MCECS Bot

ECE 578 | 11.15.2012

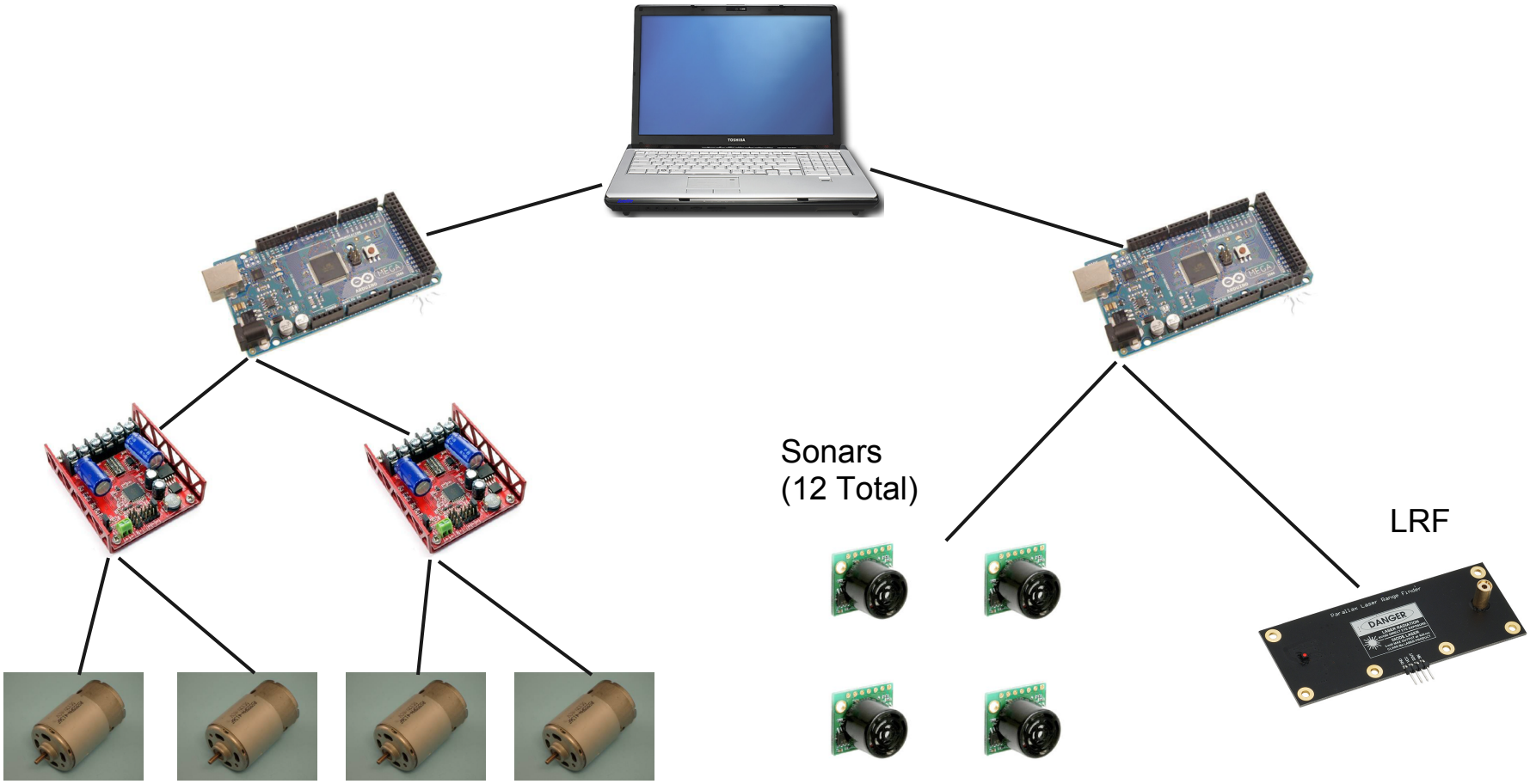
Mitch Barton
Jeramy Barrett
Conor O'Connel
Jesse Adams
Phil Lamb

MCECS Bot Overview

Systems Overview

- 12 sonar sensors
- Laser range finder
- Collision detecting bumper buttons
- Wheel encoders
- 4 Omni-wheels
- 4 motors
- 2 motor controllers
- Arduino Uno
- Arduino Mega 2560

MCECES Bot Connectivity

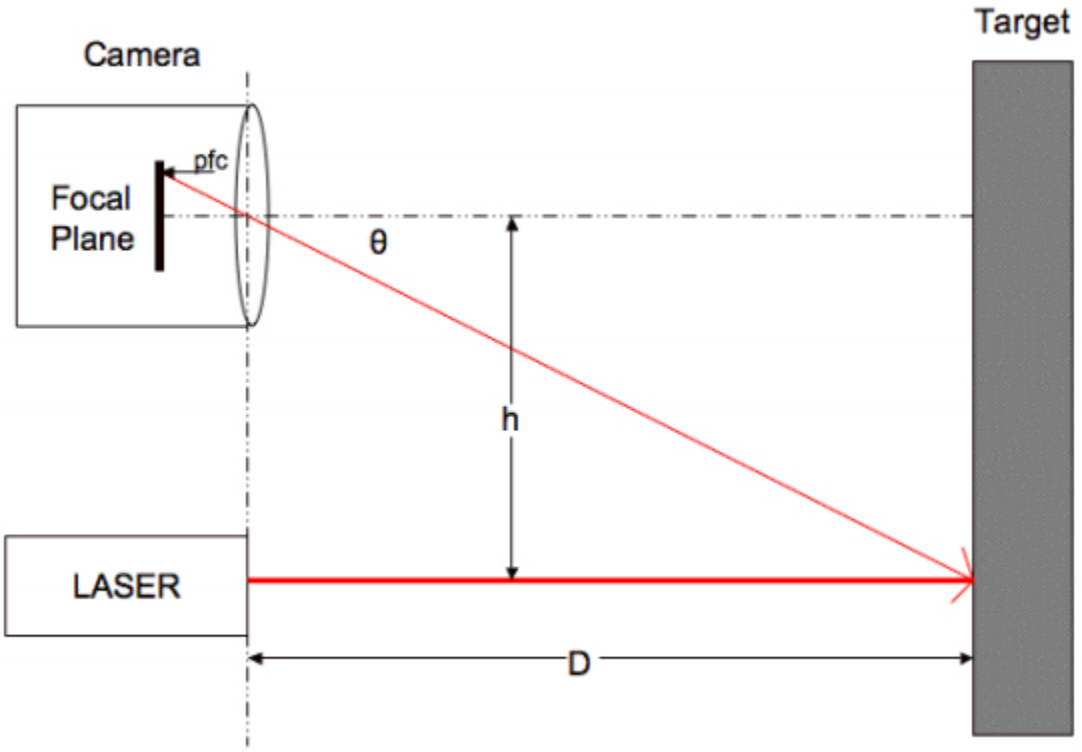


MCECS Bot Status

- Concerning the overall development of the MCECS bot, we determined that all modules (sensors and actuators) should be installed and functioning at a basic level before proceeding with integration.
- The motors are working with raw commands, but need to be integrated with the encoders such that we can abstract the motor commands at a higher level, i.e. go forward 1ft rather than wheel 1, 2, 3, and 4 move forward at 4ft/min for 15sec, etc.
- Bumpers are installed but need to be programmed and tested.
- LRF is ready for MCECS bot integration, with minimal code left to write.
- The sonar sensors have been tested preliminarily, but still need to be integrated and tested all together.

LRF Theory of Operation

- Fixed offset between camera and laser (h)
- As D increases, so does the laser spot's distance from the center of the FPA, or pixels from center (pfc).
- A linear relationship between pfc and Θ, of the form y=mx+b, is derived from experimental data.
- Once the laser spot is seen by the camera, Θ can be calculated and passed to the trigonometric function to determine D.



LRF Theory of Operation

Once the laser spot is seen by the camera, there are additional image processing steps taken to maximize accuracy, using the Parallax propeller –Q44.

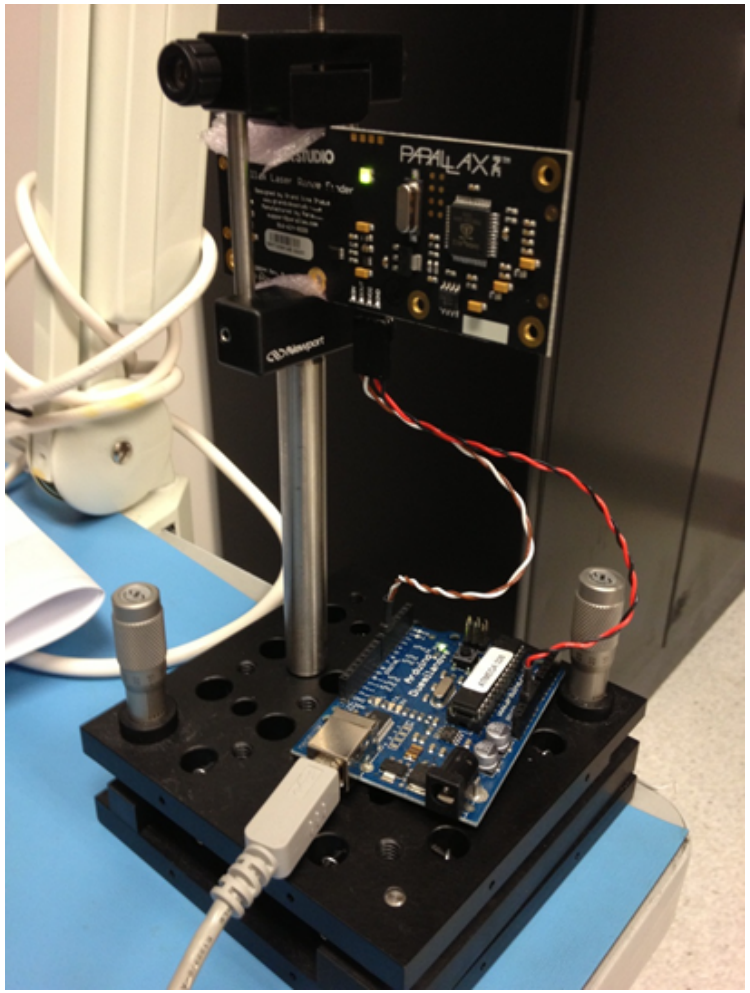
1. Background subtraction
2. Thresholding
3. Column sum
4. Blob detection
5. Mass/Centroid calculation

LRF Serial Command Set

- R – single range measurement (4 digit decimal in mm)
- B – single range measurement (4 byte binary in mm)
- L – repeated range measurement
- E – adjust camera for current lighting conditions
- S – reset to default settings
- V – print version info
- H – print list of available commands
- O – display coordinate, mass, and centroid info
- X – calibrate camera system for range finding
- G – capture and send single frame (greyscale 160p x 128p)
- C – capture and send single frame (YUV422 640p x 16p)
- P – capture and send single background subtracted frame (YUV422 640p x 16p)

LRF Initial Testing

- Initial testing performed using an Arduino Duemilanove (Uno essentially) to power the LRF and communicate between it and the PC.
- Found that image acquisition only works at 115,200 baud, and the software serial library for the Arduino Uno does not support that speed.
- Because of this and other reasons LRF will be operated with the Arduino Mega 2560, which has multiple hardware serial ports.



LRF Initial Testing

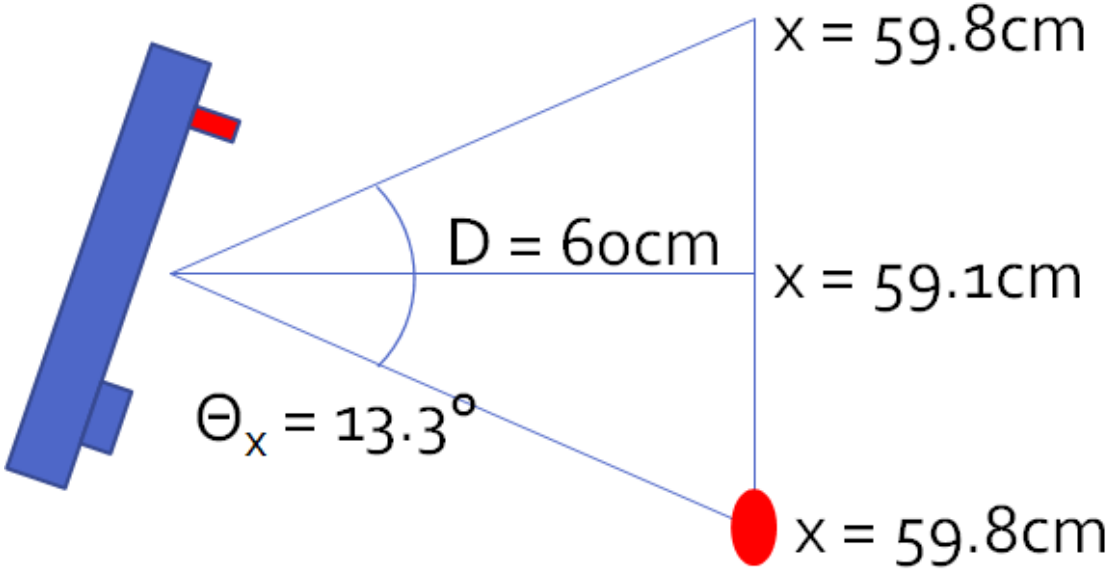
- The LRF is spec'ed to provide ranges between 15-122cm with an average error of 3%, not to exceed 5%, and I confirmed this with data below.
- I found that the maximum distance I was able to measure was 242.3cm - this was very inaccurate, but could be useful for long range object detection only.

Actual Distance (cm)	**Calculated Distance (cm)**	**Difference (cm)**	**Error (%)**
20.0 cm	20.1 cm	0.1 cm	0.50
30.0 cm	30.0 cm	0 cm	0
40.0 cm	40.1 cm	0.1 cm	0.50
50.0 cm	49.8 cm	-0.2 cm	0.40
60.0 cm	59.1 cm	-0.9 cm	1.5

Initial Testing

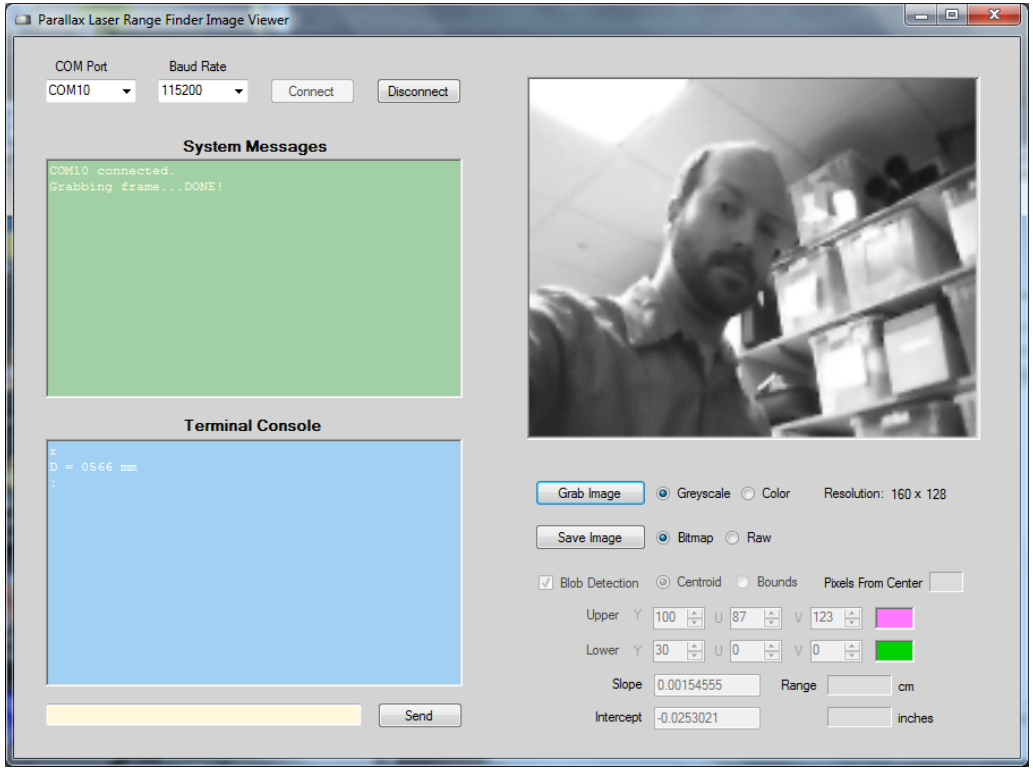
- The measured distance did not change as it should have with angle, i.e. there is an insensitivity to angular offset, and it is less accurate in azimuth, that is, the plane in which the camera and laser are the same distance from the floor.

- In azimuth, I measured the same distance, 59.1cm, through an angle of 13.3°, after which the measurement jumped to 59.8cm. In elevation the angle was only 1.64° before the measurement again jumped to 59.8cm.



LRF Initial Testing

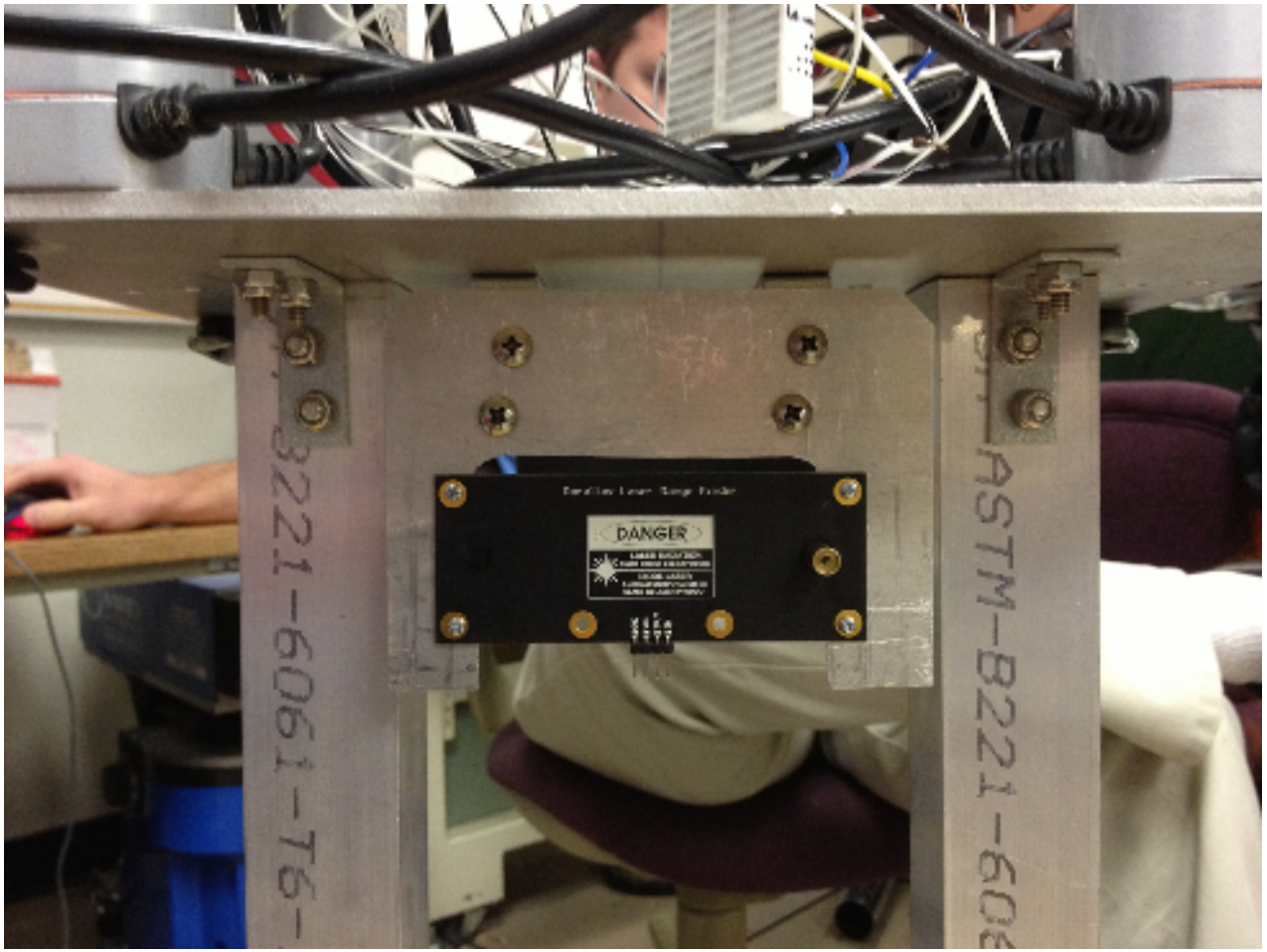
- The makers of the LRF also developed a GUI to communicate and capture images which is shown at the right.
- With the LRF running at 115,200 baud, it is possible to obtain all of the imagery off the camera.



LRF Integration

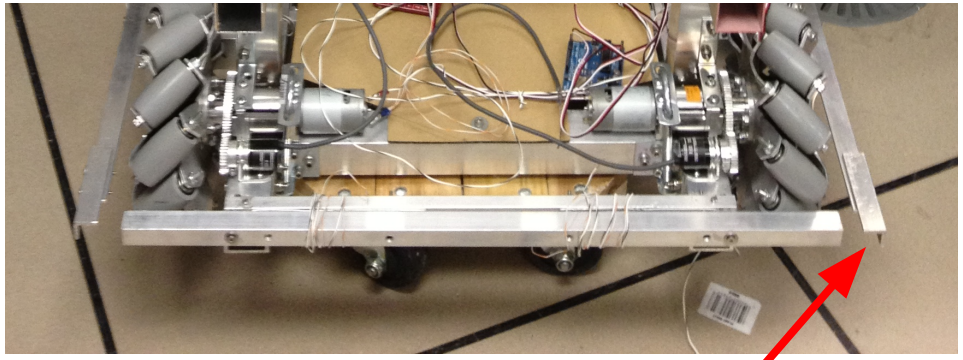
- We chose the upper body plate as the mounting point for the LRF, because that location is recessed such that the minimum distance the LRF can measure is within the volume of the robot.
- The following hardware was required to mount the sensor:
  - 2 2"x5/8" with 4 holes L-brackets
  - Custom interface plate to mate LRF with brackets

LRF Integration



LRF as mounted

Roboclaw/Bumper Integration



Ouch

Roboclaw/Bumpers

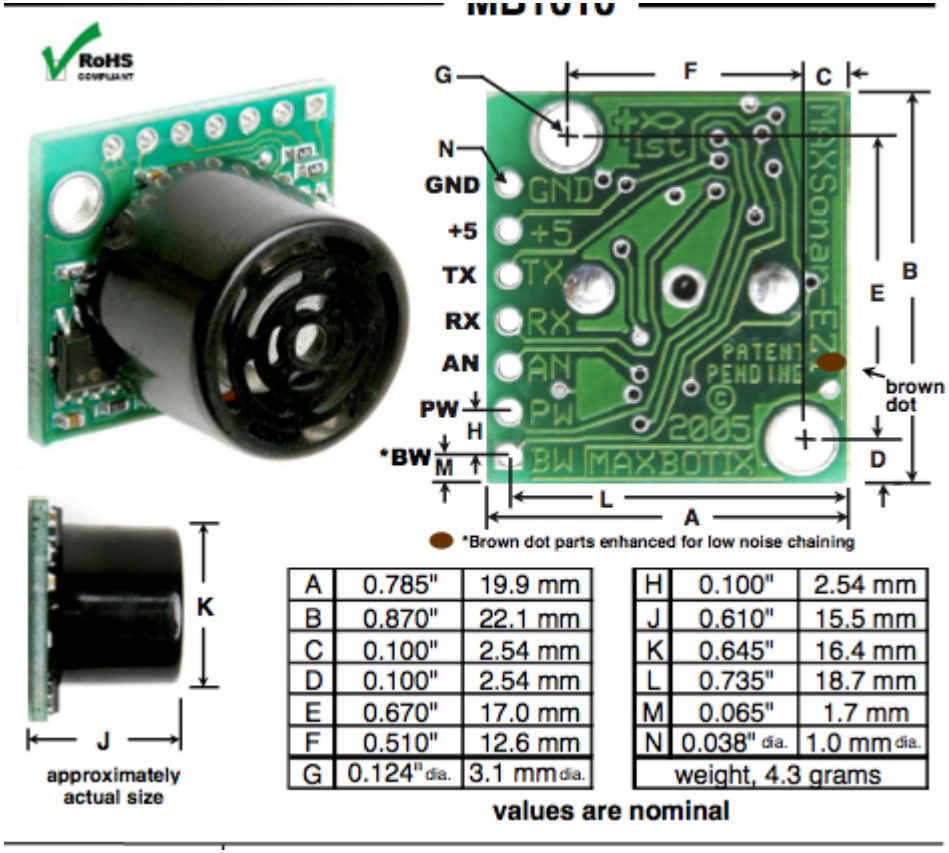
- Motor drivers and bumpers will all be connected to a single Arduiono Uno
- Four bumpers; one on each side
- Bumpers will either be connected to the kill function of the Motor Driver or to the Arduino
- Benefit of Arduino connected bumpers is the ability to latch

Roboclaw Motors/Encoders

- The encoders needed connectors for ease of testing and integration.
- We will communicate with the roboclaw motor controller in mode 4 (packet serial) using an Arduino Uno, with motor control by quadrature encoders commands.
- Goal is to integrate all motor/encoder components with the MCECS bot base, and replicate the examples from the data sheet.

Sonars

LV MaxSonars



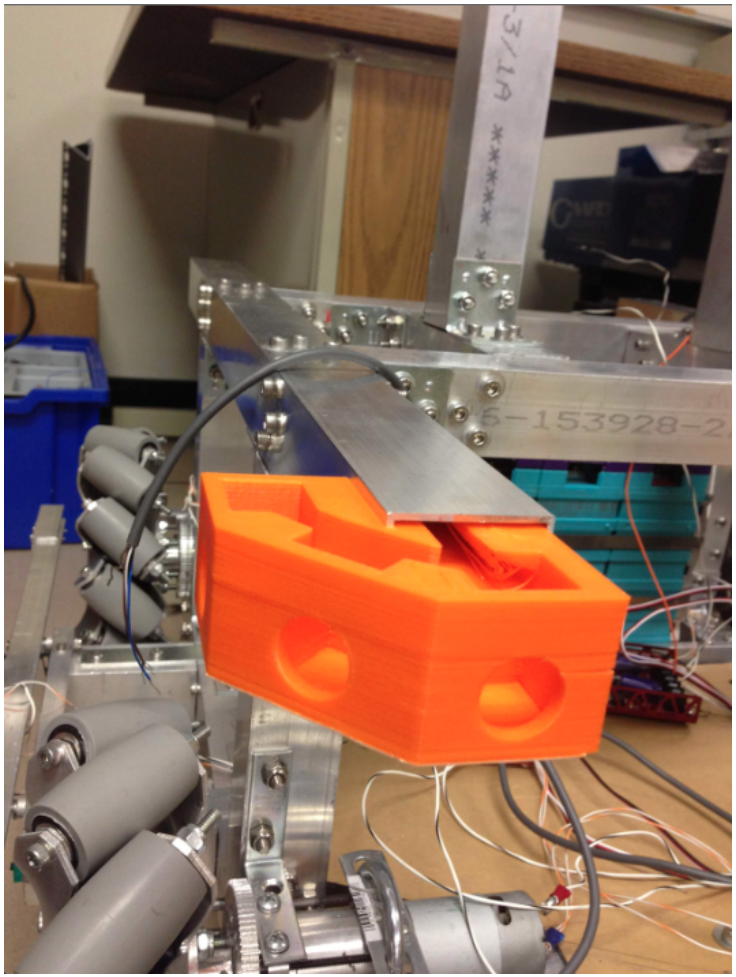
A	0.785"	19.9 mm
B	0.870"	22.1 mm
C	0.100"	2.54 mm
D	0.100"	2.54 mm
E	0.670"	17.0 mm
F	0.510"	12.6 mm
G	0.124" dia.	3.1 mm dia.

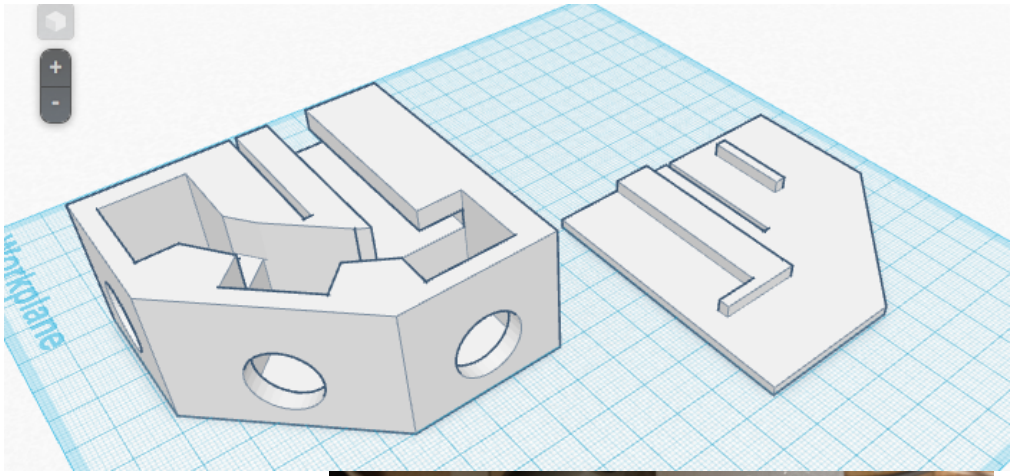
H	0.100"	2.54 mm
J	0.610"	15.5 mm
K	0.645"	16.4 mm
L	0.735"	18.7 mm
M	0.065"	1.7 mm
N	0.038" dia.	1.0 mm dia.
weight, 4.3 grams		

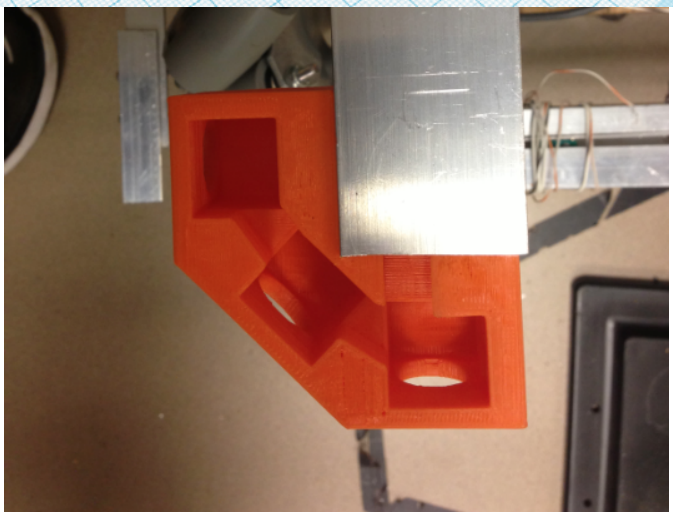
values are nominal

- Using 12 Sonars for comprehensive obstacle detection system.
- Multiple methods of interacting with sonars
- Range of 6.5 meters
- Feedback provides a distance from some object.

Sonar Mounting







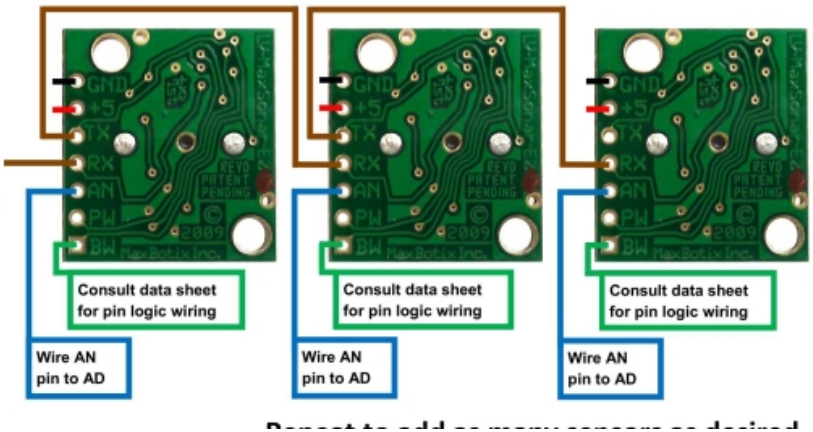
Sonar Interfacing

Challenges to overcome

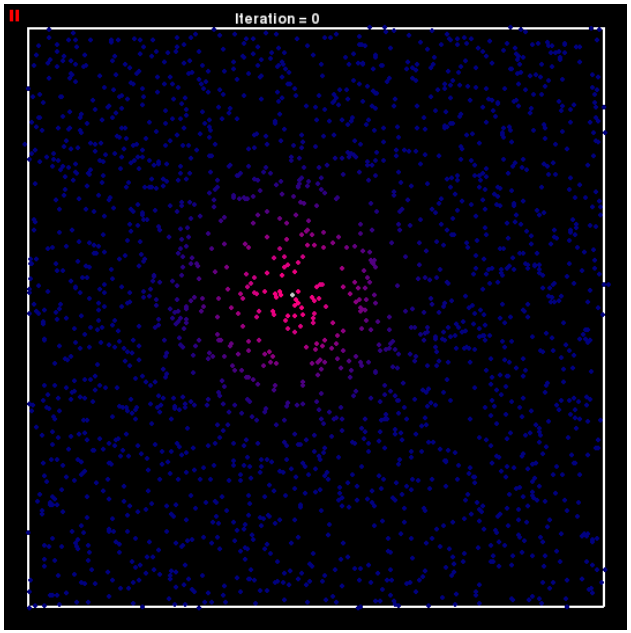
- Multiple methods of receiving data.
  - Serial
  - Pulse width
  - Analog

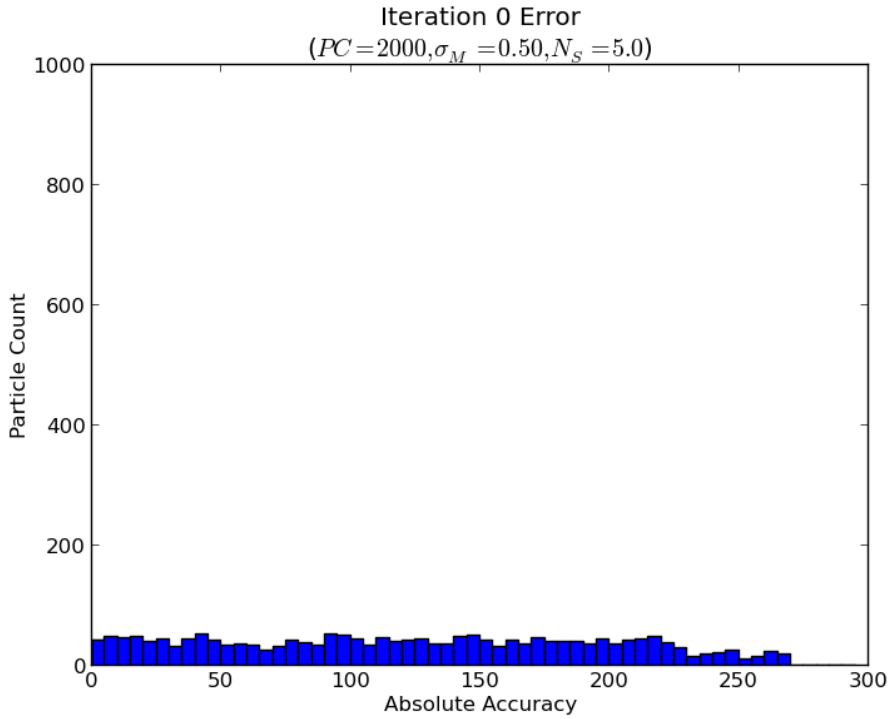
- Sonars cannot be used at the same time due to interference.

- Sonars can take readings ever 49mS

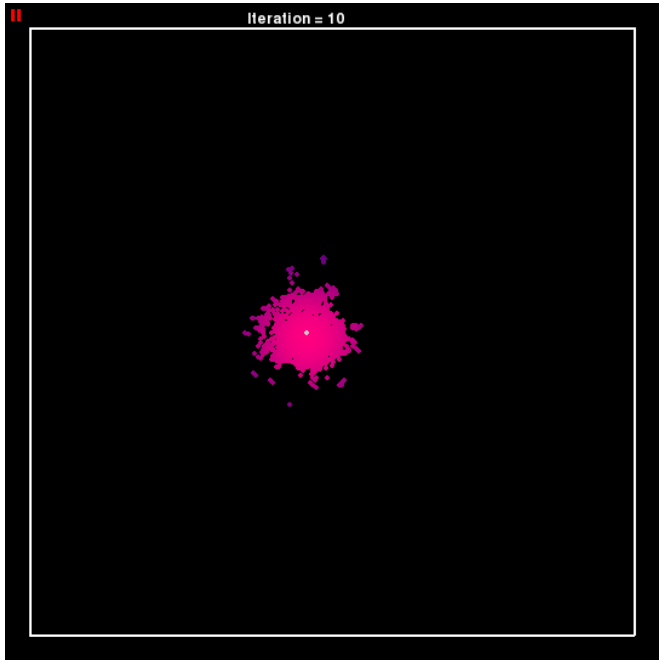


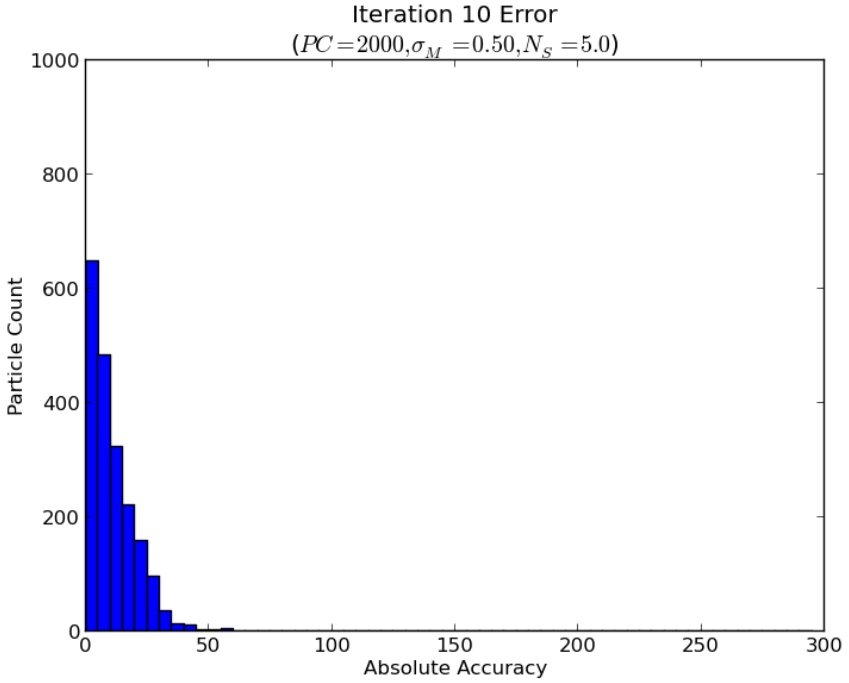
Particle Filter Simulation



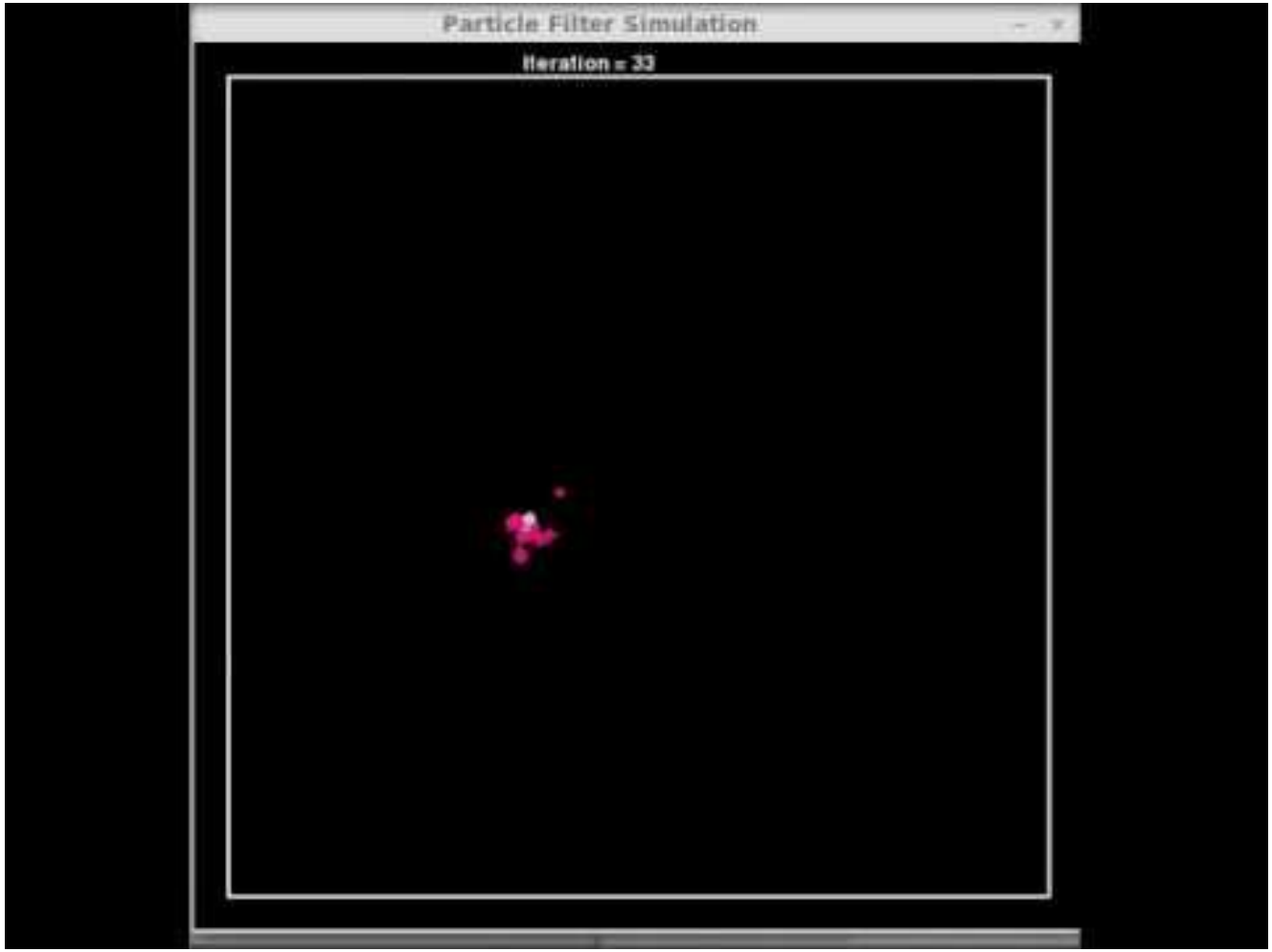


Particle Filter Simulation

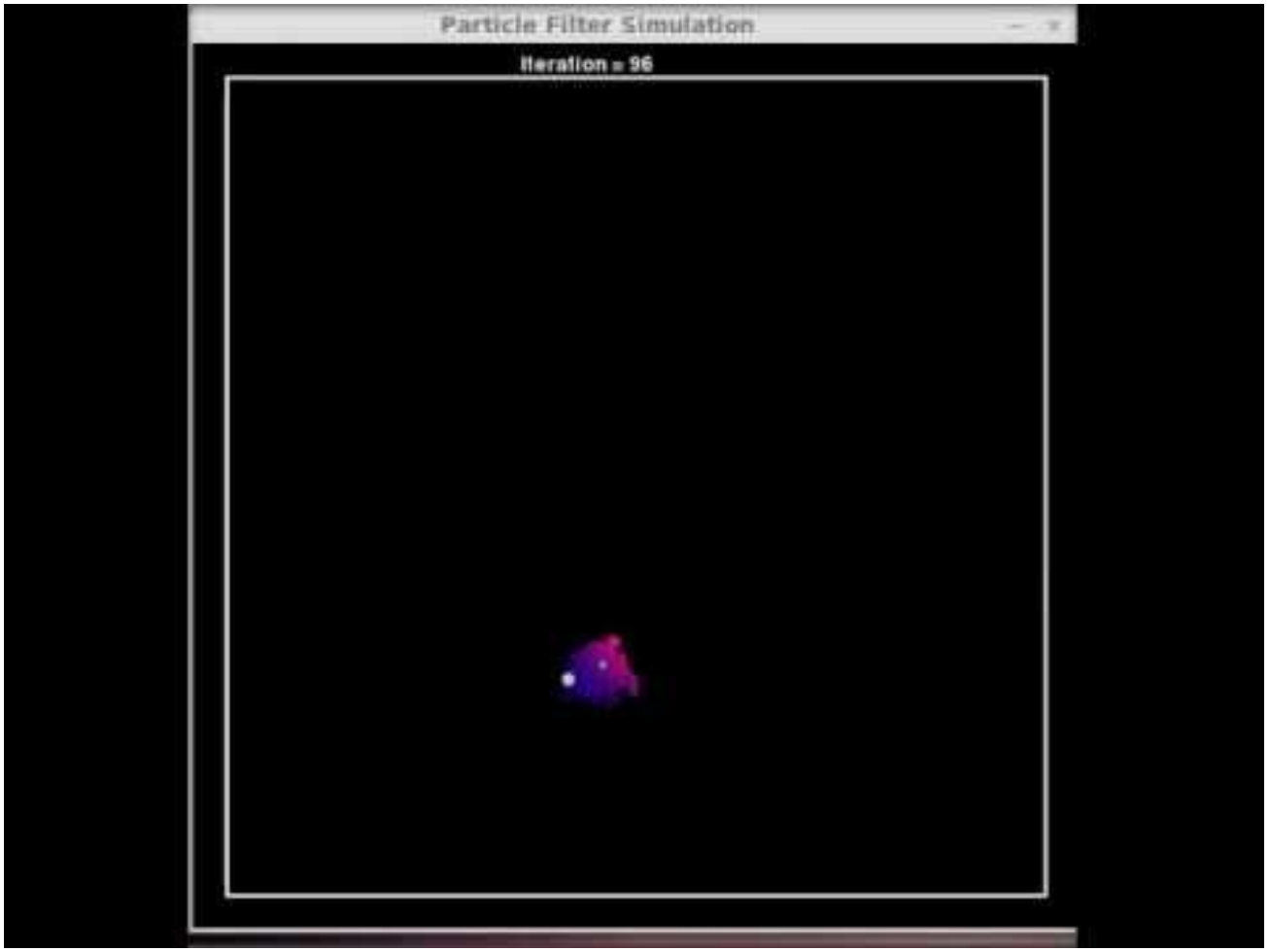




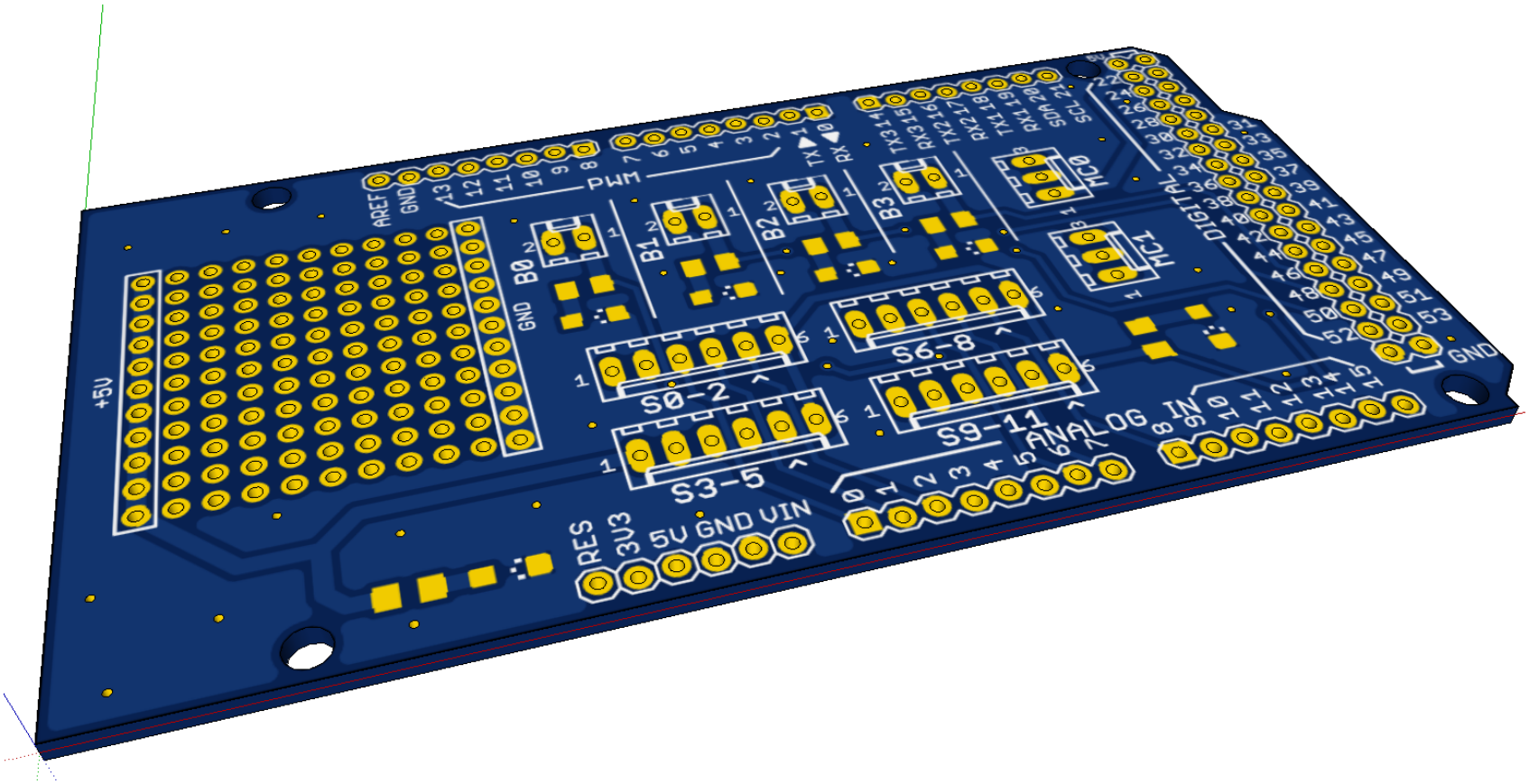
Particle Filter Simulation



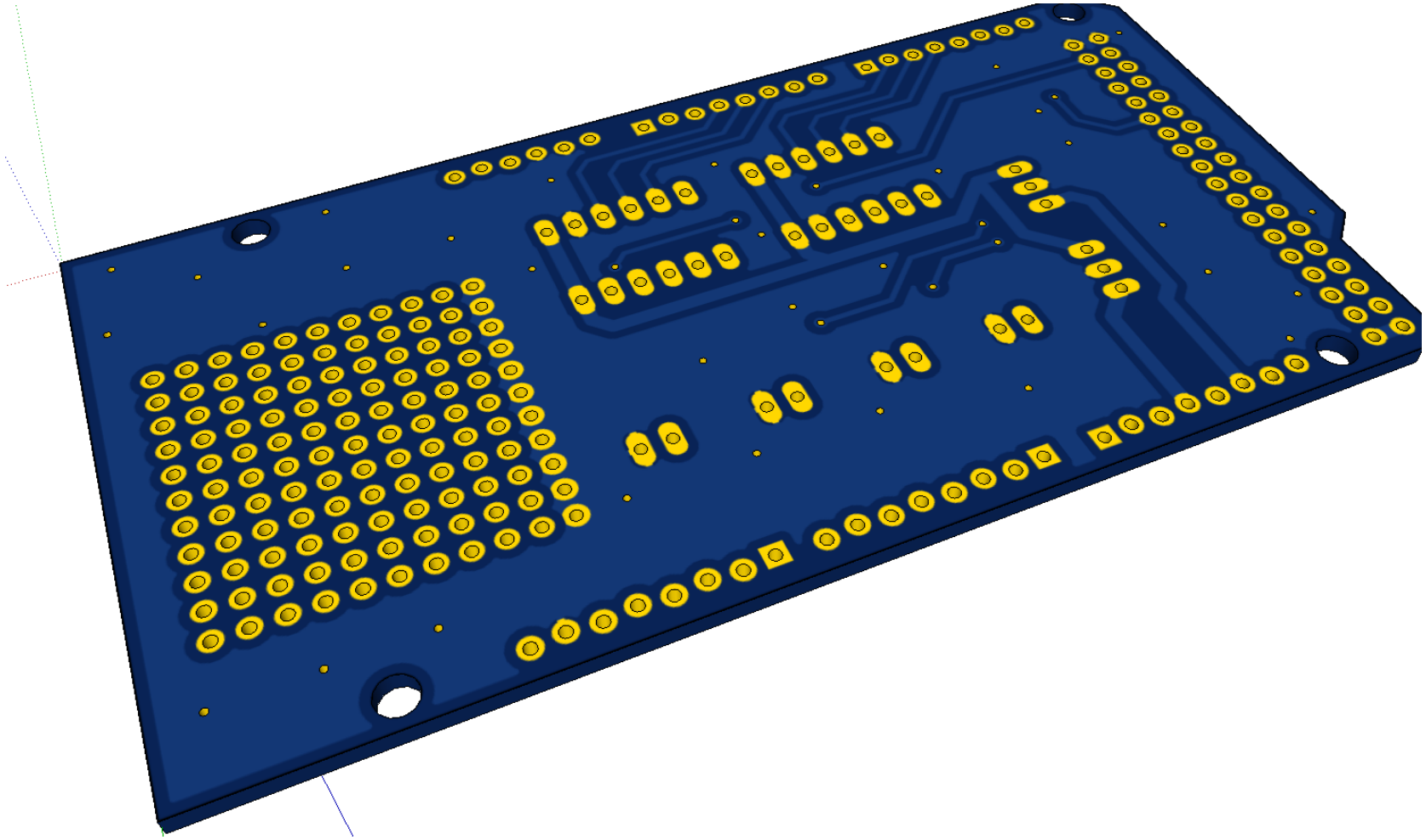
Particle Filter Simulation



Arduino/Connector Interface



Arduino/Connector Interface



Arduino/Connector Interface