NEA630 – MICROPROCESSADORES

Projeto para P1

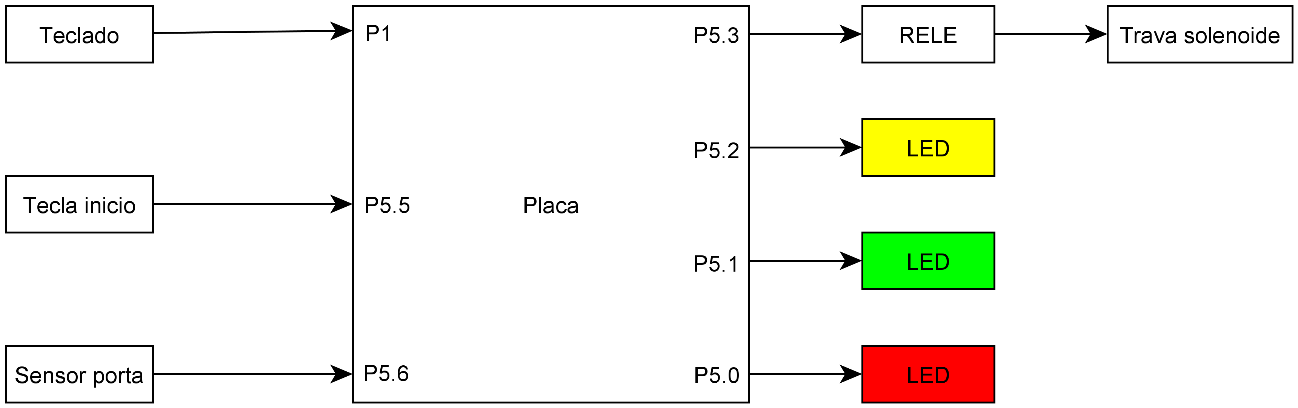
Gustavo Ryuji Sanomia 12.115.481-9

**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.

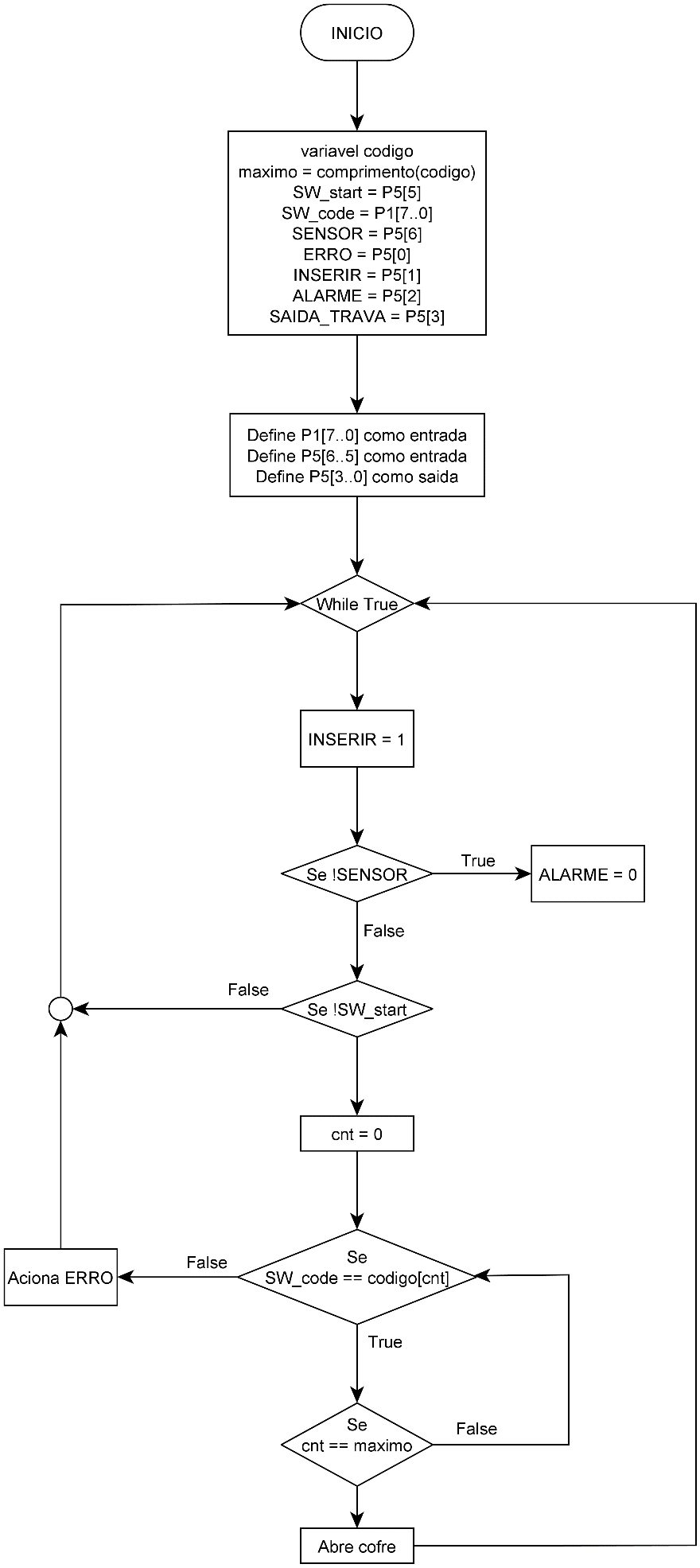
* A senha é programada por software.
* Existe um sensor para detectar se a porta foi aberta ou não.
* Um alarme é acionado se a porta for violada.
* Se o usuário errar a senha não vai ativar o alarme.
* Uma vez acionado, o alarme só pode ser desativado através de uma chave que ativa o reset do sistema.

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**

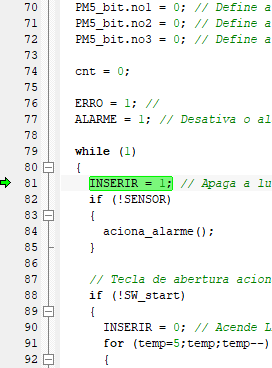
|  |
| --- |
| #include "ior5f100le.h"  #include "ior5f100le\_ext.h"  #include "intrinsics.h"  #include "myRL78.h"  // Configura watchdog = desligado  #pragma location = "OPTBYTE"  \_\_root \_\_far const char opbyte0 **=** WDT\_OFF**;**  // Configura detector de baixa tensão= desligado  #pragma location = "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 P1x 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 cdigo)  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.

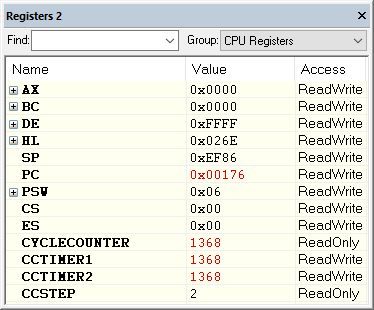
**Simulação**

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.

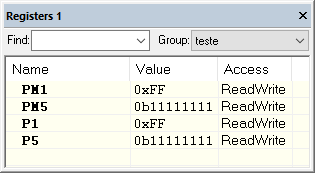
Execução até o início do loop:



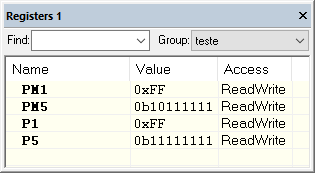
Até o início do loop eterno, o sistema demorou 1370 ciclos ou 42,81e-6 s, valor baixo de inicialização, pois não há nenhuma verificação de componente.



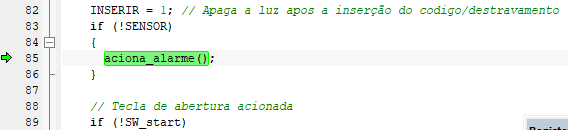
A tabela dos registradores das portas, para simplificar, criei uma lista personalizada apenas com os registradores que estão sendo utilizados.



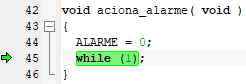
Alterando o valor do bit 6 da porta 5 simulamos um arrombamento

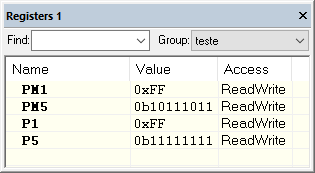


E o código executa a operação para ativar o alarme.

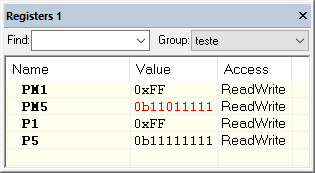


Onde ele fica no loop eterno até o reset.

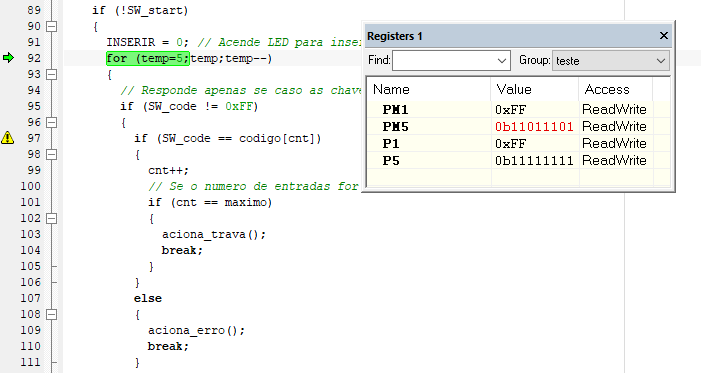




Ao simular a inserção de código.

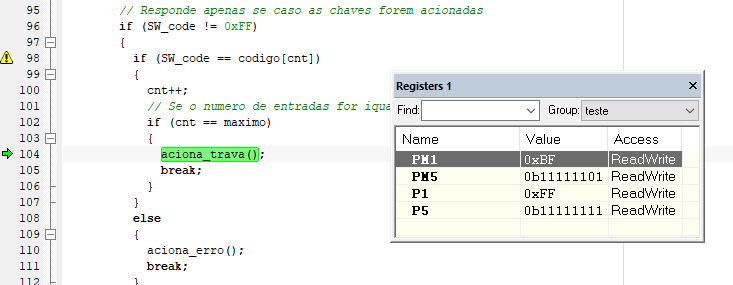


O código entra no loop para inserção do código.

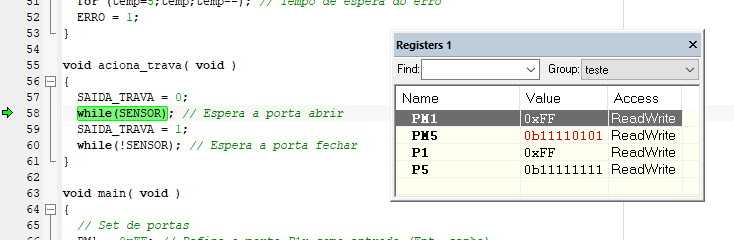


O *for* demora 4 ciclos para configurar, e o ciclo em si demora 27 ciclos, quando a tecla é pressionada e ocorre o evento de comparação são mais 16 ciclos, considerando 2 minutos como um bom tempo para inserir a senha, o valor do loop poderá ser de 142.222.222 ciclos de loop.

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

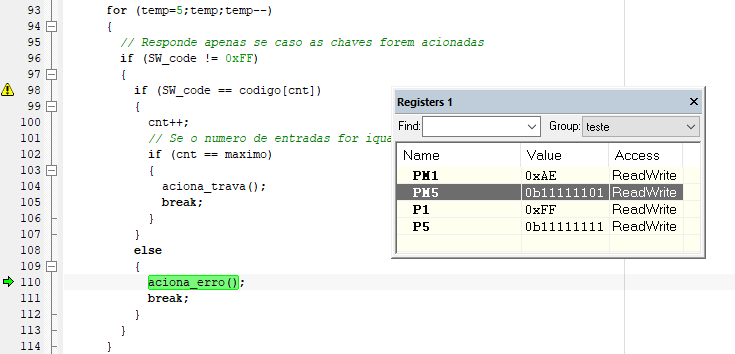


Essa rotina envia um sinal para o rele, para ativar a trava.

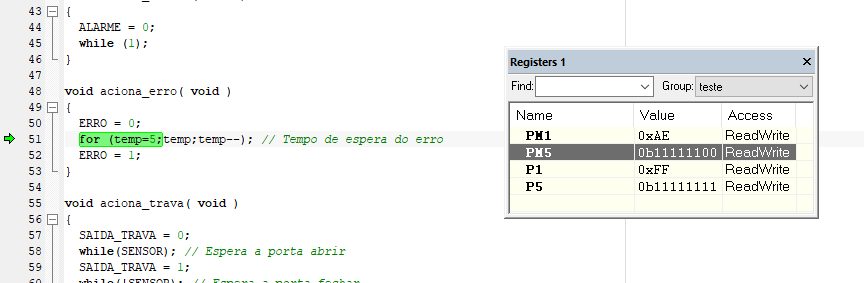


O programa aguarda a abertura da porta antes de desativar a trava, e então, aguarda a porta se fechar para voltar para o programa principal, evitando que o alarme acione.

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



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



**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.