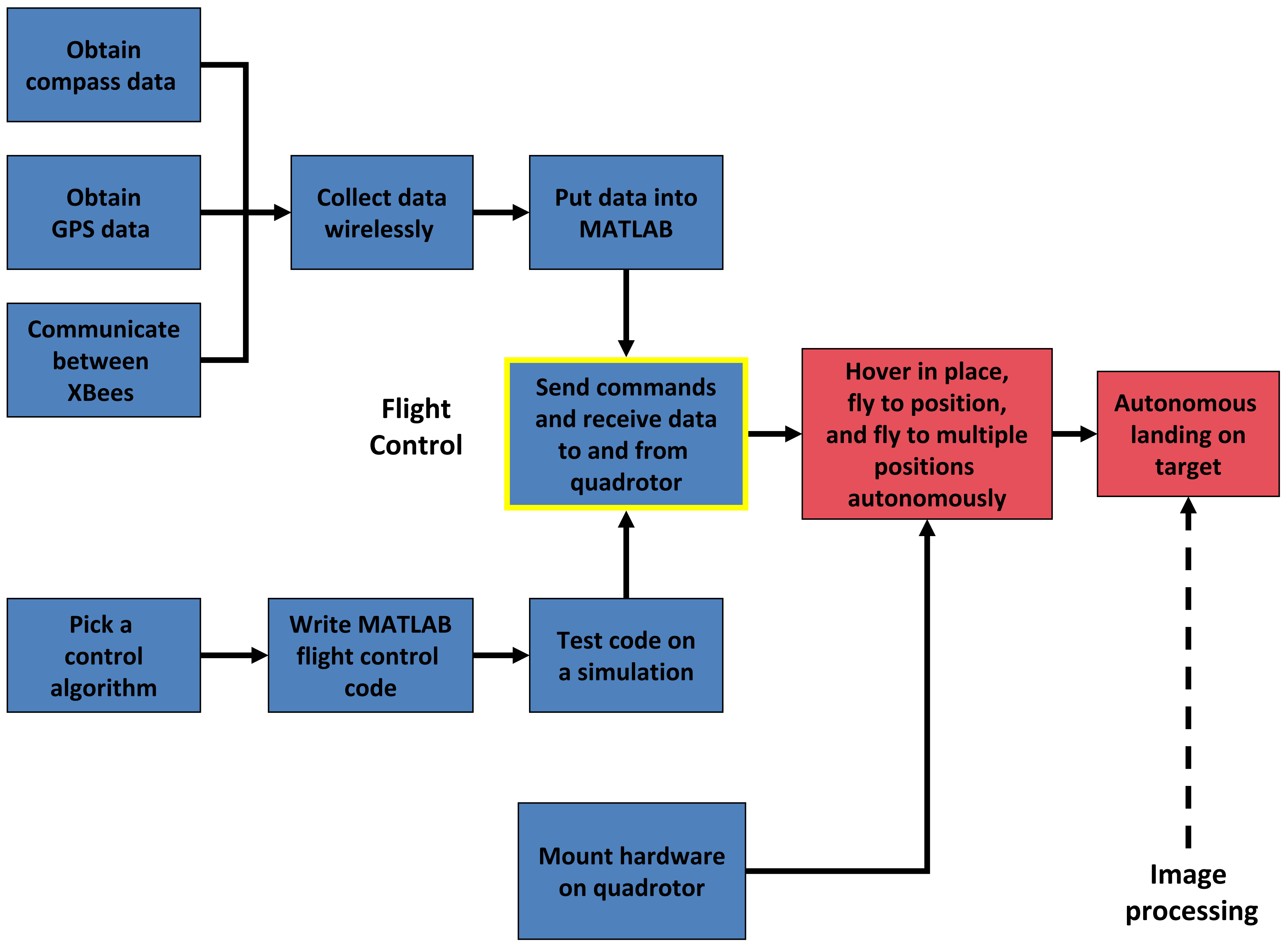
Autonomous Navigation of a Quadrotor Helicopter Using GPS and Vision Control

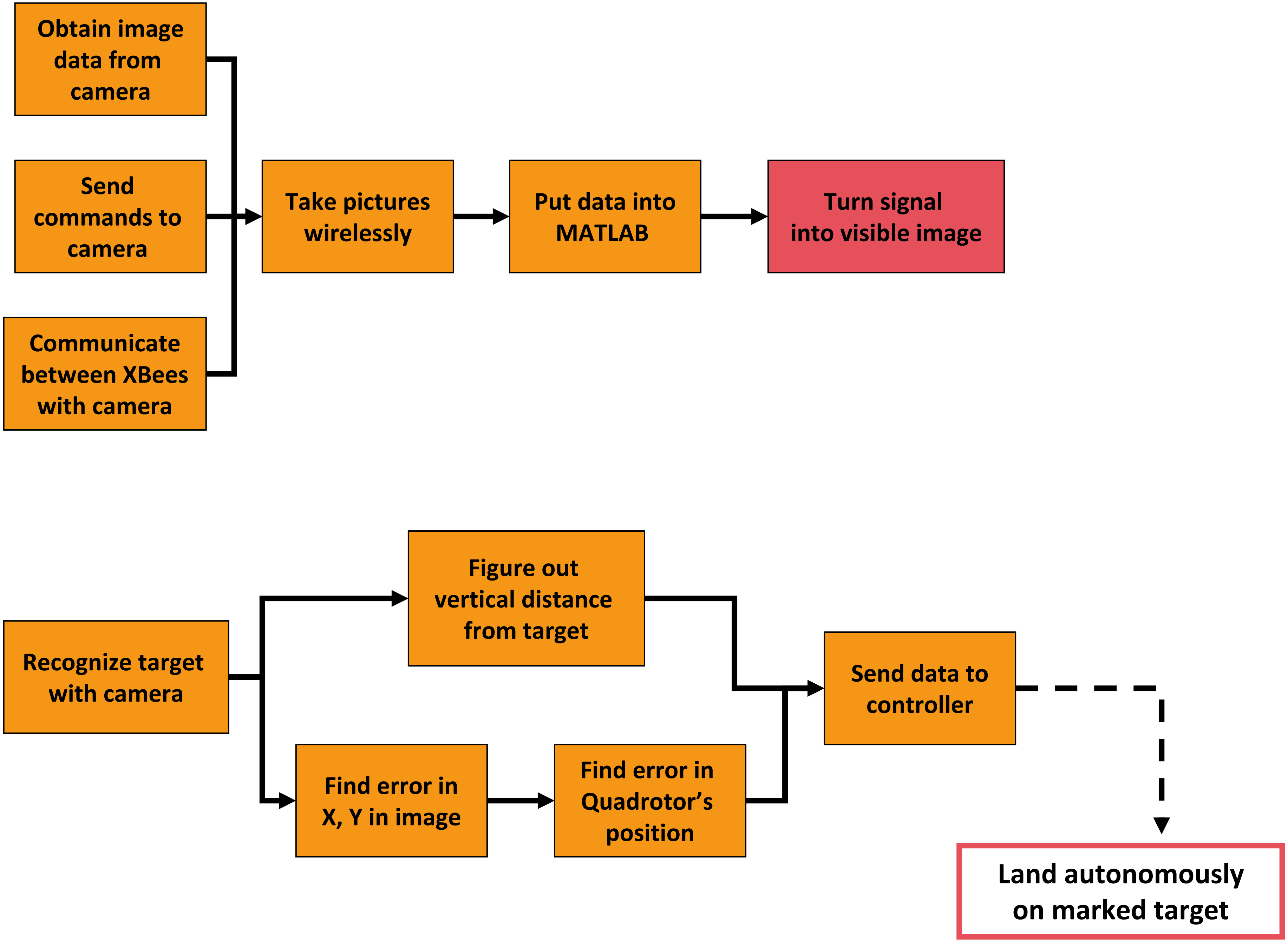
Photo of the Ascending Technologies Hummingbird Autopilot Quadrocopter removed due to copyright restrictions.

Group 1

Project Goals

- Fly helicopter to a predetermined location using GPS feedback
- Take pictures at this location
- Fly a planned path along GPS coordinates
- Take pictures along the reference path
- Use GPS and camera feedback to visually servo to and land on a marked target


Obtain compass data
Obtain GPS data
Communicate between XBees
Collect data wirelessly
Put data into MATLAB
Flight Control
Send commands and receive data to and from quadrotor
Hover in place, fly to position, and fly to multiple positions autonomously
Autonomous landing on target
Pick a control algorithm
Write MATLAB flight control code
Test code on a simulation
Mount hardware on quadrotor
Image processing



Obtain image data from camera
Send commands to camera
Communicate between XBees with camera
Take pictures wirelessly
Put data into MATLAB
Turn signal into visible image
Recognize target with camera
Figure out vertical distance from target
Find error in X, Y in image
Find error in Quadrotor's position
Send data to controller
Land autonomously on marked target

Software and Compass

Student A

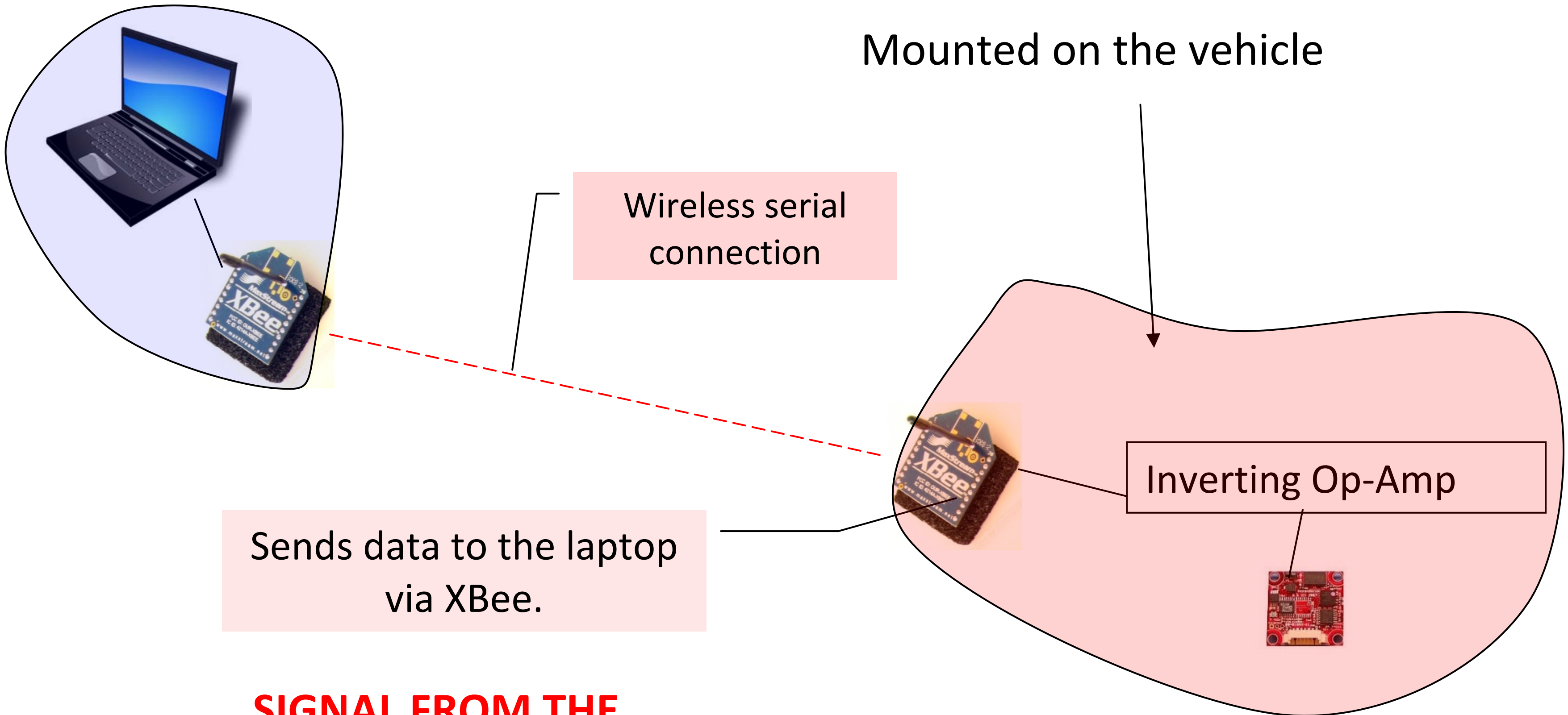
Accomplished

-> Serial connection with XBees

-> Wireless connection between the compass and my laptop

->Importing the readings to a MATLAB variable

Compass



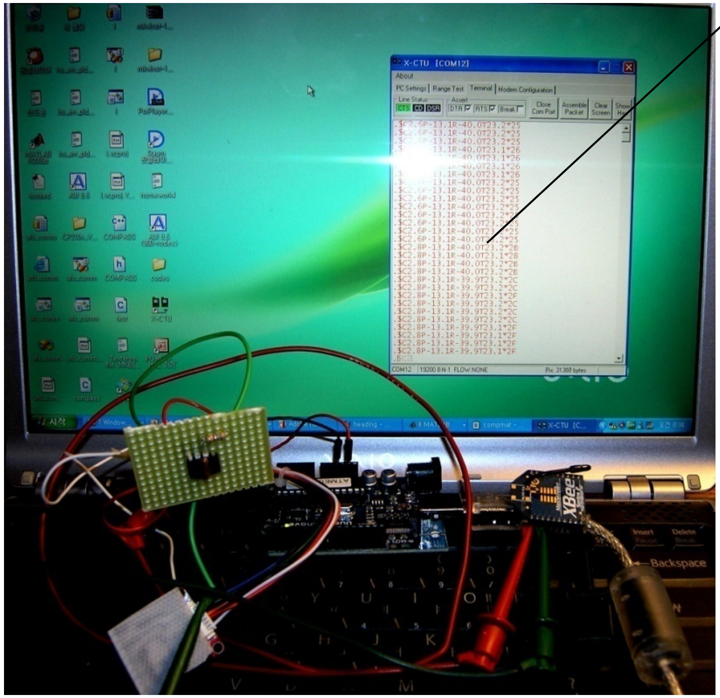
SIGNAL FROM THE COMPASS MUST BE INVERTED BEFORE IT IS FED TO THE XBEE!

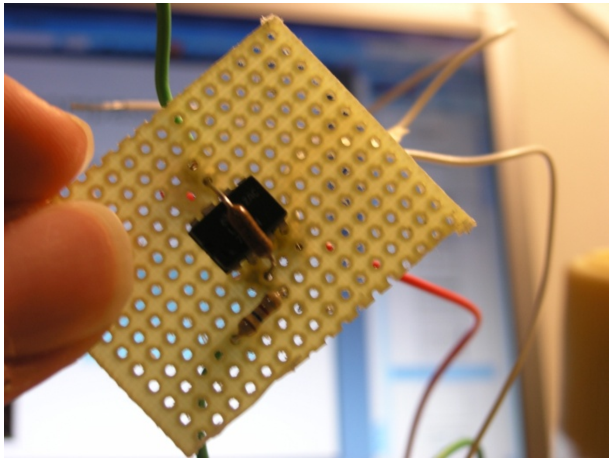
Images from the [OpenClipArt Library]() and [mangonha]() on Flickr.

Getting the compass to work

- In order to collect data, the output signal from the compass must be inverted.

Receiving serial data from the compass via XBee. This data can be imported to MATLAB.





Importing Serial data to MATLAB

- Luckily, we already have a VC++ code that can read the heading of the compass.

- “MexFunction” and some other lines are added to the C Code in order to send the heading to matlab

```
……
……….
#define              YP_OUT              plhs[0]
  void mexFunction( int nlhs, mxArray *plhs[], int nrhs, const mxArray* prhs[] ) {   //start of
the mexfunction.
   double *yp;
                    float tempheading;
                    opencompass();  //opens compass

                                        tempheading = readcompass();
                                        WaitForSingleObject(headingMutex, INFINITE);
                                        heading = tempheading;
                                        ReleaseMutex(headingMutex);
.............
...
```

What the code is doing

- The mex function “compmat” reads the heading of the compass at that instant.

(returns one float type number)

- Example: code that reads the heading for 10 seconds (10 readings per sec)

```
for i=1:100
   heading(i) = compmat;
   pause(0.1);
end
```

results

heading <1x100 double>

33	34	35	36	37
157.1000	133.6000	142.1000	151.2000	149.5000

Future Plans

- Solder the whole thing up so it can be attached to the vehicle.
- The XBee and the compass needs 3.3V. Figure out how to get this voltage from the quadrotor's battery.
- Try out the control code by flying the quadrotor with the compass and the GPS attached.
- After we start getting decent results, we can take off all the XBees and replace it with an Arduino; which will be much more elegant.

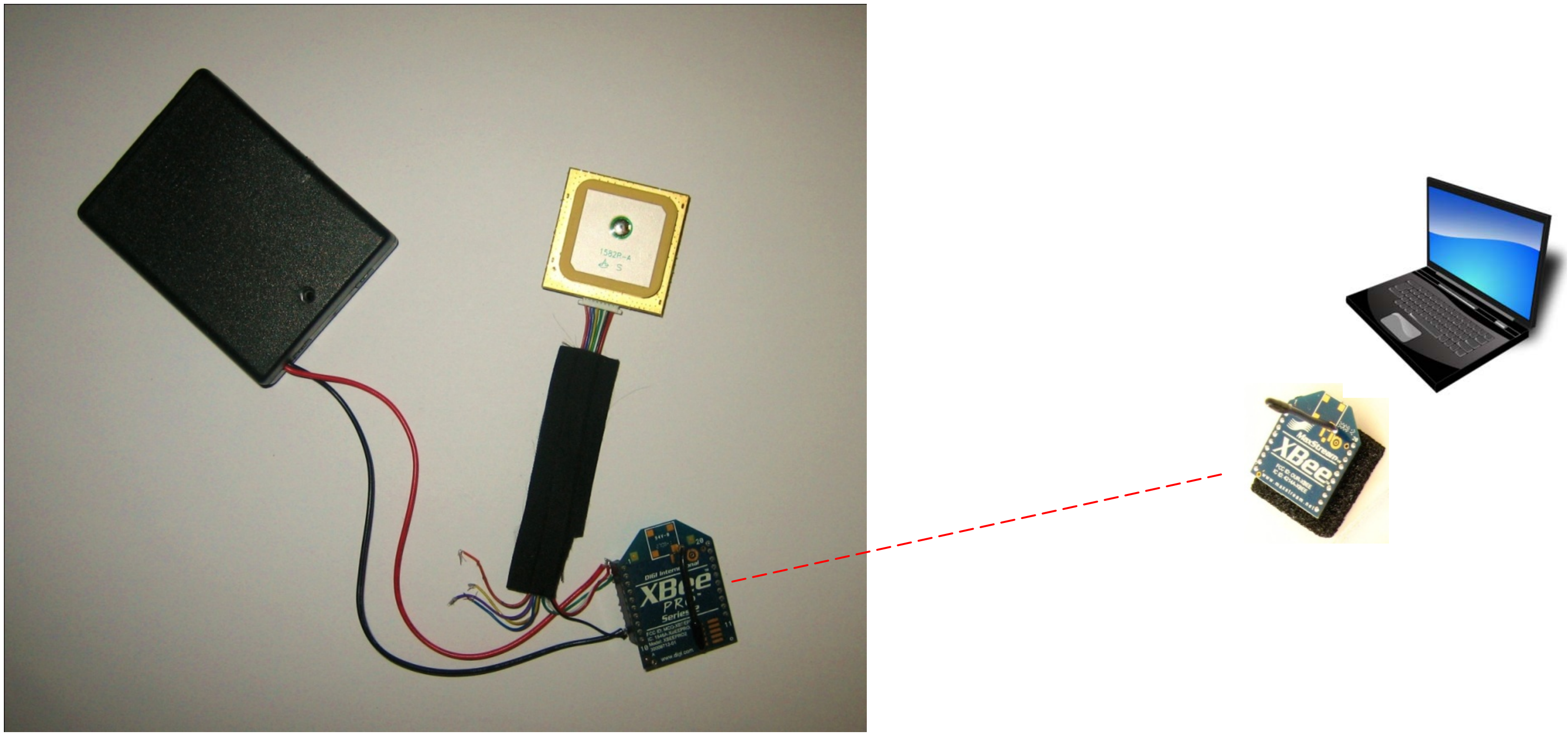
GPS Integration and Hardware

Students C and E

Progress

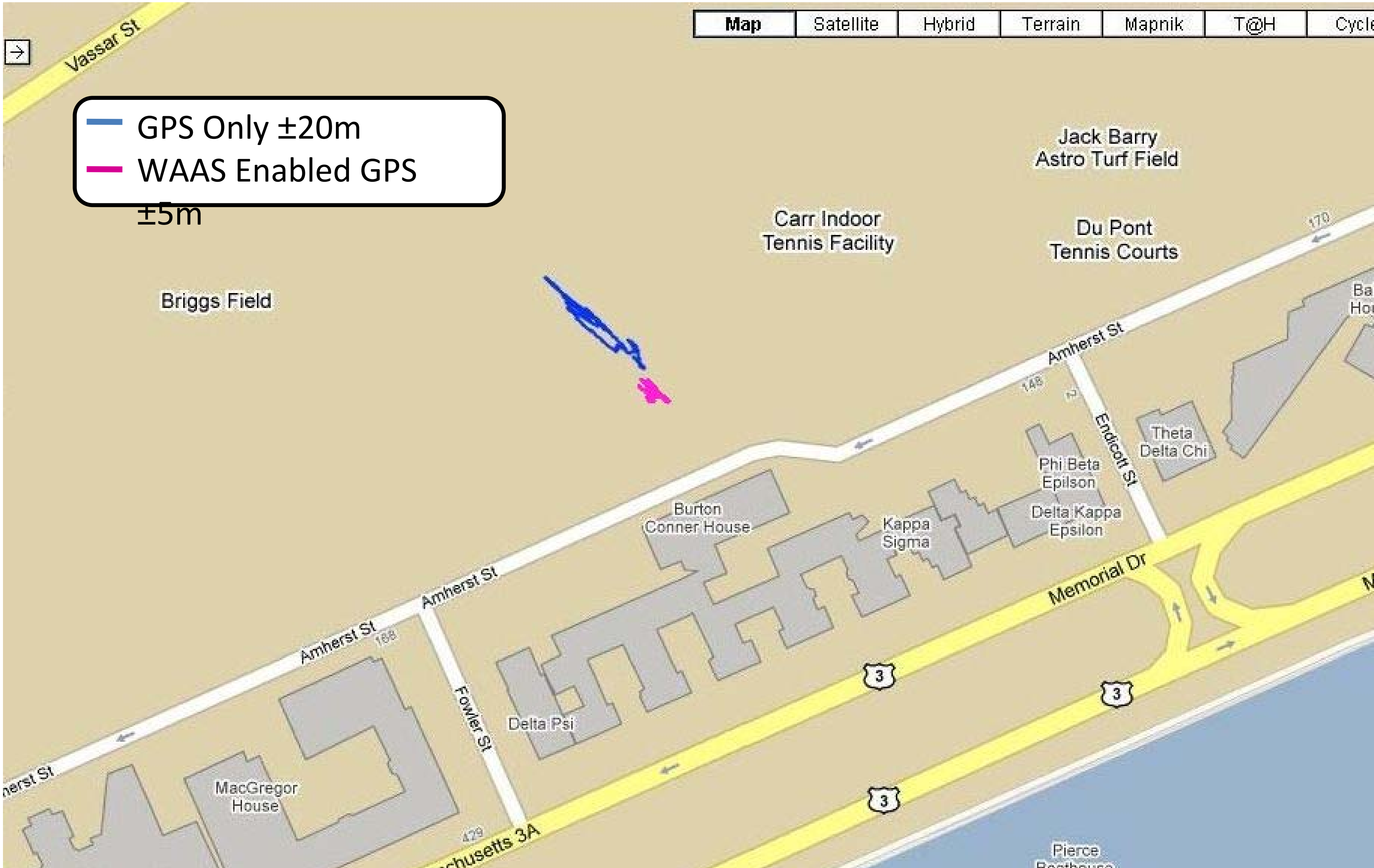
- ☑ Read GPS signal directly connected to computer
- ☑ Successful XBee communication
- ☑ Read GPS signal through XBee communication
- ☑ Transmit GPS signal from Quadrotor
- ☐ Send data to MATLAB

Current Set-up



Images from the [OpenClipArt Library]() and [mangonha]() on Flickr.

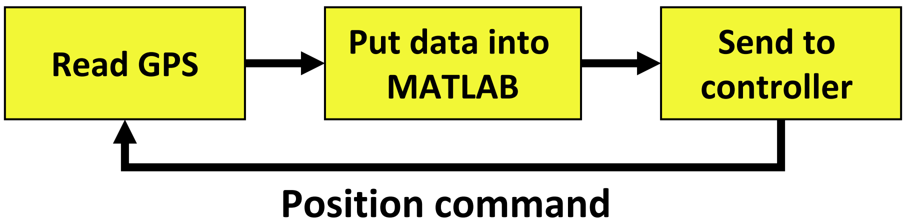
Accuracy Tests



Quadrotor Test

Mission Goals:

- ❑ Fly to preset GPS coordinate
- ❑ Maintain position
- ❑ Fly a planned path based on GPS



The Next Step:

- Get MATLAB to read data from the XBee
- Build a more permanent housing for the GPS

Control System

Student B

Deliverables

- ☑ Demonstrate closed loop control on a LTI model of the quadrotor
- ❑ Demonstrate closed loop control of the quadrotor

Control System Design

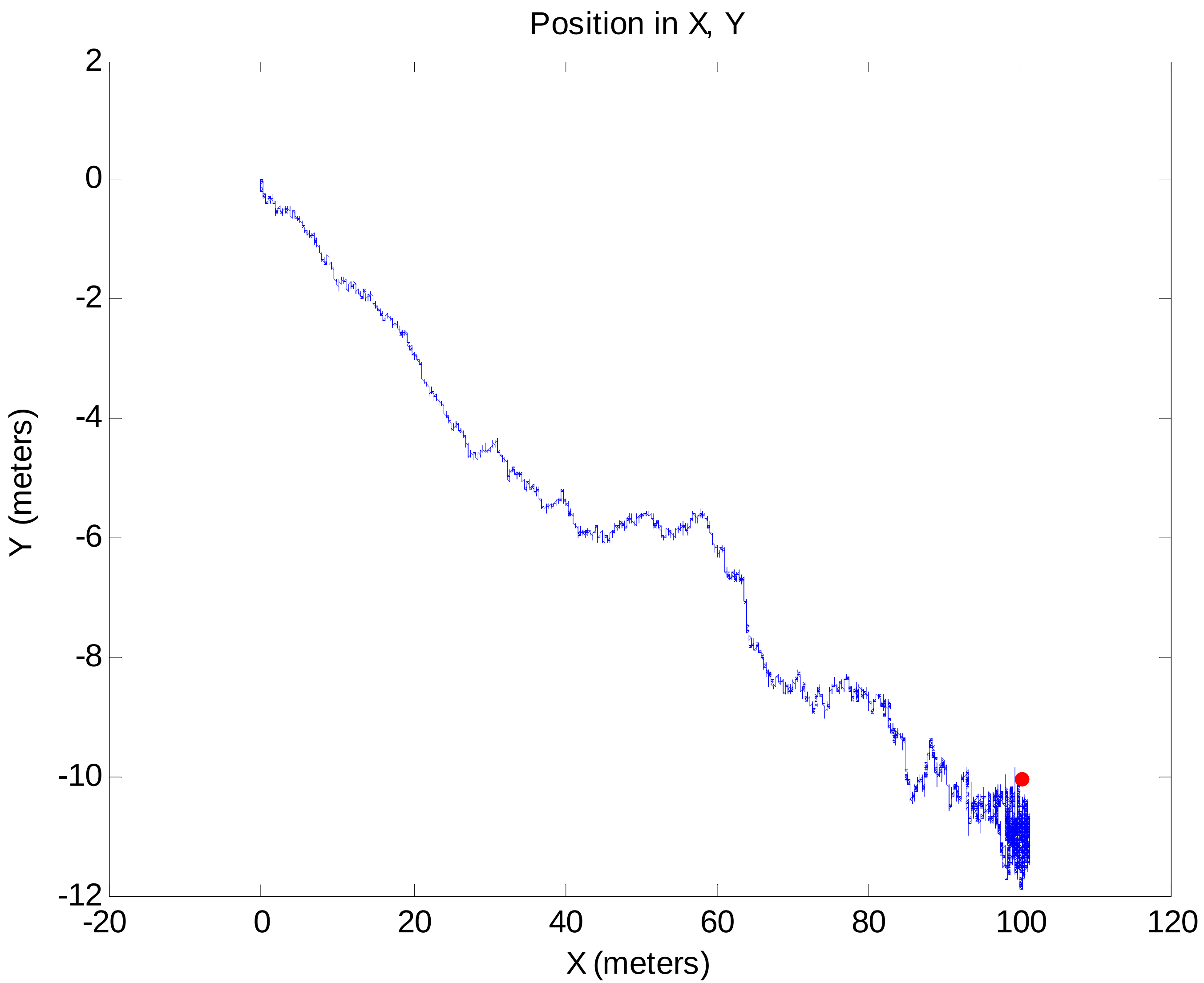
Strategy: Point and go

- Yaw is set initially and is static
- Depends on ability of quadrotor to pivot
- Controlled variables
  - Yaw rate: points at the target
  - Roll: keeps on line to target
  - Pitch: determines speed forward
  - Thrust: offsets gravity and brings rotor to correct height
- Measured variables
  - Yaw: compass
  - X, Y positions: GPS, camera
  - Height: Pressure sensor, GPS

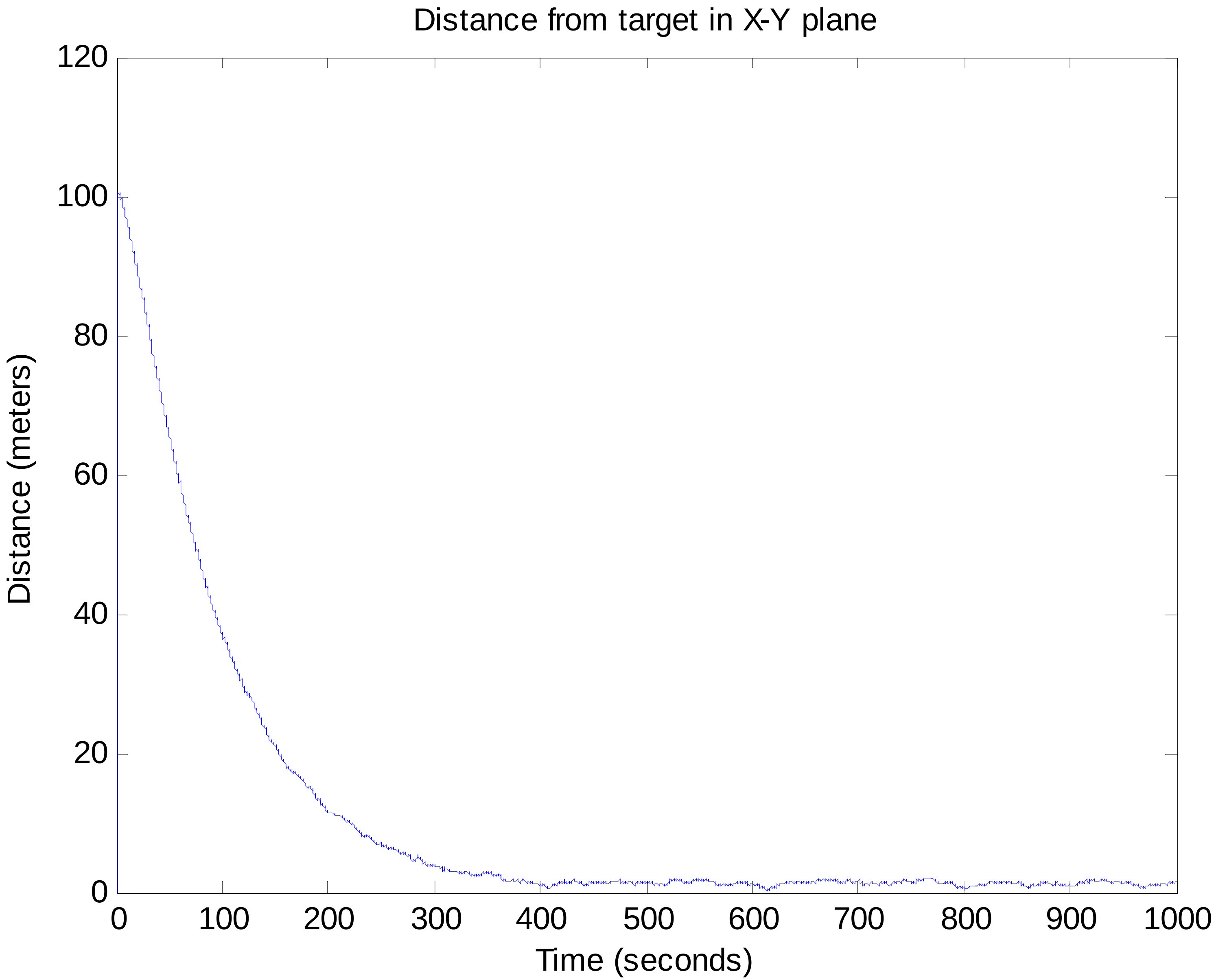
Model Assumptions

- Linear Time Invariant
- Small angle pitch and roll
- Max, Min thrust = 1.25*m*g and 0.75*m*g
- Delta t=0.001 seconds
- Added in a lot of random noise to x_dot, y_dot, z_dot and yaw

Pretty Pictures



Pretty Pictures



Pretty Pictures

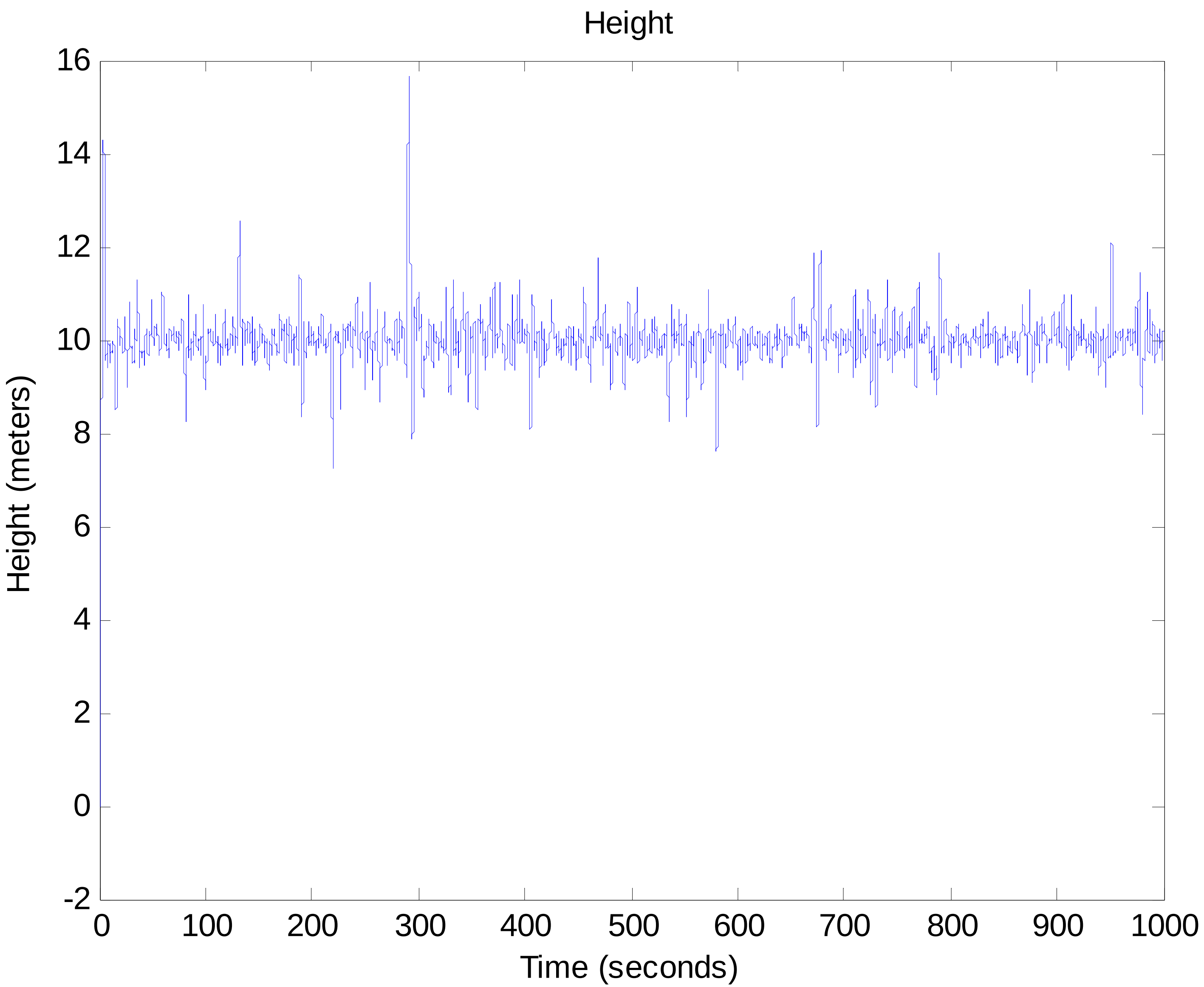


Image Processing Goals

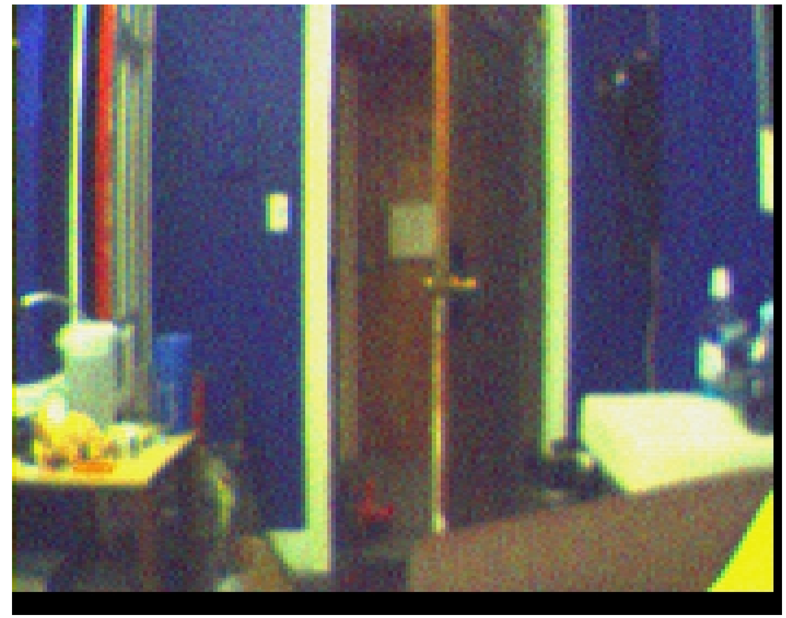
- Take pictures of a predetermined location and also along a reference flightpath
- Track a landing target at a known location
- Visually servo to the target using feedback from the image

The CMUCam2+

Photo of the CMUCam2+ removed due to copyright restrictions.

- Specifications
  - Low res: 87x142
  - Hi res: 174x254
- Features
  - Tracks motion
  - Tracks colors
  - Makes real-time histograms
  - Face recognition

Low Resolution Image



High Resolution Image

Obstacles in Using the CMUCam

- Poor implementation
  - GUI can directly execute functions, but the data goes nowhere
  - GUI can't take frames and data simultaneously
- Serial communications
  - CMU's recommendations don't work
  - Crashes MATLAB
- Camera cannot directly save an image

Relevant CMUCam Functions

- Frame Difference (FD)
- Track Color (TC)
- Track Window (TW)
- We choose to exclusively use TC
  - FD only works when the camera is stationary
  - TW only tracks colors in the center of the frame

T Packets

Centroid of tracked data

Bounding box coordinates

T 84 132 4 1 172 250 255 12

Indicates a color tracking data packet

Number of pixels that match the tracked color

Confidence

This is all the data we need to achieve the mission goals

What's Next

- Set up an XBee to work with CMUCam and MATLAB (without crashing)
- Test CMUCam parameters in a mission environment
- Establish visual servoing procedures

Arduino Microcontroller and Communication

Student D

Accomplished

- Reading output from compass in Arduino Mega

Benefits of On Board Data Collection

- Range:
  - 120 m (line of sight, outdoors) XBee
  - 40 m (indoor) XBee
  - Not an issue with on board Arduino
- Always in contact:
  - On board Arduino will not lose contact with sensors due to range or signal issues

Disadvantages of on Board Data Collection

Weight:

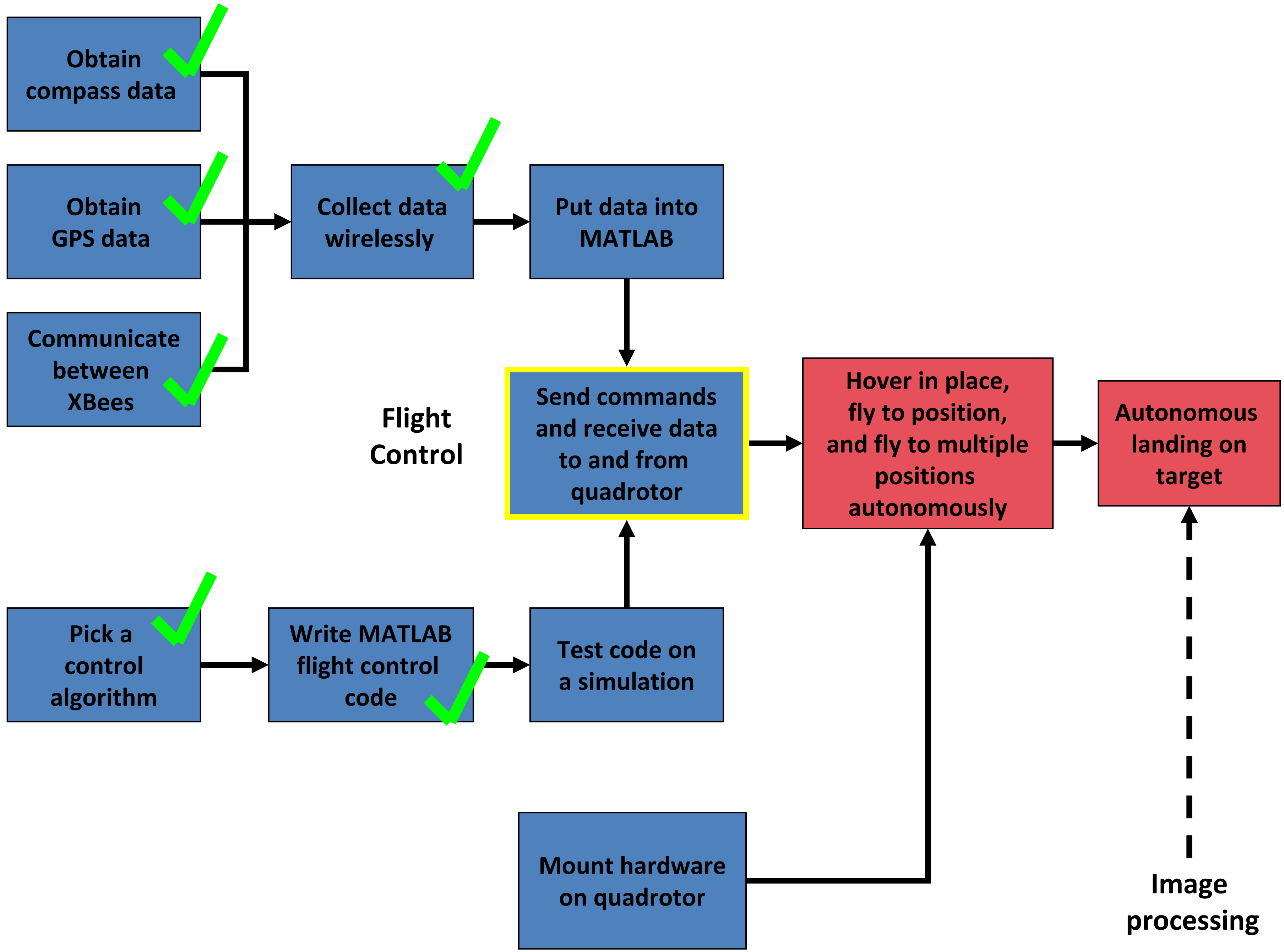
- 4 XBee's weigh 12-16g (depending on type)
- Arduino Mega weighs 40g

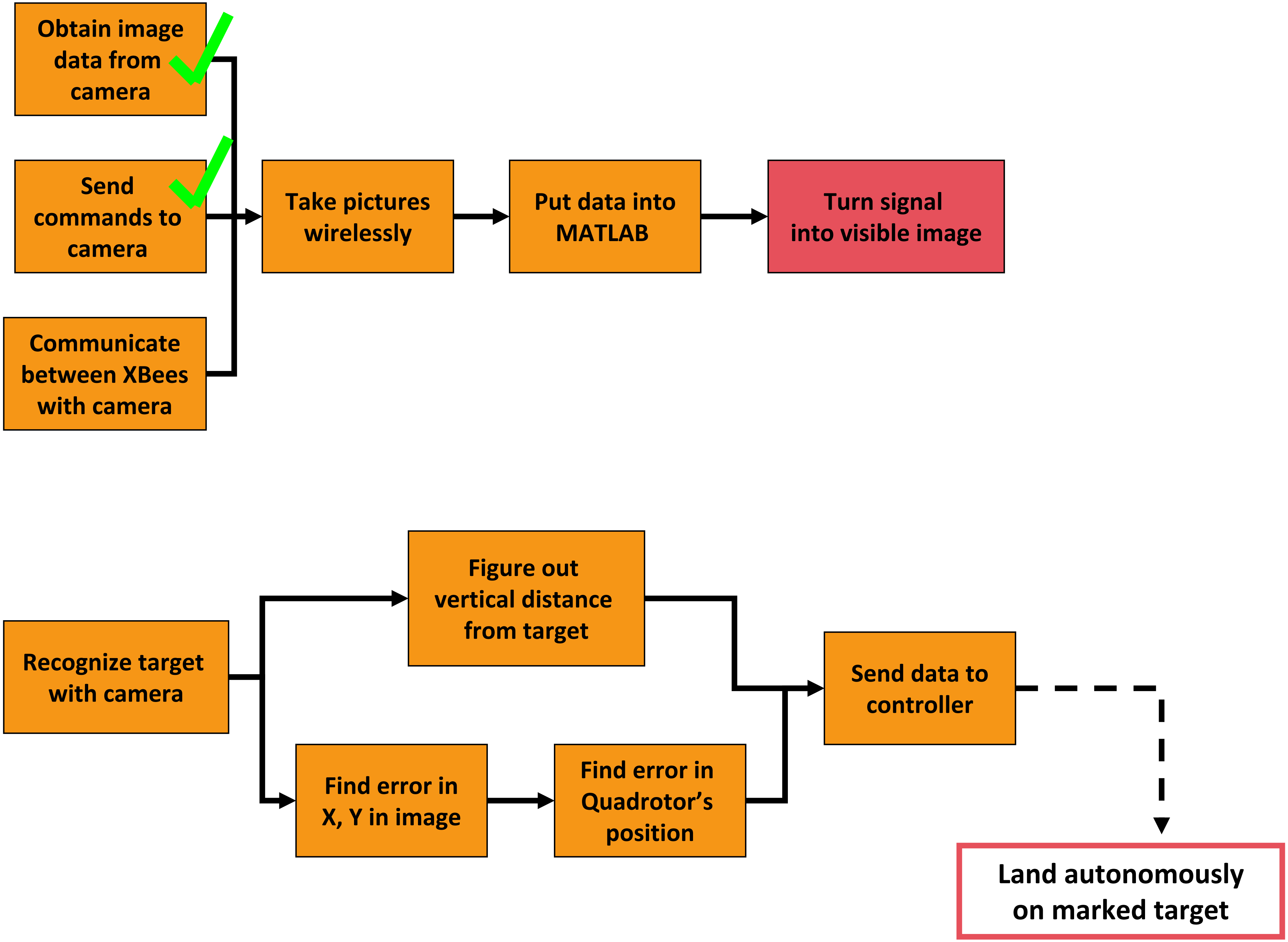
Work Left

- Read GPS output
- Communicate with CMU cam
- Communicate with quadrotor
- Communicate with computer via XBee

Possible On Board Control

- Benefits
  - 'cordless' – no XBee tether
  - Elegant
- Disadvantages
  - Less computing power
    - Mega has only 124 KB of flash memory
  - Programming restrictions
    - XBee can receive instructions from any code/program on computer (MATLAB ...)
    - Arduino has only C/C++ and Arduino language extensions


Obtain compass data
Obtain GPS data
Communicate between XBees
Collect data wirelessly
Put data into MATLAB
Flight Control
Send commands and receive data to and from quadrotor
Hover in place, fly to position, and fly to multiple positions autonomously
Autonomous landing on target
Pick a control algorithm
Write MATLAB flight control code
Test code on a simulation
Mount hardware on quadrotor
Image processing


Obtain image data from camera
Send commands to camera
Communicate between XBees with camera
Take pictures wirelessly
Put data into MATLAB
Turn signal into visible image
Recognize target with camera
Figure out vertical distance from target
Find error in X, Y in image
Find error in Quadrotor's position
Send data to controller
Land autonomously on marked target

Questions?