UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
SISTEMAS DE INFORMAÇÃO

SISTEMAS OPERACIONAIS
SISTEMA DE ALARME RESIDENCIAL

JOSÉ DE FARIA LEITE NETO
WILIAN HENRIQUE CAVASSIN
FERNANDO LEVY SILVESTRE DE ASSIS

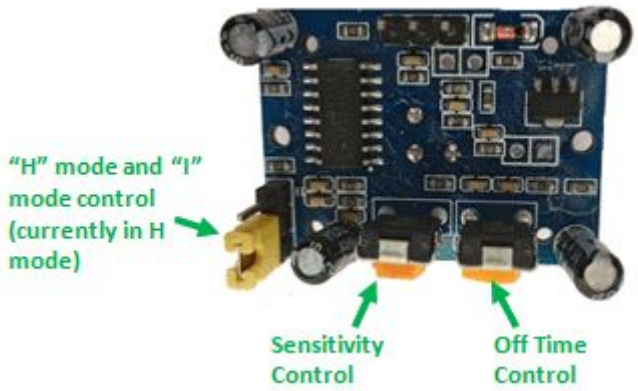
CURITIBA
2018

Sumário

1 - INTRODUÇÃO

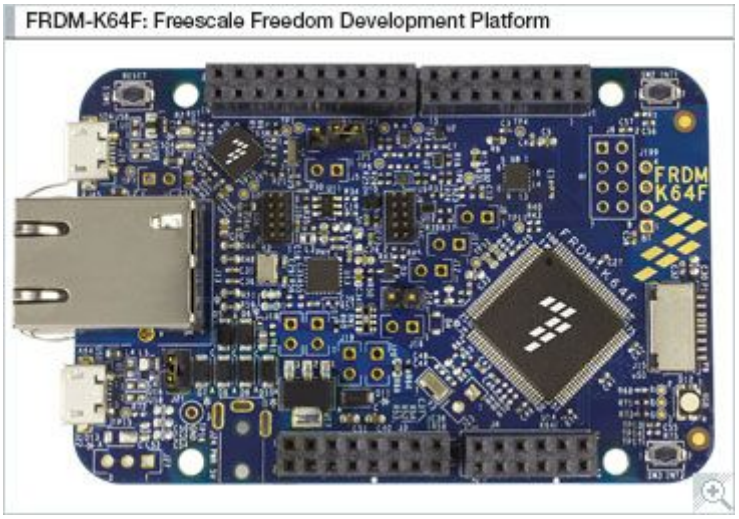
O objetivo do projeto é simular um sistema de alarme residencial com 5 alarmes, para o melhor uso dos conceitos aprendidos durante a disciplina, foram atribuídas prioridades diferentes nos alarmes, supondo que a residência possui regiões mais críticas que as demais, precisando de uma prioridade maior nos alarmes.
Na parte de hardware, para simulação do sistema, foram usados 4 push buttons e 1 sensor de movimento, cada um representando um sensor alarme da residência. Também foi usado um buzzer para simular a saída do sistema assim como os leds da própria placa, e tudo isso foi construído usando uma placa frdm-k64f, concedida pelo professor.

Modelo de push button usado no projeto



Modelo de sensor usado no projeto

Modelo de buzzer utilizado



Modelo de placa frdm-k64f, a qual foi usada no sistema

Falando de software, toda a implementação do projeto foi desenvolvida na plataforma mbed, usando a biblioteca padrão da plataforma “mbed.h” e biblioteca “rtos.h” que disponibiliza a implementação de threads.

No total, foram usadas 5 threads no projeto, sendo 4 para cada input (botões e sensor), e 1 thread para o escalonador. Os conceitos de SO usados foram: sistema não-preemptivo, com comunicação de threads por memória compartilhada, e um escalonador por prioridades, timer e interrupção. O sistema não-preemptivo foi usado porque o tempo de execução dos processos é curto, então seria humanamente impossível o invasor disparar um segundo alarme antes do primeiro terminar de tocar (seguindo nossa abstração), a comunicação por memória compartilhada foi a escolha por ser a maneira mais simples de implementar a fila de processos, considerando que o escalonador e os sensores utilizam da mesma região de memória, o escalonamento por prioridades foi usado porque, para melhor aproveitamento dos conceitos que aprendemos na disciplina, supomos que cada sensor varre uma região mais ou menos crítica que os outros, por exemplo, o sensor supostamente varre a região que dá acesso aos quartos, logo ele possui maior prioridade. As interrupções são disparadas quando algum dispositivo de input é acionado, e assim, suas rotinas são executadas. Foi também usado um timer próprio da placa para o tempo de execução dos processos.

2 - COMUNICAÇÃO ENTRE THREADS

Como citado no capítulo anterior, foi usada uma região de memória compartilhada para a comunicação entre as threads. Basicamente, as threads de input, acionadas pelos botões e pelo sensor, ficam aguardando o acionamento dos mesmos, quando isso acontece, uma interrupção é acionada dentro do sistema. A região de memória alocada para o sistema tem tamanho fixo, considerando que não é possível ativar um sensor duas vezes (seguindo nossa abstração de alarmes), e quando a interrupção é ativada, uma parte dessa memória é setada para um valor que corresponde ao input ativado, por exemplo, se o botão 3 é acionado, a posição 2 da memória recebe o identificador do botão e sua prioridade. Quando isso acontece, o escalonador (que está a todo momento varrendo a região de memória), percebe que há um processo na região, ele então verifica se é a maior prioridade, se sim, o processo é executado e a região de memória volta para seus valores originais.

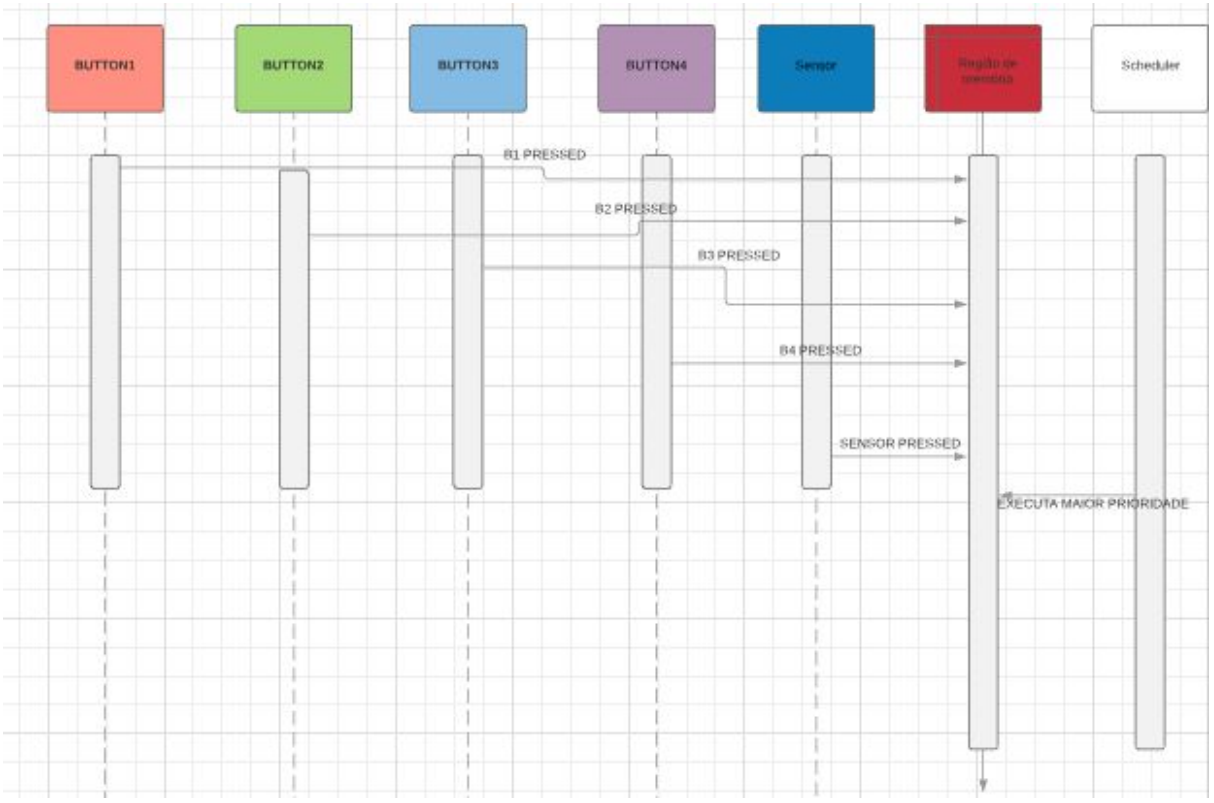


Diagrama de sequência final do projeto

3 - ALGORITMO DE ESCALONAMENTO

O escalonamento usado foi por prioridade, como citado anteriormente, foi suposto que cada sensor atuaria em uma região mais ou menos crítica em relação às outras, logo, essas regiões mais críticas possuem uma prioridade maior. O escalonador desempenha o papel de escolher o processo mais prioritário e executá-lo. Ele faz isso varrendo a região de memória constantemente, quando ele encontra um processo, ele armazena sua prioridade em uma variável e verifica se há outro processo com uma prioridade maior, caso sim, ele armazena essa nova prioridade maior, e assim ele encontra o processo com maior prioridade da região. Logo em seguida ele executa a rotina específica para esse processo, cada processo ativa o buzzer e acende uma cor específica do led da placa por 3 segundos. Após a execução, o escalonador seta a região de memória para sua prioridade e identificador original, assim como reseta a variável que armazena a maior prioridade.

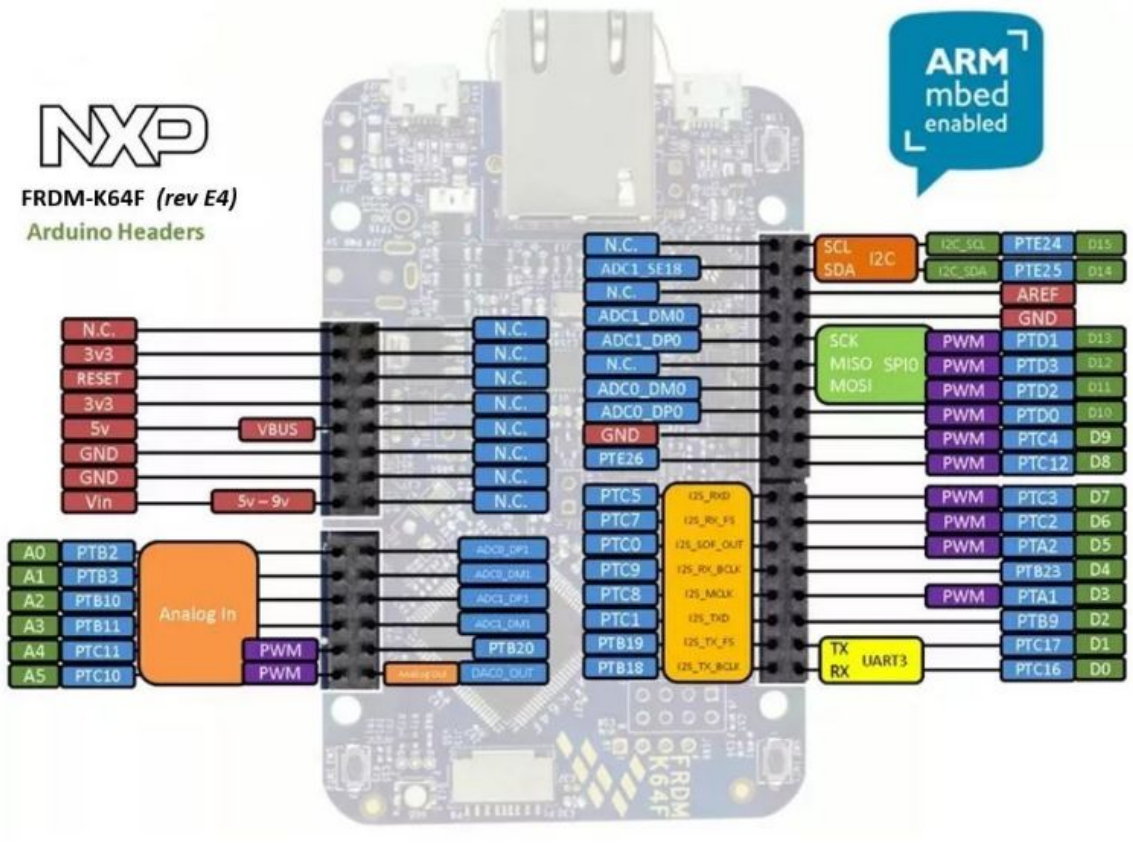
4 - DESCRIÇÃO GERAL DO TRABALHO

Os 5 sensores estão ligados em 5 pontos de interrupção da placa:

Botão 1: D7
Botão 2: D6
Botão 3: D5
Botão 4: D4
Sensor de movimento: D3

O sensor de movimento tem sua alimentação ligada à saída de 5V da placa.

As saídas do sistema são os leds RGB da própria placa, e um Buzzer conectado na saída D2. O terra do sistema está conectado na saída D0.



Esquemático das saídas da placa

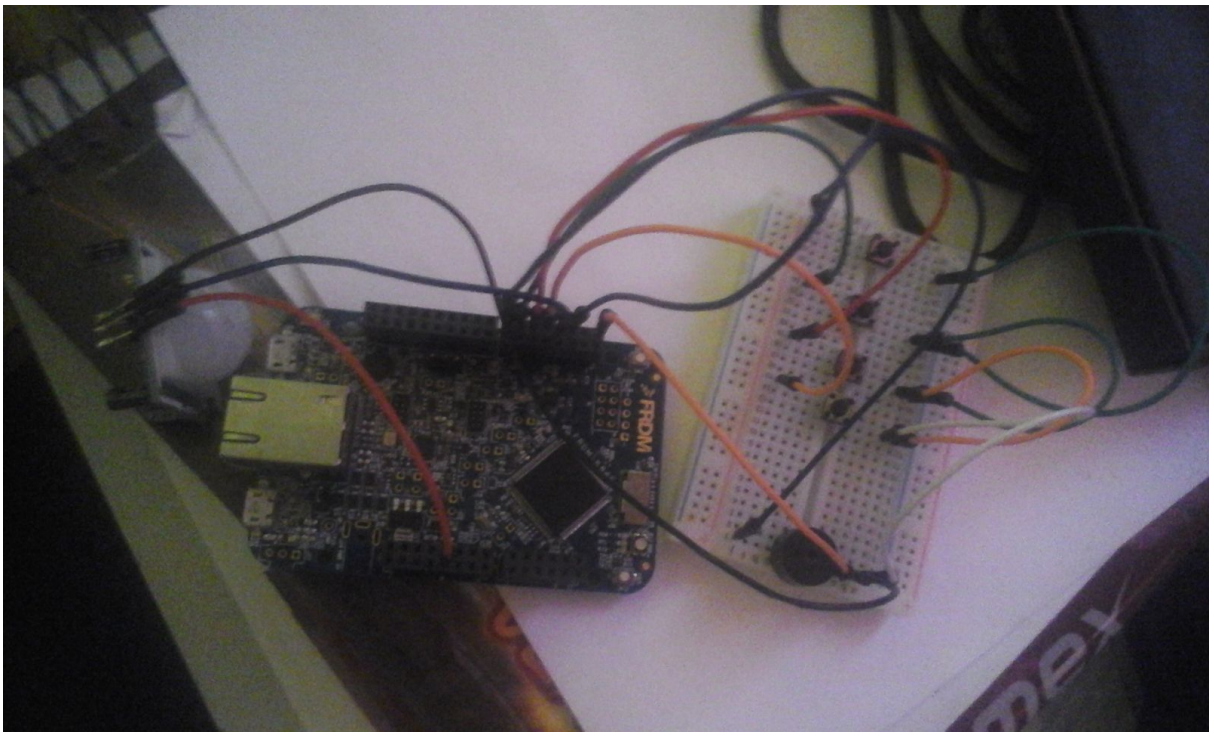


Foto da montagem final da placa

Globalmente, são alocadas as regiões de memória para os sensores, atuadores e para a região que será compartilhada entre os sensores e o escalonador.
Cada botão representa um sensor do alarme, assim como o sensor de movimento representa o 5º sensor. Logo no começo da execução, o sistema seta a prioridade de cada sensor:
Botão 1: prioridade 3
Botão 2: prioridade 2
Botão 3: prioridade 1
Botão 4: prioridade 3
Sensor de movimento: prioridade 4

Então, o valor da saída do terra é setado para zero, os leds começam apagados, e todas as seis threads são iniciadas (sensores e escalonador). Em seguida, o sistema fica aguardando a ativação de algum sensor, e quando isso acontece, é acionada uma interrupção no sistema, uma rotina de tratamento acontece e o escalonador seleciona o processo de acordo com a política de prioridades, como já foi descrito no capítulo anterior. Após isso, o alarme é disparada, e o(s) led(s) e o buzzer executam por 3 segundos, então, o próximo processo da fila é executado (se houver). Abaixo está descrito a rotina para cada sensor:

Botão 1: Led vermelho + buzzer
Botão 2: Led azul + buzzer

Botão 3: Led verde + buzzer
Botão 4: Led verde + Led azul + buzzer
Sensor de movimento: Led vermelho + Led azul + buzzer

5 - CONCLUSÃO

O trabalho foi de vital importância para o exercício de diversos conceitos adquiridos durante a disciplina. Foi uma experiência nova e única para nós que somos alunos de Sistemas de Informação, assim como importante para aprendermos como os softwares gerenciam recursos. A parte de hardware trouxe algumas dificuldades na hora de montagem e funcionamento dos componentes, mas foram todas contornadas.
No fim, o sistema cumpre o que foi proposto de ser uma abstração de um sistema de alarme residencial, e foi vital saber todos os conceitos da disciplina e saber usá-los da maneira certa para cumprir nossos objetivos.