



Relatório de Atividade

Braço Robótico

LORRAN PIRES VENETILLO DUTRA, MICHAEL CASSEMIRO OLIVEIRA, VÍTOR
GABRIEL REIS CAITITÉ, WILLIAN BRAGA DA SILVA

Petianos responsáveis pelo projeto, em ordem alfabética

Resumo

O projeto desenvolvido é inspirado no braço robótico FoamArmDS desenvolvido pela EasyDS. O Braço Robótico possuiu sua estrutura construída com spumaPaper (material facilmente modelado), além de servo motores, placa de circuito impresso e um microcontrolador Arduino. É um projeto que tem como objetivo transmitir conceitos básicos de robótica.

CONTEÚDO

I	Introdução	1
II	Materiais e Métodos	2
III	Resultados	12
IV	Discussão	13

I. INTRODUÇÃO

O Braço Robótico foi um projeto desenvolvido na disciplina de Laboratório de Circuitos Elétricos 1. Ao final da disciplina os alunos envolvidos com o projeto decidiram que o projeto seria doado ao grupo PETEE. No grupo o Braço Robótico é utilizado principalmente nas atividades de extensão, como a Engenharia na Escola (ENE) e Mostra de Profissões da UFMG. Dessa forma, por meio do protótipo é possível explicar, ao público interessado, conceitos básicos de robótica e programação.

II. MATERIAIS E MÉTODOS

Materiais para a construção do Braço Robótico:

- 6x Micro Servo 9g
- 4x Chave Tátil 12x12x4,3mm
- 6x Potenciometro Linear Mini 1K
- 5x Led Vermelho Difuso 3mm
- 5x Led Amarelo Difuso 3mm
- 5x Led Verde Difuso 3mm
- 2x Placa CI De Fenolite Virgem 10x20
- Jack J4 DC-002
- Jack J4 DC-005 P/ Placa 2,1mm X 5,5mm
- Jack J4 DC-005 P/ Placa 2,5mm X 5,5mm
- 4x Chave Tátil 12x12x4,3mm 4 Terminais
- Spumapaper
- 2x Chave Tátil 12x12x12mm 4 Terminais
- Knob AD-110 Azul Para Eixo Estriado
- Knob AD-110 Verde Para Eixo Estriado
- Knob AD-110 Branco Para Eixo Estriado
- Arduino Uno
- suporte para 4 pilhas
- pilhas de 1.2V cada
- jumpers macho/fêmea

A primeira etapa da construção do Braço Robótico foi a realização de testes no protoboard. Para isso foi realizada a montagem da Figura 1 e o código foi carregado no Arduino Uno.

II MATERIAIS E MÉTODOS

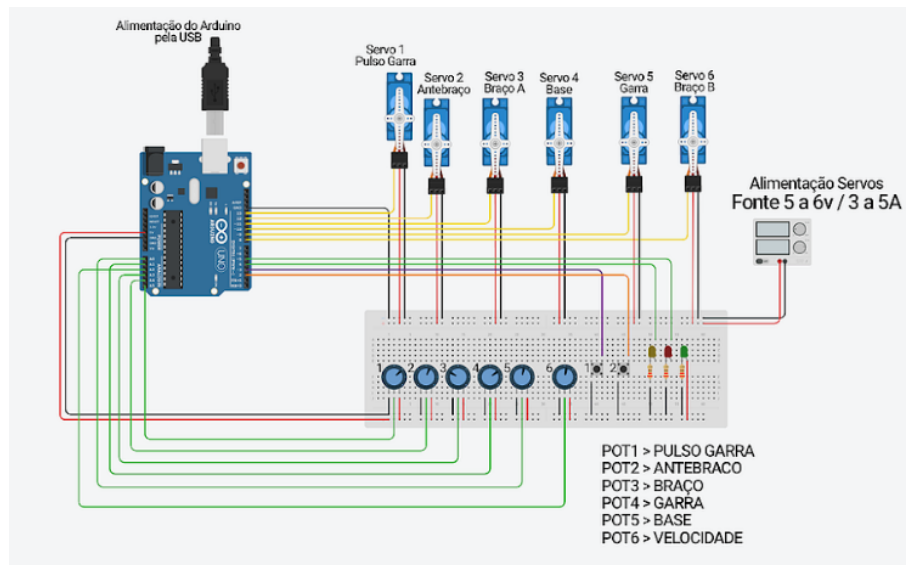


Figura 1: Diagrama de montagem no protoboard

Código

```
\begin{ArduinoSketchBox}{title}

#include <VarSpeedServo.h>
#include <EEPROM.h>
#define espacoMemoria 199
#define tempoPausaEntreMovimentos 500
//configura o tempo de pausa entre cada movimento

#define pinServo1 13
#define pinServo2 12
#define pinServo3 11
#define pinServo4 10
#define pinServo5 9
#define pinServo6 8

#define pinBotao1 3
#define pinBotao2 2

#define pinLedA 5
#define pinLedB 4

#define pinPot1 A5
#define pinPot2 A4
#define pinPot3 A3
#define pinPot4 A0
#define pinPot5 A1

\end{ArduinoSketchBox}
```

```
#define pinPot6      A2

VarSpeedServo servo1;
VarSpeedServo servo2;
VarSpeedServo servo3;
VarSpeedServo servo4;
VarSpeedServo servo5;
VarSpeedServo servo6;

byte pinBotao1Modo();
bool pinBotao2Retencao();
void pinLedAPisca(bool reset = false);
void setMemoria(byte posicao, byte servo, byte valor);
byte readMemoria(byte posicao, byte servo);

int ultMemoria;

void setup() {
    servo1.attach(pinServo1);
    servo2.attach(pinServo2);
    servo3.attach(pinServo3);
    servo4.attach(pinServo4);
    servo5.attach(pinServo5);
    servo6.attach(pinServo6);

    pinMode(pinLedA, OUTPUT);
    pinMode(pinLedB, OUTPUT);

    pinMode(pinBotao1, INPUT_PULLUP);
    pinMode(pinBotao2, INPUT_PULLUP);

    ultMemoria = EEPROM.read(0);
}

void loop() {
    static byte modo = 0;
    static byte modoAnt;
    static byte movimento = 0;
    static byte posMemoria = 0;
    static unsigned long delayPausa;

    //Modo Normal
    if (modo == 0) {
        digitalWrite(pinLedA, LOW);
```

II MATERIAIS E MÉTODOS

```
if (pinBotao2Retencao()) {
    digitalWrite(pinLedB, HIGH);

    //executa um movimento
    if (movimento == 0) {
        byte velocidade = map(analogRead(pinPot6), 0, 1023, 0, 255);
        servo1.slowmove(readMemoria(posMemoria,0), velocidade);
        servo2.slowmove(readMemoria(posMemoria,1), velocidade);
        servo3.slowmove(readMemoria(posMemoria,2), velocidade);
        servo4.slowmove(readMemoria(posMemoria,3), velocidade);
        servo5.slowmove(readMemoria(posMemoria,4), velocidade);
        //servo6.slowmove(readMemoria(posMemoria,2), velocidade);
        servo6.slowmove( map(readMemoria(posMemoria,2), 0, 179, 179, 0), velocidade);
        movimento = 1;
    }

    //aguarda término de um movimento para selecionar o próximo movimento
    if (movimento == 1) {
        if ( (servo1.read() == readMemoria(posMemoria,0)) &&
            (servo2.read() == readMemoria(posMemoria,1)) &&
            (servo3.read() == readMemoria(posMemoria,2)) &&
            (servo4.read() == readMemoria(posMemoria,3)) &&
            (servo5.read() == readMemoria(posMemoria,4)) ) {

            posMemoria++;
            if (posMemoria > ultMemoria) { posMemoria = 0; }

            delayPausa = millis();
            movimento = 2;
        }
    }

    //pausa entre movimentos
    if (movimento == 2) {
        if ((millis() - delayPausa) > tempoPausaEntreMovimentos) {
            movimento = 0;
        }
    }
} else {
    digitalWrite(pinLedB, LOW);

    if (pinBotao1Modo() == 2) {
```

```
        modo = 1;
    }
}

//Modo Gravação
if (modo == 1) {

    if (modoAnt == 0) {
        pinLedAPisca(true);
        ultMemoria = -1;
        EEPROM.write(0, ultMemoria);
    }

    digitalWrite(pinLedB, LOW);
    pinLedAPisca();

    if (pinBotao1Modo() == 2) {
        modo = 0;
    }

    servo1.write( map(analogRead(pinPot1),0,1023,0,179) );
    servo2.write( map(analogRead(pinPot2),0,1023,0,179) );
    servo3.write( map(analogRead(pinPot3),0,1023,0,179) );
    servo4.write( map(analogRead(pinPot4),0,1023,0,179) );
    servo5.write( map(analogRead(pinPot5),0,1023,0,179) );
    //servo6.write( map(analogRead(pinPot3),0,1023,0,179) );
    servo6.write( map(analogRead(pinPot3),0,1023,179,0) ); //Servo
    //sincronizado inversamente ao Servo3

    if (pinBotao2Apertado()) {
        ultMemoria++;
        EEPROM.write(0, ultMemoria);
        setMemoria(ultMemoria, 0, map(analogRead(pinPot1),0,1023,0,179));
        setMemoria(ultMemoria, 1, map(analogRead(pinPot2),0,1023,0,179));
        setMemoria(ultMemoria, 2, map(analogRead(pinPot3),0,1023,0,179));
        setMemoria(ultMemoria, 3, map(analogRead(pinPot4),0,1023,0,179));
        setMemoria(ultMemoria, 4, map(analogRead(pinPot5),0,1023,0,179));

        digitalWrite(pinLedB, HIGH);
        delay(250);
        digitalWrite(pinLedB, LOW);

        if ( ultMemoria == (espacoMemoria - 1)) {
```

II MATERIAIS E MÉTODOS

```
        modo = 0;
    }
}

modoAnt = modo;
}

byte pinBotao1Modo() {
#define tempoDebounce 40 //(tempo para eliminar o efeito Bounce
//EM MILISEGUNDOS)
#define tempoBotao 1500

bool estadoBotao;
static bool estadoBotaoAnt;
static byte estadoRet = 0;
static unsigned long delayBotao = 0;
static byte enviado = 0;

if ( (millis() - delayBotao) > tempoDebounce ) {
    estadoBotao = digitalRead(pinBotao1);
    if (estadoRet == 0) {
        if ( !estadoBotao && (estadoBotao != estadoBotaoAnt) ) {
            estadoRet = 1;
            delayBotao = millis();
        }
    }

    if (estadoRet == 1) {
        if ( estadoBotao && (estadoBotao != estadoBotaoAnt) ) {
            estadoRet = 0;
            delayBotao = millis();
        }
    }
    estadoBotaoAnt = estadoBotao;
}

if (estadoRet == 1) {
    if ((millis() - delayBotao) > tempoBotao) {
        estadoRet = 2;
        delayBotao = millis();
    }
}
}
```

```
if (estadoRet == 2) {
    enviado++;

    if (enviado >= 2) {
        estadoRet = 0;
        delayBotao = millis();
        enviado = 0;
    }
}

return estadoRet;
}

bool pinBotao2Retencao() {
#define tempoDebounce 40 //(tempo para eliminar o efeito Bounce EM
//MILISEGUNDOS)

    bool estadoBotao;
    static bool estadoBotaoAnt;
    static bool estadoRet = false;
    static unsigned long delayBotao = 0;

    if ( (millis() - delayBotao) > tempoDebounce ) {
        estadoBotao = digitalRead(pinBotao2);
        if ( !estadoBotao && (estadoBotao != estadoBotaoAnt) ) {
            estadoRet = !estadoRet;
            delayBotao = millis();
        }
        if ( estadoBotao && (estadoBotao != estadoBotaoAnt) ) {
            delayBotao = millis();
        }
        estadoBotaoAnt = estadoBotao;
    }

    return estadoRet;
}

bool pinBotao2Apertado() {
#define tempoDebounce 40 //(tempo para eliminar o efeito Bounce EM MILISEGUNDOS)

    bool estadoBotao;
    static bool estadoBotaoAnt;
    static bool estadoRet;
    static unsigned long delayBotao = 0;
```


II MATERIAIS E MÉTODOS

```
estadoRet = false;
if ( (millis() - delayBotao) > tempoDebounce ) {
    estadoBotao = digitalRead(pinBotao2);
    if ( !estadoBotao && (estadoBotao != estadoBotaoAnt) ) {
        estadoRet = true;
        delayBotao = millis();
    }
    if ( estadoBotao && (estadoBotao != estadoBotaoAnt) ) {
        delayBotao = millis();
    }
    estadoBotaoAnt = estadoBotao;
}

return estadoRet;
}

void pinLedAPisca(bool reset) {
static unsigned long delayPisca = 0;

    if (reset) {
        delayPisca = millis();
    } else {

        if ((millis() - delayPisca) < 500) {
            digitalWrite(pinLedA, LOW);
        } else {
            digitalWrite(pinLedA, HIGH);
        }

        if ((millis() - delayPisca) >= 1000) {
            delayPisca = millis();
        }
    }
}

void setMemoria(byte posicao, byte servo, byte valor) {
int posMem;

    posMem = ((posicao * 5) + servo) + 1;
    EEPROM.write(posMem, valor);
}

byte readMemoria(byte posicao, byte servo) {
```

```

int posMem;

    posMem = ((posicao * 5) + servo) + 1;
    return EEPROM.read(posMem);
}
\end{ArduinoSketchBox}

```

Após a realizada a montagem da Figura 1 acima e verificado o correto funcionamento na protoboard, foi realizado o projeto da placa de circuito impresso no software Eagle e montagem (confecção e soldagem dos componentes), como mostra as figuras abaixo.

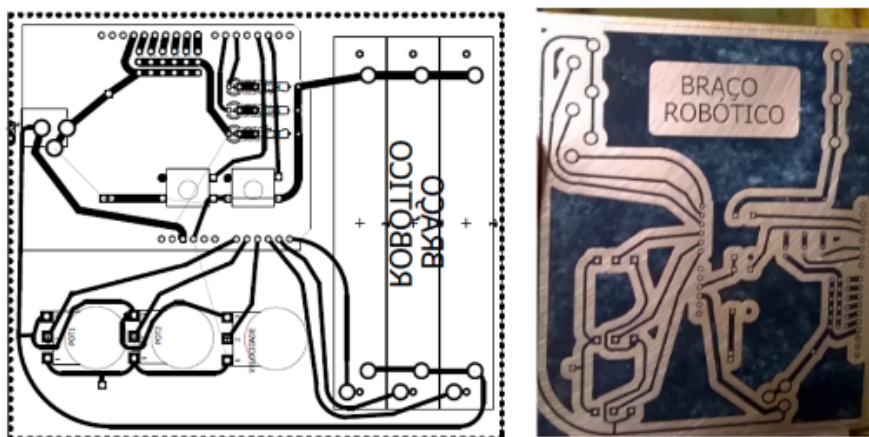


Figura 2: Esquemático de trilhas e placa de circuito impresso



Figura 3: Placa finalizada

Com a placa finalizada foram confeccionadas todas as peças de spumapaper necessárias pra a estrutura do Braço Robótico. Na Figura 4 é ilustrado as peças confeccionadas e na Figura 5 é mostrado a estrutura geral do Braço Robótico.

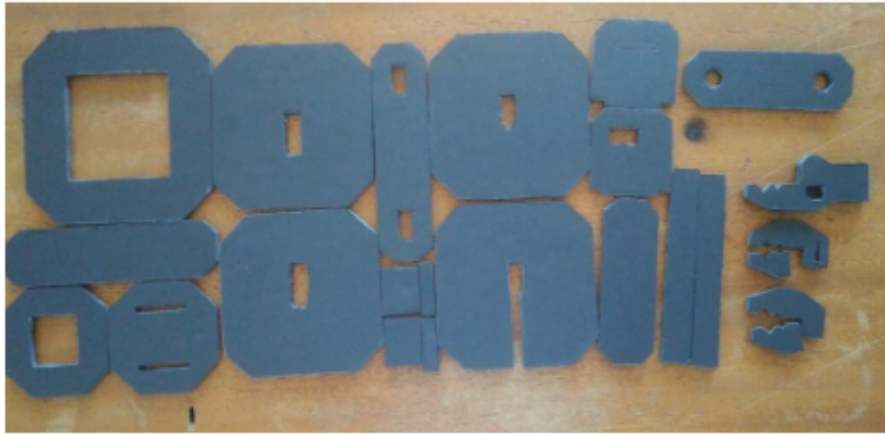


Figura 4: Peças confeccionadas

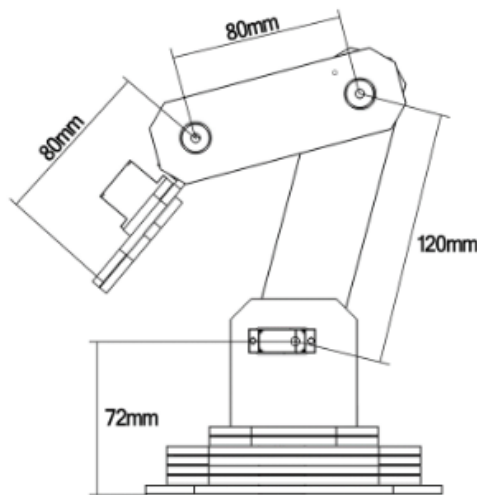


Figura 5: Estrutura geral do Braço Robótico

Funcionamento do Braço Robótico O Braço Robótico é alimentado por uma fonte de 5V (4 pilhas de 1.2V em série), o Led vermelho estará piscando e o verde aceso quando o projeto estiver em funcionamento. Existem dois modos de funcionamento, o modo manual e o modo de gravação. No modo manual os servos motores são todos controlados pelos potenciômetros (exceto o potenciômetro de knob verde) da placa de circuito impresso que está conectada com o Arduino Uno. No modo de gravação poderão ser gravados até cinco movimentos do Braço Robótico. Para isso a cada posição de interesse é preciso apertar e soltar o push button que está logo abaixo dos Leds, o Led amarelo irá piscar indicando que a posição foi gravada. Após as gravações das posições apertar-se e segura o push button mais distante dos Leds até que o Led vermelho pare de piscar e se apague, depois deve-se clicar no outro push button e o Braço Robótico passará a executar as posições que foram gravadas e o Led amarelo acenderá. O potenciômetro de knob verde determina a velocidade de execução dos movimentos que foram gravados no modo de gravação.

III. RESULTADOS

O projeto apresentou resultados satisfatórios desde sua construção e cumpriu com as funções que foram determinadas para ele. É capaz de executar movimentos em diferentes direções tanto de forma manual como de forma automática pelo modo de gravação. Alguns pontos do projeto que se deve ter uma atenção é a questão da alimentação, que para o correto funcionamento deve estar sempre em 5V, portanto problemas de bateria descarregadas podem acontecer com frequência. Além disso, o material e o encaixe dos servos não possuem muita qualidade, dessa forma pode acontecer de alguma das partes se soltarem ou até mesmo se danificarem. Por fim, o projeto apresenta bom funcionamento durante as apresentações e gera muito interesse do público. Na Figura 6 é apresentado o projeto finalizado do Braço Robótico.

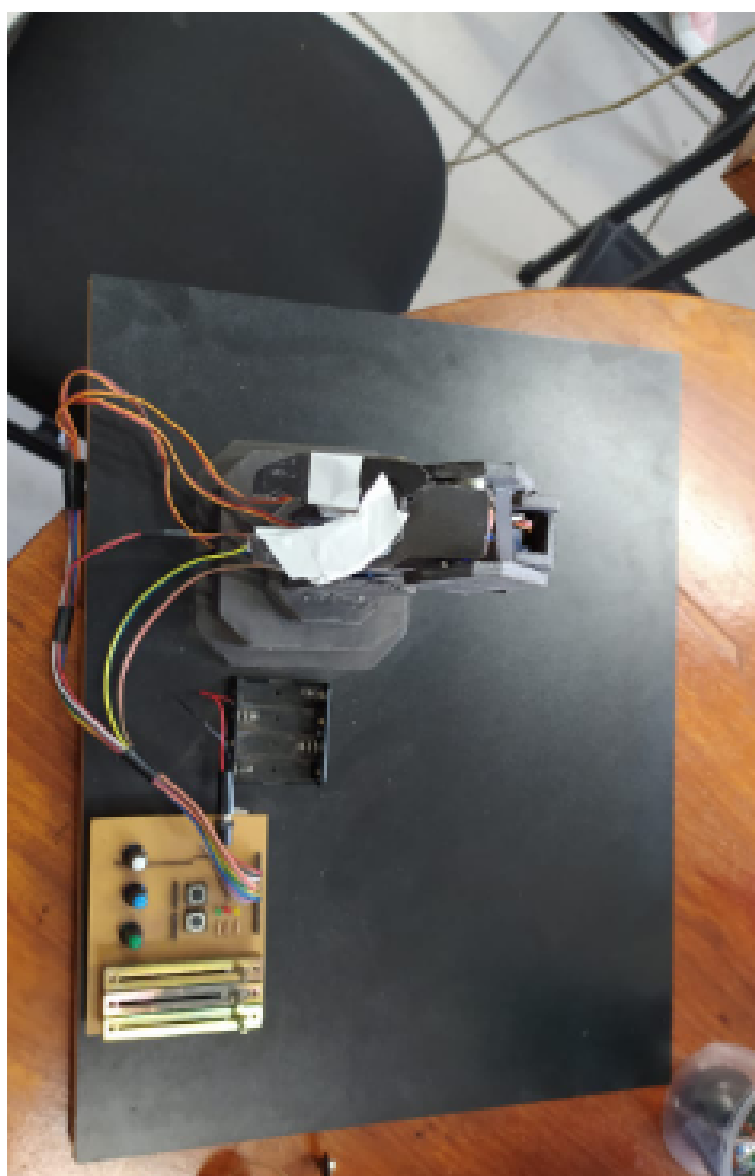


Figura 6: Braço Robótico

REFERÊNCIAS

IV. DISCUSSÃO

O Braço Robótico não sofreu alterações significativas desde a sua construção, resolver os problemas de alimentação e de fragilidade citados anteriormente pode melhorar significativamente o protótipo. Além disso, tomando como base o modelo desenvolvido é possível elaborar um braço robótico em maior escala, mais robusto e com maior grau de movimentos.

REFERÊNCIAS

<<https://www.youtube.com/watch?v=vVOyWQZ25M8>>