DUSTIN MENDOZA

Engineering Portfolio

dmendoza@alaska.edu

Introduction

My name is Dustin Mendoza, and I am an engineering student in the College of Engineering at the University of Alaska Anchorage. This portfolio aims to supplement my resume with examples of my previous work.

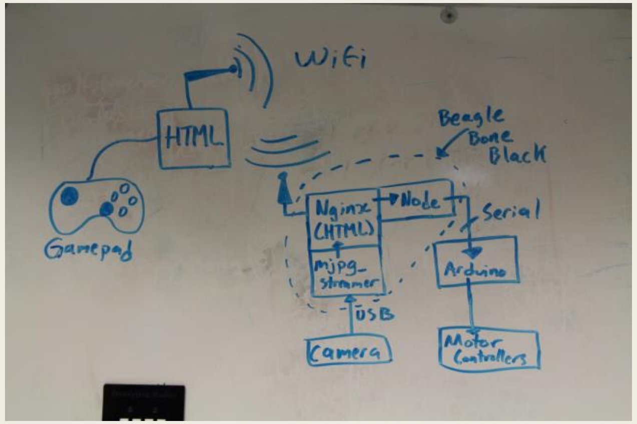
I believe the best quality of my work is simplicity and elegance in design. During my free time I mentor students in robotics, help work on the controls system for the UAA rocket club's rocket and work on side projects.

Summer 2014

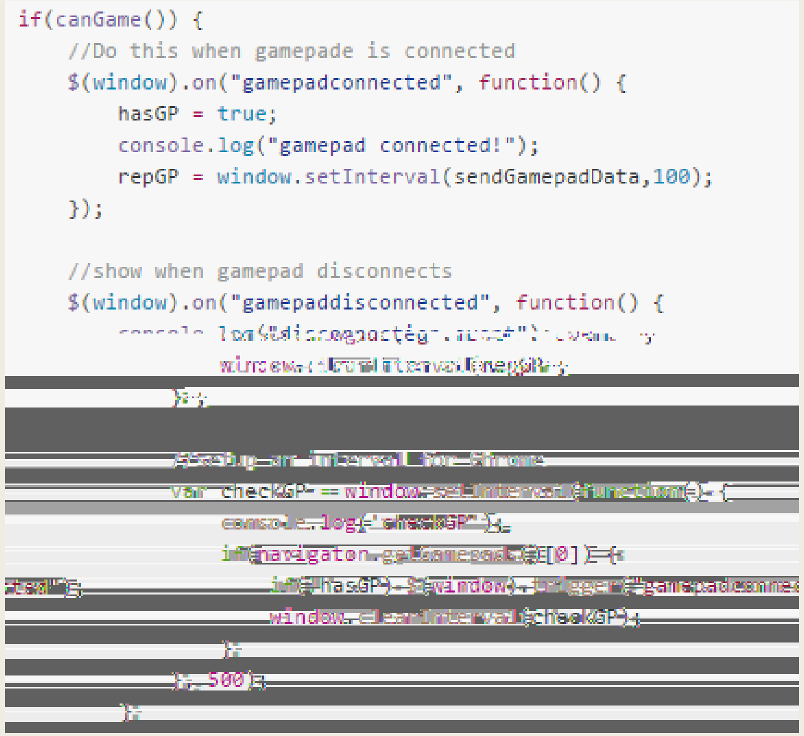
Wolf Tracker

This project aims to help scientists at the California Wolf Center to remotely observe wolves. I joined the team in the last 2 weeks of the program to help rebuild the controls system for the robot from the ground up. We decided to scrap the old design, which relied heavily on QT and utilized an Intel Development board. We replaced it with a lighter Beagle Bone Black and an Arduino for the Motor controls. Individually, I worked on writing the video streaming code, motor controls, web user interface, and the Arduino to Beagle Bone communication.

High Level Diagram

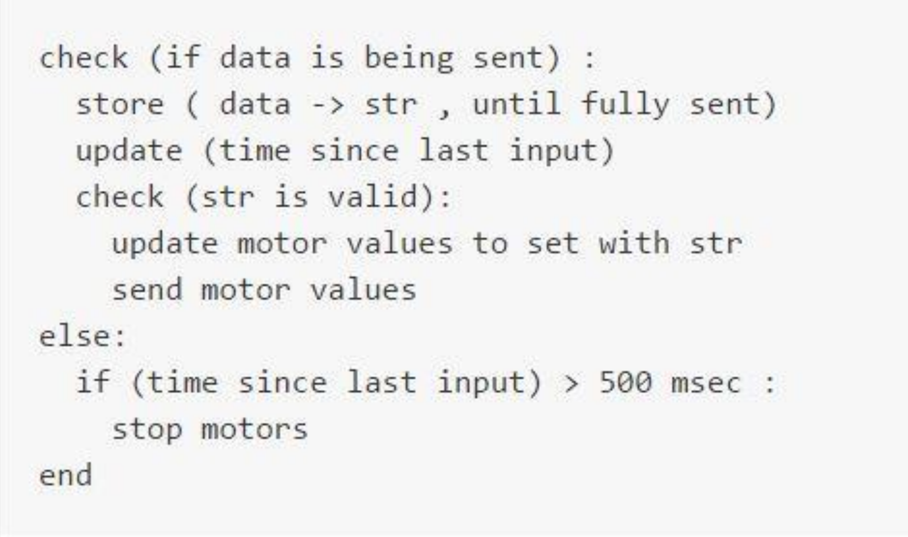


UI Webpage Snippet



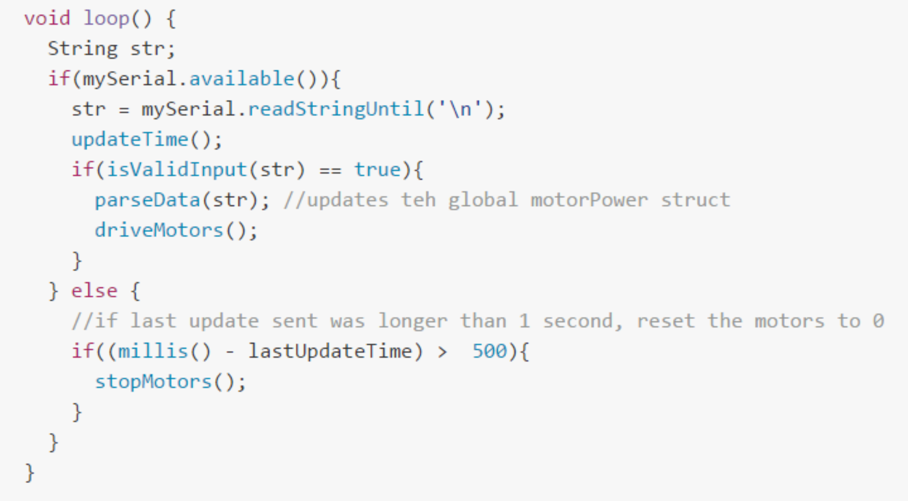
```
if(canGame()) {
    //Do this when gamepade is connected
    $(window).on("gamepadconnected", function() {
        hasGP = true;
        console.log("gamepad connected!");
        repGP = window.setInterval(sendGamepadData,100);
    });

    //show when gamepad disconnects
    $(window).on("gamepaddisconnected", function() {
        console.log("disconnected [illegible]");
        window.clearInterval(repGP);
    });

    //setup an interval for Chrome
    var checkGP = window.setInterval(function() {
        console.log('checkGP');
        if(navigator.getGamepads()[0]) {
            if(!hasGP) $(window).trigger("gamepadconnected");
            window.clearInterval(checkGP);
        }
    }, 500);
}
```

Motor Control Pseudocode



```
check (if data is being sent) :
   store ( data -> str , until fully sent)
   update (time since last input)
   check (str is valid):
      update motor values to set with str
      send motor values
else:
   if (time since last input) > 500 msec :
      stop motors
end
```

Arduino Snippet

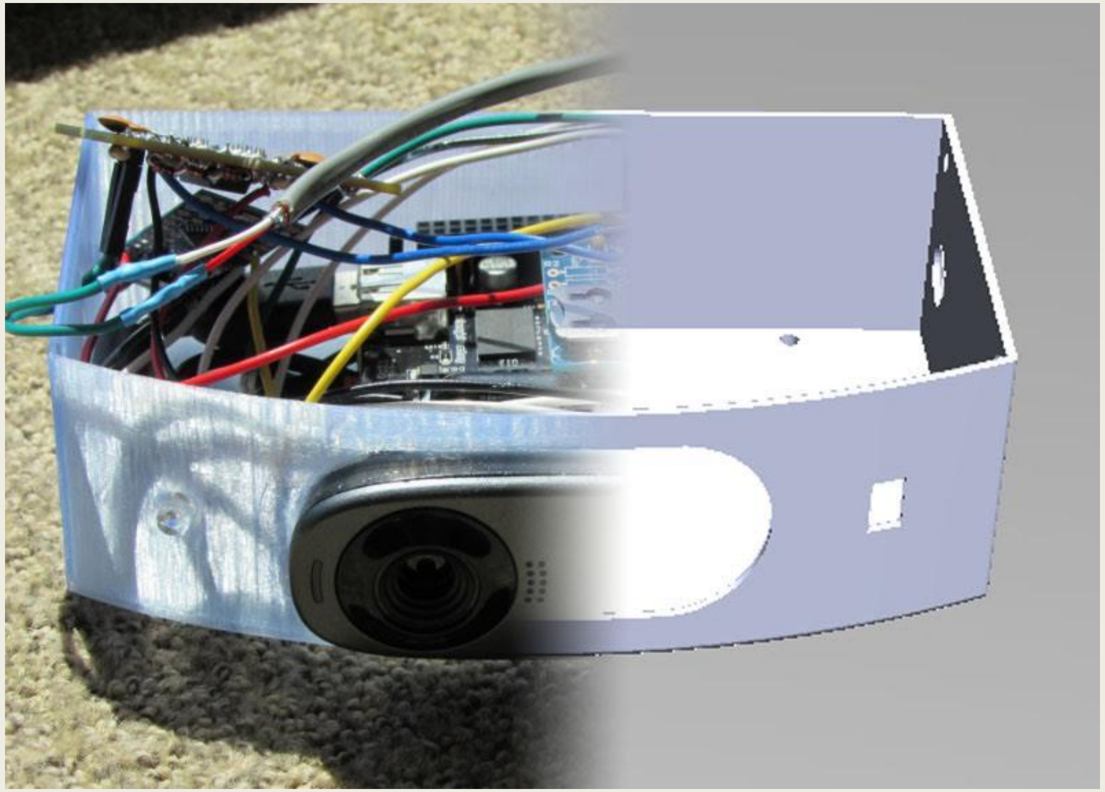


```
void loop() {
  String str;
  if(mySerial.available()){
    str = mySerial.readStringUntil('\n');
    updateTime();
    if(isValidInput(str) == true){
      parseData(str); //updates teh global motorPower struct
      driveMotors();
    }
  } else {
    //if last update sent was longer than 1 second, reset the motors to 0
    if((millis() - lastUpdateTime) >  500){
      stopMotors();
    }
  }
}
```

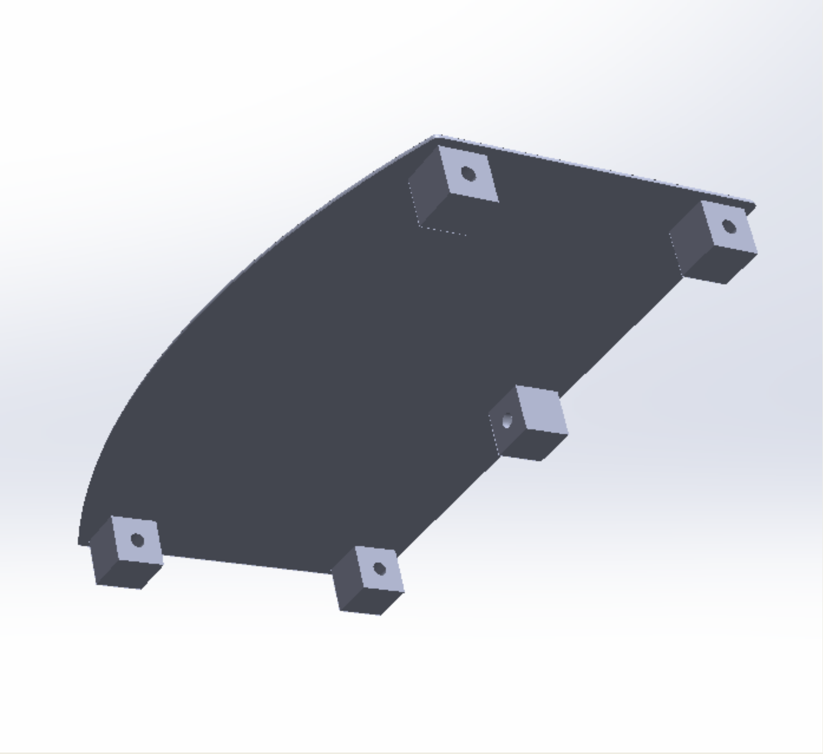
Summer 2014

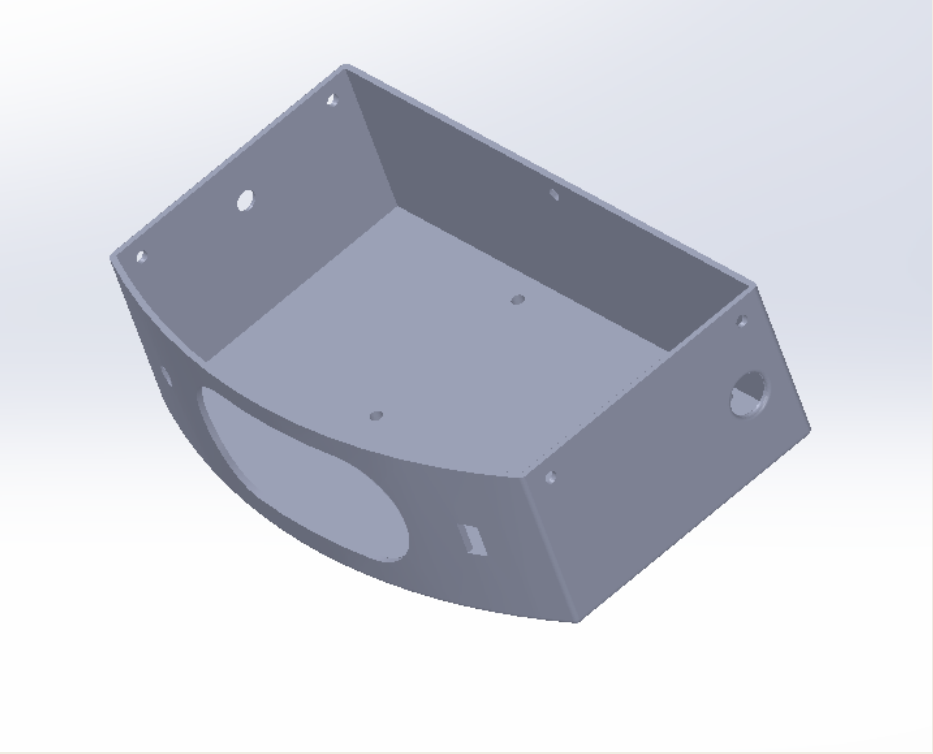
Angry Birds aka Bird Collision Detection and Recording

Every year, thousands of birds collide into the windows of homes and businesses. This autonomous system helps researchers empirically collect and record the number of bird collisions. With this system, researchers would be able to determine the effectiveness of different methods in reducing bird collisions. During the course of the project, I developed the Makefiles and config scripts, contributed to the code base, and was the Github maintainer. I helped create a protective housing for the system in SolidWorks and an IR remote that would turn off the system when the windows were cleaned to avoid recording false positives.



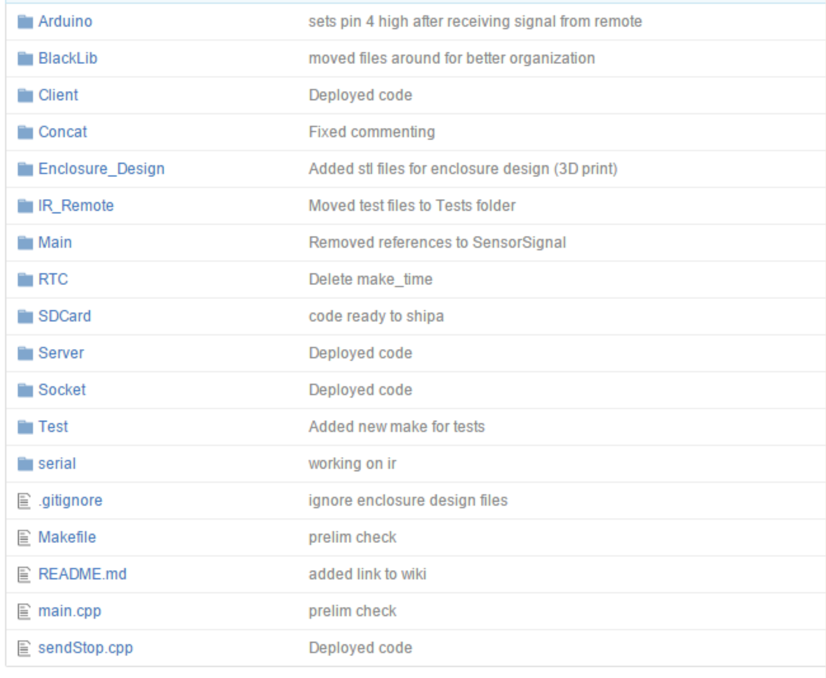
Enclosure Design in SolidWorks





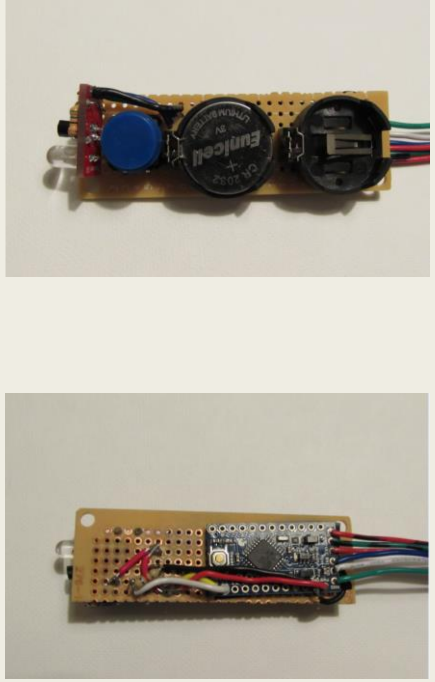
I designed our code base so that it could be easily reused in future projects and ensured it was modularized so everyone on the programming team could program concurrently. I worked on writing the data to the SD-card, communicating between the beagle bone and Arduino remote control system, constructing the IR remote, and programming the overall codebase for the logging/detection program.

Codebase Overview



Arduino	sets pin 4 high after receiving signal from remote
BlackLib	moved files around for better organization
Client	Deployed code
Concat	Fixed commenting
Enclosure_Design	Added stl files for enclosure design (3D print)
IR_Remote	Moved test files to Tests folder
Main	Removed references to SensorSignal
RTC	Delete make_time
SDCard	code ready to shipa
Server	Deployed code
Socket	Deployed code
Test	Added new make for tests
serial	working on ir
.gitignore	ignore enclosure design files
Makefile	prelim check
README.md	added link to wiki
main.cpp	prelim check
sendStop.cpp	Deployed code

IR remote



Finished Design