

1

1.1

O0: é o nível padrão, não há qualquer tentativa de otimização, o código é executado diretamente;

O1: o gcc tenta reduzir o tempo de compilação e o tamanho do executável, ficando menor e mais rápido que o anterior, 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);

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

1.2

- VOLATILE

Indica ao compilador que a variável pode ser modificada sem o conhecimento do programa principal. Dessa forma, o compilador não pode prever com segurança se pode otimizar trechos de programa onde esta variável se encontra, evitando erros de otimização.

- CONST

Essas “variáveis” não podem ter seus valores alterados durante o programa.

- STATIC

O funcionamento destas variáveis depende se estas são globais ou locais:

-Globais funcionam como variáveis que não são (e nem podem ser) conhecidas em outros módulos (arquivos). Isto é útil se quisermos isolar pedaços de um programa para evitar mudanças acidentais em variáveis globais. Isso é um tipo de *encapsulamento* — que é, simplifiadamente, o ato de não permitir que uma variável seja modificada diretamente, mas apenas por meio de uma função.

-Locais estáticas são variáveis cujo valor é mantido de uma chamada da função para a outra.

1.3

O objetivo de Makefile é definir regras de compilação para projetos de software utilizado pelo programa Make para agilizar e simplificar a compilação de programas. 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.

O MakeFile evita a compilação de arquivos desnecessários. Automatiza tarefas rotineiras como limpeza de vários arquivos criados temporariamente na compilação.

1.4

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 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.

2

2.1

A gravação é feita através de 4 pinos, TDI(5), TMS(7), TCK(9), TDO(13). Sendo TCK o sinal de clock, TMS que seqüência a máquina de estados amostrando na borda de subida e TDO a saída.

Daisy Chain é um esquema de fiação em que vários dispositivos são conectados juntos em seqüência ou em um anel., portanto, JTAG Daisy Chain significa conectar múltiplos dispositivos pelas suas portas e controlá-los com um único conector.

2.2

JP3: Reinicializa o conteúdo da memória Flash e alguns dos seus NVM bits;

JP9: NCS0 permite a seleção do chip NAND Flash;

JP11: Mantém o casamento de impedância para a interface RS485.

2.3

Ele gera internamente um clock “lento” (SLCK), o qual é o único clock permanente do sistema; MAINCK que é a saída da seleção do clock principal ou um oscilador de cristal ou um rápido oscilador RC de 4/8/12 MHz; PLLACK/PLLBCK, é a saída do Divisor e PLL programável de 60 a 130 MHz.

2.4

- VDDOUT

Saída do regulador de tensão interno.

- VDDIO

Energiza os periféricos das linhas I/O. **1,62V – 3,6V;**

- VDDIN

Energiza o regulador de tensão interno, ADC, DAC e fontes de alimentação do comparador analógico. **1,8V – 3,6V;**

- VDDPLL

Energiza o PLLA, PLLB e o oscilador de 12 MHz. **1,62V – 1,95V;**

- VDDCORE

Energiza o núcleo, incluindo o processador, incorporando memórias e periféricos. **1,62V – 1,95V;**

2.5

Existem 3 LEDs na placa um azul no pino D2 um verde no D3 e um vermelho no D4. Os LEDs azul e verde são definidos pelo usuário e controlados pelo GPIO, o LED vermelho funciona como um indicativo que a alimentação de 3,3 V está ativa, ele também pode ser definido pelo usuário e controlado pelo GPIO tendo como principal diferença que é controlado através de um transistor MOS.

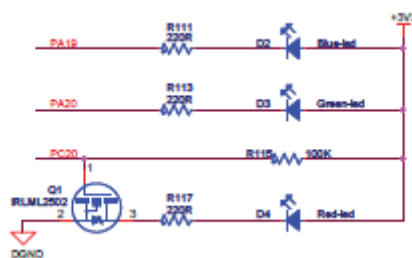


Figura 1 Esquemas LEDs

2.6

O módulo possui dois botões, tais botões são controlados pelo usuário, o primeiro botão está na linha PB3 do I/O e controla o periférico A UTXD1 e o B PCK2, o segundo está na linha PC12 do I/O e controla o periférico ANCS3.

2.7

Tendo como exemplo o ZigBee percebemos que existe um conector macho de 10 pinos para o modulo, A resistência de 0 Ohm deve de ser implementadas em série com as linhas de PIO que são usadas em outros lugares no projeto.

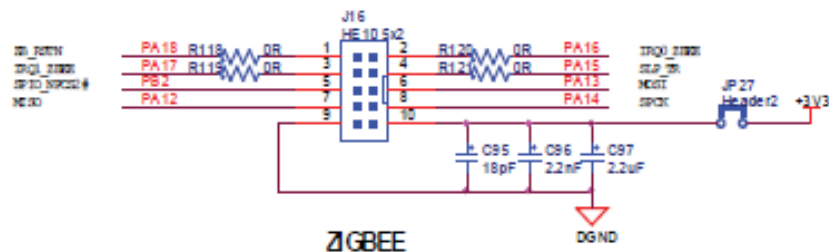


Figura 2 interface do ZigBee

3

3.1

O microcontrolador SAM4SD32C tem uma memória Flash de 2 x 1024 Kbytes e uma SRAM de 160 Kbytes.

3.2

Para o intervalo de temperatura de funcionamento de -40°C a 105°C , o microcontrolador tem uma IOH de -30 mA e uma IOL de 30 mA .

3.3

Brown out em caso de uma queda de tensão funciona como um um circuito que faz com que ocorra um reset, sem esse circuito o processador poderia funcionar mal ou travar, em alguns casos, se um processador recebe uma queda de tensão indesejada e continua operando de qualquer maneira, ela acaba produzindo uma saída sem sentido.

3.4

Um watchdog timer é um dispositivo eletrônico temporizador que dispara um reset ao sistema se o programa principal, devido a alguma condição de erro, deixar de fazer reset no watchdog timer. Este termo Watchdog é utilizado muito em software de medidores eletrônicos, onde tem a finalidade de fiscalizar o processamento e quando necessário aplicar correções e até mesmo um reset no hardware do medidor. Trata-se de um sistema emergencial, quando ativado, precisamos zerar o Watchdog, caso contrário, ele vai estourar e resetar o sistema. Muito utilizado para prevenir as possíveis falhas do sistema.

3.5

O Parallel I/O é utilizado para fazer conexões tanto de entrada de dados como de saída, ele pode ser usado como um controle para os periféricos. Pelo fato de ser paralelo o dispositivo acaba por conseguir fazer operações simultâneas de entradas e saídas.

3.6

De acordo com um fornecedor arrow cada unidade do microcontrolador ATSAM4S-EK2 custa \$151.81 informação retirada de:

https://www.arrow.com/en/products/search?q=SAM4SD32C&filters=Manufacturer_name:Atmel