Remote Touchscreen-Controlled Defense Turret

Szu-yu Huang, Courtney Mann, and Bradley Clymer

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

***Abstract* — Sensitive or intruder-vulnerable areas are, by definition, in need of some form of defense. In such cases, it is potentially desirable to remove human physical presence from the defense, without eliminating human influence. Thus, it was decided that a turret should be made to guard these types of areas, which could be remotely controlled with only simple user input. Due to the variety of potential applications for this technology, scalability was important: lasers may be used for "painting" or specifying targets, and more forceful firing mechanisms may be used when necessary. The system designed is modular, and may be easily adapted to utilize a multitude of firing platforms.**

***Index Terms* — Computer vision, defense industry, intelligent robots, microcontrollers, servomotors, wireless communication.**

I. Introduction

The motivation for the creation of the defense turret was a desire to build a human-selective defense platform that does not expose the operator to direct risk, while minimizing the need for both the training time and manual dexterity to mount a working defense in a tactically important or personnel-sensitive area.

The system takes in images from a wireless camera via a Wi-Fi connection, and processes them on the touchscreen tablet-PC. They are presented to the user, and he or she is prompted to simply touch the target of their choice, at which point the system fires upon the selected target; this is the entirety of the user experience; the interface was specifically engineered this way to minimize complexity and eliminate training; in an emergency situation, the difference between successful defense and compromised security may depend entirely upon the speed and simplicity with which a user may adapt to the interface.

The turret system is intended to be used for defensive purposes, and as such the initial design has been made with a laser pointer rather than a mechanical projectile-firing mechanism. However, scalability was deemed a priority for this project, so the housing, armature, and servos were selected and designed to accommodate heavier projectile mechanisms and/or larger optical or audio deterrents.

II. Overview

The overall flow of the system operates relatively simply. First, visual information is received by the camera, a Cisco WVC80N. This camera compresses and transmits the 640x480 video stream to the tablet, where an OpenCV-based software interface runs on the Acer Iconia tablet-pc. The tablet processes the visual information, and presents it to the user.

The user proceeds to select a target by touching its location on the screen, and the U.I. software calculates the center-pixel of the area touched; this pixel information is transmitted to the microcontroller through XBee, and the on-board software turns those coordinates into Pulse-Width-Modulation signals, which are the control signals that determine servo angle.

Simultaneously, a digital HIGH logic level of 5V acts to turn on the laser, indicating firing to the user; 500ms later, the laser logic is turned LOW again, and the firing is completed.

III. Mechanical Design

The mechanical design consisted of the turret construction itself, as well as the casing for the electrical components.

A. Turret Armature

The turret armature shown in Figure 1 is machined aluminum, capable of both pan and tilt motion. The motion is achieved by two servos, controlled by the group-designed Atmega-328 based microcontroller. Its lightweight construction is easily controllable by the servos, while sturdy enough to allow for heavier firing

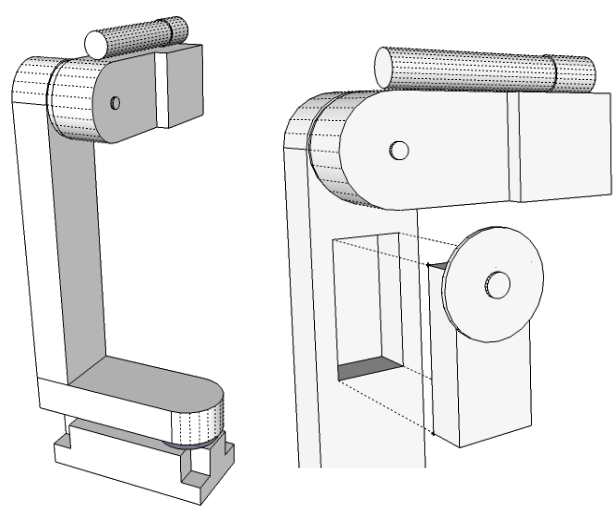


Fig. 1. Turret armature, and detail of servo insertion

mechanisms. This extends to the servos, which normally only draw about 100mA of current, but can max out at 500mA each when required 12.1kg/cm of torque; this is typical of what might be produced by the rotation of a paintball gun, for instance.

B. Hardware Housing

The housing for the project, Figure 2, was constrained by requirements of durability and accessibility, and visibility. In terms of durability, it needed to be able to

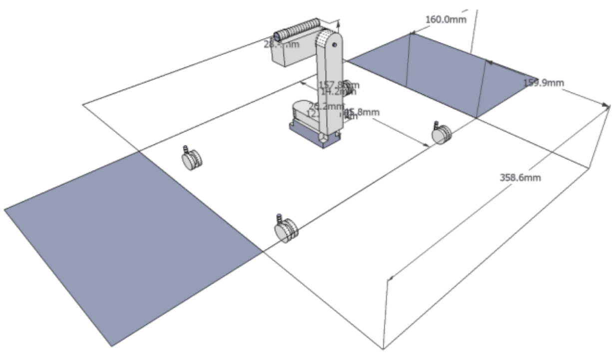


Fig. 2: Deployed hardware housing

withstand the torque applied by the servos to the armature, as well as frequent transport and disassembly.

To address the durability requirement, a material with a strong tensile strength but low brittleness was required; 8mm Lexan was chosen. Accessibility was of primary concern both because of the necessity for the project to be worked on after it would be initially constructed, and because the evaluators might desire to inspect the internal components more closely than a cursory, external visual exam could allow. The solution was a hardware housing as shown in Figure 3, with cotter-pinned construction for quick deployment and inspection.

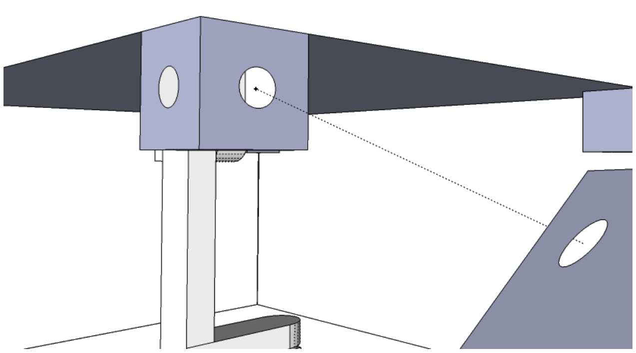


Fig. 3: Corner of lid and sidewalls, showing cotter-pinned construction

IV. Electrical Hardware

The hardware block diagram, shown in Figure 4 below, illustrates the basic connections of the hardware components. Initially the wireless camera used to capture the visual target field relays its information wirelessly to the Windows tablet through Wi-Fi. Once the tablet has processed the camera's image for target tracking, it sends the coordinate information to the XBee explorer to determine how far the gun must be moved. These instructions are transmitted wirelessly to the microcontroller, which interprets them into servo movements and firing commands. The system is powered through AC.

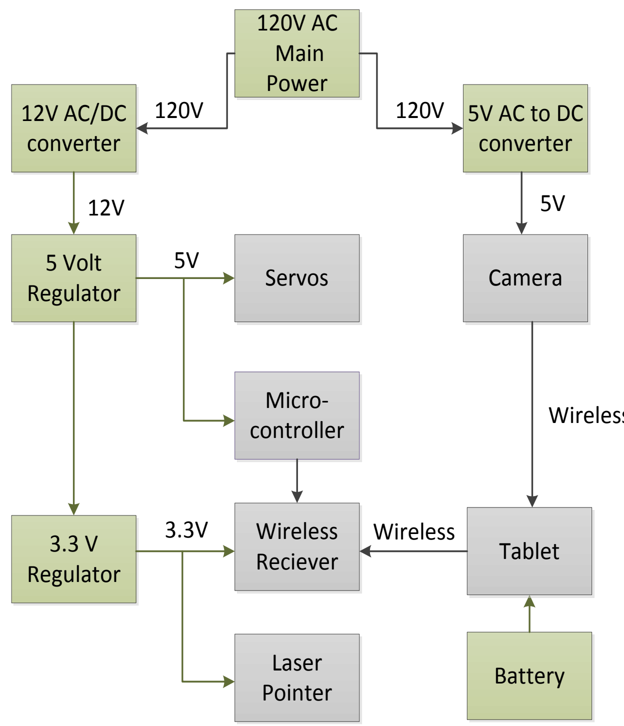


Fig. 4: Hardware Block Diagram

A. Camera

The chosen camera was a Cisco WVC80N, an IP camera which offers a resolution of 640x480 pixels [2], sufficient for use by the computer vision portion of the system. Its hardware casing has a center-mounted pin, which easily mounts on the pan-axis of the armature. This is necessary to reduce parallax errors in vision processing, and eliminate the need for intensive range calculation.

B. Tablet

One of the biggest challenges of the project was the integration of the tablet into the system. The tablet acted as the main computational source for the image processing. Because of the complex nature of the calculations, a large amount of processing power was needed, and since the tablet was already available for the user interface, it was decided that the task of target acquisition and tracking could be assigned to it as well. This allowed the group to devote the onboard system processor to the simpler tasks of motor control and firing commands. The specifications for the chosen tablet are listed in Table 1 below.

The tablet initially receives input from the camera in the form of captured frames. It then analyzes the images and performs the programmed operations to successfully detect and track moving targets, as explained in the software. A user application is open to display the processed images to the user, which consists of the incoming frames with the enhancements added to them by the program, as well as a binary image of the detected target. In the incoming frame, the targets are outlined in specific colors, and the turret fires upon the chosen targets. When the user has selected their targets, the program calculates its position based on the location of its centroid, and sends this information wirelessly to the microcontroller, which then interprets the signal into commands for the movement of the servos. In addition, it has the capability to fire upon manually chosen stationary targets.

In order to accomplish these tasks, the tablet had to meet certain requirements. First of all, it had to have the capability to interface wirelessly with both the camera and the system processor, where it sends the objects' locations so the servos can be properly oriented. The tablet had to also be easily programmable, since an application needed to be created to serve as the user interface to select the targets. Another factor was the necessity of a touch screen interface, since one of the primary project goals is to create a touch-operated system. Additional considerations included computing power and processing speed, which were important for processing the images with minimal delay.

There were a variety of tablets and tablet systems that were looked into for use. Since it would be involved in both the user interface and the image processing computations, concern for the requirements of both of these processes had to be taken into account. This means a high quality display for viewing of the video streaming from the camera, as well as adequate memory and computing power for processing the images. The systems brought into considerations were Apple, Android, and Windows 7. Windows can be programmed in C++, which would be compatible with the OpenCV functions. Android is mostly Java based, however it is also possible to run programs written in C or C++. Apple, however, uses Objective-C language to write programs in. This restriction quickly eliminated the Apple iPad from consideration, leaving the choice between the two remaining options. The Windows tablet had the advantage in that the group would be using OpenCV on a Windows-based PC before moving it to the tablet, so having the same operating system on both would greatly reduce the

TABLE I
TABLET SPECIFICATIONS

Feature	Specification
Operating System	Genuine Windows® 7 Home Premium 32-bit
CPU and chipset1	AMD C-Series dual-core processor C-50 (1 MB L2 cache, 1 GHz, DDR3 1066 MHz, 9 W) AMD A50M Fusion™ Controller Hub
Memory[1, 2,]	Up to 2 GB of DDR3 onboard system memory
Display1	10.1" HD 1280 x 800 resolution, high-brightness (350-nit), 146 PPI Acer CrystalBrite™ LED-backlit TFT LCD with integrated multi-touch screen, supporting finger touch and image auto rotation Wide viewing angle up to 80/80/80/80 degrees (up/down/left/right) Mercury-free, environment-friendly
Storage	Solid state drive 32 GB or larger, with mini-SATA (mSATA™) interface connector[1, 5,]
Wireless and networking	WLAN Acer InviLink™ Nplify™ 802.11b/g/n Wi-Fi CERTIFIED™ Acer InviLink™ 802.11b/g Wi-Fi CERTIFIED™ (available only in Russia, Pakistan, Ukraine) Supporting Acer SignalUp™ wireless technology WPAN:1 Bluetooth® 3.0+HS LAN: Fast Ethernet on the dock
Dimensions and weight	Dimensions 275 (W) x 190 (D) x 15.95 (H) mm (10.83 x 7.48 x 0.63 inches) for the tablet 275 (W) x 190 (D) x 11/19.5 (H) mm (10.83 x 7.48 x 0.43/0.77 inches) for the dock Weight 0.97 kg (2.14 lbs.)9 with 3-cell embedded battery for the tablet 0.61 kg (1.34 lbs.)9 for the dock

changes that would need to be made. On the other hand, the Android operating system is much more popular than Windows as a tablet platform, and the large amount of support information available to developers make it an appealing choice for beginning app builders.

Ultimately, the group selected Windows 7 as the chosen platform, due to the familiarity of the operating system, as well as the ease of implementing a wireless webcam compared to an Android system or Apple iOS. Microsoft Visual Studios 2010 was utilized to create the tracking program, with external libraries included to increase program functionality, including OpenCV 2.2, cvBlobs (created by Cristobal Carnerolinan), and a serial library for Windows in C++ (created by Ramon de Klein). The tablet used in the project was the Acer Iconia, which met both our graphics and processing requirements.

C. Wireless System

Network acquisition time, transmission range, and data rate were the major factors that were taken into account while deciding which protocol would be most sufficient for the project. Since this project was meant to be in real-time, network acquisition time became crucial. For this project to succeed, the transmission range was determined to be at least 30m, data rate had to be at least 15bps, and the network acquisition time needed to be less than 50ms. Based on the complexity of implementation and project requirements, the group decided to use XBee.

XBee is a protocol that can directly be embedded into various applications at the frequency 2.4GHz and link with multiple nodes. It has low power consumption, low network acquisition time, and low data rate. It is the most suited protocol for monitoring and controlling applications when no massive data information needs to be transmitted. Since only coordinates of the designated target were sent to the microcontroller, XBee's low data rate and its capability of transmitting periodic and intermittent data were sufficient for the specific part of the project. In order for the tablet to communicate with the microcontroller, an XBee USB explorer was connected to the tablet and the module was set to be XBee coordinator. The other XBee module which was communicating to the microcontroller on the PCB was reconfigured to be XBee Pro Router/ End device. Table 2 below shows the XBee specifications.

D. Microcontroller

At the core of the mechanical portion of the system, control is achieved by the Atmega-328. Around this microprocessor, the group designed a printed circuit board, onto which a large portion of the components from the Arduino were loaded.

TABLE II
XBEE SPECIFICATIONS

Specification	XBee 802.15.4
Topology	Point-to-Multipoint
Frequency	2.4 GHz
RF line of sight range	90m
Transmission Power	0dBm
Receiver Sensitivity	-92dBm
RF data rate	250Kbps
MSRP	$29.95

Among the many viable options in this field, the group had narrowed the choices down to Texas Instruments' MSP430 and the Arduino Uno, which contains the Atmel ATmega328 as its core processor. The MSP430 was a strong contender, mainly due to the fact that the group was in possession of a few already, but in the end the Arduino was chosen for a number of reasons. First of all was its 'ease of use' factor. Besides the fact that its programming environment was beginner-friendly, the software and hardware were both well-documented, and there existed numerous pre-built libraries that greatly helped in the coding process. Of all the boards in the Arduino family, the Arduino Uno was singled out because it contained all of the features that were needed without too many extraneous ones. According to the datasheet, it has 6 analog inputs and 6 digital input/output pins, which were used to connect the servo motors, in addition to a USB connection. For memory storage, it includes 2 KB of SRAM, 1 KB of EEPROM, and 32 kb of flash memory, although of that 0.5 kB are used by the bootloader to upload programs onto the board [1]. Table 3 given below summarizes the main features of the board. Another key factor in the decision was the necessity of wireless communication between the board and the tablet. The XBee shield for Arduino was designed to interface well with the Arduino Uno, but is not compatible with many of the other Arduino boards, eliminating them as possible options.

E. Servo Motors

Servos are controlled by sending them a pulse of variable width within duration in a closed loop system. A servo is essentially a positionable motor, which takes two inputs: the current position and the desired position. The control wire is used to send this pulse, which has three parameters: minimum pulse (ground), maximum pulse (power), and a repetition rate (command). Given the rotation constraint of the servo, neutral is defined to be the position where the servo had the exact same amount of potential rotation. Different servos have different constraints on their rotations but they all have the same neutral position, which is always around 1.5 milliseconds. Servos are commanded through "Pulse Width Modulation." Basically, the width of a pulse defines the position. To move to motor 90 degrees and hold in that position, a 1.5 ms pulse is required to be sent to the servo. Also, the command has to be send every 20ms or at a frequency of 50Hz. It is possible to damage a servo if commands are sent at an improper frequency. Unlike stepper motors which can have changing speed by varying the voltage applied to drivers, the majority of servos have fixed speed rate in degrees/second.

To decide which servo motors would be most successful for this particular case, it was necessary to determine how much torque would be needed and how fast the servo system operates. Another factor to be taken into account was the cost. Since the group used laser painting instead of a paintball marker, the system needed a little torque and responsive servos for the system to achieve its best performance After comparing the internet prices and performances of motors, the group decided to choose two HS 325-HB BB Deluxe digital servos motor from servocity.com at the price of $12.99 each. These motor can handle torque up to 51 oz/in or 3.7kg/cm and has a standard operating voltage from 4.8V to 6V with speed from 0.19 second per 60 degrees to 0.15 second to 60 degrees. Since the weight of the laser painting implemented in the system was ultra-light, this motor was able to provide enough torque so that it did not burn out under the stress of responsive tracking of designated targets.

TABLE III
ATMEGA328 SPECIFICATIONS

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA

The Arduino's programming language made PWM easy to use; it simply called analogWrite(pin, dutycycle), where the duty cycle was from 0-255 and the connected pin was one of the PWM pins (3, 5, 6, 9, 10, or 11). The analogWrite function provides a simple interface to the hardware PWM, but does not provide any control over frequency. Despite the name analogWrite, the function output is a digital signal. The Arduino Servo library functions were utilized in the project.

F. Laser Pointer

Unlike a paintball marker, which involves calculating the gravitational effect on the shooting angle, a laser pointer was implemented to illuminate the designated target by a bright spot of light for practical demonstration purpose. The requirements for a laser pointer were simple. The range of the laser pointer had to be at least 30 meters and the brightness needed to be noticeable and easy for the user to see even in day time. The laser pointer was connected to one of the microcontroller digital pins, which outputs 5V to turn on the laser pointer when a firing command is received (HIGH).

H. Power

The system contains a wide range of components. Table 4 summarizes the expected peak power requirements for the turret portion of the system, since the charging of the user interface will not be relevant during times of operation.

Since these components require large amounts of current (and therefore power) at any given moment, the system required either a large battery (which would inhibit transport) or the use of AC from a generator or wall outlet (which limits installation location and make the project susceptible to interruptions of service from the power grid).

The microcontroller is normally powered by an AC adaptor which outputs 9V DC. Laser pointers – most notably the model which the group initially selected – are typically powered by two AA batteries, which are 1.5V cells in series. Thus, a 3V DC adaptor, properly wired, could serve as a suitable power supply for this device. Additionally, the servos were controlled by the

TABLE IV
POWER REQUIREMENTS

Part	Peak Voltage	Peak Current	Quantity	Peak Power
Servos	6.00 V	500.00 mA	2	6.00 W
Atmega328	12.00 V	50.00 mA	1	0.60 W
Laser Pointer	5.00 V	33.33 mA	1	0.10 W
Totals	23.00 V	1.08 A		6.70 W

microcontroller, but this device does not supply nearly enough current to operate them, so they needed some external form of power.

From the above specifications, it was determined early-on in the project that AC should be the power source of choice for the project. The user interface tablet that would allow control of the system would have battery power, but would fundamentally be charged via AC; and more importantly, the servos that control the aiming of the targeting system would draw excessive amounts of current for short periods of time, which would either require large lithium-ion batteries, or readily available DC after conversion from wall-AC. Since standard wall AC is readily available, the group opted to use it.

Overall, the project used a consumer-grade power strip. Into this was plugged the adaptors for the Arduino and the driver board. The Arduino ran on a 9V DC “wall wart” style adaptor with a 2.1mm barrel plug, utilizing a positive tip. The servo motors required 2.857A of current and 5V through a separate AC/DC adaptor. The laser pointer did not require any extra power source, as it was actuated through the Arduino.

The User Interface Tablet was not powered by the system during operation – it was charged separately prior through the tablet’s included 36 Wh 3260 mAh 3-cell Li-polymer embedded battery.

V. PCB

The PCB was based off of the Arduino Uno, which was utilized in the project. To minimize the budget, only needed components were kept on the project’s PCB. There were four major systems: the wireless module, the microcontroller, and two voltage regulators. The voltage regulator, NCP1117ST50T3G from OnSemiconductor, was used, since both the PCB board and the Atmel Atmega328 operated at 5V. The other voltage regulator, LP2985from TI, was implemented for the wireless module, which was powered at 3.3V. The board was designed using the free version of EAGLE and manufacture by Advanced Circuits. The size of the board was restricted to be 4 x 3.2 inches due to the limitations of the free version. Since the size of the board was not spacious, the space arrangement of the PCB became important. In order to have an efficient PCB layout, understanding wire connection between each unit was essential. The Atmel Atmega328 in particular had multiple connections, so, decent knowledge of its architecture was required before designing the PCB. Figure 5 below shows the electrical schematic of the PCB.

VI. SOFTWARE

Targeting acquisition involves the visual processing of the images obtained from the camera to determine the existence and the location of a target. The software must be capable of tracking individual moving targets at a time.

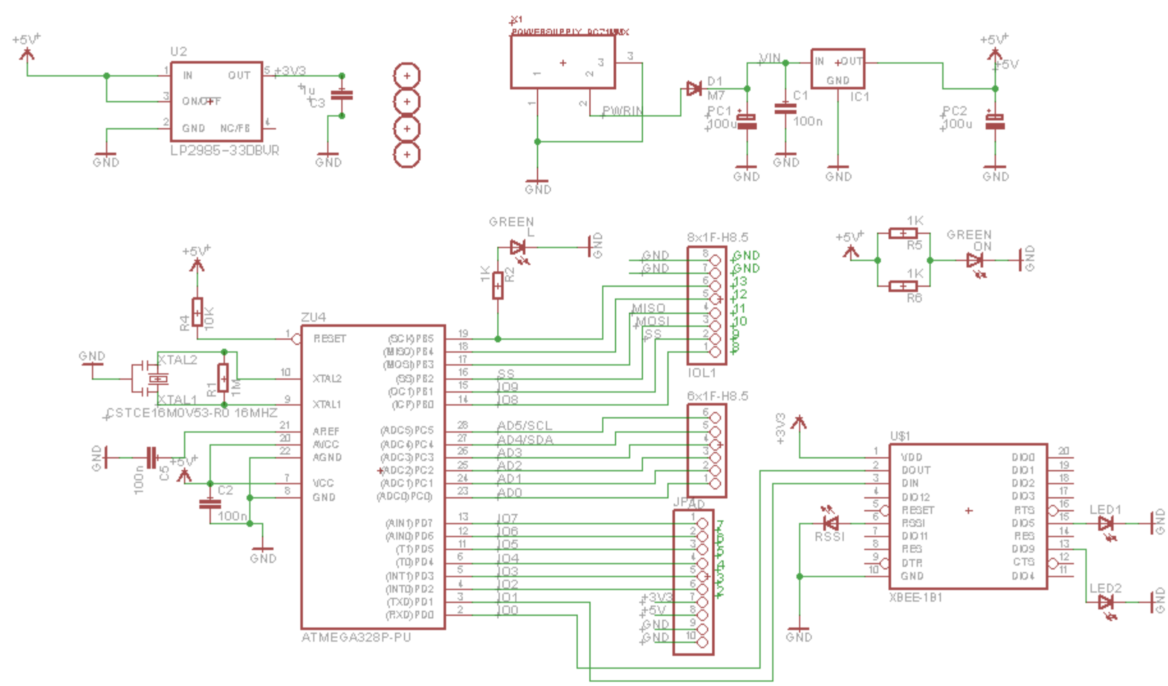


Fig. 3: Electrical Schematic of PCB

First it must recognize that a new subject in the field of view has been selected by the user for targeting, at least within the turret's 30m range, such as a person walking by. Then it has to find the location of the target, so the motors can be directed where to aim.

It was observed that supplementing the tracking program with a prebuilt library, such as Intel's Open Computer Vision or Aforge.NET library, would greatly reduce the amount of code needed and simplify a number of intricate programming details. Extensive research was done to measure the capabilities of each library, and determine which would be the best fit for the RTCDT. Prior to choosing the tablet, the group narrowed the choice down to two libraries. AForge.NET is based in C#, and has a lot of filters, which make it an excellent for features such as edge detection and thresholding. It is also widely held to be the easier of the two to use. Unfortunately, it was unclear how exactly this could be implemented on an Android system. OpenCV, on the other hand, is geared towards C/C++ developers, and is useful for more sophisticated image manipulations, such as facial recognition system, gesture recognition, object identification, and motion tracking. It also includes wrappers for languages such as C#, and Java, which will be useful for the transition to the Android platform. One of the other benefits of using OpenCV is the existence of a large support base due to its popularity among developers, where the group can troubleshoot problems and find solutions for common issues. Ultimately, OpenCV was determined to be the best fit for the project.

Object detection was implemented with the use of a color recognition program. The user selects an object on the screen that they would like to track, and the program will target that pixel and store its value, which represents the color. This value will then be fed into the program, along with a specified threshold, which serves to filter out any pixels that fall outside the given range. A new image is created that contains a thresholded binary image, where all the pixels of the selected color are converted to the maximum value (white) and the rest of the pixels, which fall outside the threshold are given the minimum value(black). This image is displayed to the user for debugging purposes. A smoothing filter is applied to the incoming frame using a median filter, which analyzes the neighboring values of each pixel and sets it to the calculated median. Next each image is converted from RGB to HSV. Another included library, cvBlobs, detected consecutive pixels of the chosen color as blobs, and represented them to the user by outlining them in colored rectangles. Additionally, a "filter by area" function was applied so that only blobs of a minimum size would be read by the program, thus limiting errors. An example of the user interface is shown in Figure 6 below.

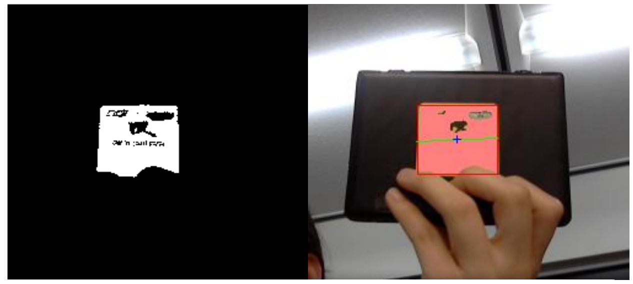


Fig. 6: Software Example

The position of the tracked target is found by using moments. By comparing the difference of location between the current frame and the previous frame, the object can be accurately tracked. This will inform the microcontroller how much it needs to move the servos to maintain its aim with the target. If the gun is in its default position facing straight ahead, the program must compare the target's position with this location in order to orient the gun correctly. The coordinates are displayed to the screen for the user. Finally, an included serial library is used to send the x and y coordinates to the serial port, where an XBee explorer transmits the information to another XBee connected to the microcontroller, which in turn converts the coordinates to PWM signals and outputs it to the servo motors for tracking, along with a firing command to the laser pointer.

Another feature of the system is the ability to select stationary targets. When a user has selected a random point in the viewing window, the system must recognize the specified point as the new target. As with the moving targets, the location must be calculated, however since the object does not move, the 'tracking' is unnecessary, so this calculation only needs to be done once. The coordinates are found and sent to the XBee transmitter, along with a firing command, which the microcontroller receives and interprets by moving the turret to the specified point and flashing the laser pointer.

VII. Conclusion

The integration of computer vision processing, wireless communication, and motor control – along with an easy-to-use and intuitive user interface – provides a nucleus for basic intelligent systems, many of which are vital to newly-graduated electrical engineers. An intelligent approach to the implementation of these systems in a turret of the type shown herein enables a robust, upgradable, and powerful platform for remote defense.

The group discovered – as was likely the primary intent of the of the CECS faculty – that the development of such a project represents an enormous coalescence of the sum-total of engineering skills learned thus far, along with a significant accumulation of new abilities and experience.

ACKNOWLEDGEMENT

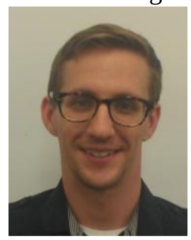
At the conception of this project, it was evident that the financial burden would not be light, due to the large amount of components required and the complex technology needed to implement such a design. Workforce Central Florida allotted financial support for projects in the categories of homeland or cyber security, renewable and sustainable energy, biotechnology, and digital media or modeling. This gave the group the choice to submit the proposal to WCF under the Homeland Security label. The group chose the latter option, and the budget was approved for full funding from the company. Szu-yu Huang would like to acknowledge the support by WORKFORCE CENTRAL FLORIDA. Brad Clymer would like to acknowledge the support by WORKFORCE CENTRAL FLORIDA. Ms. Mann would like to acknowledge the support by WORKFORCE CENTRAL FLORIDA.

The industry mentor chosen to meet the Workforce Central Florida requirements is Dr. Robert Muise. Dr. Muise is a Senior Staff Engineer at Lockheed Martin Missiles and Fire Control. He holds a PhD in mathematics from the University of Central Florida. He is responsible for leading engineering teams in applied research in the areas of image processing, compressive/computational sensing, automatic target detection/recognition, and image/data compression. His expertise is in algorithms for signal/image processing, computational linear algebra, and integrated sensing and processing. Dr. Muise holds two patents, has published many journal and conference papers, is a member of SIAM, and a senior member of IEEE. He is also part of the faculty in the University of Central Florida Department of Mathematics.

In addition, the group asked Dr. Niels da Vitoria Lobo to act as an unofficial mentor to aid with issues in the computer vision portion of the project. Dr. Lobo received his B. Sc. (Honors) degree in Mathematics and Computer Science from Dalhousie University, Canada, and the Ph.D. in Computer Science from the University of Toronto.

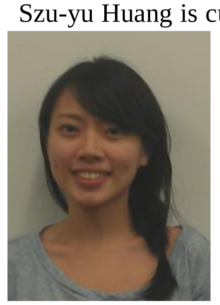
BIOGRAPHIES

Brad Clymer is currently a senior at University of Central Florida and will receive his Bachelors of Science in Electrical Engineering in May of 2012. His interests include biomedical nanotechnology and micro electro mechanical engineering. He begins his graduate studies at the University of Central Florida in fall 2012, focusing on Surface-Acoustic Wave devices.



Courtney Mann is currently an electrical engineering senior at University of Central Florida. Her interests lie in communication and controls. She is graduating in May 2012. She is planning on going to graduate school and pursuing a working career in the electrical engineer profession.

Szu-yu Huang is currently an electrical engineering senior at University of Central Florida. Her interests include optics and biomed electrical engineering. She is graduating in May 2012. She is planning on continuing studies for master degrees in Biomed Engineering and pursing a working career in electrical engineer profession.



REFERENCES

[1] *Arduino*. (2011). Retrieved 11 2, 2011, from Arduino: http://arduino.cc/en/Main/ArduinoBoardUno

[2] LINKSYS by Cisco. (n.d.). *Wireless-N Internet Home Monitoring Camera*. Retrieved November 23, 2011, from LINKSYS by Cisco: http://downloads.linksysbycisco.com/downloads/datasheet/WVC80N_DS_V10_NC-WEB.pdf