

FreeRtos

Configurações no/do compilador

Felipe Bandeira, 15/06/2012

Ferramentas utilizadas:

- Mplab X, versao 1.20
- C30, versao 3.24
- P24FJ256GB106, P24FJ128GA010 e PICKIT3

PIC30-as:

Em **Additional Options** utilizar "-g" para debug, padrao no freertos. Desmarcar todas as opções em **Other Options**.

Options for pic30-as (v3_24_0)

Option categories: **Other Options** [Reset]

Allow CALL optimizations	<input type="checkbox"/>
Keep local symbols	<input type="checkbox"/>
Diagnostics level	Generate warnings
Include source code	<input type="checkbox"/>
Expand macros	<input type="checkbox"/>
Include false conditionals	<input type="checkbox"/>
Omit forms processing	<input type="checkbox"/>
Include assembly	<input type="checkbox"/>
Link symbols	<input type="checkbox"/>

Additional options: -g

Option Description: Generated Command Line

Options for pic30-as (v3_24_0)

Option categories: **Symbols & Macros** [Reset]

Preprocessor macro definitions (.S only)	...
Assembler symbols	...
Preprocessor Include directories	.././include;.././include;.././include;.././include
Assembler Include directories	..

Additional options: -g

Option Description: Generated Command Line

PIC30-gcc:

Em **Preprocessor macros** a string "MPLAB_PIC24_PORT" é utilizada para informar na compilação do FreeRTOS que a troca de contexto será desenvolvida para o PIC24. Outra alternativa é colocar no arquivo de configuração do FreeRTOS a definição: "#define MPLAB_PIC24_PORT". Na categoria **Optimization** desmarcar todas as opções e desabilitar todas as preotimizações. O nível (level) de otimização utilizado pelos exemplos é o **1**. Na categoria **Memory Model** utilizar "Large code model" para **Code Model**. Inserir "-fno-omit-frame-pointer" em **Additional options**.

Options for pic30-gcc (v3_24_0)

Option categories: **General** [Reset]

Have symbols in production build	<input checked="" type="checkbox"/>
Support all ANSI-standard programs	<input type="checkbox"/>
Isolate each function in a section	<input type="checkbox"/>
Make warnings into errors	<input type="checkbox"/>
Use 64-bit double	<input type="checkbox"/>
Additional warnings	<input checked="" type="checkbox"/>
Strict ANSI warnings	<input type="checkbox"/>
Enable large arrays	<input type="checkbox"/>
Preprocessor macros	MPLAB_PIC24_PORT
Include directories	.././include;.././include;.././include;.././include

Additional options: -fno-omit-frame-pointer

Options for pic30-gcc (v3_24_0)

Option categories: **Memory Model** [Reset]

Code model	Large code model
Data model	Default for data model
Scalar model	Default for scalar model level
Location of constants	Default for location of constants

Additional options: -fno-omit-frame-pointer

Options for pic30-gcc (v3_24)

Option categories: Optimization Reset

optimization-level	1
Unroll loops	<input type="checkbox"/>
Omit frame pointer	<input type="checkbox"/>
Procedural abstraction	<input type="checkbox"/>
Pre-optimization instruction scheduling	Disable
Post-optimization instruction scheduling	Disable

Additional options: -fno-omit-frame-pointer

PIC30-ld:

Cuidado com o **Heap size**, a pilha TCP/IP da microchip utiliza acima de 2048.

Options for pic30-ld (v3_24 0)

Option categories: General Reset

Heap size (bytes)	0
Stack size (bytes)	
Allow overlapped sections	<input type="checkbox"/>
Initialize data sections	<input checked="" type="checkbox"/>
Pack data template	<input checked="" type="checkbox"/>
Create handles	<input checked="" type="checkbox"/>
Create default ISR	<input checked="" type="checkbox"/>
Remove unused sections	<input type="checkbox"/>
Symbols	Keep all

Aumentando o tempo de compilação:

Como o C30 utiliza como construtor dos objetos o "make". Sendo o "make" um projeto GNU, temos algumas características, como criação de jobs para otimizar a compilação.

Apenas usando a sintaxe:

```
make -j [quantidade]
```

Quanto maior a quantidades de jobs mais memoria é processamento é exigido. Para configurar o MPLABX, faça:

Em tools -> options

No meu caso eu deixei quantidade infinita!

