Instituto Federal Catarinense (Campus Blumenau)

Professor: Carlos Augusto Machado Monteiro

Matéria: Sistemas Embarcados

Nomes:Gabrielli Danker, Gabriel Rodrigues de Carvalho e Lucas Taveira de Sena

Turma: BCC 2025.1

Data de entrega: 09 de Abril de 2025

Relatório- Lab 5

Na atividade 1 montamos o circuito do motor com a ponte H e acionando o botão 1 (gira para a direita) que está ligada na porta 3, botão 2 (gira para a esquerda) que está ligada na porta 2 e o botão 3 desliga o motor.

Código:

void setup() {

pinMode(3, INPUT\_PULLUP); // Direita

pinMode(2, INPUT\_PULLUP); // Esquerda

pinMode(4, INPUT\_PULLUP); // Desligar

pinMode(9, OUTPUT); // IN2

pinMode(10, OUTPUT); // IN1

pinMode(11, OUTPUT); // GN

}

void loop() {

if (digitalRead(3) == LOW) {

digitalWrite(9, HIGH);

digitalWrite(10, LOW);

analogWrite(11, 255);

}

else if (digitalRead(2) == LOW) {

digitalWrite(9, LOW);

digitalWrite(10, HIGH);

analogWrite(11, 255);

}

else if (digitalRead(4) == LOW) {

digitalWrite(9, LOW);

digitalWrite(10, LOW);

analogWrite(11, 0);

}

}

Na atividade 2 utilizamos o mesmo circuito com o motor, fizemos um programa para aumentar a velocidade do motor (mínimo 10 velocidades) ao pressionar o botão 1, diminui ao pressionar o botão 2 e ao pressionar o botão 3 o motor para.

Código:

#define Delay 250 // milissegundos

int velocidade = 0; // Velocidade inicial

bool motorLigado = false; // Variável para verificar o estado do motor

unsigned long lastPressTime = 0; // Armazena o último momento de pressionamento de botão

void setup() {

pinMode(3, INPUT\_PULLUP); // Aumenta a velocidade

pinMode(2, INPUT\_PULLUP); // Diminui a velocidade

pinMode(4, INPUT\_PULLUP); // Desliga

pinMode(9, OUTPUT); // IN2 (direção)

pinMode(10, OUTPUT); // IN1 (direção)

pinMode(11, OUTPUT); // PWM para velocidade

}

void loop() {

unsigned long currentMillis = millis();

// Evita múltiplos acionamentos rápidos com debounce

if (currentMillis - lastPressTime > Delay) {

// Aumenta a velocidade

if (digitalRead(3) == LOW && motorLigado) {

velocidade = min(255, velocidade + 25);

lastPressTime = currentMillis; // Atualiza o tempo do último pressionamento

}

// Diminui a velocidade

if (digitalRead(2) == LOW && motorLigado) {

velocidade = max(0, velocidade - 25);

lastPressTime = currentMillis; // Atualiza o tempo do último pressionamento

}

// Desliga o motor

if (digitalRead(4) == LOW) {

motorLigado = false;

digitalWrite(9, LOW);

digitalWrite(10, LOW);

analogWrite(11, 0); // Desliga o motor

velocidade = 0; // Reseta a velocidade

lastPressTime = currentMillis; // Atualiza o tempo do último pressionamento

}

// Liga o motor novamente

if (!motorLigado && (digitalRead(3) == LOW || digitalRead(2) == LOW)) {

motorLigado = true;

lastPressTime = currentMillis; // Atualiza o tempo do último pressionamento

}

// Controla o motor com a direção mantida

if (motorLigado) {

// Define a direção (apenas uma vez, não dependendo da velocidade)

digitalWrite(9, HIGH); // IN2

digitalWrite(10, LOW); // IN1

analogWrite(11, velocidade); // Aplica a velocidade ajustada

}

}

}

Circuito esquemático:

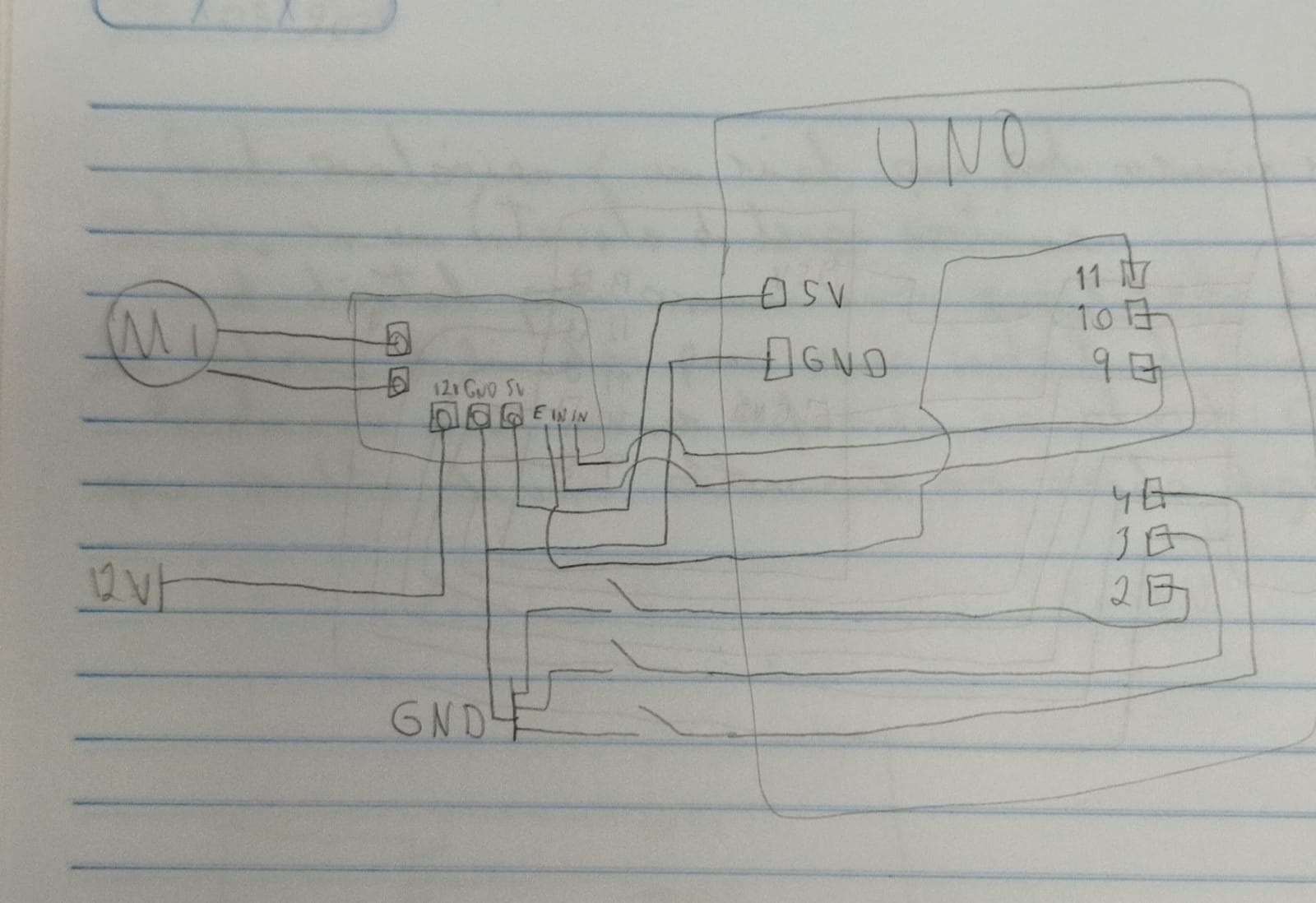


Foto da montagem:

