

# EA871 – LAB. DE PROGRAMAÇÃO BÁSICA DE SISTEMAS DIGITAIS

## Projeto Final – Relógio Digital com Cronômetro, Termômetro e comunicação com um Terminal

Profa. Wu Shin-Ting

**OBJETIVO:** Desenvolvimento de um projeto de média complexidade com uso de máquina de estados.

**ASSUNTOS:** máquina de estados, concorrência de recursos, integração de funções existentes.

### O que você deve ser capaz ao final deste experimento?

Modelar a solução de um problema como uma máquina de estados.

Reusar as funções existentes.

Contornar o uso concorrente de recursos.

Planejar os valores a serem configurados nos registradores de um microcontrolador para que o comportamento do sistema atenda a especificação do projeto.

## INTRODUÇÃO

Uma máquina de estados é um modelo de computação baseado em um número finito de estados e um número finito de transições entre estes estados. Ao representar o comportamento esperado de uma máquina como uma máquina de estados pode facilitar a otimização dos códigos como a identificação de transições válidas, como é demonstrado neste projeto final.

O projeto consiste num relógio digital com 3 funcionalidades: relógio, cronômetro e termômetro. Pode-se **fazer ajustes com base no horário atual** e **disparar cronometragens** via 3 chaves. Uma chave, IRQ12, é responsável por selecionar a unidade de horário (hora, minuto ou segundo) a ser ajustada. A segunda chave **IRQ5 serve para incrementar o valor** da unidade selecionada, enquanto a chave NMI, para decrementar o valor da unidade selecionada. A chave **IRQ5 é, porém, também usada para controlar o início e o fim de uma cronometragem**. Vamos ver que, ao descrever o comportamento de uma máquina em estados, fica simples o controle da função que a chave IRQ5 deve assumir num dado instante.

É integrada ao relógio digital uma **interface serial com um Terminal**, de forma que seja possível fazer consultas do horário e da temperatura corrente, como também do último valor cronometrado. Neste caso, podem ocorrer acessos concorrentes do horário e do valor cronometrado pelo Terminal (para leitura) e pelas chaves (para atualização), de forma que os valores lidos possam ser inválidos. Vamos ver que uma das formas mais simples para evitar essas concorrências é **evitar operações paralelas** sobre um mesmo recurso, desabilitando um dos módulos.

O esboço de um diagrama de blocos contendo os módulos controladores das funções do relógio digital é apresentado na Fig. 1. Essencialmente, o dispositivo tem três modos de operação:

**Operação normal:** são mostrados no visor do LCD o horário do relógio (no formato HH:MM:SS a partir do endereço 0x00), o *bitmap* do relógio no endereço 0x0F e a temperatura do sensor (no formato XX.X C a partir do endereço 0x40). Os valores da temperatura são divididos em 6 faixas: menor que 22, [22,23], [23,24],[24,25],[25,26], e maior que 26. Cada faixa é codificada por uma cor, mostrada pelo *led* RGB. Neste modo de operação é permitida a consulta via Terminal do horário (ao

digitar H/h), da temperatura (ao digitar T/t) e da última cronometragem (ao digitar C/c). As mensagens de resposta são enviadas ao Terminal via um *buffer circular*.

**Operação de ajuste de horário:** são mostrados no visor do LCD o horário do relógio (no formato HH:MM:SS a partir do endereço 0x00), o *bitmap* do relógio no endereço 0x0F e o cursor piscando na unidade do horário que está sendo configurado. Toda vez que se aciona a chave IRQ12, entra-se no modo de ajuste. O horário mostrado é o horário corrente do relógio. A chave IRQ12 seleciona ciclicamente as unidades HH → MM → SS → HH. Para cada unidade, o valor é incrementado (em 1) a cada acionamento da chave IRQ5 e decrementado (em 1) a cada acionamento da chave NMI. A estratégia para voltar ao modo de operação normal é por *timeout* de 2s controlado pelo temporizador SysTick.

**Operação cronometragem:** são mostradas no visor do LCD as contagens em segundos a partir do endereço 0x00, o *bitmap* do cronômetro no endereço 0x0F e o resultado da cronometragem no formato HH:MM:SS a partir do endereço 0x40. As temperaturas amostradas são apenas visualizadas através das cores do *led* RGB. Inicia-se a cronometragem quando se aciona a chave IRQ5 e pára-se a cronometragem quando a mesma chave é reacionada. O resultado da cronometragem é mostrado em seguida no visor LCD e volta-se automaticamente para o modo de operação normal.

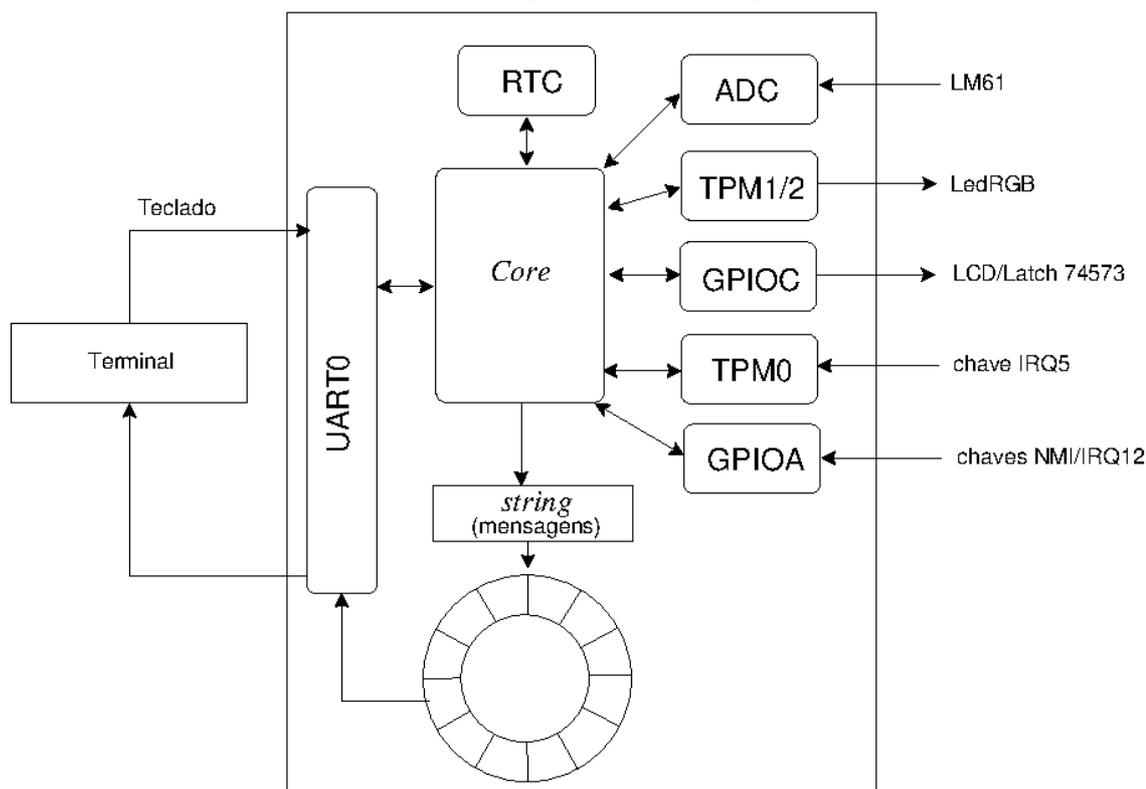


Fig. 1: Diagrama de blocos do Relógio Digital.

Para a operação deste dispositivo, foram classificados 13 estados:

- RELOGIO (0): operação normal.
- AJUSTE\_HORA (1): cursor piscante sobre a unidade de hora.
- AJUSTE\_HORA\_INC (2): incremento da hora
- AJUSTE\_HORA\_DEC (3): decremento da hora.
- AJUSTE\_MINUTO: (4) cursor piscante sobre a unidade de minuto.
- AJUSTE\_MINUTO\_INC (5): incremento de minuto.

- AJUSTE\_MINUTO\_DEC (6): decremento de minuto.
- AJUSTE\_SEGUNDO: (7) cursor piscante sobre a unidade de segundo.
- AJUSTE\_SEGUNDO\_INC (8): incremento de segundo.
- AJUSTE\_SEGUNDO\_DEC (9): decremento de segundo.
- CRONOMETRO\_INI (10): início de cronometragem.
- CRONOMETRO\_FIM (11): fim de cronometragem.
- UART0\_IN (12): consulta via Terminal.

Fig. 2 mostra uma proposta de transições entre os 13 estados. Por uma questão de legibilidade, foram omitidas as transições dos estados 1 a 9 para o estado 0 por um *timeout* em 2s, ou seja se as chaves ficarem ociosas por mais que 2s, voltar-se-á automaticamente para a operação normal (estado 0). Uma implementação desta máquina em C se encontra no projeto [maquina\\_de\\_estados.zip](#) [1].

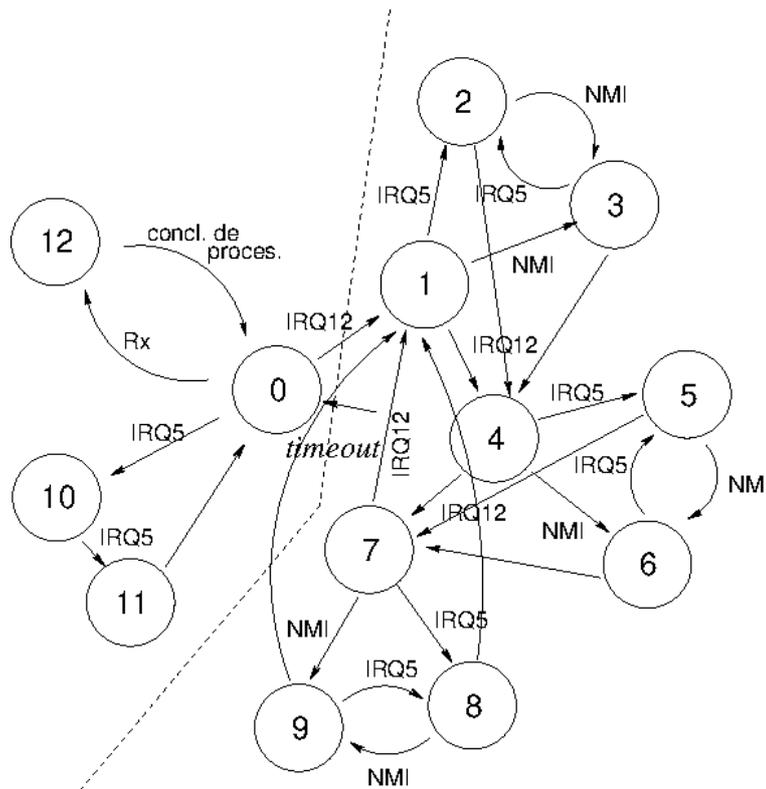


Fig. 2: Transições entre os estados, dependentes das chaves acionadas e dos estados.

Como já comentado, os registradores onde estão armazenados os dados de relógio, termômetro e cronometragem podem ser acessados simultaneamente pelo Terminal e pelas chaves. Para evitar situações como ler um valor cujo processamento ainda não é concluído, foi aplicada uma solução simples que consiste em desabilitar o canal Rx do Terminal quando o dispositivo estiver no modo de ajuste e no modo de cronômetro.

Adicionais detalhes técnicos que devem ser atendidos na implementação do projeto:

1. A frequência de *bus clock* seja 10485760.
2. A fonte de clock do módulo RTC em 1 kHz é gerado pelo temporizador LP0.
3. O *timeout* de 2s é controlado pelo SysTick.
4. Os *leds* vermelhos do *shield* FEEC 871 devem ficar apagados.

5. O modo de operação dos módulos TPM1 e TPM2 é: contagem crescente; contagem máxima num período, 255; pré-escala, 8; e polaridade *Low*.
6. O modo de operação do módulo ADC é disparo por *hardware*, na frequência 2Hz, usando o módulo PIT  
frequência ADCK: 2621440Hz  
resolução de 10 *bits*  
tempo de amostragem longo habilitado, com 6 ciclos de ADCK extras.  
alta velocidade de conversão habilitada  
média habilitada para 8 amostras por conversão
7. O modo de operação do módulo UART0 é *baud rate 19200*, caracter de 8 *bits*, 1 *stop bit*, *bit* de paridade par e taxa de amostragem 12x.

## RELATÓRIO

Relatório técnico do projeto final, incluindo

1. uma descrição funcional do projeto, tomando como base a máquina de 13 estados,
2. um diagrama de blocos mostrando as relações entre os módulos do Kinetis usados no projeto,
3. detalhamento da configuração dos registradores de controle destes módulos, especialmente dos módulos RTC (relógio), PIT + ADC (amostragem dos sinais do sensor LM61), TPMx (TPM0 para controle de cronômetro e TPM1/2 para controle das cores do *led* RGB), PORTx + GPIOx (PORTA para controle das chaves e PORTC para controle de LCD/latch 74573) e UART0 (comunicação serial com o Terminal), justificando a escolha dos valores com base nas especificações técnicas.
4. pseudocódigos (do fluxo principal e do tratamento do evento que ocasionou transições) de cada um dos 13 estados definidos.
5. uma descrição do projeto da interface com usuário.
6. testes de validação realizados.
7. códigos-fonte documentados do projeto.

Elabore um texto endereçado aos itens 1–6 e organize-o num arquivo em **pdf**. Exporte o projeto no ambiente IDE CodeWarrior para um arquivo em formato zip. Suba **os dois arquivos, em separado**, no sistema [Moodle](#). Não se esqueça de identificar todos os seus arquivos de códigos com a palavra reservada “@author” de Doxygen.

## REFERÊNCIAS

[1] [maquina\\_de\\_estados.zip](#)

[http://www.dca.fee.unicamp.br/cursos/EA871/2s2020/ST/codes/maquina\\_de\\_estados.zip](http://www.dca.fee.unicamp.br/cursos/EA871/2s2020/ST/codes/maquina_de_estados.zip)

[2] Roteiros dos experimentos 1-13