Pesquisa 3 - C para embarcados e kit de desenvolvimento

1. C

1.1. Otimizações

Quais são os níveis de otimização que o GCC suporta?

O **gcc** pode aperfeiçoar o programa de modo a aumentar seu desempenho e diminuir o tamanho do código de máquina gerado. Por padrão, o gcc não realiza nenhuma otimização (-O0).

Há três níveis de otimização: 1, 2 e 3. Quanto maior o nível, maior deve ser a melhora no desempenho; mas também deve ser maior o tempo de compilação. Para ativar a otimização, deve-se usar as opções -O1, -O2 ou -O3, de acordo com o nível de otimização desejado.

1.2. volatile/const/static

O que são variáveis volatile/const/static?

Na programação em C, existem diferentes tipos de variáveis, cada uma com suas respectivas características:

- A variável **volatile** impede o compilador de realizar a otimização no código envolvendo objetos voláteis, garantindo assim, cada atribuição de variável volátil ler o acesso de memória correspondente.

Exemplo entre variável volatile:

```
int x = 100;

while(x == 100)
{
// codigo
}
```

Nesse caso, o compilador verificará que pode fazer essa otimização, pois o valor de `x` nunca é alterado:

```
while(true)
{
// codigo
}
```

O que pode ser visto no `assembly` gerado:

```
$LL2@main:
    jmp SHORT $LL2@main
```

Enquanto, usando `volatile`, essa otimização não é feita.

```
volatile int x = 100;

while(x == 100)
{
// codigo
}
```

Como pode ser visto no `assembly` gerado:

```
$LL2@main:
    cmp DWORD PTR _x$[ebp], 100
    je  SHORT $LL2@main
```

- **Const** é uma variável do tipo “apenas leitura”, ela pode ser utilizada tal como qualquer outra variável do seu tipo, mas o seu valor não pode ser alterado. Caso o programador atribua um valor neste tipo de variável, ocorrerá erro no programa.

- O funcionamento das variáveis declaradas como **static** depende de se estas são globais ou locais. Variáveis globais static funcionam como variáveis globais apenas no programa em que foram declaradas. Isto funciona como um tipo de encapsulamento. Já as variáveis locais estáticas são variáveis cujo valor é mantido de uma chamada da função para a outra.

1.3. MakeFile

O que é um makefile e qual a sua utilização?

À medida que os programas ficam mais complexos, é mais trabalhoso compilar; principalmente quando existe mais de um arquivo e várias opções adicionais de linha de comando. Como solução pra este problema existe o **Makefile**:
O makefile é um utilitário para configuração de compilação utilizado pelo programa Make, cuja idéia é simplificar e agilizar a compilação de programas. Como exemplo, ele evita a compilação de arquivos desnecessários e analisa as dependências do programa para apenas depois atingir os objetivos do código.

1.4. ASCII

O que é ASCII, e quando é utilizado?

ASCII é conhecido como *Código Padrão Americano para o Intercâmbio de Informação.* Ele é um código binário codificado em caracteres. É um conjunto de 128 sinais:

- 95 sinais gráficos (letras, sinais de pontuação e sinais matemáticos).
- 33 sinais de controle.

A codificação ASCII é usada para representar textos em computadores, equipamentos de comunicação, entre outros dispositivos que trabalham com texto. Na linguagem C, pode-se verificar qual número corresponde ao respectivo símbolo ao declarar uma variável do tipo **Char** e fazê-la receber tal número.

Bin	Oct	Dec	Hex	Sinal
0010 0000	040	32	20	(espaço)
0010 0001	041	33	21	!
0010 0010	042	34	22	"
0010 0011	043	35	23	#
0010 0100	044	36	24	$
0010 0101	045	37	25	%
0010 0110	046	38	26	&
0010 0111	047	39	27	'
0010 1000	050	40	28	(
0010 1001	051	41	29	)
0010 1010	052	42	2A	*
0010 1011	053	43	2B	+
0010 1100	054	44	2C	,
0010 1101	055	45	2D	-
0010 1110	056	46	2E	.
0010 1111	057	47	2F	/
0011 0000	060	48	30	0
0011 0001	061	49	31	1
0011 0010	062	50	32	2
0011 0011	063	51	33	3
0011 0100	064	52	34	4
0011 0101	065	53	35	5
0011 0110	066	54	36	6
0011 0111	067	55	37	7
0011 1000	070	56	38	8
0011 1001	071	57	39	9
0011 1010	072	58	3A	:
0011 1011	073	59	3B	;
0011 1100	074	60	3C	<
0011 1101	075	61	3D	=
0011 1110	076	62	3E	>
0011 1111	077	63	3F	?

Bin	Oct	Dec	Hex	Sinal
0100 0000	100	64	40	@
0100 0001	101	65	41	A
0100 0010	102	66	42	B
0100 0011	103	67	43	C
0100 0100	104	68	44	D
0100 0101	105	69	45	E
0100 0110	106	70	46	F
0100 0111	107	71	47	G
0100 1000	110	72	48	H
0100 1001	111	73	49	I
0100 1010	112	74	4A	J
0100 1011	113	75	4B	K
0100 1100	114	76	4C	L
0100 1101	115	77	4D	M
0100 1110	116	78	4E	N
0100 1111	117	79	4F	O
0101 0000	120	80	50	P
0101 0001	121	81	51	Q
0101 0010	122	82	52	R
0101 0011	123	83	53	S
0101 0100	124	84	54	T
0101 0101	125	85	55	U
0101 0110	126	86	56	V
0101 0111	127	87	57	W
0101 1000	130	88	58	X
0101 1001	131	89	59	Y
0101 1010	132	90	5A	Z
0101 1011	133	91	5B	[
0101 1100	134	92	5C	\
0101 1101	135	93	5D	]
0101 1110	136	94	5E	^
0101 1111	137	95	5F	_

Bin	Oct	Dec	Hex	Sinal
0110 0000	140	96	60	`
0110 0001	141	97	61	a
0110 0010	142	98	62	b
0110 0011	143	99	63	c
0110 0100	144	100	64	d
0110 0101	145	101	65	e
0110 0110	146	102	66	f
0110 0111	147	103	67	g
0110 1000	150	104	68	h
0110 1001	151	105	69	i
0110 1010	152	106	6A	j
0110 1011	153	107	6B	k
0110 1100	154	108	6C	l
0110 1101	155	109	6D	m
0110 1110	156	110	6E	n
0110 1111	157	111	6F	o
0111 0000	160	112	70	p
0111 0001	161	113	71	q
0111 0010	162	114	72	r
0111 0011	163	115	73	s
0111 0100	164	116	74	t
0111 0101	165	117	75	u
0111 0110	166	118	76	v
0111 0111	167	119	77	w
0111 1000	170	120	78	x
0111 1001	171	121	79	y
0111 1010	172	122	7A	z
0111 1011	173	123	7B	{
0111 1100	174	124	7C	|
0111 1101	175	125	7D	}
0111 1110	176	126	7E	~

Tabela 1 - Sinais gráficos com seus respectivos códigos numéricos.

2. SAM4s-EK2

2.1. Gravador / Debug

Como funciona a gravação via JTAG (Joint Test Action Group), quais são os pinos utilizados do microcontrolador. O que é jtag daisy chain e qual a sua funcionalidade ?

O conector JTAG é um dos padrões utilizados no microcontrolador da Atmel SAM4S-EK2 para depuração de um projeto. Ele possui 20 pinos distribuídos para comunicação com emulador SAM-ICE que funciona como ponte entre o computador e o microcontrolador.

Após a instalação dos devidos softwares e drivers, a comunicação ocorre entre o USB (serial) e conector JTAG (paralelo). Os pinos do microcontrolador utilizados são PB4, PB5, PB6, PB7, PB8. Daisy chain é utilizado para conexão de diversos dispositivos conectados em série (com diferentes clocks) que serão controlados.

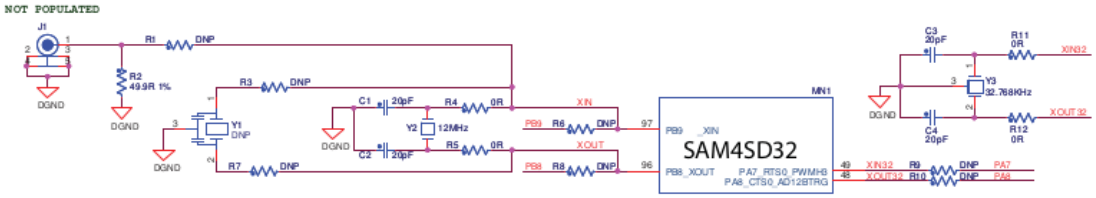
2.2. Jumpers

Qual a função dos seguintes "jumpers" da placa:

- **JP3**
- **JP9**
- **Outro jumper de sua escolha**
- O jumper JP3 serve para reinicializar a memória flash e memórias não-voláteis;
- O jumper JP9 serve como controle de memória estática;
- O jumper JP13 seleciona o LCD do microcontrolador.

2.3. Clock

Analise o esquemático e a documentação e descreva como funcionam os clocks da placa e microcontrolador e quais são suas opções.



O gerador de clock do microcontrolador é composto de:

- Oscilador de Baixa Potência de 32,768 kHz de clock lento com modo "bypass" , o único clock permanente do sistema;
- Oscilador de Cristal de 3 a 20 Hz, que pode ser substituído em conexões USB (que necessitam de 12 MHz) e pode ser selecionado na saída de Oscilador de Clock Principal;
- Oscilador RC interno programado de fábrica, com freqüências selecionáveis de 4, 8 ou 12 MHz e que também pode ser selecionado na saída de Oscilador de Clock Principal;
- Um PLL (PLLB) ou *Phase Locked Loop* de 60 a 130 MHz, gerador de clock para o Controlador USB de Alta Velocidade;
- Um PLL (PLLA) de 60 a 130 Hz programável, capaz de gerar clock MCK para o processador e os periféricos. A freqüência de entrada do PLLA é de 7,5 a 20 MHz.

O SAM4S-EK2 vem equipado com um cristal de 12MHz, Ressonador Óptico Piezelétrico Cerâmico de 12 MHz, um cristal de 32,768 kHz e uma conector de entrada para clock externo opcional.

2.4. Alimentação

Qual o nível de tensão de operação do microcontrolador? Como é feita a sua alimentação?

O produto tem diferentes tipos de pinos de alimentação:

- Pino VDDIN: alimentação para o regulador interno de tensão, conversores A/D, D/A, e comparador analógico de tensões. O nível de tensão vai de 1,8 a 3,6 V;
- Pino VDDIO: alimentação para as linhas periféricas I/O. O nível de tensão vai de 1,62 a 3,6 V;
- Pino VDDOUT: saída do regulador de tensão interno;
- Pino VDDCORE: alimentação para o núcleo, incluindo processador, memórias e periféricos. O nível de tensão vai de 1,62 a 1,95 V;
- Pino VDDPLL: alimentação para o PLL A, PLL B e oscilador de 12MHz. O nível de tensão vai de 1,62 a 1,95 V.

2.5. LEDs

Como funcionam os leds da placa? Quais são os pinos do microcontrolador dedicados para eles? Qual deve ser o valor nos pinos para ligar e desligar os LEDs?

Há 3 LED's na placa, sendo um azul, um verde e um vermelho. O azul (pino PA19) e o verde (pino PA 20) são definidos e controlados pelo GPIO. O vermelho (pino PC20) indica que a trilha está energizada com 3,3 V, mas também pode ser controlado pelo GPIO e ser usado como um dos LED's anteriores, a diferença é que ele é controlado por um transistor MOS. Quando a linha PIO está desabilitada, um resistor pull-up controla o MOS para acender o LED quando a energia está ligada.

2.6. Botões

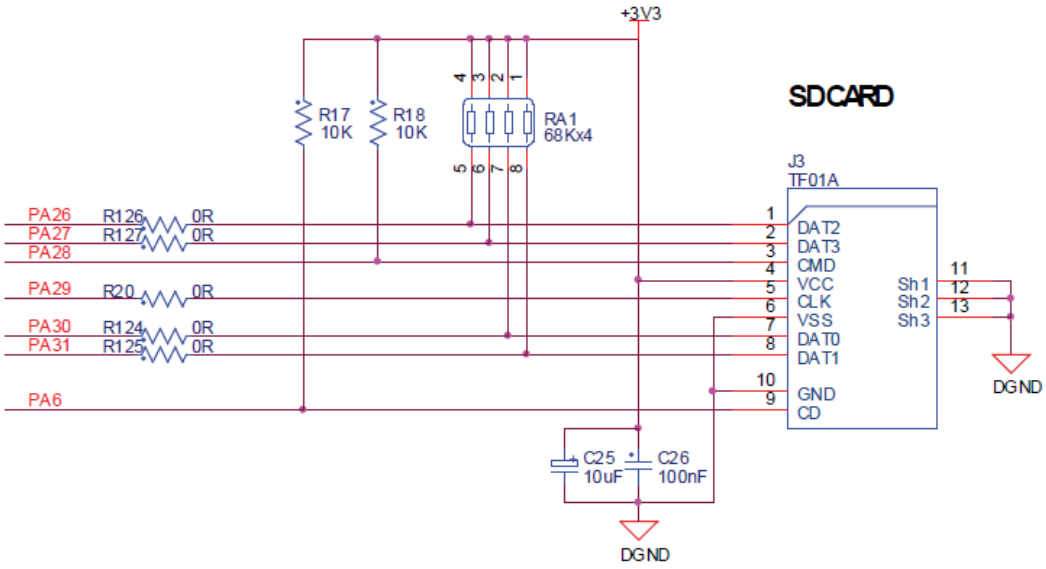
Como funcionam os botões (push buttons) da placa? Quais são os pinos do microcontrolador dedicados para eles?

Há dois botões mecânicos na placa conectados às linhas de PIO que definem "direita" e "esquerda" por padrão e um que controla o "reset" do sistema. Os pinos dedicados a eles são, respectivamente, PB3, PC12 e NRST.

2.7. Periféricos

Escolha um dos periféricos do kit de desenvolvimento (LCD, SDCARD, Microfone, Speaker, NAD FLASH, RS232, USB, ZigBee, QTouch) e explique sua funcionalidade descrevendo os pinos utilizados e a solução de hardware empregada (analise o esquemático e os componentes empregados).

SDCARD: O SAM4S - EK2 tem uma interface MMC de multimídia de 4-bits de alta velocidade, o qual está conectado a um slot para microcard SD/MMC de 4-bits com um interruptor de detecção de placa. Os pinos utilizados são PA26, PA27, PA28, PA29, PA30, PA31 e PA6.



3. SAM4SD32C

3.1. Memória

Quais são as memórias internas do uC ATSAM4SD32C e seus tamanhos?

O µC estudado contém 2MB de memória Flash, 160KB de memória SRAM, e 2KB de memória Cache integrada.

3.2. IOH, IOL

Qual a corrente máxima suportada de entrada (I_{OH}) e de saída (I_{OL})?

A corrente máxima suportada pelo I_{OH} e I_{OL} é de 30 mA.

3.3. Brownout

O que é Brownout?

O BrownOut é utilizado para reiniciar a CPU automaticamente quando a tensão de alimentação cair abaixo de um nível seguro para a operação do circuito; deste modo, no momento de reset, a CPU executará as instruções a partir do vetor de reset (0x000 do código). O BrownOut deve ser habilitado nos bits de configurações do µC, e dependendo dos µC os níveis de tensão seguros de operação em µC são fixos, ou selecionados pelo usuário.

3.4. Watchdog Time

Que é o Watchdog Time e qual o seu uso?

O Watchdog Timer, abreviadamente WDT, é um dispositivo de tempo que provoca um reset no uC quando o tempo programado expira. O WTD serve para reiniciar o programa de tempos em tempos, seja pelo fato dele poder travar durante a execução ou entrar em loop infinito ou qualquer outra razão que se tenha para manter o programa sempre ativo. O WDT é ativado e desativado pelos bits de configuração e, em alguns uC, isso pode ser feito no software. O WDT possui um oscilador independente do oscilador do uC, com um período de timeout configurável em milissegundos a segundos, e pode ficar ativo durante o modo sleep.

3.5. PIO

Descreva as funcionalidades do PIO:

Os PIOs servem para gerar uma liberdade ao usuário de fazer leitura e enviar sinais pelo µC. Pode-se configurar os PIOS para que o µC tenha acesso a sensores, e envie sinais de controle, como no caso, por exemplo,do controle da velocidade de rotação de um motor.

3.6. Custo

Pesquise nos fornecedores o valor de mercado desse chip:

Fazendo a pesquisa em alguns fornecedores encontrou-se o valor médio de 13 Euros (R$ 56,00 reais) para a compra somente do chip ATSAM4SD32C.