

IBMEC RJ – ARQUITETURA DE COMPUTADORES
PROF CLAYTON J A SILVA
TRABALHO PROPOSTO AP1-2024.1

1. Visão Geral

O projeto aplica conceitos teóricos de dispositivos de entrada/saída, sensores, atuadores e representação de dados, enfatizando a prática de programação através de modularização, uso de funções, e estruturas de controle.

Desenvolver um sistema de alarme digital programável com o Arduino Mega 2560.

A ativação do alarme é definida por chave seletora (*on-off*), sinalizada por LEDs. A definição de hora de início é definida no formato binário por um conjunto de chaves (*dip-switches*). A definição de hora de término é também definida por chaves (*dip-switches*).

O Arduino Atmega2560 é o sistema computacional que: lê a chave seletora e as chaves de hora de início-término; lê a hora atual – pelo monitor serial; mantém o relógio atualizado; lê o sinal detectado; e gera o alarme no caso de invasão.

Uma característica distintiva do projeto é a utilização de uma saída de 5 bits, representada por LEDs, para mostrar o tempo restante (em horas) para o desligamento do alarme, utilizando a representação em complemento de 1.

A figura do anexo apresenta a visão geral do sistema.

2. Material Utilizado

- **Arduino Mega 2560:** A base para a implementação do sistema de alarme.
- **LEDs:** Para indicar o status do sistema e o tempo restante para o desligamento do alarme em complemento de 1.
- **Buzzer:** Para funcionar como o alarme sonoro.
- **Chaves dip switch:** Para configurar os horários de ativação e desativação do alarme em formato 24h binário puro.

3. Descrição do Shield

O *shield* configurado para este projeto incluirá:

- **LEDs de Status:** Um verde para indicar o sistema ativo e um vermelho para o sistema inativo. LEDs verde e vermelho do Seletor e Indicador de Sistema Ativo da figura anexa.
- **Chaves *dip switch*:** Para a entrada dos horários de início e término de ativação do alarme. Chaves I₀ a I₄, T₀ a T₄ da figura anexa.
- **Buzzer:** Para alertas sonoros quando o sistema detectar uma presença no período ativo.
- **LEDs de Contagem Regressiva:** Um grupo de LEDs para representar, em complemento de 1, o tempo restante (em horas) para o desligamento do alarme. LEDs S₀ a S₄ da figura anexa.
- **Sensor:** O sensor de presença será simulado por uma chave, sinalizando a detecção da presença ou não de um invasor em um perímetro.
- **Observação:** verificar que a figura apresenta a sugestão de pinagem utilizada para entradas e saídas do Arduino.

4. Funcionalidades Desejadas do Sistema

- **Ativação/Desativação do sistema:** Mediante acionamento da chave seletora, o usuário pode a qualquer tempo armar ou desarmar o alarme. LEDs específicos mostram se o sistema está ativo ou inativo.
- **Configuração de Horários de Início/Término de Alarme:** Através das chaves *dip switch*, o usuário poderá definir as horas limites (em binário puro) entre as quais, se ocorrer uma invasão, o alarme deverá soar. Se o usuário inserir uma hora acima de 24h e a chave de seletora estiver em *on*, os LEDs de saída devem piscar todos simultaneamente.
- **Configuração de Hora Atual e Relógio:** Através do monitor serial o usuário deverá definir a "hora atual" e o sistema deverá manter o "relógio" atualizado a partir da hora inserida.
- **Determinação de Hora Restante:** O sistema deve determinar e manter atualizado quantas horas faltam para a hora de término programada.
- **Contagem Regressiva em Complemento de 1:** A quantidade de horas restantes para o desligamento do alarme é mostrada nos LEDs de contagem regressiva em complemento de 1. Quando o alarme estiver desativado, todos esses LEDs devem ficar apagados
- **Acionamento e Desligamento Automático do Alarme:** O alarme é acionado durante o período ativo ao detectar presença e desliga automaticamente conforme programado. A detecção de presença deve ser simulada por uma entrada digital do Arduino.

5. Testes

O sistema deve ser modularizado tanto na arquitetura do hardware quanto na estruturação do *sketch*.

Os seguintes testes de hardware serão realizados para assegurar o funcionamento adequado do sistema

- **Ativação/Desativação do sistema:** Quando a chave for acionada para ligar o alarme, o LED verde deve acender; em caso contrário, o LED vermelho deve acender. Verificar se a entrada digital do Arduino está lendo corretamente o sinal de ativação.
- **Configuração de Horários de Início/Término de Alarme:** Verificar se as entradas digitais correspondentes às chaves estão sendo lidas corretamente.
- **Contagem Regressiva em Complemento de 1:** Verificar se as saídas do Arduino que se conectam com os LEDs que indicam a quantidade de horas restantes para o desligamento do alarme estão adequadas. Verificar se os LEDs estão acendendo e apagando conforme as saídas 1 ou 0.
- **Acionamento do Alarme:** Verificar se a saída digital do Arduino está gerando o sinal para o *buzzer* de alarme. Verificar se o *buzzer* está tocando com o tempo e com a frequência desejados.
- **Deteção de intrusão:** Verificar se o simulador de detecção está sendo lido corretamente pelo Arduino.
- **Configuração de Horários e Resposta do Sistema:** Garantir que o sistema aceita a configuração de horários corretamente e responde de acordo com essas configurações.

Os seguintes testes de software serão realizados para assegurar o funcionamento adequado do sistema:

- **Configuração de Horários de Início/Término de Alarme:** Conversão binária-decimal no formato HH.
- **Configuração de Hora Atual:** Leitura do monitor serial para o usuário deverá definir a "hora atual".
- **Relógio:** Funcionamento do "relógio" no formato HH:MM a partir da hora inserida.
- **Contagem Regressiva:** Funcionamento do timer, no formato HH, indicando quantas horas estão restantes a partir da hora inserida. Conversão HH decimal-binária. Geração do complemento de 1.

- **Verificação da Hora Atual está no Intervalo de Alarme:** Verificar se o alarme está ativado ou não e verificar se o relógio está no intervalo de alarme.
- **Teste de Overflow de Hora de Início Hora de Término:** Verificar se os LEDs piscam quando uma hora inadequada é inserida.
- **Testes de leitura e saída de dados.**

Sugestão de funções para estruturação do sketch:

- *setup()*: Configuração inicial do Arduino, definindo os pinos dos LEDs, buzzer, chaves dip switch e configurações iniciais do monitor serial.
- *loop()*: Função principal que verifica o estado do sistema (ativado/desativado), lê a hora atual via monitor serial, e executa as rotinas de verificação e acionamento do alarme.
- *readDipSwitches()*: Lê as chaves dip switch e converte o valor binário para decimal, para as horas de início e término do alarme.
- *updateCurrentTime()*: Lê a hora atual inserida pelo usuário via monitor serial e atualiza o relógio interno.
- *activateAlarm()*: Verifica se a hora atual está dentro do intervalo definido para o alarme e ativa o sistema, acendendo o LED verde e aguardando detecção de presença.
- *deactivateAlarm()*: Desativa o sistema de alarme, acendendo o LED vermelho e parando o buzzer.
- *checkIntrusion()*: Verifica a entrada simulada de presença e, se dentro do intervalo de horário ativo, aciona o buzzer.
- *calculateRemainingTime()*: Calcula quantas horas faltam até o término do alarme e atualiza os LEDs de contagem regressiva em complemento de 1.
- *updateCountdownLEDs(int hours)*: Atualiza os LEDs de contagem regressiva com o valor em complemento de 1 das horas restantes.
- *invertBits(int value)*: Recebe um valor e retorna o seu complemento de 1 para a representação nos LEDs.
- *checkTimeOverflow()*: Testa se o usuário não inseriu uma hora fora do range de 0 a 24h.

6. Apresentação da Solução

A solução final deverá incluir:

- **Descrição Detalhada do Shield:** Explicação sobre a montagem e configuração dos componentes no *shield*.
- **Código Fonte (Sketch) Comentado:** O *sketch* (código desenvolvido para o Arduino), detalhando a implementação das funcionalidades, especialmente a lógica para o cálculo do tempo restante em complemento de 1.

- **Demonstração de Funcionalidade:** Mostrar o sistema em operação, destacando a capacidade de configurar horários, a acurácia dos LEDs indicadores de status e tempo restante, e a funcionalidade do sistema de alarme conforme o esperado.

