

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

JONATHAN ANTÔNIO BUCHEBUAM

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA
MONITORAMENTO DE CRIPTOMOEDAS**

CAXIAS DO SUL

2022

JONATHAN ANTÔNIO BUCHEBUAM

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA
MONITORAMENTO DE CRIPTOMOEDAS**

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Engenheiro de
Controle e Automação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Orientador: Prof. Me. Ricardo Leal
Costi

CAXIAS DO SUL

2022

JONATHAN ANTÔNIO BUCHEBUAM

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA
MONITORAMENTO DE CRIPTOMOEDAS**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Engenheiro de Controle e Automação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Aprovado em .../.../.....

BANCA EXAMINADORA

Prof. Me. Ricardo Leal Costi
Universidade de Caxias do Sul - UCS

Prof. Dra. Marilda Machado Spindola
Universidade de Caxias do Sul - UCS

Prof. Me. Angelo Zerbetto Neto
Universidade de Caxias do Sul - UCS

Este trabalho é dedicado à Deus e a minha família, os quais são a razão de eu querer ser uma pessoa melhor e me esforçar para isso.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus, por me dar a vida, força, resiliência e sabedoria para superar as dificuldades encontradas ao longo do curso.

Agradeço aos meus pais Geli Antônio Buchebuam (in memorian) e Janete Yung Guidini, que apesar das dificuldades, sempre me motivaram e sofreram comigo durante os momentos difíceis.

A minha esposa Eduarda Bertotti Buchebuam, minha filha Júlia Bertotti Buchebuam e meu filho Theo Bertotti Buchebuam, estes que representam toda a minha força e são o motivo de eu nunca pensar em desistir, e sempre melhorar como pessoa.

Agradeço ao meu professor orientador Ricardo Leal Costi, responsável pela orientação deste trabalho, e aos meus avaliadores Angelo Zerbetto Neto e Marilda Machado, os quais abdicaram de momentos particulares para me orientar, avaliar e direcionar.

“Um vencedor é um sonhador que nunca desiste”

Nelson Mandela

RESUMO

O trabalho tem como objetivo apresentar uma solução para investidores e entusiastas de criptoativos, os quais por obrigatoriedade ou até mesmo ansiedade, precisam estar constantemente verificando as cotações das moedas de interesse e das moedas presentes em suas carteiras digitais. Quando se fala de investimento de criptoativos, é essencial estar supervisionando o mercado minuto a minuto, esse acompanhamento *real time*, pode alavancar os lucros e minimizar as perdas, se analisado de maneira adequada. Uma boa análise de tendência de criptoativos, baseada em históricos e gráficos, pode aumentar exponencialmente as chances de sucesso em operações, pois este tipo de ativo tem a tendência de repetir comportamentos ao longo do tempo e com base em movimentos do mercado. A solução proposta foi o desenvolvimento de um *gadget*, o qual facilitaria esse monitoramento, pois esse produto estará constantemente atualizando o seu banco de dados e atualizando as informações na tela do equipamento de forma gráfica, dinâmica e intuitiva, essas informações estarão sempre sob o olhar do investidor, municiando-o para estar sempre analisando tendências e tomando as melhores decisões condizentes ao momento do mercado. O produto possuirá basicamente as seguintes funcionalidades: buscar na internet os dados das criptomoedas selecionadas, receber os dados por meio de uma API, transmitir os dados do módulo Wi-Fi para o microcontrolador via UART e disparar alertas, de acordo com os limites pré-definidos pelo usuário. Este conceito de produto já existe em versões para aplicativos de celular e computador, porém a inspiração em desenvolvê-lo surgiu a partir do entusiasmo do mercado em explorar diversas maneiras de acompanhar as movimentações dos ativos, onde esse dispositivo apresenta uma vantagem frente as demais soluções, pelo fato de que será um dispositivo dedicado a esta tarefa, pois estará sob a mesa de trabalho exibindo constantemente as informações ao usuário, sempre acessível ao campo de visão. O produto foi idealizado para atingir o público de investidores aficionados em criptomoedas, podendo ser utilizado como uma ferramenta a mais de trabalho.

Palavras-chave: Criptomoedas. Monitor de criptomoedas. Bitcoin. BTC. Ethereum. ETH.

ABSTRACT

The work aims to present a solution for investors and crypto enthusiasts, who out of obligation or even anxiety, need to be constantly checking the quotes of the currencies of interest and the currencies present in their digital wallets. When it comes to investing in crypto assets, it is essential to be monitoring the market minute by minute, this real-time monitoring can leverage profits and minimize losses, if analyzed properly. A good cryptocurrency trend analysis, based on histories and charts, can exponentially increase the chances of success in operations, as this type of asset has a tendency to repeat behaviors over time and based on market movements. The proposed solution would be the development of a gadget, which would facilitate this monitoring, as this product will be constantly updating its database and updating the information on the equipment screen in a graphic, dynamic and intuitive way, this information will always be under the eye of the investor, equipping him to always be analyzing trends and making the best decisions in line with the market moment. The product will basically have the following functionalities: search the internet for data on selected cryptocurrencies, receive data via an API, transmit data from the Wi-Fi module to the microcontroller via UART and trigger alerts, according to the limits predefined by the user. This product concept already exists in versions for mobile and computer applications, but the inspiration to develop it came from the market's enthusiasm to explore different ways of tracking asset movements, where this device has an advantage over other solutions. , due to the fact that it will be a device dedicated to this task, as it will be under the work table constantly displaying information to the user, always accessible to the field of vision. The product was designed to reach the audience of investors who are passionate about cryptocurrencies, and can be used as a extra work tool.

Keywords: Cryptocurrencies. Cryptocurrency Monitor. Bitcoin. BTC. Ethereum. ETH.

LISTA DE FIGURAS

Figura 1 – Representação da estrutura do equipamento	15
Figura 2 – Símbolo oficial do Bitcoin	18
Figura 3 – Tela de acompanhamento do gráfico da criptomoeda Bitcoin (BTC)	20
Figura 4 – Representação do hardware simplificado por diagrama de blocos	21
Figura 5 – <i>Representação</i> de dois modelos de transmissão: serial e paralela	23
Figura 6 – Representação de uma transmissão assíncrona em uma comunicação serial	24
Figura 7 – Representação de um diagrama de conexões de um protocolo de comunicação I2C	24
Figura 8 – Ilustração da estrutura de uma API	28
Figura 9 – Exemplo de uma aplicação desenvolvida no <i>TouchGFXDesigner</i>	31
Figura 10 – Diagrama de Blocos de Representação do Esquemático Preliminar	31
Figura 11 – Esquemático do MCU	32
Figura 12 – Esquemático da fonte de alimentação	32
Figura 13 – Dados retirados do datasheet AMS1117 - 3.3	32
Figura 14 – Esquemático do LCD	33
Figura 15 – Esquemático da memória RAM	34
Figura 16 – Representação da <i>demo board</i> que será utilizada para o desenvolvimento do protótipo	35
Figura 17 – Representação do RTOS	37
Figura 18 – <i>Workspace</i> da TouchGFX utilizada nesse projeto	38
Figura 19 – Detalhamento dos dados da tela	38
Figura 20 – Arquitetura <i>Model-View-Presenter (MVP)</i>	39
Figura 21 – Fluxograma geral do firmware	40
Figura 22 – Fluxograma Inicialização do firmware	40
Figura 23 – Diagrama de mensagens transmitidas e recebidas	41
Figura 24 – Fluxograma atualização e envio das criptomoedas	42
Figura 25 – Requisição das informações atualizadas do Bitcoin, na API do portal Mercado Bitcoin	43
Figura 26 – Requisição da cotação do dia 12/11/2022 do Bitcoin, na API do portal Mercado Bitcoin	43
Figura 27 – Fluxograma verificação e envio do alerta	44
Figura 28 – Protocolo de envio dos valores das criptomoedas	44
Figura 29 – Dados observados no monitor serial	45
Figura 30 – Applet para envio de e-mail	45
Figura 31 – E-mail de alerta	46
Figura 32 – Vista de todos os componentes	47

Figura 33 – Ambiente de desenvolvimento do monitor de criptomoedas	49
Figura 34 – Teste do monitor de criptomoedas da moeda ETH, dia 03/12/2022 hora 14:43	50
Figura 35 – Cotação ETH na <i>Exchange</i> Mercado Bitcoin, dia 03/12/2022 hora 14:42 . .	50
Figura 36 – Gráfico ETH do dia 30/11/2022 ao dia 03/12/2022	50
Figura 37 – Monitor de criptomoedas com alarme ativo, simulando dia 03/12/2022 às 14:39	51
Figura 38 – E-mail recebido informando que ocorreu um alarme	51
Figura 39 – Produto no involucrio fechado	52
Figura 40 – <i>Workspace</i> de desenvolvimento de PCB da plataforma EasyEDA	57
Figura 41 – Representação do modelo 3D da vista inferior da PCB	60
Figura 42 – <i>Workspace</i> da STM32CubeIDE	61

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
PC	Personal Computer CC] Trabalho de Conclusão de Curso
UCS	Universidade de Caxias do Sul
TCP	Transmission Control Pro
ARM	Advanced RISC Machine
FTP	File Transfer Protocol
I2C	Inter Integrated Circuit
IDE	Integrated Development Environment
IEEE	Institute Of Electrical And Electronics Engineers
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
MCU	Microcontroller Unit
NRZI	Non-Return to Zero Inverted
PDA	Personal Digital Assistant
RISC	Reduced Instruction Set Computer
RS	Recommended Standard
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memotocol
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
UWB	Ultra-Wide-Band
WPAN	Wireless Personal Area Networks
bps	<i>bits por segundo</i>
dBm	<i>decibel metro</i>
GHz	<i>Giga-Hertz</i>
GND	<i>Ground (Terra)</i>
kbps	<i>kilobits por segundo</i>

kB	<i>kilo-Byte</i>
m	<i>metro</i>
mA	<i>mili Ampère</i>
MB	<i>Mega-Byte</i>
Mbps	<i>Megabits por segundo</i>
V	<i>Volt</i>
Vdc	<i>Volt (Direct Current)</i>

LISTA DE SÍMBOLOS

A_e	Área efetiva da antena
λ	Comprimento de onda
Γ	Coefficiente de reflexão
B	Largura de faixa em que o ruído é medido em Hertz
d	Distância em metros
I	Corrente elétrica
k	Constante de Boltzmann's
K	Eficiência de irradiação
M	Varição do patamar de ruído em função da RBW
N	Condutor de neutro
NF	Figura de ruído
P	Potência
R	Resistência
E	Campo elétrico
FA	Fator da antena
Gr	Ganho de recepção
h	Altura efetiva ou comprimento efetivo de uma antena
S_i	Potência do sinal na entrada
S_o	Potência do sinal na saída
t	Tempo
V	Tensão
Z_L	Impedância da antena
Z_o	Impedância de referência (50Ω)

SUMÁRIO

1	INTRODUÇÃO	14
1.1	JUSTIFICATIVA	14
1.2	OBJETIVO GERAL	15
1.3	OBJETIVO ESPECÍFICOS	15
1.4	Restrições	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Criptomoedas	17
2.1.1	A história das criptomoedas	17
2.1.2	Transações de criptomoedas	19
2.2	Hardware	21
2.2.1	Comunicação de dados	21
2.2.2	Interface do usuário	26
2.3	<i>Application Programming Interface (API)</i>	27
2.4	<i>Layout do produto</i>	28
3	METODOLOGIA	29
3.1	Conexão <i>WIRELESS</i>	29
3.2	Processamento (Microcontrolador)	30
3.3	<i>Hardware</i>	30
3.4	<i>Firmware</i>	34
3.4.1	<i>Firmware STM429ZIT6</i>	35
3.4.2	<i>Firmware ESP8266</i>	39
3.5	Invólucro	46
4	CONSIDERAÇÕES FINAIS	48
4.1	TRABALHOS FUTUROS	51
	REFERÊNCIAS	53

1 INTRODUÇÃO

O termo criptomoeda é usado para referenciar moedas digitais, sendo o termo digital uma referência, ao fato desse tipo de moeda não possuir uma referência física como uma cédula, por exemplo uma nota de euro, real ou dólar. Ao falar de moedas digitais é necessário falar sobre sistemas de segurança no modelo de *blockchains*, com a tradução literal corrente de blocos. Esse mecanismo de segurança funciona basicamente com diversos trechos de códigos fonte criptografados, sendo que todos esses códigos são conectados entre si e cada um deles possui os registros de todas as transações realizadas com essa moeda. Através desse processo, não é possível assim que algo ou alguém, multiplique ou altere o valor de suas moedas digitais, pois essa alteração não poderia ser feita em todos os códigos fontes distribuídos, espalhados estes em diversas carteiras digitais de diversos donos.

Ao falar sobre criptomoedas é obrigatório e instintivo falar sobre o Bitcoin (BTC), pois essa criptomoeda é considerada a primeira do gênero a ser criada. A história do Bitcoin é rodeada de mistérios onde até hoje não se sabe ao certo a identidade do seu criador ou criadores. Seguindo o sucesso do Bitcoin, surgiram milhares de outras moedas digitais até os dias presentes, e esse número só aumenta com lançamentos de novas moedas quase que diariamente. Esse aumento de moedas disponíveis, também é proporcional ao volume de novos investidores que ingressaram nesse mercado, onde somente no Brasil, segundo a matéria publicada no portal de notícias E-investidor (OLIVEIRA, 2022), foi realizado um levantamento pela gestora Hashdex, enviado com exclusividade ao E-Investidor, o qual revelou que o Brasil registrou um crescimento de 1.266 por cento no número de investidores alocados em fundos e ETFs de criptoativos em 2021.

Em números, esse estudo revelou que o volume de pessoas que investem nessa categoria de ativo passou de 30mil, em 2020, para mais de 410mil no ano de 2021, demonstrando assim um crescimento impressionante, considerando que o mundo estava limitado economicamente pela pandemia, referente à COVID-19.

1.1 JUSTIFICATIVA

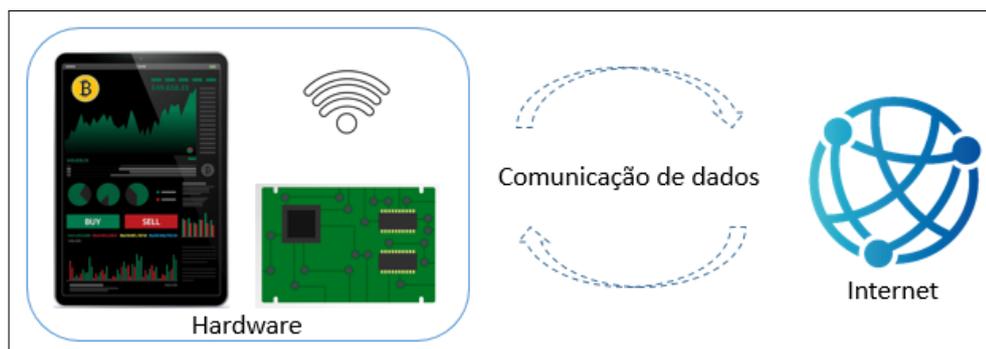
A motivação para desenvolver este trabalho surgiu ao observar o aumento do interesse das pessoas em investir e monitorar as suas moedas digitais, o qual é compartilhado pelo autor desse trabalho, pois o mesmo ingressou nesse modelo de aplicação no início de 2022 e já acumulou rendimentos muito superiores as suas demais aplicações. Considerando que para maximizar os lucros significativos e minimizar as perdas, é necessário entender o comportamento desse tipo de investimento e ter uma ferramenta exclusiva para acompanhar a movimentação do mercado, isto pode representar uma substancial diferença nos resultados ao longo do tempo.

Assim surgiu a ideia de desenvolver um sistema embarcado para ser utilizado como ferramenta de trabalho, auxiliando investidores e aficionados de criptoativos. A utilização desse *gadget* além de auxiliar diretamente nas operações como ferramenta de trabalho, impressionaria e valorizaria o ambiente, tendo em vista que o produto terá uma interface gráfica com interação via *touchscreen*.

1.2 OBJETIVO GERAL

Desenvolver um *gadget* funcional para monitorar e estudar comportamento e oscilações em criptomoedas de interesse do usuário. O dispositivo coletará as informações previamente armazenadas em um banco de dados e atualizará a interface do dispositivo com estes valores, caso a criptomoeda sob análise ultrapasse um determinado valor, pré-definido pelo usuário, o equipamento terá a função de emitir um alerta através do display e via e-mail. A Figura 1 ilustra como será o conceito do sistema e quais componentes irão compor a solução.

Figura 1 – Representação da estrutura do equipamento



Fonte: O autor (2022)

1.3 OBJETIVO ESPECÍFICOS

Para alcançar o objetivo geral será necessário atender os objetivos específicos:

- Pesquisar sobre formas de armazenamento em banco de dados;
- Pesquisar sobre fontes de consulta de criptomoedas;
- Pesquisar sobre possíveis hardwares;
- Estudo de *layout* de produto;
- Pesquisar sobre técnicas de desenvolvimento de firmware;
- Desenvolver um projeto de *hardware* que atenda aos requisitos do dispositivo;

- Desenvolver um código de *firmware* que atenda aos requisitos de funcionamento esperado para o dispositivo;
- Realizar testes para a validação do *gadget* desenvolvido.

1.4 RESTRIÇÕES

O dispositivo possuirá os seguintes limites de funcionalidade de aplicação:

- Possibilidade de cadastrar até 3 moedas digitais para o monitoramento;
- Possibilidade de cadastrar 2 alertas por criptomoeda, sendo um máximo e um mínimo;
- Cada moeda cadastrada possuirá um histórico de 20 cotações anteriores.
- Os alertas serão realizados de forma gráfica através do display e via e-mail.

2 FUNDAMENTAÇÃO TEÓRICA

O investimento em criptoativos tem atraído investidores de alto risco do mundo todo, os quais tentam aproveitar a alta volatilidade para alavancar seus ganhos.

2.1 CRIPTOMOEDAS

Conforme o autor Pedro Brandão resumiu no artigo "Criptomoeda: o Bitcoin"(BRANDÃO, 2020), pode se definir uma criptomoeda como uma espécie de dinheiro, como as demais moedas tradicionais conhecidas, porém esse dinheiro é utilizado de maneira totalmente digital, não existindo cédulas por exemplo. Essa moeda digital tem como pilar principal o fato de que qualquer pessoa ou instituição pode adquirir e movimentar suas criptomoedas de maneira livre, ou seja, de modo independente sem a intervenção de governos e/ou instituições financeiras. As transações podem acontecer sem intermediadores, através do P2P (*Peer to peer*) ou por meio de corretoras de criptoativos, também chamadas *exchanges*. Essas *Exchanges* são intermediadoras e moderadoras de operações entre os compradores e vendedores, facilitando a compra e venda, obtendo lucro através de uma cobrança de taxa por operação.

Ao falar de criptomoedas é imprescindível também falar sobre (*blockchain*), onde Lara Chaves (CHAVES, 2021), define o termo (*blockchain*) como um mecanismo que emprega um método de criptografia, o qual utiliza um conjunto de algoritmos matemáticos específicos para criar e verificar uma estrutura de dados em crescimento contínuo, na qual os dados só podem ser adicionados, nunca removidos, o que leva a uma cadeia de "blocos de transação", funcionando com uma razão de distribuição.

2.1.1 A história das criptomoedas

Ainda segundo o estudo do autor Pedro Brandão, entre 2008 e 2009, o Bitcoin surgiu, sendo a primeira moeda digital descentralizada desenvolvida por Satoshi Nakamoto, o que aparentemente se trata do nome fictício. Nakamoto, no dia 31 de outubro de 2008, publicou um *whitepaper* sobre o Bitcoin, chamado de ("*Bitcoin: a Peer-to-Peer Electronic Cash System*") (NAKAMOTO, 2008). Segundo relatos esse documento foi enviado por e-mail para os seus contatos interessados em criptografia e publicado, sendo esse o primeiro registro de alguém falando sobre uma moeda digital. Na mensagem ele comentou que estava trabalhando em um novo sistema de monetização totalmente eletrônica, baseado no conceito (*Peer to peer*), o qual funcionaria sem a intervenção de sistemas monetários tradicionais.

O *whitepaper* publicado por Nakamoto, continha instruções de como esse sistema monetário deveria funcionar. Neste manual de nove páginas, o desenvolvedor Nakamoto resumiu os quatro pilares do Bitcoin, sendo eles os listados abaixo:

- Ser uma rede peer-to-peer, evitando assim que essas moedas fossem duplicadas e enviadas mais de uma vez;
- Transações sem intermediário, como bancos;
- Garantia de anonimato dos compradores e vendedores;
- Utilização de um algoritmo conhecido como Prova de Trabalho, o qual é conhecido pelo processo de mineração.

Figura 2 – Símbolo oficial do Bitcoin



Fonte: pattymalajak (2021)

Além dos pilares, no documento o autor determinou que a oferta de BTC seria finita, reduzindo, mas não garantindo, a flutuação e dificultando a perda de valor de mercado da moeda ao longo do tempo. Estima-se que apenas 21 milhões de unidades podem ser mineradas ou criadas até o ano de 2140, tornando-o assim escasso no mercado, sendo que segundo o agregador Coingecko, já foram emitidas 18,8 milhões de unidades de moedas até o final de 2021. O primeiro bloco da (*blockchain*) só foi minerado no dia 3 de janeiro de 2009, mesmo tendo sido lançado no final de 2008. Neste arquivo, com o nome de Gênese, Nakamoto deixou escrita a seguinte mensagem criptografada “The Times 03/Jan/2009 Chancellor on brink of second bailout for banks”. Esta frase pode ser traduzida ao português como o Chanceler à beira do segundo resgate aos bancos, o que remeteria à manchete do jornal britânico The Times do dia da sua criação. Esse texto pode ser interpretado como a motivação que impulsionou Nakamoto a criar essa moeda digital.

A identidade do criador do Bitcoin ainda permanece desconhecida, algumas pessoas já vieram a público e afirmaram ser ele, entretanto ninguém conseguiu provar a identidade. Acredita-se que ele esteja vivo, baseado nas suas participações em fóruns e blogs de discussões sobre a criptomoeda, onde por exemplo existem registros de duas aparições, sendo uma em novembro de 2009, onde ele participou ativamente por aproximadamente um ano em um fórum de discussão sobre Bitcoin, postando aproximadamente 600 mensagens e a outra participação em 12 de dezembro de 2010, neste mesmo fórum discutindo sobre a segurança da rede. Evidentemente a comunidade acredita que Nakamoto ele, porém novamente não foi possível autenticar a identidade.

2.1.2 Transações de criptomoedas

Devido ao surgimento das criptomoedas, um mercado bilionário foi se desenvolvendo ao redor dessas moedas digitais. Esse movimento fomentou todo um setor da economia formando novas empresas, empregos e produtos, sem precedentes até o momento. Avaliando esse ecossistema vale destacar o surgimento das *exchanges*, também denominadas corretoras, as quais tem como principal objetivo intermediar as negociações entre compradores e vendedores, tendo como maior fonte de lucro a cobrança sobre as operações realizadas.

Em alguns casos as corretoras guardam as criptomoedas, sob custódia, daqueles investidores que não querem manter suas criptos em carteiras próprias, (*hardware* dedicado por exemplo. Conforme descrito na matéria publicada no portal de notícias InfoMoney (EXCHANGE, 2022), basicamente essas corretoras funcionam similarmente a corretoras de valores, onde o usuário precisa preencher um cadastro, confirmando sua identidade e aceitando os termos do contrato proposto pela empresa. Ao comparar uma corretora de valores com uma corretora de criptomoedas pode-se destacar algumas características:

- *Exchanges* somente oferecem moedas digitais para transações, como Dogecoin (DOGE), Solana (SOL) e XRP (XRP).
- Diferentemente das corretoras de valores, as *exchanges* não são supervisionadas pelo Banco Central e pela Comissão de Valores Mobiliários (CVM).
- Cada *exchange* funciona como uma bolsa de valores independente, ou seja, os valores variam entre as *exchanges*, o que por vezes se torna um artifício de alguns investidores, que se valem dessa diferença para fazer arbitragem (operação de compra e venda em locais diferentes para realizar lucro).
- As *exchanges* em sua grande maioria funcionam 24 horas por dia, 7 dias por semana.

A Figura 3 representa uma das telas da *Exchange* Binance, considerada a maior corretora de criptomoedas do mundo:

Figura 3 – Tela de acompanhamento do gráfico da criptomoeda Bitcoin (BTC)



Fonte: Aplicativo Binance (2022)

As *exchanges* também exigem o pagamento de uma taxa em operações de saque de ativos digitais. O preço pago pela *exchange* no momento do saque varia conforme a criptomoeda, as regras de estrutura de cada moeda (*blockchains*) e oscilações no tráfego de informação da internet. Apesar de ser o modo mais fácil de transacionar ativos digitais, essas corretoras não são o único canal para realizar operações, sendo possível também realizar operações através de fundos de investimento, ETF (*Exchange Traded Funds*) ou até mesmo P2P (*Peer to peer*).

Fundos de investimento – Funcionam como comunidades de investidores, onde se reúne os recursos de um grupo para realizar operações em conjunto, esses movimentos podem seguir um índice ou serem gerenciados pelo responsável do grupo, o qual definirá os métodos para maximizar os lucros e minimizar as perdas.

ETF – É um modelo de fundo de investimento, a diferença dos demais é que esse ativo pode ser negociado na bolsa de valores tradicional, assim como uma ação. Além disso, esse fundo normalmente segue índices similares as ações tradicionais.

P2P – Esse modelo de transação, que recebe o nome de P2P (*Peer to peer*), representa as compras e vendas sem intermediadores, ou seja, transações que acontecem diretamente entre as carteiras de compradores e vendedores. Esse modelo de transferência, foi o proposto pelo criador do Bitcoin.

2.2 HARDWARE

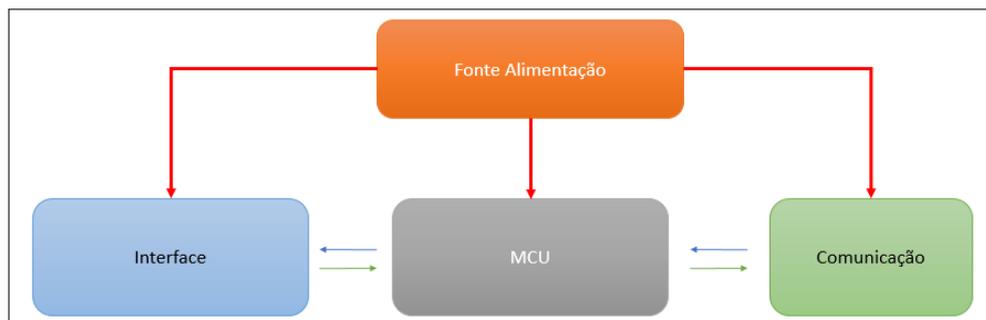
Pode-se definir o hardware como o conjunto de circuitos e componentes eletrônicos que compõe uma placa de circuito eletrônico, esse arranjo de componentes e ligações elétricas é projetado de tal modo que os componentes interligados, cada um com a sua função, atinjam o objetivo específico proposto para o projeto.

O hardware do equipamento será desenvolvido por seções e subseções, onde cada seção representará uma funcionalidade específica do sistema, as seções principais são:

- Fonte de alimentação;
- Comunicação de dados;
- Microcontrolador;
- Interface Gráfica.

A Figura 4 representa a estrutura do hardware do equipamento:

Figura 4 – Representação do hardware simplificado por diagrama de blocos



Fonte: O autor (2021)

2.2.1 Comunicação de dados

Para que exista a troca de informação entre dispositivos, é necessário existir uma comunicação e para que isto ocorra, uma série de pré-requisitos deve ser atendida, tais como velocidade de transmissão/recebimento, sincronização, formatação de bits de dados, entre outros. Segundo Edmur Canzia (CANZIAN, 2009), "a comunicação de dados tem como objeto de estudo os meios de transmissão de mensagens digitais para dispositivos externos ao circuito

originador da mensagem.", Edmur Canzia também define em outro trecho que, "A função de qualquer sistema de comunicação é fornecer a maior taxa de transmissão possível, com a menor potência e com o menor ruído possível."

Em uma comunicação de um ou mais dispositivos, é um canal para tal comunicação, sendo o caminho pelo qual a informação irá trafegar. Este caminho, pode ser um fio ou então um rádio ou outra fonte de energia radiante. Quanto aos tipos de canais, estes podem ser divididos em 3 modelos:

- Simplex: a direção de transmissão possui um único sentido (unilateral);
- *Half-Duplex*: a direção pode ser revertida, ou seja, as mensagens podem ser revertidas, porém, neste caso, nunca ao mesmo tempo;
- *Full-Duplex*: permite a troca de mensagens simultaneamente em ambas as direções.

A taxa de transmissão (*Baud Rate*) é o parâmetro de configuração da comunicação que pode ser definido como a velocidade com que os dados são compartilhados em um canal (medido em transições elétricas por segundo). Por exemplo, uma taxa de 9600bps corresponde a uma transferência de 9600 dados por segundo.

A comunicação de dados também pode ser classificada dois principais modelos: a comunicação de dados serial e a comunicação de dados paralela. Esses diferentes modelos se diferenciam basicamente pela maneira em que os dados são trafegados, por um ou mais canais de comunicação.

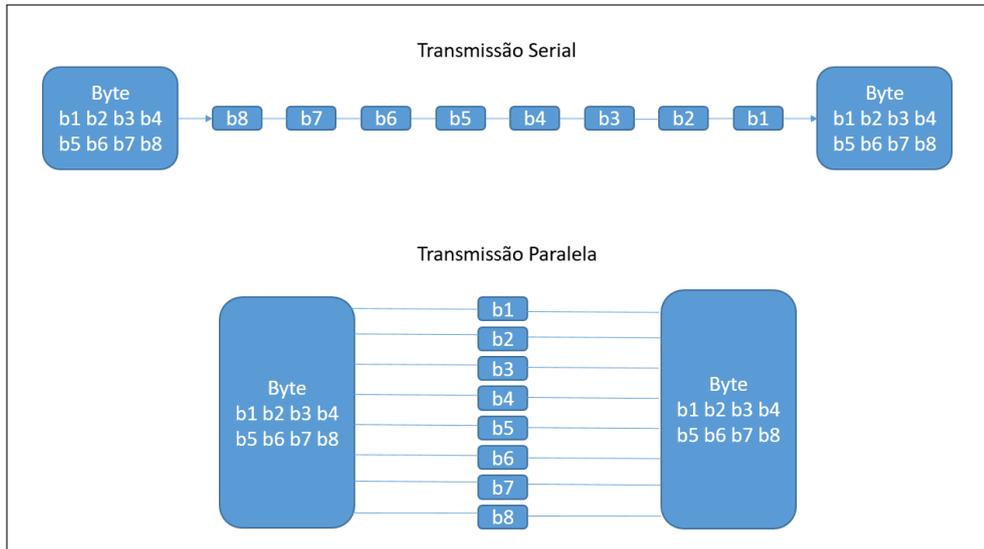
A comunicação serial é um modelo de comunicação de dados, onde os dados são transmitidos de forma sequencial bit a bit, através do canal de comunicação, por exemplo um fio. Nesse modelo de comunicação caso ocorra um erro na transmissão, somente o dado que faltou é reenviado. Nessa configuração de comunicação destacam-se os padrões RS-232, RS-485, SPI, I2C e USB.

A comunicação paralela pode ser definida como um conjunto de canais de transmissão simultânea, os quais normalmente transmitem as mensagens em um padrão de 8 bits ou de múltiplos de 8. Os bits são enviados simultaneamente e de maneira sincronizada, onde é necessário um fio para cada bit transmitido, devido a essa característica, esse modelo de comunicação normalmente demanda muitos canais de transmissão de dados, e com isso cabos e conectores maiores.

Pode-se concluir a primeira vista, que as conexões paralelas aparentam ser muito mais rápidas que as conexões em série, porém para transmitir dessa forma é necessário manter a sincronia para evitar perdas e interferências (por ter muitos cabos), e para que essa sincronia seja mantida, é necessário diminuir a taxa de transferência de dados. Como as comunicações seriais

possuem menos canais e por consequência menos interferência, pode ser aplicado frequências muito altas na transmissão, fazendo com que aumente consideravelmente a sua velocidade.

Figura 5 – Representação de dois modelos de transmissão: serial e paralela



Fonte: O autor

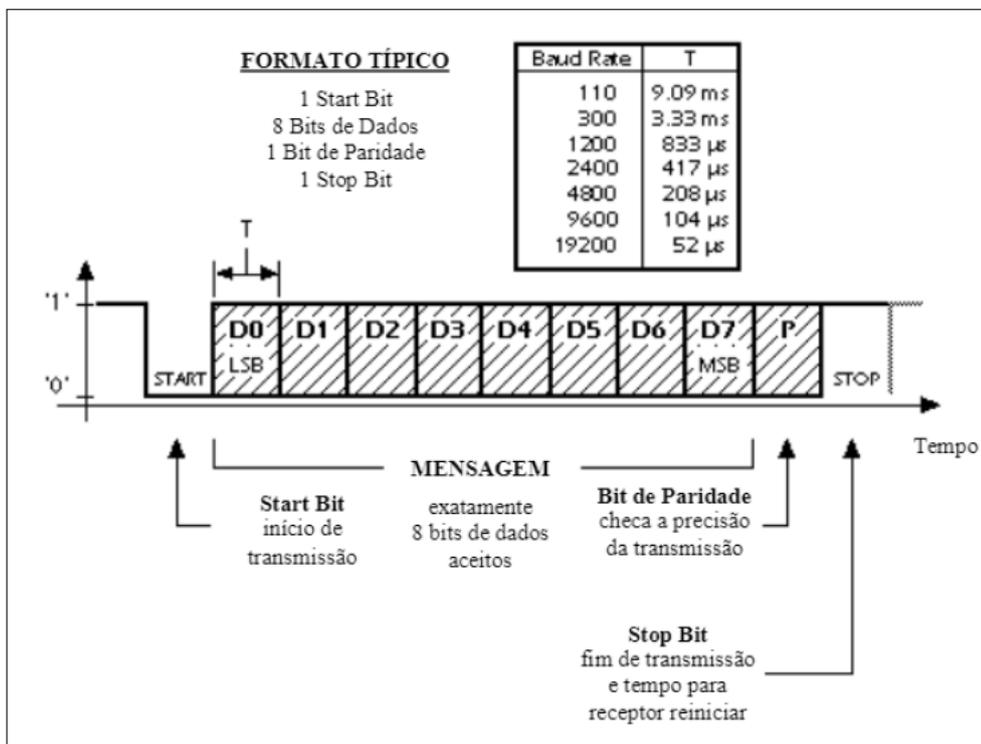
A respeito da transmissão, segundo Edmun Canzian (CANZIAN, 2009), ela pode ainda ser caracterizada como síncrona ou assíncrona. Normalmente, os dados serializados não são enviados de maneira uniforme através de um canal, mas sim pacotes regulares de informação são enviados seguidos de uma pausa. O receptor deve saber o momento adequado para ler os bits do canal, sabendo quando o pacote começa e o tempo decorrido entre bits. Quando estes itens são atendidos, o receptor está então sincronizado com o transmissor. Falhas na sincronização implicam perda de dados.

Na transmissão síncrona, são utilizados canais separados para transmissão de dados e informação de tempo (pulso de *clock*). De acordo com os pulsos recebidos no canal de tempo, o circuito avalia o momento adequado e então lê os dados e armazena os valores dos bits.

Já na transmissão assíncrona, o transmissor e o receptor devem ser previamente configurados para que a comunicação ocorra adequadamente. São utilizados sinais de *clock* interno para cada um, de forma que sejam muito similares entre si. Ao transmitir mensagens com esse modelo de transmissão, o transmissor sinaliza ao receptor que iniciará a transmissão dos dados, assim o receptor se prepara para receber e armazenar os dados.

Nos protocolos de comunicação serial mais comuns, normalmente são utilizados pacotes de dados de 10 ou 11 bits, dos quais 8 bits formam a mensagem em si. O canal quando está em repouso mantém a tensão em nível lógico alto (1), ao iniciar a comunicação o transmissor coloca a tensão do canal no nível baixo (0) para indicar a *start bit*, ou seja, sinalizando o início da mensagem. A seguir vêm os 8 bits de dados no *baud rate* especificado, um bit de paridade (opcional) e um *stop bit*. Isto pode ser melhor entendido com a ilustração da Figura 6 abaixo:

Figura 6 – Representação de uma transmissão assíncrona em uma comunicação serial

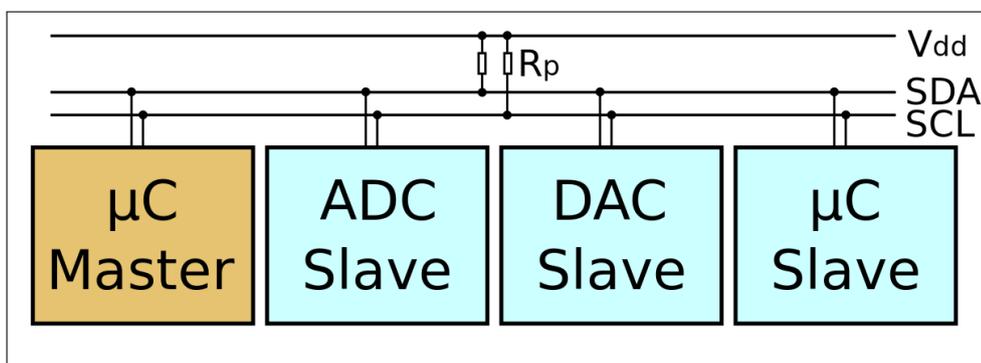


Fonte: (CANZIAN, 2009)

O protocolo de comunicação I2C (*Inter-integrated circuit*), descreve o funcionamento de um modelo de comunicação serial que acontece por meio de dois fios, o qual foi desenvolvido pela Philips no início dos anos 90. Esse modelo de comunicação somente pode ocorrer quando ambos os periféricos possuem suporte a este protocolo, seja via hardware ou via software, onde o funcionamento do protocolo é emulado bit a bit, através da técnica *bit-bang*.

O barramento I2C é composto de dois fios, SCL e SDA, onde a linha SDA é responsável pela transmissão dos dados e a linha SCL pelo clock do barramento, além da referência de alimentação positiva. A figura 7 representa um barramento de comunicação I2C.

Figura 7 – Representação de um diagrama de conexões de um protocolo de comunicação I2C



Fonte: (HTTPS://UPLOAD.WIKIMEDIA.ORG, 2022)

Para que os dispositivos se comuniquem através desse protocolo, é necessário que so-

mente um dispositivo seja configurado como mestre, sendo os demais dispositivos, configurados como escravos. A diferença principal entre um dispositivo mestre e um dispositivo escravo, é de que o dispositivo mestre será o responsável pela geração do *clock* base da transmissão dos dados.

Esse tipo de comunicação é muito utilizado em projetos microcontrolados de pequeno porte o protocolo I2C se torna uma excelente opção, pois é prático, simples e possui um baixo custo de implementação. Esse tipo de rede de comunicação possui uma fácil manutenção, pois possibilita uma modularidade de dispositivos, onde cada dispositivo pode ser protegido separadamente.

As comunicações sem fio se tratam de uma tecnologia que permite acessar redes e serviços sem a necessidade de uso de cabos, promovendo flexibilidade e mobilidade. Isto é um benefício em relação às redes cabeadas, eliminando o próprio custo do cabeamento e sendo relativamente simples de desenvolver. O cenário dos principais protocolos *wireless* de curta distância pode ser reduzido aos itens especificados nas normas IEEE 802.15.1, 802.15.3, 802.15.4 e 802.11a/b/g, tratando-se das tecnologias *Bluetooth*, UWB, *ZigBee* e Wi-Fi, respectivamente (WILLIG, 2003).

Conforme Paul H. Young (2005) descreve, o Wi-Fi (*wireless fidelity*), foi patenteado em 1999 pela Wireless Ethernet Compatibility Alliance (WECA), sendo este concebido como uma evolução direta do padrão de rede local (LAN) sem fio IEEE 802.11b. Ele define a operação ISM com taxas de dados de 5,5 Mb/s e 11Mb/s, junto com as taxas de fallback de 1Mb/s e 2Mb/s. Essa camada física usa dois esquemas de modulação: chaveamento em código complementar (CCK) e codificação convolucional em pacote binário (PBCC).

O padrão IEEE 802.11, conhecido como Wi-Fi, refere-se a uma família de especificações a respeito da tecnologia *wireless* LAN. Os autores Accardi e Dodonov (2012) citam as principais divisões como:

- 802.11a: Opera em frequência entre 5,1 e 5,8GHz e velocidade de 54 Mbps;
- 802.11b: Opera em frequência entre 2,4 e 2,485GHz e velocidade de 11 Mbps;
- 802.11g: Opera em frequência entre 2,4 e 2,485GHz, e velocidade de 54 Mbps;
- 802.11n: Opera em frequência de 2,4 e/ou 5GHz com velocidade de até 600Mbps

Biegelmeier (2015) complementa, explicando que a divisão 802.11a usa o esquema OFDM (*Orthogonal Frequency Division Multiplexing*) no lugar do FHSS (*Frequency-hopping spread spectrum* ou espectro de difusão em frequência variável) ou DSSS (*Direct Sequence Spread Spectrum* ou sequência direta de espalhamento do espectro) e é incompatível com a 802.11b, devido à sua frequência de trabalho. A divisão 802.11b (*High Rate*), por sua vez, usa somente codificação DSSS, sendo uma extensão do 802.11 que permite um funcionamento

semelhante às redes Ethernet. Tem um alcance por volta de sete vezes maior que o 802.11a. O padrão 802.11g é compatível com o 802.11b e o 802.11n é hoje em dia o mais comercializado em produtos eletrônicos.

O padrão Wi-Fi já tem suporte ao protocolo TCP/IP, altamente difundido ao redor do mundo na Internet. Outro fator importante para esta tecnologia é o fato de já existir uma infraestrutura construída, como, por exemplo, em residências, empresas, lugares públicos, etc. Isto torna fácil e versátil sua aplicação. (BIEGELMEYER, 2015)

Conforme Castañeda (2012), o *File Transfer Protocol* (FTP - Protocolo de Transferência de Arquivos) é um protocolo de comunicação para transferir arquivos entre sistemas conectados por meio de uma rede do tipo *Transmission Control Protocol* (TCP - Protocolo de Controle de Transmissão), com base numa arquitetura do tipo cliente-servidor. Um usuário executa um programa cliente que se conecta a outra máquina com um programa servidor e, de conexão estabelecida, podem trocar arquivos de diversos tipos.

Um servidor FTP trata-se de um programa que se executa na máquina servidora, normalmente conectada à internet e seu papel é permitir a troca de informações entre diferentes máquinas. De maneira geral, os servidores não ficam nas máquinas pessoais e o usuário deve usar o protocolo para estabelecer comunicação com o servidor remotamente.

As aplicações mais comuns de servidores FTP são de hospedagem na web, nas quais os clientes podem usar o serviço para fazer upload de suas páginas da web e seus arquivos correspondentes, ou como servidor de backup (backup) de arquivos importantes.

Um cliente FTP usa o protocolo para se conectar a um servidor para transferir arquivos. Os modos de conexão podem ser do tipo Ativo ou Passivo.

No modo ativo, o servidor sempre cria o canal na porta 20, já no cliente o canal será uma porta aleatória maior que 1024. O cliente envia então um comando PORT destinado ao servidor indicando o número de porta, para que o servidor possa abrir uma conexão de dados pela qual os arquivos serão transferidos. Este modo acaba gerando um problema de segurança, pois a máquina cliente estará disposta a aceitar qualquer conexão de entrada em uma porta maior que 1024. A fim de sanar este problema, desenvolveu-se o modo passivo.

No modo passivo, o cliente envia um comando PASV e o servidor indica a porta (maior que 1023) na qual o cliente deve se conectar. O cliente começa então uma conexão da próxima porta de controle para a porta do servidor especificada anteriormente.

2.2.2 Interface do usuário

Pode-se definir o termo Interface como o modo em que ocorre a comunicação entre o usuário e o sistema e utiliza os canais sensoriais do usuário para estabelecer uma comunicação com uma máquina. A comunicação deve ser feita de modo que ambos os envolvidos interpretem

o que o outro deseja comunicar, para isso é válido destacar alguns tipos de interfaces:

A interface gráfica do usuário (GUI): É o conteúdo gráfico por meio do qual se expõe uma informação do equipamento em uma tela, podendo exibir essas informações por meio de gráficos, imagens, animações, etc.

Interface de voz (VUI): Trata-se de programas capazes de reconhecer comandos de voz do usuário, sendo o exemplo mais comum as assistentes de voz como a Siri (Apple), a Alexa (Amazon) e o Google Assistente por exemplo.

Interface de texto: É utilizado principalmente no ambiente de programação de sistemas, esse tipo de interface pode ser considerado a evolução das interfaces primitivas de comando, as quais eram utilizadas nos primeiros programas de computador.

Interface natural: É chamada assim o tipo de interface que identificam e interpretam expressões faciais e movimentos naturais dos seres humanos, esse tipo de interface pode ser observado em jogos de vídeo game que possuem câmeras para interação.

Interface cérebro-computador: Esse tipo de interface pode ser considerado a mais inovadora que existe até o momento, pois é a interface utilizada para interpretar as ondas cerebrais e controlar as respectivas próteses biônicas.

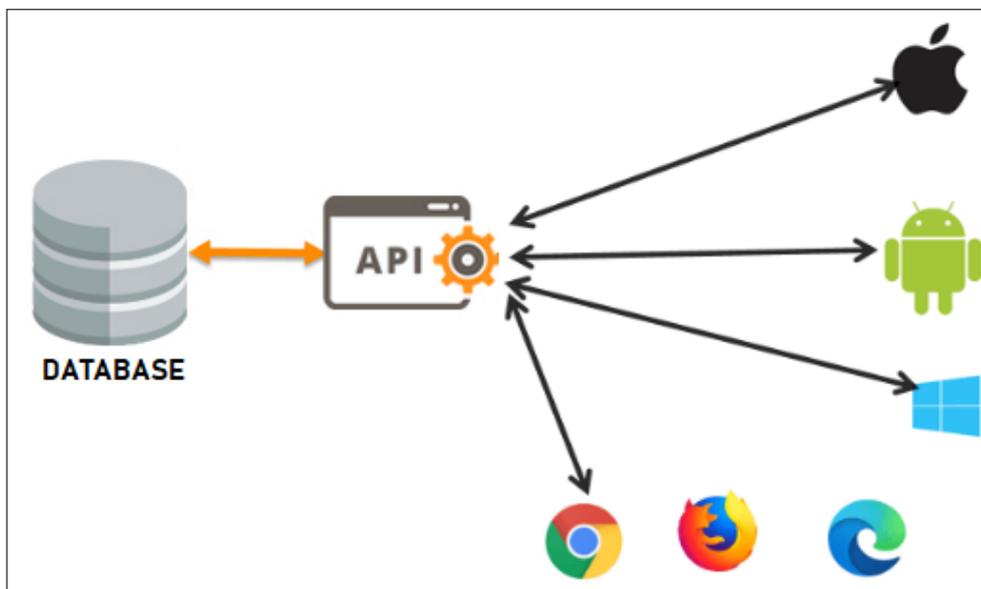
2.3 *APPLICATION PROGRAMMING INTERFACE (API)*

A sigla API é derivada da expressão *Application Programming Interface*, a qual pode ser traduzida literalmente como uma interface de programação de aplicação, segundo o portal de informações da IBM (EDUCATION, 2022), uma API permite que as empresas abram os dados e as funcionalidade de seus aplicativos para desenvolvedores externos. Isso permite que aplicações se comuniquem e desfrutem das funcionalidades uns dos outros, por meio de uma interface documentada.

Ainda segundo informações disponibilizadas no portal da IBM, uma API funciona seguindo os seguintes passos:

- 1. Um aplicativo cliente inicia uma chamada de API para recuperar informações, também conhecida como solicitação. Essa solicitação é processada de um aplicativo para o servidor da Web, por meio do URI (Uniform Resource Identifier) da API e inclui um verbo de solicitação, cabeçalhos e, às vezes, um corpo de solicitação.
- 2. Após receber uma solicitação válida, a API faz uma chamada para o programa externo ou servidor web.
- 3. O servidor envia uma resposta à API com as informações solicitadas.
- 4. A API transfere os dados para o aplicativo solicitante inicial.

Figura 8 – Ilustração da estrutura de uma API



Fonte: O autor

Em resumo, essa interface é uma ferramenta desenvolvida a fim de facilitar o trabalho dos desenvolvedores, pois por meio desta interface é possível acessar e interagir de maneira muito mais fácil com outras aplicações, sendo essa interação guiada por um protocolo já estabelecido.

2.4 LAYOUT DO PRODUTO

De modo geral o encapsulamento deve proteger e valorizar o produto, ao ponto de que o circuito eletrônico não fique exposto ao cliente, possibilitando assim que o cliente usufrua do produto por um longo período. O material mais utilizado para esse tipo de invólucro é o ABS, pois é um material com preço relativamente baixo, leve e fácil de moldar, tendo ótimas propriedades, sendo resistente ao impacto, à tração e à abrasão.

O invólucro e o seu material construtivo deverão ser definidos conforme a aplicação e o ambiente em que o dispositivo será instalado ou será utilizado, podendo assim cumprir com a função desde a proteção física do produto, como até mesmo a valorização do produto, utilizando um invólucro transparente, por exemplo.

3 METODOLOGIA

Este capítulo do trabalho expõe as etapas e as decisões executadas a respeito do desenvolvimento do projeto, afim de que o mesmo atenda a necessidade descrita no capítulo 1.1, bem como a escolha dos componentes a serem utilizados e suas especificações.

A metodologia para o desenvolvimento deste dispositivo, pode ser organizada em 5 etapas:

- Etapa 1: estabelecer uma base de estudo referencial teórico e literário sobre os itens relacionados ao dispositivo, que são eles: criptomoedas, hardware, firmware, armazenamento em banco de dados e *layout*;
- Etapa 2: desenvolver um sistema para gerenciar e armazenar os dados das criptomoedas no banco de dados;
- Etapa 3: desenvolver um hardware capaz de atender as especificações do equipamento;
- Etapa 4: desenvolver um firmware capaz de atender as especificações do equipamento;
- Etapa 5: desenvolver um *layout* de encapsulamento para o equipamento.

Para desenvolver uma solução que atenda à demanda vista neste trabalho, é possível aplicar os conceitos revisados em um projeto de dispositivo eletrônico.

3.1 CONEXÃO WIRELESS

Para acessar os arquivos desejados no servidor de dados, será utilizada uma comunicação *wireless*, pois o dispositivo será desenvolvido com a premissa de ser móvel, não sendo necessário assim, uma conexão via cabo Ethernet. Será adotado como padrão de comunicação sem fio, a rede Wi-Fi, pois esta já está presente na maioria dos lugares em que o dispositivo será utilizado, como escritórios e ambientes residenciais.

Para executar a comunicação Wi-Fi entre o dispositivo e o servidor, pode ser utilizado módulos dedicados a esta aplicação como o ESP32, o SP806 e o ESP8266. Neste caso será utilizado um módulo ESP8266 (Espressif Systems), pois é um dispositivo que o autor já possui e já o utilizou em outros projetos pessoais.

Esse acessório possui uma ampla biblioteca já desenvolvida pela comunidade e baixo custo de implementação, quando comparado ao desenvolvimento de um módulo Wi-Fi desde o início. Além do fácil acesso a esse módulo, este dispositivo possui uma comunidade ativa e um grande acervo de bibliotecas já desenvolvidas, por ter sido desenvolvido na base do Arduino.

O módulo opera conforme o padrão IEEE 802.11 b/g/n, entre 2,4 e 2,5GHz, com velocidades de até 72,2 Mbps e poder de transmissão de 14dBm (@802.11n). Esse acessório possui pinos de UART (*Universal asynchronous receiver/transmitter*) para comunicação serial e esse será o meio de comunicação em que o microcontrolador utilizará, para enviar/receber os dados ao servidor.

3.2 PROCESSAMENTO (MICROCONTROLADOR)

Durante o desenvolvimento de dispositivos eletrônicos alguns obstáculos como tempo e falta de conhecimento normalmente podem atrasar e atrapalhar o andamento do projeto, pensando nisto os fabricantes de componentes eletrônicos habitualmente fornecem aos clientes *kits* de desenvolvimentos. Este tipo de ferramenta já fornece uma solução funcional para facilitar a implementação e teste desse componente, agilizando assim o processo de validação.

Para o desenvolvimento desse projeto poderia ser utilizado diversos modelos de microcontroladores de diversos fabricantes, como por exemplo: PIC, Arduino, STM32, ST, entre outros.

O microcontrolador será usado para gerenciar a interface *touch*, atualizando telas e processando interações, gerenciar o recebimento e envio de dados por meio da comunicação serial com o módulo Wi-Fi. O *chip* STM32F429ZIT6U é compatível com as tarefas necessárias, esse foi o componente escolhido para o projeto, pois possui um controlador dedicado ao controle e gerenciamento do LCD, *touchscreen* e memória RAM externa, além de possuir os demais periféricos I2C, SPI, UART e USART, que podem ser utilizados para a comunicação com o módulo Wi-Fi.

Esse *chip* também possui 2MB de memória Flash e 256+4kB de SRAM e sua alimentação é de 3,3V. Possui 4 portas UART, que podem servir de canal para uma comunicação serial a ser realizada entre microcontrolador e módulo ESP8266. Os demais recursos disponíveis no microcontrolador STM32F429ZIT6U podem ser visualizados no Anexo A (que ilustra a capa do Datasheet deste componente).

Além disto, o fabricante deste item também disponibiliza um software gratuito para o desenvolvimento da interface gráfica, o *TouchGFXDesigner*, o qual poderá ser usado para o desenvolvimento das mais diversas aplicações, as quais envolvam elementos de LCD e *touch*, a Figura 9 abaixo representa uma aplicação desenvolvida com essa interface:

3.3 HARDWARE

Como o desenvolvimento inicial se dará em uma etapa com o uso do *kit* de desenvolvimento supracitado e o mesmo já possui um circuito com diversos recursos de hardware, somente será necessário adicionar uma conexão para o funcionamento do módulo ESP8266. Entretanto, é possível pensar em um circuito que leve em consideração somente os recursos necessários. A

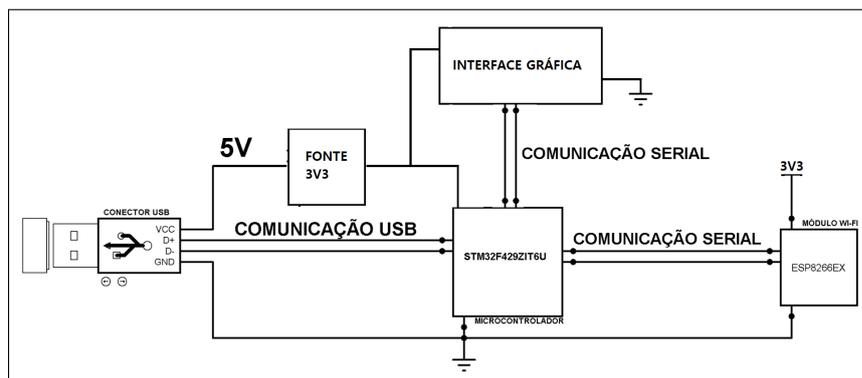
Figura 9 – Exemplo de uma aplicação desenvolvida no *TouchGFXDesigner*



Fonte: (STMICROELETRONICS, 2022a)

partir das definições realizadas, pode-se montar um simples diagrama com os circuitos básicos, que é ilustrado na Figura 10.

Figura 10 – Diagrama de Blocos de Representação do Esquemático Preliminar



Fonte: O autor (2022)

O esquemático e a PCB do produto foram desenvolvidos através da plataforma online EasyEDA, essa plataforma possui uma vasta biblioteca de componentes, e permite a criação de esquemático e placas de circuito impresso de forma gratuita. A Figura do Anexo B ?? representa o ambiente de desenvolvimento comentado acima.

O esquemático do MCU, apresentado na Figura 11, foi desenvolvido levando em consideração as informações e recomendações presentes no *datasheet* do fornecedor, considerando as necessidades da aplicação em questão.

O projeto do dispositivo prevê que o mesmo funcione conectado à uma conexão USB, essa conexão disponibilizará 5V e GND para a alimentação do circuito. Como o restante do circuito é alimentado a uma base de tensão de 3,3V, será necessário rebaixar a tensão à este nível.

O esquemático da fonte de alimentação, apresentado na Figura 12, foi desenvolvido levando em consideração as informações e recomendações presentes no *datasheet* do fornecedor.

Figura 11 – Esquemático do MCU

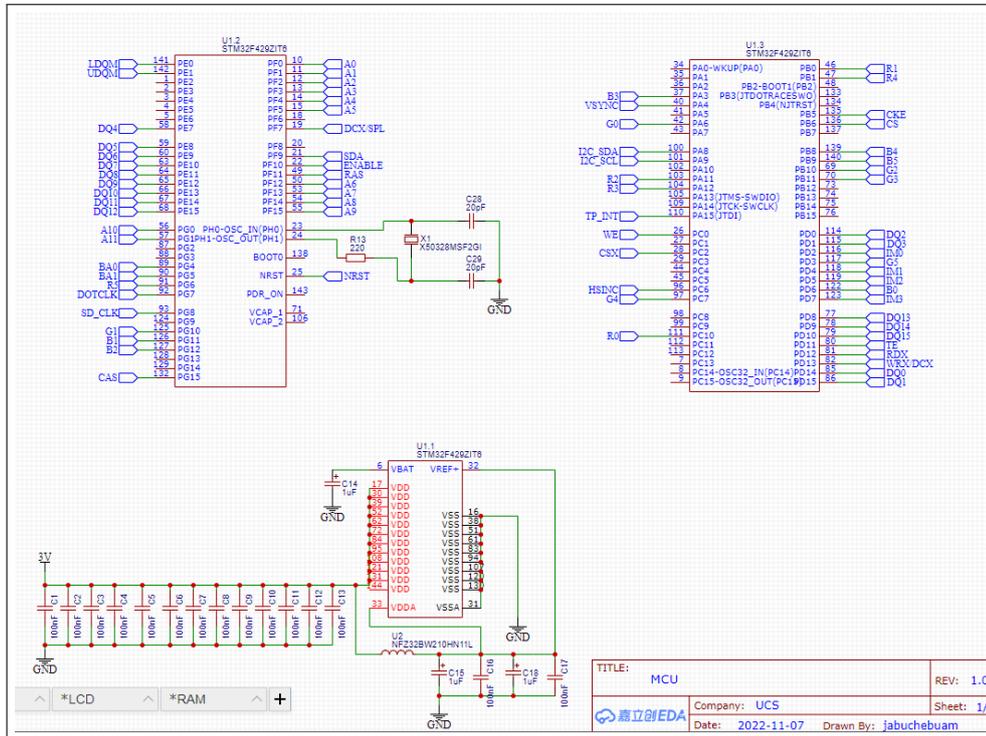
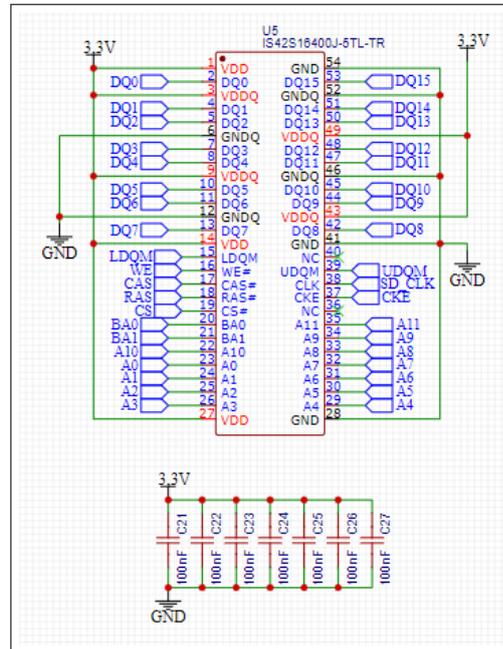


Figura 15 – Esquemático da memória RAM



Fonte: O autor (2022)

Assim como o esquemático, a PCB também foi desenvolvida pelo ambiente da plataforma EasyEDA, conforme é possível observar no Anexo C. Ao desenvolver a PCB, foi optado em manter a placa com as menores dimensões possíveis, assim refletiria no tamanho final do produto. Em decorrência deste fator, foi necessário utilizar 8 *layers* de sinais, para que fosse possível realizar todas as conexões previstas no esquemático do produto. As informações da PCB podem ser observadas no Anexo D.

Ao finalizar o posicionamento dos componentes e conectar todos as conexões elétricas, foi possível observar uma prévia de como a dispositivo ficaria, após montado todos os componentes na PCB. As representações em 3D da vista superior e inferior, podem ser observadas no Anexo E e no Anexo F respectivamente.

3.4 FIRMWARE

O *firmware* do microcontrolador STMF429ZIT6 e do módulo Wi-Fi 8266 foram ambos desenvolvidos em linguagem C, no qual para o microcontrolador foi utilizado a IDE (*Integrated Development Environment*) STM32CubeIDE Versão:1.10.1 e para o módulo Wi-Fi foi utilizado a IDE Arduino Versão 1.8.19.

Estas IDEs, além de servir como editor e facilitar o desenvolvimento do código com seus recursos, também possuem compiladores integrados, que transformam o código em linguagem C, para um arquivo hexadecimal que possa ser gravado no componente.

Como os dois firmwares, apesar de serem desenvolvidos na mesma linguagem de pro-

gramação, foram desenvolvidos em plataformas de desenvolvimento diferentes, isto também implicou em estruturas distintas de desenvolvimento, as quais serão detalhadas a seguir.

3.4.1 *Firmware STM32F429ZIT6*

Ao desenvolver projetos de hardware é comum que os desenvolvedores utilizem placas de demonstração para a economia de tempo e de dinheiro em seus projetos. Esses dispositivos são fornecidos pelos fabricantes, com um hardware já validado e com bibliotecas demonstrativas do hardware, justamente para facilitar o desenvolvimento e apresentar o seu produto ao seu cliente, neste caso, o desenvolvedor. Devido à disponibilidade do recurso do autor, será utilizada a *demoboard* STM32F429 DISCOVERY da STMicroelectronics, esse modelo de placa de demonstração possui todos os requisitos necessários para a aplicação, sendo assim cumprirá com a finalidade de auxiliar no desenvolvimento do projeto e apresentar uma solução funcional do dispositivo. A Figura 16 mostra o *kit* de desenvolvimento utilizado.

Figura 16 – Representação da *demo board* que será utilizada para o desenvolvimento do protótipo



Fonte: (STMICROELETRONICS, 2022b)

Esta placa tem como elemento foco o microcontrolador STM32F429ZIT6U e possui recursos de *hardwares* periféricos como: conexão USB, LCD de 2.4", sensor de movimento, botões de interação com o usuário, LEDs de indicação, memória SDRAM de 64 Mbits, conec-

tores ligados aos pinos do microcontrolador, possibilidade de alimentação do circuito por USB, ferramenta de gravação de firmware ou fonte externa, entre outros.

Para detalhar o desenvolvimento do firmware do MCU, é possível considerar uma estrutura de 3 partes: camada de abstração de hardware (HAL – *Hardware Abstraction Layer*), drivers e aplicação.

A camada HAL é responsável por acessar os periféricos do microcontrolador, sendo assim foram utilizadas as bibliotecas disponibilizadas pelo próprio fabricante do MCU (*ST-Microelectronics*). Essas bibliotecas são desenvolvidas para auxiliar no processo de desenvolvimento, e possuem funções específicas para configurar e ajustar esses periféricos, de acordo com a aplicação. Os principais periféricos do MCU utilizados nesse desenvolvimento foram: UART, para a comunicação com o módulo Wi-Fi, TIMER para organizar a execução operações no *firmware*, I2C para a comunicação com o driver do *touch* e LCD para a comunicação com a memória RAM externa e controlador do LCD. Todas as bibliotecas da camada HAL acessam diretamente os registradores do microcontrolador.

A camada de *drivers* é responsável por facilitar o acesso e manipulação de uma funcionalidade já pré-configurada pelo fabricante. Por exemplo, no driver da UART ao enviar uma mensagem, somente é necessário informar a uma função a mensagem a ser enviada, assim a função que compõe o driver da UART se encarregará de manipular a camada HAL, afim de organizar os registradores do MCU a refletirem no pino de TX, a mensagem a ser enviada.

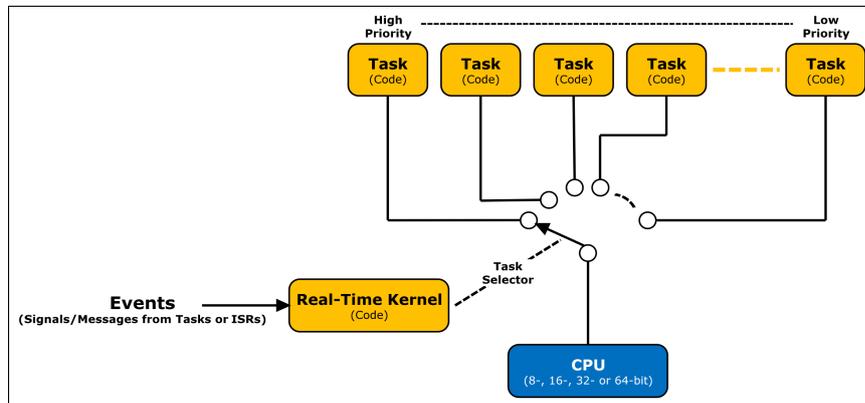
A camada de aplicação é o local onde são desenvolvidas todas as lógicas do produto, como por exemplo, a manipulação gráfica dos componentes da tela, protocolos de comunicação e algoritmos de controle. No projeto de firmware deste dispositivo, a aplicação foi dividida em bibliotecas isoladas. O objetivo é fazer com que estas bibliotecas tenham acesso aos drivers, que por sua vez, acessam a camada HAL, proporcionando ao programador uma melhor compreensão da estrutura do código. Esta divisão em camadas possibilita a utilização de bibliotecas já desenvolvidas em outros projetos, sem grandes reestruturações no código fonte. É possível visualizar, através da Figura 42, a tela do ambiente de desenvolvimento STMCubeIDE com algumas das bibliotecas e *drivers* utilizados no projeto de firmware.

Além de entender a camada de aplicação, é importante descrever como funciona a manipulação de interfaces gráficas controladas por microcontroladores, trabalhando em conjunto ao acesso à periféricos. Para que seja possível alcançar um bom resultado, neste tipo de aplicação, é necessário implementar o modo de trabalho RTOS, ou seja, dividir o processamento da aplicação em tarefas e processar essas tarefas em paralelo, diferentemente do conceito habitual de funcionalidade, o qual executa as tarefas de forma serial.

Foi necessário utilizar esse modo de trabalho no firmware, pois todas as funcionalidades do MCU, sendo as principais delas: Touch, LCD e comunicação, devem responder instantaneamente e independente das outras tarefas. Por exemplo, caso uma mensagem esteja sendo lida

pela UART e ocorrer uma interação via *touch* do usuário, tanto a mensagem da UART deve ser armazenada e processada, quanto a interação do usuário deve ser avaliada e processada. A Figura 17 é uma representação de como uma estrutura RTOS processa as tarefas.

Figura 17 – Representação do RTOS



Fonte: ([HTTPS://THECUSTOMIZEWINDOWS.COM](https://thecustomizewindows.com), 2022)

A comunicação com o módulo Wi-Fi será realizada por meio dos pinos PA9 e PA10 do MCU (UART1), os quais podem ser configurados para funcionar de acordo com essa função. Para que a comunicação ocorra adequadamente, entre o módulo Wi-Fi e o monitor de criptomoedas, o pino TX do MCU deverá ser conectado no pino RX do módulo Wi-Fi, o pino RX do MCU deverá ser conectado ao pino TX do módulo Wi-Fi e os GNDs deverão estar conectados, assim a conexão poderá ser estabelecida adequadamente.

Nesta aplicação, ambos os dispositivos foram configurados para funcionar com *Baud Rate* de 9600, essa velocidade foi escolhida, pois é uma velocidade relativamente baixa o que reduz o percentual de falhas na comunicação e essa velocidade é suficiente para o pleno funcionamento da operação.

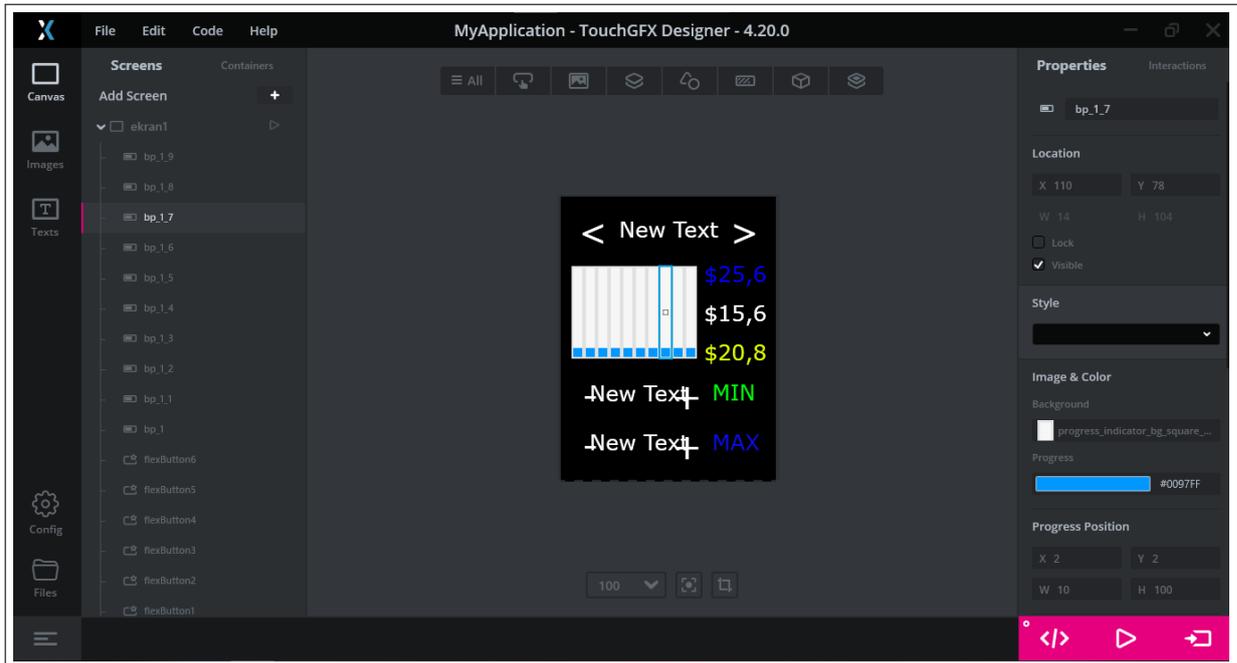
O recebimento e o envio de dados pela UART foram realizados utilizando a biblioteca `stm324xxhal-uart.h`, a qual é disponibilizada pelo fabricante do *chip*, o uso dessas bibliotecas do fabricante, agiliza o processo de desenvolvimento e facilita a manipulação dos drivers.

A interface gráfica do monitor de criptomoedas, foi desenvolvida por meio da ferramenta *TouchGFX Designer*, representada na Figura 18. Essa é uma ferramenta gratuita fornecida pelo mesmo fabricante do MCU, e tem como principal objetivo facilitar a criação de interfaces gráficas e agilizar o processo de desenvolvimento de soluções.

Na tela desta IDE é possível adicionar elementos de interação como botões, entradas numéricas, *sliders*, entre outros. Por meio dessa interface também é possível adicionar elementos de informação como imagens, saídas numéricas, saídas de texto, barras de progresso, elementos geométricos.

Para essa aplicação foram incluídos elementos de saídas de texto, saídas numéricas, botões e barras de progresso, onde todos esses elementos foram manipulados durante o fun-

Figura 18 – *Workspace* da TouchGFX utilizada nesse projeto

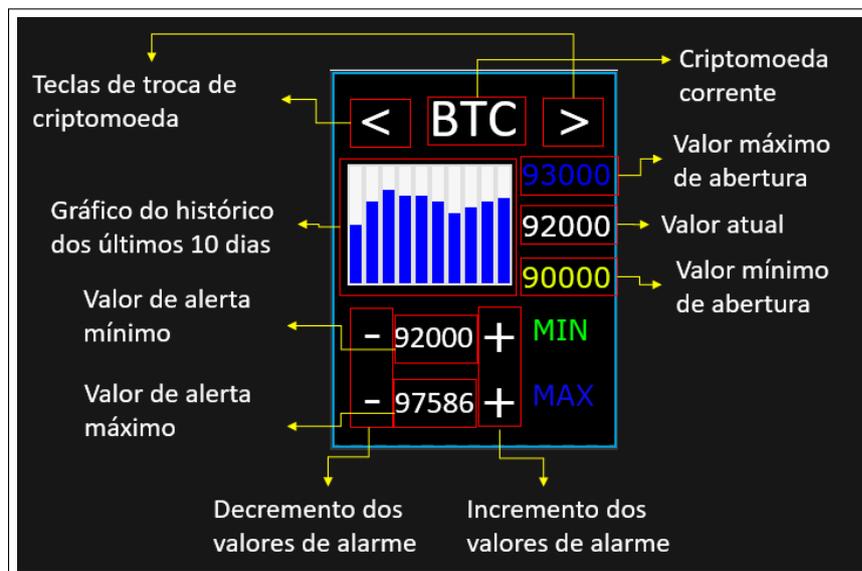


Fonte: O autor (2022)

cionamento do firmware, por meio de comandos. Nas situações que envolvem interações via *touch*, foram criados eventos, ou seja, quando existir a interação do usuário com aquele determinado elemento, ocorrerá um evento. Assim foi possível interpretar a interação do usuário e atualizar o algoritmo de acordo com essa interação.

Esses componentes foram adicionados e posicionados seguindo a ideia de design desenvolvida pelo autor, esta detalhada na Figura 19 abaixo:

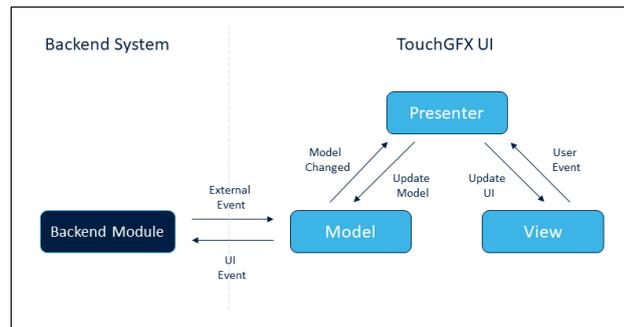
Figura 19 – Detalhamento dos dados da tela



Fonte: O autor (2022)

Ao utilizar a plataforma TouchGFXDesigner, foi necessário entender o modo em que a interface gráfica e os dados são manipulados, considerando o desenvolvimento do firmware em C. Ao desenvolver uma interface gráfica por meio desta ferramenta, arquivos são adicionados a pasta raiz do firmware, e é através destes arquivos que se manipula os dados da interface gráfica, sendo estes: *Model*, *View* e *Presenter*, os quais podem ser representados pela Figura 20.

Figura 20 – Arquitetura *Model-View-Presenter* (MVP)



Fonte: ([HTTPS://SUPPORT.TOUCHGFX.COM](https://support.touchgfx.com), 2022)

Ao considerar a arquitetura MVP, sempre que uma informação é atualizada no *Backend*, no caso dessa aplicação o laço principal, para que essa informação seja atualizada na tela, é necessário que passe por todas as camadas, primeiro pelo *Model*, depois pelo *Presenter* e por fim o *View*, que é a camada que atualizará a informação na tela. Esse processo ocorrerá da mesma forma, porém no caminho inverso, caso ocorra um evento por meio do *touch*, como um click, ou a alteração de uma variável, por exemplo.

Nessa aplicação, as informações das criptomoedas são recebidas pela UART, filtradas no *Backend* e alocadas em variáveis, as quais serão transferidas entre as camadas até que por fim, a tela seja atualizada com o novo valor.

3.4.2 *Firmware* ESP8266

Para o módulo Wi-Fi, somente a área de aplicação pode ser modificada, a camada de abstração de hardware (HAL – *Hardware Abstraction Layer*) e drivers é modificada através das bibliotecas adicionadas na aplicação. Sendo assim, caso seja necessário acessar um driver ou alterar algo nos registradores internos, terá de ser executado através da aplicação. A plataforma para desenvolver o firmware está representada no Anexo H.

O desenvolvimento do firmware do módulo Wi-Fi, pode ser separado em etapas e seguirá uma sequência de instruções para alcançar o funcionamento desejado. Para demonstrar as etapas e o fluxo de funcionamento, foi desenvolvido um fluxograma geral, o qual pode ser observado na Figura 21.

A primeira etapa do firmware é a inicialização, nessa etapa as bibliotecas são carregadas, as configurações das bibliotecas são realizadas e as funções e definições são declaradas. Para

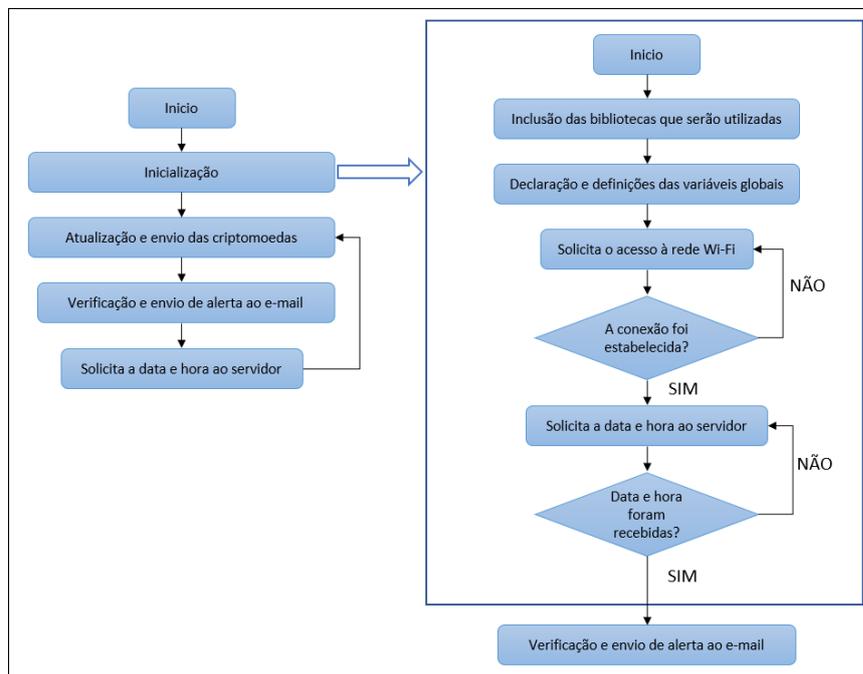
Figura 21 – Fluxograma geral do firmware



Fonte: O autor (2022)

a aplicação desse firmware, a etapa de inicialização servirá também para conectar o módulo a rede local de Wi-Fi e após isso, realizar a consulta da data e hora local através do protocolo NTP (*Network Time Protocol*). A Figura 22 representa a sequência de inicialização do firmware.

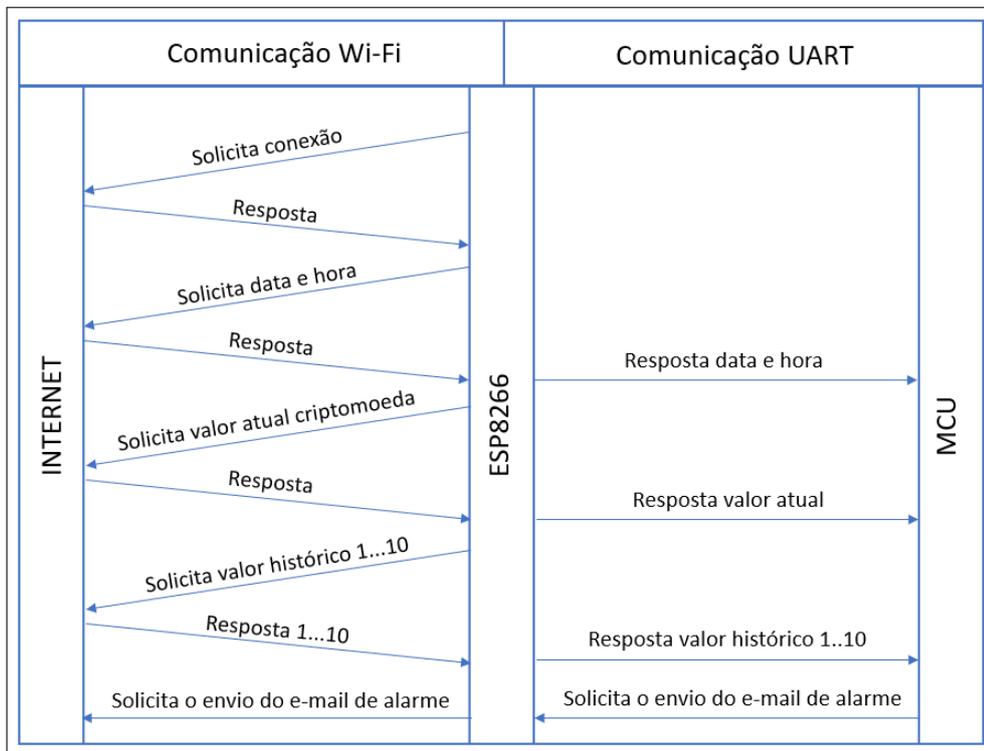
Figura 22 – Fluxograma Inicialização do firmware



Fonte: O autor (2022)

Nessa topologia de funcionamento, podemos considerar que temos 3 componentes participando da comunicação, sendo eles: a internet, o módulo Wi-Fi e o microcontrolador (MCU). Esses componentes se comunicarão em uma rede por meio de uma comunicação Wi-Fi, Internet e módulo Wi-Fi, e uma comunicação serial (UART), módulo Wi-Fi e microcontrolador. O diagrama apresentado na Figura 23, demonstra a sequência em que as mensagens são transmitidas e recebidas, pelos componentes participantes dessa rede de comunicação.

Figura 23 – Diagrama de mensagens transmitidas e recebidas



Fonte: O autor (2022)

Para conectar o módulo Wi-Fi à rede Wi-Fi local, foi utilizado a biblioteca `ESP8266WiFi.h` do Arduino, a qual realiza a conexão automaticamente sem a necessidade de comandos AT, neste caso somente será necessário informar o nome e a senha da rede a qual o módulo será conectado, e solicitar a conexão por meio do comando `WiFi.begin` dessa biblioteca.

Nessa etapa, será imprescindível que a conexão seja realizada adequadamente, caso contrário o firmware não seguirá adiante e ficará solicitando a conexão à rede local até que essa conexão seja estabelecida.

Após realizar a conexão, o módulo Wi-Fi realizará a requisição das informações de data e hora atualizadas no nosso fuso horário, via NTP (*Network Time Protocol*). Conforme descrito no portal NTP Brasil (BRASIL, 2022), o NTP ou Protocolo de Tempo para Redes, é o padrão utilizado para a sincronização dos relógios dos dispositivos de uma rede como servidores, estações de trabalho, roteadores e outros equipamentos a partir de referências de tempo confiáveis. Além do protocolo de comunicação em si, o NTP define uma série de algoritmos utilizados para consultar os servidores, calcular a diferença de tempo e estimar um erro, escolher as melhores referências e ajustar o relógio local.

Para realizar a coleta das informações de data e hora via NTP, foi utilizada a biblioteca `WiFiUdp.h`, a qual foi desenvolvida exatamente com esse propósito, buscar o horário via NTP e armazenar essas informações em registradores da biblioteca.

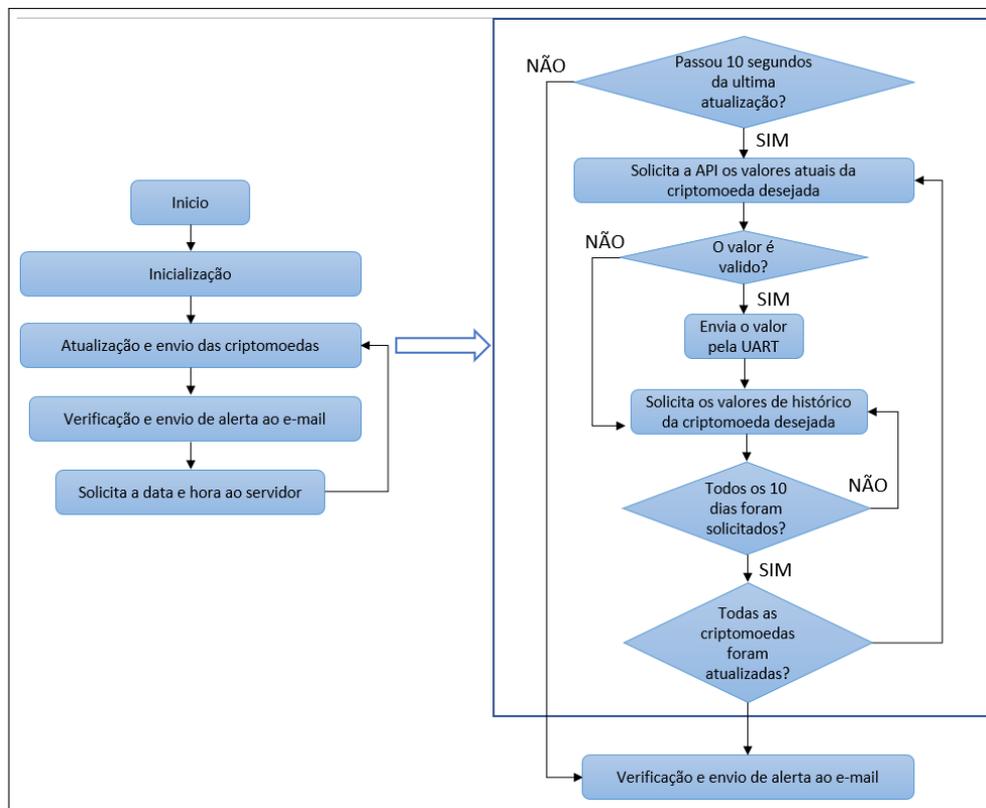
Após a inicialização concluída, seguirá o fluxo de operação e entrará no laço principal,

no qual o firmware basicamente realizará 3 processos, sendo eles:

- Processo 1: Atualização e envio das criptomoedas via UART, se o valor recebido for válido;
- Processo 2: Verificação e envio de alerta ao e-mail, caso seja necessário;
- Processo 3: Solicitar as informações atualizadas de data e hora ao servidor.

Na etapa de atualização e envio das criptomoedas, o firmware seguirá o fluxo de funcionamento conforme a Figura 24:

Figura 24 – Fluxograma atualização e envio das criptomoedas

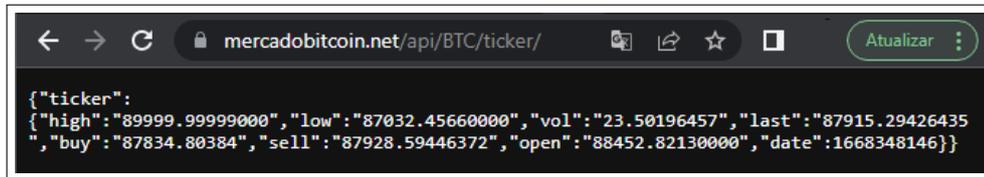


Fonte: O autor (2022)

Para controlar a primeira etapa desse processo, foi utilizado um laço de tempo utilizando a função `millis()` do Arduino. Essa função utiliza um temporizador interno, e é utilizada para atualizar variáveis a cada atualização desse temporizador. Na prática toda vez que esse temporizador é atualizado, as variáveis de controle incrementam 1 ao seu valor atual, assim é possível controlar as funções que devem ser acessadas periodicamente. No caso dessa aplicação, foi utilizado um temporizador para atualizar o valor das criptomoedas a cada 10 segundos, evitando assim que o algoritmo fique buscando a todo momento atualizar os parâmetros das criptomoedas, o que para essa aplicação atenderia perfeitamente.

Caso seja o momento de atualizar as criptomoedas, o módulo Wi-Fi solicitará a API o valor atual das criptomoedas, a Figura 25 representa a requisição dos parâmetros atuais da moeda Bitcoin (BTC).

Figura 25 – Requisição das informações atualizadas do Bitcoin, na API do portal Mercado Bitcoin



```
mercadobitcoin.net/api/BTC/ticker/
{"ticker":
{"high": "89999.99999000", "low": "87032.45660000", "vol": "23.50196457", "last": "87915.29426435", "buy": "87834.80384", "sell": "87928.59446372", "open": "88452.82130000", "date": "1668348146"}}
```

Fonte: O autor (2022)

Após requisitado o valor da cotação atual da criptomoedas, foi requisitado o histórico de preços dos últimos 10 dias, a Figura 26 representa a requisição dos valores do histórico da moeda Bitcoin (BTC).

Figura 26 – Requisição da cotação do dia 12/11/2022 do Bitcoin, na API do portal Mercado Bitcoin



```
mercadobitcoin.net/api/BTC/day-summary/2...
{"date": "2022-11-12", "opening": "89228.83238232", "closing": "88991.75485744", "lowest": "87628.15076", "highest": "89999.99999", "volume": "3055321.91697551", "quantity": "34.35709635", "amount": "2897", "avg_price": "88928.40902067"}
```

Fonte: O autor (2022)

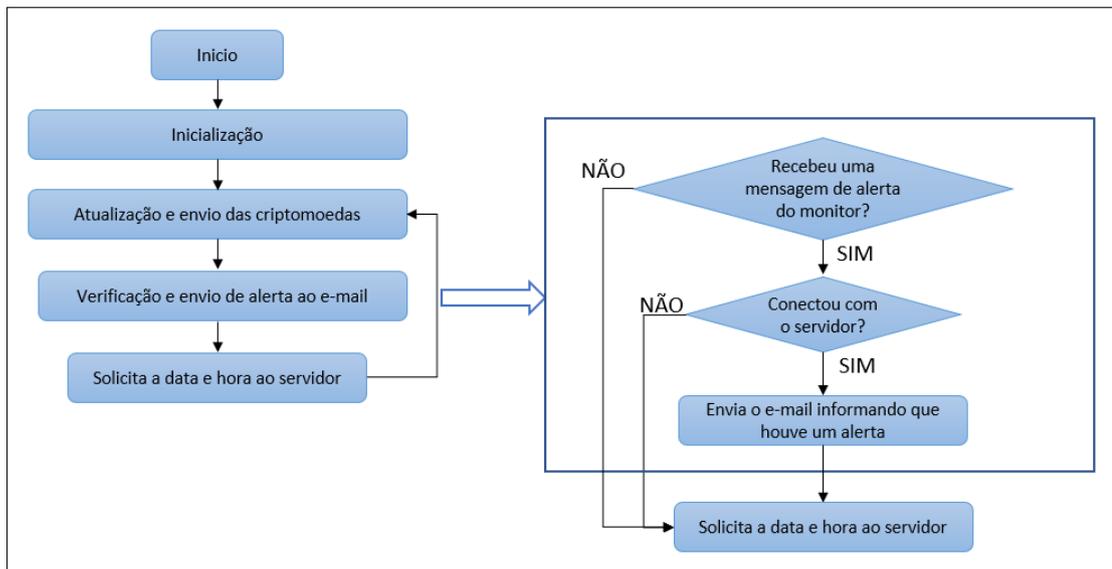
A requisição de acesso e a aquisição dos dados da API foram realizados pelo protocolo JSON (*JavaScript Object Notation*), segundo o portal (ORG, 2022), esse é um protocolo leve para intercâmbio de dados, sendo baseado em um subconjunto do *JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999*. O JSON é um formato de texto completamente independente de linguagem, mas similar as linguagens C, C++, C, Java, JavaScript, Perl e Python.

Em resumo, o JSON retorna uma coleção de pares nome/valor, ou seja, uma lista ordenada de valores como uma matriz, vetor, lista ou sequência. Assim como é possível observar no retorno da API, demonstrado previamente na Figura 25 e na Figura 26.

Na etapa de verificação e envio do alerta, o firmware seguirá o fluxo de funcionamento conforme a Figura 27:

Ao solicitar esses dados via JSON, todos esses dados são armazenados em uma variável temporária, filtrados e validados, após isto estas mensagens são realocadas nas variáveis que serão enviadas via UART, para o monitor de criptomoedas. As mensagens são enviadas seguindo o protocolo detalhado na Figura 28 e podem ser observados através do monitor serial representado pela Figura 29.

Figura 27 – Fluxograma verificação e envio do alerta



Fonte: O autor (2022)

Figura 28 – Protocolo de envio dos valores das criptomoedas

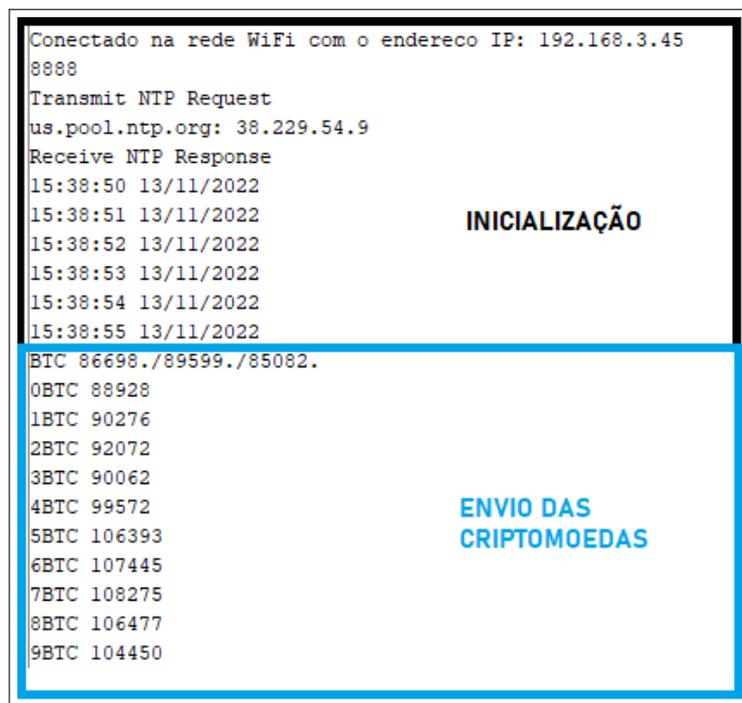
Valor corrente					Valor Histórico		
	SIGLA DA CRIPTOMOEDA	Valor Atual	Maior valor	Menor valor	ID	SIGLA DA CRIPTOMOEDA	Valor Atual
mensagem:	BTC	: 86681	/ 89599	/ 85082	mensagem: 0	BTC	88928
					mensagem: 1	BTC	90276
					mensagem: 2	BTC	92072
					mensagem: 3	BTC	90062
					mensagem: 4	BTC	99572
					mensagem: 5	BTC	106393
					mensagem: 6	BTC	107445
					mensagem: 7	BTC	108275
					mensagem: 8	BTC	106477
					mensagem: 9	BTC	104450

Fonte: O autor (2022)

A realização desta etapa do firmware está condicionada ao recebimento de uma mensagem de gatilho, a qual o monitor de criptomoedas enviará ao módulo Wi-Fi, essa mensagem informará ao sistema que uma das criptomoedas teve o seu limite atingido, sendo assim o usuário deverá receber um aviso, via e-mail. Caso esse gatilho não seja acionado o firmware seguirá o fluxo natural, ou seja, irá para a etapa de solicitar a data e hora atualizado ao servidor.

Para realizar o envio do e-mail ao usuário foi utilizado a interface de integração IFTTT (*If This Then That*), segundo consta no site oficial (IFTTT, 2020), essa plataforma web se auto descreve como a melhor maneira a integração de aplicativos, dispositivos e serviços. Essa plataforma funciona com a criação de um Applet, que é basicamente um pequeno software que executa uma atividade específica, dentro do contexto de outro programa maior. Essa Applet basicamente funciona com a seguinte estrutura: um evento de gatilho e uma ação, ou seja, caso o evento gatilho ocorra, a ação será executada. A plataforma IFTTT foi escolhida, pois pos-

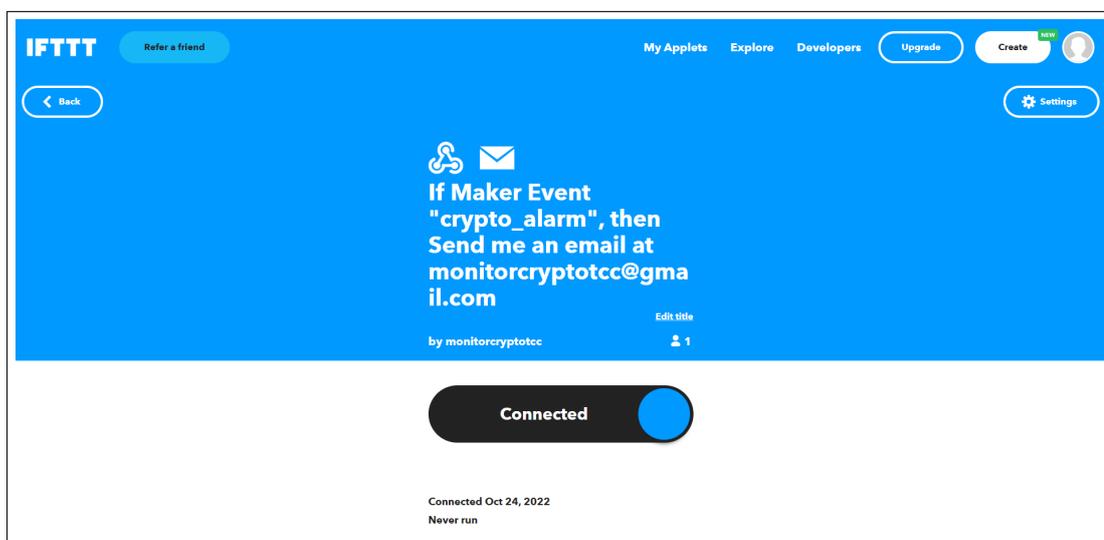
Figura 29 – Dados observados no monitor serial



Fonte: O autor (2022)

sui uma grande diversidade de gatilhos possíveis e ações que podem ser combinados. No caso dessa aplicação o gatilho escolhido foi o acesso a uma URL específica, ou seja, quando ocorrer o acesso a essa URL, um e-mail será enviado ao destinatário pré-definido. Na Figura 30 é possível observar o software de integração dedicado ao envio do e-mail, caso a criptomoeda ultrapole o limite.

Figura 30 – Applet para envio de e-mail

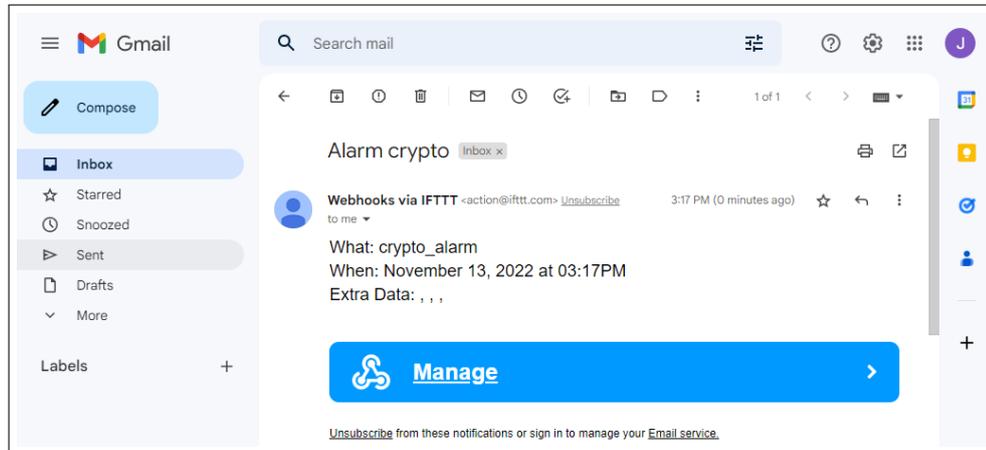


Fonte: O autor (2022)

Assim sendo, caso o módulo Wi-Fi receba uma mensagem de alerta, esse acessá a URL

pré-definida na Applet e a interface de integração IFTTT, enviará um e-mail ao endereço cadastrado. A Figura 31 abaixo demonstra como seria a mensagem de alarme recebida por e-mail.

Figura 31 – E-mail de alerta



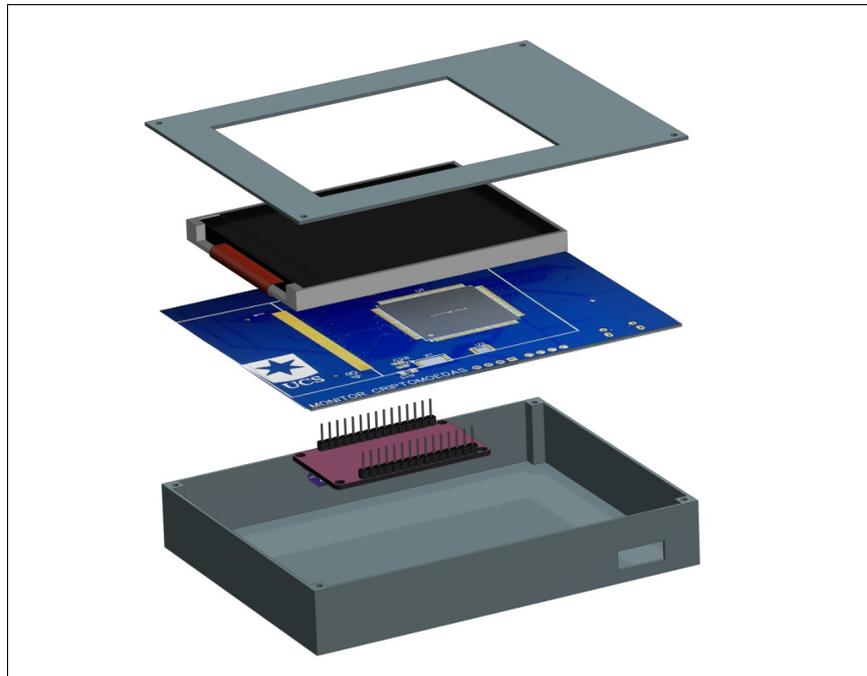
Fonte: O autor (2022)

A etapa de solicitação de data e hora atualizada é realizada da mesma forma que foi realizado na inicialização, é feita através do protocolo NTP e nesse contexto servirá para manter o dispositivo sempre atualizado, em situações em que esse dispositivo ficaria ligado por varios dias. Ao final desta etapa, o firmware retornará ao início do laço principal e reiniciará as etapas.

3.5 INVÓLUCRO

O procedimento para desenvolvimento do invólucro consiste em quatro etapas: elaborar a tampa do invólucro, elaborar o corpo do invólucro, modelar a placa e posteriormente realizar a montagem, com os componentes: tampa, monitor de criptomoedas, módulo Wi-Fi (modelo baixado no GrapCad) e o corpo do invólucro. Esse desenvolvimento foi realizado por meio da ferramenta NXCad, onde podemos observar a interface desse ambiente no Anexo I e a representação dos componentes na placa pela Figura 32 .

Figura 32 – Vista de todos os componentes



Fonte: O autor (2022)

4 CONSIDERAÇÕES FINAIS

Atualmente existem dispositivos similares, ao proposto nesse projeto já disponíveis no mercado, conforme o modelo representado no ANEXO J. Esses dispositivos possuem uma ideia de funcionamento diferente, pois as moedas são selecionadas através de um software de computador, o qual descarrega as informações no dispositivo via USB, diferentemente da ideia proposta, a qual possibilita selecionar a moeda digital através da própria tela, por meio do recurso de *touchscreen*.

A solução proposta foi desenvolvida para que seja possível, acompanhar o valor atual e o histórico de valores das principais criptomoedas do mercado, através de uma interface gráfica. A tomada de decisões baseadas nas oscilações do mercado, é uma necessidade para todos os investidores desse modelo de investimento. Nestas aplicações, a possibilidade de acompanhar os dados da moeda digital constantemente, estão diretamente vinculados ao resultado final das aplicações. Considerando essas condições, todas as formas de acompanhar e monitorar as oscilações do mercado em tempo real são benéficas, o que motivou o autor deste trabalho a desenvolver um dispositivo, dedicado exclusivamente a este propósito. Essa característica, permitirá que o usuário acompanhe a criptomoedas sem a necessidade de acessar um aplicativo, site ou software dedicado.

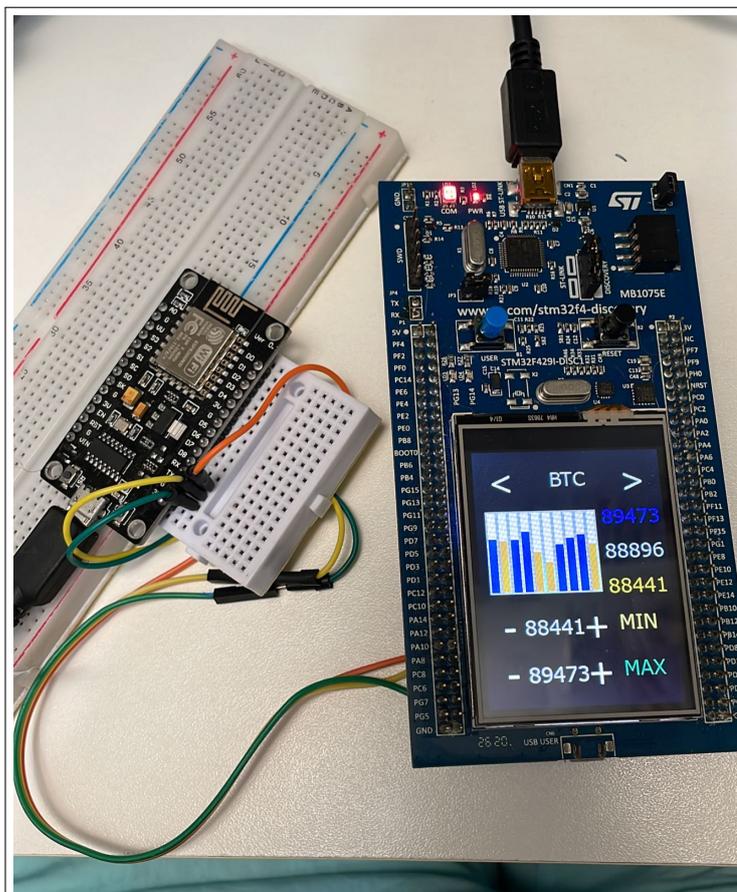
Para implementar uma solução funcional do dispositivo monitor de criptomoedas, foi necessário desenvolver o esquemático elétrico, a PCI (placa de circuitos impresso), o firmware do monitor de criptomoedas, o firmware do módulo Wi-Fi e o invólucro do produto. Como esse dispositivo é considerado um protótipo, a produção da PCI ficou condicionada a uma segunda etapa do desenvolvimento, esta podendo ser realizada após a aprovação do protótipo.

Devido a disponibilidade de recursos do autor, o projeto foi testado e validado em uma *demoboard* STM32F429 DISCOVERY e uma ESP8266, de propriedade do autor, assim se aproximando de todos os recursos necessários para atender a aplicação. Por meio destas plataformas de desenvolvimento, foi possível validar o funcionamento dos firmwares e o conceito de funcionamento geral do dispositivo, na Figura 33 é possível observar o ambiente em que a solução foi desenvolvida.

Durante o processo de validação do protótipo, os dados exibidos no monitor de criptomoedas foram certificados por cotações do portal Mercado Bitcoin (MERCADOBITCOIN, 2022), e o dispositivo desenvolvido apresentou resultados satisfatórios nos testes realizados. Basicamente os testes foram separados no monitoramento das criptomoedas e na verificação do disparo dos alarmes.

Para validar o monitoramento das criptomoedas foi registrado uma imagem do dispositivo e os dados confrontados com os valores disponíveis na internet. Na Figura 34 é possível

Figura 33 – Ambiente de desenvolvimento do monitor de criptomoedas



Fonte: O autor (2022)

observar que no momento do registro, a criptomoeda estava com o valor atual de 6669 reais, valor mínimo 6650 reais e valor máximo de 6809 reais, e os valores de alarme no momento do registro estavam em 6656 reais como mínimo e 6673 reais como máximo.

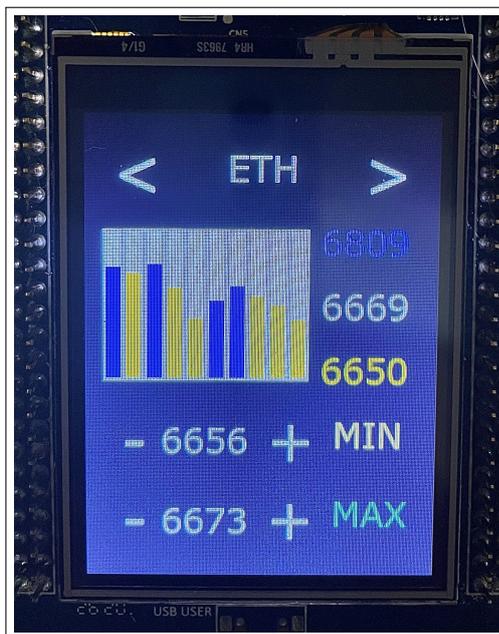
Após registrar o valor do dispositivo monitor de criptomoedas, foram coletadas as informações do mesmo período na *Exchange* Mercado Bitcoin (MERCADOBITCOIN, 2022), conforme representado na Figura 35 e na Figura 36.

O funcionamento do dispositivo também foi simulado nas situações de alarme, essa simulação foi feita alterando os valores limites, assim no momento em que o valor foi ultrapassado, o texto indicativo do valor limite teve a sua cor modificada, conforme representado na Figura 37.

Assim que o alarme foi disparado, um e-mail de alarme foi recebido na caixa de entrada, conforme representado e a Figura 38.

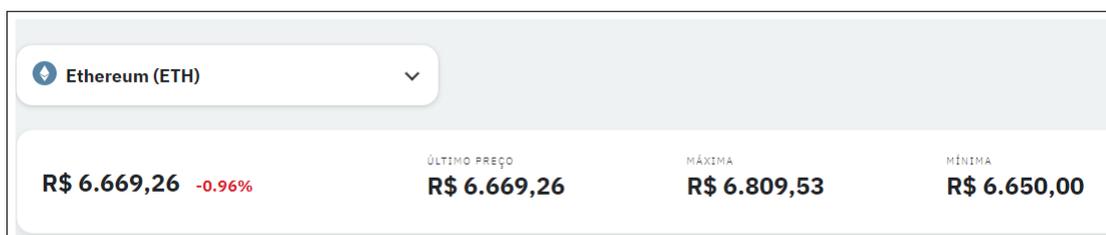
Através da análise dos resultados dos testes, é possível concluir que o dispositivo desenvolvido, atende aos requisitos propostos no início do trabalho. O projeto dos firmwares, está disponível na plataforma GitHub (BUCHEBUAM, 2022), possibilitando assim, que os testes sejam replicados e aferidos por terceiros.

Figura 34 – Teste do monitor de criptomoedas da moeda ETH, dia 03/12/2022 hora 14:43



Fonte: O autor (2022)

Figura 35 – Cotação ETH na Exchange Mercado Bitcoin, dia 03/12/2022 hora 14:42



Fonte: (MERCADOBITCOIN, 2022)

Figura 36 – Gráfico ETH do dia 30/11/2022 ao dia 03/12/2022



Fonte: (MERCADOBITCOIN, 2022)

A divisão do projeto em etapas proporcionou um excelente aprendizado de diferentes tecnologias, em três etapas desenvolvidas: hardware, firmware e invólucro. Assim pode-se considerar que o dispositivo aqui desenvolvido, além de agregar conhecimento ao autor nas etapas supracitadas, pode sim ser uma solução viável para o mercado, no que diz respeito a ideia de produto.

Figura 37 – Monitor de criptomoedas com alarme ativo, simulando dia 03/12/2022 às 14:39



Fonte: O autor (2022)

Figura 38 – E-mail recebido informando que ocorreu um alarme



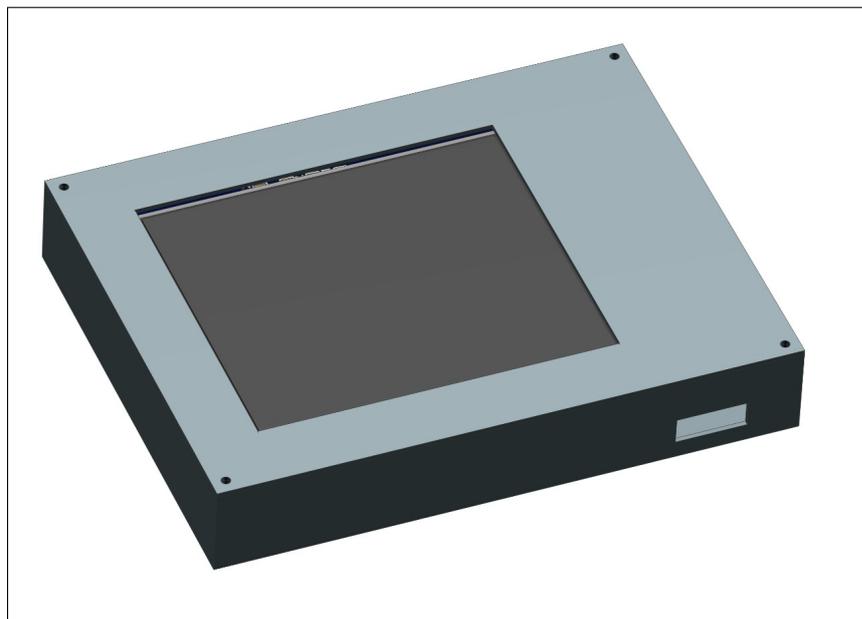
Fonte: O autor (2022)

4.1 TRABALHOS FUTUROS

Após a etapa de aprovação do protótipo, torna-se necessário a produção da PCI e montagem dos componentes eletrônicos na mesma e a produção do invólucro. Assim todos os componentes do produto estarão disponíveis e dispositivo poderá ser montado, consolidando a estrutura física do dispositivo. A Figura 39 representa como o dispositivo ficaria após o fechamento do invólucro:

A segunda etapa do desenvolvimento futuro será o gerenciamento de um banco de dados específico para essa aplicação. O objetivo dessa implementação será a validação e redundância

Figura 39 – Produto no involucro fechado



Fonte: O autor (2022)

dos dados, sendo assim possível a partir desse momento dobrar a confiabilidade dos gráficos. Essa funcionalidade permitirá criar variações nos gráficos apresentados, como por exemplo o histórico das últimas 10 variações nas últimas 1h, 4h ou 24h. A redundância também será implementada para a aquisição dos dados da API, sendo assim, o sistema somente salvaria os dados no banco de dados se confirmado a veracidade dos mesmos.

Após o estudo das APIs, foi avaliada a possibilidade de executar compra e venda de criptoativos por meio de comandos enviados a esta API. A implementação desta funcionalidade representará um substancial avanço no conceito do produto, pois ao invés de somente alertar ao usuário, quando uma moeda digital ultrapassa um limite, seria possível executar ordens de compra e venda quando esses limites forem ultrapassados. Esta alteração, seria uma terceira etapa do desenvolvimento futuro, a qual não foi desenvolvida nesta fase do protótipo, pois ao movimentar valores dentro de uma *Exchange*, uma série de fatores de segurança devem ser atendidos, e para atender a estas condições, deverá ser realizado um profundo embasamento teórico, o que neste momento inviabilizaria atender aos objetivos desse projeto.

A quarta etapa do desenvolvimento futuro, será o desenvolvimento de um teclado alpha numérico, essa ferramenta será utilizada para adicionar novas criptomoedas que possam surgir futuramente, assim o dispositivo poderia ampliar o seu portfólio de criptomoedas disponíveis para o acompanhamento.

REFERÊNCIAS

- ACCARDI, A.; DODONOV, E. Automação residencial: Elementos básicos, arquiteturas, setores, aplicações e protocolos. **Tecnologias, Infraestrutura e Software**, p. 156, nov. 2012.
- ALLDATASHEET.COM. **STM32F429ZI Datasheet (PDF) - STMICROELECTRONICS**, month = jun, year = 2021, note = Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/882039/STMICROELECTRONICS/STM32F429ZI.html>> Acesso em: 22 de jun. de 2021.
- AMS. **AMS1117-3.3 Datasheet**. 2022. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/205691/ADMOS/AMS1117-3.3.html>> Acesso em: 14 de nov. de 2022.
- BIEGELMEYER Anderson. Internet Of Things – Desenvolvimento e Aplicação de uma Casa Inteligente. 2015.
- BRANDÃO, P. **Criptomoeda: o Bitcoin**. 2020. Disponível em: <<http://hdl.handle.net/10400.2/10377>> Acesso em: 01 de jul. de 2022.
- BRASIL, N. **Network Time Protocol**. 2022. Disponível em: <<https://ntp.br/conteudo/ntp/>> Acesso em: 13 de nov. de 2022.
- BUCHEBUAM, J. **Repositório Firmware Monitor Criptomoeda**. 2022. <https://github.com/jabuchebuam/CryptoMonitor.git,2022> Acesso em: 04 de dez. de 2022.
- CANZIAN, E. **MINICURSO Comunicação Serial – RS232**. Cotia, São Paulo, Brasil: [s.n.], 2009.
- CASTAÑEDA, R. G. Implementación y ejecución de un protocolo de transferencia de archivos (FTP). 2012.
- CHAVES, L. **BLOCKCHAIN e CRIPTOMOEDAS**. 1. ed. Curitiba, Paraná, Brasil: InterSaberes, 2021. 146 p.
- EASYEDA. **Plataforme de desenvolvimento de Esquemático e PCB**. 2022. <https://easyeda.com/,2022> Acesso em: 14 de nov. de 2022.
- EDUCATION, I. C. **Interface de programação de aplicativos (API)**. 2022. Disponível em: <<https://www.ibm.com/cloud/learn/api>> Acesso em: 11 de nov. de 2022.
- EXCHANGE. **Exchange de criptomoedas: o que são e como escolher uma?** 2022. Disponível em: <<https://www.infomoney.com.br/guias/exchange-de-criptomoedas>> Acesso em: 01 de jul. de 2022.
- [HTTPS://SUPPORT.TOUCHGFX.COM](https://support.touchgfx.com). **MVP**. 2022. <https://support.touchgfx.com/4.14/docs/development/ui-development/software-architecture/model-view-presenter-design-pattern/,2022> Acesso em: 12 de nov. de 2022.

HTTPS://THECUSTOMIZEWINDOWS.COM. **RTOS**. 2022.

<https://thecustomizewindows.com/2017/01/difference-real-time-os-rtos-embedded-linux/>, 2022 Acesso em: 12 de nov. de 2022.

HTTPS://UPLOAD.WIKIMEDIA.ORG. **Barramento I2C**. 2022.

<https://upload.wikimedia.org/wikipedia/commons/thumb/3/3e/I2C.svg/220px-I2C.svg.png>, 20221 Acesso em: 01 de jul. de 2022.

IFTTT. **ifttt**. 2020. Disponível em: <https://ifttt.com/explore/new_to_ifttt > Acesso em : 13 de nov. de 2022.

MARCIANO, L. C. **Arquitetura De Telas – Com Microcontroladores E Interfaces Graficas Em Sistemas Embarcados**. 2019.

MERCADOBITCOIN. **Cotação EHT Mercado Bitcoin**. 2022.

<https://www.mercadobitcoin.com.br/negociacoes/eth>, 2022 Acesso em: 03 de dez. de 2022.

NAKAMOTO, S. **Bitcoin: A Peer-to-Peer Electronic Cash System**. 2008. Disponível em: <<https://bitcoin.org/bitcoin.pdf>> Acesso em: 20 de jul. de 2022.

OLIVEIRA, I. **Fundos cripto: número de investidores cresce 1.266 por cento no Brasil em 2021**. 2022. Disponível em: <<https://investidor.estadao.com.br/criptomoedas/investidores-fundo-criptomoedas-crescem-2021/>> Acesso em: 01 de jul. de 2022.

ORG, J. **JSON**. 2022. Disponível em: <<https://www.json.org/json-en.html>> Acesso em: 13 de nov. de 2022.

STMICROELECTRONICS. **Bluetooth Low Energy**. 2021. Disponível em: <<https://www.st.com/en/applications/connectivity/bluetooth-low-energy.html>> Acesso em: 21 de jun. de 2021.

STMICROELECTRONICS. **Datab TouchGFX Designer**. 2022.

https://www.st.com/resource/en/data_brief/touchgfxdesigner.pdf, 2022 Acesso em : 13 de nov. de 2022.

_____. **Demoboard**. 2022. <https://www.st.com/en/evaluation-tools/32f429idiscovery.html>, 2022 Acesso em: 01 de jul. de 2022.

WILLIG, A. **An architecture for wireless extension of Profibus**. Roanoke, : IEEE Int. Conf. Ind. Electron., 2003.

WWW.ETSY.COM. **gadget**. 2022. https://www.etsy.com/listing/979870136/ticrthing-bitcoin-price-display-crypto?ga_order=most_relevant&ga_search_type=all&ga_view_type=gallery&ga_search_query=gadget+bitcoin&ref=sr_gallery-1-4organic_search_click = 1 Acesso em : 01 de jul. de 2022.

YOUNG, P. H. **Técnicas de Comunicação Eletrônica**. 5. ed. [S.l.]: Pearson Universidades, 2005. 590 p.

ANEXO A - DATASHEET DO MICROCONTROLADOR STM32F429XX



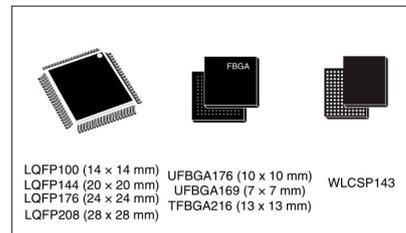
STM32F427xx STM32F429xx

ARM Cortex-M4 32b MCU+FPU, 225DMIPS, up to 2MB Flash/256+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 comm. interfaces, camera & LCD-TFT

Datasheet - production data

Features

- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 180 MHz, MPU, 225 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Memories
 - Up to 2 MB of Flash memory organized into two banks allowing read-while-write
 - Up to 256+4 KB of SRAM including 64-KB of CCM (core coupled memory) data RAM
 - Flexible external memory controller with up to 32-bit data bus: SRAM, PSRAM, SDRAM/LPDDR SDRAM, Compact Flash/NOR/NAND memories
- LCD parallel interface, 8080/6800 modes
- LCD-TFT controller up to XGA resolution with dedicated Chrom-ART Accelerator™ for enhanced graphic content creation (DMA2D)
- Clock, reset and supply management
 - 1.7 V to 3.6 V application supply and I/Os
 - POR, PDR, PVD and BOR
 - 4-to-26 MHz crystal oscillator
 - Internal 16 MHz factory-trimmed RC (1% accuracy)
 - 32 kHz oscillator for RTC with calibration
 - Internal 32 kHz RC with calibration
- Low power
 - Sleep, Stop and Standby modes
 - V_{BAT} supply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
- 3×12-bit, 2.4 MSPS ADC: up to 24 channels and 7.2 MSPS in triple interleaved mode
- 2×12-bit D/A converters
- General-purpose DMA: 16-stream DMA controller with FIFOs and burst support
- Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 180 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
- Debug mode
 - SWD & JTAG interfaces
 - Cortex-M4 Trace Macrocell™



- Up to 168 I/O ports with interrupt capability
 - Up to 164 fast I/Os up to 90 MHz
 - Up to 166 5 V-tolerant I/Os
- Up to 21 communication interfaces
 - Up to 3 × I²C interfaces (SMBus/PMBus)
 - Up to 4 USARTs/4 UARTs (11.25 Mbit/s, ISO7816 interface, LIN, IrDA, modem control)
 - Up to 6 SPIs (45 Mbits/s), 2 with muxed full-duplex I²S for audio class accuracy via internal audio PLL or external clock
 - 1 × SAI (serial audio interface)
 - 2 × CAN (2.0B Active) and SDIO interface
- Advanced connectivity
 - USB 2.0 full-speed device/host/OTG controller with on-chip PHY
 - USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
 - 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII
- 8- to 14-bit parallel camera interface up to 54 Mbytes/s
- True random number generator
- CRC calculation unit
- RTC: subsecond accuracy, hardware calendar
- 96-bit unique ID

July 2016

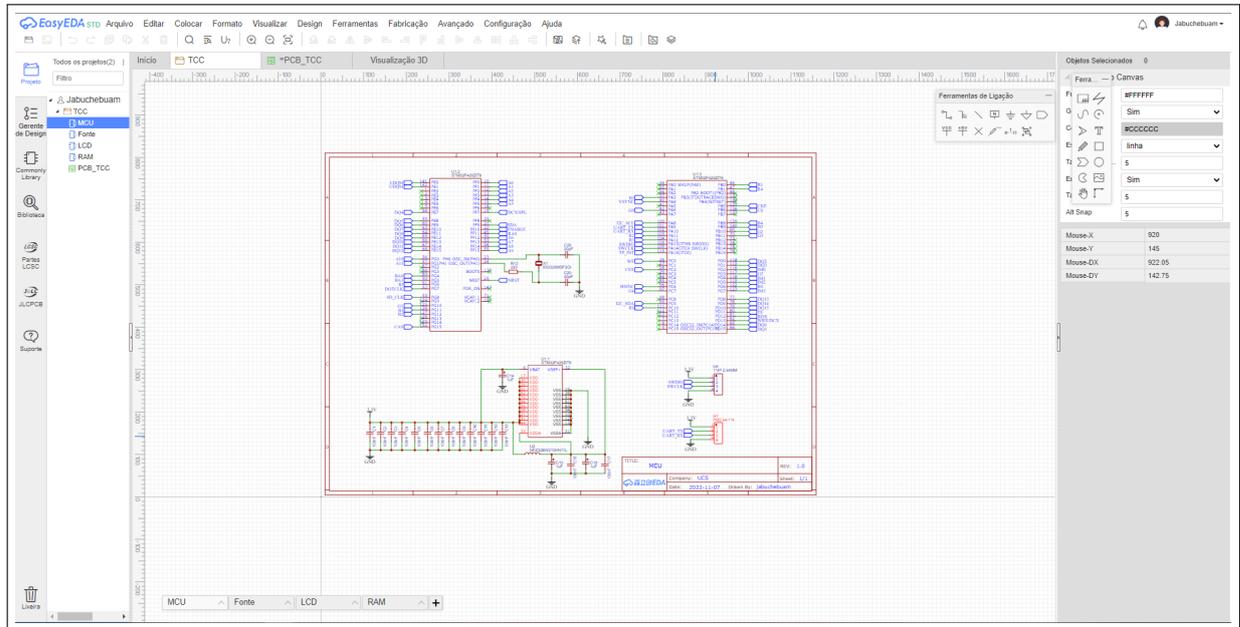
DocID024030 Rev 9

1/238

This is information on a product in full production.

www.st.com

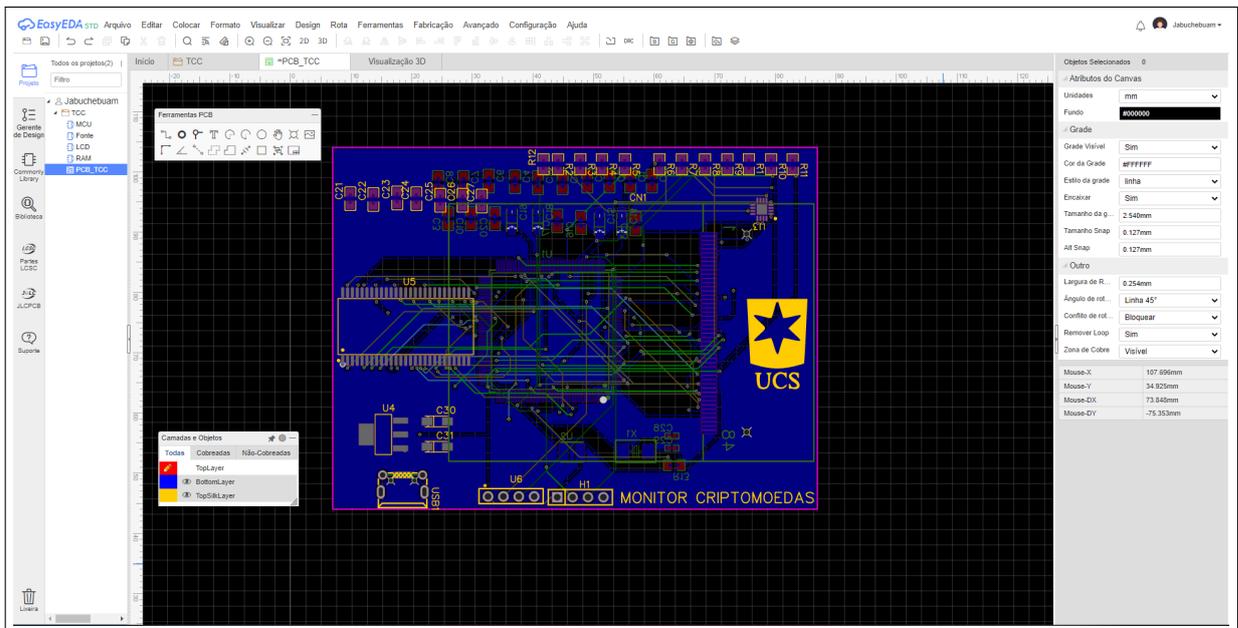
ANEXO B - *WORKSPACE* DE DESENVOLVIMENTO DE ESQUEMÁTICO DA PLATAFORMA EASYEDA



Fonte: (EASYEDA, 2022)

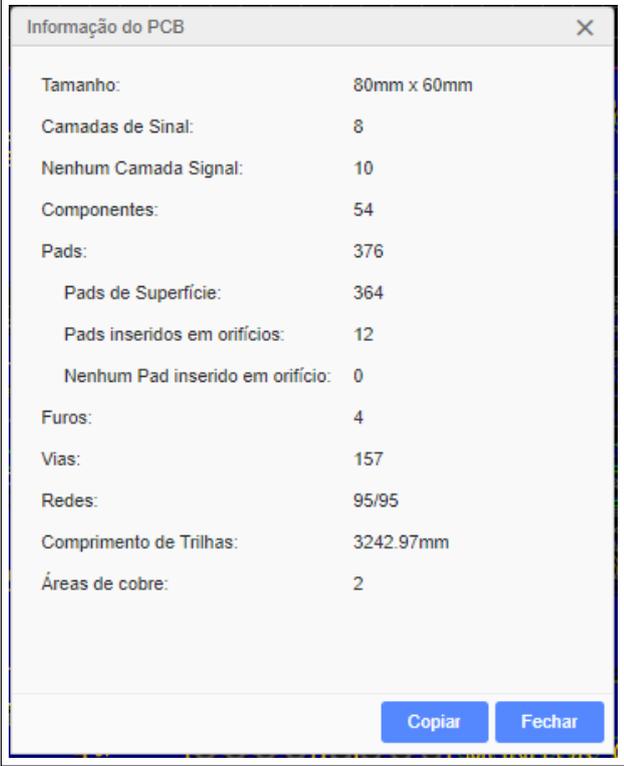
ANEXO C - WORKSPACE DE DESENVOLVIMENTO DE PCB DA PLATAFORMA EASYEDA

Figura 40 – *Workspace* de desenvolvimento de PCB da plataforma EasyEDA



Fonte: (EASYEDA, 2022)

ANEXO D - INFORMAÇÕES DA PCB

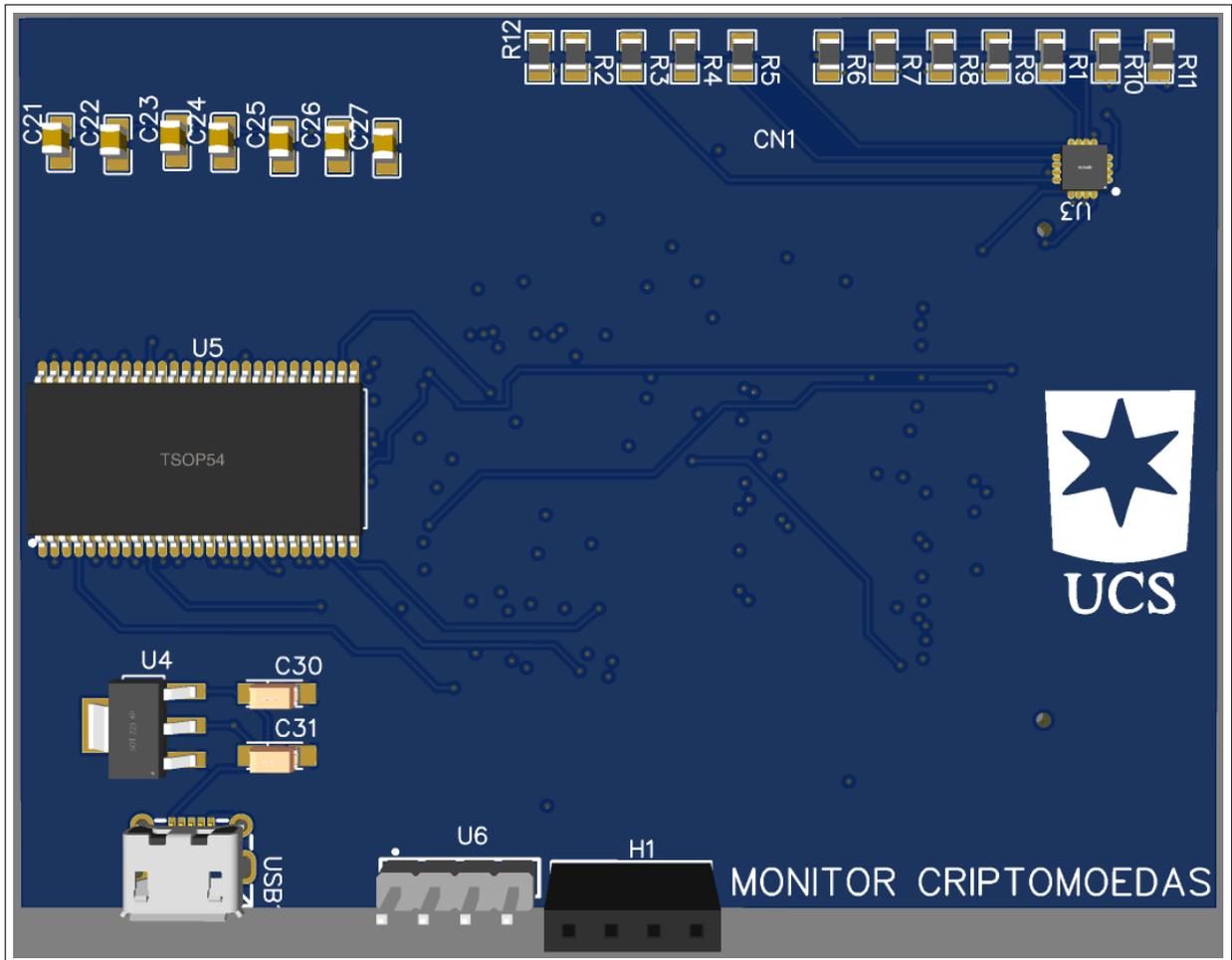


The image shows a screenshot of a software window titled "Informação do PCB" (PCB Information). The window contains a list of PCB statistics and their corresponding values. At the bottom right of the window, there are two buttons: "Copiar" (Copy) and "Fechar" (Close).

Propriedade	Valor
Tamanho:	80mm x 60mm
Camadas de Sinal:	8
Nenhum Camada Signal:	10
Componentes:	54
Pads:	376
Pads de Superfície:	364
Pads inseridos em orifícios:	12
Nenhum Pad inserido em orifício:	0
Furos:	4
Vias:	157
Redes:	95/95
Comprimento de Trilhas:	3242.97mm
Áreas de cobre:	2

Fonte: O autor (2022)

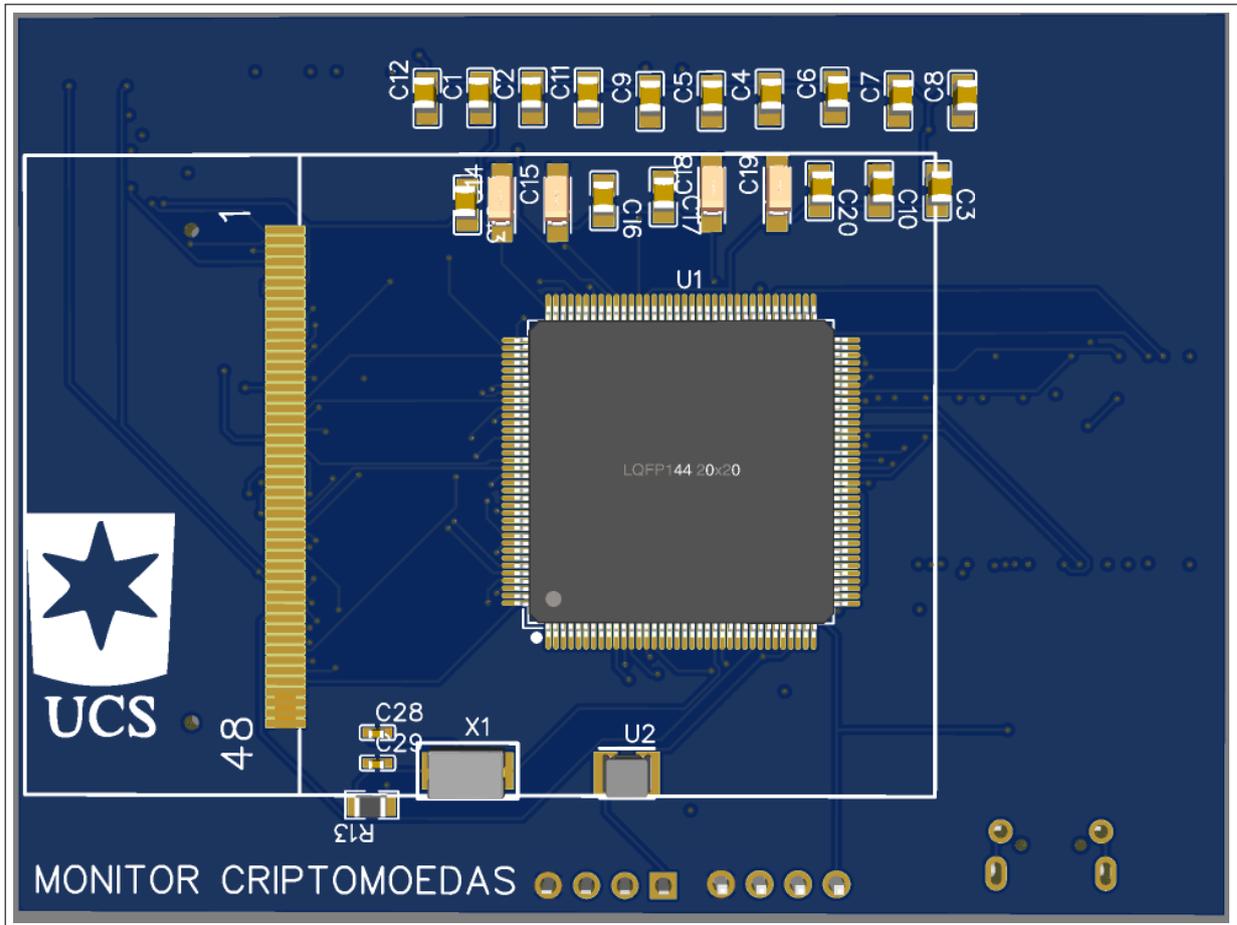
ANEXO E - REPRESENTAÇÃO DO MODELO 3D DA VISTA SUPERIOR DA PCB



Fonte: O autor (2022)

ANEXO F - REPRESENTAÇÃO DO MODELO 3D DA VISTA INFERIOR DA PCB

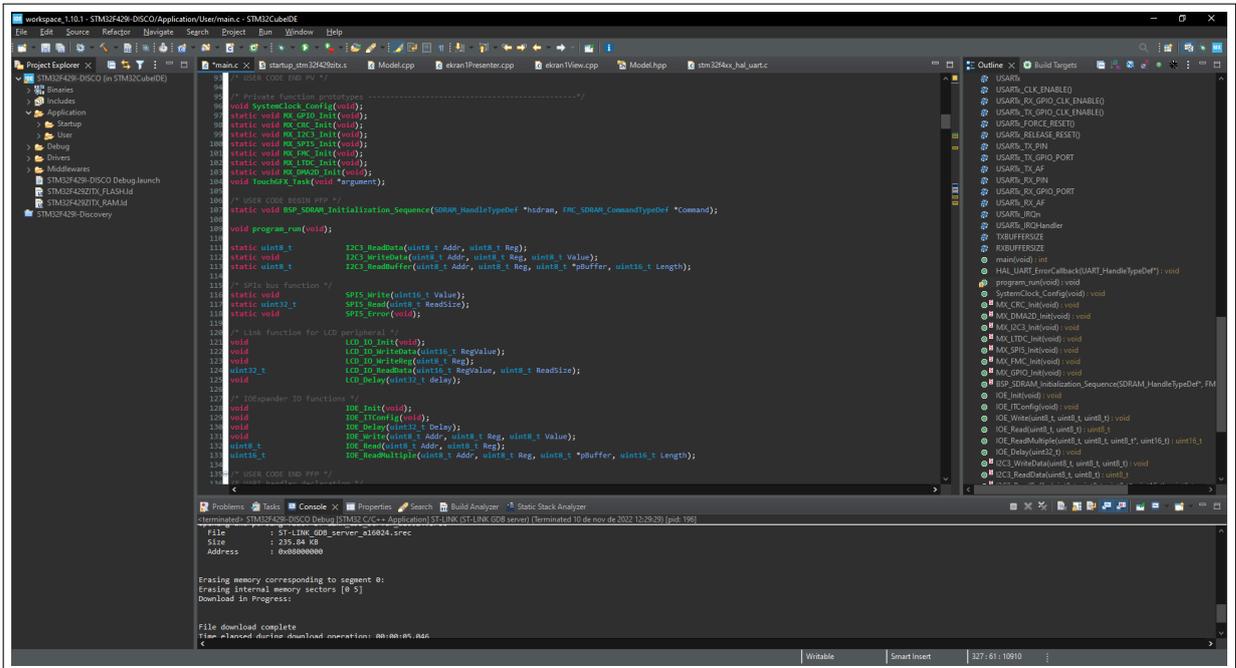
Figura 41 – Representação do modelo 3D da vista inferior da PCB



Fonte: O autor (2022)

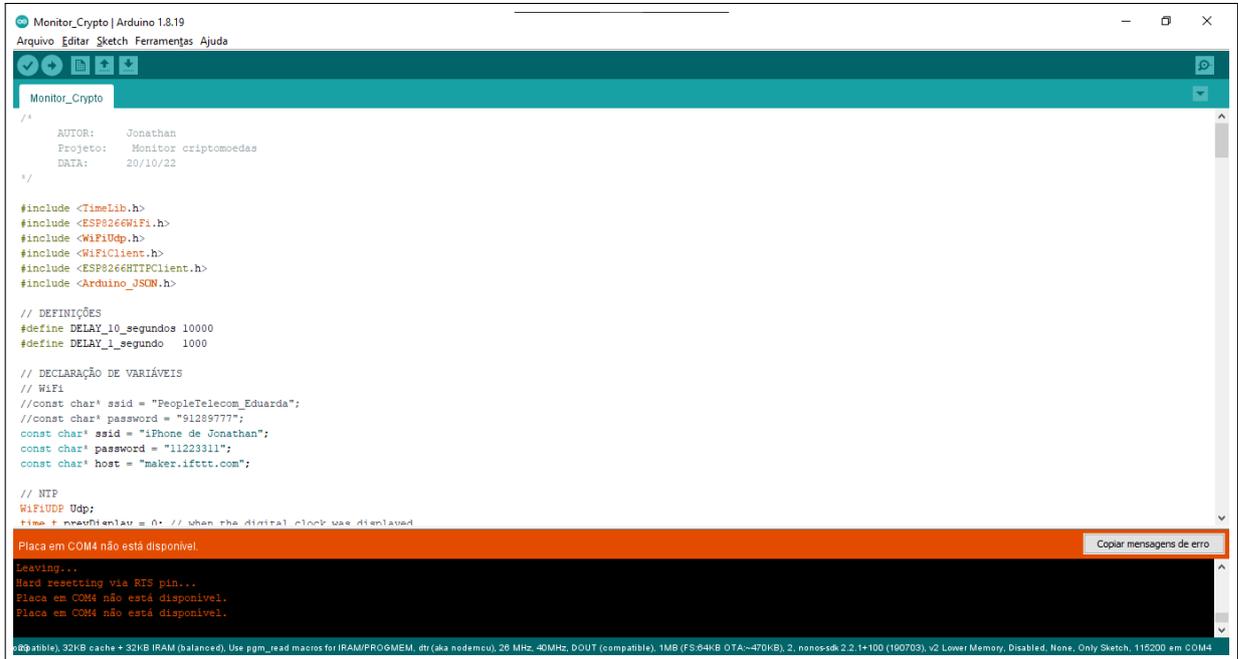
ANEXO G - WORKSPACE DA STM32CUBEIDE

Figura 42 – *Workspace* da STM32CubeIDE



Fonte: (STMICROELECTRONICS, 2021)

ANEXO H - INTERFACE DE PROGRAMAÇÃO DO ARDUINO



```
Monitor_Crypto | Arduino 1.8.19
Arquivo  Editar  Sketch  Ferramentas  Ajuda

Monitor_Crypto

/*
  AUTOR:    Jonathan
  Projeto:  Monitor criptoeddas
  DATA:    20/10/22
*/

#include <TimeLib.h>
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <Arduino_JSON.h>

// DEFINIÇÕES
#define DELAY_10_segundos 10000
#define DELAY_1_segundo  1000

// DECLARAÇÃO DE VARIÁVEIS
// WiFi
//const char* ssid = "PeopleTelecom_Eduarda";
//const char* password = "91289777";
const char* ssid = "iPhone de Jonathan";
const char* password = "11223311";
const char* host = "maker.ifttt.com";

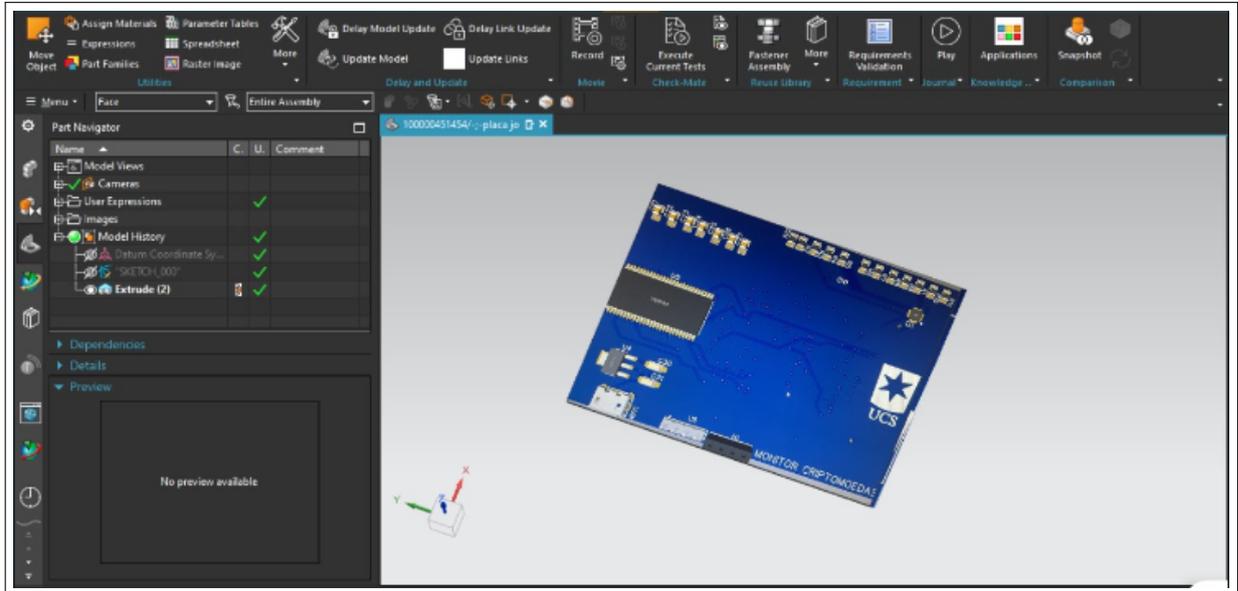
// NTP
WiFiUDP Udp;
time_t nowOffset = 0; // when the digital clock was displayed

Placa em COM4 não está disponível. Copiar mensagens de erro
Escaving...
Hard resetting via RTS pin...
Placa em COM4 não está disponível.
Placa em COM4 não está disponível.

@920kb) 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM. dtu (aka nodemcu), 26 MHz, 40MHz, DOUT (compatible), 1MB (FS:64KB OTA~470KB), 2, nonosdk 2.2.14-100 (160703), v2 Lower Memory, Disabled, None, Only Sketch, 115200 em COM4
```

Fonte: O autor (2022)

ANEXO I - AMBIENTE DO NXCAD USADO NO DESENVOLVIMENTO DO INVÓLUCRO



Fonte: O autor (2022)

ANEXO J - MONITOR DE CRIPTOMOEDA DO MERCADO



Fonte: (WWW.ETSY.COM, 2022)