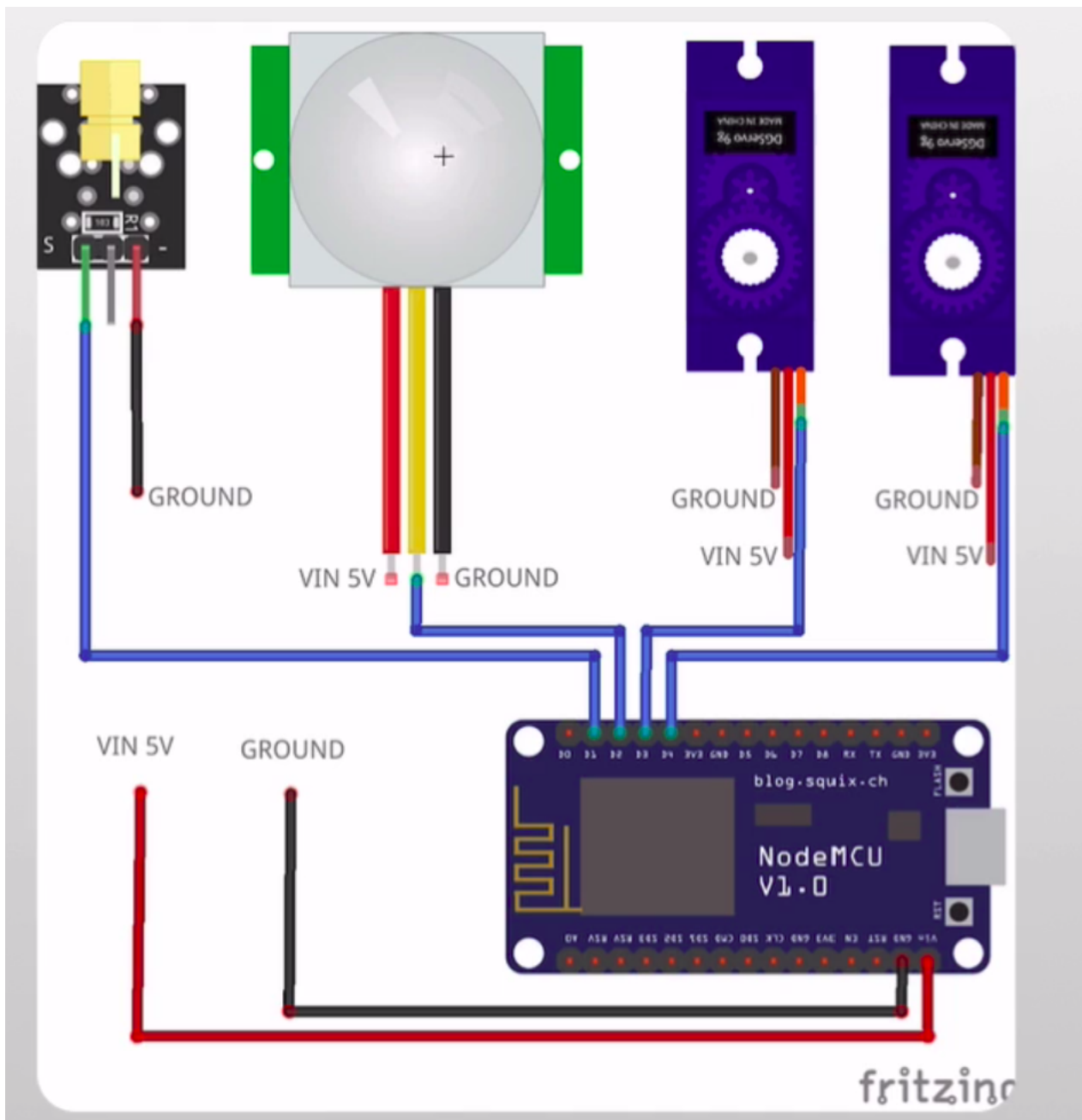


Inicializando o Laser

Transcrição

Faremos a adição do *laser* em nosso projeto, mas antes, temos uma observação. Apesar de usarmos um *laser* que vem embutido em sua própria placa, é recomendado que o mesmo seja substituído por um que venha apenas com os fios no produto final. Isso evitará que ocorram problemas de mau contato. Vejamos nosso diagrama.



Primeiro devemos conectar o ponto negativo no *GROUND* e depois, o ponto positivo o pino *D1*. O *laser* funcionará como um *led*, porém, ele será conectado em um pino, em vez de no conector de 5 volts porque nós teremos o controle quando o laser acende ou apaga via *NodeMcu*.

Antes de continuarmos, vamos comentar o código que imprime as coordenadas dos servos motores, além de remover a função de posicionamento de dentro do *loop*, já que eram só testes.

```
#include <Servo.h>

#define POS_INICIAL 90
#define X_MINIMO 30
#define Y_MINIMO 50
#define X_INTERVALO 12
#define Y_INTERVALO 4

Servo servoX;
Servo servoY;

void setup() {
    servoX.attach(D3);
    servoY.attach(D4);
    servoX.write(POS_INICIAL);
    servoY.write(POS_INICIAL);
    // Serial.begin(9600);
}

void loop() {

}

// FUNÇÕES AUXILIARES

void posiciona_motores() {
    int posicaoX = (random(0, (X_INTERVALO)) * 10 + X_MINIMO);
    int posicaoY = (random(0, (Y_INTERVALO)) * 10 + Y_MINIMO);
    servoX.write(posicaoX);
    servoY.write(posicaoY);
    // Serial.print(posicaoX);
    // Serial.print(" , ");
    // Serial.println(posicaoY);
    delay(2000);
}
```

Em seguida, inicializaremos nosso *laser*, primeiro criando uma variável que identifique seu pino.

```
#define LASER D1
```

O próximo passo será configurar o pino em si como saída de dados e deixando o mesmo desligado. No bloco `setup` teremos:

```
pinMode(LASER, OUTPUT);
digitalWrite(LASER, LOW);
```

Por último, antes de testar, criaremos nossas funções auxiliares que ligam e desligam o *laser*:

```
void liga_laser() {
    digitalWrite(LASER, HIGH);
}
```

```
void desliga_laser() {  
    digitalWrite(LASER, LOW);  
}
```

Dentro do bloco `loop` , vamos intercalar a chamada das duas funções com a função `delay()` para que possamos ver claramente o *laser* funcionando.

Após ligarmos, aguardaremos dois segundos, em seguida, iremos desligar e esperar dois segundos antes de religá-lo.

```
void loop() {  
    liga_laser();  
    delay(2000);  
    desliga_laser();  
    delay(2000);  
}
```

Nosso código final se encontra assim:

```
#include <Servo.h>  
  
#define POS_INICIAL 90  
#define X_MINIMO 30  
#define Y_MINIMO 50  
#define X_INTERVALO 12  
#define Y_INTERVALO 4  
#define LASER D1  
  
Servo servoX;  
Servo servoY;  
  
void setup() {  
    servoX.attach(D3);  
    servoY.attach(D4);  
    servoX.write(POS_INICIAL);  
    servoY.write(POS_INICIAL);  
  
    pinMode(LASER, OUTPUT);  
    digitalWrite(LASER, LOW);  
    // Serial.begin(9600);  
}  
  
void loop() {  
    liga_laser();  
    delay(2000);  
    desliga_laser();  
    delay(2000);  
}  
  
// FUNÇÕES AUXILIARES  
  
void posiciona_motores() {  
    int posicaoX = (random(0, (X_INTERVALO)) * 10 + X_MINIMO);  
    int posicaoY = (random(0, (Y_INTERVALO)) * 10 + Y_MINIMO);  
    servoX.write(posicaoX);
```

```
servoY.write(posicaoY);  
// Serial.print(posicaoX);  
// Serial.print("  ", " ");  
// Serial.println(posicaoY);  
delay(2000);  
}  
  
void liga_laser() {  
  digitalWrite(LASER, HIGH);  
}  
  
void desliga_laser() {  
  digitalWrite(LASER, LOW);  
}
```

Vamos testar? Lembre-se de fazer as conexões físicas do dispositivo.