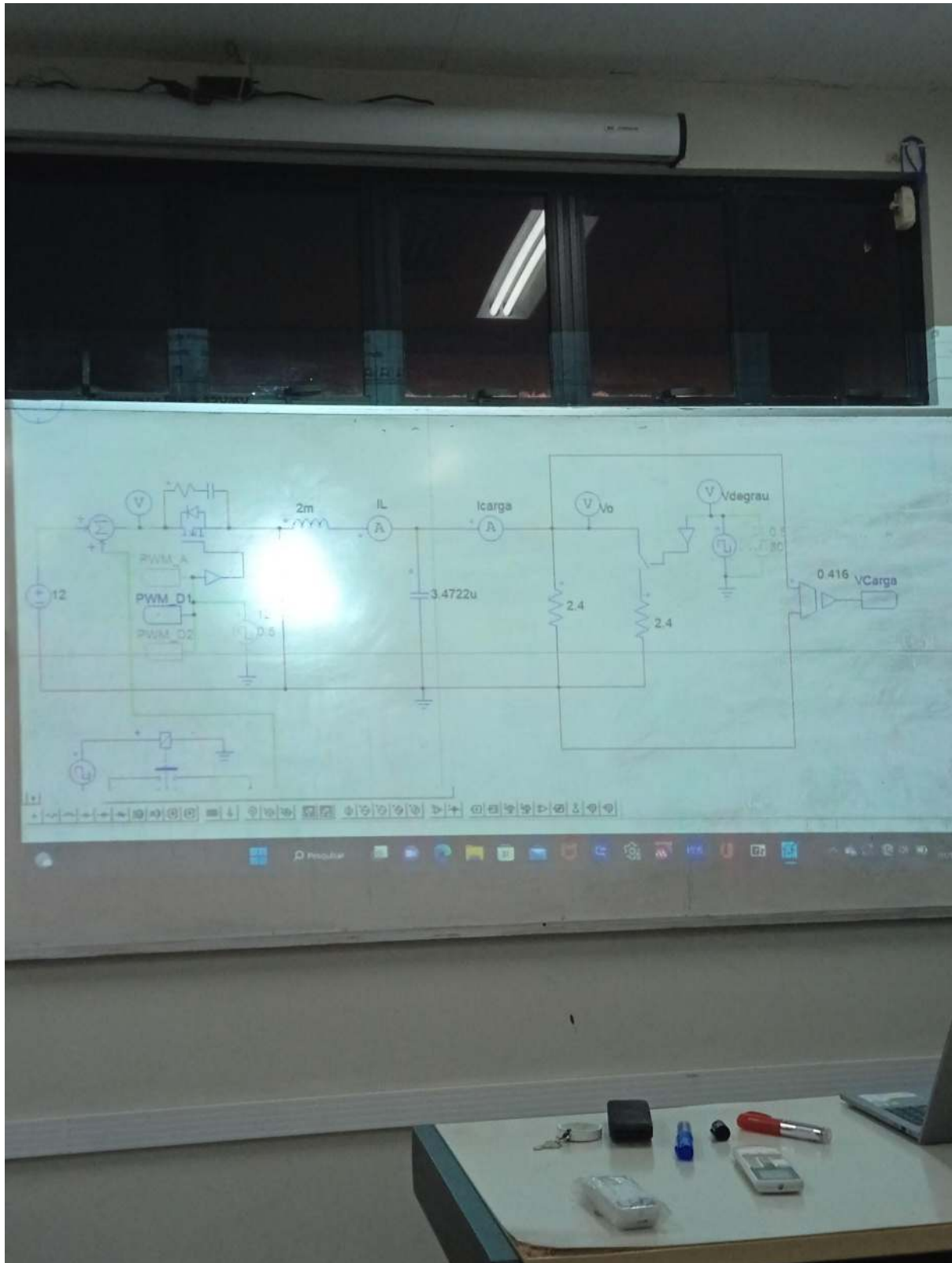
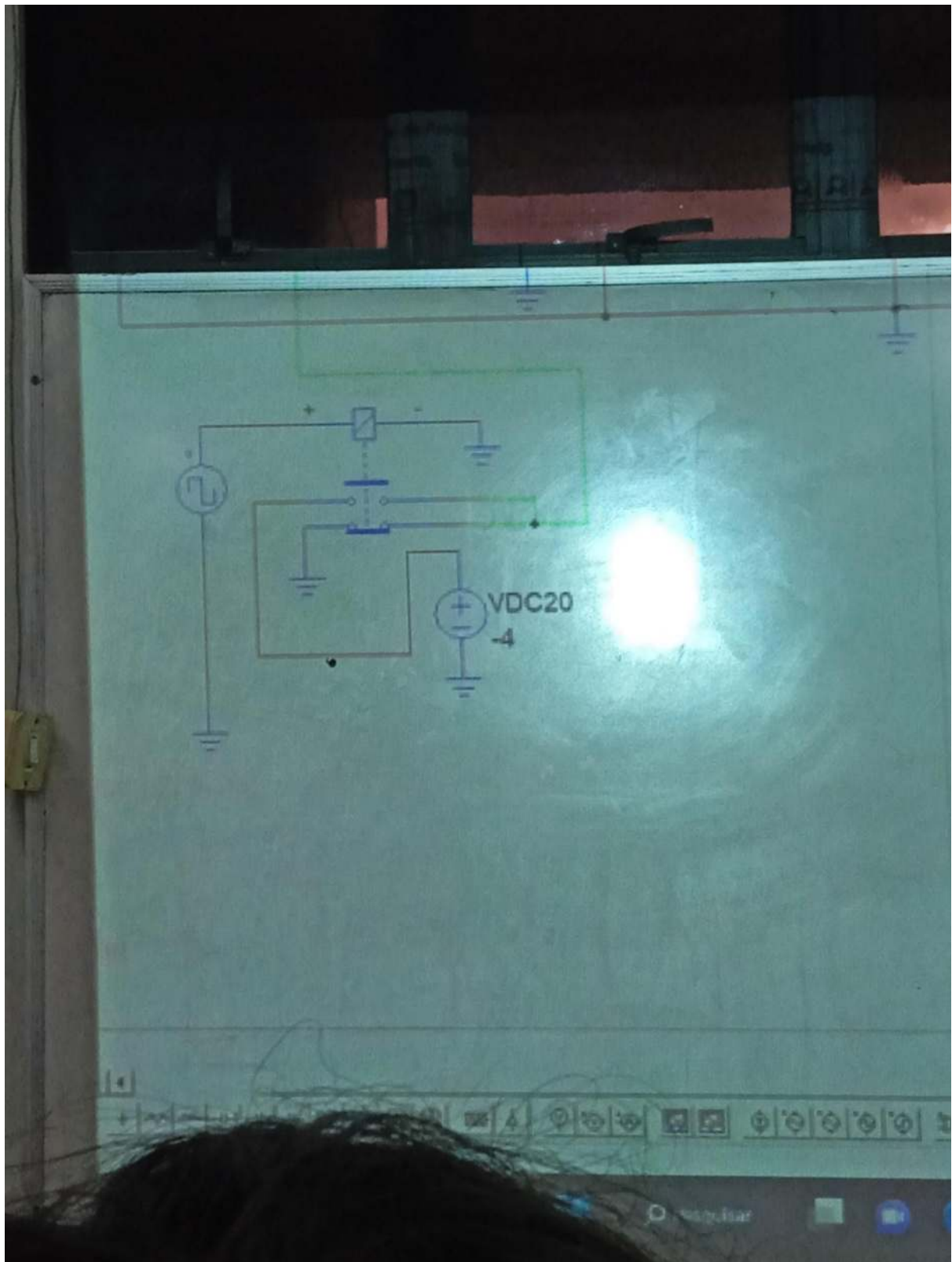


Psim. Questão de prova

Esquema eléctrico do conversor CC-CC

Trabalho, replicar no Proteus

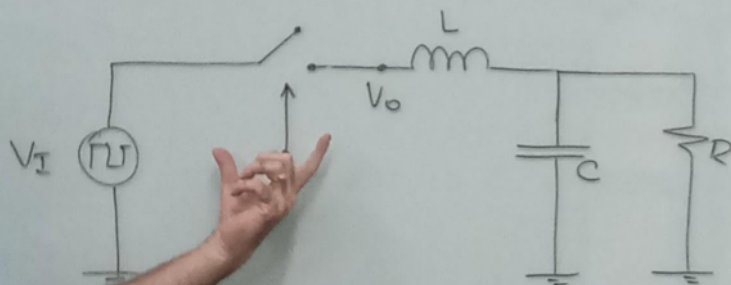




Representando no quadro o Conversor CC-CC

→ CONVERTER CC-CC:

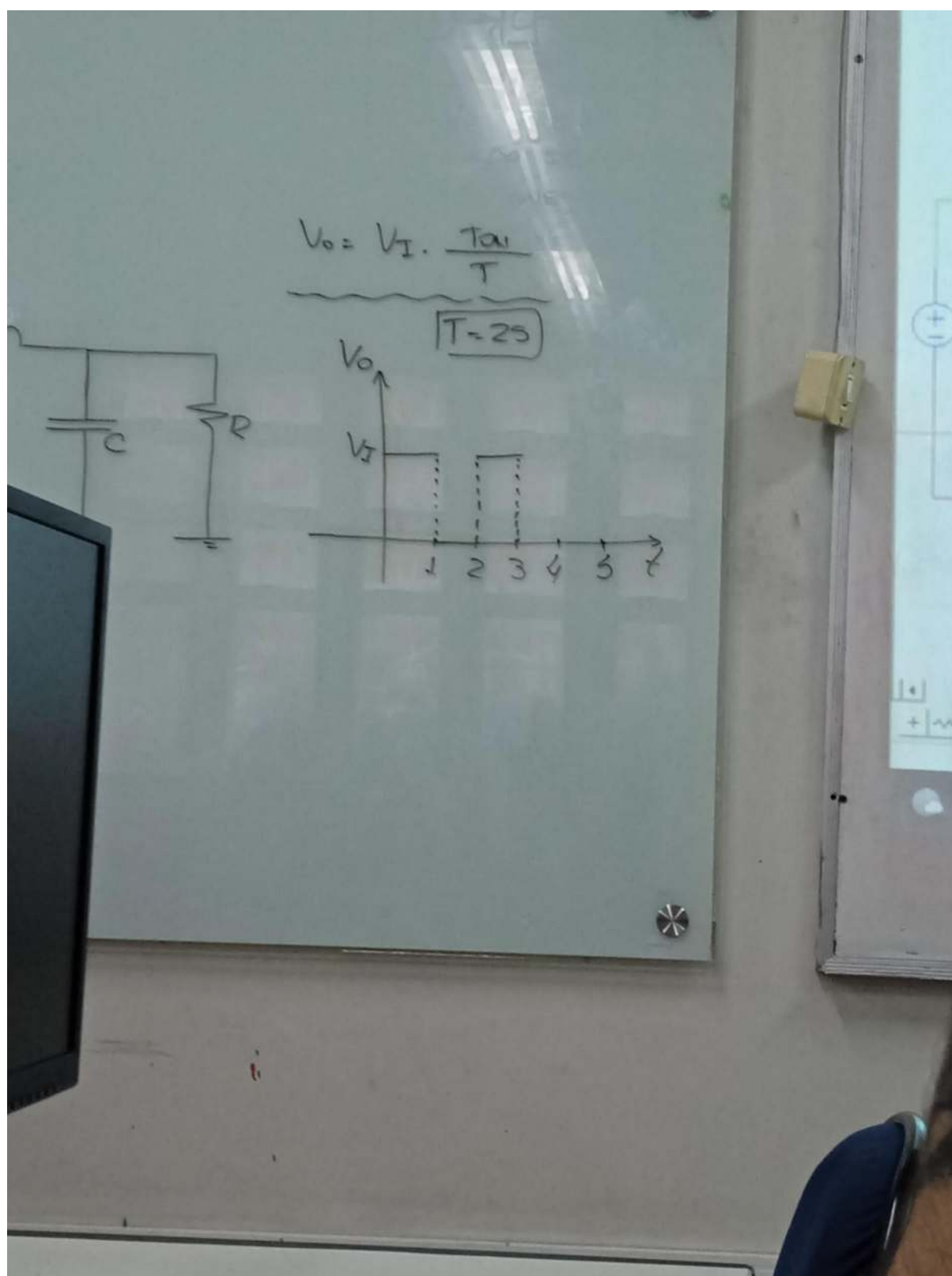
$$\underline{V_o = V_I \cdot \frac{T_{on}}{T}}$$



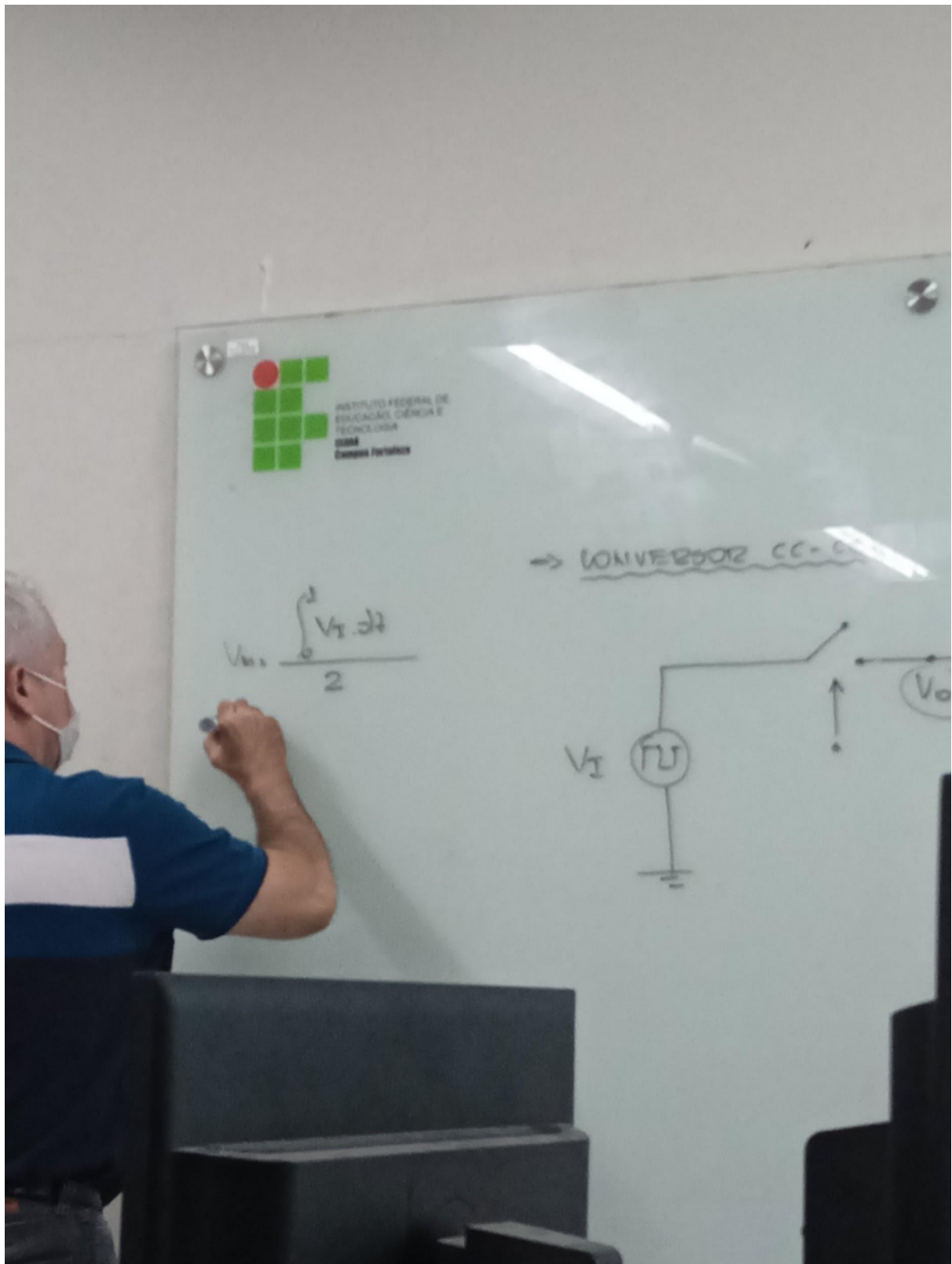
Calcular a tensão inicial

Ton: tempo que a chave fica fechado

Supondo que o  $T = 2s$

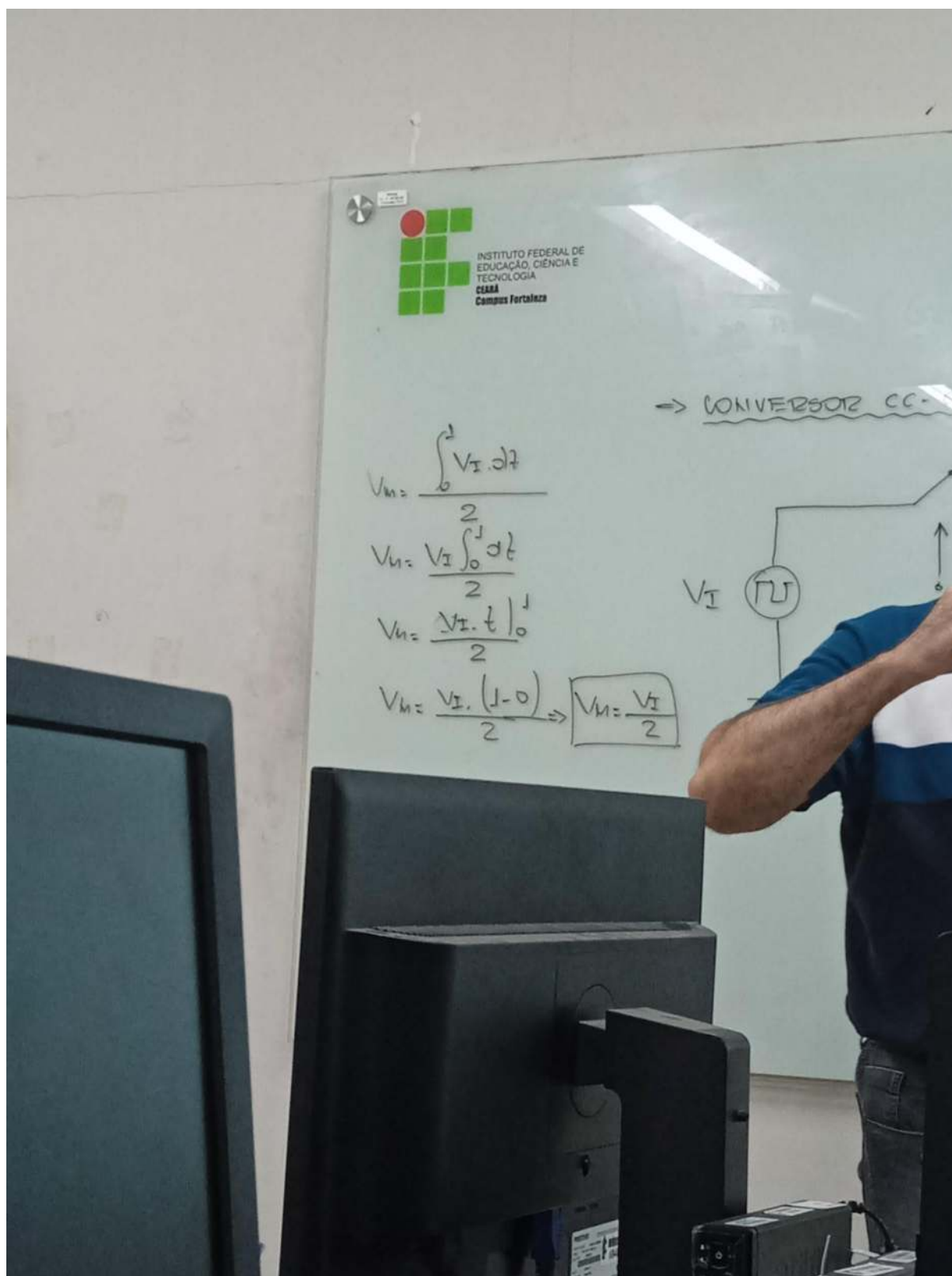


Calculando a tensão média:

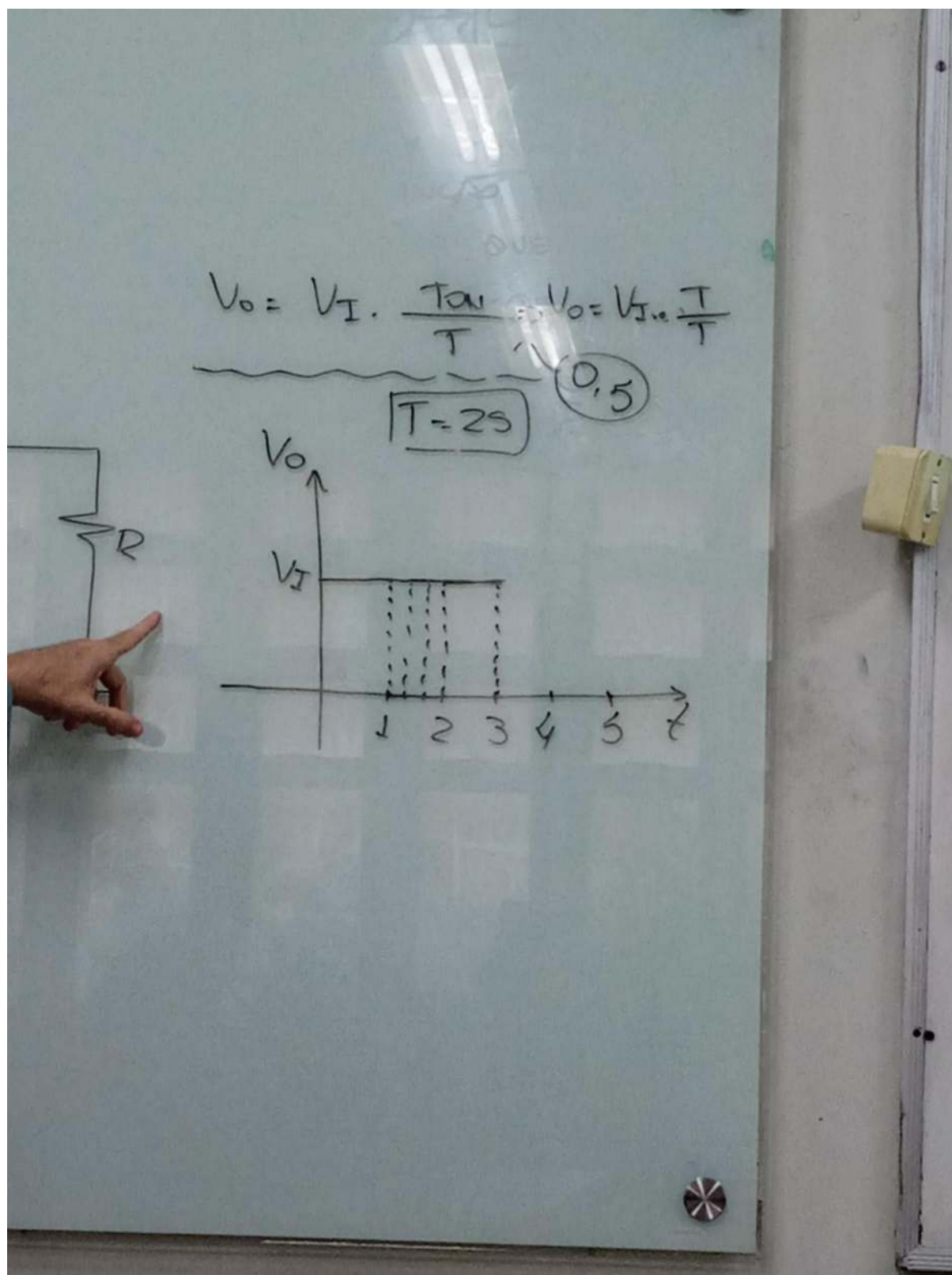


"Integral de 0 até 1"

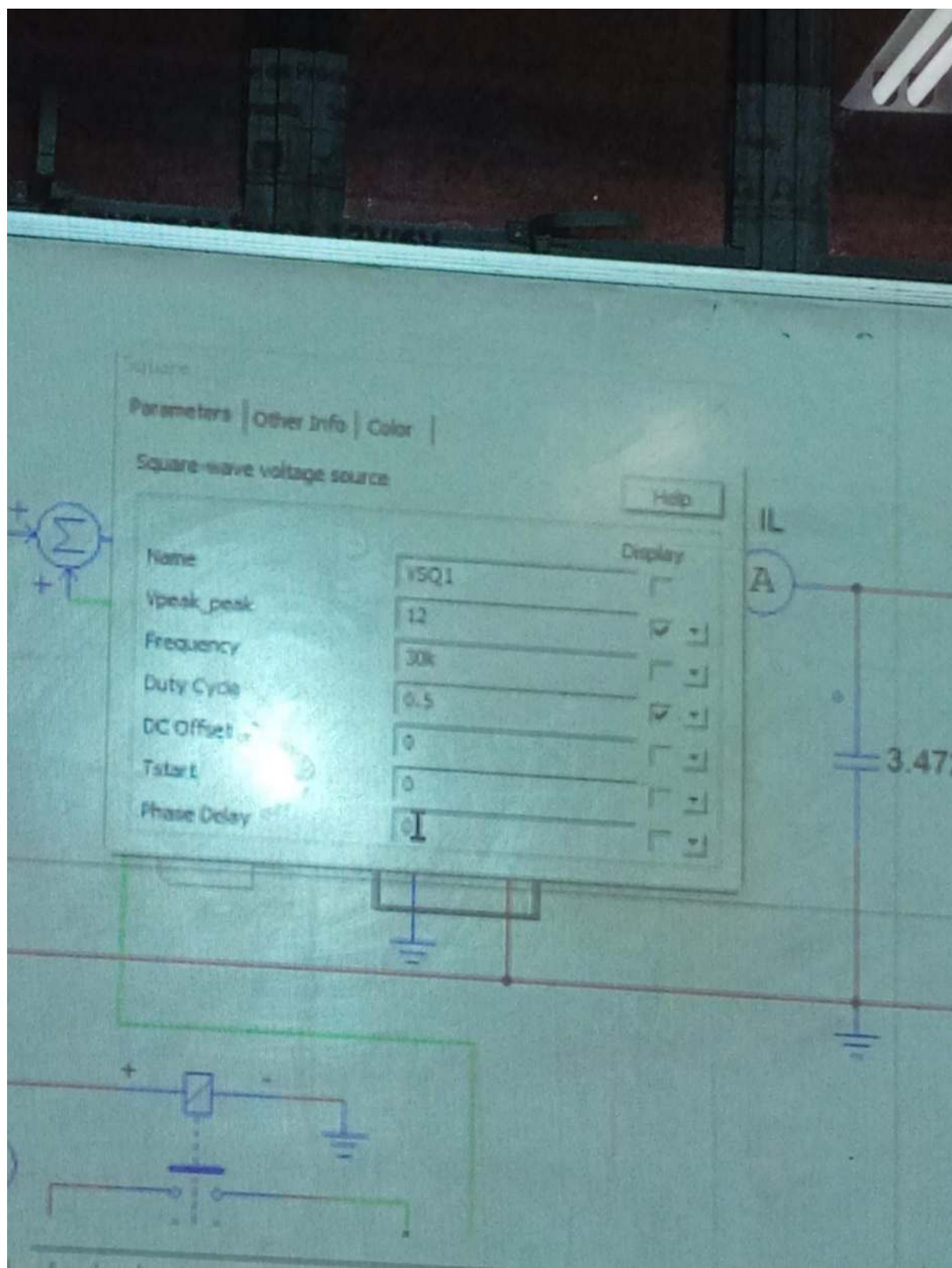




Considerando uma relação  $T_{on}/T = 0,5$ , onde  $T_{on} = 1$  e  $T = 2$



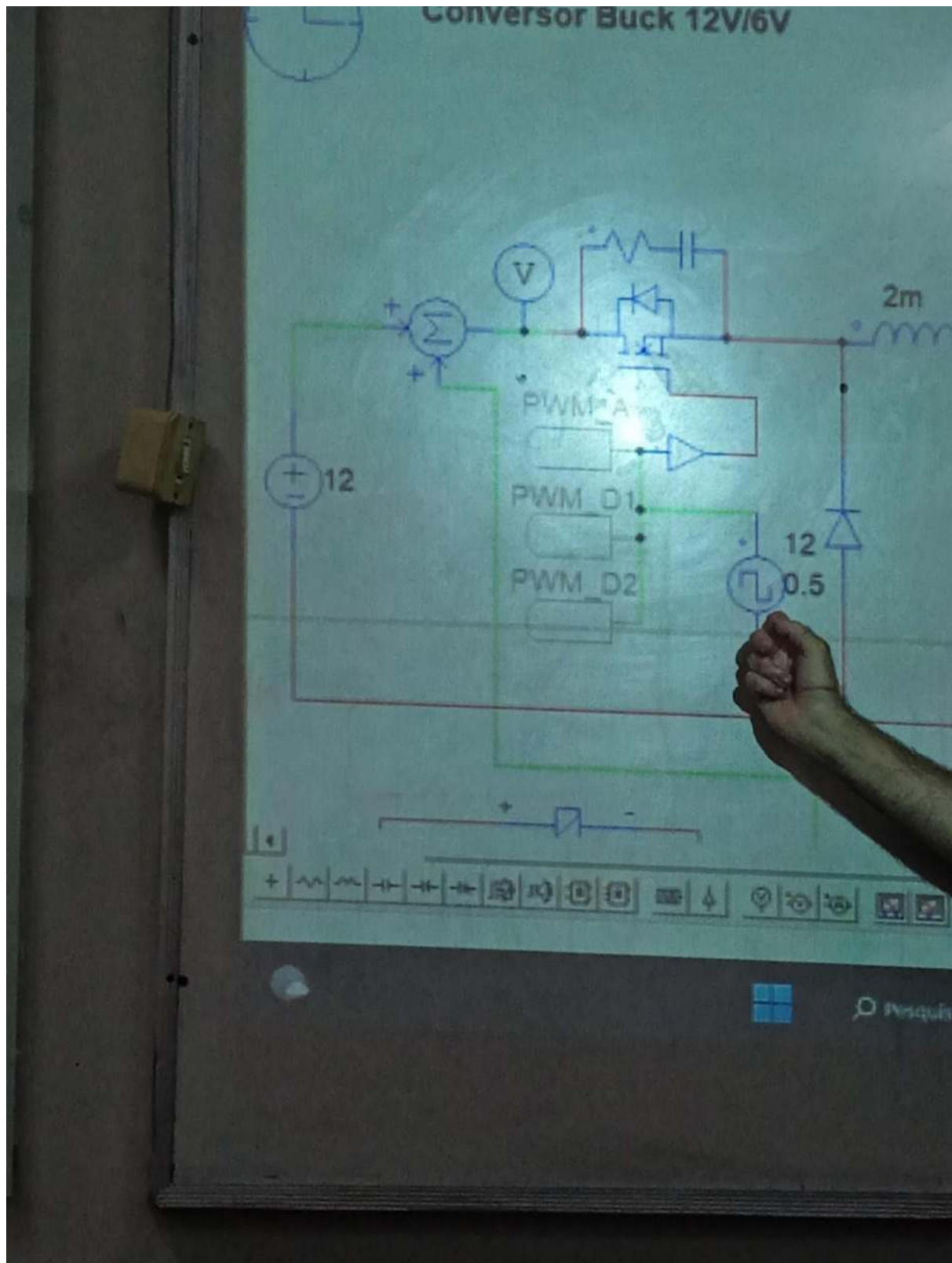
Abrindo a fonte do psim



Duty Cycle fixo em 0,5. Logo, será 50% do ciclo

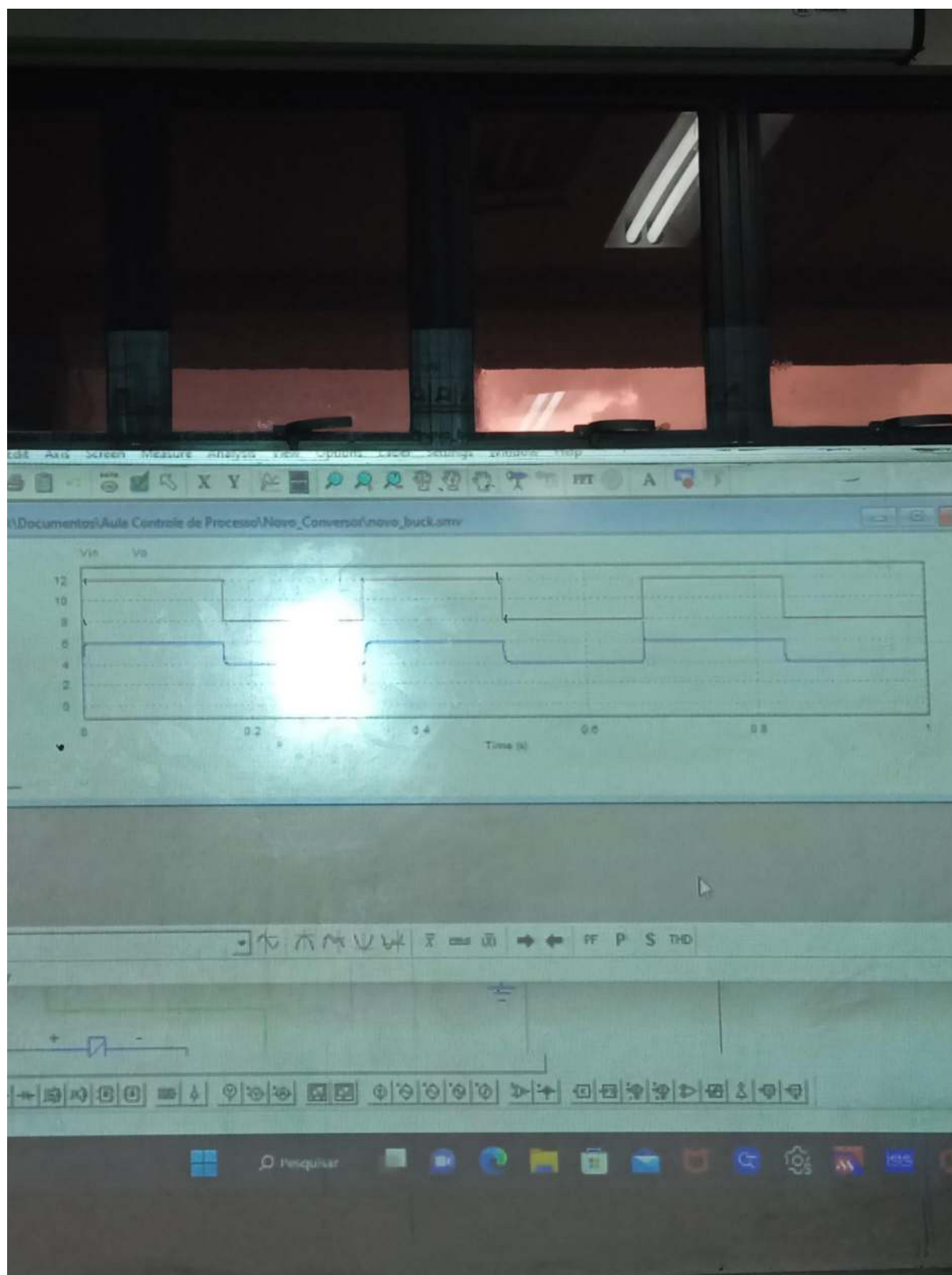
...





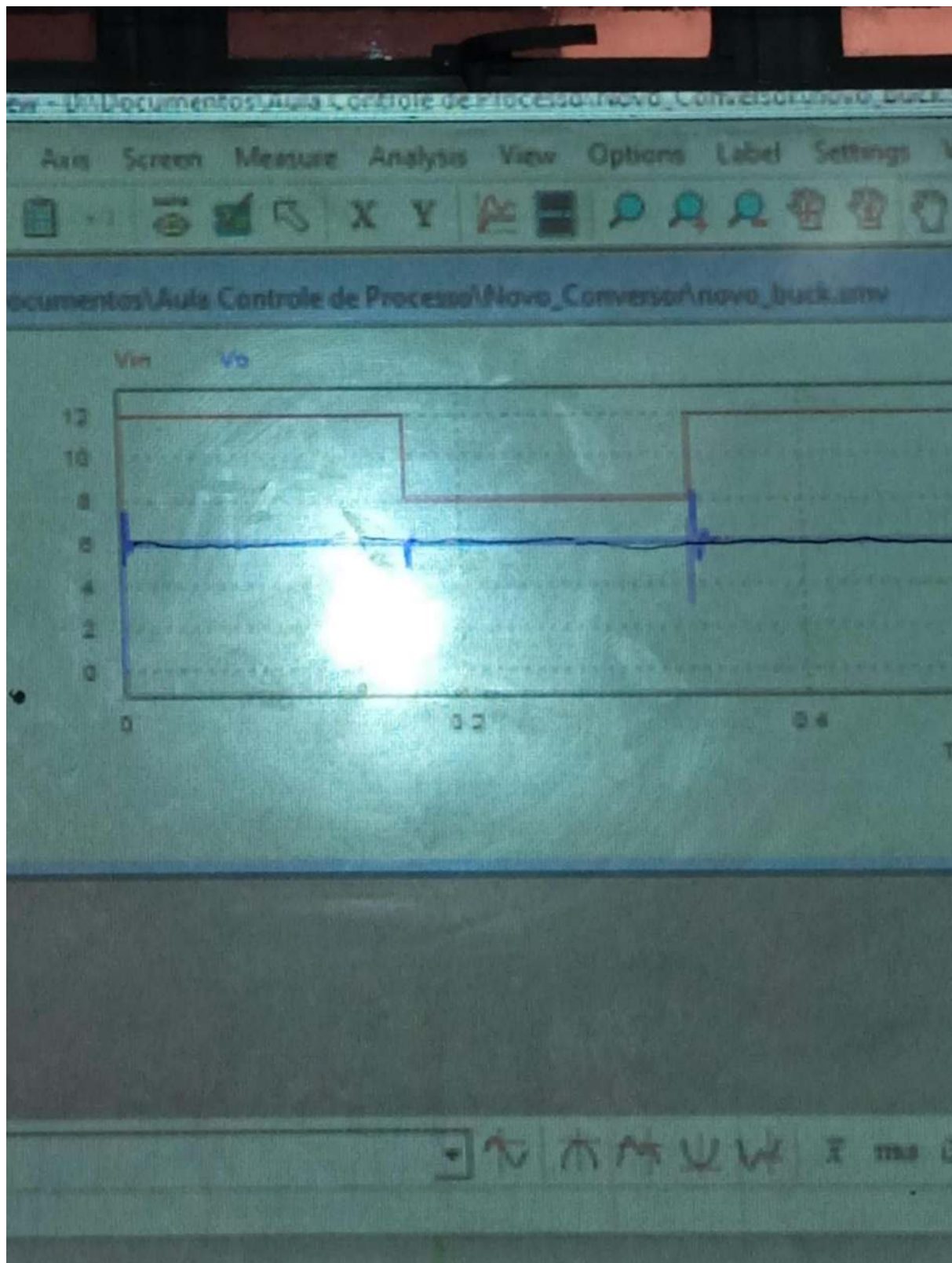
V fica alternando entre  $V = 12 + 0$  e  $V = 12 - 4$

Verificando a tensão



Desligando a fonte 0.5

O gráfico fica :



Estabilizou  $V_o$  em 6

Conceito do Controlador genérico, seja CC ou CA:

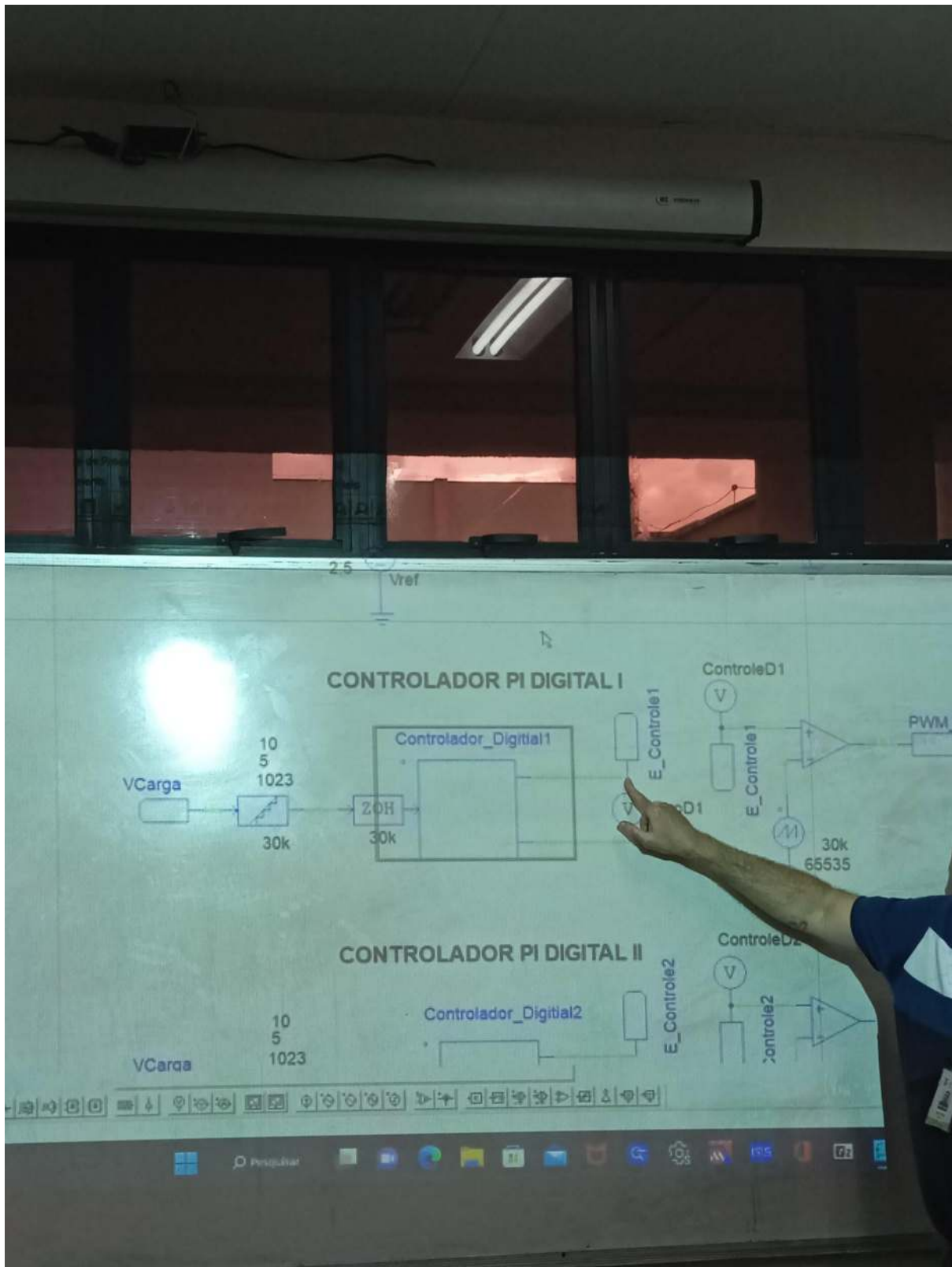
O controlador garante que a tensão de saída seja sempre um valor fixo, por exemplo, a saída seja 6V, logo garante que a saída será 6V independente da variação até a tensão de

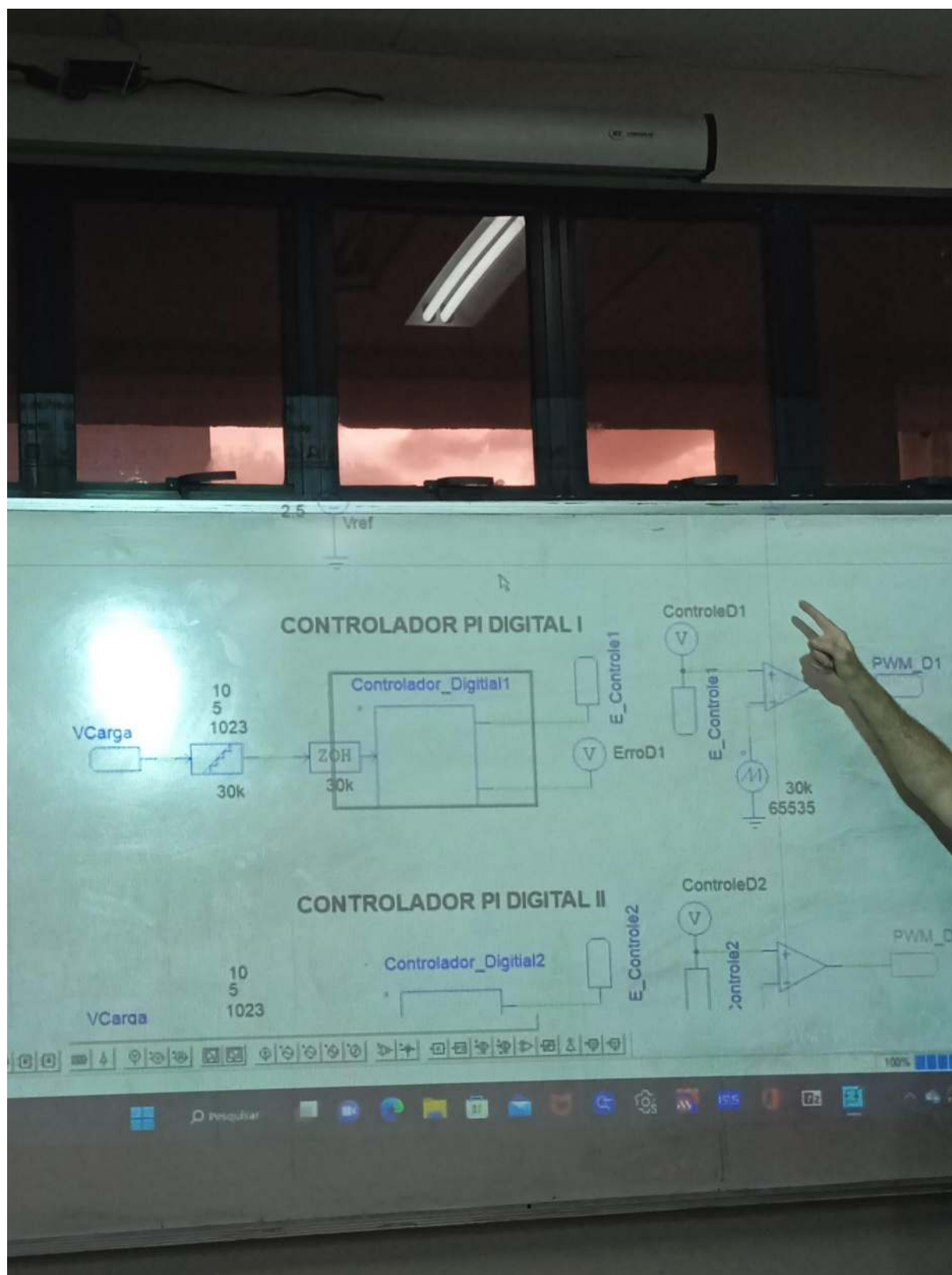
entrada seja 6V. Se for um controlador Buck-booster, a variação da tensão de entrada mesmo em 2V, ele garante 6V na saída. Exemplo o Power Bank

Controlador

Mostrando o circuito







Sinal PwM\_D1 será usado no Mosfet

Mostrando o código em C do controlador

# Simplified C Block

Parameters | Color

Simplified C Block

Help

Block

Name:

Number of Input/Output Ports

Inputs:

1

Outputs:

2

## C Code

Following variables are valid: 1

Input x1

Output y1, y2

```
static double ek=0, ek_1=0, uk=0, uk_1=0, rk_1=0, rk=0;  
static double ref=511; // Equivalente a um valor de tensão de 2.5V (Análogo)  
static float ts =0.000333;  
static float to;
```

```
ek = ref - x1;  
rk = ek + rk_1;  
uk = 1.67rk + 1.837rk_1;
```

```
if(uk>65535)
```

```
{  
    uk =65535;
```

```
if(uk<0)
```

```
{  
    uk=0;
```

```
}
```

Check Code

Block



Pesquisar



## Simplified C Block

Help

Block

Name:  ☒

Number of Input/Output Ports

Inputs

Outputs

C Code

Following variables are valid: t, delt

Input x1

Output y1, y2

```
#if(uk>65535)
{
    uk =65535;
}
#if(uk<0)
{
    uk=0;
}
y1=uk;
y2=-uk;
rk_1 = rk;
```

Edit Image

Check Code

Simplified C Block



Pesquisar

13h



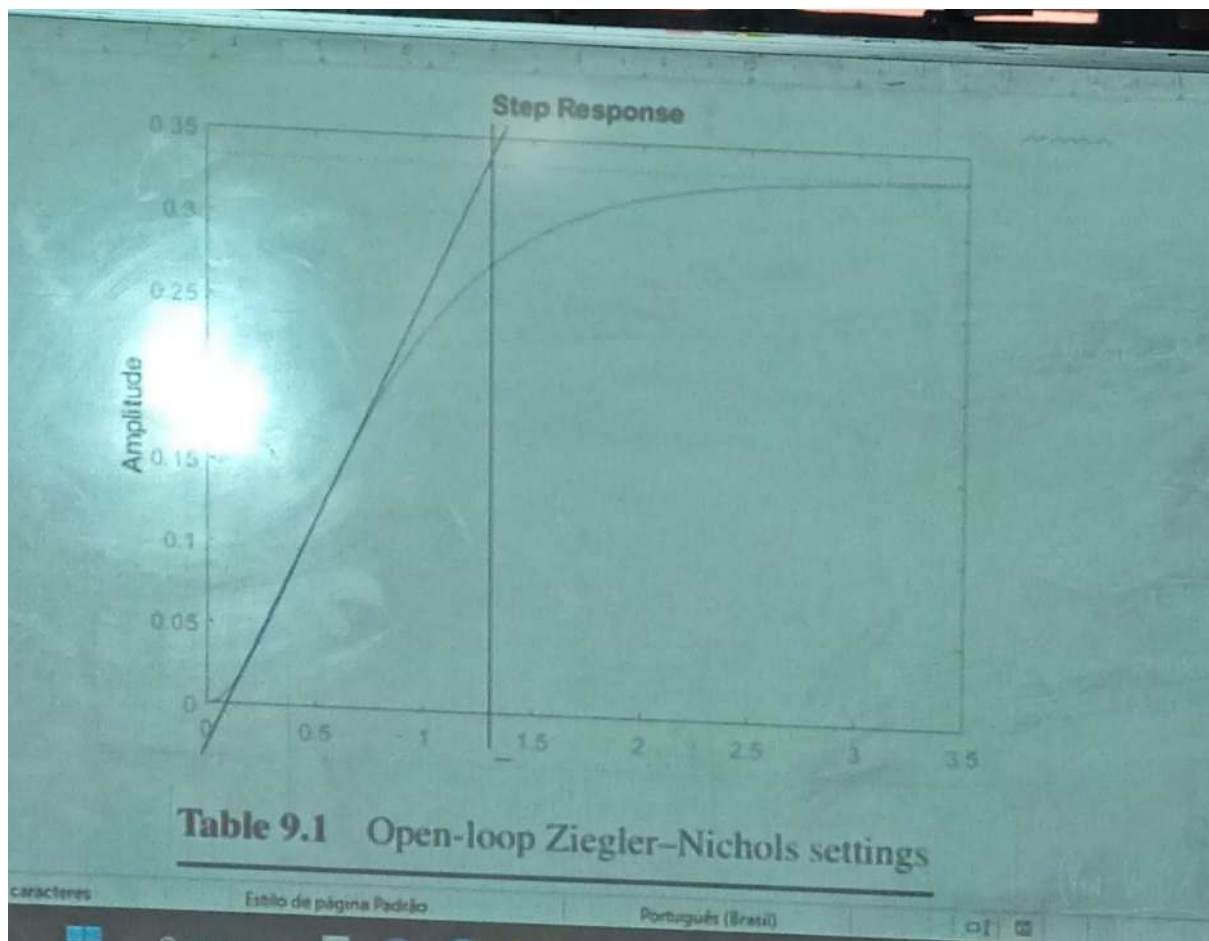
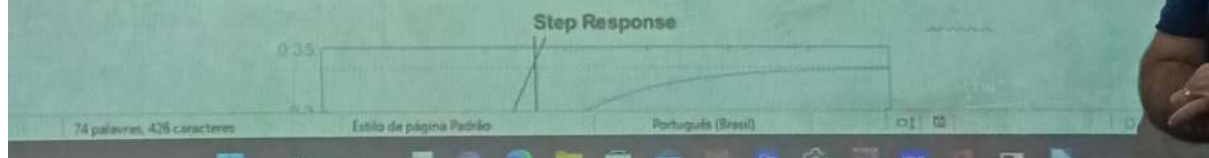
## Questão de prova:(CQP)

### Exercício

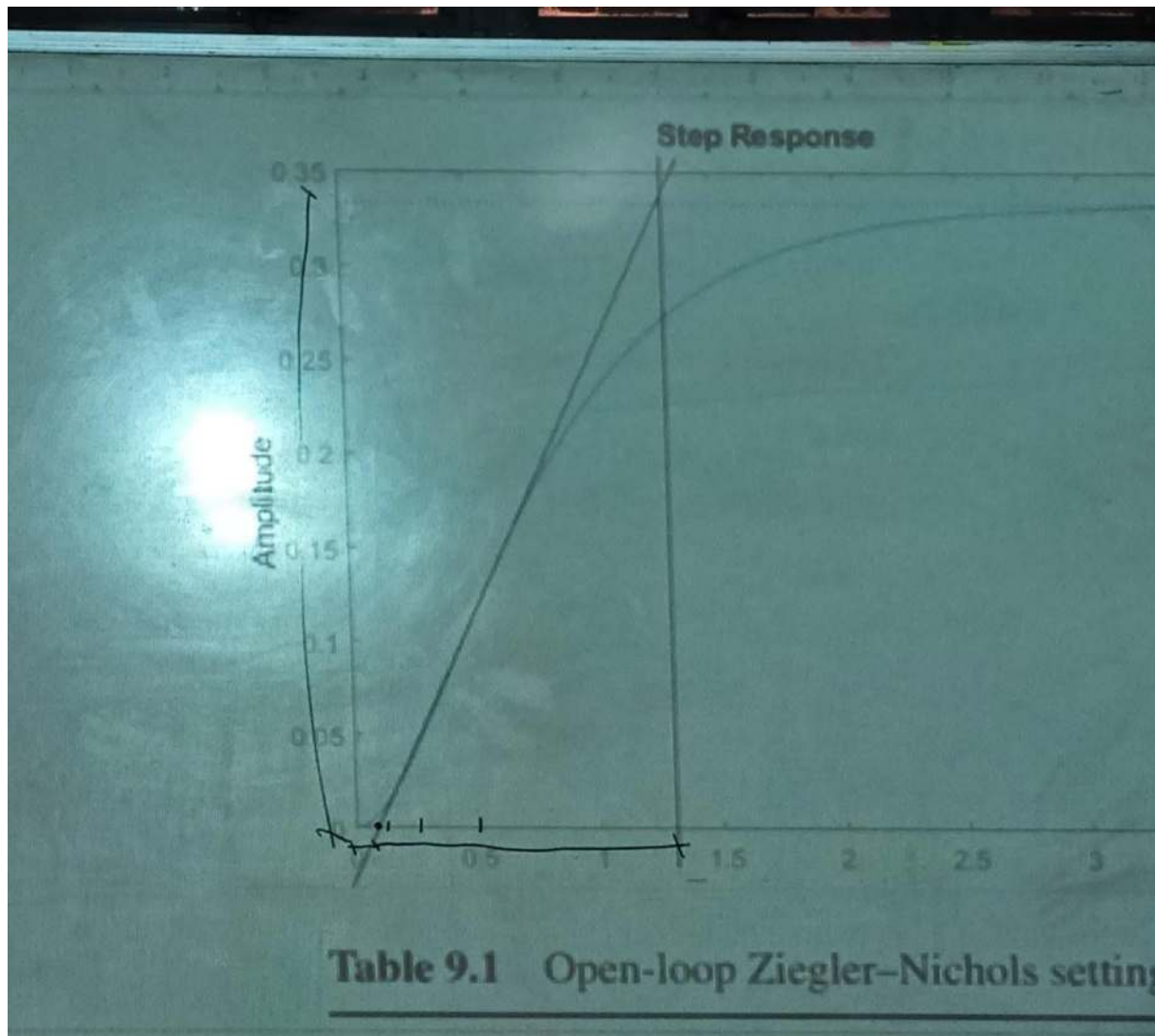
01. Elabore a rotina de interrupção de um programa, em linguagem C para PIC, de um controlador PID para a planta abaixo:

$$C(s) = 2/(s^2 + 5s + 6)$$

*Obs: a resposta à função degrau unitário aplicada à planta é apresentada abaixo. Aplique a regra de Ziegler-Nichols para determinar a FT do controlador PID no plano S:*



Usando o método Ziegler-Nichols



K é a linha pontilhada paralelo ao eixo X

$K = 0,34$

$TD = 0,12s$

$T1 = 1,35 - 0,12$

$T1 = 1,23s$

Projetando, com cálculos,  $K_p$ ,  $T_i$  e  $T_d$



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA  
CSAA  
Campus Fortaleza

$$\underline{K = 0,34}$$

$$\underline{T_D = 0,12s}$$

$$T_I = 1,35 - 0,12 \Rightarrow \underline{T_I = 1,23s}$$

$\Rightarrow$  CÁLCULOS DE  $K_P, T_I, T_D$ :

$$K_P = \frac{1,2 \times T_I}{K \times T_D} = \frac{1,2 \times 1,23}{0,34 \times 0,12} = 36,17$$

$$T_I = 2 \times T_D \Rightarrow T_I = 2 \times 0,12 \Rightarrow \underline{T_I = 0,24s}$$

$$T_D = 0,5 \times T_D \Rightarrow \underline{T_D = 0,5 \times 0,12 = 0,06s}$$

Com isso, podemos determinar a Função de Transferência (FT) do controlador PID, no plano S: e aplicando a transformada Z

⇒ TUNIÇÃO DE TRANSFERÊNCIA DO  
CONTROLADOR PID NO PLANO S:

$$D(s) = K_p + \frac{K_p}{T_i \cdot s} + K_p \cdot T_d s$$

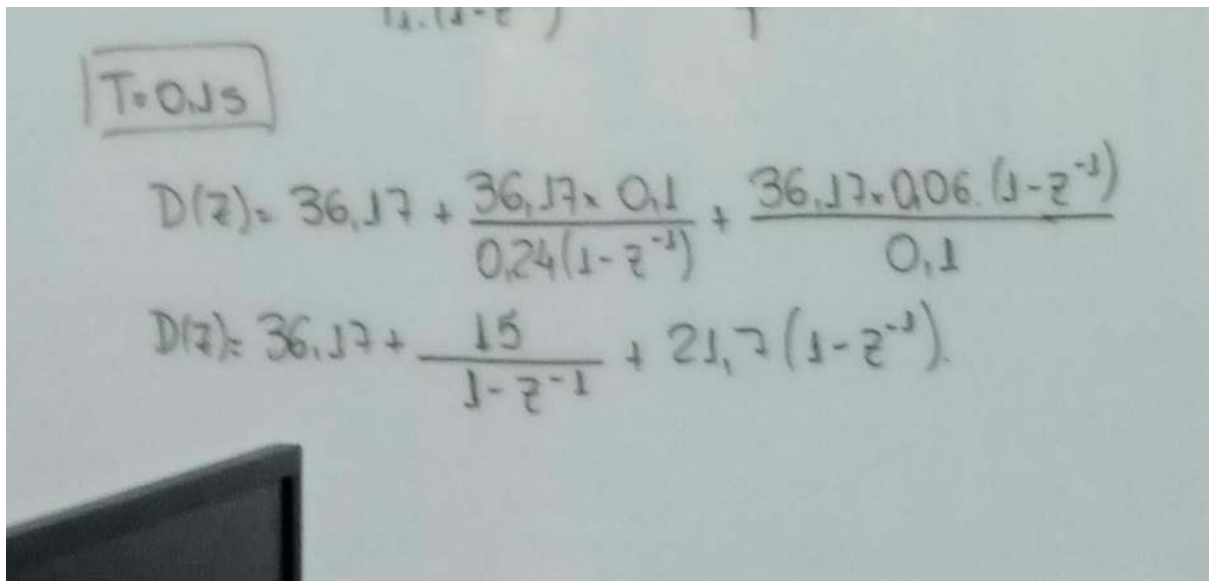
---

APLICANDO A TRANSFORMADA Z:

$$D(z) = K_p + \frac{K_p \cdot T}{T_i \cdot (1 - z^{-1})} + \frac{K_p \cdot T_d \cdot (1 - z^{-1})}{T}$$



Adotando  $T=0,1s$ :



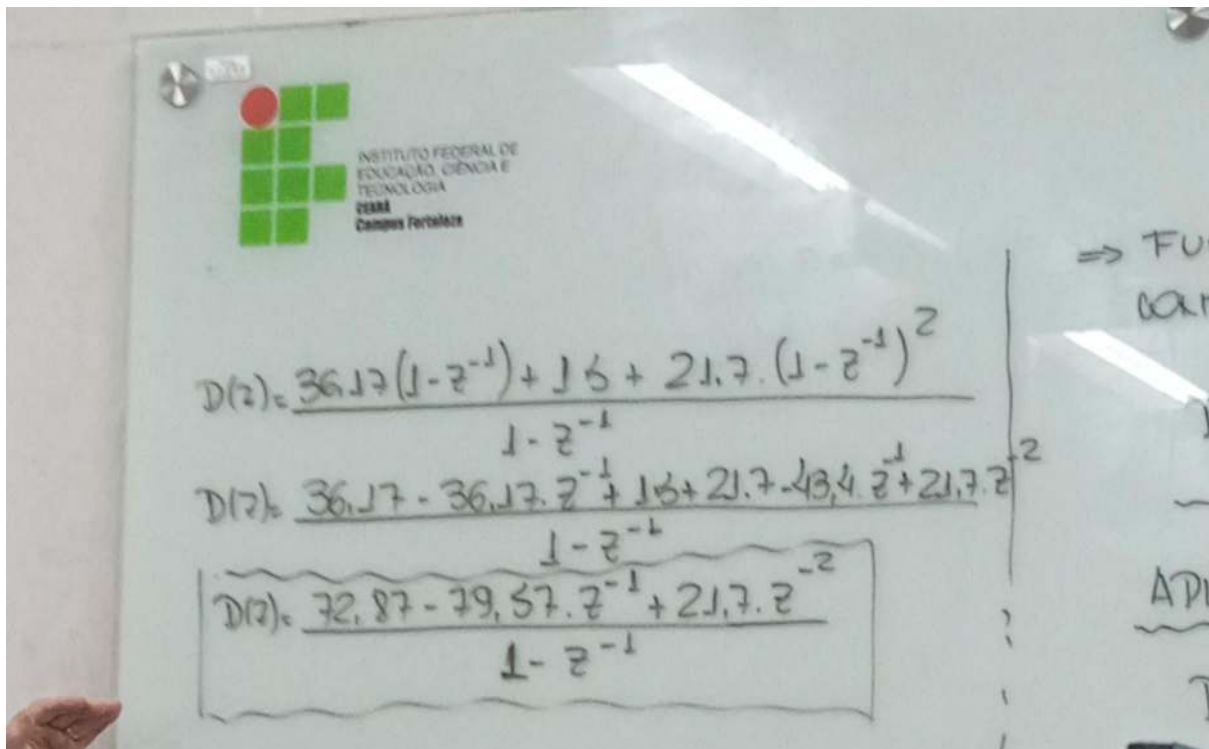
Handwritten equations on a whiteboard:

$$T=0,1s$$

$$D(z) = 36,17 + \frac{36,17 \times 0,1}{0,24(1-z^{-1})} + \frac{36,17 \times 0,06(1-z^{-1})}{0,1}$$

$$D(z) = 36,17 + \frac{15}{1-z^{-1}} + 21,7(1-z^{-1})$$

Dando sequência:



Handwritten equations on a whiteboard:

$$D(z) = \frac{36,17(1-z^{-1}) + 15 + 21,7(1-z^{-1})^2}{1-z^{-1}}$$

$$D(z) = \frac{36,17 - 36,17z^{-1} + 15 + 21,7 - 43,4z^{-1} + 21,7z^{-2}}{1-z^{-1}}$$

$$D(z) = \frac{72,87 - 79,57z^{-1} + 21,7z^{-2}}{1-z^{-1}}$$

$a_0 = 72,87$   
 $a_1 = -79,57$   
 $a_2 = 21,7$   
 $b_1 = -1$

Determinar as equações de diferença, utilizando as estrutura Direta Canônica para descobrir o  $r_k$  e o  $u_k$ , temos:

$$D(z) = \frac{36,17(1-z^{-1}) + 16 + 21,7 \cdot (1-z^{-1})^2}{1-z^{-1}}$$

$$D(z) = \frac{36,17 - 36,17 \cdot z^{-1} + 16 + 21,7 \cdot 43,4 \cdot z^{-1} + 21,7 \cdot z^{-2}}{1-z^{-1}}$$

$$D(z) = \frac{72,87 - 79,57 \cdot z^{-1} + 21,7 \cdot z^{-2}}{1-z^{-1}}$$

$$\begin{cases} a_0 = 72,87; & a_1 = -79,57; & a_2 = 21,7 \\ b_1 = -1 \end{cases}$$

UTILIZANDO A ESTRUTURA DIRETA  
CANÔNICA:

$$r_k = e_k - b_1 \cdot r_{k-1};$$

$$u_k = a_0 \cdot r_k + a_1 \cdot r_{k-1} + a_2 \cdot r_{k-2}.$$

Montando as duas equações de diferença, temos:

$$\begin{cases} r_k = e_k + r_{k-1}; \\ u_k = 72,87 \cdot r_k - 79,57 \cdot r_{k-1} + 21,7 \cdot r_{k-2}; \end{cases}$$

Obs: Agora vamos programar, em C, a rotina de interrupção. SOMENTE A ROTINA, não precisa do código inteiro.

O código da Rotina de interrupção é representado por:

```

VOID CONTROL-PID() {

    VALOR = READ-ADC();
    VALOR = VALOR *  $\left(\frac{5000}{255}\right)$ ; // mV

    ek = Sk - VALOR;
    rk = ek + rk-1;
    uk = 72.87 * rk - 79.57 * rk-1 + 21.7 * rk-2;
    yk = (byte)  $\left(u_k * \frac{255}{5000}\right)$ ;

    OUTPUT-B(yk);
    OUTPUT-LOW(PIN-CO);
    OUTPUT-HIGH(PIN-CO);
    rk-2 = rk-1;
    rk-1 = rk;
}

```

Lembrando de configurar o Timer em 0,1s

// Finalizado a questão da prova

Obs: S<sub>k</sub> = 2500 mV

Lembre de inicializar as variáveis em 0.