

Pesquisa 3

C para embarcados e kit de desenvolvimento

1 C

1.1 Otimizações

Os níveis de otimização que o GCC suporta são:

O0: nível padrão, no qual não há qualquer tentativa de otimização;

O1: o gcc tenta reduzir o tempo de compilação e o tamanho do executável, possibilitando ainda o uso de depuradores;

O2: melhor do que o nível O1, não causa aumento do arquivo por causa disso. Também é o melhor, mais seguro (por conta da portabilidade) e o mais usado nos dias atuais na distribuição de softwares em Linux;

O3: opção com melhor nível de otimização; entretanto, é a que mais causa efeitos colaterais: arquivos maiores, maior uso de memória RAM e poucas chances de fazer uso de um depurador sobre o programa executável;

1.2 volatile/const/static

Static é um programa que permite ao compilador identificar quando a variável sofre ou não modificação, dando liberdade para ele escolher se irá alocar na RAM ou na ROM. *Volatile* diz ao compilador para não otimizar determinada variável e sempre carrega-la da RAM, o que é bem útil quando uma variável é alterada pelo hardware. *Const* é utilizado quando não se pretende mudar o valor de certa variável e como os Microcontroladores costumam ter mais ROM do que RAM é possível poupar memória RAM usando *const*.

1.3 MakeFile

O objetivo do Makefile é definir regras de compilação para projetos de software. Tais regras são definidas em arquivo chamado Makefile. O programa make interpreta o conteúdo do Makefile e executa as regras lá

definidas. Alguns Sistemas Operacionais trazem programas similares ao make, tais como gmake, nmake, tmake, etc. O programa make pode variar de um sistema a outro pois não faz parte de nenhuma normalização.

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

ASCII (American Standard Code for Information Interchange) é um código binário que codifica um conjunto de 128 sinais: 95 sinais gráficos (letras do alfabeto latino, sinais de pontuação e sinais matemáticos) e 33 sinais de controle. Cada código binário possui 8 bits (equivalente a 1 byte), sendo 7 bits para o propósito de codificação e 1 bit de paridade (detecção de erro).

A codificação ASCII é usada para representar textos em computadores, equipamentos de comunicação, entre outros dispositivos que trabalham com texto. Desenvolvida a partir de 1960, grande parte das codificações de caracteres modernas a herdaram como base.

Os sinais não-imprimíveis, conhecidos como caracteres de controle, são amplamente utilizados em dispositivos de comunicação e afetam o processamento do texto.

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

2 SAM4s-EK2

2.1 Gravador/Debug

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; Pino 11 – Entrada de retorno do sinal de clock vindo do alvo; Pino 13 – saída de dados vindos do alvo.

Daisy Chain é um esquema de fiação em que vários dispositivos são conectados juntos em sequência ou em um anel que pode ser usado para a alimentação, sinais analógicos, dados digitais, ou uma combinação dos mesmos.

2.2 Jumpers

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

JP9 – Abilita o NAND FLASH Chip Select

JP - 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 até 20 MHz o qual pode ser contornado (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 programável de 60 até 130MHz PLL (PLLA) capaz de prover um clock MCK para o processador e para os periféricos. A frequência de entrada do PLLA vai 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 diferentes tipos de pinos de alimentação:

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

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

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

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

2.5 LEDs