

**IBMEC RJ – ARQUITETURA DE COMPUTADORES**  
**PROF CLAYTON J A SILVA**  
**TRABALHO PROPOSTO AP2-2024.1**

## **1. Visão Geral**

### **Parte 1. Contemplada na AP1**

Desenvolver um sistema de alarme digital programável com o Arduino Mega 2560. A ativação do alarme é definida por chaves seletoras (*on-off*), sinalizadas por LEDs. A definição de hora de início é definida no formato binário por um conjunto de chaves. A definição de hora de término é também definida por chaves. 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. 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.

### **Parte 2. A ser apresentada na AP2**

Utilizando um sensor de movimento ou sensor de presença, substituir a chave que simula o invasor da fase 1 do projeto, ajustando tanto o

As referências para uso de ambos podem ser encontradas nas práticas:

[https://github.com/claytonjasilva/claytonjasilva.github.io/blob/main/arq\\_aulas/pratica\\_sensorultrassonico.md](https://github.com/claytonjasilva/claytonjasilva.github.io/blob/main/arq_aulas/pratica_sensorultrassonico.md)

[https://github.com/claytonjasilva/claytonjasilva.github.io/blob/main/arq\\_aulas/pratica\\_sensormovimento.md](https://github.com/claytonjasilva/claytonjasilva.github.io/blob/main/arq_aulas/pratica_sensormovimento.md)

**DATA DE APRESENTAÇÃO: NO DIA DA APLICAÇÃO DA AP2, LOGO APÓS A PROVA.**

**TODOS OS COMPONENTES DO GRUPO DEVERÃO ESTAR PRESENTES. O COMPONENTE DO GRUPO AUSENTE TERÁ GRAU 0,0 NO TRABALHO.**

## 2. Material Utilizado

Além de todo o material utilizado na fase 1,

- **Sensor ultrassônico** HCSR04, para detectar a presença de invasor quando se aproximar a uma distância do sensor; **ou**
- **Sensor de presença** PIR

## 3. Descrição do Shield

O shield configurado para este projeto incluirá, conforme detalhado na figura anexa:

- **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 I<sub>0</sub> a I<sub>4</sub>, T<sub>0</sub> a T<sub>4</sub> 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 S<sub>0</sub> a S<sub>4</sub> da figura.

ATÉ AQUI - CONFORME JÁ IMPLEMENTADO NA FASE 1 DO PROJETO, incluindo

- **Sensor:** Utilizar um sensor ultrassônico ou PIR, sinalizar a detecção da presença ou não de um invasor em um perímetro, para substituir a chave com jumper utilizada anteriormente.

## 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.

Como definido na fase 1 do projeto, complementadas com

- **Sensoriamento de invasor:** O sistema precisa detectar o invasor se aproximando do limite do perímetro de segurança do sistema. Dependendo do tipo de sensor escolhido essa funcionalidade será implementada de forma diferente.

## 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.
- **Configuração de Horários e Resposta do Sistema:** Garantir que o sistema aceita a configuração de horários corretamente e responde de acordo com essas configurações.

Como realizado na fase 1, incluindo

- **Deteção de intrusão:** Verificar se o simulador de detecção está sendo lido corretamente pelo Arduino.

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.**
- Como os testes anteriores da fase 1, incluindo na fase 2
- **Teste de invasão:** Verificar se a detecção do invasor está acontecendo, de acordo com o tipo de sensor utilizado.

## 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.

