Acadêmico: Rômulo Caetano Guerra

Relatório 2º Código Circuitos Digitais

A primeira coisa a ser feita, foi declarar cada uma das portas que estavam conectadas ao Arduíno, referente aos segmentos do visor.

```
struct DisplayPorts {
  int a = 3;
  int b = 4;
  int c = 6;
  int d = 7;
  int e = 8;
  int f = 9;
  int g = 10;
  int dp = 5;
} display1;
```

Após isso declarei fiz algumas declarações **void printNumber(int n),** sendo o **printNumber** a declaração do numero que será escrito no visor de 7 segmentos, e **int n** o case selecionado, para decidir qual numero será mostrado.

O **void displayControl(struct DisplayPorts dp, int conf[]),** foi a declaração da forma que será executado o comando e como sairá no visor. Sendo **struct DisplayPorts dp**, está apontado como qual saída foi apontado e **int conf[]** a configuração desejada.

No **void setup(),** é configurado os **pinMode,** que são declarações para saber se o pino é Entrada ou Saida.

```
void printNumber(int n);

void displayControl(struct DisplayPorts dp, int conf[] );

void setup() {
  pinMode(13, OUTPUT);

  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);

  pinMode(3T);

  pinMode(13,INPUT);
  pinMode(12,INPUT);
  pinMode(10,INPUT);
  pinMode(5,INPUT);
}
```

No **void loop(),** é declarado e configurada como irá funcionar o circuito. O **digitalWrite(13, HIGH),** é o comando de mandar ou não, energia para a porta apontada, no caso a **13** e se define se vai mandar energia ou não, com **HIGH** ou **LOW.**

Já os **int A=digitalRead(*),** é uma leitura para saber se o pino está **on "1"** ou **off "0",** o *, é para definir qual pino.

Os **out[*],** são as comparações feitas para que o **Interruptor DIP DPST x4**, venha a ter 9 entradas em vez de 4, de acordo com a BCD 4511.

```
void loop() {
  digitalWrite(13, HIGH);
  int A=digitalRead(5), B=digitalRead(10), C=digitalRead(12), D=digitalRead(13);
  A=false;B=false;C=true;D=false;
  int out[8];
  out[0] = !((!A&&C)||(B&&C)||(C&&!D)||(!A&&B&&C)||(A&&!D)||(A&&!B&&!C));
  out[1] = !((!A&&!C&&!D)||(!B&&!C)||(A&&!C&&D)||(!A&&!B)||(A&&C&&D)||(!B&&C&&!D));
  out[2] = !((!A&&!C)||(!A&&D)||(B)||(!B&&!C&&D));
  out[3] = !((!A&&!D)||(!B&&C&&D)||(B&&!C)||(!C&&!D)||(B&&!D)||(A&&!C));
  out[4] = !((C&&!D)||(!A&&!B&&!D)||(A&&B)||(A&&C)||(A&&D));
  out[5] = !((!C&&!D)||(A&&!B)||(A&&C)||(!A&&B&&C)||(B&&!D));
  out[6] = !((C&&!D)||(A&&!B)||(!A&&B&&!C)||(A&&D)||(!B&&C));
  out[7] = 1;
  displayControl(display1, out);
}
```

Aqui temos a declaração **switch(n),** que na tradução seria troca, ou seja, de acordo com o numero inserido, o arduíno responde com a definição daquela entrada.

Já o **confDisplay[8] = {0,0,0,0,0,0,1,1},** está a definir a configuração de saída. O "**8",** define quantas entradas irá receber, cada número é para um casa de segmento + a porta DP, ou seja, assim que receber o código abaixo, o mesmo irá imprimir o primeiro numero no segmento.

g	f	e	d	c	b	a	dp
0	0	0	0	0	0	1	1

```
void printNumber(int n) {
  switch (n) {
    case 0: {
      int confDisplay[8] = {0,0,0,0,0,0,1,1};
      displayControl(display1, confDisplay);
      break;
    }

    case 1: {
      int confDisplay[8] = {1,0,0,1,1,1,1,1};
      displayControl(display1, confDisplay);
      break;
    }
```

Já aqui temos a escrita no segmento, **digitalWrite(dp.a,conf[0]),** como já dito antes, **digitalWrite** para "escrever", **dp.a** para decisão de entrada e **conf[*],** para definição da saída.

```
void displayControl(struct DisplayPorts dp, int conf[] ){
  digitalWrite(dp.a,conf[0]);
  digitalWrite(dp.b,conf[1]);
  digitalWrite(dp.c,conf[2]);
  digitalWrite(dp.d,conf[3]);
  digitalWrite(dp.e,conf[4]);
  digitalWrite(dp.f,conf[5]);
  digitalWrite(dp.g,conf[6]);
  digitalWrite(dp.dp,conf[7]);
}
```