

**PBLE01**  
*Co-projeto de produtos eletrônicos*

**Manual**  
Grupo 6

## Sumário

1	Identificação dos integrantes .....	3
2	Introdução .....	4
3	Esquema elétrico.....	5
3.1	Requisitos .....	5
3.2	Circuito.....	7
3.3	Esquemas elétricos .....	8
3.3.1	Unidade de processamento.....	8
3.3.2	Alimentação .....	9
3.3.3	Relógio de Tempo Real .....	10
3.3.4	Teclado .....	11
3.3.5	Sinalização .....	12
3.3.6	Entrada Analógica .....	13
3.3.7	Entradas diferenciais .....	14
3.3.8	Saída Analógica.....	15
3.3.9	Barras de expansão.....	16
3.3.10	Comunicação serial .....	17
3.3.11	Visor LCD .....	18
3.4	Memorial de cálculos .....	19
3.4.1	Entradas diferenciais .....	19
3.4.2	Saída Analógica.....	20
3.5	Relatório de verificação de erros de projeto.....	21
4	Placa de circuito impresso .....	22
4.1	Requisitos .....	22
4.2	Desenho da placa de circuito impresso .....	23
4.3	Visão tridimensional do circuito projetado .....	24
4.4	Relatório de verificação de erros de projeto.....	27
5	Especificações elétricas .....	28
5.1	Níveis de alimentação suportados .....	28
5.2	Faixa de níveis de consumo estimadas.....	28
5.3	Correntes e tensões máximas de entrada.....	28
5.4	Correntes e tensões máximas de saída .....	28
6	Programa embarcado de validação .....	29
7	Montagem.....	30
7.1	Lista de compras.....	30
7.2	Análise de Custos .....	31
8	Anexos.....	32
8.1	Código-fonte do programa de validação .....	32
8.2	Programas utilizados .....	45
9	Bibliografia.....	46

# 1 Identificação dos integrantes

André Luís Oliveira de Almeida – 2019008624

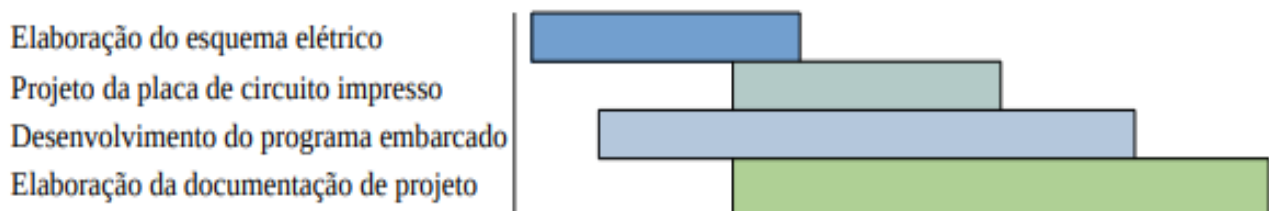
João Roberto da Silva Couto - 2018000944

Lucas Ferreira Machado - 2019000537

## 2 Introdução

O objetivo da disciplina era certamente nos incluir no ramo do mercado de trabalho para projetos, protótipos e lançamentos de PCIs (placas de circuito impresso) e ideias correlacionadas, por promover uma experiência de mercado de trabalho, com o maior número possível de detalhes ao ponto que a disciplina em si não ensina quase nada novo, pois foi visto o que se requiere no projeto em períodos anteriores.

As principais atividades a serem desenvolvidas durante o transcorrer da disciplina são o projeto da placa de circuito impresso, o desenvolvimento de programa embarcado de validação e a elaboração da documentação de projeto, etapas que não devem seguir um planejamento de inícios e terminos, com etapas sendo discutidas e aprimoradas ao mesmo tempo, como se observa no desenho abaixo:



*Figura 1 - Diagrama de avanço de tarefas*

A etapa de projeto da placa de circuito ainda pode ser subdividida nas etapas de elaboração do esquema elétrico e de elaboração da PCI.

a documentação de projeto, a etapa que aqui vos instrui, a ser desenvolvido consiste em um manual a partir do qual é possível confeccionar e validar a placa de circuito impresso projetada. Para tanto, tal documento deve pormenorizar características do circuito elaborado, como seu esquema elétrico, suas características elétricas, os passos para sua validação bem como a relação de seus componentes e os fornecedores para estes.

Dizem que o engenheiro é aquele que aplica os conhecimentos dos cientistas, então este projeto, e consequentemente esta disciplina é a oportunidade perfeita para isso.

### 3 Esquema elétrico

A atividade de elaboração de esquema elétrico consiste em desenvolver a representação do circuito eletroeletrônicos relativo à placa de circuito impresso em projeto. Essa atividade costuma ser realizada por meio de ferramentas de desenvolvimento gráficas. Em linhas gerais, por meio dessas ferramentas, são elencados (e representados) os componentes eletrônicos discretos a fazerem parte da placa de circuito impresso em projeto e a interconexão entre eles.

#### 3.1 Requisitos

Requisito	Classe	Especificação
R2	Alimentação	<ol style="list-style-type: none"><li>1 Suportar tensão de entrada na faixa de 7 a 12V em corrente contínua;</li><li>2 Empregar um conector de alimentação do tipo <i>jack J4</i>;</li><li>3 Empregar proteção contra tensão reversa;</li><li>4 Empregar um regulador linear com saída de 5V;</li><li>5 Empregar um diodo emissor de luz para sinalizar a presença de alimentação na placa.</li></ol>

Requisito	Classe	Especificação
R3	Operação	<ol style="list-style-type: none"><li>1 Empregar microcontrolador PIC18F4550 como unidade de processamento e controle;</li><li>2 Possuir uma barra de pinos compatível com a padrão de gravação <i>ICSP</i> para o microcontrolador empregado;</li><li>3 Possuir uma chave tátil para reiniciar a operação do microcontrolador.</li></ol>

Requisito	Classe	Especificação
R4	Interação com o usuário	<ol style="list-style-type: none"><li>1 Possuir um teclado numérico de cinco (5) teclas confeccionado a partir de chaves tácteis;</li><li>2 Possuir um visor de 16x2 caracteres da família JHD162A, com luminosidade a ser controlada por meio de <i>trimmer</i>;</li><li>3 Possuir conjunto de, pelo menos, quatro (4) diodos emissores de luz (<i>LED</i>) a partir do qual seja possível sinalizar estados diversos de operação da unidade de processamento e controle;</li><li>4 Possuir um <i>trimpot</i> com o qual permitir a entrada de valores contínuos por parte dos usuários.</li></ol>

<b>Requisito</b>	<b>Classe</b>	<b>Especificação</b>
R5	Periféricos e expansão	<ol style="list-style-type: none"> <li>1 Empregar um relógio de tempo real da família MCP7940;</li> <li>2 Possuir uma barra de expansão que contemple os sinais de referência e de alimentação da placa e quatro entradas analógicas a formarem dois pares de entradas analógicas diferenciais. Tais pares devem ser condicionados de forma a se gerar níveis entre 0V e 3V a serem disponibilizados ao microcontrolador empregado;</li> <li>3 Possuir uma saída analógica formada por sinal em modulação PWM;</li> <li>4 Possuir uma barra de expansão de sinais que contemple os sinais de referência (terra) e de alimentação do circuito assim como os demais pinos não utilizados do microcontrolador.</li> </ol>

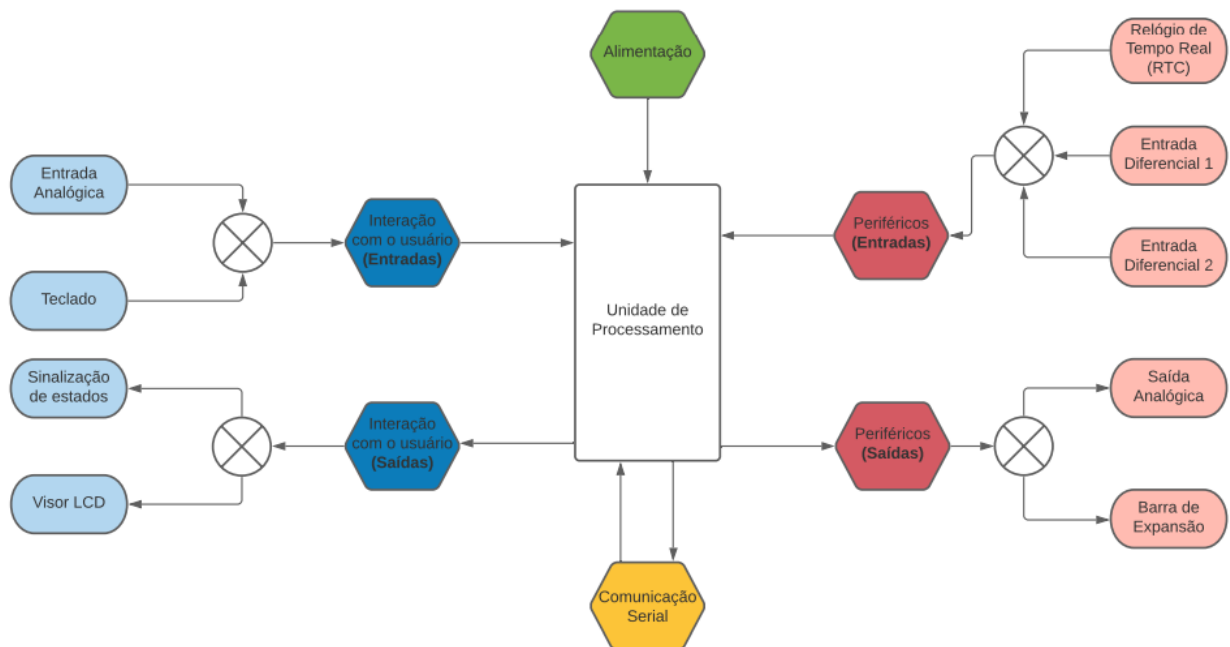
<b>Requisito</b>	<b>Classe</b>	<b>Especificação</b>
R6	Requisitos adicionais	<ol style="list-style-type: none"> <li>1 Empregar um conversor USB-serial da família MCP2200 para se permitir a comunicação serial com a placa desenvolvida.</li> </ol>

## 3.2 Circuito

O dispositivo almejado deve permitir, como principal meta, a interação entre diversas entradas e saídas com a unidade de processamento e os subcircuitos necessários para a efetivar a requisição do usuário.

Dentre entradas, algumas diretamente vindas do usuário (como o teclado), outras pré-definidas por estes (as entradas analógicas), outras vindas de terceiros (diferenciais) e ainda uma entrada do próprio dispositivo (RTC).

Dentre saídas, existem as que tanto visam informar o usuário (sinais de estados e LCD) quanto exportar um resultado (saída analógica).



*Figura 2 - representação em blocos funcionais do circuito*





### 3.3.2 Alimentação

A subdivisão da alimentação é responsável por suprir devidamente a demanda energética do sistema, reduzindo a tensão da fonte de alimentação externa para o nível necessário para o projeto em questão. Confira abaixo o esquema elétrico.

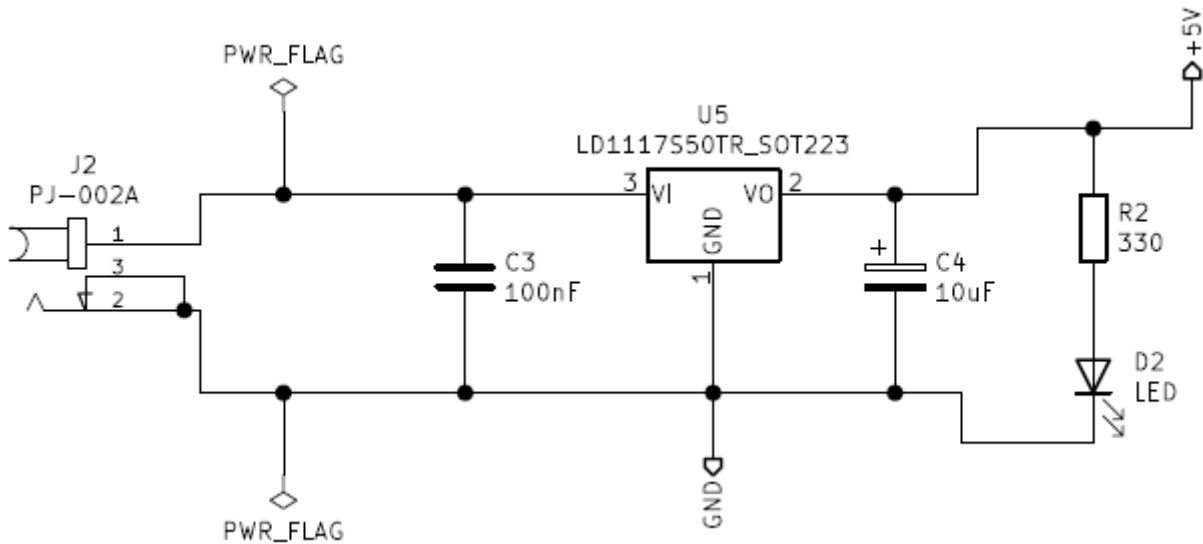


Figura 4- Esquema elétrico do subcircuito de alimentação

### 3.3.3 Relógio de Tempo Real

Esta subdivisão baseia-se em comunicação com o microcontrolador através do protocolo I2C e pode fornecer informações de data, hora, ano e afins, assim como pode também receber dados do PIC. Confira abaixo seu esquema elétrico.

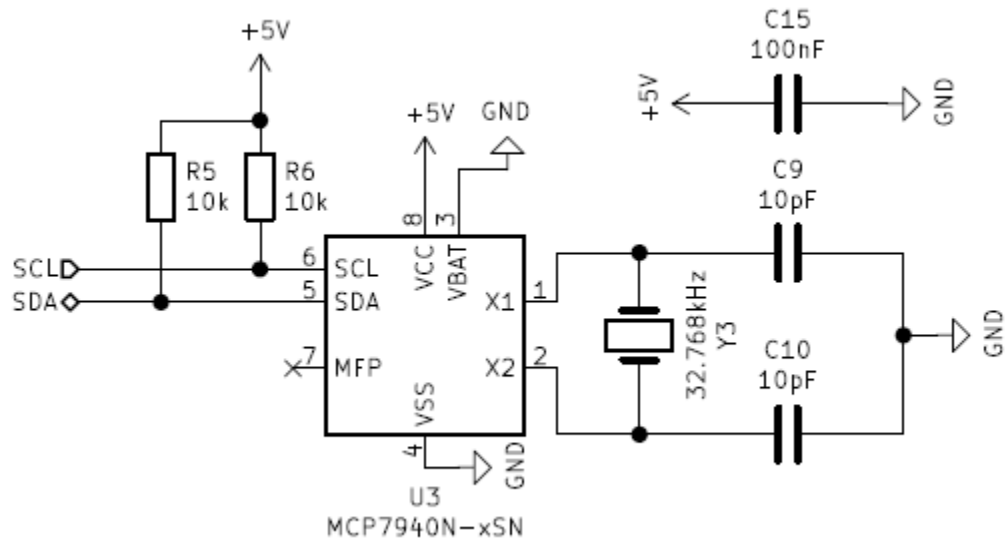
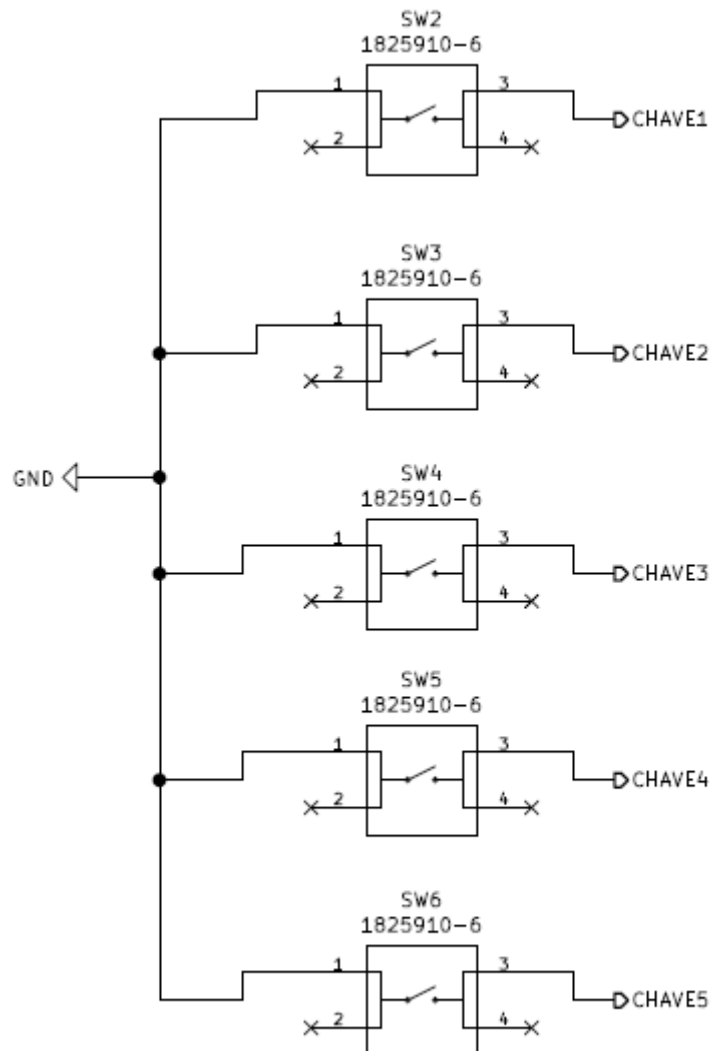


Figura 5- Esquema elétrico do relógio de tempo real

### 3.3.4 Teclado

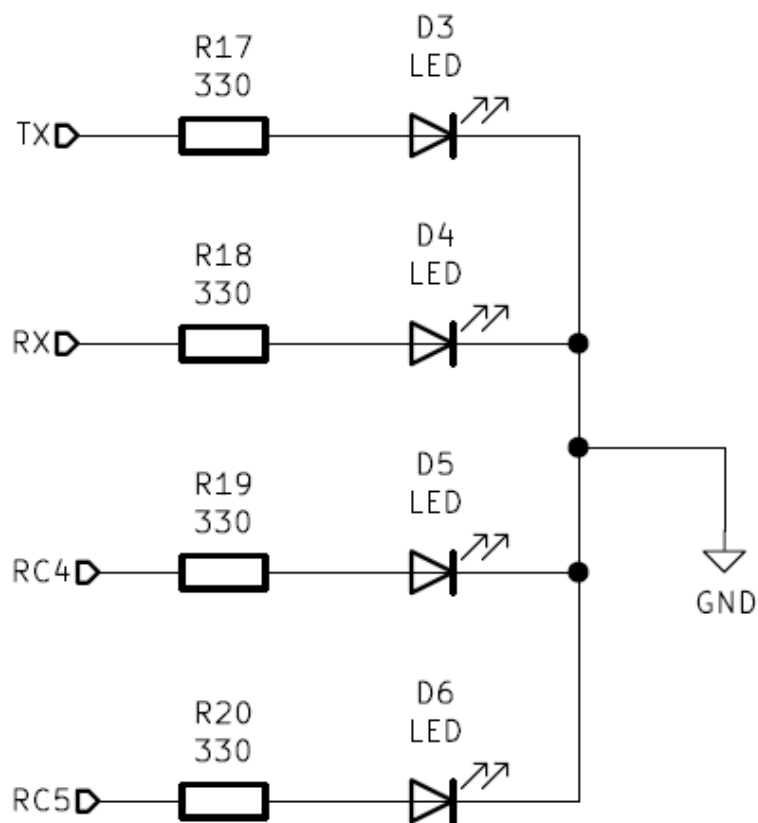
O teclado, parte de uso exclusivo do usuário para com o usuário, é constituído por 5 botões e permitem que o usuário forneça valores digitais para o sistema, assim como configurar o que deve aparecer no LCD, que dado específico planeja obter, etc. Confira abaixo seu esquema elétrico.



*Figura 6- Esquema elétrico do Teclado*

### 3.3.5 Sinalização

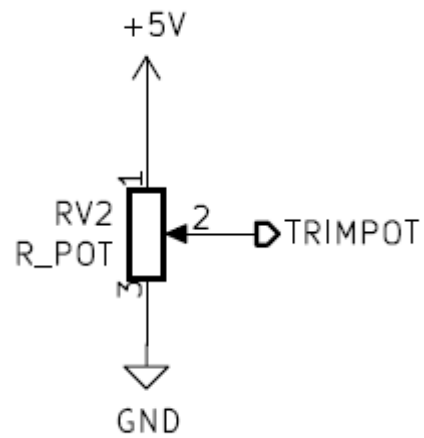
A sinalização para o usuário é formada por um conjunto de 4 *leds* e tem a função de fornecer uma sinalização luminosa sobre determinados pinos do microcontrolador. Confira abaixo seu esquema elétrico.



*Figura 7- Esquema elétrico da sinalização luminosa*

### 3.3.6 Entrada Analógica

Esta consiste em apenas 1 *trimpot* e tem a função de fornecer um valor analógico para o microcontrolador. Confira abaixo seu Esquema elétrico.



*Figura 8- Esquema elétrico da entrada analógica*

### 3.3.7 Entradas diferenciais

Este é constituída por dois circuitos que realizam a mesma função. Cada circuito possui duas entradas analógicas, que são subtraídas uma da outra e geram um sinal de tensão que é proporcional à diferença entre as duas entradas. Tal sinal é fornecido para o microcontrolador. (mais informações são encontradas no memorial de cálculo e especificações elétricas). Confira abaixo seus esquemas elétricos

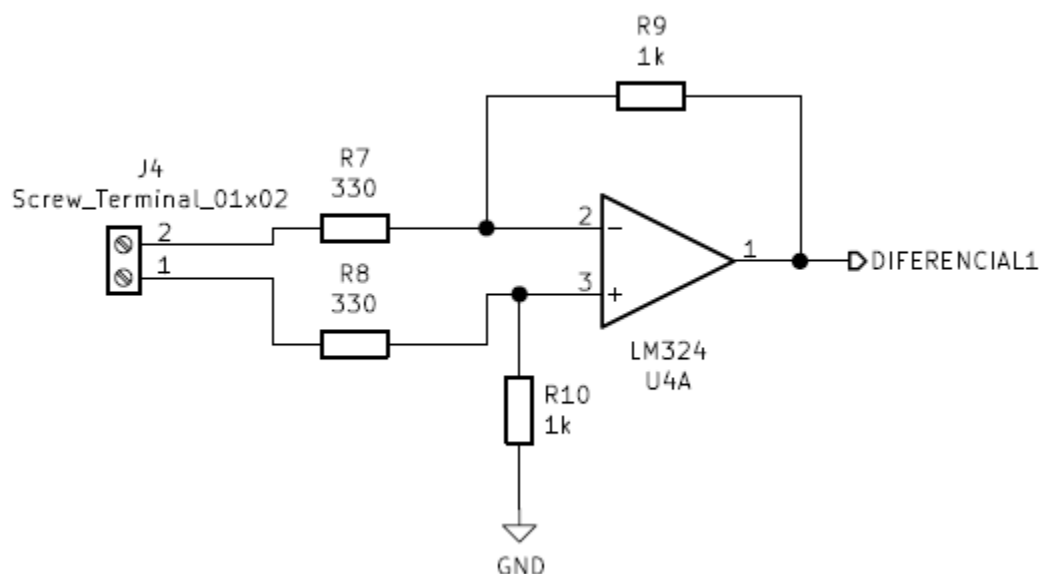


Figura 9- Esquema elétrico da entrada diferencial 1

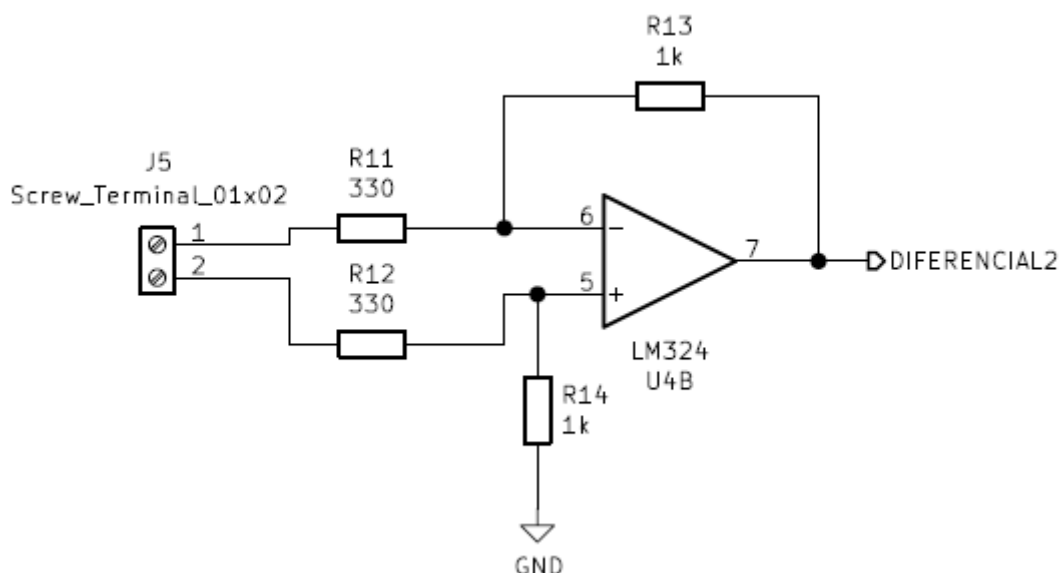
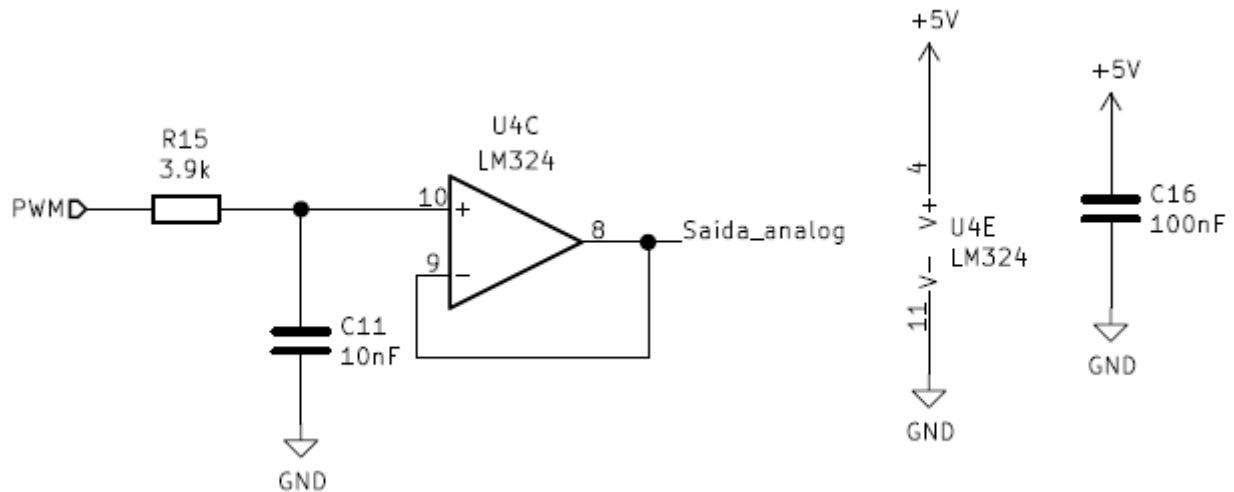


Figura 10 - Esquema elétrico da entrada diferencial 2

### 3.3.8 Saída Analógica

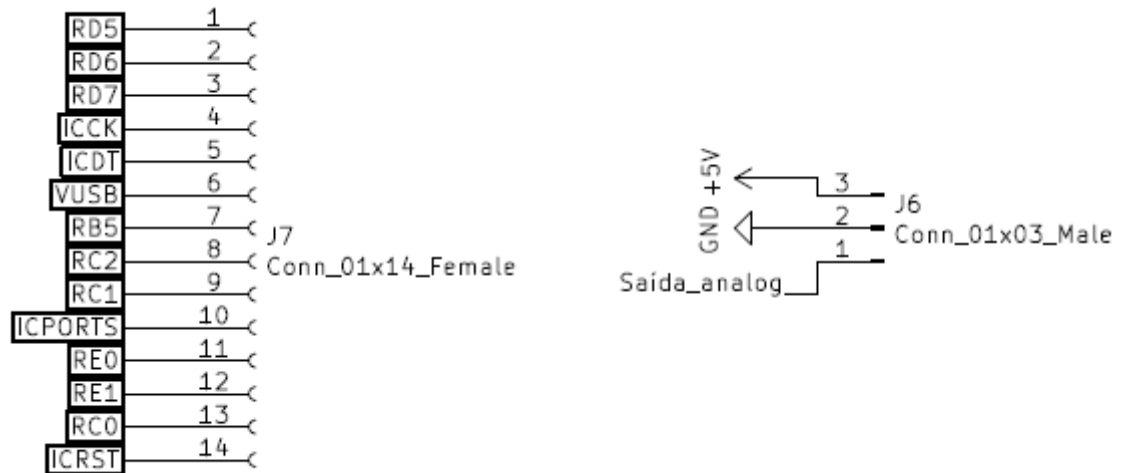
Essa parte é formada por um filtro passa-baixa e tem a função de fornecer um nível de tensão para o usuário. Esse nível de tensão é gerado a partir de uma saída PWM do microcontrolador. Confira abaixo seu esquema elétrico.



*Figura 11- Esquema elétrico da saída analógica*

### 3.3.9 Barras de expansão

uma barra contempla todos os pinos do microcontrolador que não estão sendo usados. A outra barra contempla os sinais de alimentação do circuito e a saída analógica que foi mostrada na saída analógica. Confira abaixo seu esquema elétrico.



*Figura 12- esquema elétrico das barras de expansão*



### 3.3.10 Comunicação serial

Esse sub-circuito é principalmente constituído por um conversor USB-serial e um conector USB. Ele tem a função de estabelecer a comunicação entre o microcontrolador e, por exemplo, um computador que esteja conectado à placa através de um cabo USB. Confira seu esquema elétrico na imagem abaixo.

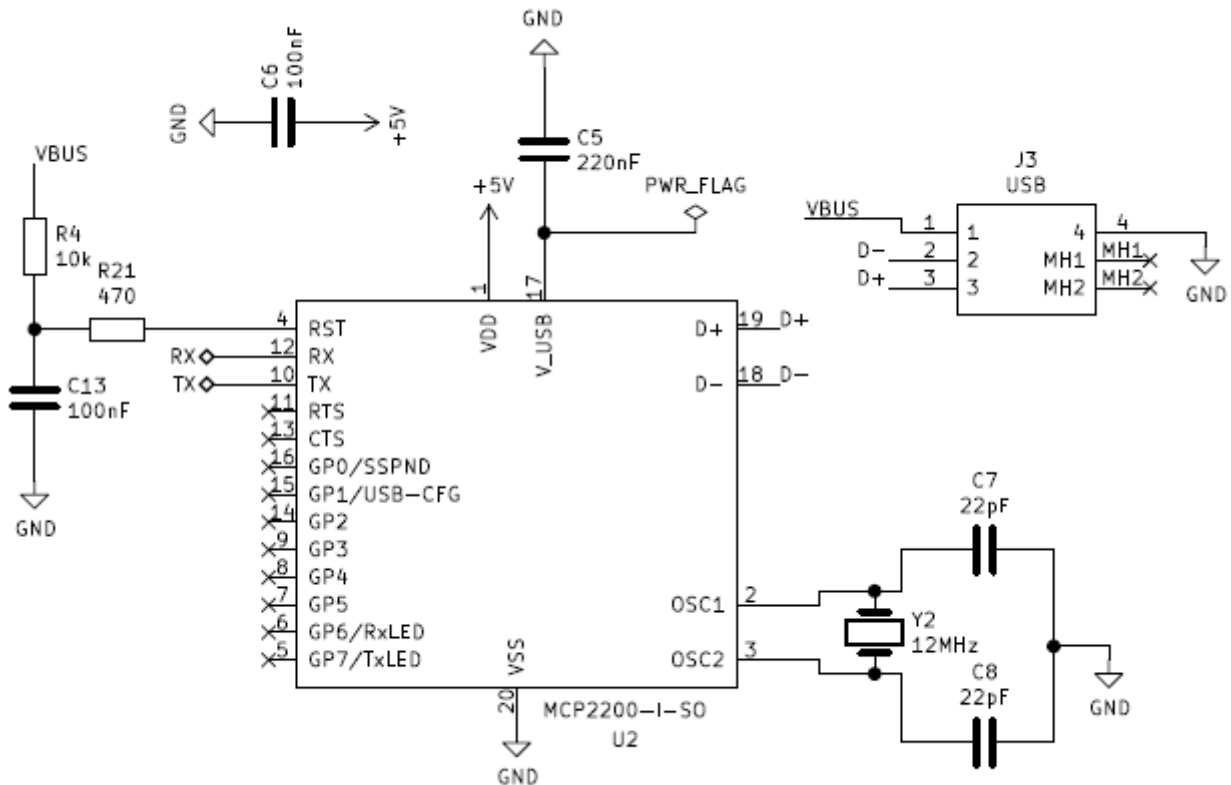


Figura 13- Esquema elétrico do subcircuito de comunicação serial

### 3.3.11 Visor LCD

Essa parte é constituída por um visor de cristal líquido e um *trimmer*. Tem o objetivo de fornecer informações em formato de texto para o usuário sobre algum estado do sistema. O *trimmer* serve para alterar o contraste do visor. Confira abaixo seu esquema elétrico

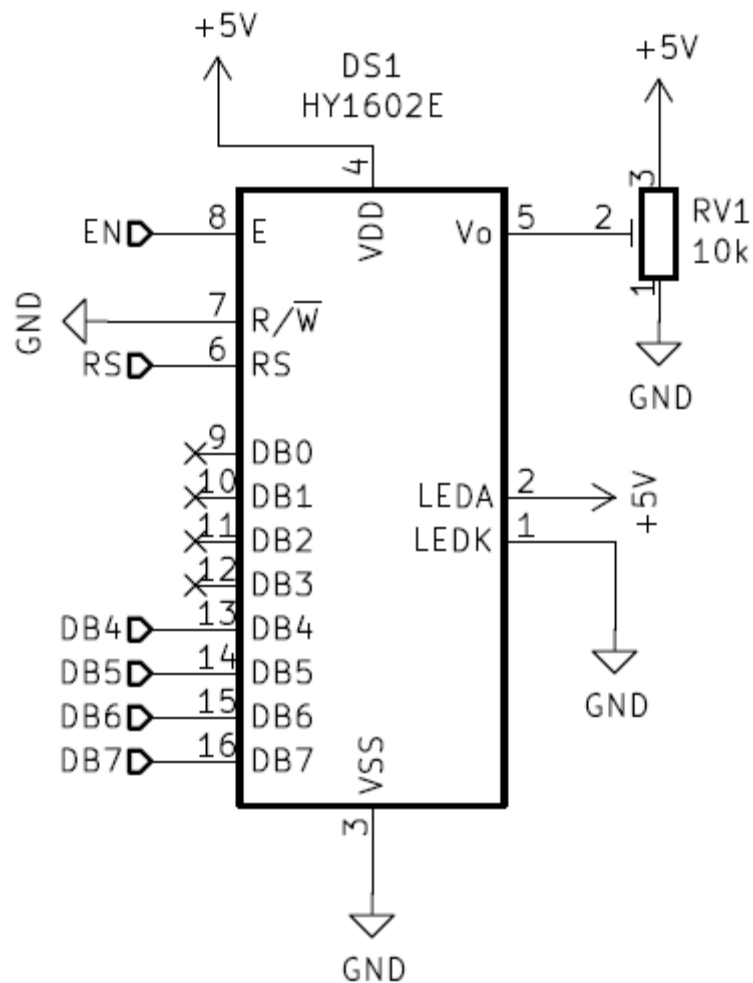
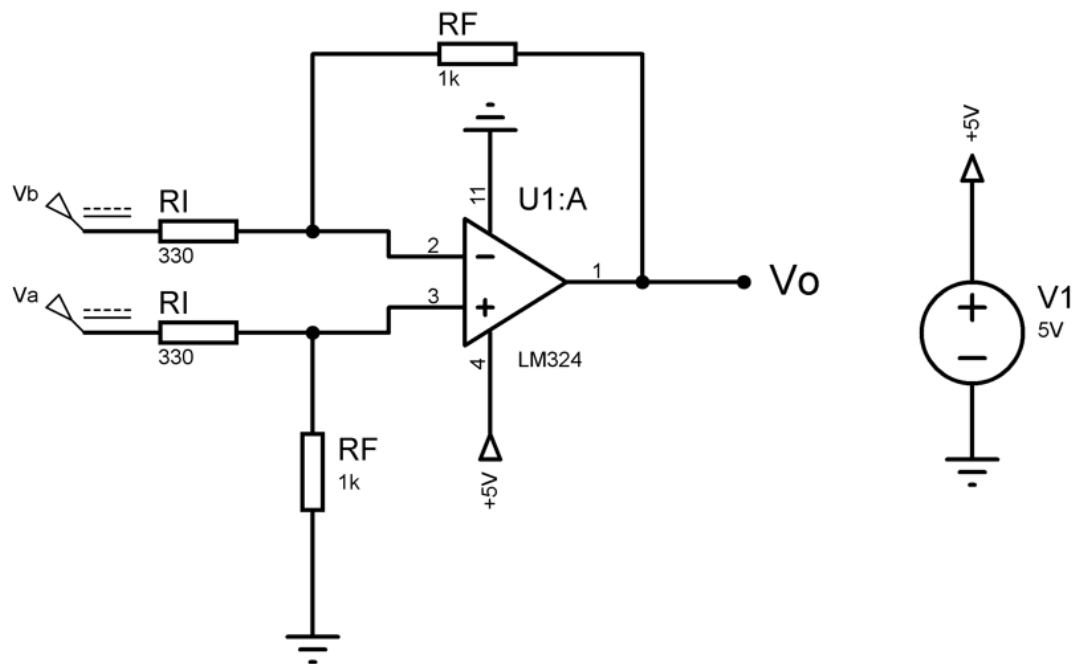


Figura 14- Esquema elétrico do visor LCD

### 3.4 Memorial de cálculos

#### 3.4.1 Entradas diferenciais



Ambas entradas diferenciais apresentam o circuito apresentado acima. A equação para a tensão de saída é mostrada abaixo:

$$V_o = \frac{R_F}{R_I} * (V_A - V_B)$$

Já que a saída deveria variar de 0 a 3V, o ganho escolhido foi de:

$$\frac{R_F}{R_I} = \frac{1000}{330} = 3,03$$

A partir desse ganho, a entrada diferencial ( $V_A - V_B$ ) poderia variar de 0 a 1V. Porém, a tensão inferior de saturação do amplificador operacional escolhido é de 0.5V (característica inerente ao amplificador), e por causa disso, se a entrada diferencial for menor do que aproximadamente 0.17V, a saída ficará limitada ao valor de 0.5V, como é mostrado a seguir:

$$V_o = 3,03 * (0,17) = 0,51 V$$

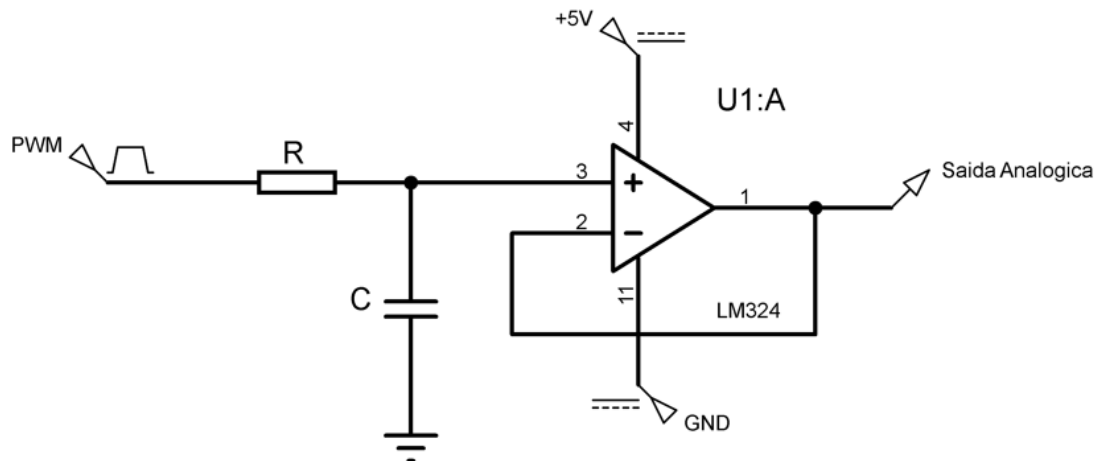
**Conclusão:**

**Para ( $V_A - V_B$ ) variando de 0.17V a 1V**

**$V_o$  vai variar de 0.5V a 3V**

### 3.4.2 Saída Analógica

Projeto do filtro passa-baixa:



A frequência de corte escolhida para o filtro foi de 4 kHz. Para descobrir a frequência do PWM, foi usada a seguinte relação (onde  $f_c$  é a frequência de corte e K é uma constante):

$$f_{PWM} = K * f_c$$

Escolhendo uma constante  $K=25$ , com o objetivo de gerar uma frequência bem alta para o PWM, chegou-se ao seguinte resultado:

$$f_{PWM} = K * f_c \Leftrightarrow f_{PWM} = 25 * 4000 \Leftrightarrow f_{PWM} = 100 \text{ kHz}$$

A seguir, foram calculados os valores de R e C:

$$R * C = \frac{1}{2 * \pi * f_c} \Leftrightarrow R * C = 39,78 * 10^{-6}$$

Escolhendo um  $R = 3,9 \text{ k}\Omega$  (valor comercial), foi encontrado o valor de C:

$$C = \frac{39,78 * 10^{-6}}{3900} \Leftrightarrow C = 10,2 * 10^{-9}$$

Logo, os valores finais comerciais escolhidos e a frequência do PWM são resumidos abaixo:

$$R = 3,9 \text{ k}\Omega$$

$$C = 10 \text{ nF}$$

$$f_{PWM} = 100 \text{ kHz}$$

### **3.5 Relatório de verificação de erros de projeto**

O relatório de erros do esquema elétrico não apresentou nenhum aviso.

## 4 Placa de circuito impresso

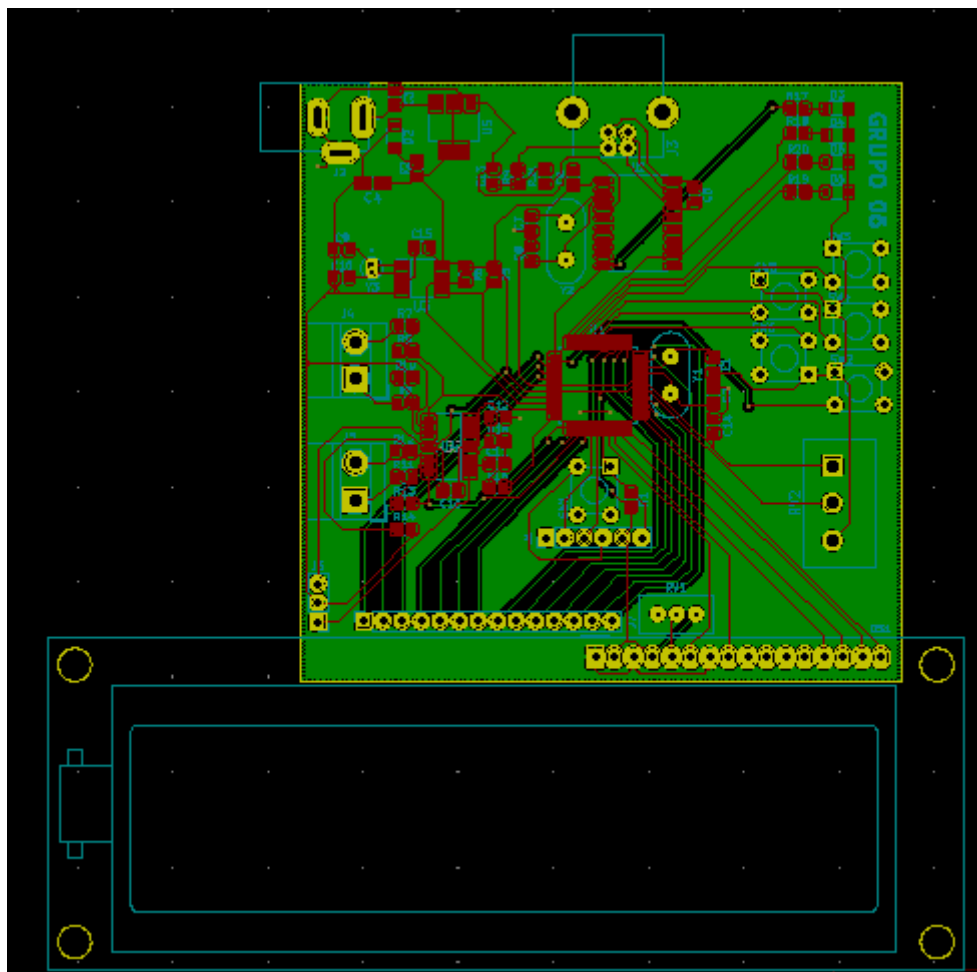
O projeto da placa de circuito impresso se trata de especificar como deve ocorrer a transposição do esquema elétrico desenvolvido para uma PCI real. Nesta etapa, os componentes que fazem parte do circuito são associados às pegadas (*footprints*), e suas interconexões, às trilhas de sinais. Ainda nesta etapa, é também produzida uma lista de compra para os componentes eletroeletrônicos que fazem parte da placa de circuito impresso projetada.

### 4.1 Requisitos

Requisito	Classe	Especificação
R1	Características físicas da placa de circuito impresso	<ul style="list-style-type: none"><li>4 Possuir dimensão de até 8x8 cm<sup>2</sup>;</li><li>5 Ser de dupla face e utilizar a face inferior somente como plano de terra;</li><li>6 Possuir capacitores de supressão de tensão entre a alimentação e o sinal de terra de cada circuito integrado utilizado pelo circuito;</li><li>7 Sua face superior deve possuir uma camada de texto (<i>silk</i>) na qual haja a identificação de cada componente eletroeletrônico e a identificação do grupo que a desenvolveu;</li><li>8 Possuir quadro furos de fixação dispostos em seus cantos;</li><li>9 Possuir identificação de todas as conexões de entrada e de saídas presentes na placa.</li></ul>

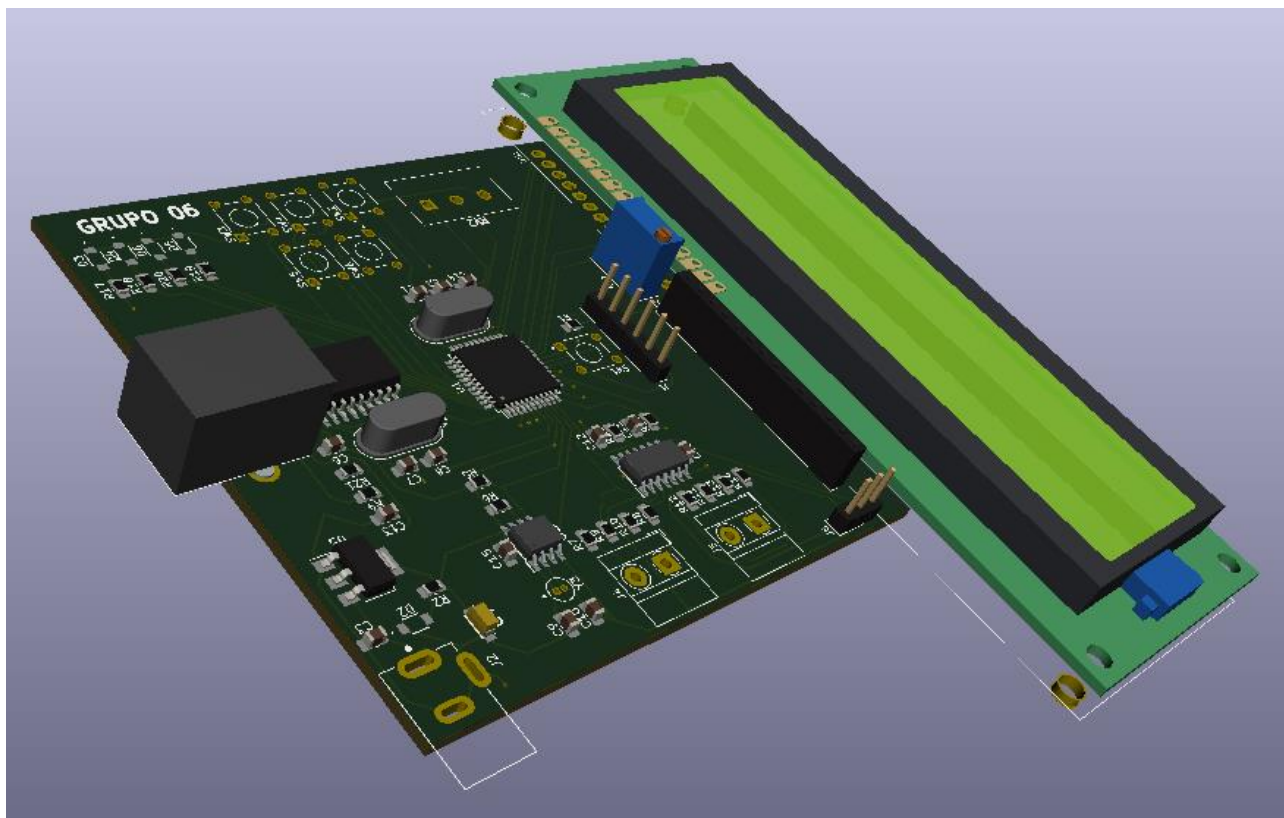
Requisito	Classe	Especificação
R8	Espaçamento e dimensões de trilhas e afins	<ul style="list-style-type: none"><li>1 Utilização do formato de arquivos eletrônicos de fabricação <i>Gerber RS274X</i>;</li><li>2 Mínima largura para trilhas de sinais de 8 mils;</li><li>3 Mínima largura para trilhas de alimentação de 12 mils;</li><li>4 Mínimo espaçamento entre trilhas, furos e ilhas de 8 mils</li><li>5 Mínimo diâmetro de furo de vias de 12 mils;</li><li>6 Mínimo diâmetro de ilhas de vias de 25 mils;</li><li>7 Não utilizar microvias.</li></ul>

## 4.2 Desenho da placa de circuito impresso



*Figura 15- Desenho da PCI*

### 4.3 Visão tridimensional do circuito projetado



*Figura 16- Visão tridimensional da PCI*



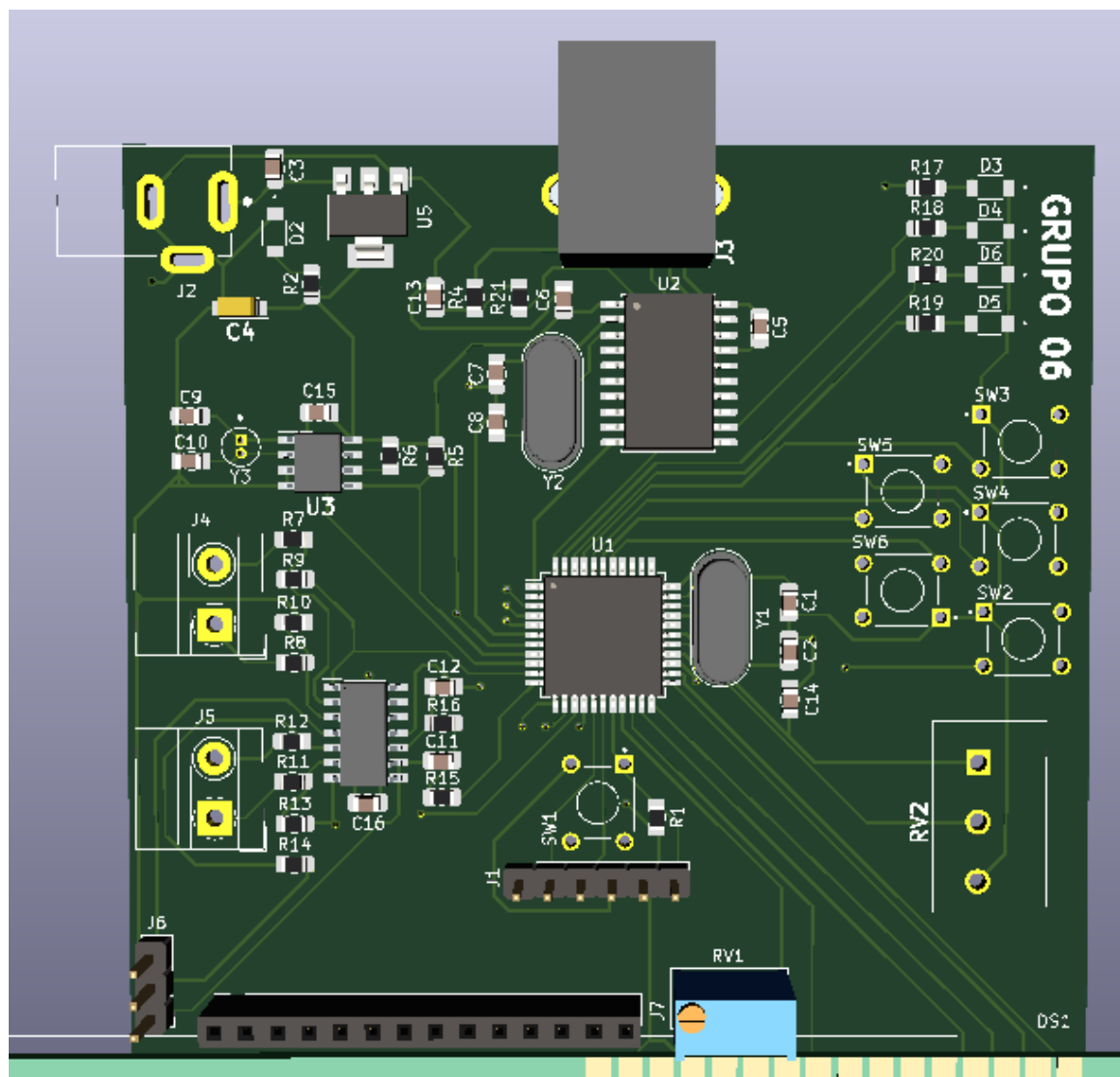
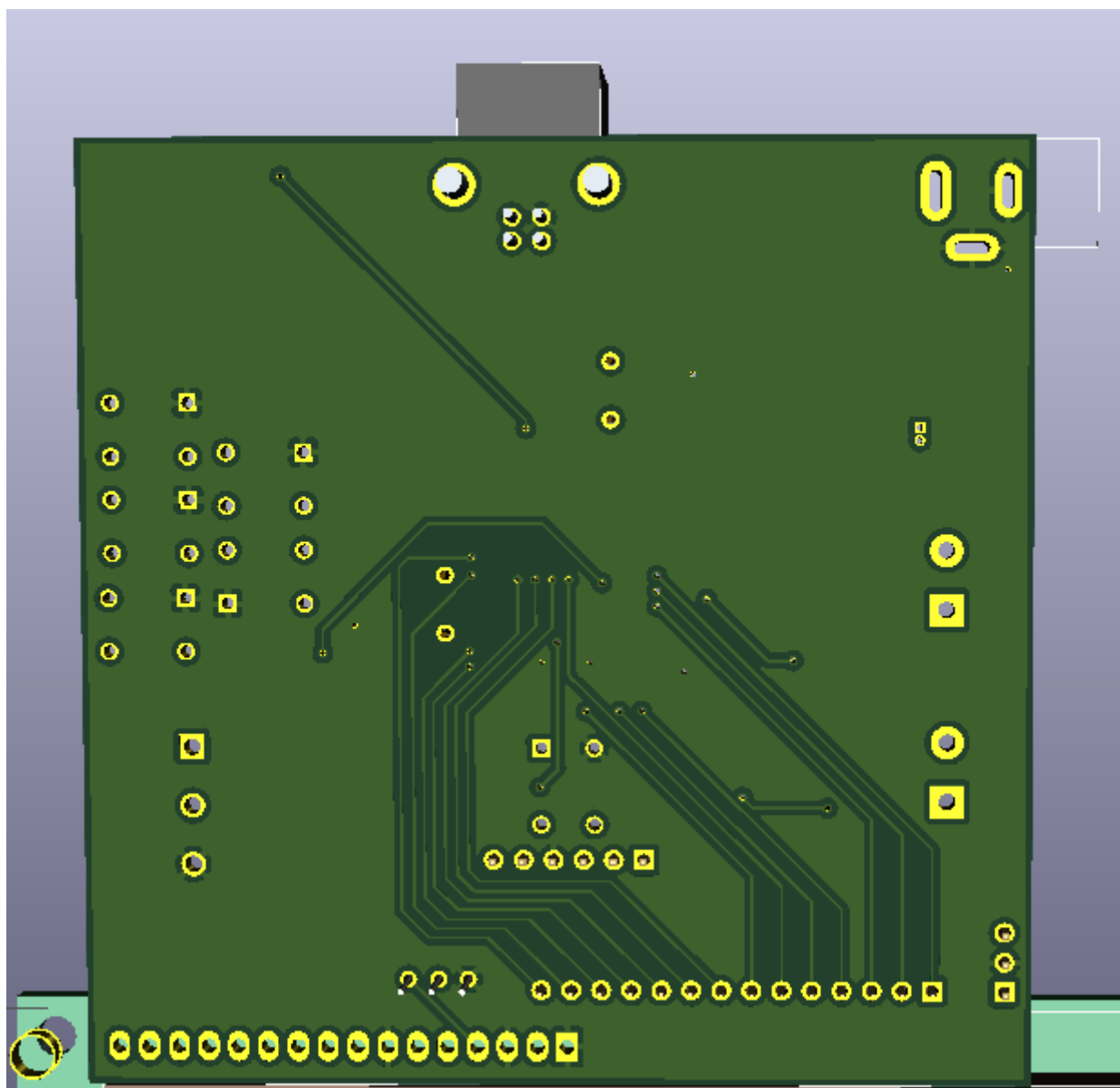


Figura 17- Visão da parte frontal da PCI



*Figura 18- Visão traseira da PCI*

## 4.4 Relatório de verificação de erros de projeto

\*\* Drc report for C:\Users\Lucas F. Machado\Desktop\PBLE01\PBLE01\_Grupo06\PBLE01.kicad\_pcb \*\*

\*\* Created on \*\*

\*\* Found 4 DRC errors \*\*

ErrType(19): Ilha muito próxima a ilha

@(115,570 mm, 71,942 mm): Ilha 1 de Y3 em Todas as camadas de cobre

@(115,570 mm, 73,042 mm): Ilha 2 de Y3 em Todas as camadas de cobre

ErrType(45): Sobreposição de pátios

@(147,053 mm, 56,363 mm): Footprint J3 em F.Cu

@(142,314 mm, 60,325 mm): Footprint C6 em F.Cu

ErrType(45): Sobreposição de pátios

@(147,053 mm, 56,363 mm): Footprint J3 em F.Cu

@(138,633 mm, 60,198 mm): Footprint R21 em F.Cu

ErrType(45): Sobreposição de pátios

@(176,847 mm, 98,770 mm): Footprint RV2 em F.Cu

@(161,064 mm, 93,467 mm): Footprint C14 em F.Cu

\*\* Found 0 unconnected pads \*\*

\*\* End of Report \*\*

## 5 Especificações elétricas

### 5.1 Níveis de alimentação suportados

A placa suporta fonte de corrente contínua externa que seja entre 7V a 15V.

Tensões inferiores a 7V também podem ser aplicadas, porém não é recomendado, pois o sistema pode não operar da forma correta.

### 5.2 Faixa de níveis de consumo estimadas

Estamos usando 22 portas do microcontrolador para acionar saídas e ler entradas. Como cada porta tem a capacidade de drenar ou fornecer 20mA de corrente, então, considerando o pior caso (onde todas as portas estão operando com o máximo de sua capacidade e ao mesmo tempo), o sistema vai consumir aproximadamente:

22 portas x 20mA = **440mA**

### 5.3 Correntes e tensões máximas de entrada

Entradas Diferenciais → 0 a 32V em cada entrada de cada par diferencial.

### 5.4 Correntes e tensões máximas de saída

Saída analógica → 0,5V a 4,5V (é limitada pela saída PWM).

## **6 Programa embarcado de validação**

Na etapa de desenvolvimento do programa embarcado, é desenvolvido um sequenciamento de ações mínimo a ser utilizado exclusivamente para se testar as funcionalidades básicas de componentes do circuito impresso projetado, como, por exemplo, para se verificar se uma unidade de processamento é capaz de detectar e identificar interações com demais componentes do circuito. Esse programa embarcado não corresponde ao programa final de aplicação da placa desenvolvida, mas apenas a um subproduto de projeto utilizado exclusivamente para se realizar certas validações de projeto.

## 7 Montagem

### 7.1 Lista de compras

ITEM	QUANTIDADE	FABRICANTE	PART NUMBER
Amplificador Operacional	3	On Semi	LM324DG
Capacitor 100nF	6	Diversos	445-175390-1-ND
Capacitor 10nF	1	Diversos	C321C103J3G5TA-ND
Capacitor 10pF	2	Diversos	399-7815-2-ND
Capacitor 10uF	2	Diversos	493-1057-ND
Capacitor 15pF	2	Diversos	399-4152-ND
Capacitor 220nF	1	Diversos	C356C224K1G5TA-ND
Capacitor 22pF	2	Diversos	399-4220-ND
Chaves tácteis	6	TE Connectivity	1825910-6
Con. Barra de expansão	1	TE Connectivity	A31112-ND
Con. Entradas diferenciais	2	On Shore Inc.	ED2580-ND
Conector alimentação	1	CUI Devices	PJ-002A CUI
Cristal de 12KHZ	1	CTS Electronic	CTX1077-ND
Cristal de 20KHZ	1	CTS Electronic	CTX1106-ND
Cristal de 32.768KHZ	1	ABRACON	535-9034-ND
LEDs	5	Lite On	160-1169-2-ND
Microprocessador	1	Microchip	PIC18F4550-I/PT-ND
Regulador de tensão	1	STMicroelectronics	497-1238-2-ND
Relógio de tempo real	1	Microchip	MCP7940N-I/SN
Resistor 10k Ohms	4	Diversos	10KQBK-ND
Resistor 1k Ohms	4	Diversos	S1KHTR-ND
Resistor 330 Ohms	9	Diversos	330QBK-ND
Resistor 3k9 Ohms	1	Diversos	3.9KW-2-ND
Resistor 470 Ohms	1	Diversos	470QBK-ND
Transceptor USB-serial	1	Microchip	MCP2200-I/SO-ND
Trimmer de 10k	1	Bourns Inc	3296W-1-103RLFTR-ND
Trimpot 10k	1	TT Electronics	P160KN-0QC15B100K
Visor LCD	1	Precise	HY1602E

## 7.2 Análise de Custos

ID NO E.Elétrico	FORNECEDOR	PREÇO UNITÁRIO
Diversos	Digikey	\$1.25
C11	Digikey	\$0.17
Diversos	Digikey	\$0.20
Diversos	Digikey	\$0.28
Diversos	Digikey	\$0.55
C05	Digikey	\$1.67
Diversos	Digikey	\$0.20
Diversos	Digikey	\$0.11
J7	Digikey	\$0.25
J4, J5	Digikey	\$0.76
J2	Digikey	\$0.60
Y2	Digikey	\$0.36
Y1	Digikey	\$0.36
Y3	Digikey	\$0.32
Diversos	Digikey	\$0.31
U1	Digikey	\$6.66
U5	Digikey	\$0.54
U3	Digikey	\$0.80
Diversos	Digikey	\$0.10
Diversos	Digikey	\$0.10
Diversos	Digikey	\$0.10
R15	Digikey	\$0.44
R21	Digikey	\$0.10
U2	Digikey	\$2.16
RV1	Digikey	\$1.81
RV2	Digikey	\$0.43
DS1	AliExpress	\$3.39

Não foi possível comportar a tabela toda no manual, então ela estará em anexo caso necessário.

## 8 Anexos

### 8.1 Código-fonte do programa de validação

```
1  #include "basic.h"
2  #include "config.h"
3  #include "adc.h"
4  #include "i2c.h"
5  #include "kp.h"
6  #include "lcd.h"
7  #include "pwm.h"
8  #include "rtc.h"
9  #include "serial.h"
10 #include "tmr.h"
11
12 unsigned char Start_slot(void);
13 unsigned char LED_slot(void);
14 unsigned char PWM_slot(void);
15 unsigned char ADC_slot(void);
16 unsigned char Trimpot_slot(void);
17 unsigned char Serial_slot(void);
18 unsigned char RTC_slot(void);
19
```



```

20 void main() {
21
22     LCD_Init();
23     Kp_Init();
24     ADC_Init();
25     PWM_Init();
26     I2C_Init();
27     Serial_Init(9600);
28
29     RTC_Write_Clock(0x04, 0x20, 0x00, 0x00);
30     RTC_Write_Calendar(0x07, 0x06, 0x93, 0x00);
31
32     unsigned char slot;
33
34     for(;;) {
35
36         switch(slot) {
37             case 0:
38                 slot = Start_slot();
39                 break;
40             case 1:
41                 slot = LED_slot();
42                 break;
43             case 2:
44                 slot = PWM_slot();
45                 break;
46             case 3:
47                 slot = ADC_slot();
48                 break;
49             case 4:
50                 slot = Trimpot_slot();
51                 break;
52             case 5:
53                 slot = Serial_slot();
54                 break;
55             case 6:
56                 slot = RTC_slot();
57                 break;
58             default:
59                 slot = 0;
60                 break;
61         }
62
63     }
64 }

```

```

66 unsigned char Start_slot(void){
67     if(!BitTst(Kp_Read(), 0)){
68         while(!BitTst(Kp_Read(), 0));
69         return 1;
70     }
71     if(!BitTst(Kp_Read(), 4)){
72         while(!BitTst(Kp_Read(), 4));
73         return 1;
74     }
75     if(!BitTst(Kp_Read(), 3)){
76         while(!BitTst(Kp_Read(), 3));
77         return 6;
78     }
79
80     RTC_Read_Clock(0);
81     LCD_Command(0x80);
82     LCD_String("PBLE01 ");
83     LCD_Char(((RTC_Read_Hour() & 0x3f) / 16) % 16 + 48);
84     LCD_Char((RTC_Read_Hour() & 0x3f) % 16 + 48);
85     LCD_Char(':');
86     LCD_Char(((RTC_Read_Min() / 16) % 16) + 48);
87     LCD_Char((RTC_Read_Min() % 16) + 48);
88     LCD_Char(':');
89     LCD_Char(((RTC_Read_Sec() / 16) % 16) + 48);
90     LCD_Char((RTC_Read_Sec() % 16) + 48);
91
92     RTC_Read_Calendar(3);
93     LCD_Command(0xC0);
94     LCD_String(" G.06 ");
95     LCD_Char((RTC_Read_Date() / 16) % 16 + 48);
96     LCD_Char((RTC_Read_Date() % 16) + 48);
97     LCD_Char('.');
98     LCD_Char((RTC_Read_Month() / 16) % 16 + 48);
99     LCD_Char((RTC_Read_Month() % 16) + 48);
100    LCD_Char('.');
101    LCD_Char((RTC_Read_Year() / 16) % 16 + 48);
102    LCD_Char((RTC_Read_Year() % 16) + 48);
103
104    return 0;
105
106 }

```

```

108 unsigned char LED_slot(void){
109     BitClr(TRISC, 4);
110     BitClr(TRISC, 5);
111     LCD_Clear();
112
113     for(;;){
114         LCD_Command(0xC0);
115         LCD_Char(0x7F);
116         LCD_Command(0xCF);
117         LCD_Char(0x7E);
118         LCD_Command(0xC6);
119         LCD_String("LEDs");
120
121         if(!BitTst(Kp_Read(), 4)){
122             while(!BitTst(Kp_Read(), 4));
123             return 2;
124         }
125         if(!BitTst(Kp_Read(), 3)){
126             while(!BitTst(Kp_Read(), 3));
127             return 0;
128         }
129         if(!BitTst(Kp_Read(), 0)){
130             while(!BitTst(Kp_Read(), 0));
131             LCD_Clear();
132             while(BitTst(Kp_Read(), 0)){
133                 LCD_Command(0x80);
134                 LCD_String("LED.1 ");
135                 if(BitTst(PORTC, 4))LCD_String("ON ");
136                 else LCD_String("OFF");
137                 if(!BitTst(Kp_Read(), 1)){
138                     while(!BitTst(Kp_Read(), 1));
139                     BitFlp(PORTC, 4);
140                 }
141                 LCD_Command(0xC0);
142                 LCD_String("LED.2 ");
143                 if(BitTst(PORTC, 5))LCD_String("ON ");
144                 else LCD_String("OFF");
145                 if(!BitTst(Kp_Read(), 2)){
146                     while(!BitTst(Kp_Read(), 2));
147                     BitFlp(PORTC, 5);
148                 }
149             }
150             while(!BitTst(Kp_Read(), 0));
151             LCD_Clear();
152         }
153     }
154 }

```

```

156  unsigned char PWM_slot(void){
157      LCD_Clear();
158      for(;;){
159          LCD_Command(0xC0);
160          LCD_Char(0x7F);
161          LCD_Command(0xCF);
162          LCD_Char(0x7E);
163          LCD_Command(0xC6);
164          LCD_String("PWM");
165
166          if(!BitTst(Kp_Read(), 4)){
167              while(!BitTst(Kp_Read(), 4));
168              return 3;
169          }
170          if(!BitTst(Kp_Read(), 3)){
171              while(!BitTst(Kp_Read(), 3));
172              return 1;
173          }
174          if(!BitTst(Kp_Read(), 0)){
175              while(!BitTst(Kp_Read(), 0));
176              LCD_Clear();
177              unsigned char dtc = 0;
178              unsigned int freq = 0;
179
180              while(BitTst(Kp_Read(), 0)){
181
182                  LCD_Command(0x80);
183                  LCD_String("Duty Cycle:");
184                  LCD_Command(0x8F);
185                  LCD_Char('%');
186
187                  LCD_Command(0xC0);

```

```

188 LCD_String("Frequency:");
189 LCD_Command(0xCD);
190 LCD_String("kHz");
191
192 LCD_Command(0x8C);
193 LCD_Char(((dtc / 100) %10) +48);
194 LCD_Char(((dtc / 10) %10) +48);
195 LCD_Char(((dtc / 1) %10) +48);
196
197 LCD_Command(0xCA);
198 LCD_Char(((freq / 100) %10) +48);
199 LCD_Char(((freq / 10) %10) +48);
200 LCD_Char(((freq / 1) %10) +48);
201
202 PWM_Set(dtc);
203 PWM_Freq_Set(freq * 1000);
204
205 if(!BitTst(Kp_Read(), 1)){
206     while(!BitTst(Kp_Read(), 1));
207     if (dtc < 100) dtc = dtc + 5;
208 }
209 if(!BitTst(Kp_Read(), 2)){
210     while(!BitTst(Kp_Read(), 2));
211     if(dtc > 0) dtc = dtc - 5;
212 }
213 if(!BitTst(Kp_Read(), 4)){
214     while(!BitTst(Kp_Read(), 4));
215     if (freq < 100) freq = freq + 5;
216 }
217 if(!BitTst(Kp_Read(), 3)){
218     while(!BitTst(Kp_Read(), 3));
219     if(freq > 0) freq = freq - 5;
220 }
221 }
222 while(!BitTst(Kp_Read(), 0));
223 LCD_Clear();
224 }
225 }
226 }

```

```

228  unsigned char ADC_slot(void){
229      LCD_Clear();
230      for(;;){
231          LCD_Command(0xC0);
232          LCD_Char(0x7F);
233          LCD_Command(0xCF);
234          LCD_Char(0x7E);
235          LCD_Command(0xC6);
236          LCD_String("ADC");
237
238          if(!BitTst(Kp_Read(), 4)){
239              while(!BitTst(Kp_Read(), 4));
240              return 4;
241          }
242          if(!BitTst(Kp_Read(), 3)){
243              while(!BitTst(Kp_Read(), 3));
244              return 2;
245          }
246          if(!BitTst(Kp_Read(), 0)){
247              while(!BitTst(Kp_Read(), 0));
248              LCD_Clear();
249              unsigned int adc1;
250              unsigned int adc2;
251
252              while(BitTst(Kp_Read(), 0)){
253
254                  adc1 = ADC_Read(8);
255                  adc2 = ADC_Read(11);
256                  adc1 = adc1 / 6,138;
257                  adc2 = adc2 / 6,138;
258
259                  LCD_Command(0x80);

```

```

260 LCD_String("In.Dif. 1:");
261 LCD_Command(0x8F);
262 LCD_Char('v');
263
264 LCD_Command(0xC0);
265 LCD_String("In.Dif. 2:");
266 LCD_Command(0xCF);
267 LCD_String("v");
268
269 LCD_Command(0x8B);
270 LCD_Char(((adc1 / 100) %10) +48);
271 LCD_Char(',');
272 LCD_Char(((adc1 / 10) %10) +48);
273 LCD_Char(((adc1 / 1) %10) +48);
274
275 LCD_Command(0xCB);
276 LCD_Char(((adc2 / 100) %10) +48);
277 LCD_Char(',');
278 LCD_Char(((adc2 / 10) %10) +48);
279 LCD_Char(((adc2 / 1) %10) +48);
280
281 }
282 while(!BitTst(Kp_Read(), 0));
283 LCD_Clear();
284 }
285 }
286

```

```

288 unsigned char Trimpot_slot(void){
289     LCD_Clear();
290     for(;;){
291         LCD_Command(0xC0);
292         LCD_Char(0x7F);
293         LCD_Command(0xCF);
294         LCD_Char(0x7E);
295         LCD_Command(0xC5);
296         LCD_String("Trimpot");
297
298         if(!BitTst(Kp_Read(), 4)){
299             while(!BitTst(Kp_Read(), 4));
300             return 5;
301         }
302         if(!BitTst(Kp_Read(), 3)){
303             while(!BitTst(Kp_Read(), 3));
304             return 3;
305         }
306         if(!BitTst(Kp_Read(), 0)){
307             while(!BitTst(Kp_Read(), 0));
308             LCD_Clear();
309             unsigned int trm, trm_100, trm_5;
310
311             while(BitTst(Kp_Read(), 0)){
312
313                 trm = ADC_Read(7);
314                 trm_100 = trm / 10.23;
315                 trm_5 = trm / 2.045;
316

```



```

316
317 LCD_Command(0x80);
318 LCD_String("Trimpot : ");
319 LCD_Char(((trm_100 / 100) %10) +48);
320 LCD_Char(((trm_100 / 10) %10) +48);
321 LCD_Char(((trm_100 / 1) %10) +48);
322 LCD_Char('%');
323
324 LCD_Command(0xC2);
325 LCD_Char(((trm / 1000) %10) +48);
326 LCD_Char(((trm / 100) %10) +48);
327 LCD_Char(((trm / 10) %10) +48);
328 LCD_Char(((trm / 1) %10) +48);
329
330 LCD_Command(0xC9);
331 LCD_Char(((trm_5 / 100) %10) +48);
332 LCD_Char(',');
333 LCD_Char(((trm_5 / 10) %10) +48);
334 LCD_Char(((trm_5 / 1) %10) +48);
335 LCD_Char('v');
336
337 }
338 while(!BitTst(Kp_Read(), 0));
339 LCD_Clear();
340 }
341 }
342 }
343

```

```

344 unsigned char Serial_slot(void){
345     LCD_Clear();
346     for(;;){
347         LCD_Command(0xC0);
348         LCD_Char(0x7F);
349         LCD_Command(0xCF);
350         LCD_Char(0x7E);
351         LCD_Command(0xC5);
352         LCD_String("Serial");
353
354         if(!BitTst(Kp_Read(), 4)){
355             while(!BitTst(Kp_Read(), 4));
356             return 6;
357         }
358         if(!BitTst(Kp_Read(), 3)){
359             while(!BitTst(Kp_Read(), 3));
360             return 4;
361         }
362         if(!BitTst(Kp_Read(), 0)){
363             while(!BitTst(Kp_Read(), 0));
364             LCD_Clear();
365             LCD_Command(0x80);
366             LCD_String("RX.");
367             LCD_Command(0xC0);
368             LCD_String("TX.");
369
370             Serial_Send_String("TX/RX.");
371             Serial_Send_Char(0x3A);
372

```

```

373 unsigned char data_out = 65;
374
375 while(BitTst(Kp_Read(), 0)){
376
377     LCD_Command(0xC5);
378     LCD_Char(data_out);
379     LCD_Command(0x85);
380     //LCD_Char(Serial_Receive_Char());
381
382     if(!BitTst(Kp_Read(), 1)){
383         while(!BitTst(Kp_Read(), 1));
384         if(data_out > 65) data_out --;
385     }
386     if(!BitTst(Kp_Read(), 2)){
387         while(!BitTst(Kp_Read(), 2));
388         if (data_out < 90) data_out ++;
389     }
390
391     if(!BitTst(Kp_Read(), 4)){
392         while(!BitTst(Kp_Read(), 4));
393         Serial_Send_Char(data_out);
394     }
395 }
396 while(!BitTst(Kp_Read(), 0));
397 LCD_Clear();
398 }
399 }
400 }

```

```

404 unsigned char RTC_slot(void){
405     LCD_Clear();
406     for(;;){
407         LCD_Command(0xC0);
408         LCD_Char(0x7F);
409         LCD_Command(0xCF);
410         LCD_Char(0x7E);
411         LCD_Command(0xC6);
412         LCD_String("RTC");
413
414         if(!BitTst(Kp_Read(), 4)){
415             while(!BitTst(Kp_Read(), 4));
416             return 0;
417         }
418         if(!BitTst(Kp_Read(), 3)){
419             while(!BitTst(Kp_Read(), 3));
420             return 5;
421         }
422
423         if(!BitTst(Kp_Read(), 0)){
424             while(!BitTst(Kp_Read(), 0));
425             LCD_Clear();
426             while(BitTst(Kp_Read(), 0)){
427                 RTC_Read_Clock(0);
428                 LCD_Command(0x80);
429                 LCD_String("Hour: ");
430                 LCD_Char(((RTC_Read_Hour() & 0x3f) / 16) % 16 + 48);
431                 LCD_Char((RTC_Read_Hour() & 0x3f) % 16 + 48);
432                 LCD_Char(':');
433                 LCD_Char(((RTC_Read_Min() / 16) % 16) + 48);
434                 LCD_Char((RTC_Read_Min() % 16) + 48);
435                 LCD_Char(':');
436                 LCD_Char(((RTC_Read_Sec() / 16) % 16) + 48);
437                 LCD_Char((RTC_Read_Sec() % 16) + 48);
438
439                 RTC_Read_Calendar(3);
440                 LCD_Command(0xC0);
441                 LCD_String("Date: ");
442                 LCD_Char((RTC_Read_Date() / 16) % 16 + 48);
443                 LCD_Char((RTC_Read_Date() % 16) + 48);
444                 LCD_Char('.');
445                 LCD_Char(((RTC_Read_Month() / 16) % 16) + 48);
446                 LCD_Char((RTC_Read_Month() % 16) + 48);
447                 LCD_Char('.');
448                 LCD_Char(((RTC_Read_Year() / 16) % 16) + 48);
449                 LCD_Char((RTC_Read_Year() % 16) + 48);
450             }
451             while(!BitTst(Kp_Read(), 0));
452             LCD_Clear();
453         }
454     }
455 }

```

## 8.2 Programas utilizados

FERRAMENTA	VERSÃO	USO
KiCad	5.1.10	Desenvolvimento do esquema elétrico e PCI
Proteus	8.5	Simulação do sistema
MPLab	5.20	Produção do embarcado de validação

## 9 Bibliografia

- Link para cálculo do filtro disponível em: <http://ww1.microchip.com/downloads/en/AppNotes/00538c.pdf>
- Aulas e proposta de projeto da disciplina, disponível em: <https://sigaa.unifei.edu.br/sigaa/ava/index.jsf>
- Datasheet do amplificador disponível em: [https://www.ti.com/product/LM324LV?utm\\_source=supplyframe&utm\\_medium=SEP&utm\\_campaign=not\\_alldatasheet&DCM=yes&dclid=CN73ktOmpfICFUomuQYdCu-cHcQ](https://www.ti.com/product/LM324LV?utm_source=supplyframe&utm_medium=SEP&utm_campaign=not_alldatasheet&DCM=yes&dclid=CN73ktOmpfICFUomuQYdCu-cHcQ)
- Datasheet do microprocessador disponível em: <https://www.microchip.com/en-us/product/PIC18F4550>
- Datasheet do transceptor disponível em: <https://www.microchip.com/en-us/product/MCP2200>
- Datasheet do relógio disponível em: <https://www.microchip.com/en-us/product/MCP7940N>