

# Mão Protética

Amauri da Costa Junior  
FGA  
UNB-Universidade de Brasília  
Gama, Brasil  
Amauri.cj@hotmail.com

Thássio Gabriel Farias dos Santos  
FGA  
UNB-Universidade de Brasília  
Gama, Brasil  
Thassio.96@gmail.com

**Resumo**—Este documento mostra de forma detalhada a elaboração do projeto de uma mão protética controlada pelo microcontrolador MSP430 ao descrever toda a sua composição de Hardware e Software.

**Palavras-Chave**—Mão, Prótese, Software, Hardware.

## I. INTRODUÇÃO

A perda de um dos membros do corpo gera algum tipo de dificuldade para a pessoa que o perdeu. Quando o membro perdido é a mão, funções básicas acabam sendo perdidas como, por exemplo, escrever e carregar objetos. Poder recuperar nem que sejam parcialmente essas funções, é sem dúvidas algo desejado por aqueles que não conseguem mais executar essas funções e foi pensando nisso que se decidiu elaborar um projeto com foco nesse tema. Para que fosse possível recuperar essas funções, elaborou-se a ideia de criar uma mão protética.

Para controlar a mão tem-se como ideia usar o microcontrolador MSP430 que receberá em sua entrada um sinal analógico da deformação sofrida pelo Piezoelétrico e baseado nesse sinal será controlado de forma devida um servomotor.

## II. DESENVOLVIMENTO

### A. Descrição de Hardware

Para a realização deste projeto foi utilizada a seguinte lista de materiais

Table I. BILL OF MATERIALS

BILL OF MATERIALS		
ITEM	UNIDADE	QUANTIDADE
MSP430G2553 launchpad	CADA	1
PIEZOELÉTRICO	CADA	1
SERVOMOTOR	CADA	1
LINHA DE NYLON	METRO	1

MÃO DE PLÁSTICO	CADA	1
FITA ISOLANTE	METRO	3
Elástico	CADA	2
JUMPER	CADA	3
PULSEIRA ELÁSTICA	CADA	1

O hardware consiste em uma mão de plástico que possui fios de nylon que funcionam como tendões que são puxados por um servo motor controlado pela MSP430. O servo motor necessita de alimentação externa, pois a MSP430 não possui a corrente necessária para o motor funcionar como desejado. Também é utilizada uma pulseira onde fica o sensor de pressão que receberá dados do pulso do usuário, esses dados são enviados para o microcontrolador que processa esses sinais.

As conexões que compõem a parte eletrônica do projeto foram feitas de acordo com o anexo 1.

Nela, o servo recebe alimentação externa de uma bateria e sua entrada PMW se conecta ao pino 2.1 do servomotor. A leitura do sinal enviado pelo piezoelétrico é feita pela porta 1.5. O botão utilizado é o que já vem instalado na launchpad e é equivalente ao pino 1.3.

### B. Descrição de Software.

O código desenvolvido foi feito com base na linguagem arduino e usou-se as bibliotecas *Servo.h* para controlar o Servomotor e a biblioteca *msp430.h* para utilizar as funções do microcontrolador *msp430g2553*.

O software criado para a MSP430 recebe os dados do sensor de pressão em valores de tensão, ela transforma os valores de tensão em valores que variam de 0 a 1023, com 0 sendo 0 volts e 1023 sendo 5 volts. De acordo com o valor de 0 a 1023 o microcontrolador faz com que o motor puxe os fios da mão, o código leva em consideração uma variação de 300 antes de mandar a ordem para o motor, isso foi criado para que

o servo motor não ficasse realizando movimentos a cada mínima variação na coleta de dados do sensor.

O código também conta com um sistema de calibragem para melhor adequar o equipamento ao usuário. Quando o usuário aperta o botão o valor máximo que é de 1023 diminui em 100, isso serve para que se o sensor estiver recebendo um valor menor a mão possa fazer um movimento mais amplo, ao invés de apenas pequenos movimentos. Existem sete níveis diferentes de calibragem, quando o botão é apertado pela sétima vez ele retorna para a calibragem inicial de valor 1023.

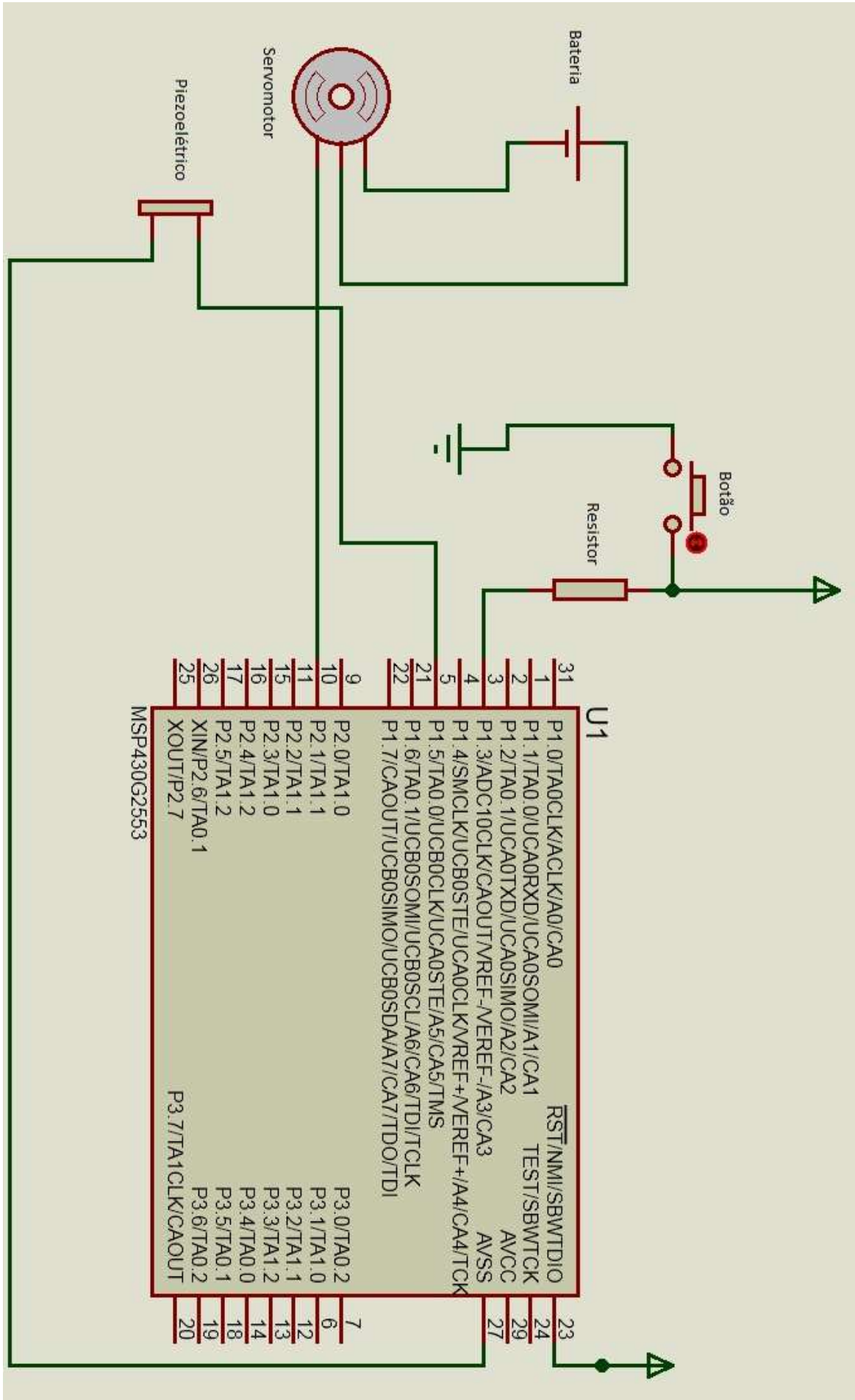
O código funciona de acordo com o fluxograma encontrado no anexo 2 e o código na íntegra se encontra no anexo 3.

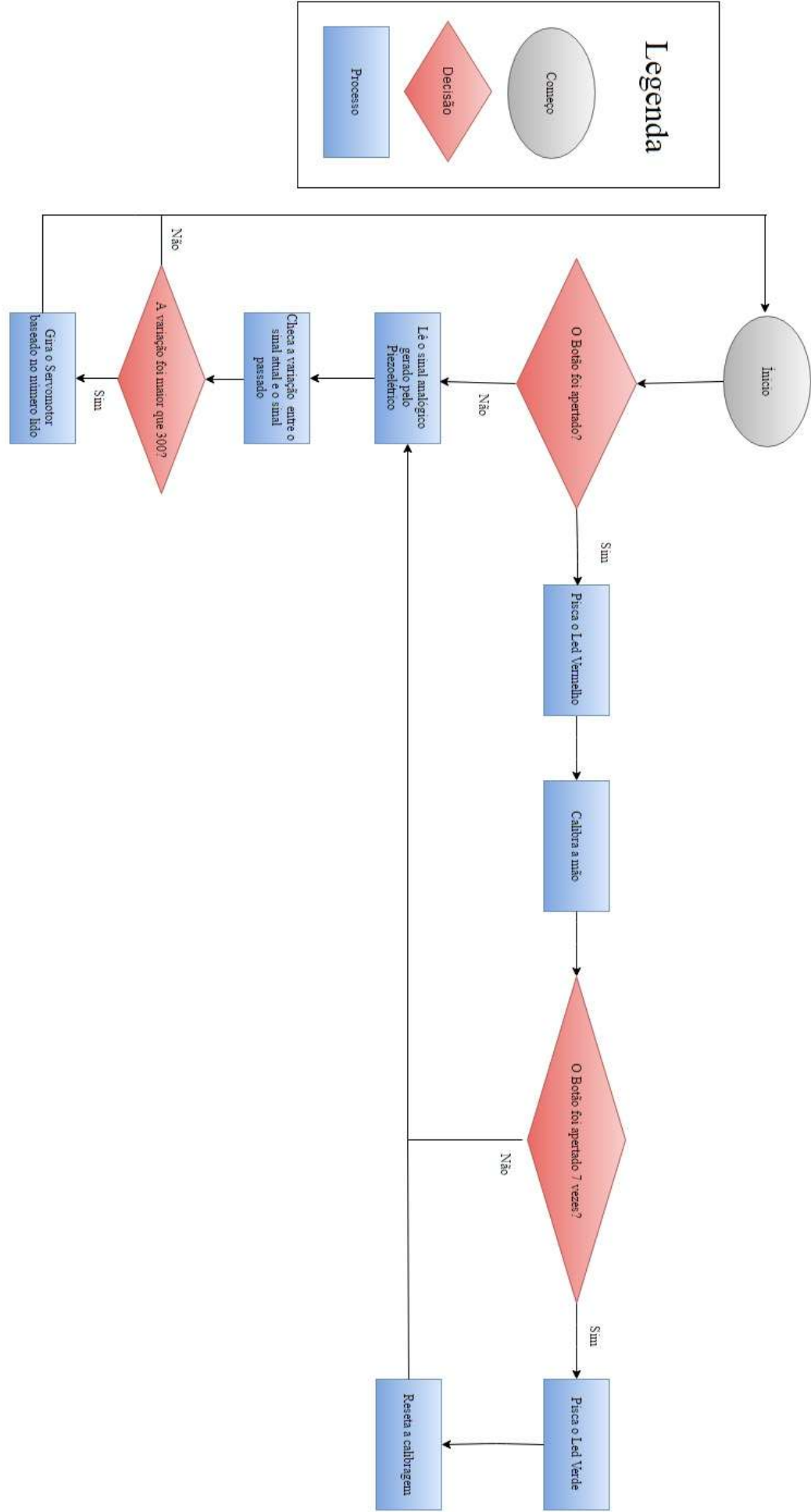
### III. RESULTADOS

Foram realizados diversos testes com a mão biônica, como segurar pesos, e tempo de resposta entre o movimento real da mão. A mão biônica conseguiu manter pesos de objetos do dia-a-dia sem derrubá-los, o único erro percebido foi que após algum tempo o sensor devolvia um valor menor, e esse valor ia diminuindo, após essa variação superar os 300 de segurança, a mão começa a relaxar. O teste de tempo de resposta foi satisfatório, o sistema completo respondeu quase que imediatamente ao movimento da mão.

### IV. CONCLUSÃO

Após o fim dos testes e várias calibrações no sensor e no código pode-se concluir que o projeto é viável de ser realizado utilizando a MSP430. Tudo que foi proposto no ponto de controle 1 foi realizado utilizando a energia para programar a MSP, alguns pontos que poderiam ser melhorados é a perda de força da mão com o passar do tempo, e aumentar as possibilidades de movimento da mão biônica.





### Anexo 3

```
#include <Servo.h>
#include <msp430.h>
```

```
Servo myservo;
```

```
int PinoServo = 9;
int leitoranalogico = 7;
int val;
int x;
int op;
int ValorServo = 1023;
```

```
int botao = PUSH2;
int ledverde = GREEN_LED;
int ledvermelho = RED_LED;
int estadob = 0;
int conta = 0;
```

```
void setup() {
  myservo.attach(9);
  Serial.begin(9600);
  pinMode(ledverde, OUTPUT);
  pinMode(ledvermelho, OUTPUT);
  pinMode(botao, INPUT_PULLUP);
  pinMode(leitoranalogico, INPUT);
}
```

```
void loop() {
```

```
  estadob = analogRead(botao);
```

```
  if (estadob == LOW) {
```

```
    digitalWrite(ledvermelho, HIGH);
    conta = conta + 1;
    ValorServo = ValorServo - 100;
    delay(500);
```

```
  }
  else {
    digitalWrite(ledvermelho, LOW);
  }
```

```
  if ( conta == 6){
    digitalWrite(ledverde, HIGH);
    conta = 0;
    ValorServo = 1023;
    delay(500);
```

```
  }else digitalWrite(ledverde, LOW);
```

```
    val = analogRead(leitoranalogico);
    val = map(val, 0, ValorServo, 0, 180);
    op = val - x;
    if (op < 300 && op > -300)
    {
      myservo.write(val);
    }

    delay(15);
    x = val;
  }
```