

## Projeto Elevador de Carga

Nesse projeto, a lógica que eu usei foi colocar várias verificações para saber se o motor deve subir ou descer, ou apenas permanecer onde está (ex: caso a pessoa aperte o botão do andar que ela já está ele não pode fazer nada), então eu fiz todas as verificações. Outra coisa que eu fiz foi calcular o tempo que o motor deve levar para subir ou descer levando em consideração que de um andar para outro leva 1 segundo, então eu calculo “1000 \* andar – andarAtual” para subir ou “1000 \* andarAtual – andar” para descer, e coloco o resultado no delay e enquanto está no delay eu chamo o método para continuar lendo os valores dos botões e ele saber o próximo andar para ir. Eu usei um método que mostra o andar que está, mas era apenas um debug pra eu não ficar perdido na hora de programar a lógica e resolvi deixar.

Código:

```
const int botaoT = 8;
const int botao1 = 9;
const int botao2 = 10;
const int botao3 = 11;
const int botao4 = 12;
const int controlePonteH = 7;
const int motorSubida = 6;
const int motorDescida = 5;
int ligadoT = 0;
int ligado1 = 0;
int ligado2 = 0;
int ligado3 = 0;
int ligado4 = 0;
char direcao = 'S';
int andarAtual = 0;
int tempo = 0;
int AndarApertado;
```

```
void setup()
{
  pinMode(botaoT, INPUT_PULLUP);
  pinMode(botao1, INPUT_PULLUP);
  pinMode(botao2, INPUT_PULLUP);
  pinMode(botao3, INPUT_PULLUP);
  pinMode(botao4, INPUT_PULLUP);
  pinMode(controlePonteH, OUTPUT);
  pinMode(motorSubida, OUTPUT);
  pinMode(motorDescida, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop()
{
  digitalWrite(controlePonteH, HIGH);
  VerificaBotao();
  VerificaAndar();
}
```

```
void VerificaAndar(){
  // Verifica o andar pressionado enquanto descia ou subia
  if(AndarApertado == 0){
    ligadoT = 0;
  }
  else if(AndarApertado == 1){
    ligado1 = 0;
```

```
}  
else if(AndarApertado == 2){  
    ligado2 = 0;  
}  
else if(AndarApertado == 3){  
    ligado3 = 0;  
}  
else if(AndarApertado == 4){  
    ligado4 = 0;  
}  
// Seta a direção correta  
if(ligado1 == 0 && direcao == 'S' && andarAtual > 1 )  
{  
    direcao = 'D';  
}  
else if(ligado1 == 0 && direcao == 'D' && andarAtual < 1 )  
{  
    direcao = 'S';  
}  
else if(ligado2 == 0 && direcao == 'S' && andarAtual > 2 )  
{  
    direcao = 'D';  
}  
else if(ligado2 == 0 && direcao == 'D' && andarAtual < 2 )  
{  
    direcao = 'S';  
}
```

```
else if(ligado3 == 0 && direcao == 'S' && andarAtual > 3 )
{
    direcao = 'D';
}
else if(ligado3 == 0 && direcao == 'D' && andarAtual < 3 )
{
    direcao = 'S';
}
else if(ligado4 == 0 && direcao == 'S' )
{
    direcao = 'D';
}
else if(ligado4 == 0 && direcao == 'D' && andarAtual < 4 )
{
    direcao = 'S';
}
else if(ligadoT == 0)
{
    direcao = 'D';
}
```

//Verificação para ver se estou subindo

```
if(ligado1 == 0 && direcao == 'S' && andarAtual != 1)
{
    SubirMotor(1);
}
if(ligado2 == 0 && direcao == 'S' && andarAtual != 2)
```

```
{
    SubirMotor(2);
}
if(ligado3 == 0 && direcao == 'S' && andarAtual != 3)
{
    SubirMotor(3);
}
if(ligado4 == 0 && direcao == 'S' && andarAtual != 4)
{
    SubirMotor(4);
    direcao = 'D';
}
//Verificação para ver se estou descendo
if(ligado3 == 0 && direcao == 'D' && andarAtual != 3)
{
    DescerMotor(3);
}
if(ligado2 == 0 && direcao == 'D' && andarAtual != 2)
{
    DescerMotor(2);
}
if(ligado1 == 0 && direcao == 'D' && andarAtual != 1)
{
    DescerMotor(1);
}
if(ligadoT == 0 && direcao == 'D' && andarAtual != 0)
{
```

```

        DescerMotor(0);
        direcao = 'S';
    }

}

void VerificaBotao(){
    ligadoT = digitalRead(botaoT);
    ligado1 = digitalRead(botao1);
    ligado2 = digitalRead(botao2);
    ligado3 = digitalRead(botao3);
    ligado4 = digitalRead(botao4);

    if(ligadoT == 0){
        AndarApertado = 0;
    }
    else if(ligado1 == 0){
        AndarApertado = 1;
    }
    else if(ligado2 == 0){
        AndarApertado = 2;
    }
    else if(ligado3 == 0){
        AndarApertado = 3;
    }
    else if(ligado4 == 0){
        AndarApertado = 4;
    }
}

```

```
}
```

```
void SubirMotor(int andar){  
    digitalWrite(motorSubida, HIGH);  
    digitalWrite(motorDescida, LOW);  
    tempo = 1000 * (andar - andarAtual);  
    while(tempo >= 0){  
        tempo = tempo - 15;  
        VerificaBotao();  
        delay(15);  
    }  
    digitalWrite(motorSubida, LOW);  
    digitalWrite(motorDescida, LOW);  
    andarAtual = andar;  
    MostrarAndar(andar);  
    delay(1000);  
}
```

```
void DescerMotor(int andar){  
    digitalWrite(motorSubida, LOW);  
    digitalWrite(motorDescida, HIGH);  
    tempo = 1000 * (andarAtual - andar);  
    while(tempo >= 0){  
        tempo = tempo - 15;  
        VerificaBotao();  
        delay(15);  
    }  
}
```

```
digitalWrite(motorSubida, LOW);  
digitalWrite(motorDescida, LOW);  
andarAtual = andar;  
MostrarAndar(andar);  
delay(1000);  
}
```

```
void MostrarAndar(int andar){  
    switch (andar)  
    {  
        case 1:  
            Serial.println("Primeiro Andar");  
            break;  
        case 2:  
            Serial.println("Segundo Andar");  
            break;  
        case 3:  
            Serial.println("Terceiro Andar");  
            break;  
        case 4:  
            Serial.println("Quarto Andar");  
            break;  
        default:  
            Serial.println("Terreo");  
            break;  
    }  
}
```



