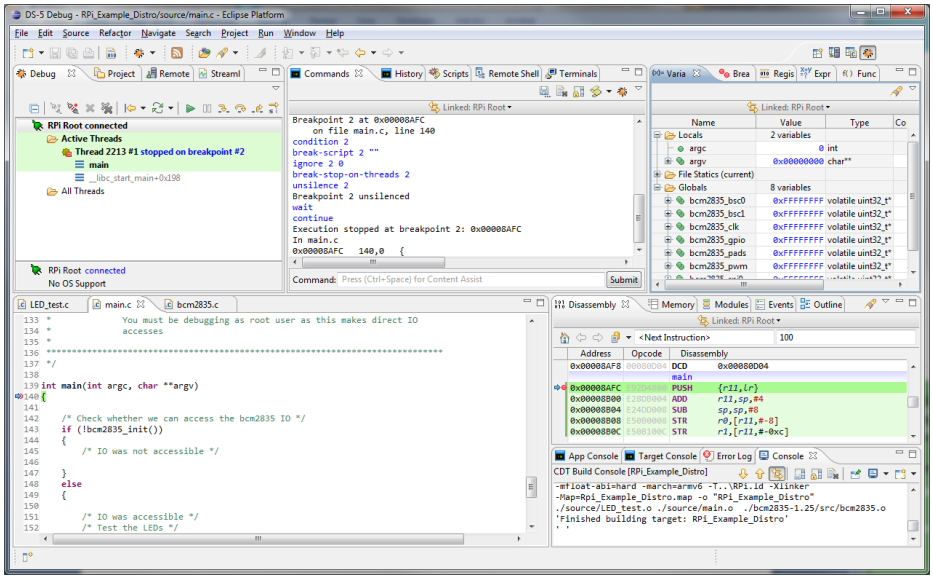
ARM DS-5 For Raspberry Pi: A professional PC-based C/C++ Eclipse development and debug environment for free!

Written by Mike Beach, Hitex UK Ltd

As dyed-in-the-wool C programmers, our first encounter with the Raspberry Pi was a bit frustrating. Whilst it is true that you can write C programs in a command-line fashion on the Pi, trying to run the normal Eclipse IDE completely overwhelms it. There is a lightweight C IDE that runs on the Pi, but it does not have any form of debugger. It is possible to use a C cross-compiler on a PC with the GDB debugger linked to the Raspberry Pi via Ethernet. There are a number of websites that give examples of using Eclipse on a Windows PC to write and debug Raspberry Pi programs but having tried them, none worked properly.

Our thoughts turned to the ARM DS-5 Eclipse-based debugger for Linux targets. This has predefined support for a large number of professional Linux/ARM boards but sadly there was no Raspberry Pi in the list. This is a very powerful tool that can undertake Linux and bare-metal development as well as perform complex tracing and analysis. However it can be used in a simple Linux debug mode for free via the Community Edition which just what the frustrated Raspberry Pi C programmer wants.

After some experimentation, we managed to produce a DS-5 Eclipse-based C/C++ cross development environment for the Raspberry Pi running on a Windows PC. This gives a professional PC-based development environment with the powerful DS-5 debugger built in. Programs are downloaded into the Pi via Ethernet and can be started, stopped and single-stepped in the usual way. It will make any C programmer feel instantly at home in the Pi world. The example is built using static libraries at the moment but we hope to have a shared library version soon.



You can find the DS-5 workspace containing the example at:
http://www.hitexuk.co.uk/RPi/DS-5 Workspace Export.zip (143MB direct download)

You can get the free DS-5 Community Edition from ARM at:
http://ds.arm.com/ds-5-community-edition/getting-started/

where there are instructions on how to download and install it.

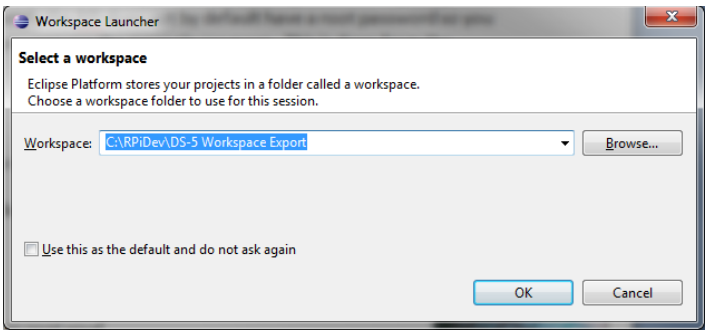
For the example program, we used Mike McCauley's bcmm2835 IO library to make a simple pin toggling program via direct access to the Broadcom chip's IO pins. This requires the Raspberry Pi to have a root user password. Unfortunately the RPi does not by default have a root password so you have to make one on your Pi before you can use the example program. This is done from the normal Pi user login with:

sudo passwd root

You have now set a root password you can login with e.g "yourrootpassword".

To install the example program, unzip it to "C:\RPiDev\DS-5 Workspace Export". This is the default as we zipped it up with the "full path info" option.

Once you have downloaded and installed DS-5 Eclipse, when you start it, you will be asked to specify a Workspace:

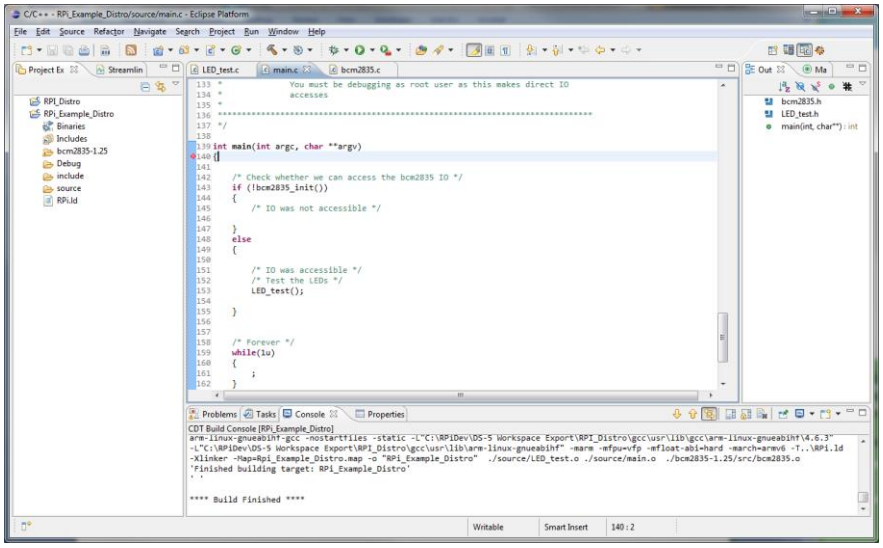


By default the example program will have installed to:

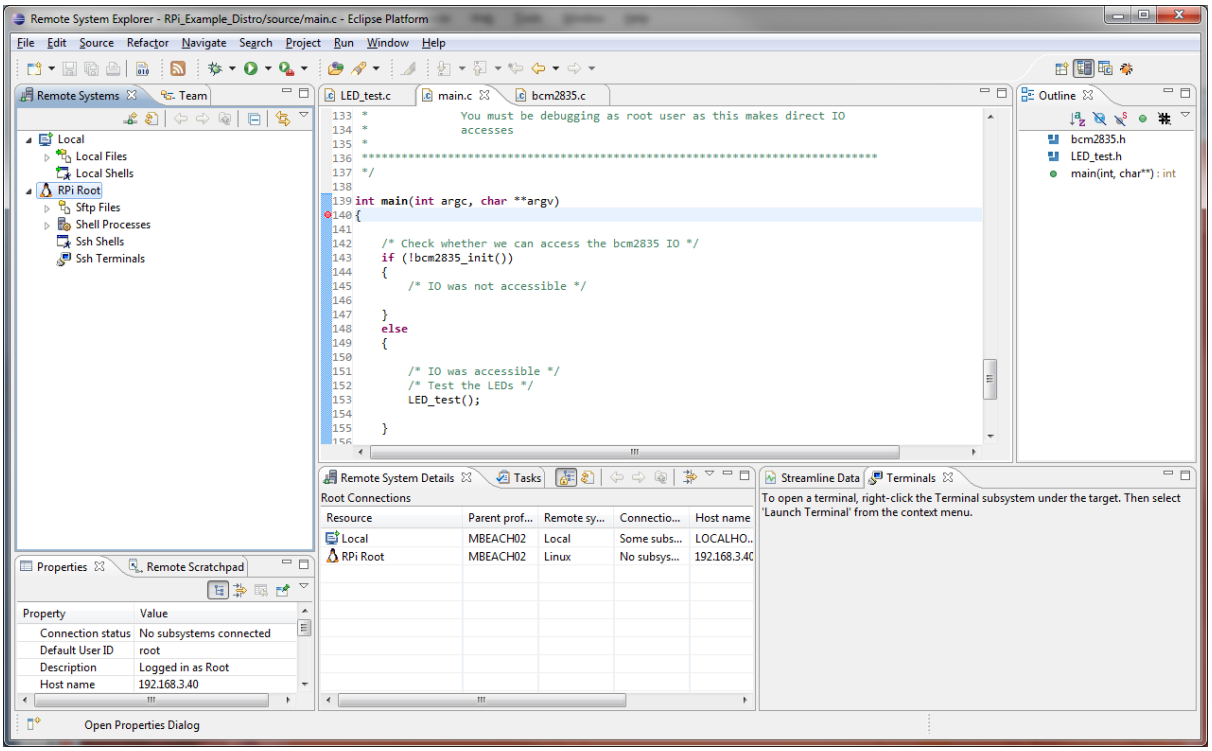
"C:\RPiDev\DS-5 Workspace Export"

so enter this path and click "OK".

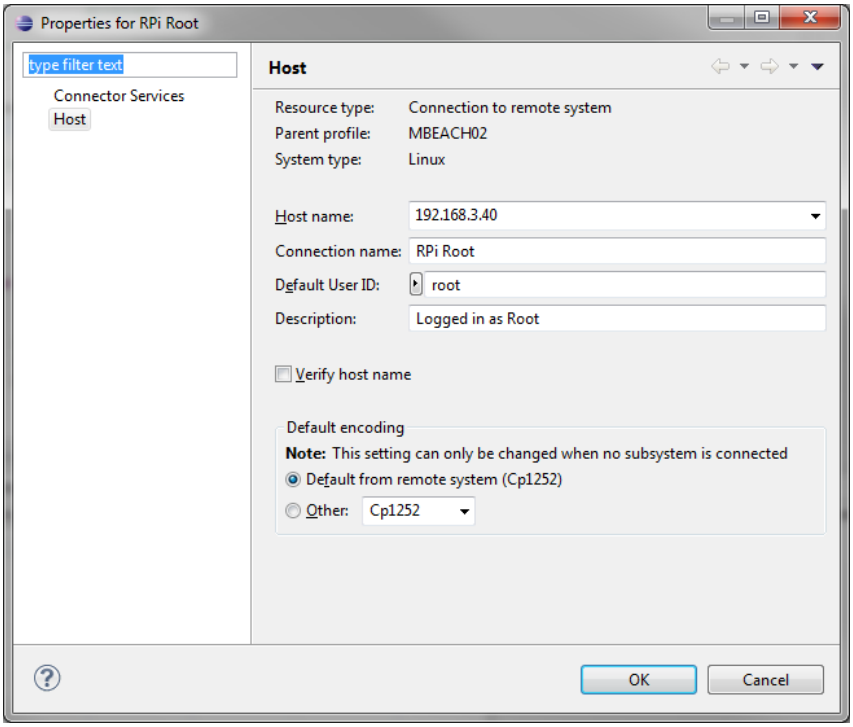
Before trying to download into the Pi, you must build the example program by using "Project-Build All" or clicking the  icon.



Next you will need to configure the Ethernet link to the Pi. This done under the Remote System Explorer menu. You can access this by clicking on the  icon. This changes DS-5 to show:

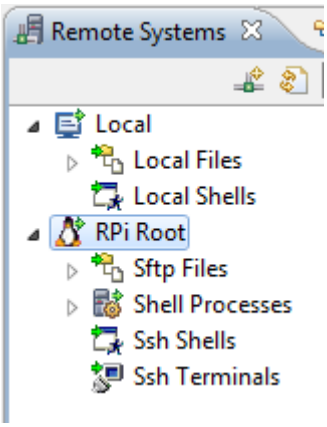


Now right-click on RPi Root to the left of the penguin. Now you need to set the IP address of your Pi in the host name box.

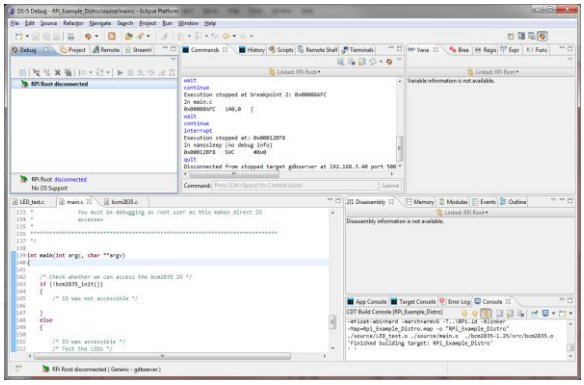


On our network the Pi was on 192.168.3.40 but yours will be different to this. You only need to change this item in the "Properties for RPI Root" menu. Save this new IP address by clicking on "OK".

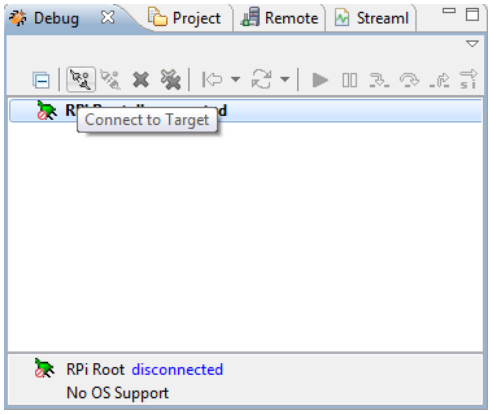
Now right-click RPi Root again and select "Connect". This should now connect to your Pi. The first time you do this you will be asked for your root user password. Enter it and tick the Save Password box. DS-5 should now be connected and green arrows will appear.



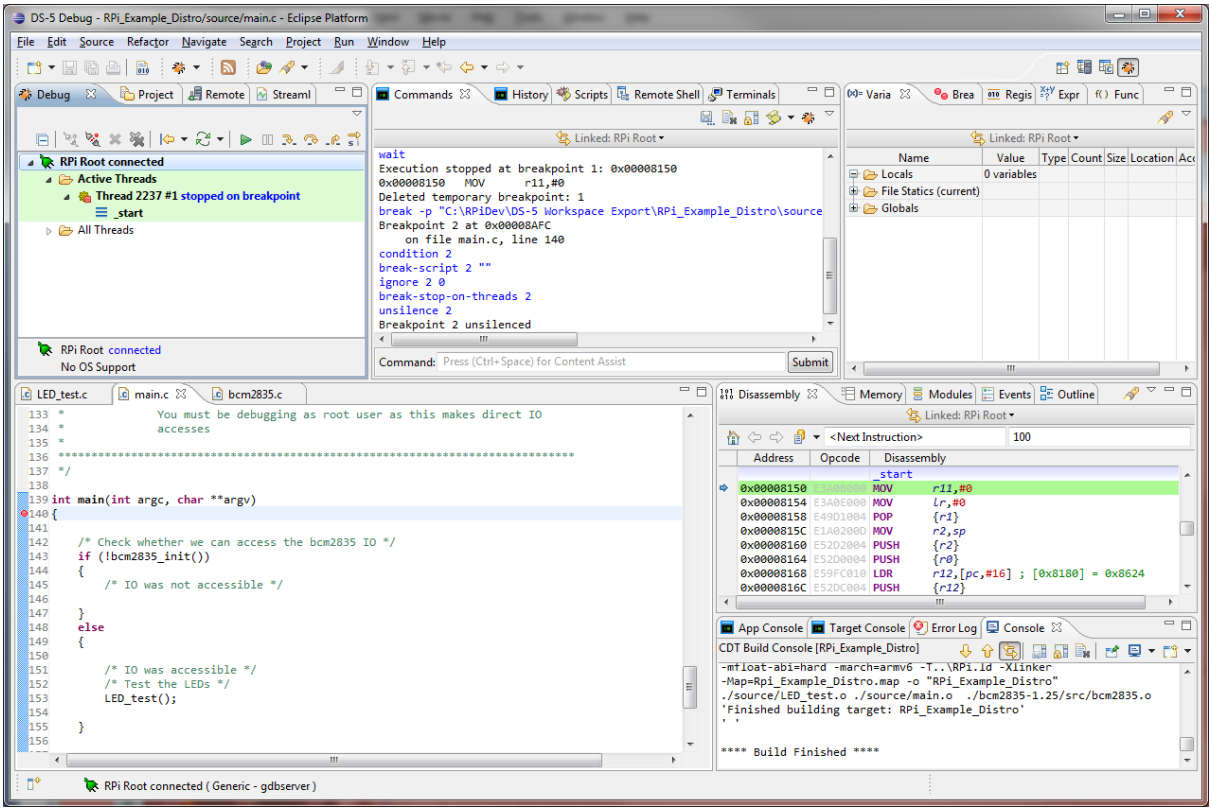
To download the example program into the RPi, click on the Debug perspective icon . However the DS-5 debugger will not be connected to the RPi yet:



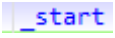
To connect it and download the program, click on the  icon.



Now the debugger will be sitting at the start of the program in the assembler code before main(). If it not there, click the reset icon  to reset the program counter to the start.



We have left a breakpoint at main() (the red dot) so click the Run  and the program should reach main and stop. Now if you run again, the RPI pins should toggle every second. You can stop it with the stop icon  .

To modify the C program, stop the debugger executing  and then reset it  . Now click on the C perspective icon  . When you rebuild it (), the new version will be automatically downloaded into the Pi and be waiting at the  symbol. Click on the Debug icon  to get to the debugger. If it is not at _start, quickly disconnect the Pi  and then reconnect it  . This will restore things to the correct state.

And there you have it: a professional cross C/C++ Eclipse development and debug environment for free!