ColorTrack

Catarina Carvalho de Sousa
dept. de Eletrónica Industrial
(of Universidade do Minho)
Universidade do Minho
Braga, Portugal
a97259@uminho.pt

Isabel Maria Carvalho da Costa Gomes
dept. de Eletrónica Industrial
(of Universidade do Minho)
Universidade do Minho
Braga, Portugal
a104868@uminho.pt

Rui Jorge Carvalho Sampaio
dept. de Eletrónica Industrial
(of Universidade do Minho)
Universidade do Minho
Braga, Portugal
a101037@uminho.pt

Este projeto consistiu no desenvolvimento de um *robot* capaz de seguir linhas coloridas, simulando o comportamento de um veículo num ambiente urbano. O *robot* desloca-se por diferentes áreas da maquete e reage conforme instruções previamente definidas. Foram estabelecidos requisitos que fornecem as funcionalidades pretendidas ao *robot*, nomeadamente: o seguimento de linhas na maquete desenvolvida, a capacidade de reconhecer a cor da linha, e a resposta adequada a cada cor detetada. Assim, perante uma linha verde, o *robot* avança à velocidade máxima; perante uma linha azul, reduz a velocidade; e perante uma linha vermelha, para.

Palavras-Chave - *robot*, seguidor de linha, maquete, linhas coloridas, sensor de linha, ponte H, *STM32, TCS3200*.

I. Introdução

No âmbito da unidade curricular de Projeto Integrador em Engenharia Eletrónica Industrial e Computadores 2, foi desenvolvido um *robot* autónomo com capacidade para seguir linhas coloridas. O principal objetivo consistiu na implementação de um sistema que integrasse perceção, tomada de decisão e controlo, com base na interpretação de diferentes cores no percurso. Para tal, foi necessário integrar sensores de cor, motores e lógica de controlo, permitindo ao *robot* adaptar o seu comportamento com base nas cores da linha que percorre. Durante o desenvolvimento, foram definidos requisitos específicos que guiaram a implementação das funcionalidades desejadas. Entre eles destaca-se a capacidade de seguir com precisão o percurso estabelecido, identificar diferentes cores de linha e responder de forma coerente: acelerar, abrandar ou parar. Este relatório descreve as etapas de conceção, desenvolvimento e testes do *robot*.

II. Materiais e Métodos

A. Dispositivos e Componentes utilizados

Os principais componentes utilizados no desenvolvimento do *robot* foram: microcontrolador STM32, módulo seguidor de linha com 5 sensores infravermelhos, sensor de cor RGB TCS3200, 2 pontes H e o kit carrinho *robot*.

B. Funções dos dispositivos/componentes utilizados

A *STM32* foi utilizada como unidade principal de processamento e controlo do sistema, sendo responsável pela aquisição de dados dos sensores, execução dos algoritmos de seguimento da linha e tomada de decisão em tempo real. O módulo seguidor de linha, composto por cinco sensores infravermelhos, permitiu detetar com elevada precisão a posição relativa da linha no solo, garantindo que o *robot* se mantinha no percurso definido, mesmo em trajetos com curvas. A disposição linear dos sensores possibilita ao sistema calcular o desvio da linha em relação ao centro do *robot* e ajustar a direção de deslocação de forma proporcional. O sensor de cor *RGB TCS3200* foi utilizado para identificar a cor da linha, permitindo ao *robot* adaptar dinamicamente o seu comportamento de acordo com o contexto. Através da análise da frequência de saída associada a cada cor, o sistema consegue distinguir as diferentes cores. A estrutura móvel foi construída com recurso a um *kit robot*, que serviu de base para todos os componentes, assegurando a estabilidade e mobilidade do sistema. A deslocação foi assegurada por motores *DC*, controlados através da ponte H, que permite comandar tanto o sentido de rotação como a velocidade dos motores. Este controlo foi realizado com recurso ao *PWM*, gerado pela STM32, garantindo uma resposta suave e precisa durante o movimento do *robot*.

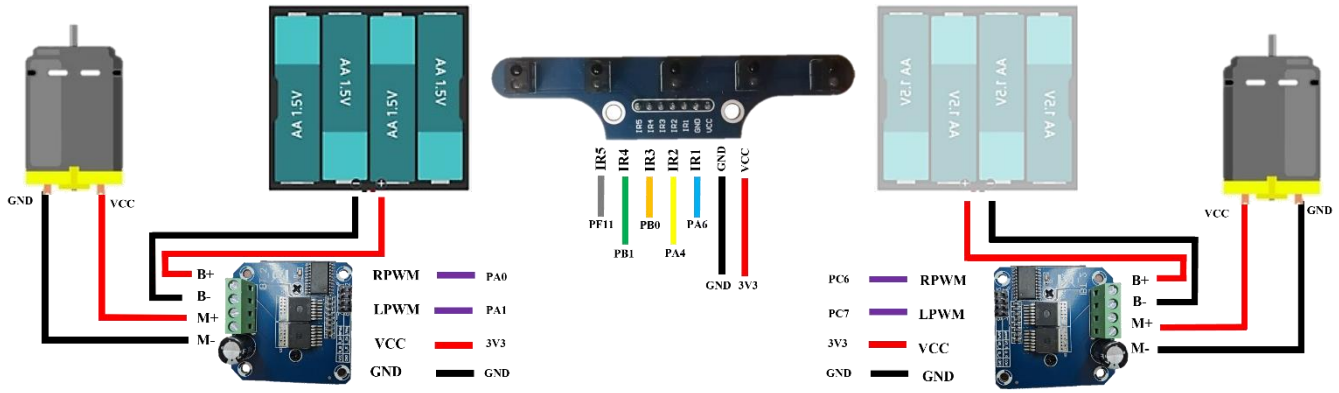

Fig. 1. Esquema de ligações

III. MÓDULO SEGUIDOR DE LINHA

Nesta secção III, são descritos os métodos utilizados que conferem ao *robot ColorTrack* a capacidade de seguir a linha, conforme será detalhado nas subsecções abaixo.

A. Princípio de funcionamento do Seguidor de linha

Para que o *robot* conseguisse efetuar o seguimento da linha, foi utilizado um módulo composto por cinco sensores infravermelhos, como representado na Fig. 2. Este módulo funciona através da emissão e receção de luz infravermelha, sendo capaz de identificar variações de contraste na superfície, como linhas escuras sobre fundo claro e vice-versa.

Conforme ilustrado na Fig. 3, cada sensor é constituído por um emissor (LED infravermelho) e um recetor (fototransístor ou fotodíodo). Quando a luz emitida incide numa superfície clara, é refletida de volta ao recetor, sinalizando a presença de fundo claro. Em superfícies escuras, a luz é absorvida e pouco ou nenhum sinal é refletido, permitindo identificar a linha.

O módulo disponibiliza sinais analógicos ou digitais que indicam a intensidade da luz refletida. Estes sinais permitem ao *robot* reconhecer a localização da linha e ajustar a direção de movimento, garantindo que segue corretamente o percurso. A disposição linear dos cinco sensores permite detetar com exatidão desvios laterais da linha em relação ao centro do *robot*.

Um exemplo ilustrado na Fig. 4: Quando o sensor IR4 deteta uma superfície escura, o motor desse lado acelera, forçando o *robot* a curvar na direção correspondente. A mesma lógica aplica-se ao sensor IR2.

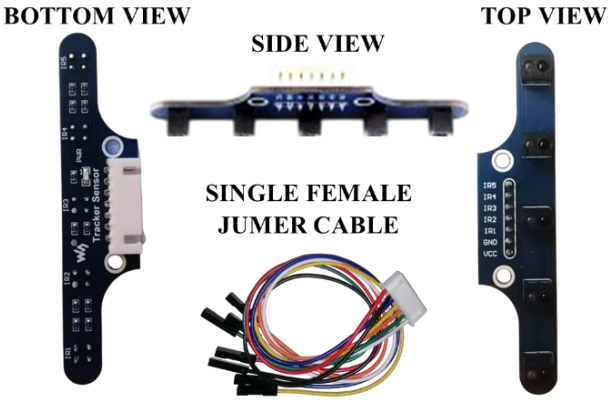

Fig. 2. Módulo seguidor de linha com 5 sensores IR

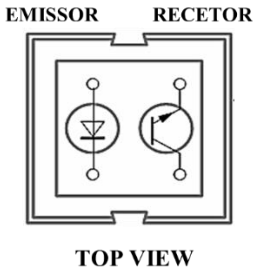

Fig. 3. Composição de cada um dos sensores

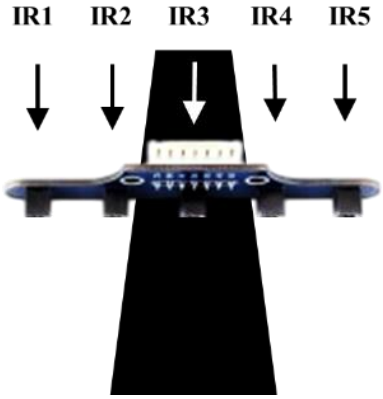

Fig. 4. Exemplo prático de funcionamento

B. Valores obtidos dos sensores

Estão descritos nas Tabela I e Tabela II, os valores obtidos dos cinco sensores, perante a cor branca e preta, respetivamente.

Nota: O módulo foi isolado, para que a luz natural incidente não afetasse os resultados obtidos e assim, obter uma maior precisão.

Com uma tensão de 3,3V:

Perante a cor branca					
Sensores	IR1	IR2	IR3	IR4	IR5
Analógico	3,17 V	3,16 V	3,17 V	3,16 V	3,17 V
Digital	3433	3424	3395	3413	3395

Tabela I

Perante a cor preta					
Sensores	IR1	IR2	IR3	IR4	IR5
Analógico	1 V	0,8 V	0,9 V	0,9 V	0,85 V
Digital	900	745	734	748	752

Tabela II

No caso das leituras analógicas, a gama de tensão varia entre 0 V e 3,3 V, correspondente à faixa de operação do conversor analógico-digital (ADC) da STM32. Já os valores digitais resultantes da conversão encontram-se compreendidos entre 0 e 4095, dado que o ADC utilizado possui uma resolução de 12 bits (2^{12} = 4096 níveis possíveis).

A análise dos dados apresentados nas tabelas confirma o correto funcionamento dos cinco sensores infravermelhos. Os valores obtidos estão de acordo com a relação matemática típica da conversão ADC representada na Equação 1.

$$\text{Valor digital} = \frac{\text{Tensão medida}}{3{,}3\ \text{V}} \times 4096 \qquad (1)$$

C. *Implementação da leitura dos sensores da linha na STM32*

O sistema de seguimento de linha funciona continuamente em loop, com um pequeno atraso (*delay*) entre iterações, garantindo um seguimento dinâmico e em tempo real da linha, reagindo constantemente às variações do trajeto, o diagrama está representado na Fig. 5.

A lógica principal é composta por quatro funções fundamentais:

1) readSensors() - Esta função é responsável por adquirir os valores brutos dos sensores infravermelhos e está demonstrada na Fig. 6:

- Configura sequencialmente cada canal ADC associado a um sensor IR;
- Inicia a conversão analógica-digital, aguarda a sua conclusão e armazena os dados no vetor *sensorValues[i].*

2) readLine() - Processa os dados recolhidos pelos sensores, Fig. 7:

- Inverte os valores lidos com a fórmula (*MAX_SENSOR_VALUE – raw*) para que superfícies pretas correspondam a valores mais altos;
- Calcula a média ponderada da posição da linha, multiplicando o índice do sensor por 1000;
- Retorna a posição estimada da linha, num intervalo de 0 a 4000.

3) PID_LineFollow() - Esta função executa o algoritmo de controlo PID para manter o r*obot* alinhado com a linha, demonstrado o código na Fig.8:

- Calcula o erro, definido como a diferença entre a posição ideal (2000) e a posição atual lida;
- Atualiza os termos P (Proporcional), I (Integral) e D (Derivativo);
- Computa o valor de controlo $PIDvalue = Kp \cdot P + Ki \cdot I + Kd \cdot D$;
- Ajusta as velocidades dos motores esquerdo e direito em função de *PIDvalue*, respeitando os limites de ±255;
- Envia os comandos através da função *motorDrive(leftSpeed, rightSpeed).*

4) *setMotorA(int speed) / setMotorB(int speed)* - Estas funções aplicam os sinais PWM aos motores com base no sinal de controlo, Fig. 9:

- Se o *toggle_flag* estiver ativo, convertem o valor (positivo ou negativo) em sinais de *PWM* e direção;
- O canal *RPWM* recebe $|speed|$ se $speed \geq 0$; caso contrário, recebe 0;
- O canal *LPWM* recebe $|speed|$ se $speed < 0$; caso contrário, recebe 0.

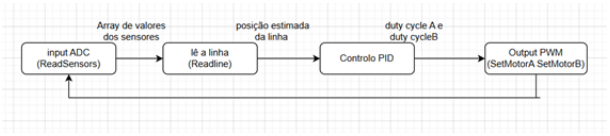

Fig. 5. Modelo de seguimento da linha

```c
void readSensors(void) {
    ADC_ChannelConfTypeDef sConfig = {0};

    for (int i = 0; i < SENSOR_COUNT; i++) {
        HAL_ADC_Stop(&hadc1); // Garante que o ADC não está ativo

        sConfig.Channel = sensorChannels[i];
        sConfig.Rank = ADC_REGULAR_RANK_1;
        sConfig.SamplingTime = ADC_SAMPLETIME_64CYCLES_5;
        sConfig.SingleDiff = ADC_SINGLE_ENDED;
        sConfig.OffsetNumber = ADC_OFFSET_NONE;
        sConfig.Offset = 0;

        if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK) {
            printf("Erro a configurar canal %d\n", i);
            continue;
        }

        HAL_ADC_Start(&hadc1);

        if (HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY) == HAL_OK) {
            sensorValues[i] = HAL_ADC_GetValue(&hadc1);
            printf("Sensor %d: valor %lu\r\n", i + 1, sensorValues[i]);
        } else {
            printf("Timeout no canal %d\n", i);
        }

        HAL_ADC_Stop(&hadc1);
    }
}
```

Fig. 6. Código da função readSensors

```c
uint16_t readLine(void){
  uint32_t weighted_sum = 0;
  uint32_t sum = 0;

  for (int i = 0; i < SENSOR_COUNT; i++)
  {
    uint16_t value = MAX_SENSOR_VALUE - sensorValues[i];
    weighted_sum += (uint32_t)value * (i * 1000);
    sum += value;
  }

  /* if (sum < 5000) return 2000;  // linha perdida (valor neutro)*/

  return weighted_sum / sum;
}
```

Fig. 7. Código da função readLine

```c
void PID_LineFollow(int error)
{
  P = error;
  I += error;
  D = error - previousError;

  PIDvalue = (Kp * P) + (Ki * I) + (Kd * D);
  previousError = error;

  int lsp = baseSpeed - PIDvalue;
  int rsp = baseSpeed + PIDvalue;

  if (lsp > 255) lsp = 255;
  if (lsp < -255) lsp = -255;
  if (rsp > 255) rsp = 255;
  if (rsp < -255) rsp = -255;

  motorDrive(lsp, rsp);
}
```

Fig. 8. Código da função PID_LineFollow

```c
void setMotorB(int speed) // Controla o motor B
{
    if(toggle_flag){
  if (speed >= 0)
  {
      __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, speed);  // RPWM
      __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_2, 0);      // LPWM
  }
  else
  {
        speed = -speed;
      __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, 0);      // RPWM
      __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_2, speed);  // LPWM
  }
}
```

Fig. 9.Código da função setMotorB (int speed)

IV. SENSOR *RGB TCS3200*

Para a deteção da cor da linha, foi utilizado o sensor de cor *RGB TCS3200*, representado na Fig. 10, capaz de detetar e medir uma grande variedade de cores visíveis com precisão.

A. Princípio de funcionamento do sensor detetor de cor

O sensor *TCS3200* é constituído por uma matriz de 8×8 fotodíodos, totalizando 64, em que os fotodíodos da mesma cor estão ligados em paralelo. A distribuição dos filtros é a seguinte:

- Dezasseis fotodíodos com filtro azul;
- Dezasseis fotodíodos com filtro verde;
- Dezasseis fotodíodos com filtro vermelho;
- Dezasseis fotodíodos transparentes (sem filtro).

Este sensor combina fotodíodos de silício configuráveis com um conversor de corrente para frequência (light-to-frequency converter) num único circuito integrado. A saída do sensor é uma onda quadrada com Duty-Cycle de 50%, cuja frequência é proporcional à intensidade da luz recebida, como ilustrado na Fig. 11 e resumido na Tabela IV.

Os pinos S2 e S3, conforme indicado na Tabela III, são utilizados para selecionar o grupo de fotodíodos ativo (vermelhos, verdes, azuis ou transparentes), ajustando assim o tipo de luz a ser medida.

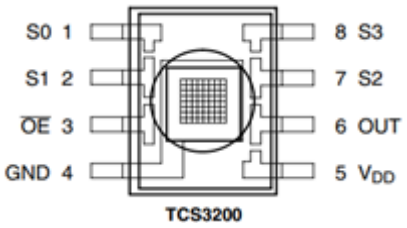

Fig. 10. Sensor de cor RGB TCS3200

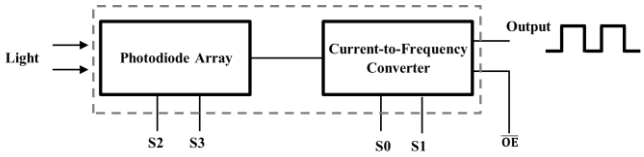

Fig. 11. Funcionamento do TCS3200

S0	S1	OUTPUT FREQUENCY SCALING (fo)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

Tabela IV

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Tabela III

B. Valores obtidos

Relativamente ao *sensor* de cor, não foram incluídos valores medidos no presente relatório, uma vez que, não foi possível finalizar essa parte da implementação até à data de entrega. No entanto, a estrutura do código e a integração física do sensor encontram-se preparadas para testes e validação numa fase posterior.

C. Implementação da leitura do Sensor de Cor TCS3200

Para a deteção de cores no projeto, foi utilizado o sensor TCS3200, que converte a intensidade da luz refletida numa frequência de onda quadrada à saída digital (pino OUT). Esta frequência varia consoante a intensidade da cor selecionada, sendo os filtros de cor ativados através dos pinos de controlo S2 e S3, que permitem alternar entre os canais vermelho, verde e azul.

Na interface com a STM32, o pino OUT do TCS3200 foi ligado a um pino configurado como interrupção externa (EXTI), sensível à borda de subida. A cada transição de nível lógico baixo para alto, uma interrupção é gerada e um contador é incrementado via *software*. Este contador armazena o número total de pulsos recebidos num intervalo de tempo fixo (por exemplo, 100 ms), definido por um *timer* auxiliar ou *delay*, como demonstrado no código da Fig. 12.

Funcionamento da Captura de Frequência

A lógica de leitura da frequência desenvolvida assenta em três etapas principais:

- Seleção da cor: os pinos S2 e S3 são controlados via *GPIO* da STM32 para ativar o filtro de cor pretendido (vermelho, verde ou azul).
- Contagem de pulsos: durante um intervalo de tempo conhecido (e.g., 100 ms), o sistema conta o número de bordas de subida detetadas no pino OUT do TCS3200. Cada borda representa um ciclo da onda gera da pelo sensor.
- Cálculo da frequência: após o tempo decorrido, calcula-se a frequência da onda emitida de acordo com a fórmula descrita na Equação 2:

$$Frequência\ (Hz) = \frac{Número\ de\ pulsos}{Tempo\ de\ medição\ (\triangle t)} \quad (2)$$

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) { //ISR_T
    if (GPIO_Pin == GPIO_PIN_5) {
        count++;
    }
}
```

Fig. 12. Código para o sensor de cor

Este valor é posteriormente analisado e comparado com intervalos de referência, permitindo identificar a cor predominante refletida no momento.

V. Maquete utilizada para o desempenho do colortrack

A maquete desenvolvida constitui o ambiente de circulação do *robot ColorTrack*, onde este realiza a movimentação completa de forma autónoma, representada na Fig. 13. Foi construída sobre uma base plana de cor preta, contendo um percurso fechado com secções de diferentes cores: verde, azul e vermelha, cada uma associada a um comportamento específico do sistema. As zonas verdes representam maioritariamente segmentos de avanço em linha reta, onde o *robot* pode atingir maior velocidade, as azuis correspondem a curvas com velocidade reduzida e os trechos vermelhos indicam pontos de paragem obrigatória. Esta configuração permite simular um cenário urbano simplificado, onde o *robot* segue a linha, identifica variações de cor e reage conforme programado. A construção da maquete teve em conta critérios de clareza visual, coerência de percurso e contraste suficiente para garantir uma deteção eficiente por parte dos sensores do *robot*. A espessura da linha foi escolhida de forma a cobrir toda a base dos cinco sensores infravermelhos dispostos na parte inferior do *robot*, assegurando leituras fiáveis mesmo durante as curvas. Além disso, a disposição do percurso fechado permite que o *robot* funcione de forma contínua, sem necessidade de intervenção externa

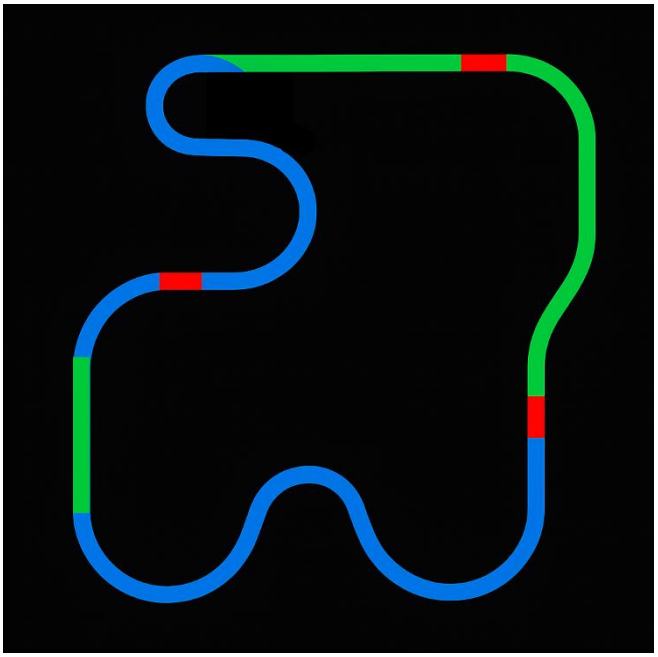

Fig. 13. Protótipo da Maquete desenvolvida

VI. Resultado final do colortarck

A imagem apresentada na Fig. 14, representa o resultado final do *ColorTrack*, já montado e funcional. Nela é possível observar a disposição dos principais componentes: a placa *STM32*, os sensores de seguimento de linha, os motores *DC* com as respetivas rodas, e os módulos *ponte H*. A estrutura foi organizada de forma compacta e equilibrada, de forma a tentar garantir estabilidade durante o movimento e fácil acesso para futuras manutenções ou alterações.

VII. Tabela de custo de componentes utilizados

Na Tabela V apresenta-se a lista completa dos componentes utilizados na construção do *robot ColorTrack*, incluindo os custos associados. Esta análise permite ter uma visão clara do orçamento necessário para a implementação do projeto:

Dispositivos/Componentes	Quantidade	Preço
STM32	1 x	30,5 €
Seguidor de Linha c/5 sensores com cabo- WS	1 x	6,50 €
Sensor de Cor RGB – TCS3200	1 x	8,90 €
Kit carrinho robot	1 x	12,80 €
Protoboard 78x58mm	1 x	1,50 €
Suporte Ic 40 Pinos	1 x	2,40 €
Ponte H	2x	18,00€
	TOTAL:	80,60 €

Tabela V

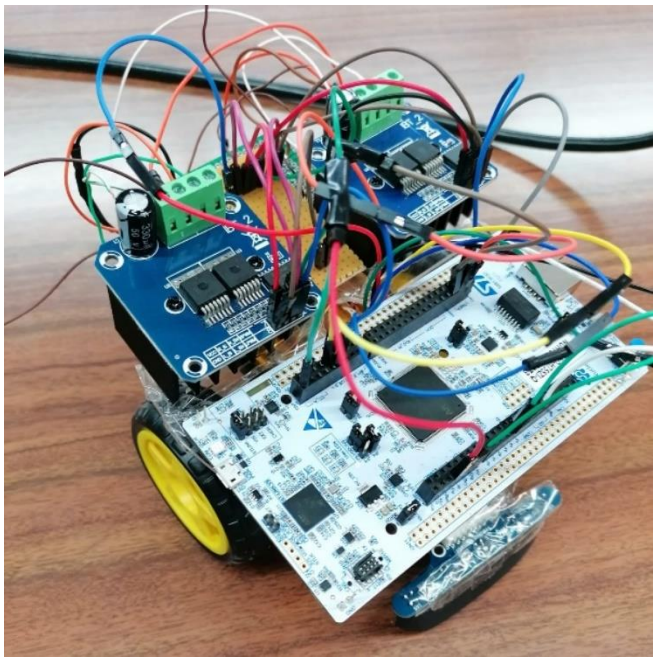

Fig. 14. Resultado final do robot

VIII. CONCLUSÃO

O *robot* apresentou algumas limitações que influenciaram o seu desempenho e a eficiência geral do sistema. Estas limitações, embora não tenham comprometido o funcionamento essencial do ColorTrack, revelaram-se importantes na análise crítica do projeto.

- Controlo manual dos parâmetros *PID*:

O ajuste do controlo *PID* foi realizado manualmente, exigindo várias iterações para encontrar os valores adequados dos parâmetros proporcional, integral e derivativo. Este processo foi demorado e sujeito a tentativa e erro, dificultando a obtenção de um equilíbrio ideal entre a estabilidade do movimento e a velocidade de resposta.

- Sensibilidade dos sensores:

Embora os sensores analógicos utilizados tenham funcionado de forma geral, a sua sensibilidade revelou-se vulnerável a fatores externos, como variações na iluminação ambiente e imperfeições na linha. Estas interferências originaram oscilações nas leituras, dificultando a interpretação precisa da posição da linha pelo sistema de controlo.

- Peso do *robot*:

A instalação de dois módulos ponte H, como é possível verificar na Fig.14, necessária para o controlo independente de ambos os motores DC, contribuiu para um aumento significativo do peso total do *robot*. Este aumento teve impacto direto na aceleração e na capacidade de manobra, exigindo maior esforço dos motores e tornando o sistema mais suscetível a irregularidades presentes na maquete.

- Tensão da bateria:

O sistema depende integralmente de alimentação para o funcionamento da *STM32*, dos sensores e dos motores. Verificou-se que flutuações ou quedas de tensão da bateria resultavam numa redução de velocidade e desempenho, afetando especialmente os momentos de arranque ou exigência de esforço adicional.

- Desigualdades mecânicas entre motores e rodas:

Apesar da utilização de motores idênticos, foram detetadas diferenças no desempenho de cada um, particularmente ao nível da rotação e do binário. Adicionalmente, as rodas apresentavam ligeiros desalinhamentos, provocando desvios mesmo quando ambos os motores recebiam o mesmo sinal PWM. Esta assimetria exigiu compensações adicionais no controlo PID e aumentou a complexidade de manter uma trajetória estável e centrada na linha.

Apesar das limitações identificadas, o ColorTrack demonstrou ser funcional e cumpriu o objetivo traçado: seguir corretamente a linha ao longo de todo o percurso.