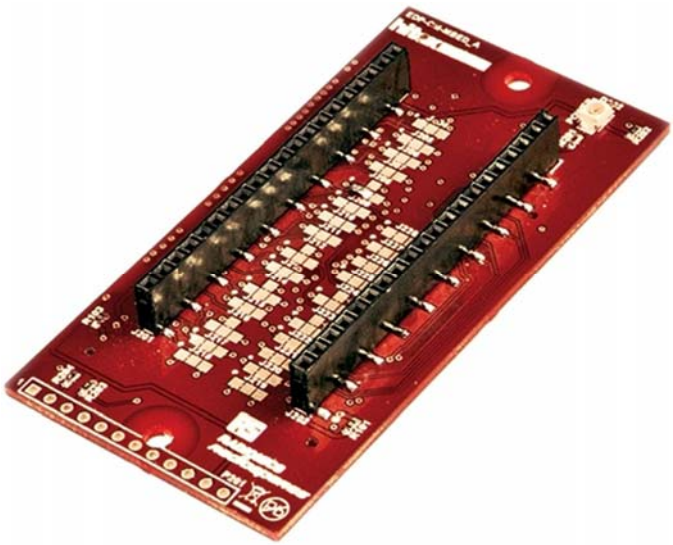
Embedded Development Platform

Getting Started Guide for the ARM mbed Carrier Module

EDP-CM-mbed



Version 3.11

February 2011

EMBEDDED DEVELOPMENT PLATFORM

Contents

1. Introduction

To get the most out of the EDP platform it's important to understand the concept of the EDP system correctly. This is detailed in the user manual for the Base Boards which can be downloaded as a pdf file **RS EDP-BB-SystemBaseBoard User Manual Vx**, from the RS EDP website.

The base boards come in both 2 position and 4 position formats and share a common user manual. Please read this manual to get an understanding of the system.

Each of the Command Modules (CM) and Application Modules (AM) has its own user manual, so again these documents must be read to get an understanding on how to use the modules.

Each of the boards comes with its own suite of software to fully exercise the EDP Application Modules and the peripherals available on the MCU device.

An EDP system usually consists of one Command Module and one or more Applications Modules plugged into a Base Board. A minimum system just has a Command module and Base Board, for example a simple web server operating through an Ethernet connection.

The Command Module dictates whether the whole system uses a supply voltage of +3.3V or +5.0V. This particular CM module use a +3.3V microcontroller (MCU) and so the board is configured as such. The user can check the Vcc_CM signal on the Base Board break-out header to confirm the system voltage.

There are 100 pins on the MCU of which 40 are made available on the mbed module. These are connected via various link options on the mbed carrier to the Base Board. The Base Board (BB) then provides these signals to the Application Modules thereby allowing the CPU Module to communicate with the Application Modules.

As many of the MCU pins have more than one function it can make the mapping of the connections rather complex so there are additional support documents available to help you with this. The first is the Pin Allocation Spread Sheet. One spread sheet is available for each of the CM Modules. The one for the NXP family of ARM-based modules is called:

Pin Allocation - 100 pin mbed Command Module Rev xx

This spreadsheet also forms part of the User Manual for the LPCxxxx CM module. It details which pins are mapped to the Base Board backplane and the various link options which need to be configured to connect them accordingly.

To get an appreciation of how the Application Modules are mapped to the backplane and how the CPU Module can connect to them, a Mapping Aid exists. The one for the mbed module is called:

Mapping Aid RS-EDP mbed Rev xx
(Name may vary slightly for other NXP modules)

This mapping aid also forms part of the User Manual for the mbed carrier Module and at a glance you can see what resources are required to get the best out of each Application Module.

Other useful documents you will need are the circuit diagrams for the modules you wish to use. These are contained in the back of each user manual.

So before you start to use the RS EDP system make sure you have to hand the following documents:

- Base Board User Manual
- mbed CM module User Manual
- Application Module User Manuals (as required)

2. Prepare to run the 'Hello World' Program

The ARM mbed module uses a completely different technique for program development than the conventional locally-based IDE method which involves programming/debugging via a suitable JTAG interface. Instead, all development tools are run from a remote server on the Internet at http://mbed.org/ . You must log in to this site to register your mbed module which gives you access to the Compiler and pages to document your activities.

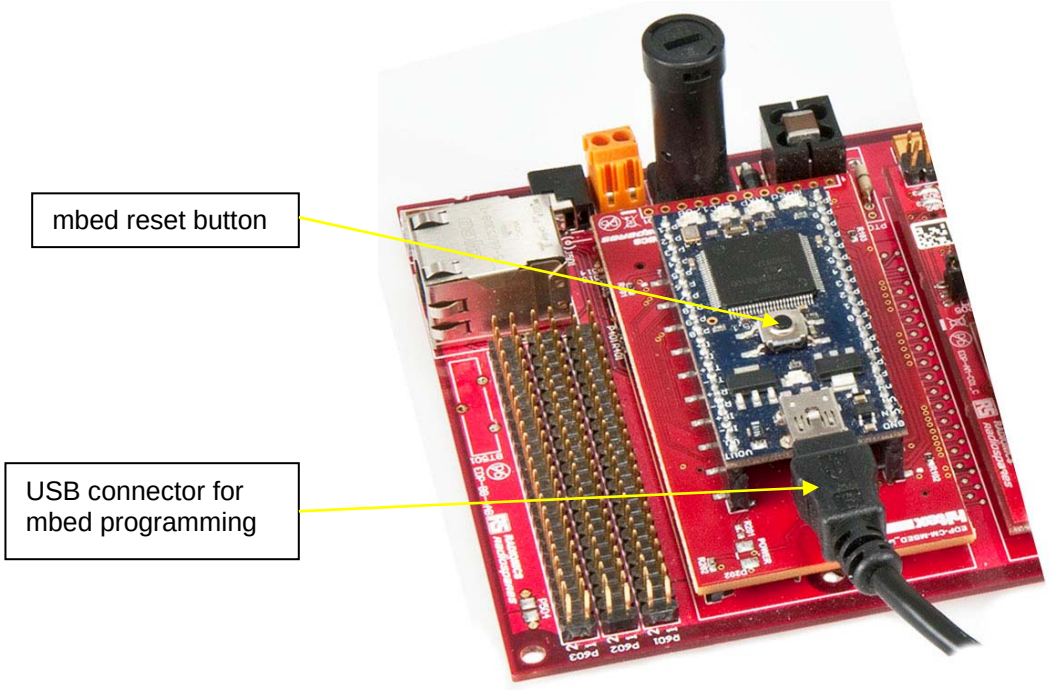
2.1 Software requirements

There are no expensive additional software requirements for mbed! All you need to generate some code in an object-oriented version of C language, compile it and program the mbed Flash memory is available on the web site. Once it is plugged into an EDP system however, you may need the drivers for any Application Modules you may be using and these can be downloaded from the RS EDP website. You might not need these at all because the mbed driver library contains all you need to drive, for example, an RS-232 port on the Communications module or the system I^2C bus. The mapping guide will show which mbed pin functions are connected to corresponding busses on EDP.

Connect the mbed module to an Internet-enabled PC with the USB cable provided and navigate to the mbed site with a suitable Browser. The module is powered from the USB connection and does not need to be plugged into anything else at this point. Follow the instructions to link up your module. You will also need to download a serial port driver and make sure you have a suitable serial terminal emulator: Windows HyperTerminal is just about adequate, but it is recommended you use one of the free alternatives such as **TeraTerm**.

2.2 Hardware requirements

As a minimum to get an EDP based system up and running you will need the mbed Carrier Module (EDP-CM-mbed) and an EDP Base Board. A simple 'Hello World' program that flashes an LED on the CPU module will work without the EDP, so to provide more of a confidence check the program used here will also print 'Hello World' on a terminal emulator via an EDP Communications module (EDP-AM-CO1).



Note correct orientation of mbed on the carrier module

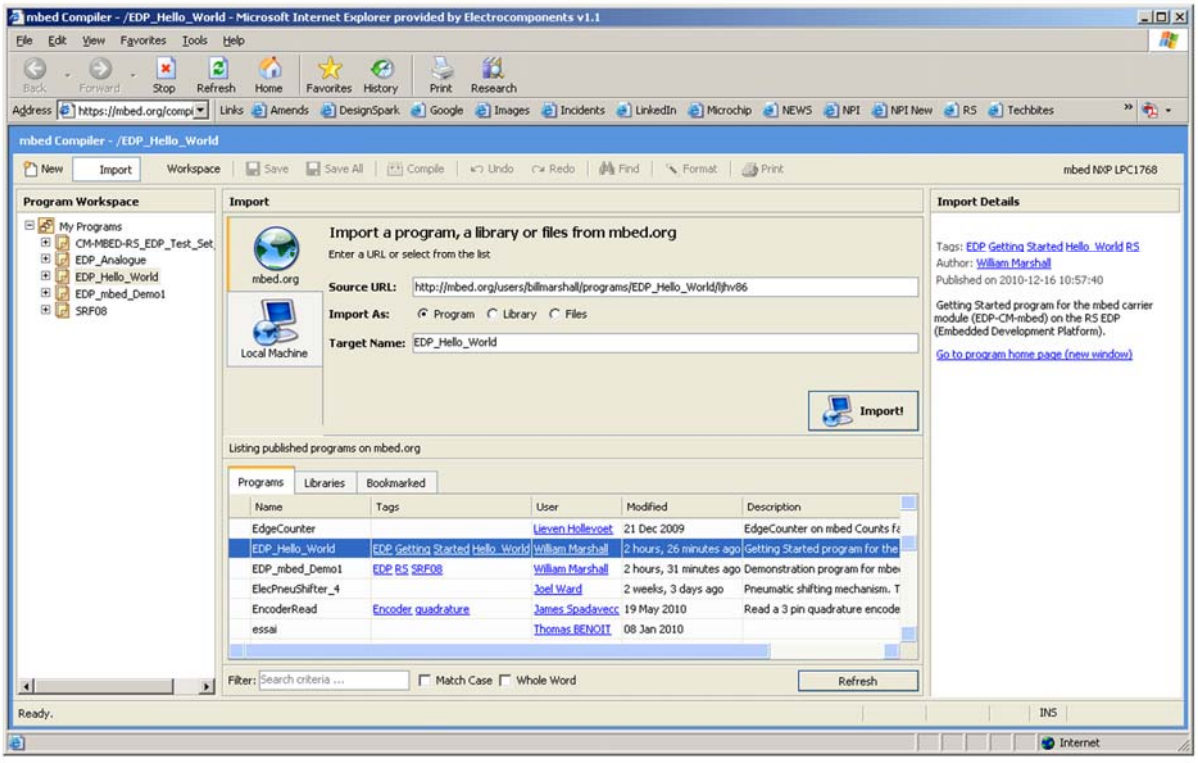
3. Load and Run ‘Hello World’

3.1 Set up hardware

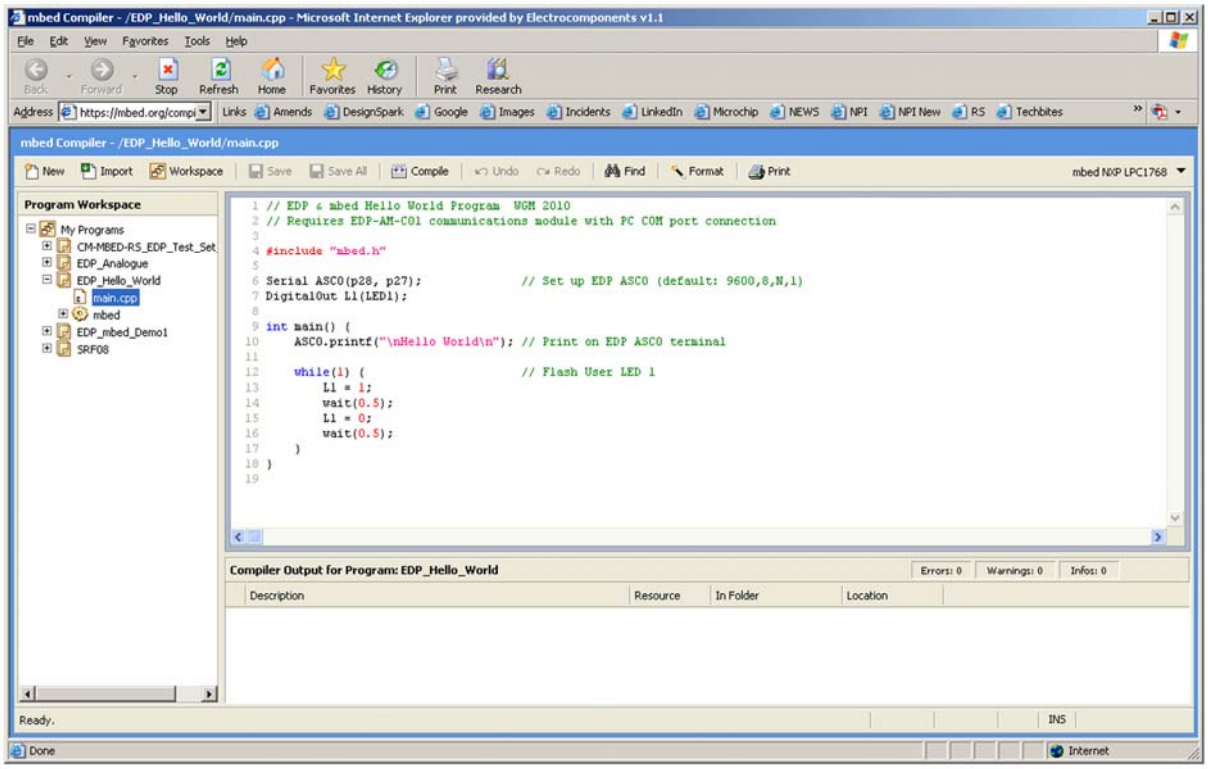
- Insert the mbed module into the adapter board. (Ensure correct orientation)
- Insert the complete CPU Module into a slot on the base board
- Use a USB cable to connect the mbed module directly to a PC USB port. Do NOT use the mini-USB socket on the Base Board.
- Use an RS-232 serial cable to connect the ASC0 port on the Communications module to a COM port on the PC. If the PC has no COM port, use an RS-232 to USB adapter.
- Connect a +12V DC power supply to the Base Board.

3.2 Set up software

- Run a Terminal Emulator on the host PC, select the ASC0 serial COM port and set it up for 9600 baud, 8 data bits, one stop bit and no parity.
- Login to the mbed website and start the mbed **Compiler**.
- Click on **Import**. Scroll down the list of available programs until you find **EDP_Hello_World**. Click to select it and then click the **Import!** button.



- Compile, build and save the code to the mbed module in the normal way.
- The mbed should be RESET via the Base Board or the mbed RESET button. Either will work.
- After resetting, the MCU Flash memory will be programmed with the new code.
- The message ‘Hello World’ should appear in the Communications channel terminal window, and User LED 1 will flash at 1Hz.



4. Conclusion

Now create your own programs using the low-level device drivers from the mbed **Handbook** and adapting high-level code from the **Cookbook**.

5. Appendix 1 EDP I²C Bus Device Addresses

8-bit address format is used: the LSB is reserved as the read-write bit. These addresses are defined in file: I2C-Directory.h

BaseBoard

BB_DIP = 0x40

8-bit DIP Switch

Address set by jumper link JP501 providing possible range of addresses: 0x40 (default), 0x42.

BB_EEPROM = 0xA2

Serial 4KB EEPROM memory

Address set by jumper links J601 to J603 providing possible range of addresses: 0xA0, 0xA2 (default), 0xA4, 0xA6, 0xA8, 0xAA, 0xAC, 0xAE.

Communications Module EDP-AM-CO1

CO1_RTC = 0xA0

Real-Time Clock and SRAM

Address set by jumper link J304 providing possible range of addresses: 0xA0 (default), 0xA2.

Digital I/O Module EDP-AM-DIO54

DIO54_Out = 0x46

Digital output latch

Address set by jumper links B305 to B307 providing possible range of addresses: 0x40, 0x42, 0x44, 0x46 (default), 0x48, 0x4A, 0x4C, 0x4E.

DIO54_In = 0x44

Digital input latch

Address set by jumper links B302 to B304 providing possible range of addresses: 0x40, 0x42, 0x44 (default), 0x46, 0x48, 0x4A, 0x4C, 0x4E.

Analogue Input Module EDP-AM-AN16

AN16_ADC = 0x6A

12-channel 10-bit Analogue to Digital Converter

AN16_Pot = 0x58

Digital Potentiometer for setting filter cut-off frequency of channels AN0 and AN1

Address set by jumper links J305 & J306 providing possible range of addresses: 0x58 (default), 0x5A, 0x5C, 0x5E.

Note that jumper links J204 & J205 select between CTRL_I2C (default) and I2C_GEN0 bus channels.

Dual BLDC Motor Module EDP-AM-MC2

MICROCHIP_MOTOR_DRIVER_BASE = 0x80

dsPIC U201

Address set by jumper links J201 to J203 providing possible range of addresses: 0x80 (default), 0x82, 0x84, 0x86, 0x88, 0x8A, 0x8C, 0x8E.

dsPIC U202

Address set by jumper links J205 to J207 providing possible range of addresses: 0x80, 0x82 (default), 0x84, 0x86, 0x88, 0x8A, 0x8C, 0x8E.

Note that spare addresses allow up to four MC2 modules per base board

Miscellaneous

SRF08 = 0xE0

Daventech SRF08 Ultrasonic Rangefinder module