



**UNIVERSIDADE
FEDERAL DE JUIZ DE FORA**

**FACULDADE DE ENGENHARIA ELÉTRICA
SOFTWARE EMBARCADO**

**Enzo Gonçalves Pusiol
Humberto Barbosa Coutinho
Vinícius Franklin Pedroso Mansur de Azevedo**

Protótipo de um Relógio Inteligente

Juiz de Fora
2025

SUMÁRIO

1	Introdução	2
2	Sensores Utilizados	3
2.1	Módulo GPS	3
2.1.1	Obtenção do Delta de Altitude	3
2.1.2	Cálculo da Diferença de Latitude e Longitude	3
2.1.3	Cálculo da Distância Utilizando a Fórmula de Haversine	4
2.1.4	Velocidade Média	4
2.1.5	Velocidade Instantânea	5
2.1.6	Distância Total Percorrida	5
2.1.7	Cálculo de Calorias Gastas	5
2.1.8	Incertezas do Módulo GPS e Limitação de Erros	6
2.1.9	Conclusão	6
2.2	Acelerômetro ADXL345	7
2.2.1	Características Técnicas	7
2.2.2	Aplicação em Smartwatches	7
2.2.3	Implementação no Código	7
2.3	Display OLED	8
2.3.1	Características Técnicas	8
2.3.2	Aplicação no Smartwatch	8
2.3.3	Implementação no Código	9
2.4	DHT11	9
2.4.1	Implementação no Código	9
3	Conclusão	11
	REFERÊNCIAS	12

1 Introdução

Nos últimos anos, o apelo para o uso de dispositivos remotos como *smartwatches* tem se popularizado. Este fator é tanto impulsionado pelo avanço nas tecnologias embarcadas - que atuam através de sensores mais precisos e de baixo consumo, juntamente com a possibilidade de conectividade com outros dispositivos, tornando-o uma ferramenta essencial para qualquer público -, quanto na necessidade de monitorar a saúde e o desempenho físico.

Neste contexto, esses equipamentos geralmente são estruturados para fornecer informações como contagem de passos, medição de distâncias, calorias despendidas, oscilações de altitude e características como temperatura e umidade.

Portanto, diante desse cenário, este trabalho apresenta o desenvolvimento de um protótipo semelhante a um smartwatch, capaz de medir distância, passos, velocidade, calorias gastas, altitude, temperatura e umidade. O projeto explora a aplicação de software embarcado para garantir a coleta, processamento e apresentação dessas informações de forma eficiente e acessível ao usuário.

2 Sensores Utilizados

2.1 Módulo GPS

O módulo GPS utilizado no projeto tem como função principal fornecer dados de localização geográfica para o cálculo de **distância percorrida, velocidade, altitude, tempo e calorias gastas pelo usuário**. A comunicação com o microcontrolador ocorre via **UART**, e os dados recebidos são processados para extrair informações como **latitude, longitude e horário**.

Este relatório detalha a implementação do código do GPS, explicando como são feitas a obtenção do **delta de altitude**, a **diferença de latitude e longitude**, o **cálculo da distância percorrida utilizando a Fórmula de Haversine** e a **estimativa de calorias gastas**. Além disso, detalharemos os cálculos de velocidade média e instantânea.

2.1.1 Obtenção do Delta de Altitude

A variação de altitude é um parâmetro importante para determinar o esforço físico do usuário. Subidas e descidas influenciam diretamente no consumo calórico. No código, a altitude é extraída das leituras do GPS e armazenada para comparação com a leitura anterior.

A obtenção do **delta de altitude** ocorre da seguinte forma:

1. O código mantém uma variável que armazena a última altitude registrada.
2. Quando uma nova leitura do GPS é recebida, a altitude atual é comparada com a anterior.
3. Se houver uma diferença significativa, esse valor é armazenado e utilizado para análise posterior.

Essa abordagem permite estimar com maior precisão o impacto do terreno no desempenho do usuário.

2.1.2 Cálculo da Diferença de Latitude e Longitude

Os valores de **latitude** e **longitude** são extraídos das sentenças NMEA enviadas pelo GPS. Essas sentenças contêm informações de localização no formato decimal e precisam ser processadas para cálculos de distância.

No código, os valores de latitude e longitude são armazenados em variáveis para cálculos posteriores. A função `toRadians()` converte os valores de graus para radianos, preparando-os para os cálculos trigonométricos necessários.

A diferença entre as coordenadas dos dois pontos será fundamental para determinar a distância percorrida, o que será mais detalhado na próxima seção.

2.1.3 Cálculo da Distância Utilizando a Fórmula de Haversine

A **Fórmula de Haversine** é usada para calcular a distância entre dois pontos na superfície da Terra considerando a curvatura do planeta.

O cálculo da distância utiliza a Fórmula de Haversine, que envolve a conversão das coordenadas de latitude e longitude para radianos, o cálculo das diferenças angulares e a aplicação de funções trigonométricas para determinar a distância entre os pontos. O resultado é multiplicado pelo raio médio da Terra para obter a distância em quilômetros.

A Fórmula de Haversine é descrita por:

$$d = 2R \cdot \arctan \left(\sqrt{a}, \sqrt{1 - a} \right)$$

onde:

$$a = \sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\Delta\lambda}{2} \right)$$

e R é o raio médio da Terra (6371 km).

Explicação da Fórmula

1. Conversão para radianos: As coordenadas de latitude e longitude são convertidas para radianos.
2. Cálculo da diferença angular: São determinadas as variações dLat e dLon.
3. Cálculo do fator haversine: A equação utiliza a função $\sin()$ para calcular um fator que mede a separação entre os pontos.
4. Aplicação da fórmula: O resultado é multiplicado pelo raio médio da Terra (6371 km) para obter a distância real em quilômetros.
5. Esse método é amplamente utilizado por sistemas de navegação devido à sua precisão e baixo custo computacional.

2.1.4 Velocidade Média

A velocidade média é calculada ao final do percurso, dividindo a distância total percorrida pelo tempo total gasto. O cálculo é realizado utilizando o tempo inicial e final do percurso, que são armazenados no início e no final da corrida, respectivamente.

2.1.5 Velocidade Instantânea

A velocidade instantânea é calculada a cada nova leitura de GPS, dividindo a distância entre os dois pontos mais recentes pelo tempo decorrido entre essas leituras. Este cálculo é realizado utilizando as coordenadas de latitude e longitude e os tempos de leitura dos pontos mais recentes.

2.1.6 Distância Total Percorrida

A distância total percorrida é acumulada a cada nova leitura de GPS, somando a distância calculada entre os pontos mais recentes. Esta soma é realizada utilizando a Fórmula de Haversine para calcular a distância entre os pontos e acumulando o resultado em uma variável que armazena a distância total percorrida.

2.1.7 Cálculo de Calorias Gastas

A estimativa de calorias queimadas leva em consideração **a distância percorrida, a velocidade média e o peso do usuário**. Baseado no site:

[https : //sites.google.com/site/compendiumofphysicalactivities/](https://sites.google.com/site/compendiumofphysicalactivities/)

[Activity – Categories/running](#)

E a equação do MET foi baseada em:

[https : //nutritotal.com.br/pro/material/met – multiplos – de – equivalentes – metabolicos/](https://nutritotal.com.br/pro/material/met-multiplos-de-equivalentes-metabolicos/)

O cálculo no código é baseado na determinação do MET (Metabolic Equivalent of Task) com base na velocidade do usuário e na aplicação de uma fórmula que leva em consideração o MET, o peso do usuário e o tempo de exercício.

Explicação do Cálculo

1. Determinação do MET (Metabolic Equivalent of Task): O MET é um fator que representa o gasto calórico com base no tipo de exercício.
2. Fórmula da caloria: Multiplica-se o MET pelo peso do usuário e pelo tempo de exercício em horas.
3. Ajuste energético: O fator 1.05 representa uma média do gasto extra devido a variações de metabolismo.

Essa abordagem permite estimar com boa precisão as calorias gastas durante a atividade física.

2.1.8 Incertezas do Módulo GPS e Limitação de Erros

É importante notar que o módulo GPS apresenta algumas incertezas, especialmente quando o usuário está em repouso. Pequenas variações nas leituras do GPS podem resultar em flutuações tanto na distância quanto na altitude, gerando valores falsos.

Para mitigar esses problemas, foram estabelecidos **limiares (thresholds)** no código. Esses limiares ajudam a filtrar variações insignificantes, garantindo que apenas mudanças significativas sejam consideradas nos cálculos de distância e altitude. Especificamente:

- **Variação de Altitude:** Pequenas variações na altitude, inferiores a 0.3 metros, são ignoradas. Isso evita que flutuações irrelevantes afetem o cálculo do delta de altitude.
- **Variação de Distância:** Distâncias inferiores a 0.003 quilômetros (3 metros) entre duas leituras consecutivas também são desconsideradas. Isso impede que pequenas variações na posição do GPS, quando o usuário está parado, sejam interpretadas como movimento real.

Esses limiares foram definidos após testes e calibrações para garantir que o sistema forneça dados precisos e confiáveis, mesmo em condições onde o GPS possa apresentar pequenas variações.

Com essas calibrações, o sistema se torna mais robusto e preciso, fornecendo dados mais confiáveis para os cálculos de distância, altitude e calorias gastas.

2.1.9 Conclusão

A implementação do módulo GPS no sistema embarcado permite monitorar a localização do usuário em tempo real e calcular métricas importantes como **distância percorrida, velocidade e calorias queimadas**.

A obtenção do **delta de altitude** fornece informações sobre a variação do terreno, influenciando diretamente o cálculo do esforço físico. O cálculo da diferença de **latitude e longitude**, combinado com a **Fórmula de Haversine**, garante medições precisas da distância percorrida.

A estimativa de calorias permite que o usuário acompanhe seu desempenho e ajuste sua rotina de atividades. Futuramente, melhorias podem ser feitas na **calibração dos sensores** e na **otimização do consumo de energia**, aumentando a autonomia do dispositivo.

2.2 Acelerômetro ADXL345

O ADXL345 é um acelerômetro digital de baixo consumo de energia, projetado para medir a aceleração em três eixos (X, Y e Z) com alta precisão. Ele é utilizado em dispositivos remotos, como smartwatches e rastreadores de atividades, devido à sua capacidade de detectar movimentos, inclinação e quedas.

2.2.1 Características Técnicas

O ADXL345 opera em uma faixa de tensão de 2,0 V a 3,6 V e pode ser conectado via I2C ou SPI, permitindo integração flexível com microcontroladores e sistemas embarcados. Suas principais características incluem:

- Faixa de medição ajustável: $\pm 2g$, $\pm 4g$, $\pm 8g$ ou $\pm 16g$;
- Alta resolução: Sensibilidade de até 4 mg/LSB em modo de alta resolução ($\pm 2g$);
- Taxa de amostragem configurável;
- Sensores de queda livre e detecção de movimento;
- Baixo consumo de energia: Apenas 40 μA em modo de operação normal e 0,1 μA no modo de espera.

2.2.2 Aplicação em Smartwatches

No contexto do dispositivo proposto, o ADXL345 é utilizado para contagem de passos: O acelerômetro detecta oscilações características da caminhada e corrida para calcular o número de passos dados pelo usuário. A integração do ADXL345 com o software embarcado do dispositivo requer a implementação de algoritmos de manipulação das leituras para identificar variações e eliminar ruídos indesejados. O uso de média móvel ajuda a suavizar as variações abruptas nos dados coletados.

2.2.3 Implementação no Código

O sensor é inicializado em uma função, onde o registrador de controle de energia é configurado para ativar o dispositivo. A leitura dos valores dos eixos X, Y e Z ocorre por meio da função `adxl345_read()`, que acessa o registrador e recupera 6 bytes de dados, correspondentes aos três eixos. A função `task_adxl345()` é responsável por processar os dados brutos do sensor. Ela calcula a magnitude da aceleração usando a equação:

$$\text{Magnitude} = \sqrt{x^2 + y^2 + z^2} \quad (2.1)$$

Além disso, é aplicado um filtro de média móvel para suavizar os valores de aceleração e reduzir ruídos. Por fim, compara-se a magnitude com um valor limite de detecção de passos ($STEP_THRESHOLD = 280$) e incrementa o contador de passos.

2.3 Display OLED

O Display OLED (Organic Light-Emitting Diode) é uma tecnologia muito utilizada para telas de dispositivos, sendo amplamente utilizado em celulares e smartwatches devido à sua eficiência energética, alto contraste e excelente visibilidade em diferentes condições de iluminação. Diferente das telas LCD tradicionais, os displays OLED não necessitam de um painel de retroiluminação, pois cada pixel emite sua própria luz, resultando em imagens mais nítidas.

2.3.1 Características Técnicas

Os displays OLED utilizados em sistemas embarcados geralmente possuem as seguintes características:

- Baixo consumo de energia: Como os pixels pretos não consomem energia, o consumo é reduzido, tornando o OLED ideal para dispositivos portáteis;
- Interfaces de comunicação versáteis: Pode operar via I2C ou SPI, permitindo fácil integração com microcontroladores e processadores embarcados;
- Resolução e tamanhos variados

2.3.2 Aplicação no Smartwatch

No dispositivo em desenvolvimento, o Display OLED desempenha um papel fundamental na exibição das informações coletadas pelos sensores embarcados. Suas principais aplicações incluem:

- Apresentação de dados em tempo real: Exibição de passos, distância percorrida, velocidade, calorias queimadas, temperatura, umidade e altitude;
- Interface interativa: Telas iterativas para o usuário;

A implementação do Display OLED no software embarcado do smartwatch requer a utilização de bibliotecas específicas para controle gráfico.

2.3.3 Implementação no Código

A inicialização do display ocorre na função `ssd1306_init()`, que configura o modo de carregamento, a orientação da tela e a ativação do display. A função `ssd1306_clear_display()` é usada para limpar o conteúdo exibido e a exibição de texto ocorre em `ssd1306_display_text()`, que:

- Percorre a string a ser exibida e escreve os caracteres um a um usando como dicionário a fonte `font8x8_basic`.
- Suporta múltiplas linhas e avança para a próxima linha ao encontrar um caractere de nova linha (`\n`).

A função `task_ssd1306_display_text()` controla a interface do usuário, permitindo alternar entre diferentes telas pressionando um botão (`BT_IO`):

- Tela 0: Exibe a contagem de passos coletada do ADXL345.
- Tela 1: Exibe a temperatura e a umidade coletadas do sensor DHT11.

Essa interface garante que o usuário possa visualizar informações relevantes de forma dinâmica.

2.4 DHT11

O sensor DHT11 é um dispositivo amplamente utilizado para medir temperatura e umidade relativa do ar em sistemas embarcados. Ele se comunica com o microcontrolador através de um único pino digital, enviando leituras discretas a cada segundo. No contexto deste projeto, ele é utilizado para coletar e exibir informações do ambiente no Display OLED SSD1306. A seguir, será detalhada a implementação do DHT11, incluindo a inicialização, a leitura dos dados e a exibição das informações coletadas.

2.4.1 Implementação no Código

A comunicação com o DHT11 ocorre através da função `vTaskGet()`, que executa um loop contínuo para capturar e processar os dados do sensor. Primeiramente, a função inicia um log de atividade para indicar que a medição está em andamento. Em seguida, os valores de temperatura e umidade são lidos por meio da função `DHT11_read()`, que retorna uma estrutura contendo as leituras coletadas. Esses valores são armazenados nas variáveis globais `temp` e `umi`, permitindo que outras partes do programa acessem e utilizem as informações. Após cada leitura, os dados são enviados ao log do ESP32, garantindo o monitoramento contínuo da temperatura e umidade em tempo real.

O código prevê a exibição das leituras de temperatura na Tela 1 do display OLED, permitindo alternar entre diferentes dados ao pressionar um botão. A coleta periódica de dados do DHT11 possibilita o monitoramento do ambiente, sendo uma ferramenta útil para atividades físicas que exigem controle térmico ou análise de umidade.

3 Conclusão

O desenvolvimento do protótipo de smartwatch demonstrou a viabilidade da aplicação de sistemas embarcados para monitoramento de atividades físicas. A integração dos sensores, como o GPS, acelerômetro ADXL345, display OLED e sensor DHT11, permitiu a obtenção de dados sobre distância percorrida, velocidade, calorias gastas, número de passos, altitude, temperatura e umidade.

A implementação dos algoritmos, incluindo a Fórmula de Haversine para cálculo de distância e a estimativa de gasto calórico, garantiu medições essenciais para a usabilidade do dispositivo. Além disso, o uso do FreeRTOS possibilitou a execução das tasks responsáveis pela captura e exibição das informações.

Apesar dos resultados positivos, algumas melhorias podem ser exploradas no futuro, como a otimização do consumo energético, a calibração mais precisa dos sensores e a implementação de novas funcionalidades, como monitoramento de frequência cardíaca. Dessa forma, o projeto pode evoluir para um dispositivo ainda mais útil para o acompanhamento da saúde e do desempenho físico dos usuários.

REFERÊNCIAS

- 1 AOSONG ELECTRONICS. DHT11 Humidity Temperature Sensor Datasheet. Disponível em: <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>. Acesso em: 16 mar. 2025.
- 2 ANALOG DEVICES. ADXL345 Datasheet: 3-Axis, ± 2 g/ ± 4 g/ ± 8 g/ ± 16 g Digital Accelerometer. Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf>. Acesso em: 17 mar. 2025.
- 3 FREE RTOS. FreeRTOS – Real-time operating system for microcontrollers microprocessors. Disponível em: <https://www.freertos.org/>. Acesso em: 17 mar. 2025.
- 4 YANBE. SSD1306 ESP-IDF I2C Driver. Disponível em: <https://github.com/yanbe/ssd1306-esp-idf-i2c>. Acesso em: 17 mar. 2025.