



UNIVERSIDADE ESTADUAL DE CAMPINAS

---

## ES070 - Projeto de Sistemas Embarcados

### Robô Seguidor de Linha

---

<i>Nomes</i>	<i>RA</i>
Gabriel Erick Matheus Imamura	197301
Guilherme Augusto Vieira Palma	173610
Henrique Meneguetti Bianchi	217802
João Vitor Gomes Do Nascimento De Siqueira	237870
Paulo Roberto de Araújo Júnior	204440
Suellen Ribeiro	187215

## Sumário

<b>1 Resumo</b>	<b>2</b>
<b>2 Documentação do Sistema</b>	<b>3</b>
<b>3 Display LCD</b>	<b>21</b>
3.1 Características Gerais . . . . .	21
3.1.1 Comunicação I2C . . . . .	21
3.1.2 Modos de Operação . . . . .	21
3.1.3 Funcionalidades . . . . .	21
3.1.4 Funções principais . . . . .	21
<b>4 Bluetooth</b>	<b>22</b>
4.1 Configuração do Módulo Bluetooth HC-05 . . . . .	22
4.1.1 Passos para Configuração . . . . .	22
4.2 Aplicativo de Controle . . . . .	22
4.2.1 Modos de Controle . . . . .	22
4.3 Conexão Inicial com o Smartphone . . . . .	23
<b>5 Encoders</b>	<b>24</b>
<b>6 Sensor Ultrassônico</b>	<b>25</b>
6.1 Características Gerais . . . . .	25
6.1.1 Princípio de Funcionamento . . . . .	25
6.1.2 Especificações . . . . .	25
6.1.3 Conexão com o STM32 . . . . .	25
6.2 Funcionalidades . . . . .	25
6.3 Funções principais . . . . .	26
<b>7 Manual de Utilização</b>	<b>27</b>
<b>8 Problemas Identificados e não Resolvidos</b>	<b>28</b>
<b>9 Autoria dos Códigos Fornecidos</b>	<b>29</b>

## 1 Resumo

O desenvolvimento de robôs seguidores de linha tem se tornado uma prática comum em projetos de automação e robótica, proporcionando uma excelente plataforma para o estudo de sistemas embarcados, controle de motores e integração de sensores. Este projeto propõe o desenvolvimento de um robô seguidor de linha utilizando o kit de desenvolvimento NUCLEO-G474RE, em conjunto com uma placa dedicada e sensores, visando criar um sistema autônomo de movimentação e controle.

O robô foi projetado para seguir trajetórias delineadas por faixas, sendo equipado com 5 sensores de faixa que permitem detectar a posição relativa da linha a ser seguida. Além disso, ele possui um sensor ultrassônico, que garante a detecção de obstáculos frontais, evitando colisões de maneira eficiente. O sistema de controle do robô é baseado em um firmware desenvolvido na IDE STM32CubeIDE, fornecida pela STMicroelectronics, com uma configuração detalhada dos periféricos essenciais para o funcionamento do robô.

As funcionalidades do robô podem ser controladas por meio de um aplicativo para Android, que permite ao usuário ajustar a velocidade dos motores, controlados via PWM, e gerenciar os movimentos do robô. A comunicação entre o robô e o aplicativo é realizada via Bluetooth, utilizando a interface USART configurada no firmware. Além disso, um display LCD controlado via I2C oferece feedback visual em tempo real sobre o estado do robô e suas funções.

## 2 Documentação do Sistema

O objetivo principal deste projeto é desenvolver um robô autônomo seguidor de linha que possa navegar de forma precisa e eficiente em um circuito pré-definido, sendo controlado e monitorado por um aplicativo móvel. Além de seguir trajetórias, o robô deverá ser capaz de lidar com desafios comuns de navegação, como cruzamentos e obstáculos, respondendo de maneira autônoma às mudanças do ambiente. O robô oferecerá uma interface de controle e monitoramento acessível e intuitiva, compatível com dispositivos móveis, permitindo ao usuário configurar parâmetros e acompanhar o desempenho do sistema em tempo real.

### Requisitos Funcionais

1. **Detecção de Desvio:** O robô deve identificar quando se desvia da linha e corrigir automaticamente sua trajetória para manter-se dentro dos limites.
2. **Cruzamentos:** Deve ser capaz de navegar por cruzamentos e continuar seguindo a linha corretamente após esses pontos.
3. **Velocidade de Operação:** O robô permitirá configurar sua velocidade de operação por meio do aplicativo móvel, oferecendo diferentes níveis de desempenho.
4. **Parada em Caso de Obstáculo:** Ao detectar um obstáculo frontal, o robô deverá parar e emitir um alarme sonoro.
5. **Parada Emergencial em Caso de Colisão:** Em caso de colisão frontal, os motores devem ser interrompidos imediatamente como medida de segurança.
6. **Parada no Fim do Percurso:** O robô deve parar automaticamente ao chegar ao final do percurso predefinido.
7. **Monitoramento:** O sistema deve monitorar a distância percorrida, a velocidade atual e média, bem como a postura, exibindo coordenadas  $x$ ,  $y$  e orientação  $\theta$ .
8. **Exibição no Display LCD:** O display LCD deve mostrar o nível de carga da bateria em %, a autonomia estimada em metros, a distância percorrida e a postura atual do robô.
9. **Controle de Movimento via Celular:** A interface do celular permitirá ao usuário controlar o movimento do robô através de comunicação Bluetooth.
10. **Configuração de Parâmetros via Interface:** O usuário poderá ajustar parâmetros como intensidade do alarme e velocidade diretamente na interface do aplicativo.
11. **Monitoramento dos Parâmetros pelo Celular:** Todos os parâmetros monitorados pelo robô estarão disponíveis para visualização no aplicativo móvel.
12. **Indicação de Status com LEDs:** O robô utilizará LEDs para indicar seu status atual, como em movimento, parado, em espera ou em recarga.

13. **Compatibilidade com Dispositivos Móveis:** A interface de controle e monitoramento será compatível com dispositivos Android e iOS.
14. **Indicação de Nova Configuração:** Toda vez que uma nova configuração for concluída, o buzzer e a luz verde do robô deverão acionar simultaneamente para confirmar a alteração.

### Requisitos Não Funcionais

1. **Suavidade de Movimento:** O robô deverá movimentar-se suavemente ao longo da linha, minimizando solavancos para evitar desgaste mecânico e aumentar a precisão.
2. **Taxa de Atualização das Grandezas Monitoradas:** A taxa de atualização dos parâmetros monitorados (distância, velocidade, postura) deve ser de 1 Hz para garantir um monitoramento em tempo real.
3. **Interface de Usuário Intuitiva no Celular:** A interface de controle e monitoramento no aplicativo móvel deve ser projetada para ser fácil de usar e intuitiva, facilitando o uso pelo usuário final.
4. **Responsividade:** O robô deve responder a comandos de movimento recebidos via Bluetooth com uma latência máxima de 100 ms, proporcionando controle ágil e preciso.

### Diagrama de Blocos

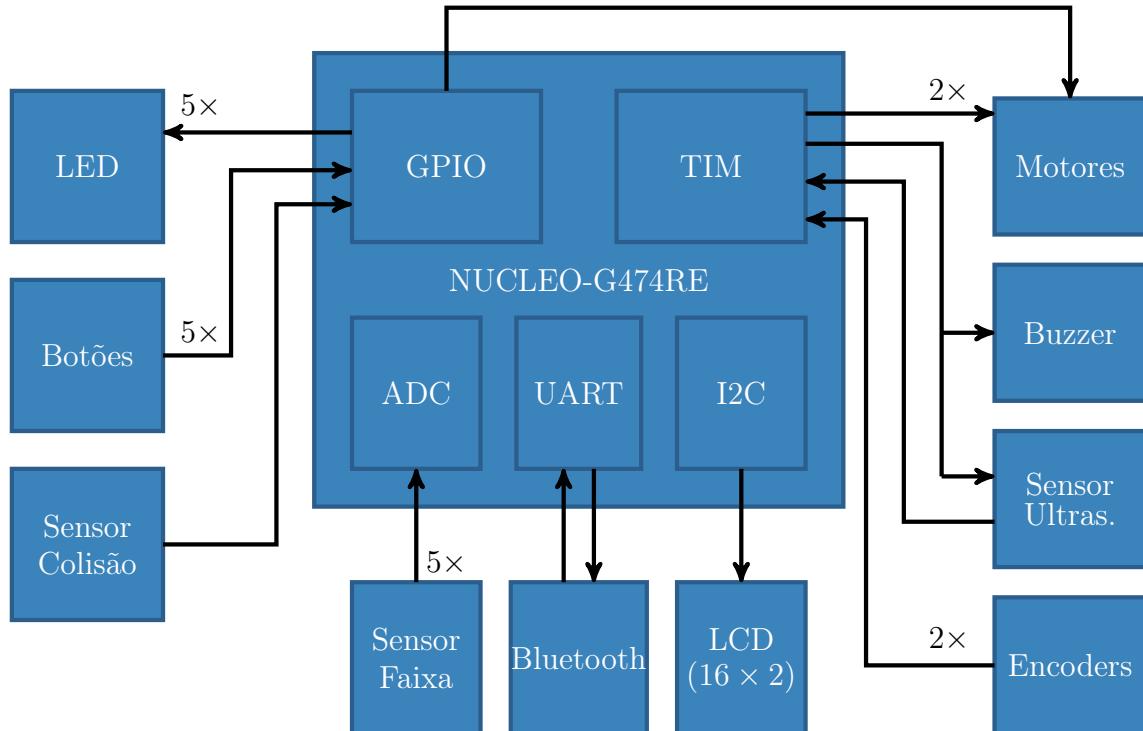


Figura 1: Diagrama de blocos do sistema.

## Componentes de Hardware

- NUCLEO-G474RE
  - Processador ARM Cortex-M4F
  - Alimentação via micro USB (5 V)
  - Gravador e Debugger integrado (ST-LINK V3E)
  - Comunicação UART via COM Virtual
  - Botão e LED programáveis
  - Fontes de clock integradas (24 MHz e 32 KHz)
  - Barramento de conexão compatível com Arduino UNO V3
  - Barramento de conexão compatível com ST Morpho
  - IDE: STM32CubeIDE, IAR, Keil, CoIDE, MULTI, CrossWorks, TrueSTUDIO etc.
  - Custo: R\$ 165,02

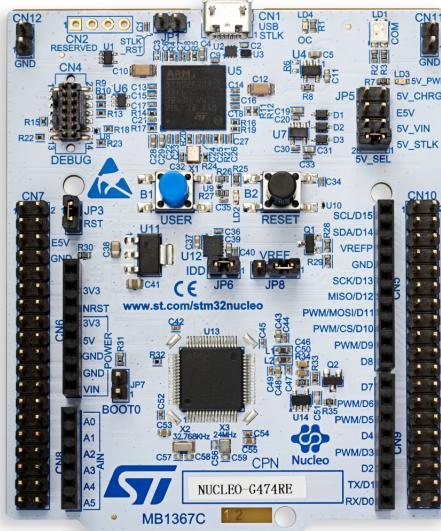


Figura 2: NUCLEO-G474RE.

- Microcontrolador target STM32G474RE
  - Núcleo: ARM Cortex M4
  - Tamanho da memória do programa: 128 kB
  - Largura do barramento de dados: 32 bit
  - Resolução ADC:  $5 \times 12$  bit
  - Frequência de operação máxima: 170 MHz

- Número de I/Os: 52 I/O
- Tamanho RAM dos dados: 128 kB
- Tensão de alimentação - Mín: 1.71 V
- Tensão de alimentação - Máx: 3.6 V
- Temperatura operacional mínima:  $-40^{\circ}\text{C}$
- Temperatura operacional máxima:  $+85^{\circ}\text{C}$
- Tensão de alimentação analógica: 1.62 V a 3.6 V
- Resolução DAC: 12 bit
- Tipo de dados Ram: SRAM
- Voltagem I/O: 5 V
- Tipo de interface: 12C, SPI, UART, USART, USB
- Tipo de memória do programa: Flash 960
- Custo: R\$ 42,56



Figura 3: Target STM32G474RE.

- Microcontrolador debugger, gravação e comunicação COM (STLINK V3E)
  - Fonte de tensão programável de 1,6 V a 3,6 V
  - Classificação de corrente de saída de 500 mA com proteção contra sobrecorrente (OCP) em 550 mA
  - Taxa de amostragem programável de 1 SPS a 100 kSPS
  - Medição dinâmica: corrente de 100 nA a 550 mA / medições de potência de 160 nW a 1,65 W / Largura de banda de 50 kHz / aquisição de 1,6 MHz / precisão de 2%
  - Fonte de tensão de saída auxiliar de 1,6 V a 3,6 V com até 2 A (sem medição de corrente, OCP em 2,5 A)
  - JTAG/Depuração por fio serial (SWD): Suporte de comunicação SWD (Depuração por fio serial) e SWV (Visualizador por fio serial) até 10 MHz. Suporte de comunicação JTAG até 20 MHz

- Interface UART na porta COM virtual (VCP) com frequência de até 12 MHz
- Ponte multipath USB para SPI/I2C/CAN/GPIOS
- Adaptador de nível integrado de voltagem I/O de 1,6 V a 3,6 V
- Quatro LEDs bicolores fornecendo o estado da sonda
- Três cabos planos STDC14 para MIPI10/STDC14 / MIPI20 com conectores de passo de 1,27 mm
- Quatro cabos (dois macho/macho e dois macho/fêmea)
- Conector USB Type-C:
  - Alimentado através de USB Type-C (5 V/3 A máximo)
  - Interface USB 2.0 de alta velocidade
  - Atualização do firmware da sonda por USB
- Suporte direto da ferramenta de software STM32CubeMonitor-Power, Keil, IAR.



Figura 4: Microcontrolador debugger.

- Conversor USB / Serial - UART - CP2102

- Baud Rate: 300 Kbps - 1 Mbps;
- Chip de controle: CP2102;
- Tensão de trabalho: 4 V 5.25 (Alimentado pela porta USB);
- Sistemas operacionais: Windows XP / 7 / Linux / Mac OS etc.
- Tensão de saída: 3.3 V e 5 V (Dois pinos);
- LED’s Indicadores: Alimentação (Power) e TXD (Transmissão);
- Corrente limite: 100 mA (Com proteção);
- Custo: R\$ 18,60

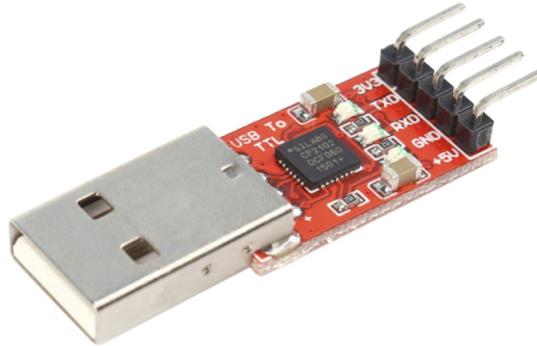


Figura 5: Conversor USB / Serial - UART - CP2102.

- LCD 16 × 2 alfanumérico
  - Tipo - display de caracteres
  - Tensão de alimentação - 5 V
  - Controlador HD44780
  - 4 ou 8 pinos de dados
  - Custo: R\$ 27,65

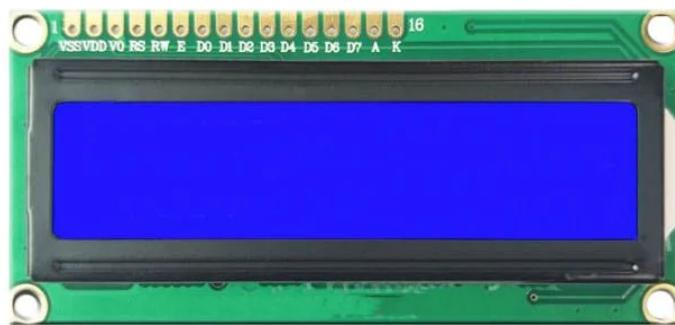


Figura 6: LCD 16 × 2 alfanumérico.

- Comunicação I2C
  - Endereço 12C: 0x20-0x27 (Padrão 0x20 mas pode ser modificado)
  - Compatível com Display LCD 16 × 2 e LCD 20 × 4
  - Tensão de operação: 5 V

- Dimensões:  $55 \times 23 \times 14$  mm
- Peso: 5 g
- Custo: R\$ 16,82

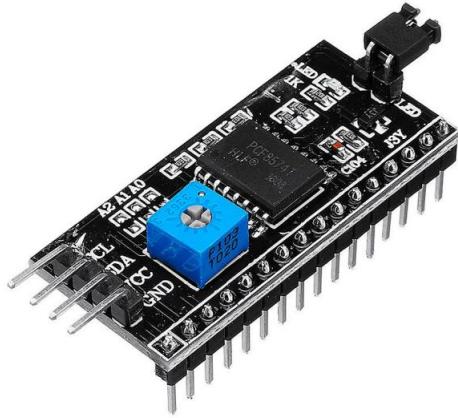


Figura 7: Comunicação I2C.

- Conversor DC/DC - Boost - XL6009
  - Chip Base: XL6009 (Datasheet XL6009);
  - Tensão de Entrada: 3 à 32 VDC;
  - Tensão de Saída: 5 à 35 VDC (Sempre maior que a entrada);
  - Corrente de entrada máxima: 4A;
  - Eficiencia de conversão: 88% (típica), 94% (máxima);
  - Frequencia de chaveamento: 400 KHz (Típica);
  - Temperatura de Trabalho: -40°C à +85°C;
  - Acrécimo de Temperatura em plena carga: 40°C;
  - Recomendável Utilizar dissipador para potencias acima de 25 W;
  - Dimensões: 43 x 21 x 14 (mm);
  - Peso: 10 g;
  - Custo R\$ 7,25



Figura 8: Conversor DC/DC - Boost - XL6009.

- Push button
  - Tensão máxima - 12 V
  - Corrente máxima - 0,5 A
  - Dimenções -  $6 \times 6 \times 6$  mm
  - Peso - 0,3 g
  - Custo R\$ 0,40



Figura 9: Push button.

- LED

- Corrente de trabalho: 20 mA
- Tensão: 1.9 V - 2.1 V
- Custo: R\$ 0,33



Figura 10: LED.

- Buzzer

- Tensão de operação: 4 a 8 V;
- Tensão recomendada: 5 V
- Corrente máxima: 40 mA
- Diâmetro: 12 mm
- Altura: 8,5 mm
- Peso: 1,1 g
- Custo: R\$ 2,50



Figura 11: Buzzer.

- Sensor de Distância Ultrassônico - HC-SR04
  - Tensão de trabalho: 5 VDC;
  - Corrente de repouso: inferior a 2 mA;
  - Corrente de trabalho: 15 mA;
  - Distância de detecção: 2 cm - 400 cm;
  - Resolução: 0,3 cm;
  - Ângulo efetivo: inferior a 15°;
  - Ângulo de medição: 30°;
  - Largura do pulso de entrada do acionador: 10  $\mu$ s
  - Dimensões: 45 x 20 x 15 mm;
  - Peso: 9 g
  - Custo: R\$ 7,80



Figura 12: Sensor de Distância Ultrassônico - HC-SR04.

- Módulo Segue Faixa IR - 5 canais
  - Número de saídas: 5;
  - Tensão de operação: 3,3 a 5V;
  - Leitura: 10 a 20cm;
  - Dimensões: 96,5mm x 20,3mm;
  - Tipo de saída: Digital;
  - Nível da saída digital: 5V;
  - Custo: R\$ 25,00

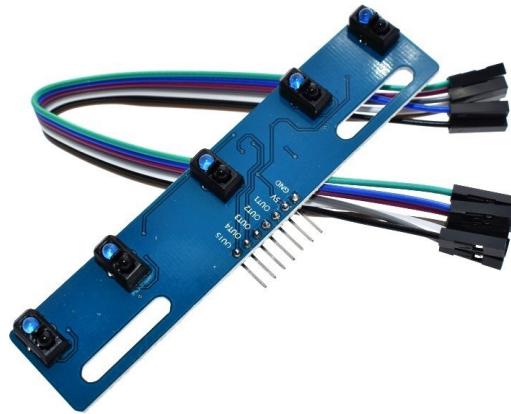


Figura 13: Módulo Segue Faixa IR - 5 canais.

- Sensor Velocidade Módulo Encoder Acoplador Óptico
  - Tensão de operação: 3,3 a 5 V;
  - Abertura disco encoder: 5 mm;
  - Saída Digital e Analógica;
  - Dimensões: 32 x 14 x 12 mm;
  - Peso: 2 g;
  - Custo: R\$ 7,50

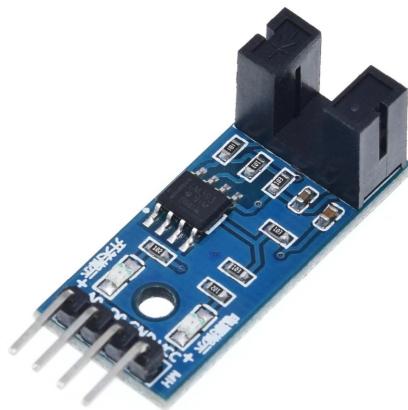


Figura 14: Sensor Velocidade Módulo Encoder Acoplador Óptico.

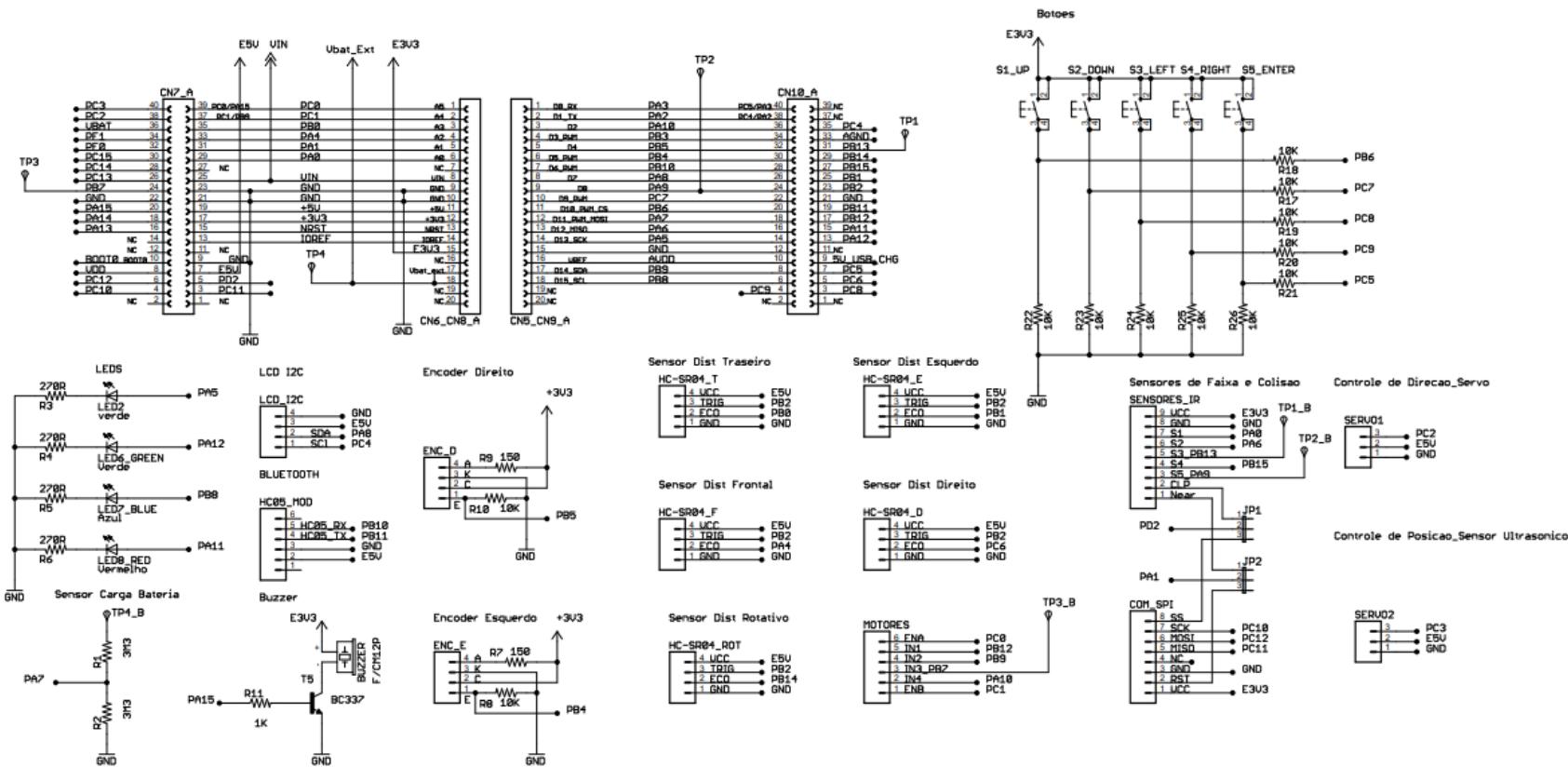
- Kit Motor Dc 3-6 V + Roda - Redutor Plastico
  - Tensão de Operação: 3-6 V;
  - Redução: 1:48;
  - Corrente sem carga:  $\leq 200 \text{ mA}$  (6 V) e  $\leq 150 \text{ mA}$  (3 V);
  - Velocidade sem carga: 200 RPM (6 V) e 90 RPM (3 V);
  - Diâmetro da roda: 68 mm;
  - Largura da roda: 26 mm;
  - Furo central: 5,3 x 3,66 mm (Semicírculo);
  - Peso: 80 g;



Figura 15: Kit Motor Dc 3-6 V + Roda - Redutor Plastico.

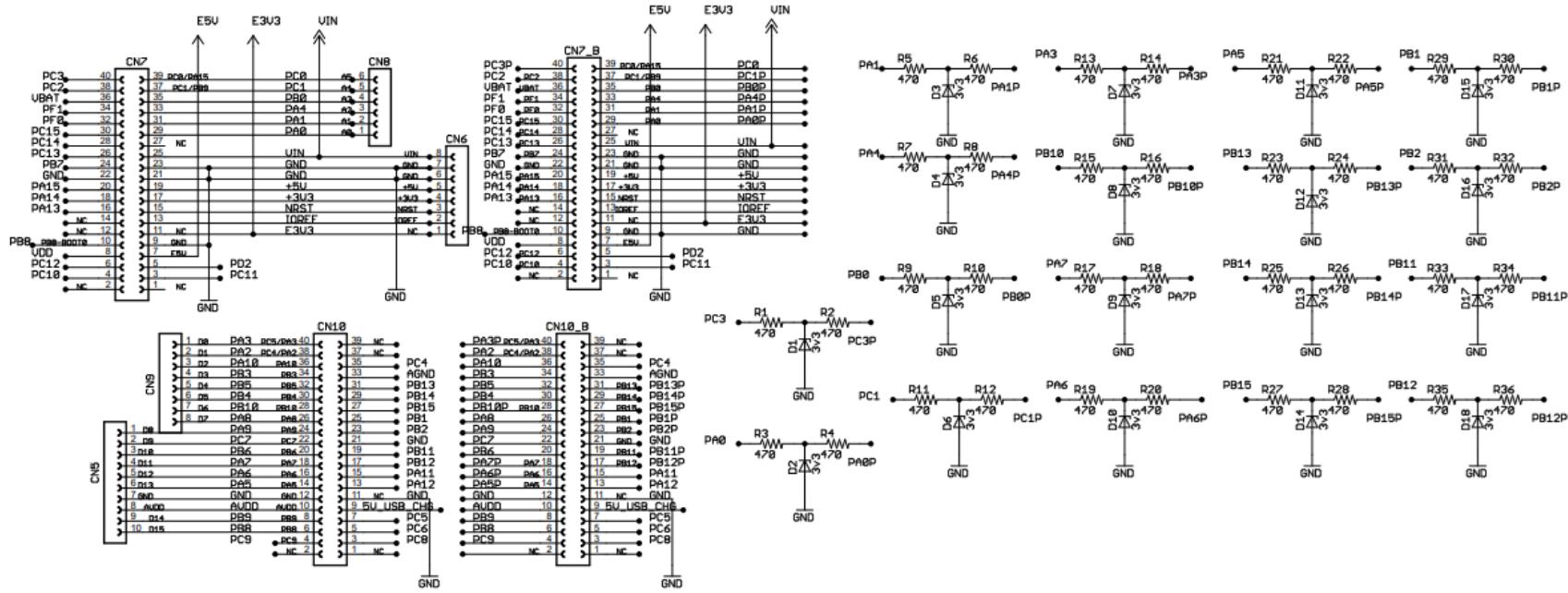
## Esquemático Placa Seguidor de Linha (V2)

STM32G4\_Projetos RSL rev1.2



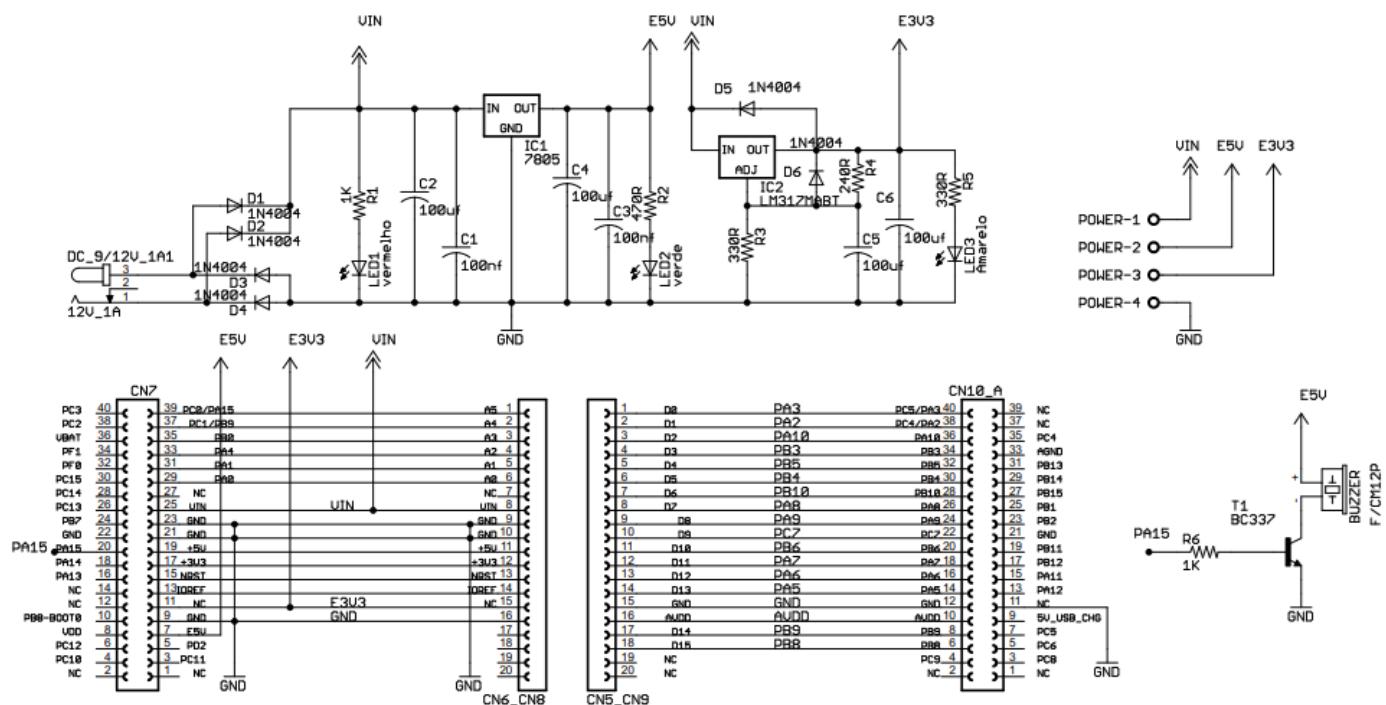
## Esquemático Placa de Proteção

STM32G4\_FireWall rev2.0



## Esquemático Placa Fonte

STM32G4\_Fonte\_Base rev0.0



## Descrição dos pinos da Placa Robô Seguidor de Linha

Função	Barramento Kit	Pino Kit	Pino ARM	Nome Pino ARM	Função	Periférico ARM	Periférico Externo ao Kit ARM
Controle de velocidade do motor esquerdo	CN7	38	8	PC0	PWM	TIM1_CH1	Motor_Esq_PWM
Controle de velocidade do motor direito	CN7	36	9	PC1	PWM	TIM1_CH2	Motor_Dir_PWM
Controle de direção	CN7	35	10	PC2	PWM	TIM1_CH3	Direcao_Servo_PWM
Controle de rotação do sensor de distância	CN7	37	11	PC3	PWM	TIM1_CH4	Distancia_Servo_PWM
Sensor de faixa (IR1)	CN7	28	12	PA0	Entrada Analógica	ADC1_IN1	IR1_AD
Sensor de distância infravermelho frontal	CN7	30	13	PA1	Entrada Analógica	ADC1_IN2	IR_Fr_AD
Eco do sensor de distância ultrassônico frontal	CN7	32	18	PA4	Timer - Captura	TIM3_CH2	Ultra_Fr_Eco_TIM
Sensor de faixa (IR2)	CN10	13	20	PA6	Entrada Analógica	ADC2_IN3	IR2_AD
Sensor de carga da bateria	CN10	15	21	PA7	Entrada Analógica	ADC2_IN4	Bateria_AD
Controle do Display LCD por I2C	CN10	34	22	PC4	I2C SCL	I2C2_SCL	LCD_I2C_SCL
Botão Enter	CN10	6	23	PC5	Entrada Digital	GPIO_EXTI5	BT_Enter
Eco do sensor de distância ultrassônico traseiro	CN7	34	24	PB0	Timer - Captura	TIM3_CH3	Ultra_Tr_Eco_TIM
Eco do sensor de distância ultrassônico esquerdo	CN10	24	25	PB1	Timer - Captura	TIM3_CH4	Ultra_Es_Eco_TIM
Trigger de todos os sensores ultrassônicos	CN10	22	26	PB2	PWM	TIM20_CH1	Ultra_All_Trig_PWM
Comunicação Bluetooth ou Wi-Fi (UART TX)	CN10	25	30	PB10	UART TX	USART3_TX	HC05_RX
Comunicação Bluetooth ou Wi-Fi (UART RX)	CN10	18	33	PB11	UART RX	USART3_RX	HC05_TX
Controle do sentido de rotação do motor esquerdo	CN10	16	34	PB12	Saída Digital	GPIO_Output	Motor_Dir_IN1
Sensor de faixa (IR3)	CN10	30	35	PB13	Entrada Analógica	ADC3_IN5	IR_3_AD
Eco do sensor de distância ultrassônico rotativo	CN10	28	36	PB14	Timer - Captura	TIM15_CH1	Ultra_Rot_Eco_TIM
Sensor de faixa (IR4)	CN10	26	37	PB15	Entrada Analógica	ADC4_IN5	IR4_AD
Eco do sensor de distância ultrassônico direito	CN10	4	38	PC6	Timer - Captura	TIM3_CH1	Ultra_Di_Eco_TIM
Botão Baixo	CN10	19	39	PC7	Entrada Digital	GPIO_EXTI7	BT_Baixo
Botão Esquerda	CN10	2	40	PC8	Entrada Digital	GPIO_EXTI8	BT_Esq
Botão Direita	CN10	1	41	PC9	Entrada Digital	GPIO_EXTI9	BT_Dir
Controle do Display LCD por I2C	CN10	23	42	PA8	I2C SDA	I2C2_SDA	LCD_I2C_SDA
Sensor de faixa (IR5)	CN10	21	43	PA9	Entrada Analógica	ADC5_IN2	IR5_AD
Controle do sentido de rotação do motor direito	CN10	33	44	PA10	Saída Digital	GPIO_Output	Motor_Esq_IN4
Led Vermelho	CN10	14	45	PA11	PWM	TIM4_CH1	LED_R_PWM
Led Verde	CN10	12	46	PA12	PWM	TIM4_CH2	LED_G_PWM
Buzzer	CN7	17	51	PA15	PWM	TIM8_CH1	Buzzer_PWM
Conexão IMU (SPI SCK)	CN7	1	52	PC10	SPI SCK	SPI3_SCK	IMU_SPI_SCK
Conexão IMU (SPI MISO)	CN7	2	53	PC11	SPI MISO	SPI3_MISO	IMU_SPI_MISO
Conexão IMU (SPI MOSI)	CN7	3	54	PC12	SPI MOSI	SPI3_MOSI	IMU_SPI_MOSI
Sensor de colisão frontal	CN7	4	55	PD2	Entrada Digital	GPIO_EXTI2	Switch_Fr
Encoder roda esquerda	CN10	27	57	PB4	Timer - Captura	TIM16_CH1	Encoder_Esq_TIM
Encoder roda direita	CN10	29	58	PB5	Timer - Captura	TIM17_CH1	Encoder_Dir_TIM
Botão Cima	CN10	17	59	PB6	Entrada Digital	GPIO_EXTI6	BT_Cima
Controle do sentido de rotação do motor direito	CN7	21	60	PB7	Saída Digital	GPIO_Output	Motor_Esq_IN3
Led Azul	CN10	3	61	PB8-BOOT0	PWM	TIM4_CH3	LED_B_PWM
Controle do sentido de rotação do motor esquerdo	CN10	5	62	PB9	Saída Digital	GPIO_Output	Motor_Dir_IN2

### Diagrama de Camadas

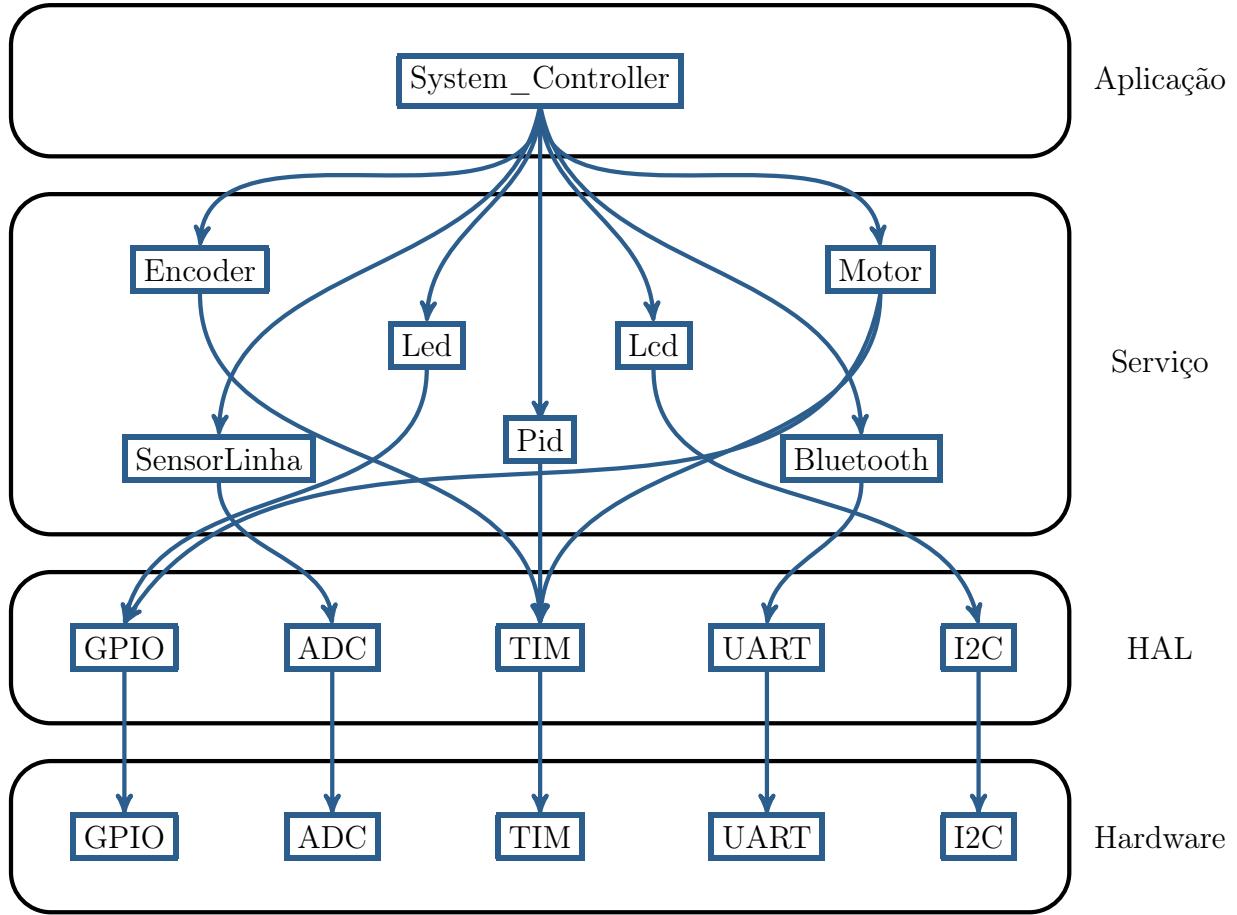


Figura 16: Diagrama de camadas do sistema.

## Diagrama de Classes

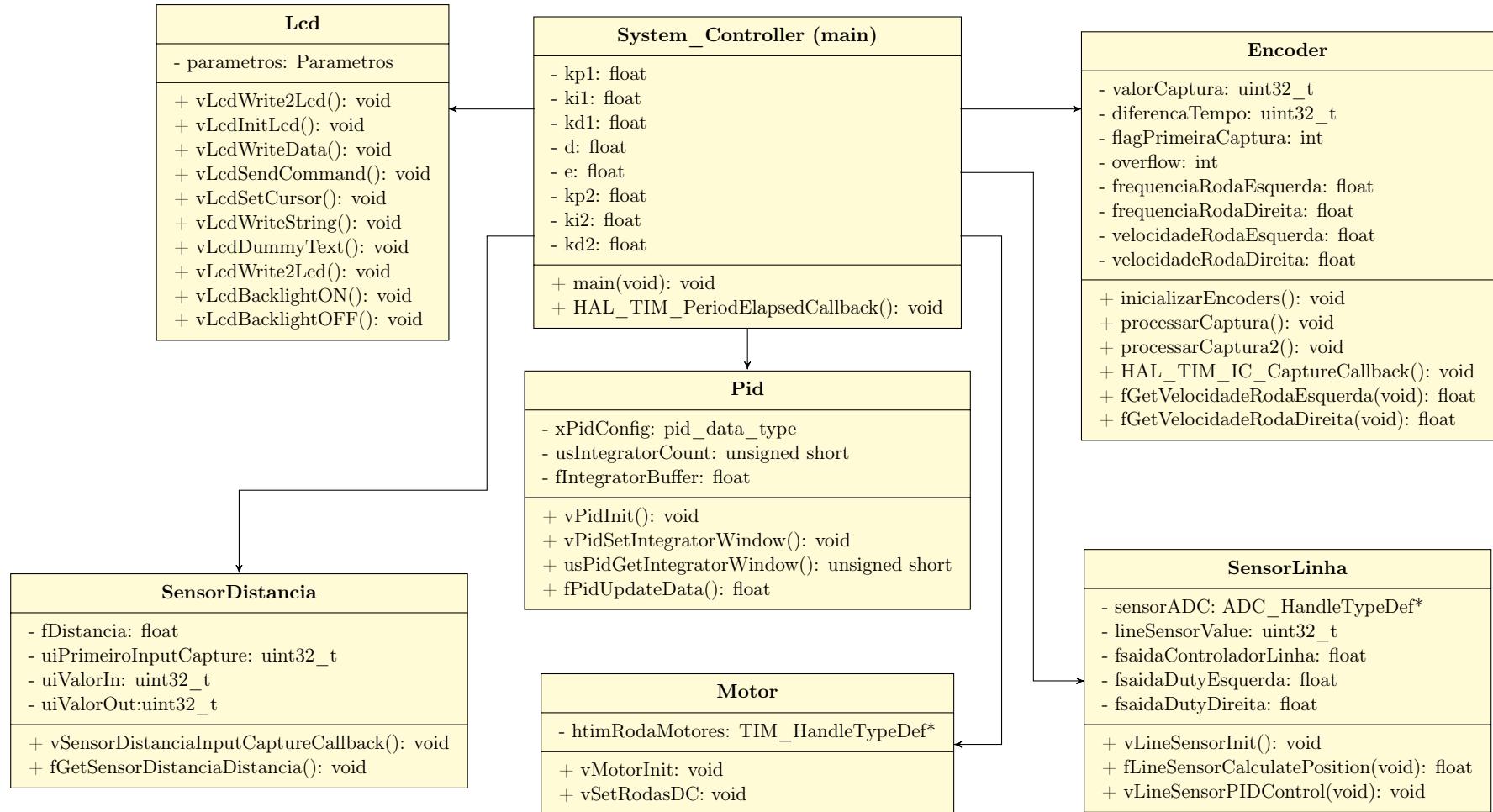


Figura 17: Diagrama de classes do sistema.

## 3 Display LCD

A biblioteca utilizada foi desenvolvida para a interface e controle de displays LCD baseados no chip Hitachi HD44780U utilizando comunicação via I2C. Ela foi criada por Nikita Bulaev em 2017 e modificada pelo Prof. Bacurau para utilização no projeto desenvolvido.

### 3.1 Características Gerais

#### 3.1.1 Comunicação I2C

A conexão com o display se dá por meio de um barramento I2C, permitindo controle eficiente através dos pinos (SDA e SCL), além da alimentação do dispositivo.

#### 3.1.2 Modos de Operação

Possui suporte para operação em modo de 4 bits, reduzindo a quantidade de conexões físicas necessárias entre o microcontrolador e o display.

#### 3.1.3 Funcionalidades

- Inicialização do display com parâmetros ajustáveis, como número de linhas e colunas.
- Controle do cursor (habilitar/desabilitar, movimentar à direita/esquerda, ativar/desativar piscamento).
- Controle da iluminação de fundo (backlight).
- Impressão de strings e caracteres.
- Posicionamento livre do cursor no display.
- Limpeza de dados do display.

#### 3.1.4 Funções principais

- lcdPrintStr: Exibe uma string a partir da posição atual do cursor.
- lcdSetCursorPosition: Move o cursor para uma posição especificada do display.
- vPrintMotorSpeed: Projetada para exibir as velocidades das rodas esquerda e direita no formato “Speed L: XX,YY” e “Speed R: XX,YY”, com precisão de duas casas decimais, sendo a velocidade da roda esquerda na primeira linha e a da roda direita na segunda linha.

## 4 Bluetooth

### 4.1 Configuração do Módulo Bluetooth HC-05

Para a utilização do módulo Bluetooth HC-05 no robô, é necessária uma configuração inicial, realizada com o auxílio do software **Config HC**, desenvolvido pelo Professor Leandro Dantas, e um adaptador USB.

#### 4.1.1 Passos para Configuração

1. Conectar o módulo HC-05 ao adaptador USB.
2. Utilizar o software Config HC para definir as configurações iniciais, como o *Baud Rate*. No nosso caso, o valor utilizado foi **38400 Hz**.
3. Salvar as configurações e desconectar o módulo.

Após a configuração inicial, o módulo pode ser conectado ao robô e emparelhado com o smartphone.

### 4.2 Aplicativo de Controle

O controle do robô é realizado por meio do aplicativo **Arduino Bluetooth Controller**, disponível na Play Store. Este aplicativo oferece três modos de controle distintos:

#### 4.2.1 Modos de Controle

**D-Pad** Este modo utiliza um *joystick* para controlar os movimentos do robô:

- **Seta para cima:** Avançar.
- **Seta para baixo:** Parar os motores.
- **Seta para a direita:** Mover para a direita.
- **Seta para a esquerda:** Mover para a esquerda.

**Botões** Além dos controles direcionais (1, 2, 3, 4), este modo permite o acionamento de periféricos como LEDs e a coleta de dados. Por exemplo:

- Ligar/desligar LEDs.
- Coletar informações de velocidade e colisão.

**Terminal** O terminal permite inserir caracteres específicos para controlar o robô diretamente. A tabela abaixo resume as ações disponíveis:

Comando	Ação
9	Acender LEDs
8	Apagar LEDs
7, 6	Ações a definir
5	Relatório do funcionamento
4	Parar as rodas
3	Mover para a esquerda
2	Mover para a direita
1	Andar reto

Tabela 1: Comandos no modo Terminal.

### 4.3 Conexão Inicial com o Smartphone

Após a configuração, o módulo HC-05 deve ser emparelhado com o smartphone:

1. Ativar o Bluetooth do smartphone.
2. Procurar por dispositivos e selecionar o HC-05.
3. Inserir o código PIN padrão (**1234**).

Uma vez conectado, o aplicativo pode ser utilizado para controlar o robô.

## 5 Encoders

O código foi projetado para capturar os sinais dos encoders das duas rodas do robô seguindo de linha e calcular suas velocidades. Ele utiliza as interrupções baseadas nos timers para medir a diferença de tempo entre as capturas de pulsos dos encoders, converte essas informações em frequências e depois as usa para calcular a velocidade de cada roda. A velocidade final é calculada com base no raio do encoder e nas configurações do microcontrolador, permitindo que o robô ajuste seu movimento conforme necessário. De posse desta velocidade, o lcd pode exibir a velocidade atual e o PID consegue também receber o retorno dos ajustes feitos na rota pelo método de pesos utilizado.

## 6 Sensor Ultrassônico

O sensor ultrassônico HC-SR04 foi integrado ao projeto para realizar medições de distância com precisão. Ele utiliza pulsos ultrassônicos para determinar a distância entre o sensor e um objeto em seu caminho. Desta forma, impede que o carrinho colide com algum obstáculo à sua frente.

### 6.1 Características Gerais

#### 6.1.1 Princípio de Funcionamento

O sensor emite ondas ultrassônicas através de seu pino TRIG e detecta o eco refletido no pino ECHO. O tempo entre o envio do pulso e o retorno do eco é utilizado para calcular a distância, utilizando a fórmula:

$$\text{Distância (em cm)} = \frac{\text{Duração do Pulso (em } \mu\text{s)} \times 0,034}{2}$$

#### 6.1.2 Especificações

- Faixa de medição: 2 cm a 400 cm.
- Resolução: 0,3 cm.
- Tensão de operação: 5V.
- Interface: GPIO para pinos TRIG e ECHO.

#### 6.1.3 Conexão com o STM32

- **VCC:** Alimentação (5V).
- **GND:** Terra.
- **TRIG:** Conectado a um pino GPIO configurado como saída.
- **ECHO:** Conectado a um pino GPIO configurado como entrada, monitorado por um temporizador.

## 6.2 Funcionalidades

- Medição de distâncias em tempo real.
- Integração com temporizadores do STM32 para medir a duração do pulso no pino ECHO.
- Calibração e adaptação às condições ambientais (temperatura e obstáculos).

### 6.3 Funções principais

- **Sensor Trigger:** Envia o pulso TRIG para iniciar a medição.
- **Echo:** Calcula a duração do pulso recebido no pino ECHO.
- **Calcular Distância:** Realiza o cálculo da distância com base na duração do pulso capturado.

## 7 Manual de Utilização

O **Manual de Utilização** do robô seguidor de linha tem como objetivo orientar o usuário na operação eficiente e segura do dispositivo. Antes de iniciar, assegure-se de que todas as conexões de hardware estão devidamente estabelecidas e que a bateria está carregada. Para ligar o robô, pressione o botão de energia; a inicialização será exibida no display LCD. Verifique o funcionamento básico dos sensores de faixa posicionando o robô sobre uma linha e garantindo que todos os sensores estejam corretamente calibrados para detectá-la.

O controle do robô pode ser realizado de duas formas: remotamente, via aplicativo Android (ANDROID BLUETOOTH CONTROLLER), ou em modo autônomo. No modo remoto, utilize o aplicativo para enviar comandos como avançar, virar à esquerda ou à direita, e ajustar a luzes LED. Além disso, o aplicativo permite acessar relatórios detalhados, incluindo a velocidade das rodas e outros dados operacionais. No modo autônomo, posicione o robô sobre a linha desejada e ative a funcionalidade de seguimento. O robô ajustará sua trajetória automaticamente, utilizando sensores de faixa para detectar a linha e o sensor ultrassônico para evitar colisões.

Durante o uso, o display LCD exibirá informações úteis, como nível de carga da bateria, distância percorrida, e estimativa de autonomia restante. Caso o robô detecte um obstáculo, ele interromperá o movimento e soará um alarme para alertar o operador. Se necessário, revise o alinhamento dos sensores e a configuração do firmware para garantir a máxima eficiência do sistema.

Para manter o robô em boas condições, realize inspeções periódicas nos sensores, conexões elétricas e bateria. Limpe os sensores de faixa para evitar falhas de detecção e atualize o firmware sempre que novas versões estiverem disponíveis. Com essas orientações, você poderá utilizar o robô de forma prática e eficiente, explorando todas as suas funcionalidades.

## 8 Problemas Identificados e não Resolvidos

A implementação do sensor ultrassônico no robô não foi concluída, o que impede o funcionamento pleno da funcionalidade de detecção de obstáculos. A maior dificuldade estava relacionada a chamada da função callback do modo input capture. Caso seja implementada com sucesso, será apresentada na última entrega. Além disso, o sensor de colisão apresentou problemas. Em ambos os carrinhos o sensor frequentemente não alterava o nível (alto/baixo) ao ser acionado, tornando-o incapaz de detectar impactos de maneira confiável. A odometria, responsável pelo monitoramento da distância percorrida e da postura do robô, também não está funcionando, comprometendo o acompanhamento preciso das grandezas relacionadas ao movimento.

## 9 Autoria dos Códigos Fornecidos

O código necessário para implementar o funcionamento do sensor ultrassônico foi obtido da internet, com o objetivo de garantir uma solução eficiente e testada previamente. Esse código foi integrado ao projeto para atender aos requisitos específicos de detecção de obstáculos no robô seguidor de linha.

HCSR04 Ultrasonic sensor and STM32. Disponível em:  
<https://controllerstech.com/hcsr04-ultrasonic-sensor-and-stm32/>. Acesso em: 2024.