilística deu o primeiro passo para a padronização da tecnologia embarcada nos veículos, a implementação do OBD - On-Board Diagnostic, sistema de autodiagnóstico do veículo. No Brasil, a implementação do OBD, se tornou obrigatório apenas em 2010. Desde então, diversas fabricantes estão implementando novas tecnologias em seus veículos que variam de sensores até sistemas inteligentes de condução semiautônoma do veículo. Apesar do desenvolvimento tecnológico na indústria automobilística ter avançado bastante desde 1886, a tecnologia utilizada nos veículos, ainda é bastante restrita ao fabricante do veículo e pouco divulgada no contexto acadêmico.

Se por um lado, a tecnologia embarcada demorou para ser prioridade na indústria automobilística, na indústria eletrônica, os microcontroladores tiveram um desenvolvimento muito rápido. O primeiro microcontrolador foi desenvolvido pela Intel em 1971 e tinha apenas 4 bits para processamento. Após 45 anos do primeiro microcontrolador, em 2016, já era possível encontrar microcontroladores com 32 bits de processamento, ou seja, 8 vezes a capacidade de processamento do primeiro microcontrolador.

Com o rápido desenvolvimento de microcontroladores e a evolução da conectividade dos dispositivos, surgiu uma nova categoria de dispositivos inteligentes, a Internet das Coisas (*Internet of Things* - IoT), baseada em dispositivos pequenos, conectados e eficientes.

O paradigma de comunicação Máquina a Máquina (*Machine to Machine* – M2M), é um conceito simples e versátil que sugere integração entre dispositivos eletrônicos, como o próprio nome sugere. A base do conceito é a comunicação entre máquinas através de um meio de comunicação.

A junção do IoT com o M2M, gera diversas possibilidades de aplicações que podem ser criadas. A Internet das Coisas se define com dispositivos e sistemas embarcados que possuem tecnologia de conectividade através da Internet e são interconectados

para a execução de determinada tarefa com seu potencial máximo, criando espaço para a inovação de serviços e automações.

Dispositivos IoT já são populares na sociedade, são facilmente encontrados em forma de assistentes virtuais, lâmpadas, sensores de clima, vestíveis, entre outros dispositivos que fazem parte do dia a dia da sociedade moderna. A diversidade de dispositivos IoT associado à internet, gera uma vasta gama de possibilidades de automações, dando oportunidade para integrações entre dispositivos que antes disso, era pouco viável, como é o caso das automações nos veículos automotores. Com a chegada da IoT, se torna viável a integração entre as tecnologias do carro com um dispositivo inteligente, que a partir desta automação, é possível realizar a leitura dos sensores, se conectando a rede CAN Bus. Nesse contexto, é possível criar análises dos dados para a obtenção de aspectos que variam da condução feita por uma motorista em uma viagem, até a customização dos módulos mecânicos do veículo.

Dado o contexto, a pesquisa realizada e descrita neste artigo, propor realizar a análise da condução veicular, que será baseada em dados coletados do veículo durante viagem, usando como parâmetro, os dados estabelecidos no manual do carro para uma boa condução, que visa estabelecer um perfil comportamental para o condutor do veículo.

1.1 Problemática e hipóteses

Em 2010, as fabricantes atuantes no Brasil, adotaram a padronização OBD II nos seus veículos. O OBD II possui diversos protocolos de comunicação que dão acesso as informações do carro em tempo real. Apesar de essa tecnologia estar estabelecida a mais de 10 anos no nosso país, o acesso a esta tecnologia e o conhecimento sobre a como extrair as informações ainda é muito escasso. Pensando nisso, a problematização deste trabalho é explorar as possibilidades de comunicação entre dispositivos IoT e os veículos que possuem essa tecnologia embarcada.

As hipóteses sobre essa comunicação seguem abaixo:

- A leitura dos dados do veículo será possível;
- A leitura dos dados do veículo não será possível;
- Os dados do veículo estarão criptografados pela montadora;
- O dispositivo poderá interagir com o sistema do veículo, alterando dados;

1.3 Objetivos da pesquisa

1.3.1 Objetivo geral

Realizar um estudo sobre a condução veicular através de dispositivos IoT e algoritmos que possam fazer previsões sobre o modo de condução do motorista.

1.3.2 Objetivos específicos

- Construir um dispositivo protótipo IoT, usando tecnologias e arquiteturas de redes em sistemas que estão presentes no contexto de Internet das Coisas;
- Coletar os dados de sensores do veículo, processar e enviar para análise de comportamento;
- Demonstrar os benefícios de forma clara na utilização de Internet das Coisas em nossa rotina.

1.4 Justificativa e delimitação da pesquisa

1.4.1 Justificativa

A tecnologia está constantemente evoluindo para dispositivos menores, mais rápidos e cada vez mais inteligentes. Através dessa evolução, surgiu a Internet das Coisas, como uma nova habilidade da internet em se conectar e coletar dados de dispositivos de onde não eram possíveis as comunicações com a rede de computadores.

Através de dispositivos pequenos, com alta capacidade de processamento, baixo consumo de energia e conexão sem fio, a Internet das Coisas abre uma porta para diversas inovações tecnológicas.

Nossa rotina está cada vez mais automatizada por dispositivos inteligentes como os celulares, wearables e até cafeteiras inteligentes, porém, veículos automotivos, apesar de possuírem tecnologia embarcada, ainda não estão inseridos no contexto da Internet das Coisas, isso foi a principal causa que motivou esse estudo a explorar novas possibilidades de automação e análises feitas a partir da coleta de dados do automóvel.

1.4.2 Delimitação e Universo da Pesquisa

O estudo tem seus limites definidos dentro de cada contexto, como é mostrado a seguir:

- Dispositivo IoT, se limitando a coleta e envio de dados do veículo, velocidade atual e rotação do motor, somente durante um percurso em que o carro está em movimento.

- Nuvem privada, se limitando a comunicação com o dispositivo IoT, com foco no recebimento e processamento dos dados coletados para análise de condução do veículo.
- Pesquisa e Estudo, se limitando a um único veículo específico utilizado.
- Comunicação IoT, se limitando a rede GSM para troca de informações

1.5 Metodologia

- Análises bibliográficas
- Compreensão sobre o tema
- Protótipo de dispositivo e rede IoT

1.6 Viabilidade da pesquisa

1.6.1 Cronograma

Tabela 1 – Cronograma de Atividades

#	Atividades	jan/21	fev/21	mar/21	abr/21	mai/21	jun/21
1.	Desenvolvimento Monografia						
2.	Desenvolvimento Hardware						
3.	Desenvolvimento Software						
4.	Desenvolvimnto Firmware						
5.	Testes						

Fonte: Autoria Própria

1.6.2 Recursos

Para o desenvolvimento da pesquisa e prototipação do dispositivo, foram necessários os itens apresentados na tabela a seguir.

Tabela 2 – Recursos

Recurso	Quantidade	Custo	Já Possuía?
Protoboard	2	R\$ 20,00	Sim
Jumps	50	R\$ 9,90	Não
LilyGo T-Call (ESP32)	1	R\$ 150,00	Não
Raspberry Pi 4	1	R\$ 450,00	Não
SDCard 64GB	1	R\$ 70,00	Não
Módulo CAN BUS SN65HVD230	1	R\$ 25,90	Não
OLED SSD1306	1	R\$ 35,00	Não
RTC 3231	1	R\$ 25,00	Não
Total	58	R\$ 785,80	

Fonte: Autoria Própria

O presente estudo será dividido em ??? capítulos sendo estes:

- 1 – Introdução incluindo apresentação dos objetivos, metodologia e delimitação do problema;
- 2 – Referencial teórico sobre definição, arquitetura e tecnologias;

2 Revisão Bibliográfica

2.1 Hardware

2.1.1 Raspberry PI 4

O Raspberry PI é um pequeno computador de baixo custo, criado pela Federação Raspberry PI Em 2006 no Reino Unido. Sua primeira versão disponível para venda, teve o preço de \$35 dólares, muito abaixo dos valores de computadores pessoais encontrados nas lojas. A ideia dos criadores era desenvolver um produto com um preço acessível, que fosse pequeno e funcional, capaz de integrar facilmente o desenvolvimento de projetos eletrônicos com software. Atualmente esta placa contém 4 grandes versões, cada versão lançada se refere a atualização do modelo anterior. A última versão disponível do Raspberry PI é a 4, ela possui apenas o modelo B, que é equipado com o processador da Broadcom, o BCM2711, Quad core Cortex-A72 (ARM v8), processamento de 64-bit SoC e frequência de 1.5GHz, três opções de escolha para a memória RAM, podendo ser de 2GB , 4GB ou 8GB LPDDR4-3200 SDRAM, Conectividade WiFi 2.4 GHz e 5.0 GHz IEEE 802.11ac, Bluetooth 5.0, BLE, Gigabit Ethernet, 2 portas USB 3.0 e 2 portas USB 2.0. O Raspberry também possui 2 saídas de vídeo micro-HDMI, com capacidade de transmissão 4K em cada monitor. O dispositivo não contém nenhum equipamento para o armazenamento de dados integrado a placa, porém, é compatível com cartões micro SD, sendo necessário que o usuário insira um cartão com o sistema operacional para que ele funcione. Na figura 1, é possível ver um exemplar do Raspberry 4 model B.

Figura 1 – Raspberry Pi 4 – Model B

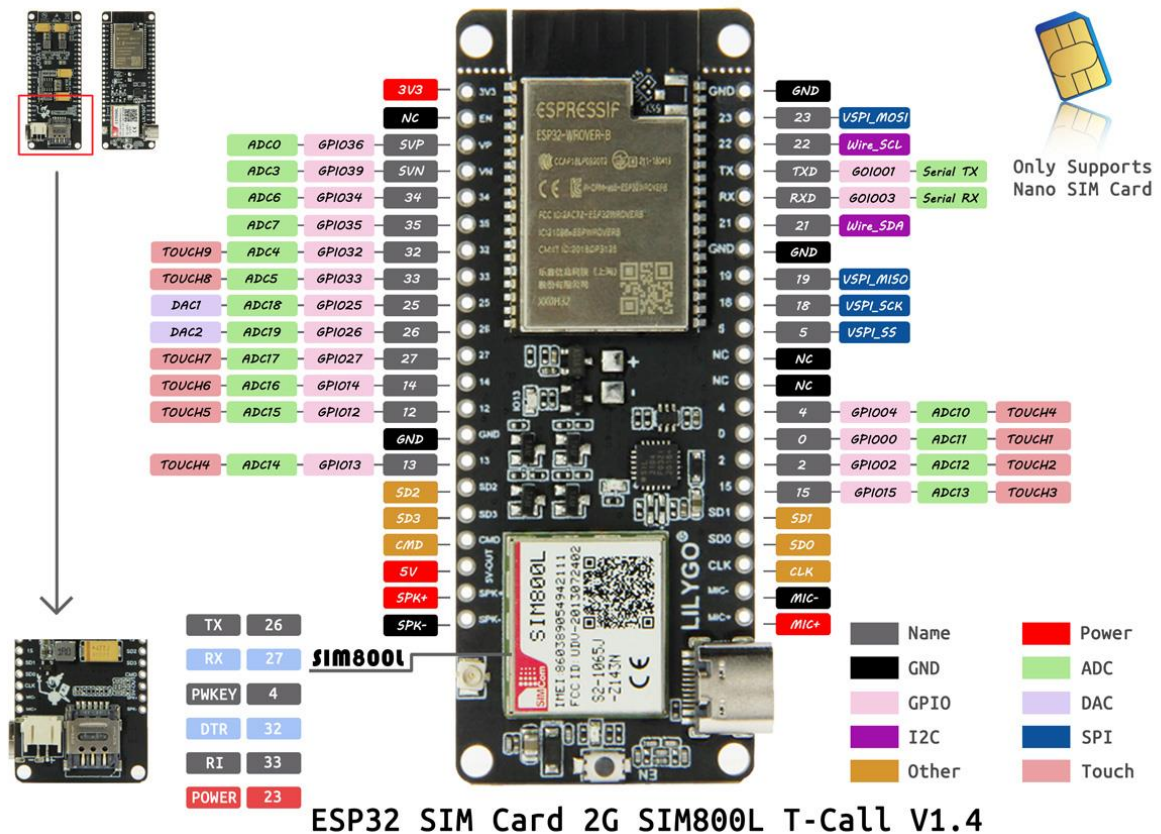


Fonte: Retirado de [1]

2.1.2 LilyGo TTGO T-Call

A LilyGo TTGO T-Call é uma placa de desenvolvimento de hardware baseado no MCU ESP32, integrado a uma placa GSM SIM800L. O ESP32 é fabricado pela Espressif e foi criado em 2016 com o intuito de facilitar e baratear o desenvolvimento de dispositivos inteligentes, além de melhorar seu antecessor, o ESP8266. Para isso o ESP32 possui as seguintes configurações: conectividade WIFI, Bluetooth e BLE, têm processamento ajustável, podendo variar de 80 MHz a 240MHz, baixo consumo de energia e memória flash suficiente para grandes aplicações. A versão utilizada do ESP32 é a WROVER-B, com 4MB de Flash SPI e 8MB de PSRAM. O SIM800L, que também integra o LilyGo, é um módulo para comunicação GSM que necessita de um chip para seu funcionamento e possui interfaces de entrada e saída de áudio que estão integradas nos pinos do LilyGo. A placa de desenvolvimento é alimentada por uma entrada USB-C e têm tensão de funcionamento entre 2.7v e 3.6v. Na figura 2 é possível ver um exemplar da placa.

Figura 2 - LilyGo TTGO T-Call

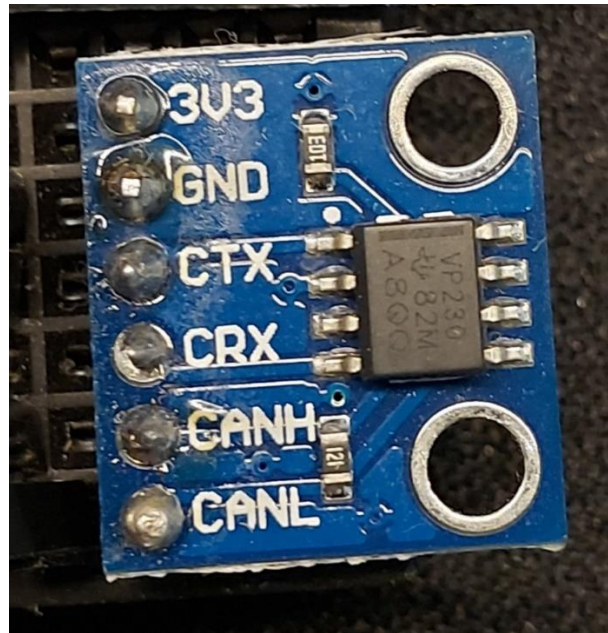


Fonte: Retirado de [2]

2.1.3 SN65HVD23

O SN65HVD23 é um transceptor para comunicação com a rede CAN (*Controller Area Network*). Esse dispositivo foi projetado para trafegar dados a taxas de até 1 megabit por segundo (Mbps), e incluem muitos recursos de proteção que fornecem aos dispositivos e a rede CAN mais robustez. Os dispositivos destinam-se ao uso em aplicações que empregam a camada física de comunicação serial CAN de acordo com o padrão ISO 11898. Na figura 3 é possível ver um exemplar do módulo.

Figura 3 – Módulo SN65HVD23 CAN Bus

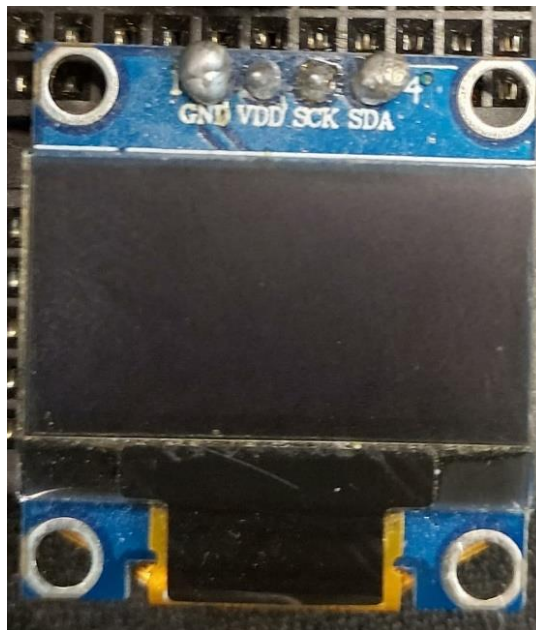


Fonte: Autoria Própria

2.1.4 SSD1306

SSD1306 é um driver CMOS OLED / PLED de chip único com controlador para emissão de luz (orgânica / polímero) em sistema de exibição gráfica por matriz de pontos de diodo. Consiste em 128 segmentos e 64 comuns. Este IC é projetado para painel OLED tipo cátodo comum. O SSD1306 incorpora controle de contraste, RAM de exibição e oscilador, o que reduz o número de componentes externos e consumo de energia. Possui controle de brilho de 256 etapas. Dados / comandos são enviados do MCU geral através da interface paralela compatível com a série 6800/8000 selecionável por hardware, I2C ou Serial. É adequado para muitas aplicações portáteis compactas, como sub-display de telefone celular, MP3 player, calculadora etc. Na figura 4 é possível ver um exemplar do visor.

Figura 4 – Módulo OLED SSD1306

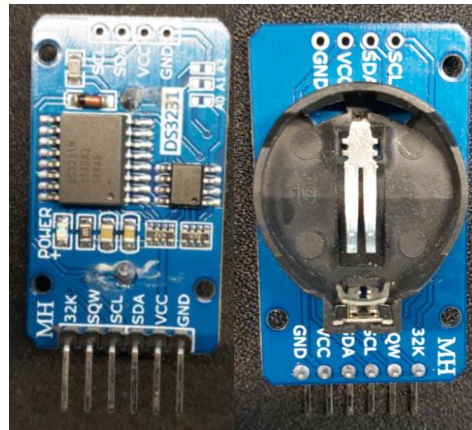


Fonte: Autoria Própria

2.1.5 RTC DS3231

O DS3231 é um relógio de tempo real I2C (RTC – *Real Time Clock*) extremamente preciso e de baixo custo, possui um oscilador de cristal com compensação de temperatura (TCXO) e cristal integrado. O dispositivo incorpora uma entrada de bateria e mantém uma cronometragem precisa quando a alimentação principal do dispositivo é interrompida. A integração do ressonador de cristal aumenta a precisão de longo prazo do dispositivo e reduz a contagem de peças em uma linha de fabricação. Na figura 5 é possível ver um exemplar do módulo.

Figura 5 – Módulo RTC DS3231



Fonte: Autoria Própria

2.2 Software

2.2.1 Linux

O Linux é um sistema operacional de código aberto desenvolvido por Linus Torvalds e lançado sua primeira versão em 1992. O desenvolvimento do sistema operacional ainda é liderado pelo seu criador, porém, conta com a ajuda de diversos programadores ao redor do mundo, uma vez que está disponível na internet para que qualquer pessoa possa fazer modificações. O objetivo do sistema Linux era inicialmente replicar e superar um sistema operacional já existente na época, o Minix, porém, com a política de código aberto, o sistema se tornou muito popular e hoje em dia é possível encontrar diversas versões deste SO rodando em computadores pessoais e servidores. O Kernel do Linux recebe diversas atualizações da comunidade de código aberto e em 2002 eles lançaram uma atualização para o Arch Linux, que tornou ele compatível com os processadores de arquitetura ARM, o que permitiu que vários dispositivos que utilizam processadores baseados nessa arquitetura pudessem agora rodar um sistema operacional Linux. A Raspberry PI Foundation desenvolveu o sistema operacional Raspberry PI OS para seus dispositivos a partir do código fonte do Debian, uma distribuição do Linux.

2.2.2 PlatformIO

PlatformIO é uma plataforma para desenvolvimento de firmwares em sistemas embarcados. A plataforma é open-source e possui seu código aberto para que outros desenvolvedores possam contribuir para seu desenvolvimento. A ferramenta tem o principal objetivo de transformar a experiência do desenvolvimento de firmware em algo simples. Teve sua primeira versão lançada em 2014 no GitHub, até o momento, já foram lançadas mais de 107 versões do software. A plataforma é

compatível com as mais modernas placas de desenvolvimento de hardware como o Arduino e o ESP32. Além de promover um ambiente amigável ao desenvolvedor, a ferramenta conta com alguns recursos de auto completar, análises estáticas de código e sugestões de melhores práticas durante o desenvolvimento.

2.2.3 Node-Red

Node-RED é uma ferramenta de programação open-source que tem como objetivo conectar dispositivos de hardware, API (*Application Programming Interface*) e serviços online de maneira inovadora e simples. Ele fornece um editor web, acessado pelo navegador, que torna fácil a experiência do usuário. A ferramenta possui diversos módulos pré-configurados que são capazes de executar funções que muitas vezes são demoradas em sua implementação, como conectores de banco de dados e exposição de end-points de API. As funções desenvolvidas na ferramenta podem ser facilmente exportadas para JSON (JavaScript Object Notation) e importadas novamente na ferramenta.

2.2.4 MongoDB

MongoDB é um banco de dados não relacional open-source, que armazena seus dados no formato de documentos, ele foi projetado para facilitar o desenvolvimento e o dimensionamento de aplicações que precisam de alta velocidade de resposta. Os dados armazenados no MongoDB estão no formato JSON, que permite criar dados com alta complexidade de informações e estrutura de dados.

2.2.5 MQTT (Protocolo)

O MQTT (*Message Queue Telemetry Transport*) foi inventado e desenvolvido pela IBM no final dos anos 90. Sua aplicação original era vincular sensores em pipelines de petróleo a satélites. Como seu nome sugere, ele é um protocolo de mensagem com suporte para a comunicação assíncrona entre as partes. Um protocolo de sistema de mensagens assíncrono desacopla o emissor e o receptor da mensagem tanto no espaço quanto no tempo e, portanto, é escalável em ambientes de rede que não são confiáveis. Apesar de seu nome, ele não tem nada a ver com filas de mensagens, na verdade, ele usa um modelo de publicação e assinatura. No final de 2014, ele se tornou oficialmente um padrão aberto OASIS (*Organization for the Advancement of Structured Information Standards*), com suporte nas linguagens de programação populares, usando diversas implementações de software livre. Com isso, muitos softwares foram lançados implementando essa tecnologia que se popularizou entre os dispositivos de IoT.

2.2.6 MQTT Mosquitto

MQTT Mosquitto é um software open-source de mensageria que implementa o protocolo MQTT versões 5.0, 3.1.1 e 3.1. O Mosquitto foi projeto para ser leve e adequado para uso em todos os dispositivos, desde microcontroladores, com o ESP32 até Servidores.

REFERÊNCIAS

- [1] Raspberry PI Foundation. In: Product Brief. **Raspberry Pi 4 Model B**. 2021. Disponível em: <https://datasheets.raspberrypi.org/rpi4/raspberry-pi-4-product-brief.pdf>. Acesso em 15 mai. 2021.
- [2] LilyGo. LILYGO® TTGO T-Call V1.4. Disponível em: http://www.lilygo.cn/prod_view.aspx?Typeld=50044&Id=1127&FId=t3:50044:3. Acesso em 15 mai. 2021.
- [3] SN65HVD2X. SN65HVD23. Disponível em: https://www.ti.com/lit/ds/symlink/sn65hvd23.pdf?ts=1621113940379&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FSN65HVD23%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dasc-null-null-GPN_EN-cpc-pf-google-ww%2526utm_content%253DSN65HVD23%2526ds_k%253DSN65HVD23%2526DCM%253Dyes%2526gclid%253DEAlaIQobChMI6YP_gNDM8AIViw6RCh0gpAwzEAYASAAEgJWV_D_BwE%2526gclidsrc%253Daw.ds. Acesso em 15 mai. 2021.