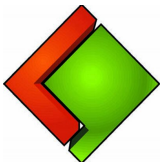


LIBERATO SALZANO VIEIRA DA CUNHA TECHNICAL SCHOOL
FOUNDATION TECHNICAL

COURSE IN ELECTRONICS

Matheus M. Schumacher

Three-Phase Soft Starter Report with TCA785 and Arduino Control

Subject: Power Electronics
Professor: João Neves

Novo Hamburgo, 2021

INTRODUCTION TO THE SOFT STARTER:

It consists of driving a motor using a firing circuit with **TRIACs**, which allow the control of the motor and the configured acceleration and deceleration ramps. It controls the power circuit by varying the firing angle of the Triacs and consequently varying the effective voltage applied to the motor. It also features a **200% overcurrent protection system** of the motor's nominal current, **DC Braking**, and an **HMI** (Human-Machine Interface) on a 128x64 OLED display that, together with an **Arduino UNO**, are responsible for turning the motor on/off and changing the time values of the previously mentioned ramps.

TCA785 (ZERO CROSSING/PPM/CONTROL):

It was chosen due to its practicality; this IC has an **internal zero-crossing detector**, an output with the **PPM signal synchronized** with it, and allows the control of the motor's acceleration and deceleration through its **pin 11**.

OPTOCOUPLER:

The **MOC3021** was used in this project, which has an LED that drives 2 antiparallel diodes. When a current of 10mA passes through this LED, the signal at its output will trigger a current in the gate sufficient to activate the **TRIAC** and turn on the motor.

PROTECTION STAGE:

Using a **5mA current transformer**, the current of one of the three-phase motor's phases is measured. At the TC output, there is a rectifier bridge that transforms the signal into a **DC level**, followed by a **non-inverting amplifier** that boosts the signal and takes it to a **comparator**. The comparator checks if the value is greater than the value corresponding to a current near **200%**. If so, the comparator's

output is connected to the **Arduino**, which checks this value, and if positive, the motor will be switched off.

Código do Arduino:

Menu Display - Example:

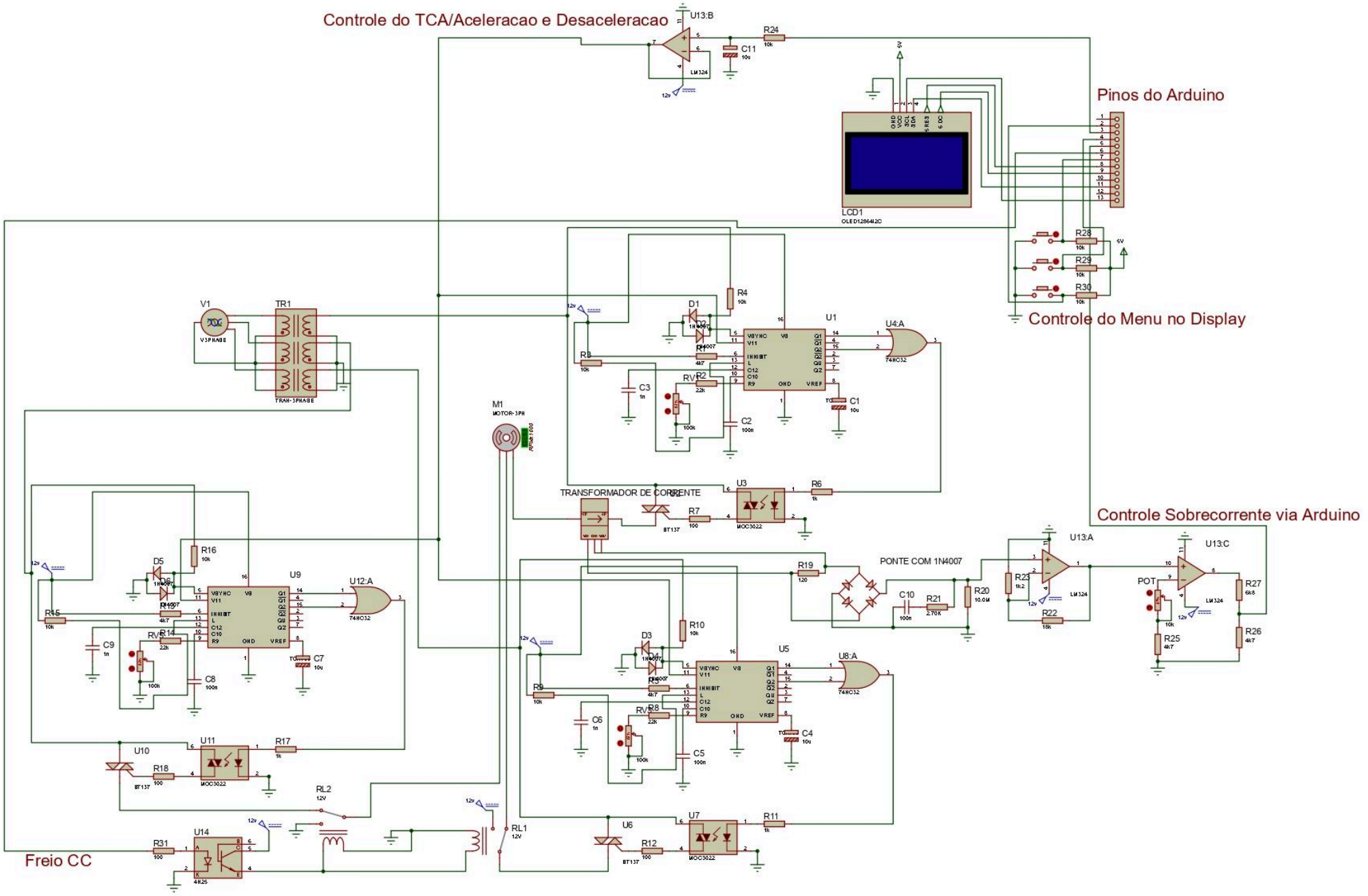

```
#include <Arduino.h>
#include <U8g2lib.h>
#ifdef U8X8_HAVE_HW_SPI
#include <SPI.h>
#endif
#ifdef U8X8_HAVE_HW_I2C
#include <Wire.h>
#endif
const int buttonPin = 7;
const int buttonPin2 = 4;
const int buttonPin3 = 2;
int onoff=0;
int i=0;
int counter=1;
float aceleration=1;
float desaceleration=1;
char* ACE;
char* DES;
unsigned long tempoAnterior=0;
unsigned long tempoAtual=0;
unsigned long ti=0;
unsigned long to=0;
int2str function (converts integer to string) //Function omitted, available online
U8G2_SSD1306_128X64_NONAME_1_4W_SW_SPI u8g2(U8G2_R0, /* clock=*/ 13, /* data=*/ 11, /* cs=*/ 10, /* dc=*/ 9, /* reset=*/ 8);

void setup(void) {
  pinMode(9, OUTPUT);
  digitalWrite(10, 0);
  digitalWrite(9, 0);
  pinMode(buttonPin, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
```

```
  pinMode(3,OUTPUT);
   pinMode(5,INPUT);
    pinMode(6,OUTPUT);
  Serial.begin(9600);
     u8g2.begin();
     u8g2.setFont(u8g2_font_6x12_tf);
     u8g2.firstPage();
}
void loop(void) {
  int buttonState;
  int buttonState2;
  int buttonState3;
  int freiocc;
  int checkoverload;
  static int select=0;
  do {
  tempoAtual=millis();
     if(onoff==0 && i<255 && checkoverload==LOW){
    for(i=45;i<=255;i++)
    {
  analogWrite(3,i);
  delay(desaceleration*4.76);
    }
  }
    if(onoff==1 && i>=0 && checkoverload==LOW)
    { for(i=255;i>=35;i--)
    {
  analogWrite(3,i);
  delay(aceleration*4.54);
    }
      }
      if(checkoverload==HIGH){
        analogWrite(3,0);
      }
  buttonState = digitalRead(buttonPin);
  buttonState2 = digitalRead(buttonPin2);
  buttonState3 = digitalRead(buttonPin3);
  checkoverload = digitalRead(5);
if (buttonState == LOW && tempoAtual-tempoAnterior>=200) // light the LED
  {tempoAnterior=millis();
```

```
  u8g2.clearDisplay();
  if(counter==1 && select%2!=0)
  {aceleration++;
  }else if(counter==2 && select%2!=0){
    desaceleration++;
    }else {counter++;
    }
}
 if (buttonState2 == LOW && tempoAtual-tempoAnterior>=200) // light the LED
{tempoAnterior=millis();
  u8g2.clearDisplay();
   if(counter==1 && select%2!=0)
   {aceleration--;
    }else if(counter==2 && select%2!=0){
      desaceleration--;
      } else {counter--;
      }
}
 if (buttonState3 == LOW && tempoAtual-tempoAnterior>=200) // light the LED
{tempoAnterior=millis();
  select++;
  u8g2.clearDisplay();
}
switch(counter)
{
  case 1:
if(select%2==0){
    u8g2.drawStr(13,9,"Softstarter Menu");
    u8g2.drawStr(8,26,"-Acceleration Time");
    u8g2.drawStr(8,39," Deceleration Time");
    u8g2.drawStr(40,53," Turn On");
    }else {
      u8g2.drawStr(13,9,"Acceleration Time");
      ACE=int2str(aceleration);
      String fixed = ACE;
    u8g2.drawStr(13,30,fixed.c_str());
    u8g2.drawStr(40,30,"Seconds");
    }
    break;
  case 2:
```

```
  if(select%2==0){
    u8g2.drawStr(13,9,"Softstarter Menu");
    u8g2.drawStr(8,26," Acceleration Time");
    u8g2.drawStr(8,39,"-Deceleration Time");
    u8g2.drawStr(40,53," Turn On");
    }else {
    u8g2.drawStr(13,9,"Deceleration Time");
    DES=int2str(desaceleration);
    String name = DES;
    u8g2.drawStr(13,30,name.c_str());
    u8g2.drawStr(40,30,"Seconds");
    }
    break;
    case 3:
      if(select%2==0){onoff=0;
    u8g2.drawStr(13,9,"Softstarter Menu");
    u8g2.drawStr(8,26," Acceleration Time");
    u8g2.drawStr(8,39," Deceleration Time");
    u8g2.drawStr(40,53,"-Turn On");
    }else {onoff=1;
     u8g2.drawStr(20,39,"-Turn Off");
      }
    break;
  default:
    counter=1;
    u8g2.clearDisplay();
    break;
}
  } while ( u8g2.nextPage() ); }
```


Controle do TCA/Aceleracao e Desaceleracao
Pinos do Arduino
Controle do Menu no Display
Controle Sobrecorrente via Arduino
Freio CC
TRANSFORMADOR DE CORRENTE
PONTE COM 1N4007
LCD1
M1
V1
TR1
U1
U3
U4:A
U5
U6
U7
U8:A
U9
U10
U11
U12:A
U13:A
U13:B
U13:C
U14
RL1
RL2
POT

CONCLUSION:

During the project's construction, several difficulties arose. Among the many problems that kept surfacing, one after the other, were regulating the current in the optocoupler's LED to values closer to the ideal (10mA), testing the DC Braking with 2 relays, where its instability when activating without any kind of delay caused a Triac to burn out. Controlling pin 11 of the TCA785 through the Arduino was not possible due to a small DC level that was present when making a direct connection between the two. Therefore, it was necessary to use the configuration shown in the schematic using an Op Amp to correct the problem. When building the Menu, the display used was not the common model, so figuring out how to connect it was also a challenge. But after overcoming all the obstacles, the soft starter worked properly, and the time invested paid off in learning.