

# Projetos com Arduino: Carro Seguidor de linha (Kit Quantum)

Gustavo Vanin Bernardino de Souza  
Professor Especialista em Currículo - Tecnologia  
Diretoria de Ensino - Região de Botucatu



# Sumário

<b>1</b>	<b>Introdução</b>	<b>5</b>
<b>2</b>	<b>Projetos comentados</b>	<b>9</b>
2.1	Projeto 1 - Buzina Simples . . . . .	9
2.2	Projeto 2 - Botão Acende LED . . . . .	10
2.3	Projeto 3 - Buzina com Notas Musicais . . . . .	11
2.4	Projeto 4 - Motor Direito . . . . .	12
2.5	Projeto 5 - Motor Esquerdo . . . . .	13
2.6	Projeto 6 - Sensor Ultrassônico com LED . . . . .	14
2.7	Projeto 7 - Sensor seguidor de linha . . . . .	16
2.8	Projeto 8 - Carro Segue linha . . . . .	17
2.9	Projeto 9 - Carro Segue linha com Parada por obstáculo . . . . .	19



# Capítulo 1

## Introdução

Este material tem como objetivo apresentar projetos com Arduino de forma prática e acessível para estudantes do Ensino Médio e foi desenvolvido especialmente para os kits Quantum. A motivação deste tutorial é que a escola Napoleão Corule em Pardinho-SP possui cinco kits e precisa de orientações de como utilizá-los.

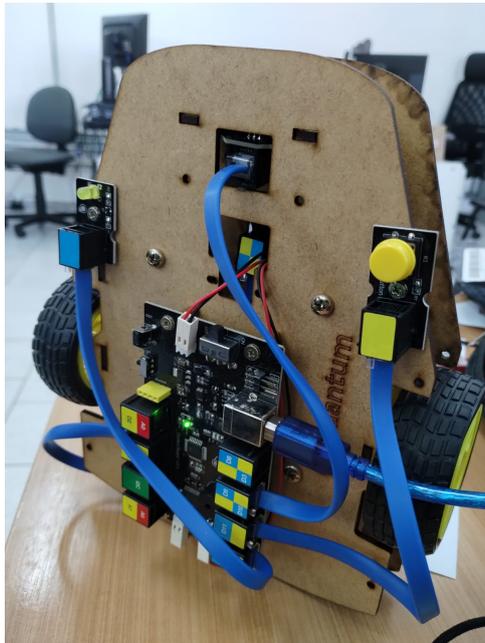


Figura 1.1: Kit Quantum.

Para começar, é necessário instalar a **IDE do Arduino**, que pode ser baixada gratuitamente no site oficial:

<https://www.arduino.cc/en/software/>

## Instalação de drivers

Algumas placas Arduino (especialmente clones) exigem a instalação de drivers adicionais:

- **CP210X (USB to UART Bridge)**

<https://www.silabs.com/software-and-tools/usb-to-uart-bridge-vcp-drivers>

- **CH34X (CH340/CH341)**

[https://www.wch-ic.com/downloads/CH341SER\\_EXE.html](https://www.wch-ic.com/downloads/CH341SER_EXE.html)

## Selecionando a placa e porta correta

Após instalar a IDE e conectar a placa, abra o software e selecione:

- Tipo de placa: **Arduino Uno**
- Porta COM: geralmente será detectada automaticamente (ex: COM3, COM22 etc.)  
Obs: nem sempre será a COM22.
- Se desplugar e plugar novamente a lista de portas atualizarão e você poderá identificar qual a porta correspondente. Se a lista não atualizar a placa não foi reconhecida. Nestes casos geralmente por conta da ausência do drive instalado ou por conta de problemas no cabo que lina no usb do computador.

## Enviando o código



Figura 1.2: Selecionando a placa e porta na IDE do Arduino

Após escrever ou colar o código, clique no botão com uma seta apontando para a direita (lado superior esquerdo) para **enviar o código para o Arduino**.

Certifique-se de que a placa esteja conectada corretamente via USB antes de enviar.

# Componentes do Kit

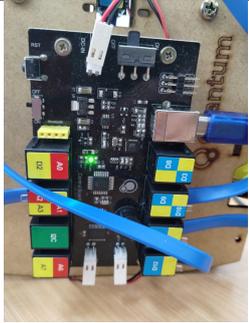
Componente	Imagem
Botão	
LED	
Placa Controladora	
Sensor Seguidor de Linha	
Sensor Ultrassônico	

Tabela 1.1: Componentes do robô.

## Visão geral dos códigos para iniciantes

- Os códigos estão anexados dentro deste pdf e podem ser aberto dando duplo clique no cliques que aparece antes do código;
- `void setup(){ código A... }` é executado uma única vez na inicialização do programa e executa o código que tiver entre as chaves. Neste exemplo seria o "código A...";
- `void loop(){ código B... }` roda repetidamente em um loop infinito o que estiver entre chaves;
- `//comentário` são comentários no código, ou seja, instruções ignoradas pelo compilador.
- `Serial.print("Texto...")` imprime no monitor serial. Para visualizar o texto na IDE do Arduino ir no menu (tools) Ferramentas → (Serial Monitor) Monitor Serial.

# Capítulo 2

## Projetos comentados

### 2.1 Projeto 1 - Buzina Simples

#### Objetivo do Projeto

Este projeto emite um som intermitente por uma buzina conectada à porta digital D13. Ideal como primeiro teste de saída digital.

#### Análise do Código

O pino D13 é colocado em nível alto (HIGH), e aciona a buzina intermitentemente com ajuda da função delay (espere) com um intervalo de 1 segundo (1000 ms).

#### Código

Duplo clique no cliques ao lado para abrir: 

```
1 int pinoBuzina = 13; // Altere aqui para testar outras portas
2
3 void setup() {
4   pinMode(pinoBuzina, OUTPUT);
5 }
6
7 void loop() {
8   tone(pinoBuzina, 1000); // Toca som de 1000 Hz
9   delay(1000);           // Espera 1 segundo
10  noTone(pinoBuzina);    // Para o som
11  delay(1000);           // Espera mais 1 segundo
12 }
```

## 2.2 Projeto 2 - Botão Acende LED

### Objetivo do Projeto

Este projeto acende um LED na porta D12 quando um botão na porta D11 é pressionado.

### Análise do Código

O botão é lido com `digitalRead`. Quando pressionado (HIGH), o LED na D12 é aceso com `digitalWrite`.

### Código

Duplo clique no cliques ao lado para abrir: 

```
1 int botao = 11;          // Pino do botão
2 int led = 12;           // Pino do LED
3 int val = 0;            // Variável que armazena o estado do botão
4
5 void setup() {
6   pinMode(botao, INPUT); // Define o botão como entrada
7   pinMode(led, OUTPUT);  // Define o LED como saída
8   Serial.begin(9600);    // Inicia comunicação serial
9 }
10
11 void loop() {
12   val = digitalRead(botao); // Lê o botão
13   Serial.println(val);      // Mostra o valor no monitor serial
14
15   if (val == LOW) {        // Se o botão estiver pressionado (puxado
16     digitalWrite(led, HIGH); // Acende o LED
17   } else {
18     digitalWrite(led, LOW);  // Apaga o LED
19   }
20 }
```

## 2.3 Projeto 3 - Buzina com Notas Musicais

### Objetivo do Projeto

Este projeto toca notas musicais com a buzina quando o botão é pressionado.

### Análise do Código

O botão na porta D11 ativa frequências específicas no buzina (D13), usando a função `tone()`.

### Código

Duplo clique no cliques ao lado para abrir: 

```
1 int botao = 11;           // Pino do botão
2 int led = 12;            // Pino do LED
3 int buzzer = 13;        // Pino da buzina
4 int val = 0;            // Estado do botão
5
6 // Frequências das notas musicais (Hz)
7 int notas[] = {262, 294, 330, 349, 392, 440, 494}; // C D E F G A B
8
9 void setup() {
10  pinMode(botao, INPUT);
11  pinMode(led, OUTPUT);
12  pinMode(buzzer, OUTPUT);
13  Serial.begin(9600);
14 }
15
16 void loop() {
17  val = digitalRead(botao);
18  Serial.println(val);
19
20  if (val == LOW) { // Botão pressionado (ligado ao GND)
21    digitalWrite(led, HIGH); // Acende o LED
22
23    // Toca as 7 notas
24    for (int i = 0; i < 7; i++) {
25      tone(buzzer, notas[i]); // Toca nota
26      delay(400);             // Duração da nota
27      noTone(buzzer);        // Para a nota
28      delay(100);           // Pequena pausa entre notas
29    }
30
31    digitalWrite(led, LOW); // Apaga o LED depois da música
32    delay(300);             // Pequeno atraso para não repetir várias
33                             vezes
34 }
```

## 2.4 Projeto 4 - Motor Direito

### Objetivo do Projeto

Este projeto aciona o motor direito do carro.

### Análise do Código

A porta D6 liga ou desliga o motor. A porta D7 define a direção do giro. No comando `analogWrite(motorVel, 255)`, o valor 255 é a intensidade do giro que pode variar de 0 até 255, ou seja, a porta D11 é uma porta PWM (Pulse Width Modulation) onde é possível controlar a voltagem da saída entre 0 e 5 Volts.

### Código

Duplo clique no clipe ao lado para abrir: 

```
1 int motorVel = 6; // Pino de velocidade (PWM)
2 int motorDir = 7; // Pino de direção (sentido)
3
4 void setup() {
5   pinMode(motorVel, OUTPUT);
6   pinMode(motorDir, OUTPUT);
7   Serial.begin(9600); // Inicializa comunicação serial
8 }
9
10 void loop() {
11   // Giro roda direita (ex: direção LOW) - carro vai para trás
12   digitalWrite(motorDir, LOW);
13   analogWrite(motorVel, 255);
14   Serial.println("Carro para trás - roda direita (sentido LOW) por 5
15     segundos.");
16   delay(5000);
17
18   // Parar por 1 segundo
19   analogWrite(motorVel, 0);
20   Serial.println("Roda esquerda parada por 1 segundo.");
21   delay(1000);
22
23   // Giro roda direita (ex: direção HIGH) - carro vai para frente
24   digitalWrite(motorDir, HIGH);
25   analogWrite(motorVel, 255);
26   Serial.println("Carro para frente roda direita (sentido HIGH) por 5
27     segundos.");
28   delay(5000);
29
30   // Parar por 1 segundo
31   analogWrite(motorVel, 0);
32   Serial.println("Roda esquerda parada por 1 segundo.");
33   delay(1000);
34 }
```

## 2.5 Projeto 5 - Motor Esquerdo

### Objetivo do Projeto

Controla o motor esquerdo do carro.

### Análise do Código

A porta D5 aciona o motor. A D4 define se ele gira para frente ou trás.

### Código

Duplo clique no cliques ao lado para abrir: 

```
1 int motorVel = 5; // Pino de velocidade (PWM)
2 int motorDir = 4; // Pino de direção (sentido)
3
4 void setup() {
5   pinMode(motorVel, OUTPUT);
6   pinMode(motorDir, OUTPUT);
7   Serial.begin(9600); // Inicializa comunicação serial
8 }
9
10 void loop() {
11   // Giro para a esquerda (ex: direção LOW) - carro vai para trás
12   digitalWrite(motorDir, LOW);
13   analogWrite(motorVel, 255);
14   Serial.println("Carro para trás - roda esquerda (sentido LOW) por 5
15     segundos.");
16   delay(5000);
17
18   // Parar por 1 segundo
19   analogWrite(motorVel, 0);
20   Serial.println("Roda esquerda parada por 1 segundo.");
21   delay(1000);
22
23   // Giro para a esquerda (ex: direção HIGH) - carro vai para frente
24   digitalWrite(motorDir, HIGH);
25   analogWrite(motorVel, 255);
26   Serial.println("Carro para frente roda esquerda (sentido HIGH) por 5
27     segundos.");
28   delay(5000);
29
30   // Parar por 1 segundo
31   analogWrite(motorVel, 0);
32   Serial.println("Roda esquerda parada por 1 segundo.");
33   delay(1000);
34 }
```

## 2.6 Projeto 6 - Sensor Ultrassônico com LED

### Objetivo do Projeto

Acende um LED quando algum obstáculo é detectado pelo sensor ultrassônico.

### Análise do Código

O pino Trig (D9) envia um pulso ultrassônico que se propaga pelo ar. Quando esse pulso atinge um objeto, ele é refletido de volta ao sensor. O pino Echo (D10) então mede o tempo que o eco levou para retornar. Com esse tempo, o Arduino calcula a distância até o objeto. Se a distância calculada for menor que um valor limite definido, isso significa que há um objeto próximo — então o Arduino acende o LED como forma de alerta visual.

### Código

Duplo clique no cliques ao lado para abrir: 

```
1 const int trigPin = 9; // Pino Trig do sensor
2 const int echoPin = 10; // Pino Echo do sensor
3 const int ledPin = 12; // LED conectado ao D8
4
5
6 long duracao;
7 int distancia;
8
9 void setup() {
10   pinMode(trigPin, OUTPUT);
11   pinMode(echoPin, INPUT);
12   pinMode(ledPin, OUTPUT);
13   Serial.begin(9600);
14 }
15
16 void loop() {
17   // Envia pulso de 10 microssegundos
18   digitalWrite(trigPin, LOW);
19   delayMicroseconds(2);
20   digitalWrite(trigPin, HIGH);
21   delayMicroseconds(10);
22   digitalWrite(trigPin, LOW);
23
24   // Lê tempo de resposta do Echo
25   duracao = pulseIn(echoPin, HIGH);
26
27   // Converte para distância em centímetros
28   distancia = duracao * 0.034 / 2;
29
30   // Exibe no monitor serial
31   Serial.println(distancia);
32
33   // Liga o LED se estiver a menos de 30 cm
34   if (distancia < 30) {
35     digitalWrite(ledPin, HIGH);
36   } else {
37     digitalWrite(ledPin, LOW);
```

```
38 }  
39  
40 delay(500); // Aguarda meio segundo  
41 }
```

## 2.7 Projeto 7 - Sensor seguidor de linha

### Objetivo do Projeto

LED indica se o sensor central detecta a linha.

### Análise do Código

Sensores infravermelhos (A1, A2, A3) leem a cor do solo. Se o sensor central (A2) detectar linha, o LED (D12) acende.

### Código

Duplo clique no cliques ao lado para abrir: 

```
1 int s1, s2, s3;           // Variáveis dos sensores
2 const int ledPin = 12;   // LED no pino D12
3
4 void setup() {
5   pinMode(A1, INPUT); //esquerda
6   pinMode(A2, INPUT); //desprezar este sensor
7   pinMode(A3, INPUT); //direita
8   pinMode(ledPin, OUTPUT);
9
10  Serial.begin(9600);
11  delay(100);
12 }
13
14 void loop() {
15   // Lê os sensores como valores analógicos (0 a 1023)
16   s1 = analogRead(A1); // esquerda
17   s2 = analogRead(A2); // centro que não funciona e deve ser desprezado. (
18     pelo menos no sensor que estou usando)
19   s3 = analogRead(A3); // direita
20
21   // Mostra valores no Serial Monitor
22   Serial.print("S3 = ");
23   Serial.print(s3);
24   Serial.print(" | S2 = ");
25   Serial.print(s2);
26   Serial.print(" | S1 = ");
27   Serial.println(s1);
28
29
30   // Verifica se qualquer sensor está sobre a linha preta (valor BAIXO)
31   if (s1 < 100 || s2 < 100 || s3 < 100) {
32     digitalWrite(ledPin, HIGH); // Liga o LED
33   } else {
34     digitalWrite(ledPin, LOW); // Desliga o LED
35   }
36
37   delay(50); // Aguarda meio segundo
38 }
```

## 2.8 Projeto 8 - Carro Segue linha

### Objetivo do Projeto

O carro segue a linha utilizando dois sensores infravermelhos localizados nas extremidades. O percurso deve ter uma linha é grossa o suficiente para ser mais larga que a distância entre os sensores, garantindo que o carro a detecte corretamente.

### Análise do Código

A1 e A3 identificam a posição da linha. As portas D4-D7 controla a velocidade e direção de rotação dos motores direitos e esquerdos.

### Código

Duplo clique no cliques ao lado para abrir: 

```
1 // Sensores
2 const int sensorE = A1;
3 const int sensorD = A3;
4
5 // Motores
6 const int dirE = 4;
7 const int veE = 5;
8 const int dirD = 7;
9 const int veD = 6;
10
11 void setup() {
12   // Sensores
13   pinMode(sensorE, INPUT);
14   pinMode(sensorD, INPUT);
15
16   // Motores
17   pinMode(dirE, OUTPUT);
18   pinMode(veE, OUTPUT);
19   pinMode(dirD, OUTPUT);
20   pinMode(veD, OUTPUT);
21
22   // Comunicação serial
23   Serial.begin(9600);
24   delay(100);
25 }
26
27 void loop() {
28   int leituraE = digitalRead(sensorE);
29   int leituraD = digitalRead(sensorD);
30
31   // Direção padrão:
32   digitalWrite(dirE, HIGH); // Motor esquerdo para frente
33   digitalWrite(dirD, LOW); // Motor direito CORRIGIDO --> agora gira para
34   frente
35
36   Serial.print("Sensor E = ");
37   Serial.print(leituraE);
38   Serial.print(" | Sensor D = ");
```

```
38 Serial.print(leituraD);
39 Serial.print(" --> Ação: ");
40
41 if (leituraE == 1 && leituraD == 1) {
42     analogWrite(velE, 200);
43     analogWrite(velD, 200);
44     Serial.println("Frente");
45 }
46 else if (leituraE == 0 && leituraD == 1) {
47     analogWrite(velE, 100);
48     analogWrite(velD, 200);
49     Serial.println("Virando para ESQUERDA");
50 }
51 else if (leituraE == 1 && leituraD == 0) {
52     analogWrite(velE, 200);
53     analogWrite(velD, 100);
54     Serial.println("Virando para DIREITA");
55 }
56 else {
57     analogWrite(velE, 0);
58     analogWrite(velD, 0);
59     Serial.println("PARADO - Linha perdida");
60 }
61
62 delay(100);
63 }
```

## 2.9 Projeto 9 - Carro Segue linha com Parada por obstáculo

### Objetivo do Projeto

O carro segue a linha (projeto anterior) e para quando encontra um obstáculo.

### Análise do Código

Adiciona um sensor ultrassônico ao seguidor de linha. Se detectar obstáculo, os motores param.

### Código

Duplo clique no cliques ao lado para abrir: 

```
1 // Sensores de linha
2 const int sensorE = A1;
3 const int sensorD = A3;
4
5 // Motores
6 const int dirE = 4;
7 const int veLE = 5;
8 const int dirD = 7;
9 const int veLD = 6;
10
11 // Sensor Ultrassônico
12 const int TRIG = 9;
13 const int ECHO = 10;
14
15 // Buzina
16 const int buzina = 13;
17
18 void setup() {
19   // Sensores de linha
20   pinMode(sensorE, INPUT);
21   pinMode(sensorD, INPUT);
22
23   // Motores
24   pinMode(dirE, OUTPUT);
25   pinMode(veLE, OUTPUT);
26   pinMode(dirD, OUTPUT);
27   pinMode(veLD, OUTPUT);
28
29   // Ultrassônico
30   pinMode(TRIG, OUTPUT);
31   pinMode(ECHO, INPUT);
32
33   // Buzina
34   pinMode(buzina, OUTPUT);
35
36   // Serial
37   Serial.begin(9600);
38   delay(100);
```

```

39 }
40
41 void loop() {
42     // ----- VERIFICA OBSTÁCULO -----
43     long distancia = medirDistancia();
44
45     if (distancia > 0 && distancia < 15) {
46         // Obstáculo detectado --> parar e buzinar
47         pararMotores();
48         buzinar();
49         Serial.print("OBSTÁCULO a ");
50         Serial.print(distancia);
51         Serial.println(" cm --> PARADO E BUZINANDO");
52         return; // Sai do loop aqui
53     }
54
55     // ----- SEGUIR LINHA NORMALMENTE -----
56     int leituraE = digitalRead(sensorE);
57     int leituraD = digitalRead(sensorD);
58
59     digitalWrite(dirE, HIGH);
60     digitalWrite(dirD, LOW); // Corrigido: motor direito gira para frente
61
62     Serial.print("Sensor E = ");
63     Serial.print(leituraE);
64     Serial.print(" | Sensor D = ");
65     Serial.print(leituraD);
66     Serial.print(" --> Ação: ");
67
68     if (leituraE == 1 && leituraD == 1) {
69         analogWrite(velE, 200);
70         analogWrite(velD, 200);
71         Serial.println("Frente");
72     }
73     else if (leituraE == 0 && leituraD == 1) {
74         analogWrite(velE, 100);
75         analogWrite(velD, 200);
76         Serial.println("Virando para ESQUERDA");
77     }
78     else if (leituraE == 1 && leituraD == 0) {
79         analogWrite(velE, 200);
80         analogWrite(velD, 100);
81         Serial.println("Virando para DIREITA");
82     }
83     else {
84         pararMotores();
85         Serial.println("PARADO - Linha perdida");
86     }
87
88     delay(100);
89 }
90
91 // ----- FUNÇÕES AUXILIARES -----
92
93 // Mede distância com o sensor ultrassônico
94 long medirDistancia() {
95     digitalWrite(TRIG, LOW);
96     delayMicroseconds(2);

```

```

97  digitalWrite(TRIG, HIGH);
98  delayMicroseconds(10);
99  digitalWrite(TRIG, LOW);
100
101  long duracao = pulseIn(ECHO, HIGH, 20000); // timeout: 20ms = 340 cm
102  long distancia = duracao * 0.034 / 2;
103  return distancia;
104 }
105
106 // Para os motores
107 void pararMotores() {
108     analogWrite(velE, 0);
109     analogWrite(velD, 0);
110 }
111
112 // Buzina intermitente
113 // Buzina intermitente usando tom audível
114 void buzinar() {
115     for (int i = 0; i < 5; i++) {
116         tone(buzina, 1000); // Frequência de 1000 Hz (agudo e forte)
117         delay(300);
118         noTone(buzina);
119         delay(200);
120     }
121 }
122 }

```



# Conclusões

Espero que tenha ajudado em sua jornada com o kit. Se tiver alguma dúvida ou sugestão fique a vontade para entrar em contato no meu Whats App (14)99753 – 6155.