

Guia de criação de projeto para F28379D

Instalação do ambiente de trabalho

Faça o download da IDE Code Composer Studio [neste link](#).

ATENÇÃO: este guia foi criado visando a versão 12.2.0. Não garantimos que os passos serão os mesmos para outras versões. A versão 20.X é drasticamente diferente da 12, e portanto não a recomendamos.

Durante a instalação, selecione a opção “Custom installation”, após a qual você deverá selecionar **apenas** a opção “C2000 real-time MCUs”. Você poderá concluir a instalação normalmente após isso.

Instalação da biblioteca da placa

Faça o download da biblioteca C2000Ware [neste link](#).

ATENÇÃO: este guia foi criado visando a versão 5.04.00.00. Não garantimos que os passos serão os mesmos para outras versões.

Durante a instalação, lembre-se de anotar o caminho utilizado, você precisará dele em seguida. De agora em diante chamaremos este caminho de “\$CW”

Criação de uma biblioteca estática

Se você já criou uma biblioteca estática como esta antes, você pode reutilizá-la em outros projetos, e portanto pode pular esta etapa.

Abra o Code Composer Studio (CCS), e clique em **File→New→CCS Project**.

Em **Target**, escreva **F28379D** e selecione a única opção disponível (**TMS320F28379D**).

Em **Project name**, escreva **C2000WareLib** (este nome não importa)

Expanda **Project type and tool-chain**:

Em **Output type** selecione **Static Library**.

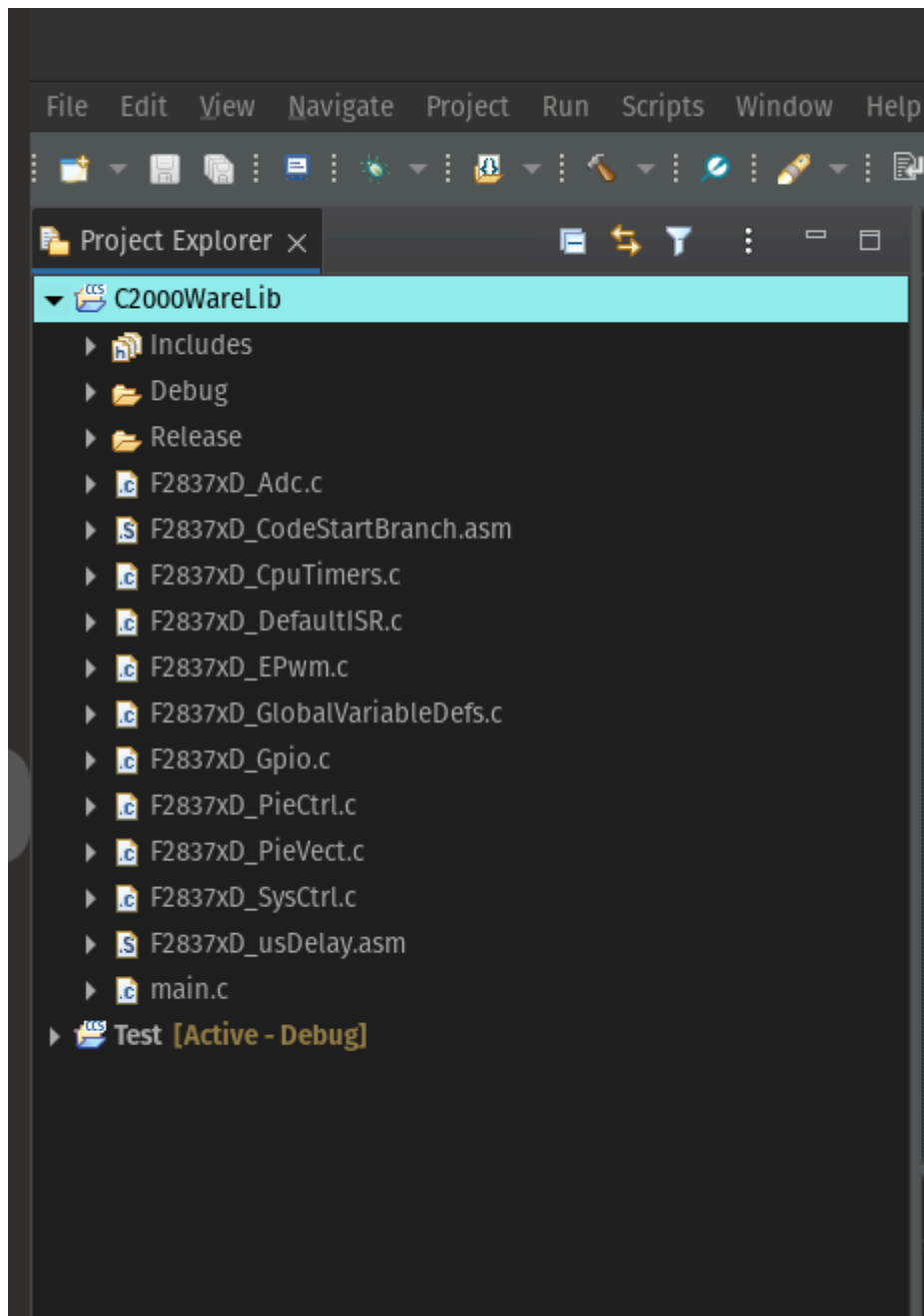
Em **Output format** selecione **eabi (ELF)**.

Expanda **Project templates and examples**:

Selecione **Empty Project** (sem nada mais).

Clique em **Finish**

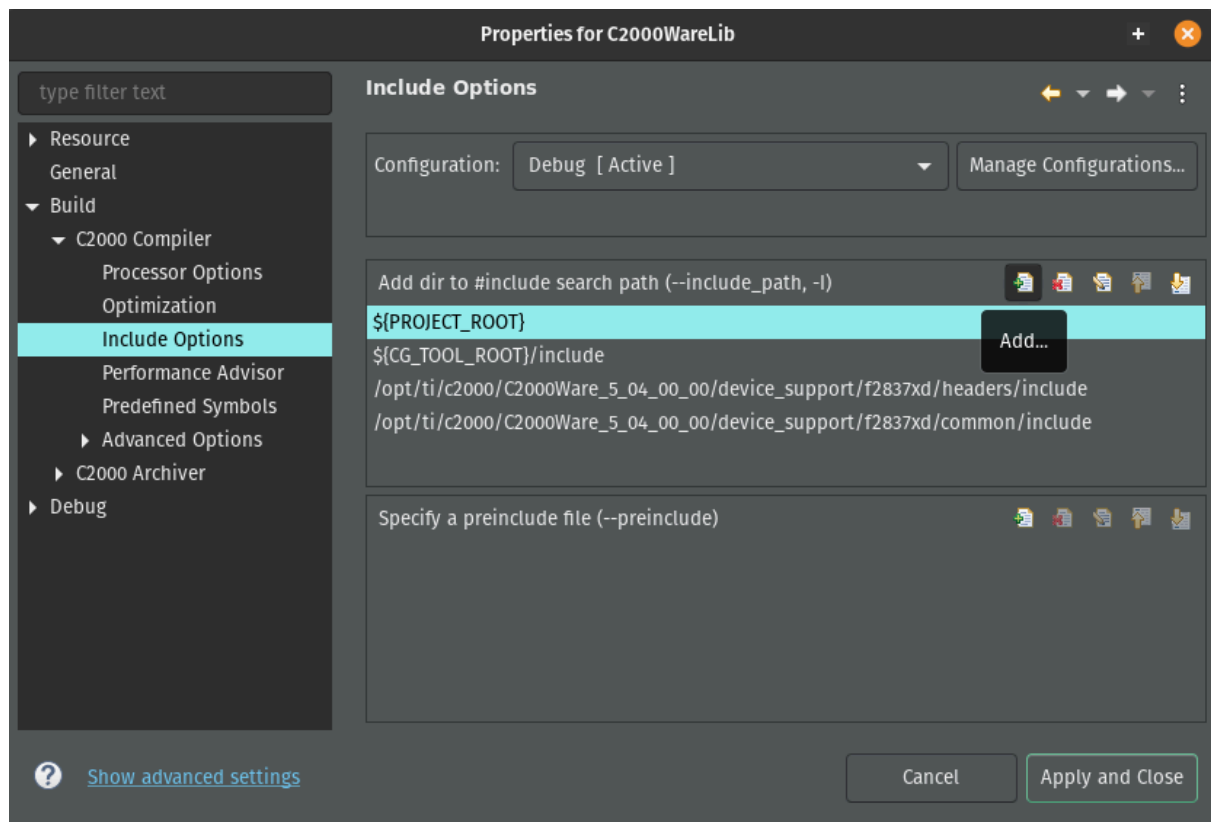
Com a biblioteca criada, ela deve aparecer na barra lateral na esquerda, como na imagem a seguir:



Se você não consegue ver a aba dos projetos, clique em **Window**→**Show View**→**Project Explorer**.

Com o botão direito na base do projeto, selecione **Properties**.

Clique em **Build**→**C2000 Compiler**→**Include Options**, e então selecione o botão de adicionar marcado na figura abaixo:



Clique em **Add**, e coloque o seguinte caminho, trocando \$CW pelo caminho onde você instalou a biblioteca C2000Ware: “\$CW/device_support/f2837xd/headers/include”.

Repita o mesmo processo, e adicione o caminho “\$CW/device_support/f2837xd/common/include”.

Clique em **Build**→**C2000 Compiler**→**Predefined Symbols**, e então adicione os seguintes nomes: “CPU1” e “_LAUNCHXL_F28379D”.

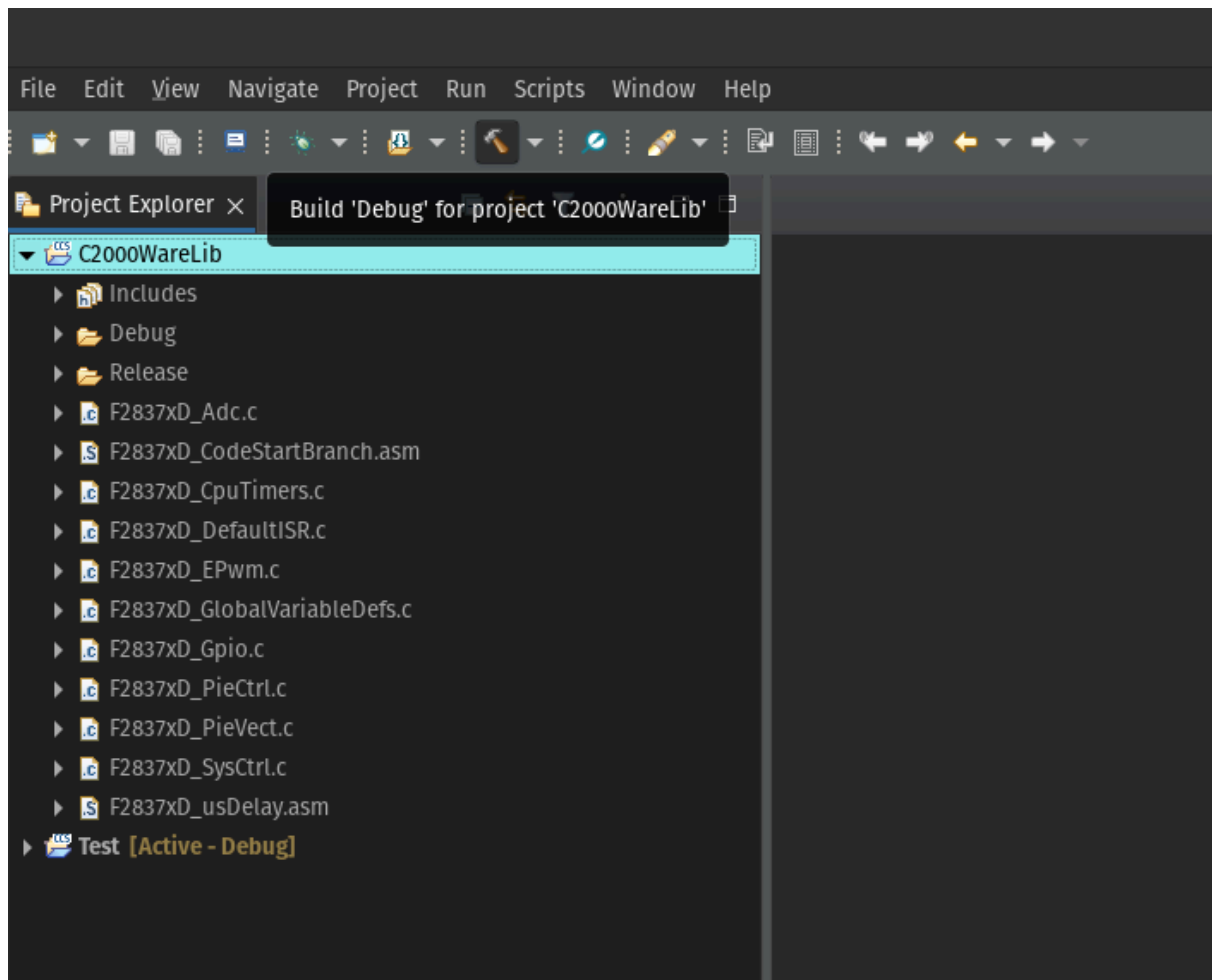
Clique em **Apply and Close**.

Novamente clique com botão direito na base do projeto, mas desta vez clique em **Add Files**. Navegue até:

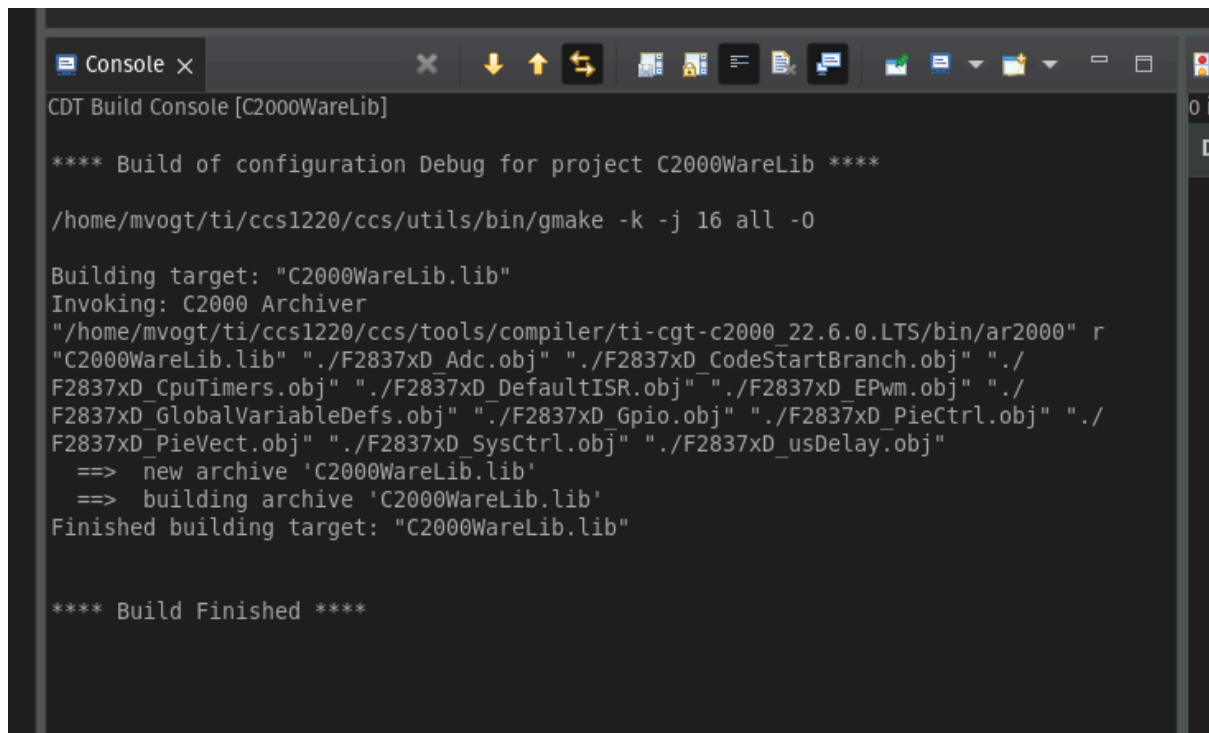
- “\$CW/device_support/f2837xd/headers/source”, e importe:
 - GlobalVariableDefs.c
- “\$CW/device_support/f2837xd/common/source”, e importe:
 - Adc.c
 - CodeStartBranch.asm
 - CpuTimers.c

- DefaultISR.c
- EPwm.c
- Gpio.c
- PieCtrl.c
- PieVect.c
- SysCtrl.c
- usDelay.asm

Com todos os arquivos importados, faça a build do projeto, clicando no ícone do martelo:



Se você não teve nenhum erro de compilação ou de linkagem, você fez tudo corretamente.



```
CDT Build Console [C2000WareLib]

**** Build of configuration Debug for project C2000WareLib ****

/home/mvogt/ti/ccs1220/ccs/utils/bin/gmake -k -j 16 all -O

Building target: "C2000WareLib.lib"
Invoking: C2000 Archiver
"/home/mvogt/ti/ccs1220/ccs/tools/compiler/ti-cgt-c2000_22.6.0.LTS/bin/ar2000" r
"C2000WareLib.lib" "./F2837xD_Adc.obj" "./F2837xD_CodeStartBranch.obj" "./
F2837xD_CpuTimers.obj" "./F2837xD_DefaultISR.obj" "./F2837xD_EPwm.obj" "./
F2837xD_GlobalVariableDefs.obj" "./F2837xD_Gpio.obj" "./F2837xD_PieCtrl.obj" "./
F2837xD_PieVect.obj" "./F2837xD_SysCtrl.obj" "./F2837xD_usDelay.obj"
==> new archive 'C2000WareLib.lib'
==> building archive 'C2000WareLib.lib'
Finished building target: "C2000WareLib.lib"

**** Build Finished ****
```

Criação de um projeto

Agora que você já possui a biblioteca estática, vamos criar um projeto para piscar a LED da placa.

Clique em **File**→**New**→**CCS Project**.

Em **Target**, selecione a placa **TMS320F28379D**

Em **Project name**, dê um nome para seu projeto.

Expanda **Project type and tool-chain**.

Em **Output type**, selecione **Executable**.

Em **Output format**, selecione **eabi (ELF)**.

Clique em **Finish**.

Com o botão direito na base do projeto, selecione **Properties**.

Clique em **Build**→**C2000 Compiler**→**Include Options**

Clique em **Add**, e coloque o seguinte caminho, trocando \$CW pelo caminho onde você instalou a biblioteca C2000Ware: "\$CW/device_support/f2837xd/headers/include".

Repita o mesmo processo, e adicione o caminho

"\$CW/device_support/f2837xd/common/include".

Clique em **Build**→**C2000 Compiler**→**Predefined Symbols**, e então adicione os seguintes nomes: "CPU1" e "_LAUNCHXL_F28379D".

Clique em **Build**→**C2000 Linker**→**File Search Path**, e em **Include library file or command file as input**:

- Clique em **Add**, e então em **Workspace**. Expanda o projeto da biblioteca estática, e então expanda **Debug**, e selecione o arquivo **.lib**.
- Adicione os seguintes caminhos:
 - "\$CW/device_support/f2837xd/headers/cmd/F2837xD-Headers_nonBIOS_cpu1.cmd"
 - "\$CW/device_support/f2837xd/common/cmd/2837xD_RAM_Ink_cpu1.cmd"

Clique em **Apply and Close**.

Com isso, o projeto está configurado. Para implementar o código para piscar o LED, clique com o botão direito na base do projeto e selecione **New**→**File** e nomeie como **main.c**, por exemplo. Os DSPs da Texas Instruments possuem suporte a códigos de exemplo disponíveis em "\$CW/device_support/f2837xd/examples/cpu1". Nesta pasta, navegue para "**blinky/cpu01**", copie o código do arquivo **blinky_cpu01.c** e cole em seu arquivo **main.c**. Este código deve piscar o LED incorporado ao DSP intermitentemente.

Guia de criação de projeto para F28335

Criação de uma biblioteca estática

Abra o Code Composer Studio (CCS), e clique em **File**→**New**→**CCS Project**.

Em **Target**, escreva **F28335** e selecione **TMS320F28335**.

Em **Project name**, escreva **C2000WareLib** (este nome não importa)

Expanda **Project type and tool-chain**:

Em **Output type** selecione **Static Library**.

Em **Output format** selecione **eabi (ELF)** ou **legacy COFF**.

Expanda **Project templates and examples**:

Selecione **Empty Project** (sem nada mais).

Clique em **Finish**

Com o botão direito na base do projeto, selecione **Properties** e em **Build**→**C2000 Compiler**→**Include Options**, adicione os caminhos para as duas pastas de “include”, que estão localizadas nos seguintes diretórios “\$CW/device_support/f2833x/headers/include” e “\$CW/device_support/f2833x/headers/include”.













Não defina nenhum símbolo em **Build**→**C2000 Compiler**→**Predefined Symbols**.

Novamente clique com botão direito na base do projeto, clique em **Add Files**. Navegue até os diretórios abaixo e inclua os seguintes arquivos:

- “\$CW/device_support/f2833x/headers/source”:
 - DSP2833x_GlobalVariableDefs.c
- “\$CW/device_support/f2833x/common/source”, e importe:
 - DSP2833x_Adc.c
 - **DSP2833x_ADC_cal.asm** (este não existe no F28379D)
 - DSP2833x_CodeStartBranch.asm
 - DSP2833x_CpuTimers.c
 - DSP2833x_DefaultIsr.c
 - DSP2833x_EPwm.c

- DSP2833x_Gpio.c
- DSP2833x_PieCtrl.c
- DSP2833x_PieVect.c
- DSP2833x_SysCtrl.c
- DSP2833x_usDelay.asm

Os arquivos ficarão assim:

- >  DSP2833x_ADC_cal.asm
- >  DSP2833x_Adc.c
- >  DSP2833x_CodeStartBranch.asm
- >  DSP2833x_CpuTimers.c
- >  DSP2833x_DefaultIsr.c
- >  DSP2833x_EPwm.c
- >  DSP2833x_GlobalVariableDefs.c
- >  DSP2833x_Gpio.c
- >  DSP2833x_PieCtrl.c
- >  DSP2833x_PieVect.c
- >  DSP2833x_SysCtrl.c
- >  DSP2833x_usDelay.asm

Com todos os arquivos importados, faça a build do projeto, clicando no ícone do martelo na barra superior do Code Composer Studio. Se você não teve nenhum erro de compilação ou de linkagem, você fez tudo corretamente.

Criação de um projeto

Agora que você já possui a biblioteca estática, vamos criar um projeto em banco.

Clique em **File→New→CCS Project**.

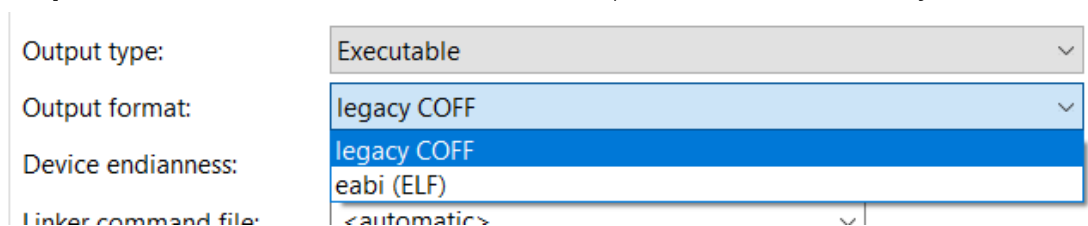
Em **Target**, selecione a placa **TMS320F28335**

Em **Project name**, dê um nome para seu projeto.

Expanda **Project type and tool-chain**.

Em **Output type**, selecione **Executable**.

Em **Output format**, selecione o mesmo formato que você definiu na criação da biblioteca.



Clique em **Finish**.

Com o botão direito na base do projeto, selecione **Properties**.

Clique em **Build**→**C2000 Compiler**→**Include Options**

Clique em **Add**, e coloque o seguinte caminho, trocando \$CW pelo caminho onde você instalou a biblioteca C2000Ware: "\$CW/device_support/f2833x/headers/include".

Repita o mesmo processo para "\$CW/device_support/f2833x/common/include".

Não adicione nenhum símbolo em **Build**→**C2000 Compiler**→**Predefined Symbols**.

Clique em **Build**→**C2000 Linker**→**File Search Path**, e em **Include library file or command file as input**:

- Clique em **Add**, e então em **Workspace**. Expanda o projeto da biblioteca estática, e então expanda **Debug**, e selecione o arquivo **.lib**.
- Adicione os seguintes caminhos:
 - "\$CW/device_support/f2833x/headers/cmd/DSP2833x_Headers_nonBIOS.cmd"
 - "\$CW/device_support/f2833x/common/cmd/28335_RAM_Ink.cmd"

Clique em **Apply and Close**.

Com isso, o projeto está configurado. Para criar um arquivo de código, clique com o botão direito na base do projeto e selecione **New**→**File** e nomeie como **main.c**, por exemplo. Os DSPs da Texas Instruments possuem suporte a códigos de exemplo que estão disponíveis em **"\$CW/device_support\f2833x\examples"**.