



# STANDARD SMALL SATELLITE RESEARCH PLATFORM FOR LIFE SCIENCE RESEARCH

# ORGANIZATIONAL CHART



**DR. SAMUEL RICHIE**

ECE Technical Advisor



**DR. MARK HEINRICH**

CS Technical Advisor



**DR. JOSE NUNEZ**

NASA KSC Chief, Flight ECE Technical Advisor  
Technologies Branch



**DR. LEI WEI**

ECE Technical Advisor



**NICOLAS EL TENN**

Computer Engineer



**RAQUEL GUZMAN**

Computer Engineer



**NOAH HEIKES**

Computer Science



**SHIVANI KUMAR**

Computer Science



**MATTHEW PHILPOTT**

Electrical Engineer

# PROJECT SOLICITATION

- Florida Space Grant Consortium
- NASA Kennedy Space Center
- Development of a standard Small Satellite (e.g. CubeSat range) platform for Life Sciences research
- Usage of interchangeable research modules usable on Earth, in Low Earth Orbit, lunar orbit, aboard a lunar outpost (e.g. Gateway), and on the Lunar surface

# PROJECT TECHNICAL GAPS

Goal	Category	Gap	Sub-gap	Description
Main	Hardware	Environmental Monitoring	Root Zone Sensors	Lack of effective Root Zone Moisture Sensor
Stretch	Crops	Crop Performance	Yield Database	Incomplete data sets on yield of candidate crops in space environments

# GOALS AND OBJECTIVES

## **Mission:**

Our mission is to design an environmental monitoring system for plants in microgravity (Lunar Orbit) and tackle NASA Technical Gaps.

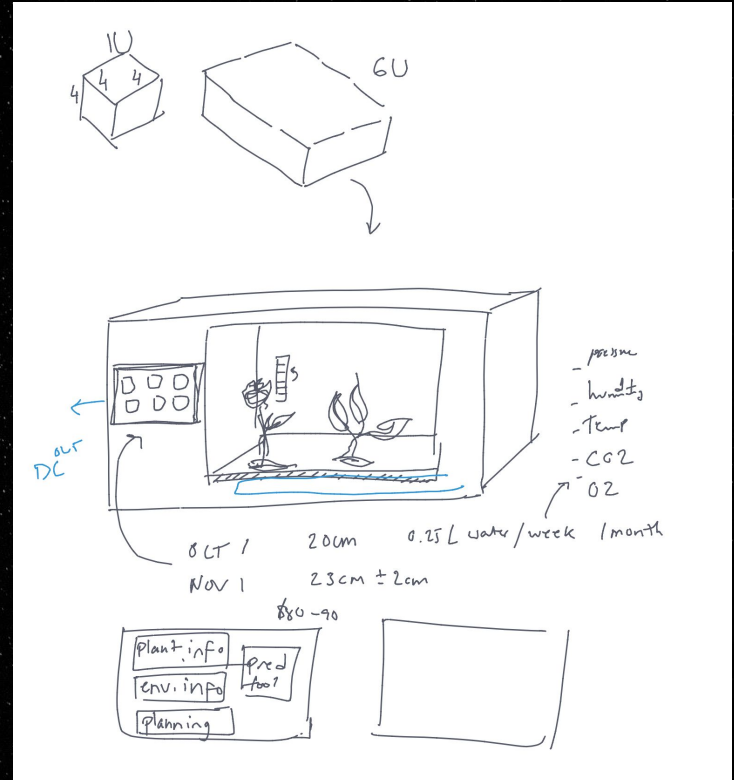
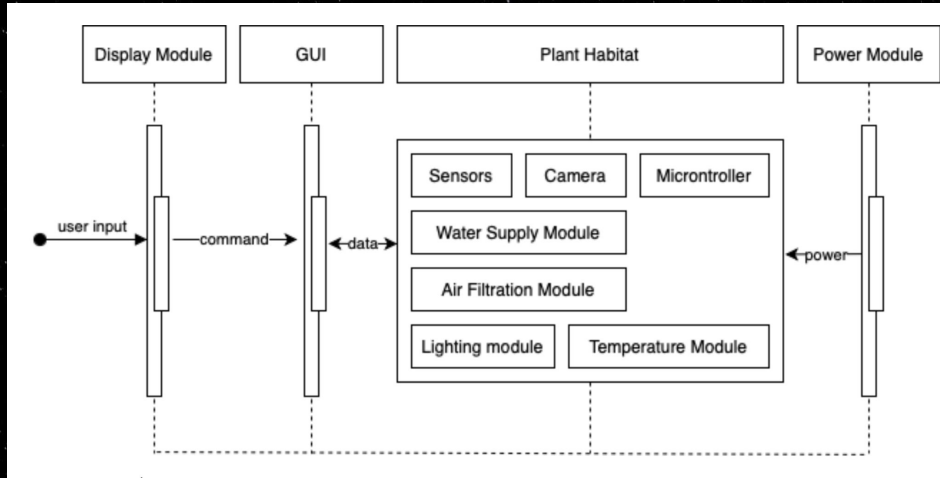
## **Objectives:**

- The data will successfully be gathered and transferred to the database from the humidity sensor, temperature sensor, light sensor, and camera units.
- The web app will display the sensor information and images of the plant outlining ideal conditions for plant growth.

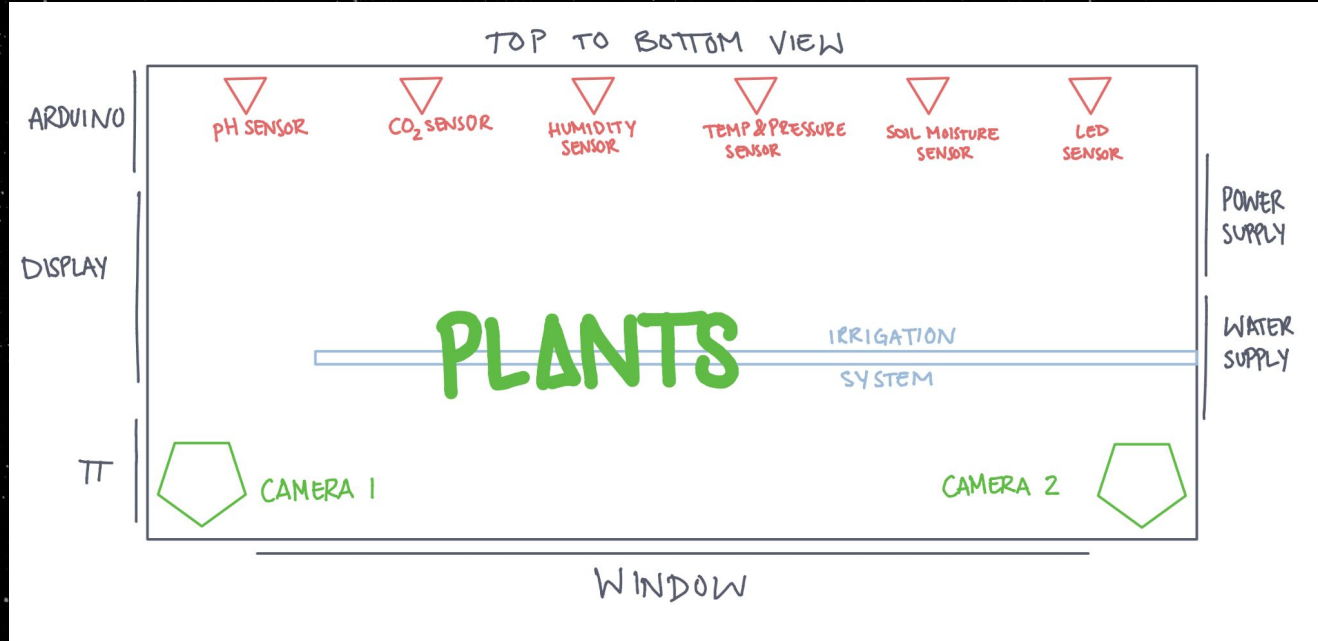
# INTERVIEWS CONDUCTED

Name	Background	Discussion
Dr. Allen	UCF MAE Department	CubeSat structure
Dr. Colwell	UCF Physics Department	CubeSat feasibility
Karen Debaere	UCF Biology Department	Tissue culture
Stephen Dick	UCF System Administrator	Communication and antennas
Dr. Mason	Plant Physiology	Plant research
Dr. Massa	NASA KSC Horticulture	Plants in space
Dr. Nunez	NASA KSC Flight Technologies	Systems Engineering
Julia Walton	Biology Laboratory	Sterilization and labs

# SYSTEM ARCHITECTURE

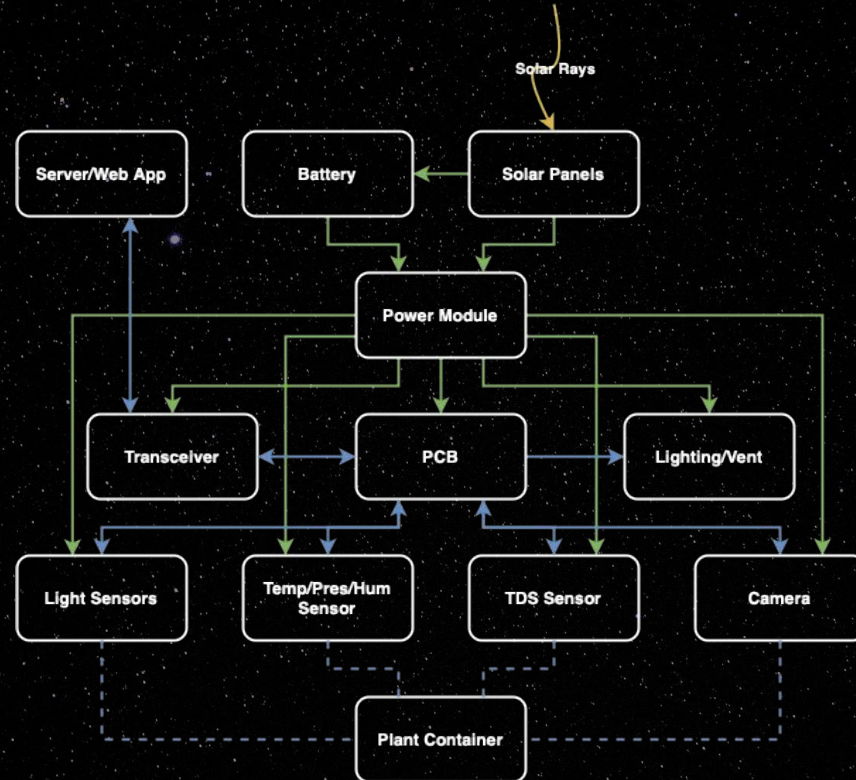


# SYSTEM ARCHITECTURE





# SYSTEM ARCHITECTURE



# ARABIDOPSIS THALIANA SPECIFICATIONS AND REQUIREMENTS

Aspect	Description
Grown Size	~ 2-3 centimeters in diameter
Life Cycle	Short life cycle about 6-8 weeks from germination to seed maturation
Grown Mass	~ 2.5 grams
Watering frequency	Sealed container retains moisture sufficient for operational period
Photoperiod	16:8



# NICOTIANA TABACUM SPECIFICATIONS AND REQUIREMENTS

Aspect	Description
Photoperiod	18:6
Germination	7-10 days
Temperature	20-25 degrees C
Watering frequency	Sealed container retains moisture sufficient for operational period
Growth medium	Murashige and Skoog
Weeks since planted	7 weeks by recording 8 weeks by final presentation



# HARDWARE SPECIFICATIONS AND REQUIREMENTS

- Dimensions: 6U (20cm x 10cm x 34.05cm)
- Camera: 720p-1080p, Images taken to be processed by server
- Temperature:  $\pm 1.0^{\circ}\text{C}$  accuracy
- Humidity:  $\pm 3\%$  accuracy
- Pressure:  $\pm 1$  meter or better accuracy
- Light Sensor: IR 550nm-1000nm (centered on 800), Visible 400nm-800nm (centered on 530)
- Spectral Color Sensor: able to detect not only the amount of light present, but also the amounts of light within 400nm - Near IR wavelengths

# PAYLOAD SPECIFICATIONS

- Review on small cube sizes:
  - 1U CubeSat is 10 cm × 10 cm × 11.35 cm.
  - **6U CubeSat is 20 cm × 10 cm × 34.05 cm.**
  - 12U CubeSat is 20 cm × 20 cm × 34.05 cm.
  - Smallest existing CubeSat design is 0.25U and largest is 27U.
- Payload Dimensions: 6U (9.7 x 19.7 x 22.3cm)
- Research shows that:
  - Payload mass- 7.2-7.8 kg (configuration dependent)
  - Available Payload volume 97 x 197 x 223 mm<sup>3</sup>
- Box is laser cut 0.25 plywood(10 x 15 x 20 cm)

# CAMERA SELECTION

	ArduCAM Mini (2MP)	ESP32-CAM (2MP)	IMX219 (8MP)
Image Resolution	1600 x 1200	1600 x 1200	3280 x 2464
Interface	I2C/SPI	I2C/SPI	I2C
Voltage	3.3V - 5.0 V	3.3V - 5.0V	3.3V
Price	\$25.99	2 for \$17.99	\$27.99

# ENVIRONMENTAL SENSOR SELECTION

	BME280	MPL115A2	DS18B20
Interface	I2C/SPI	I2C	1-Wire protocol
Voltage	1.8V - 5.0V	3.0V - 5.0V	3V - 5.5V
Accuracy (temp., humidity, pressure)	$\pm 1^{\circ}\text{C}$ , $\pm 3\%$ , $\pm 1$ meter	$\pm 1^{\circ}\text{C}$ , no humidity, $\pm 0.1$	$\pm 0.5^{\circ}\text{C}$ , no pressure, no humidity
Price	\$11.99	\$9.95	\$9.99

# LIGHT/SPECTRAL COLOR SENSOR SELECTION

	<b>AS7341</b>	SI1145	TSL2591
Interface	I2C	I2C	I2C
Voltage	3.3V - 5.0V	3.0V - 5.0V	3.3V - 5.0V
Operating Range	IR, Visible Spectrum (8 channels)	IR, Visible Spectrum	IR, Visible Spectrum
Price	\$25.79	\$15.99	\$8.95

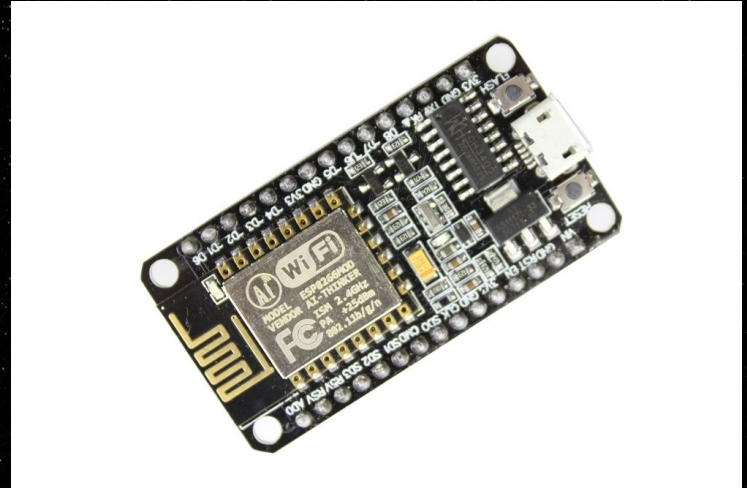


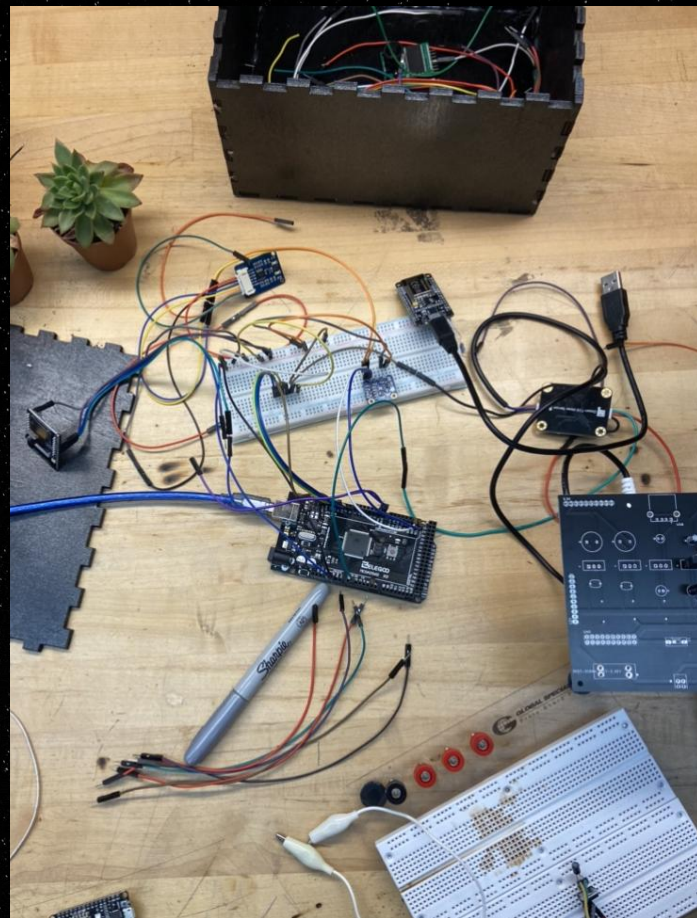
# LIGHTING MODULE SELECTION

	WS2812 LED Matrix	WS2811 LED Strip	TRYSOMDIO Grow Lights
Voltage	5V	5V	5V
Addressable LEDs?	Yes	Yes	No
Size	3.15 in x 3.15 in	3.2 ft	12.87 in x 6.77 in x 3.3 in
Price	\$11.99	\$9.99	\$21.99

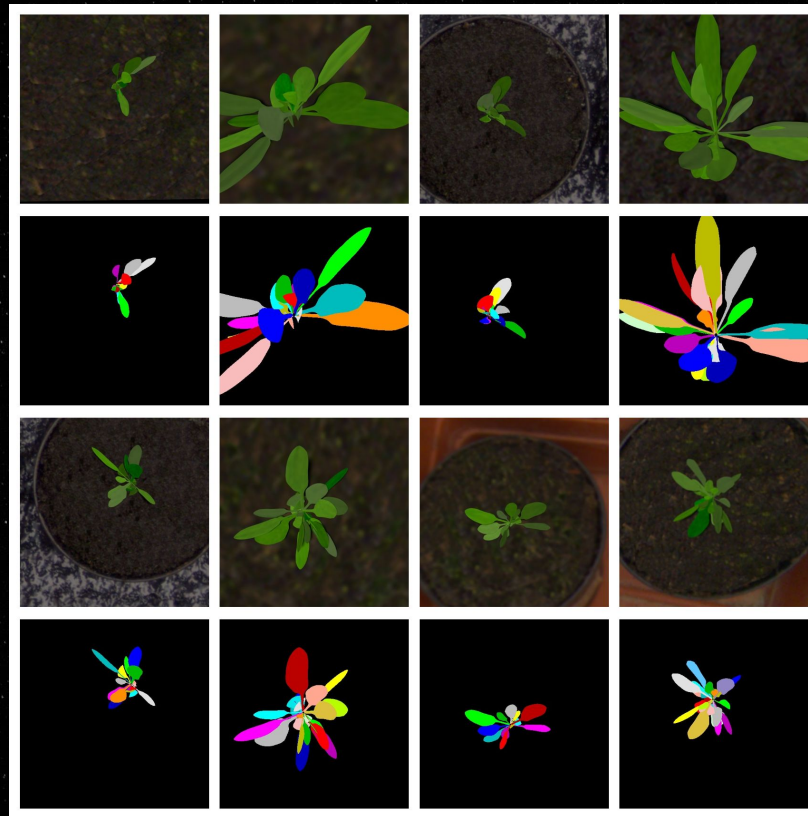
# WIFI MICROCONTROLLER - ESP8266

- Allows wireless transfer of data to our server
- Full WiFi front-end (client/access point)
- TCP/IP stack with DNS support
- Antenna trace: 2.4 GHz

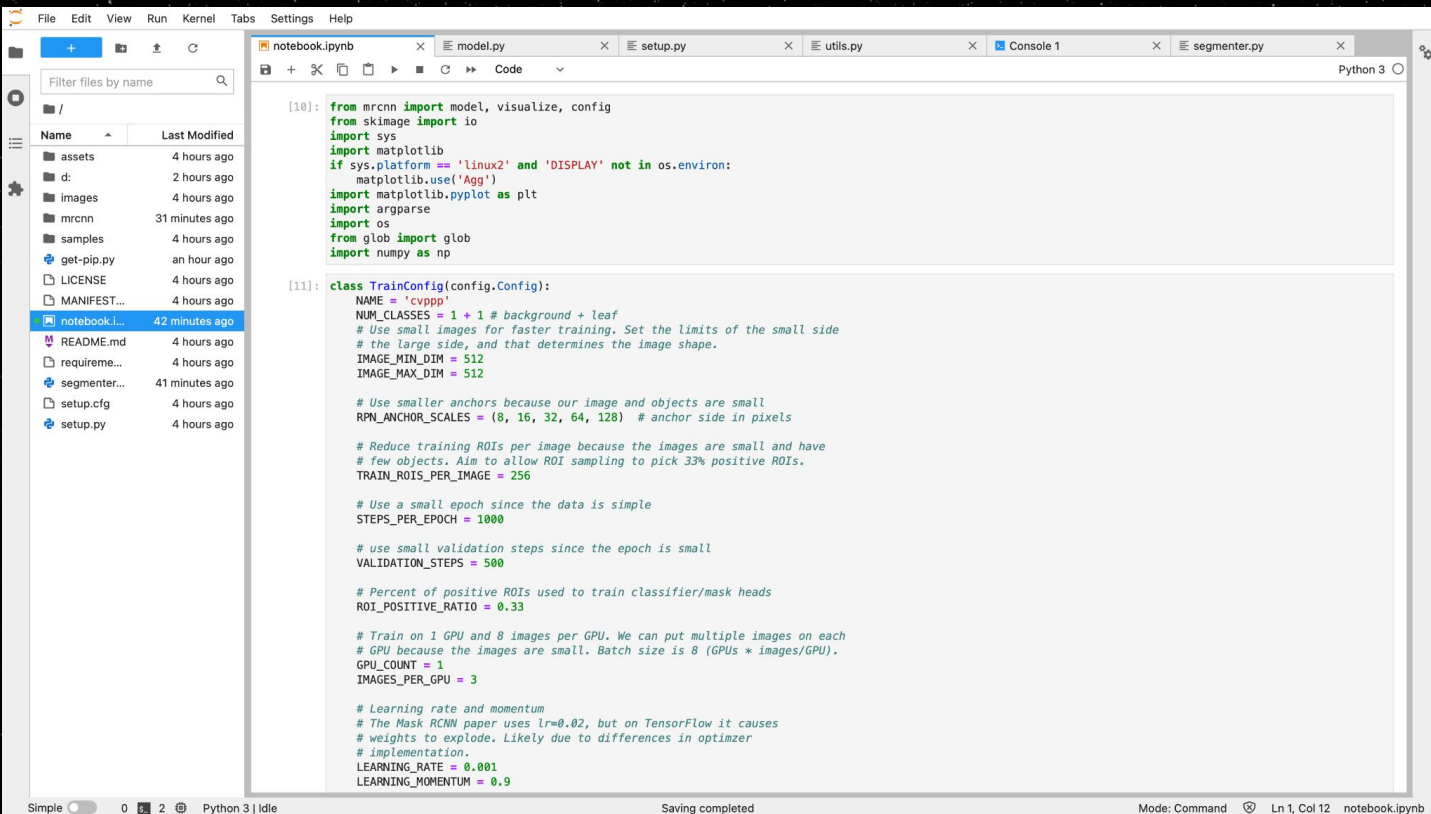




# MACHINE LEARNING RAW AND LABELED IMAGES



# MACHINE LEARNING JUPYTER NOTEBOOK



The screenshot displays the Jupyter Notebook interface. On the left, a file browser shows a directory structure with files like 'assets', 'd:', 'images', 'mrcnn', 'samples', 'get-pip.py', 'LICENSE', 'MANIFEST...', 'notebook.i...', 'README.md', 'requireme...', 'segmenter...', 'setup.cfg', and 'setup.py'. The 'notebook.i...' file is selected, showing it was modified 42 minutes ago.

The main area shows a code editor with the following Python code:

```
[10]: from mrcnn import model, visualize, config
      from skimage import io
      import sys
      import matplotlib
      if sys.platform == 'linux2' and 'DISPLAY' not in os.environ:
          matplotlib.use('Agg')
      import matplotlib.pyplot as plt
      import argparse
      import os
      from glob import glob
      import numpy as np

[11]: class TrainConfig(config.Config):
      NAME = "cvppp"
      NUM_CLASSES = 1 + 1 # background + leaf
      # Use small images for faster training. Set the limits of the small side
      # the large side, and that determines the image shape.
      IMAGE_MIN_DIM = 512
      IMAGE_MAX_DIM = 512

      # Use smaller anchors because our image and objects are small
      RPN_ANCHOR_SCALES = (8, 16, 32, 64, 128) # anchor side in pixels

      # Reduce training ROIs per image because the images are small and have
      # few objects. Aim to allow ROI sampling to pick 33% positive ROIs.
      TRAIN_ROIS_PER_IMAGE = 256

      # Use a small epoch since the data is simple
      STEPS_PER_EPOCH = 1000

      # use small validation steps since the epoch is small
      VALIDATION_STEPS = 500

      # Percent of positive ROIs used to train classifier/mask heads
      ROI_POSITIVE_RATIO = 0.33

      # Train on 1 GPU and 8 images per GPU. We can put multiple images on each
      # GPU because the images are small. Batch size is 8 (GPUS * images/GPU).
      GPU_COUNT = 1
      IMAGES_PER_GPU = 3

      # Learning rate and momentum
      # The Mask RCNN paper uses lr=0.02, but on TensorFlow it causes
      # weights to explode. Likely due to differences in optimizer
      # implementation.
      LEARNING_RATE = 0.001
      LEARNING_MOMENTUM = 0.9
```

The status bar at the bottom indicates the notebook is in 'Simple' mode, running Python 3 | Idle, with the file 'notebook.ipynb' saved. The current cursor position is at Line 1, Column 12.

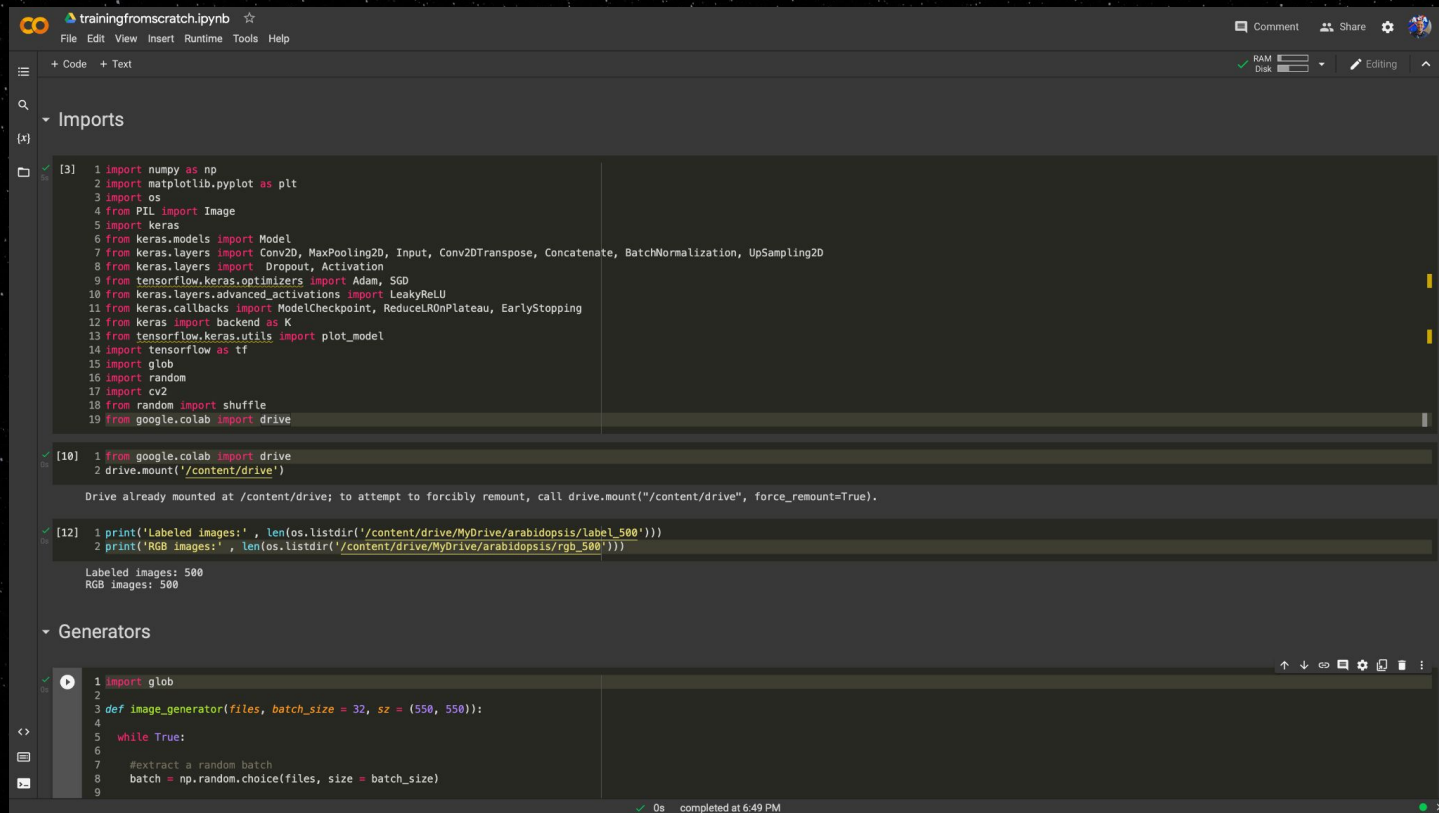
# INSTANCE SEGMENTATION ON SERVER

The screenshot displays a server environment with three main windows:

- Left Window:** A window titled "plantimglight1.png" showing a segmented image of a plant. The image is divided into numerous colored regions, representing individual leaves or parts of the plant. The status bar at the bottom indicates "2000 x 2000 pixels 49.0 kB 53%".
- Middle Window:** A window titled "plantimglight1.png" showing the original image of the plant. The status bar at the bottom indicates "2000 x 2000 pixels 3.7 MB 53%".
- Right Window:** A window titled "leafCounts.csv - LibreOffice Calc" showing a spreadsheet with a table of image counts. The table has columns for "Image" and "Count". Below the spreadsheet, a file manager window titled "output" displays a grid of files, including "leafCounts.csv" and several "plantimglight\*.png" files. The status bar at the bottom indicates "leafCounts" and "Sheet 1 of 1".

Image	Count
2. plantimglight1.png	36
3. plantimglight2.png	35
4. plantimglight3.png	14
5. plantimglight4.png	0
6. plantimglight5.png	13
7. plantimglight6.png	33
8. plantimglight7.png	12
9. plantimglight8.png	24
10. plantimglight9.png	6
11. plantimglight10.png	5

# COMPARISON MODEL USING U-NET



The image shows a Jupyter Notebook window titled "trainingfromscratch.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with icons for code and text, and a RAM/Disk usage indicator. The notebook content is organized into sections: "Imports", "Generators", and "Outputs".

```
[3] 1 import numpy as np
2 import matplotlib.pyplot as plt
3 import os
4 from PIL import Image
5 import keras
6 from keras.models import Model
7 from keras.layers import Conv2D, MaxPooling2D, Input, Conv2DTranspose, Concatenate, BatchNormalization, UpSampling2D
8 from keras.layers import Dropout, Activation
9 from tensorflow.keras.optimizers import Adam, SGD
10 from keras.layers.advanced_activations import LeakyReLU
11 from keras.callbacks import ModelCheckpoint, ReduceLRonPlateau, EarlyStopping
12 from keras import backend as K
13 from tensorflow.keras.utils import plot_model
14 import tensorflow as tf
15 import glob
16 import random
17 import cv2
18 from random import shuffle
19 from google.colab import drive

[10] 1 from google.colab import drive
2 drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[12] 1 print('Labeled images:', len(os.listdir('/content/drive/MyDrive/arabidopsis/label_500')))
2 print('RGB images:', len(os.listdir('/content/drive/MyDrive/arabidopsis/rgb_500')))

Labeled images: 500
RGB images: 500

Generators

[ ] 1 import glob
2
3 def image_generator(files, batch_size = 32, sz = (550, 550)):
4
5     while True:
6
7         #extract a random batch
8         batch = np.random.choice(files, size = batch_size)
9
```

0s completed at 6:49 PM

# CAMERA TESTING

ESP32 OV2640

Not Secure | 192.168.87.229

Toggle OV2640 settings

Resolution: CIF(400x296)

Quality: 10 / 63

Brightness: -2 / 2

Contrast: -2 / 2

Saturation: -2 / 2

Special Effect: No Effect

AWB:

AWB Gain:

WB Mode: Auto

AEC SENSOR:

AEC DSP:

AE Level: -2 / 2

AGC:

Gain Ceiling: 2x / 128x

BPC:

WPC:

Raw GMA:

Lens Correction:

H-Mirror:

V-Flip:

DCW (Downsize EN):

Color Bar:

Face Detection:

Face Recognition:

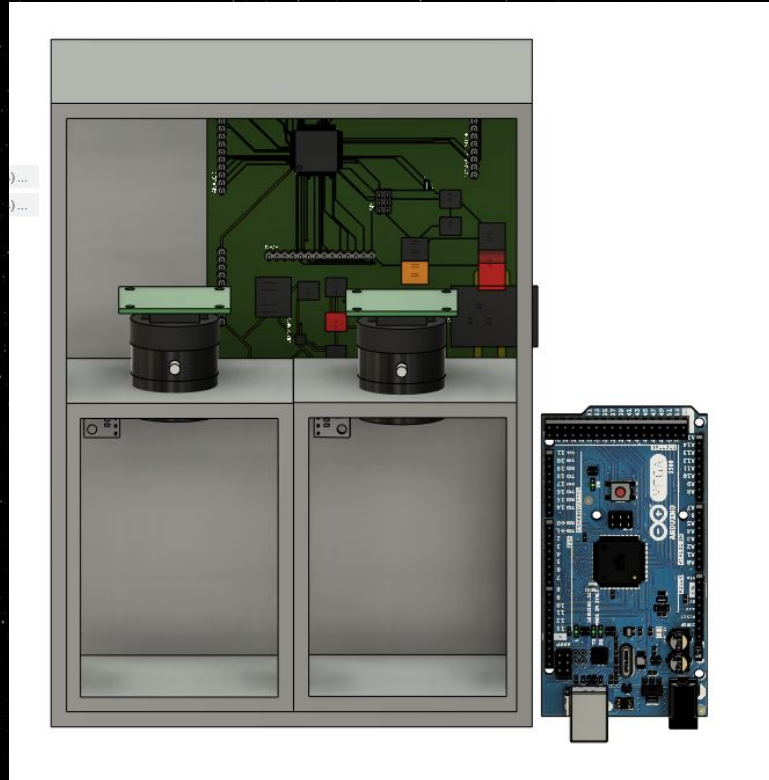
Get Still Stop Stream Enroll Face



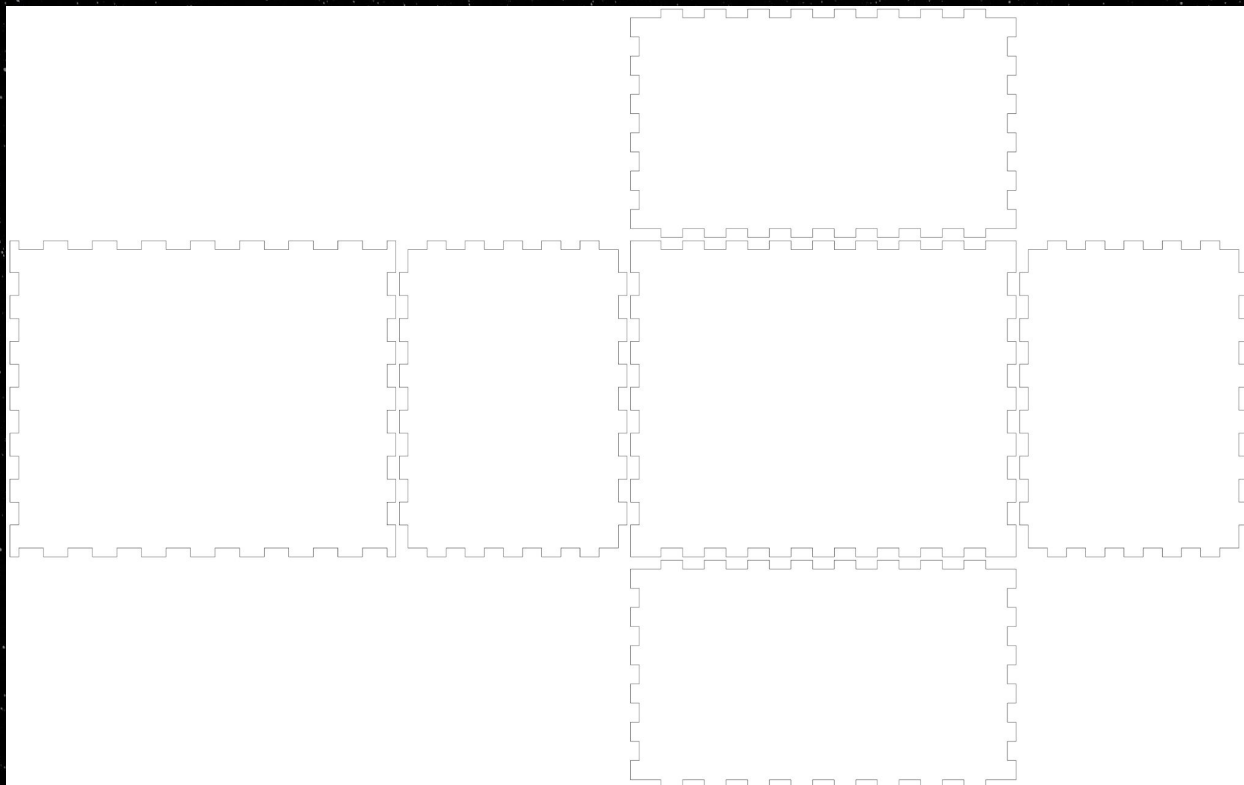


# SYSTEM DESIGN

# PAYLOAD DESIGN ON FUSION 360



# PAYLOAD DESIGN



# POWER REQUIREMENTS

Designing a power budget:

The budget is determined by orbit, efficiency of the photovoltaic cells.

Evaluation can be simulated using tools like orbit simulators such as Systems tool kit(STK). This tool is also useful for communication simulations for satellites.

Between this and data supplied by cubesat vendors

Other Subsystems will require power that are necessary for space flight

# POWER DESIGN

Electrical Power Systems

Given Estimated Power Production

19.2 watts per panel

38.4 watts

Estimated Average Payload Power Available

10-30 W

3.3V, 5V outputs

# POWER SUPPLY

Because the system is a research platform we ended up using a external power supply using a 2.1mm power jack.

A variable power supply was ordered to provide the system a range of voltages for testing and system testing of the research platform.

Universal Power Adapter 4V 5V 6V 9V 12V 15V 18V 19V 20V 24V  
Adjustable Switching Power Supply.

The system can also be run on a external 12v battery if ground testing requires.



# MCU SELECTION

Item	Description	Price
atmega2560	16 MIPS and operates at 16 MHz with 256 kBytes flash memory	\$15.67
atmega168	20 MHz and 16KBytes of In-System Self-Programmable Flash program	\$4.77
MSP430	24-MHz clock and features 32KB of embedded FRAM	\$15.59

Several factors in determining the appropriate microcontroller for this project.

Cost and capacity.

Can it store camera data and is their support?

There was only one choice choice is obvious among these options.

# ATMEGA2560

- Is a low-power microcontroller that operates between 4.5 and 5.5 volts
- Capable of 16 MIPS and operates at 16 MHz using an external crystal oscillator.
- 100 pins with 86 general purpose I/O lines
- 16 10-bit A/D converter
- Operating temp -40 - 85





# HARDWARE DESIGN (ORIGINAL)

Voltage Converter:

12V to 3.30V @ 3 (TPS56339)

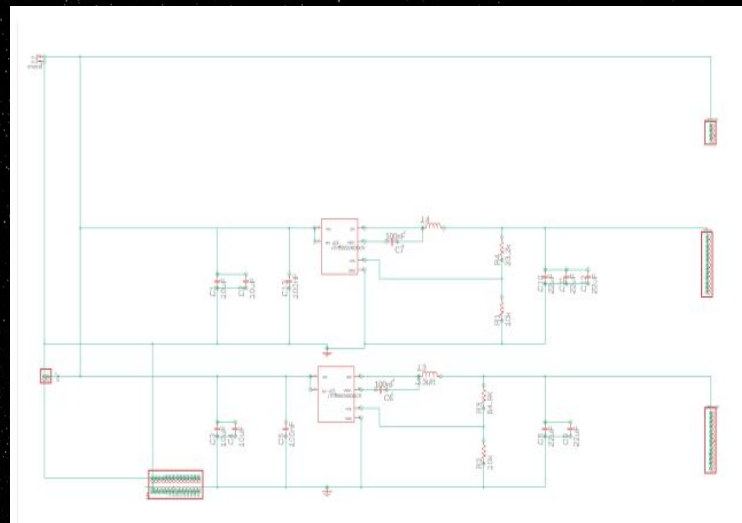
Efficiency: 89.8%

Topology: synchronous switching step-down(Buck)

12v to 5.00V @ 5A using (TSP565208)

Efficiency: 94.3%

Topology: synchronous switching step-down(Buck)



# HARDWARE DESIGN

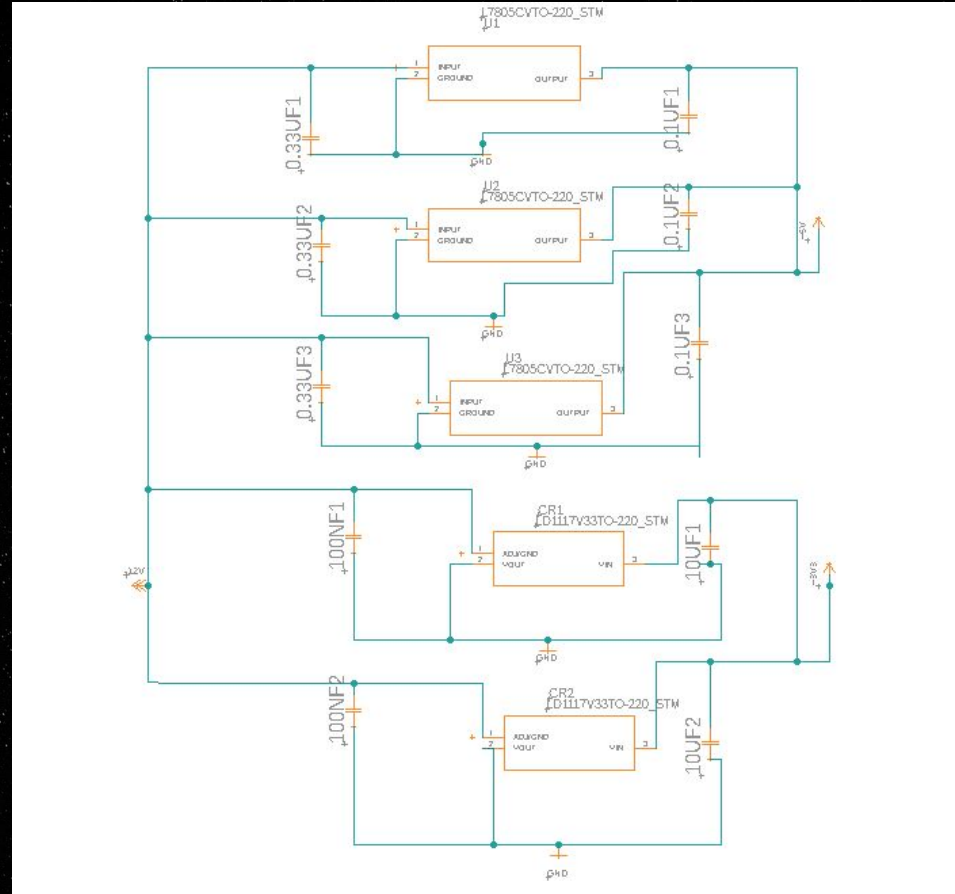
Voltage Converter:

12V to 3.30V @ 800mA each

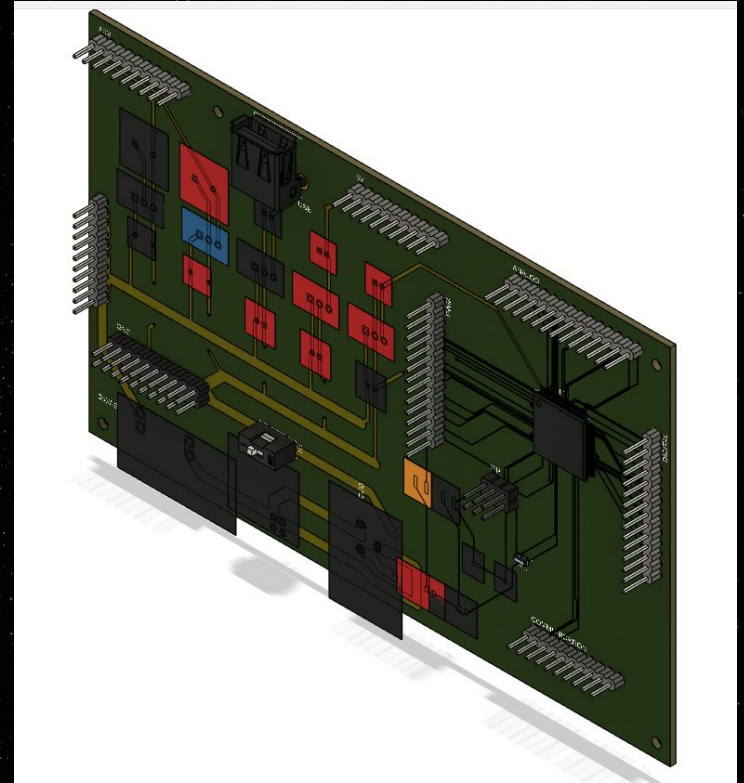
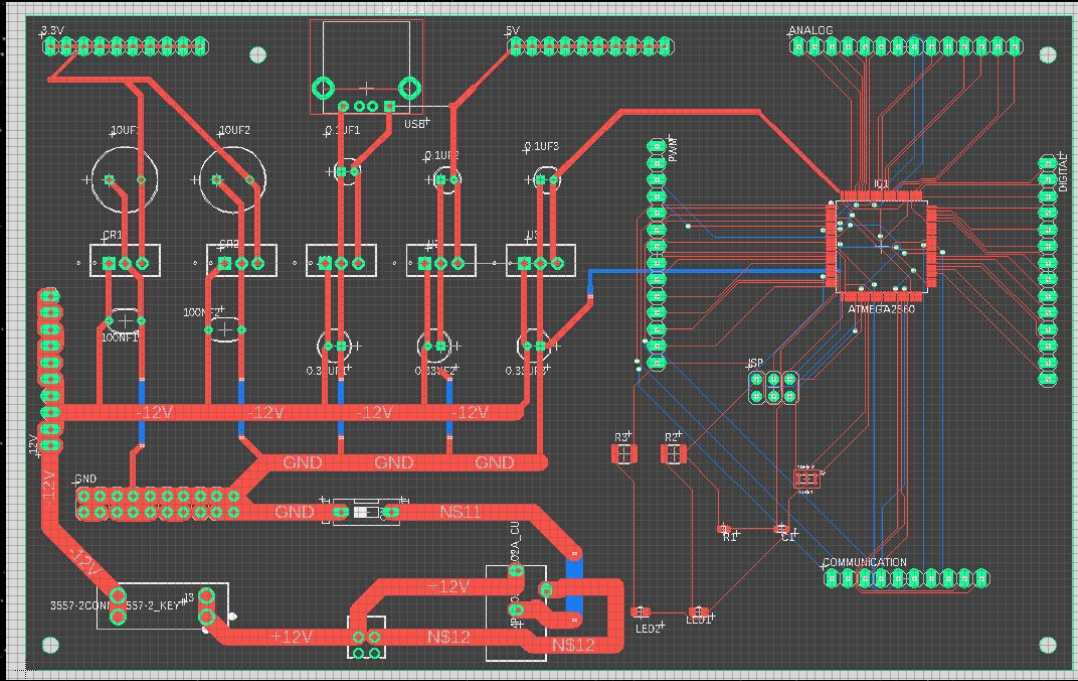
Topology: linear Voltage Regulator

12v to 5.00V @ 1.5A each

Topology: linear Voltage Regulator



# HARDWARE DESIGN



# ARDUINO IDE

Arduino Integrated Development Environment -  
or Arduino Software (IDE) -

It contains a text editor for writing code,  
a message area and it connects to the  
Arduino hardware to upload programs and  
communicate with them.

Many of the sensors have libraries on github  
that allowed for simple integration with the  
arduino platform.



# ARDUINOJSON

ArduinoJson is one of the best libraries that allows for the serialization and transfer of data to the ESP transceiver module and then on send to the server.

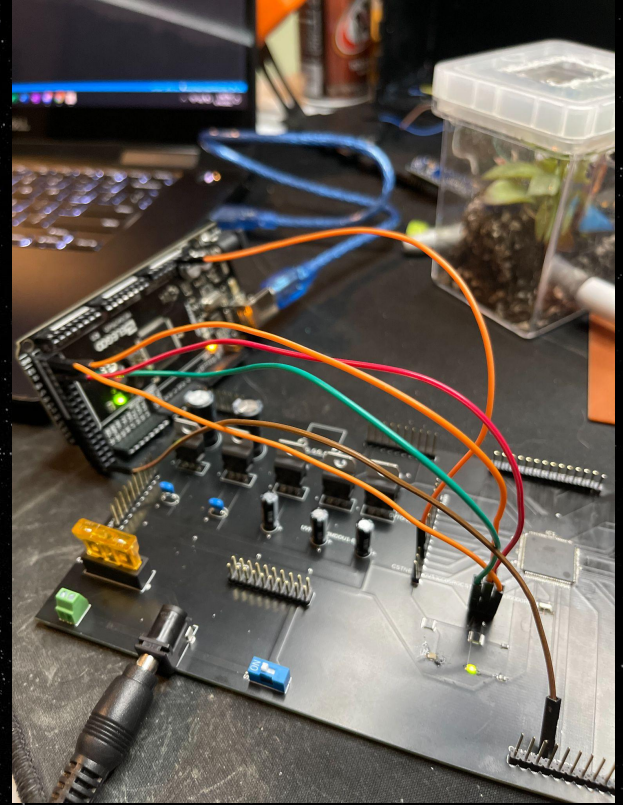
This was also useful for the parsing of data when it came time to display data on the server side.



# PROGRAMMING THE ATMEGA 2560

Using ISP (AKA. ICSP) Arduino platform comes integrated with the ability to use its development boards as programmers for any of its supported chips.

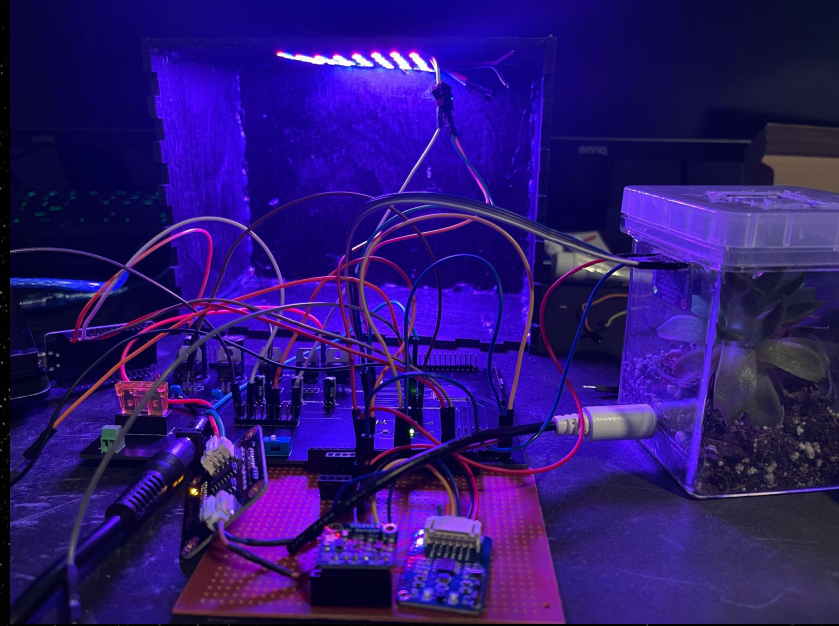
Future integration of a usb for communication was determined to be beneficial because of the space constraints for plugging in programming leads.



# LESSONS LEARNED

Proper connectors would be beneficial to avoid the loose connection and issues associated with that. Or in the case of this system integrated circuit boards with leads to the sensors used. This relies on the fact is the entire system can not iterate easily.

The Arduino platform is great for initial development but a Real-Time operating system for timing actions of sensors and sending data. This will reduce power consumption for some components being in an always on state.



# MACHINE LEARNING SPECIFICATIONS AND REQUIREMENTS

- Analyze pretrained model: Mask R-CNN using Synthetic Arabidopsis Dataset
  - Dataset size: 10,000 images
  - Images (width × height): 550×550 pixels
  - Output: instance segmentation
- Train new model: YOLOv5/U-NET using Synthetic Arabidopsis Dataset
  - Dataset size: 10,000 images
  - Images (width × height): 550×550 pixels
  - Output: instance segmentation
- Compare models
  - Accuracy of YOLOv5/U-NET to Mask R-CNN



# MACHINE LEARNING REMOTE SERVER

```
root@PlantPodServer:~/plantpod/api/machine_learning# sudo python3 segmenter.py --inputImage './plant-test.png' --weightsPath ./leafSegmenter0005.h5 --useCPU
Using TensorFlow backend.
2022-03-22 20:20:45.656430: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlopen error: libcuda.so.1: cannot open shared object file: No such file or directory
2022-03-22 20:20:45.656666: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2022-03-22 20:20:45.656773: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (PlantPodServer): /proc/driver/nvidia/version does not exist
2022-03-22 20:20:45.657183: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2022-03-22 20:20:45.670496: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2494085000 Hz
2022-03-22 20:20:45.670893: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7fca24000b60 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2022-03-22 20:20:45.671014: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2022-03-22 20:20:48.958220: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 51380224 exceeds 10% of free system memory.
2022-03-22 20:20:49.095825: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 51380224 exceeds 10% of free system memory.
2022-03-22 20:20:49.149676: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 51380224 exceeds 10% of free system memory.
2022-03-22 20:20:51.004048: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 51380224 exceeds 10% of free system memory.
2022-03-22 20:20:58.049938: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 33554432 exceeds 10% of free system memory.
RESULTS!!!!!!!!!!!! >>>14
root@PlantPodServer:~/plantpod/api/machine_learning#
```

# MACHINE LEARNING UCF SERVER PROGRESS

```
Traceback (most recent call last):
  File "segmenter.py", line 2, in <module>
    import mrcnn.config as config
  File "/lustre/fs0/home/neltenn/env/Mask_RCNN/GitHub/plantpod/api/machine_learning/mrcnn/config.py", line 10, in <module>
    import numpy as np
  File "/lustre/fs0/home/neltenn/env/lib64/python3.6/site-packages/numpy/__init__.py", line 140, in <module>
    from . import core
  File "/lustre/fs0/home/neltenn/env/lib64/python3.6/site-packages/numpy/core/__init__.py", line 48, in <module>
    raise ImportError(msg)
ImportError:
```

IMPORTANT: PLEASE READ THIS FOR ADVICE ON HOW TO SOLVE THIS ISSUE!

Importing the numpy C-extensions failed. This error can happen for many reasons, often due to issues with your setup or how NumPy was installed.

We have compiled some common reasons and troubleshooting tips at:

<https://numpy.org/devdocs/user/troubleshooting-importerror.html>

Please note and check the following:

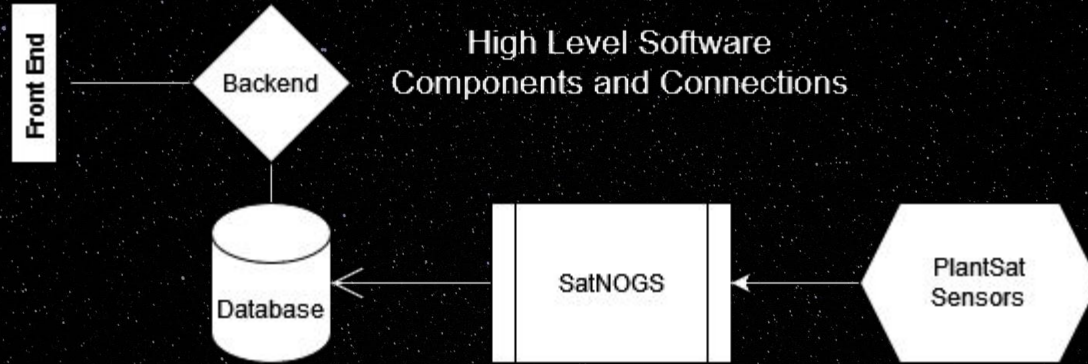
- \* The Python version is: Python3.6 from "/lustre/fs0/home/neltenn/env/bin/python3"
- \* The NumPy version is: "1.19.5"

and make sure that they are the versions you expect.  
Please carefully study the documentation linked above for further help.

Original error was: PyCapsule\_Import could not import module "datetime"

```
Segmentation fault (core dumped)
(env) [neltenn@evuser1 machine_learning]$ █
```

# SOFTWARE SIDE - OVERVIEW



- Easily manage and compare data for scientists.
- Website, Database, and API

# WEB REQUIREMENTS

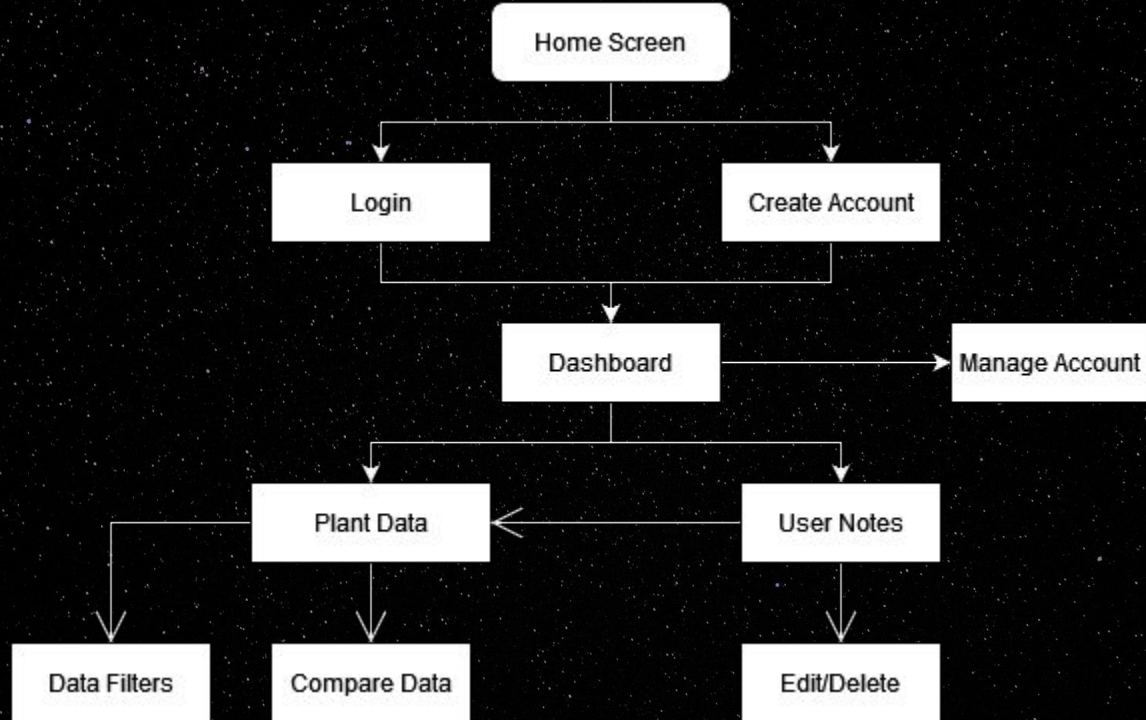
Users can log in to see:

- History
- Notes
- Photographs

Notes:

- Compare
- Edit
- Delete

# FRONTEND BLOCK DIAGRAM







# PROTOTYPE DATA COMPARISON PAGE

First Row

Data 1 Label	Data 1
Data 2 Label	Data 2
Data 3 Label	Data 3
Data 4 Label	Data 4
Data 5 Label	Data 5
Data 6 Label	Data 6
Data 7 Label	Data 7

First Row Image

Second Row

Data 1 Label	Data 1
Data 2 Label	Data 2
Data 3 Label	Data 3
Data 4 Label	Data 4
Data 5 Label	Data 5
Data 6 Label	Data 6
Data 7 Label	Data 7

Second Row  
Image

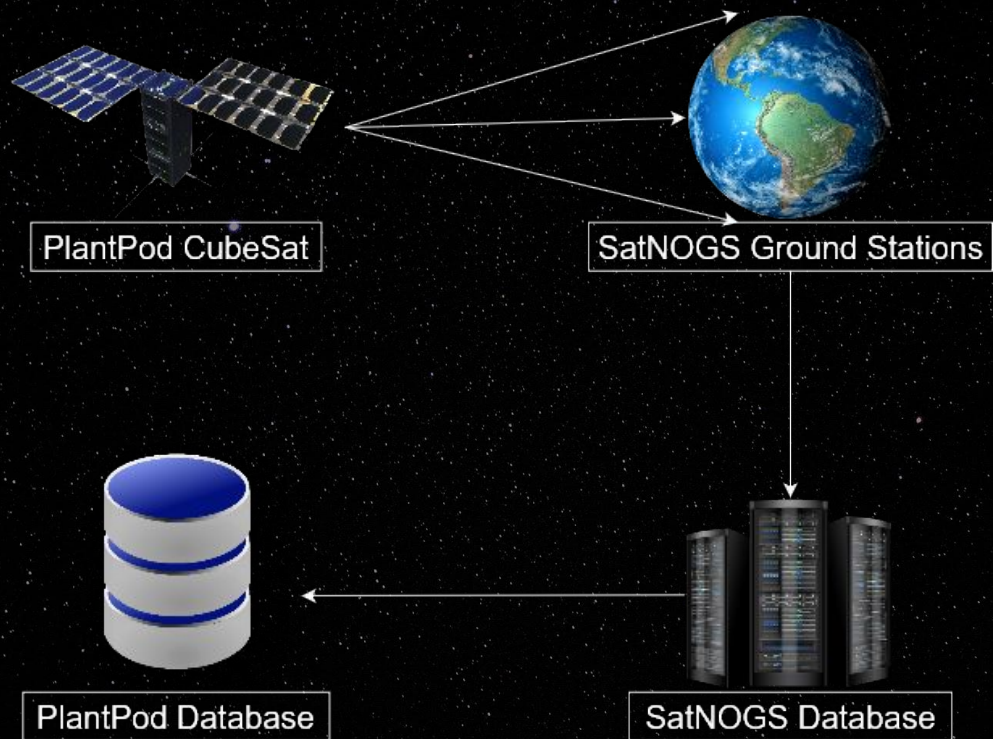
Difference

Data 1 Label	Data 1
Data 2 Label	Data 2
Data 3 Label	Data 3
Data 4 Label	Data 4
Data 5 Label	Data 5
Data 6 Label	Data 6
Data 7 Label	Data 7

Crossfade?



# DATA PATH



# TECHNOLOGIES USED

Websites & Apps	Web	Modified MERN Stack
GitHub	JavaScript	SQLite3
VSCode	CSS	ExpressJS
DigitalOcean	HTML	ReactJS
		NodeJS

# WEB APPLICATION HOSTING



	Digital Ocean	GitHub
Price	Free \$100 from Github Education	Free
Purpose	Virtual Machine Hosting	Collaboration

# SOFTWARE SUCCESS AND CHALLENGES

## Successes

- Useful libraries for React
- Hosting on DigitalOcean as a Droplet

## Difficulties

- Transfer of data from physical sensors into database
- Swapping project focus from ISS module to satellite form

# TEAM ORGANIZATION

	Discord weekly meetings
	Trello board
	Google Drive
	GitHub

# BUDGET AND FINANCE THROUGH UCF

Item	Amount	Cost	Total
Arabidopsis Wild-Type Seed, Pack of 200	1	\$16.10	\$16.10
HiLetgo 5pcs Micro SD TF Card Adater Reader Mod	1	\$6.99	\$6.99
Adafruit SI1145 Digital UV Index/IR / Visible Light	1	\$15.99	\$15.99
LED Grow Lights for Indoor Plants, Full Spectrum P	1	\$21.99	\$21.99
Magenta GA-7 Plant Culture Box with Lid, Karter Sc	2	\$6.99	\$13.98
Bicool AS7341 Spectral Color Sensor 8X Visible Spe	2	\$24.99	\$49.98
BTF-LIGHTING WS2812B ECO RGB Alloy Wires 5050SMD I	1	\$11.49	\$11.49
2 Pack ESP32-CAM WiFi Bluetooth Camera Module Deve	1	\$17.99	\$17.99
Arabidopsis Germination Medium	1	\$2.23	\$2.23
Total			\$151.81

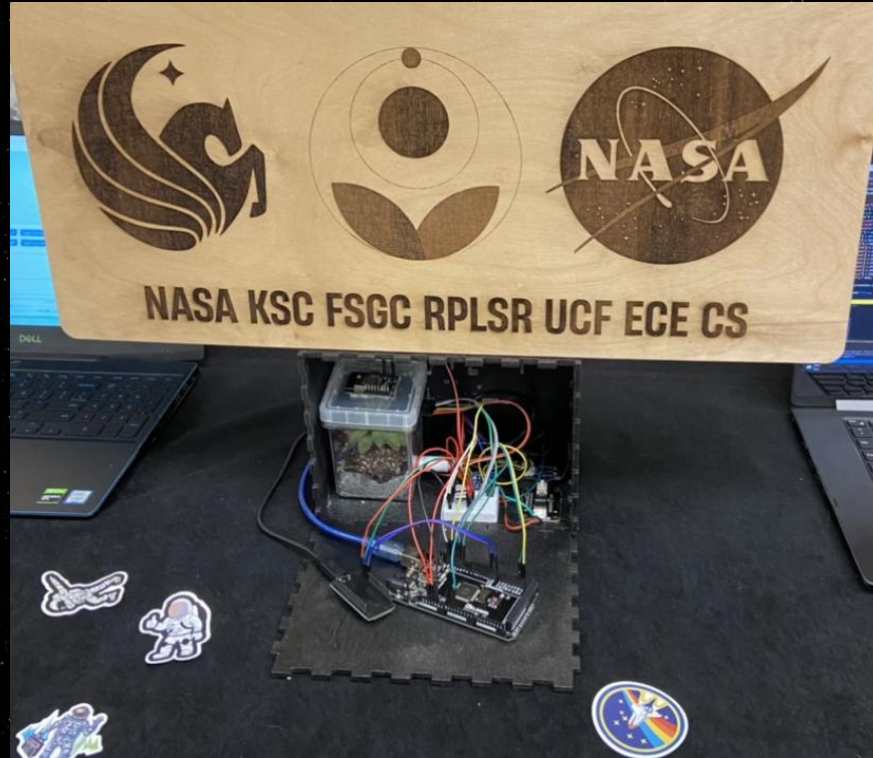
Excluding shipping and handling expenses.

# BUDGET AND FINANCE THROUGH MATT AND NICK

Item	Amount	Cost	Total
Printed Circuit Boards	1	\$21.00	\$21.00
ESP8266	1	\$15.98	\$15.98
Arducam	2	\$25.99	\$51.98
Adafruit BME Sensor	2	\$20.96	\$41.92
Total Dissolved Solids Sensor	1	\$13.99	\$13.99
Jumper cables	1	\$6.79	\$6.79
Solder 2 sizes	2	\$8.99	\$17.98
Power Supply	1	\$15.95	\$15.95
ATMega2560 Chips	1	\$38.99	\$38.99
Printed Circuit Board Components	1	\$50.00	\$50.00
Petri dishes	1	\$12.99	\$12.99
MS Medium	1	\$23.36	\$23.36
Micro SD Card	1	\$13.99	\$13.99
Total			\$324.92

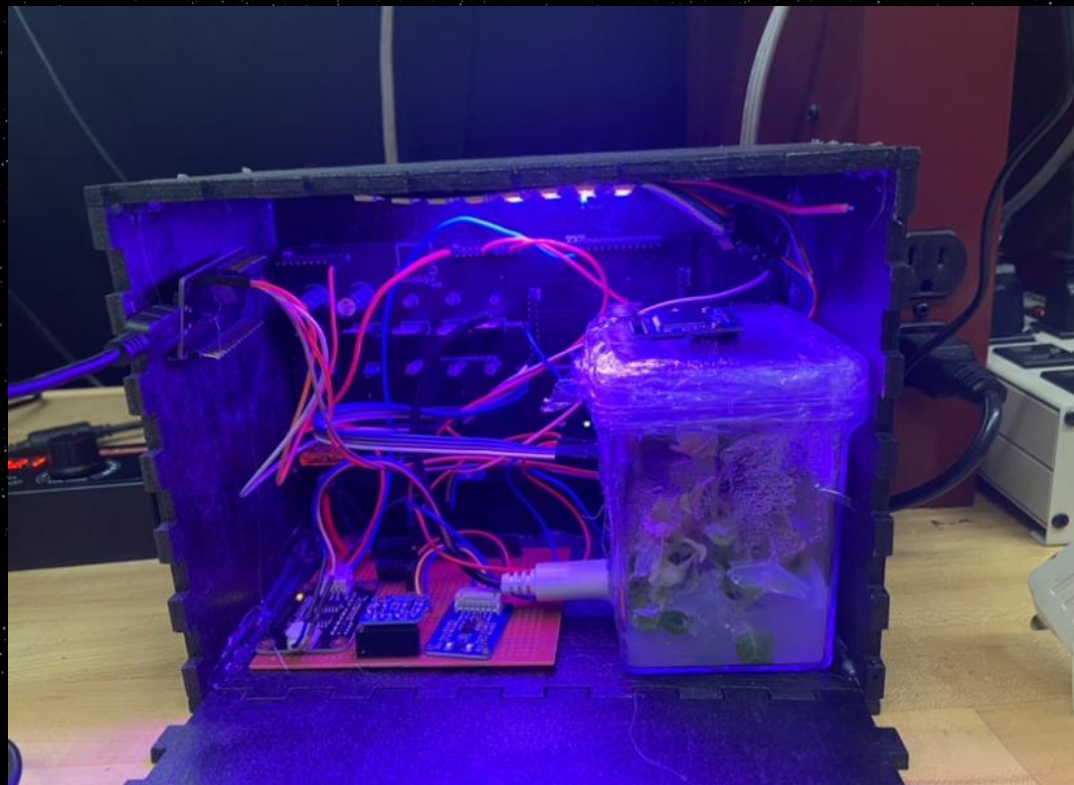
Excluding shipping and handling expenses.

# STEM DAY AT UCF





# WORKING PROTOTYPE



# NEXT STEPS FOR FUTURE SENIOR DESIGN TEAMS

- Focus on edible leafy greens that help meet nutritional goals
- Grow test plants media in collaboration with Biology Department
- Incorporate a vent system for the payload for rate of diffusion
- Research Wisconsin Fast Plant (Brassica) instead of Arabidopsis
- Research connecting with the Deep Space Network for communication
- Conduct system vibration testing to simulate rocket launch
- Implement 3 layers of redundancy instead of radiation hardening
- Integrate and test solar panels to ensure independent power supply



**THANK YOU FOR WATCHING**