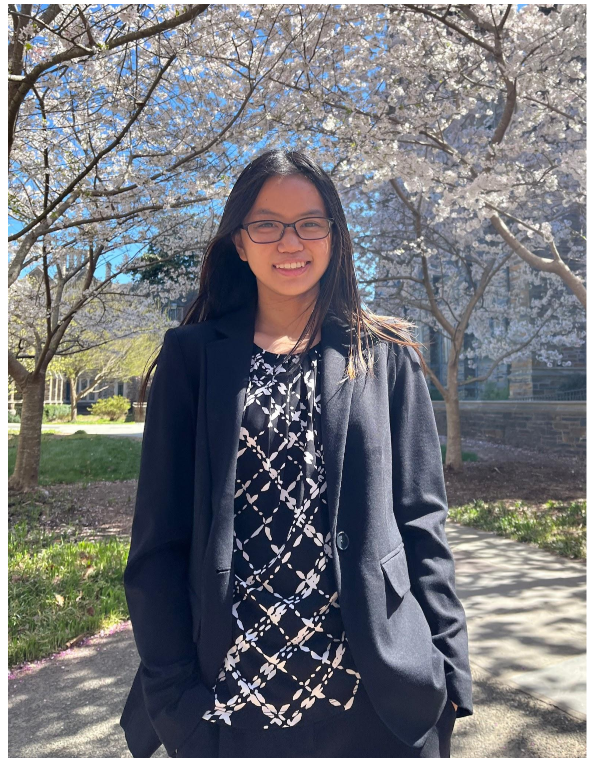
Engineering Portfolio

Lucy Huo

Table of Contents

About Me

My name is Lucy – I am an aspiring engineer looking to help others by transforming ideas into tangible solutions.



As an electrical and computer engineering student at Duke University, I get excited about design and innovation. This portfolio showcases a few of my school and personal projects, highlighting my creativity and problem-solving capabilities.

Within the ECE field, I am actively looking for opportunities to collaborate, learn, and contribute. Thank you for viewing my portfolio!

Interested in my work? Feel free to reach out!

- Email: lucyhuo25@gmail.com
- Phone: +1 972 357 0912
- LinkedIn: https://www.linkedin.com/in/lucy-huo-00a697223/
- Github: https://github.com/lucyhuo214

GardenGuru: Soil Monitoring System with Custom Microprocessor

---

Overview

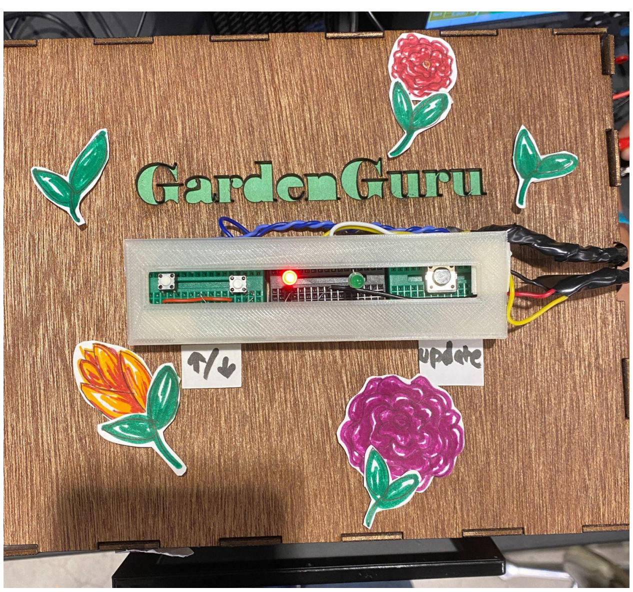
We created a soil monitoring system allowing users to track pH and moisture levels for optimal plant growth. Our solution implements a 5-stage pipelined microprocessor in Verilog supporting a custom ISA.

- Class project for ECE350: Digital Systems at Duke University, taught by Dr. John Board
- Member of a 2-person project team
- Problem statement: Plant owners have difficulty maintaining optimal soil conditions for their plants

---

Our solution: We designed and built a soil monitoring tool to support optimal plant growth.

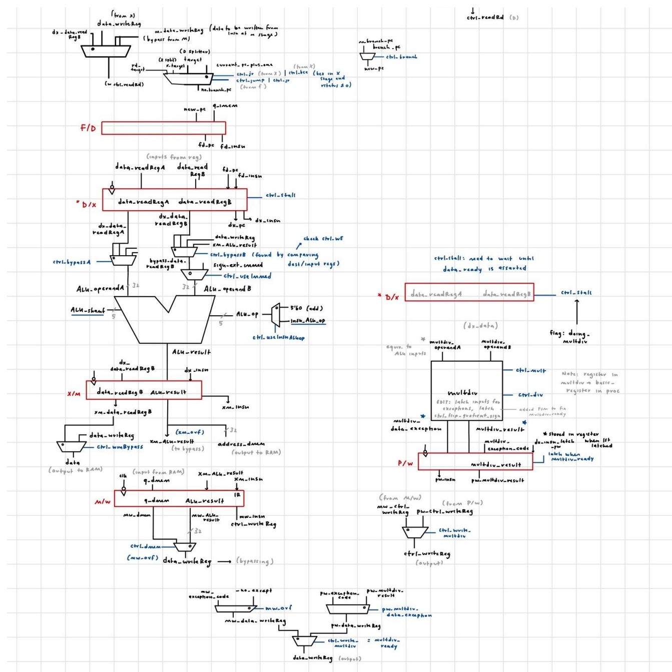
- A moisture sensor and pH sensor continually monitor soil conditions.
- The system reads the sensor data and indicates whether the moisture and pH levels are within a desired range.
- The system is driven by a custom microprocessor implemented using Verilog and SystemVerilog on an FPGA.



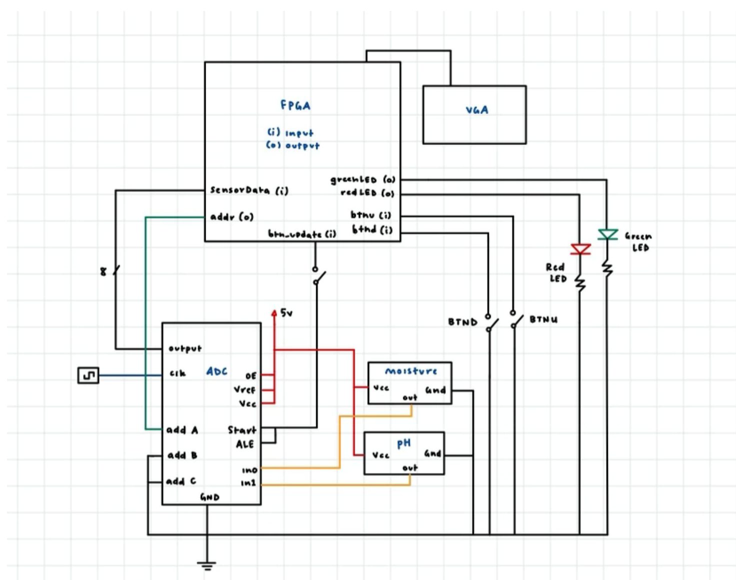
- Shown above, a VGA display continually updates with the current moisture and pH levels of the soil. LEDs indicate when the soil conditions are not ideal, prompting the user to change the moisture or pH level of the soil.

My Contribution

- I implemented a 5-stage Verilog microprocessor supporting a custom instruction set architecture that suited our system's needs.
- For instance, our ISA included specialized instructions to read data values from the moisture and pH sensors within 1 clock cycle.
- The diagram to the right shows the microprocessor implementation in detail.



- I designed and implemented an electrical circuit which collected sensor data via an A/D converter.
- The diagram to the right shows the circuit diagram for the completed system.



- I implemented FPGA I/O elements including buttons, switches, LEDs, and a

VGA display.

- These elements allowed the user to easily modify the desired ranges for moisture and pH, and easily view the soil conditions.
- Custom Verilog modules and MIPS assembly instructions were used to evaluate sensor data and continually update the output elements in real time.

Results

- The system was able to successfully monitor soil conditions including moisture and pH level. It was able to immediately indicate to users when moisture or pH deviated outside of the optimal range.
- The final project report is linked here: Final Report

Relevant skills: Verilog, SystemVerilog, CPU Design, Circuit Design, Digital Systems, Computer Architecture, FPGA, MIPS Assembly

PenguinBot: Walking Bipedal Robot

Overview

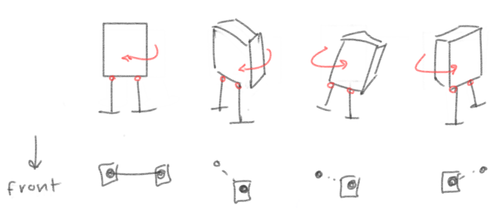
I designed and constructed a penguin-inspired walking bipedal robot.

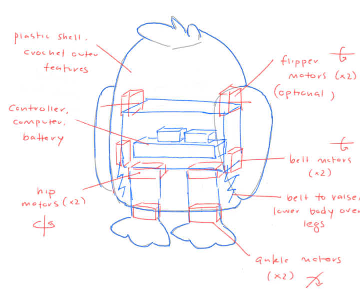
- Class project for ECE590: Robotics Studio at Duke University, taught by Dr. Boyuan Chen
- Problem statement: Create a 2-legged or 4-legged robot with the ability to walk and dance.

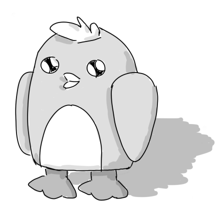
My solution: PenguinBot!

- Dimensions: 34cm x 30cm x 18cm.
- The 3D-printed robot body houses an internal electrical harness and battery.
- A Raspberry Pi drives four LX16A servo motors to enable robot movement.

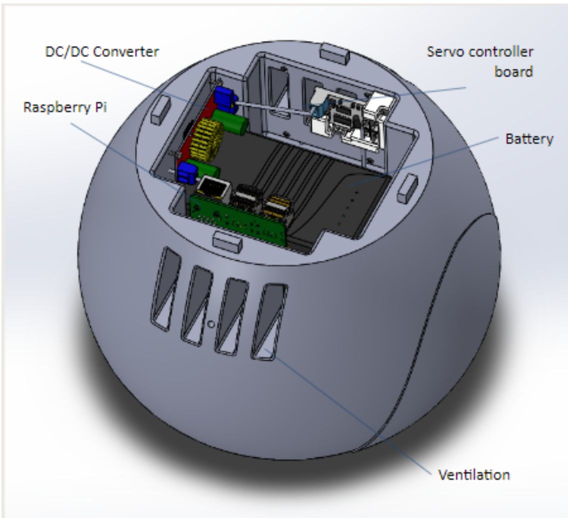
- Concept sketches are shown from the early stages of the design process.
    - Early brainstorming considered gait analysis and internal component placement.

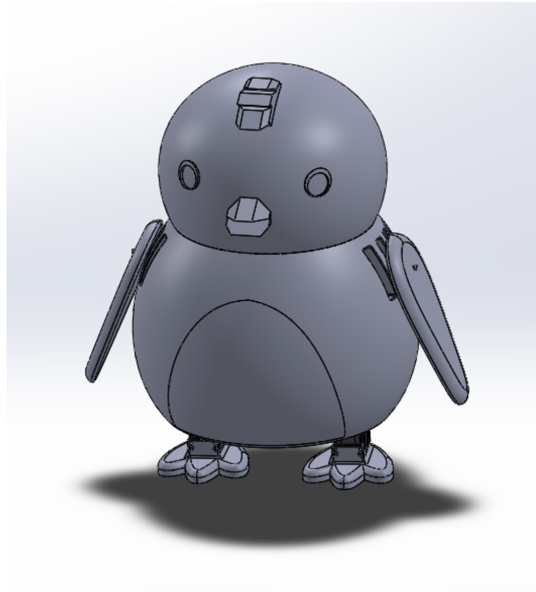




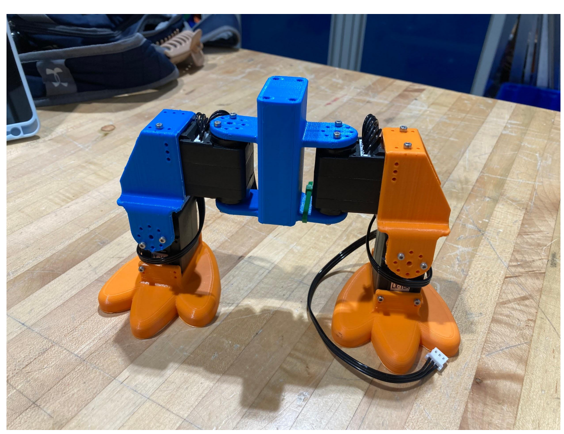


- The robot design was modeled in SolidWorks.





- I 3D-printed the robot body, created the electrical harness, and programmed the servo motors to create the robot gait.
    - The modular body design allows for easy assembly of the robot.
    - Individual parts can be quickly reprinted as needed.
- The servo motors were programmed in Python using the PyLX16A library.
    - This library interfaced easily with the motors and allowed for rapid trial-and-error when developing the robot gait.



Results

- The fully assembled robot was able to walk and dance without requiring external connections to power or control signals.
- The Github repository for the project is linked here: [Github Repo]
- To see the full design process and the robot in action, watch my full journey video linked here: [Journey Video]

- Relevant Skills: SolidWorks, 3D Printing, Python, Linux/Ubuntu, Raspberry Pi, Robotics

Combat Robotics

Overview

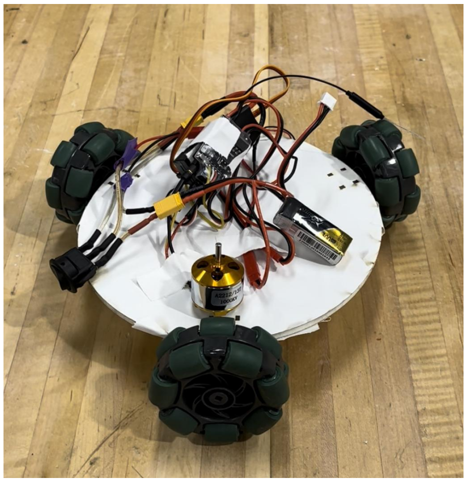
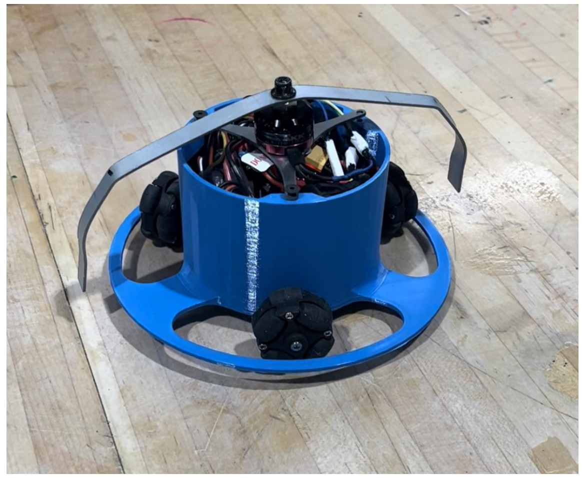
We designed, built, and deployed a combat robot with an IMU-based holonomic drive system.

- Independent study project under the direction of Dr. Tyler Bletsch at Duke University
- Member of a 3-person project team
- Problem statement: Create a combat robot in compliance with SPARC antweight robot guidelines.

Our solution: We designed an antweight combat robot with a holonomic drive system utilizing 3 omni-wheels and kiwi drive.

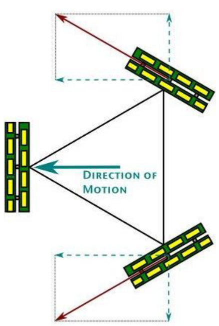
- Holonomic drive can introduce veering, which is usually regulated using encoders on each wheel.
- However, our microcontroller system uses absolute orientation data from an IMU sensor to regulate our holonomic drive system.
    - This solution is lighter and more space-efficient than the encoders, allowing the robot to meet the <1 lb weight requirement.

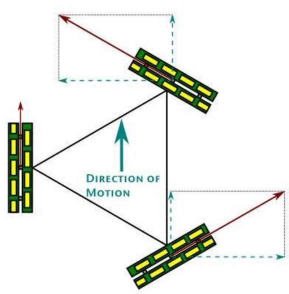
- Early prototypes of the design.

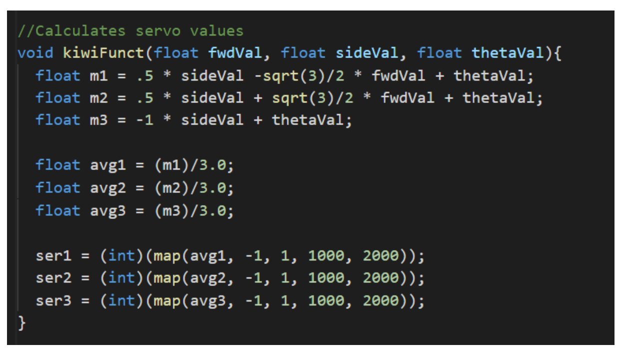



My Contribution

- I implemented the software for the kiwi drive system using the Arduino IDE.
    - We chose kiwi drive because our omnidirectional weapon was compatible with a rotationally symmetric robot body and drive system.

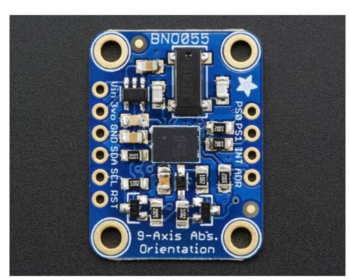




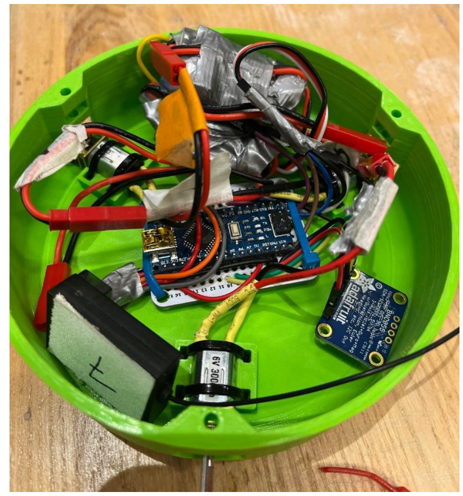


```
//Calculates servo values
void kiwiFunct(float fwdVal, float sideVal, float thetaVal){
  float m1 = .5 * sideVal -sqrt(3)/2 * fwdVal + thetaVal;
  float m2 = .5 * sideVal + sqrt(3)/2 * fwdVal + thetaVal;
  float m3 = -1 * sideVal + thetaVal;

  float avg1 = (m1)/3.0;
  float avg2 = (m2)/3.0;
  float avg3 = (m3)/3.0;

  ser1 = (int)(map(avg1, -1, 1, 1000, 2000));
  ser2 = (int)(map(avg2, -1, 1, 1000, 2000));
  ser3 = (int)(map(avg3, -1, 1, 1000, 2000));
}
```

- I implemented the microcontroller system which regulated the robot's holonomic drive system.
    - I integrated the BNO055 sensor because of its ability to accurately measure absolute orientation and angular velocity.

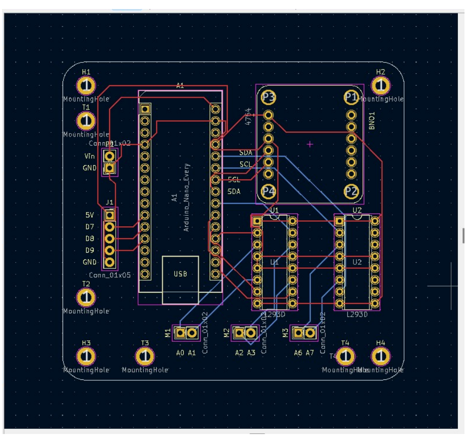


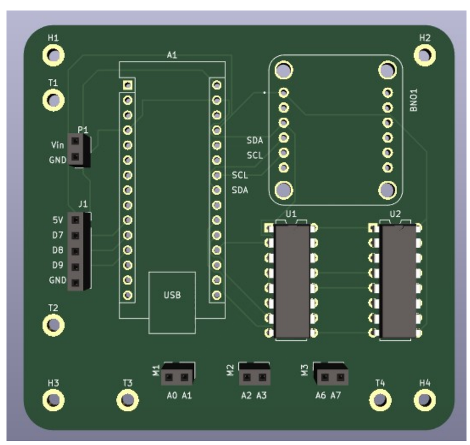
- Early prototypes used an Arduino microcontroller but the final design used an Arduino Nano to reduce the robot size and weight.

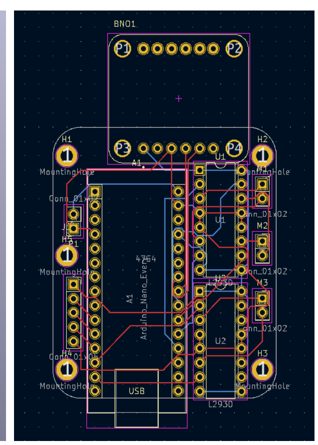


- I created a PID control loop based on the sensor data to prevent the robot from veering off course.
- The images on the right show the IMU and its implementation in one of our later prototypes.

- I designed a custom PCB for the robot using KiCad integrating the microcontroller, IMU, and motor drivers.
    - This reduced the size and weight of the electrical harness by >50%
    - The electrical connections were less likely to fail or be damaged in combat.
- The images below show an early version/rendition of the PCB layout, alongside a later iteration which was reduced in size.







Results

- The robot successfully competed in the Duke Combat Robotics competition with a functional holonomic drive system.
- The Github repositories for this project are linked here: Spring 2022 and Fall 2022

- Relevant Skills: Arduino, C++, KiCad, Microcontrollers, Sensors, SolidWorks, Robotics, Control Theory

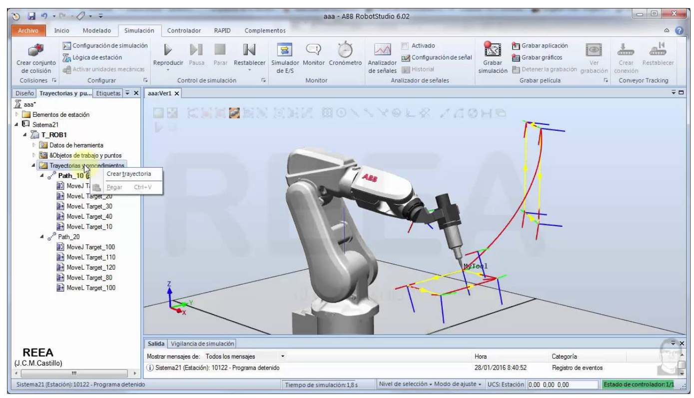
Internship at SpaceX

Overview

At SpaceX, I developed a control suite for an ABB robotic arm to improve near-field emissions scanning.

- Internship project for Summer 2023
- Worked as an engineering intern on the Electromagnetic Environmental Effects team under the supervision of Scott Shermer

- The image on the right shows a screenshot of the RobotStudio software used in this project.
- Specific details about project implementation and results are proprietary.



- Relevant Skills: Python, ABB RobotStudio, Automation, Scripting, Robotics, Signal Generator, Spectrum Analyzer, RF Testing and Measurement

Skin Disease Classification App

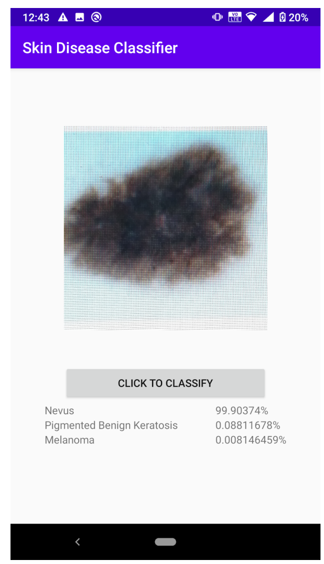
Overview

We implemented a mobile application to identify common skin diseases using image classification and artificial neural networks.

- Personal project
- Member of a 2-person project team
- Problem statement: Skin cancer is often misdiagnosed in its early stages when it is most curable.

Our Solution: We created a mobile application which identifies a skin disease based on a user-provided image.

- Our neural network model successfully classifies common skin diseases with >90% accuracy.



- Relevant Skills: Machine Learning, Python, Keras, TensorFlow, Android Studio

Internship at UnitedHealth Group - Optum

Overview

At Optum, I developed a web dashboard to display real-time health insurance data for use by 500 clients and 450K members.

- Internship project for Summer 2022
- Member of 4-person intern team
- Worked as a software intern on the UHC Student Resources team under the supervision of Archna Kalra

---

- Specific details about project implementation and results are proprietary.

- Relevant Skills: C#, SQL, AngularJS, HTML, CSS