**Preparação do Ambiente de Desenvolvimento**

O tema principal é preparar um ambiente de desenvolvimento de programas para o kit ARM Ravina, este utiliza o micro controlador lpc1752 de arquitetura ARM (Advanced RISC Machine) com núcleo Cortex-M3. O ambiente não só serve exclusivo para o KIT Ravina, mas sim para toda linha ARM independente de qual família ou fabricante. Neste ambiente podemos editar, compilar os nossos programas, e depois gravá-lo e depurá-lo em seus respectivos processadores.

Tendo um ambiente de desenvolvimento e entendendo como ele funciona, o desenvolvedor tem condições de estudar e explorar o mundo dos sistemas embarcados, executando muitos ideias e projetos desejados. Para os iniciantes recomendo a um bom livro de programação em C, independente qual arquitetura computacional, ler os datasheets dos processadores alvos, e se basear nos exemplos e em outros projetos publicados.

Os motivos principais para que essa estrutura de ambiente de desenvolvimento foi que:

1. Ficar livre de pagamento de licenças ferramentas de desenvolvimentos (IDEs) como KEIL, IAR e outros, utilizando somente as ferramentas GNU de uso livre. Isto é muito bom para hobbystas e para profissionais com poucos orçamentos;
2. Ficar livre de quaisquer ferramentas de desenvolvimento e de sistemas operacionais. Deixando o programador a escolher a ferramenta que melhor servir;
3. Obter o conhecimento necessário de como preparar um sistema e executá-lo em uma determinada arquitetura computacional. Isto envolve a criação de startup do processador, organização e utilização de suas memórias via linker script do compilador, e como organizar tudo isto junto com as fontes do programa usando o makefile. Pois, as IDEs abstraem este trabalho para você, simplificando tudo para o projetista, porém deixando muitas vezes o programador desconhecer desses detalhes técnicos. Com essa desvinculação das IDEs que proponho, permite criarmos um sistema integrado de aplicativos e bibliotecas que sejam independentes das IDEs dedicadas, dos sistemas operacionais e de quaisquer outras ferramentas de trabalho.

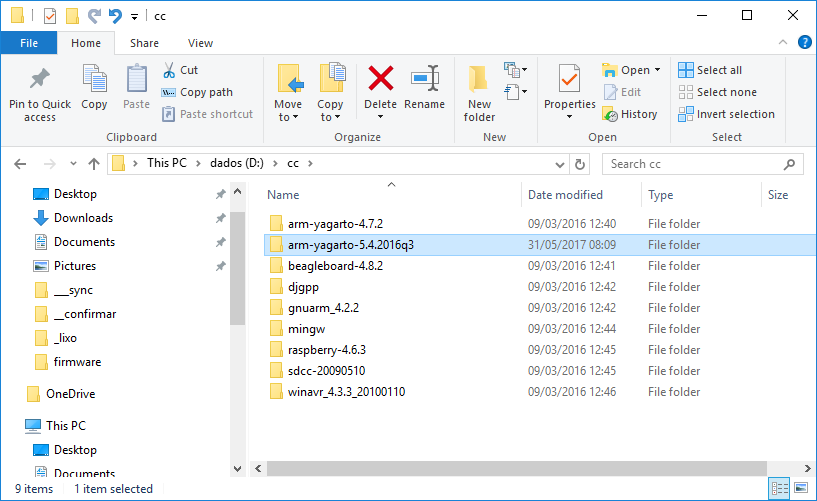
A preparação dos ambientes de desenvolvimentos descrito aqui foi feito em um PC rodando Windows 10 x64, porém o mesmo procedimento já foi usado em outras versões anteriores do Windows, e com algumas mudanças sutis também pode ser usado em ambiente Linux.

**Instalando o Compilador ARM - Yagarto**

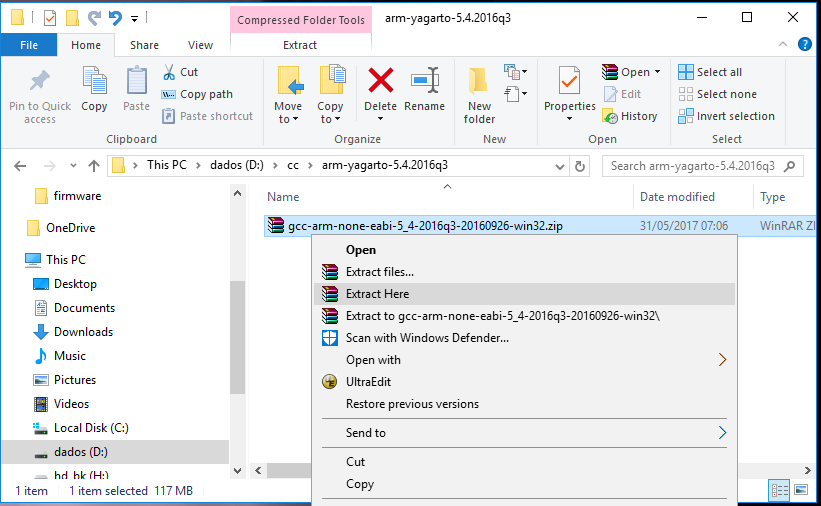
O compilador GCC para ARM que vamos instalar é na realidade cross compiler, ou seja, um compilador que gera códigos para outro processador que não seja o processador que está executando o compilador. Que em nosso caso estamos usando um x86 para gerar códigos para o ARM.

Há muitos tipos e versões de cross compilers para arquitetura ARM, entretanto, utilizamos o compilador Yagarto. O compilador Yagarto é outro projeto GNU GCC destinado a muitos processadores ARMs, é um similar ao GNUARM. Foi escolhido este compilador por ter o GDB modificado para aumentar o desempenho na depuração em sistemas embutidos, e também por estar constantemente sendo atualizado.

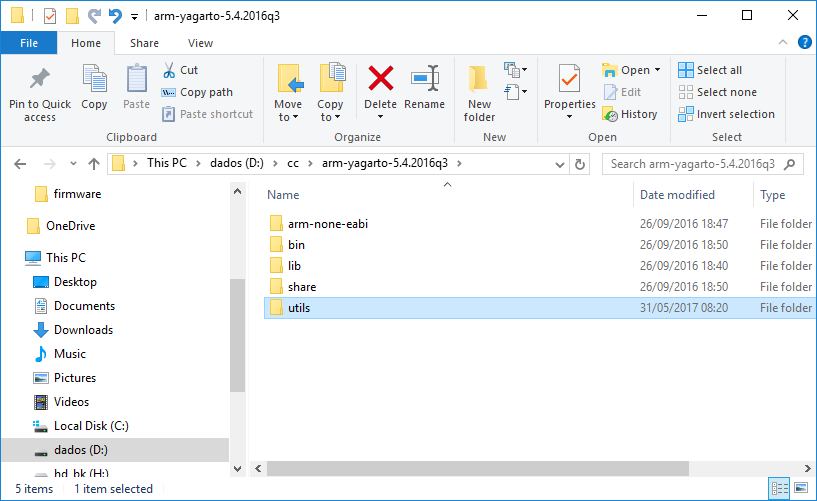
Em nosso estilo de trabalho para cada linha de processador ou versões diferentes de compiladores, criamos uma pasta exclusiva para ele. Isto garante as compatibilidades com os programas que já foram compilados com um determinado compilador, e permite testar novos compiladores. Então, crie uma pasta em um local adequado para instalar o compilador, neste processo sempre instalamos os compiladores em uma pasta chamada “cc” na partição D do HD, logo foi criada ali a pasta “arm-yagarto-5.4.2016q3”:



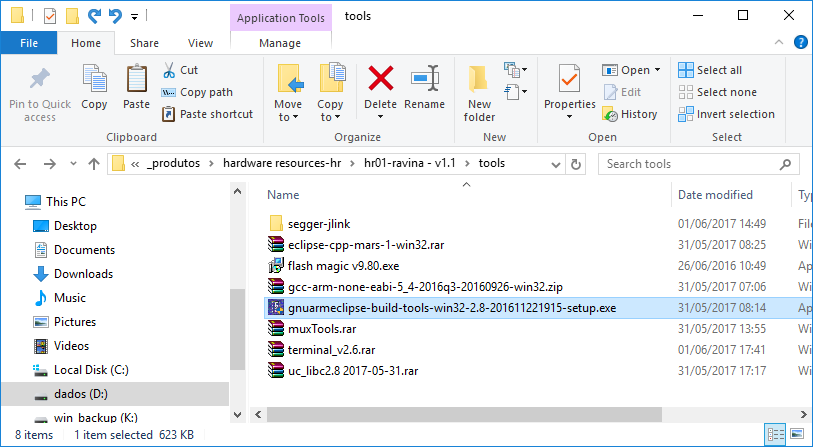
Copie o compilador compactado para a pasta recém criada e o descompacte:



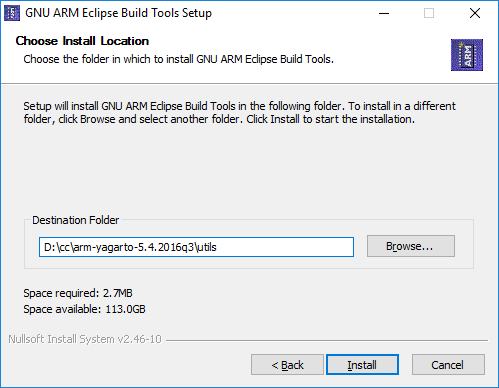
A instalação do compilador do ARM está concluída, mas ainda o trabalho não está terminado. É preciso instalar os utilitários necessários para usar o compilador no Windows. Dentro a pasta do compilador crie um diretório “utils” para gravar esses utilitários:



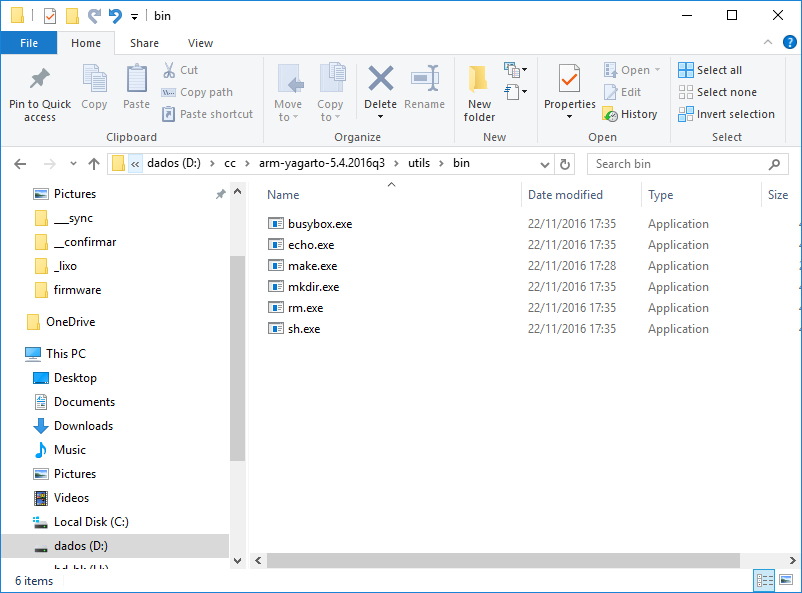
Execute o instalador dos utilitários:



Durante a instalação quando pedir o local para ser salvo os utilitários, aponte para a pasta “utils”:



Após o termina da instalação os utilitários já estão nos seus devidos lugares:



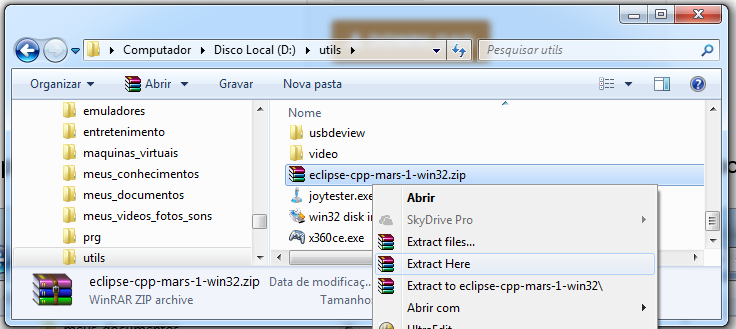
O compilador e as ferramentas foram pegos nos seguintes sites:

* Compilador: gcc-arm-none-eabi-5\_4-2016q3-20160926-win32.zip
  + <http://www.yagarto.org/>
* Ferramentas: gnuarmeclipse-build-tools-win32-2.8-201611221915-setup.exe
  + <https://github.com/gnuarmeclipse/windows-build-tools/releases/tag/v2.8>

**Instalando o Eclipse**

O ambiente Eclipse é uma excelente ferramenta de desenvolvimento na área de programação de softwares e firmwares de uso livre. O ambiente Eclipse é independente de qualquer linguagem de programação, e agrega muitos recursos necessários para um desenvolvimento, especialmente com suporte à depuração. O Eclipse é versátil em diversas áreas de programação, como JAVA, PHP, C/C++, VHDL, Verilog e muito outras. A outra vantagem de usar o Eclipse é que nos habituamos a sempre usar o mesmo ambiente de desenvolvimento para diversas áreas de projeto, nos acostumando sempre a utilizar os mesmos métodos e comandos de trabalho.

O Eclipse que vamos usar é a versão Mars (4.5). O Eclipse não é instalável, basta baixá-lo do site oficial e descompactar na pasta desejada. Além disso é preciso instalar os plug-ins CDT (C Development ToolKit), Zylin e GNUARM. Para poupar esse trabalho o Eclipse que acompanha esse Kit ARM já está preparado para nosso uso. Copie o Eclipse do CD do kit para um local em seu computador e o descompacte:

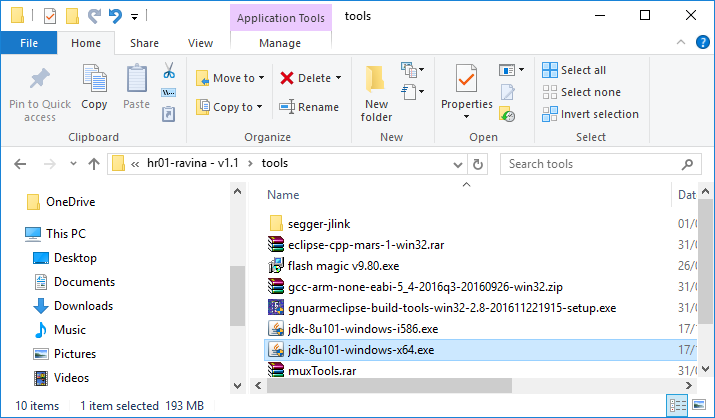


Eclipse com plug-in foi baixado do site: <http://www.eclipse.org/downloads/>

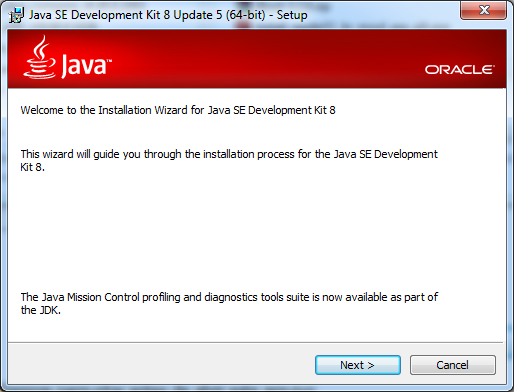
**Instalando Máquina Java**

Contudo, o Eclipse roda sob uma máquina Java, isto para ser portátil em muitos tipos de sistemas operacionais, logo, é preciso ter a máquina Java instalado no seu sistema operacional. Há dois tipos de pacotes de instalação do Java, um pacote chamado de Java Runtime Environment (JRE) que contém somente a máquina Java para rodar os aplicativos, e outro pacote mais completo tendo suporte para desenvolvimento chamado de Java Development Kit (JDK), que já contém o JRE. Logo, vamos usar a distribuição JDK para atender todas as necessidades de desenvolvimento que podem vir a necessitar.

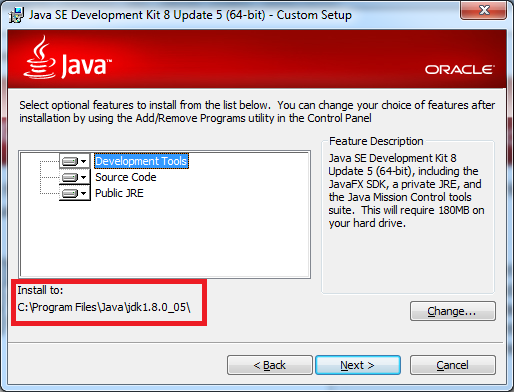
Vamos instalar as versões Java 8u101, mas podemos usar outras versões mais atuais. Execute o instalador JDK de 64 bits “jdk-8u101-windows-x64.exe” que acompanha o kit:



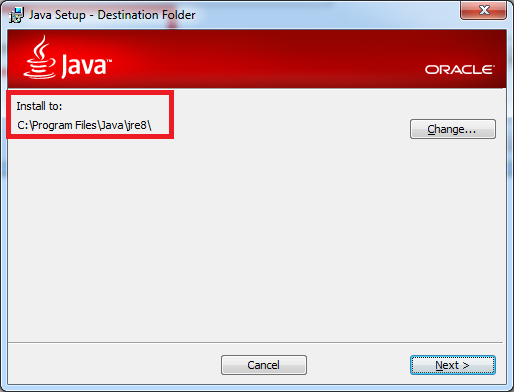
Esta é a janela de apresentação e prossiga com a instalação:



Deixe a opção no padrão e prossiga com a instalação:



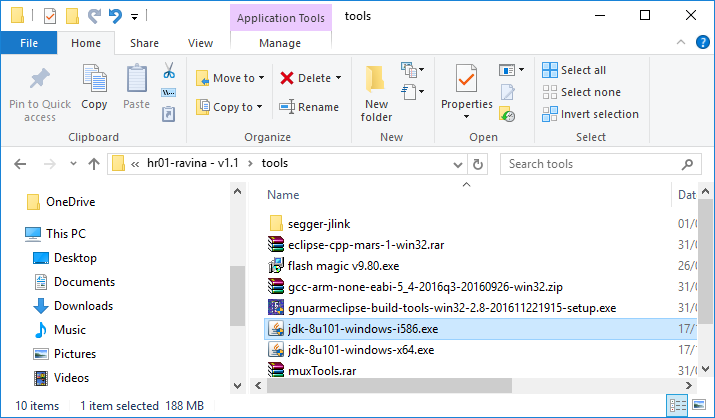
Após a instalação do JDK, vai aparecer à janela para instalação do JRE, prossiga com a instalação:



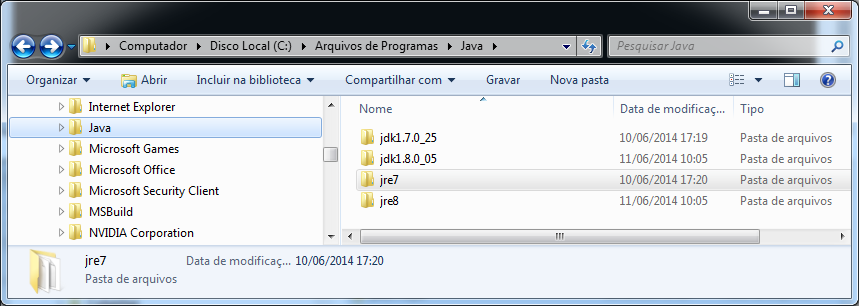
Conclua a instalação, opcionalmente clique em “Next Steps” para leituras de tutoriais e documentos:



Agora vamos instalar a máquina virtual de 32 bits executando o instalador “jdk-8u5-windows-i586.exe”:



Prossiga com a instalação seguindo os mesmos passos da versão x64. No fim da instalação o sistema operacional Windows terá duas versões do Java instaladas:



**Instalando Emulador J-Link**

O emulador que vamos usar para gravar e depurar os nossos programas no ARM via JTAG é o J-Link da Segger, este foi desenhado para muitas famílias de processadores ARMs.

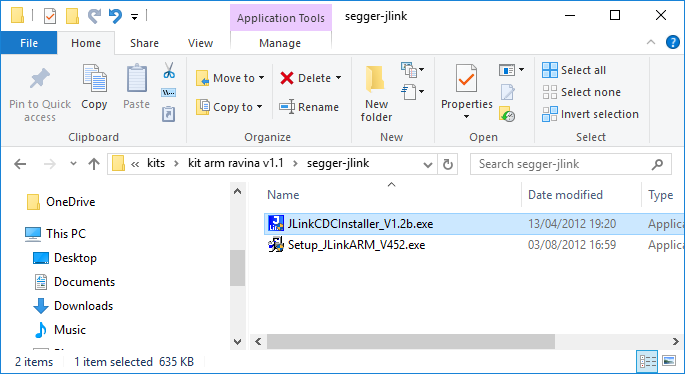
<http://www.segger.com/index.html>



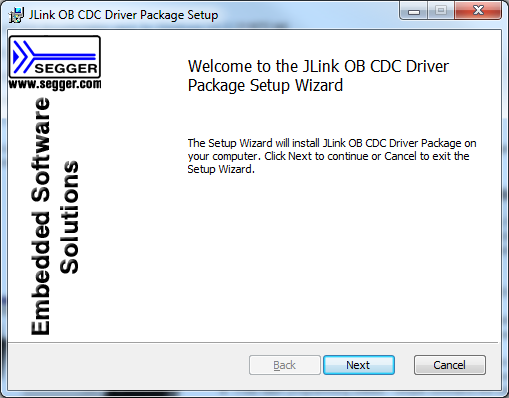
***O uso do emulador J-Link não é obrigatório***, podemos usar o aplicativo FlashMagic para somente gravar os nossos programas no ARM via porta UART, isto será descrito mais tarde de como se fazer. Entretanto, se não usar o emulador J-Link não será possível fazer as depurações, que são muito úteis no desenvolvimento de aplicativos.

**Instalando o Driver J-Link**

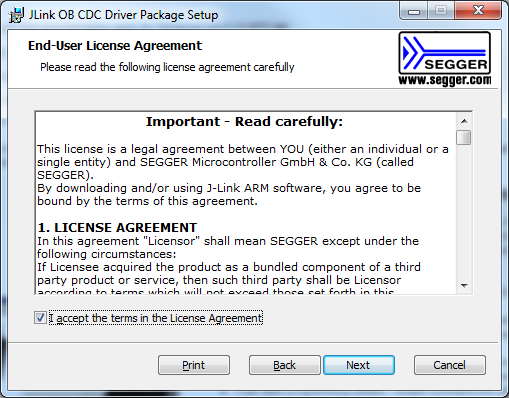
Vamos começar instalando o driver do J-Link. Execute o “JLinkCDCInstaller\_V1.2b.exe”:



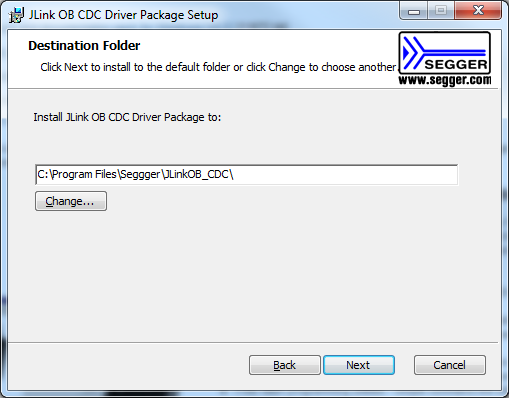
Na janela de apresentação clique em Next para continuar a instalação:



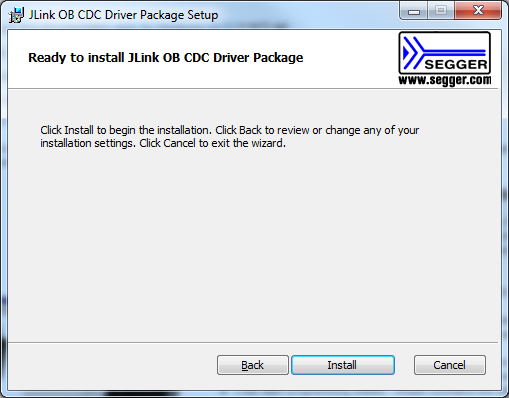
Aceite a licença e clique em próximo para continuar a instalação:



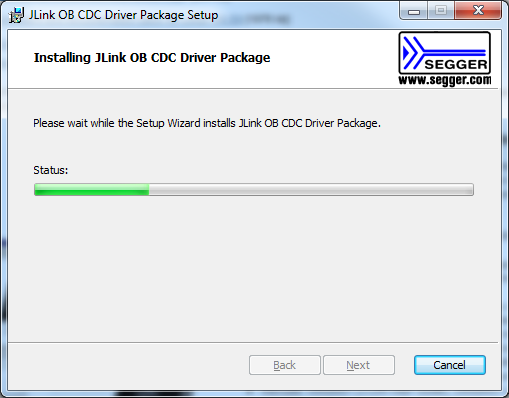
Opcionalmente aponte outro local onde instalar o driver e clique em próximo para continuar a instalação:



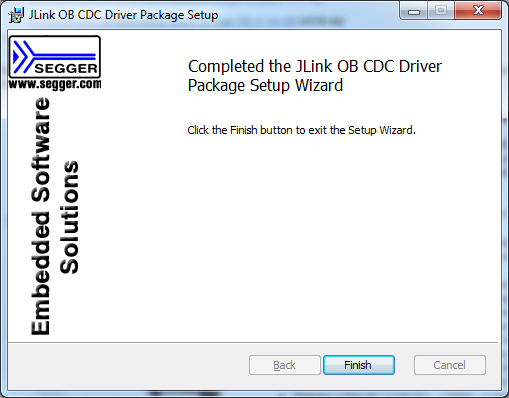
Clique em instalar para efetuar a instalação de fato:



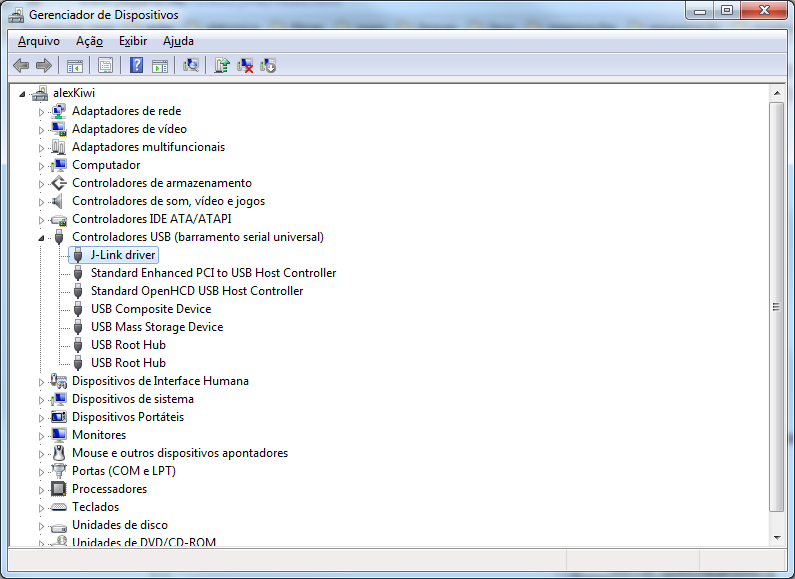
Espere pela instalação:



O driver do J-Link está instalado com sucesso, clique em finalizar para concluir a instalação:

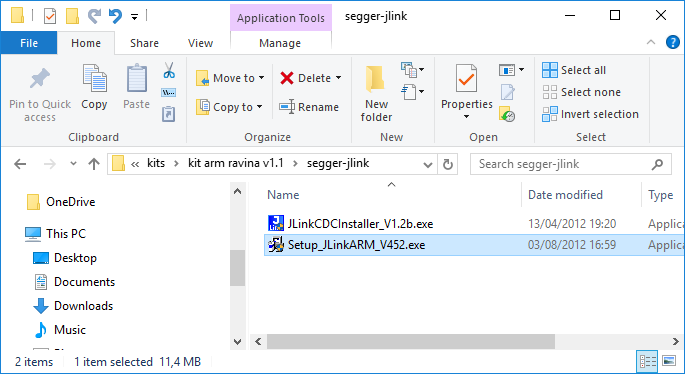


Conecte o emulador J-Link e espere que o mesmo seja reconhecido pelo Windows. Verifique se a instalação foi bem sucedida, para isto vá ao gerenciador de dispositivo do Windows e cheque se driver J-Link foi reconhecido como um dispositivo USB:

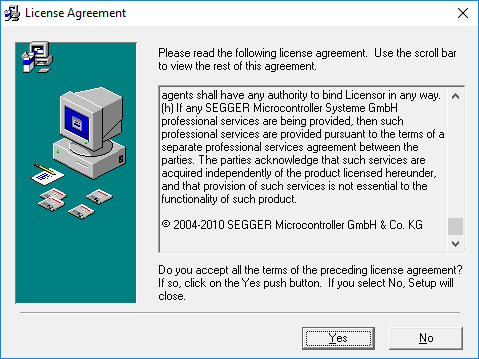


**Instalando o Servidor J-Link**

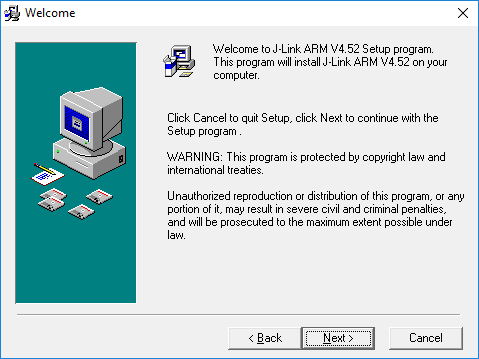
Execute o “Setup\_JLinkARM\_V452.exe” para iniciar a instalação do servidor J-Link:



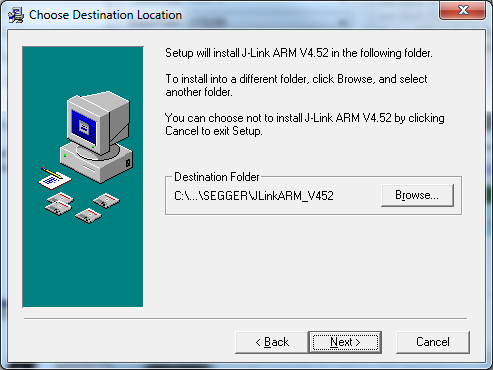
Na tela de licenças clique sim para aceita-la e continuar com a instalação:



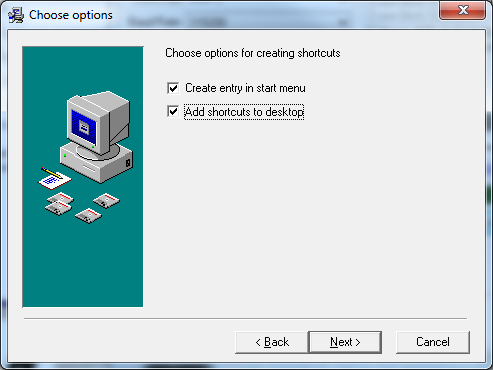
Na tela de apresentação clique próximo para continuar com a instalação:



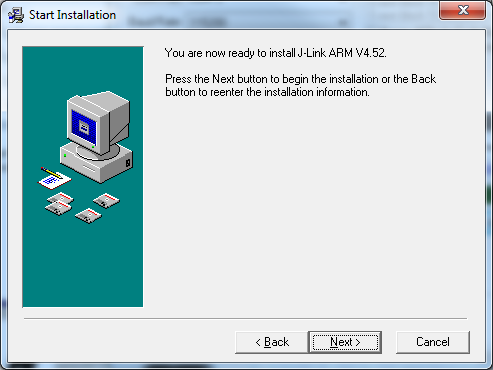
Opcionalmente aponte outro local onde instalar o servidor e clique em próximo para continuar a instalação:



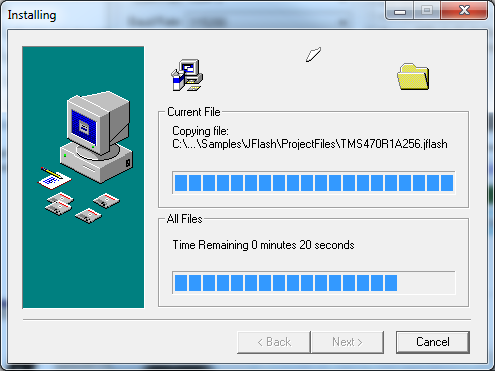
Opcionalmente selecione as opções de instalação e clique em próximo para continuar a instalação:



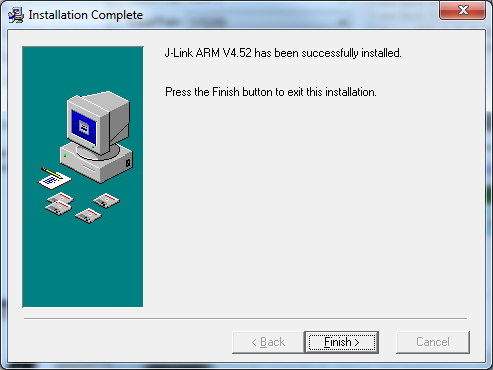
Inicie a instalação de fato clicando em próximo:



Espere pela conclusão da instalação:



O servidor do J-Link está instalado com sucesso e clique em finalizar para concluir a instalação:



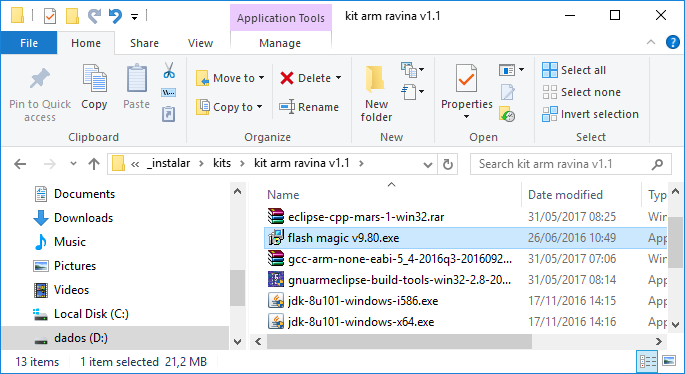
O servidor J-Link e depurador ARM J-Link está pronto para uso.

**Instalando o FlashMagic**

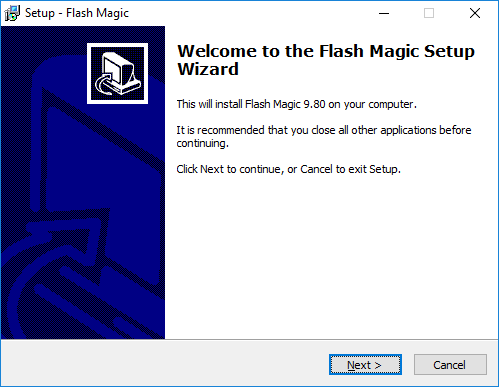
O FlashMagic é um aplicativo livre desenvolvido pela mesma empresa NXP, é a mesma empresa que fabrica o ARM lpc1752 usado pelo Kit ARM. Com FlashMagic podemos acessar o ARM via UART para diversas necessidades, os mais importantes são:

* Gravar os nossos programas do ARM. Uma alternativa de gravar nossos programas caso não disponibilizamos de emuladores JTAG/SWD;
* Visualizar os conteúdos de programa (FLASH) ou de dados (RAM);
* Apagar a memória de programa (FLASH);
* Checar ID e nível de proteção de segurança do código.

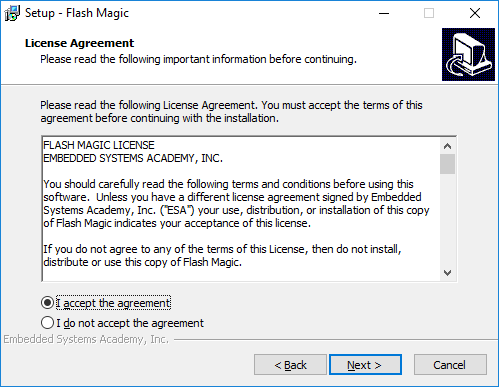
A versão do FlashMagic não é importante desde que tenha suporte para o ARM lpc2752 do Kit Ravina. Execute o “flash magic v9.80.exe” para iniciar a instalação da ferramenta:



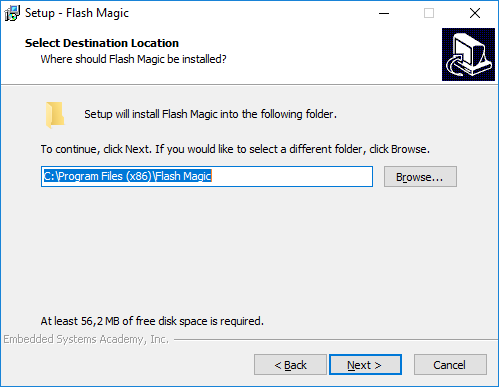
Na tela de apresentação clique próximo para e continuar com a instalação:



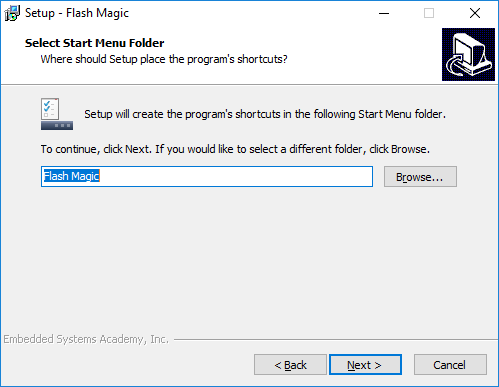
Na tela de licenças clique sim para aceita-la e continuar com a instalação:



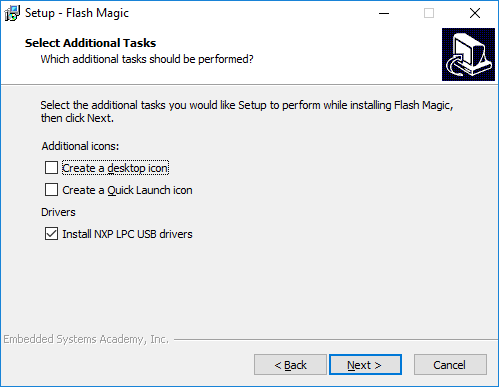
Opcionalmente aponte outro local onde instalar o FlashMagic e clique em próximo para continuar a instalação:



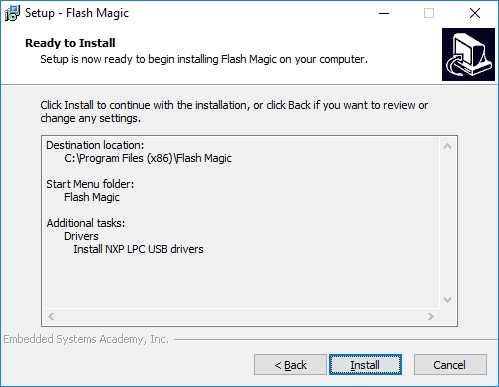
Opcionalmente aponte outro o item de menu e clique em próximo para continuar a instalação:



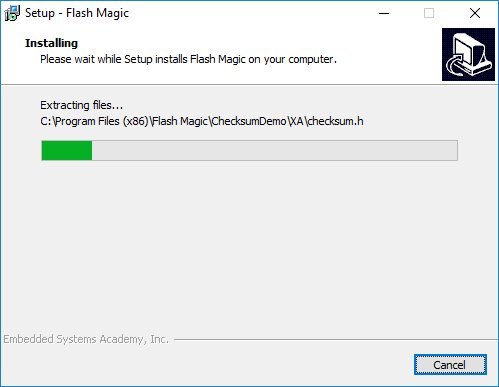
Opcionalmente selecione as opções de instalação e clique em próximo para continuar a instalação:



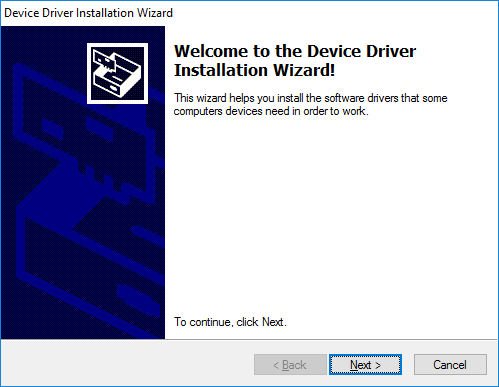
Inicie a instalação de fato clicando em instalar:



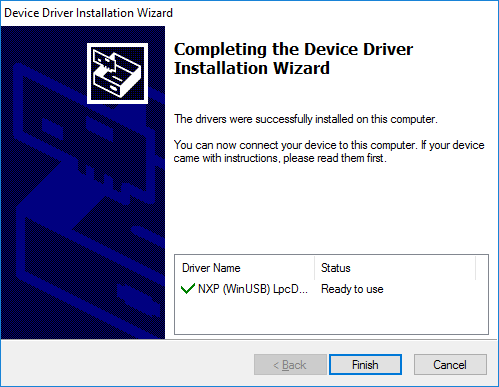
Espere pela conclusão da instalação:



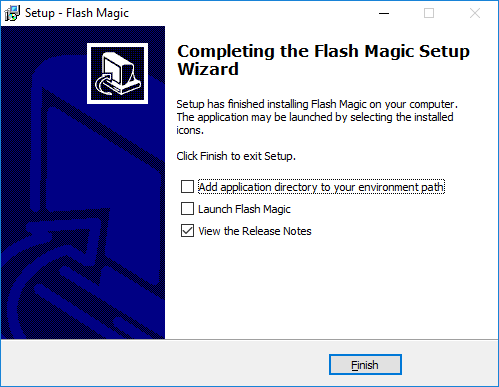
Agora clique em próximo para instalar um driver no Windows para usar o FlagsMagic:



O driver fora instalador com sucesso e conclua clicando e finalizar:



Conclua a instalação clicando e finalizar:



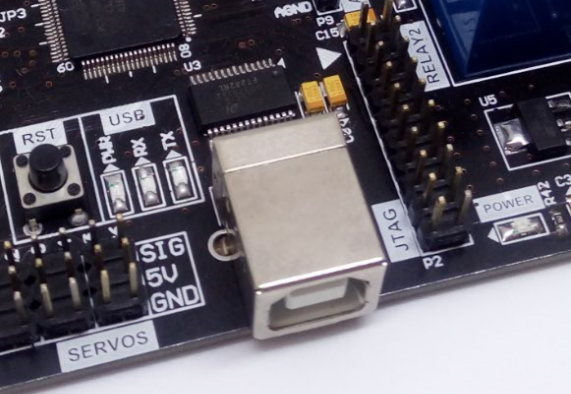
**Instalando Ferramentas de Apoio**

Junto com o Kit acompanham ferramentas para auxiliar o desenvolvedor e na criação dos programas embarcados.

**Conversor USB Serial**

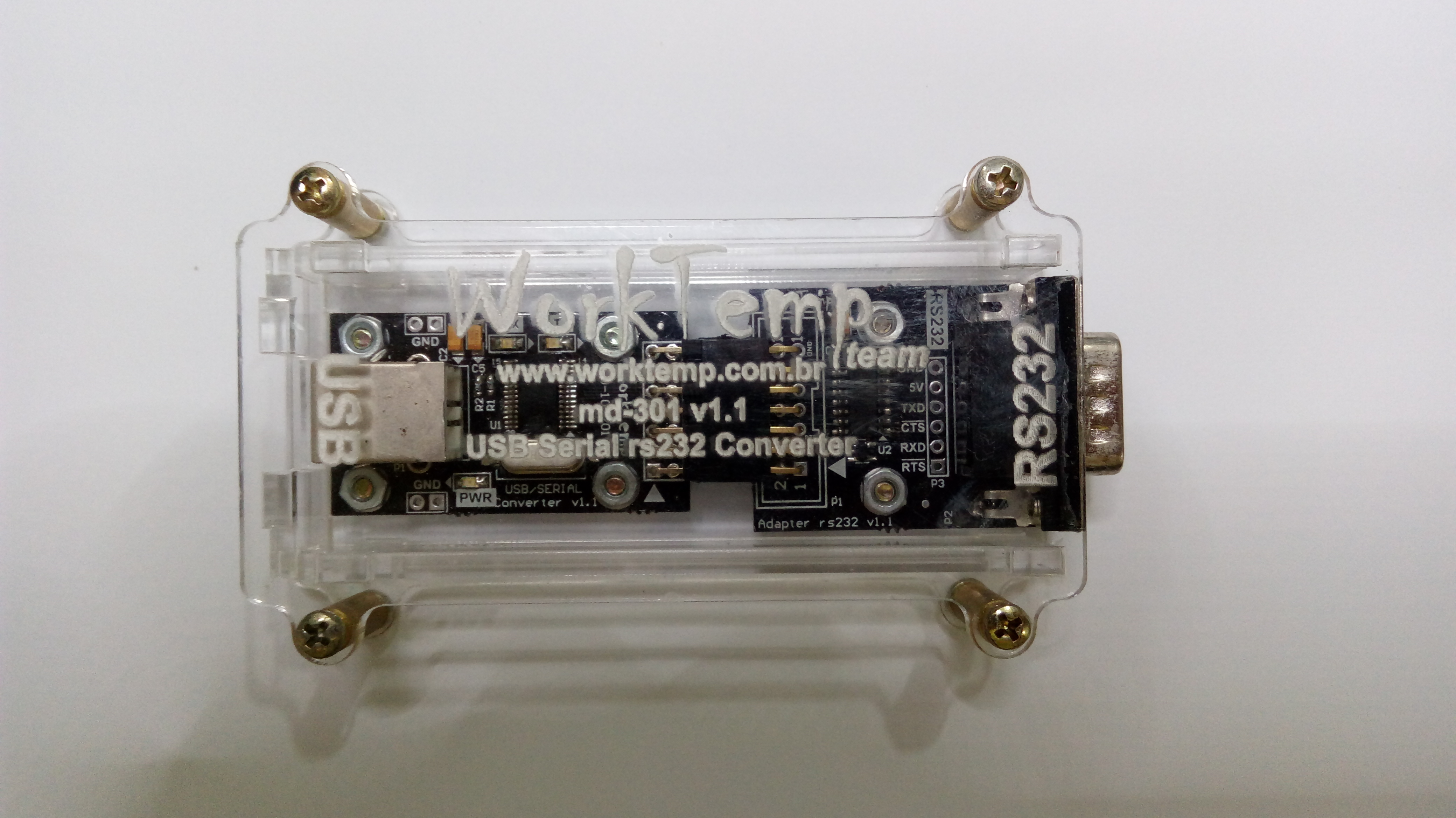
Hoje em dia é raro o computador que tenha uma porta serial UART disponível para uso, principalmente em notebooks. Como em ambientes embarcados é muito comum o uso dessa porta, é indispensável ter esse tipo de conexão com o seu ambiente de desenvolvimento.

O kit ARM Ravina já contém um conversor USB/SERIAL, e para instalar o driver no computador e como personalizar consulte o documento “usando conversor USB serial com FTDI.docx” que acompanha o kit.



Caso o seu computador não contenha uma porta COM UART e necessita usar mais de uma comunicação serial com o kit, a solução é usar os conversores USB para Serial. Os modelos de conversores baseado nas FTDIs é que foi obtido melhores resultados e funcionalidade tanto em Windows com Linux.

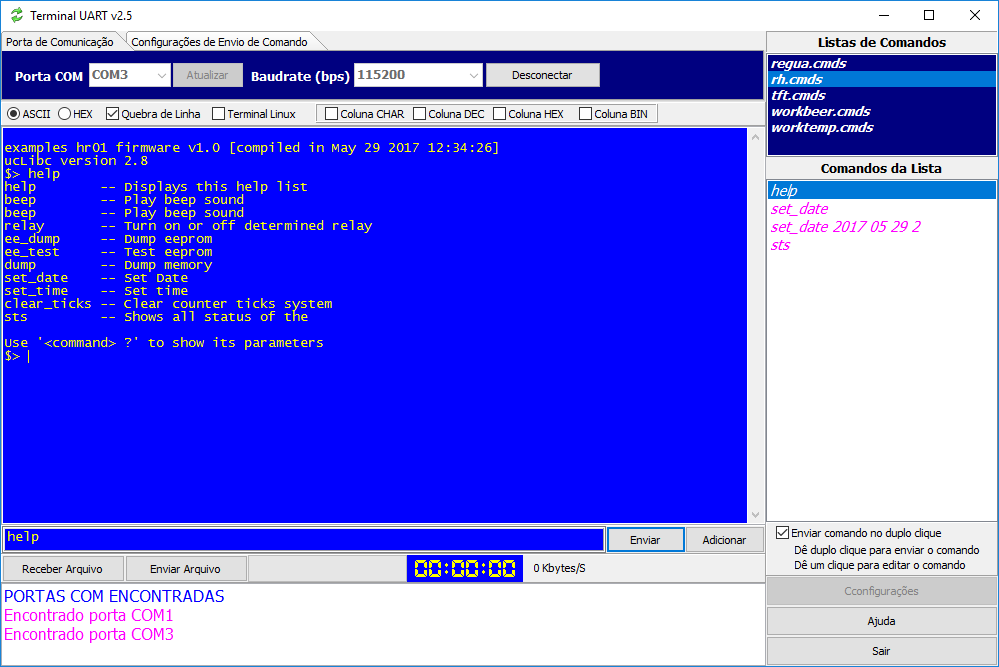




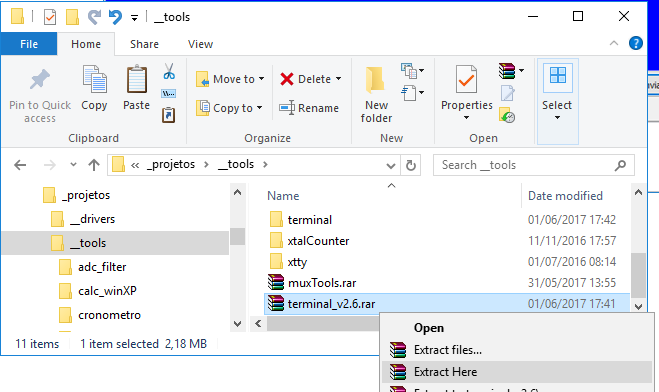
**Terminal UART**

O terminal UART é um terminal comum de comunicação via porta UART, o funcionamento essencial é o mesmo de outros programas de comunicação do gênero, porém neste foi agregado alguns recursos para facilitar o uso do interpretador de comados (CLI) gravados no ARM. O CLI (Command Line Interface) é um prompt de comandos que faz parte do pacote de bibliotecas fornecidos pelo kit, para que possamos interagir com os nossos programas gravados no ARM.

No exemplo abaixo mostra acessando a lista de comandos de um programa:

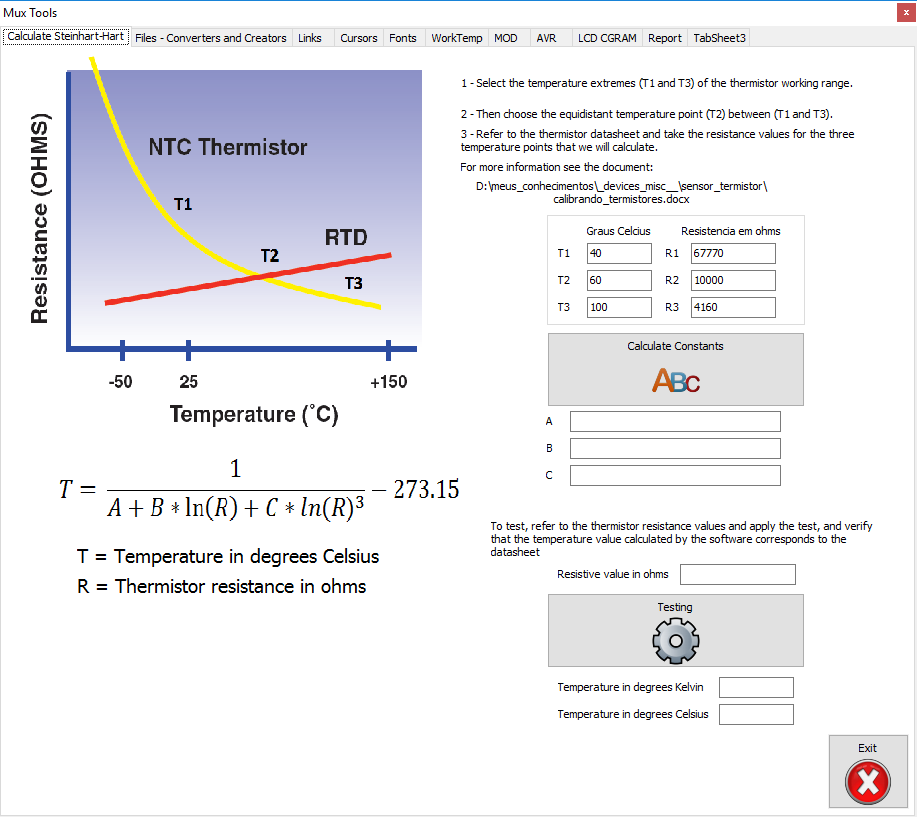


O terminal UART não é instalável, basta descompactar “terminal\_v2.6.rar” na pasta desejada:

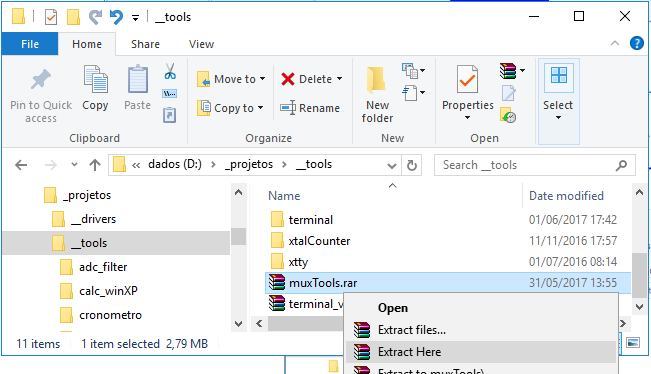


**MuxTools**

O programa muxtools contém alguns recuroso para ajudar no desenvolviemtno de sistemas embarcados, tais como: criar fontes de textos; calcular formula de sensores de temperatura; conversores de tipos e de arquivos, e entre outros:



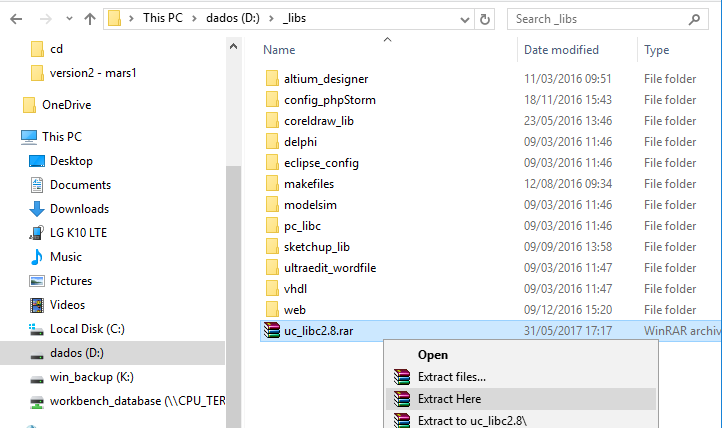
O muxtools não é instalável basta descompactar “muxTools.rar” na pasta desejada:



**Biblioteca uc\_libc**

Junto o kit é disponibilizado um conjunto de bibliotecas contendo muitos drivers e gerenciadores para acessar vários tipos atuadores; sensores, displays; sistema de arquivo FAT, freertos e muitos outros. Este conjunto de bibliotecas se chama uc\_libc, e não se restringe somente ao kit ARM Ravina, mas atende para outros processadores ARMs e AVRs.

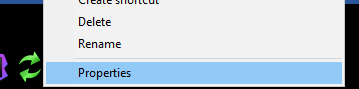
A versão atual da biblioteca é 2.8, e a mesma não é instalável, basta copiar e descompactar em um local desejado. Descompacte a uc\_libc e um local adequado:



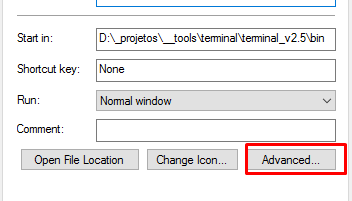
**Aplicativos Devem Rodar como Administrador no Windows 10**

Alguns aplicativos que vão acessar dispositivos de hardwares do computador devem rodar no modo administrativo, senão o Windows limitará o acesso a eles, como por exemplo o terminal UART que vai acessar a porta COM.

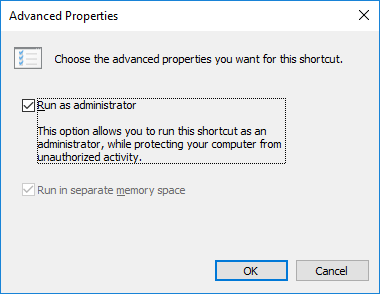
Exemplificaremos de como colocar o terminal para rodar como administrador. Clique com o botão direito do mouse sobre o ícone do terminal e escolha propriedades:



Clique na opção avançada:



E marque a opção para rodar como administrador:



**Concluindo**

O ambiente de desenvolvimento já está pronto para abrir os códigos exemplos e testá-los. Ou seja, as principais ferramentas estão prontas para uso onde possamos editar, compilar e gravar nossos programas no Kit ARM Ravina.