

## Documentação Mini Projeto Arduino – Dupla Livia e Marcelo

Nós adotamos os seguintes padrões de LEDs para representar nossos estados:  
Considerando os ícones:

**Estado 1 – CLOCK** (Exibe as horas sem acionar o alarme)

D1 

D2 

D3 

D4 

**Estado 2 – ALARM\_ON** (Exibe as horas com alarme ligado)

D1 

D2 

D3 

D4 

**Estado 3 – ALARM** (Exibe o horário do alarme, com ele ligado)

D1 

D2 

D3 

D4 

**Estado 4 – SET\_CLOCK\_H** (Exibe as horas e permite ajuste das horas do relógio)

D1 

D2 

D3 

D4 

(LEDS Piscando)

**Estado 5 – SET\_CLOCK\_M** (Exibe as horas e permite ajuste dos minutos)

D1 

D2 

D3 

D4 

(LEDS Piscando)

**Estado 6 – SET\_ALARM\_H** (Exibe horário do alarme e permite ajuste de suas horas)

D1 

D2 

D3 

D4 

(LEDS Piscando)

**Estado 7 – SET\_ALARM\_M** (Exibe horário do alarme e permite ajuste de seus minutos)

D1 ●

D2 ○

D3 ●

D4 ○

(LEDS Piscando)

O projeto possui apenas dois timers, um para a campainha do despertador e outro que marca 1 segundo. Para facilitar nossos testes, fizemos o relógio contando só até 24min, indicando os minutos do primeiro display e os segundos no outro. Depois adicionamos um contador dentro deste timer, para contar 60segundos e exibir as 24 horas como um relógio real.

O alarme apita durante 1minuto a emitindo um som à uma frequência (próximo a frequência de uma nota musical ré), seguido de mais apitos a uma frequência de 342Hz (próximo à uma nota musical fá).

No módulo "feature\_alarm\_with\_music.ino" são tocadas variações de várias notas tentando reproduzir uma música conhecida, porém vimos que a campainha não reagia bem a frequências acima de 349Hz. E tivemos alguns problemas com a função "tone()" do Arduino, ao varias muitas frequências e utiliza-la junto com outros recursos como o display.

O nosso tratamento de debounce, nos permite segurar um botão por tempo indeterminado, e só será contado uma única como apertado, somente ao soltar o botão. Ele foi feito captando o último momento de descida do botão (saindo do estado HIGH para o LOW) e depois esperamos uns 50ms para verificar se o botão ainda está em LOW ou se está em HIGH (caso tenha sido solto), a partir desse momento consideramos que o botão foi apertado.

Para utilizar a função de pressionar o botão 3+1 para voltar ao primeiro estado, deve-se pressionar primeiro o botão 3 logo em seguida o botão 1, pois ao pressionar o botão 3 por último, a sua função de ir para o próximo estado ocorrerá antes.

Todos os outros requisitos do projeto, descritos no enunciado, foram satisfeitos como esperado.