Appendix:

A Final code for all systems

```
#include <EYW.h>


//All of this is the set up library identifying the different sensors and naming them.
EYW::Camera PayloadServo;
EYW::RangeFinder rsensor;
int distance =0;


EYW:: Altimeter myaltimeter;
float current_height=0;
//This is the timer since the other one did not work.
unsigned long timerstart=0;
unsigned long duration=0;
void beginTimer(int dur)
{
  timerstart = millis();
  duration = dur;
}
bool checkTimer()
{
  if ((millis()-timerstart) > duration){
```

```
    return true;
  } else {
    return false;
  }
}

void setup() {
  // put your setup code here, to run once
PayloadServo.begin();
 // first it sounds a light
PayloadServo.calibrate();
 // the servo moves
PayloadServo.alarm();
 // the alarm sounds then the timer begins
PayloadServo.beginTimer(1000);

// This is the rangefinder calibration sequence
Serial.begin(9600);
rsensor.begin();
rsensor.alarm();
rsensor.calibrate(10);
```

```
//This is the altimeter calibration sequence
Serial.begin(9600);
myaltimeter.begin();
myaltimeter.calibrate(10);
myaltimeter.alarm();
}

void loop()
  // put your main code here, to run repeatedly:

  //This code lets the servo know to move once the height gets over the specified height below.
And only if the height is over that will the servo take the picture
{
current_height = myaltimeter.getHeightAvg (20);
if (PayloadServo.timerExpired()==true && current_height >= 10)
 {
 //PayloadServo.getPicture();
  PayloadServo.getPicture();

//If the distance is less than the specified height below in the if statement then the range finder
will go off.
```

```
distance = rsensor.getDistance();
Serial.print("Current Distance: ");
Serial.println(distance);

if (distance <10) {
  rsensor.alarm(2,1000, 250);
}
// This is the code for the altimeter to read the current height that the aerial imaging device is at.
current_height = myaltimeter. getHeightAvg (20);
Serial.print("Current Height: ");
Serial.println(current_height);

//This code actually gets the servo to work with the altimeter so that if the camera reaches 10
meters the servo will trigger and take pictures and the altimeter alarm will go off at 10 meters.

if(current_height>10) {
  PayloadServo.getPicture();
  myaltimeter.alarm(6,4000,100);
   beginTimer (1000);
}
}
```

B=Descent subsystem Code for range finder

```
#include <EYW.h>
//labeling the rangesensor and saying initial distance is 0


EYW::RangeFinder rsensor;
int distance =0;
void setup() {
  // put your setup code here, to run once:
// This is the beginning code that should happen when the arduino turns on. Rangefinder
calibrates and starts its process.
Serial.begin(9600);
rsensor.begin();
rsensor.alarm();
rsensor.calibrate(10);
}
//This tells it to do this process of sensing objects if front of the rangefinder over and over agian.
void loop(){
distance = rsensor.getDistance();
Serial.print("Current Distance: ");
Serial.println(distance);
// this says that if the distance is less than 30 meters the rangefinder should sense and go off.
if (distance <30) {
  rsensor.alarm(2,2000, 1000);
```

```
}


}
```

C= Code for subsystem structure and altimeter

```
#include <EYW.h>
//This is naming the altimeter.


EYW:: Altimeter myaltimeter;


float current_height=0;
void setup() {
//This starts the altimter calibration.
  // put your setup code here, to run once:
Serial.begin(9600);
myaltimeter.begin();
myaltimeter.calibrate(100);
myaltimeter.alarm();
}


void loop()
```

```
//This starts the loop of sensing the altitude and should contiune to go off as long as the altimeter
is higher than 0.5 meters
{
current_height = myaltimeter. getHeightAvg (20);
Serial.print("Current Height: ");
Serial.println(current_height);

if(current_height>.5)
{
  myaltimeter.alarm(6,2000,500);
}
}
```

D= Payload subsystem Code for servo

```
// this is including the EYW library
#include <EYW.h>
#include <Servo.h>
#include <Wire.h>

#include <EYW.h>
```

```
// this is naming the camera
EYW:: Camera PayloadServo;
void setup() {
// this is calibrating the servo
  // put your setup code here, to run once:
PayloadServo.begin();
PayloadServo.calibrate();
PayloadServo.alarm();
PayloadServo.beginTimer(10000);


}


void loop()
// This is putting the servo on a timer of the servo arm triggering every 5 seconds
  // put your main code here, to run repeatedly:
{
if (PayloadServo.timerExpired()==true || PayloadServo.buttonPressed()==true)
 {
 PayloadServo.getPicture();
 PayloadServo.beginTimer(5000);
 }
```

}