Denis Taylor de Brito

Matrícula: 2016016794

CONTROLE MPPT(MAXIMUN POWER POINT TRACKING)

PROJETO FINAL DE DISCIPLINA ECOP04 - PROGRAMAÇÃO EMBARCADO

**SUMÁRIO**

Conteúdo

[RESUMO 4](#_Toc78279981)

[Introdução 5](#_Toc78279982)

[Desenvolvimento 6](#_Toc78279983)

[Código Desenvolvido 7](#_Toc78279984)

[SIMULAÇÃO 12](#_Toc78279985)

[Componentes Utilizados 15](#_Toc78279986)

[Coclusão 16](#_Toc78279987)

[Referência 17](#_Toc78279988)

# RESUMO

Este relatório tem por objeto, descrever o processo de criação de um código, uutilizando linguagem em C#, desenvolvido no software MPLAB IDE X, compilador XC8 versão 1.4, com função de monitorar durante todo o seu tempo de funcionamento, nivéis de tensão, corrente e surtos, afim de controlar o fornecimento de Energia proveniente de Paineis Solares ou Geração Eólica. E com isto fazer com que o sistema se torne independente em controle e medições de valores fornecidos durante a geração de energia.

ABSTRACT

The purpose of this report is to describe the process of creating a code, using C# language, developed in MPLAB IDE X software, XC8 compiler version 1.4, with the function of monitoring throughout its operating time, voltage, current and surges, in order to control the supply of Energy from Solar Panels or Wind Generation. And with this make the system become independent in control and measurements of values ​​provided during power generation.

# Introdução

Com a alta demanda em produção de energias renováveis no âmbito residencial, houve se a necessidade de desenvolver um produto que atendece este tipo de cliente A proposta de criação deste projeto tem a finalidade de desenvolver um software e um hardware, capaz de controlar e analisar o fluxo de energia renovaveis produzidas por meio de paineis solares e pás heólicas.

Este sistema tomara as decisões pertinentes a sua operacionalização, a fim de garantir um funcionamento adequado e semi-autônomo aendendo as normas (ABNTNBR 16.149 e a de procedimento de ensaios: ABNTNBR 16.150 e NBR/IEC62116. ABNTNBR 16.149 – Sistemas fotovoltaicos (FV) – Características da interface de conexão com a rede elétrica de distribuição) :vigentes, referente a produção deste tipo de energias no Brasil.

# Desenvolvimento

Para o processo de implementação e desenvolvimento deste produto, houve a necessidade de se buscar informações a partir de requisitos exigidos pelos clientes, normas ABNT existentes referente a produção e controle deste tipo de energia e ainda de custos para o equipamento. Em posse destas informações foi possivél desenvolver todas as atividades referente ao processo de controle e de visualização de todas as informações gerenciadas pelo hardware. Para se chegar a um produto que atendesse os objetvos inicialmete proposto, foram realizadas varias simulações em softwares dedicados com objetivo de se obter todos os dados de operacionalização do sistema.

Quanto a utilização das bibliotecas pertinentes ao MPLAB X foram realizada algumas mudanças e criação de outras (acd.h, lcd.h e PWM) para que o código pudesse ser simulado também na plataforma PicSimLab.

Concluída todas as simulações, obtive-se os componentes necessários a serem empregados no circuito que seguem em lista própria.. Contudo foram aplicados no processo de criação do hardware os seguintes itens, um microcontrolador da Marca Microchip denominado PIC18F4520, o software MPLAB X IDE linguagem C#, compilador XC8 versão 1.41, simulador PicSimLab plataforma PicGenius, softwares Proteus.

. Após a Confecção da Pci, foram realizados todos os teste em bancada, a fim de validar as simulações realizadas em software e para isto foram utilizados, um Varivolt trifásico, um Trafo como indutor, paineis solares, pás eólicas e multimetros para acompanhar o funcionamento deste sistema e confirmar as informação exibidas no LCD 16x4..

Código Desenvolvido

/\*

\* File: main.c

\* Author: Denis

\*

\* Created on 12 de Julho de 2021, 20:17

\*/

#include <xc.h>

#include <pic18f4520.h>

#include <stdio.h>

#include "Config\_Fuses.h"

#include "io.h"

#include "lcd.h"

#include "bits.h"

#include "adc.h"

#include <math.h>

#include "PWM.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*variaveis global \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

#define Varredura PORTDbits.RD0

#define Sin\_Operacao PORTDbits.RD1

#define Bloq\_Controle PORTDbits.RD2

#define Botao PORTBbits.RB0

#define Sensor PORTBbits.RB1

#define Chave PORTBbits.RB2

#define \_XTAL\_FREQ 8000000

char linha\_1[16];

char linha\_2[16];

char linha\_3[16];

char linha\_4[17];

char linha\_5[17];

int Dado\_1 = 1023;

int Flag\_Botao;

int Flag\_Sensor;

int Flag\_Chave;

float Ii = 0;

float Kp = 2;

float Ki = 10;

float dT = 0.040;

float Er, Ic, Io, Ir, Vc, Vo, Vu, Po;

float PWM\_Porcento;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Função Limite de Integração\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

float limiteTensaoCorrente(float x) {

if (x < 0) {

return 0; //valor minimo de tensão

}

if (x > 5) {

return 5; //valor máximo de tensão

}

return x;

}

void main(void) {

TRISA = 0x07;

PORTA = 0x00;

TRISC = 0xC1;

PORTC = 0x00;

TRISD = 0x00;

PORTD = 0x00;

TRISE = 0x00;

PORTE = 0x00;

TRISB = 0xFF;

PORTB = 0x00;

lcdInit();

Init\_ADC();

InitPWM();

lcdCommand(0x01);

lcdCommand(0x81);

lcdString("PROJETO FINAL");

lcdCommand(0xC1);

lcdString("CONTROLE MPPT");

lcdCommand(0xD0);

lcdString("EMBARCADO UNIFEI");

\_\_delay\_ms(5000);

lcdCommand(0x01);

Sin\_Operacao = 1;

Flag\_Botao = 0;

Flag\_Sensor = 0;

Flag\_Chave = 0;

while (1) {

if (Chave == 0) {

if (Flag\_Chave == 0) {

Flag\_Chave = 1;

} else

Flag\_Chave = 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Leitura do Botão de Bloqueio \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Saída de Sinalização

if (Botao == 0) {

if (Flag\_Botao == 0) {

Flag\_Botao = 1;

Sin\_Operacao = 0; //Sinal de Conntrole Desliga

} else {

Flag\_Botao = 0;

Sin\_Operacao = 1;

}

\_\_delay\_ms(15);

}

if (Sensor == 0) {

if (Flag\_Sensor == 0) {

Flag\_Sensor = 1;

} else {

Flag\_Sensor = 0;

Flag\_Botao = 0;

Sin\_Operacao = 1;

}

\_\_delay\_ms(15);

}

//\*\*\*\*\*\*\*\*\*\*\*\*Condições para Controle\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if ((Flag\_Sensor == 1) && (Flag\_Botao == 1)) {

//Controle será habilitado

Bloq\_Controle = 0; //desliga o sinal de Sistema bloqueado e libeera

Varredura = 1; // led começa a pisca indicando o inicio da varredura

//Leitura da Tensão do Sistema

Vo = Ler\_Tensao(0); //conversão de dados 5\*Dado / 1024

//Leitura do Corrente do Sistema

Io = Ler\_Tensao(1); //5\*Dado / 1024 conversão

Po = Vo\*Io; //calculando valaor de Potencia

//Leitura de referência da corrente

Ir = Ler\_Tensao(2); //conversão de dados 5\*Dado / 1024

//Leitura da Chave de Composição da referência

if (Flag\_Chave == 1) {

Ic = Ir + 0.01 \* Vo; //Faz composição de referência

} else

Ic = Ir; //utilizaza somente a referência sem composição

//Conrole PI de Corrente de inicio

Er = Ic - Io; //Erro (reverso)da Malha de Corrente

Ii = Ii + Er\*dT; //Parte integral

//Limites de integração

Ii = limiteTensaoCorrente(Ii);

//Soma da Parte Proporcional e Integral

Vc = Kp \* Er + Ki\*Ii;

//Limite d Informação de Comando

Vc = limiteTensaoCorrente(Vc);

Dado\_1 = (int) (1023 \* Vc / 5.0); //Conversão de Dados

EscrevePWM1(Dado\_1); //Valor de Saida de Controle 1

Vu = Vc;

if (Vu < 3.333) {

Vu = 3.333; //Valor correspodente a ALFA = 120

}

\_\_delay\_ms(15);

//Indicação de Varredura

Varredura = 0;

} else {

//Desasbilita o Controle

Bloq\_Controle = 1;

//Sinaliza Controle não Atuando

Dado\_1 = 0;

Ii = 0;

EscrevePWM1(Dado\_1);

}

sprintf(linha\_1, "Vo:%1.2f", Ler\_Tensao(0)); //Grava texto linha 1

sprintf(linha\_2, "Io:%1.2f", Ler\_Tensao(1)); //Grava texto linha 2

sprintf(linha\_3, "Po:%1.2f", Ler\_Tensao(0)\* Ler\_Tensao(1)); //Grava texto linha 3

sprintf(linha\_5, "Ic:%1.2f", Ic); //Grava texto linha 4

sprintf(linha\_4, "PWM:%2.3f", EscrevePWM1(Dado\_1)); //Grava texto linha 4

lcdCommand(0x80); //posiciona o cursor na linha 1, caracter 1

lcdString(linha\_1); //escreve texto na linha 1 do lcd

lcdCommand(0xC0); //posiciona o cursor na linha 1, caracter 10

lcdString(linha\_2); //escreve texto na linha 1 do lcd

lcdCommand(0x90); //posiciona o cursor na linha 2, caracter 1

lcdString(linha\_3); //escreve texto na linha 2 do lcd

lcdCommand(0x99); //escreve testo na linha 3, caracter 5

lcdString(linha\_5); //escreve na linha 3 do lcd

lcdCommand(0xD0); //posiciona o cursor na linha 2, caracter 10

lcdString(linha\_4); //escreve texto na linha 2 do lcd

}

}

SIMULAÇÃO

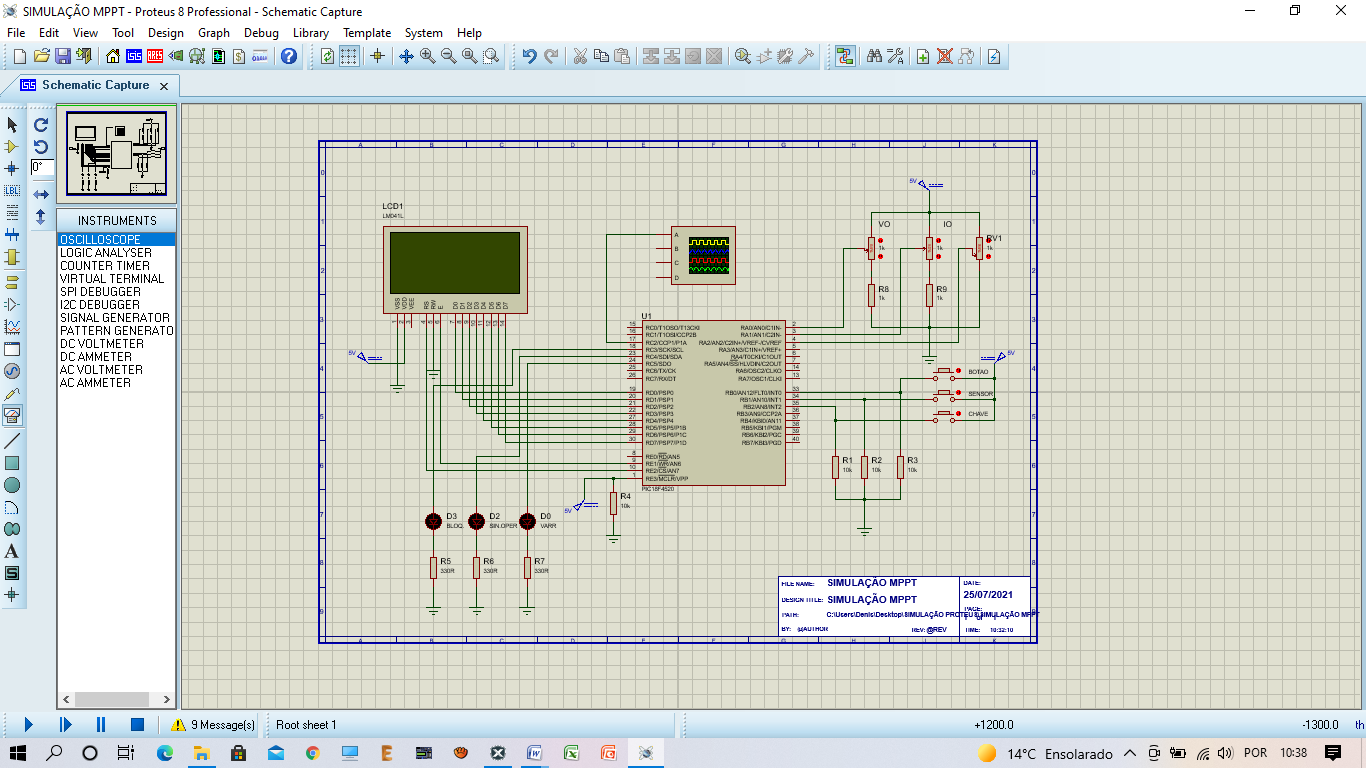


Figura 1 – Esquemático no Software Proteus versão 8.9

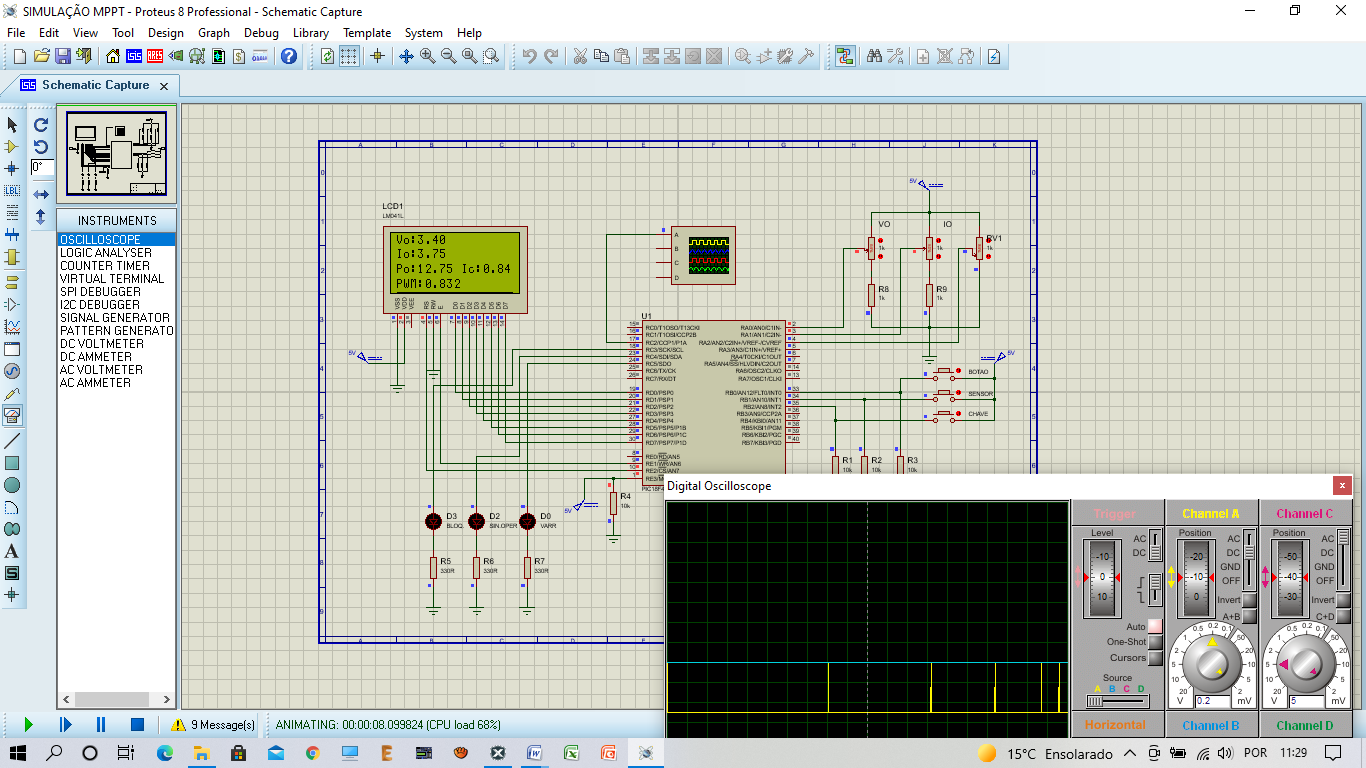


Figura 2 – Simulação no Software Proteus

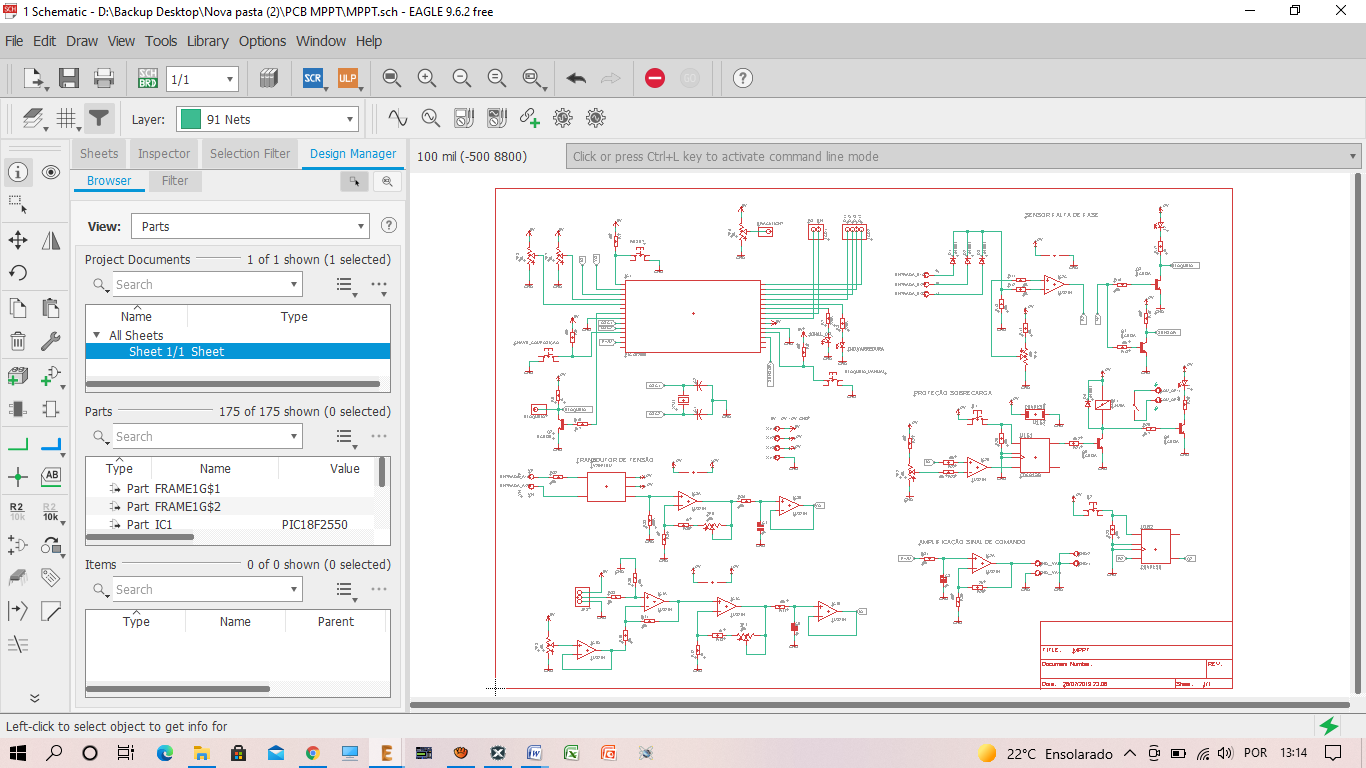


Figura 3 – Schematic no Eagle para confeccionar Pci

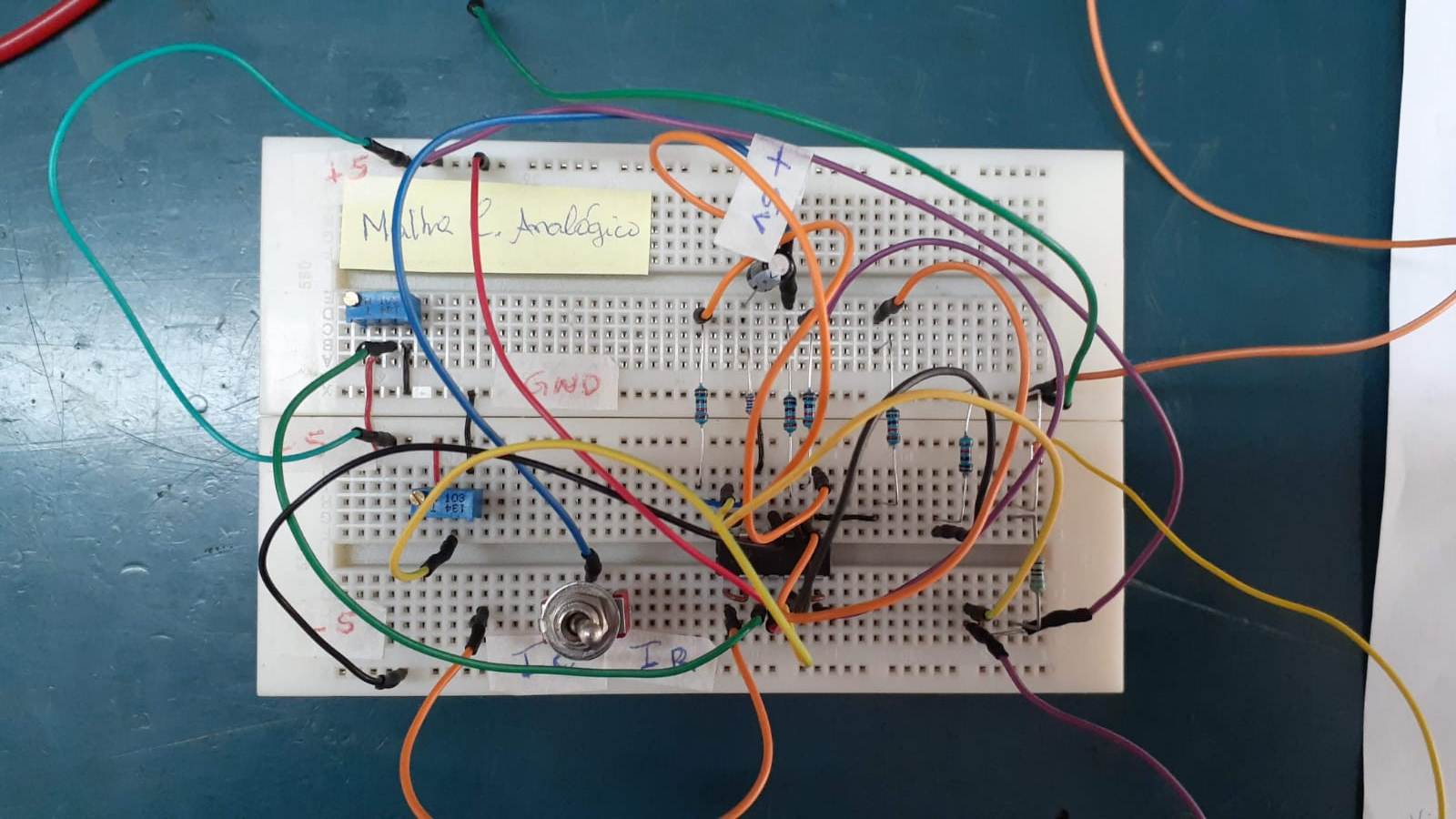


Figura 4 – Malha analógica de leitura de Tensão e Correntes

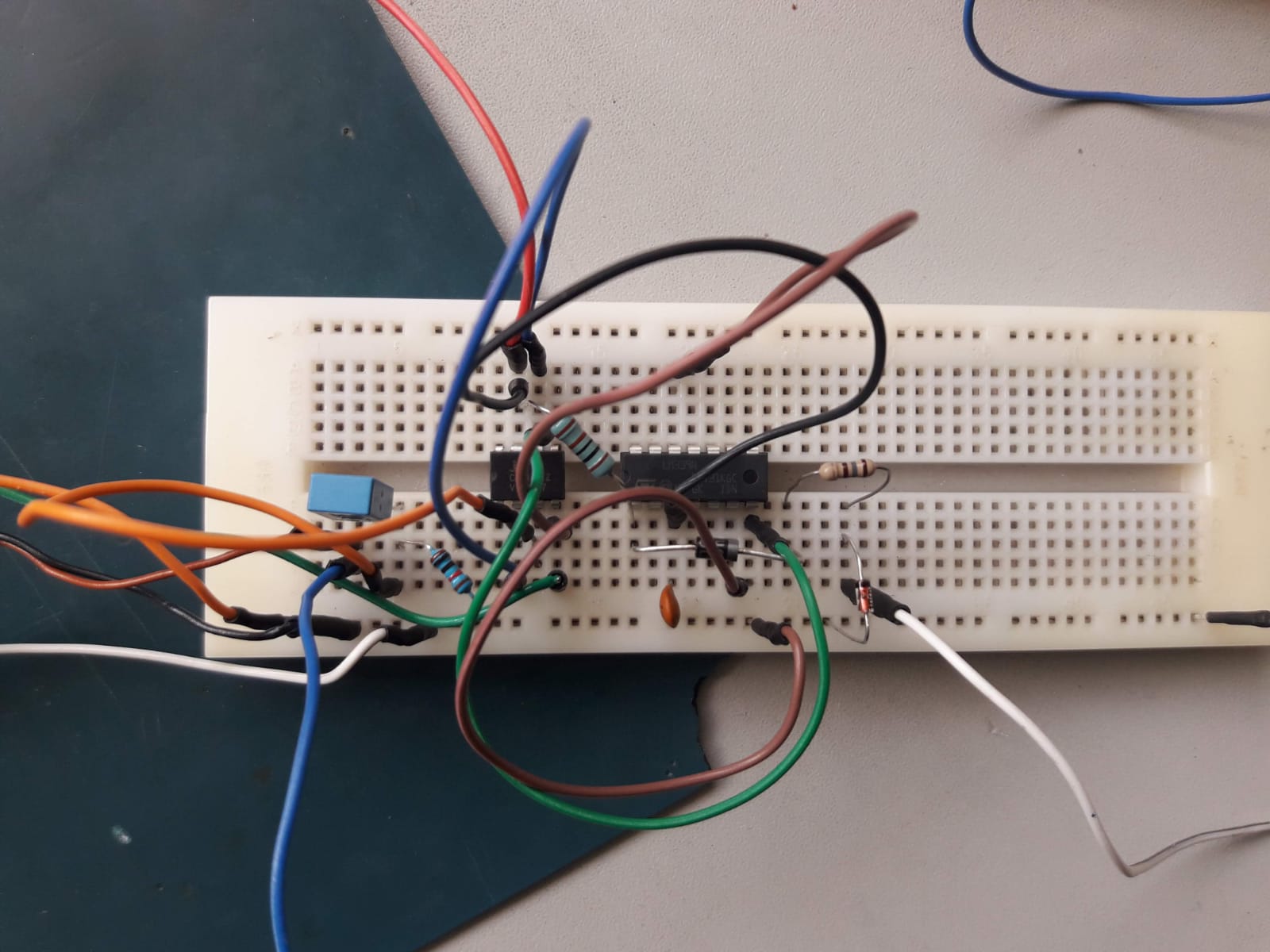


Figura 5 – Circuito Analógico Controle de Surto

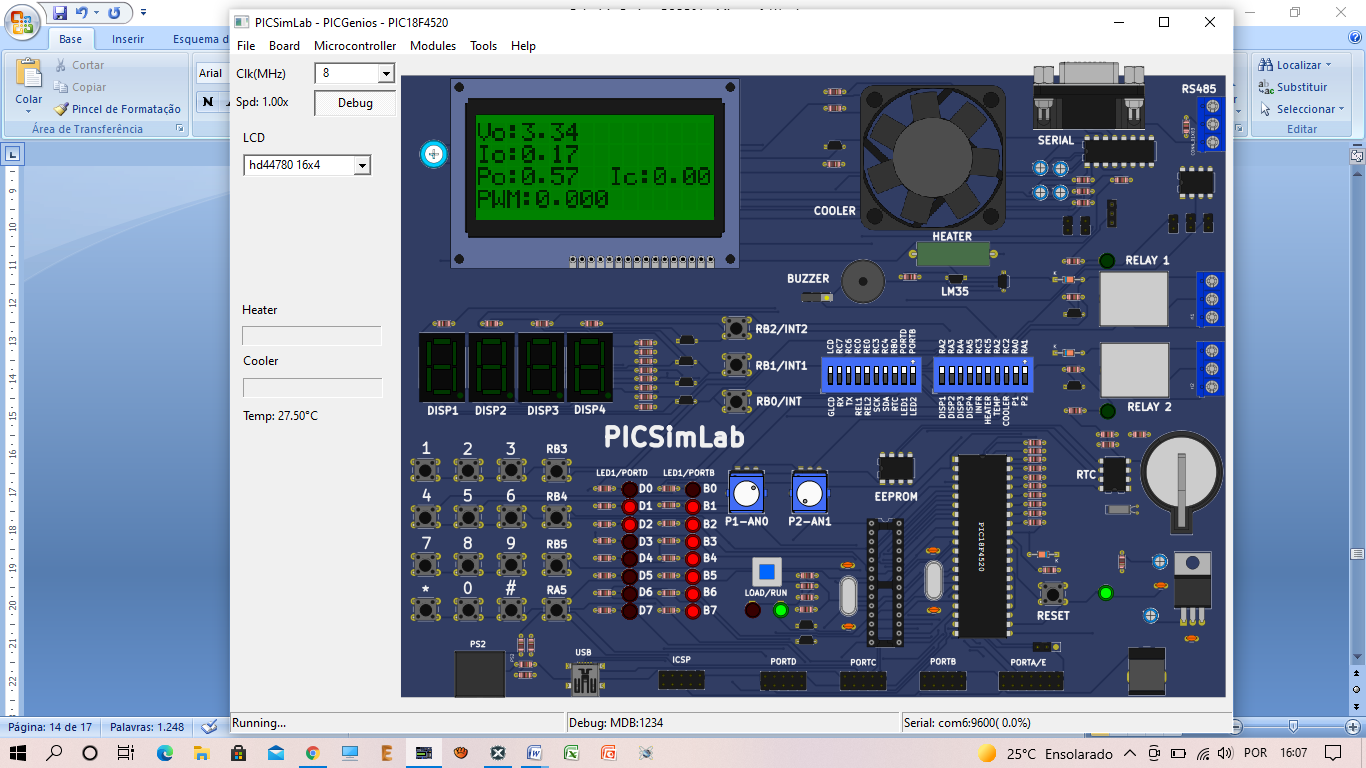


Figura 6 – Simulação PicSimLab

## Componentes Utilizados

|  |  |  |  |
| --- | --- | --- | --- |
| **Quant.** | **Descrição Material** | **Preço Unit.** | **Total** |
| 4 | Amplificador LM 324 | R$ 1,50 | R$ 6,00 |
| 1 | Sensor de Corrente ACS712 | R$ 18,90 | R$ 18,90 |
| 1 | Voltage Tranducer LV 20P | R$ 69,00 | R$ 69,00 |
| 1 | Placa de Fenolite dupla face 15x20 | R$ 24,00 | R$ 24,00 |
| 1 | Filip Flop CD4013 | R$ 2,76 | R$ 2,76 |
| 3 | Diodos 4001 | R$ 0,15 | R$ 0,45 |
| 10 | Resistores ¼ wats 10k | R$ 0.06 | R$ 0,60 |
| 4 | Diodos Leds Vermelho | R$ 2,50 | R$ 10,00 |
| 2 | Resistores ¼ walts 100k | R$ 0,06 | R$ 0,12 |
| 2 | Transistor BC548 NPN | R$ 0,30 | R$ 0,60 |
| 6 | Trimpot 10K | R$ 3,32 | R$ 19,92 |
| 1 | Display LCD 16x2 | R$ 44,50 | R$ 44,50 |
| 5 | Push Button | R$ 0,38 | R$ 1,90 |

# Coclusão

Diante da proposta para o projeto, foi possível colocar em prárica grande parte do conteúdo teórico e prático ministrado em aulas e também verificar algumas características no mundo de trabalho em desenvolvimento de sistemas embarcados. Com isto obtivemos resultados satisfatórios em controle na produção de energias renovaveis e que o produto final atende os objetivos propostos para o projeto.

Referência

<https://www.alldatasheet.com/datasheet-df/pdf/174116/ALLEGRO/ACS712.html>

<https://www.alldatasheetpt.com/datasheet-pdf/pdf/828813/LEM/LV20-P.html>

<https://www.alldatasheet.com/datasheet-pdf/pdf/112665/MICROCHIP/PIC18F4520.html>

<https://www.alldatasheet.com/datasheet-pdf/pdf/50840/FAIRCHILD/CD4013.html>