



ENGENHARIA ELETRÔNICA

EEN241 – MICROCONTROLADORES E SISTEMAS EMBARCADOS

NOTURNO

PROF. RAFAEL CORSI

HENRIQUE PEREIRA ROSA

11.02741-0

EDUARDO GALINSKAS KARWOSKI

13.01129-4

17/MARÇO/2016

1.1 Otimizações

- 00**: Não realiza otimizações no código. Apenas traduz para linguagem de baixo nível.
- 0** / -**01**: Realiza tarefas simples de otimização. É o primeiro nível de otimização.
- 02**: Um nível mais avançado de otimizações comparado ao nível 1. Ativa todas as otimizações opcionais exceto o desatamento de loop, fazer inline de função (otimizar tamanho), e a renomeação e registrador. É o mais utilizado no Linux.
- 03**: O nível máximo de otimizações em relação a tamanho e códigos de e velocidade de execução do código.

1.2 volatile/ const / static

Volatile: Avisa ao compilador para não modificar as variáveis ou funções utilizadas nessa declaração, durante as otimizações de compilação.

Const: O modificador “const” faz com que a variável não possa ser modificada no programa. Útil para declarar constantes. Faz com que o compilador

Static: Variáveis “static” funcionam como variáveis globais dentro de um módulo

1.3 MakeFile

O Makefile é um arquivo de configurações que norteia o funcionamento do Make. Um exemplo são as opções de otimização que são interpretadas por ele. O texto contido em um Makefile é usado para a compilação, ligação (linking), montagem de arquivos de projeto entre outras tarefas como limpeza de arquivos temporários, execução de comandos, etc.

Vantagens do uso do Makefile:

- Evita a compilação de arquivos desnecessários. Por exemplo, se seu programa utiliza 120 bibliotecas e você altera apenas uma, o make descobre (comparando as datas de alteração dos arquivos fontes com as dos arquivos anteriormente compilados) qual arquivo foi alterado e compila apenas a biblioteca necessária.
- Automatiza tarefas rotineiras como limpeza de vários arquivos criados temporariamente na compilação.
- Pode ser usado como linguagem geral de script embora seja mais usado para compilação.

1.4 ASCII

O padrão ASCII (American Standard Code for Information Interchange) é uma padronização entre conjuntos binários e caracteres de forma a deixar comum os comandos e informações enviadas entre sistemas de informação.

Desenvolvida a partir de 1960, seu principal objetivo é padronizar o universo computacional, porém até hoje possui utilizações tanto computadores como em sistemas embarcados, como em Display Alfanuméricos que são controlados através dos comandos ASCII.

O código ASCII é muito utilizado para conversão de Código Binário para Letras do alfabeto Maiúsculas ou minúsculas.

2.1 Gravador/Debug

JTAG é um sistema de gravação e depuração de microcontroladores criado como objetivo de ser o dispositivo padrão para essa finalidade. Hoje é adotado por muitos fabricantes, como Atmel e Motorola. Já microchip por exemplo ainda está no processo de migração.

Os pinos utilizados são:

Pino 1: É usado para checar se o “alvo” está alimentado, para criar o nível lógico para a entrada do comparador para controlar o nível lógico da saída para o alvo;

Pino 3: JTAG Reset;

Pino 5: Entrada de dados;

Pino 7: Modo de ajuste da entrada;

Pino 9: Sinal de clock do gravador;

Pino 11: Entrada de retorno do sinal de clock vindo do device;

Pino 13: Saída de dados vindos do device.

Daisy Chain é um esquema de fiação em que vários dispositivos são conectados juntos em sequência ou em um anel. Esse sistema é utilizado para gravar vários dispositivos utilizando apenas um conector.

2.2 Jumpers

JP3: Reinicializa o conteúdo Flash e alguns dos seus bits NVM.

JP9: Habilita o NAND FLASH Chip Select

JP5: Acesso para medição de corrente em VDDPLL.

2.3 Clock

O gerador de clock é composto de:

- Um oscilador de baixa potência de 32.768HZ com modo de desvio.
- Um oscilador de cristal de 3 à 20 MHz que pode ajustado para 12 MHz (necessário em caso de USB).
- Um oscilador RC interno programado de fábrica.
- Um controlador USB que provém um clock de 60 até 130MHz.
- Um controlador PLL programável de 60 à 130MHz (PLLA) capaz de prover um clock MCK para o processador e para os periféricos. A frequência de entrada do PLLA varia de 7.5 até 20MHz.

A placa do SAM4S-EK2 é equipada com um cristal de 12MHz, um de 32.768Hz e um conector de entrada de clock externo.

2.4 Alimentação

O SAM4SD32 possui diferente tipos de pinos de alimentação:

VDDIN: Alimenta o regulador interno de tensão, ADC, DAC e o comparador analógico. A tensão varia de 1.8V a 3.6V.

VDDIO: Alimenta os periféricos I/O. Tensão de 1.62V a 3.6V.

VDDCORE: Alimentação para o núcleo, incluindo o processador, memórias embutidas e periféricos. Tensão de 1.62V a 1.95V.

VDDPLL: Alimenta o PLL A, PLL B e o oscilador de 12MHz. A Tensão vai de 1.62V até 1.95V.

2.5 LEDS

Existem três LEDs na placa SAM4S-EK2: um LED azul e um verde que são usados e definidos pelo GPIO, e um LED vermelho que ligado a trilha de 3.3V, porém, também é controlado pelo GPIO e pode ser tratado como um led qualquer pelo usuário. A diferença do led vermelho é que ele é controlado através de um transistor MOS e por padrão, a linha do PIO é desabilitada e um resistor de pull-up controla o MOS para ligar o led vermelho quando a força está ligada. Os pinos são: PA12, PA20 e PC20.

2.6 Botões

Existem dois botões mecânicos no SAM4S-EK2 cujos pinos são conectados as linhas do PIO e definidos para serem esquerdo e direito por padrão. Existe também um botão para o controle de reset do sistema. Os pinos são: PB3, PC12 e NRST.

2.7 Periféricos

Entrada de Microfone: O microfone incorporado é conectado ao pré-amplificador de áudio usando o amplificador operacional TS922. O ganho é ajustado usando os jumpers JP14 e JP15; ambos devem ser ajustados ou removidos ao mesmo tempo. Modificando as posições dos jumpers, você pode selecionar cada seguinte valor de ganho:

20 dB (configuração padrão, ambos JP14 e JP15 estão desligados).

26 dB (ambos JP14 e JP15 estão ligados).

3.1 Memória

FLASH: 2 x 1024Kbytes.

SRAM: 160Kbytes.

ROM: 16Kbytes

3.2 IOH, IOL

A corrente máxima suportada em ambos é de 30 mA.

3.3 BrownOut

Uma condição na qual o nível de eletricidade fica muito reduzido por um tempo considerável. Ao contrário de um blackout, ou perda total de energia, em um brownout o fluxo de eletricidade de todos os dispositivos conectados à rede elétrica é mantido, porém, em níveis mais baixos que os normais. Por esse motivo existe uma opção de reinicialização do microcontrolador caso haja detecção desse nível

3.4 WatchDogTimer

O Watchdog em um sistema embarcado tem por função proteger o sistema com respeito a qualquer travamento ocasional que venha a ocorrer. Esse travamento pode ser causado pela ocorrência de uma condição inesperada no programa, por um bug ainda não encontrado no sistema ou até mesmo por falhas de hardware. Quando ocorre um travamento, o Watchdog provoca uma reinicialização no sistema, ajudando o mesmo a voltar a funcionar.

3.5 PIO

PIO (Parallel input Output) é um registrador responsável pelo gerenciamento dos pinos de interface do microcontrolador. Portanto é possível configurar os I/Os para interfacear com o mundo externo tendo acesso a sensores, botões, LEDs, entre outros.

3.5 CUSTO

Um microcontrolador ARM cortex M4 custa em média U\$10.