



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO



MANUAL DE UTILIZAÇÃO DO *TMS320C6713 DSP Starter Kit (DSK)*

Desenvolvedor:

Eng^o M.Sc. Jean Mário Moreira de Lima

Setembro
2018

Sumário

1	Introdução	1
2	Primeiros Passos	3
2.1	Ligando a placa	3
2.2	Desenvolvimento no <i>Code Composer</i> TM	3
2.2.1	Criando um Projeto no CCS	3
2.2.2	Programação no CCS	6
3	Perguntas Frequentes	8

1 Introdução

O *TMS320C6713 DSP Starter Kit (DSK)* desenvolvido em conjunto com a *Spectrum Digital* é uma plataforma de desenvolvimento de baixo custo projetada para acelerar o desenvolvimento de aplicações de alta precisão baseadas na geração DSP de ponto flutuante da TI (*Texas Instruments*). O kit usa comunicações USB para funcionalidade *plug-and-play* e um software completo para desenvolvimento, o *Code Composer Studio™ (CCS) Integrated Development Environment (IDE)*.



Figura 1: *TMS320C6713 DSP Starter Kit (DSK)*

O DSP TMS320C6713 é um dispositivo de 225 MHz que fornece até 1800 milhões de instruções por segundo (MIPs) e 1350 MFLOPS. Essa geração de DSP foi projetada para aplicativos que exigem alta precisão. O C6713 é baseado na plataforma DSP TMS320C6000 projetada para as necessidades de aplicações de alta precisão e desempenho, como pró-áudio, médica e diagnóstico. Outros recursos de hardware da placa incluem:

- Suporte JTAG embarcado via USB
- Codec estéreo de 24 bits de alta qualidade
- Quatro conectores de áudio de 3,5 mm para microfone, entrada de linha, alto-falante e saída de linha

- Flash de 512K e 16 MB de SDRAM
- Conector da porta de expansão para módulos plug-in
- Interface padrão IEEE JTAG integrada
- Fonte de alimentação universal de + 5V

2 Primeiros Passos

Nesta seção, explica-se brevemente os passos iniciais para utilização da placa DSP.

2.1 Ligando a placa

Para ligar a placa corretamente, conecte a fonte presente no *Starter Kit* a placa. Uma vez que a fonte é conectada, o POST (*Power On Self Test*) é executado. Os LEDs 0-3 piscarão. Quando o POST é terminado com sucesso, todos os LEDs piscam ON e OFF e permanecem ON.



Figura 2: *TMS320C6713 DSP board*

A partir desse ponto, a placa DSP está funcional e você pode conectá-la ao computador com o cabo USB também presente no kit. Agora, os próximos passos são criar, compilar e executar um projeto na placa utilizando o software de desenvolvimento *Code Composer™*.

2.2 Desenvolvimento no *Code Composer™*

2.2.1 Criando um Projeto no CCS

1. Para criar um novo projeto no CSS: **File->New->CCS Project**.

2. A caixa de diálogo de um novo projeto vai abrir. Preencha as seguintes configurações:
- (a) **Target:** C671x Floating-point DSP / TMS320C6713
 - (b) **Connection:** Spectrum Digital DSK-EVM-eZdsp onboard USB Emulator
 - (c) **Project Name:** Escolha o nome do seu projeto
 - (d) **Compiler Version:** TI V7.4.24 (versão instalada no CCS)

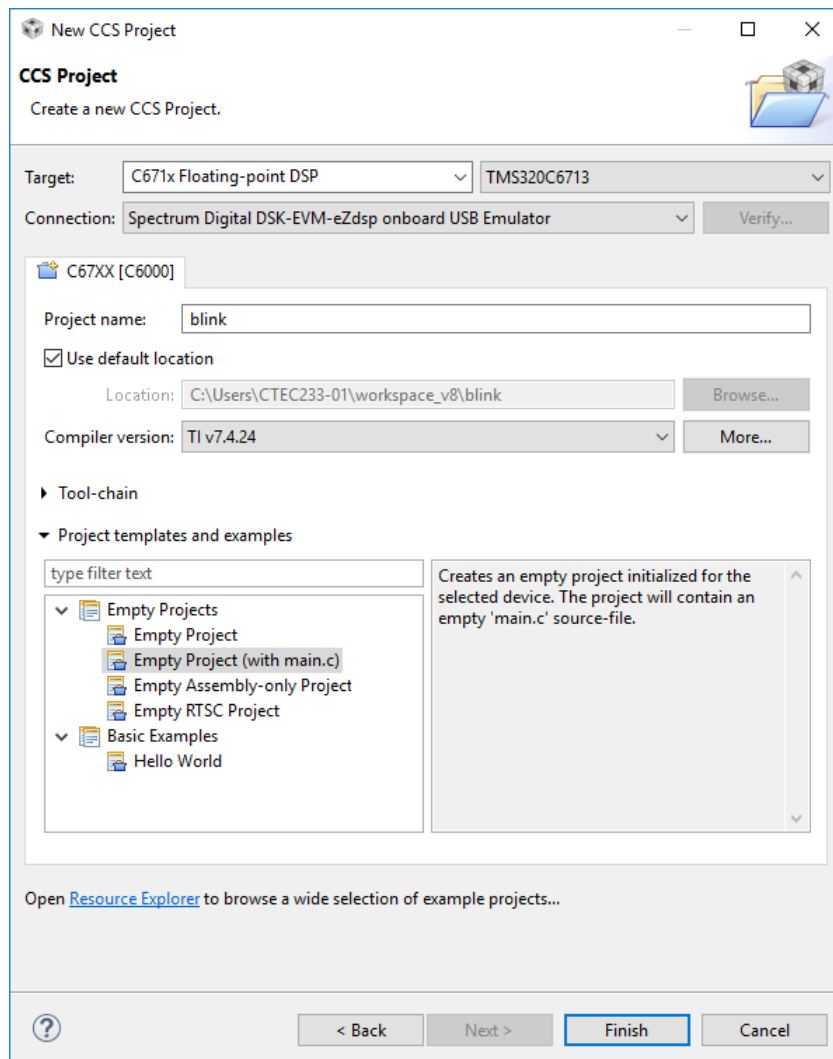


Figura 3: *New Project Dialog Box*

3. Clique em **Finish** para criar o projeto.

4. Agora é preciso "linkar" as bibliotecas com os códigos necessários para aplicações na DSP. Na janela *Project Explorer*, clique o botão direito no seu projeto e depois em **Show Build Settings**.
5. Uma nova caixa de diálogo vai aparecer. Clique **Build->C6000 Linker->File Search Path** ao lado esquerdo da janela. Acima da primeira caixa de texto, clique no ícone que é semelhante a uma folha com um sinal de mais (+) verde para adicionar a biblioteca. Digite **dsk6713bsl.lib** e clique **Ok**.

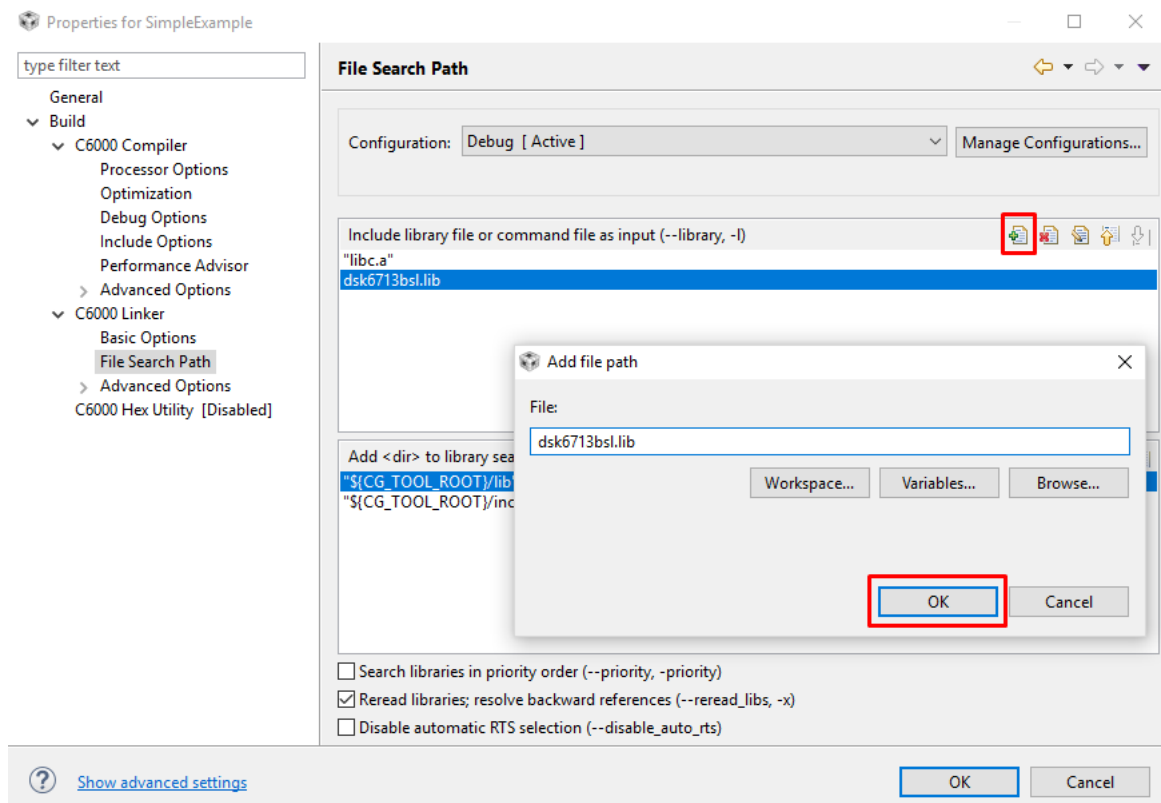


Figura 4: Adicionando Bibliotecas

6. Repita o passo 5 para adicionar outra biblioteca chamada: **cs16713.lib**.
7. Na opção **Configuration** altere a opção *Debug|Active* para *Release*. Adicione as mesmas bibliotecas de antes repetindo os passos 5 e 6. Após isso, altere mais uma vez o campo **Configuration** de volta para *Debug|Active*. Clique em **Apply and Close**.

2.2.2 Programação no CCS

Para ilustrar a programação no CCS, vamos utilizar um exemplo:

```
/*
 * main.c
 */
#define CHIP_6713

#include "dsk6713.h"
#include "dsk6713_led.h"
#include "dsk6713_flash.h"

void main(void)
{
    DSK6713_init();
    DSK6713_LED_init();

    while (1)
    {
        DSK6713_LED_toggle(2);
        DSK6713_waitusec(1000000);
    }
}
```

Ao se criar um projeto no CSS, por padrão, um arquivo *main.c* é criado. Na paleta *Project Explorer* é possível encontrá-lo. Abra o arquivo *main.c* do seu projeto e cole o código-exemplo desse material.

1. A instrução *#define CHIP_6713* é necessária para TODO e QUALQUER projeto para que as bibliotecas reconheçam que placa será usada.
2. Para importar as funções utilizadas nesse exemplo, é preciso incluir três arquivos **dsk6713.h**, **dsk6713_led.h**, **dsk6713_flash.h**. De acordo com as funções necessárias ao seu projeto, é preciso importar os arquivos .h (*headers*) as classes onde as funções estão implementadas.
3. Nos seus projetos futuros, lembre-se de chamar a função: *DSK6713_init()* e outras funções *init's* necessárias.
4. Nesse exemplo, o LED 2 vai piscar a uma taxa de 1Hz.

Após copiar o código do exemplo no seu arquivo *main.c*:

1. Click no ícone similar a um martelo para construir o binário do seu código, com extensão .out. Você vai saber se a construção foi feita com sucesso se os painéis **Console** e **Problems** não exibirem algum(s) erro(s).

- (a) Em caso de erro relacionado com "*undefined reference...*", quer dizer que, provavelmente, você cometeu algum erro no momento de linkar as bibliotecas (.lib).
- (b) Em caso de erro relacionado com "*cannot find some include .h file*", quer dizer que, provavelmente, você esqueceu de copiar os arquivos de *include* nos diretórios corretos.

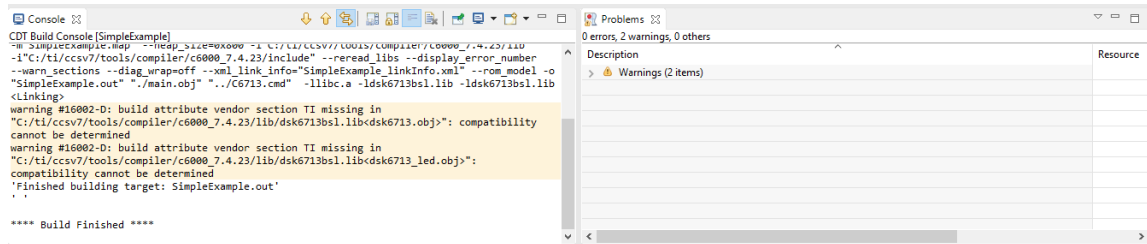


Figura 5: Tela de *Console e Problems*

2. Finalmente, agora vamos fazer carregar o código binário na placa e rodá-lo. Clique em **Run->Debug** ou pressione **F11**.
 - (a) CSS vai mudar do modo *edit* para o *run*.
 - (b) O painel **Debug**, você deve ver o nome do emulador e entre parênteses: (**Suspended - SW Breakpoint**). Isso significa que o binário foi carregado com sucesso e está esperando para ser executado.

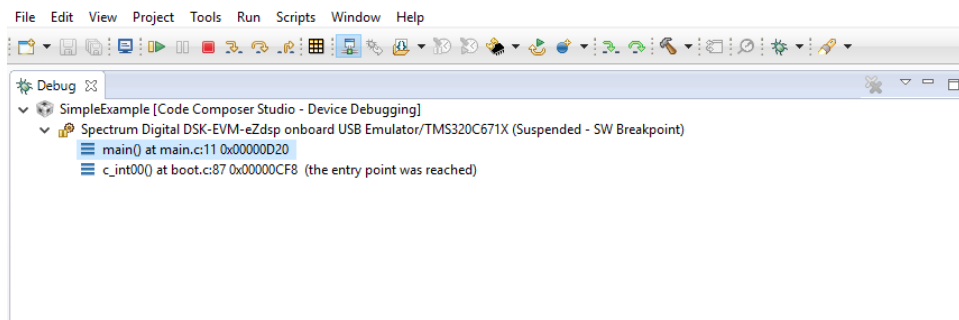


Figura 6: Tela de *Debug*

- (c) Clique no botão **Resume** ou pressione **F8**. Agora, seu código deve estar rodando e o painel **Debug** deve mostrar (*Running*).
3. Quando for necessário fazer mudanças no seu código e testá-lo novamente, não há a necessidade de executar todos esses passos novamente. Você pode seguir as instruções seguintes:

- (a) Clique no botão **Suspend** ou pressione **Alt+F8**.
- (b) Agora, você pode alterar seu código. Por fim, pressione o botão **Build**. Clique em **Yes** para recarregar o programa automaticamente. Assim, estará na mesma situação do item 2. Se clicar em **Resume** estará rodando seu código novamente.

3 Perguntas Frequentes

1. Onde posso encontrar esquemáticos, manuais técnicos e documentação do DSP?
 - Toda a documentação necessária pode ser encontrada nesse link:
[<http://c6000.spectrumdigital.com/dsk6713/revc/>](http://c6000.spectrumdigital.com/dsk6713/revc/)
2. Como saber que funções estão disponíveis para uso?
 - A BSL (*Board Support Library*) API pode ser vista no arquivo de ajuda **C6713DSK.chm** localizado dentro do diretório **docs**. Clique duas vezes no arquivo **.chm** e o menu vai abrir:
 - Além disso, todas as funções tradicionais da linguagem C estão disponíveis: *free*, *printf*, *dfracnumdentextitmalloc*, etc.

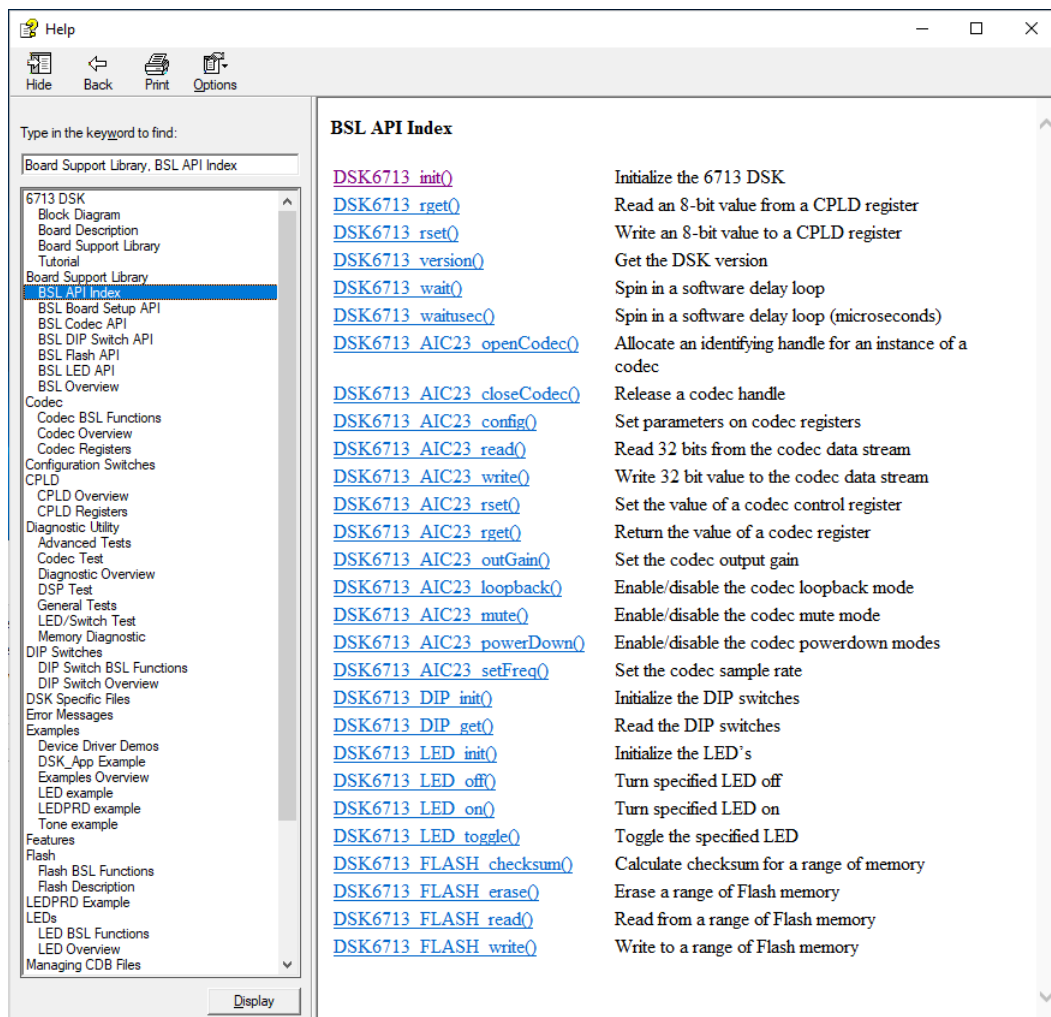


Figura 7: Tela de *Console e Problems*