

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ

EMANUEL COUTO WEISS
RENATO XAVIER MACEDO

**CRONÔMETRO SINCRONIZADO WIRELESS PARA PISTA
DE ATLETISMO**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2018

EMANUEL COUTO WEISS
RENATO XAVIER MACEDO

**CRONÔMETRO SINCRONIZADO WIRELESS PARA PISTA
DE ATLETISMO**

TRABALHO DE CONCLUSÃO DE CURSO

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Marcelo Eduardo Pelenz

CURITIBA

2018

TERMO DE APROVAÇÃO

EMANUEL COUTO WEISS
RENATO XAVIER MACEDO

CRONÔMETRO SINCRONIZADO WIRELESS PARA PISTA DE ATLETISMO

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da Universidade Federal do Paraná como requisito à obtenção do grau de Engenheiro Eletricista, pela seguinte banca avaliadora:

Orientador: Prof. Dr. Marcelo Eduardo Pellenz
Departamento de Engenharia Elétrica, UFPR

Prof. Dr. Marcos Vinicio Haas Rambo
Departamento de Engenharia Elétrica, UFPR

Prof. M.Sc. Rafael da Silva Ferraz
Departamento de Engenharia Elétrica, UFPR

Curitiba, 04 de dezembro de 2018

AGRADECIMENTOS

Em primeiro lugar a Deus que nos deu forças e nos iluminou até chegarmos aqui.

Aos nossos familiares que sempre nos apoiaram e incentivaram ao longo dessa jornada.

Aos amigos que estiveram do nosso lado, compartilhando conhecimentos, sofrendo e se divertindo juntos.

Por fim aos nossos professores que nos passaram seus conhecimentos e experiências essenciais para a nossa formação, além de nos fazerem superar desafios e crescer pessoalmente e profissionalmente.

"O que fazemos em vida, ecoa na eternidade."

(Scott, Ridley;Gladiador;2000)

RESUMO

Este trabalho tem como objetivo o desenvolvimento de um cronômetro virtual a fim de automatizar a marcação dos tempos dos atletas em provas de velocidade do atletismo. Neste tipo de modalidade a concentração do atleta é de fundamental importância, visto que nessas competições a vitória é decidida em questão de centésimos. Através da automatização das marcações de tempo, o atleta poderá ter um foco maior na sua corrida, não se preocupando em olhar o relógio ao longo do percurso ou parar a contagem manualmente ao final da corrida, o que resulta em uma marcação imprecisa e pouco confiável. Outra vantagem é a possibilidade de o atleta treinar sozinho, sem a necessidade de outra pessoa para cronometrar a sua corrida, tornando os seus treinos mais flexíveis. O sistema proposto contará com dois módulos de comunicação controlados por microcontroladores ESP32 utilizando o protocolo LoRa® para a comunicação *wireless* entre eles. Um dos módulos será posicionado junto à linha de largada da pista e o outro junto à linha de chegada. Assim que for dada a largada um sinal sonoro será emitido por um *buzzer* posicionado no módulo de largada e um sinal de rádio é enviado ao módulo da chegada para dar início a contagem do tempo. Assim que o atleta cruzar a linha de chegada um sensor ultrassônico detectará a sua passagem, fazendo com que o módulo ESP da saída trave a contagem de tempo, o qual poderá ser visto em um *display* de 7 segmentos. O projeto foi desenvolvido visando a praticidade e facilidade de uso por uma pessoa leiga em eletrônica, a qual terá como única tarefa pressionar um botão para dar início à operação do sistema, que ocorrerá de forma automática com as etapas de sinalização sonora, sincronização entre os módulos e marcação do tempo.

Palavras-chave: Cronômetro, automatização, módulos de comunicação, ESP32, protocolo LoRa, comunicação *wireless*.

ABSTRACT

This work aims the development of a virtual chronometer in order to automate the time marking of athletes in athletics speed events. In this type of modality the concentration of the athlete is of fundamental importance, since in these competitions the victory is decided in a matter of hundredths. Through the automation of time stamps the athlete may have a greater focus on their running, not bothering to look at the clock along the course or stop counting manually at the end of the race, which results in inaccurate and unreliable marking. Another advantage is the possibility of the athlete training alone, without the need of another person to time your sprint, making their trainings more flexible. The proposed system will have two communication modules controlled by ESP32 microcontrollers using the LoRa® protocol for wireless communication between them. One of the modules will be positioned next to the start line of the track and the other next to the finishing line. As soon as the start is given, a beep will be emitted by a buzzer positioned on the start module and a radio signal will be sent to the arrival module to start the time counting. As soon as the athlete crosses the finish line an ultrasonic sensor will detect its passage, causing the ESP module of the output to stop the time counting, which can be seen in a 7 segment display. The project was developed aiming the practicality and ease of use by a lay person in electronics, whose only task is to press a button to start the system operation, which will occur automatically with the steps of sound signaling, synchronization between the modules and time marking.

Key-words: Chronometer, automation, communication modules, ESP32, LoRa protocol, wireless communication.

LISTA DE FIGURAS

2.1	Fases da corrida	10
2.2	Dimensões da pista de atletismo	11
2.3	Diagrama de uma rede LoRa	16
2.4	ESP32 integrado com rede LoRa	17
2.5	Pinagem do ESP32 com rede LoRa	18
2.6	FSPL em função da distância para as frequências de 2,4 GHz, 5,1 GHz e 5,7 GHz.	21
3.1	Diagrama de atividades	22
3.2	Fluxograma para a programação do módulo ESP da largada	25
3.3	Fluxograma para a programação do módulo ESP da chegada	26
3.4	Posicionamento do módulo de largada	27
3.5	Ilustração do módulo de largada	28
3.6	Posicionamento do módulo de chegada	29
3.7	Ilustração do módulo de chegada	29
3.8	Sensor ultrassônico HC-SR04	30
3.9	Display TM1637	30
3.10	Circuito esquemático do multiplexador TM1637	31
3.11	Pista de atletismo simplificada	32
4.1	Circuito esquemático para a PCB da largada	34
4.2	Layout para a PCB da largada	34
4.3	Circuito esquemático para a PCB da chegada	35
4.4	Layout para a PCB da chegada	35
4.5	Captura da porta serial da chegada	36
4.6	Captura da porta serial da largada	36
4.7	Foto do módulo de largada	37
4.8	Foto do módulo de chegada no modo <i>stand by</i>	38
4.9	Foto do módulo de chegada após a marcação do tempo	38

LISTA DE SIGLAS

CI	Circuito Integrado
DCV	Doença Cardiovascular
ESP	Microcontrolador da empresa Espressif Systems
FSPL	Perda de Percurso do Espaço Livre
<i>GPS</i>	<i>Global Position System</i>
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
<i>IP</i>	<i>Internet Protocol</i>
<i>IoT</i>	<i>Internet of things</i>
<i>LED</i>	<i>Light Emmiting Diode</i>
<i>LoRa</i>	<i>Long Range</i>
<i>PCB</i>	<i>Printed Circuit Board</i>
<i>RF</i>	<i>Rádio Frequência</i>
<i>RFID</i>	<i>Radiofrequency Identification</i>
<i>SoC</i>	<i>System on a Chip</i>
<i>TCP</i>	<i>Transmission Control Protocol</i>

SUMÁRIO

RESUMO	6
ABSTRACT	7
LISTA DE FIGURAS	8
1 INTRODUÇÃO	7
1.1 OBJETIVOS	8
1.2 Objetivo Geral	8
1.3 Objetivos Específicos	8
2 FUNDAMENTAÇÃO TEÓRICA	9
2.1 AS PROVAS DE VELOCIDADE NO ATLETISMO	9
2.1.1 As etapas da corrida de velocidade	9
2.1.2 A pista e os fatores determinantes	10
2.2 INTERNET OF THINGS (IoT)	11
2.2.1 Conceitos de IoT	12
2.2.2 Principais Tecnologias	13
2.2.2.1 Radiofrequência	14
2.3 MÓDULO DE COMUNICAÇÃO - ESP32	15
2.3.1 Tecnologia LoRa e o protocolo LoRaWAN	15
2.3.2 ESP32 com rede LoRa	16
2.4 PROPAGAÇÃO NO ESPAÇO LIVRE	18
2.4.1 O modelo do espaço livre	18
3 DESENVOLVIMENTO	22
3.1 METODOLOGIA	23
3.2 Circuitos desenvolvidos	27
3.2.1 Módulo de largada	27
3.2.2 Módulo de chegada	28

3.3	CÁLCULOS ENVOLVENDO RF	31
4	RESULTADOS E DISCUSSÃO	33
5	CONCLUSÃO	39
5.1	TRABALHOS FUTUROS	40
A	CÓDIGOS DE PROGRAMAÇÃO	41
A.1	Cronometro Setup Principal	41
A.2	Chegada	47
A.3	Largada	52

CAPÍTULO 1

INTRODUÇÃO

A prática de esportes no ambiente acadêmico aponta para uma diminuição do risco de desenvolver doença cardiovascular (DCV) pelo menos até os 50 anos de idade. Pesquisas mostram que o exercício físico (esportes, caminhadas, escaladas, etc.) na meia idade está inversamente relacionado ao desenvolvimento tardio de DCV e morte precoce. Além disso as taxas de mortalidade são menores para homens que praticam esportes moderadamente intensos do que para homens menos ativos (JUNIOR, 1984).

Partindo da premissa de que um atleta necessita de concentração durante os treinos para obter um melhor rendimento, a automação de funções auxiliares durante estas atividades pode trazer vários benefícios. Na prática do atletismo, a automatização das marcações de tempo e distâncias parciais percorridas na pista, elimina a necessidade de marcação manual por um treinador ou pelo próprio atleta, além de fornecer informações mais precisas e confiáveis sobre a velocidade do atleta em cada ponto específico da pista. Com isso ele pode ter maior foco na execução da sua corrida, não se preocupando em olhar relógios ao longo do trajeto para conferir o seu tempo, o que na maioria das provas é algo impraticável, dada a biomecânica dos movimentos dos braços e tronco. Outra vantagem é a possibilidade do atleta treinar sozinho, não necessitando de outra pessoa para marcar o seu tempo, permitindo ao atleta treinar no horário que seja mais conveniente para ele.

Tendo em vista tais características do projeto e os benefícios que podem ser alcançados, espera-se que o mesmo possa agregar maior competitividade para a equipe de atletismo da UFPR, melhorando a autonomia dos atletas nos treinos individuais e fornecendo dados estatísticos mais precisos.

1.1 OBJETIVOS

1.2 Objetivo Geral

Desenvolver um dispositivo, a qual permita que um atleta sozinho em uma pista de atletismo possa marcar o seu tempo de percurso de forma precisa e passiva. O sistema proposto possuirá dois módulos de comunicação, um posicionado junto à linha de largada e o outro junto à linha de chegada, os quais terão comunicação *wireless*. Desta forma o módulo da saída saberá o momento em que o atleta partiu, podendo assim calcular o tempo percorrido pelo mesmo ao cruzar a linha de chegada e mostrá-lo em um *display*.

1.3 Objetivos Específicos

- Programar um cronômetro virtual com sincronia entre os módulos de comunicação, e que seja capaz de exibir o tempo percorrido na pista com precisão centesimal através de um *display*.
- Projetar e construir placas de circuito capazes de abrigar o ESP juntamente com os demais componentes do projeto;
- Realizar a simulação do projeto através de testes *indoor* e posteriormente na pista de atletismo da UFPR.
- Construir um encapsulamento para o circuito capaz de suportar as condições climáticas da pista de atletismo;

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

A automação desportiva vem sendo amplamente utilizada em países desenvolvidos para melhorar a qualidade dos treinos e das competições em diversas modalidades. Ela pode ser definida como a utilização de dispositivos os quais executam funções auxiliares de forma ativa, como marcações de tempos, pontuações, localizações de atletas ou objetos, dentre outros.

O foco deste trabalho será para as competições de atletismo, mais precisamente as provas de velocidade. Primeiramente serão vistas as características destas modalidades, bem como os conceitos envolvidos na sua prática. Posteriormente serão abordados conceitos relacionados ao dispositivo proposto neste projeto, como a *internet of things* (IoT), a propagação de sinais no espaço livre, além das características e funcionamento dos componentes escolhidos.

2.1 AS PROVAS DE VELOCIDADE NO ATLETISMO

A prova de velocidade mais popular do atletismo mundial certamente é a prova dos 100 metros rasos. É notável a melhoria nos índices alcançados pelos atletas ao longo da história, isto é, foram 113 anos entre o primeiro recorde masculino registrado na história, o recorde olímpico de 11,8 s de Thomas Burke, nos Jogos Olímpicos de Atenas, em 1896, até o recorde mundial atual, os 9,58 s de Usain Bolt, conquistado no Campeonato Mundial de 2009, em Berlim (GINCIENE; MATTHIESEN, 2012).

Porém, não só os atletas melhoraram no decorrer de pouco mais de um século, mas também as técnicas de treino, de corrida, alimentação, equipamentos e até mesmo a tecnologia de pavimentação das pistas de corrida.

2.1.1 As etapas da corrida de velocidade

As fases da corrida de 100 metros rasos podem ser divididas em:

- Fase 1 (0-20 m): aceleração inicial;
- Fase 2 (20-40 m): aceleração estendida;
- Fase 3 (40-50 m): estágio velocidade de pico inicial;
- Fase 4 (50-60 m): manutenção da velocidade I ;
- Fase 5 (60-70 m): manutenção da velocidade II;
- Fase 6 (70-80 m): velocidade máxima;
- Fase 7 (80-100 m): desaceleração.

É com base nessa divisão de fases, que pode ser visualizada na figura 2.1 que posteriormente será feito a divisão dos sensores ao longo da pista de atletismo, para que seja possível avaliar o desempenho do atleta em cada fase.

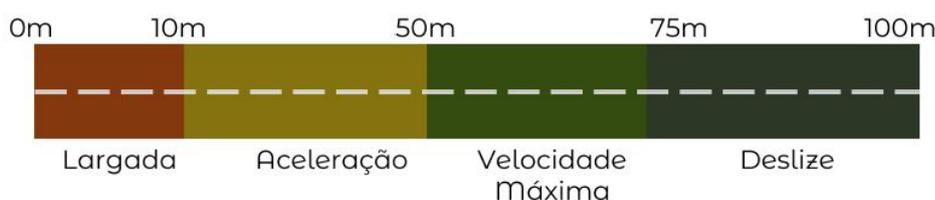


Figura 2.1: Fases da corrida

Fonte: O autor (2018)

2.1.2 A pista e os fatores determinantes

Em qualquer prova de pista ou nas provas genericamente chamadas de "corrida" a variável mais importante para um atleta é a velocidade.

Os treinadores de atletismo costumam analisar a velocidade de um ponto de vista técnico, visto que a velocidade de corrida de um atleta de velocidade, sob o ponto de vista da biomecânica, é o produto da frequência da passada de comprimento e passada (MACKALA, 2007).

Porém, no que diz respeito as medições, a velocidade será tratada como uma grandeza física vetorial, quociente de duas grandezas físicas escalares: o espaço e o tempo, como mostra a equação 2.1.

$$V = \Delta s/t \quad (2.1)$$

Através da Equação 2.1 fica evidente que é necessário mensurar do tempo relativo em que um atleta atinge certo ponto da pista, a qual possui distâncias conhecidas como mostra a Figura 2.2.

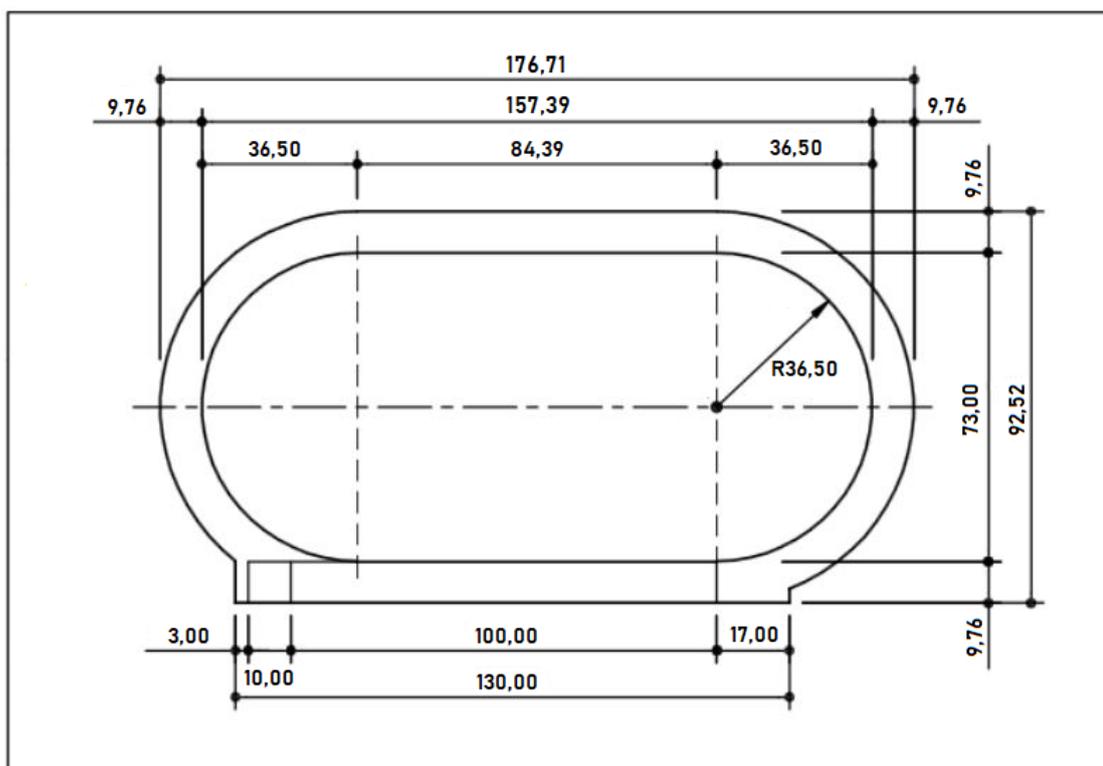


Figura 2.2: Dimensões da pista de atletismo

Fonte: (IAAF TRACK AND FIELD FACILITIES MANUAL, 2008)

Tendo as distâncias da pista conhecidas, e mensurando o tempo relativo em que o atleta passará por determinado trecho, será possível estabelecer padrões de velocidade de deslocamento do mesmo através da pista.

2.2 INTERNET OF THINGS (IoT)

A internet das coisas (IoT) é um paradigma inovador que está crescendo rapidamente e ganhando muito terreno no cenário atual, onde as redes Wi-Fi e telecomunicação possuem bastante influência. A ideia deste conceito é o simples fato de tudo estar

ligado de alguma maneira, dispositivos (sensores, celulares, atuadores, etc..) são capazes de interagir um com o outro, facilitando ainda mais o cotidiano do ser humano em diversos aspectos, fazendo com que tudo e todos estejam conectados.

A Internet das coisas (IoT) refere-se a conectividade online dos dispositivos com a capacidade de enviar e receber dados através da Internet. Telefones, eletrodomésticos, carros, relógios e muitos mais objetos do cotidiano são cada vez mais "inteligentes". Especialistas acreditam que até o final de 2015, 25 bilhões de dispositivos serão conectados à Internet e que, até 2020, o número pode crescer para 50 bilhões. Os benefícios afetam quase todos os domínios, incluindo cuidados de saúde, negócios e atividades domésticas. No entanto, o recolhimento de dados não autorizado através de sensores cria problemas de confidencialidade. Submeter suas informações pessoais, números de geolocalização ou conta bancária para qualquer dispositivo no IoT aumentam os problemas de segurança. A auto regulação por empresas, mesmo que em conjunto com a conscientização do usuário, é ainda insuficiente para fazer face a questões levantadas. Uma vez que com leis muito específicas deixam a IoT susceptível de diminuir seu potencial de crescimento, formuladores de políticas devem centrar-se na promulgação de leis de segurança de dados geral. Emitir avisos de consumidor pode impedir violações de segurança que levam ao uso indevido de dados. As leis de privacidade geral, se melhoradas, podem incutir confiança nos consumidores levando-os a experimentar essa tecnologia inteligente (FTC, 2015).

2.2.1 Conceitos de IoT

O termo *internet of things* é utilizado como uma palavra-chave para cobrir vários aspectos relacionados à extensão da Internet e da Web no âmbito físico, por meio da implantação generalizada de dispositivos distribuídos espacialmente com identificação, detecção e / ou capacidades de atuação. A IoT prevê um futuro no qual entidades digitais e físicas podem ser ligadas, por meio de tecnologias apropriadas de informação e comunicação, para possibilitar uma nova classe de aplicativos e serviços (MIORANDI et al., 2012).

Analisando hábitos cotidianos de um indivíduo moderno típico, como, por exemplo, na escola, na empresa e mesmo na vida privada fica evidente que esses e outros hábitos estão fortemente integrados a esse novo conceito de Internet das coisas. Outro exemplo: considere o uso também crescente de etiquetas inteligentes (*radiofrequency identification* - RFID) em grandes armazéns e lojas de varejo, conectando objetos a redes de computadores. Acrescente a isso a evolução das tecnologias de sensores e nanotecnologias que, quando associadas, contribuem para o aparecimento de um crescente número de objetos inteligentes. Vão de geladeiras que preparam listas de compras a veículos que ajudam seus motoristas a localizar caminhos via comunicação com *Global Position System* (GPS).

A tecnologia sem fios e a globalização são fatores importantes no desenvolvimento humano, que estão mutuamente relacionados. A mobilidade é a característica de definição da tecnologia sem fios. Embora a tecnologia sem fios seja global, as diferenças regionais emergiram em modelos de uso de acordo com a da tecnologia exigida. Nos Estados Unidos, a rádio é principalmente um instrumento de negócio; na Europa e a Ásia, a rádio é uma necessidade apenas caseira. A indústria sem fios é intensamente e globalmente competitiva. O computador e as indústrias de comunicações, incluindo rádio, convergem para uma integração. Dentro de algum tempo, os circuitos continuarão encolhendo-se e as baterias terão maior duração, evidenciando que a falta de faixa de espectro e as dificuldades de integração são dificuldades iminentes desta indústria. *wireless* (SHARMA; NAKAMURA, 2003).

2.2.2 Principais Tecnologias

A identificação de radiofrequência é uma tecnologia crescente para a IoT, mesmo dado o papel mais representativo da Internet banda larga para as aplicações cotidianas (XIA et al., 2012).

As aplicações de IoT formam um extenso espaço de design com várias dimensões que incluem: implantação; mobilidade contínua ocasional ou realizado por alguns ou todos as "coisas" no ambiente; custo, recursos e energia; heterogeneidade ilimitada

como diferentes hierarquias e modalidade de comunicação; a própria comunicação - rádio frequência eletromagnética, óptica, acústica, capacitiva e indutiva acoplado a infra-estrutura de comunicação; e também a topologia da rede, como por exemplo malha e/ou multi-camadas (SUNDMAEKER et al., 2010).

A tecnologia sem fios e a globalização reforçam mutuamente fatores no desenvolvimento da tecnologia. A mobilidade é a característica de definição da tecnologia sem fios. Embora a tecnologia sem fios seja global, as diferenças regionais emergiram em modelos de uso e as espécies da tecnologia exigida. Nos Estados Unidos, a rádio é principalmente um instrumento do negócio; na Europa e a Ásia, a rádio é uma utilidade doméstica. A indústria sem fios é intensamente e globalmente competitiva. O computador e as indústrias de comunicações (especialmente rádio) convergem. Os serviços de dados sem fios são especialmente valiosos quando os consumidores têm de responder imediatamente a fatos se modificam ou onde uma transação é sem complicação. com o passar do tempo, os circuitos continuarão encolhendo e as baterias durarão mais tempo. A falta de faixas espectro e as dificuldades de integração são questões resistentes na indústria *wireless*.

2.2.2.1 Radiofrequência

Nos tempos presentes os procedimentos de identificação por rádio tem sido muito usados em diversos segmentos da indústria, segmentos de logística, vendas e diversos outros. Esses procedimentos existem para prover informações sobre pessoas, bem e serviços, animais e produtos em transporte (FINKENZELLER, 2010).

Os códigos de barra se tornaram onipresentes em praticamente todos os produtos controlados algum tempo atrás, porém, hoje, esses códigos não se mostraram apropriados a diversas situações, uma vez que os mesmo são fixos e não reprogramáveis, mesmo sendo extremamente baratos. Passando pelos cartões magnéticos de contato usados principalmente por bancos, o qual exigem uma estação específica para contato direto, chega-se a conclusão de que uma fonte de transmissão de informações a distância é bem mais flexível para aplicações em qualquer campo (FINKENZELLER,

2010).

Uma estação geralmente contém um módulo de Rádio frequência(RF), compostos por transmissor e receptor, uma unidade de controle e um dispositivo de ligação com o *transponder*. Alguns dispositivos ainda possuem interface com outros sistemas como computadores, controladores etc. (FINKENZELLER, 2010).

2.3 MÓDULO DE COMUNICAÇÃO - ESP32

O módulo ESP32 é um SoC (*System on Chip*) com protocolo TCP/IP integrado que consegue dar a qualquer microcontrolador o acesso a rede Wi-Fi. Este módulo possui potencial de hospedar uma aplicação sozinho ou descarregar as funções de redes Wi-Fi a partir de outro processador. Cada módulo ESP32 possui uma pré-programação com comandos básicos de configuração e acionamento quando ele está sob controle de um equipamento externo no seu *firmware* (ESPRESSIF, 2017)..

O módulo foi escolhido para o projeto por possuir um processador poderoso na placa, que faz uma integração satisfatória com sensores e outras aplicações específicas usando os GPIOs, com necessidade de pouco desenvolvimento. O chip é bastante completo, e por conta disso o ESP32 requer poucos componentes na placa, fazendo com que ocupe pouco espaço no protótipo. Além disso, o ESP32 suporta APSD (Automatic Power Save Delivery) para aplicações VOIP (Voice over Internet Protocol) e interface com *Bluetooth*.

Dentre as principais características técnicas do ESP32 vale citar o processador *dual-core* ESP32, LX6, de 32 *bits* com *clock* de 160MHz, memória RAM de 512Kb e *Flash* de 16Mb suporte ao protocolo 802.11 b/g/n (ESPRESSIF, 2017).

2.3.1 Tecnologia LoRa e o protocolo LoRaWAN

LoRa® é uma tecnologia sem fio desenvolvida pela Semtech Corporation, a qual utiliza uma técnica de modulação por espalhamento espectral na faixa de Sub-GHz, permitindo uma cobertura de longo alcance, baixo consumo de energia, alta capacidade

de rede, comunicação robusta e capacidade de localização (MICROCHIP, 2018).

O LoRaWAN™ define o protocolo de comunicação e a arquitetura do sistema para a rede, enquanto a camada física LoRa® permite o link de comunicação de longo alcance. O protocolo e a arquitetura de rede têm a maior influência na determinação da vida útil da bateria de um nó, da capacidade da rede, da qualidade do serviço, da segurança e da variedade de aplicativos atendidos pela rede.

Os módulos enviam e recebem dados de Gateways específicos (similar as redes Wi-Fi, porém com alcance muito maior), que os encaminham via conexão IP para servidores locais ou remotos. O alcance de comunicação pode chegar a mais de 3 km dependendo do enlace. A Figura 2.3 apresenta um diagrama esquemático de uma rede com dispositivos LoRa (ALLIANCE, 2018).

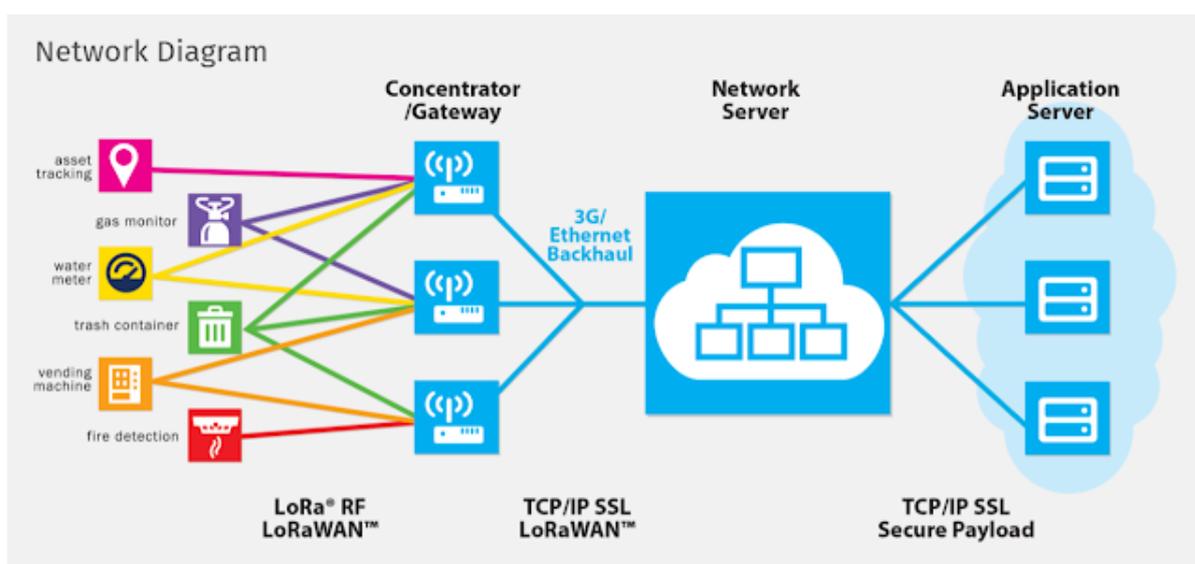


Figura 2.3: Diagrama de uma rede LoRa

Fonte: <https://lora-alliance.org>

2.3.2 ESP32 com rede LoRa

O ESP32 com rede LoRa®, combina o ESP32, um microcontrolador programável que inclui os protocolos Wi-Fi e Bluetooth, com um transceptor LoRa RFM95W para criar uma rede LoRa de canal único. Esta placa também pode servir como uma plataforma de desenvolvimento ESP32 de uso geral. Dessa forma pode ser utilizada como um dispositivo LoRa, podendo usar o poder do microcontrolador ESP32 para

monitorar sensores, hospedar em servidor web, realizar uma exibição, entre outras funcionalidades (SPARKFUN, 2018).

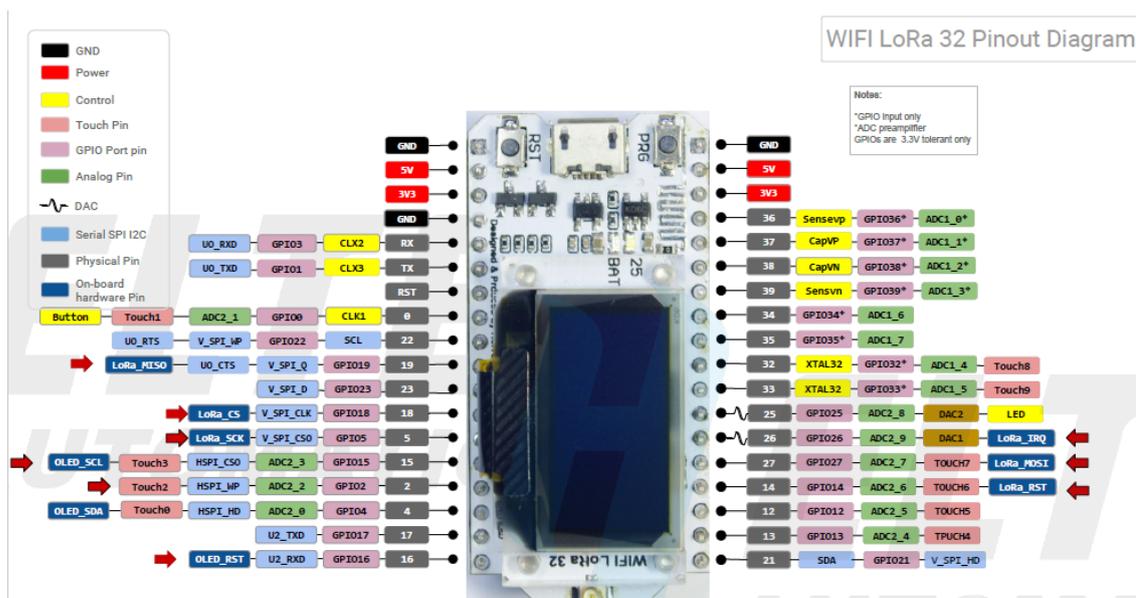
Neste trabalho será utilizado o módulo ESP32 com rede LoRa, desenvolvido pela Heltec, o qual possui um display OLED integrado à placa para visualização dos dados de comunicação. O módulo em questão pode ser visto na Figura 2.4.



Figura 2.4: ESP32 integrado com rede LoRa

Fonte: <https://pt.aliexpress.com>

A pinagem do ESP32, exemplificada na Figura 2.5 demonstra a utilização possível de cada pino. Alguns pinos desse modelo são *input only* e podem ser aproveitados para aplicações como botões ou leituras de sensores, para que não ocupem portas que precisam ser usadas como *input/output* como configuração de *displays* de 7 segmentos. Alguns pinos do ESP32 já estão sendo usados internamente no modelo de placa da Figura 2.4 para rotinas de controle dos dispositivos *built-in* como o *display* OLED e dispositivo de rede LoRa.



espaço livre à uma distância d do transmissor (FRIIS, 1946).

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (2.2)$$

Onde P_t é o sinal de potência transmitido. G_t e G_r representam os ganhos do transmissor e receptor respectivamente. $L(L \geq 1)$ é a perda do sistema, e λ é o comprimento de onda.

A perda de percurso do espaço livre (FSPL) é a perda na força do sinal de uma onda eletromagnética que resulta de um caminho de linha de visada através do espaço livre (geralmente ar), sem interferência de obstáculos que podem causar reflexão ou difração. De acordo com o padrão IEEE é definida como "A perda entre dois radiadores isotrópicos no espaço livre, expressa como uma relação de potência". Esta relação é geralmente expressa em dB (BERNARD, 2001).

A perda de percurso do espaço livre é proporcional ao quadrado da distância entre o transmissor e o receptor, e também proporcional ao quadrado da frequência do sinal de rádio, como pode ser observado na Equação 2.3.

$$FSPL = \left(\frac{4\pi df}{c} \right)^2 \quad (2.3)$$

Onde:

- λ é o comprimento de onda do sinal (em metros);
- f é a frequência do sinal (em hertz);
- d é a distância a partir do transmissor (em metros);
- c é a velocidade da luz no vácuo ($2,99792458 \times 10^8 m/s$).

A expressão do FSPL (2.3) pode gerar uma interpretação errônea de que o espaço livre atenua uma onda eletromagnética de acordo com sua frequência. Este não é o caso, pois não há mecanismo físico que possa causar isso. A expressão para FSPL na verdade possui dois efeitos, os quais serão explicados a seguir.

a) Dependência da Distância

A dependência da FSPL sobre a distância é causada pela propagação da energia eletromagnética no espaço livre e é descrita pela lei do inverso do quadrado, que é dada pela Equação 2.4.

$$S = P_t \frac{1}{4\pi d^2} \quad (2.4)$$

Onde:

- S é a potência por unidade de área ou densidade espacial de potência (em watts por metro quadrado) à uma distância d ;
- P_t é a potência radiada isotropicamente equivalente (em watts).

b) Dependência da Frequência

A perda de percurso é uma função da frequência pois é definida sobre o uso de uma antena receptora isotrópica ($G_r = 1$). O que pode ser comprovado derivando a FSPL a partir da equação de transmissão de Friis (Equação 2.5).

$$FSPL = \frac{P_t}{P_r} G_t G_r \quad (2.5)$$

Dessa forma a perda de percurso é uma ferramenta conveniente, pois representa uma hipotética perda de potência recebida que ocorreria se a antena receptora fosse isotrópica. Assim, a dependência de frequência da perda de percurso é causada pela dependência da frequência da abertura da antena receptora, no caso de o ganho da antena ser fixo. A abertura da antena, por sua vez, determina quão bem uma antena pode captar energia de uma onda eletromagnética de entrada (BERNARD, 2001). A dependência da abertura da antena a partir do ganho da antena é descrita pela Equação 2.6.

$$A = G \frac{\lambda^2}{4\pi} \quad (2.6)$$

Esta fórmula representa um fato bem conhecido, que quanto menor a frequência

(quanto maior o comprimento de onda), maior será a antena necessária para se obter certo ganho de antena. Portanto, para uma antena teórica isotrópica ($G_r = 1$), a potência recebida P_r é descrita pela Equação 2.7.

$$P_r = S \frac{\lambda^2}{4\pi} \quad (2.7)$$

Onde S é a densidade de potência de uma onda eletromagnética em um local de antena receptora isotrópica teórica, a qual é totalmente dependente do comprimento de onda, que é como o comportamento dependente da frequência surge.

A Figura 2.6 apresenta um gráfico da FSPL em função da distância para algumas frequências fixas.

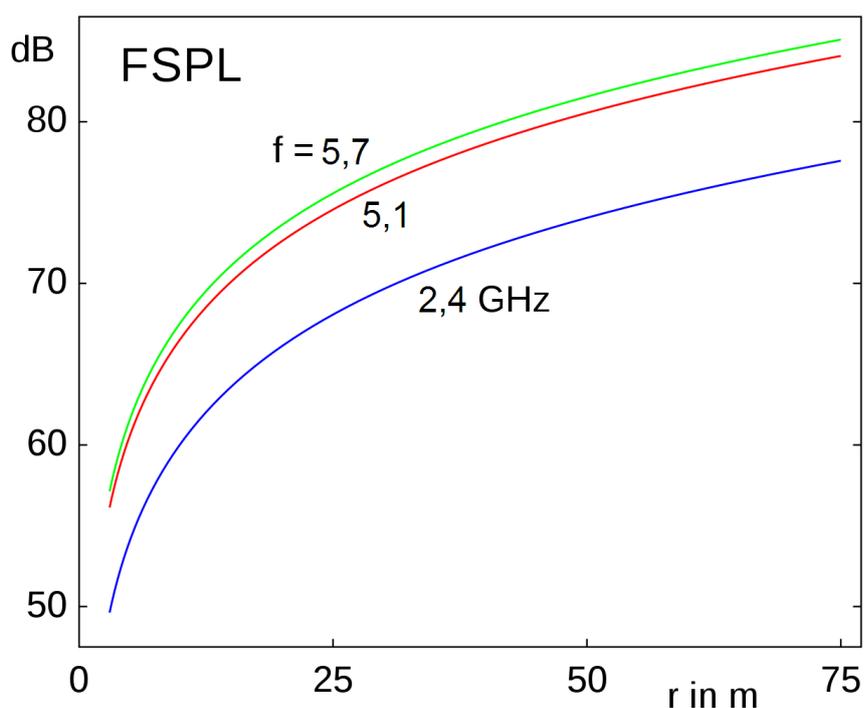


Figura 2.6: FSPL em função da distância para as frequências de 2,4 GHz, 5,1 GHz e 5,7 GHz.

Fonte: <https://ipfs.io/wiki/Free-space-path-loss.html>

CAPÍTULO 3

DESENVOLVIMENTO

Após um estudo do cenário envolvido no projeto e das tecnologias utilizadas no seu desenvolvimento, o presente capítulo tem como objetivo apresentar uma visão geral do sistema proposto, bem como descrever o seu funcionamento e a escolha dos componentes utilizados. A Figura 3.1 apresenta um diagrama simplificado das principais etapas envolvidas no desenvolvimento do projeto.

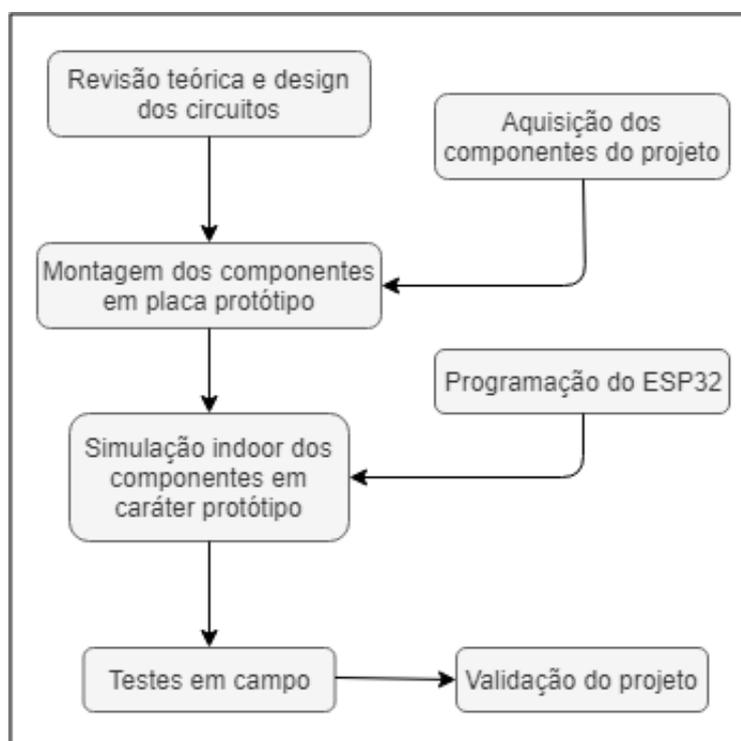


Figura 3.1: Diagrama de atividades

Fonte: O autor (2018)

Primeiramente foram projetados os circuitos para os módulos utilizados com a ajuda de um software de desenvolvimento online(Circuito.IO). Em seguida foram adquiridos os componentes necessários em cada módulo. Após a aquisição dos componentes foram realizadas a montagem e simulação *indoor* dos componentes em caráter protótipo. Paralelamente foi realizada a programação dos microcontroladores ESP32, de acordo com o funcionamento esperado do sistema. Esta programação será abordada posterior-

mente. Na sequência foram projetados os circuitos esquemáticos para a confecção das placas usinadas utilizando-se o software Eagle. Após a montagem dos componentes foram realizados os testes na pista de atletismo da UFPR.

3.1 METODOLOGIA

O monitoramento do atleta na pista de atletismo será iniciado através do acionamento de um botão no módulo de largada o qual faz uma requisição para sincronizar com o módulo de chegada, enquanto esse processo acontece um LED de cor verde acende intermitentemente para mostrar que o módulo entrou no processo de sincronização e o mesmo LED permanece aceso quando o pedido for respondido pelo módulo da chegada, para sinalizar que os módulos estão sincronizados e o atleta pode se preparar para a partida. Após a sincronização um contador de 10 segundos (preparando largada) é acionado, e o *buzzer* sinaliza o final desse contador e o começo do contador de 20 segundos (às suas marcas) que ao final emitirá o sinal sonoro através do *buzzer* simulando o comando "set", onde o atleta ergue o quadril, prestes a deixar o bloco de partida. O próximo comando sonoro que simboliza o "tiro de largada" virá em um tempo aleatório entre 0,3 e 1,4 segundos, para evitar que o atleta "adivinha" a largada após usar o sistema algumas vezes. Após partir o módulo da largada envia um sinal indicando a partida do atleta e ao receber tal sinal o módulo da chegada captura o horário do *clock* interno em milisegundos. No módulo de chegada será posicionado um sensor ultrassônico (modelo HCSR04) para detectar a passagem do atleta e capturar o tempo de chegada do *clock* interno.

Com o tempo da largada, o atraso e o tempo da chegada o processador tem os parâmetros necessários para calcular o tempo de percurso de acordo com a equação 3.1

$$T_{tot} = T_{CH} - (T_L + Atr) \quad (3.1)$$

onde:

- T_{tot} = Tempo total de percurso

- T_{CH} = Tempo de chegada
- T_L = Tempo de largada
- Atr = Atraso de comunicação

Tal Atraso foi calculado em modelo teórico no capítulo 2 e o mesmo está na grandeza de centésimos de segundo, visto que nesse faixa o mesmo não é influente no tempo registrado, poderia ser descartado; no entanto as condições ideais não levam em conta fatores como interferência por outras ondas e atrasos de *hardware*, portanto a exatidão das marcações terá maior confiabilidade se o atraso for calculado. Neste processo o módulo da largada envia uma constante e o módulo de chegada retorna a mesma constante, e com o tempo do *clock* da saída e da chegada dessa informação o módulo calcula a variável "Atraso". Porém consideramos como atraso apenas a informação que vaja da largada até a chegada, porém só conseguimos calcular o atraso das operações de ida e volta juntas, então considerando que os módulos são idênticos e a operação é idêntica, aproximamos com grande confiabilidade o atraso de ida para metade do atraso total, como mostrado na Equação 3.2.

$$Atr = (Atr_{ida} + Atr_{volta})/2 \quad (3.2)$$

onde:

- Atr = Tempo total de atraso
- Atr_{ida} = Tempo de atraso da ida
- Atr_{volta} = Tempo de atraso da volta

A programação dos microcontroladores ESP32 de cada módulo foi realizada utilizando a linguagem de programação C, através da IDE Arduino. Para um melhor entendimento das etapas de programação, foram utilizados como base os fluxogramas das Figuras 3.2 e 3.3.

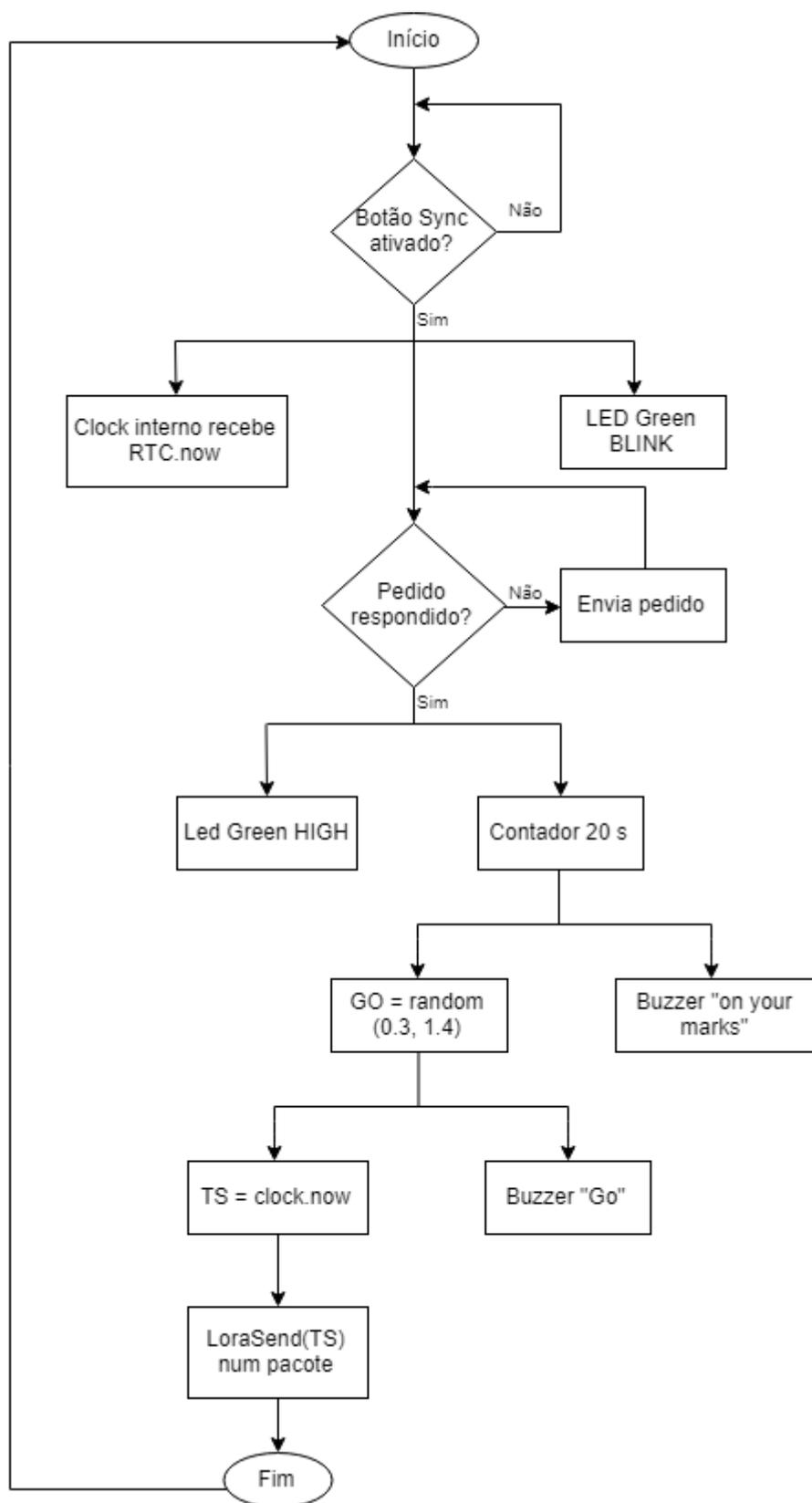


Figura 3.2: Fluxograma para a programação do módulo ESP da largada

Fonte: O autor (2018)

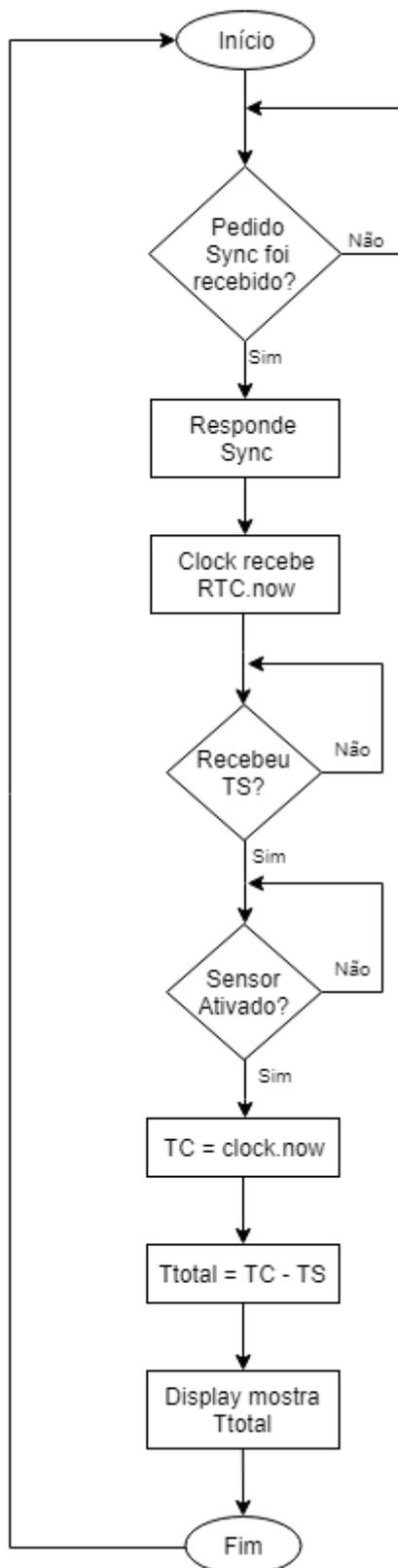


Figura 3.3: Fluxograma para a programação do módulo ESP da chegada

Fonte: O autor (2018)

3.2 Circuitos desenvolvidos

Nesta seção serão abordados os arranjos utilizados para o desenvolvimento dos circuitos dos módulos de largada e chegada, além de uma breve explicação sobre o funcionamento de alguns componentes chave do projeto.

3.2.1 Módulo de largada

O circuito do módulo de largada será posicionado na pista como ilustra a Figura 3.4, próximo a linha de largada. Este módulo contará com um botão e um *buzzer*, para sinalizar as etapas da largada do atleta. Quando o botão for pressionado, o módulo buscará a sincronização com o módulo da chegada. Quando a comunicação for estabelecida o *buzzer* enviará sinais sonoros indicando o posicionamento do atleta para a largada: o sinal de preparação e o sinal para partida. Este módulo contará ainda com dois LEDs de sinalização, um para indicar que o circuito está ligado e o outro para indicar a comunicação com o módulo da saída. Um circuito esquemático ilustrativo do módulo de largada é ilustrado na Figura 3.5.

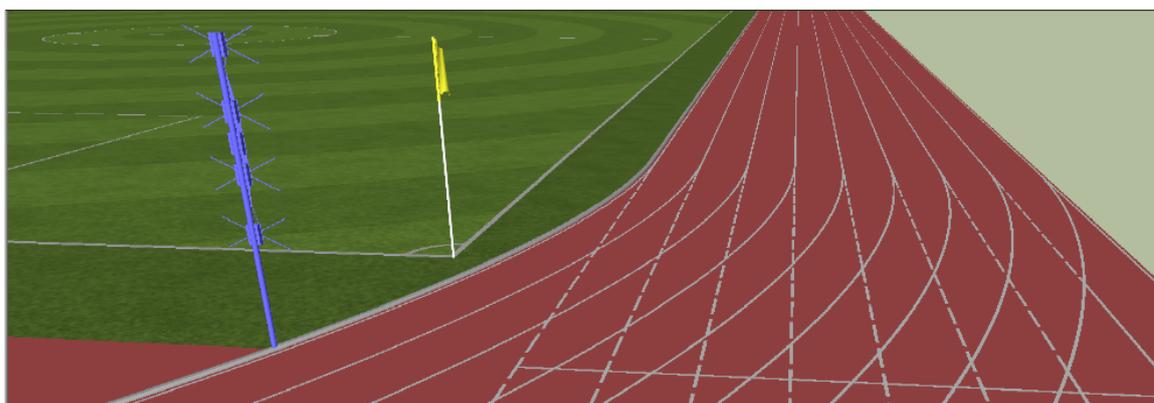


Figura 3.4: Posicionamento do módulo de largada

Fonte: O autor (2018)

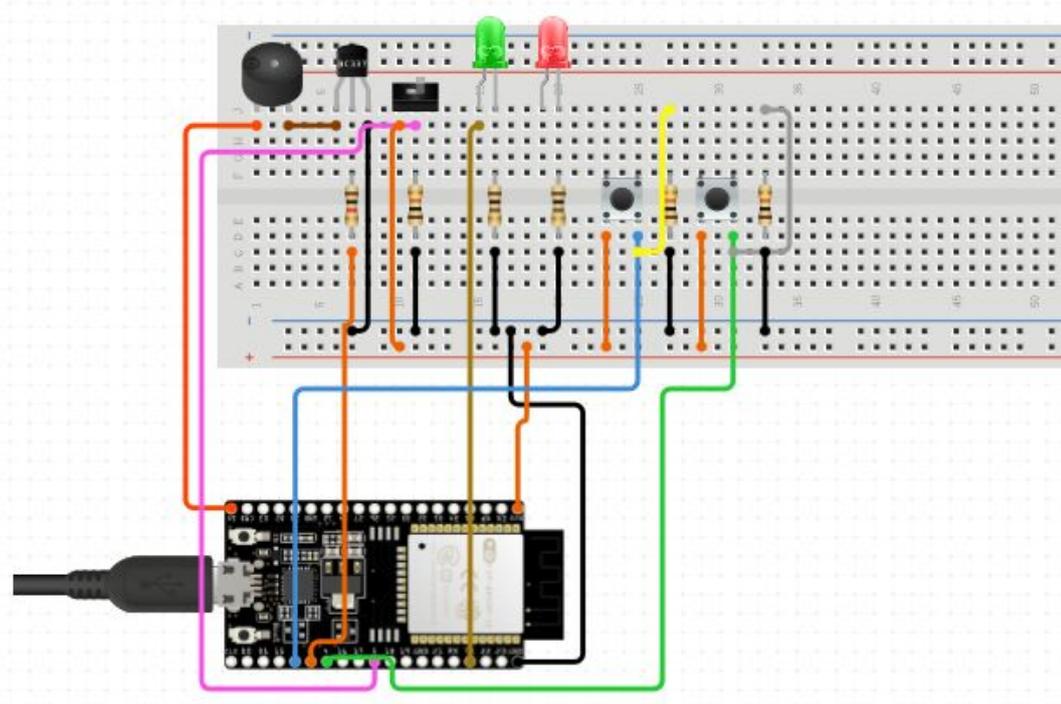


Figura 3.5: Ilustração do módulo de largada

Fonte: O autor (2018)

3.2.2 Módulo de chegada

O circuito do módulo de chegada, posicionado na pista como ilustra a Figura 3.8, contará com um sensor ultrassônico para detecção da passagem do atleta pela linha de chegada e um *display* (TM1637) para mostrar o tempo de percurso. Assim como no módulo de largada, o circuito será alimentado com uma bateria de 9 V e regulado para 5 V utilizando um transistor L7805. O circuito esquemático do módulo de chegada pode ser visto na Figura 3.7.

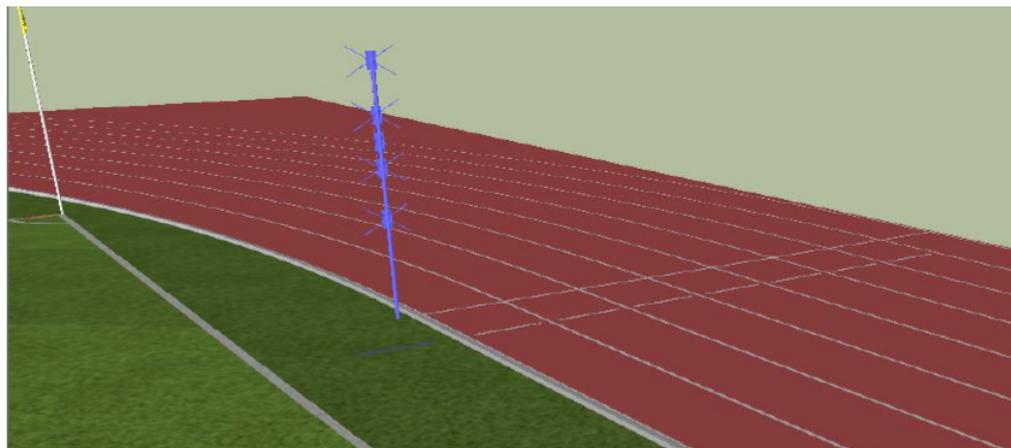


Figura 3.6: Posicionamento do módulo de chegada

Fonte: O autor (2018)

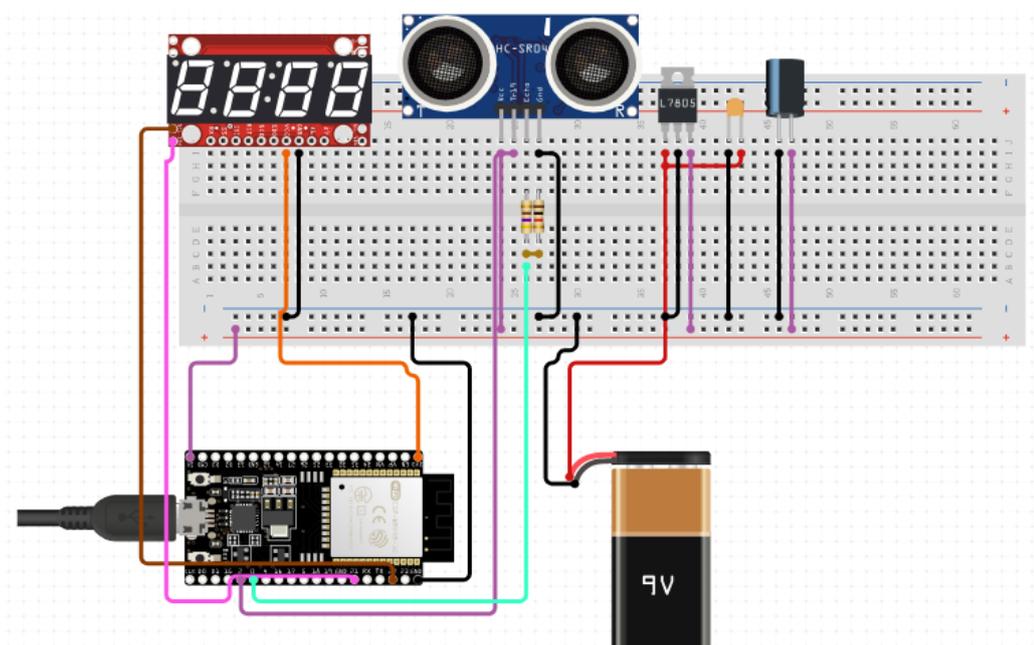


Figura 3.7: Ilustração do módulo de chegada

Fonte: O autor (2018)

O princípio de funcionamento dos sensores ultrassônicos está baseado na emissão de ondas sonoras de alta frequência, e na medição do tempo levado para a recepção do eco produzido quando a onda é refletida por um objeto. O sensor pode calcular a distância do alvo detectado com base no intervalo de tempo entre a emissão do sinal e o recebimento do eco.

O modelo do sensor utilizado neste projeto foi HC-SR04 (Figura), o qual é capaz

de medir distâncias de 2cm a 4. Este módulo possui um circuito com emissor e receptor acoplados e 4 pinos (VCC, Trigger, ECHO, GND). Os sinais ultrassônicos são enviados pelo pino trigger, se algum sinal de retorno for identificado pelo receptor (echo), o sensor gera um sinal de nível alto no pino de saída cujo tempo de duração é igual ao tempo calculado entre o envio e o retorno do sinal ultrassônico. Este sinal é detectado pelo módulo ESP resultando no travamento da contagem do cronômetro.



Figura 3.8: Sensor ultrassônico HC-SR04

Fonte: <https://www.eletrogate.com>

O modelo do display utilizado no projeto é o TM1637, o qual pode ser visto na Figura 3.9. O seu funcionamento é baseado na multiplexação de 4 *displays* de 7 segmentos usando o circuito TM1637, o qual opera recebendo os pulsos de atualização do display por apenas 2 pinos, além do pino Vcc e GND, como pode ser visto no esquemático da Figura 3.10

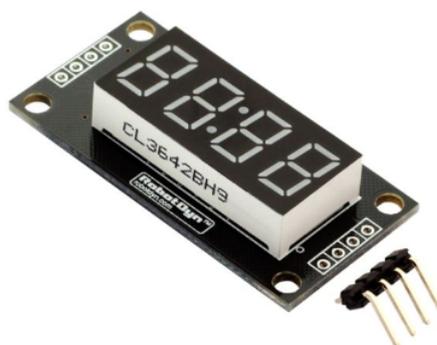


Figura 3.9: Display TM1637

Fonte: www.robotdyn.com

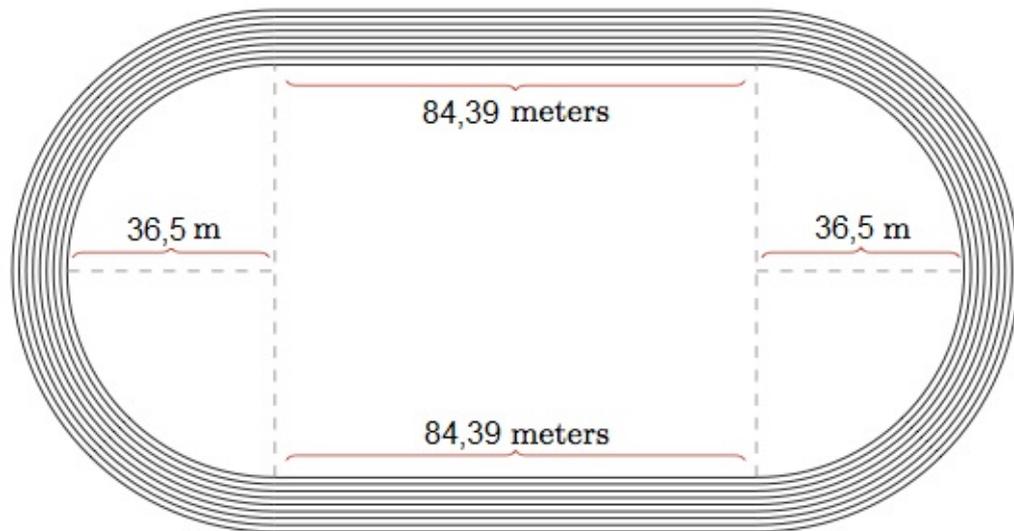


Figura 3.11: Pista de atletismo simplificada

Fonte: (ILLUSTRATIVE MATHEMATICS, 2018)

Um módulo instalado no começo de uma das curvas, e no final da reta oposta estaria a uma distância que pode ser calculada usando a equação:

$$D_{max} = \sqrt[2]{(84.39^2 + 36.5^2)} \quad (3.3)$$

O que resulta em:

$$D_{max} \approx 110m \quad (3.4)$$

Portanto se 2 módulos forem capazes de uma comunicação a essa distância sem que haja atrasos consideráveis, todos os módulos intermediários estariam em condições iguais ou melhores de funcionamento.

Posteriormente é necessário usar essa distância encontrada para cálculos mais complexos que utilizarão dados do modelo de antena nativa do ESP32, a distância D_{max} encontrada, a potência de transmissão no ESP32 e a potência mínima esperada no módulo receptor, para que então, seja possível concluir se existe necessidade de instalação de uma antena externa e se existir, qual o ganho necessário para tal antena.

CAPÍTULO 4

RESULTADOS E DISCUSSÃO

A operação do cronômetro foi simplificada para a utilização de um atleta leigo em eletrônica, tendo seu uso restrito a apenas um botão para o usuário, o qual inicia toda a operação de sincronia, disparos e marcações. O LED vermelho sinaliza se o circuito está alimentado, e caso esteja apagado será fácil notar pelo usuário que o módulo está com a bateria fraca e necessita troca, algo que não acontecerá com frequência dado o baixo consumo de energia pelo sistema. O LED verde que sinaliza o processo de sincronia e posteriormente a partida preparada também é um sinal fácil de entender para o usuário.

O sistema foi projetado com a premissa de conseguir exatidão centesimal e a mesma pôde ser alcançada através do cálculo do atraso de comunicação, que ficou em médias próximas a 1038 ms na maior parte dos testes de campo e como o mesmo é calculado a cada ciclo de marcação qualquer variação será medida e incorporada ao tempo total, porém a variável "atraso" será iniciado no código com o valor 1046, por ser o valor mais frequente em testes e em caso de perda de pacote na marcação, o atleta não perca a medição e possa ter seu tempo calculado baseado no atraso médio das medições em pista.

Os circuitos esquemáticos e *layouts* para o desenvolvimento das placas de circuito impresso foram implementados utilizando o *software* Eagle. O circuito esquemático e *layout* para o módulo de largada são apresentados nas Figuras 4.1 e 4.2. Já o circuito esquemático e *layout* para módulo da chegada podem ser vistos nas Figuras 4.3 e 4.4.

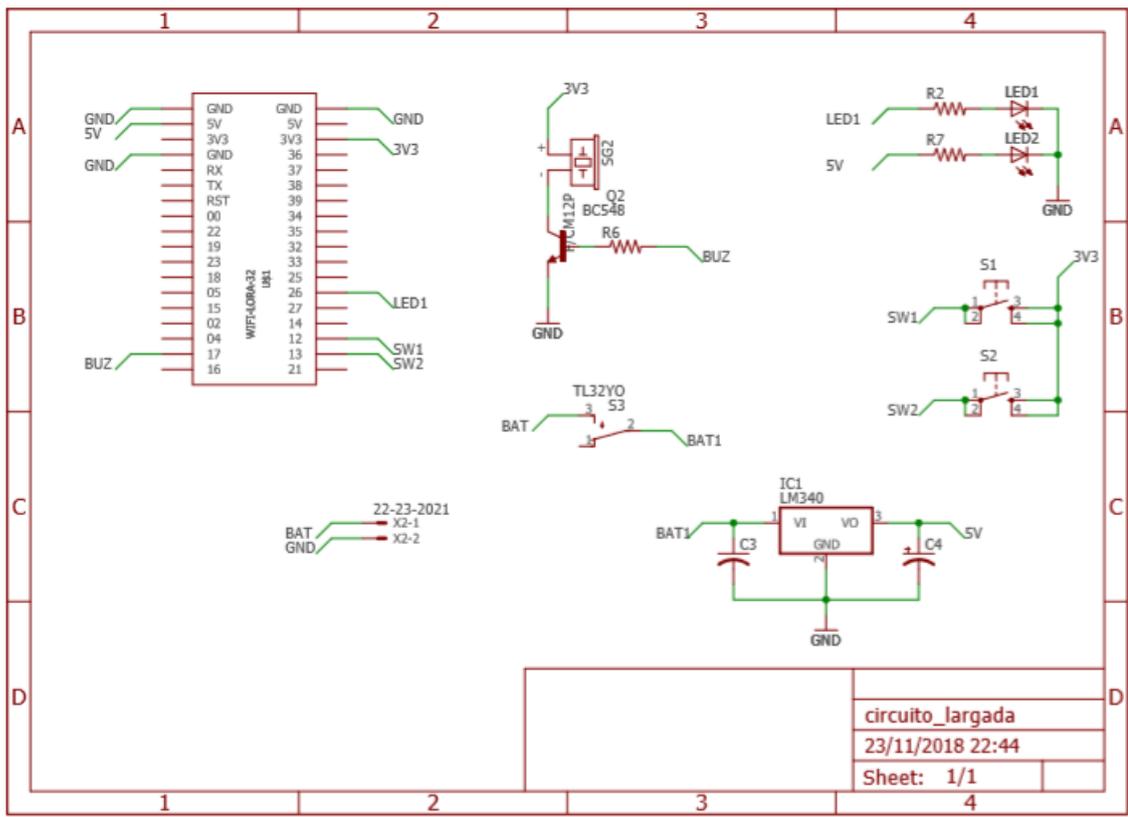


Figura 4.1: Circuito esquemático para a PCB da largada

Fonte: O autor (2018)

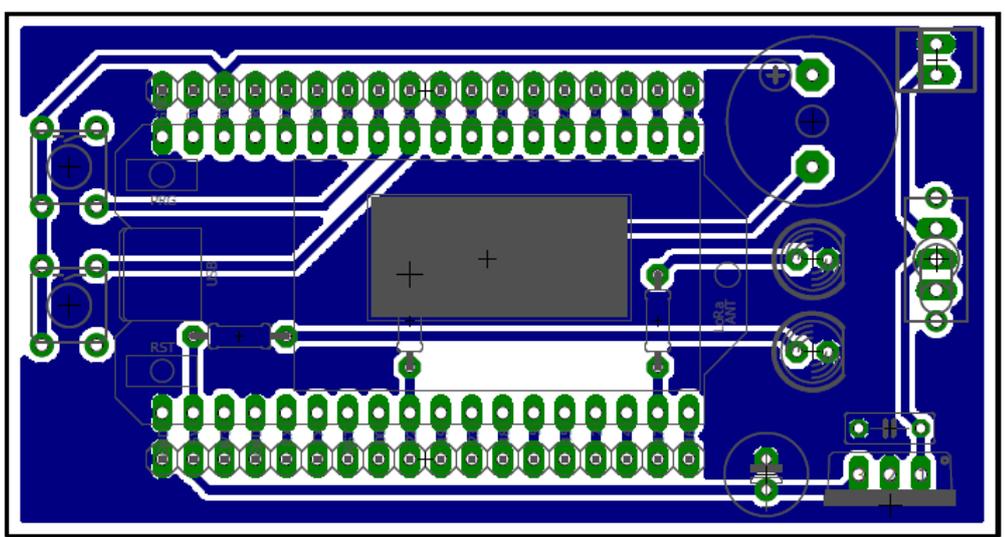


Figura 4.2: Layout para a PCB da largada

Fonte: O autor (2018)

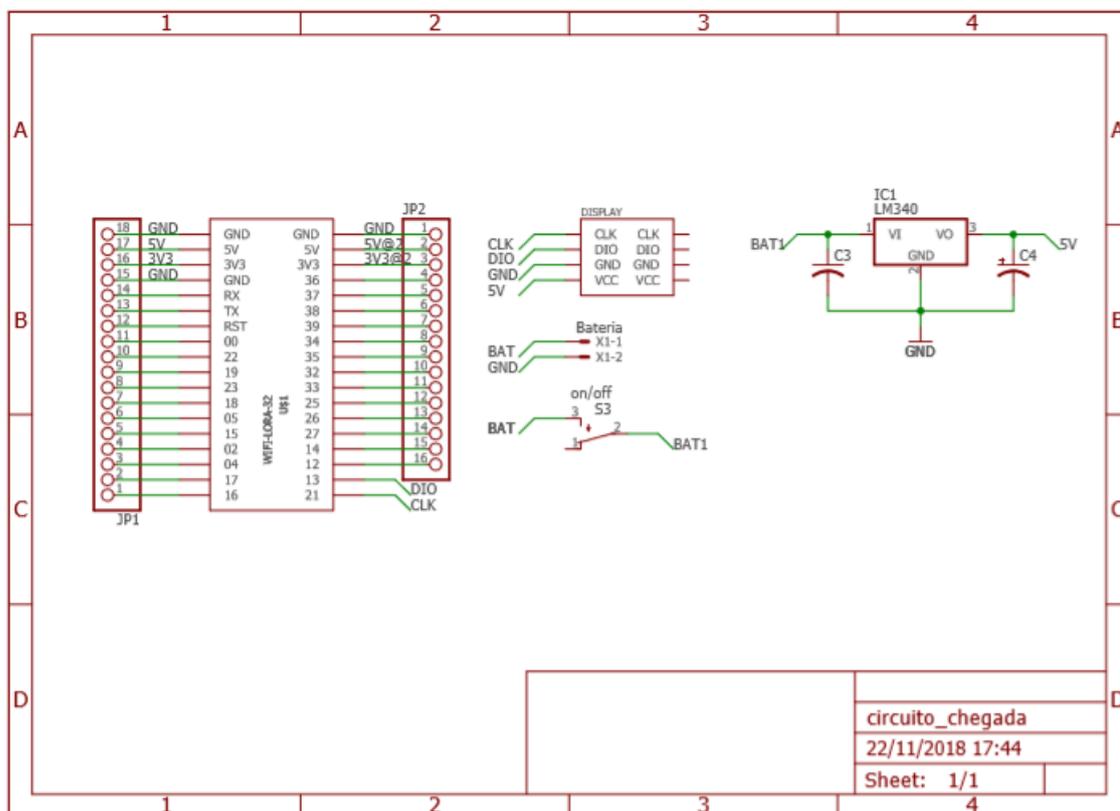


Figura 4.3: Circuito esquemático para a PCB da chegada

Fonte: O autor (2018)

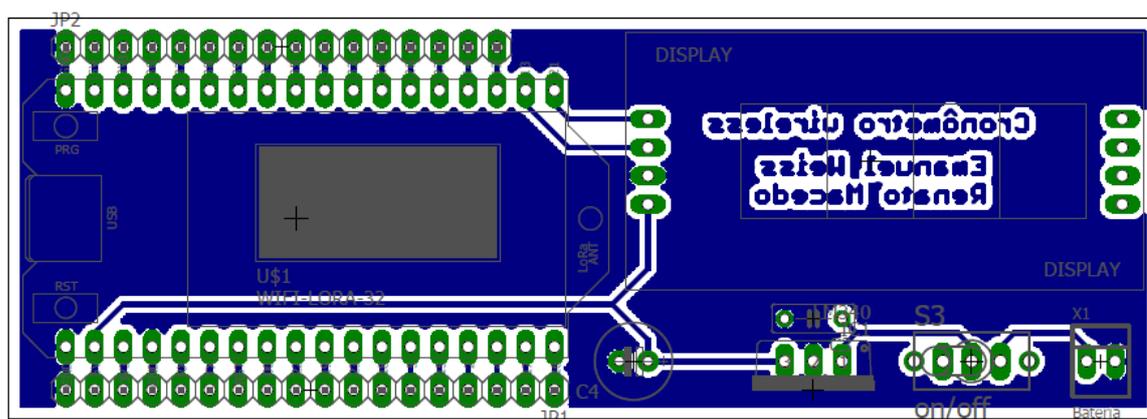


Figura 4.4: Layout para a PCB da chegada

Fonte: O autor (2018)

No funcionamento do código programado é possível analisar a comunicação entre os 2 módulos quando conectados a uma porta serial USB e é possível visualizar as variáveis enviadas e recebidas como mostra a Figura 4.5 e na Figura 4.6

```

COM7
|
|
recebeu pedidoSYNC/ STATUS = 0
Enviando respondeSYNC
STT 0// i 0// TS // k 0
STT 0// i 1// TS // k 0
STT 0// i 2// TS // k 0
STT 0// i 3// TS // k 0
STT 0// i 4// TS // k 0
STT 0// i 5// TS // k 0
STT 0// i 6// TS // k 0
STT 0// i 7// TS // k 0
STT 0// i 8// TS // k 0
STT 0// i 9// TS // k 0
STT 0// i 10// TS // k 0
STT 0// i 11// TS // k 0
STT 1// i 12// TS 83070// k 5
aguardando sensor // Tcron = 47870// Distância = 0 cm
distancia 0
adj 10.00// numc 83070// TC 49076
Ttot -33994 Ttot/adj -3399.40 Ttot/TotDsp -3399// Tcron2= 1206

```

Figura 4.5: Captura da porta serial da chegada

Fonte: O autor (2018)

```

COM7
|
|
Enviando pedidoSYNC
Enviando pedidoSYNC
Enviando pedidoSYNC
0
1073422196
0
1073422196
1
1073422196
2
recebeu respondeSYNC/ STATUS = 0
1
3
1
Enviando 110797

```

Figura 4.6: Captura da porta serial da largada

Fonte: O autor (2018)

Foram realizados testes *indoor* do sistema operando em todas as suas etapas. Na Figura 4.7 é possível ver o módulo de largada pronto para sincronização. Na Figura 4.8 é mostrado o módulo de chegada no modo *stand by*, com o valor inicial pré-programado de 9,58 s, já na Figura 4.9 pode ser visto o módulo de chegada após a detecção da passagem do atleta, com a indicação do tempo de percurso no *display*.

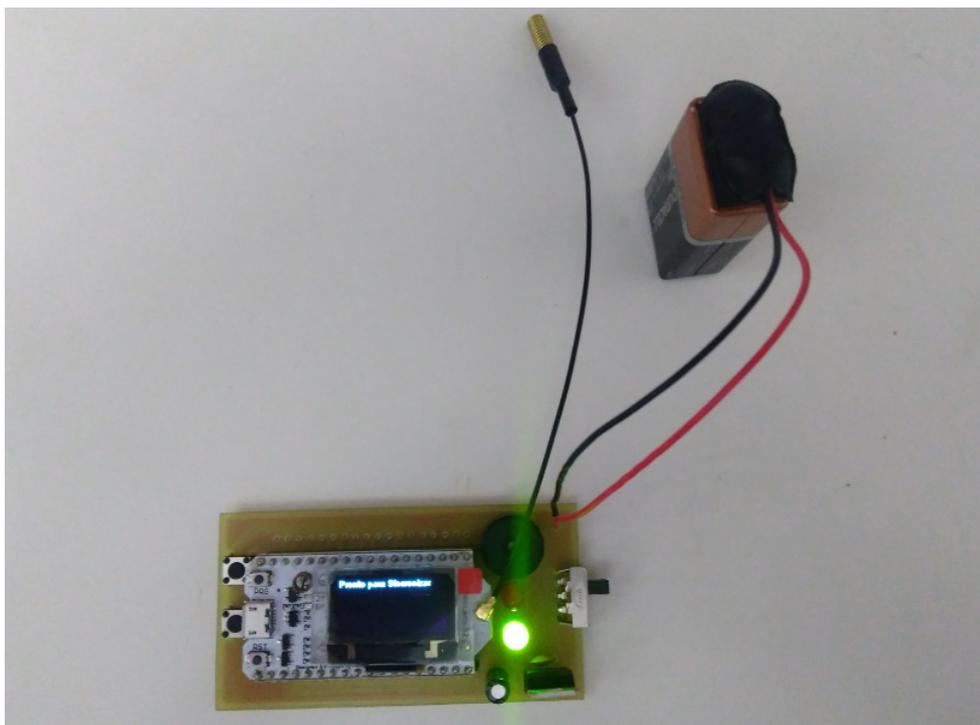


Figura 4.7: Foto do módulo de largada

Fonte: O autor (2018)

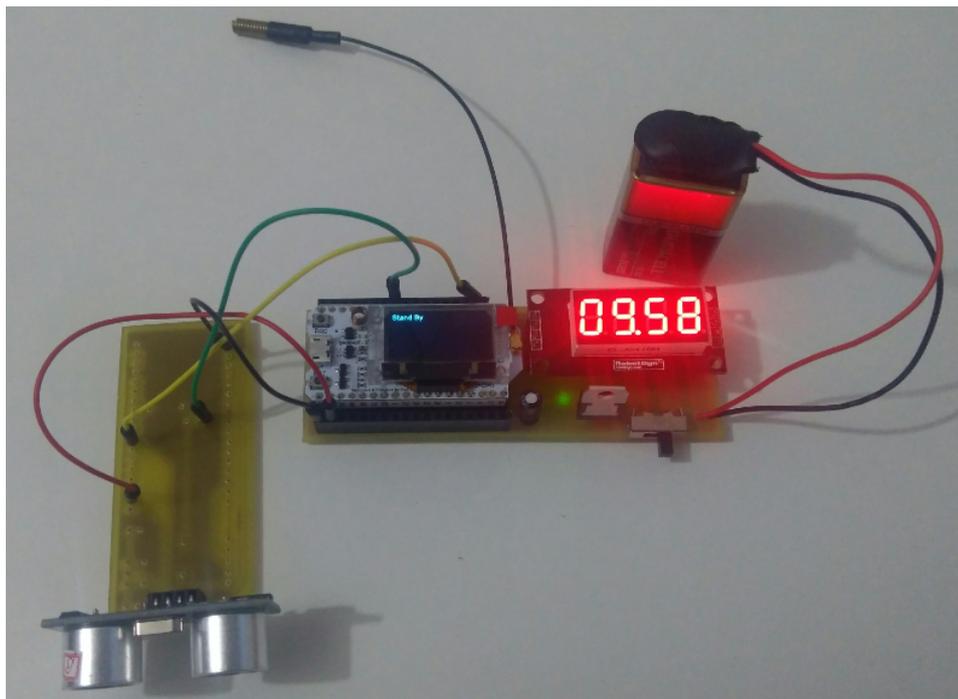


Figura 4.8: Foto do módulo de chegada no modo *stand by*

Fonte: O autor (2018)

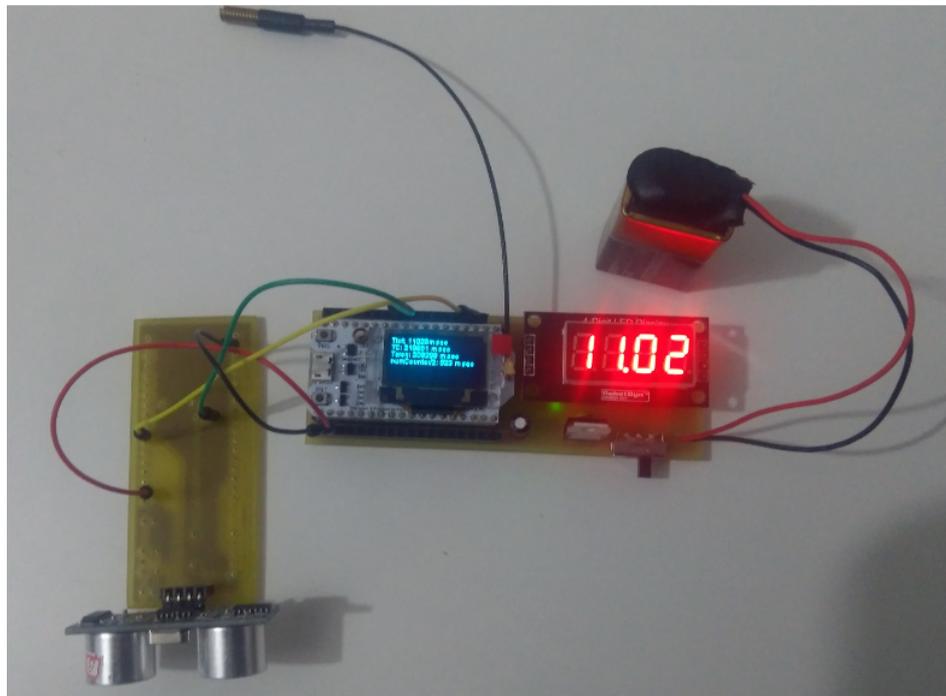


Figura 4.9: Foto do módulo de chegada após a marcação do tempo

Fonte: O autor (2018)

CAPÍTULO 5

CONCLUSÃO

O trabalho apresentado tem um objetivo bem definido: automatizar as marcações de tempo dos atletas em pistas de atletismo. Outra contribuição deste projeto será o desenvolvimento dos alunos nos conceitos fundamentais de IoT e na utilização de módulos de comunicação como o ESP32. Esse conhecimento pode ser aplicado em outras áreas importantes como a automação residencial, *smart grids*, entre outras.

A interface facilitada será de grande valia para usuários reais que por um lado não detêm conhecimentos em eletrônica ao mesmo tempo que não desconcentram o atleta de sua principal necessidade que é a corrida, gerando ganhos para o mesmo em:

- Ergonomia, pela facilidade de uso;
- Autonomia, uma vez que o atleta não precisa do auxílio de um terceiro para fazer sua própria marcação;
- Agilidade, pois a amostragem é rápida e pode ser feita consecutivas vezes;
- Exatidão, dado que as marcações são livres de interferência humana e podem até mesmo ser submetidas a fatores de correção caso necessário.

Além de indicações como um LED vermelho para sinalizar o circuito ligado e um LED verde para indicar que o módulo está pronto pra partir.

Dentre algumas dificuldades encontradas pode-se citar a interação entre os componentes e a aquisição dos sensores a *laser* que não são muito comuns fora das aplicações industriais. Porém o sensor HC-SR04 se mostrou apto a realizar a medição pretendida com a precisão exigida. A interação entre o RTC escolhido e o ESP32 não foi possível como planejado inicialmente, visto que os RTC's acessíveis economicamente para os autores não cobriam requisitos de resolução e também não tinham comunicação com o a placa escolhida, mesmo após várias tentativas e métodos diferentes, incluindo fabricar uma biblioteca específica para o projeto em questão.

Tendo em vista que uma das premissas da IoT é a mobilidade o protótipo se mostrou de fácil transporte e utilização dado o tamanho reduzido, peso desprezível e

alimentação independente feita por baterias comuns de 9 V que podem encontradas no comércio. O *display* de 7 segmentos que mostra o tempo de percurso do atleta se revelou uma escolha certa, dado ao fato de que a luz solar ofusca displays de LED e telas semelhantes quando em local aberto como a pista de atletismo, porém o *display* escolhido possui um nível de brilho excelente e bom contraste, facilitando a visada.

5.1 TRABALHOS FUTUROS

Mesmo após o término do projeto poderão ser exploradas algumas implementações adicionais a fim de melhorar o funcionamento do projeto e também para fins de aprendizado para os desenvolvedores, tais como:

- Desenvolvimento de um aplicativo para interface de comando através do smartphone;
- Implementação de um RTC em cada módulo configurados com o mesmo horário para comparação entre os tempos de chegada e saída de acordo com os *clock* dos mesmos.
- Substituir o sensor ultrassônico por um sensor a *laser* do tipo LIDAR (light radar) devido a luz ter velocidade muito superior ao som e ser capaz de detectar a passagem do atleta com maior rapidez, elevando a confiabilidade da marcação.

Tais implementações também podem ser feitas por alunos do setor de Engenharia Elétrica da UFPR que se mostrarem interessados e queiram implementar melhorias para o sistema.

APÊNDICE A

CÓDIGOS DE PROGRAMAÇÃO

A.1 Cronometro Setup Principal

```
1 //Setup do main
2
3 #include <SPI.h>
4 #include <LoRa.h>
5 #include <Wire.h>
6 #include <SSD1306.h>
7 #include <Time.h>
8
9 #define CHEGADA
10
11 #define SCK 5
12 #define MISO 19
13 #define MOSI 27
14 #define SS 18
15 #define RST 14
16 #define DI00 26
17
18 #define BAND 433E6
19
20 const String SYNCDATA = "pedidoSYNC";
21 const String SYNCDATAZ = "respondeSYNC";
22 const String ATRASOI = "atrasoida";
23 const String ATRASOV = "atrasovolta";
24
25 String TIMEDATA = "";
26 String HEADER = "TCC";
27 long lastSendTime = 0;
28
29 int atraso = 1038; //delay medio
```

```
30
31
32 String TS = "";
33 String Tcheg = "";
34
35 int STTRCV = 0;
36 int TIME_MSG_LEN = 0;
37
38 int TIME_HEADER = 0;
39
40 int k = 0;
41 float adj =0;
42
43
44 int TotDsp = 0;
45 int TC = 0;
46 int Ttot =0;
47
48
49 SSD1306 display(0x3c, 4, 15);
50
51 void setupDisplay () {
52     pinMode(16, OUTPUT);
53     digitalWrite(16, LOW); digitalWrite(16, HIGH);
54
55     display.init();
56     display.flipScreenVertically();
57     display.setFont(ArialMT_Plain_10);
58     display.setTextAlignment(TEXT_ALIGN_LEFT);
59 }
60
61 void setupLoRa () {
62     SPI.begin(SCK, MISO, MOSI, SS);
63     LoRa.setPins(SS, RST, DI00);
64
```

```

65     if (!LoRa.begin(BAND, true)){
66         display.clear();
67         display.drawString(0, 0, "Erro ao inicializar LoRa!");
68         display.display();
69         while (1);
70     }
71
72     LoRa.enableCrc();
73     LoRa.receive();
74 }
75
76
77 //const int buttonPinZERA = 25;      // the number of the pushbutton pin
78
79
80 // variables will change:
81 int buttonState = 0;          // variable for reading the pushbutton status
82
83
84 void send(String DATAS){
85     //Inicializa o pacote
86     LoRa.beginPacket();
87     //Envia o que está contido em "DATAS"
88     LoRa.print(DATAS);
89     Serial.println("Enviando" + DATAS);
90     //Finaliza e envia o pacote
91     LoRa.endPacket();
92 }
93
94
95
96
97 int receive(String DATAR){
98     //Tenta ler o pacote
99     int packetSize = LoRa.parsePacket();

```

```

100
101 //Verifica se o pacote possui a quantidade de caracteres que esperamos
102 if (packetSize == DATAR.length()){
103     String received = "";
104
105     //Armazena os dados do pacote em uma string
106     while(LoRa.available()){
107         received += (char) LoRa.read();
108     }
109
110     if(received.equals(DATAR)){
111         //Simula a leitura dos dados
112         Serial.println("recebeu" + DATAR + "/" + STATUS + " + STTRCV);
113
114
115         return 1;
116     }else{
117
118     return 0;}
119 }
120
121
122 }
123
124 int receiveTS(String TEMPO){
125     //Tenta ler o pacote
126     int packetSize = LoRa.parsePacket();
127
128     //Verifica se o pacote possui a quantidade de caracteres que esperamos
129     if (packetSize == TEMPO.length() or packetSize == TEMPO.length() -1 or
        packetSize == TEMPO.length() + 1){
130         String received = "";
131         k = TEMPO.length();
132         //Armazena os dados do pacote em uma string
133         while(LoRa.available()){

```

```

134     received += (char) LoRa.read();
135 }
136
137     TS = received;
138
139     return 1;
140
141 }
142 return 0;
143
144 }
145
146 void atrasoLargada() {
147
148     //Marcamos o tempo que ocorreu o envio
149     lastSendTime = millis();
150     send(ATRASOI);
151     Serial.println("INTERVAL" + String(250) + "/" + STTRCV + " + String(
        STTRCV) + " / " + String(millis()) + " / " + lastSendTime + " + String
        (lastSendTime));
152
153
154     int i = 0;
155     do
156     {
157
158     STTRCV = 0;
159     STTRCV = receive(ATRASOV);
160     Serial.println("aguardando" + String(ATRASOV) + "/" + STTRCV + " +
        String(STTRCV));
161     display.clear();
162     display.drawString(0, 0, "aguardando" + String(ATRASOV));
163     display.display();
164     delay(1);
165     i++;

```

```

166 } while (STTRCV!=1 and i<150);
167 atraso = ( millis ()-lastSendTime );
168 Serial.println("atraso_" + String(atraso)+"/_miliis:_"+String(
        millis ())+"/_lastSendTime:_"+String(lastSendTime));
169 display.clear();
170 display.drawString(0, 0, "atraso_" + String(atraso));
171 display.display();
172 delay(500);
173 }
174
175
176 void atrasoChegada() {
177
178     int i =0;
179     do
180 { STTRCV = 0;
181 STTRCV = receive(ATRASOI);
182     Serial.println("aguardando_" + String(ATRASOI) + "/_STTRCV:_"+
        String(STTRCV));
183     display.clear();
184     display.drawString(0, 0, "aguardando_" + String(ATRASOI));
185     display.display();
186
187 } while (STTRCV!=1);
188
189     send(ATRASOV);
190     Serial.println("enviou_" + String(ATRASOV) + "/_STTRCV:_"+ String(
        STTRCV));
191     display.clear();
192     display.drawString(0, 0, "enviou_" + String(ATRASOV));
193     display.display();
194     delay(500);
195
196
197 }

```

A.2 Chegada

```

1
2 #ifndef CHEGADA
3 #define INTERVAL 500
4
5
6 #include <TM1637displ.h>
7 #include <Ultrasonic.h>
8
9 int counter = 0; //aux
10 int numCounter = 0; //display 7seg
11
12 const int echoPin = 34; //PINO DIGITAL UTILIZADO PELO HC-SR04 ECHO(RECEBE)
13 const int trigPin = 12; //PINO DIGITAL UTILIZADO PELO HC-SR04 TRIG(ENVIA)
14 Ultrasonic ultrasonic(trigPin,echoPin); //INICIALIZANDO OS PINOS
15 int distancia; //CRIA UMA VARIÁVEL CHAMADA "distancia" DO TIPO INTEIRO
16 String result; //CRIA UMA VARIÁVEL CHAMADA "result" DO TIPO STRING
17
18 const int CLK = 21; //Set the CLK pin connection to the displ
19 const int DIO = 13; //Set the DIO pin connection to the displ
20
21 TM1637displ displ(CLK, DIO); //set up the 4-Digit displ.
22 //funcao para operar o HCSR04
23 void hcsr04() {
24 digitalWrite(trigPin, LOW); //SETA O PINO 6 COM UM PULSO BAIXO "LOW"
25 delayMicroseconds(2); // DELAY DE 2 MICROSSEGUNDOS
26 digitalWrite(trigPin, HIGH); //SETA O PINO 6 COM PULSO ALTO "HIGH"
27 delayMicroseconds(10); // DELAY DE 10 MICROSSEGUNDOS
28 digitalWrite(trigPin, LOW); //SETA O PINO 6 COM PULSO BAIXO "LOW" NOVAMENTE
29 // FUNcao RANGING, FAZ A CONVERSAO DO TEMPO DE
30 //RESPOSTA DO ECHO EM CENTIMETROS E ARMAZENA
31 //NA VARIÁVEL "distancia"
32 distancia = (ultrasonic.Ranging(CM)); // VARIÁVEL GLOBAL RECEBE O VALOR DA
    DISTANCIA MEDIDA

```

```
33 delay(50); //INTERVALO DE 50 MILISSEGUNDOS
34 }
35
36
37
38 void setup() {
39     Serial.begin(115200);
40     //Chama a configuracao inicial do display
41     setupDisplay();
42     //Chama a configuracao inicial do LoRa
43     setupLoRa();
44     display.clear();
45     display.drawString(0, 0, "Esperando SYNC...");
46     display.display();
47     STTRCV = 0;
48     Serial.println(STTRCV);
49
50     displ.setBrightness(0x07); //set the diplay brightness
51
52     pinMode(echoPin, INPUT); //DEFINE O PINO 7 COMO ENTRADA (RECEBE)
53     pinMode(trigPin, OUTPUT); //DEFINE O PINO 6 COMO SAIDA (ENVIA)
54
55     TotDsp = 10;
56     Ttot =9580;
57 }
58
59
60 void loop() {
61
62
63     STTRCV = 0;
64     STTRCV = receive(SYNCDATA);
65
66     if (STTRCV == 1){
67
```

```

68 for (int i=0;i<8;i++){
69     send(SYNCDATAZ);
70     display.clear();
71     display.drawString(0, 0, "Enviando_" + SYNCDATAZ);
72     display.display();
73     delay(250);
74 }
75
76
77 atrasoChegada();
78
79
80 int i=0;
81 int Tcron = 0; //variavel do stopwatch
82 TIMEDATA = String(atraso, DEC);
83 k = 0;
84 TS = "";
85 do
86 { STTRCV = 0;
87 STTRCV = receiveTS(TIMEDATA);
88 Serial.println("STT_"+String(STTRCV) + " //_ " + "i_"+String(i)+ " //_ " + "TS_"
      "+String(TS)+ " //_ " + "tamanho_da_String_atraso:_" +String(k));
89 displ.showNumberDecEx(0000, (0x80 >> 1), true);
90 delay(50);
91 i++;
92 } while (STTRCV!=1 and i<4000);
93
94
95 Tcron = millis();
96
97
98 //loop do sensor de chegada
99 do
100 {
101     Serial.println("aguardando_sensor //_ Tcron=_") + String(Tcron) + "//

```

```

        _ _Distancia _=_ " + String(distancia)+" _cm");
102   display .clear ();
103   display .drawString(0, 0,"aguardadndo _sensor");
104   hcsr04 ();
105   display .display ();
106
107   delay(10);
108
109
110   i++;
111 } while(distancia >= 100 and i<10000);
112 // loop da chegada fecha
113
114 TC = millis ();
115 numCounter = TS.toInt ();
116
117
118
119   display .clear ();
120   display .drawString(0, 0,"TS=_ _" +String(TS));
121   display .drawString(0, 10,"TC=_ _" +String(TC));
122   display .drawString(0, 20,"atraso=_ _" +String(atraso));
123   display .display ();
124
125   Serial .println ("atraso=_ _" +String(atraso) + "// _" + "numCounter=_ _"+
        String(numCounter) + "// _" + "TC_" +String(TC) );
126
127
128 delay(3000);
129
130
131
132 Ttot = TC - (Tcron + (numCounter/2));
133 TotDsp = 10;
134

```

```

135
136     Serial.println("Ttot_" +String(Ttot)+ "//_TC=_" + String(TC)+ "//
        _Tcron=_" + String(Tcron)+"//_numCounter/2_" +String(
            numCounter/2));
137
138     display.clear();
139     display.drawString(0, 0, "Ttot:" + String(Ttot) + "msec");
140     display.drawString(0, 10, "TC:" + String(TC) + "msec");
141     display.drawString(0, 20, "Tcron:" + String(Tcron) + "msec");
142     display.drawString(0, 30, "numCounter/2:" + String(numCounter/2) + "
        msec");
143     display.display();
144     displ.showNumberDecEx(Ttot/TotDsp, (0x80 >> 1), true);
145     Serial.println("Ttot_" +String(Ttot)+ "//_TC=_" + String(TC)+ "//
        _Tcron=_" + String(Tcron)+"//_numCounter/2_" +String(
            numCounter/2));
146     delay(5000);
147
148
149 } //if RCV = 1
150
151
152 else{
153
154     display.clear();
155     display.drawString(0, 0, "StandBy");
156     display.display();
157     STTRCV = receive(SYNCDATA);
158     displ.showNumberDecEx(Ttot/TotDsp, (0x80 >> 1), true);
159     delay(5000);
160
161
162 } //else
163
164

```

```
165
166
167
168 } //void loop
169
170 #endif
```

A.3 Largada

```
1
2 #ifndef LARGADA
3 #define INTERVAL 250
4
5 int counter = 0;
6 int count10 = 0;
7 int count20 = 0;
8 int countGO = 0;
9 int GO = 0;
10 int TWIND = 0;
11
12 const int buttonPinSYNC = 34; // the number of the pushbutton pin
13 const int ledGreen = 17;
14 const int buzz = 22;
15 int ledState = 0;
16 int buzState = 0;
17
18 void setup() {
19   Serial.begin(115200);
20   //Chama a configuracao inicial do display
21   setupDisplay();
22   //Chama a configuracao inicial do LoRa
23   setupLoRa();
24   display.clear();
25   display.drawString(0, 0, "TCC");
26   display.display();
27
```

```

28
29 // initialize the pushbutton pin as an input:
30 pinMode(buttonPinSYNC, INPUT);
31   pinMode(ledGreen, OUTPUT);
32   pinMode(buzz, OUTPUT);
33 }
34
35
36
37
38 void loop() {
39
40   //loop do Sync ABRE
41   // read the state of the pushbutton value:
42   buttonState = digitalRead(buttonPinSYNC);
43
44   // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
45   if (buttonState == HIGH) {
46     //variavel TimerS recebe RTC
47     //LED Green BLINK
48     display.clear();
49     display.drawString(0, 0, "Botao_ON");
50     display.display();
51     delay(250);
52
53     //Loop do SEND pedido de sincronizacao abre
54     for(int i=0;i<3;i++){
55       display.clear();
56       display.drawString(0, 0, "Botao_ON");
57       display.display();
58       delay(250);
59     send(SYNCDATA);
60     delay(250);
61   }
62   STTRCV = 0;

```

```

63 Serial.println(STTRCV);
64 int i =0;
65 do
66 { STTRCV = 0;
67 digitalWrite(ledGreen, HIGH);
68 STTRCV = receive(SYNCDATAZ);
69 Serial.println(STTRCV);
70 Serial.println(i);
71 i++;
72 delay(250);
73 digitalWrite(ledGreen, LOW);
74 delay(250);
75 } while (STTRCV!=1 and i<10);
76
77 Serial.println(STTRCV);
78 if (STTRCV == 1){
79
80 //verifica se o pedido foi respondido
81 //Loop do SEND pedido de sincronizacao fecha
82 //LED Green HIGH
83 digitalWrite(ledGreen, HIGH);
84 //contador de "10s" comeca
85
86
87
88 for (count10=0;count10 < 10;count10++){
89   display.clear();
90   display.drawString(0,0, "contador_10s");
91   display.drawString(0,10, String(count10) + "_segundos");
92   display.drawString(0,20, "atraso:_" + String(atraso)+"_ms");
93   display.display();
94   delay(1000);
95 }
96 atraso = 1038;
97   atrasoLargada();

```

```
98  delay(250);
99
100
101  int count10 = 0;
102  //buzzer HIGH "on your marks"
103  digitalWrite(buzz, HIGH);
104  //delay 1000
105  delay(1000);
106  //buzzer LOW
107  digitalWrite(buzz, LOW);
108  //contador de "10s" termina
109
110  //contador de "20s" comeca
111  for (count20=0;count20 < 20;count20++){
112    display.clear();
113    display.drawString(0,0, "contador_20s");
114    display.drawString(0,10, String(count20) + "_segundos");
115    display.drawString(0,20, "As_suas_marcas");
116    display.display();
117    delay(250);
118  }
119
120  int count20 = 0;
121  //buzzer HIGH "SET"
122  digitalWrite(buzz, HIGH);
123  //delay 1000
124  delay(300);
125  //buzzer LOW
126  digitalWrite(buzz, LOW);
127  //contador de "20s" termina
128  //delay 20000
129  //GO = random(1,14)
130  GO = 0;
131  countGO=0;
132  GO = random(3,13);
```

```

133 for (countGO=0;countGO < GO;countGO++){
134     display.clear();
135     display.drawString(0,0, "contador_GO_" + String(GO) + "_sorteado");
136     display.drawString(0,10, String(countGO) + "_segundos");
137     display.drawString(0,20, "aleatorio");
138     display.display();
139     delay(100);
140 }
141 display.clear();
142 display.drawString(0,0, "disparou_GO_" );
143
144 display.display();
145
146 //buzzer HIGH "GO"
147 digitalWrite(buzz, HIGH);
148 TIMEDATA = String(atraso, DEC);
149 //delay 1000
150 delay(400);
151 digitalWrite(buzz, LOW);
152
153
154
155
156 //Loop do SEND tempod e largada abre
157 for(int i=0;i<8;i++){
158     send(TIMEDATA);
159     display.clear();
160     display.drawString(0,0, "disparando_largada_");
161     display.drawString(0,10, "atraso_de:" +String(atraso)+"_ms");
162     display.display();
163     delay(500);
164 }
165
166 //send TIMEDATA fecha
167

```

```
168 }else{
169     display.clear();
170     display.drawString(0,10, "Nao_conseguiu_Sincronizar_");
171     display.display();
172     delay(2000);
173
174 }
175
176 } else {
177     // turn LED off:
178     digitalWrite(ledGreen, LOW);
179     display.clear();
180     display.drawString(0, 0, "Pronto_para_Sincronizar_");
181     display.display();
182     delay(250);
183 }
184
185 }
186
187 //loop do Sync FECHA
188
189
190
191 #endif
```

BIBLIOGRAFIA

ALLIANCE, L. *LoRaWAN*. 2018. <https://lora-alliance.org/about-lorawan>. (Accessed on 11/16/2018).

BERNARD, S. *Digital communications fundamentals and applications*. Prentice Hall, USA, 2001.

ESPRESSIF. 2017. <https://www.espressif.com>. (Accessed on 05/10/2018).

FINKENZELLER, K. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. [S.l.]: John Wiley & Sons, 2010.

FRIIS, H. T. A note on a simple transmission formula. *Proceedings of the IRE*, v. 34, n. 5, p. 254–256, May 1946.

FTC. *Internet of Things: Privacy & Security in a Connected World*. January 2015. (Accessed on 08/23/2018).

GINCIENE, G.; MATTHIESEN, S. Q. O sistema de partida em corridas de velocidade do atletismo. 2012. (Accessed on 04/29/2018).

HELTEC. *HelTec*. 2018. <http://www.heltec.cn/project/wifi-lora-32/?lang=en>. (Accessed on 11/26/2018).

IAAF TRACK AND FIELD FACILITIES MANUAL. [S.l.: s.n.], 2008.

ILLUSTRATIVE MATHEMATICS. 2018. <https://www.illustrativemathematics.org>. (Accessed on 05/10/2018).

JUNIOR, P. A natural history of athleticism and cardiovascular health. *Jama*, v. 252, n. 4, p. 491–495, 1984.

MACKALA, K. Optimisation of performance through kinematic analysis of the different phases of the 100 metres. 2007. (Accessed on 04/29/2018).

MICROCHIP. *Lora Technology*. 2018. <https://www.microchip.com/design-centers/wireless-connectivity/low-power-wide-area-networks/lora-technology>. (Accessed on 11/16/2018).

MIORANDI, D. et al. Internet of things: Vision, applications and research challenges. *Ad hoc networks*, Elsevier, v. 10, n. 7, p. 1497–1516, 2012.

SHARMA, C.; NAKAMURA, Y. *Wireless Data Services Summary*. 2003. (Accessed on 08/29/2018).

SPARKFUN. *ESP32 LoRa*. 2018. <https://learn.sparkfun.com/tutorials/esp32-lora-1-ch-gateway-lorawan-and-the-things-network>. (Accessed on 11/16/2018).

SUNDMAEKER, H. et al. Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commission*, v. 3, n. 3, p. 34–36, 2010.

WILLIS, D. M. *An introduction to radiowave propagation*. 2007. http://www.mike-willis.com/tutorial/free_space.htm. (Accessed on 08/29/2018).

XIA, F. et al. Internet of things. *International Journal of Communication Systems*, v. 25, n. 9, p. 1101, 2012.