

Software:

<http://robotics.ee.uwa.edu.au/courses/des/rasp/>

Windows:

<http://robotics.ee.uwa.edu.au/courses/des/rasp/bin-win/>

FileZilla (<https://filezilla-project.org/>)

Win32DiskImager

Putty (<http://www.putty.org/>)

Mac:

<http://robotics.ee.uwa.edu.au/courses/des/rasp/bin-mac/>

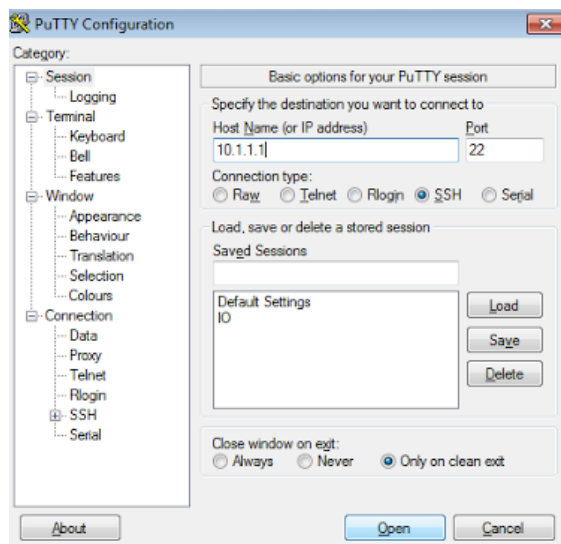
ApplePi-Baker

FileZilla

Connect to RPi SSH (Secure Shell - Remote Login):

- 1) Connect your computer to RPi Wifi network:
i.e: SSID: Rpi77a7x7f
Password: password
Password: raspberry
- 2) Connect to RPi through SSH
Default RPi IP Address: 10.1.1.1

Windows:



Write the IP Address and the Open

MAC:

Open a terminal:

Type:

`ssh pi@10.1.1.1`

3) Default settings:

User name: pi

Password: raspberry

4) Now you are connected remotely to RPi in command line.

Useful command line functions in Raspbian (RPi Operative System):

CMD	Description	Example
ls	List content of current directory	ls
cd	changes directory: Login directory Root directory	<code>cd ~</code> <code>cd /</code>
gccarm	Compile a program from C source file	<code>gccarm [file.c] -o [outputname]</code>
make	Calls a MakeFile, if any, with instructions for compiling	make
./	Executes a program	./[filename]

5) The usr files path is:

`/home/pi/usr/software`

In order to access to that directory you can use:

`cd /home/pi/usr/software`

or

`cd ~/usr/software/`

It is recommended to work in this directory since that directory is mapped in the Eyebot interface (LCD) and you can execute your programs from there too.

Connect to RPi FTP Client:

In order to create a C file, it is recommended to edit the source file in your computer and then transfer the file using the FTP, in this way you will always keep a copy of the source.

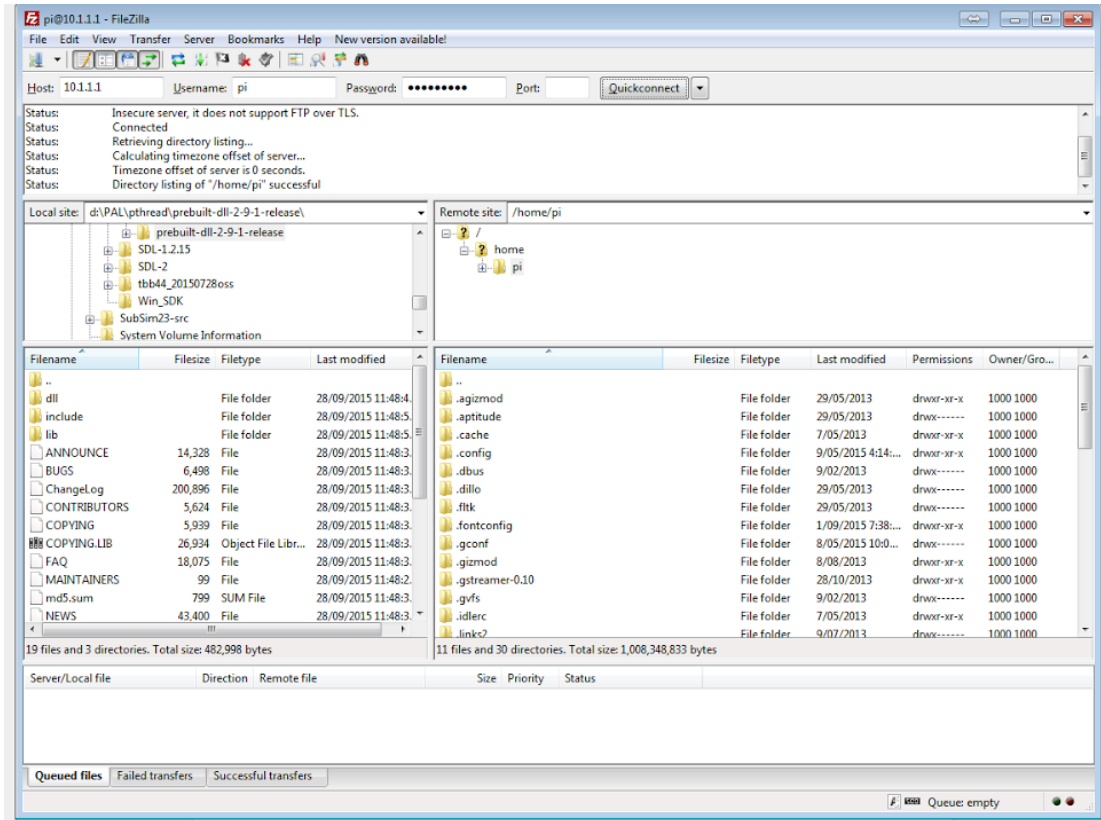
1) Open FileZilla

2) Connect to RPi using the default settings:

Host: 10.1.1.1

Username: pi

Password: raspberry



With the FTP connection you can send and receive files from the RPi. The left panel correspond to the tree of your computer and the right panel to the RPi.

Upload a file to RPi:

- 1) In the panel of the left, locate your working directory (i.e. C:\demo).
- 2) In the panel of the right, locate your RPi working directory (i.e. /home/pi/usr/software)
- 3) By double clicking to a file it is automatically uploaded or downloaded from / to your PC to / from the RPi

Hello World:

1) Write the following program in a text editor in your computer:

```
1  #include "eyebot.h"
2  // writes "Hello World" on the LCD Screen and waits for a key for exiting.
3  int main(void)
4  {
5      KEYInit();
6      LCDInit();
7      LCDClear();
8      LCDMenu("Bye", "Bye", "Bye", "Bye");
9      LCDSetPos(10,10);
10     LCDPrintf("Hello World!");
11     KEYWait(ANYKEY); //exits when any key is pressed
12     return(0);
13 }
14
```

2) Save it as a .c file i.e. HelloWorld.c

3) Transfer the file to the RPi to `~/usr/software/`

4) Compile it using gccarm

```
cd usr/software
```

```
gccarm HelloWorld.c -o HelloWorld
```

5) Execute the program from the RPi LCD Interface or from a Remote Desktop

Eyebot Libraries:

In order to use the Eyebot library include the library: "eyebot.h" in your files.

```
#include "eyebot.h"
```

C library High level:

<http://robotics.ee.uwa.edu.au/eyebot7/Robios7.html>

Low level - serial communication with IOBoard - Informative Only:

<http://robotics.ee.uwa.edu.au/eyebot7/>

Additional files:

HDT file:

```
/home/pi/eyebot/bin
```

```
hdt.txt
```

Include headers:

```
/home/pi/eyebot/include
```

Lab 6

Handling files:

```
1  #include "eyebot.h"
2  #include <stdio.h>
3
4  int main() {
5
6      FILE *fp;
7      fp = fopen("/home/pi/usr/software/output.csv", "w");
8      fprintf(fp, "starting the csv\n");
9      fclose(fp);
10
11     return 0;
12 }
13
```

Timed function:

```
1  #include "eyebot.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <time.h>
5
6
7  int i;
8  void timed_function(void) {
9      i++;
10     printf("i: %d \n",i);
11 }
12
13 int main() {
14
15     OSAttachTimer(500, &timed_function); // 1KHz clock
16     // 1000 --> 1 sec --> 1 Hz
17     // 500 --> 0.5 sec --> 2 Hz
18     while (1)
19     {
20         usleep(100);
21     }
22
23     return 0;
24 }
25
```

Hint:

Reading and writing to the IOBoard requires some time, a timed function executed at a 20 - 30Hz should work fine when reading the encoder and driving a motor.

Encoder / Motor Functions:

```
1  #include "eyebot.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <time.h>
5
6  int main() {
7      int encoder_m1;
8
9      MOTORInit();
10     encoder_m1 = ENCODERRead(1);
11     MOTORDriveRaw(1, 100);
12
13     return 0;
14 }
15
```

Hint:

Max Speed:

10060 ticks / second

LCD Plot:

LCD: 320 x 240

```
1 //Author: Marcus Pham
2 #include "eyebot.h"
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <time.h>
6
7 int randomArray(int* arr) {
8     srand(time(NULL));
9     for(int i =0; i<320; i++) {
10         arr[i] = rand()%100 +1;
11     }
12     return 0;
13 }
14
15 int main() {
16     int arr[320];
17     randomArray(arr);
18     LCDInit();
19     usleep(1000);
20     LCDClear();
21     for(int i=0; i<320; i++) {
22         LCDPixel(i, 50+arr[i], WHITE);
23     }
24
25     sleep(10);
26     return 0;
27 }
28
```