Capstone Design Abstracts

December 2016

Capstone NAO

Team Members: Anthony Cicchetti, Adam Slattery, Sean Andrews, David Stewart, Matthew Henderson

Advisor: Professor Charles DiMarzio

Abstract

The Capstone NAO team, over the course of the past several months, has created a code viewer and editor for students utilizing the Aldebaran Softbank Robotics NAO platform. This viewer and editor allows middle and high school aged students who have no experience with code or programming concepts to easily and intuitively create and run their own programs on the NAO robot or its various simulators. Oftentimes during programming education, the feedback students receive from their programs is simply the output of a terminal emulator or IDE standard output viewer. This form of output does not offer students much insight into what operations their code is performing, and what insight it does offer is not visually engaging. The team recognized this problem and addressed it by utilizing two open source development tools, Aldebaran's NAOqi SDK and Google for Education's Blockly visual code platform, to create an engaging and accessible means for new students to learn basic programming concepts such as loops, conditionals, and function creation. Students and instructors also have the ability to extend their programming education by creating new blocks and functions of their own for advanced functionality.

This program, by transforming the aforementioned programming concepts into a visual block syntax, allows students to use concepts such as if-blocks and for-loops significantly more easily, while also giving them much more hands-on experience interacting with the robot. This paradigm is a far more intuitive way for students to learn the process of design and development in a straightforward and easily understood environment, as opposed to far more complex IDEs which would otherwise be used in the classroom setting. This program also provides a simple way for instructors to create additional operations and seamlessly integrate them into the editor. The stock operations provided by the Capstone NAO team were carefully selected to allow maximum functionality for students to work and interact with, while providing a simple way for any instructors or enterprising students to build upon them in a constructive and interactive way.

This project also prompted the team to create example lesson plans to assist instructors in using the system to accomplish their individual goals. These lesson plans start with simple instructions and guiding every step of the way, for example by having a student move the robot or say "Hello World", and gradually transition into using previous knowledge to have students do more advanced tasks. Later into the coursework, students are introduced to concepts such as looping, by using a for-loop to repeat an action instead of using the same blocks multiple times, and conditionals, such as using an if-block to make sure the robot completed its tasks successfully before executing another one. This natural progression from simple tasks to more complex ones prepares students for future programming courses where they will not have the luxury of such a simple interface. It also gives them an opportunity to continue tinkering and finding the limits of their curiosity.

Spinostim: Data Acquisition and Test Device for Back Pain Treatment using a High Electrode Density Paddle

Team Members: Colleen Finnegan, Michael Hoang, Kenneth Staley Jr, Eric Su, Alexander Teixeira, Liane Wong

Advisor: Professor John Kimani

Abstract

Spinostim is an implantable medical device designed to treat chronic lower back pain. Spinal cord stimulation (SCS) works to eliminate pain by applying an electrical current to nervous system via the spine. The high electrode density paddle is implanted into the epidural space of the spinal column and makes contact with the dura. Spinostim includes three components: a graphical user interface, a printed circuit board (PCB), and a high electrode density paddle.

The GUI is the interface that clients will use to configure the paddle. We have designed the GUI to represent the layout of the electrodes on the paddle, where every electrode has a respective button that can set the electrode as a stimulator or receiver. In addition, the GUI allows the configuration of up to 8 groups, to allow users to configure multiple uses for the paddle. Additional groups are optional and can be activated, de-activated, and edited at will. Each group has configurable settings, including the amplitude, pulse width, pulse duration, etc. These settings are all saved locally when users switch between groups and all groups can be reset to give the user a clean slate. Finally, when a user is ready to use a group, they can choose to implement charge balancing before running the configuration on the paddle.

The PCB is the physical interface between the GUI and the paddle. It consists of a power block, a micro-controller, a current stimulator, and a selector chip. The power block is made of two 9-volt batteries, a DCDC converter, and three linear regulators. The micro-controller, TI's CC2650, is used to control all of the communication between the GUI, the current stimulator, and the selector chip. It receives commands from the GUI via serial and communicates with both the current stimulator and the selector chip via SPI. The current stimulator sends specific currents through the selector chip to the paddle. The selector chip simply acts as an amalgamation of MUXs that create the paths for the current to travel from the stimulator to each selected electrode.

The high electrode density paddle has 48 stimulating electrode pads. It will be built using a Micro-Leads proprietary manufacturing process. The paddle will be implanted within the epidural space of the spinal column and it conforms to the shape of the spine. Each electrode receives the generated waveform via wires extending from the selector chip outputs. The stimulating ends of the electrodes, farthest from the selector chip, make contact with the spine and deliver the therapeutic waveforms. The ground electrode(s), selected within the GUI, completes the circuit and receives the waveform to send it back to the PCB.

High Dimensionality Synchronous Data Acquisition (DAQuLA)

Team Members: David Butler, Lauria Clarke, Dominic Harkness, Peter MacLellan, Tyler McCarthy, Marie Mitchell

Advisor: Professor Bahram Shafai
Sponsor: Gunar Schirner, StreetScan

Abstract

The DAQuLA team has designed and built a high throughput synchronous data acquisition system. The system has a vast range of applications including detecting foreign objects in manholes, mapping underground utilities, and profiling road conditions and hazards. Outdoor open event security currently relies on police dogs to detect the presence of explosives. However, the DAQuLA system would allow event security to scan streets many weeks prior to the event, such as a marathon route, and then rescan those streets the night before the event. This enables security to check for any new foreign objects that seem suspicious in a more efficient and reliable manner than currently exists.

The DAQuLA system is capable of simultaneously acquiring data from 32 analog input channels at up to 200 ksps. These input channels are attached to a custom printed circuit board, which contains four 8-channel analog-to-digital converters (ADCs) and a Zynq-7000 SoC. Digital hardware implemented in the programmable logic fabric of the Zynq-7000 is responsible for two main actions: (1) handling the high speed serial peripheral interface (SPI) communication with the ADC chips; (2) aggregating the sampled data and passing it to a direct memory access (DMA) controller, where it gets stored in shared memory accessible to the Linux OS on the Zynq ARM core. From there, the team created a program utilizing a special-purpose kernel driver to pull the raw data out of the DMA memory block and send it to an external machine via a transmission control protocol (TCP) Ethernet connection. On the external machine, an application is responsible for three main functions. First, the application receives and stores all of the incoming data in binary format to files on disk. These files can be converted to a more useful HDF5 format later, as an offline batch process, for application-specific analysis using the included python script. Second, the application provides a live-updating plot of the gathered data on the graphical user interface (GUI). In order to ensure full speed lossless recording of data to files, the plotting pipeline is able to drop samples as needed to maintain speed in the more critical path. Lastly, users are able to configure the sample rate of the ADCs from the GUI or command line, which propagates down through the Linux server to the programmable logic to configure the actual hardware. The user is only able to change this sample rate when connecting and disconnecting from the Linux server, preventing accidental changes during operation.

Testing was completed to ensure that data was able to pass through the entire system, from the ADCs to the output binary files. Wireshark was used to verify the data throughput speed, which confirmed the system was processing at 200 ksps. After testing and verification of operation was complete, the DAQuLA system was handed over to StreetScan for outside testing and implementation.

Classio: An Integrated Classroom Solution

Team Members: Cole Bush, Tyler Paskowski, Jose Ramirez, Jon Smith, Tao Tang

Advisor: Professor Charles DiMarzio

Abstract

The average university lecture structure, while effective, cannot cater to all of the varied learning and teaching styles of its participants. The Classio project aims to bring fluidity and efficiency into the classroom environment through several features that facilitate organization, clear communication of ideas, and flow of information in hopes of creating a more effective learning environment for both students and professors. Classio: An Integrated Classroom Solution takes a modular approach to address some of the shortcomings of the modern classroom as well as the unique needs of any given student or lecturer. This first iteration of the Classio project seeks to address these more common classroom issues with independently functioning features while also laying the groundwork to introduce more niche features in future iterations.

Classio is implemented on a Raspberry Pi with a USB microphone to create a solution that is straightforward and portable for the professor. To keep things simple, Classio just needs to be plugged in and it will start automatically. In the event of any issues with the device, it can be rebooted with a reset button. Otherwise, all features are configured and accessed through the Classio web interface hosted on a Django server at a provided IP address. Three features of Clasio are discussed here: (1) In order to increase the efficiency of the classroom interaction, Classio provides a way for students to go to any slide in the lecture or snap to the current slide and follow along with the presentation. The professor follows simple instructions to obtain a PowerPoint URL which is then embedded into Classio's student interface for guided lecturing. (2) The professor will also be able to utilize the Clarity Rating feature that gives the professor feedback on whether or not they are loud enough for their classrooms. This is accomplished by listening to the lecture and comparing the average levels of the last few seconds with a threshold. (3) In the event there are students who are hard of hearing, Classio also has an audio streaming feature. Students can plug hearing devices into their phone or computer and clearly listen to the live lecture with minimal latency. This is accomplished using the mic input signal and sending it to a sound server integrated with Django. Classio has additional features not described here addressing other common problems seen in the classroom.

The final Classio product is a collection of several features that can be used to enhance the classroom experience and encourage a productive flow of information. In practice, some of the desired features were limited by the processing constraints of the platform. For instance, the Clarity Rating feature could be improved by a more complex sound to noise ratio analysis algorithm using two microphones as inputs. Further iterations would see Classio implemented on a more powerful base platform to allow for the addition of more complex features.

‘Hi’: An In-Ear Minimal Latency Translator

Team Members: Halah Abuateeq, Tammy Huynh, Upendra Naidoo, Menglu Xie, Luxi Zeng, Hao Zhen

Advisor: Professor John Kimani

Abstract

The rise of globalization introduced various language barriers. To address these problems, we designed “Hi,” a wearable in-ear translation device that attempts to work with minimal latency. Our product derives its name from one of the most common greeting words in the world. “Hi” is composed primarily of two parts: the earpiece and a Raspberry Pi, which serves as the processing unit and provides an additional microphone for voice detection from an external source.

The processing unit receives data in two ways: (1) wirelessly through the earpiece and (2) through a 3.5mm wired microphone equipped with automatic gain control. The earpiece communicates with the processing unit via Bluetooth and a 2.4 GHz S band radio frequency. The received audio is then translated to the target language by the Google Translate API, installed on the system, and played through the respective speaker. The processing unit automatically toggles between the two microphones by utilizing a sound sensor on the main unit and a voice recognition chip on the earpiece.

The earpiece consists of an Arduino Uno Board, battery, wireless microphone, earbud, and a voice recognition and Bluetooth chip. The Bluetooth chip communicates with the processing unit with two protocols: (1) A2DP for streaming stereo audio and (2) RFCOMM for serial data. The low-power voice recognition chip can store up to 7 commands, which are used to switch between microphones, enter translation or sleep mode. The commands are transmitted to the processing unit via Bluetooth RFCOMM. These commands could be trained via a button press on the earpiece followed by the designated phrase. The wireless microphone has adjustable gain to optimize the audio quality depending on the ambient noise level of the room.

The processing unit consists of a Raspberry Pi running Ubuntu Mate, sound and SD card, battery, AGC microphone, wireless microphone receiver, sound sensor, speaker, and an analog multiplexer. The translation program converts speech to text and passes the converted text to the Google Translate API, which supports 64 languages, and synthesizes the result back to speech in Python 3. To provide minimal-latency, we used Google Cloud services for speech recognition and translation. As a way to provide accurate speech detection, a Python function was added to listen for ambient noise from the microphones for 1 second and sets the baseline for noise. The multiplexer was used to switch between the 2 microphones as well as an external microphone port. Finally, the selection signal for the multiplexer is determined by the sound sensor on the processing unit and the voice recognition chip.

Air Pollution Monitoring and Visualization with AirMap

Team Members: Taylor Skilling, Will Johnson, Nick Tosta, Chris Kenyon, and Brandon Nguyen

Advisor: Professor Bahram Shafai

Abstract

The AirMap team has designed and produced a handheld system for accurate, geo-located monitoring and reporting of air pollutants and atmospheric conditions. A functional website allows anyone with an Internet connection to view data collected by AirMap devices and view air quality levels by region and over time. For the researcher in need of a standalone device capable of collecting time-stamped, GPS tagged, fine particulate data for hours on a single charge or the concerned expecting mother curious about the quality of air in her neighborhood, AirMap provides an easy-to-use, handheld platform. While air quality sensors exist on the market today, none of the solutions are cost effective in large numbers, are capable of collecting geo-located data, or include a clear and accessible data visualization platform.

As an end-user, deploying the AirMap device is extremely simple. Three mechanical switches determine the power and sampling control logic and whether or not OLED display is active. Underneath the opaque black polycarbonate exterior lays a custom printed circuit board and a variety of sensors and electronic modules. A Raspberry Pi Zero microcontroller, running a distribution of Linux, aggregates data from each sensor, writes to a local MongoDB database, and transmits each sample via a 2G cellular network to the web server backend. Each sample contains a precise time-stamped GPS latitude, longitude, and altitude, along with temperature, relative humidity, and pressure and a count of particulate matter fewer than 2.5 micrometers in diameter (PM 2.5) and Ozone. The onboard software contains provisions for error handling and logging, allowing the user to continue sampling without a cellular connection by simply saving data only to the local database. The web front end, accessed via http://airmap.tosta.io/, allows the user to view not only data collected via AirMap devices around the United States, but existing EPA data as well. Further, the website provides users with site and device registration, making it easy to view data from only the devices you own.

Extensive testing in a controlled lab environment ensured that AirMap devices continue to accurately log atmospheric data in cellular denied environments and purge data without geo-location. Multiple devices were used in Boston, Massachusetts to collect data at various locations and during different periods of the day. Data collected from the device was compared to existing EPA data and visualized on the deployed website. Large matrices of data containing information about atmospheric and pollutions condition varying over time and location can be formed through the AirMap community. Using singular valued decomposition (SVD) or principle component analysis (PCA), data trends can be discerned from these matrices, providing valuable information about which factors have the most influence on air pollution concentrations, informing public policy, healthcare, and efforts to reduce air pollution.

Microwave Tomography (MWT) Scanner

Team Members: Emma Bobola, Joshua Jameson, Erica Penniman, Janna Shaftan, Matthew Tivnan

Advisor: Professor Charles DiMarzio

Abstract

Breast cancer is the second most common form of cancer-related death in women world-wide after lung cancer. Of women who are diagnosed with breast cancer, nearly 1 in 5 will die from it. Early detection is crucial to successful treatment and current methods such as mammography have between 6% to 46% false negative rates leading to delayed diagnosis and treatment. Improved accuracy is needed. Preliminary evidence has shown that the hybridization of x-ray and microwave tomography (MWT) is a promising venture to produce high-contrast high-resolution images which reduce false-positives and false-negatives present in conventional mammography.

The aim of this project is to design and build a functional prototype MWT scanner that would physically verify computer simulations indicating that microwave tomography can optimize imaging of human tissue. After penetrating a medium in the viewing chamber with microwaves, the amplitude and phase measurements of the EM field are collected to reconstruct the composition of the tissue. Denser objects, like cancerous tissue, appear in higher contrast. A system of twelve monopole antennas mounted on moving platforms and submerged in water will be used to perform a scan. Each of the twelve antennas can act as both a transmitter and receiver in turn. All possible transmitter-receiver pairs will be used once during a scan. The measurements collected will be processed in MATLAB to reconstruct the images using the back-projection method. This method works by summing the magnitude-weighted lines of response to show the location of an object at the point of highest signal magnitude. Data from our preliminary imaging experiments has been used to successfully reconstruct images in this way.

EyeoT: Context-Aware Gaze Interface to Assistive Devices for Persons with Limited Fine Motor Control

Team Members: Jenna Czeck, Harrison Dimmig, Filmon Elias, Gen Ohta, Francesca Sally, Eric Tseng

Advisor: Professor John Kimani

Abstract

The group implemented a natural eye-tracking solution to acquire a user's gaze, enabling a person with limited fine motor control to accurately and wirelessly control assistive devices, improving their quality of life by increasing independence in known environments. Because patients with neurodegenerative diseases typically maintain a high degree of eye control throughout their illness, we view a gaze-controlled assistive device as an impactful, long-term method of helping millions in need.

An existing eye tracking solution's (Pupil Labs') hardware and SDK is used for data acquisition. The Pupil transmits three streams of data over TCP: gaze coordinates, pupil measurements, and worldview frames. Together, these streams are used to determine objects of interest and controls for the system by filtering and setting parameters around the data.

A custom 3D-printed chassis was designed for the simplest binary state device: a light switch. A servomotor brace was specifically engineered to keep the motor sufficiently secure to prevent its torque from causing it to fall out of alignment. This turnkey solution also provided space for all the circuitry and LED indicator strips in addition to a secure mount for a Bluetooth® Smart enabled microcontroller. The computer processing the gaze data serves as the master to any number of slave Bluetooth® assistive devices. To determine what object a user wants to control, the slaves' LED indicators are illuminated with a sufficient number of different colors given the total number of connected slaves. The master, in turn, will determine which device to send commands to based upon our machine learning (gaze control) algorithm.

The system stays in an idle mode for most of the time, waiting for user input to wake the system and start control of devices. Using the calculated confidence of pupil measurements – which approaches zero when the user blinks – we are able to discern a long, controlled blink which is then used as both a back control and a way to wake the system from its idle mode. Once in active mode, the system determines the object of interest that the user wants to control. By projecting gaze fixation points onto the worldview frames, the system intelligently crops a subset of the image down to a region of perceived interest. Next, a k-means++ algorithm bifurcates the remaining pixels into two colored regions: background and the potential object. Following further filtering, a final bounding box which encompasses the object is found, and the average color of that box is determined. If the determined color matches the illuminated object, we determine that object to be the one to control. An audio prompt states the selected device and the user can go back if it is not their intended device. Once we have determined the object to control, the user can deterministically move their eyes to control the device.

The implemented system can be expanded to further future devices to allow a user to have independent control over their environment.

Autonomous Object Tracking UAV

Team Members: Samuel Lindemer, Oliver Scott, Nathan Winn, Robert Peterson, Robert Lebel, Robert Daigle

Advisor: Professor Bahram Shafai

Abstract

The Autonomous Object Tracking UAV team successfully designed, built, and tested an unmanned aerial vehicle (UAV, or drone) capable of identifying a target object and following it without the need for manual controls. The tracking capabilities are ideal for leader-follower or pursuer-evader scenarios. This technology is applicable in areas such as security, surveillance, commercial, and military. The team's design uses a custom-built drone with autonomous flight achieved with a Raspberry Pi single-board computer mounted on the drone's frame.

The design of the drone takes advantage of off-the-shelf and open-source components. The target's position and size data are determined by a single Pixy CMUcam5 camera. This data is sent to the Raspberry Pi over a serial connection where it is further processed. The Raspberry Pi employs PID controllers on the four axes (yaw, pitch, roll, thrust) to calculate the command values necessary to track the target at a fixed distance. These commands are then sent to the flight controller over a serial connection. At any time, the operator can issue a command to end the tracking process, allowing the drone to descend. The Raspberry Pi was configured as a WiFi access point, allowing for connection using any WiFi enabled device connecting with SSH.

Testing was performed using both fixed and moving targets. Commands to start and stop the drone were given to the drone from a laptop computer. The drone was successful in autonomously tracking a target in all directions and away from it. The team believes that this tracking drone has the potential to be implemented in many diverse applications. Of particular interest would be the addition of a mechanism for capturing an unfriendly drone. Furthermore, one can increase the number of drones with the same capabilities and employ them in cooperative control of multi-agent systems to reach consensus and perform the desired tasks as required in the aforementioned applications.

High Altitude Remote Aircraft Measuring Botanical Energy

Team Members: Evan Hovaniak, Gregory Cusano, Joseph Grasso, Michael Bibinski, Michael Youngstrom, Joseph Beaudreault

Advisor: Professor Charles Dimarzio

Abstract

Our team has built a method for the proper analysis of vegetation, that through the use of a specifically filtered camera and specialized data analysis software gives users detailed information about the health of the plant in question. The ideal use of the system would be combined with a small, autonomous aircraft to fly over large swaths of farmland and relay photographs of the crop to a ground station, where such analysis performed and give the user a sense of how the crop is progressing throughout the growing season. This will provide a quick, efficient and cost effective way to realize which areas of the farm need more attention as harvest season approaches, and can be used season after season with proper maintenance of the aircraft.

The use of the aircraft relies heavily on the Pixhawk autopilot module and the Raspberry Pi on-board processor and camera. The Pixhawk guides the aircraft through the air on a path of navigation set by the user preflight, while the Raspberry Pi takes pictures of the area below. The pictures taken from the Pi are sent back to the ground control station using a serial bus and over a 915 MHz channel used by one antenna in the plane. The user can take manual control of the plane by talking to the Pixhawk over the 433MHz antenna installed in the interior of the plane if redirection is desired.

The testing and validation of the HARAMBE project was approached from two angles, one focus being on the aircraft and the other being on the pictures and working with the Raspberry Pi to fully implement our method of image analysis. The testing of the image analysis yielded strong results that ensures that the filters and indexes used will give a proper indication of the crop health and can distinguish vegetation versus non vegetation. The data transmission and communication also yielded fair results. The team was able to establish communication both to steer and control the plane whilst airborne through the Pixhawk. We were also able transmit the data from the raspberry pi camera, through the pi board and antenna before finally arriving at the user on the ground.The first iteration of the aircraft was built using a fuselage that was purchased and then reinforced using a fiberglass and epoxy combination to give it a hard outer shell to protect the Styrofoam. After trials with this iteration of the plane, it was clear that weather issues and the sensitivity of the RC glider to wind and manual control made results with the flight unsatisfactory. In the opinion of the team, a more rigid plane that can withstand weather and minor crashes more routinely needs to be used in any advancement of this part of the project or for any real product that will be based on this proposal.

Total Knee Replacement Therex

Team Members: Ryan Doherty, Gurtaj Khatra, Kenny Krug, Izaak Branch, and Jay Chinnaswamy

Advisor: Professor John Kimani

Abstract

The purpose of our project was to create a product that will help in the rehabilitation process for patients that have undergone total knee replacement surgery. This product will allow patients to recover and strengthen key muscles and ligaments with only limited assistance from traditional rehab aides. Furthermore, this product will limit frequency of travel to facilities, saving both time and money for the care providers and patients. To accomplish this, we utilized a Microsoft Kinect to track a user as they perform a variety of exercises commonly completed at rehabilitation centers.

The body-positioning component of the Kinect API will allow us to track the patient's motion, while an avatar that we create will provide a visual representation of the necessary exercises. The user's motion data is streamed back to our application running on a local machine where it is compared to baseline model. Our interactive system makes it easy and seamless to select either an individual or a collection of exercises and edit details such as the number of repetitions. Following the completion of each exercise set, a number of test statistics will be displayed. These statistics will provide overviews for the user's performance relative to a baseline. The user can log these results and use them to track performance increases over the course of the rehabilitation period.

Monarc: Autonomous UAV with Stereovision Obstacle Avoidance

Team Members: Nigil Lee, Nathan VanBenschoten, Benjamin Soper, Justin Knichel, Chris Canal, Nate Casale

Advisor: Professor Bahram Shafai

Abstract

As the drone industry evolves, it's becoming clear that designing and building a machine from the ground up for every unique task is both economically and temporally inefficient. A need has arisen for a standard base model which can readily be modified to serve different roles. This is the mission of Monarc – provide foundational functionality while leaving ample room for post-purchase customization.

The Monarc team designed an autonomous quadcopter UAV equipped with stereoscopic vision, and ultrasonic sensors - all working together to provide autonomous flight and collision avoidance. In theory, the drone will be able to pilot itself from point A to point B using just GPS coordinates and onboard sensor data. Using its array of sensors, it will then be able to complete complex tasks such as identifying a person of interest, delivering a package, or escorting a group of people to safety. The components of Monarc are lightweight and have been chosen specifically to allow for high levels of accuracy and performance. The powerful motors provide enough lift to get the copter off the ground with thrust to spare; thrust that can be used for additional equipment necessary for more complex objectives. A well-conceived system architecture ensures that tasks are executed with precision and consistency. On board stereoscopic cameras help Monarc identify its surroundings and feed the collision detection system with data necessary to keep the copter airborne without injury. All these facets of Monarc's design come together to create a well-tuned, consistent and feature filled aerial drone that users can easily customize to accomplish whatever task the user commands.

The system is built around a custom printed circuit board we designed for avionics. This board integrates all of our primary system including: Nvidia T30 Computer, Flight controller, and an NXP LPC1759 microcontroller. To achieve flight, Monarc needs reliable communication between these three systems. The navigation computer makes flight decisions based on inputs from a stereoscopic camera, IMU, GPS, ultrasonic sensor, and adc telemetry. The system controller provides quick communication, relaying sensor information to the navigation computer and flight controls to the flight computer. The flight controller translates the flight commands that it receives into inputs for each of the motors. These three systems are connected with our custom designed circuit boards.

In its current state, Monarc is a UAV that is capable of user controlled flight through a remote controller or with limited autonomy. We have currently tested autonomous takeoff and landing. The software for more extensive autonomous flight has been implemented, but has not gone under significant tuning or testing.