INSTITUTO FEDERAL
Ceará
Campus Fortaleza

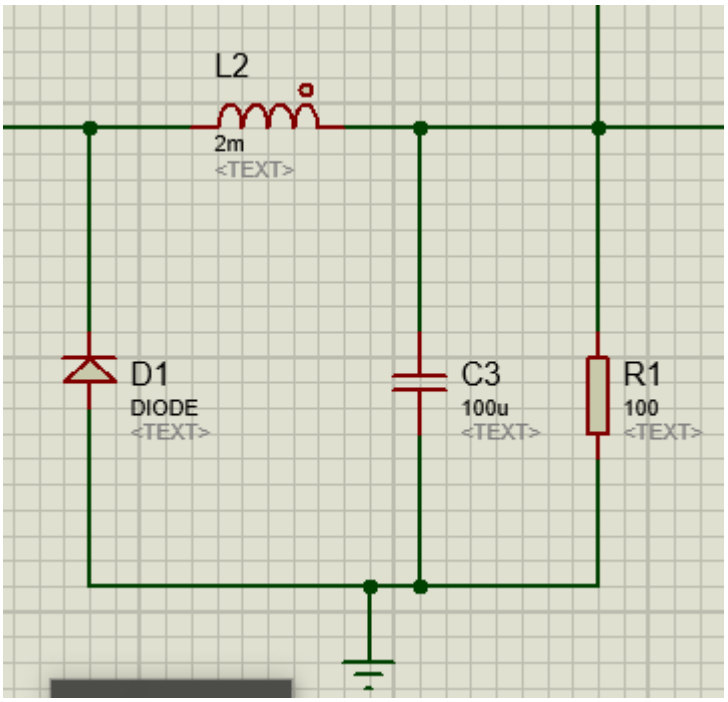
Projeto final de Aplicações de Controle

Alunos: Cristiano Coutinho, Gustavo Vasconcelos, João Gabriel, Angelo Nascimento

Passos da atividade

Cálculos manuais

Temos inicialmente a equação:
$V_0/(V_{in} * D) = R/(RLCs^2 + Ls + R)$



Do circuito acima (Informado pelo professor) sabemos que os valores das variáveis são:
$R = 100; L = 2 * 10^{-3}; C = 100 * 10^{-6}$

Substituindo os valores e simplificando, temos:

$$V_0/(V_{in} * D) = R/(RLCs^2 + Ls + R) =$$
$$= 100/(100 * 2 * 10^{-3} * 100 * 10^{-6} * s^2 + 2 * 10^{-3} * s + 100) =$$
$$\Rightarrow V_0/(V_{in} * D) = 100/(2 * 10^{-5} * s^2 + 2 * 10^{-3} * s + 100)$$

Matlab

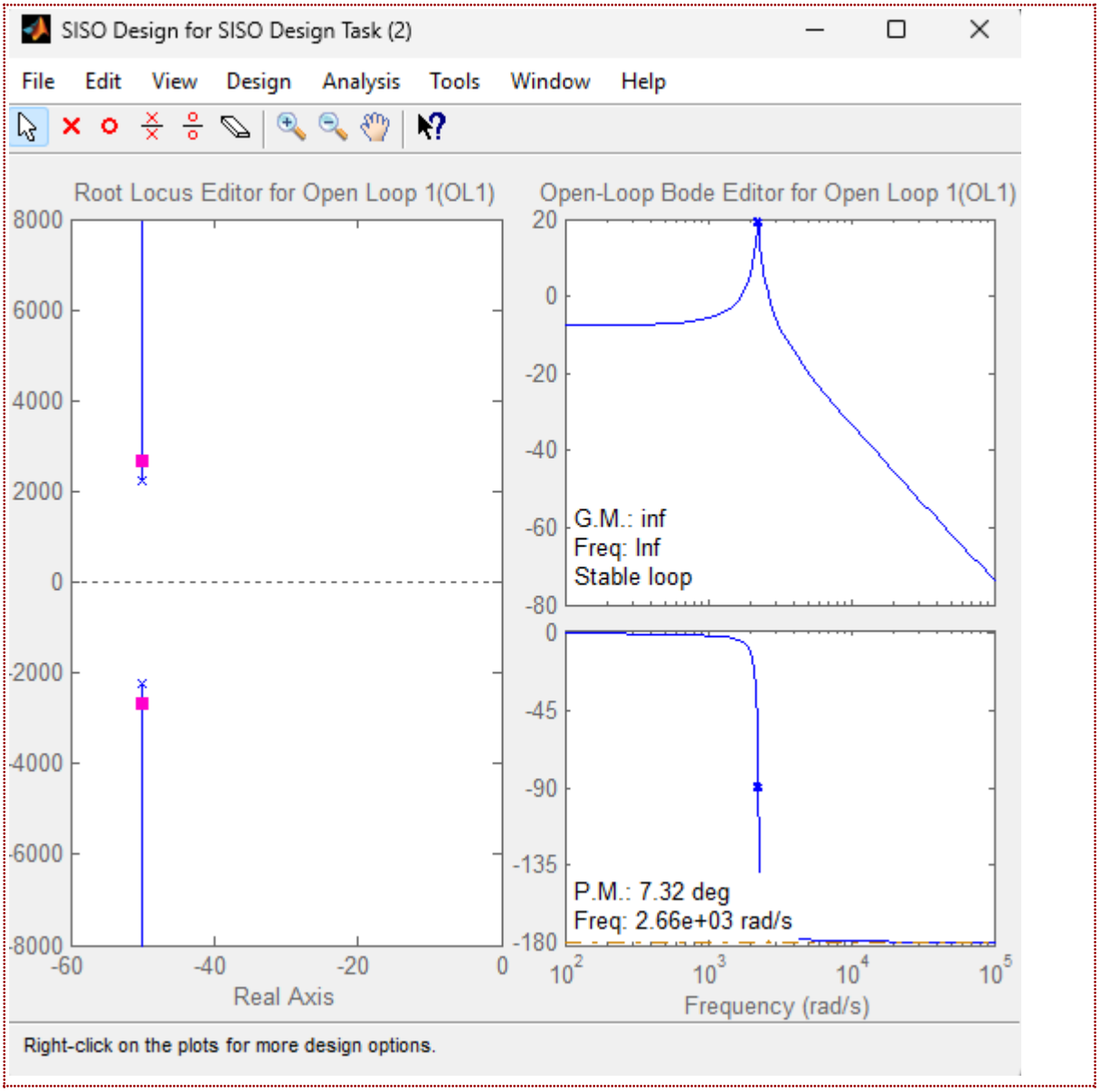
Criada uma conta para uso da versão Trial e instalado o Matlab, foram instalados os seguintes add-ons:

- Control System Toolbox
- Simulink
- Symbolic Math Toolbox.

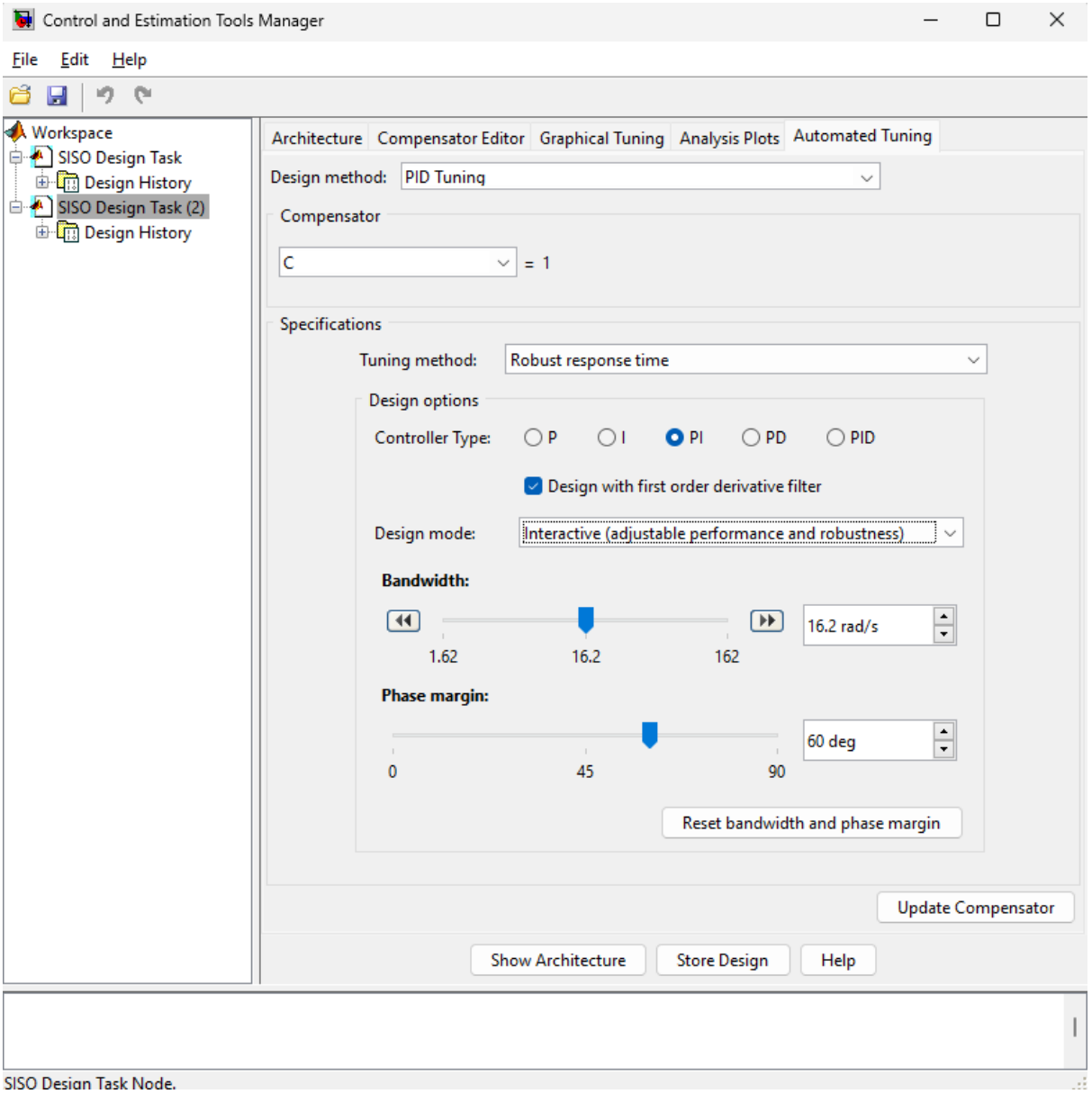
Após abrir o Matlab, foram inseridos os seguintes comandos na linha de comando:

```
syms s

R = 100;
L = 2 * 10^(-3);
C = 100 * 10^(-6);
D = 10/24;  % Fator de ganho

num_f = [0 0 R];
den_f = [R*L*C L R];
f = tf(num_f, den_f);
f = f * D;

sisotool(f)
```

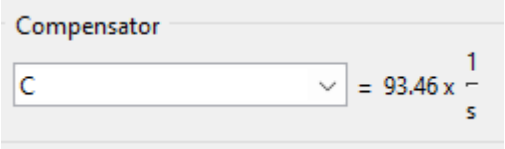
Na execução do `sisotool(f)`, é aberta a seguinte janela:



Clicando em “Tuning Method > PID Tuning”, selecione as seguintes opções e “Update compensator”:



Assim, temos o resultado abaixo:



Discretizando a ft do controlador com zoh:

```
num_c = [0 93.46];
den_c = [1 0];

c = tf(num_c, den_c);

cz = c2d(c, 0.001, 'zoh')
```

Temos o resultado:

```
>> cz = c2d(c, 0.001, 'zoh')

cz =

  0.09346
  -------
   z - 1

Sample time: 0.001 seconds
Discrete-time transfer function.
```

Transformando para expoente negativo:

$cz = (0.09346 * z^{-1})/(1 - z^{-1})$

Extraindo os valores necessários:

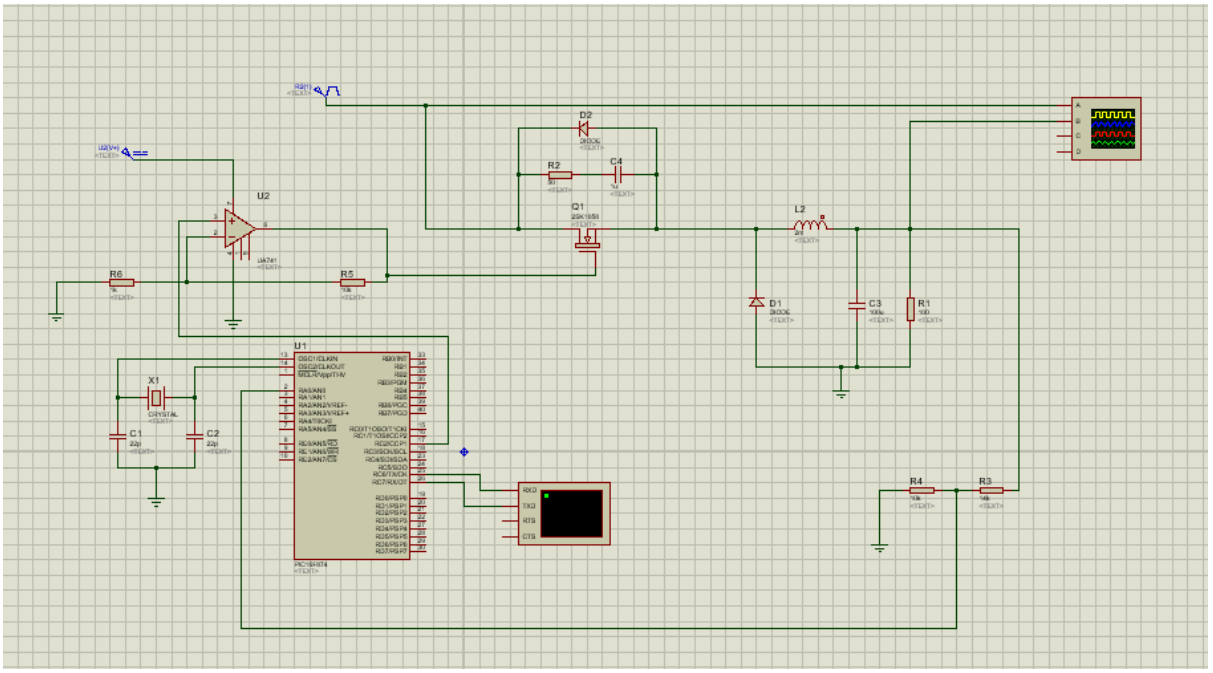
$a_1 = (0.09346 * z^{-1});\ b_1 = -1$

Usando a estrutura direta não canônica:

$u_k = a_1 * e_{k-1} - b_1 * u_{k-1} = 0.09346 * e_{k-1} + u_{k-1}$

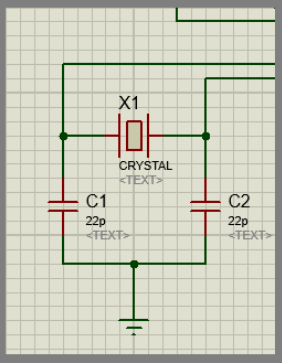
Esquemático no Proteus

>>> Print do circuito completo:

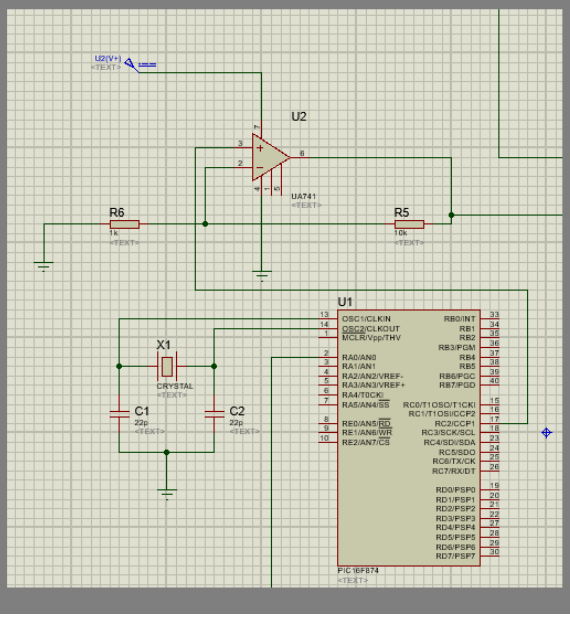


>>> Trechos do esquemático - Parte 01:

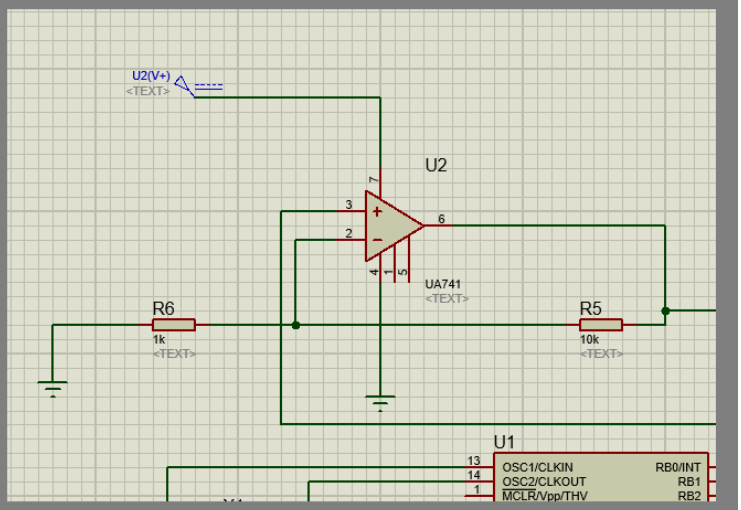
** CRISTAL ALIMENTADOR DO MICROCONTROLADOR 'PIC' 16F874*



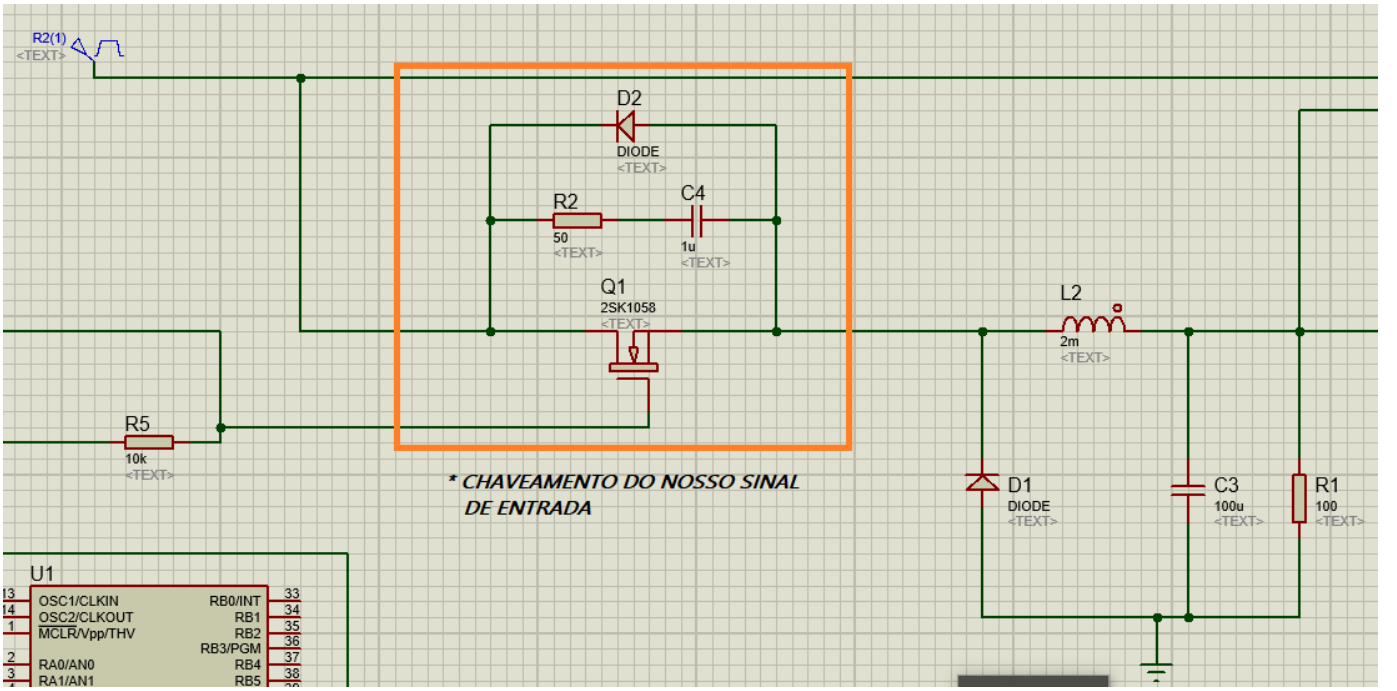
**TRECHO DO ESQUEMÁTICO RESPONSÁVEL POR CONFIGURAR O HARDWAARE DO PIC:*

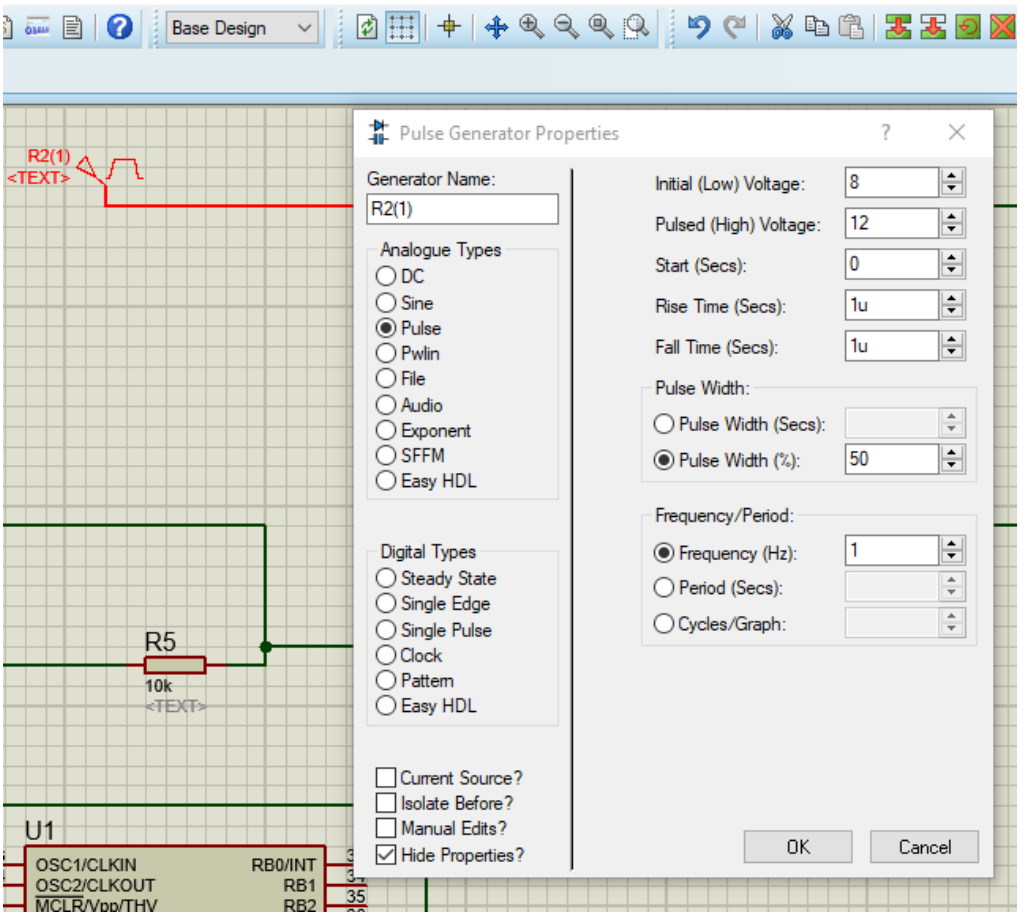


** AMP. OP ALIMENTADO POR DC DE 12V PARA ESTABILIZAR SINAL DE SAIDA CONTROLADOR (DO PIC) NA CHAVE MOSFET:*



>>> Trechos do esquemático - Parte 02:





(Obs. A frequência pedida foi de 1KHz mas utilizamos 1Hz na imagem acima para facilitar estudo do sinal no osciloscópio)

CÓDIGO (CCS C/Pcw.exe)

Na pasta PICC fornecida pelo professor utilizamos o compilador abaixo para o código em .C de configuração do PIC utilizado (Para gerar seu arquivo “.HEX” usado no esquemático do Proteus):

PCW C Compiler IDE

File Project Edit Options Compile View Tools Debug Help

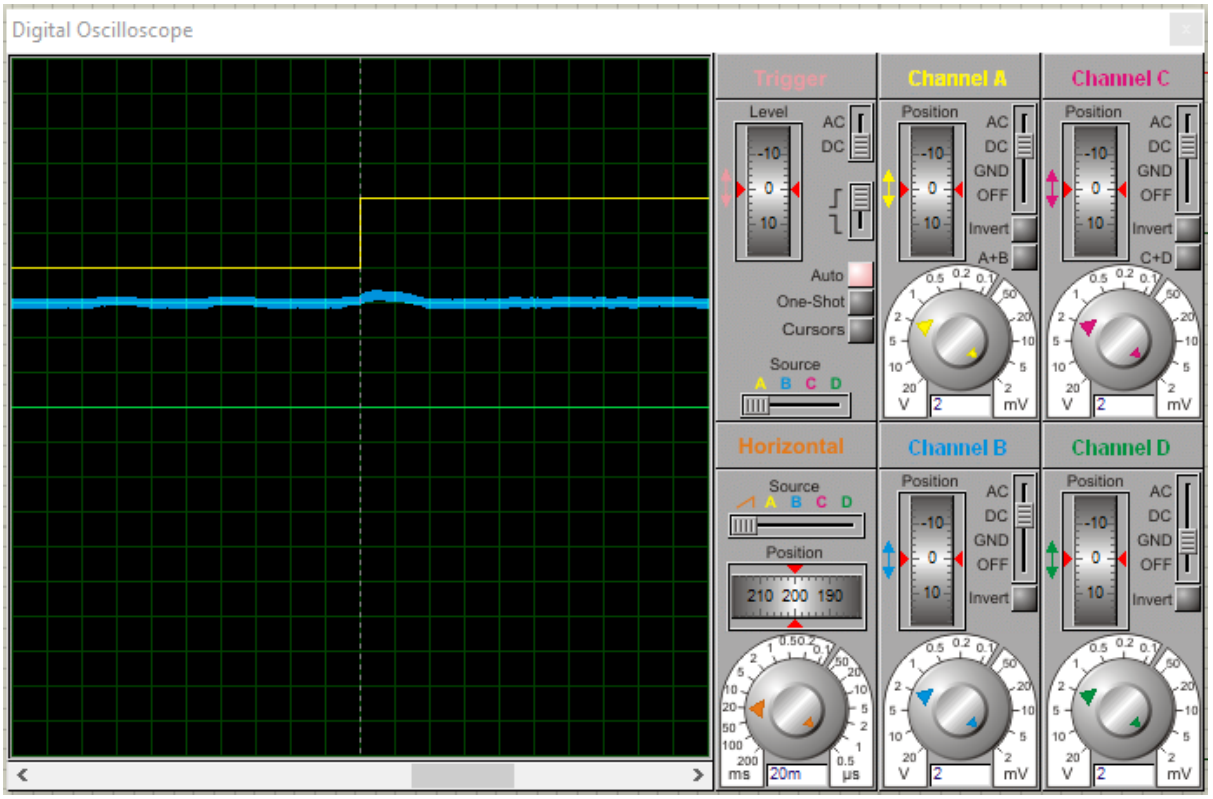
Microchip 14 bit

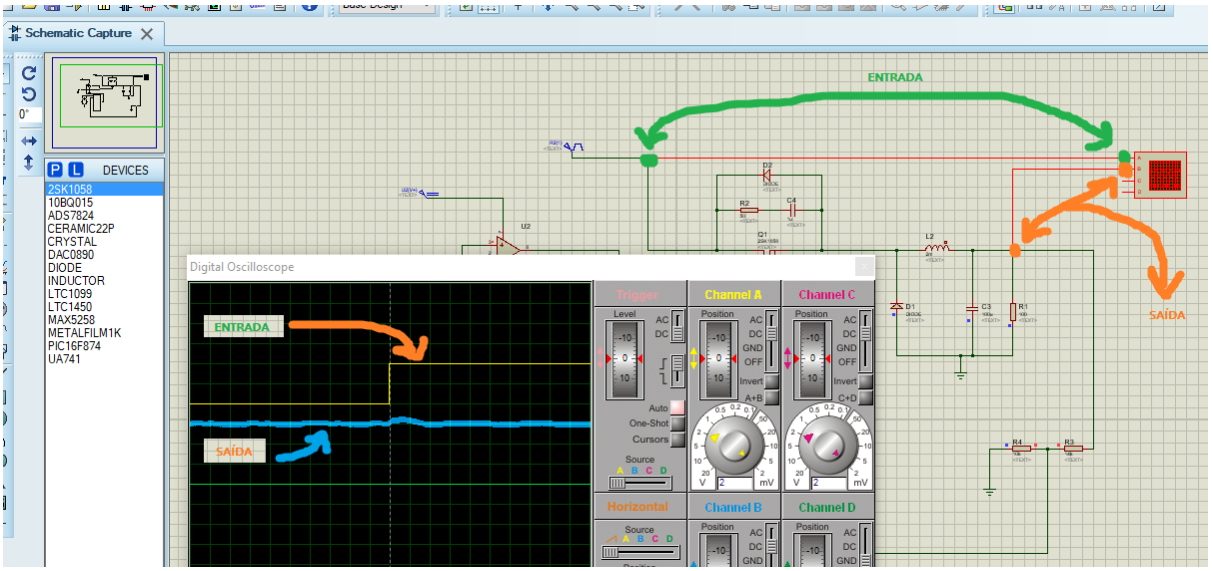
Codigo_Controlador_PI_Trab_N2_Aplic_Control_2022_2.c

```c
// Inclusão de bibliotecas e componentes utilizados
#include "16F874.h" // Precisa ser essa a biblioteca do controlador usado!
#device adc=8
#use delay(clock=1000000)
#fuses XT,NOWDT,NOPUT

void main()
{
   float ek=0, ek_1=0, uk=0, uk_1=0; // Instanciando variaveis utiliadas

   int out = 0, ref=127; // Instanciando variaveis utiliadas

   //**********************//
   float yk = 0;      //**// <---------- O MAIS IMPORTANTE PARA ESTABILIZAR O SINAL NO PROTEUS FOI ISSO AQUI!
   //**********************//            (Colocar o 'yk' para tipo 'float' kkkkk sim isso msm! Cuidado com os tipos da variáveis isso
                                      // pode afetar o resultado do circuito !!!)

   // "Setando" configurações iniciais dos componentes usados no PIC
   setup_adc_ports(ALL_ANALOG);
   setup_adc(ADC_CLOCK_INTERNAL);
   set_adc_channel(0);
   setup_timer_2(T2_DIV_BY_4,65,1); // *Importante ser 65 um dos parâmetros!
   setup_ccp1(CCP_PWM);

   enable_interrupts(GLOBAL | INT_TIMER2);
   delay_us(100);

   // Iniciando o laço
   while(1)
   {
      yk = read_adc();
      // yk = yk*(5000/255);
      ek = ref - yk;

      // Estrutura direta não canônica
      uk = 0.09346 * ek_1 + uk_1; // Coloque aqui os valores obtidos nos cálculos das equações!
      out = (int)(uk);

      // Tratando o sinal para se estabilizar em 6V (Que é o pedido no trabalho!)
      if (yk > 255){
         yk = 255;
      }
      if (yk < 0){
         yk = 0;
      }

      // Config pwm signal (Pedido no trabalho)
      set_pwm1_duty(out);


      // Estrutura direta não canônica (Continuando o ciclo no laço 'while true')
      uk_1 = uk;
      ek_1 = ek;

   }
}
```

17:1 C:\Users\Cristiano Coutinho\Desktop\Trab N2 Aplic Control 2022_2\Codigo_Controlador_PI_Trab_N2_Aplic_Control_2022_2.c

RESULTADOS CONSEGUIDOS NO OSCILOSCÓPIO (Obs. Os prints foram no Proteus 8 o professor geralmente usa o Proteus 7 Portable isso pode alterar os resultados e muito!)





**Fim // ~*

- *Muitas bençãos pro senhor!*

(Muito Obrigado por TUDO prof! Feliz Natal 🎅 e Fim de Ano 🥂 !!!)