Installing a Hosted and Embedded C Environment

In the first year you are taught how to program in both Hosted and Embedded C. To do this you need an environment to design, compile, test and run your code (or load it onto a chip). This document will guide you through the installation and setup of the recommended environment for Linux, Mac and Windows.
Notation

This document uses the following conventions:

command input  Command to be entered at the command line
1 Introduction

This guide details the setup of Hosted and Embedded environments which in a general form include:

- Compilers, both GCC and AVR-GCC for Hosted and Embedded environments respectively
- A GNU build system, Autotools for Mac and Ubuntu and MinGW for windows
- AVR C standard library and Simulator
- An IDE to write and develop code in, in this case Eclipse is used
- USB drivers for c232hm and USBAsp
- AVRDUDE to program ATmega chips
- Either Putty or Screen for UART communication

By the end of this you should have everything required for the first year C module and a good base for expansion beyond.

Though the final environment has been built with the aim of being platform independent, the install varies between platforms. Most of the programs required for the windows install are either provided by WinAVR or has a reasonably simple setup, both Ubuntu and Mac installations make heavy use of package management systems, homebrew for mac and the debian package manager for Ubuntu along with some more manual installation.

1.1 Software Overview

1.1.1 Eclipse

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. Being written in Java it is cross-platform and varies little between platforms, this being the main reason it was chosen over other IDEs. The IDE supports many languages including C and C++. In addition a software update enables AVR programming support, with a UI for working with AVR chips (programming, fuses, etc).
1.1.2 AVRDUDE

AVRDUDE (AVR Downloader/UploadEr) runs on Linux, FreeBSD, Windows, and Mac OS X, and supports a variety of in-system programming hardware, including Atmel AVR ISP mkII, Atmel JTAG ICE, older Atmel serial-port based programmers, and various third-party and "do-it-yourself" programmers.

1.1.3 GCC

The GNU Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain. As well as being the official compiler of the unfinished GNU operating system, GCC has been adopted as the standard compiler by most other modern Unix-like computer operating systems, including Linux, and the BSD family. Also included in this installation guide is AVR-GCC, a version of GCC built to build AVR code, it supports C and some of C++ (e.g. no C++ std or no 'new' operator).

1.1.4 binutils

The GNU Binary Utilities, or binutils, comprise a collection of programming tools capable of creating and managing binary programs, object files, libraries, profile data, and assembly source code originally written by programmers at Cygnus Solutions. The GNU binutils are typically used in conjunction with Compilers such as the GNU Compiler Collection (gcc), build tools like make, and the GNU Debugger (gdb).

1.1.5 avr-libc

AVR Libc is a Free Software project whose goal is to provide a high quality C library for use with GCC on Atmel AVR microcontrollers. Together, avr-binutils, avr-gcc, and avr-libc form the heart of the Free Software toolchain for the Atmel AVR microcontrollers.

1.1.6 screen

Screen is a full-screen window manager that multiplexes a physical terminal between several processes, typically interactive shells. It can be used to talk to devices over UART connections and comes with all Unix systems.
1.1.7 PuTTY

PuTTY is a free implementation of Telnet and SSH for Windows and Unix platforms, along with an xterm terminal emulator. It will be used to communicate of UART on windows, as getting screen to work on windows, though possible, is far too much work in this case.

1.1.8 libusb

Libusb is a C library that gives applications easy access to USB devices on many different operating systems. libusb is an open source project, the code is licensed under the GNU Lesser General Public License version 2.1 or later. There are two version you will encounter here, the standard libusb1.x.x and the libusb-compat with the latter being a wrapper for the former allowing programs that used libusb 0.1.x to use the newer version, also there is a version of 0.1 for windows called libusb-win32.

1.1.9 MinGW

MinGW (or Minimalist GNU for Windows) is a software port of the GNU Compiler Collection (GCC) and GNU Binutils to, it functions as a toolchains to build applications for Windows. It also comes with other software such as MSYS (Minimal System) which provides Unix-like shell commands such as make. It is similar to another program, Cygwin, which also includes many similar features such as a GCC compiler but aims to more emulate Unix in windows rather than build native programs.

1.1.10 Homebrew

Homebrew is a package management system that is simple and easy to use and uses ruby, it is similar to the other main package management systems on OS X Fink and Macports. It was chosen over these however due to ease of use and installation. It require that Command Line Tools be installed (aka autotools from apple).
2 Windows

2.1 Hosted Environment Setup

2.1.1 MinGW

To start the installation download the MinGW installer and run it. When presented with the options "Use pre-package repository catalogues" or "Download latest repository catalogues" pick the latter, accept the terms and keep the installation directory as default. When selecting components to install, select the C compiler, C++ compiler, MSYS Basic System and MinGW Developer ToolKit, click next and then install and wait for MinGW to install and download everything.

After this add the /bin folders of MinGW and MSYS to the Environmental Path Variable:

1. Go start
2. Right click Computer
3. Select Properties
4. Click Advanced system settings
5. Then Environment Variables
6. In the System variables find the path Variable
7. Click edit and append the install locations of MinGW and MSYS, with a standard install these would be: "C:\MinGW\bin;" and "C:\MinGW\msys\1.0\bin;"
8. Click ok and then again to confirm complete

This is done so that other programs know where to find MinGW executables.

2.1.2 Eclipse

Make sure you have the latest version of JAVA installed. Download the Eclipse IDE for C/C++ Developers, extract the archive to a reasonable install location and run the eclipse executable within.

2.2 Embedded Environment Setup

2.2.1 avr-gcc toolchain

Download WinAVR and run the installer.
2.2.2 **AVRDUDE**

Browse to the installation location of WinAVR (e.g. C: \WinAVR – 20100110) and enter the bin folder; create a folder here called old and move the files avrdude.exe and avrdude.conf from the bin folder into it. To update AVRDUDE to the latest version 6.3 which includes support for the C232HM cable download avrdude.exe and avrdude.conf and place them in the WinAVR bin directory (e.g. C: \WinAVR – 20100110\bin).

2.2.3 **Zadig**

Download and install Zadig. Run Zadig and select the WinUSB driver for the usbasp device (you will need to connect your Il Matto and enter the boot loader).

2.2.4 **Eclipse**

Open Eclipse and go Help - Install New Software... - then click Add... - Enter AVR for Name: and http://avr-eclipse.sourceforge.net/updatesite/ for Location: - click ok - check "AVR Eclipse Plugin" - click next and install. Then go Window - Preferences, in AVRude under AVR edit the AVRDude path (path source custom) to the folder 'old' you created in the bin folder of WinAVR, which contains the version of AVRDude that came with WinAVR.

2.2.5 **PuTTY**

Go to the PuTTY homepage and select the Windows MSI installer package for everything except PuTTYtel.

2.3 **Using the C232HM Cable**

This cable is versatile and can be used as an AVR programming cable or as a serial interface to the host computer. You will need to instruct the host how you wish to use it. If you wish to use it as an AVR programming cable you will need to do the following:

*Use Zadig to set the WinUSB driver for the C232HM cable*

If you wish to use it as a serial interface you will need to do the following:

*Use Zadig to set the FTDI driver for the C232HM cable*
3  Mac

Make sure you have the homebrew package manager installed. Open a terminal prompt and enter the command:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Then enter the following command:

```
brew install caskroom/cask/brew-cask
```

3.1  Host Environment Setup

3.1.1  Command Line Tools

Try entering the following command at the terminal prompt:

```
xcode-select --install
```

If that fails, you are using an old version of Mac OS X and then you will have to install the tools by going to the Apple Developer Site. Firstly register a free Apple developer account and then login and download the command line tools.

3.1.2  GCC Compiler

```
brew install gcc
```

3.1.3  Eclipse

```
brew cask install java eclipse-cpp
```

Eclipse will be installed in under the /opt/homebrew-cask/Caskroom/eclipse-cpp/ directory. Move Eclipse.app into the Applications folder and put it on the Doc for simple access.
3.2 Embedded Environment Setup

3.2.1 avr-gcc toolchain

Open a terminal prompt and enter the command:

```
brew tap osx-cross/avr
brew install avr-libc
```

3.2.2 AVRDUDE

```
brew link libusb libelf libusb-compat libftdi0
brew install avrdude --with-usb
```

3.3 Using the C232HM Cable

This cable is versatile and can be used as an AVR programming cable or as a serial interface to the host computer. You will need to instruct the host how you wish to use it. If you wish to use it as an AVR programming cable you will need to issue the command:

```
sudo kextunload /System/Library/Extensions/FTDIUSBSerialDriver.kext
```

If you wish to use it as a serial interface you should issue the command:

```
sudo kextload /System/Library/Extensions/FTDIUSBSerialDriver.kext
```
4 Linux

Make sure that your version of the package manager is updated with the terminal command:

```
sudo apt-get update
```

4.1 Hosted Environment Setup

4.1.1 GNU Build Tools

To install build and debugging tools open a terminal window and enter:

```
sudo apt-get install build-essential automake autoconf gdb
```

4.1.2 Eclipse

To install Eclipse-CDT open a terminal window and enter:

```
sudo apt-get install eclipse eclipse-cdt
```

4.2 Embedded Environment Setup

4.2.1 avr-gcc toolchain

Open a terminal window and enter:

```
sudo apt-get install gcc-avr binutils-avr gdb-avr avr-libc
```

4.2.2 AVRDUDE

Open a terminal window and enter:

```
sudo apt-get install avrdude
```
4.3 Using the C232HM Cable

This cable is versatile and can be used as an AVR programming cable or as a serial interface to the host computer. You will need to instruct the host how you wish to use it. If you wish to use it as an AVR programming cable you will need to issue the command:

?  

If you wish to use it as a serial interface you should issue the command:

?