

NEA630 – MICROPROCESSADORES

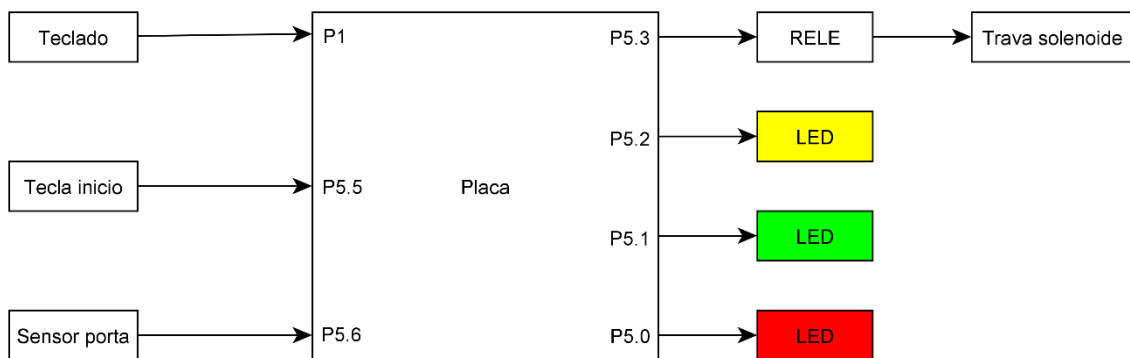
Projeto para P1

Projeto

A proposta foi projetar o software de um cofre. O funcionamento seria simples, o cofre permanece trancado, até o acionamento de um botão para a inserção da senha e destravamento da porta.

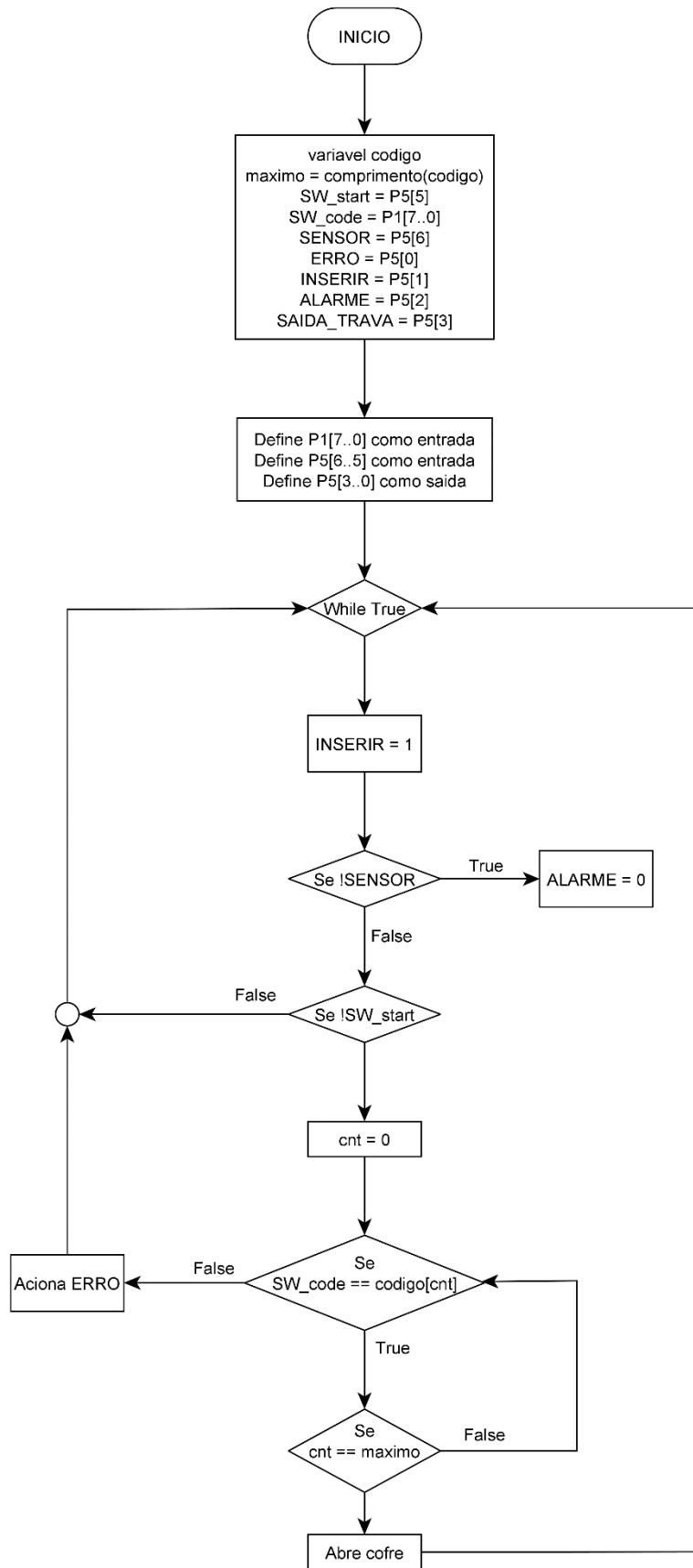
- A senha é programada por software.
- Existe um sensor para detectar se a porta foi aberta ou não.
- Um alarme é acionado se a porta for violada.
- Se o usuário errar a senha não vai ativar o alarme.
- Uma vez acionado, o alarme só pode ser desativado através de uma chave que ativa o reset do sistema.

Diagrama simplificado do projeto:



As simulações e códigos foram feitos no software "IAR EW for Renesas RL78 4.20.1", configurado para o hardware "RL78 - R5F100LE".

Fluxograma



Código fonte

```
#include "ior5f100le.h"
#include "ior5f100le_ext.h"
#include "intrinsics.h"
#include "myRL78.h"

// Configura watchdog = desligado
#pragma location = "OPTBYTE"
__root __far const char opbyte0 = WDT_OFF;
// Configura detector de baixa tensão= desligado
#pragma location = "OPTBYTE"
__root __far const char opbyte1 = LVD_OFF;
// oscilador 32MHz flash high speed
#pragma location = "OPTBYTE"
__root __far const char opbyte2 = FLASH_HS | CLK_32MHZ;
// debug ativado, com apagamento em caso de falha de autenticação
#pragma location = "OPTBYTE"
__root __far const char opbyte3 = DEBUG_ON_ERASE;
/* Configura security ID */
#pragma location = "SECUID"
__root __far const char senha[10] = {0,0,0,0,0,0,0,0,0,0};

#define SW_start PM5_bit.no5
#define SW_code PM1
#define SENSOR PM5_bit.no6

#define ERRO PM5_bit.no0
#define INSERIR PM5_bit.no1
#define ALARME PM5_bit.no2
#define SAIDA_TRAVA PM5_bit.no3

volatile unsigned long int temp;
volatile unsigned int cnt;

// Senha em hexadecimal
unsigned char codigo[2] = {
    0xF7,
    0xBF
};
// Numero de digitos da senha
unsigned int maximo = sizeof(codigo);

void aciona_alarme( void )
{
    ALARME = 0;
    while (1);
}

void aciona_erro( void )
{
    ERRO = 0;
    for (temp=5;temp;temp--); // Tempo de espera do erro
    ERRO = 1;
}

void aciona_trava( void )
{
    SAIDA_TRAVA = 0;
    while(SENSOR); // Espera a porta abrir
    SAIDA_TRAVA = 1;
}
```

```

while(!SENSOR); // Espera a porta fechar
}

void main( void )
{
    // Set de portas
    PM1 = 0xFF; // Define a porta Plx como entrada (Ent. senha)
    PM5_bit.no5 = 1; // Define a porta P55 como entrada (Tecla inicio)
    PM5_bit.no6 = 1; // Define a porta P56 como entrada (Sensor porta)

    PM5_bit.no0 = 0; // Define a porta P50 como saída (LED de erro)
    PM5_bit.no1 = 0; // Define a porta P51 como saída (LED de inserção
de código)
    PM5_bit.no2 = 0; // Define a porta P52 como saída (Alarme)
    PM5_bit.no3 = 0; // Define a porta P53 como saída (Trava)

    ERRO = 1; //
    ALARME = 1; // Desativa o alarme
    SAIDA_TRAVA = 1; // Deixa a porta travada

    while (1)
    {
        cnt = 0;
        INSERIR = 1; // Apaga a luz apos a inserção do
código/destravamento
        if (!SENSOR)
        {
            aciona_alarme();
        }

        // Tecla de abertura acionada
        if (!SW_start)
        {
            INSERIR = 0; // Acende LED para inserir o código
            for (temp=5;temp;temp--)
            {
                // Responde apenas se caso as chaves forem acionadas
                if (SW_code != 0xFF)
                {
                    if (SW_code == codigo[cnt])
                    {
                        cnt++;
                        // Se o numero de entradas for igual a do código aciona
a trava
                        if (cnt == maximo)
                        {
                            aciona_trava();
                            break;
                        }
                    }
                    else
                    {
                        aciona_erro();
                        break;
                    }
                }
            }
        }
    }
}

```

Nota: os valores nos temporizadores são valores teóricos para simulação, não são valores práticos. O cálculo dos valores práticos na parte de simulação.

Simulação

Para o cálculo do tempo decorrido, utilizamos o registrador do cycle conter, que nos indica o número de ciclos da máquina, o clock utilizado é de 32 MHz, assim podemos calcular o tempo multiplicando o número de ciclos por $1/32e6$.

Execução até o início do loop:

```
70 PM5_bit.no1 = 0; // Define a
71 PM5_bit.no2 = 0; // Define a
72 PM5_bit.no3 = 0; // Define a
73
74 cnt = 0;
75
76 ERRO = 1; //
77 ALARME = 1; // Desativa o al
78
79 while (1)
80 {
81     INSERIR = 1; // Apaga a lu
82     if (!SENSOR)
83     {
84         aciona_alarme();
85     }
86
87     // Tecla de abertura acion
88     if (!SW_start)
89     {
90         INSERIR = 0; // Acende L
91         for (temp=5;temp;temp--)
92         {
```

Até o início do loop eterno, o sistema demorou 1370 ciclos ou $42,81e-6$ s, valor baixo de inicialização, pois não há nenhuma verificação de componente.

Registers 2		
Find:	Group: CPU Registers	
Name	Value	Access
AX	0x0000	ReadWrite
BC	0x0000	ReadWrite
DE	0xFFFF	ReadWrite
HL	0x026E	ReadWrite
SP	0xEF86	ReadWrite
PC	0x00176	ReadWrite
PSW	0x06	ReadWrite
CS	0x00	ReadWrite
ES	0x00	ReadWrite
CYCLECOUNTER	1368	ReadOnly
CCTIMER1	1368	ReadWrite
CCTIMER2	1368	ReadWrite
CCSTEP	2	ReadOnly

A tabela dos registradores das portas, para simplificar, criei uma lista personalizada apenas com os registradores que estão sendo utilizados.

Registers 1		
Find:		Group: teste
Name	Value	Access
PH1	0xFF	ReadWrite
PH5	0b11111111	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

Alterando o valor do bit 6 da porta 5 simulamos um arrombamento

Registers 1		
Find:		Group: teste
Name	Value	Access
PH1	0xFF	ReadWrite
PH5	0b10111111	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

E o código executa a operação para ativar o alarme.

```

82  INSERIR = 1; // Apaga a luz apos a inserção do codigo/destroa
83  if (!SENSOR)
84  {
85      aciona_alarme();
86  }
87
88  // Tecla de abertura acionada
89  if (!SW_start)

```

Onde ele fica no loop eterno até o reset.

```

42  void aciona_alarme( void )
43  {
44      ALARME = 0;
45      while (1);
46  }

```

Registers 1		
Find:		Group: teste
Name	Value	Access
PH1	0xFF	ReadWrite
PH5	0b101111011	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

Ao simular a inserção de código.

Registers 1		
Find:		Group: teste
Name	Value	Access
PM1	0xFF	ReadWrite
PM5	0b11011111	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

O código entra no loop para inserção do código.

```
89  if (!SW_start)
90  {
91      INSERIR = 0; // Acende LED para inserção
92      for (temp=5;temp;temp--)
93      {
94          // Responde apenas se caso as chaves
95          if (SW_code != 0xFF)
96          {
97              if (SW_code == codigo[cnt])
98              {
99                  cnt++;
100                 // Se o numero de entradas for
101                 if (cnt == maximo)
102                 {
103                     aciona_trava();
104                     break;
105                 }
106             }
107             else
108             {
109                 aciona_erro();
110                 break;
111             }
112         }
113     }
114 }
```

Registers 1		
Find:		Group: teste
Name	Value	Access
PM1	0xFF	ReadWrite
PM5	0b11011101	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

O *for* demora 4 ciclos para configurar, e o ciclo em si demora 27 ciclos, quando a tecla é pressionada e ocorre o evento de comparação são mais 16 ciclos, considerando 2 minutos como um bom tempo para inserir a senha, o valor do loop poderá ser de 142.222.222 ciclos de loop.

Após inserir o código corretamente, o programa avança para a rotina de destravamento da porta

```

95 // Responde apenas se caso as chaves forem acionadas
96 if (SW_code != 0xFF)
97 {
98     if (SW_code == codigo[cnt])
99     {
100         cnt++;
101         // Se o numero de entradas for igual
102         if (cnt == maximo)
103         {
104             aciona_trava();
105             break;
106         }
107     }
108     else
109     {
110         aciona_erro();
111         break;
112     }
113 }

```

Name	Value	Access
PM1	0xBF	ReadWrite
PM5	0b11111101	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

Essa rotina envia um sinal para o rele, para ativar a trava.

```

51 for (temp=5;temp;temp--); // tempo de espera do erro
52 ERRO = 1;
53 }
54
55 void aciona_trava( void )
56 {
57     SAIDA_TRAVA = 0;
58     while(SENSOR); // Espera a porta abrir
59     SAIDA_TRAVA = 1;
60     while(!SENSOR); // Espera a porta fechar
61 }
62
63 void main( void )
64 {
65     // Set de portas
66     PM1 = 0xFF; // P1
67     PM5 = 0xFF; // P5
68 }

```

Name	Value	Access
PM1	0xFF	ReadWrite
PM5	0b11110101	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

O programa aguarda a abertura da porta antes de desativar a trava, e então, aguarda a porta se fechar para voltar para o programa principal, evitando que o alarme acione.

No caso de um dígito errado, o programa aciona a rotina de erro

```

93 for (temp=5;temp;temp--);
94 {
95     // Responde apenas se caso as chaves forem acionadas
96     if (SW_code != 0xFF)
97     {
98         if (SW_code == codigo[cnt])
99         {
100             cnt++;
101             // Se o numero de entradas for igual
102             if (cnt == maximo)
103             {
104                 aciona_trava();
105                 break;
106             }
107         }
108         else
109         {
110             aciona_erro();
111             break;
112         }
113     }
114 }

```

Name	Value	Access
PM1	0xAE	ReadWrite
PM5	0b11111101	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

Onde acende o LED de erro e aguarda um tempo antes de retornar ao loop principal.

43 {
44 ALARME = 0;
45 while (1);
46 }
47
48 void aciona_erro(void)
49 {
50 ERRO = 0;
51 for (temp=5;temp;temp--); // Tempo de espera do erro
52 ERRO = 1;
53 }
54
55 void aciona_trava(void)
56 {
57 SAIDA_TRAVA = 0;
58 while(SENSOR); // Espera a porta abrir
59 SAIDA_TRAVA = 1;
60 while(!SENSOR); // Espera a porta fechar

Registers 1

Find: Group: teste

Name	Value	Access
PM1	0xAE	ReadWrite
PM5	0b11111100	ReadWrite
P1	0xFF	ReadWrite
P5	0b11111111	ReadWrite

Considerações

O sistema da RENESAS RL78 é bem completo, conta inclusive com um sistema de Real Time Clock (RTC), que pode ser utilizado nesse projeto para gerar logs de acesso e tokens/hash de senhas periodicamente.