Advanced Mechatronics: AR Parrot Drone Control Mini Project #2

Engineering Team Members:

Ashwin Raj Kumar
Feng Wu
Henry M. Clever

Advanced Mechatronics: Project Plan

Phase 1: Design Testing Platform

Phase 2: Automated Landing Sequence

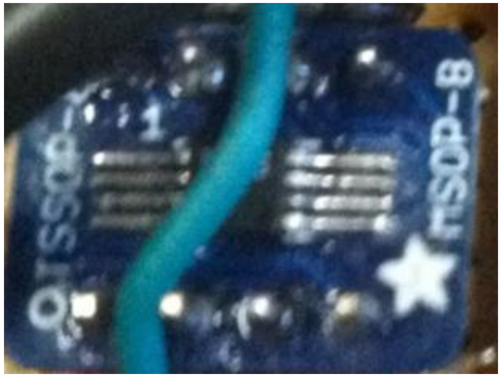
Phase 3: Battery Charging Station +
Improved Robustness of Landing

Infrared Light Camera-1

- Nintendo Wii remote has an infrared LED camera that tracks coordinates of up to 4 LEDs within a 20 degree aperture angle
- Range in the X direction: 1023 units
- Range in the Y direction: 773 units
- Tracking LEDs can be used to detect distance and rotation of an object wirelessly

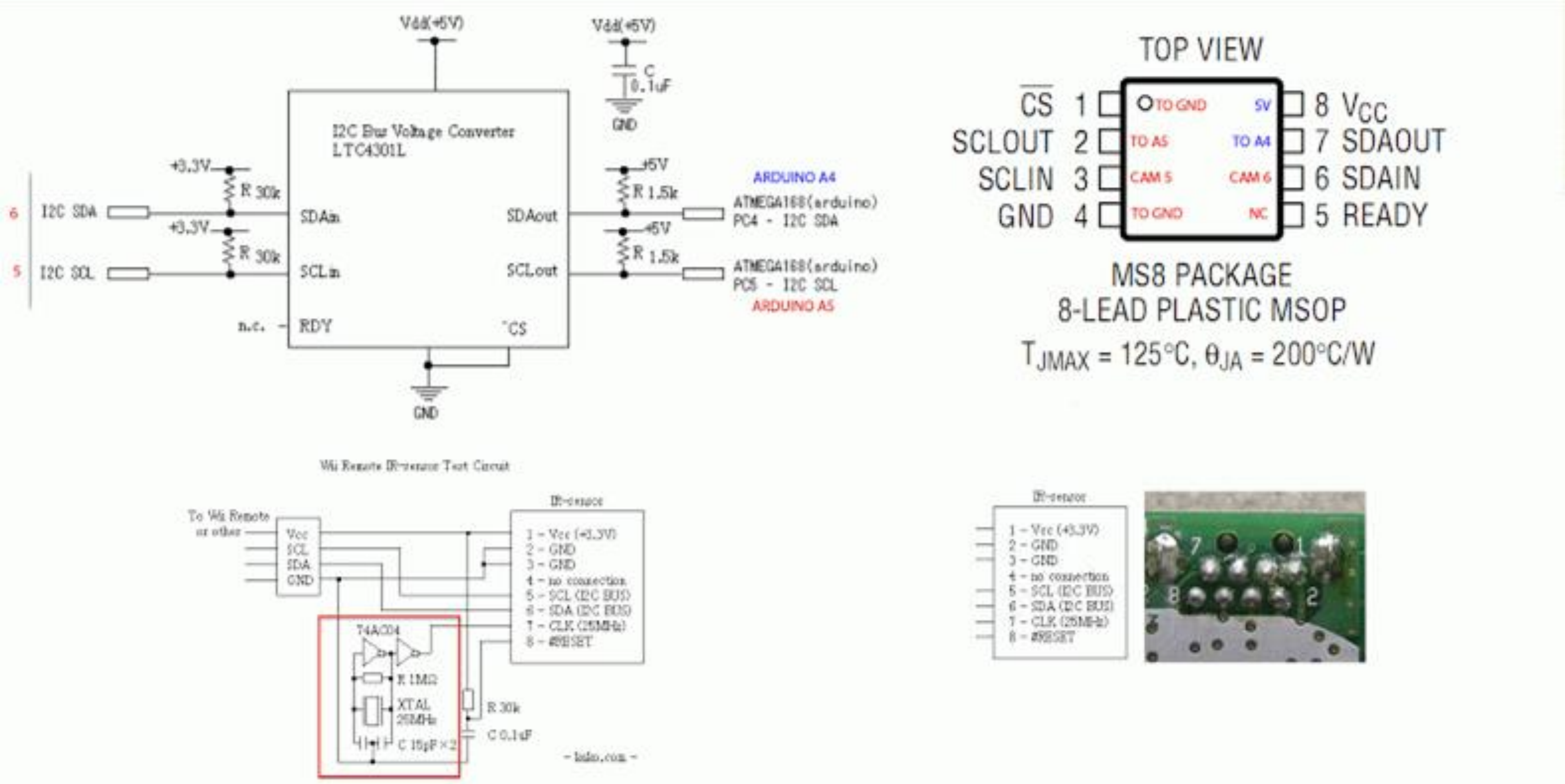
Infrared Light Camera-2

- To configure the LED camera to work with a microcontroller such as Arduino or Propeller, an additional circuit is necessary to enable I2C communication of LED coordinates
- LED camera requires a 20-25 MHz clock to enable video capture and I2C communication over a 3.3 volt line.
  - Because the camera must be mounted on the AR Drone 2.0, we will use an Arduino micro for RF communication between camera and Parallax Propeller
  - Therefore, an additional LTC4301 chip must be used to convert the 3.3V I2C signal to a useable 5V I2C signal that the Arduino Micro Requires



Infrared Light Camera-3

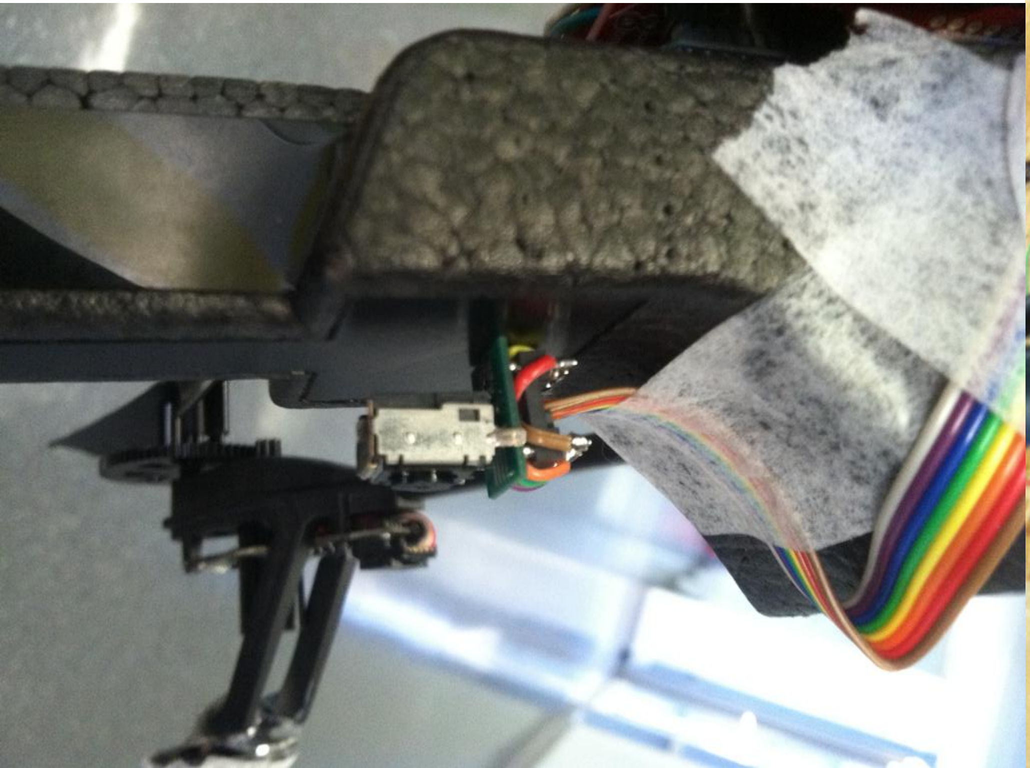
- Additional NOT gates, resistors, and capacitors are necessary to configure Wii camera for proper connection of crystal and I2C.

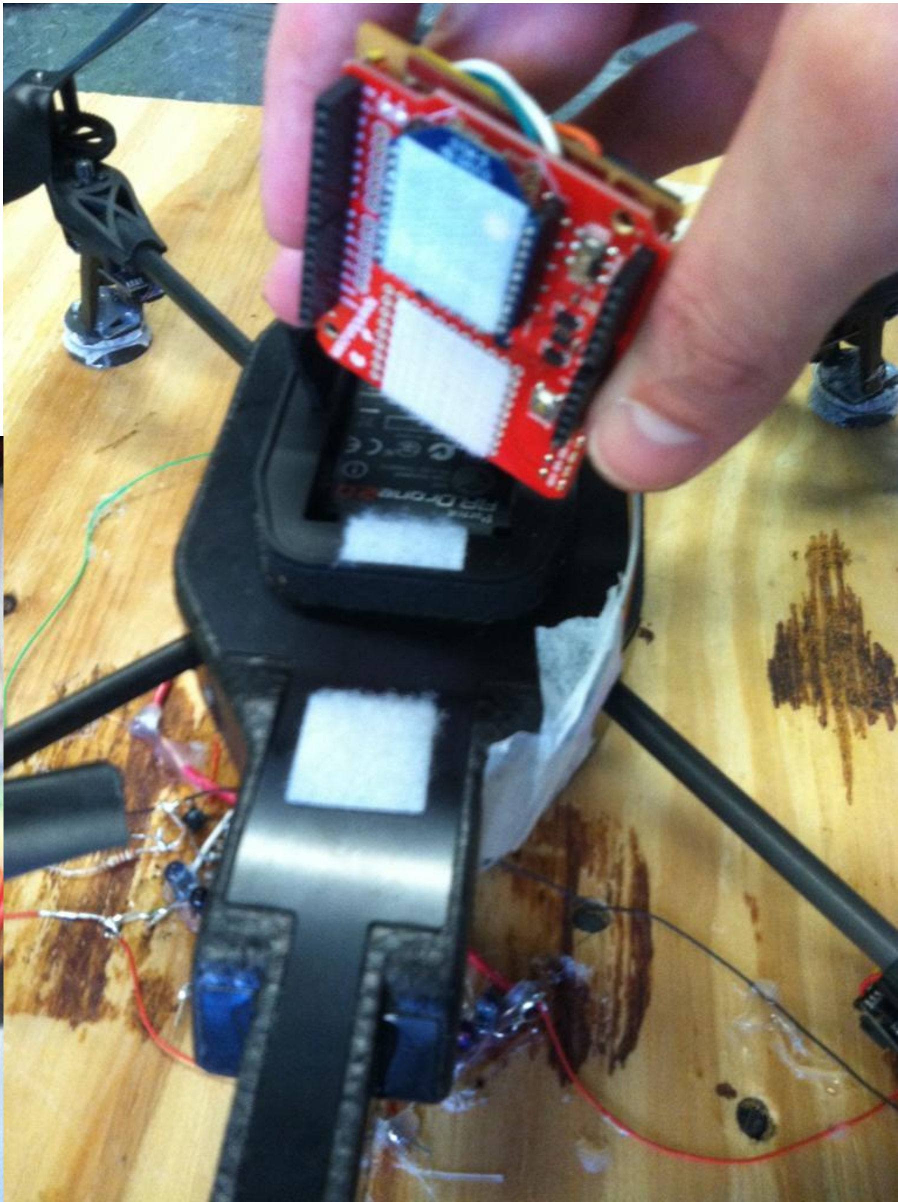


Infrared Light Camera-4

Infrared Light Camera-5

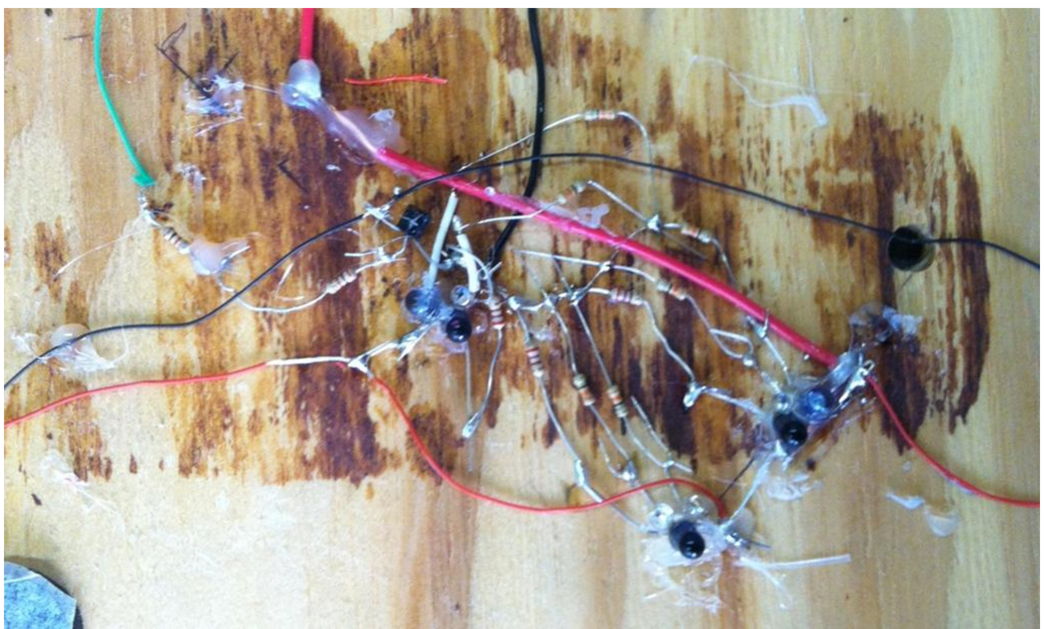
- Camera attached with 8 wire bus taped to AR Drone structure
- XBee and electronics velcro to drone for easy removal

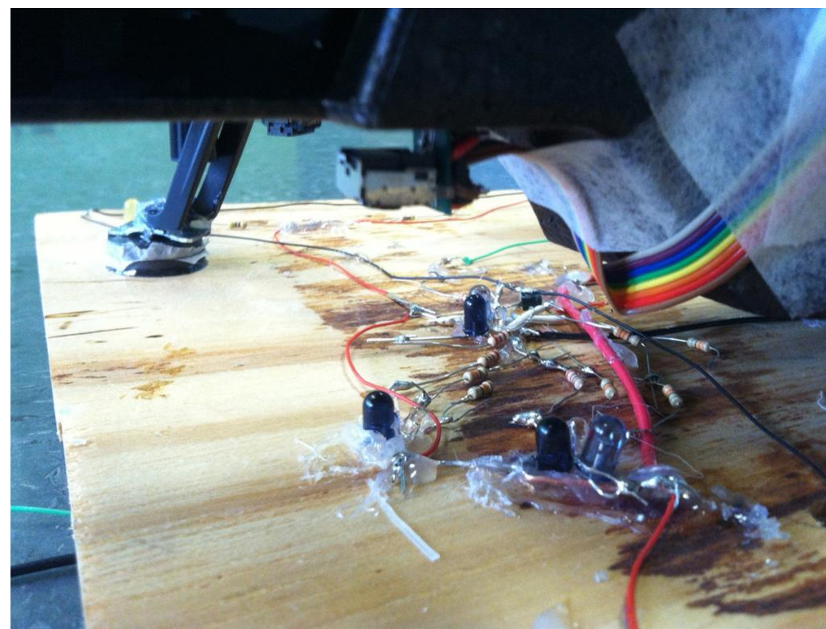




Landing Pad: IR LED Layout-1

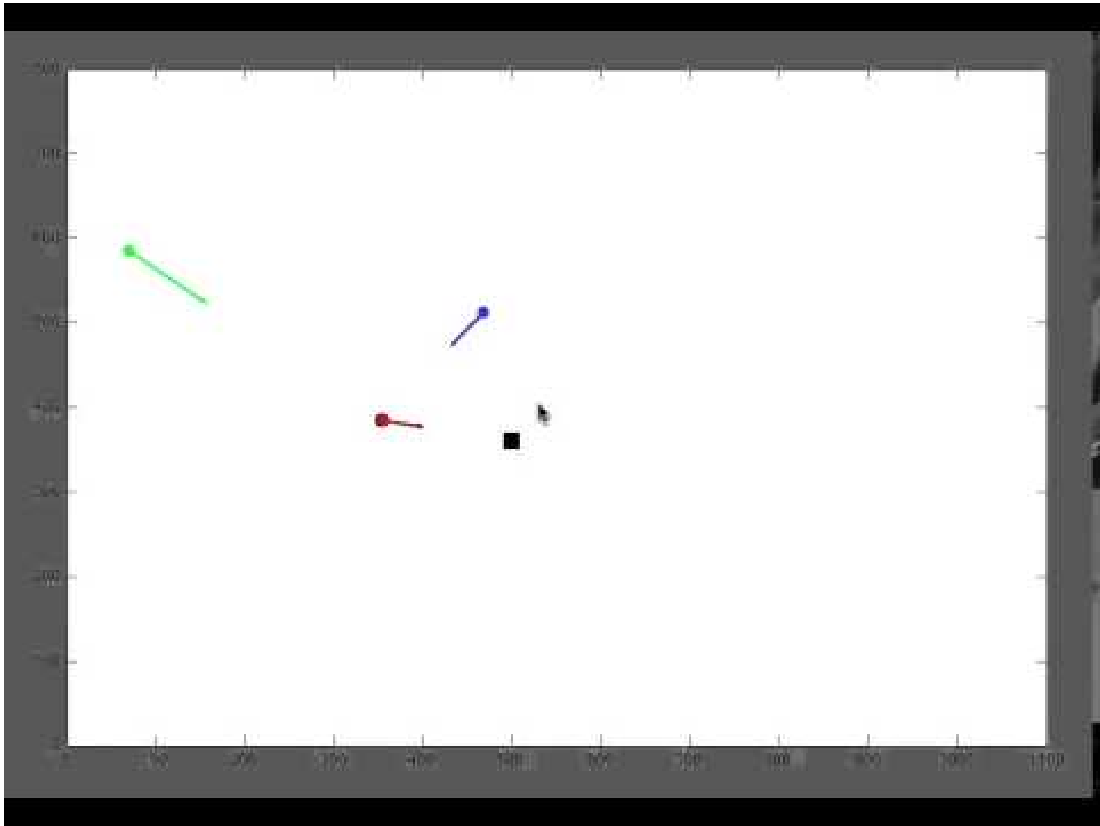
- IR LEDs are spaced about a right triangle
  - Distance and location of the drone from the landing pad can be calculated from coordinate mapping of the right triangle
  - Although the point seen by the camera is arbitrary, a calculation is performed to find the hypotenuse so the system can always identify the same point on the triangle correctly
  - Due to poor aperture angle of the LED light output, each point of the triangle has a cluster of 3 LEDs.





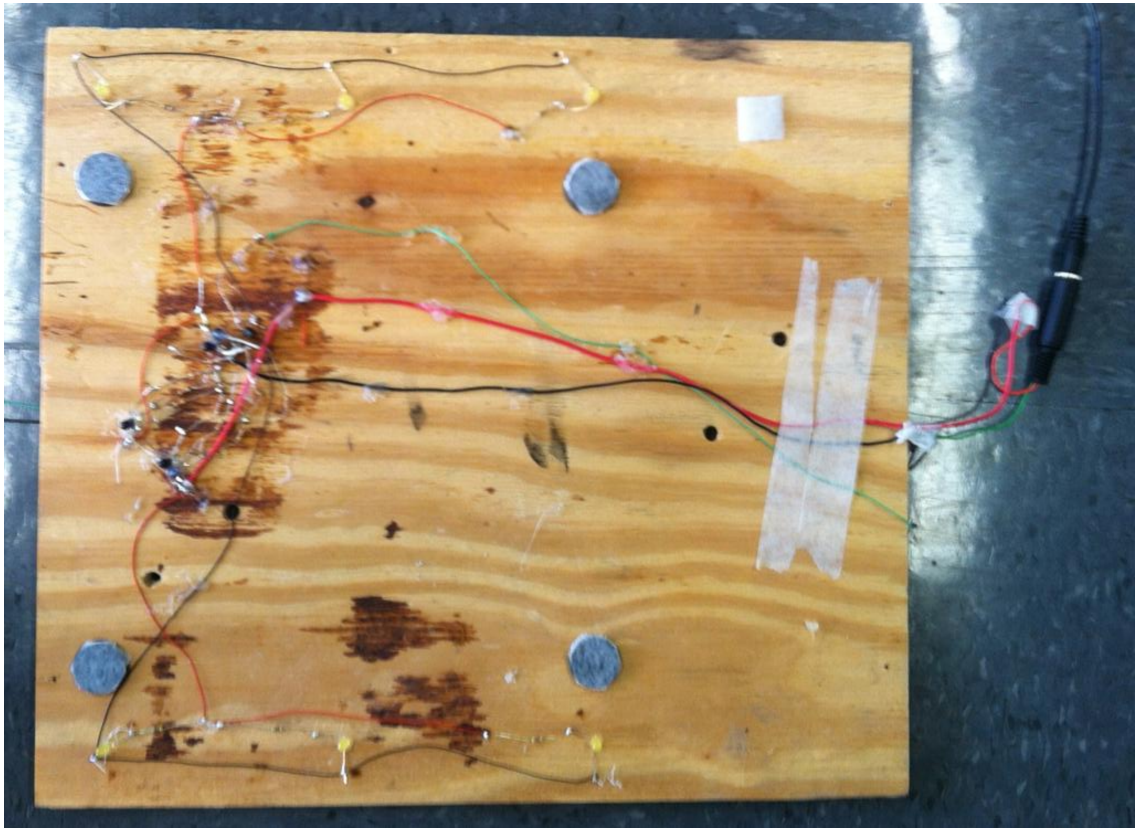
Landing Pad: IR LED Layout-2

- For visualization, a Matlab video is displayed below to show the camera sensing each cluster of 3 LEDs.
  - Although one might expect there to be a 'doubling problem' from the presence of these LEDs, the camera does an excellent job of seeing them as a single point.
- Signal Processing Feedback:
  - <u>Length of hypotenuse</u>: increasing as AR Drone flies lower
  - <u>Rotation of triangle</u>: yaw angle of AR Drone

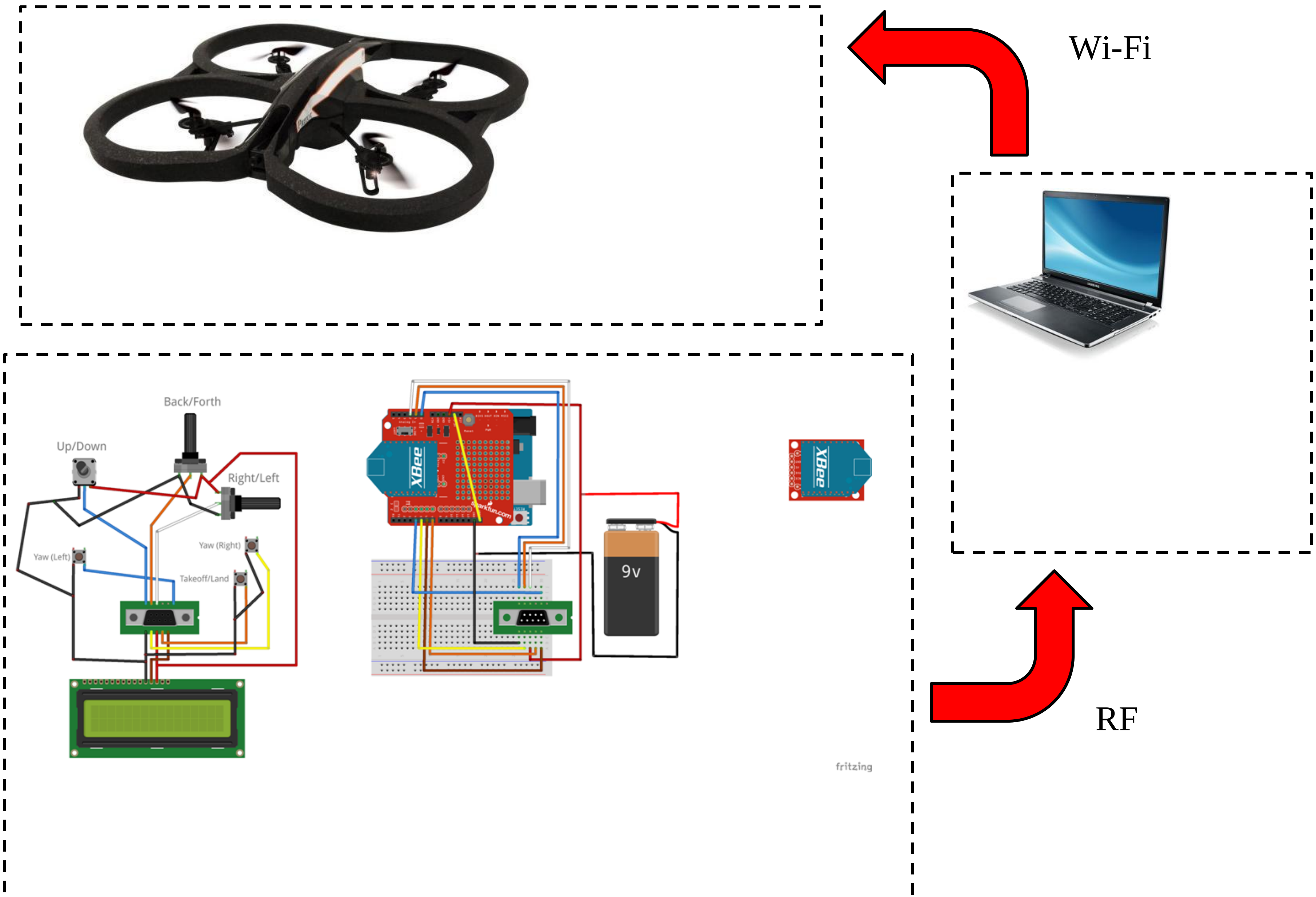


Landing Pad: Magnets

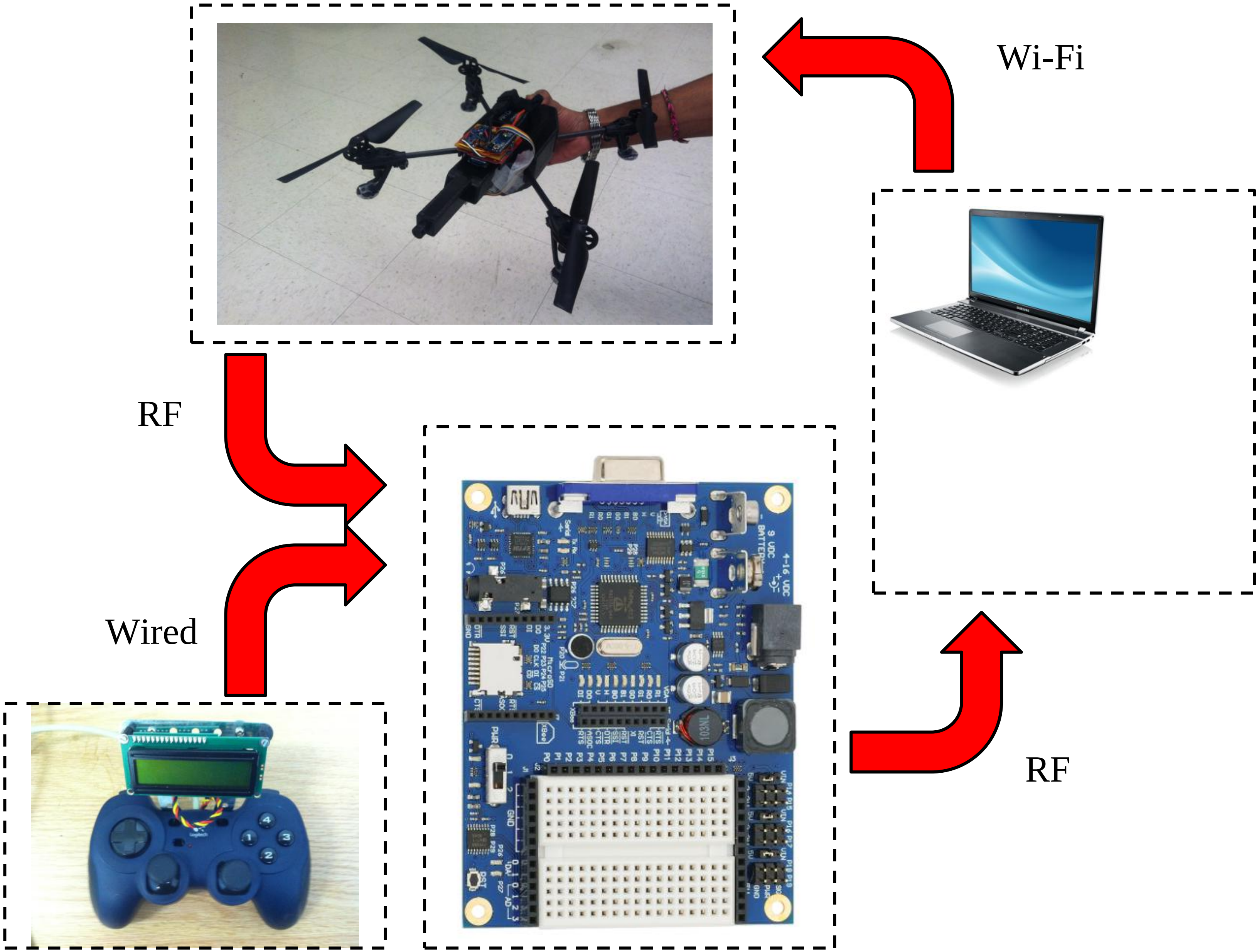
- Although the landing pad with IR LED and control sequence may perform well standalone, magnets are added to the feet of the AR Drone and the landing pad to ensure exact landing
  - This allows contact to be made in the automatic battery charging station: the goal of phase 3



AR Drone System Schematic: Phase 1



AR Drone System Schematic: Phase 2



Mechatronics Features-1

- Parallax Digital LCD controlled through digital output signal
  - Mounted on joystick for easy visual reference
- 3 buttons on customized joystick used for control of:
  - Takeoff and Landing
  - Yaw movement (rotation about the y, or up/down axis)
- 3 analog pins used for velocity in x,y,z directions
  - Joystick knobs only move through ½ of a 270 degree pot: thus, 3.3V is used as an analog reference for resolution recovery. With this we are able to utilize about ¾ of the 10 bit resolution.
- XBee RF system: serial signals sent between Arduino computer processing system through pins 2&3
- 9-pin D-Sub connector to link Arduino to joystick
- 9-V battery to power wireless mechatronic system
- Computer receives RF serial signal through Processing software - language is the same as Arduino but it can interface to Wi-Fi through our programming architecture
- Computer communicates control to AR Drone through Wi-Fi

Bill of Materials

Table: Bill of Materials.

Sl. No.	Component	Price per piece	Quantity	Total Price
1.	Propeller Board of Education	$129.99	1	$129.99
2.	IR Emitter	$2.00	6	$12.00
3.	Nintendo wii camera	$25.99	1	$25.99
4.	Arduino micro	$25.57	1	$25.57
5.	XBee Explorer Regulated	$17.95	3	$53.75
6.	SparkFun XBee Shield	$14.95	1	$14.95
7.	Joystick	$8.45	1	$8.45
8.	Experiment breadboard	$5.88	1	$5.88
9.	DB9 RS232 Serial Cable	$3.49	1	$3.49
10.	USB cable	$4.67	1	$4.67
11.	9 Volt battery	$2.99	1	$2.99
12.	Yellow LED	$0.99	1	$0.99
13.	Parrot AR drone Quadricopter	$0.00	1	$0.00
	TOTAL PRICE			***$288.82***

Propeller Codes for camera

```
cogstart(&autoLanding, NULL, stack1, sizeof(stack1));
cogstart(&joystickControl, NULL, stack2, sizeof(stack2));
```

```
void autoLanding(void *par){

  while(1){
    j=0;
    if(fdserial_rxReady(xbee)!=-1){
    do{
     ch=fdserial_rxChar(xbee);
     }while(ch!='\n');

    j=0;
    do{
     ch=fdserial_rxChar(xbee);
     x1[j]=ch;
     j++;
     }while(ch!='\t');
    j=0;
    do{
     ch=fdserial_rxChar(xbee);
     y1[j]=ch;
     j++;
     }while(ch!='\t');
```

Xbee receives characters and put them into char array.

```
X1=atoi(x1);
Y1=atoi(y1);
X2=atoi(x2);
Y2=atoi(y2);
X3=atoi(x3);
Y3=atoi(y3);
```

Strings are changed to int type for further calculation.

Propeller Codes for camera

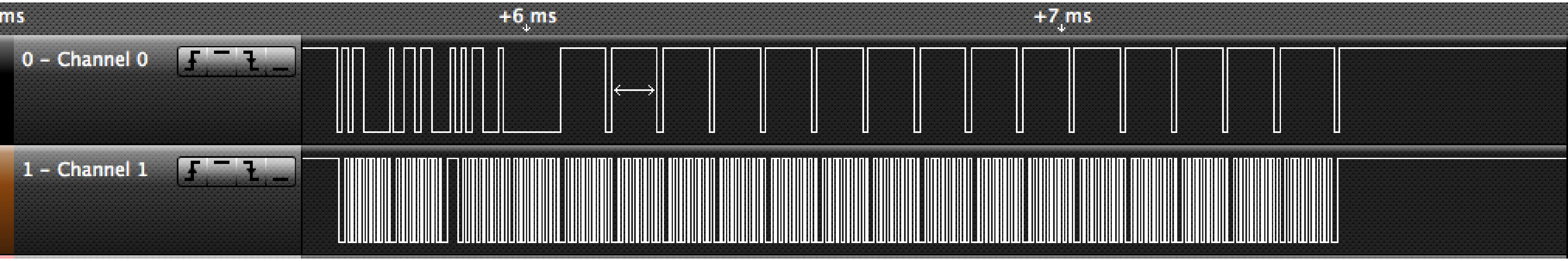
```
#include "my_includes_.h"
```

Include this h file for ADC pins.

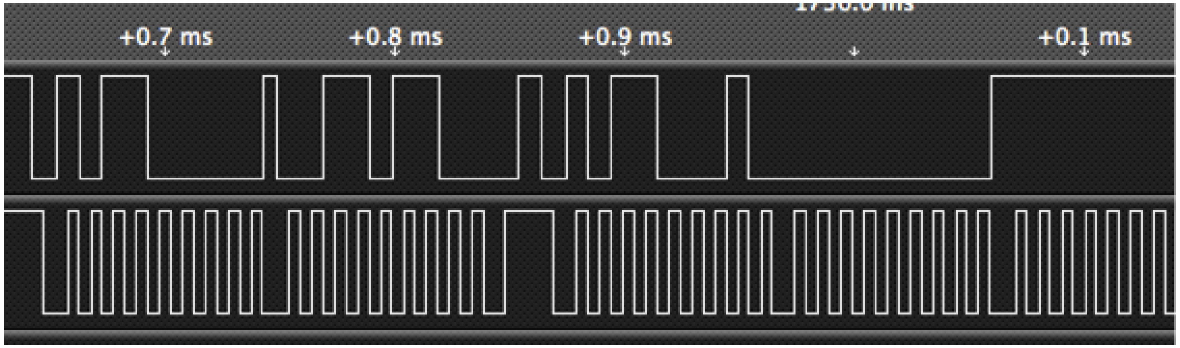
```
int map(int number, int min, int max, int maxRange, int minRange);
```

```
void joystickControl(void *par){
  while(1){
    pause(50);
    RL = read_adc_i2c(4)&0x3FF;
    BF = read_adc_i2c(1)&0x3FF;
    UD = read_adc_i2c(2)&0x3FF;

    if(UD<350){
      ud = map(UD, 125, 350, 1, 5);
    }else if(UD>400){
      ud = map(UD, 400, 653, -5, -1);
    }else{
      ud = 0;
    }

    if(RL<300){
      rl = map(RL, 35, 300, 1, 5);
    }else if(RL>400){
      rl = map(RL, 400, 550, -5, -1);
    }else{
      rl = 0;
```

Wii camera I2C




```
Ix1 = data_buf[1];
Iy1 = data_buf[2];
s   = data_buf[3];
Ix1 += (s & 0x30) <<4;
Iy1 += (s & 0xC0) <<2;
Ix2 = data_buf[4];
Iy2 = data_buf[5];
s   = data_buf[6];
Ix2 += (s & 0x30) <<4;
Iy2 += (s & 0xC0) <<2;
Ix3 = data_buf[7];
Iy3 = data_buf[8];
s   = data_buf[9];
Ix3 += (s & 0x30) <<4;
Iy3 += (s & 0xC0) <<2;
```

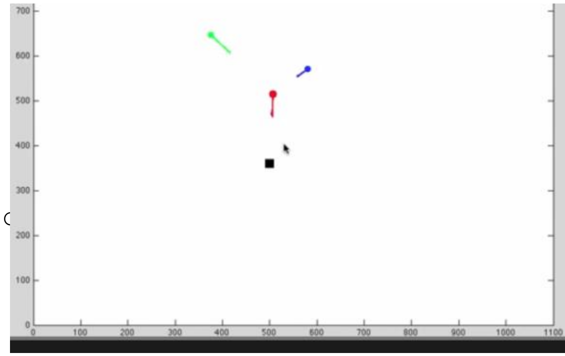


0x30=> 00011110
0xC0=>11000000

```
(s & 0x30) <<4 => 960
(s & 0xC0) <<2 => 768
```

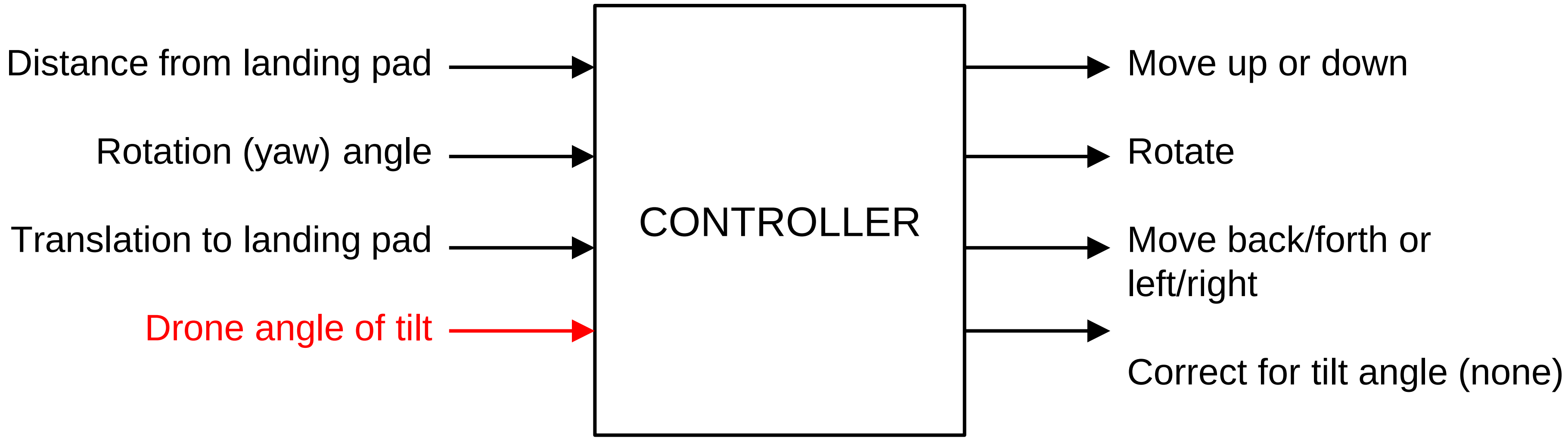
Drone Autonomous Landing

```
if(x1a>450 && x1a<550 && y1a>310 && y1a<410)
//seeing it red point 1 is close to center
      {
        printf("x1a found");
        if(d1a>=220)
        {
          printf("land");
          ud=0; rl=0;bf=0;c=0; d=0;
        }
        else if (d1a<220)    //d1a=220
        {
          printf("lowering");
          if(c!=0){ud=-1;rl=0;bf=0;c=100;};
        }
       }
      else if ((x1a-500)>50)
      {
        printf("go left");
        if(c!=0){ud=0;rl=-1;bf=0;c=100;};
        if ((360-y1a)>50)
        {
          printf("go forward");
            if(c!=0){bf=-1;}
        }
        else if ((y1a-360)>50)
        {
          printf("go back");
          if(c!=0){bf=1;}
        }
      }
     else if ((500-x1a)>50)
      {
        printf("go right");
        if(c!=0){ud=0;rl=1;bf=0;c
        if ((360-y1a)>50)
        {
          printf("go forward");
          if(c!=0){bf=-1;}
        }
        else if ((y1a-360)>50)
        {
          printf("go back");
          if(c!=0){bf=1;}
        }
      }
      else if ((360-y1a)>50)
      {   if(c!=0){ud=0;rl=0;bf=-
1;c=100;};
        printf("go forward");
      }
      else if ((y1a-360)>50)
      {
        if(c!=0){ud=0;rl=0;bf=1;c=100;};
        printf("go back");
      }
      else
if(c!=0){ud=0;rl=0;bf=0;c=100;}
    }
```



Results-1

- We have developed an autonomous landing pad that has capabilities to steer the drone and land it on the magnets of the landing pad
  - The propeller can control the drone through input from the joystick
  - With a good hand on the manual controller, propeller successfully transfers to automatic control and the AR Drone lands accurately.
- Our automatic landing controller is a multiple input multiple output (MIMO) system.
  - Angle of tilt not accounted for: causes instability
  - Develop and implement linear quadratic regulator (LQR)



Results-2

Future Improvements-2

- Because of many crashes in testing the AR Drone, the blades are damaged and need replacement. There may be additional damages as well.
  - Solution: Buy a new AR Drone
- The current control system has not been tested enough to optimize the current control configuration
  - Solution: Perform more testing
- The electronics onboard AR Drone are off center from the COM and cause possible drift.
  - Solution: Modify electronics to fit in the middle
- MIMO systems perform better with a linear quadratic regulator (LQR)
  - Solution: Implement LQR with control scheme
- There are 3 inputs and 4 outputs on the MIMO loop
  - Solution: Use another sensor to monitor and correct for tilt angle