

**INTRODUÇÃO AO
CODEWARRIOR 10.5
PARA A DISCIPLINA EA871**

**Prof. Antonio A. F. Quevedo
FEEC / UNICAMP**

Revisado em fevereiro de 2014

INTRODUÇÃO

O CodeWarrior 10 (CW10) é o ambiente de desenvolvimento de *software* desenvolvido pela Freescale para os microcontroladores e microprocessadores por ela desenvolvidos. Ao contrário das versões anteriores, desenvolvidas de maneira totalmente proprietária, a versão 10 foi desenvolvida a partir do **Eclipse**.

Eclipse é um ambiente de desenvolvimento multi-linguagem que contém um *Integrated Development Environment* (IDE), e um sistema de *plug-ins*. Foi lançado nos termos da *Eclipse Public License*, assim o ambiente Eclipse é gratuito e de código aberto. Vem se tornando um padrão cada vez mais utilizado em empresas que desenvolvem projetos de *software*.

Os *plug-ins* garantem funcionalidade expansível ao sistema. Assim, pode-se adicionar novas linguagens de programação, bem como ferramentas de teste e depuração de código, integradas ao ambiente. Pode-se ainda adicionar funcionalidades relativas a famílias específicas de processadores / controladores. Por exemplo, quando a Freescale lança uma nova família de processadores, também lança um *plug-in* para o CW10, para que este possa gerar código objeto corretamente para aquela nova família. Outro exemplo é o *plug-in* de terminal serial, que permite a implementação de um terminal para comunicação serial dentro do ambiente, não sendo mais necessário o uso de outro programa para esta finalidade no computador *host*. Assim, quando se depura um programa que se comunica com um computador através de porta serial, antes era necessário ficar alternando entre as janelas de depuração do IDE e do terminal serial. Agora, as janelas de depuração e a do terminal ficam integradas no mesmo ambiente.

Um conceito fundamental do Eclipse é o *workspace*, que é um conjunto de metadados sobre um espaço de arquivos. Na prática, podemos ter *workspaces* diferentes no mesmo computador, e em cada *workspace* podemos ter múltiplos projetos. O sistema permite importar e exportar projetos individuais ou *workspaces* completos. Neste curso, podemos definir um *workspace* para cada dupla de alunos, os quais podem colocar todos seus projetos dentro do mesmo. Assim, na mesma máquina podem conviver espaços de trabalho distintos.

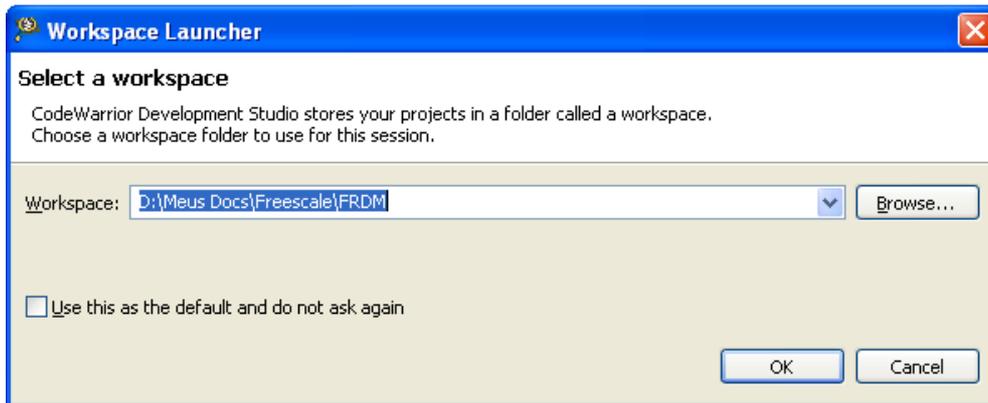
Outro conceito fundamental é a **perspectiva**. Esta pode ser definida como um conjunto de janelas que povoam seu ambiente de trabalho, conjunto este definido para melhor utilização do sistema, de acordo com o que está sendo feito naquele momento. Neste curso, vamos usar 2 perspectivas básicas: **Programação** e **Depuração**. Na perspectiva de programação, temos janelas para edição de código, árvore de arquivos do projeto, localização de módulos de código, recepção de mensagens de compilação, etc. Na perspectiva de depuração, temos janelas para visualização de ponto de execução, controles para execução passo-a-passo do código, janelas para apresentação de variáveis em tempo real, código *assembly* gerado a partir do código fonte, janela do terminal serial, etc.

Estes conceitos se desenvolverão melhor com o uso da ferramenta. Seguindo o passo-a-passo das próximas páginas, certamente os alunos estarão usando adequadamente a ferramenta em um curto espaço de tempo.

Obs: A aparência de algumas janelas pode ser ligeiramente diferente em alguns computadores, mas as diferenças não são suficientes para prejudicar este tutorial.

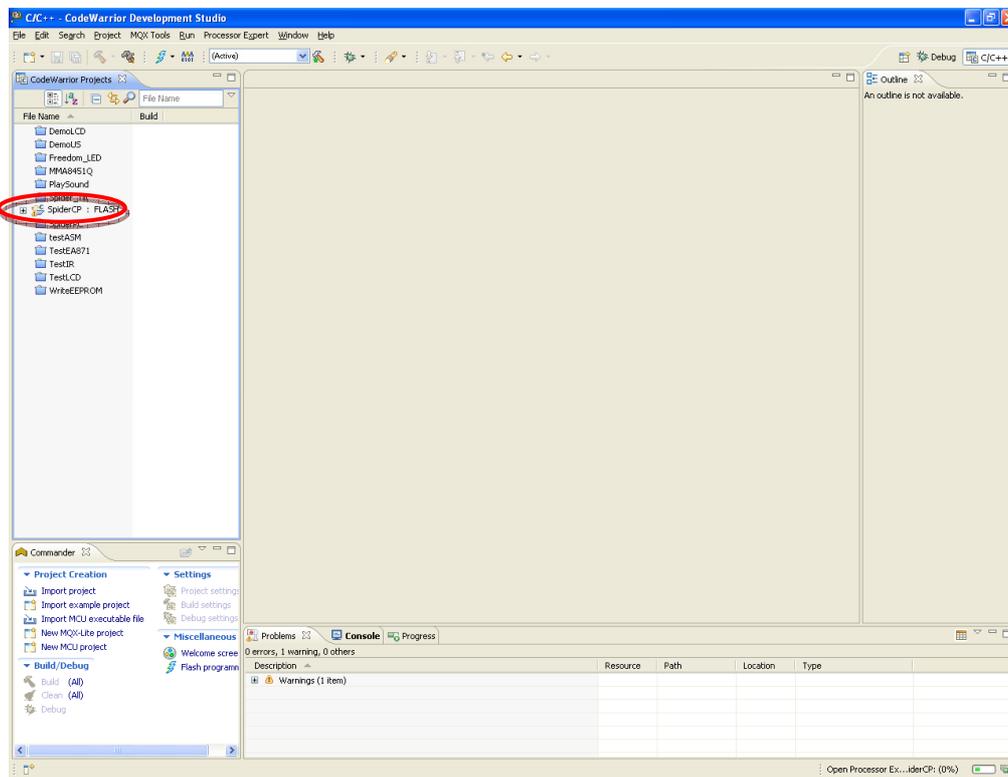
CRIANDO UM NOVO PROJETO

1. No Windows, vá ao menu Iniciar - Programas - Freescale CodeWarrior - CW for MCU v10.5 – CodeWarrior, ou chame o programa através do ícone do CodeWarrior 10.5 na área de trabalho.
2. Aparece o *Workspace Launcher*. Aqui você determina qual *workspace* você deseja utilizar. Clique em *Browse* para escolher uma pasta para ser seu *workspace* ou selecione uma já existente na lista *drop-down*. Clique em OK. Sugerimos que cada grupo crie seu próprio *workspace*.

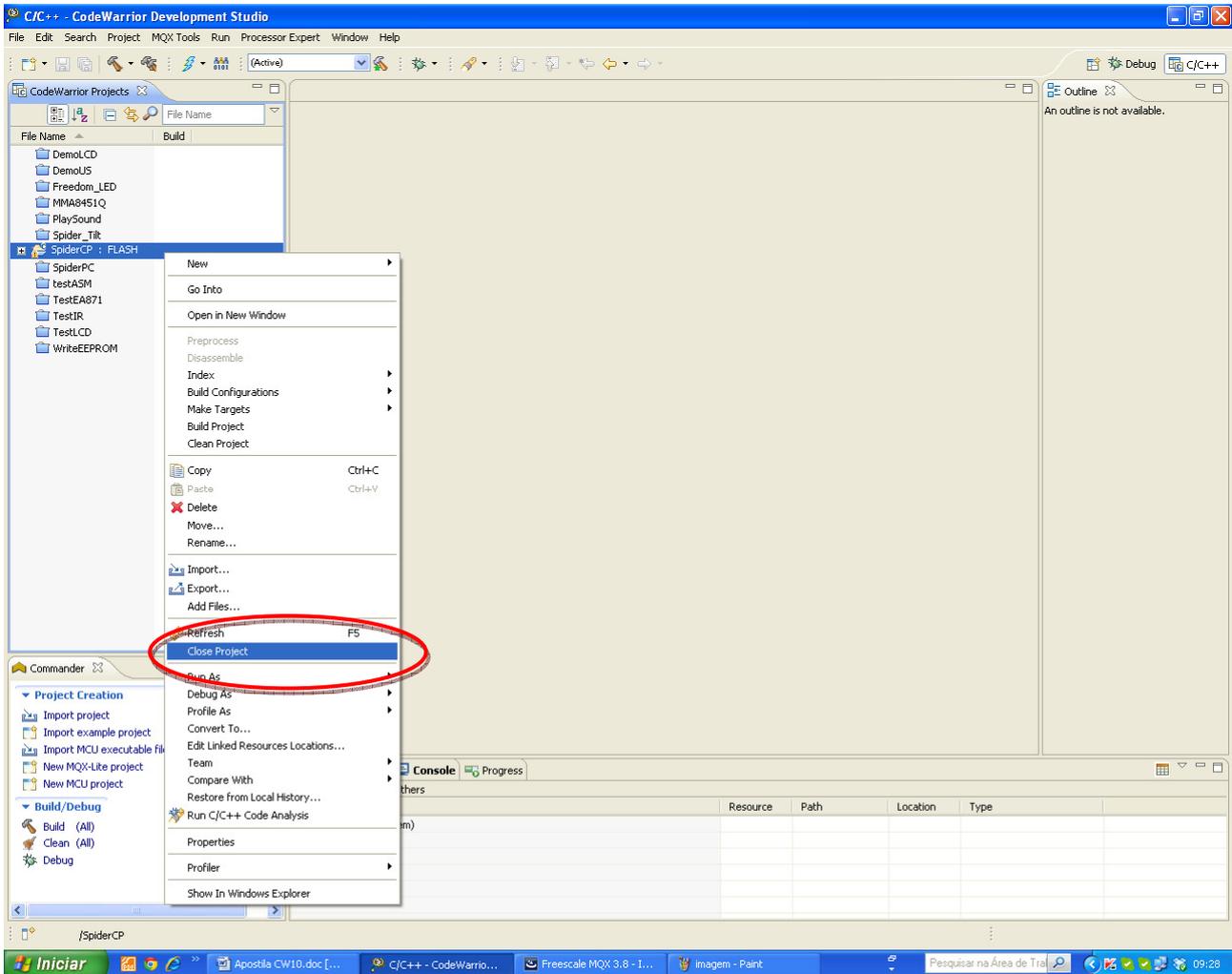


3. O CodeWarrior 10 vai abrir na perspectiva de programação. Pode-se ver a esquerda um painel com todos os projetos do *workspace*. Uma pasta aberta significa que aquele projeto está aberto.

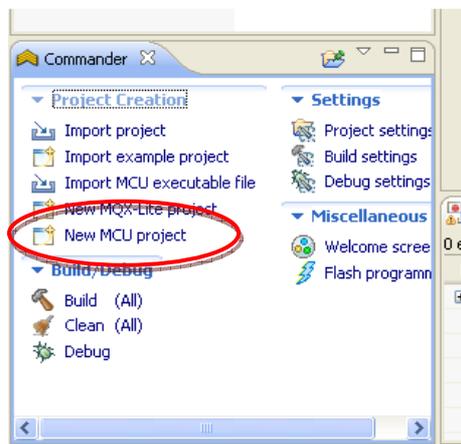
Na primeira utilização do *workspace*, ao invés da perspectiva de programação, pode aparecer um menu geral. Neste caso, basta escolher a opção “Go to Workspace”.



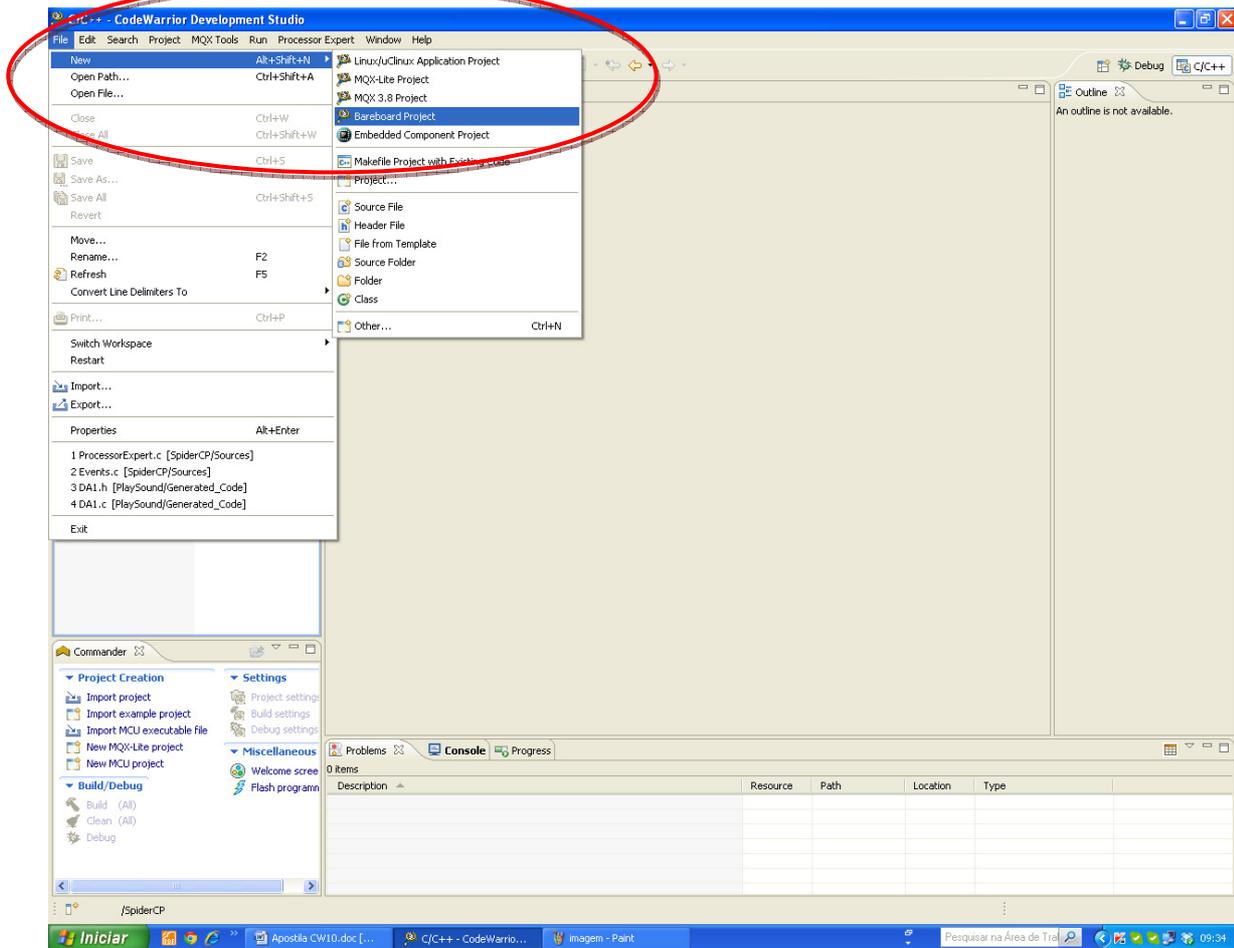
4. Para evitar problemas, evite ter mais de um projeto aberto de cada vez. Se houver diversos projetos abertos, feche os desnecessários clicando com o botão direito sobre a pasta do projeto aberto e escolhendo a opção "Close Project".



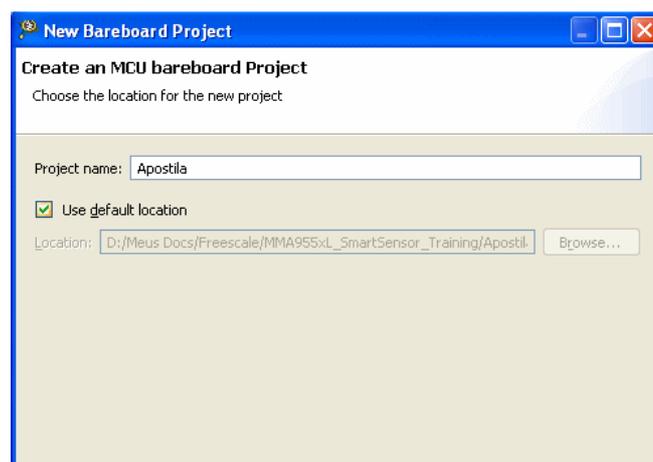
5. A janela *Commander* no canto inferior esquerdo da perspectiva agrupa os comandos mais comuns. Para criar um novo projeto, basta clicar em *New MCU project* nesta janela.



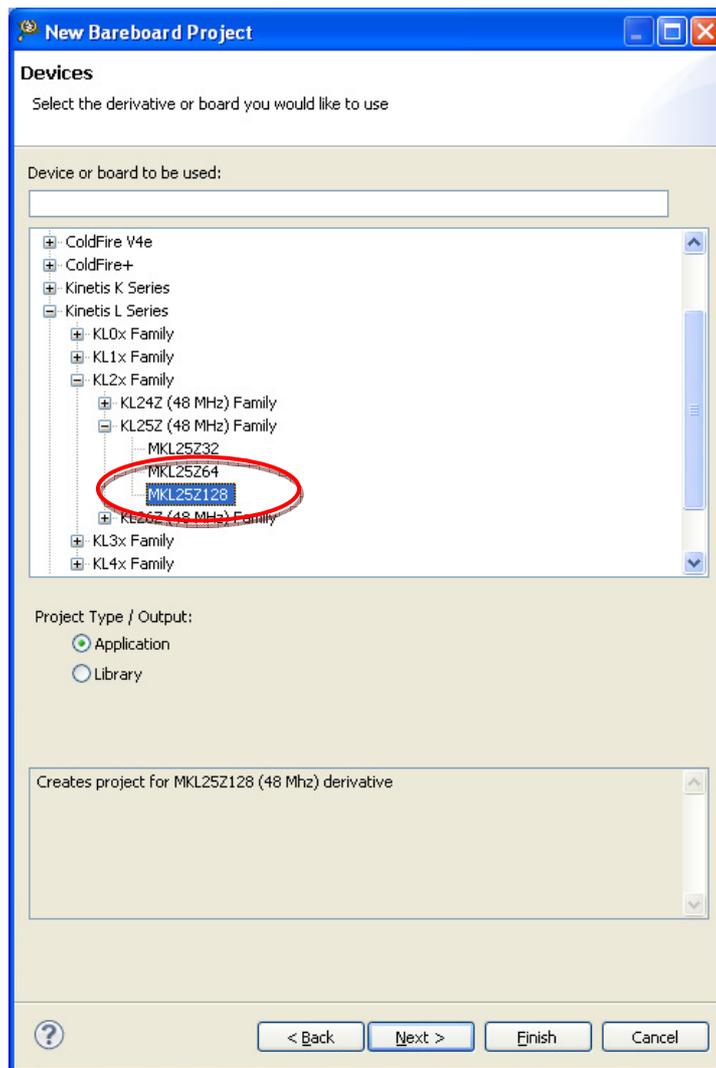
Alternativamente, no menu superior selecione *File - New - Bareboard Project*.



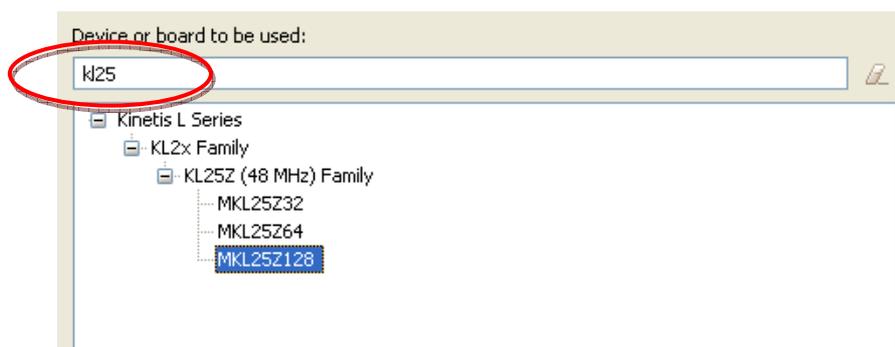
6. Na janela que se abre, escolha um nome para o projeto. No exemplo, foi escolhido o nome "Apostila". Se a caixa de seleção "Use default location" estiver selecionada, os arquivos serão salvos na pasta do *workspace*, que pode ser vista logo abaixo da caixa. Desmarcando a caixa, pode-se escolher outra pasta para colocar os arquivos. Depois, clique no botão "Next".



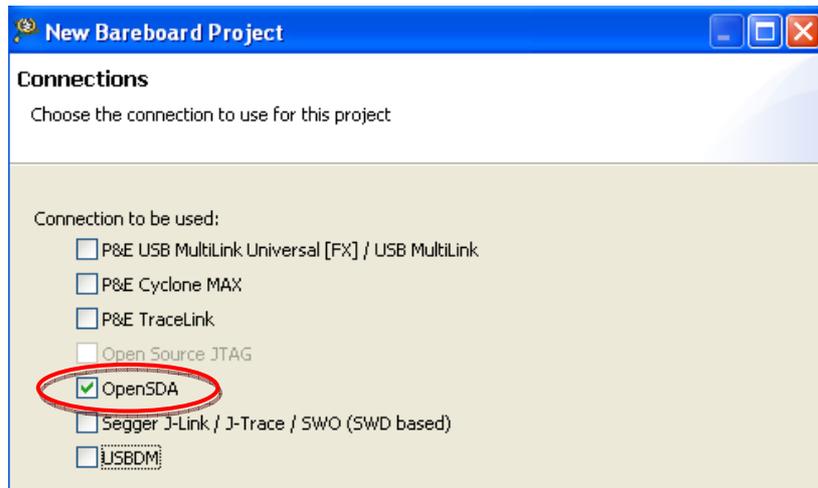
7. Agora deve-se escolher o microcontrolador ou a placa de desenvolvimento a ser utilizada. Neste curso usamos o controlador MKL25Z128, na placa FRDM-KL25. Na árvore de opções, abra o grupo "Kinetic L Series", depois o grupo "KL2x Family", depois "KL25Z (48MHz) Family" e selecione "MKL25Z128" Clique "Next".



Alternativamente, pode-se digitar “KL25” na caixa de pesquisa (intitulada “Device or board to be used”). Neste caso, o programa filtra as opções, apresentando apenas a sub-árvore dos processadores KL25, e assim pode-se selecionar mais facilmente o processador-alvo.

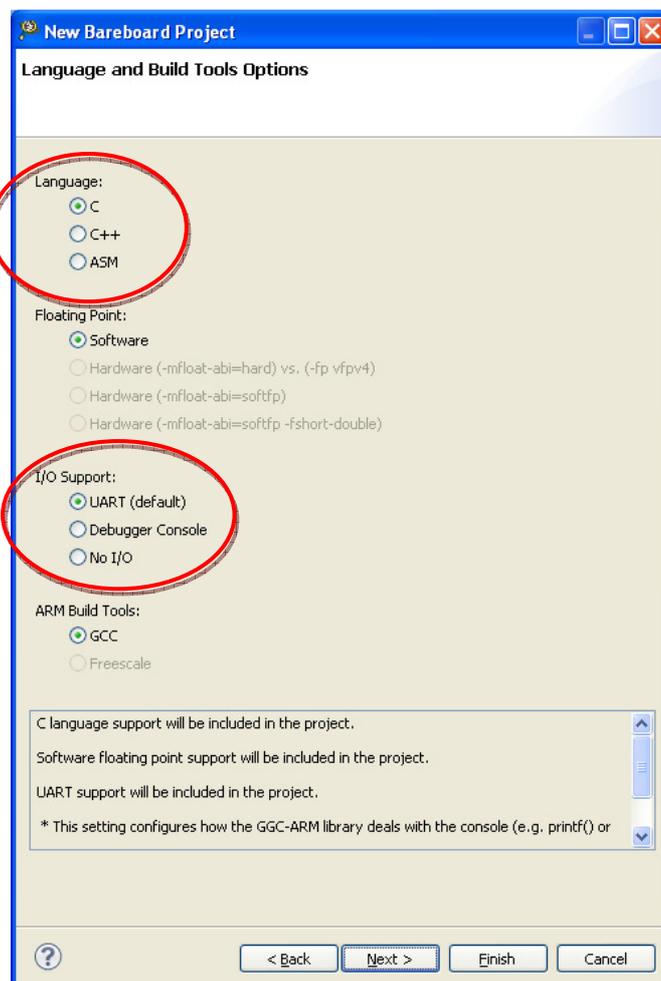


8. Agora, aparece uma lista de possíveis conexões. A conexão estabelecida entre o computador e a placa permite transferir os programas para a placa, bem como executar a depuração em tempo real. Neste caso, selecione a opção "OpenSDA", que é o *hardware* de conexão existente na placa de desenvolvimento. Desmarque quaisquer outras opções selecionadas por padrão, e depois clique em "Next".

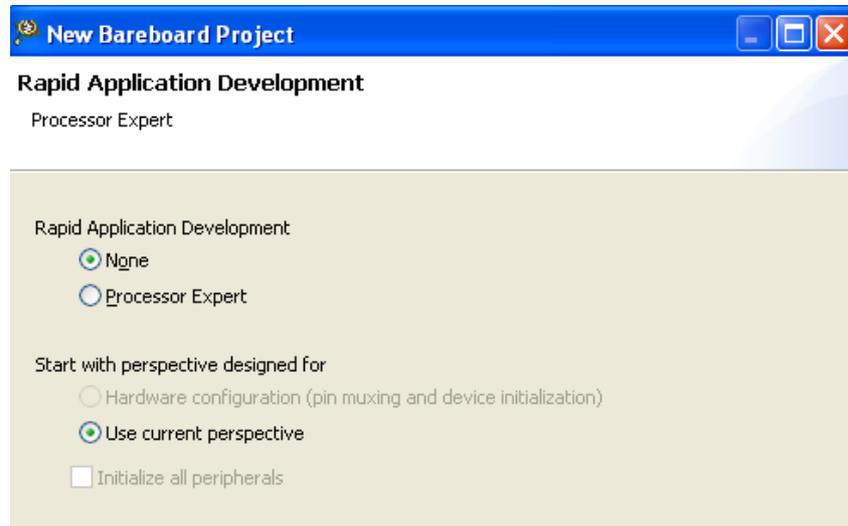


9. Na próxima janela, mantenha as opções no padrão e clique em "Next". Estas opções se referem à linguagem utilizada, uso de ponto flutuante, suporte a uso de porta serial ou console de debug, e compilador a ser usado.

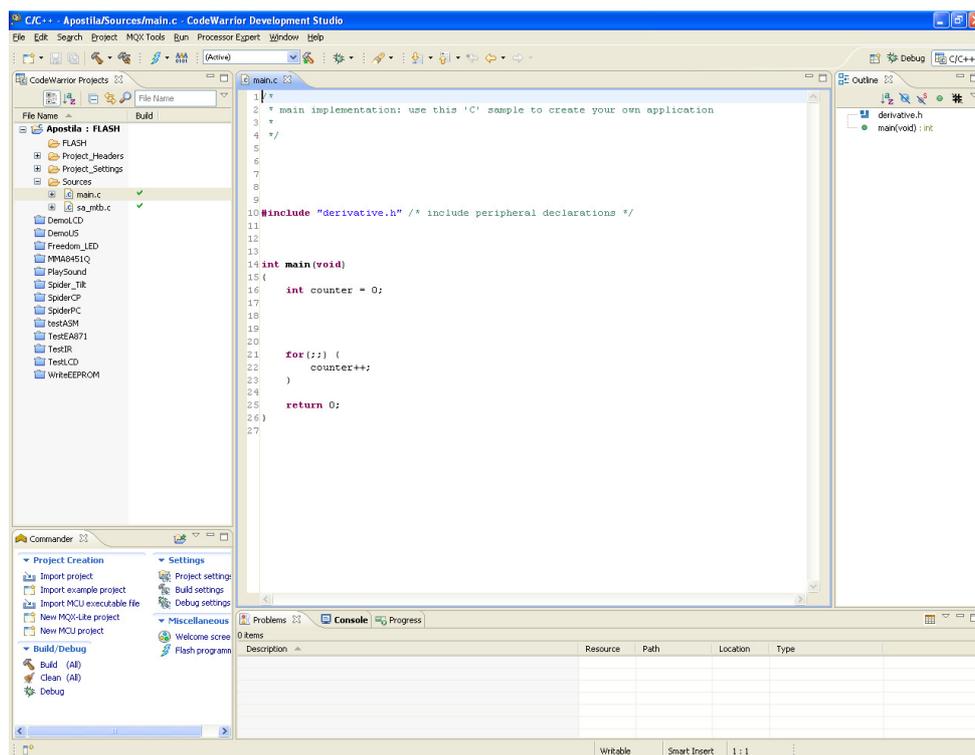
Obs: Na opção “I/O Support”, a opção *default* (UART) e a opção *Debugger Console* acrescentam código para suporte a funções *printf* e *scanf* através da porta serial ou de uma janela de console. Quando estas funções não são necessárias, pode-se marcar a opção “No I/O” para reduzir o tamanho do código final e se evitar inicializações desnecessárias. Sugerimos usar a terceira opção, a menos que o experimento use as funções acima citadas (não é o caso no curso EA871).



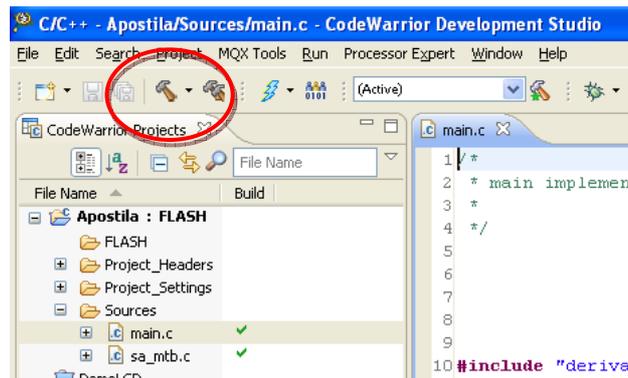
10. Nesta próxima janela, definiremos se o CodeWarrior usará ou não uma ferramenta que facilita a geração de códigos para a inicialização dos módulos da MCU e funções em alto nível para controlar estes módulos (*Processor Expert*). Neste curso, esta ferramenta não será utilizada, e a mesma só está disponível se a linguagem utilizada for o C (quando se usa *assembly*, esta opção está indisponível). Mantenha a opção "None" selecionada e clique em "Finish".



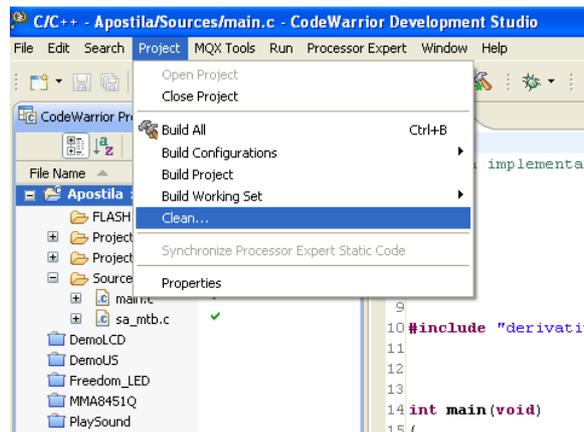
11. Veja que agora na lista de projetos, temos o projeto "Apostila" criado e aberto. Clicando no símbolo de "+" (ou no pequeno triângulo se estiver usando Windows 7/8) à esquerda do folder, podemos expandir o projeto e ver seus componentes pré-criados pelo CodeWarrior. Dentro do folder "Sources", encontraremos o arquivo "main.c" (em projetos *assembly*, o arquivo será o "main.s"). Dando um duplo-clique neste arquivo, abrimos o editor e podemos ver uma estrutura básica de código já feita pelo CodeWarrior, onde se inicializa uma variável inteira com o valor 0 e a incrementa a cada iteração de um *loop* infinito. É neste arquivo onde colocaremos os programas dos roteiros. Para este teste, manteremos o programa como está.



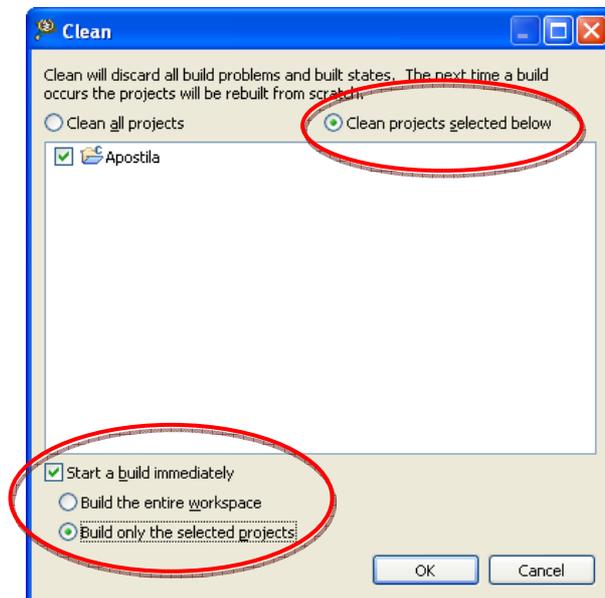
12. Agora precisamos compilar o projeto. Para isto, podemos clicar no botão da barra superior com o ícone do martelo, ou ainda no menu selecionar "Project - Build Project".



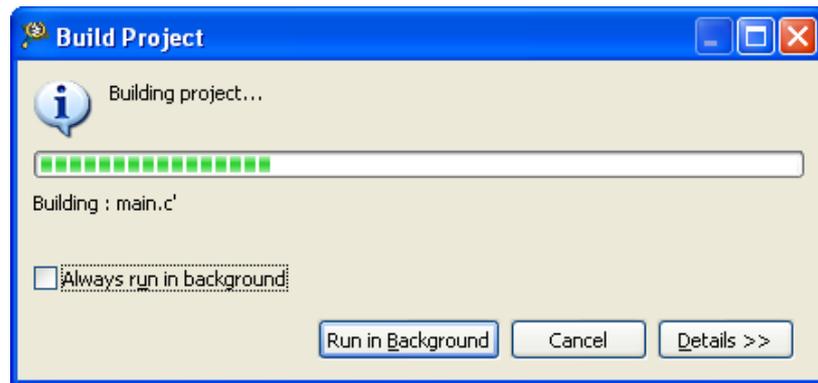
Entretanto, é uma boa prática garantir sempre que o projeto esteja “limpo” antes de compilá-lo. Isto significa apagar arquivos gerados por compilações anteriores. Assim, pode-se selecionar no menu principal a opção “Project – Clean”.



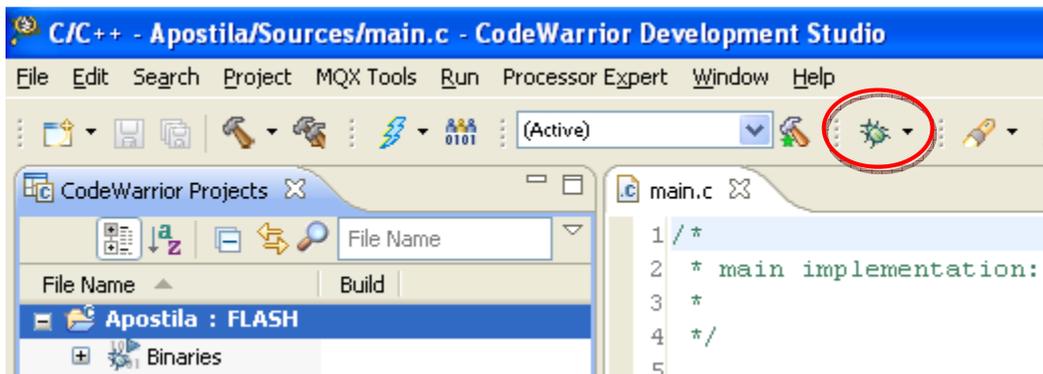
13. Na janela que se abre, selecione a opção *Clean projects selected below* e verifique se seu projeto está selecionado na lista. Marque a caixa *Start a build immediately* e selecione a opção *Build only the selected projects*. Desta forma, o projeto ativo será limpo e logo após será iniciado o *build* do mesmo.



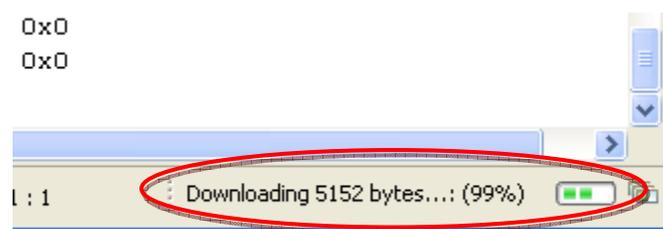
14. Teremos uma janela mostrando as etapas do clean, que costumam ser rápidas. Logo após, uma janela mostrando os passos do *Build* aparecerá. Aguarde até que ela desapareça.



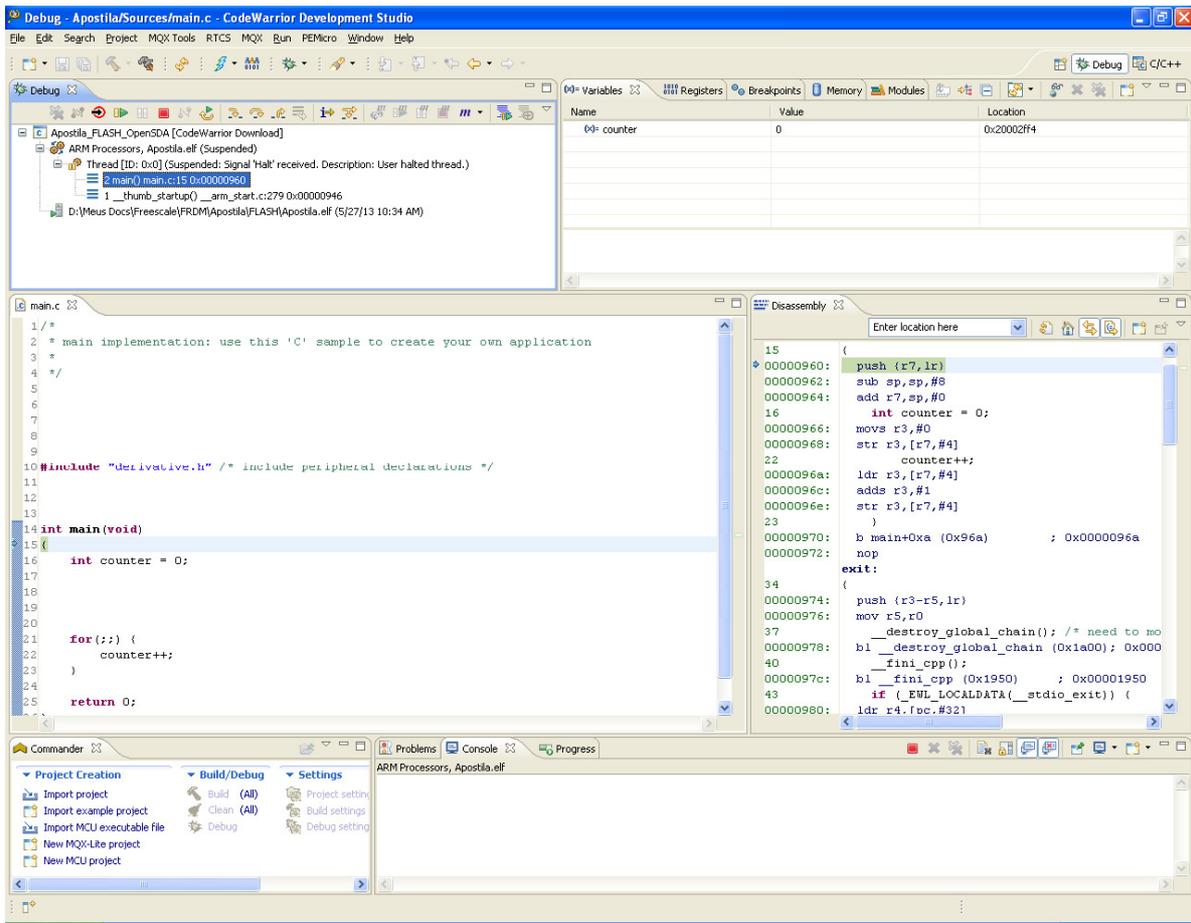
15. Agora vamos acessar a placa em modo *debug* (depuração), fazer a transferência do programa e passar para a perspectiva de depuração. Para isso, basta clicar no botão da barra superior com o ícone do besouro, ou no menu principal selecionar "Run - Debug", ou ainda pressionar a tecla F11. Certifique-se que o projeto "Apostila" esteja selecionado na janela de projetos.



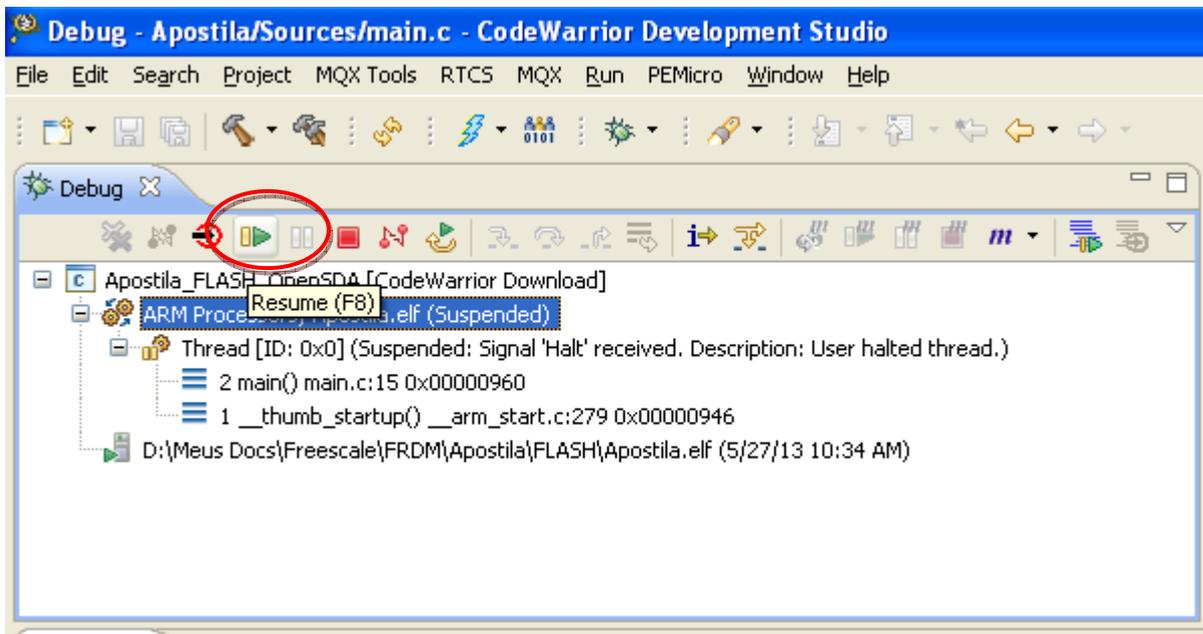
16. A perspectiva de projeto muda para *Debug*. Aguarde a carga do código (isto pode ser visto no canto inferior direito).



17. Ao acabar, veremos na janela superior esquerda (*Debug*), que a MCU está parada no início do "main" (selecionado na janela). Abaixo desta janela, aparece o código fonte do "main". Ao lado deste, o resultado da compilação em *assembly*.

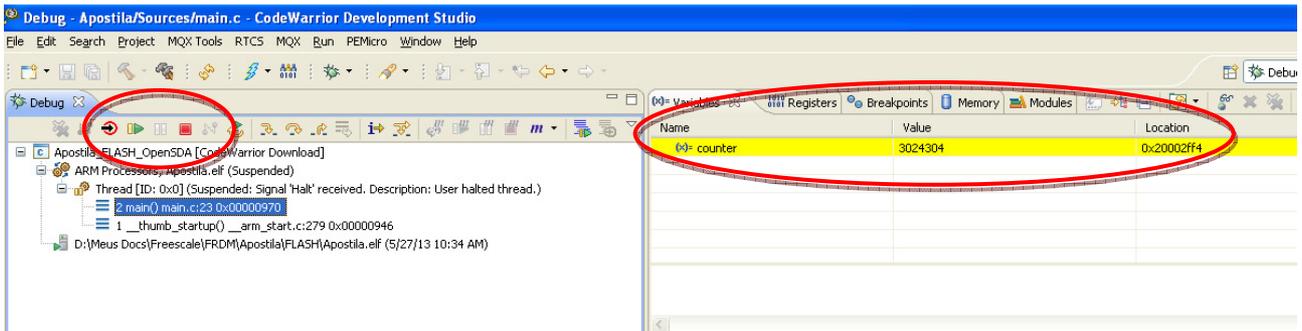


18. Nesta mesma janela, clique no triângulo verde (*Resume*), ou pressione F8 para iniciar a execução.

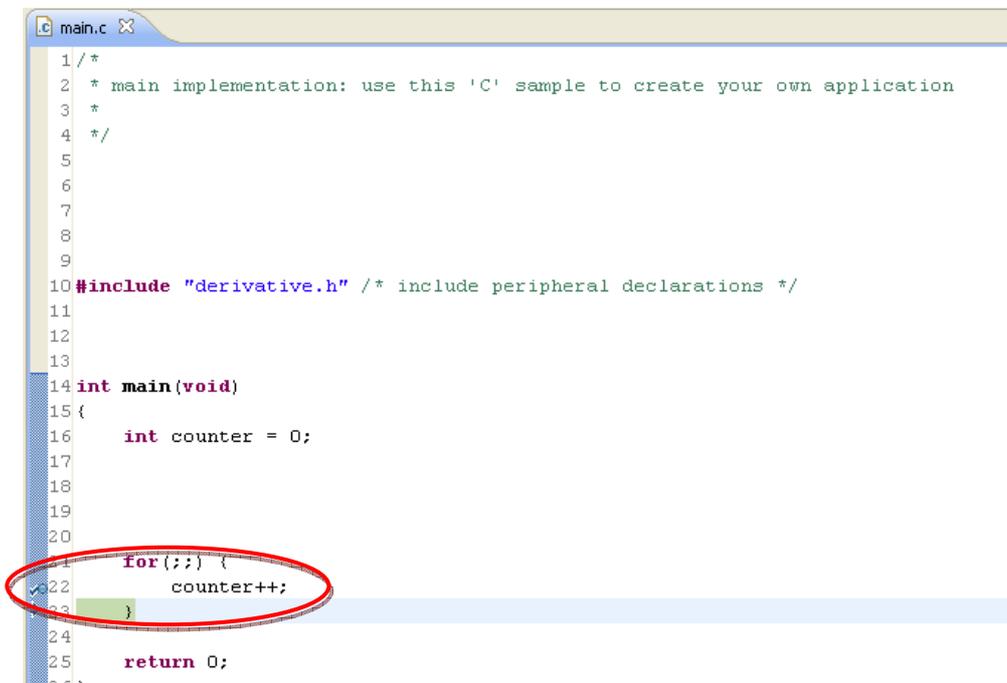


19. Junto ao botão de "Resume", há ainda os botões "Reset" (círculo vermelho com seta preta), "Suspend" (símbolo de pausa), "Terminate" (quadrado vermelho) e "Disconnect" (uma linha em forma de "N"). Estes botões permitem respectivamente colocar a MCU em seu estado inicial (aguardando o "Resume"), pausar a execução (também aguardando o "Resume"), interromper

definitivamente a execução, até que se faça um "Reset", e continuar executando o programa sem a conexão ao IDE. Clicando no botão "Suspend", o programa é interrompido e na aba "Variables", pode-se visualizar o valor da variável "counter". Se o programa voltar a ser executado, e novamente pausado, pode-se ver outro valor para a variável, mostrando que ela vai sendo incrementada a cada iteração do "loop". O valor da variável pode ser modificado durante a depuração, se a execução estiver pausada. Basta clicar sobre o valor da variável e reescrevê-lo. Clicando com o botão direito sobre a variável ou seu valor, pode-se escolher na opção *Format* o formato do dado: decimal, hexadecimal, binário, etc. Clicando com o botão direito em uma área livre do quadro, podemos adicionar variáveis que eventualmente não estejam aparecendo.

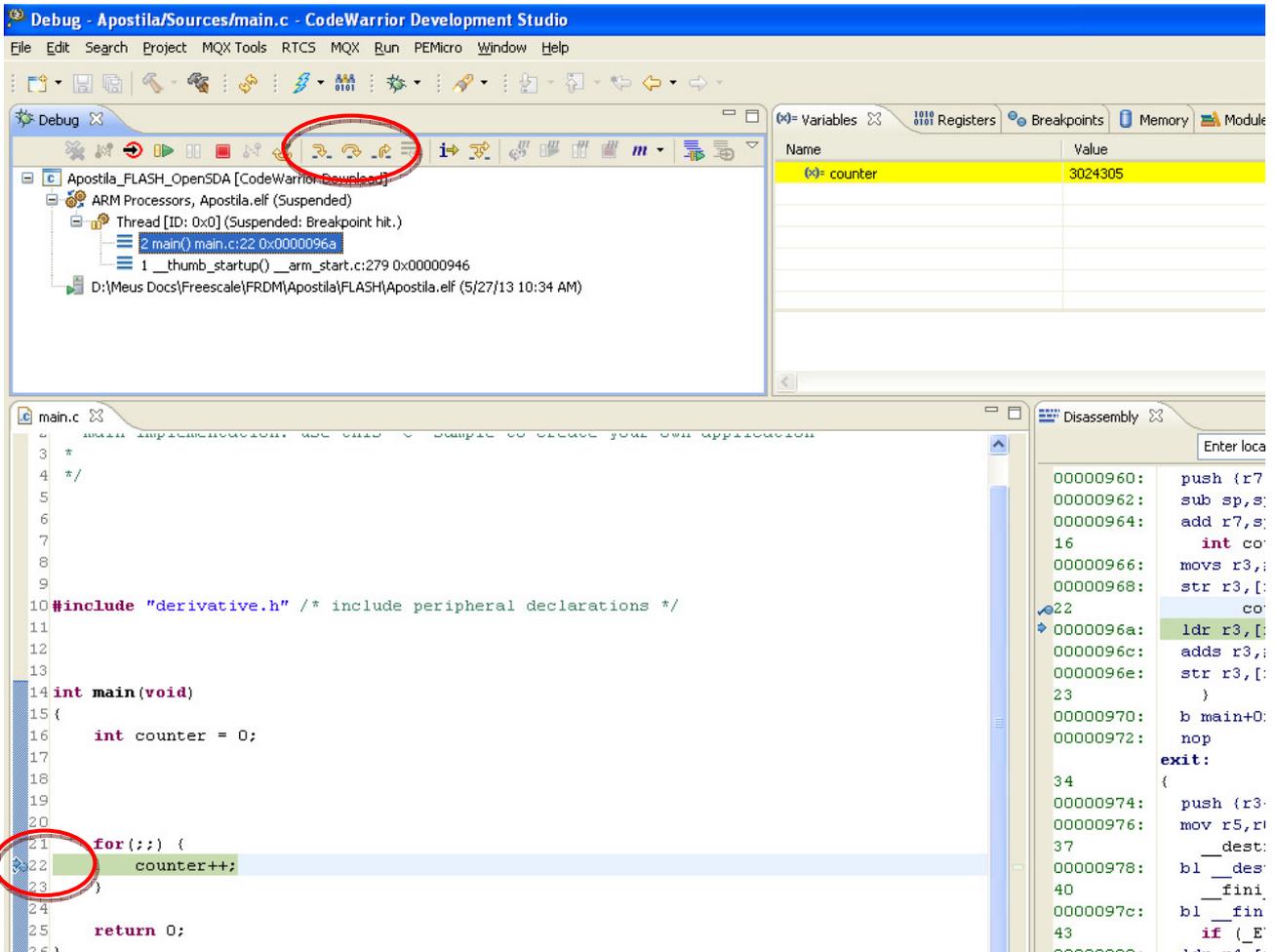


20. Pode-se estabelecer *breakpoints* no código, para que o mesmo seja interrompido toda vez que sua execução chegar a um determinado ponto. À esquerda da linha "counter++;", sobre a área azulada, dê um duplo-clique. Aparecerá um pequeno círculo azul. Agora toda vez que o programa chegar neste ponto, será interrompido, podendo ser retomado através do botão "resume".

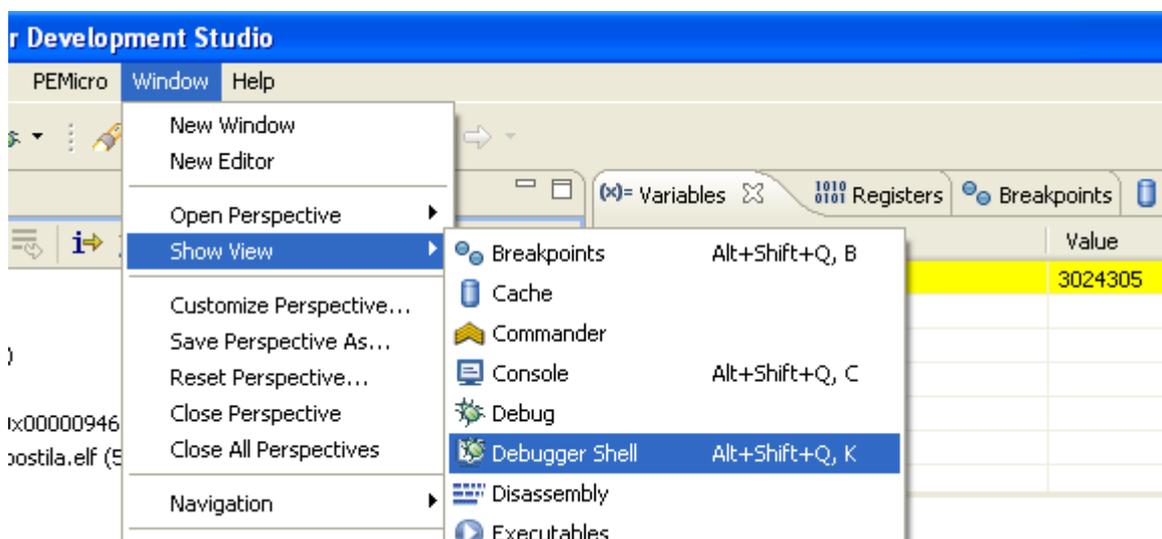


21. Continue a execução do programa com o botão "Resume". O mesmo será interrompido, e é possível ver uma seta azul à esquerda da linha onde o *breakpoint* foi colocado, indicando o ponto no qual o programa está parado. Próximo aos botões de "Resume", "Suspend" e os demais, existem botões que permitem executar "steps", ou seja, instrução por instrução. Use o botão "step into" para avançar uma instrução de cada vez. Pode-se ver o *loop* acontecendo, sendo que o programa passa

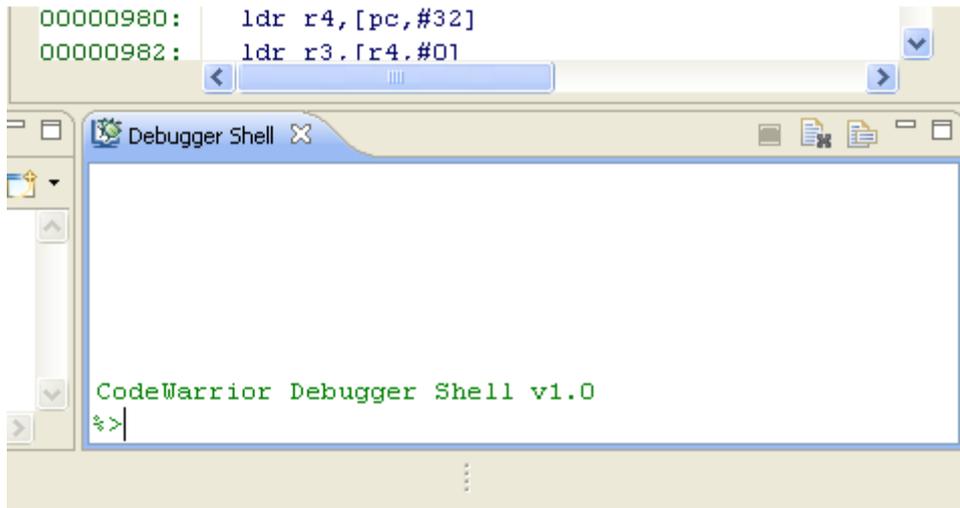
pelas instruções "for" e "counter++". O valor do contador vai sendo incrementado de um em um, cada vez que a execução passa por "counter++". Se o botão "Resume" for pressionado, a execução normal é retomada, mas para logo depois, quando a execução novamente atinge o *breakpoint*.



22. Há um recurso muito útil em determinadas situações, que é o *Debugger Shell*. Este recurso é uma janela de texto, que permite usarmos comandos em linha de texto para modificar posições de memória e registradores, além de outras funções de depuração. Para que esta janela apareça, na perspectiva *Debug*, vá ao menu superior e clique em “Window – Show View – Debugger Shell”.

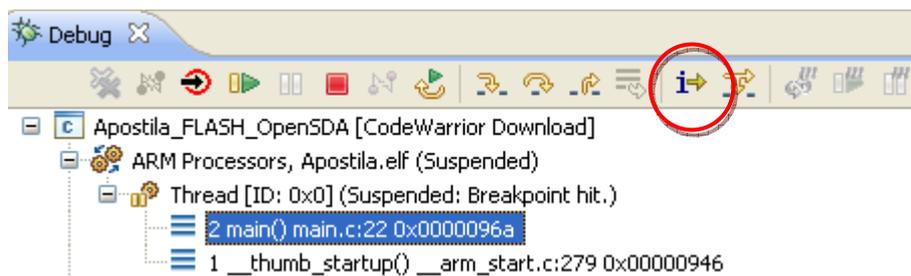


23. No canto inferior direito, a aba *Debugger Shell* irá aparecer.

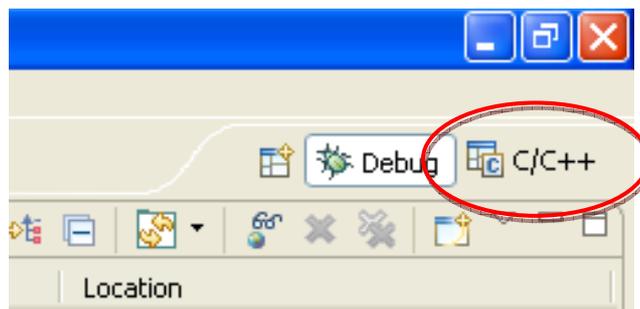


24. Digitando “help”, pode-se ver a lista de comandos possíveis. Digitando “help <comando>”, pode-se ver a forma de utilização para cada comando. Para cada janela, existem botões para maximizar, minimizar, restaurar tamanho original e fechar. As divisões entre janelas também podem ser deslocadas, modificando-se assim os tamanhos das diversas janelas.

25. Outro recurso útil é o *Instruction Stepping Mode*. Neste modo de operação, cada *step* de execução avança uma instrução de máquina (estas instruções são visíveis na janela *Disassembly*, à direita). Para entrar ou sair deste modo, clique no botão correspondente na janela de *Debug*.

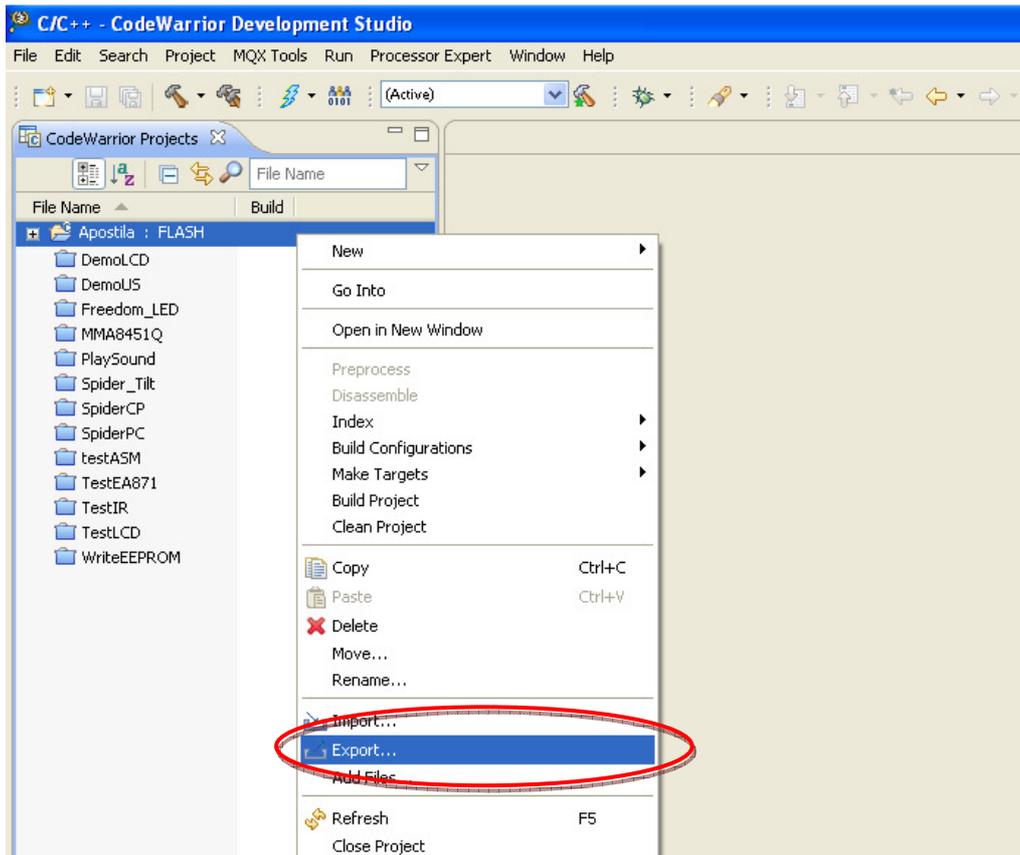


26. Para retornar à perspectiva de programação, clique no botão "C/C++" no canto superior direito da janela.

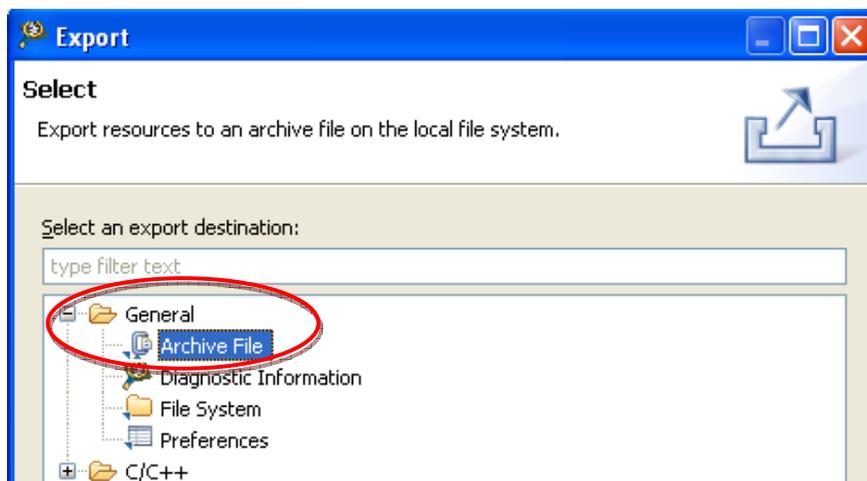


EXPORTANDO / IMPORTANDO UM PROJETO

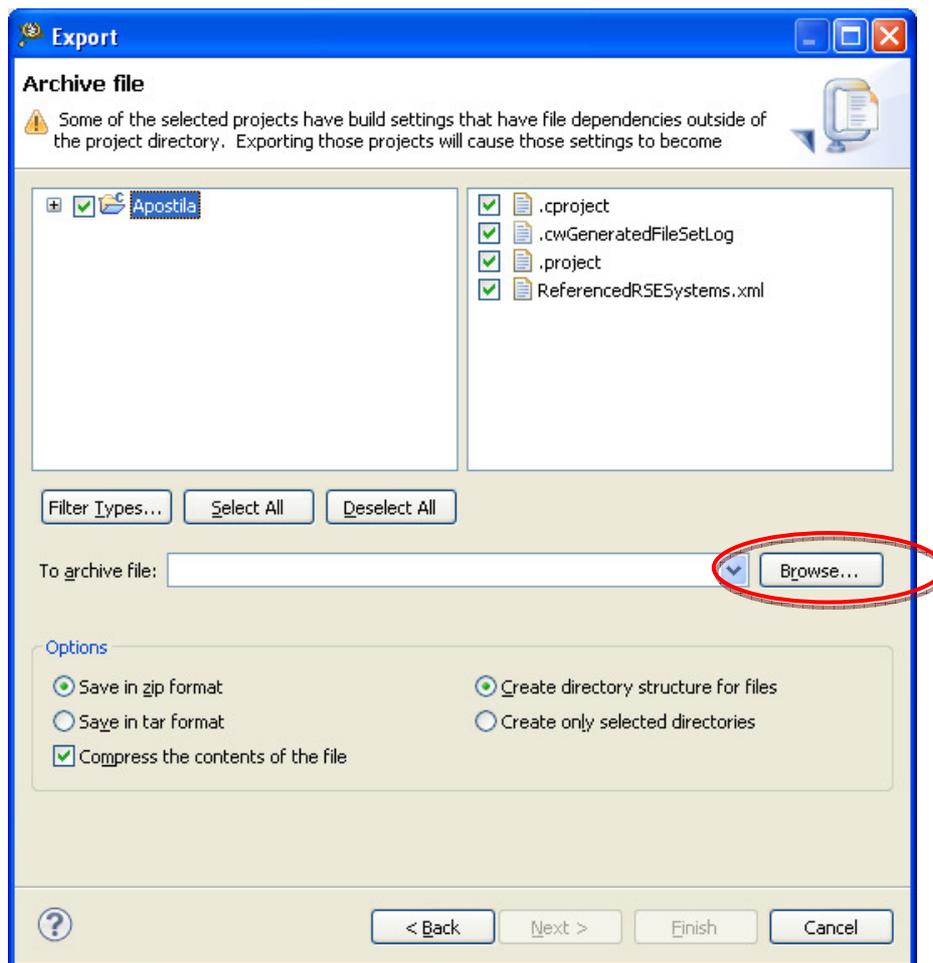
1. Na perspectiva de programação, feche todos os projetos abertos (conforme visto anteriormente), exceto aquele que se deseja exportar. Se o projeto desejado está fechado, abra-o, clicando sobre a pasta dele com o botão direito e selecionando a opção "Open Project".
2. Clique novamente na pasta do projeto e selecione "Export...". Alternativamente, pode-se usar o menu "File", opção "Export...".



3. Uma janela se abrirá. Abra o grupo "General" e selecione "Archive File". Depois, clique em "Next".

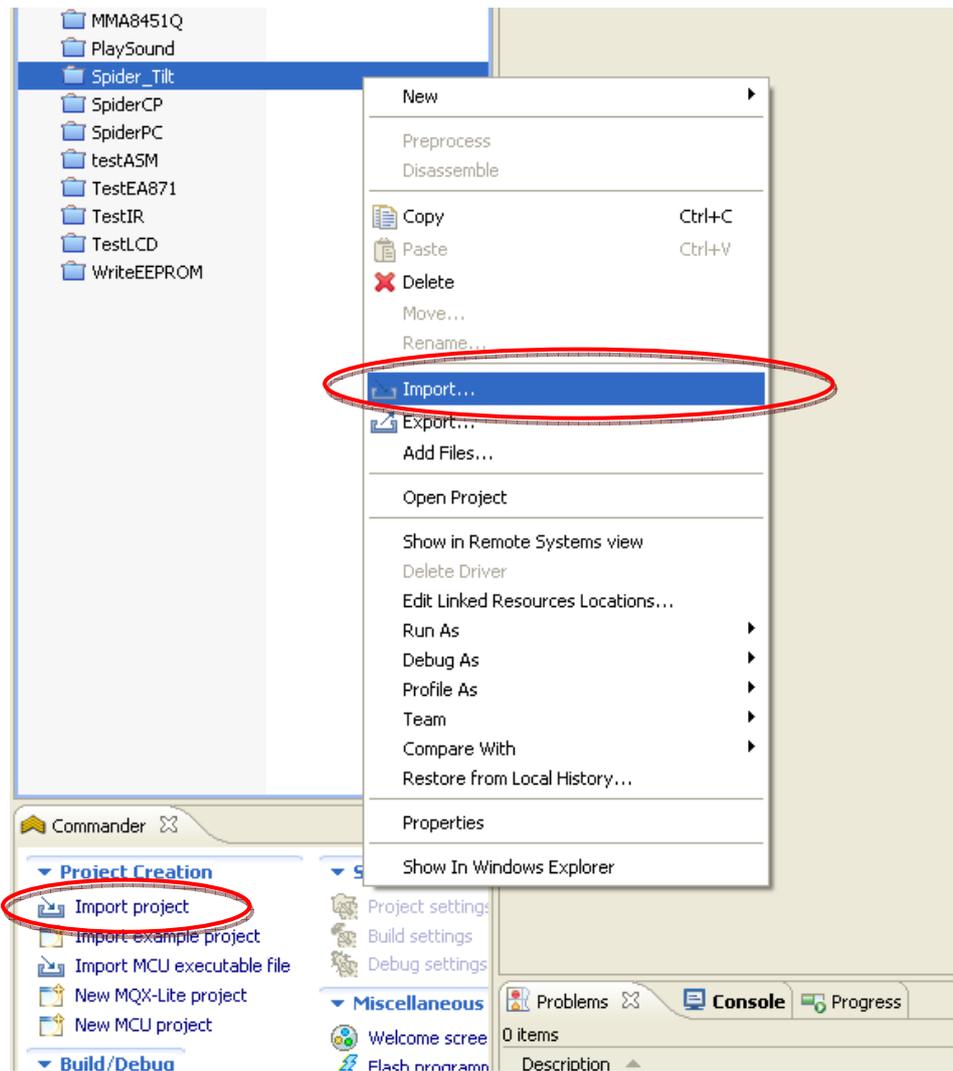


4. Nesta janela, mantenha as opções padrão, e clique no botão "Browse...". Selecione uma pasta para guardar o arquivo de exportação, e o nome do arquivo. Clique em "Finish".

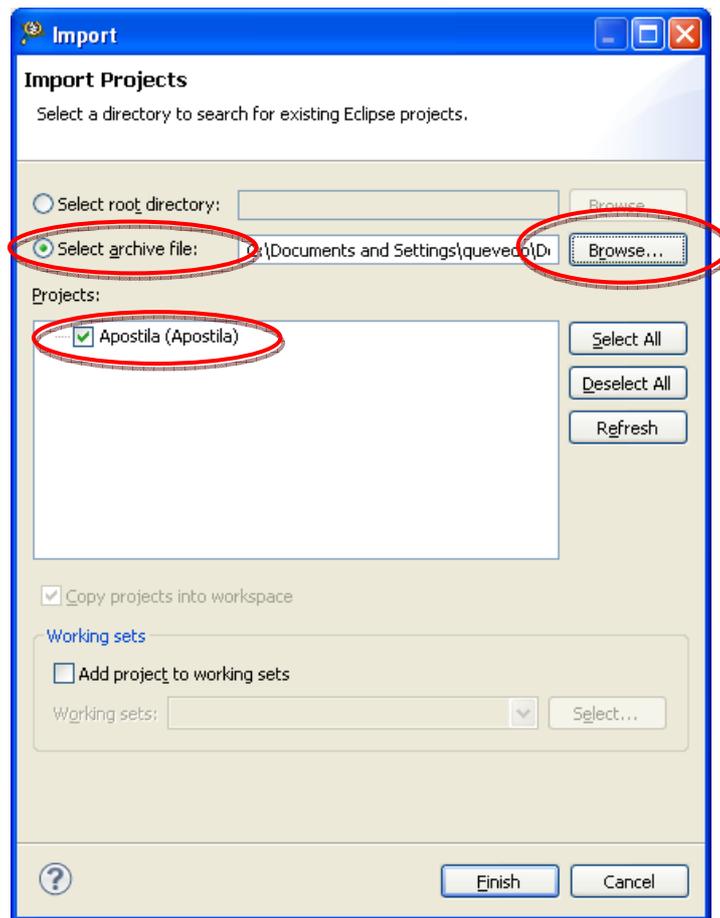


5. Um arquivo ZIP com o nome escolhido será gravado no local selecionado. Para fazer a importação, apague totalmente o projeto, inclusive os arquivos originais (conforme a seção "Apagando um projeto" abaixo). Se um projeto com o mesmo nome de outro projeto no mesmo *workspace* tentar ser importado, aparecerá uma mensagem de erro.

6. Para importar, clique com o botão direito na área dos projetos e selecione "Import...", ou selecione "Import Project" na janela "Commander", ou ainda use o menu "File - Import..."



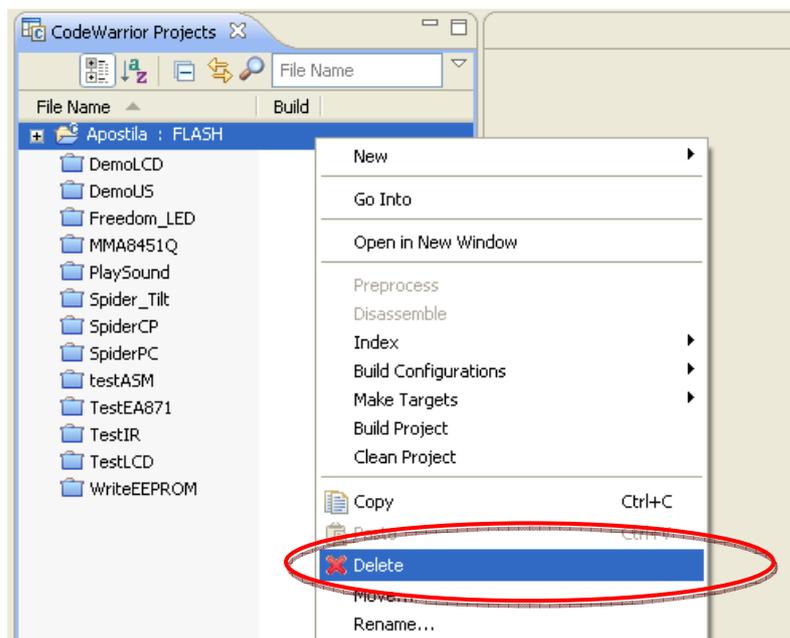
7. Marque a opção "Select archive file" e clique no botão "Browse". Localize e selecione o arquivo ZIP que contém o projeto exportado anteriormente. Confirme que o projeto está selecionado e clique em "Finish".



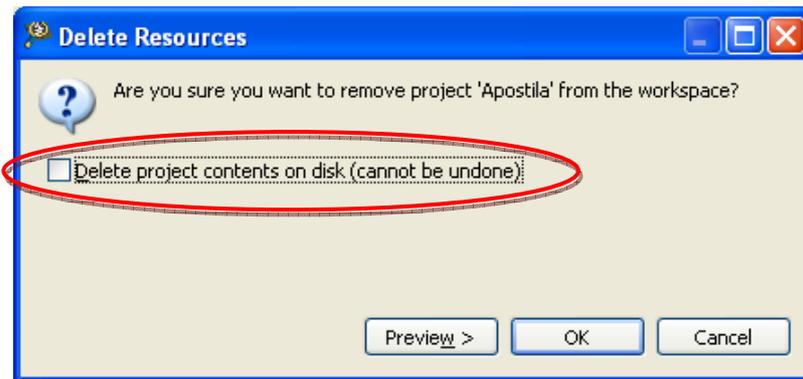
8. Pronto, seu projeto aparece agora no *workspace*.

APAGANDO UM PROJETO

1. Clique com o botão direito no projeto e selecione a opção "Delete". Alternativamente, selecione o projeto com um clique do botão esquerdo, e pressione a tecla "Delete".



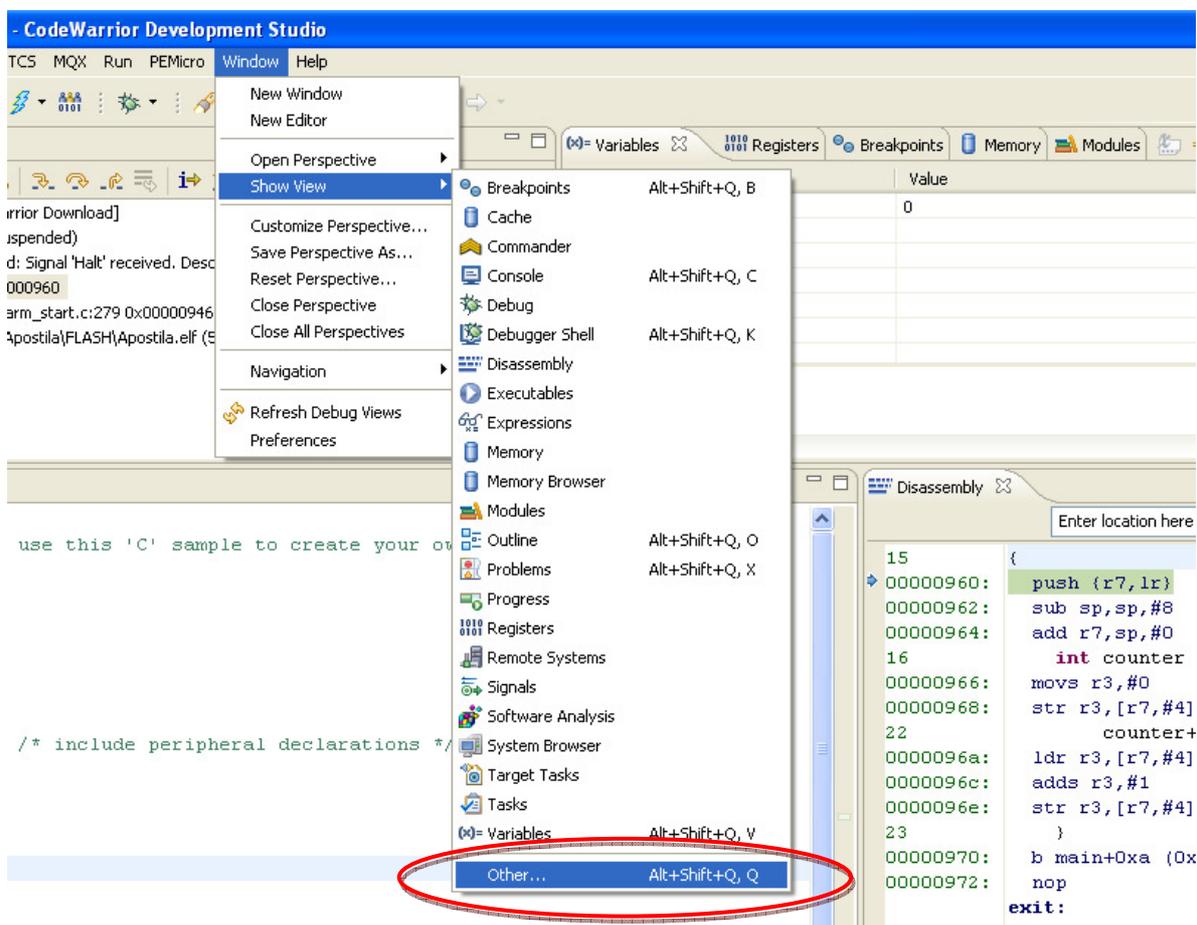
3. Uma nova janela se abrirá para confirmar o apagamento ou cancelá-lo. Por *default*, os arquivos não serão apagados. O projeto apenas sairá do *workspace*, mas continuará fisicamente na mesma pasta. Se quiser realmente apagar os arquivos do projeto, marque a caixa "Delete project contents on disk (cannot be undone)". Use esta opção com cuidado.



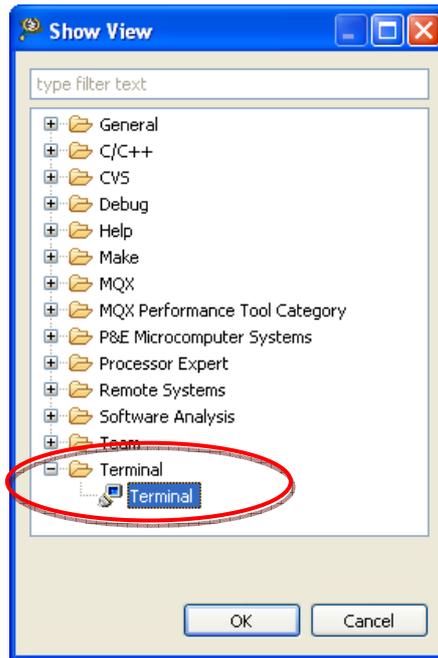
4. Clique em OK. Seu projeto foi apagado.

USANDO O TERMINAL SERIAL

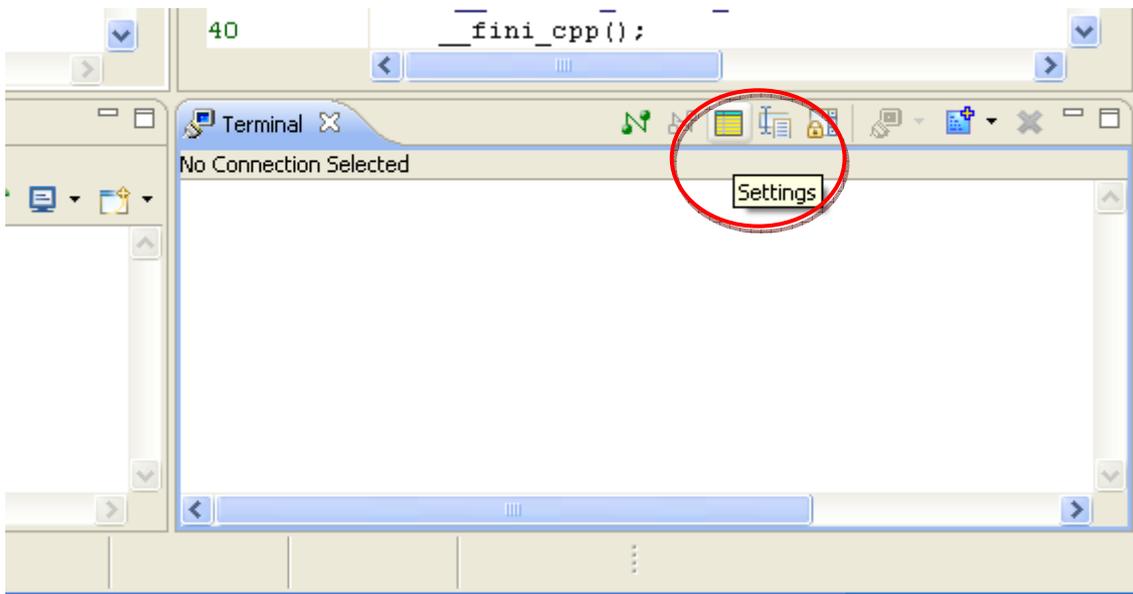
1. Para acessar o terminal serial, na perspectiva de Debug vá ao menu principal e selecione Window – Show View – Other...



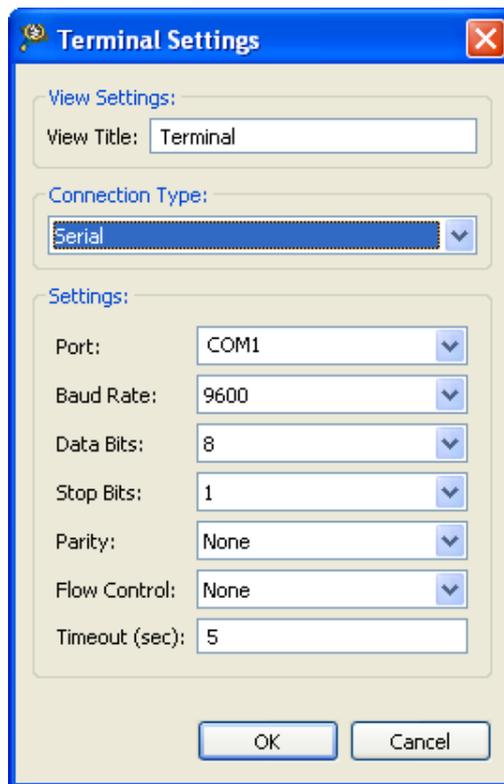
2. Na janela que se abre, expanda o grupo *Terminal* e selecione a opção *Terminal*. Clique OK.



3. Veja que na parte inferior direita da perspectiva, aparece uma nova aba chamada *Terminal*. Agora só falta configurar este terminal. Clique no botão com o símbolo de uma planilha amarela (*Settings*).



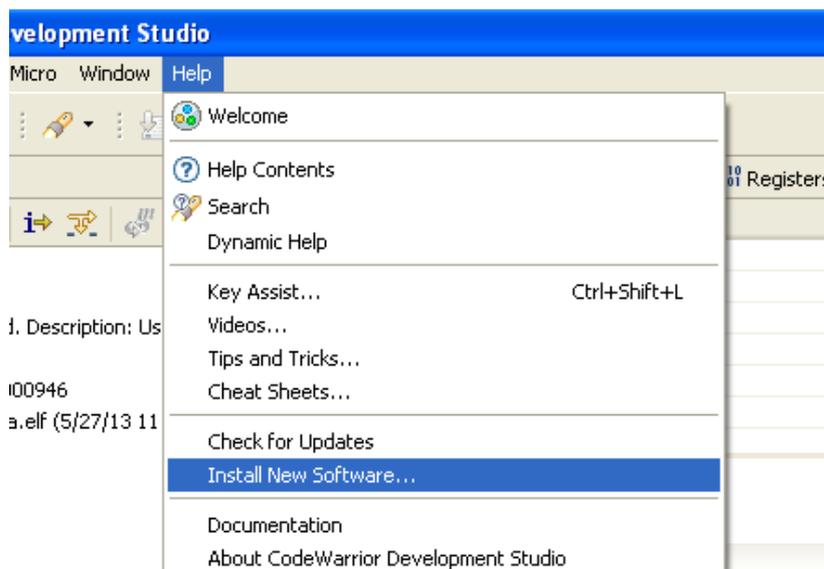
4. Na janela que se abre, temos as opções. Seleciona-se *Connection Type* para *Serial*, e se faz os ajustes necessários nos campos correspondentes.



5. Ao finalizar, o terminal será automaticamente conectado. Há botões na janela para se realizar conexão e desconexão, bem como outras funções.

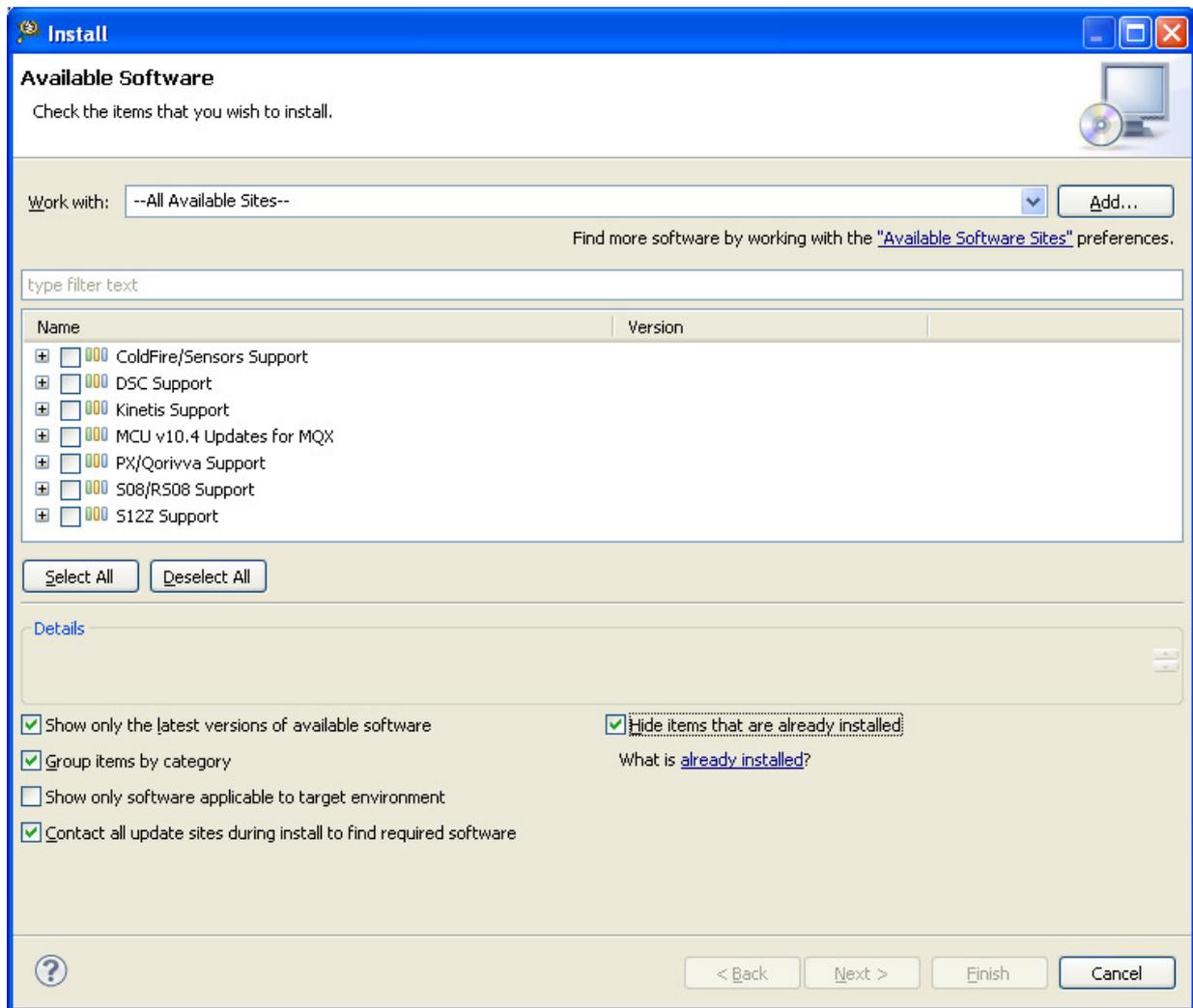
ADICIONANDO UM *PLUGIN*

1. Para se verificar atualizações, *patches* e outros *plugins*, usa-se a opção do menu principal *Help – Install New Software...* **Obs:** É necessário que o computador esteja conectado à Internet.



2. Na janela que se abre, podemos selecionar em *Work With* a opção *All Available Sites*. O botão *Add* permite que se adicione pastas de arquivos ou websites que contenham outros *plugins*. No CW10, o padrão é a página de atualizações da *Freescale*. Pode-se marcar as opções desejadas. É

interessante marcar a opção *Hide items that are already installed*. Após esta seleção, clicando-se em *Next* segue-se um procedimento similar à instalação de um programa, que pode incluir leitura e aceitação de termos, bem como confirmação de confiança no site de onde foi feito o *download*.



Revisado pelo Prof. Antonio Quevedo em 27/05/2013

As janelas apresentadas são de telas do CodeWarrior 10.4, executando em Windows XP. Pequenas diferenças podem aparecer em futuras versões do programa, e em outros sistemas operacionais.