CONTROL OF
AUTONOMOUS FLIGHT OF A QUADCOPTER
USING CRIUS ALL-IN-ONE PRO V2

by

MAYUKA SRINIVASAN

A project report submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2014

ABSTRACT

In this project an attempt has been made to edit the arducopter code to autonomously take-off, hover at a known altitude, detect an IR beacon using an on-board wii-mote camera and land if the beacon is not detected.

This involved changing the arducopter code to function without the use of GPS signals and to write libraries for Wii-mote camera and integrate it to the arducopter code. This report gives an explanation of the changes made to the code to achieve autonomous flight in an indoor environment, the extra components that were added to the Quadcopter apart from the one's mentioned in the user guide and the results obtained in the testing phase. The report ends with some of the future work that can be done using this progress.

Table of Contents

TABLE OF FIGURES..... II

LIST OF COMPONENTS III

CHAPTER 1 : BACKGROUND AND INTRODUCTION.....1

CHAPTER 2 : MATERIALS AND METHODS 2

HARDWARE.....2
X-BEE
WII-MOTE CAMERA BREAK-OUT BOARD
IR-BEACON

SOFTWARE.....5
ARDUCOPTER CODE
PROGRAMMING THE TRANSMITTER
BINDING THE TRANSMITTER AND RECEIVER
FLIGHT TEST AND PID TUNING IN THE MANUAL MODE

MANIPULTING THE ARDUCOPTER CODE FOR AUTONOMOUS FLIGHT.....9

INTEGRATING WII-MOTE CAMERA TO THE ARDUCOPTER CODE.....11

CHAPTER 3: CONCLUSION AND FUTURE WORK.....14

CHAPTER 4:REFERENCES.....15

LIST OF FIGURES

Figure 1: X-BEE communication block diagram 2

Figure 2: Setting destination address on X-CTU 3

Figure 3: X-BEE pins on Crius board 3

Figure 4:Wii-mote camera connection to Crius board 4

Figure 5: IR beacon when OFF and ON 4

Figure 6:Arducopter code on arduino 5

Figure 7:List of libraries 5

Figure 8:Mission Planner 6

Figure 9:Connecting Radio to PC 6

Figure 10: Selecting the correct serial port 7

Figure 11: Software when communicating and not-communicating with radio 7

Figure 12: BAT port on receiver 8

Figure 13:Changing flight modes in Manual Mode using Mission Planner 9

Figure 14:P vision.h and Wii-camera libraries in arduino library folder 11

Figure 15:Code snippets for Wii-camera navigation 12

Figure 16:Pictorial representation of a camera detecting the beacon 12

Figure 17:Mission Planner graphical outputs for the camera 13

LIST OF COMPONENTS TO BUILD AN ST-360 QUADCOPTER

1. ST-360 FRAME
2. 4 PROPELLERS(2 PUSHER,2 NORMAL)
3. POWER DISTRIBUTION BOARD
4. ST-2210 BRUSHLESS MOTORS (4)
5. 2200 mAH,11.1V BATTERY
6. ELECTRONIC SPEED CONTROLLERS (4)
7. SERIES-2 X-BEE's (2)
8. CRIUS All-in-One Pro V2 BOARD
9. NINENTDO-WII REMOTE CONTROLLER CAMERA & BREAK-OUT BOARD
10. FLY-SKY 6-CHANNEL TRANSMITTER AND RECEIVER

CHAPTER 1: BACKGROUND AND INTRODUCTION

This project is an extension of the work done by Cory Engel, a Graduate student from University of North Carolina at Charlotte and Audrow Nash, an Undergraduate student from University of North Carolina at Charlotte. Audrow developed an algorithm to detect and track an IR beacon using a Wii mote camera. Cory's work involved the construction of the Quadcopter and to write an algorithm to maintain a given altitude and test both the algorithms on by sending inputs to Crius all-in-one pro V2 board (that controls and maintains Quadcopter flight in the Manual mode) from a pre-programmed Red Board.

In this project, an attempt has been made to eliminate the use of Red Board to control the Quadcopter and use only the Crius all-in-one pro V2 board to achieve an autonomous flight and track the IR beacon.

Research and testing of control for an Autonomous flight of a Quadcopter in an indoor environment has increased recently. Several solutions have been proposed, to achieve an autonomous flight in the absence of a GPS signal. Wii mote camera has become an interesting tool for hacking and is a great sensor for tracking infrared sources.

In this project, we describe the control of a Quadcopter and autonomously track an IR beacon using an on-board Wii-mote camera. Our aim was to develop an unmanned aerial vehicle to autonomously take-off, detect an IR beacon, follow it in an indoor environment and land if the signal from an IR beacon is lost or in a case when the IR beacon is not detected.

In order to get the complete system working, the project was divided into two major parts. The first part was to manipulate the existing arducopter code to achieve autonomous flight in an indoor environment. The second part involved the creation of a library to detect IR beacons using a Wii mote camera and including it to the arducopter code. Each of these parts needed periodic testing and error correction. The key contribution of this project, is the creation of a Wii-camera library for arducopter and achieve an autonomous flight and tracking eliminating the use of a Red Board.This project focuses on achieving an autonomous flight and tracking by manipulating the arducopter code on the Crius board by solely relying on Sonar, since the systems cannot rely on GPS (Global Positioning system) lock in indoor environments.

In this report the first few sections gives a brief explanation of the background, construction, initial setup and testing in manual modes. The next few sections describe the arducopter code manipulation and changes made to the existing code to attain autonomous flight. The following sections describe the integration of the library code to the arducopter code and the tests conducted to check it's working. The report concludes with the tests, results and future work using the current results.

CHAPTER 2: MATERIALS AND METHODS

HARDWARE

A quadcopter frame was built using all the materials mentioned in the List of components to build an ST-360 Quadcopter mentioned above and the guide book written by Cory. The steps for integrating an X-BEE and a Wii mote camera to the Crius board and circuit for an IR beacon are not included in the guide book. These steps have been explained here under.

X-BEE:

The X-BEE's are used to establish a wireless connection between the Mission Planner and the Quadcopter while in flight. It is this way, easier to control the Quad, monitor various functionalities and to adjust the PID values and tune the Quadcopter while in air.

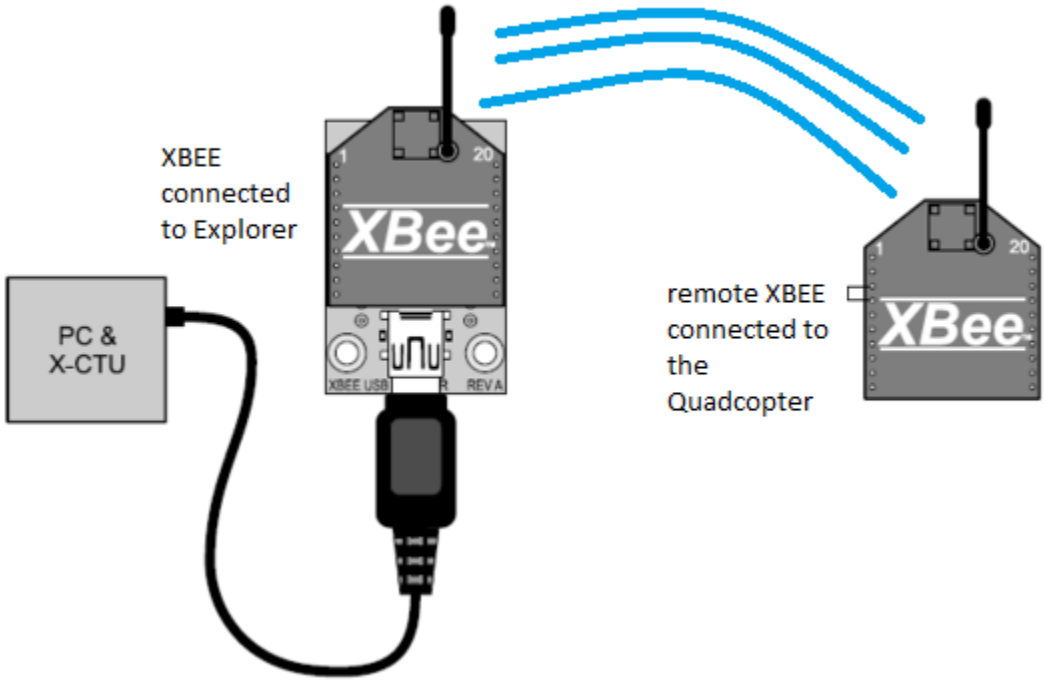

Fig.1: XBEE communication block diagram[2]

1. Install X-ctu from the below given link.

http://www.digi.com/support/productdetail?pid=3352&osvid=57&type=utilities

2. Open X-ctu and using an X-Bee explorer, hard code the **destination address** between two series-2 X-bee's, assign them with the same **pan Id** and set the baud rate to **56700** and program the X-bee's to begin communication as shown in the pictures below.

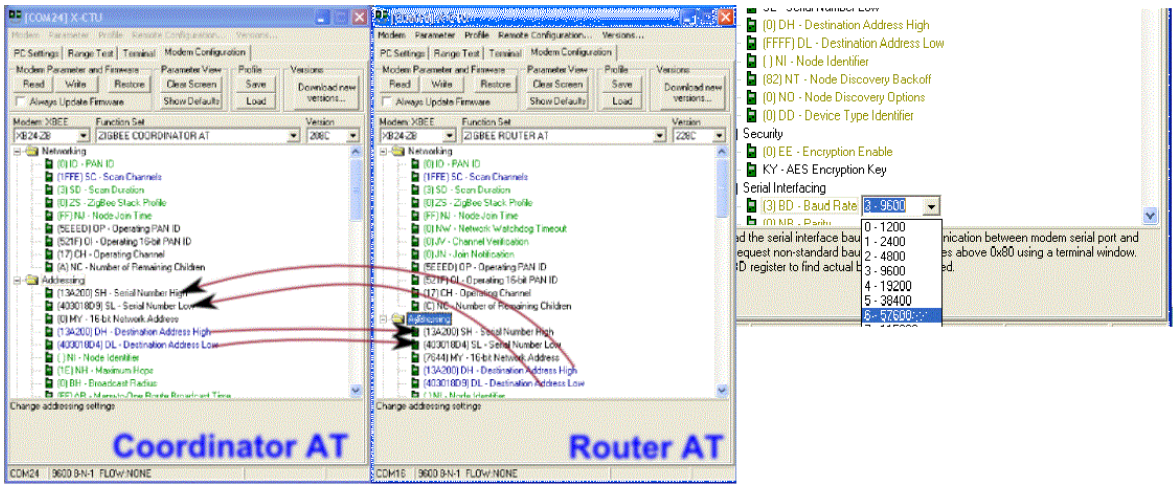

Fig.2: setting destination address and baudrate in x-ctu[2]

3. Connect the X-bee to the highlighted pins on Crius board, as shown in the figure using the connectors that came along with the board.

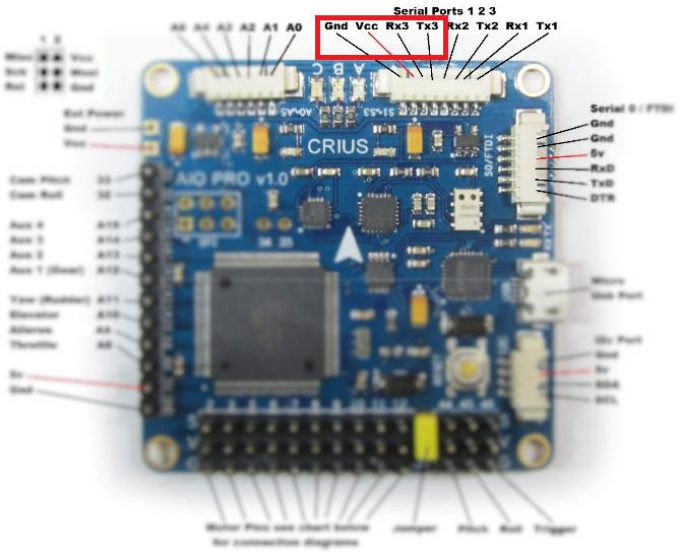

Fig.3: X-Bee pins on Crius board.

4. One X-Bee will be connected to the Pc using an explorer and the other is connected to the Quadcopter.

Wii- Mote Camera Break-out board:

The Wii-mote camera is basically a Nintendo Wii remote controller. It uses a Pixart sensor to detect up to four blobs (ir sources). This can be used to identify the direction of rotation and position of the beacon. The details of the working will be explained in the section integration of Wii-mote camera to arducopter.

1. Using the connectors that came along with the Crius all-in-one pro V2 board, connect the Wii-mote camera break-out board and the Crius board as shown in the figure below.

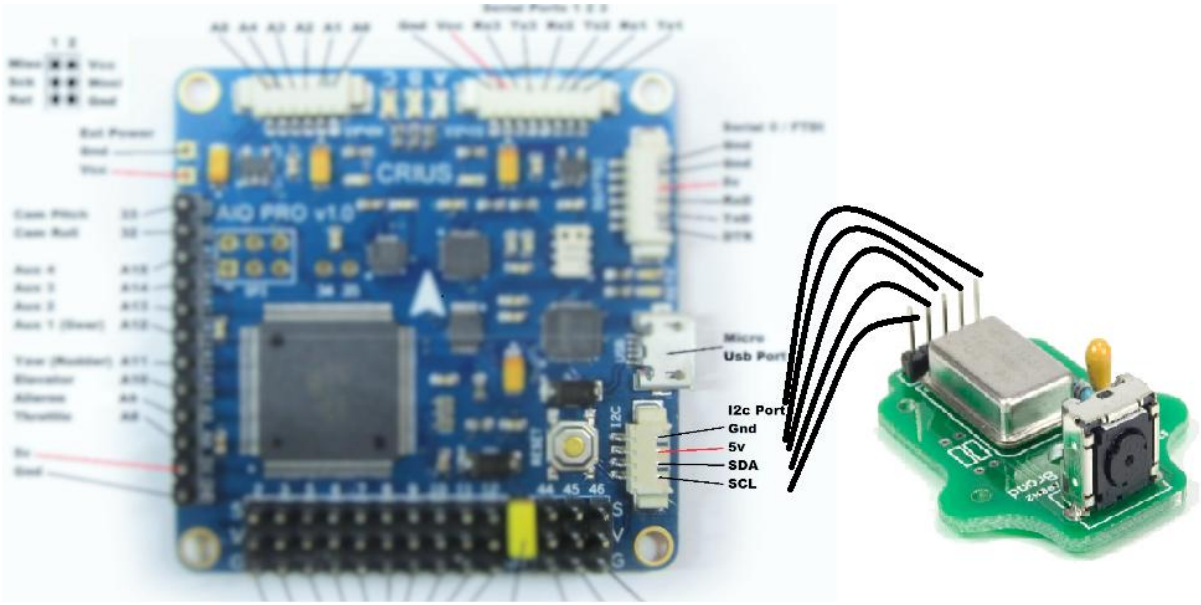

Fig.4: Wii-mote camera connection to Crius board.

IR Beacon:

An IR beacon is a set of IR LEDs that are connected in series to form a bar, that can be switched ON and OFF using a switch and is powered using three AA batteries. The only specification for constructing a beacon is that, it should have three or four IR LEDs in a straight line. The Beacon that was used for this project is as given below.

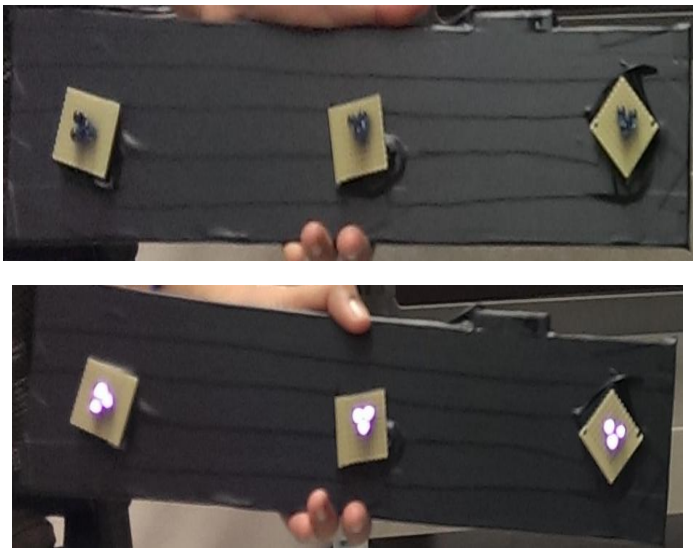

Fig.5: IR Beacon when OFF and ON

SOFTWARE

Arducopter code :

The arducopter code is an open source autopilot system. The code is made up of the main ArduCopter code which resides in its own directory, and the libraries which are shared with Arduino.

All the tutorials and explanation of the code is given in http://dev.ardupilot.com/. The code also includes Mavlink coding that is used as scripting language for the Mission Planner.

Any changes made to the arducopter code or the mission planner will make changes to the other by default.

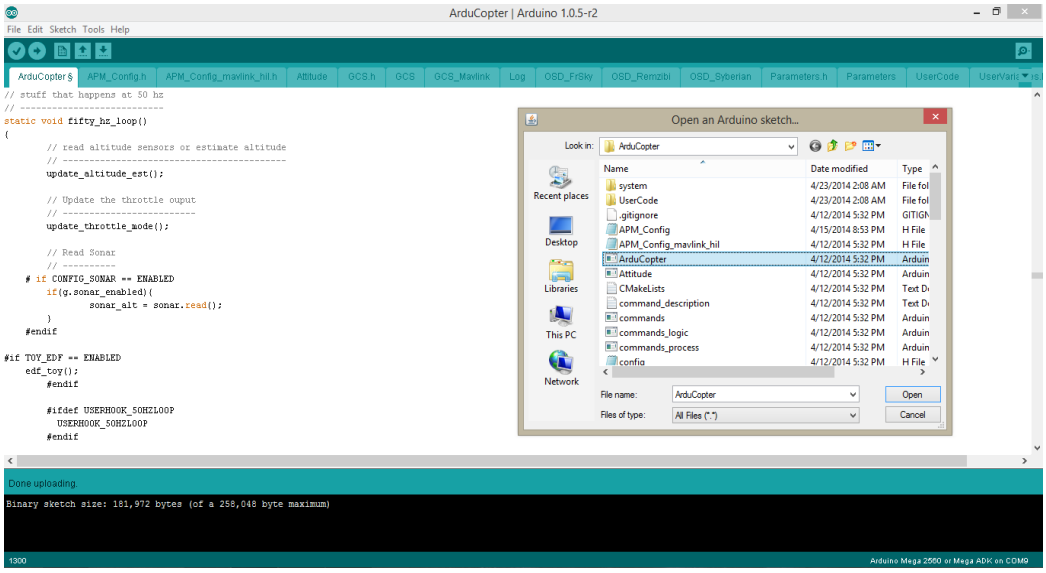

Fig.6: Arducopter code on arduino.

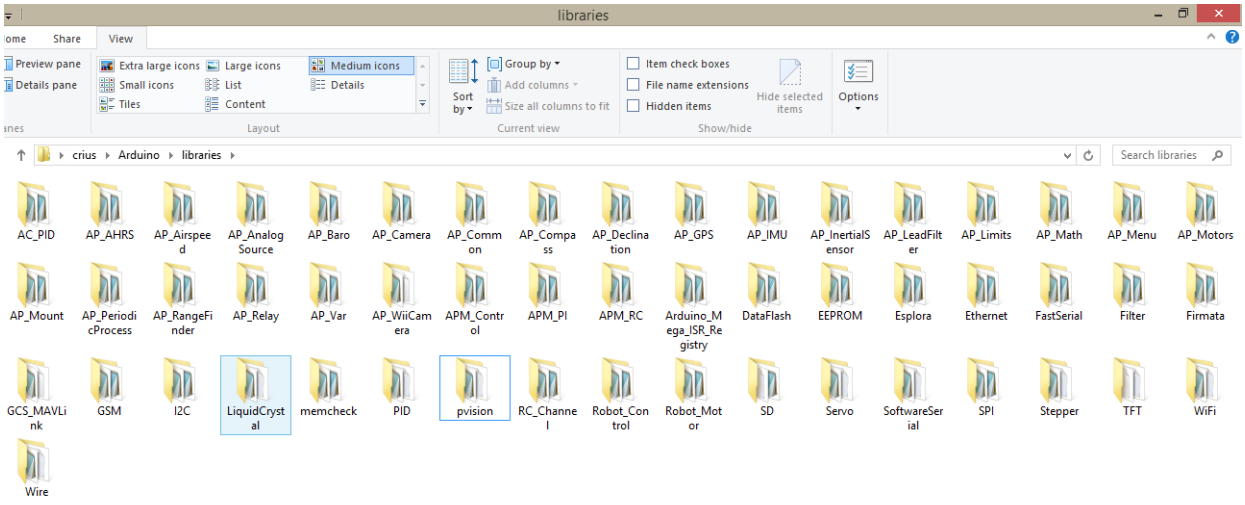

Fig.7: List of libraries.

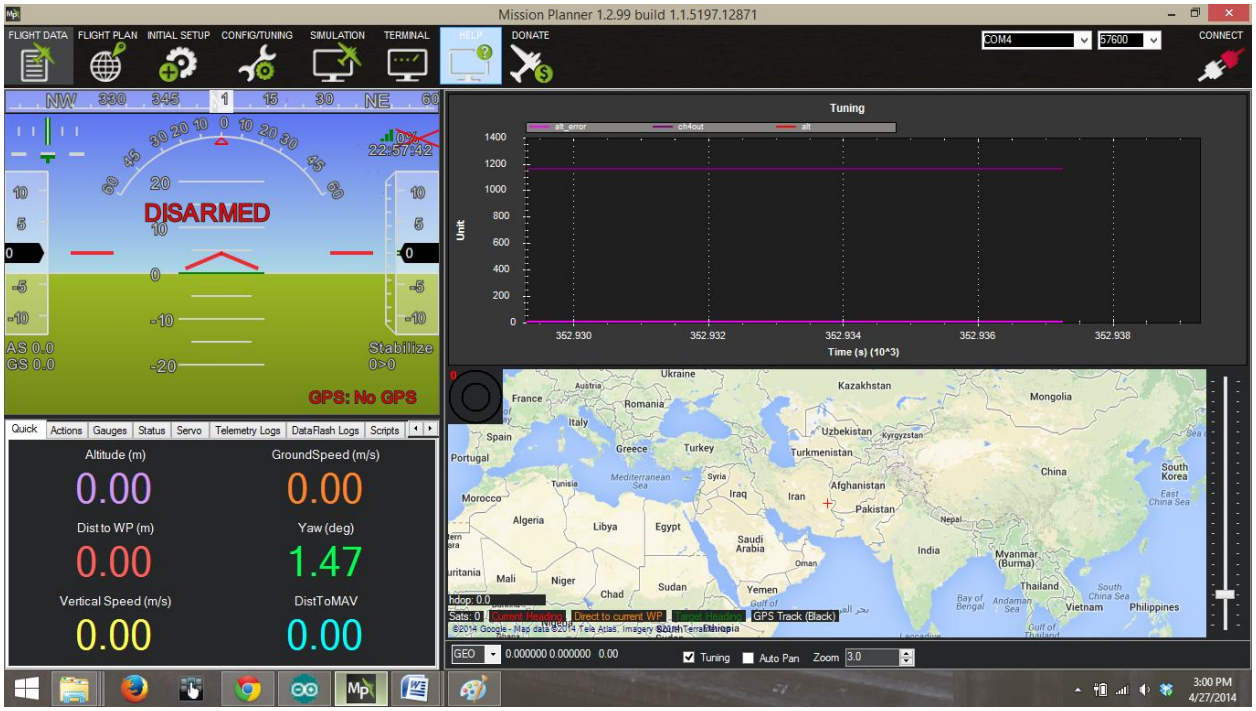

Fig.8: Mission Planner

Understanding the code usually takes a little time. Each of the libraries has a specific function and is called in the main code and is used to control the Quadcopter.

Programming the Transmitter/Receiver:

The following steps explain the procedure to program the 6 channel FS-CT6B fly-sky transmitter and to use the channel 5 (Aux-1) to change flight modes. The transmitter is user configurable and all the channels need to be programmed using the instructions given below before starting to use it.

1. Connect the transmitter to your PC using the USB cable that comes along with the kit.

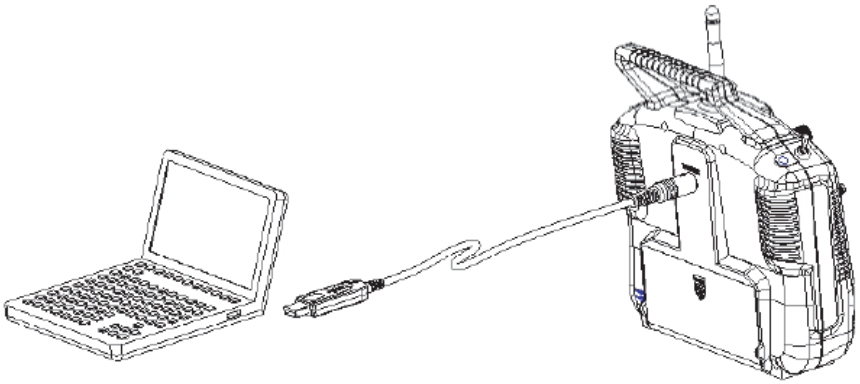

Fig.9: Connecting the radio to PC[11]

2. Install the drivers for the USB cable. The driver installs USB-UART bridge software that helps in communicating between the PC and the transmitter.

3. The drivers can be found at http://www.mycoolheli.com/t6config.html

4. Check for the COM port that appears as a serial port in PC settings.

5. Install T6 config. a programming software, from Radio sub-heading mentioned in the link http://www.mycoolheli.com/

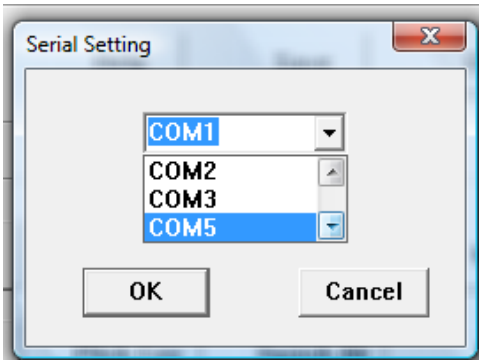

Fig.10: Selecting the correct serial port[11]

6. Click open T6 config. and set the COM port to the serial port that was installed using the settings Tab.

7. Check if the transmitter is communicating. The bars on the screen move when the sticks on the transmitter are moved.

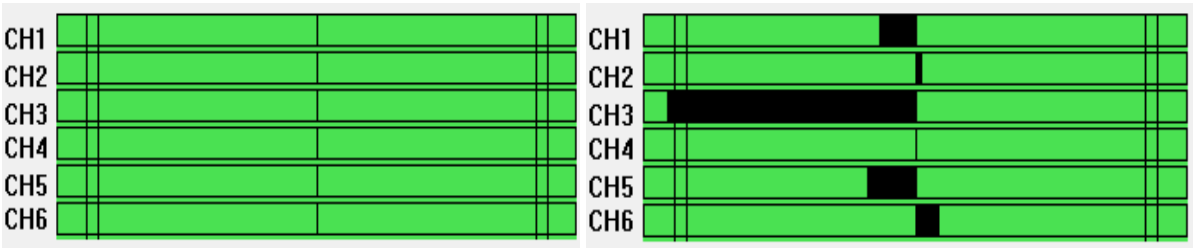

Fig.11: The software when not-communicating and communicating with the radio[11]

6. Click on **Get User**, after establishing communication with the radio. This forces the T6Config.exe software to clear its memory and read the real settings from the radio back into the software. Failure to do this may cause issues or loss of configuration information when you make changes or save the information to a file.

7. Select **Mode**, and click on Mode 2, to set channel 3 as throttle, channel 4 as Yaw, and Channel 1 and 2 as Roll and Pitch respectively.

8. Click on **Mix**, to change tune channel 5 on the radio.

9. Select **Mix 1**, source as CH5, Destination as VRA knob on the radio and Up and Down rate as 100% and finally select SWA and click OK. This enables the knob VRA on the transmitter to change flight modes when the Switch A is ON.

Binding the transmitter and receiver:

1.Install the battery to the transmitter, and turn it off.

2.Insert the binding plug into the BAT port of the receiver

Fig.12: BAT port on receiver[11]

3.Connect the ESC to Channel 3, Power up the receiver using connecting the main battery to the ESC. The LEDs should start to flash.

4.Press and hold the lower left button on the transmitter, and then switch on the transmitter's power switch.

5.Observe the LED lights on the receiver. Once the LEDs stop flashing, the receiver is bound to the transmitter. It will take about 10 seconds (or less) for the binding process to complete.

6. Release the match button on the transmitter. Remove power from the receiver, and turn off the transmitter.

7. Remove the receiver binding plug.

8. Test by turning on the receiver without pressing the match/bind switch. Power the receiver, and the LEDs should light steady meaning it is bound to the transmitter.

9. If the test failed, repeat this process.

Flight test and PID tuning in the Manual Mode:

Once the transmitter is programmed and all the other parts of the Quadcopter are connected, prepare for first flight tests and tune the PID values for a stable flight in the manual mode using the how-to-guide written by Cory.

MANIPULATING THE ARDUCOPTER CODE TO ACHIEVE AUTONOMOUS FLIGHT

To attain autonomous flight directly from the Crius all-in-one pro V2 board, the arducopter code needs to be edited appropriately use Sonar to attain autonomous flight instead of the GPS signals.
Before editing the code, the flight modes are set to perform take-off, hold a given altitude and land in manual mode.

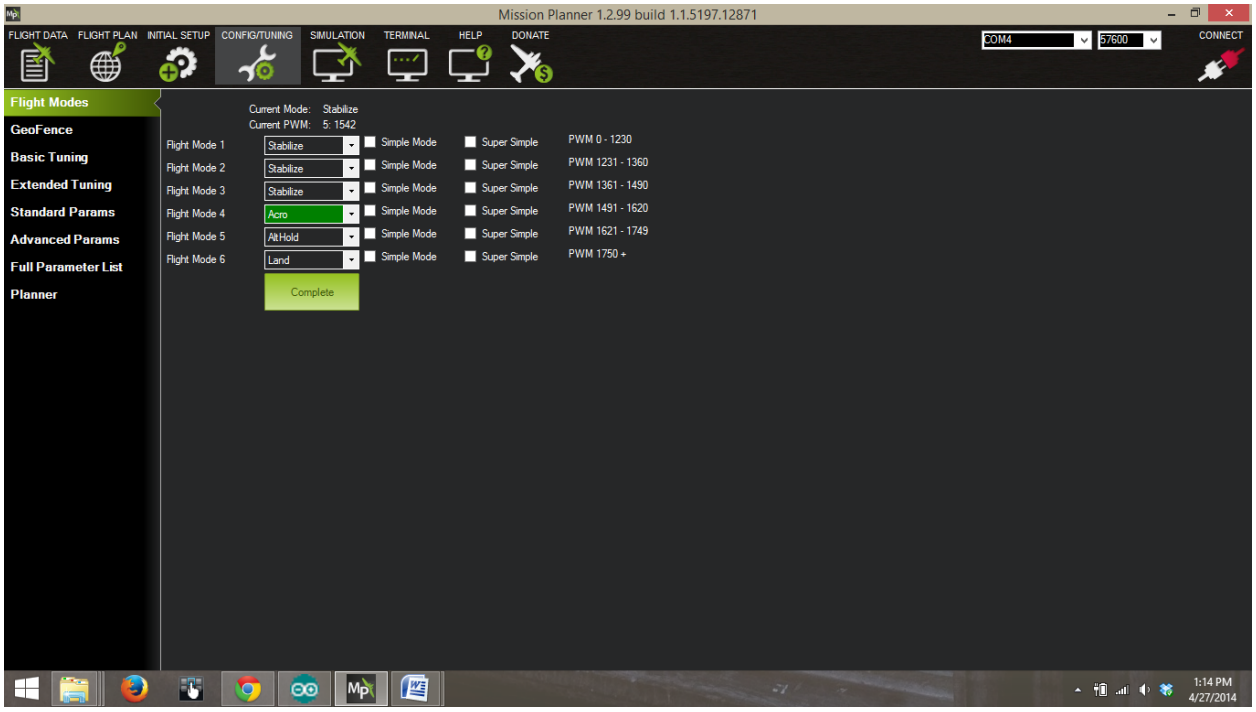

fig.13: Changing Flight modes in Manual Mode using Mission Planner

The flight modes are changed as shown in figure. Once the flight modes are set, test the working in Manual mode using radio and the Channel 5 knob that was programmed earlier to change modes. Here, **Acro** is set for Takeoff, **Alt Hold** maintains altitude and **Land** mode lands the Quadcopter smoothly.

The following changes were made to system.pde for autonomous take-off, hover and land.

```
switch(control_mode)
        {
                case ACRO:
                   yaw_mode              = YAW_HOLD;
                  roll_pitch_mode        = ROLL_PITCH_STABLE;
                  throttle_mode          = THROTTLE_AUTO;
                  // set_next_WP(&current_loc);
                  set_new_altitude(130);
                  do_takeoff();                                   // autonomous take off
```

```
            //circle_angle    = 0;
            break;

        case ALT_HOLD:
            yaw_mode                = ALT_HOLD_YAW;
            roll_pitch_mode = ALT_HOLD_RP;
            throttle_mode  = ALT_HOLD_THR;


            force_new_altitude(max(current_loc.alt, 100));
            break;

        case LAND:
            yaw_mode = LOITER_YAW;
            roll_pitch_mode = LOITER_RP;
            throttle_mode  = THROTTLE_AUTO;
            do_land();                                  // autonomous land
            break;
}
```

Changes made to arducopter.pde are as given below:

```
void autonomous_flight()
{
counter++;     //a counter flag to wait and see if it determines an IR beacon
if (counter <100)
{
set_mode(ACRO);                                                 // for autonomous take-off
}
 if (100<counter <300)
{
set_mode(ALT_HOLD);                         //when beacon is found the altitude is maintained

}
if(100<counter<600)
{
set_mode(LAND);    //when beacon is not found the Quadcopter lands
}
```

Once these mode work using the manual control, in the Arducopter code Navigate to Systems.pde and change the code as given below to set autonomous take off , hover and land. Call this function in the main Arducopter code by writing an autonomous function and defining the required variables in APM_config.h

Following the above steps, autonomous flight of a Quadcopter was achieved successfully.

INTEGRATING WII-MOTE CAMERA TO THE ARDUCOPTER CODE

Wii camera hacking and the use of wii camera for navigation was proposed by Jhonny Chung Lee and a number of articles to interface it to arduino and identify IR beacons. Using this information an attempt was made to integrate the camera to Crius aiop v2 and use it for navigation. The Wii-mote camera is first integrated to the arducopter code by creating a library for the camera code and adding it to the arduino libraries folder.

A P vision.h library that has all the definition of the Wii-mote camera to identify blobs of the IR source is also added to the arduino libraries folder.

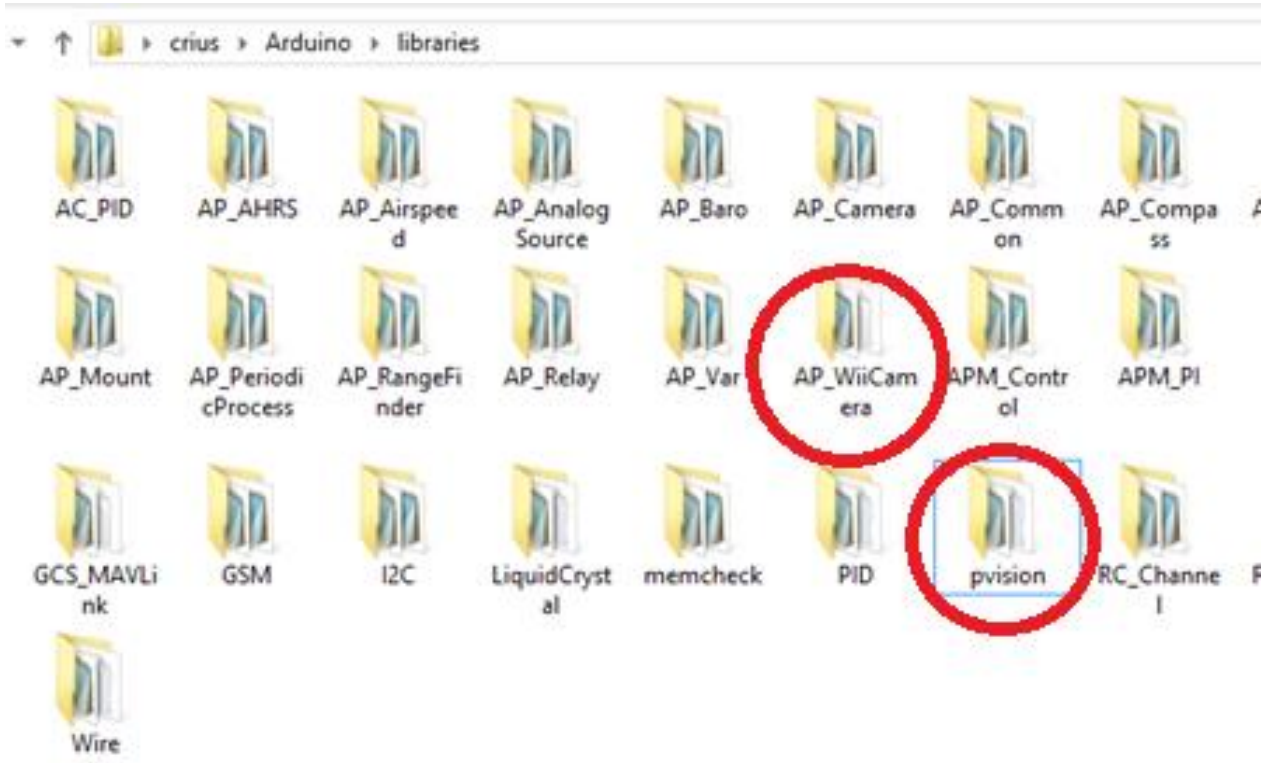

Fig.14: Pvision.h and Wii-mote libraries in the arduino libraries folder

The Wii-mote library has a .cpp file and .h file along with an examples file that has a program to identify 3 IR blobs to identify the direction of movements of the blobs. If the images are detected, the arducopter code is programmed to stay in flight and follow the blobs. If the images are not detected, the Quadcopter is programmed to land autonomously.

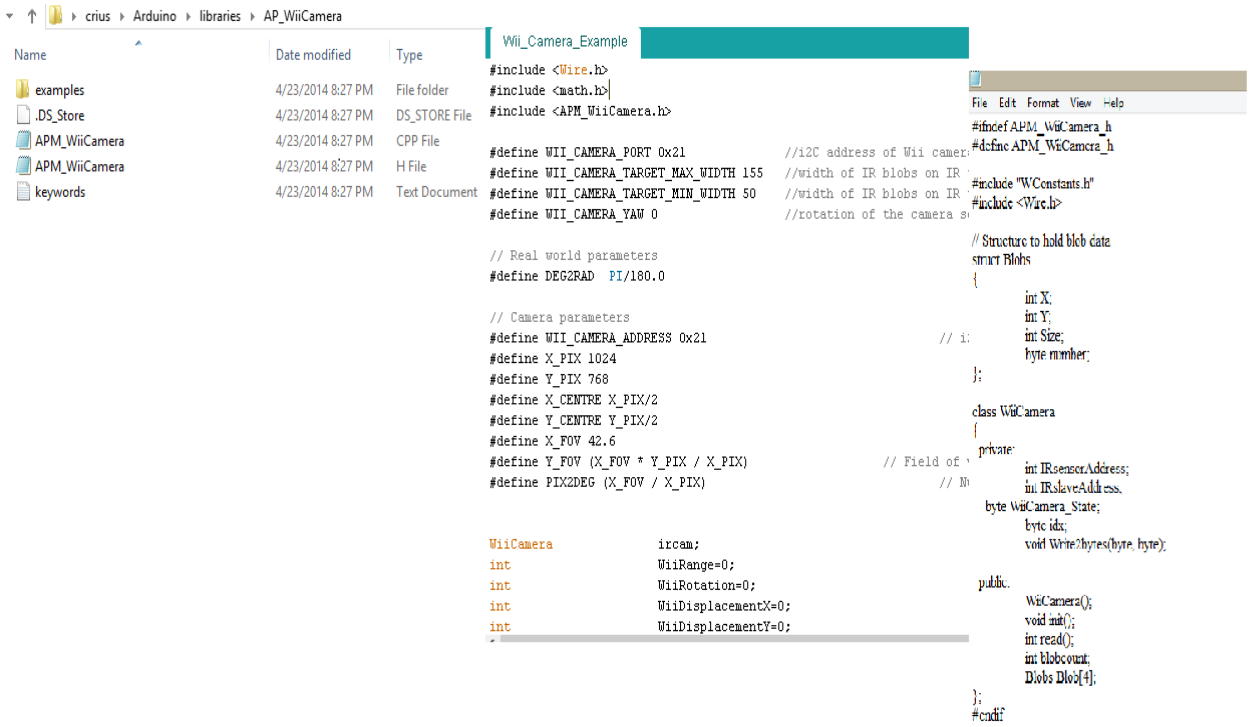

Fig.15: Code snippets for Wii camera navigation

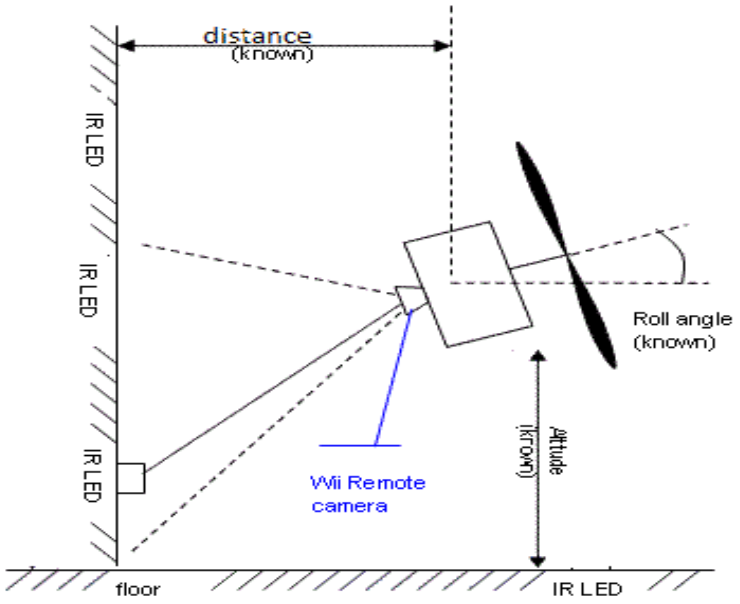

Fig.16: Pictorial representation of a camera detecting the beacon

The images of the changes that are seen when the images are detected can be tracked using Mission Planner. The graphical plot of the Wii camera and the sonar altitude are given below.

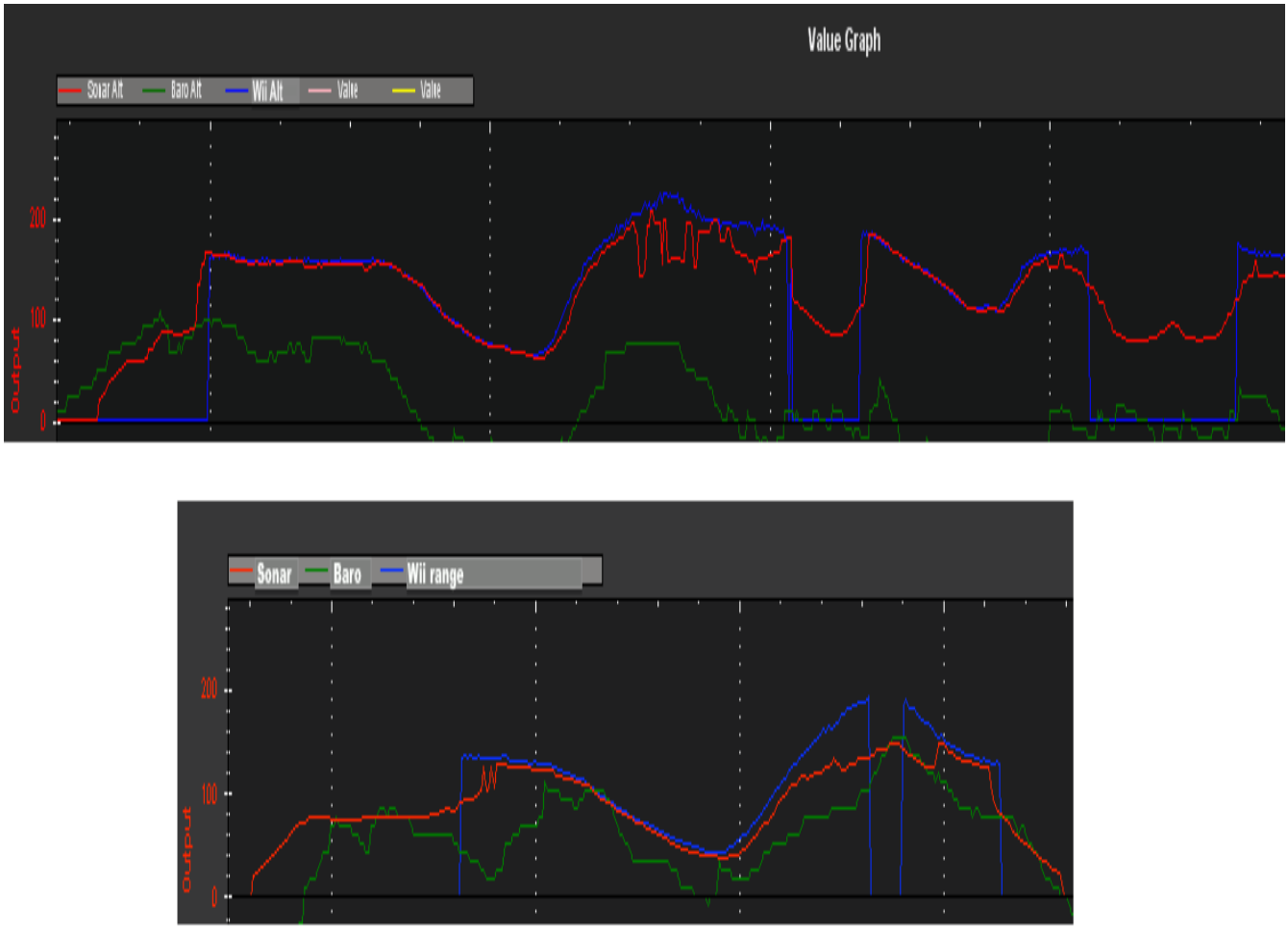

Fig.17: Mission planner graphical outputs

The Mission planner outputs clearly show the change in the Wii distance as the IR sources. Since the code is programmed to maintain a certain distance from the LEDs, the Quadcopter resets its position accordingly to maintain the given distance.

CHAPTER 3: CONCLUSION AND FUTURE WORK

The results of this project clearly indicate that the arducopter code can be edited to control the Quadcopter and attain an autonomous flight without the use of a GPS in an indoor environment. The addition of a Wii-camera library can be used to navigate the Quadcopter and follow IR beacons acting as a tracking device and control its indoor flight. The arducopter code also supports adding user written code in the USER HOOKS loop and since arducopter is controlled by loops that occur at different speeds, most functions just update things while a few functions actually control things.

This feature of the arducopter can be used to write our own code and achieve various other functionalities like obstacle detection, path planning and communication between multiple Quadcopters to attain a Swarm flight.

The next step in the autonomous navigation of Quadcopter can be to eliminate the use of IR beacons and write an algorithm to plan the path using the images obtained from the camera.

CHAPTER 4: REFERENCES

[1] http://the-force-physics.com/index.php?title=User_Hooks_and_Adding_User_Flight_Modes

[2] http://multi-science.metapress.com/content/80586kml376k2711/

[3] http://letsmakerobots.com/content/wii-ir-camera-standalone-sensor

[4] http://dev.ardupilot.com/

[5] http://copter.ardupilot.com/wiki/tuning/

[6]http://diydrones.com/forum/topics/arducopter-navigation-with-wii-camera?id=705844%3ATopic%3A649954&page=1#comments

[7]http://diydrones.com/forum/topics/arducopter-navigation-with-wii-camera?commentId=705844%3AComment%3A662476

[8] http://wiibrew.org/wiki/Wiimote#IR_Camera

[9] http://johnnylee.net/projects/wii/

[10] http://www.instructables.com/id/Wii-Remote-IR-Camera-Hack/?lang=de

[11] http://helihelp.rabbitsvc.com/

[12] HOW-TO-GUIDE.pdf by Cory Engel and research material on Wii-camera interface to arudino and vision system for miniature Quadcopters by Audrow.J Nash.