MSU Unmanned Systems 2016

Journal Paper for AUVSI-SUAS Competition

Michigan State University
College of Engineering

Figure 1: Ventus UAS

Abstract

This paper provides a comprehensive summary of MSU Unmanned System's Ventus UAS, designed to meet the mission requirements set forth by the AUVSI Student Unmanned Aerial Systems competition. Ventus is a UAS system comprising of a modified COTS airframe and a custom built GCS (Ground Control Station) designed by the team. The UAS utilizes the Pixhawk, an open-source flight control board, as the autopilot. Equipped with the autopilot and a gimballed optronic system, Ventus provides the user the ability to run reconnaissance missions on pre-defined flight paths, with the option of dynamic mid-flight re-tasking. The images captured are transmitted over a 2.4 GHz digital data link, while the live video feed is streamed to a remote video terminal over a 5.8 GHz analog downlink. The digital images are then autonomously processed by a computer to extract target data, which is then used for actionable intelligence. A modular design approach permits rapid deployment of the UAS system, while making future upgrades and on-field payload replacements straightforward. Ventus can also be controlled manually via a 433 MHz FHSS radio control system or autonomously over a 915 MHz dedicated data link. This journal also details the method of flight operations and safety guidelines employed by the team, and concludes with a description of the testing and evaluation that has been performed to show the efficacy of the vehicle in achieving mission objectives.

COLLEGE OF ENGINEERING

Table of Contents

1 Systems Engineering Approach

1.1 Mission Requirement Analysis

The AUVSI SUAS event is designed to subject the UAS to simulated real-life scenarios. Some of the tasks outlined in the rules document are:

- Providing real-time surveillance and reconnaissance to ground personnel for various applications.
- Receiving encoded messages from the SRIC (Simulated Remote Information Center).
- Potentially using thermal imaging systems to scan an Area of Interest.
- Delivering a package to personnel in need.

Barring the basic requirements, the UAS has to operate under an outlined framework with respect to navigational abilities:

- The vehicle must be able to execute missions autonomously
- Remain within a geographic boundary, or out of a “No-fly-zone”.
- Scan specific Areas of Interest while flying autonomously
- Having safeguards to ensure that vehicle lands safely in case of loss of communication.

Table 1: Mission Tasks and Achievements

Mission Task	Mission Sub-task	Status
Autonomous Flight	Waypoint Navigation	Accomplished
	Auto Takeoff and Landing	Accomplished
	GCS Display	Accomplished
Area Search	Localization, Classification	Accomplished
	Autonomous Search	Accomplished
ADLC	Localization, Classification	Accomplished
	F.A.R.	Accomplished
Actionable Intelligence	-	Accomplished
Off-axis Target	Imagery	Accomplished
	Classification	Accomplished
	Payload Autonomy	Accomplished
Emergent Target	In-flight Re-tasking	Did Not Attempt
	Autonomous Search	Accomplished
	Target Identification	Accomplished
SRIC	Download	Did Not Attempt
	Upload	Did Not Attempt
	Autonomous	Did Not Attempt
Airdrop	Release	Did Not Attempt
	Bull’s Eye Target	Did Not Attempt
Interoperability	Download and Display Obstacles and Server Time	Partially Accomplished
	Upload UAS Position	Accomplished
Sense, Detect, and Avoid	-	Did Not Attempt

1.2 Design Rationale

The design process utilized by the team was largely based on the experience gained by the team while competing at the SAE Aero Design 2014. The design was divided into three main stages:

- Initial Design
- Prototyping and Design Validation
- System Integration and Testing

In the Initial Design phase, the team evaluated the different types of aircraft available, based on the mission objectives and potential problems that could be faced during the validation/testing phase. Since the majority of the work season was during the winter, and MSU ordinances required procuring a COA from the FAA for on-campus operations/testing, the team decided against a fixed wing aircraft and instead, went with a multi-rotor aircraft. This decision was made because multi-rotors did not need as much space for testing as a fixed-wing aircraft did, and this provided the team the ability to test off-campus at smaller fields. The lack of an experienced fixed-wing RC pilot was a secondary reason behind the decision. Following this, it was determined that buying a COTS (Commercial off-the-shelf) airframe was the best way to proceed, keeping in mind the team's financial constraints. Based on some preliminary calculations, designing and building a prototype would have proven to be an economically poor decision.

The Prototyping and Design Validation phase involved using the actual propulsion system, along with the most cost-effective components and prototyping a flightworthy aircraft. This aircraft was then tested for endurance and the MTOW (Maximum Takeoff Weight), yielding values very close to the calculated capacity. This stage further involved selection of the payloads required for secondary (non-navigation) tasks and developing and fabricating a power harness/power distribution system. This approach allowed the team to test all the subsystems individually, thus proving the dependability of the subsystems.

The Systems Integration and Testing phase involved integrating the subsystems on the flight-tested aircraft, which were then tested in-flight. No further alterations were necessary as the problems encountered in the previous phase were rectified.

1.3 Expected Task Performance

Ventus was the fourth fully-autonomous vehicle developed by the team, and implemented considerable design changes compared to previously built vehicles. Keeping in mind the team's heavy emphasis on modularity, the only consistency was the concept of a LRU (Line Replaceable Unit). This was done to ensure that the team could swiftly have a back-up vehicle in case something went wrong with the primary aircraft. A new quadrotor design, coupled with a completely new propulsion system, was expected to meet mission objectives of flight endurance and payload capacity. The ADLC/target detection algorithm was the first time the team attempted such a task, and was expecting to meet only the threshold requirements. To enable maximum efficacy, the team decided to use a 3-axis gimbal (rather than a 2-axis) to allow the remote video terminal operators to point the gimbal in any required direction, independent of the orientation of the copter. Although an air-drop module was developed and tested, the team decided not to proceed with the task because of the power/endurance opportunity cost that was induced by carrying the dead weight of the 8 ounce bottle and dropping mechanism. Considering all of the above, the team expected to accomplish all primary tasks above the threshold level and four of the nine secondary tasks.

1.4 Programmatic Risks and Mitigation Methods.

Table 2: Possible Risks and Mitigation Methods

Risk	Description	Mitigation
Subsystem Failure	GPS, barometer, or other vital system failure.	-The Safety Pilot takes over control of the aircraft in Stabilize Mode.
Loss of Communication	A system glitch or interference from another frequency could disrupt communication between the vehicle and ground control.	-The autopilot is preprogrammed to return to launch position in case of communication failure with the manual transmitter. -The autopilot is programmed to continue with mission in case of loss of communication with the PGCS, as long as the safety pilot has full control.
System Crash During Flight	The vehicle could suffer any number of failures and crash during testing or competition.	- The team maintained a full kit of tools and extra parts for quick repair.
Safety	Many safety factors are present when operating an unmanned aerial vehicle.	- The team followed a thorough pre-flight checklist to ensure safety of the vehicle and crew. - The team utilized safety vests, spotters, and two-way radios to ensure safety of the team, vehicle, and surrounding environment.

2 UAS Design

2.1 Aircraft

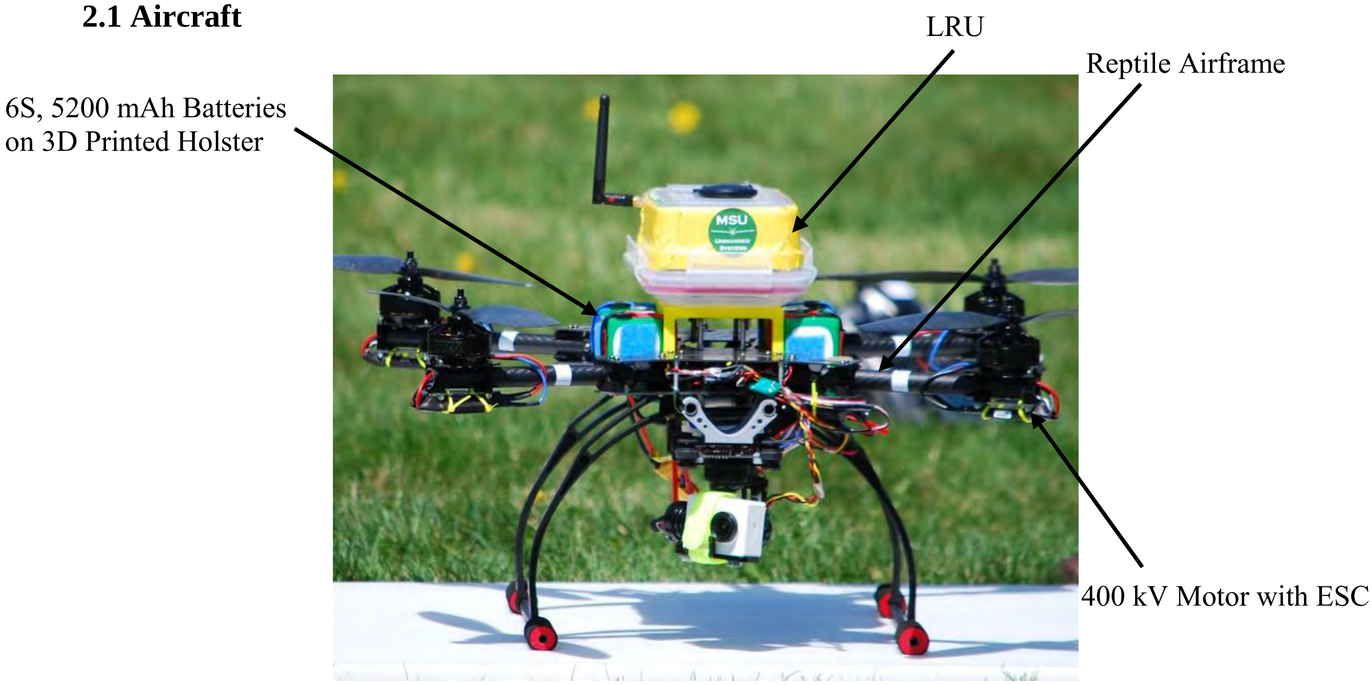

Figure 2: Front View of Ventus on Ground

2.1.1 Airframe

The Ventus frame is a H4 Reptile airframe (*Figure 2*), primarily composed of black fiberglass. This frame was chosen due to the mounting space it offered on the bottom, as well as the size-to-weight ratio. Kongcopter FQ-700, a frame that the team previously worked with, was a potential candidate for the 2016 SUAS entry, but was not chosen due to high shipping costs and lack of available spares. The selected airframe proved to be a very light frame, thus giving the user the ability to incorporate various types of power systems (up to 6 cell LiPo) and upgrade the autopilot/avionics as required.

2.1.2 Battery Holster

One drawback of the frame was the lack of a sturdy battery mounting space due to the thin fuselage profile. To alleviate this problem, the team designed and 3D printed a holster (*Figure 2*) that could be placed on the airframe like a saddle and locked into position by screwing into metal stand-offs beneath. The holster features an inverted U with circles on both sides to save weight. Slots are available on both sides for hook-and-loop straps to slide through, thus providing the ability to utilize various types of power systems (up to 6 cell LiPo). The top of the holster features holes that serve as attachment points for the LRU, and holes to access the screws on the frame for maintenance purposes.

2.1.3 LRU (Line Replaceable Unit)

To enable ease of upgrade and ensure a high degree of standardization among the team's vehicles, the concept of a Line Replaceable Unit (*Figure 2*) was utilized. Instead of fabricating a metal LRU, the team made use of watertight plastic containers as the primary housing, as they were cost-effective and lightweight. Since the autopilot used a barometric pressure sensor to determine altitude, a small slot was provided on the top to equalize the air pressure outside and inside the LRU. There were various advantages to the LRU concept, including shielding of the pressure sensor from prop wash (mitigating erroneous readings), basic protection to sensitive electronics against environmental elements, and use of a quick-connect MIL-SPEC connector. As long as all team vehicles used the same connector wiring protocol, this provided the capacity to use the same autopilot system on various vehicles and enabled a field swap of autopilots in less than 5 minutes.

The LRU was located on top of the battery holster, which was installed on top of a carbon-fiber plate. The total distance between the power distribution harness and the autopilot/magnetometer is approximately 10 cm. Due to this stand-off distance and the conductive nature of carbon-fiber, EMF and eddy currents generated have negligible impact on the electronics and do not interfere with navigation.

2.2 Propulsion System

Table 3: Ventus Components and Weight

Component	**Weight (grams)**
Frame, Arms, Landing Gear	960
Motors (4x SunnySky X4110S)	600
ESCs (4x)	105
Propellers (4x)	90
Batteries (2x Turnigy Multistar 5200 mAh)	1290
Battery Holster	120
Line Replaceable Unit (LRU)	200
Payload Mounting Plate-Mission Ready	200
Gimbal	180
Action Camera	75
Wiring	160

Ensuring a 2:1 thrust-to-weight ratio was very important to ensure efficiency of the propulsion system. In order to achieve this ratio and a minimum endurance of 30 minutes, the team utilized the E-calc software to compute the optimal battery, motor and propeller combination. An X8 configuration octocopter was chosen as the initial aircraft type, but was later abandoned due to the high cost of propulsion components required and theoretical endurance being achieved (< 15 minutes) with the batteries that the team intended to use. Instead, the team opted for a simple quadcopter design that fit the team's budget for a single aircraft and offered a theoretical flight endurance of approximately 20 minutes.

The final propulsion system consisted of four brushless, 3-phase, AC motors - each mounted on one arm of the UAS (*Figure 2*). A single motor (400 kV) combined with a 14 x 4.7 inch propeller, theoretically generated a thrust of approximately 1.55 kg at 50% throttle, providing a total thrust of 6.2 kg for hover. With the motor/prop combination described, the UAS was able to achieve a high MTOW, providing ample power for future applications and payload additions. The propulsion system demanded a LiPo capacity of 10,000 mAh. The batteries were selected in a way such that two 5,200 mAh batteries could be located on the sides of the battery holster, or a single 10,000 mAh pack could be slung under the frame, allowing the battery holster to be used as a payload mounting surface.

2.3 Autopilot

In order to execute autonomous missions in a safe and predictable manner, the autopilot was required to achieve certain parameters:

- Easy-to-use and reliable hardware and software.
- Real-time, full-duplex communication capable.
- Fully-autonomous autopilot - should be capable of self-controlled take-off, navigation and landing, while providing the ability to modify the source code, if required.
- Small size factor with low power consumption and ability to monitor vehicle health.
- Cost-effective

Based on the team's previous experience with the PX4 hardware and Mission Planner, the team decided to proceed with the 3DR Pixhawk as the flight control board. The default firmware used by the Pixhawk allows it to meet nearly all navigational objectives set forth by the rules. The autopilot on-board sensor suite (IMU + AHRS + Barometer), coupled with an external compass & GPS, is able to locate itself 3-dimensionally. The flight control schematic system is shown in Figure 3 below.

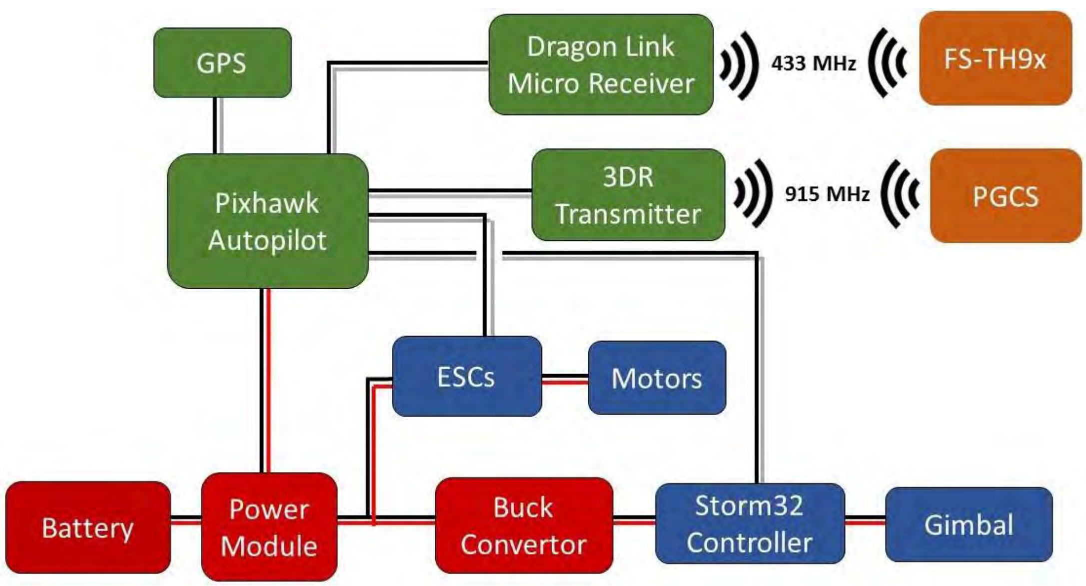

Figure 3: Flight Control Schematic

2.4 Data Links

The carbon-fiber airframe of the Ventus UAS required powerful and reliable communication modules due to the conductive nature of the material and the distance that needed to be flown by the system. Due to the multiple sources of data/communication, the team chose to go with four different frequencies.

1. For manual control and safety override, a 433MHz DragonLink 2 Tx/Rx system is used. On the ground, the DragonLink Tx is interfaced with a FlySky FS-TH9x radio and in the air, a DragonLink Micro Receiver is connected to the autopilot through the PPM input.
2. For flight controller telemetry, two 3DR 900MHz radios were chosen. To compensate for the relatively low power of the radios, a directional antenna was used on the surface, ensuring a consistent signal quality at ranges of up to 2 kilometers.
3. For transfer of imagery data, a high bandwidth link was required since the team's code was to be capable of analyzing image, as well as video feed. To ensure this requirement was met, a 2.4GHz Wi-Fi system was deemed the most cost-effective and easy-to-implement solution. For the aerial end, a USB Wi-Fi adapter with an external antenna was selected and for the surface, an Ubiquiti AirGrid 20 dBi, 2.4GHz grid antenna was used.
4. For a real-time video downlink from the gimballed action camera, an independently powered 600mW, 5.8GHz FPV transmitter was used on the UAS and a RC832 5.8GHz receiver was used on the ground.

2.5 Payload

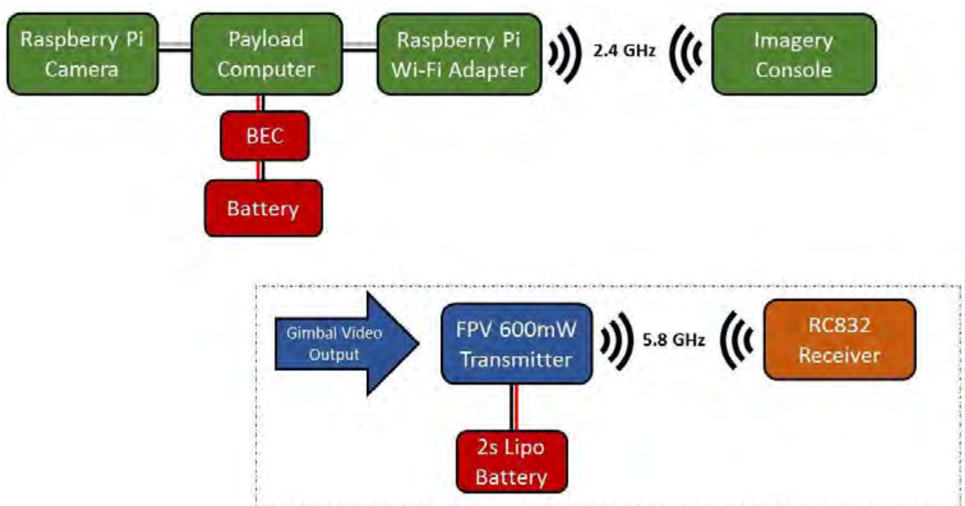

Figure 4: Payload Schematic

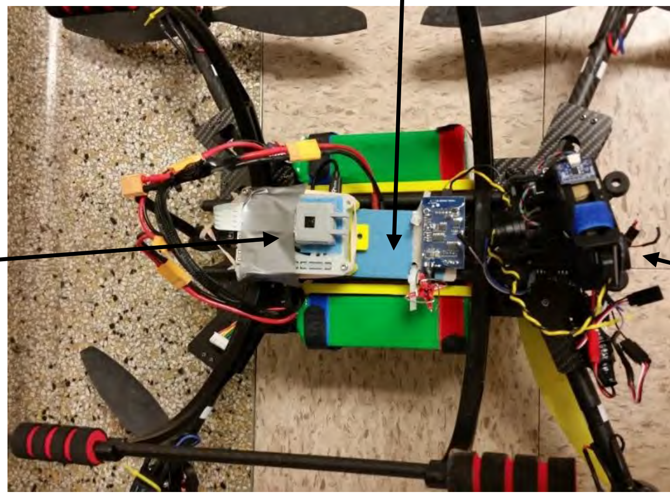

Figure 5: Payload Mounting Plate in Mission Configuration

2.5.1 Optronics/Camera

The Raspberry Pi camera (*Figure 5)* has a maximum still image resolution of 2592x1944 pixels, (~5 megapixels) which was sufficient in testing for the target recognition tasks, while other cameras tested by the team were limited to a smaller 1920x1080 pixel resolution, due to driver compatibility. The camera has a 3.6mm focal length lens with an f/2.9 aperture and a 65 degree field of view. The team chose to utilize still image frames as opposed to analyzing a video feed for target recognition due to the lesser complexity and processing power demand of this method, although the system is capable of taking and analyzing video as well. The team tested the recognition capability on targets of similar specification to the competition targets at ranges identical to those that our system will fly at during competition.

2.5.2 Payload Computer

The payload computer is based on a Raspberry Pi (*Figure 5)*, due to the extensible and configurable nature of the Raspberry Pi platform, its integration with the Raspberry Pi camera, as well as its extensive documentation and support. The approach was not to reinvent the wheel, but instead configure a system using open source and pre-existing hardware and software that fit the team's requirements and performed well with the least amount of modification, while still being open to future customization and changing operational specifications. Due to the unconventional nature of the IO bus on the Raspberry Pi, USB interfaced cameras had less available bandwidth than the first party Raspberry Pi modules that use the specialized camera interface on the Raspberry Pi board (because all USB devices share one common data bus on the Raspberry Pi). Additionally, modules based on the official camera have more software support and are better documented for integration into a python coding environment, which the team used throughout the imaging system. Other cameras lacked drivers that allowed for code-based changes to image settings and resolution, which were required for the team's purposes. For these reasons, the team chose to use the official Raspberry Pi camera as the imaging device.

2.5.3 Gimbal

To enable the user to point the camera at a region of interest - independent of the UAS orientation – the UAS utilized a 3-axis gimbal (*Figure 5*). The gimbal selected was a generic, 3-axis gimbal that used the SToRM32 board for gimbal control. This proved to be a cost-effective solution, since the team was able to manually adjust the pitch and pan of the camera as required, while the gimbal controller kept the camera pointed in the set direction. The 3-axis gimbal provided the UAS with a full field of vision that could be utilized for the emergent target task and the off-axis target acquisition task.

2.5.4 Modular Payload Mounting Plate

The Ventus aircraft used a modular Payload Mounting System located on the bottom of the aircraft frame. It was designed to allow for quick changes in payload and easy replacement in case of a part failure. The system consisted of an aluminum plate coated in hook and loop material (*Figure 5*), allowing anything with hook and loop material pasted to it to be attached to the system. The UAS had a Raspberry Pi 2, a Raspberry Pi camera, a FPV transmitter and a 2S LiPo mounted on the Modular Payload System.

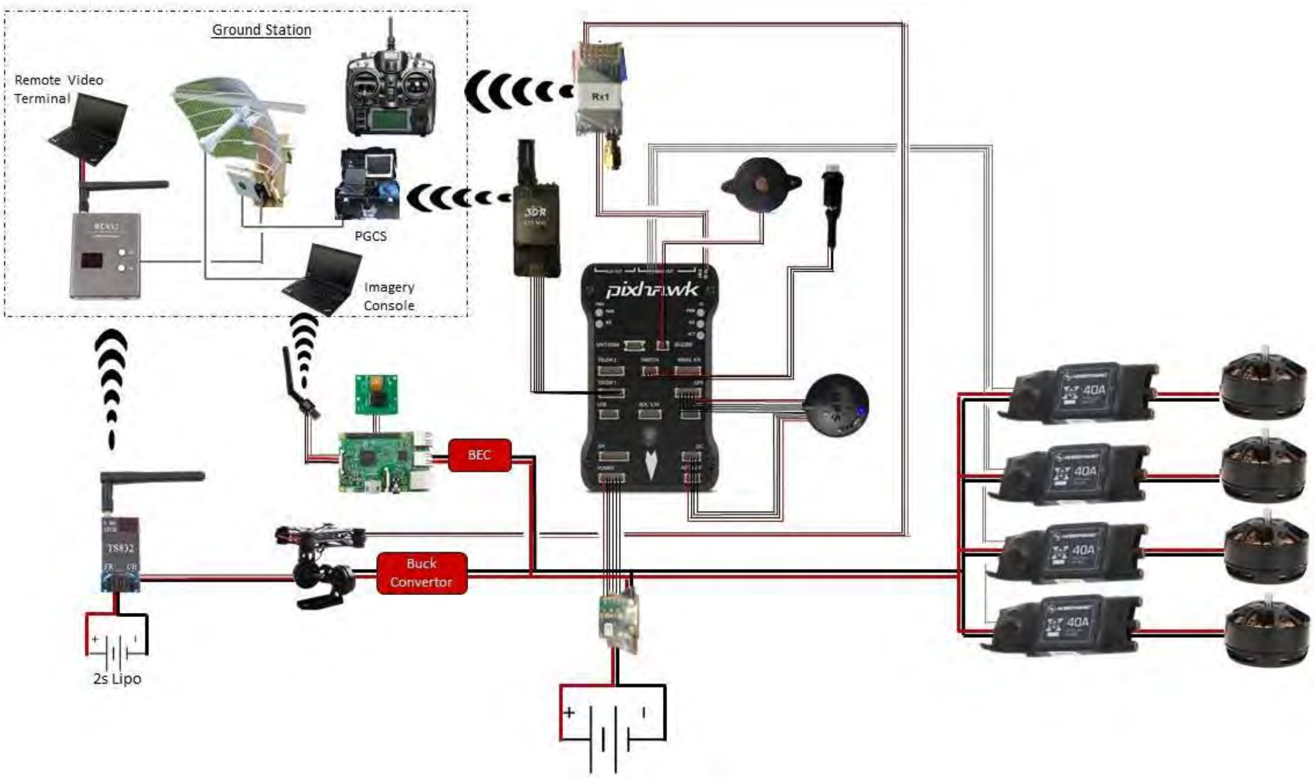

Figure 6: Complete System Architecture

2.6 Ground Control Station

Figure 7: Ground Control Setup

2.6.1 Portable GCS (PGCS)

The PGCS (*Figures 7 and 8*) comprises of a reinforced plastic box containing a laptop, secondary monitor, trackball, 6-channel servo controller and an UPS (Uninterruptible Power Supply). Power distribution for the laptops, antennas, and a provision for an autonomous antenna tracker, is built in to the box. An UPS with two 20 Ah batteries is integrated into the power distribution, thereby allowing the team to operate for up to 20 minutes without power. The PGCS also carries a variety of portable sensors such as a weighing scale, anemometer, non-contact thermometer and additional supporting equipment such as USB extensions, spare telemetry modules, antennas and propellers. The box can be stowed in a folding wagon for easy transport to the flight line.

Figure 8: Portable Ground Control Station

2.6.2 Manual Antenna Tracker (AT)

The antenna tracker (*Figures 7 and 9*) consists of three antennas mounted on a wooden pan-tilt frame, allowing a crew member to manually point it towards the aircraft. A 5.8 GHz pepperbox antenna is used to receive a real-time video feed from the on-board gimballed action camera. The pepperbox antenna was chosen over a helical or patch antenna due to the high-gain (13 dBi) and wide beam width (180 degrees) offered, as the FPV link was the only system other than the Mission Planner PFD (Primary Flight Display) that helped the safety pilot orient himself at long range. For the telemetry relay, a 900MHz patch antenna is employed. The high-gain offered (8 dBi) by the antenna allows it to have a reliable connection at long ranges. For the transfer of digital images from the Payload Computer to the Imagery Console, an Ubiquiti AirGrid 20 dBi, 2.4 GHz grid antenna was installed on the tracker.

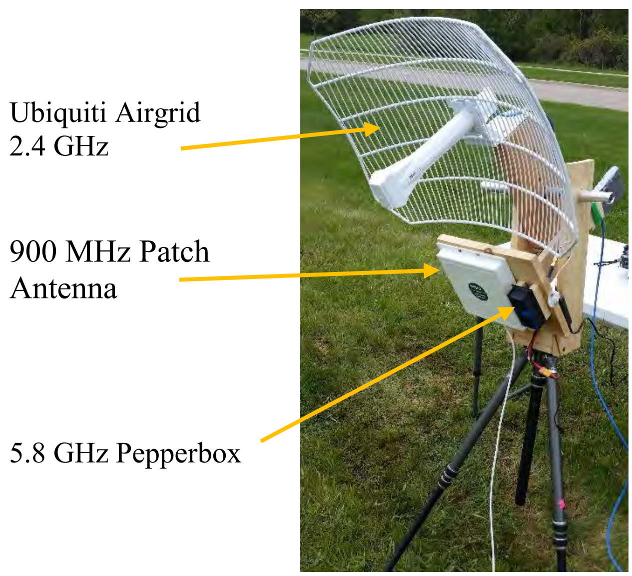

Figure 9: Manual Antenna Tracker

Figure 10: Monitor as Remote Video Terminal

2.6.3 Remote Video Terminal (RVT)

The RVT setup (*Figures 6 and 10*) used by the team consists of one to two laptops or monitors. The serial video output provided by the FPV receiver can be run directly to a stand-alone monitor for simple reconnaissance/aerial photography mission, or can be interfaced with a Linux laptop via a video capture dongle. The team opted to re-purpose an old laptop as the Remote Video Terminal, since it did not require the team to provide for an additional AC outlet needed by a stand-alone monitor. While making the system more portable, this gave the team the capacity to record the FPV stream real-time and take snapshots of an Area of Interest. The RVT operator is provided full control of the 3-axis gimbal. This is done by connecting a Turnigy 9X radio to the trainer port of the FS-TH9x transmitter, and using any of the spare channels to manually control the orientation of the gimbal. In case manual control is not required or not available on-field, the gimbal can be set to point in a specific direction before the mission commences, allowing the RVT operators to view the feed from the UAS. As long as the correct channel is used, multiple RVT systems can be set up, allowing numerous personnel on the ground to view the real-time video feed.

2.6.4 Imagery Console (IC)

The IC system was designed with efficiency and security in mind. The payload and payload computer captured a picture every few seconds. After the picture was captured, it was sent to the Imagery Console over the WiFi data link. During this process, the image recognition code on the IC constantly checked the folder for the next image to analyze. Once the picture arrived in the IC folder, it was analyzed by the code, and the information recovered was displayed. Once this cycle was completed, the IC waited for a new image to be received. This approach provided two major benefits; primarily, in case the UAS was lost due to a mishap, some images could be recovered and analyzed. Secondly, since the image processing was not executed on the Payload Computer, there was minimal power consumption, hence allowing for maximum flight time.

2.7 Image Processing

The image processing code operated on the idea of definite color. The targets on the ground always have a very uniform and bright color, whereas the terrain they rest on may vary drastically in color and may not be very uniform. With this in mind, the code began by looking in the repository folder where the Raspberry Pi stored all of the aerial pictures for a picture titled "0.png". Using C++ and OpenCV, when the picture was located in the folder, it imported it as a matrix to be analyzed. Once it had the matrix, it scanned each pixel for its Blue, Green, and Red values and checked to see if, based off those values, it was a definitive color. This was done in a single if-statement that contained a condition for color. The color red, for example, only has a small band of possible values of Blue, Green, and Red values. If the pixel fell within one of these defined ranges, then it was left alone. However, if it did not, it was changed to be purely black with Blue Green Red values of zero. It then saved these new pixel values in a new matrix as to preserve the original input matrix. The new matrix was then sent through a filtering function that performed several different methods of image filtering to clean up the image and create another new matrix that was a binary image, where all pixels that were not purely black became purely white. This binary matrix was then analyzed to find shapes. In order to find the shapes, the binary image must be scanned for all bounded binary objects based off their contours. The result of this OpenCV function yielded an array of points. These points were the vertices found for the bounded white objects. They were used to find all the features of the shape. Since it is important that all the points found are correct, the points were then sent to a function that validated them. The function began by measuring the distance from each point to the next and saving each segment length in a new array. It then found the longest segment and compared each segment in the array to that length. Since all the possible shapes are very symmetrical and most have equal side lengths, if a very small segment was found, it meant one of the points that segment was made from was incorrect, and eliminated the first one (since the points are so close, it does not matter which is deleted). The function then contained this new array of points that did not contain any points that created invalid segments. Using the new array of points, it then checked the angle at every point in the array by creating a vector from the current point to the previous point and next point in the array and using the Dot Product. Once all angles were calculated, the function checks to see if any were too large, since there could be a point that created two segments that were long enough to pass the segment length test. If a point was too large, it was deleted and the function returns a new array only containing the valid points. After the new points were calculated, the final step was to recalculate the final segment lengths and angles if they were needed, the results of which can be seen below:

Pre Validation Function

Post Validation Function

After the points were validated, the code then used them to calculate the colors of the object. To do this, it calculated the maximum and minimum of both the x and y coordinates of all the points. Once these values are created, it then uses them to create four new points to form a box around the shape on the matrix. It then scanned each point in the box; if the color was purely black, it was ignored. If it had a color, that color's category was increased by one and the next pixel was scanned. The color with the highest count was the shape color and the second highest was the letter color. If however, the letter color was too small, it means the letter may have been black or dark green and was filtered out. This was fixed later in the code during the letter calculation using the fact that if the pixel in the letter was very dark and the blue, green, and red values were all about the same, it was more than likely black, otherwise it was green. Once the colors were calculated, the next feature to look for was the shape. This was done mostly based off how many points the object had; however, occasionally more qualities needed to be examined. In the case of ten points, the object was always a star. However, a shape with four points could be a square, rectangle, diamond, or a trapezoid. In order to further differentiate the shape, the segment lengths and angles that were calculated in the end of the point validation function were looked at and used.

After the shape and colors have been calculated, the final step was to recognize the letter. This was done using the optical character recognizer called "tesseract". In order for tesseract to function, it needed to be passed pictures which contain only the letter being identified, and the letter needed to be oriented correctly. In order to accomplish this level of filtering, first the code will use the box found around the currently identified target and the binary image of the entire picture. If it is not in the box or on the white of the binary image, it is deleted. It then will filter out the target shape by deleting everything that is not the letter color or everything that is the letter shape depending on the letter color. Only the letter remains. Since the letter needs to be oriented upright when the image is pasted to tesseract, and the fact that there is no reliable way to obtain the correct orientation, the image is given to tesseract and the letter found is saved. The image is then rotated a few degrees and checked again, and the letter of that check is stored. The letter found with the most occurrences is considered to be the letter. Once this has been obtained, the code then relays the information to the interop server and displays the information in the console, then looks for the next image to start the process over.

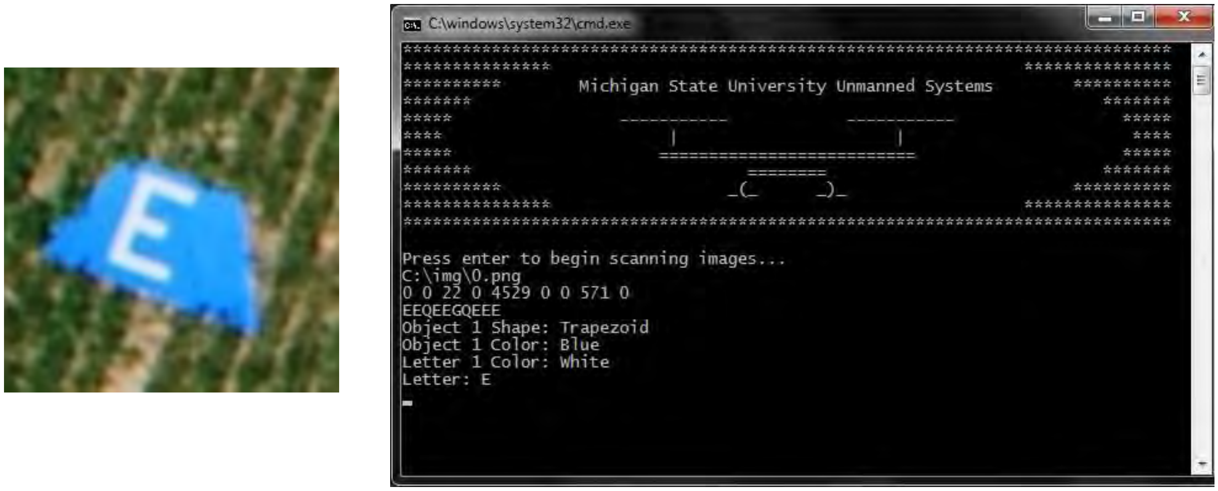

Figure 11: Sample Image and Code Display for Image

2.8 System Development and Modification for Specific Tasks

- Autonomous Flight:

In order to accurately perform autonomous flight and meet the navigational objectives set forth by the rules, the autopilot and GNSS (Global Navigation Satellite System) receiver had to be located in an optimal fashion, so as to mitigate any potential problems that could be caused by the nature of the aircraft. The Pixhawk autopilot contains a variety of sensors, of which one of the most important is the IMU (Inertial Measurement Unit). The IMU contains 3-axis accelerometers on it that are sensitive to vibrations and can potentially fail if saturated with vibrations. Based on the team's previous experience, a decision was made to use two layers of vibration attenuation. A vibration dampener was designed based on open-source attenuators, and was 3D printed. An old Pixhawk was mounted on the 3D printed dampener, using double sided foam tape and placed on a vibration testing unit for validation (*Figure 12*).

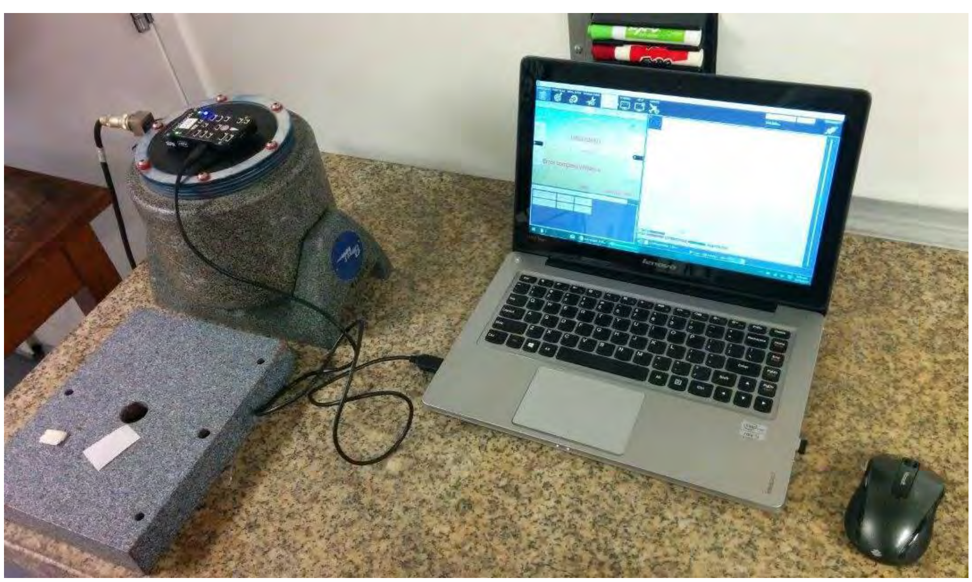

Figure 12: Pixhawk Vibration Testing

The testing yielded successful results at both lower, as well as higher frequencies. This setup was then installed in the LRU for further flight-testing.

The GNSS receiver used by the team had to be able to connect not just to GPS, but also to GLONASS and Beidou. Based on this requirement, a U-blox Neo M8M GNSS receiver was chosen for installation on the UAS. This receiver provided a good balance between quality and cost, as each receiver was very reasonably priced and was accurate to within a meter when tested in ideal conditions. The “cold-start” time on the receiver was only 40-50 seconds, which was just long enough for the autopilot to run its startup procedure. This enabled the team to be airborne in minimal time. The GNSS receiver also contained a magnetometer, which mounted on the LRU, placed it almost 10 cm above the power distribution harness and approximately 28 cm from each motor, greatly reducing the possibility of electromagnetic interference.

3 Tests and Results

3.1 Final Operation Specs for Ventus

MTOW at 50% throttle: 6.16 kg
Max Flight Time: 25 minutes
Max Airspeed: 7 m/s
Total Aircraft Weight: 4.03 kg
Theoretical Range: 2 km

3.2 Accomplished Tasks

Autonomous Flight
All flight tests conducted by the team utilized the Mission Planner software (*Figure* 13) at some point in the flight to follow set waypoints around the flight zone. Alternating routes and returns to the landing zone were successfully completed.

Figure 13: Mission Planner Flight Plan

Off-Axis Target
The aircraft was able to move into a position near the off-axis target. Objects were able to be identified – even without magnification – due to the clarity of the action camera.

Figure 14: Sample Targets Viewed from Action Camera at 30m AGL (left) and 50m AGL (right)

ADLC
The image relay system was tested independently and proved to be successful at long ranges. The images received by the Imagery Console were run through the software written by the team, and target data was successfully extracted (*Figure* 11) with a False Alarm Rate of less than 20%.

Emergent Target
Using the manually controlled gimbal, the RVT operators were able to search and detect objects on the monitor from flight altitudes required for competition.

3.3 Failures and changes

The original Ventus aircraft was based off of the Kongcopter FQ-700 frame. However, the frame was not used for SUAS 2016 because it would have been costly to purchase another frame as a reserve. Instead, the H4 Reptile frame was chosen. This frame proved effective in filling the team's requirement for modularity and low weight. The LRU and dual-battery support were maintained in all iterations of the H4 Reptile Ventus model.

In the first Reptile model, four struts made of aluminum with rubber knobs on the end acted as additional landing gear to remove the springiness from the current plastic landing gear. They worked effectively, but were removed due to weight and replication concerns.

4 Safety

To enable long-term sustained flight operations, safety was made top priority. Standard Operating Procedures were formulated and strictly adhered to. Based on the team's experience and knowledge gained from the open-source community, the team took into consideration mishaps that could possibly occur and pose a threat to team members and/or spectators. Solutions were engineered and response protocols were formulated, in case something were to go wrong.

4.1 Operational Safety Criteria

Before commencing flights, the Pilot-in-Command and Air Boss conducted a comprehensive pre-flight check. The pre-flight check involved the physical inspection of the aircraft, testing of communication systems and payload operator readiness level.
Primary pre-flight check parameters include:

- Inspection of airframe, undercarriage, propellers and prop adapters, motors and motor mounts, ESCs and all external electronic connections to ensure that everything is secured.
- Inspection of the LRU, batteries and MILSPEC connector contact, followed by a simple range test with the safety pilot transmitter.
- Inspection of payload systems and data links and antennas to the Portable GCS, Imagery Console and Remote Video Terminal.

4.2 Safety Risks and Mitigation Methods

To deal with the risk of an accidental arming and rotation of motors, the Pixhawk autopilot followed a two-step arming sequence. The autopilot boasted of a primary safety switch which needed to be turned "on" by pressing and holding for a few seconds, thereby preventing an accidental arming signal from the transmitter from powering on the motors. The secondary arming requirement prevented the motors from turning in case the safety switch was "on" and a throttle signal was given. This mitigated the risk of any mishaps that may be experienced in case of sudden rotation of propellers.
In the event of a motor, ESC or prop failure, a recovery of flight by a quadcopter was deemed unattainable. To prevent harm or damage to life and property, the team incorporated the use of the "Stabilize" flight mode offered by the Pixhawk. This flight mode allowed the Safety Pilot to cut throttle to 0 in case the aircraft lost propulsion on any arm, thus terminating flight.

In case of a loss of communication with the primary link (433 MHz Transmitter with Safety Pilot) for more than 30 seconds, the system was programmed to return to the takeoff position and land there. In compliance with Section 9.5.6.2, this (RTL) mode was also programmed as a safety flight mode, in the event of the Safety Pilot losing orientation, ground control losing data link communication or the air boss/head judge ordering a mission abort. In case of a loss of communication with the primary link for more than 3 minutes, the system will abort mission by terminating flight.

4.3 Team Safety Standards

MSU Unmanned Systems adheres to strict safety guidelines to ensure the safety of the team, the environment, and the vehicle itself. The team wears reflective safety vests during all testing operations, and maintains radio contact with spotters using two-way radios to ensure the vehicle does not leave the set fly zone or encounter any obstacles. A pre-flight checklist is used before and after every test flight. During flights, the team remains at their respective competition locations – only the Air Boss and Safety Pilot are permitted to interact with the vehicle once operations have commenced.

5 Conclusion

This journal paper summarizes the design and implementation of Michigan State University – Unmanned System Team's *Ventus* Unmanned Aerial System. The team, composed of various engineering disciplines within MSU's College of Engineering, focused on creating a vehicle that could efficiently accomplish as many primary and secondary tasks as possible from the mission parameters set forth for the AUVSI-SUAS 2016 competition. The team successfully developed and tested the Ventus aircraft, despite less-than-ideal Michigan weather conditions, and is proud to present it for this year's competition.