Aerial Intruder Removal – System for Tracking and Rendering Ineffective Knavish Enemies (AIR-STRIKE)

Kevin Chau, Scott Greenwald, Christopher Walls

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

Andrew Kirk

The College of Optics & Photonics, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This paper presents the design approaches applied in order to implement a safe and effective autonomous targeting system while operating with high-powered energy systems. With national defense interest rising, the emphasis for this project was to design a low-cost system that can target any object, of most colors, and fire a high powered energy beam at it. Using targeting and tracking algorithms, we are able to accomplish this task through a fully-autonomous system while viewing the systems information in real-time through its Wi-Fi network.**

***Index Terms* — Diode lasers, eye protection, image processing, IP networks, TCPIP, tracking loops, safety.**

I. Introduction

Our objective in the creation of AIR-STRIKE is to provide a miniature realization of an automatic anti-air defense system that uses directed energy to neutralize its targets. In this we aim to ensure reliability, precision, and safety. We define reliability as the success rate of neutralizing a target. To be reliable AIR-STRIKE must consistently target and pop balloons. We define precision as the accuracy of our direct energy weapon system. Therefore to be precise, the system must consistently hit the designated target when firing. We define safety in terms of the likelihood of harm coming to user or person in the area. To be safe this likelihood must be near zero. These three aspects are interlocked on a fundamental level in that each impacts that other.

Another significant objective is to perform all computation without relying on an external computer. Image processing is usually a fairly expensive process and as such it requires dedicated resources or a sufficiently powerful processor. We aim to accomplish this by having dedicated image processing hardware, as opposed to performing the processing upon our microcontroller. Instead this hardware, the PixyCam, will send the target coordinates to the microcontroller after performing the processing on its own. This frees up our time for handling control of the laser and translating coordinates.

II. Motivation

Tracking and directed energy systems are vital for our troops in order to locate and neutralize the enemy. The members of this group all, at one period in time, have worked in the defense industry, as engineers, in order to help improve our nation's security. Due to this common interest, and our previous knowledge for real-world defense systems, we decided to design our own targeting and tracking system. While our system is not suitable for our military to utilize, the concepts and implementation techniques that we have used are common in the real-world. Lockheed Martin, Raytheon and Northrup Grumman are all local defense contractors for our government that produce products similar to our project.

III. System Components

When outlining the system as a whole, it is easier to detail each subsystem in order to get a better understanding of how the final product will be integrated and simulated. This section will provide a technical introduction for each of our subsystems: the CC3200, Pixy, FlexMod P3, and S06J laser diode.

A. CC3200 Microcontroller

The AIRSTRIKE system requires several computationally intensive processes be completed in a timely manner in order to be effective. It was not feasible to use one processor to complete all the required tasks so AIRSTRIKE uses the "divide and conquer" methodology instead. The final design would have multiple tasks running in parallel on their own dedicated processor and communicating their results back to the main MCU. With image processing taken care of on the CMUcam5, that only left the network and main processes. While two additional microcontrollers would have fulfilled this role the choice was made to use the CC3200 single chip wireless MCU.

The CC3200 combines an ARM Cortex M4 and network processor in one package. It is a suitable microcontroller for this project because it provides integrated WiFi support and ample serial communication interfaces. Also the 80MHz clock speed and 256KB of RAM means that the microcontroller is able to store incoming information from the cameras and finish processing the data before the next set of identified targets are available from the Pixy camera. In addition the 1MB of SPI flash was sufficient to store the main program (~30KB) as well as web related files (~500KB). The alternative to the CC3200 was the TM4C1294NCPDT in combination with the CC3100 which would have provided better processing capability and greater memory, but at the cost of increased complexity on the PCB. Thus the CC3200 was decided to be the best low-cost microcontroller that satisfies the requirements.

B. Pixy (CMUcam5)

The Pixy is an all-in-one device which includes its own image processing subsystem, loaded with OpenCV. OpenCV is open-source software geared towards image processing. Geared with this, the Pixy can not only send and image to the primary processor, but also the coordinates of an object that it itself has detected.

The camera detects blobs of similar color. This works by having the camera learn a color and then having it seek for that color in an image. Once a blob of color reaches a certain size, measured by number of contiguous pixels, it is added to the list of objects to be sent to the primary microcontroller.

The Pixy has the capabilities to communicate via USB, or other I/O options such as UART, SPI, and I2C. It also has RC servo controller ports on the back. This is so that, if instructed to, the Pixy can in fact track an object on its own by controlling the servo system it's attached too. This is shown in Figure 1.

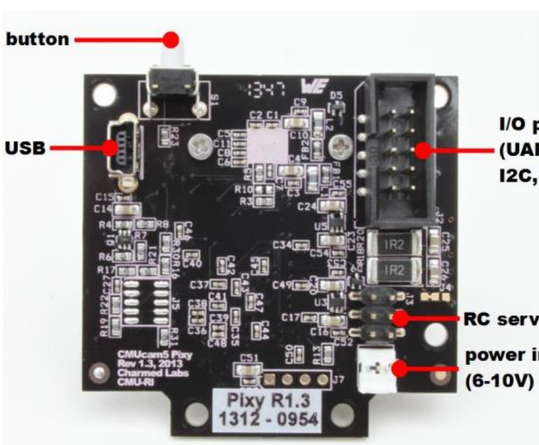


Fig. 1. Represents the various ports available on the Pixy device. Image courtesy of Charmed Labs.

For power one can use either the USB input at 5V, or unregulated input from 6V to 10V. It sports a 75° horizontal field of view, and 47° vertical field of view. It also operates at 50 frames per second (fps). Due to this, the system has capabilities to detect hundreds of objects at a time.

For communication with Pixy we've chosen to utilize SPI, primarily for its speed and its slave select capabilities in the event that we decided to use a second camera. Pixy communicates via SPI using 16-bit words as described in the table below. Separate frames are separated by two of the sync words seen in Table 1.

Bytes	Description
0, 1	Sync: 0xaa55 for a normal object, 0xaa56 for a color code object
2, 3	Checksum (sum of words 2-6, bytes 4-13)
4, 5	Signature number
6, 7	X center of object
8, 9	Y center of object
10, 11	Width of object
12, 13	Height of object

Table 1. Format of communications sent by Pixy to the host MCU.

A caveat to operating in SPI is that Pixy requires sync words to be sent from the MCU for syncing purposes. This is primarily due to the face that SPI is meant to send while it receives, and while dummy receivers and empty messages can be sent a sync message allows Pixy to keep track of the status of the MCU and if it should still be sending data. Pixy defines this sync bit to be 0x5A00 when no data is to be sent and 0x5B when you wish to send data followed by a word of the data to be sent.

C. FlexMod P3

The FlexMod P3 is the current driver that we will be using to supply the current to our directed energy system. When a diode is supplied with a constant voltage source, such as a battery, it will continuously draw current needed to produce a beam, even at the detriment of the diode. For this reason it is necessary that the diode is supplied with a regulated, constant current source to avoid any damage. The FlexMod P3 driver operates at 9V and contains a modulation input which will tell the energy system when to fire and when to turn off, or even just control the percentage of power the laser delivers.

Based on the power output vs current curve in Figure 2 we can see that the example M140 diode has a threshold current just below 200 mA but in order to achieve the power necessary to pop a balloon, a higher current would be needed. At 450 mA, our chosen laser diode (the S06J) will deliver an approximate power of 600mW at focus. The FlexMod P3 will operate from .1-4A with a modulation bandwidth of DC-160kHz. The FlexMod P3's properties are described in Table 2.

to achieve a spot size of only a couple millimeters at a power of 600mW, yielding a sufficient intensity to engage and eliminate a target within seconds.

While the diode is in operation, the current draw causes it to generate thermal energy. A thermal buildup across the semiconductor material can cause the device to malfunction and substantially decreases the diode lifetime. In order to preserve the diode, a copper heat sink was chosen to dissipate heat from the diode module. The heat sink design incorporates a solid metal slab and fin design to pull heat from the diode and dissipate it into air.

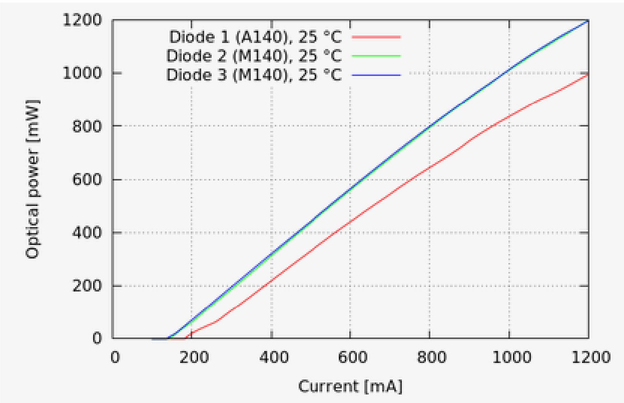


Fig. 2. Optical Power Output vs Current. Image courtesy of Achim Sack of Dodenring.

Parameter	**Specification**	**Unit**
Input Voltage (Vdd)	5 - 24	V
Max Output (Diode) Voltage	Vdd - 1.5	V
Output Current	0.01 - 4.0	A
Bias Current	0 – 1000	mA
Regulation Ripple (1Hz -100kHz)	< 1.0	%
Input Impedance	99	kΩ
Maximum diode lead loop length	3	ft
Maximum continuous power dissipation	25	W

Pin	**Description**
M+	Modulation input connection: 0-5V
G	Power ground and modulation ground connection
V+	Power input connection: 5-24V
-	Output return (Gnd) (LDK)
+	Positive current output (LDA)
Int	Interlock connection; connect to V+ via interlock loop

Table 2. FlexMod P3 Device Properties. Table courtesy of Innolasers.

D. S06J LASER DIODE

In order to successfully engage targets using directed energy, the system requires a stable and reliable laser system. A laser diode is an ideal energy source because of its compact size, yet impressive power output. The radiation emitted from the S06J laser diode lies at the edge of the UV-VIS spectrum, giving a highly energetic beam at 405nm with a linewidth of approximately 10nm.

While the diode supplies the system with the necessary energy to eliminate a threat, the emitted beam is highly divergent. In order to maintain high efficiency operation, we must be able to collect all the radiation and control the optical path so we can irradiate the target with a high intensity beam. This is achieved by using a three-lens optical system. This lens system has a large numerical aperture to collect the highly divergent energy from the diode. The optics following the initial lens will collimate the beam, effectively focusing it at infinity. This allows us

E. Gimbal/Servos

To allow for rapid targeting and accurate tracking, the motors guiding the directed energy system must be responsive and precise. Servo motors make a great choice for aligning our high energy beam onto the target since they come in compact yet effective packages. Metal gear MG90s servos were chosen for their robust design and high torque values necessary to bear the weight of the diode heat sink.

The MG90s servo motor shares the same dimensions as a standard 9g nylon servo motor, with the exception of the metal gears. These motors operate on a range of 4.8V to 6V giving torque values ranging from .1sec/60° to .08sec/60°. The speed and duration of rotation is determined by the input PWM signal from the microcontroller.

Our gimbal assembly is a Boscam 2D gimbal mount that requires two servo motors; one to travel through the azimuthal plane and another for elevation through the z-axis. The directed energy system has a custom assembly to properly mount along the center axis of the servo motors.

IV. SYSTEM CONCEPT

At the system level, a block diagram is needed to understand how each subsystem is integrated together. Figure 3 represents our system level block diagram. The general idea is to separate the subsystems and have them communicate and influence each other via the MCU.

V. HARDWARE INTEGRATION

Our subsystems are all powered and operate at a constant voltage of 5 volts, excluding the FlexMod P3 current driver which operates at 9 volts. These subsystems are powered collectively through a custom printed circuit board (PCB) detailed in section VIII.

All of our subsystems (PixyCam, both MG90 servos and the FlexMod P3 current driver) are directly or indirectly connected to the MCU's pin headers. The PixyCam

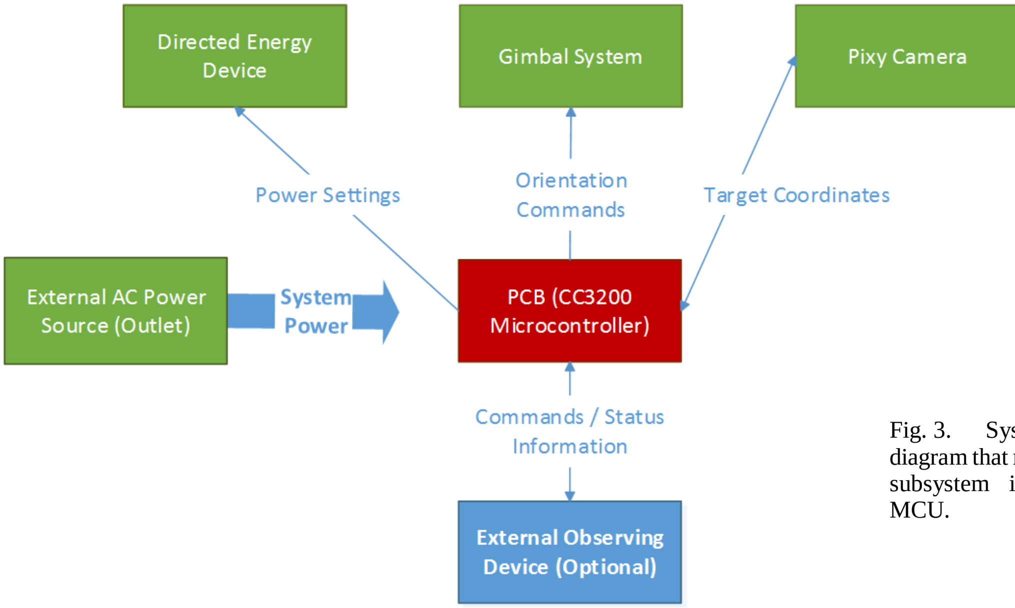


Fig. 3. System level block diagram that represents how each subsystem interacts with our MCU.

operates through the MCU's single serial peripheral interface (SPI) bus which consists of 3 different pins (MISO, MOSI, and SPI Clock). With this SPI port, the MCU is the master and the PixyCam acts as the slave. The MCU is able to transmit and receive data from the camera and then proceed along the software.

Both the FlexMod P3 current driver and the MG90 servos accept pulse width modulation (PWM) signals. These PWM signal are sent from the MCU to one of our custom PCBs in order to boost the signal from 3.3 volts to 5 volts. In this way the PCB acts as a logic level shifter.

VI. Software Detail

The software architecture is divided into tasks and after initialization the processor executes each task one after another looping indefinitely. The purpose behind this division of labor is to properly separate the individual components from the system such that communication and task management is streamlined and decluttered. The tasks are: networking, target acquisition, servo movement, and laser control.

A. *Networking*

The network code initializes the CC3200's network processor into access point mode and starts the HTTP web server. In the loop the program is notified of network actions via event callbacks. For example whenever a web page is requested or data is sent to the server an event will trigger the corresponding callback function. However the only callbacks that the program listens to is the HTTP POST and GET events because POST events mean an external user is sending data for the system to process and GET events either mean the user is requesting a web page or is requesting the latest system information. The network task serves to bridge communication between the MCU and network interfaces.

B. *Target Acquisition*

The target acquisition task is, as its name suggests, where the application selects a target and begins tracking it. To being, Pixy must be configured to track the color of the target the user intends to laze, this should be done before the user initiates defense routines. Once the primary application has been started, AIR-STRIKE begins attaining detection data from Pixy. This data will be comprised of detections ordered from largest to smallest and containing information on their position, color signature, and size. We then select the largest of these objects as our initial target. Whenever we lose a target, we once again select the largest detection.

After a target has been chosen we begin to record its positions in a circular buffer such that we can extract velocity and acceleration data, as well as build a smoother position track by averaging frames of information. We used the two most recent smoothed positions to determine velocity, and the three most recent to determine acceleration. The computed results give the system pixel coordinates which indicate where the system believes it will hit the target should it fire its laser.

This information then needs to be transformed into a polar coordinate frame from which we can extract a yaw and pitch to provide the Servo. To start with we have to define our coordinate frame. We use a standard definition for our Cartesian and Polar coordinates as seen in Figure 4, though some systems swap θ and ϕ.

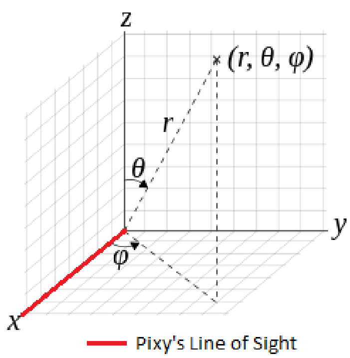


Fig. 4. Standard Cartesian and Coordinate frame relationship.

With this defined, we then convert the pixel coordinates of our target into a polar coordinates from Pixy's perspective. Pixy coordinates are defined as x_p and y_p, where x_p is our Cartesian y, and y_p is our Cartesian z. In order to being the conversion we use the equation:

$$\vec{\alpha_j} = \frac{\vec{D_j}}{\vec{P_j}} \tag{1}$$

where j a spatial dimension, $\vec{D_j}$ is the number of pixels in dimension j, $\vec{P_j}$ is the number of pixels available in dimension j, and $\vec{\alpha_j}$ is the degrees per pixel in dimension j.

Using this vector $\vec{\alpha}$, we can transform our pixel coordinate (x_p, y_p) into a polar coordinate (θ_p, Φ_p). θ_p is 90 minus our polar θ, and Φ_p is simply the opposite of our polar Φ. For range, the user must input an estimated range. This gives us a standardized polar coordinate (r, θ, Φ). Now we need to convert this into a Cartesian coordinate, using (2), (3), and (4) below.

$$x = r * \sin(\theta) * \cos(\phi) \tag{2}$$
$$y = r * \sin(\theta) * \sin(\phi) \tag{3}$$
$$z = r * \cos(\theta) \tag{4}$$

Next we add these values to a pre-measured Cartesian vector which measures the distance from our laser to Pixy in order to get the Cartesian coordinates of our target from the frame of reference of our gimbal. This can then be converted back into spherical coordinates for our gimbal via (5), (6), and (7) defined below.

$$r = \sqrt{x^2 + y^2 + z^2} \tag{5}$$
$$\theta = \cos^{-1}\left(\frac{z}{r}\right) \tag{6}$$
$$\phi = \tan^{-1}\left(\frac{y}{x}\right) \tag{7}$$

The only two values our servos need are θ and ϕ. As with before, the angle we send our pitch is actually 90 - θ, because the gimbal measures pitch as from the xy-plane as opposed to from the z-axis. After calculation these values are ready to be consumed by the Servo Movement task defined next.

C. *Servo Movement*

The servo task ensures that the correct pulse width modulation signal is being sent to the servos such that the elevation and azimuth match those calculated by target acquisition. It maps angle values to duty cycles so it can translate the target elevation and azimuth values calculated from the target acquisition task to pulse widths. Angles and degree of rotation were found to be quadratically related. The final result is the servos rotate the gimbal so the laser is aligned with the current target.

D. *Laser Control*

The laser tasks utilizes pulse width modulation to set a percent power output on the current driver which in turn drives the laser diode. Mapping pulse width modulation to power output is simple because it is directly proportional to the duty cycle. The issue here lies with when to activate the laser. As we do not get live feedback from the Servo's, we must wait until the transformed

VII. Network Interface

External Wi-Fi communication required extensive software implementation and is the main focus of the external communication task because the team planned to implement an embedded HTTP web server and use Wi-Fi in access point mode. This decision was made because with other communication mediums such as USB or Bluetooth the client device would need a custom application on the client side in order to provide a graphical user interface for the user. Also to support multiple platforms such as Windows, Linux, Mac OS, or Android a different application would be required for most of the platforms. Although the client platforms may vary, they all share a common application: a web browser.

If the external interface was implemented through a web server that served HTML and JavaScript web pages then the system would be able to communicate with a majority of platforms. Also given that even phones have web browsers, the user would have a mobile and convenient interface right in the palm of their hand. By using a web server for external communication it saved development time and made the external interface compatible with a wide variety of devices.

A. *WiFi Network*

WiFi and TCP/IP protocols are implemented through the CC3200 microcontroller which is unique in that it combines an ARM Cortex M4 and network processor in one package. This design means network processes do not take up resources on the main ARM processor and will increase system performance. The CC3200 supports the 802.11b/g/n standards, but for this implementation the system will provide an 802.11n compliant network in the 2.4 GHz band.

In order to ensure that only authorized users have access to the web server, the network is secured with WPA2-PSK. After being successfully authorized the DHCP protocol will assign the station an IP address. The protocol will also assign an IP address to the web server so the client will be able to establish a connection and request web pages. Additionally to prevent multiple users from trying to interface with AIRSTRIKE at the same time the DHCP server is configured to only lease two IP addresses: one for the server and one for a user.

B. *HTTP Server*

The HTTP server will serve web pages containing the user interface and complies with the HTTP 1.1 standard. At initialization a DNS type A record is created to associate “airstrike.com” with the IP address assigned to the server so users can just input the domain name rather than memorizing the IP address. In addition to serving web pages and scripts to the client, it also sends data between the main processor and user via HTTP POST and GET requests.

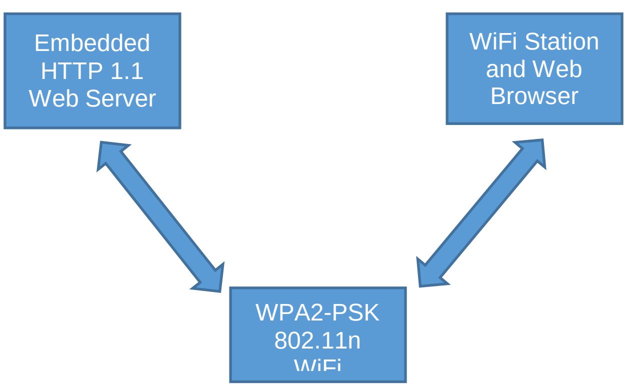


Fig. 5. Network Topology

C. *Web Interface*

The rendered HTML web pages will provide the graphical user interface for AIRSTRIKE. It provides data such as the current state of the laser, servo orientations, and current target information. Also the web interface lets users override AIRSTRIKE’s autonomous control to let users manually set the laser power and servo orientation. Utilizing WebGL the turret and targets are rendered in 3D in order to provide a more intuitive display of the system state. This means the client web browser will need to be HTML 5 compatible.

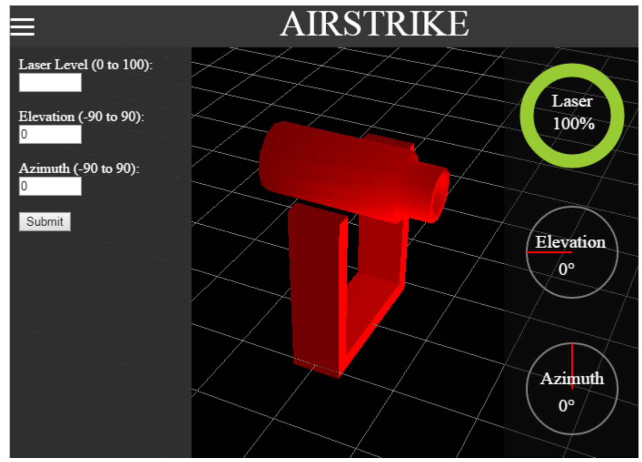


Fig. 6. GUI utilizing WebGL and HTML5

When the web interface is first loaded onto the user's browser the accompanying JavaScript code initializes a polling process that requests updates from the server every 25 milliseconds. Each request takes on the form of a HTTP GET request which is commonly used to retrieve web pages, but in this case the server responds with updated system information. However to the user updating information every 25 milliseconds would appear choppy so linear interpolation is used to ensure smooth transitions between updates. In order to send information from the user interface to the server HTTP POST requests are utilized instead. For example when the user manually sets the elevation and azimuth of the turret, the desired angle values are used as parameters in the POST request. When the request reaches the server the information is parsed and angle values sent to the servo control task. With these two forms of requests the network is able to establish bidirectional communication between the microcontroller and user interface.

VIII. MCU Board Design

Our main printed circuit board (PCB) contains both our CC3200 MCU and a FT2232D debugger chip. The design for the MCU board was completed with the Cadence software package (Orcad Schematic Capture for our schematic design and Allegro PCB Editor for our Board Layout). All of our PWM, SPI and GPIO ports are located on this board and communicate with all of our subsystems. The board is a 4-layer PCB where layers 1, 3 and 4 are conductor layers and the 2nd layer is our ground plane. The MCU board was printed through advance circuits (www.4pcb.com). Figure 7 shows the design of our MCU PCB.

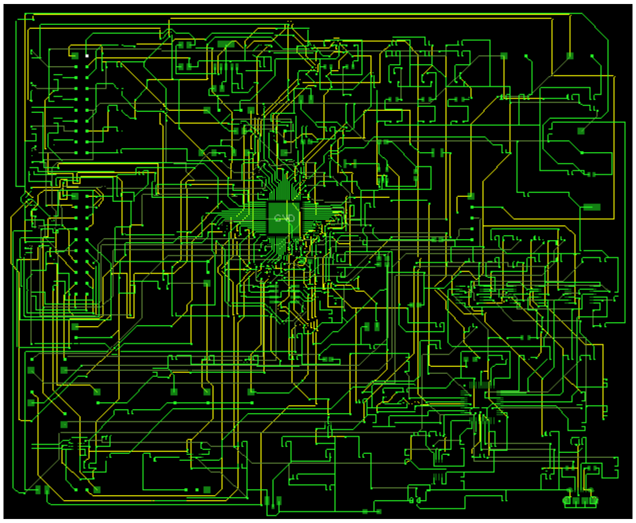


Fig. 8. MCU Board Design

IX. Power

A custom printed circuit board (PCB) was designed in order to supply all of the voltages needed to run our system. The board is powered through a barrel jack, where the input is converted initially from a 120 volts AC to 5 volts DC. A 10 pin header is connected directly to the input supplying 5 volts to each pin. All of our subsystems (including the MCU board) operate at 5 volts, therefore each subsystem will be powered through this header. The board also has 3 PWM inputs from the MCU (for both servos and the current driver) that are 3.3 volts. Through a transistor circuit, and utilizing a single voltage regulator, the 3.3 volt signal is amplified to 5 volts so that the subsystems can understand the output PWM pulses. The PCB is a simple 2-layer board. The Power PCB was created in Eagle CAD and printed through www.oshpark.com. Figure 9 shows the design of our Power PCB.

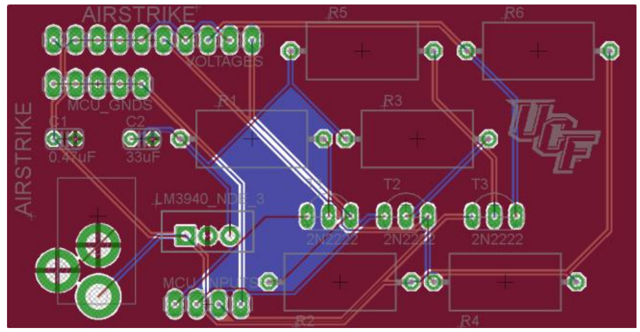


Fig. 9. Power Board Design

X. Safety

Since our project will incorporate a high power laser, it is critical that we take the proper safety measures to avoid any accidents that may occur with a stray laser beam. Lasers are divided into various class categories based on output

CLASS	US: FDA/CDRH	IEC 60825 (AMENDMENT 2)
Class 1	• No known hazards during to eye or skin *during normal operation* • Note: Service Operation may require access to hazardous embedded lasers	
Class 1M	N/A	• No known hazards to eye or skin, unless collecting optics are used
Class 2a	• Visible lasers not intended for viewing. • No known hazards up to maximum exposure time of 1000 seconds	N/A
Class 2	• Visible lasers • No known hazard with 0.25 seconds (aversion response)	
Class 2M	N/A	• No known hazard with 0.25 seconds (aversion response) unless collecting optics are used
Class 3a	• Similar to Class 2 with the exception that collecting optics cannot be used to directly view the beam • Visible only	N/A
Class 3R	N/A	• Replaces Class 3a (with different limits) • 5 x Class 2 limit for visible • 5 x Class 1 limit for some invisible
Class 3B	• Medium-powered (visible or invisible) • Intrabeam and specular eye hazard • Generally not a diffuse or scatter hazard • Generally not a skin hazard	
Class 4	• High powered lasers (visible or invisible) • Acute eye and skin hazard intrabeam, specular and scatter conditions • Non-beam hazard (fire, toxic fumes, etc.)	

Table 3. The classes of lasers and their properties according to FDA standards and more internationally regulated IEC 60825 standards.

power. Table 3 breaks down each class.

For our project, the S06J diode falls within Class IV. Because of this, we will conduct our experiments and testing in one of the UCF CREOL labs while utilizing proper eyewear. For our demonstration, we will swap out the S06J diode for the M140 diode which is a Class I laser and does not require any special safety procedures.

Since our laser will have a power output of over 500mW, we must ensure that any viewers have the proper eyewear to prevent any eye damage. The laser diode of our project will be operating at 405nm wavelength, so we will provide specified goggles with the necessary optical density filter to allow for safe viewing while in operation.

Laser safety goggles are rated based on the optical density they provide at specific wavelengths. The optical density is a ratio of transmitted power to total incident power. It is a measurement on a base ten scale, for example, an optical density (OD) of two would correspond to an attenuation factor of incident light of 102. To calculate the OD of a filter at a specific wavelength, we use (8) below:

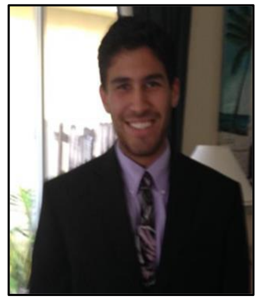
$$\lambda = \log_{10}(E_0 MPE) \tag{8}$$

where E_0 is the highest possible irradiance (proportional to power output) and MPE is the maximum permissible exposure which as the same units as E_0.

BIOGRAPHIES

Kevin Chau is studying Computer Engineering and Secure Computing and Networks at the University of Central Florida. He has interned at Lockheed Martin, DE Technologies and Dignitas Technologies from 2010 to 2014. Kevin is currently focused in computer networking and will continue graduate studies while working at a commercial business.

Scott Greenwald is studying Electrical Engineering. He has interned with Lockheed Martin as a Systems Engineer throughout the summers of 2013 and 2014. After graduation, Scott will begin full time with Lockheed Martin in Orlando while pursuing his masters in Electrical Engineering



Andrew Kirk studying Photonic Science and Engineering degree program at the College of Optics and Photonics. He has/is interning at Lockheed Martin. When he graduates he'll be the first UCF student to graduate with a bachelor's Degree in Photonic Science and Engineering.

Christopher Walls is studying Computer Engineering. He has worked in the Intelligent Systems Lab at UCF and has/is interning at Lockheed Martin as a Systems Engineer and a Software Engineer in the summers of 2013 and 2014. After graduation he plans to work full-time as a Software Engineer while pursuing a Master's Degree.

ACKNOWLEDGEMENT

The group members would like to thank Dr. Samuel Richie and Dr. Patrick LiKamWa for all of the assistance and expertise they have given us throughout the previous year. Furthermore, we'd like to extend a special thanks to Dr. Matt Drago for providing general advice and explaining concepts important in systems similar to ours. Also, we would like to thank Boeing for sponsoring our project.

REFERENCES

[1] Texas Instruments, "CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things solution, a Single-Chip Wireless MCU," June 2014. [Online]. Available: http://www.ti.com/lit/gpn/cc3200. [Accessed 14 11 2014].

[2] Nautilus Integration, "FlexMod P3 UserManual," 2010. [Online]. Available: https://innolasers.com/shop/index.php?id_product=11&controller=product.

[3] O. Svelto, Principles of Lasers FIfth Edition, Springer, 2010.