

```
/*
 * Initial CMUcam3 (cc3) data type definitions
 *
#include <stdbool.h>
#include <stdlib.h>

#include "cc3.h"
#include "cc3_pinDefines.h"
#include "cc3_hal.h"
#include "serial.h"
#include "devices.h"
#include "interrupt.h"

// Globals used by CMUCam functions
cc3_pixel_t cc3_g_current_pixel;
cc3_frame_t cc3_g_current_frame;
```

cc3.c

# CMUcam3 SDK Installation Guide

This page intentionally left blank.

## Contents

1. INTRODUCTION .....	1
2. INSTALLING DEVELOPMENT TOOLS.....	2
Cygwin .....	2
GNU ARM GCC .....	6
Philips LPC210x FLASH Utility .....	8
The CC3 Source Tree .....	9
5. BUILDING A PROJECT.....	10
Building .....	10
Projects.....	10
Makefiles.....	11
HAL vs. Projects .....	11
Environment .....	11
3. HARDWARE SETUP.....	12
Connect the Camera Module and MMC .....	13
Connect Power and Serial Cable .....	13
4. INSTALLING FIRMWARE .....	14
ISP Button.....	14
Linux .....	17
5. TESTING FIRMWARE.....	18
CMUcam2 Firmware Testing .....	18
Camera Focusing .....	19

# 1. INTRODUCTION

The CMUcam3 is an ARM7TDMI based fully programmable embedded computer vision sensor. The main processor is the Philips LPC2106 connected to an Omnivision CMOS camera sensor module. Custom C code can be developed for the CMUcam3 using a port of the GNU toolchain along with a set of open source libraries and example programs. Executables can be flashed onto the board using the serial port with no external downloading hardware required.

The development guide will step through how to install:

- Cygwin
  - Make
  - Subversion
- ARM gcc
- LPC210x FLASH Utility

We describe the process of powering up the CMUcam3, making a new project, compiling the new project and downloading a firmware image to the board.

\* Camera Properties Depend on Camera Module

## 2. INSTALLING DEVELOPMENT TOOLS

In order to compile and download code to the CMUcam3 in Windows you will need the following software:

- Cygwin
  - Make
  - Subversion
- ARM gcc
- LPC210x FLASH Utility
- CMUcam3 Frame Grab Utility (using .NET runtime v2.0 or greater)

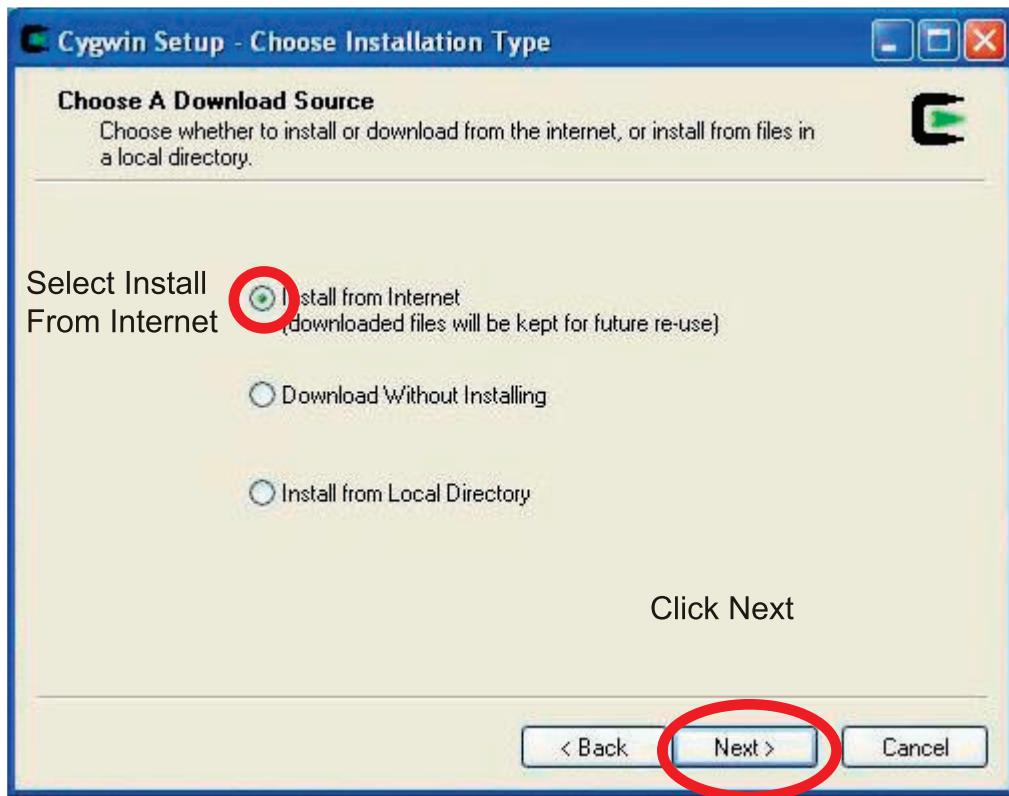
### Cygwin

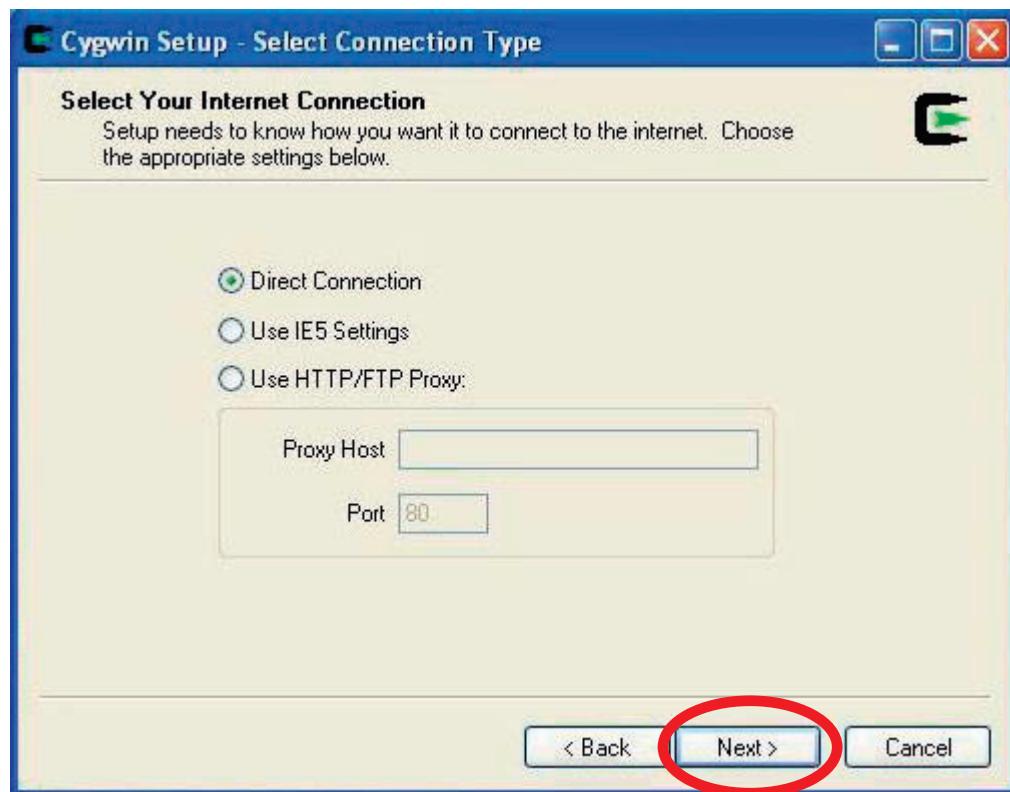
Cygwin is a Linux-like environment for Windows. It provides you with a shell and many familiar Linux tools like “make” which you will use to compile CMUcam3 code. If you are using Linux, most of these tools are already standard on your system.

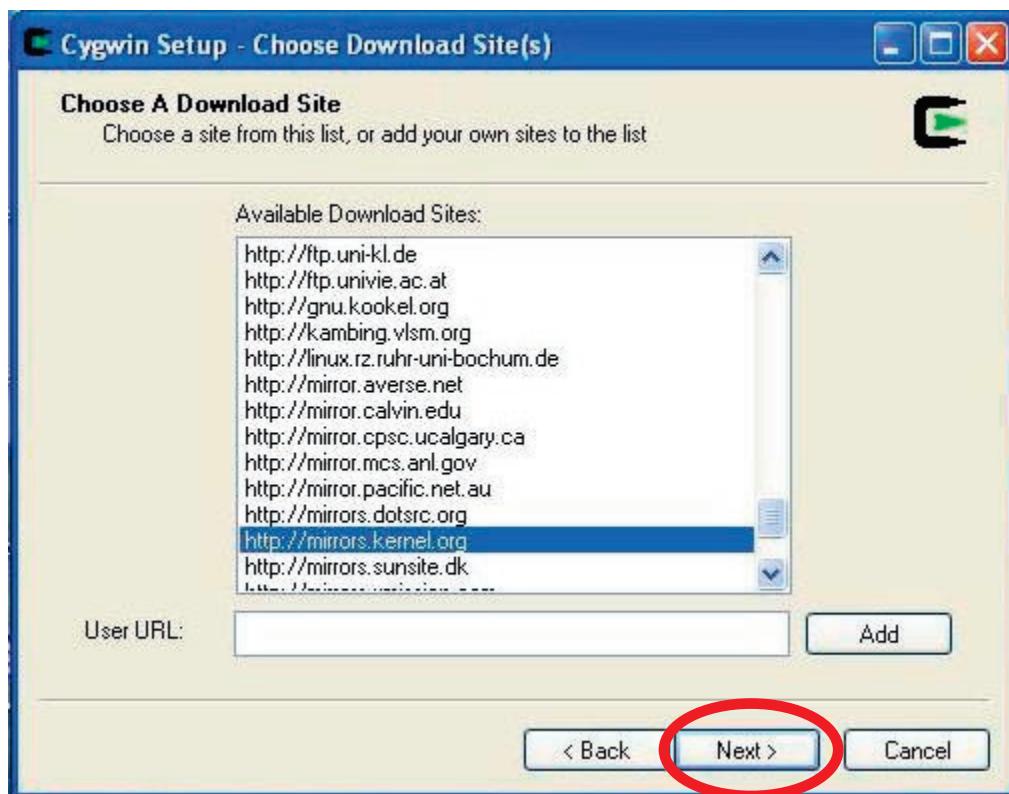
First, download the Cygwin installer from:

<http://www.cygwin.com/>

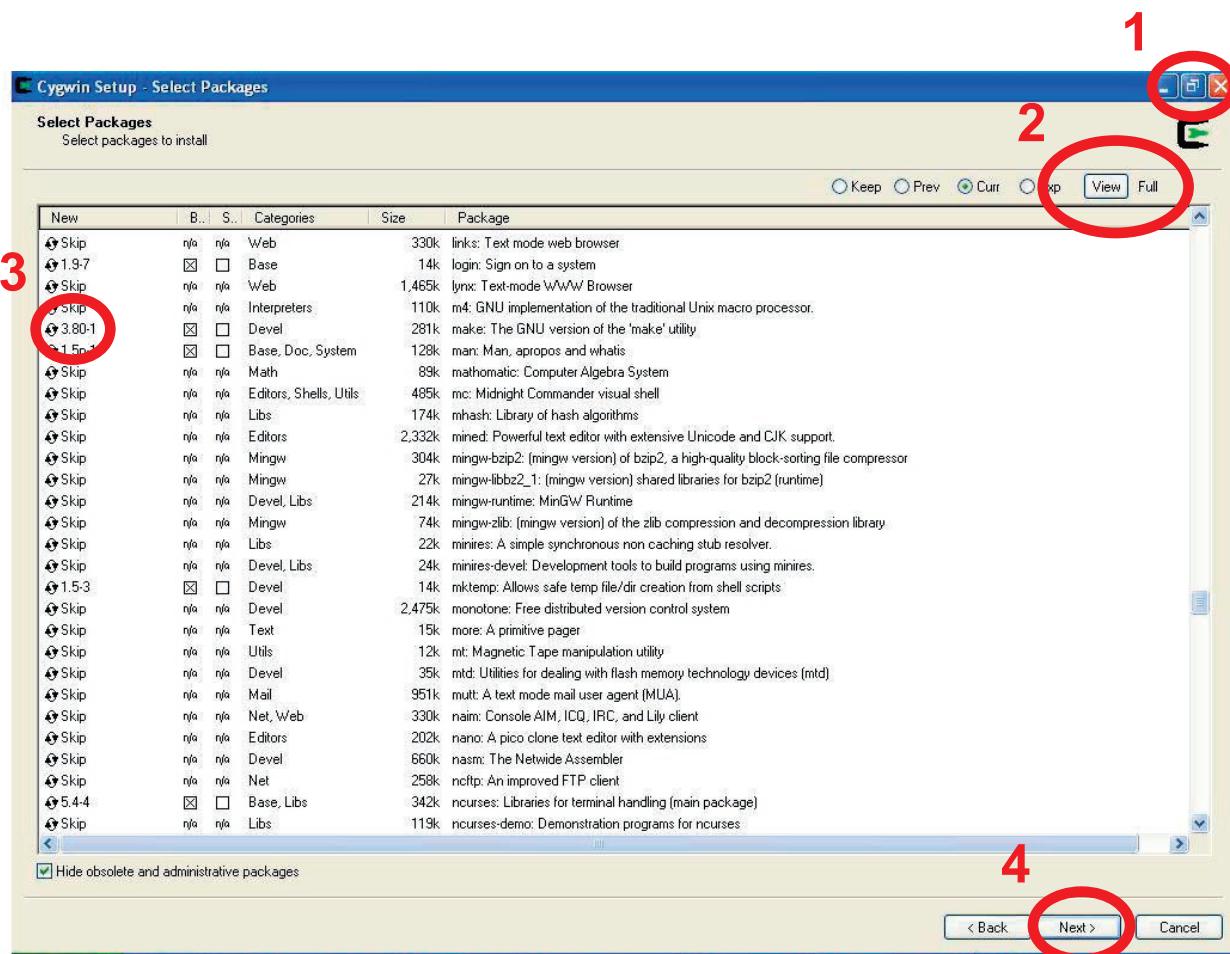
( Installer: <http://www.cygwin.com/setup.exe> )







Pick a mirror that you will use to download the rest of the cygwin software.



- 1) Maximize the Window
- 2) Click on “View” Full to see a list of all packages in alphabetical order.
- 3) Click on “skip” so that it changes into a version number to select a software package to install. Install the following extra packages:

make  
subversion  
open-ssl

- 4) Click “Next” to install. This may take about 10 minutes on a fast internet connection.

Continue clicking on next until the installation is complete.

## GNU ARM GCC

The compiler choice is left up to the hal. This allows different hardware platforms to use different compilers.

Currently, the only supported hal is “Ipc2106-cmucam3”, which uses a version of GCC optimized for the ARM processor. To build using this hal, you’ll need the arm-none-eabi-\* style GCC tools from CodeSourcery. These can be acquired here:

[http://www.codesourcery.com/gnu\\_toolchains/arm/download.html](http://www.codesourcery.com/gnu_toolchains/arm/download.html)

Get the “EABI” or “Bare Metal” target.

The screenshot shows the CodeSourcery GNU Toolchains website. At the top, there are links for C++ ABI Testsuite, QMTest, Overview, FAQ, and Downloads. On the left, there's a Mailing Lists section with Announcements and Discussion links. The main content area is titled "Select Package". It has two sections: "Host Platform" and "Target Platform". Under "Host Platform", there are four options: IA32 Windows (selected), IA32 GNU/Linux, SPARC Solaris, and Source. Under "Target Platform", there are three options: EABI (selected and circled in red), GNU/Linux, and SymbianOS. A "Download" button is located at the bottom of the "Target Platform" section.

## Installation

CodeSourcery's Windows packages use standard Windows installation packages. package, you can double-click on the file to start the interactive installation program. On GNU/Linux and Solaris, download the package and then run the following command:

Once you have downloaded the installer, double click on it and follow the rest of the on-screen instructions to finish the installation. If in doubt, select the default options.

If you feel like using a different compiler, simply change the variables in hal/lpc2106-cmucam3/defs.mk to something else. A popular thing to change is COMPILER\_PREFIX, in case you have a custom compiler built under a different name.



## Philips LPC210x FLASH Utility

1) Download and install a flash utility program. The “Philips LPC210x FLASH Utility” can be obtained from either the CMUcam (<http://www.cs.cmu.edu/~cmucam3>) or the Philips website.

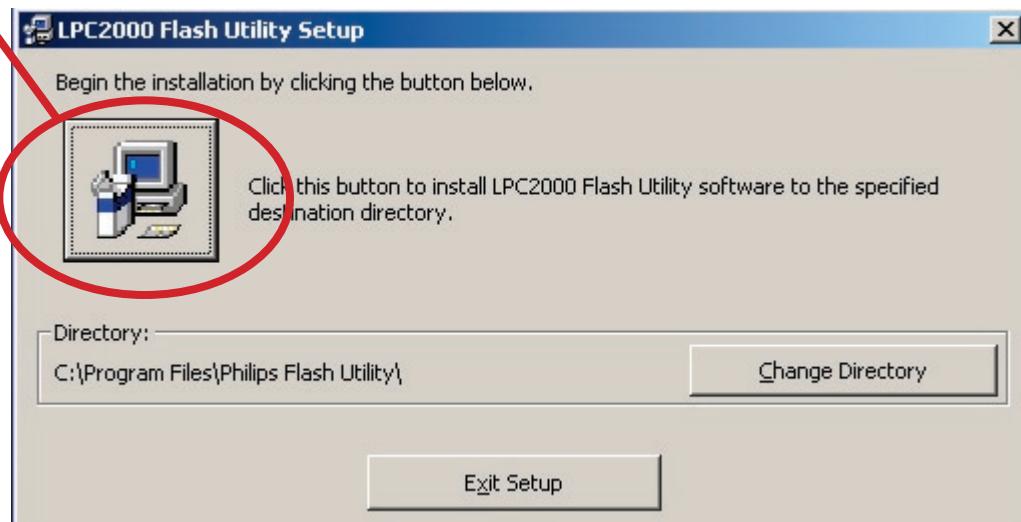
[http://www.semiconductors.philips.com/files/markets/microcontrollers/philips\\_flash\\_utility.zip](http://www.semiconductors.philips.com/files/markets/microcontrollers/philips_flash_utility.zip)  
or

<http://www.cmucam.org/wiki/Downloads>

2) Double Click on the “philips\_flash\_utility.zip” icon.

3) Next Double Click on “setup.exe”

4) Click “Ok”

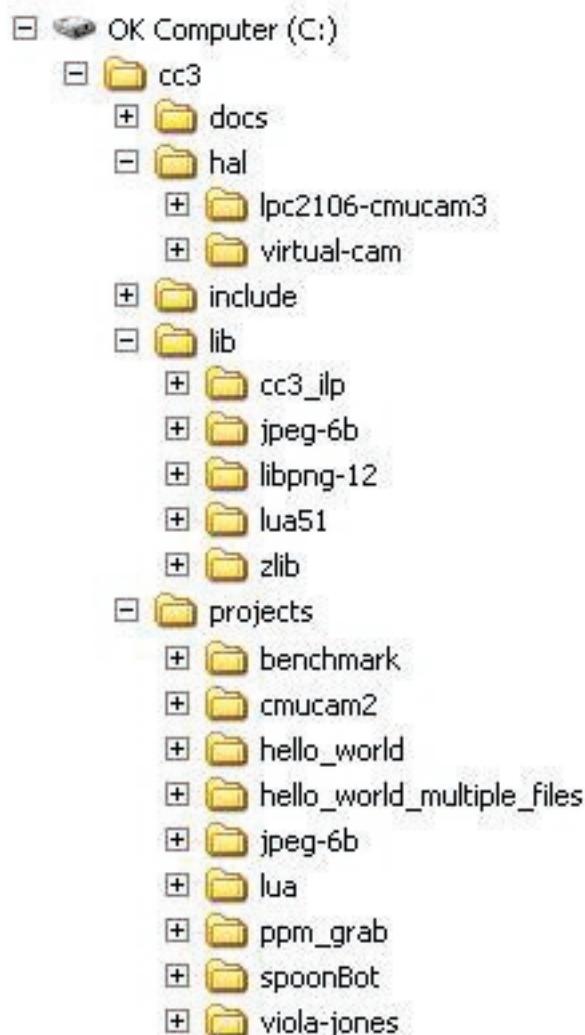


## The CC3 Source Tree

Now that you have the GNU toolchain, and the FLASH utility, you must download the CC3 source tree. This can be found on the CMUcam website or on the CMUcam sourceforge site.  
<http://www.cmucam.org/wiki/Downloads>

Unzip the directory structure in a convenient location.

The directory structure will look something like this:



## 5. BUILDING A PROJECT

### Building

Once you have the compiler and the CMUcam3 source, you need to build the system. From the top level, invoking “make” will build all the hal libraries, and will build a subset of items in the “projects” directory.

Open a Cygwin shell and type “cd c:” to move to your C drive. Next navigate to the source directory using the “cd my\_directory” command. Once in the “cc3” directory, type “make”.

Make sure you’ve built a hal before trying to build any projects!

Changing SUBDIRS in projects/Makefile will change which projects are built. Changing SUBDIRS in hal/Makefile will change which hals are built.

Invoking “make” directly from a particular project or hal directory will build that directory, regardless of SUBDIRS.

### Projects

There are a few projects in the projects directory to get you started.

The “hello\_world” project should be a nice introduction to LEDs, MMC access, servos, camera use, and serial communication.

If you want to create a project, duplicate the hello\_world directory, change the PROJECT variable in Makefile, and start editing code!

## Makefiles

When using the cc3 build system, you'll be using a Makefile in your project directory. See projects/hello\_world/Makefile for a description of how to use the file for your own projects.

## HAL vs. Projects

Generally, you will not need to make any changes to anything in hal, unless you modify or create new hardware, or if the hal you are using is missing some important functionality.

Make sure you've built a hal before trying to build any projects! If you are inside the projects directory, make does not know how to go up and build the hals. Otherwise, you will get an error like this:

```
No rule to make target `../../hal/lpc2106-cmucam3/libhal-lpc2106-cmucam3.a'
```

If this happens, go into the hal directory and run make.

## Environment

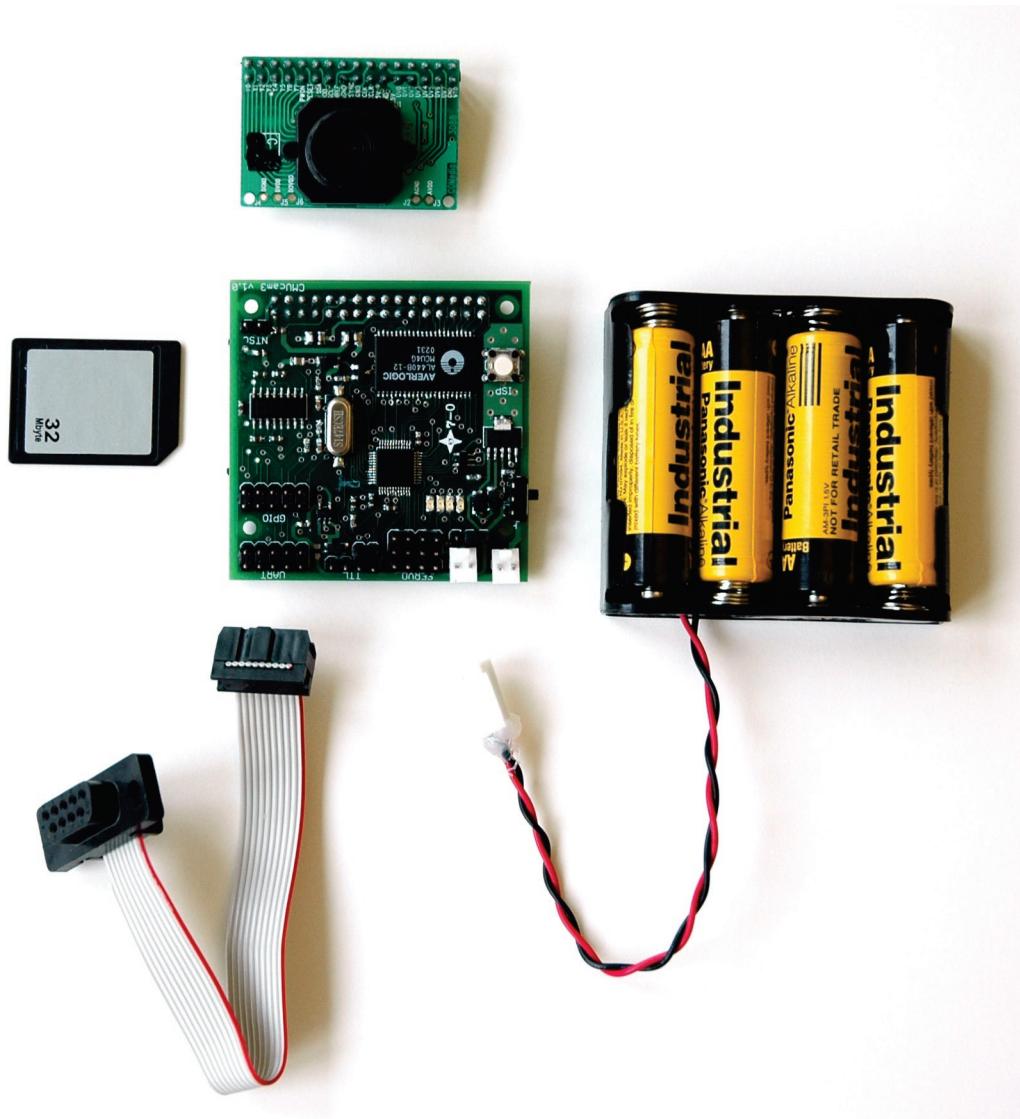
The cc3 system is build upon a standard C99 environment. For the lpc2106-cmucam3 hal, we use Newlib. Information about Newlib is here:

<http://sources.redhat.com/newlib/>

Newlib provides all the features a C programmer expects: malloc, free, printf, scanf, open, close, read, write, and friends.

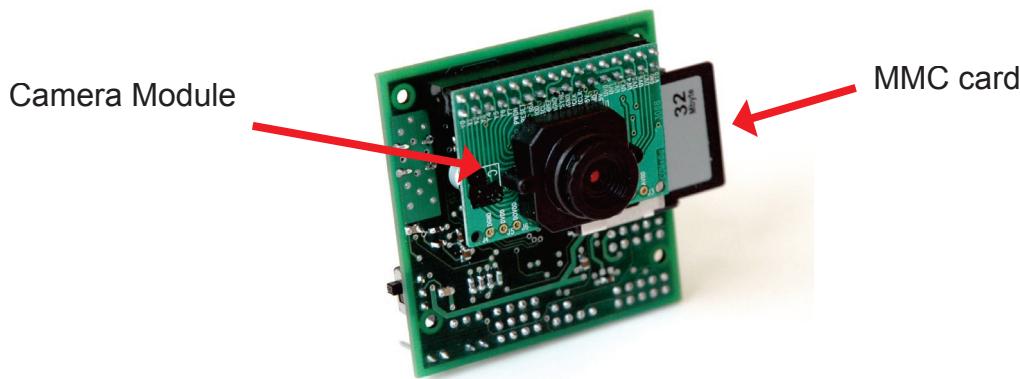
### 3. HARDWARE SETUP

This section will walk through the process of powering up and connecting your CMUcam3. The image below shows the individual components typically necessary to program the CMUcam3. The MMC card may not be included and the camera may be powered by an alternate power supply such as an AC adapter.



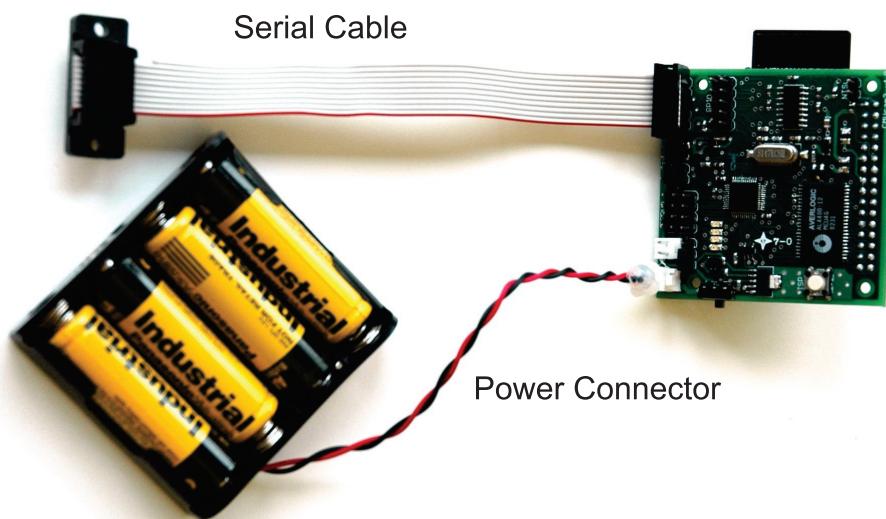
## Connect the Camera Module and MMC

Attach the camera module with the main processing board as shown below. If available, connect the MMC card.



## Connect Power and Serial Cable

Connect the serial cable and power supply as shown below. Turning on the power switch should cause the red LED to illuminate. Some serial cables may connect in the opposite orientation. If you are having difficulty communicating, try reversing the connection.

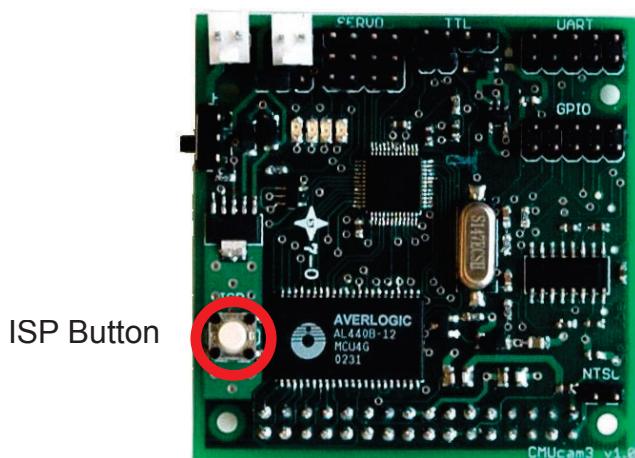


## 4. INSTALLING FIRMWARE

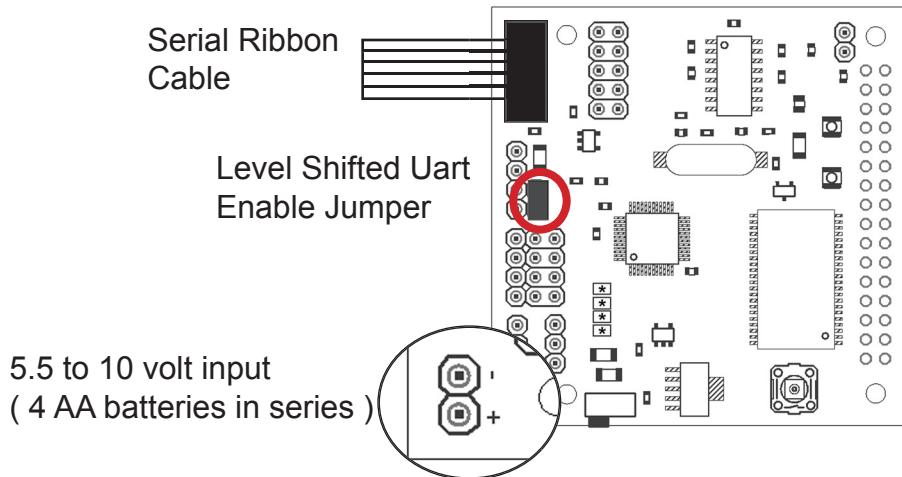
The CMUcam3 can be loaded with various firmware images. These images may perform different specific tasks. For example, the “cmucam2.hex” file will allow your CMUcam3 to emulate the functionality of the CMUcam2. When you develop your own CMUcam3 applications or would like to experiment with new firmware images, it will be necessary to flash the microprocessor. Flashing the microprocessor can be done over a standard serial port using the downloading utility described in Section 2. The next sections show step by step how to install a new firmware image onto your CMUcam3.

### ISP Button

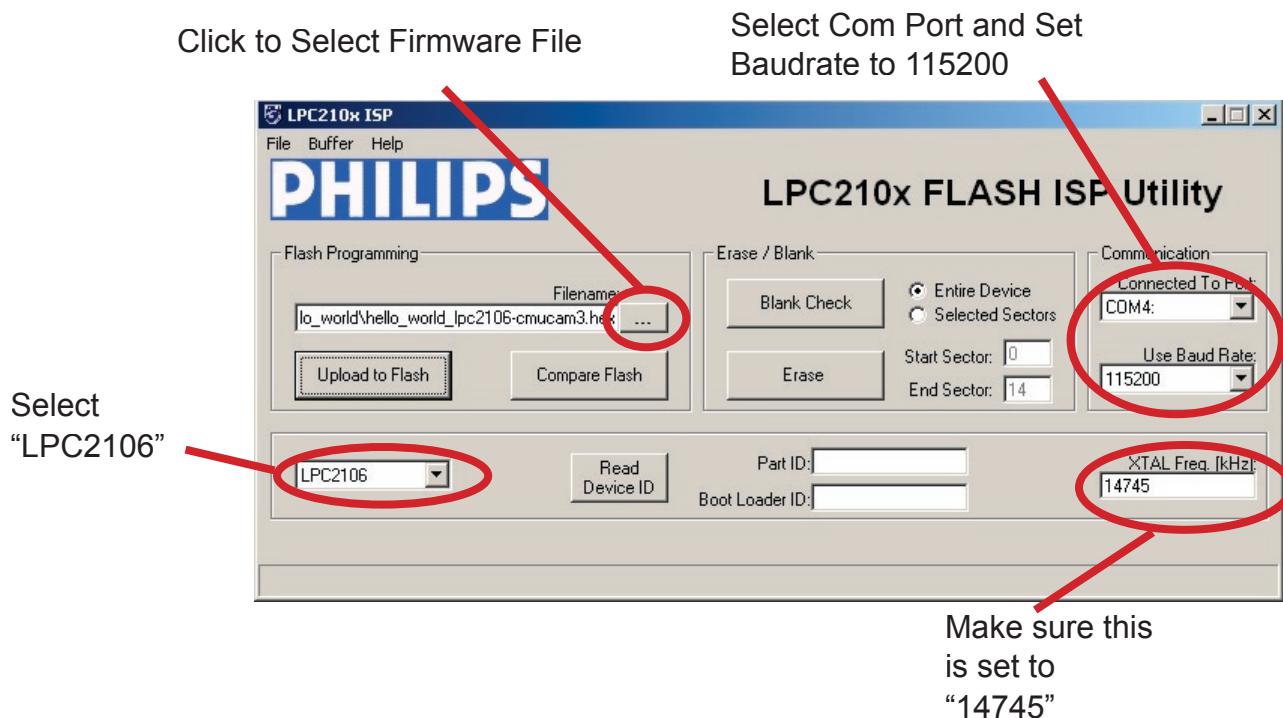
When held down on power up, the ISP button will enable the built in LPC2106 bootloader. After the processor has started up, the button can be read as normal GPIO. It is internally pulled high, and set low when depressed. The button shares the CS pin (P0.14) with the MMC. When using the MMC, CS is active low and hence the button cannot adversely affect data transfers.



6) Once the Flash Utility is installed, connect the CMUcam3 to your serial port as shown below



6) Go to the Start Menu, "All Programs", "LPC210x ISP" and execute the "LPC210x ISP" application. You should eventually see a screen similar to this:

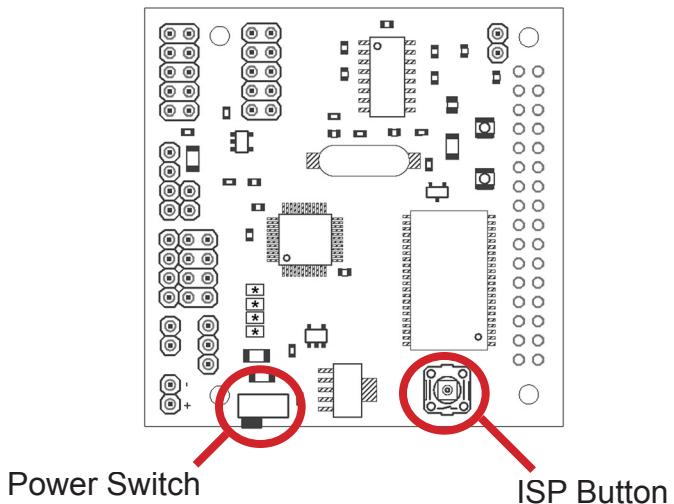


7) Click on the “...” box next to “Filename” to select a hex file to download. As an example, select the “hello\_world\_lpc2106-cmucam3.hex” file located in the CMUcam3 hello\_world project directory. Set the correct Comm Port and Device Selections.

8) Click on the “Upload to Flash” button. You should see a dialog box like below.



9) While holding down the “ISP” button on the camera, turn on the camera power. Make sure the button is held in for about a second. If the camera is already on, turn it off and on again while holding in the “ISP” button. If the button is depressed upon startup, the camera will enter its bootloader mode.



10) Press “OK” and wait for the image to download. If it fails check the connectors, reset the power and try again. The Flash Utility will say “File Upload Successfully Completed!” when the download is complete.

## Linux

In order to download to the board using linux, you must make sure you have proper access rights to the serial port. You can download a flash utility program from here:

<http://guest.engelschall.com/~martin/lpc21xx/isp/>

or

<http://www.cs.cmu.edu/~cmucam3/downloads.html>

Follow the procedure outlines in the Windows section for putting the board into its bootloader mode by holding down the ISP button while reseting power to the board.

Once installed, here is a set of sample command line arguement for downloading:

```
>: cd CMUcam3/projects/hello_world  
>: lpc21isp hello_world_lpc2106-cmucam3.hex /dev/ttyS0 115200 14745
```

## 5. TESTING FIRMWARE

### CMUcam2 Firmware Testing

This section will describe a few ways to test your CMUcam3 after you have loaded the CMUcam2 emulation firmware onto it. You should test the CMUcam3 first using a terminal program such as Minicom in Linux, Hyper Terminal in Windows, and Zterm for the Mac. Under Windows you may also use the CMUcam3 Frame Grabber Utility which acts as a lightweight terminal program as well as an image viewing utility. Please refer to the [cmucam.org](http://cmucam.org) for the latest version of the Frame Grabber Utility as well as the .NET runtime package.

#### Step 1

Inside the terminal emulator set the communication protocol to 115,200 Baud, 8 Data bits, 1 Stop bit, no parity, local echo on, no flow control and if possible turn on “add line feed” (add \n to a received \r). These setting should usually appear under “serial port” or some other similar menu option.

#### Step 2

Turn on (or reset) the CMUcam3 board; the Power LED should light up and all except for 1 status LEDs should turn on. If all LEDs remain on, there might be a problem initializing the camera. Make sure the camera module is correctly connected.

#### Step 3

You should see the following on your terminal emulator:

```
CMUcam2 v1.0 c6
```

```
:
```

If you see this, the board was able to successfully configure the camera and start the firmware.

#### Step 4

Type gv followed by the enter key. You should see the following:

```
:gv
```

```
ACK
```

```
CMUcam2 v1.0 c6
```

```
:
```

This shows the current version of the firmware. If this is successful, your computer’s serial port is also configured correctly and both transmit and receive are working.

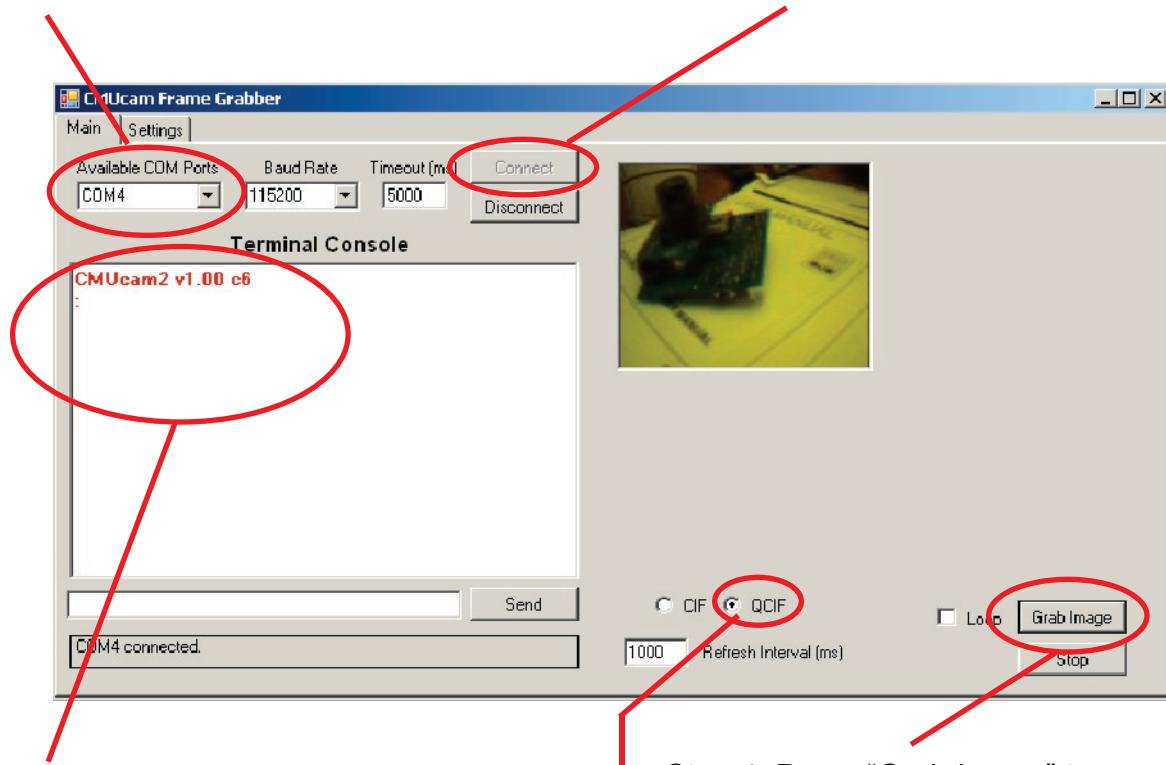
## Camera Focusing

Please refer to the CMUcam2 manual for using the CMUcam2 GUI. Under Windows, it may be easier to use the CMUcam3 Frame Grabber Utility. This can also be used as a terminal program. This allows serial transfers of JPEG compressed images for faster frame updates. You can find the CMUcam3 Frame Grabber Utility here:

<http://www.cmucam.org/wiki/Downloads>

Once you are able to capture images, rotate the front of the lens until they are in focus.

Step 1: Choose correct COM port



Step 3: Turn on Camera, or reset power.  
The CMUcam2 startup message should appear.

Step 2: Press "Connect" button

Step 4: Press "Grab Image" to see an image.

By default the camera is in low-resolution mode so the image will be scaled. Either select the "QCIF" radio box, or type "HR 1" into the text box below the startup message followed by the "Send" button to switch the camera into high resolution mode. This is a CMUcam2 function that can be found in the CMUcam2 manual.