

Sistema de Contagem de Pessoas em Sala

José Ailton Batista da Silva

Esse é um sistema microcontrolado onde através de sensores de presença ópticos detecta-se a passagem de pessoas em uma porta indicando o sentido do mesmo e mostrando a quantidade de pessoas atual em um display e no computador.

Introdução

Em certos ambientes é preciso que haja um monitoramento contínuo de pessoas como bancos, shopping, etc. O sistema descrito é uma forma de controlar a quantidade de pessoas num ambiente fechado onde exista somente uma porta como passagem para dentro ou fora do ambiente.

Circuito

Na *figura 1* é mostrado o diagrama de blocos do circuito microcontrolado.

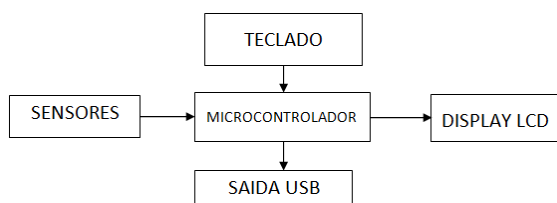


Figura 1

O bloco dos sensores é feito com sensores de presença ópticos que consiste no uso de leds infravermelhos e fotodiodos para emissão e recepção de luz. O led infravermelho ficará emitindo luz infravermelha e está de frente ao fotodiodo que estará recebendo uma grande quantidade dessa luz. Enquanto o fotodiodo estiver recebendo a luz infravermelha ele irá saturar um transistor ao qual

está ligado. Somente quando houver a passagem de algo pela porta que existirá o barramento da luz no fotodiodo fazendo com que o transistor conduza.

No coletor-comum do transistor está ligado um capacitor para a filtragem e em seguida irá para a entrada do microcontrolador. O circuito do sensor de presença óptico está mostrado na *figura 2*.

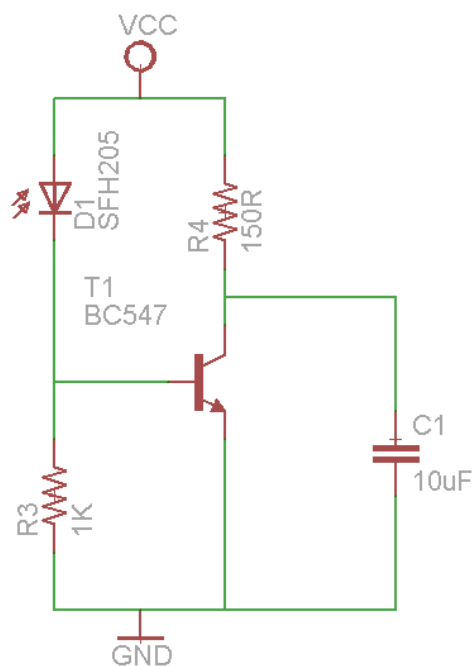


Figura 2

No sistema fazem-se necessários dois sensores de presença para detecção se a pessoa está entrando ou saindo da sala.

O bloco do microcontrolador é responsável por receber os sinais dos sensores de presença e dependendo de qual sensor foi ativado primeiro irá identificar se a pessoa está saindo ou entrando na sala.

Caso o microcontrolador identifique que o usuário está entrando na sala irá incrementar o valor da quantidade de pessoas. Já se o usuário estiver saindo da sala irá decrementar o valor da quantidade de pessoas.

O microcontrolador também receberá informações do teclado (bloco teclado) e ficará atualizando constantemente o display LCD com a quantidade de pessoas atual na sala (bloco LCD). Também enviará ao computador a quantidade de pessoas caso houver alteração nessa quantidade (bloco USB).

No caso de já haver pessoas na sala quando o sistema for iniciado, a quantidade de pessoas poderá de alterada usando o teclado. O teclado utilizado no sistema foi um do tipo 3X4 onde a pinagem é linhas por colunas.

No bloco do display LCD foi utilizado um LCD 16x2 compatível com Hitachi. No LCD será mostrado a quantidade atual de pessoas na sala.

O bloco da saída USB é utilizada para atualizar a quantidade de pessoas no computador mostrando em um software especial (sessão programa). É importante ressaltar que o microcontrolador utilizado foi o PIC18F4550, pois tem disponível comunicação USB e também que o sistema é totalmente alimentado pelo USB não necessitando de qualquer fonte extra.

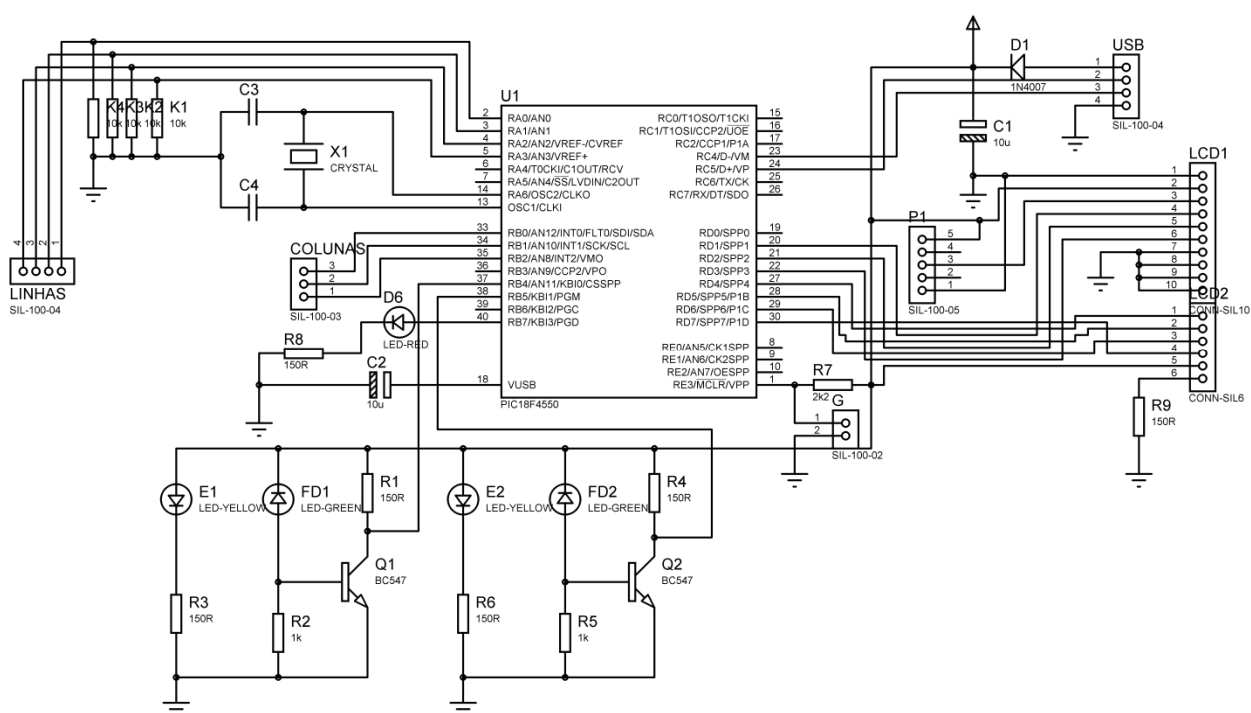


Figura 3

Montagem

A figura 4 mostra o fotolito para confecção da PCI. Nela estará o microcontrolador PIC18f4550 cujo o com o footprint tipo DIL40.

Também estará na placa todos os componentes do esquema da figura 3. Mas o teclado e LCD será pinos, pois irá ser conectada na placa através de cabos flat.

É importante dizer que os LEDs infravermelho e fotodiodos não podem ser soldados direto na placa, pois estes ficaram um de frente para

o outro na porta. Por isso devem-se soldar pinos e usar fios para ligá-los.

Programa

Como o sistema é microcontrolado, tem-se a necessidade de criar um programa para o microcontrolador PIC. O programa foi desenvolvido em linguagem C usando o PCHWD.

Um fator importante a dizer é que foi utilizado a configuração do SanUSB para programar o PIC, então fazem necessário que o microcontrolador já esteja gravado o gerenciador de

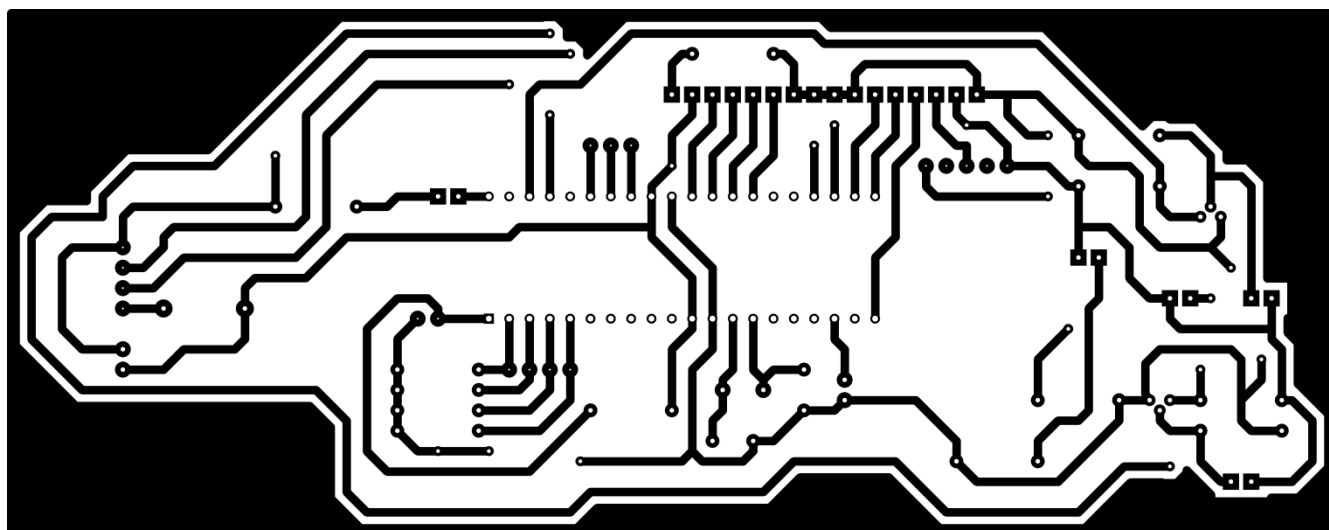


Figura 5

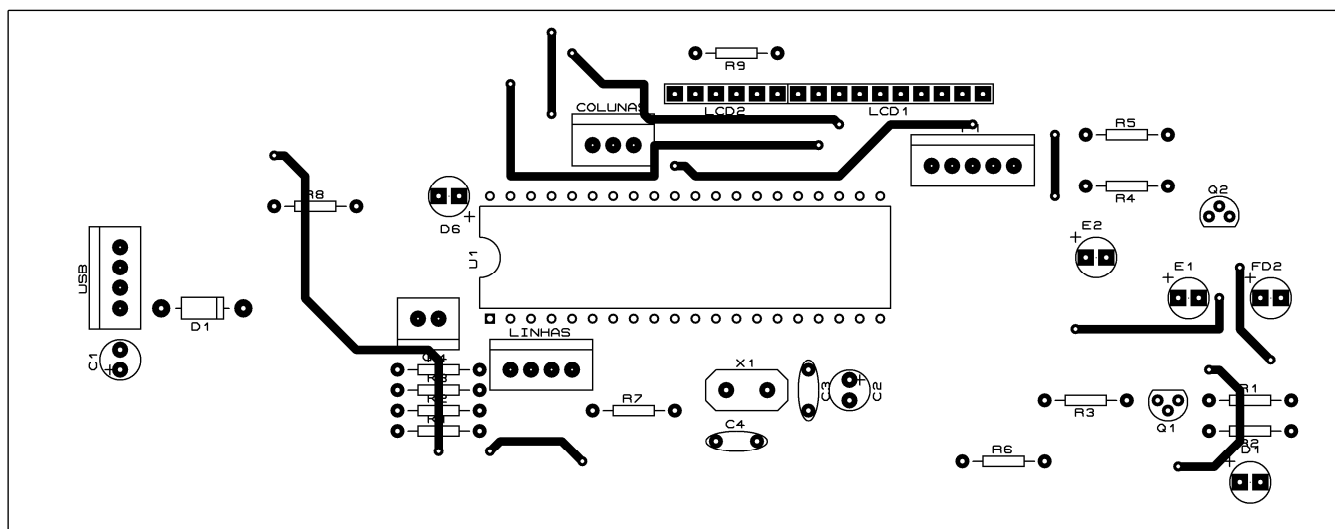


Figura 4

programas do SanUSB conhecido como *bootloader*.

Como a própria placa já tem a configuração do SanUSB, pode-se mudar o programa do PIC a qualquer momento usando o *jump* nos pinos de gravação (pinos G da figura 5).

Para a comunicação USB com o computador foi utilizado a biblioteca *usb_cdc.h* que emula uma porta serial no computador.

Para que o computador reconhecesse o USB foi utilizado o driver *Communication Port* da Microchip que vem no PCWHD.

A seguir será disponibilizado o programa para ser gravado usando a configuração do SanUSB.

```
#include <SanUSB.h>
#include <usb_cdc.h>
#include <stdlib.h>
#define LCD_ENABLE_PIN PIN_D3
#define LCD_RW_PIN PIN_D2
#define LCD_RS_PIN PIN_D1
#define LCD_DATA4 PIN_D4
#define LCD_DATA5 PIN_D5
#define LCD_DATA6 PIN_D6
#define LCD_DATA7 PIN_D7
#define LCD_TYPE 2
#include <lcd.c>
int sensor1=0;
int sensor2=0;
int antsen1=0;
int antsen2=0;
int novo=0;
int32 pessoa=0;
int pcativo=1;
int contKey=0; // setador de coluna no Teclado matricial
char tecla='N'; //Guarda tecla pressionada
int ativo=0; //Ativa captura de teclas
int posicao=0; //Guarda posicao de alteracao da String
char numero[16]="0"; //Guarda numero digitado
int debouncer=20; //usada no debouncer
void conta()
{
    if(sensor1==1 && sensor2==0) // verifica se o PRIMEIRO
    { // sensor foi acionado
        if(novo==1) //se variavel novo for 1
        entao pessoa ja passou por um sensor
            pessoa-=1; // Decrementa pessoa da
            quantidade total
    }
```

```
        pcativo=1; //ativa envio de
        dados ao PC
        novo=0; // informa que pessoa
        ja passou
    }
    else
    {
        novo=1; //informa que pessoa
        passou pelo sensor 1
    }
}
if(sensor1==0 && sensor2==1) //
verifica se o SEGUNDO
{ // sensor foi acionado
    if(novo==1) //se variavel novo for
    1 entao pessoa ja passou por um sensor
    {
        pessoa+=1; // Incrementa pessoa
        da quantidade total
        pcativo=1; //Ativa envio de
        dados para o PC
        novo=0; // informa que pessoa
        ja passou
    }
    else
    {
        novo=1; //informa que pessoa
        passou pelo sensor 2
    }
}
//fim
}
#int_TIMER0
void TIMER0_isr(void)
{
    //Parte SENSORES
    if(sensor1!=antsen1 ||
    sensor2!=antsen2)
    {
        conta();
        antsen1=sensor1;
        antsen2=sensor2;
    }
    //PARTE TECLADO
    contKey+=1; //INICIO DRIVER TECLADO
    MATRICIAL
    if(contKey>2) contkey=0;
    switch(contKey)
    {
        case 0:
            output_high(PIN_B0);
            output_low(PIN_B1);
            output_low(PIN_B2);
            break;
        case 1:
            output_high(PIN_B1);
            output_low(PIN_B0);
            output_low(PIN_B2);
            break;
        case 2:
            output_high(PIN_B2);
            output_low(PIN_B1);
            output_low(PIN_B0);
            break;
    }
}
if(debouncer==0)
{
    if(input_state(PIN_A0))
    {
        switch(contKey)
        {
            case 0:
                tecla='1';
                break;
            case 1:
                tecla='2';
                break;
        }
    }
}
```

```

        case 2:
            tecla='3';
        }
        debouncer=20;
    }
    else if(input_state(PIN_A1))
    {
        switch(contKey)
        {
            case 0:
                tecla='4';
                break;
            case 1:
                tecla='5';
                break;
            case 2:
                tecla='6';
            }
        debouncer=20;
    }
    else if(input_state(PIN_A2))
    {
        switch(contKey)
        {
            case 0:
                tecla='7';
                break;
            case 1:
                tecla='8';
                break;
            case 2:
                tecla='9';
            }
        debouncer=20;
    }
    else if(input_state(PIN_A3))
    {
        switch(contKey)
        {
            case 0:
                tecla='#';
                break;
            case 1:
                tecla='0';
                break;
            case 2:
                tecla='*';
            }
        debouncer=20;
    }
    if(tecla=='#') //Ativa captura de
    teclas caso # pressionado
    {
        ativo=1; //entra no loop para
        informa que esta capturando
        posicao=0; // muda para tecla
        capturada seja inserida no
        // primeiro algarismo
        do numero digitado
    }
    if(tecla=='*') //desativa captura de
    teclas caso * pressionado
    {
        ativo=0; //sai do loop de captura
        e volta ao normal
        pessoa=atoi32(numero); // converte
        string em numero
        pcativo=1; // ativa envio para o
        PC
    }
    if(tecla!='#' && tecla!='*' &&
    tecla!='N') //caso tecla não seja #, *
    ou nula
    {
        //capture e guarde na string
        numero[posicao]=tecla;
        posicao+=1;
    }

```

```

        numero[posicao]='\0';
    }
    tecla='N'; // anula tecla capturada
}
if(debouncer>0) debouncer-=1; //FIM
DRIVER TECLADO
}

#int_RB
void RB_isr(void)
{
    if(input_state(PIN_B5))
        sensor1=1;
    else sensor1=0;
    if(input_state(PIN_B4))
        sensor2=1;
    else sensor2=0;
}

void main()
{
    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_CLOCK_DIV_2);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    enable_interrupts(INT_TIMER0);
    enable_interrupts(INT_RB);
    enable_interrupts(GLOBAL);
    usb_cdc_init();
    usb_init();
    usb_task();
    lcd_init();
    // TODO: USER CODE!!
    while(true)
    {
        printf(lcd_putc,"\fQuantidade:\n%li",pes
        soa); //Mostra total de pessoas
        while(ativo==1) //loop para teclas
        pressionadas caso captura ativada
        {
            printf(lcd_putc,"\fAlterar
            para:\n%s",numero);
            delay_ms(4000);
            printf(lcd_putc,"\f"); //
            delays e printf para efeito de piscar
            delay_ms(4000);
        }
        while(pcativo==1) //loop para
        envio de dados para o PC caso ativado
        {
            printf(usb_cdc_putc,"%li",pessoa);
            pcativo=0;
        }
        delay_ms(1000);
    }
}

```

Software do Computador

O software do computador será responsável por mostrar a quantidade atual de pessoas na

sala. O software foi desenvolvido em Delphi 7 onde foram utilizados dois componentes aos quais não estão disponível na versão padrão do Delphi e precisam ser instalados, podendo ser baixado gratuitamente na internet. Os componentes são: *ComPort Library* e *SkinData*.

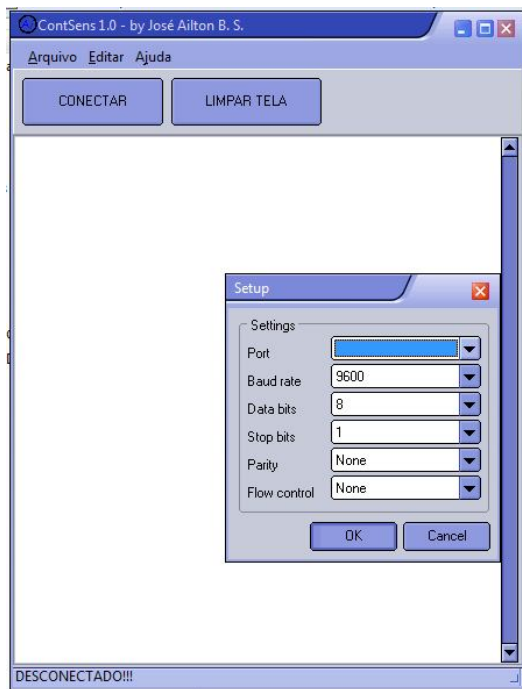


Figura 6

O código-fonte do programa será disponibilizado a seguir devendo ser digitado e compilado usando o Delphi 7 e os componentes citados acima. Caso a compilação correta o programa estará conforme a figura 6.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ComCtrls, StdCtrls,
  ExtCtrls, WinSkinData, CPort, Shellapi;

type
  TForm1 = class(TForm)
    MainMenu: TMainMenu;
    Arquivo1: TMenuItem;
    Conectar1: TMenuItem;
    Salvarinformaes1: TMenuItem;
    Fechar1: TMenuItem;
    Editar1: TMenuItem;
```

```
    Configuraes1: TMenuItem;
    Aalterarfontel: TMenuItem;
    Desconetar1: TMenuItem;
    Ajuda1: TMenuItem;
    Acessarsitel: TMenuItem;
    Sobrel: TMenuItem;
    Panell: TPanel;
    Button1: TButton;
    Memol: TMemo;
    StatusBar1: TStatusBar;
    ComPort1: TComPort;
    SkinData1: TSkinData;
    Button2: TButton;
    FontDialog1: TFontDialog;
    SaveDialog1: TSaveDialog;

    procedure Fechar1Click(Sender:
      TObject);
    procedure SobrelClick(Sender:
      TObject);
    procedure Button2Click(Sender:
      TObject);
    procedure Configuraes1Click(Sender:
      TObject);
    procedure AalterarfontelClick(Sender: TObject);
    procedure FontDialog1Apply(Sender:
      TObject; Wnd: HWND);
    procedure AcessarsitelClick(Sender:
      TObject);
    procedure Salvarinformaes1Click(Sender: TObject);
    procedure Conectar1Click(Sender:
      TObject);
    procedure Desconetar1Click(Sender:
      TObject);
    procedure ComPort1AfterClose(Sender:
      TObject);
    procedure ComPort1AfterOpen(Sender:
      TObject);
    procedure Button1Click(Sender:
      TObject);
    procedure ComPort1RxChar(Sender:
      TObject; Count: Integer);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Fechar1Click(Sender:
  TObject);
begin
  close;
end;

procedure TForm1.SobrelClick(Sender:
  TObject);
begin
  ShowMessage('DESENVOLVIDO
  POR: '+#13+'José Ailton B. S.'+#13+'E-
  MAIL: '+#13+'ailton-
  linux@hotmail.com'+#13+'SITE: '+#13+'www.
  novalogica.wordpress.com');
end;

procedure TForm1.Button2Click(Sender:
  TObject);
begin
  Memol.Lines.Clear;
end;
```

```

procedure
TForm1.Configuraes1Click(Sender:
TObject);
begin
ComPort1.ShowSetupDialog;
end;

procedure
TForm1.AalterarfontelClick(Sender:
TObject);
begin
FontDialog1.Execute;
end;

procedure
TForm1.FontDialog1Apply(Sender: TObject;
Wnd: HWND);
begin
Memol.Font:=FontDialog1.Font;
end;

procedure
TForm1.AcessarsitelClick(Sender:
TObject);
begin
ShellExecute(0,'open','http://www.novalogica.wordpress.com',nil,nil,SW_SHOW);
end;

procedure
TForm1.Salvarinformaes1Click(Sender:
TObject);
begin
if SaveDialog1.Execute then
Memol.Lines.SaveToFile(SaveDialog1.FileName);
end;

procedure TForm1.Conectar1Click(Sender:
TObject);
begin
ComPort1.Open;
end;

procedure
TForm1.Desconectar1Click(Sender:
TObject);
begin
ComPort1.Close;
end;

procedure
TForm1.ComPort1AfterClose(Sender:
TObject);
begin
StatusBar1.Panels[0].Text:='DESCONECTADO!!!';
Button1.Caption:='CONECTAR';
end;

procedure
TForm1.ComPort1AfterOpen(Sender:
TObject);
begin
StatusBar1.Panels[0].Text:='CONECTADO!!!';
Button1.Caption:='DESCONECTAR';
end;

procedure TForm1.Button1Click(Sender:
TObject);
begin
if
StatusBar1.Panels[0].Text='CONECTADO!!!'
THEN
ComPort1.Close

```

```

ELSE
ComPort1.Open;
end;

procedure TForm1.ComPort1RxChar(Sender:
TObject; Count: Integer);
var buffer: AnsiString;
    contador: integer;
begin
{caso receber dados da serial este
procedimento eh executado}
sleep(500); // espere 500 ms
contador:=ComPort1.InputCount; //
capture tamanho do dado
ComPort1.ReadStr(buffer,contador);
//guarde em Buffer
{a linha seguinte adiciona buffer ao
Memol concatenado com data e hora}
Memol.Lines.Add('-----
---');
Memol.Lines.Add('Quantidade Atual é: '+
buffer);
Memol.Lines.Add('Dia: '+
DateToStr(Date));
Memol.Lines.Add('Hora: '+
TimeToStr(Time));
end;

end.

```

Lista de Componentes

6 Resistores de 150R

2 Resistores de 1K

1 Resistor de 2K2

4 Resistores 10K

2 Capacitores de 10uF

2 Capacitores de 22p

1 Microcontrolador PIC18F4550

2 Transistores VC547

1 Diodo 1n4007

2 LED infravermelho

2 fotodiodos

30 pinos 2 cm

1 potenciômetro 1K

1 Cristal 20MHz