Autonomous Targeting Sentry

Ethan King, Daniel O'Hara, Stephen Rodriguez & James Van Gostein

School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* - This system is a combination of computer vision software and embedded hardware to implement an autonomous defensive turret, easily deployable in nearly any field. The software calculates the optical flow of moving targets by utilizing the iterative Lucas-Kanade algorithm with pyramids. The software then finds the primary target and communicates the location of the target to the embedded hardware to rotate the turret and turn on the various peripherals, such as an alarm and laser sight.**

***Index Terms* — Automatic Control, Computer Vision, Embedded Software, Microcontrollers, Motion Detection, Open Source Software, Servo Motors.**

I. INTRODUCTION

The design and implementation of the Autonomous Targeting Sentry (ATS) was fully funded by the design team. The ATS design is intended to make a valuable defense systems that is reliable, accurate, intelligent and efficient. The ATS system will use the openCV libraries to detect, track and eliminate targets. This software will be integrated with several microcontrollers controlling servo motors, laser pointer, alarm, and the trigger.

The turret will contain two servo motors for pan and tilt of the paintball gun along with wire attached to control the fire control. High definition web cameras will be used in conjecture with openCV to detect and track enemy targets. When a target enters the field of view of the web camera, the system begins tracking the target. The target will then be warned that they are entering a restricted zone. The on-board computer continually updates the targets coordinates while the microcontroller translates coordinates to pulse widths for the servo motors to move the turret into place. After the warning to the target, the system begins to fire upon the target until the target exits the field of view or the system determines the target is no longer a threat.

II. SYSTEM REQUIREMENTS

In order to make a successful prototype that can demonstrate real world applications, the ATS system must first be affordable. With a wide variety of potential targets, the ATS system should be capable of tracking multiple moving targets either fast or slow. Since many industries push newer and more advanced components, the ATS system must have components that are easily interchangeable. The ATS system must also be well protected by a sturdy base construction. To meet these requirements the following objectives were created:

A. Hardware Requirements

- Weapon shall have a hit ratio greater than 50%
- The entire turret system shall weigh no more than 30 pounds
- Yaw servo motor shall cover a lateral angle of tracking of 135 degrees
- Pitch servo motor shall cover a vertical angle of tracking of 90 degrees
- The turret shall be battery operated and operate for a minimum of 1 hour before a recharge is required
- The system shall have a lag of no more than 0.25 seconds upon command via manual control or automatic tracking.
- The battery system shall not output more than 12 V at any given time
- The gun shall be able to fire from 5-15 paintballs per second

B. Software Requirements

- ATS shall play an audio file to warn incoming intruders 1 second after entering the field of vision.
- ATS will play a second audio file, 5 seconds after the target enters the field of vision, to warn the target to alert them the gun is about to fire.
- ATS will have a response time of 1.5 seconds from finding a target to aligning the gun
- ATS will be able to track and prioritize at least 3 targets simultaneously.
- The tracking class will transmit the coordinates of the target to the microcontroller with an accuracy of 1 degree.

III. MECHANICAL DESIGN

ATS is a hanging turret built from the front fork of a children's bicycle. The bicycle fork provides the two axes required for implementing the pitch and yaw servo motors. The yaw is implemented by fitting a servo arm between the L-bracket that normally connects the steering of the bicycle to the front fork the motion is aided by the existing ball bearing located in the head stock. The pitch axis comes from the hub of the bicycle which is used normally to attach the spokes to the wheel of the bike. The servo was mounted on a plate welded to the fork and the servo arm was fitted between two screws on the gun mount. Just like the yaw the pitch motion is assisted by the ball bearings located in the hub. The gun mount is made out of a single piece of extruded aluminum attached to the hub with the use of two screws through the spoke holes. Grooves were cut in the aluminum to extend the solenoid valve wires to the gun's PCB at the top of the base. Finally the metal was powder coated flat black to match the gun and give the appearance that the turret and gun are one piece.

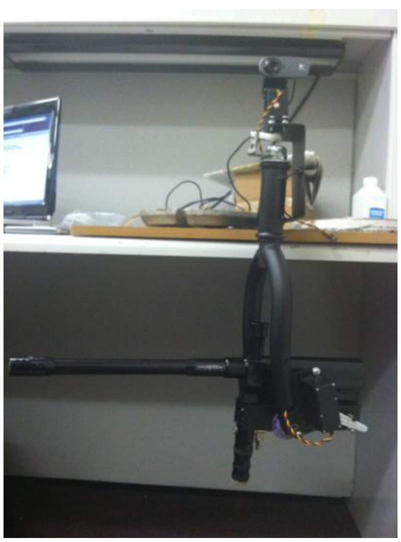


Figure 1 Gun and Mount

IV. HARDWARE COMPONENTS

A. Servo Motors

Servo motors were chosen to control the position of the turret because of the ease of implementation. High torque analog servo motors were chosen to control the pitch and yaw of the turret. Both servo motors operate at 5.0 volts for approximately 300 oz-in of torque. The servo motors are controlled via a pulse width modulation signal with 50 Hz frequency. Each servo motor has a built in potentiometer to change position based of the duty cycle of the PWM signal. This PWM signal is easily controlled by a microcontroller.

B. Microcontrollers

In order to appropriately control the ATS System, an embedded hardware solution that interfaces directly with the vision tracking software on the laptop was developed. While the laptop is performing the vision tracking algorithms, the goal of the embedded hardware is to recognize commands from the laptop and control the electrical and mechanical hardware on the system.

In order to keep development cost low, the embedded solution is primarily composed of four separately programmed Texas Instruments MSP430 value line microcontrollers working in parallel. Because of the low cost of the MCUs, each microcontroller was able to have a specific function. The four functions of the different MCUs are UART to I2C communication, servo motor controls, accessory controls, and capacitive touch controls.

The MSP430 microcontrollers used were the MSP430G2231 and the MSP430G2452. Both MCUs are running at 1 MHz clock frequency and a 3.30 operating voltage. The G2231 was included with a development board provided by Texas Instruments and only featured 2 capture and compare timer registers. This resulted in difficulties of providing accurate PWM lines for multiple servo motors.[1] The G2452 included 3 capture and compare timer registers as well as support for capacitive touch devices.[2] Due to the limitations of the MCUs, the G2231 was implemented for UART to I2C communication and accessory controls while the G2452 was implemented for capacitive touch and servo motor controls.

C. Camera

The web camera is a vital component in the system design. The sole purpose of the web camera is to provide optics for the ATS system, from there the software will be implemented so the system has motion detection and tracking capabilities. For these implementations to occur, a high definition camera will need to be used. The system will be utilizing a Logitech C310 HD Webcam. The webcam has a widescreen video at 720 pixels HD Resolution, a built-in mic and auto adjustment for poorly lit settings. The webcam will be attached to the turret above the paintball gun for accuracy of the target. The webcam will also be connected to the laptop via USB and have a real-time video capturing that will display on the laptop.

The web camera's field of vision is limited; this restricted the yaw servo motors range to around 30 degrees. The designers addressed this issue in both embedded software and the onboard tracking software.

D. Laptop to Embedded Systems Communications

Since the embedded system that controls the ATS system is physically separate from the laptop performing vision tracking algorithms, a special microcontroller and custom software was chosen to convert data from a laptop to the rest of the embedded system. The software that communicates with the vision tracking software uses the serial port class of Microsoft's .NET framework class library to transmit commands in 8 bits at 9600 baud along a USB to RS232 cable. The RS232 cable is fed into a MAX232 level changer, which is then fed into a UART to I2C microcontroller that decodes data and transmits to the appropriate MCU. Since the laptop is communicating directly with a MCU with a sole purpose of converting data communication protocols, handshaking is disabled in the software.

E. Alarm

Before designing the alarm system, a decision needed to be made based on sound volume. Since the ATS system would need to successfully warn the incoming targets, it was crucial that alarms were heard before being fired upon. To do this, the designers have opted to implement two types of alarm systems, a continuous alarm and a set of speakers to play an audio file that will verbally warn incoming targets.

The first alarm is going to be the continuous alarm. This alarm system's loudness will be based on the input current voltage, the higher the input current voltage will result in a higher decibel level, making it louder and easier to hear. This alarm will be directly connected to the microcontroller so when the software detects the incoming target, a signal will be sent from the microcontroller triggering the alarm.

The second alarm is going to be the speakers of the laptop. The decision to implement a set of speakers is because it has adjustable volume settings and the users have the option to choose verbal warning for incoming targets when they are being tracked. So when an object is being starts to be tracked, the software will trigger this alarm by playing a recorded audio file. This warning is intended for incoming humans so that they can properly understand the system's intent.

The designers choose to use both of these alarms systems to replicate real life alarm systems. Generally, when a continuous sounding alarm goes off, it is loud and high pitched, which usually initially grabs the attention of the audience. At the same time, verbal commands are being announced warning of the imminent danger or providing specific instructions. For this reason, the designers of the system chose to use both alarm systems.[3]

F. Battery (include voltage regulators)

The implementation of the battery will be a simple connection to the PCB. The main components on the PCB that will need the power provided will be the microcontrollers. The designers will be able to connect the battery to the terminal block connectors to provide power. The 12 volt lead acid battery will be fed into two regulators, a 3.3V and 5V switching regulators. Once the power is implemented onto the microcontroller, the ATS system is essentially powered to perform as designed.

G. Paintball Gun and components

The paintball gun mounted to ATS is an Airtech E-Matrix. The trigger handle was removed to provide the clean turret look the group was looking for and the gun's PCB was placed next to the main PCB and laptop. The gun is triggered by shorting two pins which then sends a pulse to the solenoid valve which then causes the spool valve to fire. The gun will be supplied with compressed air from a carbon fiber wrapped high pressure air tank, and the paintballs are fed to the gun using an electronic hopper and flexible tubing.

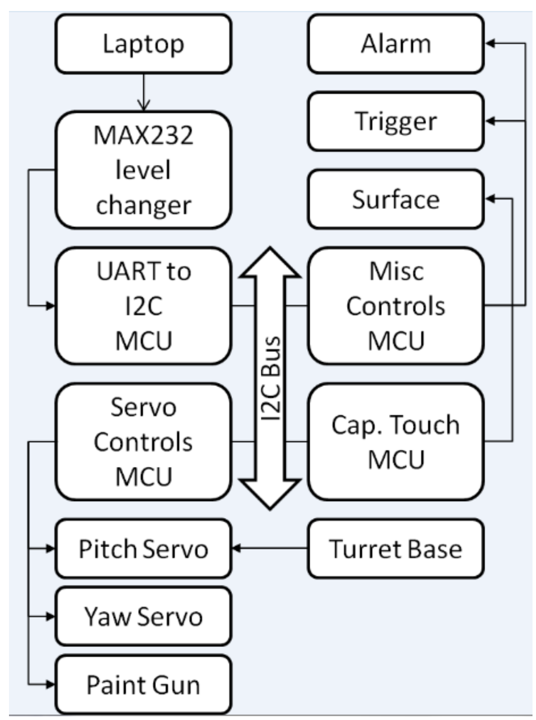


Figure 2 Hardware Block Diagram

VI. SOFTWARE COMPONENTS

The ATS software is responsible for taking in a web camera and generating a decision to fire or not and if so where to fire. To do this task, computer vision software will be needed. OpenCV's library will be used to take care of the motion tracking and then the ATS code will take that information and make a decision on what is and isn't a target based on a simple clustering algorithm, and if it should fire based on the output of the clustering algorithm. The program will write to files whether to fire or not, what the fire rate is set as, and the pitch and yaw angles that the gun needs to be at to hit the target accurately. The software was designed in C++ in the Visual Studio 2010 Professional Edition IDE.

The class diagram of the tracking software can be seen below. The ATS class contains the main function which actually performs the tracking. The other two classes, TrackPoint and TargetID, are data structures that are utilized to represent the points used in tracking and the groups of points that represent a target, respectively. Each class consists of a variety of variables and methods to operate on those variables. The details of the other two classes can be seen in the data structure section of software components. The main class follows a linear structure, where it creates the video stream, gets a set of points, tracks them, creates groups from clusters of points, find the most important cluster, determines where that target is located and then outputs the angle of that target.

A. OpenCV

Open Source Computer Vision Library, or OpenCV, is a major component in the software. ATS will be designed with the use of the version 2.3.1 of OpenCV. The system will utilize a large variety of the computer vision methods to go from the web camera image to the decision of when and where to shoot. Object detection, Object tracking, motion tracking, color detection, and facial recognition are all featured within OpenCV. With these open source algorithms along with our own code, the system will be able to make well informed decisions based on all targets found in the range of the ATS. Using these algorithms will also allow the system to be more advanced because of the time saved from having to develop computer vision algorithms for the specific application. The OpenCV object detection algorithm utilizes machine learning to be able to detect haar-like features of the object within the images it receives from the input, which, in our case, is the web camera. The OpenCV object tracking algorithm we will use is more popularly known as the Lucas-kanade algorithm. It is the most popular object tracking algorithm, and does so by using optical flow.[4]

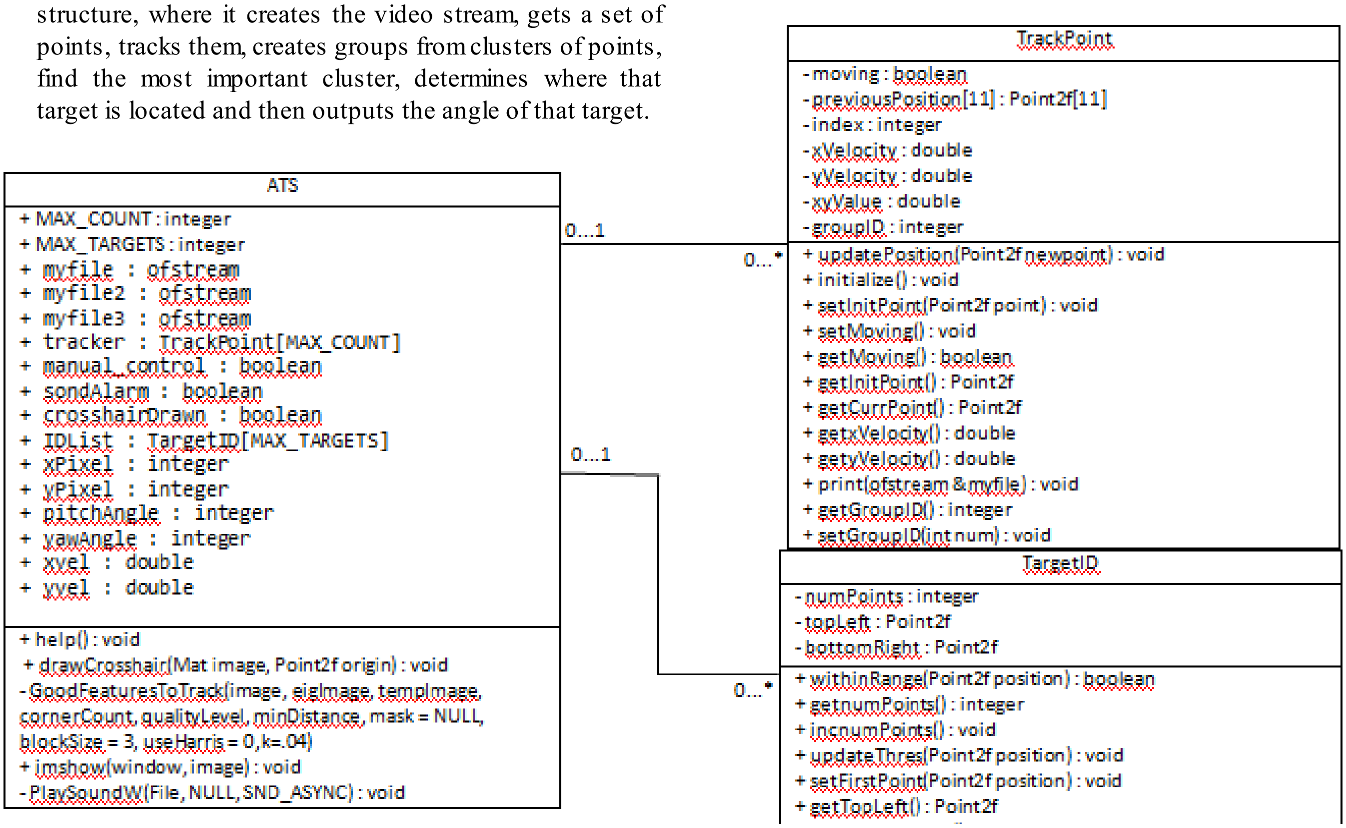


Figure 3 Software Class Diagram

B. Optical Flow

Optical flow is most commonly described as the apparent motion of objects or edges observed from a stationary observer. An optical flow algorithm, the iterative Lucas-Kanade algorithm with pyramids, will be used to track motion within the field of vision of the ATS. The optical flow algorithm is provided by openCV and takes in a set of points from frame A and then tries to locate each point in frame B where A and B are two frames from the web camera. These points will be generated from two methods that get a set of points that represent strong features and refine the locations of those points to sub-pixels, respectively. The results from the optical flow algorithm will then be used to make targets.[4]

C. Target grouping

Targets will be created from a cluster of moving points. The points received from the optical flow algorithm will be checked for movement and then all points will be placed into a group based on proximity to other points in that group. Each time a point is added to a group, the threshold grows to accommodate the area around the newly added point. This process continues until all points have been placed in a group. Once this is complete, only groups with three or more points will be classified as targets, with the one with the most points being the highest priority and the target that will be fired upon. If there are no such groups of points then the field of vision is deemed to be empty of targets and the system will remain in standby waiting for a target to enter the field of vision.

D. Data Structures

Many advanced data structures were used to improve the software. Many of these data structures were created by openCV for their algorithms, such as Mat (a matrix object), Point2f (a floating-point x,y coordinate), VideoCapture, and Video Writer (gets the web camera stream and writes it to a file, respectively) and a few other minor ones.[4] Two structures that were designed by the ATS team to simplify the code are a TrackPoint and a TargetID. The TrackPoint class is essentially a Point2f with much more data stored. A TrackPoint has its initial position (used for resetting the point when it moves off the screen), its 10 most recent positions (which are used in velocity calculations), a groupID (to determine what group that point is associated with), a moving boolean (true if moving, false if stationary) and a couple more variables used to simplify the software.

E. Embedded Software

A separate MCU was chosen control the PWM lines of the servo motors in order to keep down latency effects that may occur during I2C data transmission not pertaining to servo movement. Instead of having the same MCU waste cycles to process data not related movement, the servo motor control MCU can ignore any data not correctly addressed and, in turn, continuously update servo motor positions. The PWM signals were controlled by three capture and compare timer registers operating in “count up to mode.” To provide the 50 Hz signal required by the servo motors, one C&C register holds the period, 20000 clock cycles, and two remaining C&C registers hold the duty cycle of the 50 Hz signal. A period of 20000 clock cycles was chosen by assuming the MCU was operating at 1 MHz frequency.

The angle of the servo motors is calculated from the minimum pulse width, 900 ms, and the maximum pulse width, 2100 ms. The equation follows a simple linear $y = mx + b$ format. M is equal to the difference between the maximum and minimum pulses divided by the angle resolution of a specific servo motor. X corresponds to the angle provided by either the vision tracking software or manual control options. B corresponds to the minimum pulse width of a specific servo motor. Once the servo motor MCU receives data from the I2C bus, it immediately decodes the data, and updates the capture and compare registers using the above equation to update the PWM lines.

To accompany the servo motor controls MCU, a separate MCU controls the miscellaneous functions of the ATS system including the alarm and trigger of the paintball gun. Since this MCU is only currently controlling two functions, it has the option to control additional features such as the laser pointer, relays for additional lights, or possibly extra security features. The trigger of the paintball gun is controlled simply by interfacing one pin of the MCU, a transistor, and electronic solenoid of the paintball gun. The alarm is controlled by one pin of the MCU and a transistor to power the alarm via the 5V switching regulator.

For the embedded system and laptop to communicate, a table of codes was generated to compress all data into one signed char for easy transmission. For servo motor controls, a range from 00 to 30 hex is reserved for the yaw servo angles, and a range from 31 to 60 hex is reserved for the pitch angles. In order to apply the yaw angle, the yaw angle is directly equal to the OpCode. In order to apply the pitch angle, a value of 31 in hex is subtracted from the OpCode to retrieve the correct pitch angle. Although the ATS system currently uses approximately twenty points in either yaw or pitch directions, from web cam limitations, the system has the potential to reach 48 points in either direction. The ATS system also reserves codes between 70 to 77 in hex for miscellaneous controls. Specific codes can be viewed in the table below. In addition to compress commands into one byte, the below table allows for any MCU decoding the data to determine the correct address when transmitting to the I2C bus line.[5]

Op Code (Hex)	**Operation**	**MCU**
0x00 to 0x30	Set Yaw Angle	Servo
0x31 to 0x60	Set Pitch Angle	Servo
0x70	Set Alarm OFF	Misc
0x71	Set Alarm ON	Misc
0x75	Set Trigger OFF	Misc
0x76	Set Trigger Single	Misc
0x77	Set Trigger Automatic	Misc

Table 1 Output OpCodes

F. Pixel to Servo Motor Angle Conversion

After a moderate amount of testing, it was determined that the webcam utilized has a field of view of approximately thirty degrees. From this data, the x and y pixel values were divided by their maximum possible value, 640 by 480 pixels, and then multiplied by the maximum angle size of thirty degrees. This in turn created thirty points of interest from minimum left and top and maximum right and bottom of the webcam feed.

After further testing, it was determined that a scaling function was necessary to properly calibrate the turret. These calibrations were needed due to errors from image stretching near the edges of the video feed and a change in webcam positioning.

This calibration is performed by recording the angle of the turret in the left edge, top edge, right edge, an bottom edge of the screen. A simple linear equation of $y = mx + b$ is used to scale the vision tracking software angles to the appropriate turret angles. M corresponds to the difference between the maximum angle and minimum angle divided by the thirty. X corresponds to the software angle. B corresponds to the minimum angle. Y corresponds to the calibrated angle. This calibration was applied to both x and y angles from the vision tracking software.

G. ATS Communication

In addition to the vision tracking software, an additional program was created to read text files created by the vision tracking software. ATS COM is written in C++ using the serial port class of Microsoft's .NET Framework.

One the program is started it immediately ask the user to select a COM port to communicate with and a name for the port. The program then automatically configures to 9600 baud, no RS232 handshaking, one stop bit, and 8 bit data segments. After configuring the software, the program constantly reads text files and determines if there is a change in angle or system status. If a change was determined, the program will update the data, encode it according to the OpCode data, and transmit to the COM port specified by the user. The data is then handled by the embedded system components.

VII. Control Options

A. Automatic targeting

The primary mode of operation is the automatic targeting configuration. Automatic targeting uses vision tracking software and serial port communications to control the ATS system. The system is pre-calibrated but has the option of recalibration if the web camera and turret positions become out of sync. During automatic control, there is continuous live optical feed on board the laptop screen. This enables the user to visually monitor incoming targets or switch to manual control if desired.

B. Keyboard

The designers wanted the ability to also control the ATS system manually and to do this, the designers mapped certain characters from a keyboard to represent specific functions. To use the manual controls, one must first click the 'M' character on the keyboard, this switches the program from automatic targeting to manual control. After this point, the turret is operated by the user and can aim the turret up down, left, or right by choosing the characters 'W', 'S', 'A', 'D' respectively. All these characters were chosen as the shortcut keys for manual control. To fire upon a target, the SPACE key needs to be pressed.

C. Xbox 360 wireless controller

ATS will have the ability to be controlled via a wireless Xbox 360 controller. This feature allows the user to control the sentry from a distance with an intuitive controller. The controller interfaces with the laptop via a USB connected wireless gaming receiver. The team mapped the buttons of the controller to the corresponding keyboard controls. Gaining access to the controls is achieved by pressing the start button of the controller. Pressing the start button again will revert back to autonomous mode. The turrets movement is controlled by either the right analog stick or the directional pad, while the firing control is mapped to the right trigger. Controls to toggle the laser pointer, alarm, and reset the motion tracking points will be mapped to the A, B, and X buttons.

D. Capacitive Touch Controller

A separate MCU was chosen to read capacitance and update the ATS system angles. The capacitive touch controls use the relaxed oscillation API calls provided by Texas Instruments. In short, the RO method changes a timer frequency based on the capacitance. Controlling the ATS system via the capacitive touch controller is intuitive. If a user taps surface up, right, down, or left, the ATS system will adjust accordingly. If the user taps the middle the ATS system fires a single shot. The function of CT MCU is not to interface directly with the turret, but to send the appropriate commands to either the servo motor controls or accessory controls. All control should be immediately reflected on the physical turret.

When the capacitive MCU initializes, it immediately takes a baseline capacitance measurement and stores the raw measurement into an array. After this initialization, the MCU constantly cycles waiting for a user input. When the user does touch the capacitive touch surface, it compares the current capacitance to that stored in a the calibration array. If the current capacitance exceeds a hardcoded threshold, a touch is registered.

V. PRINTED CIRCUIT BOARD

In order to fully integrate the hardware components, a custom printed circuit board was built to handle multiple components and the high current drain from the servo motors. The PCB will house the four microcontrollers, voltage regulators, and additional components required for communication. The PCB will also feature the capacitive touch surface device. Terminal block connectors allow for external batteries and devices, such as a laser pointer, to integrate directly with the embedded system. The PCB layout can be viewed in the figure below.

The PCB was generated using the free version of Eagle CAD software. Components chosen were primarily through whole to allow for ease of soldering. Gerber files were generated using Eagle's cam processing software and sent to 4PCB board house in Colorado. Instead of soldering the microcontrollers directly to the printed circuit board, dip sockets purchased at radio shacked will be used. This will any user to freely update code on the microcontroller as well as upgrade microcontrollers.

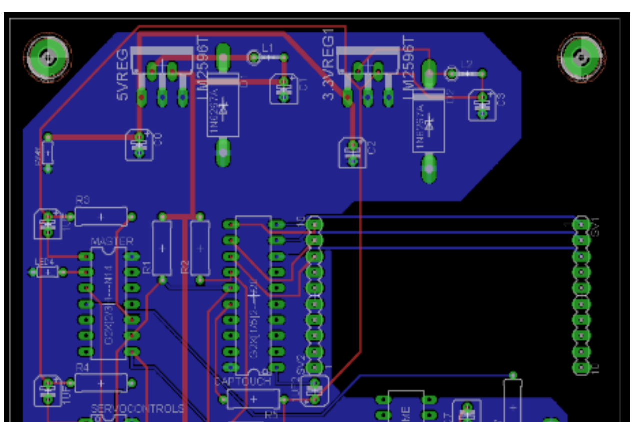


Figure 4 PCB Layout

VII. CONCLUSION

The ATS project started with an idea to create a fun, interesting, and creative product. The team intended to prototype, and demonstrate, a turret with many real world applications. ATS incorporates cutting edge computer vision and optical flow tracking with hardware components such as microcontrollers and servo motors to create a product the team is proud to have produced. ATS incorporates various fields of both computer and electrical engineering to accommodate and expand the interests of each engineer.

On the electrical engineering side, the team utilized multiple MSP430 microcontrollers to control the two servo motors as well as handle the communication between the various microcontrollers and the on-board laptop. The engineers designed and populated their custom printed circuit board powered by a 12-volt battery.

Since ATS is heavily dependent on the software it had to be reliable and accurate. The engineers decided to utilize optical flow tracking over the simpler blob or color tracking. The optical flow algorithm implemented is the Iterative Lucas-Kanade method with pyramids, which finds the optical flow between frames. Points moving in a similar space and direction were then grouped and given a target ID. The center point of the target is then converted from pixel coordinates to angles which the microcontrollers then use to rotate the turret to accurately aim at the target to eliminate it.

Designing and prototyping ATS was a rewarding experience for the team members. The team members encountered a magnitude of issues and challenging problems along the way, each of which provided a valuable learning lesson and problem solving experience. Many of the team's engineering skills were put to the test and as a result, a substantial improvement to those skills occurred.

ACKNOWLEDGEMENT

The developers of the ATS system would like to acknowledge Jon Waldmann for significant contributions to the design of the mechanical base and provision of a paintball gun.

The developers would also like to thank Texas Instruments for providing MSP430 development boards as well as hosting a seminar for embedded programming.

Lastly, the developers would like to acknowledge Dr. Richie for his guidance in communications and trigger integrations.

REFERENCES

[1] Texas Instruments (2010). MSP430G2x21, MSP430G2x31 Mixed Signal Microcontroller (Rev. I). Retrieved from http://www.ti.com/lit/ds/symlink/msp430g2231.pdf

[2] Texas Instruments (2010). MSP430G2x52, MSP430G2x12 Mixed Signal Microcontroller (Rev E). Retrieved from http://www.ti.com/lit/ds/symlink/msp430g2452.pdf

[3] Floyd Bell Inc (2012). TMC-86-530-W Technical Specifications. Retrieved from http://www.floydbell.com/products/specifications/TMC-86-530-W

[4] OpenCV (2011). OpenCV Documentation. Retrieved from http://opencv.itseez.com/

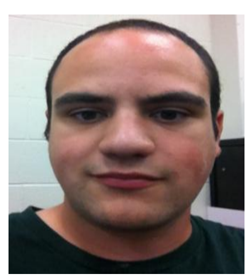
[5] Robot Electronics (2009). Using the I2C Bus. Retrieved from http://www.robot-electronics.co.uk/acatalog/I2C_Tutorial.html

BIOGRAPHY

Ethan King is 20 years old senior and graduating with a Bachelor of Science in Electrical Engineering in May 2012 from the College of Engineering at the University of Central Florida. He enjoys C and C++ programming, embedded system development, and signal processing.

Daniel O'Hara is 21 years old, graduating with a Bachelor of Science in Computer Engineering from the University of Central Florida. His primary interest is in software development with a background in C and Java. Also, he has familiarity with C++, HTML, Verilog, and MySQL. He plans to pursue a career in software development and may continue his education to obtain a master's degree in computer engineering.

Stephen Rodriguez is a 22 year old and is graduating with a Bachelor of Science in Computer Engineering with a Computer Science Minor from the University of Central Florida. He has been working as a systems engineer through the UCF College Work Experience Program for Lockheed Martin for the past two years. After graduation, he has accepted a job offer as a Systems Engineer for Lockheed Martin IS&GS in Gaithersburg, Maryland. While up there he plans to pursue a technical masters.



James Van Gostein is 21 years old and graduating with a Bachelors degree in computer engineering in May 2012 from the University of Central Florida. He is currently searching for employment as a software developer.