

Sistemas Embarcados e Robótica

Entrega 1

Projeto: Making Makers

Objetivo do projeto

Making Makers tem como objetivo automatizar a sala Maker da FECAP, de forma que se torne mais segura, mais econômica e ágil.

Vamos automatizar as bancadas de utilização de equipamentos eletrônicos, como a bancada de ferro de solda, a bancada onde se encontra o esmeril, entre outros. O painel de ferramentas terá um pequeno sistema que reconhece quando a ferramenta está no seu lugar ou não, sendo indicada por leds que se acenderão vermelho ao tirar, e verde ao guardar a ferramenta novamente. Teremos também a automação dos aparelhos de Ar condicionado e televisão, para maior controle e gerenciamento de energia.

Sensores utilizados

O sistema faz uso de dois tipos principais de sensores:

- **Encoder rotativo:** utilizado para monitorar a posição do motor, permitindo calcular a quantidade de voltas realizadas a partir da contagem de pulsos. Ele foi conectado às portas digitais D2 e D3 do Arduino, garantindo a leitura precisa do movimento.
- **Sensor de luminosidade (LDR):** responsável por medir a intensidade da luz no ambiente. Esse valor é comparado com um limiar pré-definido e, a partir dele, são acionados dois LEDs indicadores (vermelho ou verde), representando respectivamente a presença de um equipamento ou a falta dele.

Lógica de funcionamento

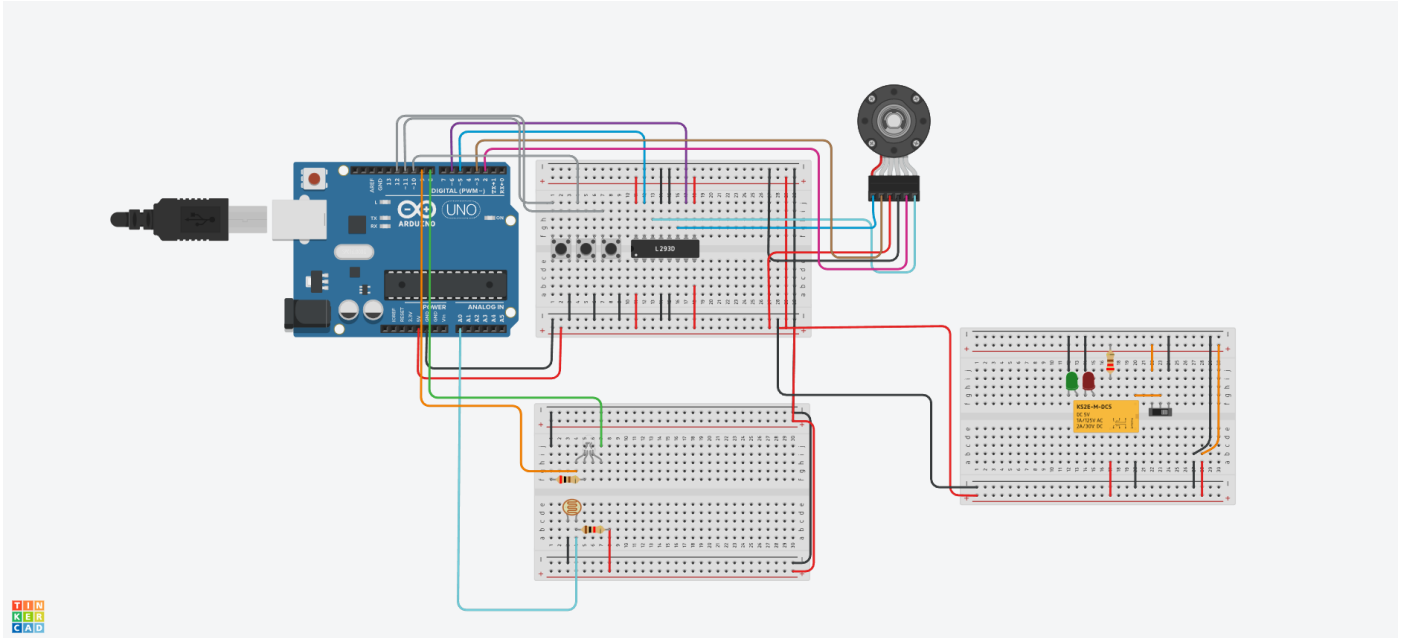
O sistema foi projetado para controlar o giro de um motor em duas direções, com monitoramento do número de voltas. Para isso, foram utilizados três botões: o primeiro aciona o motor no sentido horário por três voltas, o segundo executa o mesmo movimento em sentido anti-horário e o terceiro funciona como parada imediata, interrompendo o motor a qualquer momento.

O encoder garante que o motor seja desligado automaticamente quando atingir o número de voltas programado, mantendo precisão no controle. Já o sensor de luz complementa o projeto com uma indicação visual: quando a luminosidade ultrapassa o valor definido como limite, o LED vermelho acende, e em situações de baixa luz, o LED verde é ativado. Isso simula a ação de guardar ou retirar um equipamento.

Ao lado está uma lógica simples de ligar e desligar um relê, para simular ligar e desligar uma régua de tomadas pelo DBX-MIO.

Código INICIAL + LINK + Imagem do projeto BETA

Link: [Versão BETA: Making Makers](#)



```
#define motorA 5
```

```
#define motorB 6
```

```
// Encoder pins
```

```
int encoderPin1 = 2;
```

```
int encoderPin2 = 3;
```

```
volatile int lastEncoded = 0;
```

```
volatile long encoderValue = 0;
```

```
// LED + sensor
```

```
int sensorLuz = A0;
```

```
int ledVermelho = 8;
```

```
int ledVerde = 9;
```

```
int limiar = 500;
```

```
// Buttons

int btn1 = 10;

int btn2 = 11;

int btn3 = 12;


// Controle do motor

long targetPosition = 0;

bool motorRunning = false;

int sentido = 0;


void setup() {

    pinMode(ledVermelho, OUTPUT);

    pinMode(ledVerde, OUTPUT);

    pinMode(btn1, INPUT_PULLUP);

    pinMode(btn2, INPUT_PULLUP);

    pinMode(btn3, INPUT_PULLUP);


    Serial.begin(9600);


    pinMode(encoderPin1, INPUT_PULLUP);

    pinMode(encoderPin2, INPUT_PULLUP);


    attachInterrupt(digitalPinToInterrupt(encoderPin1), updateEncoder, CHANGE);

    attachInterrupt(digitalPinToInterrupt(encoderPin2), updateEncoder, CHANGE);


    pinMode(motorA, OUTPUT);

    pinMode(motorB, OUTPUT);

}


void loop() {

    Serial.print("A: ");
```

```
Serial.print(digitalRead(encoderPin1));

Serial.print(" | B: ");

Serial.println(digitalRead(encoderPin2));

delay(200);


// BOTÃO 1 → 3 voltas horário

if (digitalRead(btn1) == LOW && !motorRunning) {

    targetPosition = encoderValue + 3000; // ajustar pulsos por volta

    motorRunning = true;

    sentido = 1;

    giraHorario();

}


// BOTÃO 2 → 3 voltas anti-horário

if (digitalRead(btn2) == LOW && !motorRunning) {

    targetPosition = encoderValue - 3000;

    motorRunning = true;

    sentido = -1;

    giraAntiHorario();

}


// BOTÃO 3 → parada imediata

if (digitalRead(btn3) == LOW && motorRunning) {

    pararMotor();

    motorRunning = false;

}


// Controle do motor: parar automaticamente ao atingir alvo

if (motorRunning) {

    if ((sentido == 1 && encoderValue >= targetPosition) ||

        (sentido == -1 && encoderValue <= targetPosition)) {
```

```
    pararMotor();

    motorRunning = false;

}

}

// Controle da Luz

int valorLuz = analogRead(sensorLuz);

if (valorLuz > limiar) {

    digitalWrite(ledVermelho, HIGH);

    digitalWrite(ledVerde, LOW);

} else {

    digitalWrite(ledVermelho, LOW);

    digitalWrite(ledVerde, HIGH);

}

delay(20);

}

void giraHorario() {

    analogWrite(motorA, 127);

    digitalWrite(motorB, LOW);

}

void giraAntiHorario() {

    analogWrite(motorB, 127);

    digitalWrite(motorA, LOW);

}

void pararMotor() {

    digitalWrite(motorA, LOW);

    digitalWrite(motorB, LOW);
```

```
}
```

```
void updateEncoder() {
```

```
    int MSB = digitalRead(encoderPin1);
```

```
    int LSB = digitalRead(encoderPin2);
```

```
    int encoded = (MSB << 1) | LSB;
```

```
    int sum = (lastEncoded << 2) | encoded;
```

```
    if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011) encoderValue++;
```

```
    if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000) encoderValue--;
```

```
    lastEncoded = encoded;
```

```
}
```