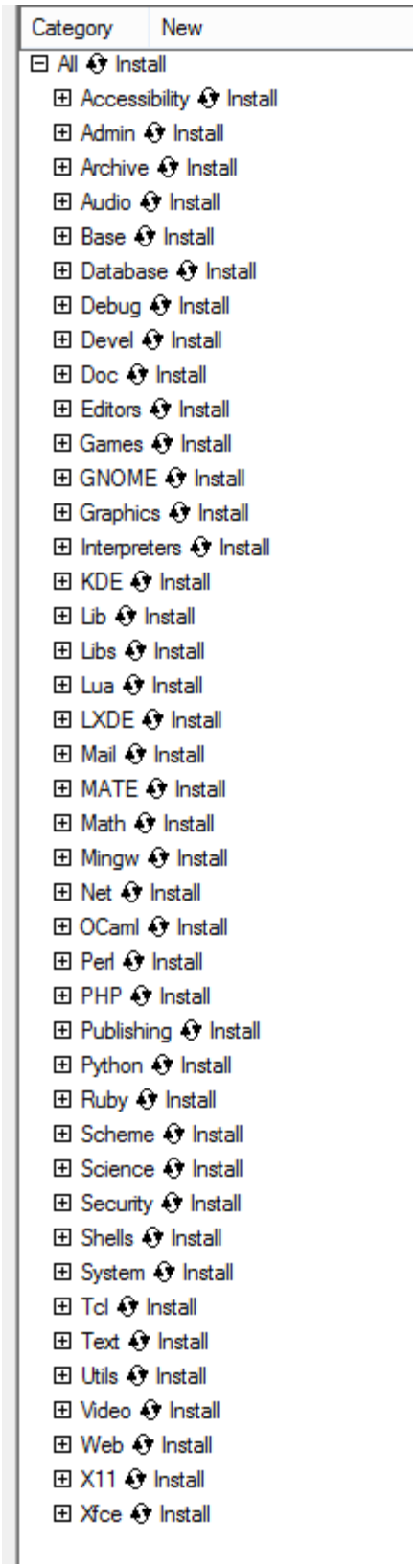
How to Install Cygwin

Cygwin is an emulator that allows to use Linux shell on Windows computer. In this project we need Cygwin to be able to compile the source code for the quad. You will use it in EE 285 too.

Go to www.cygwin.com and click “Install Cygwin” link on the left. Download and run setup-x86.exe. Click Next a few times. When you get the following window, make sure everything is selected like shown in the picture: (you can click on “Default” to change it to “Install”; ignore the errors that you get, if any)


Category
New
All Install
Accessibility Install
Admin Install
Archive Install
Audio Install
Base Install
Database Install
Debug Install
Devel Install
Doc Install
Editors Install
Games Install
GNOME Install
Graphics Install
Interpreters Install
KDE Install
Lib Install
Libs Install
Lua Install
LXDE Install
Mail Install
MATE Install
Math Install
Mingw Install
Net Install
OCaml Install
Perl Install
PHP Install
Publishing Install
Python Install
Ruby Install
Scheme Install
Science Install
Security Install
Shells Install
System Install
Tcl Install
Text Install
Utils Install
Video Install
Web Install
X11 Install
Xfce Install

This will initiate the download, which **will take a while**. After the download completes, you should be able to use Cygwin. Here is a nice guide on how to use Linux Command Prompt: http://linuxcommand.org/

Downloading/Installing Eclipse

Download Eclipse from here. Eclipse doesn't have an installer (apparently), so it comes archived in a .zip file. Unzip the file to the directory of your choice and start Eclipse. It will ask you where you want your workspace to be at. A workspace is where the code for the projects you open in Eclipse will reside. You're done! You can now start using Eclipse.

We will need a couple of add-ons for Eclipse (a Python add-on and a C/C++ add-on). Installing those should be fairly straightforward since Eclipse has an add-on manager. In Eclipse click Help -> Install New Software. Click Add, type in a random name. In the "Location" prompt, paste the following: **http://download.eclipse.org/tools/cdt/releases/8.6** and hit Ok. Check both checkboxes and hit finish. Eclipse will give you a couple of license agreement windows. You know what to do with them.

The procedure for installing the Python add-on is remarkably similar. Instead of the link provided above, copy-paste this: **http://pydev.org/updates.** Again, check both checkboxes and agree to all license agreements. You're done! Before you can compile Python code, you will need to install Python on your PC. You can get it from https://www.python.org/ . You will also need to do this .

Now you can create C/C++ or Python projects in Eclipse. Let's go ahead and import all of our source code into Eclipse. Click File -> New -> Project -> General -> Project and give it a name. Right click on the folder in your Project Explorer and click Import -> General -> File System. Navigate to the folder containing the source code for the quad and hit Finish. You're good to go.

Note: You probably won't be able to [easily] compile the C code in Eclipse. Use the method below to compile the code.

How to Compile the Code

You will need to download the source code (if you haven't already) at this link. Check this link for some software that you will need (the first two links on the page; you will need both of those). Once you've installed the software, open **Cygwin**. In Cygwin, navigate to the folder where you have the source code (the folder that contains a file named **Makefile**). Execute the following command:

```
JUSTICE@FEARLAPTOP ~/CrazyFlie_Firmware
$ make
```

You should see object files being generated:

```
CLEAN_VERSION
AS    startup_stm32f10x_md.o
CC    list.o
CC    tasks.o
CC    queue.o
CC    timers.o
CC    heap_4.o
CC    port.o
CC    misc.o
CC    stm32f10x_adc.o
CC    stm32f10x_dbgmcu.o
CC    stm32f10x_dma.o
CC    stm32f10x_exti.o
CC    stm32f10x_flash.o
CC    stm32f10x_gpio.o
CC    stm32f10x_i2c.o
CC    stm32f10x_iwdg.o
CC    stm32f10x_rcc.o
CC    stm32f10x_spi.o
CC    stm32f10x_tim.o
CC    stm32f10x_usart.o
CC    main.o
```

Once the compilation finishes, you should see a file named **cflie.bin** in the folder where your source code is. This is the binary file that you can flash onto the quad.

Note: If you wish to recompile the code, you will need to delete all of the files in the [source]/bin/dep and [source]/bin folders (leave the dep folder in [source]/bin).

Let me know if you get any errors. There may be a step in this process that I forgot about.

How to Flash Code onto the Quad

You will obviously need the quad. You will also need the Crazyradio. Once you've compiled the code, open the quad GUI that we normally use to connect to it and click Crazyflie -> Bootloader. Click Itiitiate bootloader cold boot. **You will have 10 seconds to power the quad on**, which is a bit of a challenge in our situation. Once

the connection is established, select the file you want to flash on the quad (it should be the **cflie.bin** file created by the compiler) and click Program. This fill upload the file. Once the upload is done, click Restart in Firmware Mode. You've successfully uploaded your code!