# DroneX: Drone Detection System

Harrison Kennedy, Nick King, Youssef Barsoom, and Stephan Saturne

College of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, 32826, USA

**Abstract — This paper demonstrates the electronic hardware and software design of the DroneX: Drone Detection System. The DroneX: Drone Detection System utilizes machine learning/object detection software to accurately and efficiently detect a drone. This system streams the video of a drone detected by the system and is accessible through our DroneX website. This stream uses wifi to transmit the data. This project aims to aid the drone security system market, due to the rapid growth in drone use, more nefarious activities with drones is also on an increase. Providing a system that begins to chip away at illegal drone incursions is a step in the right direction.**

***Index terms* – machine learning, object detection, Wifi, drone incursions.**

## I. Introduction

Drone incursions at airports, sports arenas, and other secure sites is a fast growing problem that has yet to be solved. Drone incursions cause several problems such as closing airports, and damaging aircraft, endangering civilians. In this society, technology is fast growing, so much that it is estimated that 2.4 million consumer drones will be sold in 2023. Each of those drones are capable of carrying out nefarious activities and putting lives at risk.

DroneX: Drone Detection System consists of 4 main sections. The first is the machine learning and object detection software that has been developed to efficiently recognize a drone from our live stream apart from other objects in the field of view, FOV, such as helicopters, birds, airplanes, clouds, etc. The second section is the Jetson Nano along with the Camera. These components house the machine learning algorithm and are the eyes to the system. Our system uses the YOLOv5 object detection model. Next, is the printed circuit board that powers the jetson nano and camera. This pcb also holds the microcontroller that will light up an LED to indicate a detection. The PCB receives power from a DC input jack and utilizes multiple voltage regulators to accurately supply each component with the desired required voltage. The final section of the system is the DroneX website. This website will present the livestream of the drone detection system, this allows the consumer to see each drone incursion simply by accessing the DroneX website.

## II. Background

Object detection is a computer vision technique that involves identifying and localizing objects within an image or video. It is a crucial task in many applications, such as autonomous vehicles, security systems, and robotics. Object detection using machine learning involves training a model to recognize and classify objects in images or videos. The model is typically trained on a large dataset of labeled images, where each image is annotated with bounding boxes around the objects of interest.

YOLOv5 uses a fully convolutional architecture, which allows it to process images of different sizes without requiring any resizing or cropping. This makes it more flexible and easier to use than earlier versions of YOLO, which had fixed input sizes.

## III. System Components

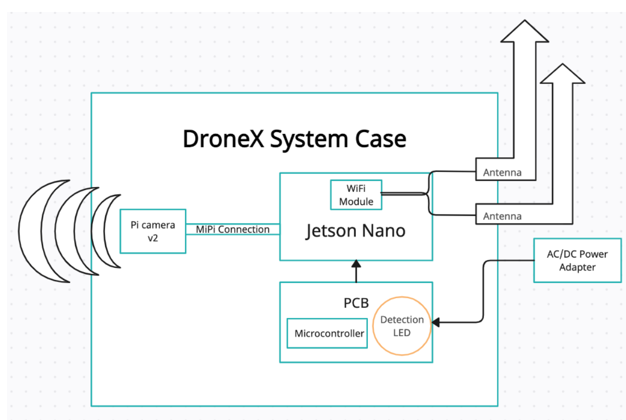



*Fig. X Hardware Block Diagram*

System components describe all of the hardware parts we selected to implement our project design. This will include all of the parts in our PCB design as well as all of the auxiliary parts which are listed: the Jetson Nano, the waveshare wireless, MSP430FR6989, Pi camera V2 8mp, Voltage regulators, and a DC power jack The functionality and reason for each parts selection will be described as well in the following sections.

## A. Nvidia Jetson Nano Developer Kit

The Nvidia Jetson Nano 4GB module is the machine learning assistant device that we have chosen. This device comes equipped with a CPU and GPU to make for machine learning on the edge device quick and easy. The CPU used is the Quad-core ARM A57 @ 1.43 GHz and the GPU is the 128-core NVIDIA Maxwell.[1] The Jetson Nano also comes preinstalled with a 40 pin header, Mipi connections for cameras and has a vast library of resource tools that all made development with this board much easier.[1] When comparing this board to other manufacturers' equivalents such as the Coral Dev Board, or even the Raspberry Pi it was clear that the Jetson was far superior in support options as well as overall functionality.

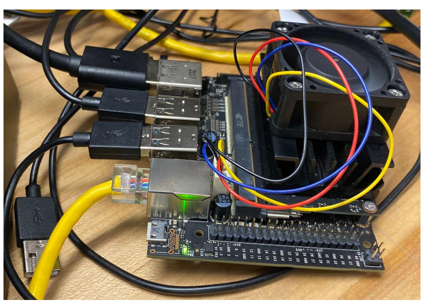


Fig 1. This is a picture of our Jetson Nano amidst testing. All of the wires seen plugged in are connected to a mouse, keyboard, monitor, and an ethernet cable for internet connection. When in use none of these wires are present.

## B. Waveshare AC8625 Wireless Module

We decided on using Wifi as our communication protocol after we discovered that a previously purchased cellular module was damaged in production and it was found that the reordering of the same cellular module was not feasible financially or within the timeline we needed for our project. Thus we decided to purchase the much cheaper and more readily available Waveshare AC8625 Wireless Module to give the Jetson Nano the Wifi integration needed to connect to the internet without the need for an ethernet connection such that we can effectively move the device to many outdoor locations with ease.

Due to the device primarily being an outdoor option it was originally a concern that WIFI would be unavailable for use in such conditions, however in our research we discovered that WIFI does have the capability to work within a 300 foot diameter outside without the impediment of obstacles, or 150 feet with obstacles in the way such as in an indoor environment with walls.

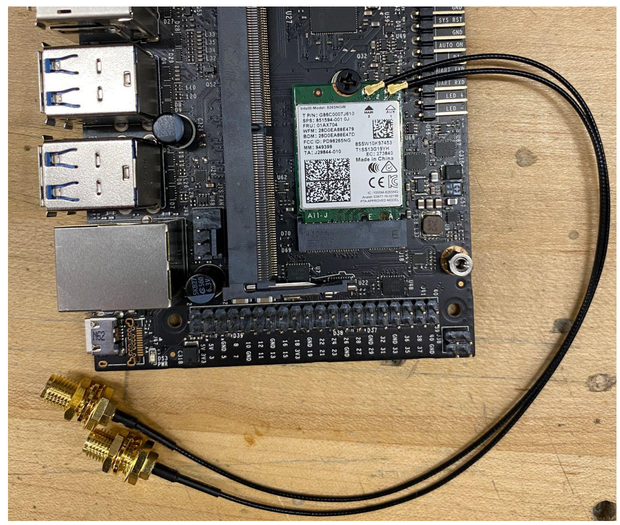


Fig 2. This is the placement of our Wifi Module as can be seen it plugs in underneath the heatsink and fan combination.

## C. MSP430FR6989 Microcontroller

For our design we originally chose to use a microcontroller to control the communication between the now broken cell module and the Jetson Nano using the UART communication protocol, however since the cell module is no longer in the design we have little need for this module. We will still show the schematic of this module and the development that took place as it was originally a crucial component of the design.

In the below section of schematic we show the connections made to our MSP430FR6989 pins. We have a direct connection to the 3.3 volt regulator output for powering of the component but we also have 5 pin header connections. These pin header connections include the development pins which allow us to program the module while on the PCB as well as the UART connection pins that will connect directly to the Jetson Nano. The rest of the pins connected to the pin headers are connected to I/O pins such that we can add additional functionality and expand from the MSP430FR6989. The flashing of the UART communication occurred by connecting to the devpins pin header allowing for quick development.

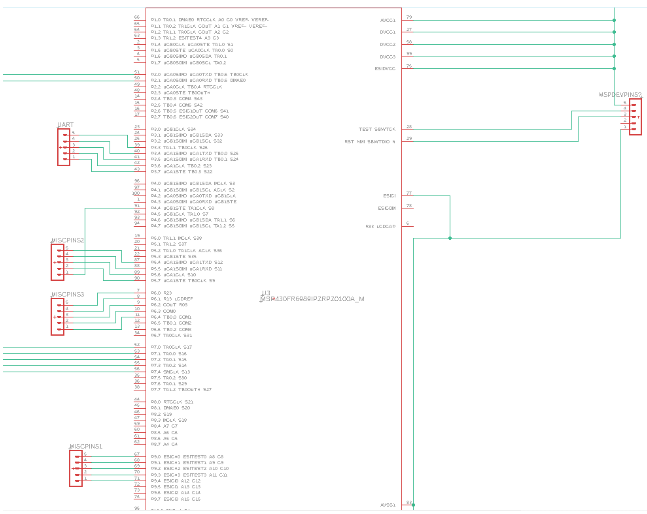


Fig 3. This figure shows the schematic connections made to our microcontroller. This is mainly UART, I/O, and Development pins.

## D. Pi Camera v2 8MP

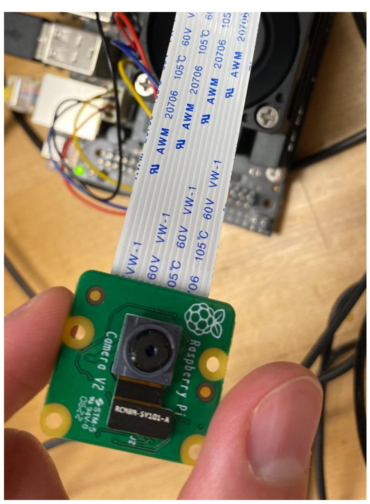


Fig 4. This is the camera Module plugged into the Jetson nano it can be seen it is connected through a MIPI wire.

The camera module we chose was highly researched and tested as we tested multiple different types of cameras to find the best one that most efficiently captured clear images we could run our model on. The Pi Camera V2 8MP was a very cheap option of camera only costing about $30 and was very easy to implement as the camera was specifically made with the Raspberry Pi and Jetson Nano in mind. The camera also has a relatively high resolution compared to the price having as the name suggests 8 Mega pixels per image which greatly increases the fidelity of the image when compared to other cameras we researched such as the e- cam cunano which had a price point 4 times larger but only a resolution of 3.4MP.

## E. Regulators (3.3V and 5V)

We used the Texas Instrument Power Designer to easily design each of our power regulators to minimize the time of production.[2] In our design we originally had 3 different regulators; a 3.3 volt, a 5 volt and a 3.8 volt regulator. Unfortunately the 3.8 volt regulator is no longer in use due to the destruction of a previously used cellular module. Thus we will only discuss the design of the 5 volt and 3.3 volt regulators.

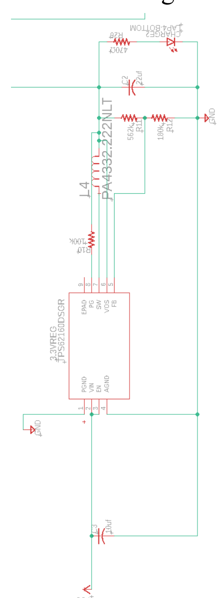


Fig 5. 3.3 volt regulator using TPS62160DSGR buck converter as well as RLC components.

The 3.3 volt regulator takes the dc jack input of 6 volts and steps it down to 3.3 volts such that it can easily power our MSP430FR6989. It is a buck converter design making it very easy to implement with the correct capacitors, resistors and inductors in the correct configuration as shown in the figure. [3]It should also be noted that the regulators present all allow for a varied DC input from 6 to 12 volts. This allows for added flexibility of the input voltage and allows for multiple DC cords to be integrated into our design without a failure. This also allows for batteries of differing sizes to be implemented in the future.

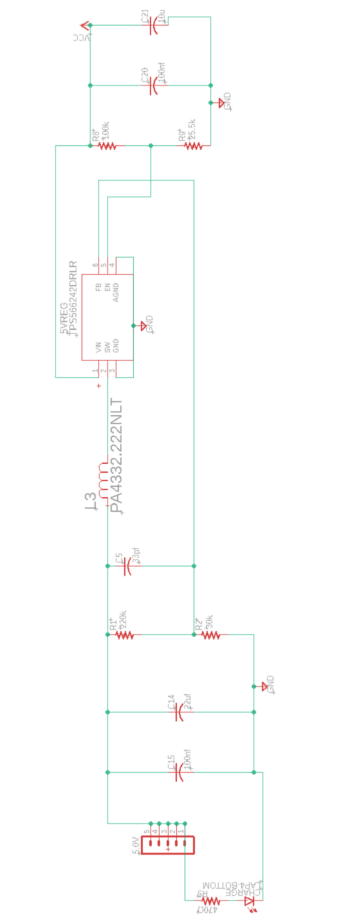



Fig 6. 5 volt regulator using TPS566242DRLR buck converter as well as RLC components

The 5 volt regulator is also powered by the dc jack input in parallel with the previous stated regulator. [4]It powers the Jetson Nano and to minimize the load on the traces on our PCB we decided to implement this 5 volt regulator scheme twice such that we can use the pinheaders of both to power the jetson to half the load on any singular line in our PCB.

## F. DC Power Jack

To power our circuit board we are using the common DC power jack connector. On our board we used this design for a couple of reasons. The first reason we decided to go with this design is due to the filtering capacitors that filter out noise and ripple voltage that can be introduced by the power supply. This helps to smooth out the voltage and provide a cleaner, more stable power supply. Another reason we chose this design was the stabilization factor. The capacitors can also be used to stabilize the voltage in a DC power circuit, especially in situations where there may be fluctuations in the power supply. This can help prevent damage to sensitive electronic components that are elsewhere in the circuit.

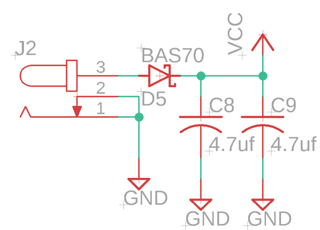
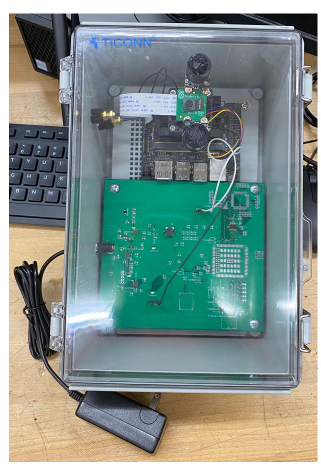



Fig 7. DC power jack schematic

This design allows the board to be plugged directly into a AC/DC adapter which plugs into a wall outlet, while also leaving the possibility of a battery to plug into the system.

# IV. DroneX Case

When designing our DroneX system we had a few requirements for the case that will hold the valuable hardware inside of it. Most importantly the case needed to be waterproof so that the system would be safe to operate in an outside environment. Another key feature of the case is that it required a transparent window or wall so that the camera would be able to see out of the case with minimal interference. Of course it needed to be big enough to house the Jetson Nano, our printed circuit board, the camera, as well as have holes for two antenna ports extending from the wireless module. The case had to be easily drilled into so we could secure the components.

Our case needed to be roughly 10” x 6” x 2.5” minimum to hold everything. We purchased spacers for the Jetson Nano and the PCB so both boards would be free of direct contact with the walls of our case. Lastly, holding our camera in place was an important aspect of the case that needed to be creatively designed as the camera is extremely light and can be easily moved if the case itself is moved. After discussion, velcro was determined to be the best method of holding the camera in place. The brackets for the camera and the 2 boards would all be screwed into place.

## V. Drone Detection Software

When initially researching this project we realized that this project would have a significant software component. Not just the machine learning but also the scripts required to cameras on the Jetson, scripts to turn on fans as well as others that also facilitated the successful completion of the project. This section will go into detail on all the different software applications we have used to implement our design.

### A. Machine Learning

The production of our machine learning algorithm was the most important portion of the project. We chose to use the YOLO(You Only Look Once) model for our algorithm. YOLO was chosen as it has one of the fastest detection rates when compared to models of similar application.[5] Yolo uses a regression process by comparing each frame to the previous and deleting any pieces that are common to each to detect objects and then comparing those objects with the shapes of known objects to identify the object. Teaching the model how to identify drones in a picture required multiple epochs. An epoch is a generation of machine learning that has learned to detect objects based on the parameters set, in our case drones. Each generation learns to detect objects better than the previous iteration in basic terms thus the more epochs the better the final model will be. Due to this we have trained 300 epochs of the model to try to maximize the models ability in detection and limiting false detections as well.

We trained this model using a premade dataset. We have combined two dataset , one that focused on drones only and other one focused on flying objects at a distance which includes drones, helicopter, airplanes and birds.[6] The second dataset was essential in our training as it focused on object at a distance and more importantly dealing with the false positive detection detected by the AI on helicopters, birds and planes.

The first dataset consisted of drones only and was created by collecting Youtube videos and labeling them. This gives a pictures of drone of multiple shapes and size, different environments and different camera resolutions,

However the second dataset consisted of drones, helicopters, birds and airplanes. It was created, captured and labeled instead of combining a set of videos. Thus giving us a more controlled dataset.

Alongside that we had high resolution images of landscapes that did not include any of the objects we are trying to detect. It includes images of different environments, weather and time. The original purpose of that dataset was to train models for landscape generation.

Overall we had 180K+ images to train with, 80% of that dataset was used for training and 20% is used for validation. The model was trained on the google colab utilizing 25GB of GPU , 60 GB of RAM and the total training time was about 96 hours.

Testing of the model has occurred in several environments to make sure that the model works consistently. We first tested it by inputting chosen videos that had an object of importance in the frame and made sure the model could detect the object and place a box around it such that a user would understand what the detection was of. After this we tested the model using a variety of cameras to ensure that the model would work within our system. We found that the higher the resolution of the camera the more reliable of detection we could produce.

### B. Jetson Software

The Jetson Nano is a small, powerful computer designed for AI and robotics applications. It runs on the NVIDIA JetPack SDK, software development kit, which includes a suite of software tools and libraries to help developers create AI applications. One component of the jetson software that was preinstalled was DeepStream. This is a streaming analytics toolkit for AI-based video and image analysis It includes pre-trained models for object detection, classification, and tracking. We decided to use deepstream as it was installed with the kernel while also having easy integration to easily stream our machine learning video.

We also had to download several software packages to integrate hardware components into our Jetson Nano. This was needed for both our fan and camera to work properly. The fan which we installed required a software that we found readily available from github called Jetson-Fan-Ctl.[7] This software allows for the integration of the fan thus the fan starts on startup and we can control the speed of the fan through this integrated software.

In designing our object detection and streaming software, there was a set of constraints we needed to keep in mind. First, our machine learning model needed to be able to utilize the 4 GB of gpu available on our Jetson Nano. In order to accomplish that, we used Pytorch and Torchvision and downloaded the appropriate CUDA to be able to run the model on the GPU.

Second, one of our main features is the ability to live stream from the Jetson Nano to our website. Thus we had to output a RTSP stream to ensure a real time view of our system. The RTSP stream was captured by our server and converted into an HLS stream so we can view it on a website. This was to ensure one of our other features which is easy accessibility of the stream from any device.

One of our other features we have is, saving recordings of detections done by our system. This is implemented by a module called Recorder Manager, which is responsible for controlling the recording and and adding the recording to our server and the necessary information regarding the recording. The information include date and time of recording, minimum and maximum precision throughout the recording and the percentage of the video the drone was captured.

## V. DroneX Website

The DroneX website is designed to be the user interface for the device. When a drone incursion is detected by the Jetson Nano we will receive a live video feed of the detection as a secondary way to ensure the detection is indeed a drone. The website will also house all recordings of any past incursions with a time and date stamp to keep a record of all incursions that happen as well as this we also have included the percent of frames that the drone was detected as well as the maximum and minimum confidence present during that period of time. This allows for missed live detections to be verified on later viewing to ensure a detection is not false. The Website also has a view live feed functionality such that a user can view the camera feed without the need for a drone incursion to be present.

## VI. Project modifications and limitations

Over the course of the last two semesters our project has been changed multiple times due to a variety of factors. Originally, the plan was to use a LoRa module to send the livestream of the drone detection to our network and we had written all of our project documentation based on this idea in the first semester. After acquiring and testing the LoRa module we realized the LoRa module was unable to provide a live stream as due to its low power sending of data its bandwidth was also very low only being able to send a few hundred bits at a time. Considering a video stream is in the magnitude of Mega Bytes thus we decided to discard this idea.

We then continued our research and came to the decision to use a cellular module to transmit our livestream video feed. The cellular module chosen was the EC25 Cell module. The most up to date design of our PCB included this cellular module as very late into development we needed to switch boards due to a trace being accidentally damaged during soldering. When desoldering the cell module to place on the new board the cell module was damaged and no longer has functionality. Fortunately we had previously looked into using WiFi for our system and were able to easily install a wireless module (Waveshare AC8625) that is compatible with the Jetson Nano.

Designing a portable and self-sustaining system through the use of a rechargeable battery and solar energy was the original design, but after a lot of testing and discussion it was determined that the system would not be able to run on its own for longer than an hour and a half. This would greatly limit the versatility of the design as a whole as we intended it to be able to monitor longer events such as football games which could take upwards of 4 hours. Thus we decided the most optimal way to increase functionality was to make the switch to a wall outlet connected by an AC/DC adapter wall plug so the system will be able to run continuously without running out of power. With this in mind a large battery could still be installed that has a DC power jack output but it was not in our budget to make the purchase of such a battery for this project.

One big limitation for our Drone Detection System is not being able to detect drones during the night due to poor lighting. The plan was to use a second camera that works on the Infrared spectrum and switch between the color spectrum camera and the IR camera whenever the background got too dark, like at night or even during cloudy weather. To install a good IR camera to accomplish this task would simply cost too much money that we did not have to spend.

Arguably the biggest limitation for this project was in terms of our budget. We did not get the opportunity to be sponsored for this project so each and every penny spent for this project came out of our own pockets. This limited the possibilities for this project in terms of being 100% self sustaining, higher camera resolution, and overall project ideas.

## XX. Conclusion

The purpose of the senior design project DroneX: Drone Detection System was to develop a system that can be placed in a secure location where drone incursions must be detected. This system will detect drones in real-time and live stream the detection to our website. With our overall design consisting of power regulators , an MCU, a Jetson Nano wifi module, and a color spectrum camera. The Jetson Nano houses our machine learning algorithm, and of course our wifi module sends the livestream to our network server and to our website for the consumer's use. We successfully applied our collective engineering knowledge to develop this device, and we worked collaboratively as a team to achieve our goal.

## Acknowledgements

We would like to express our appreciation to Samuel M. Richie and Lei Wei for their valuable feedback and guidance throughout this project. Furthermore, we extend our thanks to the committee for their support during their busy schedules to accept our invitation to be on our committee, and to the University of Central Florida for molding us into the next generation of engineers.

In addition, we would like to acknowledge and thank the UCF Engineering Department for equipping us with the necessary tools to succeed. The Senior Design Lab was particularly instrumental in the success of our project as we had spent several days and nights working in the lab. We intend to take the multitude of lessons we have learned from this experience and apply them to our future roles as engineers.

## References

[1] “Jetson Nano Developer Kit.” *NVIDIA Developer*, 28 Sept. 2022, https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[2] *Power Designer*, https://webench.ti.com/power-designer/switching-regulator?powerSupply=0.

[3] *Q TPS62903-Q1 TPS62903-Q1 3-V To 18-V, 3-A, Automotive Low I ... - Ti.com*. https://www.ti.com/lit/ds/symlink/tps62903-q1.pdf?ts=1654349224493.

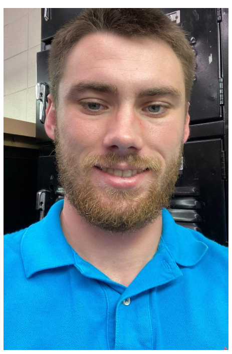
[4] *TPS56624x 3-V to 16-V Input Voltage, 6-A Synchronous Buck Converter in a SOT-563 Package 1 Features*.

[5] Redmon, Joseph, et al. "You Only Look Once: Unified, Real-Time Object Detection." *ArXiv.org*, 9 May 2016, https://arxiv.org/abs/1506.02640.

[6]DroneDetectionThesis. "Dronedetectionthesis/Drone-Detection-Dataset: Dataset Containing IR, Visible and Audio Data to Be Used to Train Drone Detection Systems." *GitHub*, https://github.com/DroneDetectionThesis/Drone-detection-dataset.

[7] Pyrestone. "Pyrestone/Jetson-Fan-CTL: Automagic Fan Control for the Nvidia Jetson Nano." *GitHub*, https://github.com/Pyrestone/jetson-fan-ctl.

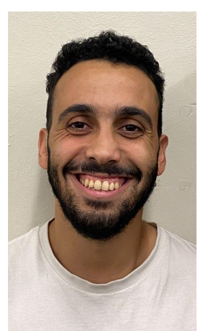
## Biographies



Harrison Kennedy is an Electrical Engineering student at the University of Central Florida planning his graduation for the spring of 2023. After graduation he plans to intensify his job search as well as take a part time position soldering in a warehouse in the Orlando area.

Nick King is an Electrical Engineering student at University of Central Florida graduating spring 2023. He has accepted a position as an associate electrical engineer at L3Harris Technologies in Palm Bay, Florida and will begin immediately following graduation.

Stephan Saturne is an Electrical Engineering student at the University of Central Florida, he will be graduating in May 2023. After graduation, He intends to continue his studying by going to graduate school at the University of Central Florida.



Youssef Barsoom is a Computer Engineering student at the University of Central Florida planning his graduation for the spring of 2023. After graduation he plans to continue his job search for a permitant position as a computer engineer for a company.