

# VEÍCULO TERRESTRE AUTÔNOMO COM DESTINO PRÉ-DEFINIDO

Samuel Moreira da Silva

**Resumo**—Neste artigo apresenta-se um sistema desenvolvido com o Arduino® capaz de se comunicar com um veículo autônomo, que, após receber as coordenadas de um mapa, seguirá até o destino desejado. O veículo deverá se comunicar através do Bluetooth® com o Smartphone, por onde o usuário irá inserir os dados necessários.

**Index Terms**—Robô, ARDUINO®, Sistemas Embarcados, Bluetooth®, Odômetro.

## 1 INTRODUÇÃO

Seja em filmes de ficção científica, missões espaciais, fins militares, ou puro hobby, veículos que se movimentam sem influência humana estão cada vez mais presentes no nosso dia-a-dia, prova disso são os drones, ou VANT - veículo aéreo não tripulado, sendo todo e qualquer tipo de aeronave que pode ser controlada nos 3 eixos e que não necessite de pilotos embarcados para ser guiada (DECEA, 2010). Os drones viraram febre mundial, e a cada dia aumentam suas vendas e surgem novas funções para estes robôs. Já os veículos terrestres são muito utilizados em ambientes hinospitais, seja pra verificar, por exemplo, se há água em um terreno, ou se existem explosivos em um local, oferecendo perigo às pessoas próximas. Seguindo a ideia de indicar através do smartphone o destino do robô, o veículo foi construído em quatro subsistemas.



Figura 1. Veículo militar não tripulado.

## 1.1 Envio De Dados

O sistema Android é responsável por coletar os dados inseridos pelo usuário, em centímetros, e enviá-los via Bluetooth para o Arduino. Figura 1 - Print App Inventor

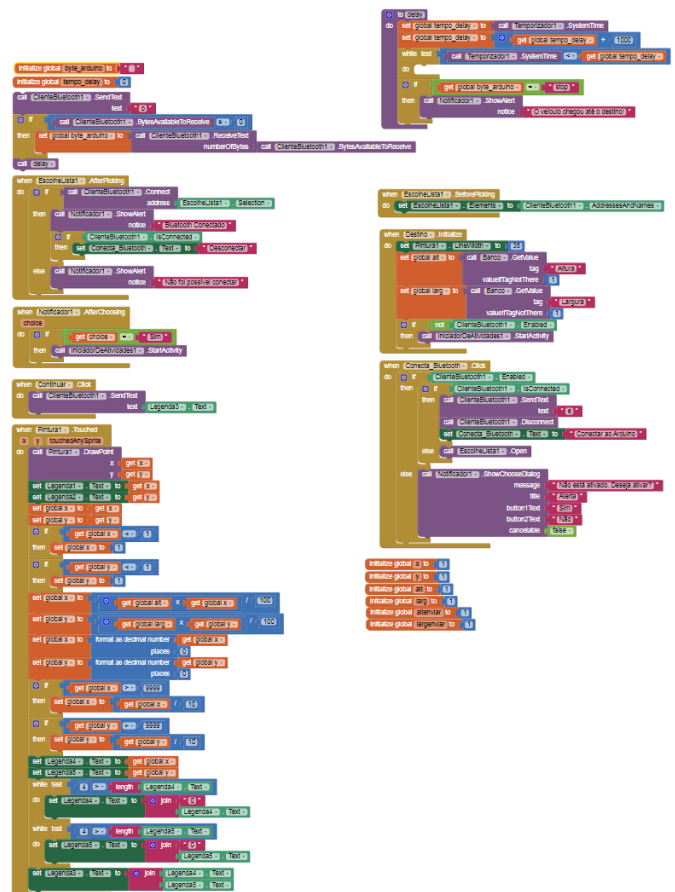


Figura 2. Blocos App Inventor.

## 1.2 Recebimento e Processamento De Dados

O Arduino® recebe os dados através do módulo Bluetooth® HC-05, e identifica, dentro do que foi enviado,

as coordenadas do ponto aonde o veículo deverá chegar.

### 1.3 Atuação do Sistema

Após calcular a distância que o robô irá percorrer, o Arduino seta as saídas dos motores pelo tempo necessário para que ele chegue ao local determinado, através da ponte H, utilizando o módulo L298N.

### 1.4 Leitura de Dados

Para verificar se o veículo chegou ao destino escolhido, foi utilizado um mouse como odômetro, realizando a leitura do deslocamento nos eixos x e y.

## 2 FORMULAÇÃO

A definição de onde o veículo deve chegar é dada pelo usuário através de um app desenvolvido para Android no AppInventor. Na primeira tela do app, é solicitado que o usuário insira as dimensões máximas (em cm) do terreno que veículo percorrerá.

Digite a altura em cm (mín. 10, máx. 1000):

100

Digite a largura em cm (mín. 10, máx. 1000):

100

Próximo

Figura 3. Primeira Tela.

Passando para a segunda tela o usuário deve selecionar na tela do Android, qual ponto do mapa o veículo deve atingir. Como as dimensões da tela são dadas em pixels, se torna necessário uma conversão simples. Após a conversão, que é feita automaticamente pelo aplicativo, o usuário requisita o envio dos dados para módulo Bluetooth®.

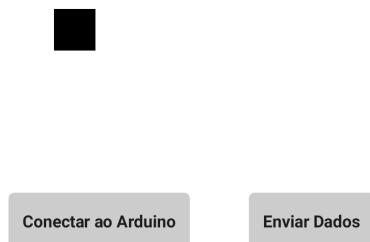


Figura 4. Segunda Tela.

A leitura dos dados pelo módulo HC-05, assim como toda comunicação serial, é feita bit a bit, para garantir que a

leitura seria fidedigna, foi determinado que o par ordenado das coordenadas seria dado, em cada eixo, por um número de 4 dígitos, limitando a 10 metros, já que esta também é distância máxima para comunicação do módulo de classe 2. Ainda sim, para garantir que o sistema lê corretamente os dados recebidos pelo módulo, foi necessário filtrar os dados com um delay de 50ms, pois a cada dois bytes lidos, um era lixo.

```
void loop() {
  while (bluetooth.available()) {
    for (i = 0; i <= 8; i++) {
      total[i] = bluetooth.read();
      delay(50);
    }
  }
}
```

Figura 5. "Filtro" de leitura.

Os dados foram lidos por um vetor de char, pois um int ocupa dois bytes, que também tornam a leitura incorreta. Com uma velocidade das rodas setada em cerca de 70 por cento da capacidade total, o veículo percorre 10 cm a cada 4 segundos, sendo assim basta dividir distância que o veículo deve percorrer por 10, e multiplicar por 4, para encontrar o tempo em que os motores deverão ficar ligados.

## 3 RESULTADOS

Apesar dos resultados não terem sido como esperado, é possível verificar a possibilidade de determinar um destino para um veículo autônomo e garantir que ele chegue ao local desejado. Em relação à leitura do deslocamento, realizada pelo mouse, se mostrou eficiente, e apesar da sensibilidade extremamente alta, cumpre o objetivo.

## 4 ARGUMENTAÇÃO

Após vários testes, ficou fácil perceber que a comunicação entre os sistemas funciona de forma correta, porém, a escolha do método de controle do veículo se mostrou ineficiente devido principalmente à presença do rodízio giratório, conhecido como roda boba, que após uma curva do veículo, permanece desalinhada com o eixo do robô, e portanto, faz com que ele permaneça em curva, mesmo quando não necessário. Além disso, a má fabricação das rodas, assim como o encaixe de seu eixo com o motor, que faz com que as rodas pareçam empenadas, o que também faz com que o veículo ande "sambando". Por fim, ainda há uma pequena diferença natural de velocidade nos motores com a mesma tensão aplicada.

## 5 CONCLUSÃO

Apesar de ter sido possível deslocar o veículo até um ponto próximo ao desejado, este trabalho se mostrou inviável. O problema é muito maior que apenas enviar as coordenadas para o microcontrolador, para garantir que o trajeto será o melhor possível, é necessário determinar um ângulo de trajetória, e um controle proporcional, ainda, para determinar a orientação inicial do veículo se torna necessário

a utilização de um giroscópio, além disso, a utilização do Arduino UNO limita o uso de módulos e sensores, devido ao número menor de pinos, quando comparado a outros modelos. [1] [2] [3]

## REFERÊNCIAS

- [1] D. Cardoso, *Ponte H L298N – Controlando a velocidade de um motor DC com PWM*. Vida de Silício, 2017.
- [2] A. Thomsen, *Como usar o Arduino Bluetooth HC-05 em modo mestre*. 2014.
- [3] jazzycamel, *OPTICAL MOUSE ODOMETER FOR (ARDUINO) RO-BOT*. Instructables, 2015.