Sistema de Controlo de Alarme Doméstico

Universidade de Aveiro

Henrique Silva, Márcia Pires

Sistema de Controlo de Alarme Doméstico

Departamento de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

Henrique Silva, Márcia Pires
(88857) macias66@ua.pt, (88747) marcia.pires@ua.pt

junho de 2018

Conteúdo

Capítulo 1

Introdução

O projeto proposto consiste na implementação de um sistema de controlo de alarme doméstico. Este alarme pode ser disparado de dois diferentes modos, o interno e o externo. No primeiro caso, este pode disparar sem que esteja armado e basta para isso acionar ou um botão de pânico ou sensores específicos. Já em modo externo, o alarme necessita estar armado, através de um botão, e, quando acionados sensores da janela ou do interior, é disparado. Em qualquer dos casos, quando o alarme é disparado, um conjunto de LED's piscará intermitentemente e tais só irão parar caso seja introduzido o código sequencial correto para proceder ao desamarmento do alarme. Ao final de 3 tentativas incorretas, o alarme irá disparar. Para permitir ao utilizador acompanhar se o alarme está desligado ou ligado, tal informação irá aparecer nos displays de 7 segmentos, bem como o temporizador de determinadas condições.

Capítulo 2

Arquitetura

O sistema, esquematizado em anexo (Anexo1), é composto por duas máquinas de estados: KeyComparator e Alarm. A KeyComparator é a responsável por comparar o código introduzido pelo utilizador com o código previamente definido. Assim sendo, a saída desta máquina de estados transparece principalmente no valor do código, ou seja, se ele é correto ('1') ou incorreto ('0') e servirá para

fazer avançar (ou não) a outra máquina de estados, o Alarm. Posto isto, esta última máquina de estados tem como objetivo controlar os estados do alarme: armado, desarmado e disparado e o avanço dos seus estados depende da saída da máquina de estados anterior. Desta forma, ambas estão sincronizadas de forma a trabalharem como se de uma grande máquina de estados se tratasse. Para controlar todo o sistema, temos o FreqDivider que é o que altera a frequência de 50MHz do clock inicial proveniente da máquina para a frequência de 1Hz.

Dois blocos bastante semelhantes, são o Blink e o RandomBlink pois o Blink serve para fazer o LEDG(0) piscar à frequência de 1Hz quando o alarme estiver ligado. Já o RandomBlink fará todos os LEDR piscarem a uma frequência mais elevada indicando que o sistema está disparado. Outros dois blocos semelhantes são o Timer e o Timer10, já que o primeiro faz uma contagem decrescente de 20 segundos, indicando o tempo que o utilizador tem para sair de casa sem fazer disparar o alarme, e o segundo, de 10, para controlar o tempo que o utilizador tem para desarmar o alarme caso um sensor de interior seja ativado. Estes valores aparecem nos displays de 7 segmentos através, primeiramente do bloco Bin2BCD que faz a conversão dos valores para BCD para que o bloco Bin7SegDecoder os consiga transmitir nos displays (HEX7, HEX6, HEX5, HEX4). Para que a indicação de que o alarme está disparado ou não, já não é necessário o bloco de Bin2BCD mas sim o Bin7SegStates que, do mesmo modo que o Bin7SegDecoder, mostra nos displays de 7 segmentos (HEX0) o pretendido.

Capítulo 3

Implementação

Tal como referido anteriormente, de todos os blocos pertencentes à arquitetura implementada, salienta-se a importância das duas máquinas de estados: KeyComparator e Alarm. Isto acontece visto que a grande ação do projeto pretendido é controlada através da combinação destas. É, por isso, pertinente analisar estas máquinas de estado detalhadamente.

A máquina de estado Alarm (anexo 2) é a que controla o estado do sistema e inicia-se no estado Unready: aqui aguarda-se que algum sensor e/ou botão de pânico seja ativado, avançando para o estado seguinte, caso contrário permanece em Unready. O estado seguinte é o Ready que é aquele onde há uma preparação da informação proveniente de todos os *inputs*. Assim sendo, se o botão de pânico

ou os sensores específicos forem ativados, ele passa imediatamente para o estado Triggered, ou seja, o alarme dispara. Se os restantes sensores forem desativados, então regressam ao estado Unready. Mas, se nenhum dos sensores estiver ativado e, enquanto isso, o alarme for ativado, através do *SW(16)*, então significa que o sistema passa para o estado Armed. Qualquer outra ação fará permanecer em Ready.

No estado Armed, passados os 20 segundos de espera, se qualquer sensor for ativado, avança para o estado Triggered, exceto se esse sensor for do interior: neste caso, aguarda 10 segundos e só se, nesse intervalo de tempo não for introduzido um código correto, passa para o estado Triggered, se não passa para o estado Disarmed, estando o alarme assim desarmado. Se com o alarme armado e nenhum sensor ativo, o código introduzido for o correto, avança para o estado Disarmed. E, no estado Triggered, à semelhança deste, avança para o estado Disarmed se o código introduzido for correto, caso contrário permanece. E, por fim, no estado Disarmed, se novamente todos os sensores estiverem desativados, retorna ao estado Unready.

Por sua vez, a máquina de estados KeyComparator (anexo 3) é aquela que nos permite verificar se o código introduzido é, de facto, correto ou não. Posto isto, iniciamos no estado S0 que aguarda pelo primeiro código da sequência: se este for correto, avança para o estado S1, mas se não, avança para o estado E1. A lógica mantém-se, ou seja, no estado S1, se o código introduzido posteriormente for correto, avança para S2, se não, para E2. E se, por fim, o último código for correto, avança para S3, e se não, para E3. Se chegar a S3, significa que o código em si está correto e, portanto pode transmitir essa informação à máquina de estados Alarm. No entanto, se o último estado for o E3, significa que não concluiu a sequência do código acertadamente. No estado E3 é também possível fazer uma contagem para verificar se o número de tentativas erradas é, ou não, superior a 3.

Capítulo 4

Validação

De modo a testar a viabilidade e fidelidade dos blocos, em particular das máquinas de estado, foram feitas *testbenches* para estas. Sendo os resultados positivos para ambas as máquinas de estado, consideramos ainda mais pertinente realizar uma *testbench* da Shell do projeto, uma vez que o mais importante seria não

o comportamento individual, mas o comportamento sincronizado, como se de uma grande máquinas de estados se tratasse. Para esses efeitos, e uma vez que apenas existe um tipo de *inputs*, foram-se variando quais destes estavam ativos e quais não e analisado o comportamento. O único problema relevante que foi detetado foram os LEDG que ficam ativos numa posição bastante inconstante e incoerente. Fora isso, o comportamento geral foi o esperado, considerando que a validação do trabalho feito foi bem sucedida.

Capítulo 5

Conclusão

Atendendo aos objetivos propostos e considerados inicialmente, a implementação não correu tal como esperado pois, após um melhor aprofundamento e consolidação dos conteúdos, foi percetível que algumas ideias eram inconcebíveis. Assim sendo, e deparados com vários obstáculos, foi necessário adequar e adaptar algumas ideias de forma a tornar o projeto o mais fiel e robusto face ao pedido. Ainda assim, de modo geral, o resultado vai de encontro ao esperado. Consoante o trabalho desenvolvido por cada elemento, consideramos sensata a distribuição percentual de 35% para o Henrique Silva e 65% para a Márcia Pires.