Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Team 2: Robotic Sorting System
Bi-Weekly Update 5

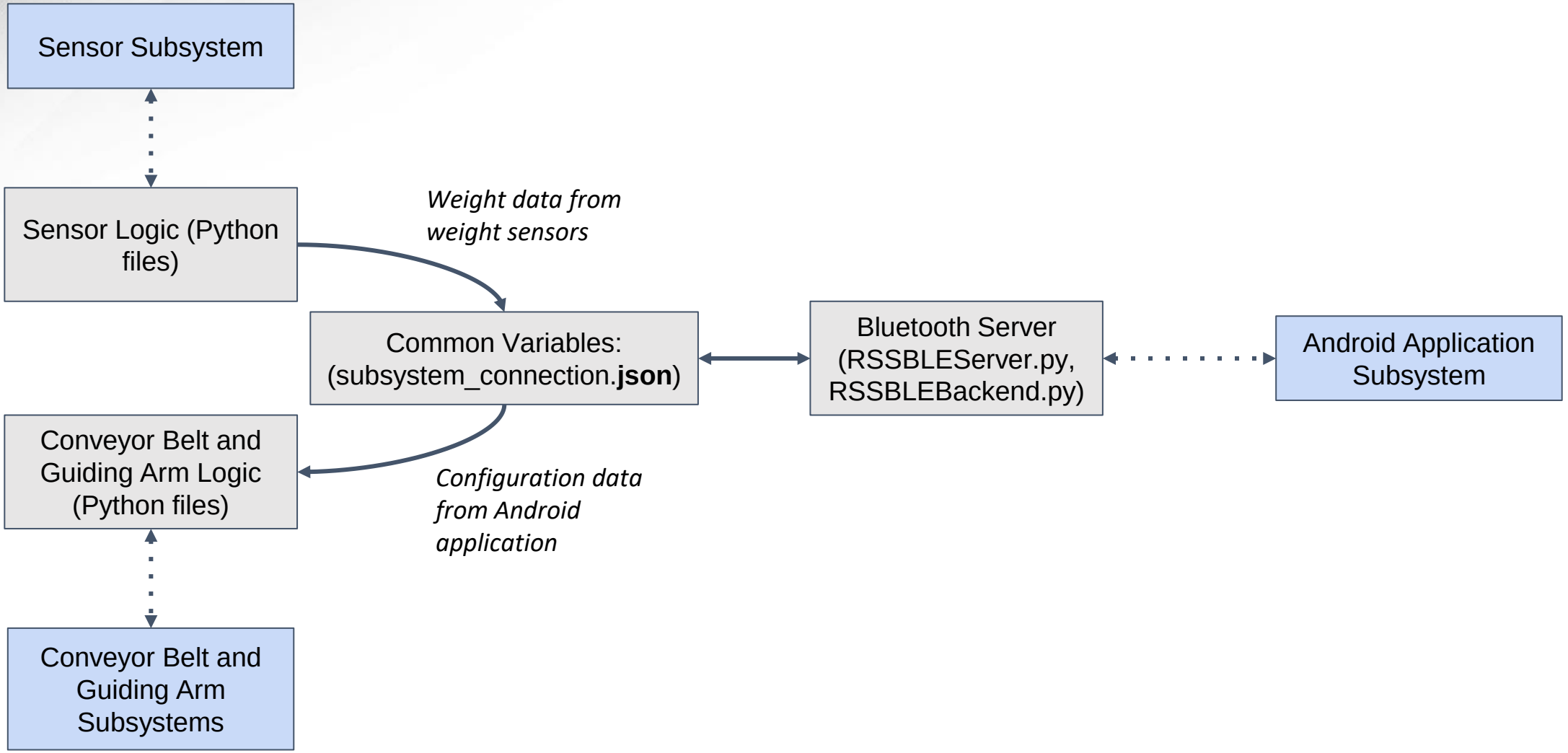
Pace Dominy, Joseph Miller, Lam Tran
Sponsor: Dalton Cyr
TA: Swarnabha Roy

Project Description

Sorting fruit by hand can be tedious and expensive.

The Robotic Sorting System is an automatic sorting system that requires limited manpower/oversight that can accurately sort fruit by their size and quality without the user having to do it themselves.

Integrated System Diagram



Project Timeline

Finished Android Application subsystem (completed 2/5)	Finished 403 Sensor subsystem (completed 2/19)	Power Control System (to complete by 3/12)	Final Integration of Android App connection (to complete by 3/12)	Finish basic Robotic Sorting System functionality (to complete by 4/2)	All subsystems integrated (to complete by 4/9)	Final Project Demonstration (to complete by 4/24)

Software Communication

Joseph Miller - 12 hours of effort

Key accomplishments:

- Moved shared project information (originally “subsystem connection”) to a JSON database
- All components of the system interact on one Raspberry Pi
- Verification of functionality based on database reads
  - e.g. pressing “Start Belt” on the app actually starts the belt

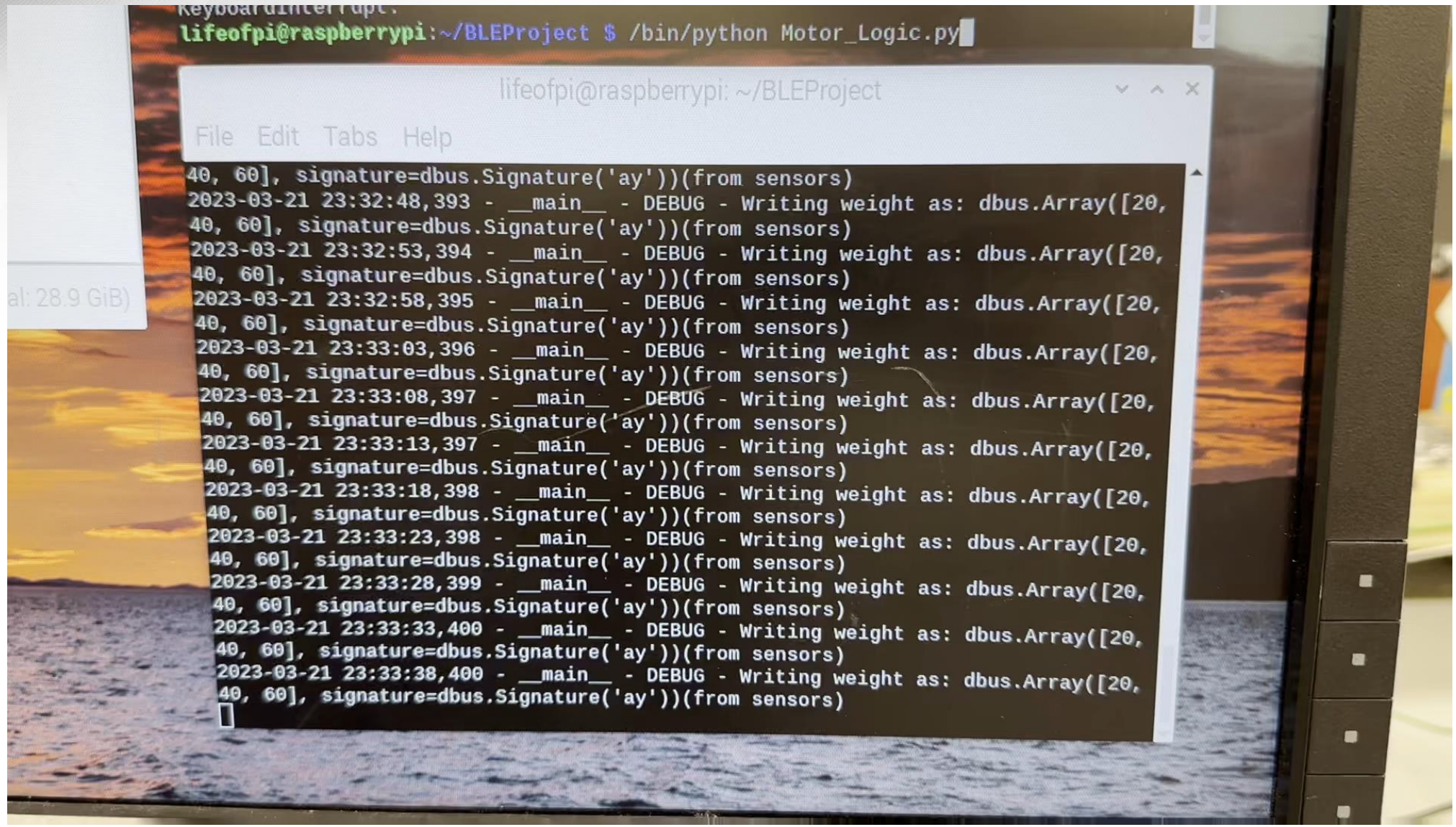
Upcoming plans:

- Optimize timing of database reads/writes based on tests and operating constraints of the system’s different components
- Validate system error conditions

Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Software Communication

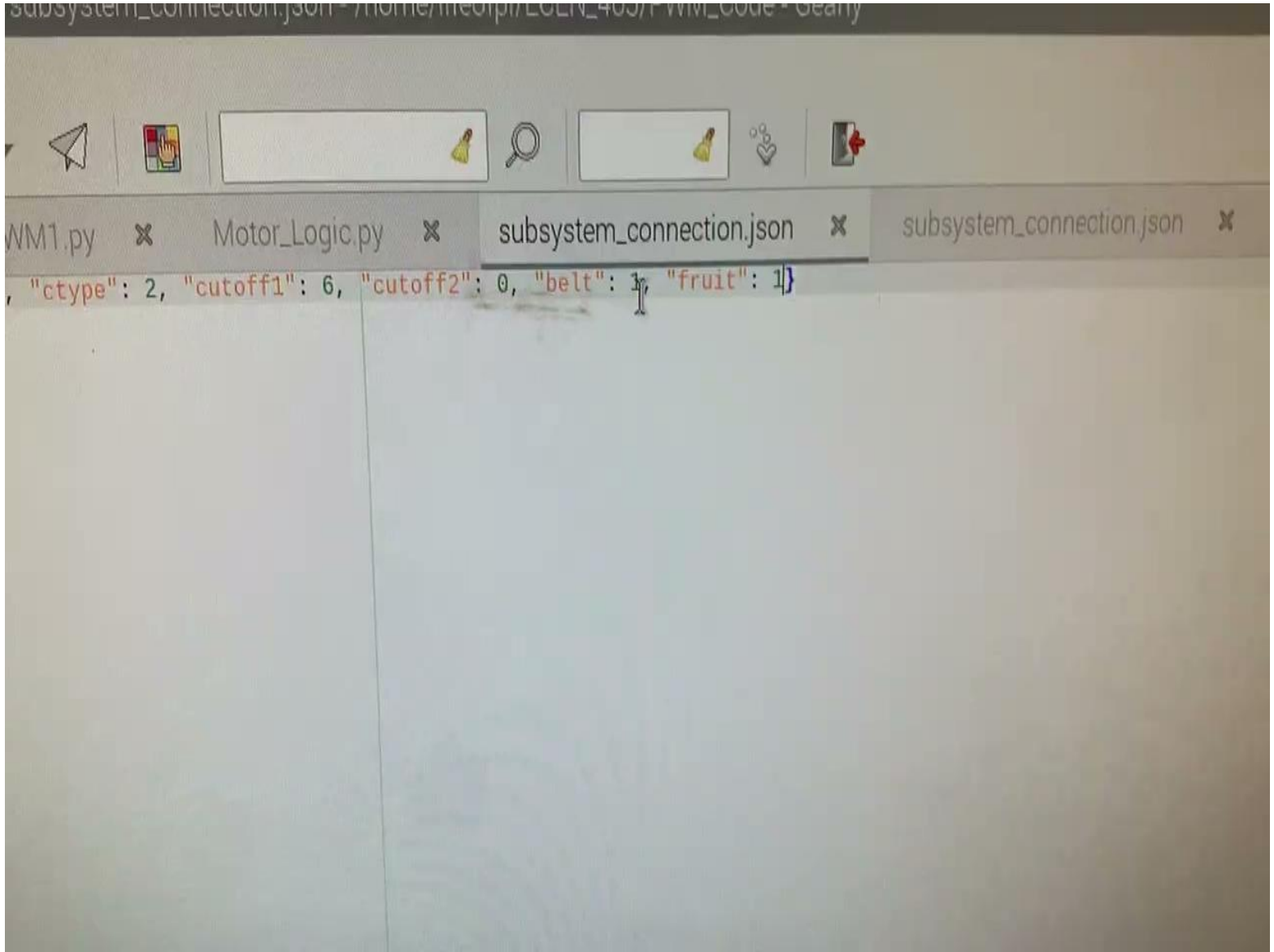
Joseph Miller - 12 hours of effort



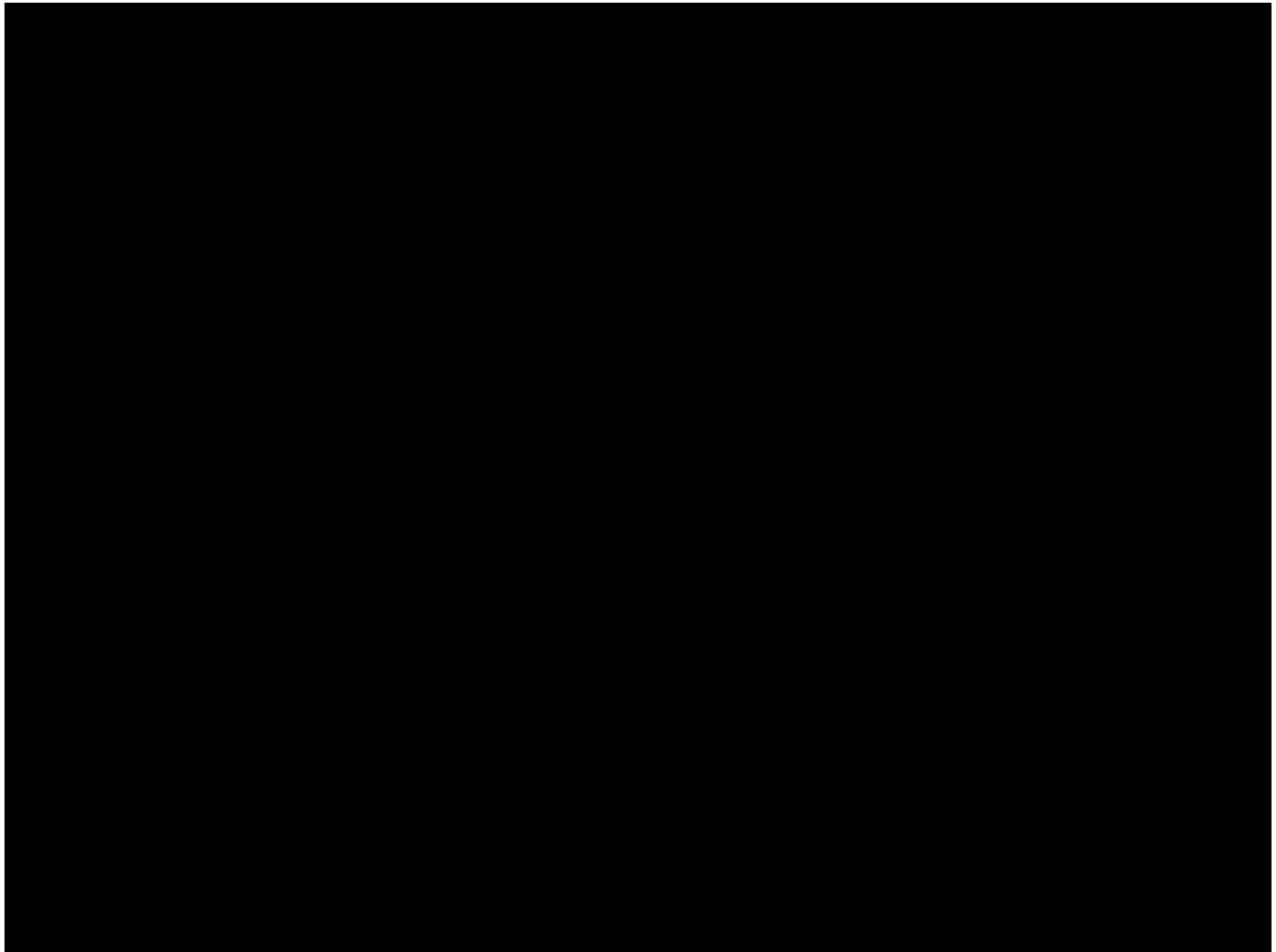
Pace Dominy

Accomplishments since Status Update 4 (28.5 Hours of Effort)	Ongoing progress/problems and plans until the next presentation
• Validated PCB, pi and load cells communicate properly • Validated guiding rails move position before conveyor belt moves • Finished python code that interacts with system database. Validated that conveyor and guiding rails take input from database and move accordingly.	• System conveyor belt must be replaced with new belt that has enough tension to hold lemons and limes (already ordered last week) • Guiding rail pulley needs new 3D printed part with larger groove in order to prevent pulley from slipping out of groove (already submitted to FEDC) • On going: Guiding rail timing must be calibrated for changes in tension of pulley as its rails near left and right lateral limits

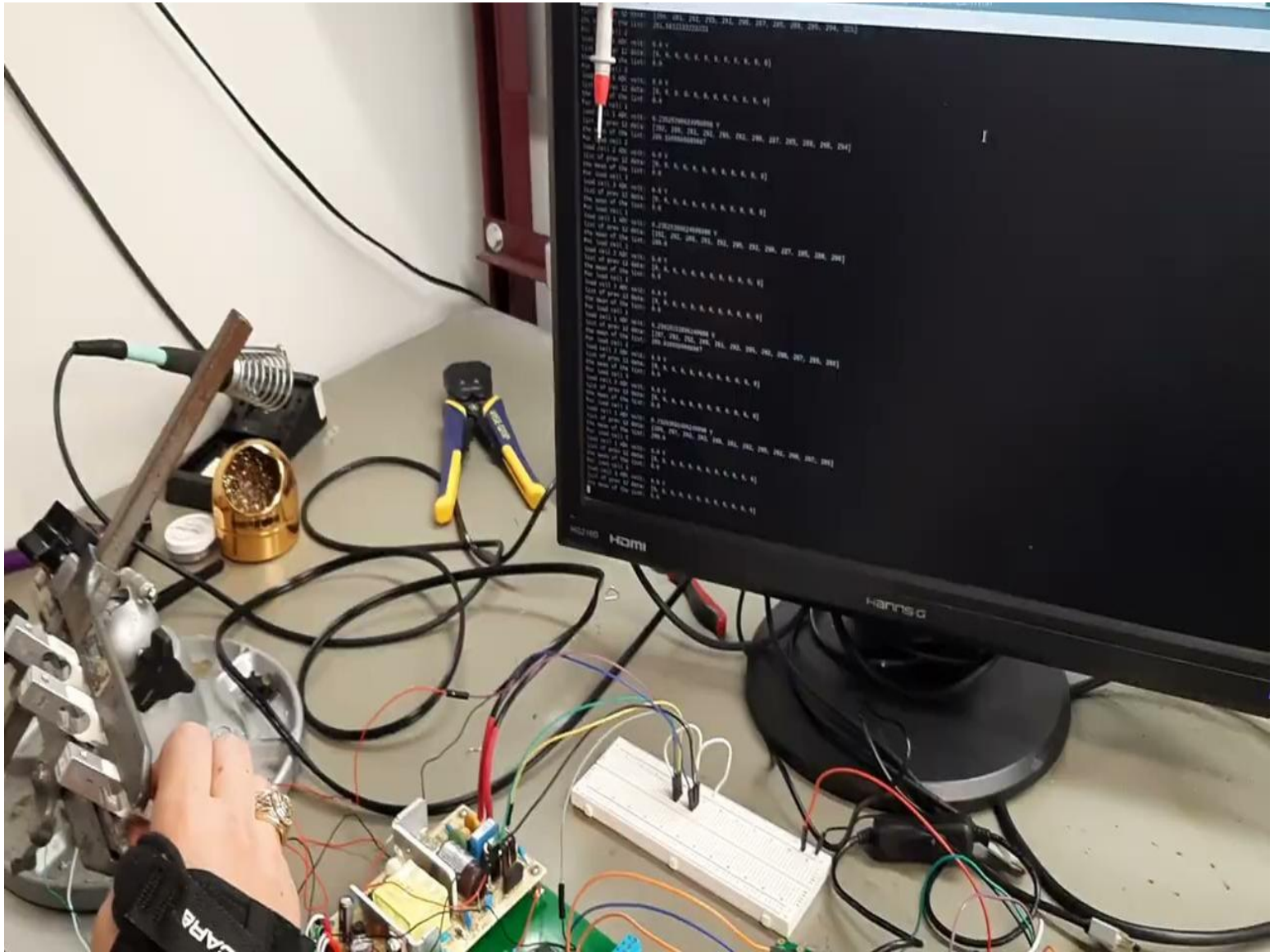
Database Demonstration



Original Guiding Rail Design



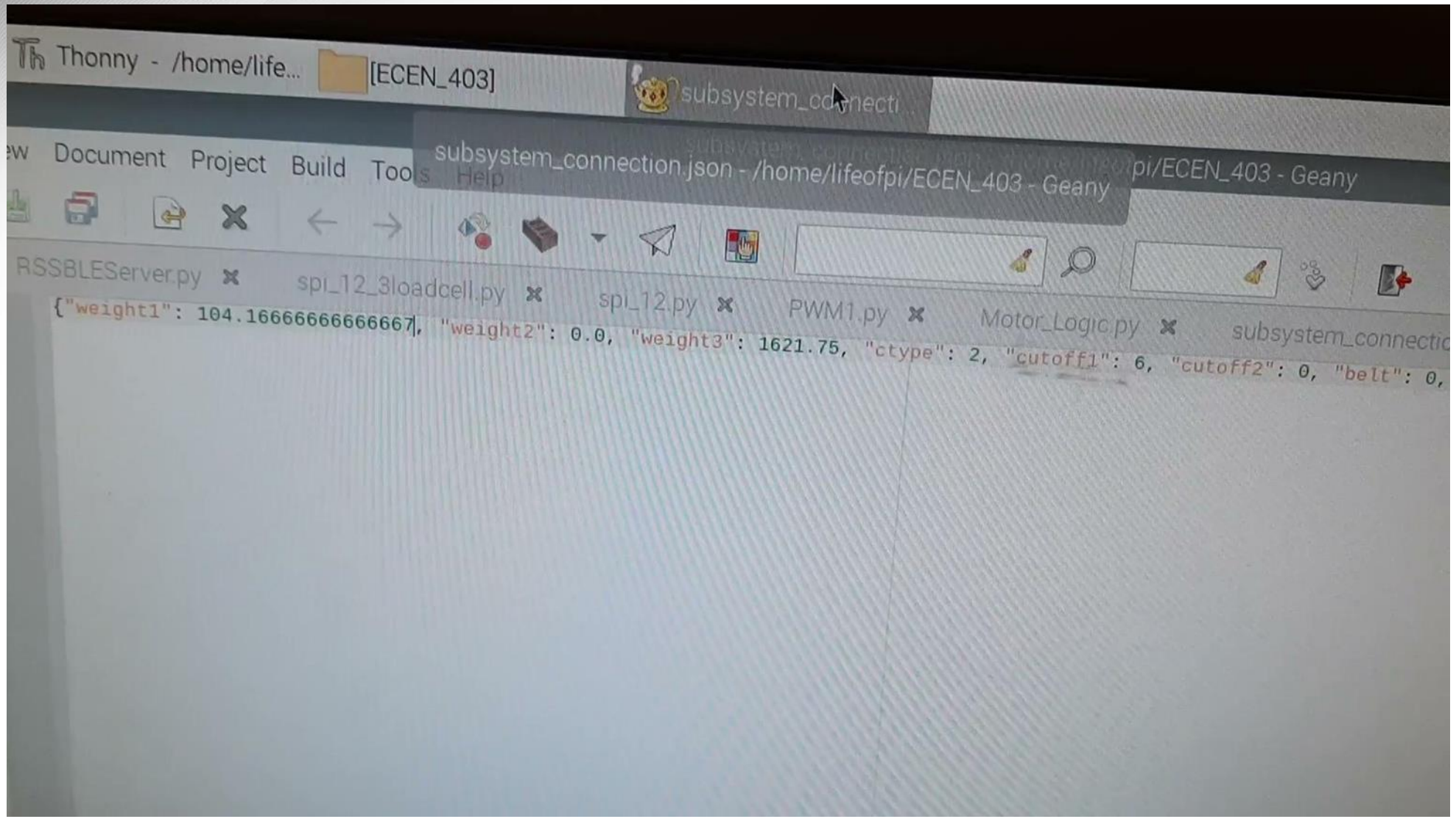
System Load Cell Validation



Lam Tran

Accomplishments since Status Update 4 (47 hrs of effort)	Ongoing progress/problems and plans until the next presentation
• Validated weight sensor connected to the PCB • Reprogrammed the weight sensor to update the JSON file and validated that it updates the file • Reprogrammed the size/color sensor code to see object from the conveyor belt • Created and tested machine learning model to detect the defects for lemons. The model failed to detect the defects	• Finish integration and validation with App subsystem (e.g. sensor error) • Finish integration and validation with the conveyor belt subsystem (e.g. output Kg measurement, size/color sensor on the conveyor belt) • After finish integrating, created a new model using TA's advice.

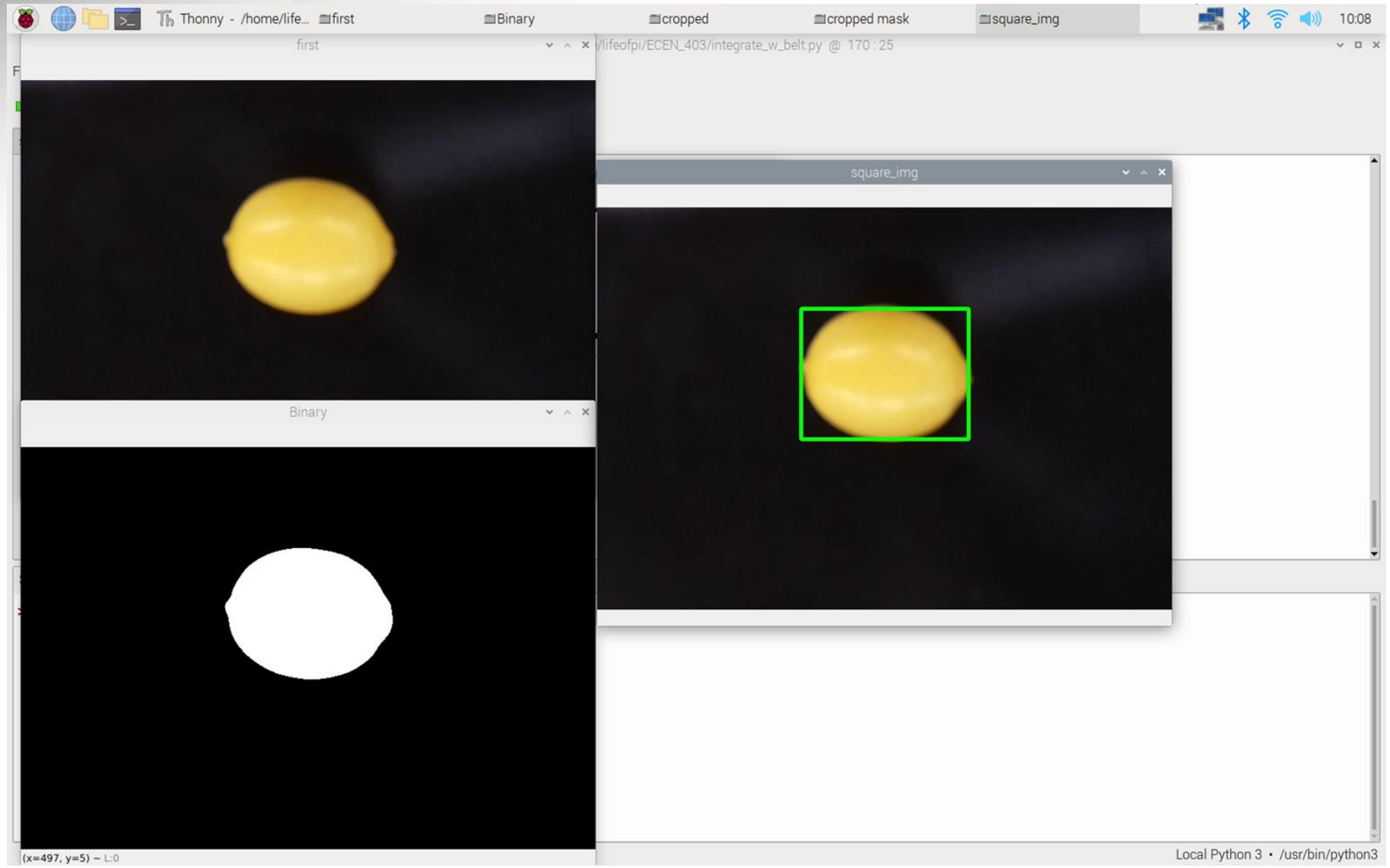
Weight Sensor and JSON database



https://youtu.be/BF9kSXDFyPc

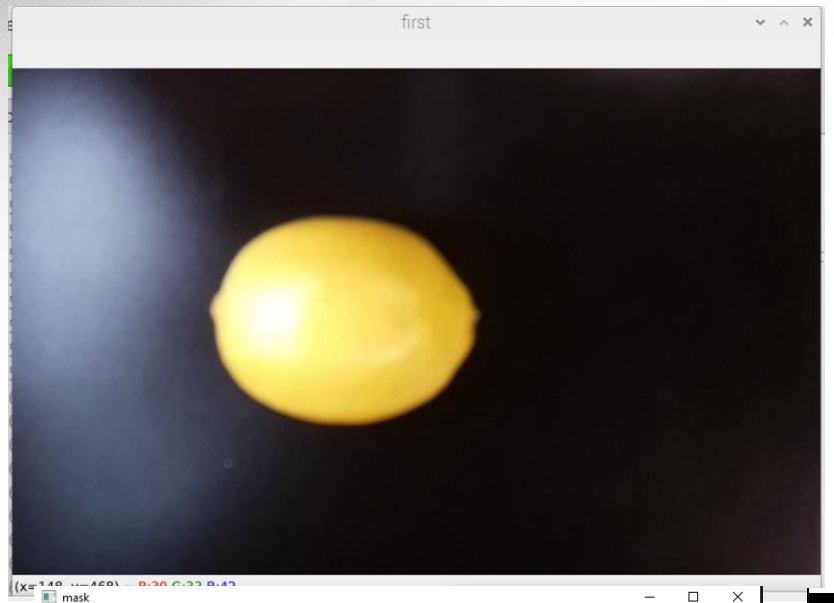
Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

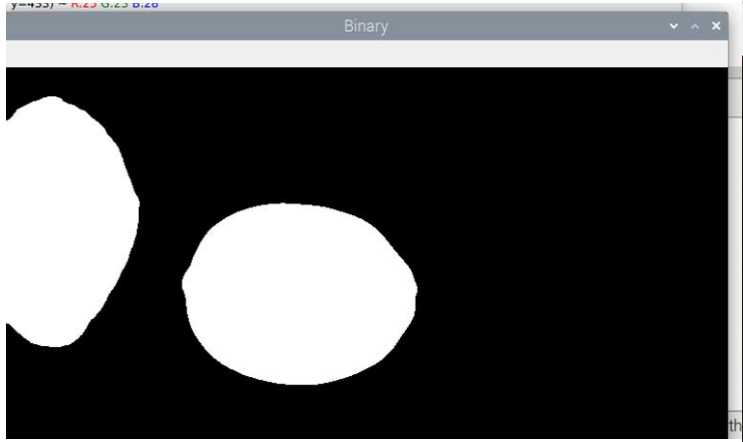
Size/Color Sensor and Conveyor belt

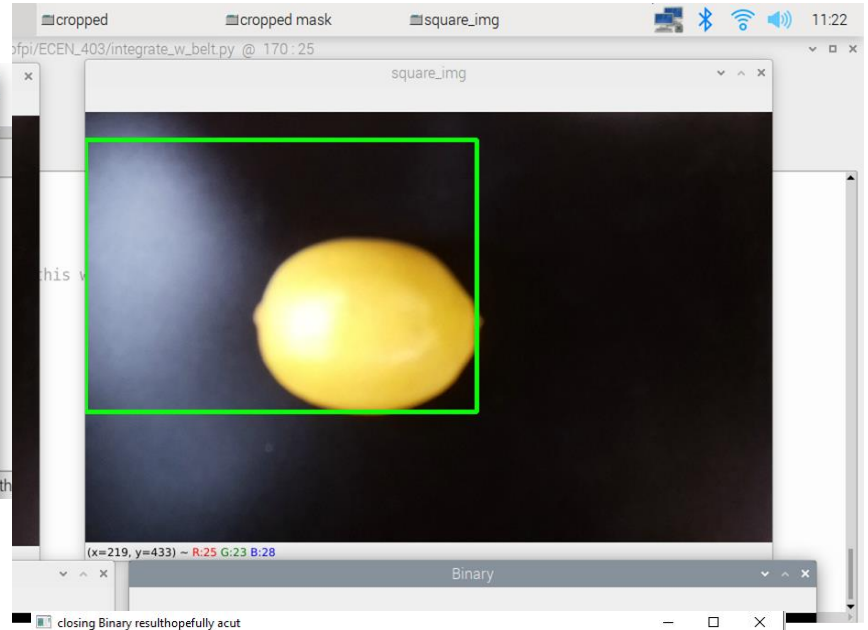


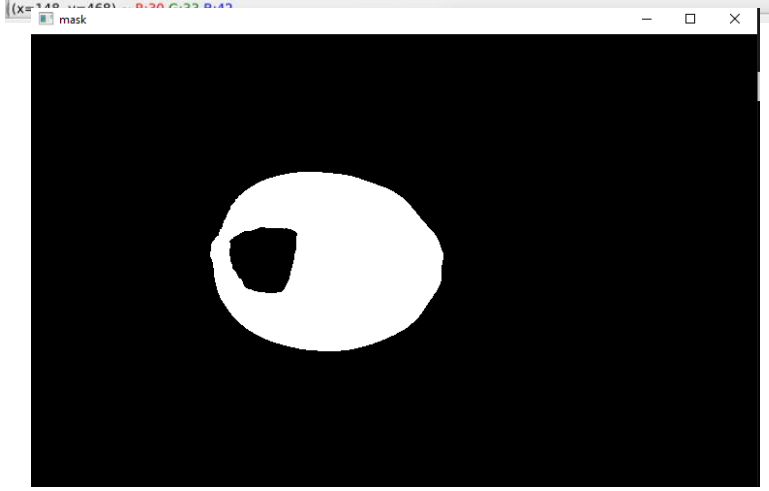
Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

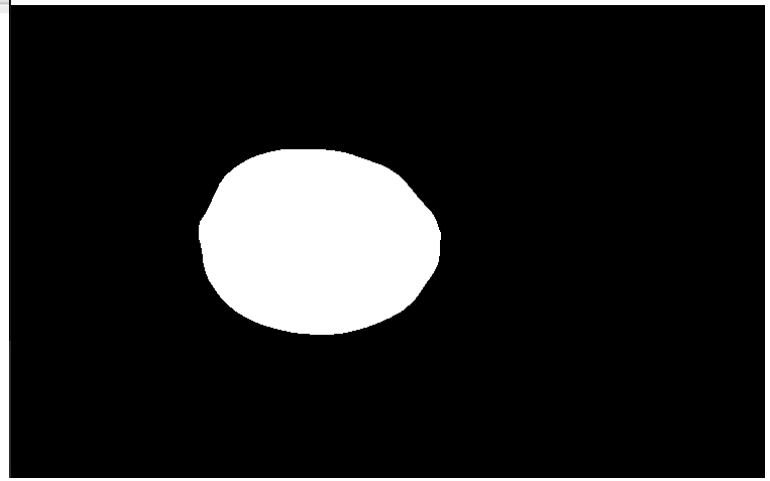
Size/Color Sensor and Conveyor belt Cont

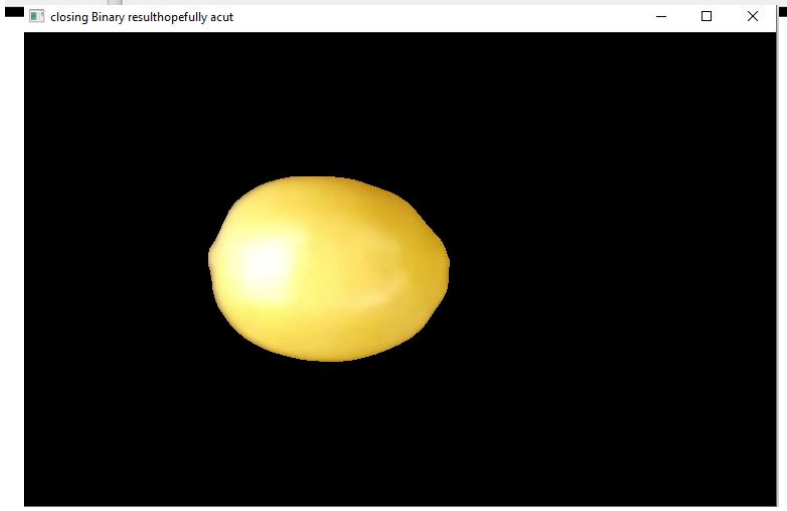




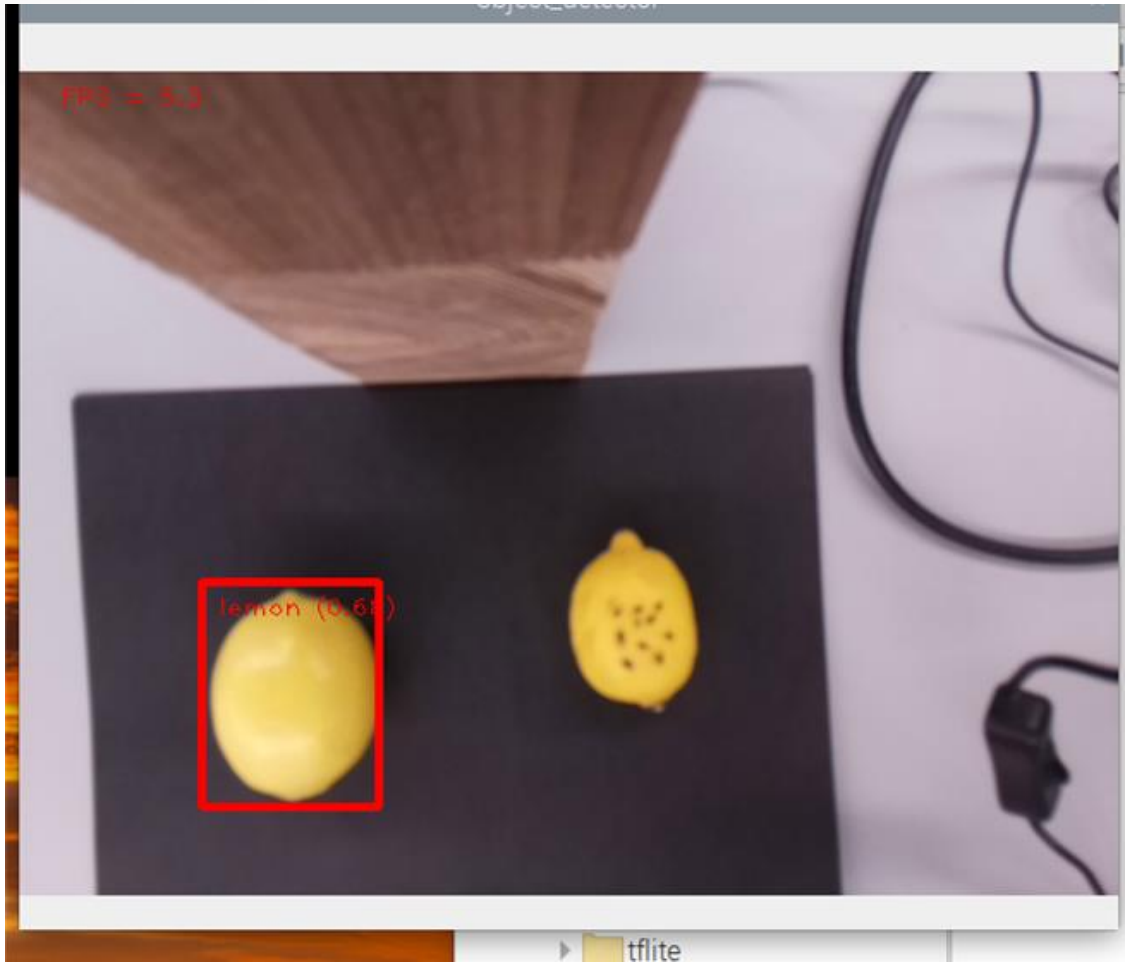




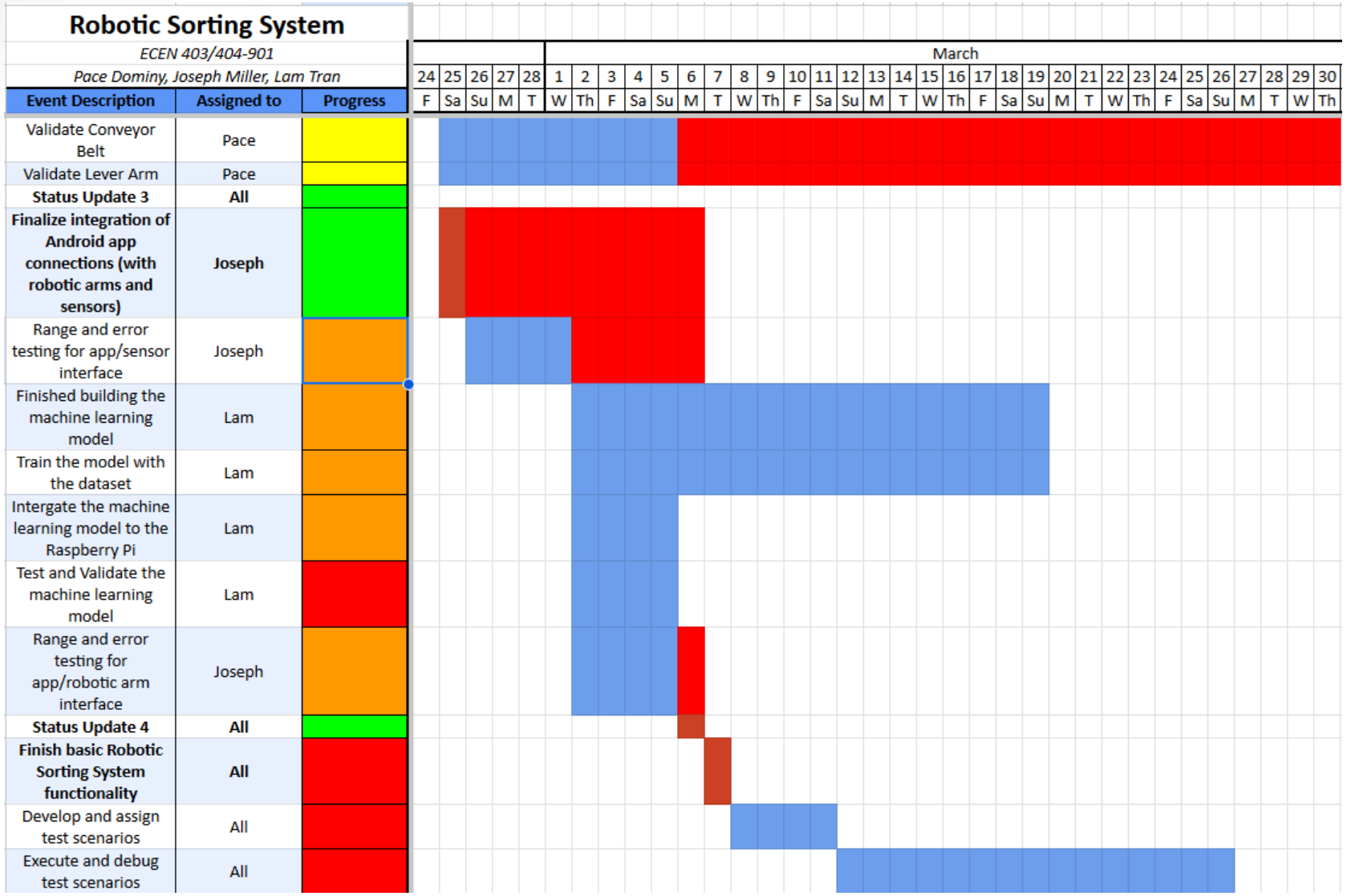




Machine Learning Model Status



Execution Plan: Recent Progress



Robotic Sorting System

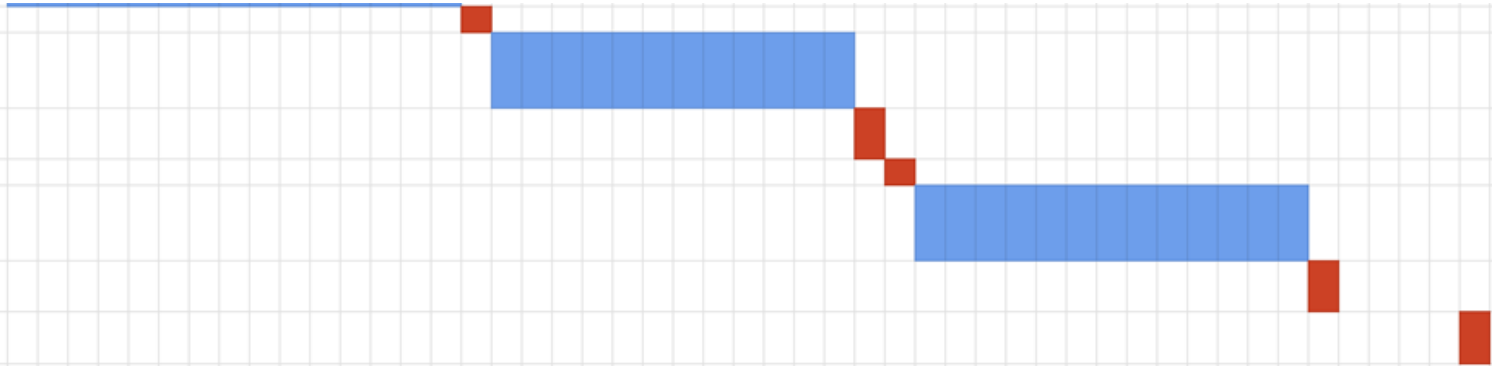
ECEN 403/404-901

Pace Dominy, Joseph Miller, Lam Tran

Event Description	Assigned to	Progress
Validate Conveyor Belt	Pace	
Validate Lever Arm	Pace	
Status Update 3	**All**	
Finalize integration of Android app connections (with robotic arms and sensors)	**Joseph**	
Range and error testing for app/sensor interface	Joseph	
Finished building the machine learning model	Lam	
Train the model with the dataset	Lam	
Intergate the machine learning model to the Raspberry Pi	Lam	
Test and Validate the machine learning model	Lam	
Range and error testing for app/robotic arm interface	Joseph	
Status Update 4	**All**	
Finish basic Robotic Sorting System functionality	**All**	
Develop and assign test scenarios	All	
Execute and debug test scenarios	All	

Execution Plan: Upcoming Plans

92	**Status Update 5**	**All**	
93	Finalize testing and functionality (buffer time)	All	
94	**All subsystems integrated**	**All**	
95	**Final Presentation**	All	
96	Finalize testing and functionality (buffer time)	All	
97	**Final Project Demonstration**	**All**	
98	**Final Report (end of ECEN 404)**	**All**	



Validation Plan

ECEN 404 (Integration/Project)					
Subsystem	*Test Name*	*FSR Reference*	*Success Criteria*	*Status*	*Responsibility*
App	Basic write test	Subsystem 3.2.4.1.1 Project 3.2.3.1	A configuration value chosen by the user using the Android application will be written to the Bluetooth server. That value can then be read by the Android app.	Complete: wrote 6, then read 6 in the Diagnostics screen	Joseph
App	Basic read test	Subsystem 3.2.4.1.2 Project 3.2.3.4	The application will read the configuration and weight values from the Bluetooth server and display those values on the Diagnostics and Machine Information screen.	Complete: read 56 for weight and 12 for configuration	Joseph
App	RSS RPi integration test	Subsystem 3.2.4.1.1, 3.2.4.1.2 Project 3.2.3.1, 3.2.3.4	The results from the Basic Write Test and Basic Read Test will be successfully replicated on the RSS Raspberry Pi.	Complete	Joseph
App	Faulty configuration protection	Subsystem 3.2.3.1, 3.2.5 Project 3.2.5	The faulty configuration values 30, 0, and -1 will not be written to the Bluetooth server and will result in a popup error.	Complete	Joseph
Sensor	Intergating the sensors with the RPI and the RasPiCam	Subsystem 3.2.2.3, 3.2.2.4	The Raspberry Pi is able to use the RasPiCam to take a image, and use it to determine the size or color of the fruit.	Complete	Lam
Sensor	Fruit's scarring/disfigurement detection	ConOp 4.3 Subsystem 3.2.2.4.3	The Raspberry Pi is able to tell if the fruit have some scarrings or disfigurements. This will determines whether or not the fruit is fit to be sold to the market.	Partially complete: Right now the model is able to detect lemon. The defects of the lemon need to be label and a model have to be trained with these labels.	Lam
Lever Arm	Position Validation	Subsystem 3.2.1	Lever arm moves to correct position based off of the data from the sensors	Partially complete: inital design worked perfectly. second design is flawed and the 3D printed part needs a larger groove to hold the pulley. Pulley gets more taunt the further it goes to its right or left lateral limits. Timing must be tuned to account for this.	Pace
App/Sensors	App/Sensor Integration	App Subsystem 3.2.1.4.2 Project 3.2.3.4	The weight value passed from the sensor to the Raspberry Pi is shown on the Diagnostic screen of the Android app.	Partially complete: Bluetooth server weight characteristic is exposed to sensor subsystem and will update when the sensor changes it. Only needs to be tested.	Joseph/Lam
App/Lever Arm	App/Lever Arm Integration	App Subsystem 3.2.4.1.1 Project 3.2.3.1 Robotic Arm Subsystem 3.2.1.2	The robotic arm moves to Bin 1 based on the configuration passed in via the Sort by Size screen.	Partially complete: configuration information from Bluetooth server is exposed to robotic arm system.	Joseph/Pace
Lever Arm & Conveyor Belt	Timing Validation	Subsystem 3.2.1	Lever arm moves to correct position and stays there. Conveyor belt then turns on and lever arm does not move until it is required to do so again	Partially complete: must be fine tuned for prototype version 2	Pace
Lever Arm/Sensors	Lever Arm/Sensors Intergration	Project 3.2.3.4.3, Sensor Subsystem 3.2.2.4.1	Lever arm moves a certain the position based off of the sensor output.	Partially complete: Belt takes input from database, that the sensors can write to, in order to determine whether or not it needs to move and then moves to the correct bin based off of its current orientation and what bin it needs to go to.	Pace/Lam
Converyor Belt/Sensors	Converyor Belt/Sensors Integration	Project 3.2.2.2, Sensor Subsystem 3.2.2.1	The camera is postion to take clear and consistent images. The weight sensor will be calibrated when it is attact to converory belt system and holding the bins.	Incomplete	Pace/Lam
App/Conveyor Belt	Belt Start/Stop	Project 3.2.3.1	The app passes a command to stop and start the belt.	Complete	Joseph
App/Sensors	Sensor Error Testing	Project 3.2.5.1.3	The sensor indicates a fault, causing an error message on the Diagnostic screen.	App subsystem-complete (app recognizes -1 from sensors as error)	Joseph/Lam
App/Conveyor Belt	Belt Movement Error	Project 3.2.5.1.1, 3.2.5.1.2	The app returns an error if the belt is obstructed or the power to the machine is cut off (stopping the belt).	App subsystem-complete (app recognizes -1 from belt as error)	Joseph
App/System	Full Functionality	App Subsystem: All Project: 3.2.1.4, 3.2.3.1, 3.2.3.4.1, 3.2.3.4.2, 3.2.5	The app passes a configuration to the machine and starts the belt. When sorting is complete, the app stops the belt and returns a notification.	Partially complete (most of this functionality is there)	Joseph

Thank you!