



## **SISTEMA DE ACESSO MICROCONTROLADO - S.A.M.**

Alexandre de Brito Melo – 2002.2.18.001-4

Victor Hugo Freitas de Oliveira – 2004.1.18.028-8

Emerson Luis Rodrigues da Costa – 2004.1.18.014-8

Joeldilza Teixeira Barbalho – 2004.1.18.017-2

Jose Soares Batista Lopes

Gerência de Indústria – CEFET-RN

Av. Salgado Filho, 1159 Morro Branco CEP 59.000-000 Natal-RN

E-mails: [alexandre@zuuumtec.com.br](mailto:alexandre@zuuumtec.com.br), [emerson@zuuumtec.com.br](mailto:emerson@zuuumtec.com.br),  
[joedilza@zuuumtec.com.br](mailto:joedilza@zuuumtec.com.br), [victor@zuuumtec.com.br](mailto:victor@zuuumtec.com.br), [jsoares@cefetrn.br](mailto:jsoares@cefetrn.br)

### **RESUMO**

Este trabalho apresenta o desenvolvimento de um Sistema de Acesso Microcontrolado, denominado “S.A.M.”, aplicando para isso um microcontrolador PIC16F877. Trata-se de um sistema onde o usuário irá interagir com um teclado, digitando uma senha, que irá habilitá-lo ou não a entrar em determinado recinto. As senhas dos usuários ficaram armazenadas no microcontrolador, o que faz do projeto um sistema embarcado, onde só é conectado o computador para cadastrar as senhas e leitura dos relatórios de acesso, sendo essa conexão feita através de cabo serial ou remotamente através de celular.

**PALAVRAS-CHAVE:** microcontroladores, automação, comunicação, celular.

## 1. INTRODUÇÃO

A correria do dia-a-dia nos leva a pensar em soluções que liberem cada vez mais a pessoa de levar coisas com elas, como é o caso das chaves. Não existe coisa pior do que você chegar em algum local e perceber que esqueceu as chaves ou, estar com as chaves, mas não encontrar a correta para aquela porta. Fora o problema de carregar-se um molho de chaves enorme. Pior ainda é quando essas chaves são perdidas e a pessoa fica sem o acesso ao seu local de trabalho e/ou moradia.

Tirar das pessoas o peso das chaves, não apenas o peso no sentido real, mas o peso da responsabilidade de estar tendo sempre que carregar alguma coisa, fica mais próximo com o uso de microcontroladores, que estão cada vez mais potentes e baratos.

Este projeto visa dar um exemplo do que pode ser feito com um microcontrolador de baixo custo, 16F877, para controle de acesso a um recinto através de um sistema totalmente microcontrolado, sendo esse sistema um facilitador para a vida das pessoas. Trata-se de um sistema onde o usuário irá interagir com um teclado, digitando uma senha, que irá habilitá-lo ou não a entrar em determinado recinto. As senhas dos usuários ficaram armazenadas no microcontrolador, o que faz do projeto um sistema embarcado, onde só é conectado o computador, para cadastrar as senhas e leitura dos relatórios de acesso.

Para execução do protótipo descrito, foi aplicado conhecimentos de robótica, automação, microprocessadores, microcontroladores, programação, comunicação de dados, eletrônica digital e analógica. Sendo então um projeto multidisciplinar.

### 1.1. Visão Geral do Sistema

O Sistema de Acesso Microcontrolado S.A.M. é constituído de 2 módulos (Fig. 1), sendo eles:

- 1.1.1. Módulo de controle: Composto do software feito em Delphi onde será feito o cadastro dos usuários e recuperação de relatórios de acesso ao recinto.
- 1.1.2. Módulo embarcado: Consiste em placa mãe do sistema, onde ficam ligados o: sensor, teclado e fechadura elétrica. Esse módulo também é conectado ocasionalmente ao Módulo de Controle para cadastro de usuários e descarga de relatórios através da porta serial, podendo também ser um acesso remoto via celular.

### 1.2. Diagrama Geral

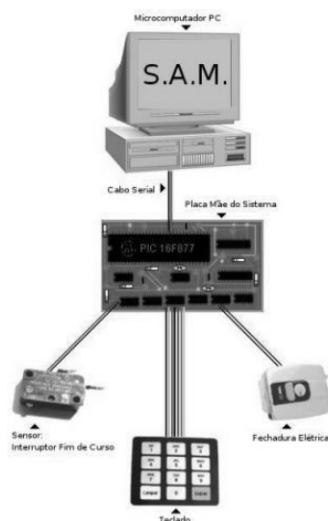


Fig. 1- Diagrama geral do sistema proposto

## 2. HARDWARE

### 2.1.Módulo de Controle

O hardware do Módulo de Controle pode ser qualquer microcomputador PC com sistema operacional Windows e também com capacidade de rodar um programa em Delphi leve, tendo ainda que ter uma porta serial livre para fazer as conexões via cabo ou celular com o Módulo embarcado. Ou seja, a grande maioria dos computadores em uso hoje em dia estão dentro das especificações.

### 2.2.Módulo Embarcado

O hardware do Módulo Embarcado consiste em uma placa principal onde é feito o controle de acesso das pessoas ao recinto. Este módulo está conectado a alguns periféricos como: sensor para verificar se a porta está aberta ou fechada, teclado para digitação da senha com liberação ou não do acesso e uma fechadura elétrica para abrir a porta.

Este módulo é baseado no microcontrolador PIC 16F877(Fig. 2), devido a sua robustez e facilidade de uso como podemos ver na sua pinagem(Fig. 3) e diagrama de blocos(Fig. 4). Para o protótipo construído, ele está sendo subutilizado, mas para o projeto final proposto, onde serão conectados vários teclados, mouses e sensores para controlar o acesso de até 8 portas para cada microcontrolador, ele está bem dimensionado.



Fig. 2 – Aspecto físico do encapsulamento do PIC 16F877 na versão DIP usada no protótipo

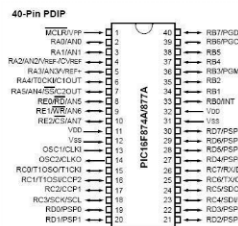


Fig. 3 – Pinagem do PIC 16F877

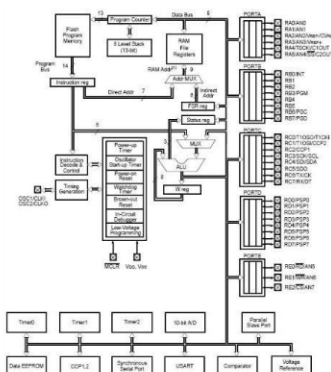


Fig. 4 – Diagrama de Blocos do PIC 16F877

Como haverá uma comunicação serial envolvida no processo de cadastro de senhas e recuperação de relatórios, optou-se por usar um oscilador externo XTAL de 4Mhz para maior precisão dos clocks na comunicação serial. O circuito oscilador externo usado no circuito consiste em 2 capacitores aterrados e ligados ao oscilador, que estará ligado aos pinos 13 e 14 do PIC, como podemos ver na figura 5

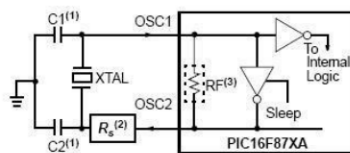


Fig. 5 – Oscilador externo

A comunicação do Módulo embarcado se dará através de uma porta serial, para tanto, precisamos transformar o sinal padrão TTL dos pinos 25 e 26 do PIC em sinal padrão RS-232. Usaremos o CI MAX232 para tal tarefa em um circuito como mostrado na figura 6.

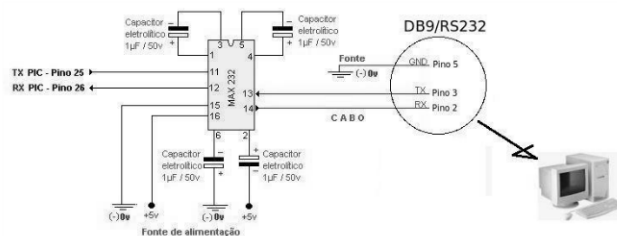


Fig. 6 – Circuito conversor serial para comunicação entre os dois módulos

Para comunicação dos módulos através de acesso remoto celular, usaremos o circuito da figura 7, que nada mais é do que dois CI's montados dentro do conector do cabo da figura 8, o qual o fabricante chama comercialmente de DLR-3P. Os dois CI's são o MAX232 e o microcontrolador 16LC63. Esse cabo é quem transforma a interface do celular em serial para ser conectado na porta do Módulo de Controle.

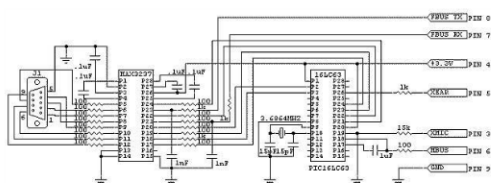


Fig. 7 – Esquema do circuito do cabo DLR-3P



Fig. 8 – Cabo DLR-3P

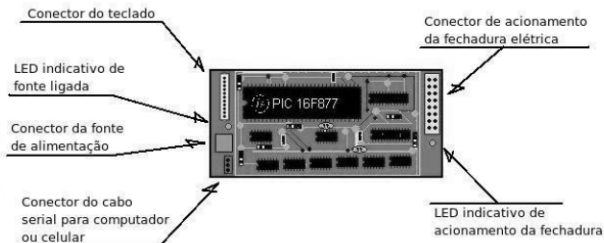


Fig. 9 – Conceção da placa do Módulo Embarcado.

Como o circuito é um protótipo, o microcontrolador usado ficou com vários pinos sobrando, o que facilitou bastante a escolha dos mesmos, optando então por pinos bem distribuídos no controlador, o que facilitaria na hora da confecção da placa. Segue abaixo (Fig. 10) os pinos e suas respectivas funções para o protótipo do sistema.

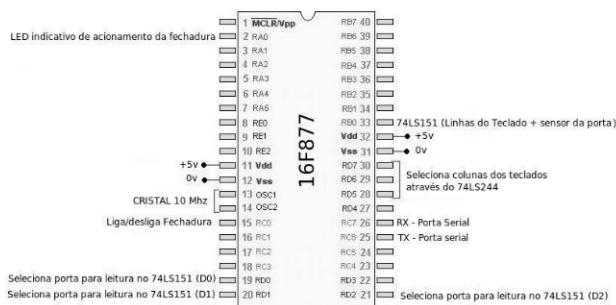


Fig. 10 – Uso dos pinos no protótipo

O PIC ficará escaneando as linhas do teclado e sensor da porta através de um circuito multiplexador 74LS151. A conexão dos circuitos será nos pinos 19, 20, 21 e 23 do portD, como mostrado na figura 11

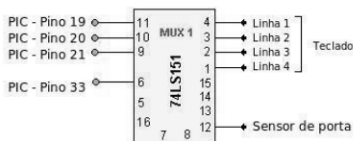


Fig. 11 – Detalhe do circuito de conexão das linhas do teclado e sensor da porta.

A entrada do sinal nas colunas do teclado será por um 74LS244, que é um buffer com saídas não invertidas. A conexão desse CI ao PIC será também pelo portD nos pinos 28, 29 e 30, como mostrado na figura 12 a seguir.

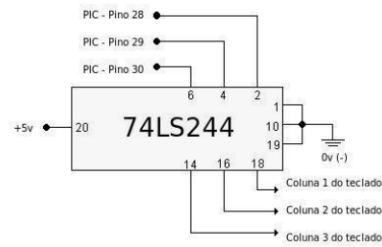


Fig. 12 – Detalhe de conexão das colunas do teclado

O etapa de potência (Fig. 13) que vai controlar a fechadura elétrica, é composta de um relé 12V/10A, um TIP31C e mais resistores e diodos. Ela receberá um sinal de 5 V do pino 15 do PIC, que acionará o TIP e por sua vez a bobina do relé, que entrará em curto nos terminais NA, ativando a fechadura.

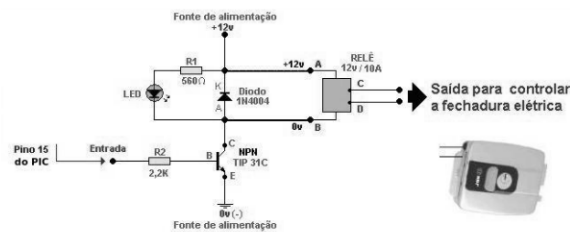


Fig. 13 – Driver de controle da fechadura

Os circuitos acima descritos foram montados e desenhados no Eagle (fig. 14), não sendo montado a placa de circuito impresso devido o número de componentes e a placa ficarem maiores do que o especificado na versão free do software usado.

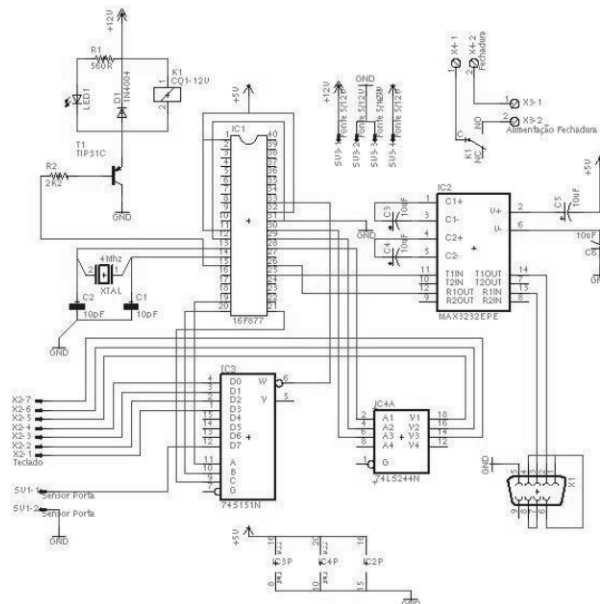


Fig. 14 – Circuito completo do Módulo Embarcado.

### 3. SOFTWARE

#### 3.1.Módulo de Controle

O software montado para rodar no Módulo de Controle foi feito em Delphi e é bastante leve. Ele faz o cadastro de senhas no microcontrolador através da serial conectada no Módulo Embarcado e solicita uma confirmação para cadastro de senha. Ainda estão sendo desenvolvidos os códigos para acesso remoto via celular e recuperação do relatório de acesso. Segue abaixo as telas e códigos do software do módulo de controle

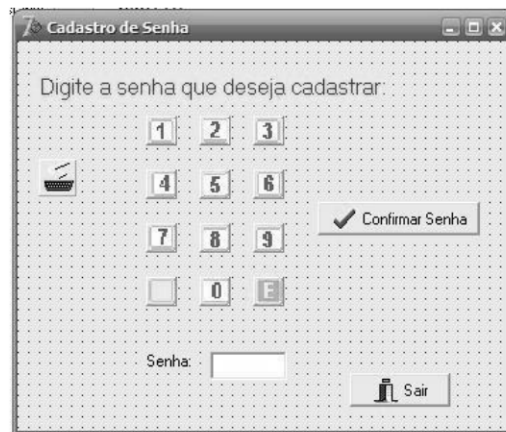


Fig. 15 – Tela principal do software do Módulo de Controle

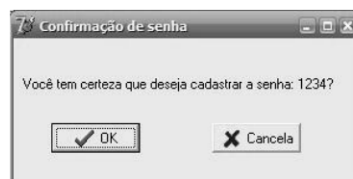


Fig. 16 – Tela de confirmação de senha

```
unit Unit1;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, Buttons, ComPort, Mask;  
  
type  
  TForm1 = class(TForm)  
    SpeedButton1: TSpeedButton;  
    SpeedButton2: TSpeedButton;  
    SpeedButton3: TSpeedButton;  
    SpeedButton4: TSpeedButton;  
    SpeedButton5: TSpeedButton;  
    SpeedButton6: TSpeedButton;  
    SpeedButton7: TSpeedButton;  
    SpeedButton8: TSpeedButton;  
    SpeedButton9: TSpeedButton;  
    SpeedButton10: TSpeedButton;  
    SpeedButton11: TSpeedButton;  
    SpeedButton12: TSpeedButton;  
    ComPort1: TComPort;  
    BitBtn1: TBitBtn;  
  end;  
end;
```

```

Label1: TLabel;
Edit1: TEdit;
Label2: TLabel;
BitBtn2: TBitBtn;
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure SpeedButton11Click(Sender: TObject);
procedure SpeedButton10Click(Sender: TObject);
procedure SpeedButton12Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);

private
{ Private declarations }
public
{ Public declarations }

end;

var
Form1: TForm1;
senha: string;
tam: integer;

implementation

{$R *.dfm}

uses Unit2;

{botão 1 pressionado, adiciona '1' na string "senha".
funcionamento identico para todos os botões numéricos.}
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    senha := senha + '1';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    senha := senha + '2';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    senha := senha + '3';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
    senha := senha + '4';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
    senha := senha + '5';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
    senha := senha + '6';
    Edit1.Text := senha;
end;

```



```

procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
    senha := senha + '7';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
    senha := senha + '8';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
    senha := senha + '9';
    Edit1.Text := senha;
end;

procedure TForm1.SpeedButton11Click(Sender: TObject);
begin
    senha := senha + '0';
    Edit1.Text := senha;
end;

{Limpa a senha.}
procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
    senha := '';
    Edit1.Text := senha;
end;

{Checa se a senha está correta e envia pela porta serial.}
procedure TForm1.SpeedButton12Click(Sender: TObject);
begin
    tam := length(senha);
    if (tam <> 4) then
    begin
        ShowMessage('A senha deve conter 4 dígitos.');
```

exit;

end;

```

Form2 := TForm2.Create(Application);
Form2.Label1.Caption := 'Você tem certeza que deseja cadastrar a senha: ' + senha + '?';
if (Form2.ShowModal = 1) then
begin
    //ShowMessage('Apos instalado o ComPort1, descomente o código');
    //ComPort1.Port := Com1;
    ComPort1.Open;
    ComPort1.Send(senha + '#13#10');
```

end;

end;

```

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    Close;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    SpeedButton12Click(Sender);
end;

end.
```

```

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons;

type
  TForm2 = class(TForm)
    Label1: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

end.

```

### 3.2.Módulo Embarcado

O Módulo Embarcado contém um programa com seu código fonte em “C” que foi feito usando-se o CCS(compilador C para microcontroladores), depois ele é transformado em assembler para então ser compilado em linguagem de máquina. Depois dessa etapa, é feita a gravação do código no microcontrolador usando um gravador conectado na serial do microcomputador.

Esse programa é responsável pela comunicação serial e armazenagem das senhas na memória flash do microcontrolador. Segue abaixo o código fonte do software descrito.

```

#include <16f877a.h>
#define delay(clock=4000000)
#define HS,NOWDT,PUT
#include <usart.c>

//Keypad connection:

#define col0 PIN_B4
#define col1 PIN_B5
#define col2 PIN_B6
#define row0 PIN_B3
#define row1 PIN_B2
#define row2 PIN_B1
#define row3 PIN_B0

// Keypad layout:
char const KEYS[4][3] = {{'1','2','3'},
                        {'4','5','6'},
                        {'7','8','9'},
                        {'*','0','#'}};

#define KBD_DEBOUNCE_FACTOR 33 // Set this number to apx n/333 where
// n is the number of times you expect
// to call kbd_getc each second

void kbd_init() {
  port_b_pullups(true);
}

```

```

short int ALL_ROWS (void)
{
    if (input (row0) & input (row1) & input (row2) & input (row3))
        return (0);
    else
        return (1);
}

char kbd_getc( ) {
    static byte kbd_call_count;
    static short int kbd_down;
    static char last_key;
    static byte col;

    byte kchar;
    byte row;

    kchar='\0';
    if(++kbd_call_count>KBD_DEBOUNCE_FACTOR) {
        switch (col) {
            case 0 : output_low(col0);
                      output_high(col1);
                      output_high(col2);
                      break;
            case 1 : output_high(col0);
                      output_low(col1);
                      output_high(col2);
                      break;
            case 2 : output_high(col0);
                      output_high(col1);
                      output_low(col2);
                      break;
        }

        if(kbd_down) {
            if(!ALL_ROWS()) {
                kbd_down=false;
                kchar=last_key;
                last_key='\0';
            }
        } else {
            if(ALL_ROWS()) {
                if(!input (row0))
                    row=0;
                else if(!input (row1))
                    row=1;
                else if(!input (row2))
                    row=2;
                else if(!input (row3))
                    row=3;
                last_key=KEYS[row][col];
                kbd_down = true;
            } else {
                ++col;
                if(col==3)
                    col=0;
            }
        }
        kbd_call_count=0;
    }
    return(kchar);
}

#rom 0x2100={'1','2','3','4'}

int num_senhas=0; //Variável que guarda a quantidade de senhas cadastradas
int i;

#int_rda
void interrupcao(){
    char aux[4]; //Recebe a senha passado pelo pc

```



#### **4. CONCLUSÃO**

O Sistema de Acesso Microcontrolado é um sistema de segurança simples, porém viável e que cumpre o seu objetivo principal que é libertar as pessoas do peso de carregar várias chaves durante a sua rotina diária.

#### **5. REFERÊNCIAS BIBLIOGRÁFICAS**

Pereira – Fábio, **Microcontroladores PIC – Programação em C**. Ed. Érica, 2003

Martins – Nardênio A., **Sistemas Microcontrolados**. Ed. Novatec, 2005

Gimenez – Salvador P., **Microcontroladores 8051**. Ed. Prentice Hall, 2002