CHAPTER ONE

INTRODUCTION

1.1 General Concepts

Throughout the history human passed through three waves. The first wave is agricultural wave in this wave land was the source of wealth and power guide who owns the land he ruled and controlled kings and princes who ruled on the ground in peoples and human beings. The second wave is oriented industrial began this wave since the seventeenth century formed the machine and raw material source of wealth for this router, and began to capital necessary to get access to more of the machines. West began this wave controlled the machine and the headwaters of wealth across the colonies occupied West fought Based on this wave means used in the two world wars, one more destructive weapons, namely nuclear weapons. The third wave is technological wave, in this wave comes knowledge and information in the first place as a source of wealth. Intellect to fully resolve the place of muscle strength and thus was both oriented impact on your human evolution. It was both oriented and means and its own language. Sickle in agricultural wave versus harvester machine in industrial wave versus food across-grain food in the laboratory in technological wave. Dagger in the wave farm as a weapon, in exchange for the gun-oriented industrial, through to laser in the technological wave. The initial idea for this project came from a military based video game where a player could put down a “sentry gun” and it would then automatically target, track and shoot at enemy players without the player having to supervise the system . To our knowledge, this system does not readily used by the military even though it has great potential. It parallels the purpose of the MQ-1 Predator made by General Atomics; an UAV used in military applications. The system would replace the active functions of an armed guard while keeping a human life out of harm’s way. Since this system would replace a human, we wanted to make it as accessible as possible which lead to idea of having the system remotely accessible and piloted by a user. To allow the user to have as much information about the place the system operates in, we wanted to add several environmental sensors can also log system location and weather conditions. This remote accessible data could allow for military planning for that

environment. Overall, we this system has a lot of potential in modern day military strategy and would help spare hundreds of our ally's lives. The Automatic Gun Fighter can autonomously and accurately track and shoot at moving targets regardless of lighting and weather conditions, while also allowing a user to remotely access and control the gun via remote computer. The remote controlling of the gun via computer provides the user with real-time video feed from the system and allows full control over the gun's pan and tilt functions, and gun triggering. These functionalities parallel the ergonomic characteristics of light weight, physically hardy and easy to operate. The system creates a reliable replacement for armed guards in harsh and hostile environments, ultimately sparing a life. [1]

1.2 Problem Statement

The protection of important places is getting more required these days but the cost of this protection is also high. By using an automatic garden system, we can save human life's and save money that can be spend as a salary for guarding.

1.3 Objectives

Our goal is to build a modified NERF sentry gun that implements a tracking system that can identify and engage targets. This tracking system will be implemented with a tracking system that relays information about possible targets to a microcontroller. The microcontroller will be programmed to take positional data from the tracking system and determine where the gun needs to be moved to and whether the target is able to be engaged.

1.4 Methodology

To achieve the Objectives of this Study, the following steps were implemented:

- Building simulation program using Proteus application.
- Using Arduino software program to build a code and to deal in Arduino.
- Building and testing the hardware circuit for the project.

1.5 Project Layout

This study consists of five chapters: Chapter One gives an introduction to the principles of the work, in addition its reasons, motivation and objectives. Chapter Two presents previous studies about some designs of gun fighters. Chapter Three gives the components of the project hardware and software. Chapter Four deal with prototype design and shows the experimental results. Finally, Chapter Five provides the conclusion and recommendations.

CHAPTER TWO

Gun Fighters

2.1 Introduction

Autonomous sentry turrets come in many different variations. From basic beginner hobbyist tutorials to expert level designs, there was a lot of previous information out there to learn the most effective ways of doing this project are and improving those designs. Instead of an airsoft gun used in this project, some designs used different weapons like paintball guns, Tasers, or real guns when done on the professional level. Researching past projects will allow us to view the success and failures of previous attempts and build upon the experience of others. After our group exploration of research, we were able to find a number of previous assignments: Remote Touchscreen-Controlled Defense Turret during fall 2011, Autonomous Targeting Sentry (ATS) during fall 2011, and Automated Targeting Proximity Turret during fall 2010. Reading through those reports we were able to come up with our own version of an automated turret and expand on their ideas. [2]

2.2 Remote Touchscreen-Controlled Defense Turret during fall 2011

Remote Touchscreen-Controlled Defense Turret was a laser turret project which monitored a designated area, highlighting multiple moving targets and sent the given information to a Windows tablet via Wi-Fi. The group displayed the moving targets through an Open-CV interface on the tablet by highlighting each moving target with different colored outlines. The user is given the option of isolating one colored target by pressing a button on the tablet. Next they calculate the centroid of the target, then send the information to an Arduino microcontroller which moves the servo motors following the target and firing a laser pointer. In order to ensure the laser is correctly pointed at the target their group used a PID controller applied in the servos. When given the coordinates of the target from the Arduino, the PID controller took that information then calculated and fixed the movement errors of the servos. That signal was then sent to the servos which moved the laser. Managing the imaging of objects on the tablet was done in three steps: object detection, object outlining, and object tracking. For object detection the group used a color recognition method. At first they used background subtraction but that was deemed too slow. With color recognition, the user selects an object on the screen, the program identifies the pixel and stores its color value. Then by setting a threshold for the color they were able to filter out objects of different color. Finally, they converted all the pixels within the threshold to white and the rest of the image to black. Next the group outlined the object and calculated the centroid. They chose to outline the objects with a

rectangle for ease of centroid calculation, but may not be ideal for certain shaped targets. The final step was tracking the object which was done by calculating the distance the centroid moved from frame to frame. This information was sent to an XBee radio module which used its library methods to convert and send the X-Y coordinates to the microcontroller. Finally, the microcontroller converts the coordinates to PWM signals and send it to the servos which aim at the intended target. [2]

2.3 Autonomous Targeting Sentry (ATS)

ATS was another Senior Design project similar to ours. Unlike RTCDT, this project attached a paintball gun to the turret. By using high definition web cams and OpenCV libraries they detected and tracked moving targets. This project had the ability to be controlled manually by the user through a connected laptop. Also when the turret initially sees a target they are given a warning to get out of the way.Where the RTCDT used an Arduino as their microcontroller, the group for ATS used a Texas Instruments MSP430. This allowed them to effectively communicate with the laptop using Microsoft's .NET Framework along with communication to the servo motors. The base structure for the turret was the front fork of a bicycle in a hanging position. The fork allowed the servos for movement for both the x and y axes. For the power supply the group used a 12V rechargeable battery which was sufficient for their needs. They attached two alarms to the project, one was simply a loud noise maker and the other was spoken audio played from speakers to warn humans.

Object detection was done with the use of several OpenCV libraries. With the use of color detection OpenCV methods, the group was able to distinguish their targets with greater accuracy. Like the previous group, they converted the RGB to grayscale for simpler tracking. Tracking was done by using OpenCV methods which took the targets from the object detection class, found the differences between every frame and calculated their velocity, direction, and range.

Manual control was done with a Manual Control Class created using Microsoft Visual Studios. The user interface is similar to ours with a video feed and buttons for rotating and firing the gun. They also had a button to initiate the warning siren. To ensure that not just anyone can access manual mode, the group implemented a facial detection and recognition class. By using more methods from OpenCV they were able to scan user's faces, sort through a database of authorized images, and determine if there was a match and allow access into manual mode. This project had a lot of schemes that the group is going for, especially a manual mode which was hard to come by during the research. Though this group did the project from a laptop and not a tablet we were able to come up with a lot of great ideas and narrow down the goals. [2]

2.4 Automated Targeting Proximity Turret (ATPT)

A third similar Senior Design turret project was ATPT which was a paintball sentry gun with the added feature of using a server to store the attack history. This project also used range detection, when a target came into the field of vision it calculated the distance they

were from the gun. When the objective came within a certain distance a warning signal occurred and if they did not leave then gun would fire.

Range detection was achieved by using an inexpensive laser pointer connected to a control board which was designed to specifically communicate the laser with the laptop. Three webcams were used for the image processing, one high definition and two low definition. The HD camera was mounted on top of the gun while the two low-def cameras were on the base. Their purpose was to keep track of the differences between frames, one focusing on the yaw and the other the pitch. Using AForge.NET libraries for motion detection and tracking and an Arduino microcontroller they were able to successfully detect motion and range of a person walking by.[4]

A database was required in order to store the history of information when the turret is active. The group kept track of every time a target was hit, taking a screenshot of the video feed in the process. All of this was done using MySQL language. The base of the turret was made of wood with the standard three servos for movement. There was also a manual mode option for the user.

This project was another one of the few we came across that had a manual mode. Most of the previous senior design projects used an Arduino microprocessor which the group is unable to use, so we had to research more for alternatives. Overall, going through these three projects, among others, we were able to learn what methods were effective and what were ineffective and not waste the time with. Likewise, we were given options for techniques that did work and were able to decide which one we would like to use.

2.5 Real Life Existents

There are many types of gun fighters such as : Samsung SGR-A1,Super aEgis II.

2.5.1 Samsung SGR-A1

The SGR-A1 is a type of sentry gun (a weapon that fires autonomously) that was jointly developed by Samsung Techwin (now Hanwha Aerospace) and Korea University to assist South Korean troops in the Korean Demilitarized Zone. It is widely considered as the first unit of its kind to have an integrated system that includes surveillance, tracking, firing, and voice recognition. While units of the SGR-A1 have been reportedly deployed, their number is unknown due to the project being "highly classified".

2.5.2 Super aEgis II

In December 2010, the South Korean firm DoDAAM unveiled the Super aEgis II, an automated turret-based weapon platform that uses thermal imaging to lock onto vehicles or humans up to 3 km away. It is able to function during nighttime and regardless of weather conditions. The system gives a verbal warning before firing, and though it is capable of firing automatically, the company reports that all of its customers

have configured it to require human confirmation. It is used at various facilities in United Arab Emirates, Abu Dhabi, and Qatar, among other places, and has been tested in the Korean Demilitarized Zone.

Figure 2.1: Super aEgis II

Figure 2.2: Samsung SGR-A1

CHAPTER THREE

HARDWARE AND SOFTWARE

3.1 Hardware Design

Overall, the Automatic gun fighter or (sentry gun) system is a project that has been demonstrated in the past, only the computer vision software is relatively new software. The hardware design on the other hand has similar components that have been completed for decades as Engineers design new up and coming technologies. The hardware design is split into 4 different sections, physical layout, power systems, arduino integration and communications. The figure 3.1 show the main block diagram of the system.

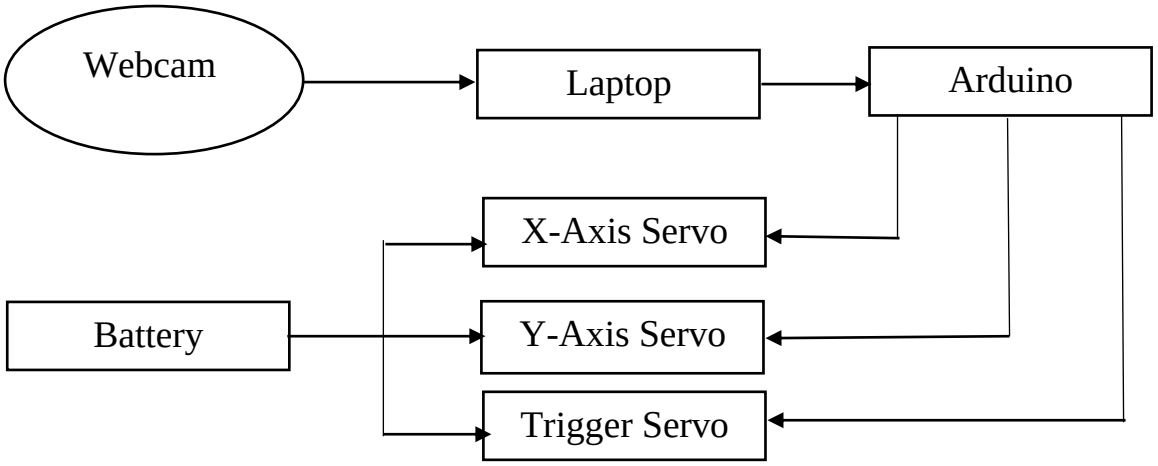

Figure 3.1: main system block diagram

3.2 The System Flowchart

When starting the program call up for the camera to determine the location of the target by turning the gun towards the target if the target has been located then the gun will fire if the target isn't located then the camera will start locating for target. The figure 3.2 show the system flowchart.

3.3 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based

on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for a lot of applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide. Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

3.3.1 Hardware

Although the hardware and software designs are freely available under copy left licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product.[5] Several Arduino-compatible products commercially released have avoided the project name by using various names ending in –Arduino.

3.3.2 Hardware structure of Arduino uno

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. Coming to basic outline of its specifications...Arduino/Genuine Uno is a microcontroller board based on the ATMEGA328P. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

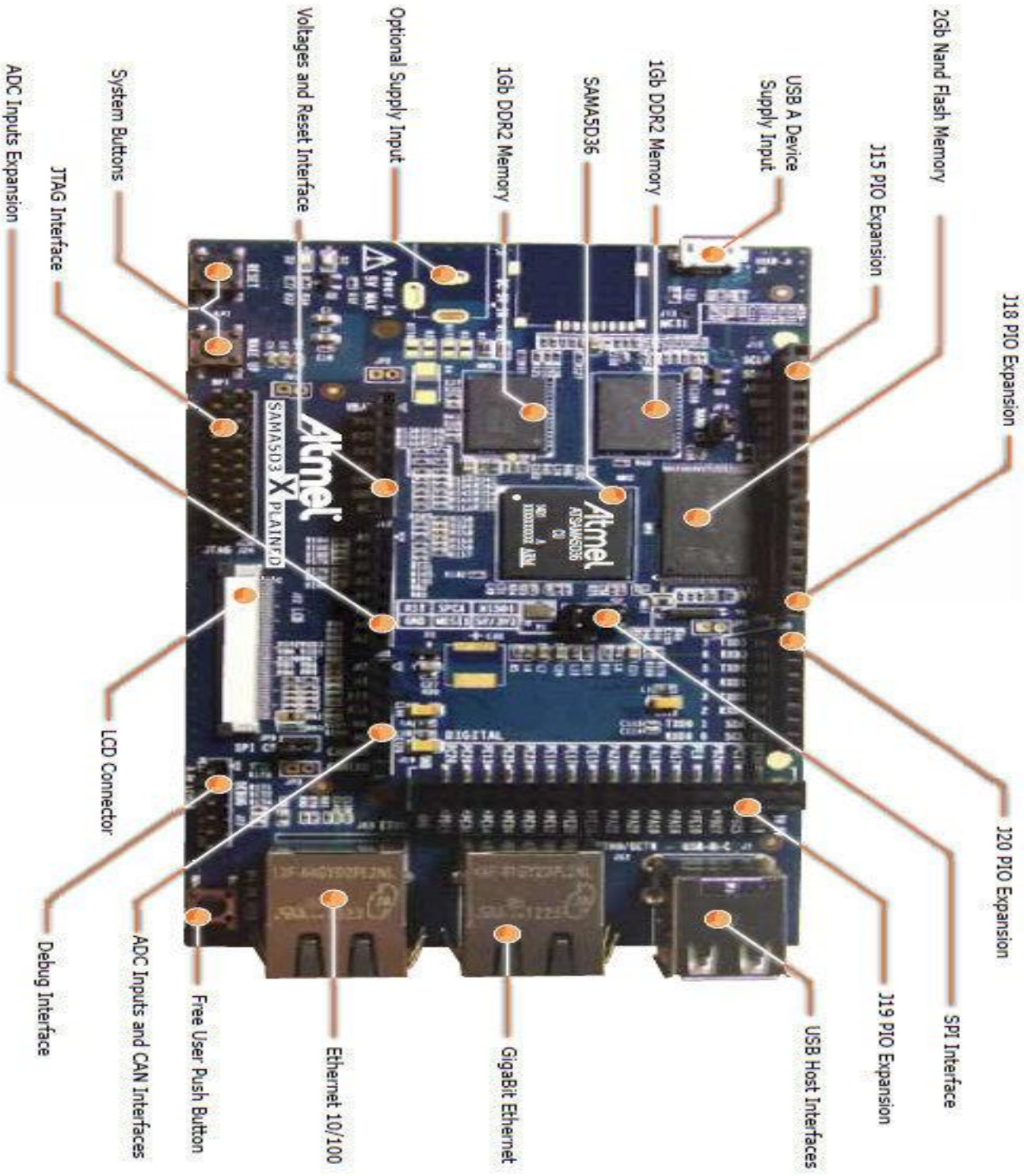

Figure 3.3: Hardware structure of Arduino uno

3.3.3 Schematics

Figure 3.4 show the Arduino uno scheme. Figure 3.5 show the pins and scheme of Atmega 168 board.

Figure 3.4: Arduino uno scheme

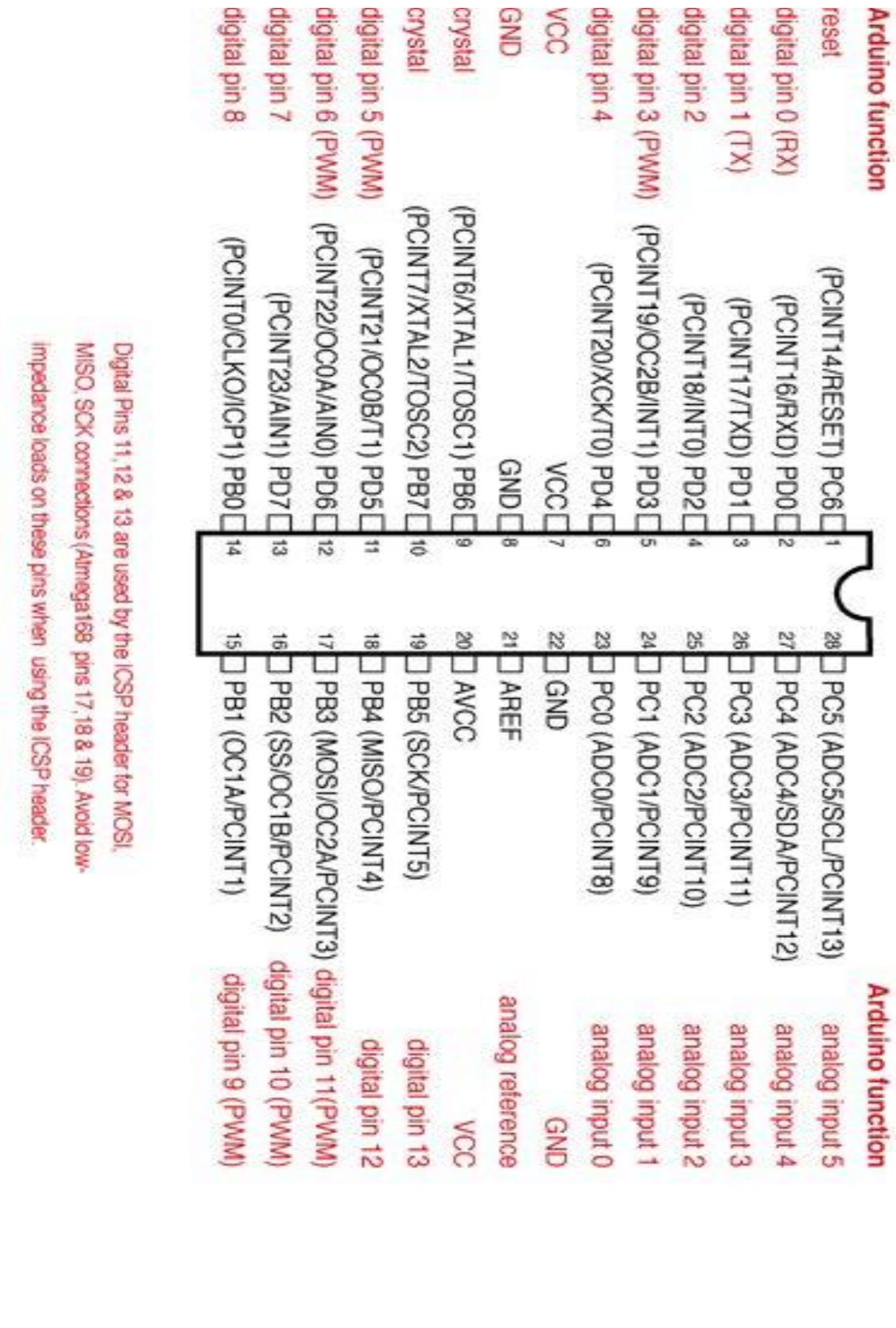

Figure 3.5: Atmega 168 pin mapping

3.3.4 Inputs and Outputs

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide

or receive 20 mA as recommended operating condition and has an internal pu ll-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. In addition, some pins have specialized functions:Serial: 0 (RX) and 1 (TX). U sed to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.There are a couple of other pins on the board:AREF. Reference voltage for the analog inputs. Used with analogReference().Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

3.3.5 Power

As we already discussed the first arduino board which can be powered via a USB connection. The Arduino/Genuino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector. The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows: Vin. The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.GND. Ground pins.IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

3.3.6 Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM.

3.3.7 Communication

Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

3.3.8 Basic features of Arduino

i. Low Cost: The plus point of Arduino board as compared to other microcontroller is that it has relatively less cost
ii. Cross-platform: Arduino can work on Linux, Windows, and Macintosh, while others are only for Windows.
iii. Simple and clear programming environment: Arduino programming is more flexible and easy, even beginners can benefit from it. As it’s based on processing environment, so professional programmers are also taking advantage of its features.
iv. Extensible and open source software: Arduino software was published as an open source tools and is available for extension. All types of source codes and libraries of Arduino environment are available openly, with which anyone can get used to Arduino.
v. Extensible hardware: The Arduino board is based on Atmel's ATMEGA8, ATMEGA328P PU and ATMEGA168 microcontrollers. It can be connected with any type of sensor, and circuits.

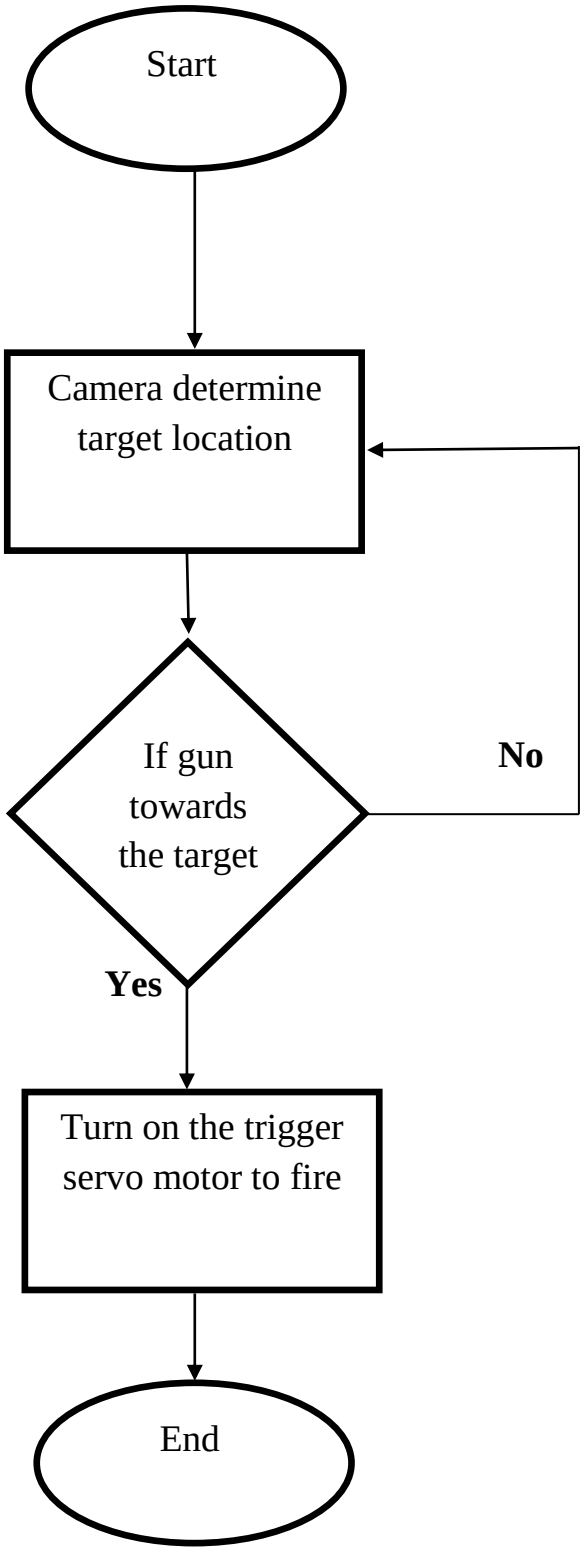

Figure 3.2: system flowchart

3.4 Servo Motor

A Servo is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. As the coded signal changes, the angular position of the shaft changes. In practice, servos are used in radio controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio controlled cars, puppets, and of course, robots.

3.4.1 Servo Motor Features

Model	Tower pro Servos MG996R
Dead Band	0.050 ms
Control System	Pulse Width Control
Working Frequency	20ms period / 50hz (Analog Control)
(RX) Required Pulse	3.0 ~ 5 Volt Peak to Peak Square Wave
Operating Voltage	4.8 ~ 6 V DC Volts
Operating Temperature Range	0 to + 55 Degree C
Operating Speed (4.8v)	0.200 sec/60° degrees at no load
Operating Speed (6v)	0.160 sec/60° degrees at no load
Stall Torque (4.8v)	9.4kg/cm
Stall Torque (6v)	11kg/cm
Motor Type	Brushed Motor
Potentiometer Drive	Direct Drive
Bearing Type	Output Bearing
Gear Type	Brass & Aluminum Gears
Case Material	Plastic
Programmable	No
connector Wire Length	32.0cm (12.6 inch)
Dimensions	40.7×19.7×42.9mm
Weight	55 grams (2.64 oz)

Table 3.1: Servo Motor Features

Figure 3.6: Tower pro Servos MG996R

3.5 Webcam

A webcam is a video camera that feeds or streams its image in real time to or through a computer to computer network. When "captured" by the computer, the video stream is also saved, viewed or sent on to alternative networks via systems such as the net, And email as an attachment. Once sent to a foreign location, the video stream may be saved, viewed or on sent there. in contrast to AN information processing camera (which uses an instantaneous affiliation mistreatment local area network or Wi-Fi a digital camera is usually connected by a USB cable, FireWire cable, or similar cable, or designed into constituent, like laptops.[4] The camera used in our design is the laptop webcam for streaming the video.

Figure 3.7: laptop webcam

3.6 Weapon

A gun may be an ordinarily hollow weapon or alternative device designed to discharge projectiles or alternative material. The projectile is also solid, liquid, gas or energy and will be free, like bullets and artillery shells, or captive like Taser probes and whaling harpoons. The means that of projection varies per style however is typically tormented by the action of pressure level, either made through the speedy combustion of a propellant or

compressed and keep by mechanical means that, operational on the projectile within AN open-ended tube within the fashion of a piston. The confined gas accelerates the movable projectile down the length of the tube transmission comfortable rate to sustain the projectile's travel once the action of the gas ceases at the top of the tube or muzzle. As an alternative, acceleration via magnetic attraction field generation is also used within which case the tube is also distributed with and a guide rail substituted [5]. A n-strike elite Nerf gun is the weapon used in our project it can shot up to ten shots in raw before loading.

Figure 3.8: n-strike elite Nerf gun

3.7 Power source

The power source was an essential component for powering the project system. It was needed since the system was portable and self-sufficient. The battery chosen to power was the 6V four aa battery pack. this battery pack had the ability to provide enough voltage to ensure reliable energy to the project system.

Figure 3.9: 6V four aa battery pack

3.8 Circuit Connection

The Arduino connects to the computer through a USB and its powering the Arduino itself and the servos are powered through a four pack battery by the ground and power wires and for the signal wires connected to the Arduino as fallow:

-x-axis servo signal wire goes to the Arduinos digital I/O pin 8

-Y-axis servo pin 9

- Trigger servo pin 10

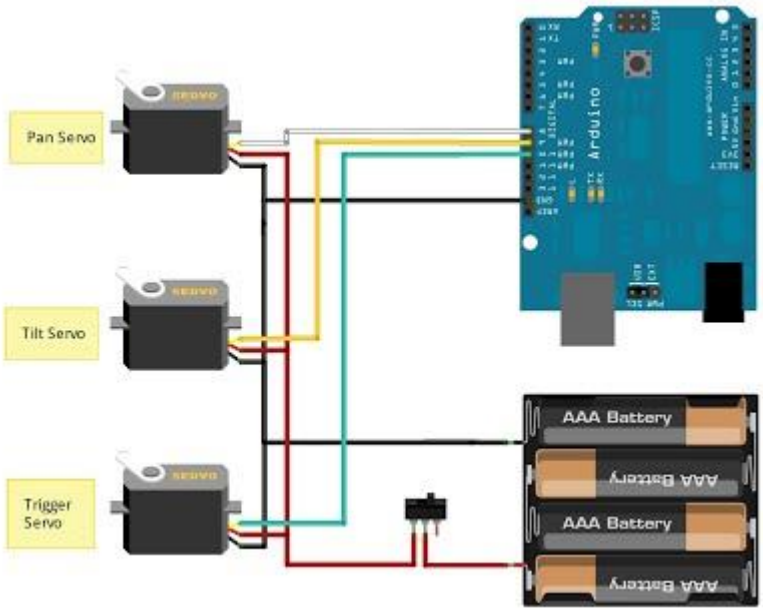

Figure 3.10: Circuit connection

3.9 Software system

For different aspects of this projects there will be different programming languages used. The first is called Processing which is an open source programming language and development environment. The language builds Java, but uses a simplified syntax and graphics programming model. It uses a sketchbook environment to create graphical and visual programs. Processing language is used for the object detection, tracking, and video streaming. The second is the Arduino IDE for tablet to microcontroller data transmission. By using the Arduino Uno microcontroller.

3.9.1 User Interface

using the programming language Processing and its JMyron library to implement the user interface as well as the tracking methods required for this project to work. The reason the Processing programming language was the top choice is because there, are a significant amount of open source tracking algorithms available that fit the needs of this project. The user interface has been developed in the Processing program language and displays a simple yet clear option screen for the user to view. The first step in creating a user interface was creating a main window that will house the commands, buttons, as well as the video feed from the camera. The first step to this was to create a main window that will be the lower layer of the UI. On top of this window sits a separate pane that houses the video input as well as the touch sensitive selector boxes and sliders that allow the user to switch between auto and manual modes, select weapon enable commands. A prewritten

Java GUI library was used to pull up button panes and image windows in the correct format. Once the GUI library was included, each UI object was created, labeled and formatted to work. The main function to create the user interface is described in Table 4.2 below.

Class:	GUI()
Description	Method the creates user control panel
Functions:	mainPanel: Main window used as destination of buttons and labels weaponOnOff_checkbox: Enables or disables the weapon's ability to fire scan_checkbox: Enables or disables the search and destroy mode autonomousOrManual_checkbox: Switches between autonomous and manual firing mode scan_checkbox: Enables or disables the search and destroy mode minBlob_slider: Creates the slider which allows the user to change the value of the minimum blob size for tracking detection

Table 3.2: Graphical User Interface Function

This user interface class will be called at the same time as the main tracking program and will have to launch before the sentry will activate. This function will depend on imported files such as the GUI library as well as the necessary microcontroller information so that button clicks can send commands that will affect the turret directly. The end result is in Figure 3.11 below. This is in autonomous mode, show the target box surrounding the moving target.

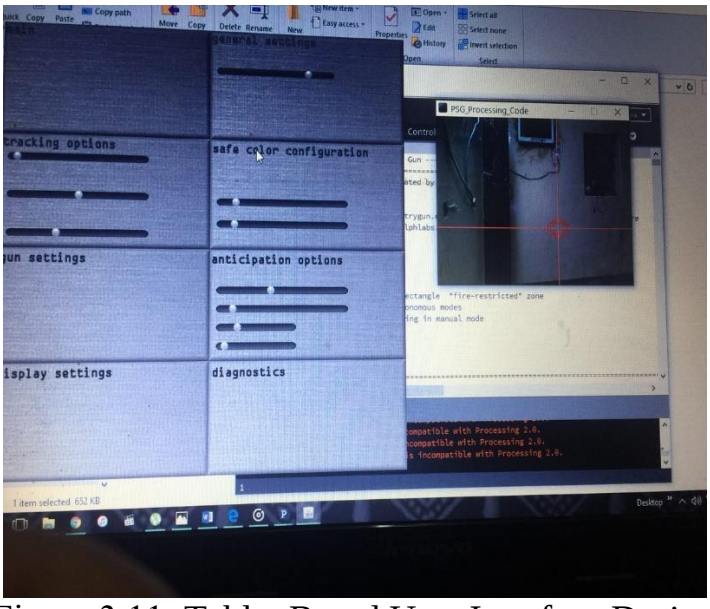

Figure 3.11: Tablet Based User Interface Design

3.9.2 Arduino IDE

To program the custom-built microcontroller, the Arduino IDE version 1.8.4 is used, which is the latest version currently. This open source IDE from the Arduino website is written in Java. It can be run on Linux, Mac, and most importantly Windows, including 32-bit. By using the default Environment. It makes writing code and uploading it to the board in an easy fashion that is aimed at beginner programmers. The language the user programs in C or C++. The IDE comes with a library called Wiring which makes many common input/output operations very simple to use. The basic Arduino code must have two function, setup and loop, described below:

- setup(): a function run once at the start of a program that can initialize settings.
- loop(): a function called repeatedly until the board powers off.

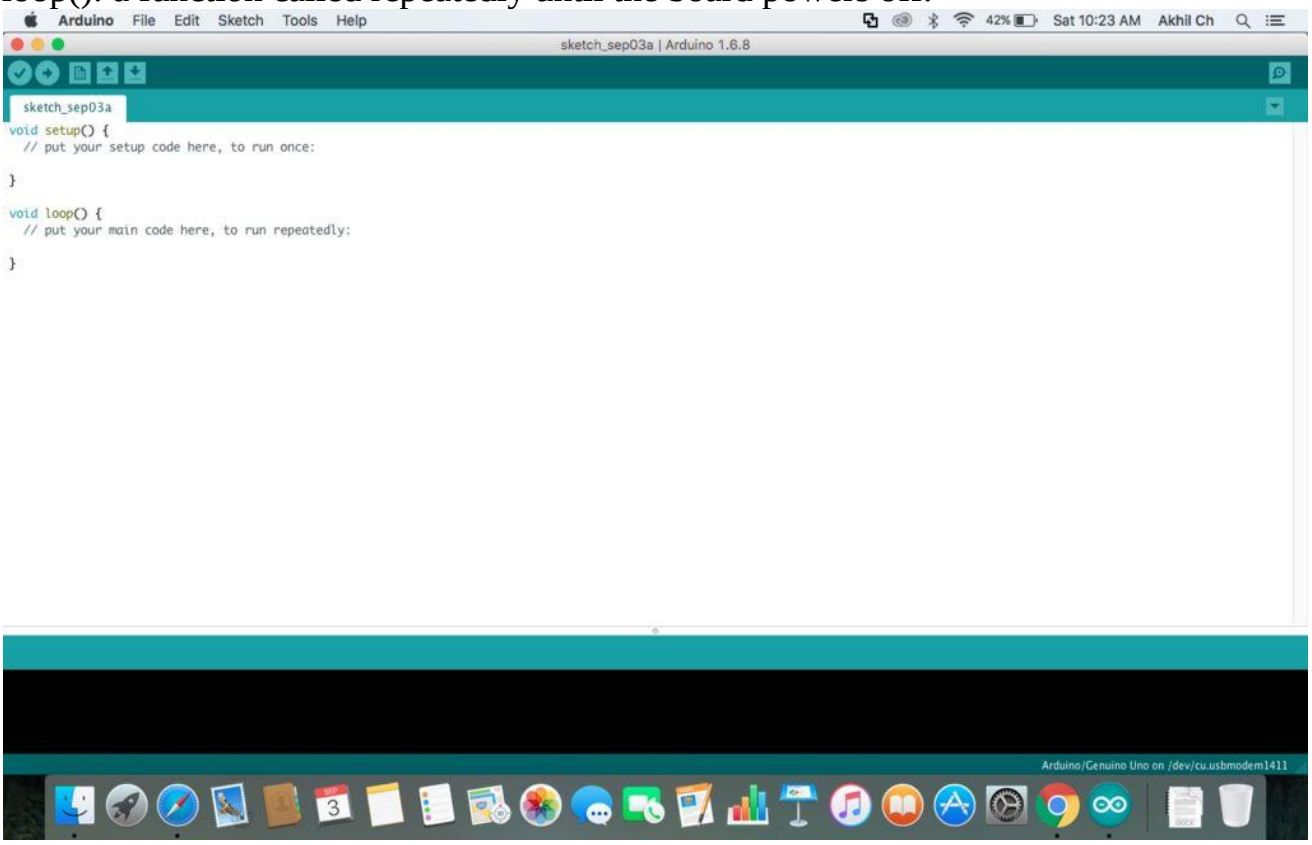

Figure 3.12: Arduino IDE

3.9.2.1 Programming

The Arduino language is merely a set of C/C++functions that can be called from your code. Your sketch undergoes minor changes (e.g. automatic generation of function prototypes) and then is passed directly to a C/C++ compiler (avr-g++). And coming to the compiler here comes the new term called Arduino Integrated Development Board(Arduino IDE).I will discuss about its interface and complete information in my next article but now we are going for its overview.The Arduino/Genuino Uno can be programmed with the (Arduino Software (IDE)). Select "Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino/Genuino Uno comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also

bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details. The ATmega16U2 firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then rese ing the 8U2.On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).[10] figure 3.12 show the Arduino IDE

3.9.3 Processing IDE

Processing is an open-source graphical library and integrated development environment (IDE) / playground built for the electronic arts, new media art, and visual design communities with the purpose of teaching non-programmers the fundamentals of computer programming in a visual context. Processing uses the Java language, with additional simplifications such as additional classes and aliased mathematical functions and operations. As well as this, it also has a graphical user interface for simplifying the compilation and execution stage.

The Processing language and IDE were the precursor to numerous other projects, notably Arduino, Wiring and P5.js.

CHAPTER FOUR
PROTOTYPE DESIGN AND RESULTS

4.1 Main Housing

The main housing of the sentry turret system is the foundation for which all additional hardware structures will stem from. The design is a square-like in design with curved edges. The top surface of the housing will hold Arduino, circuit, power supply and servo armature. Material used for the frame IS MDF wood. The housing will be built by hand, not preassembled. No components are welded to the main frame. Only screws are used to lock everything in place (and remove if necessary). The top surface will have holes drilled to feed wires through to the power supply. Figure 4.1 below depicts a top view of the surface to illustrate the position in which all the components are laid.

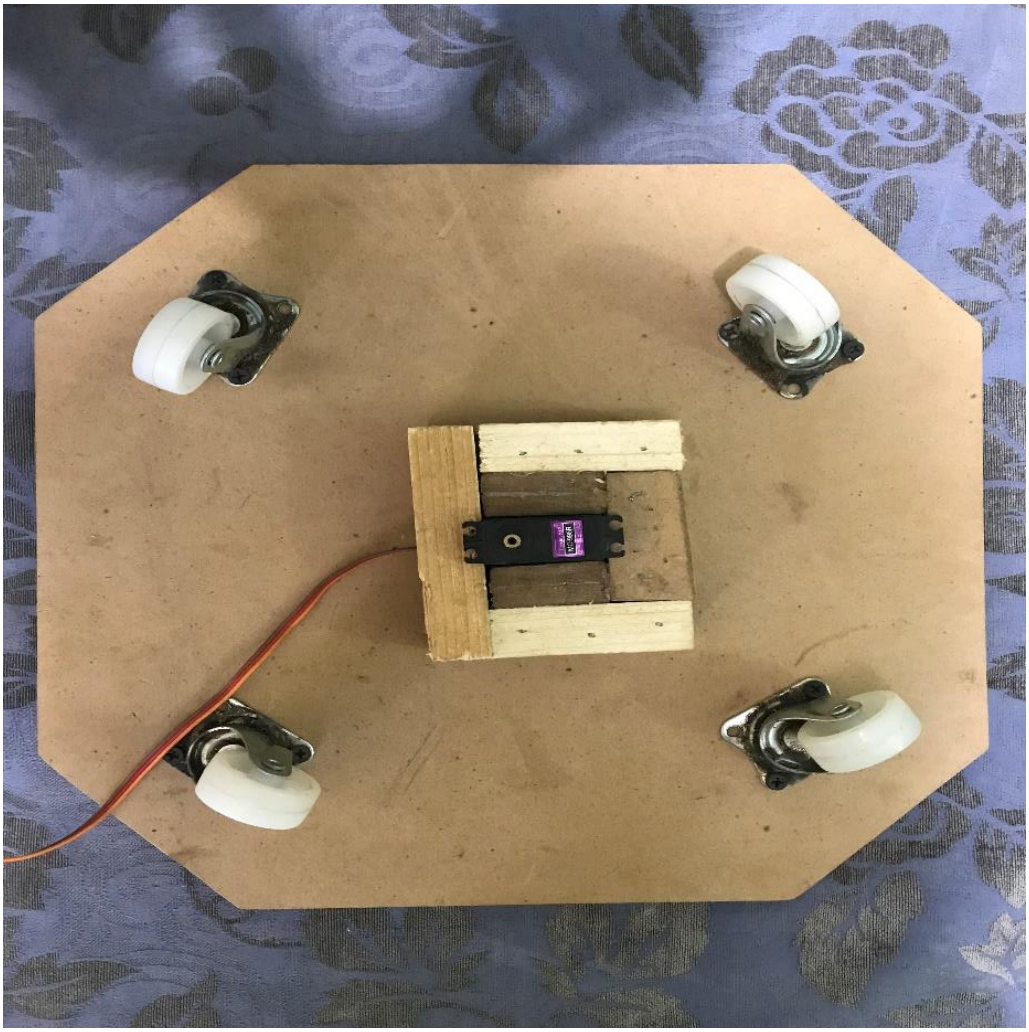

Figure 4.1: main hosing

4.2 Turret Design

The turret design is broken down into two main sections consisting of the servo armature (including the weapon platform mount), and the main housing. the final design build will be modular, mobile, and accessory friendly. The aesthetic design will be robust, yet streamlined; representing a more militaristic look. For the initial design stage of the servo 64 armature, material used will consist of MDF wood.

Below in Figures 4.2 and 4.3 are two designs for the sentry turret system. The first Figure represent a broad view of the system with the second diagram consisting of two close up views of the servo armature.

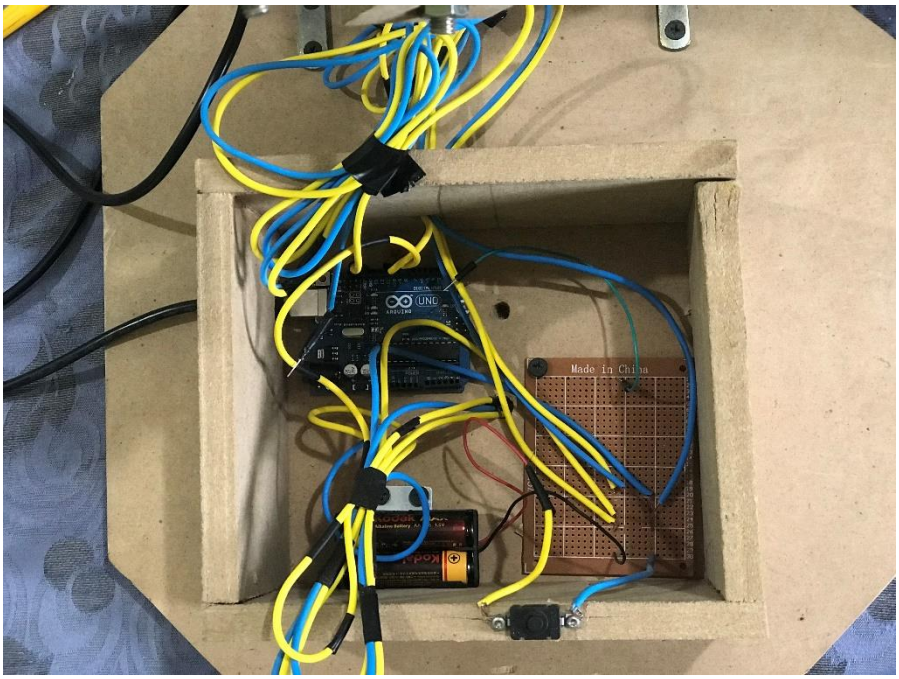

Figure 4.2: Main Housing Overview

Figure 4.3: Servo Armature Mount

4.3 Targeting Control

The open-source Arduino environment and prebuilt libraries. One of those prebuilt libraries include the Servo library. This library allows up to 12 servos, each at various

angles between 0 and 180 degrees. Below in Tables 4.1 and 4.2 are an outline of the various functions that will be applied from the Servo library.

Function	attach()
Description	Attach Servo variable to pin
Parameters	servo: a variable of type Servo pin: the number of the pin that the servo is attached to min: the pulse width, in μS, corresponding to the minimum (0- degree) angle on the servo max : the pulse width, in μS, corresponding to the maximum (180- degree) angle on the servo)
Syntax	servo.attach(pin) servo.attach(pin, min, max)

Table 4.1: Arduino Servo Library Attach

Function	detach()
Description	Detach the Servo variable from its pin
Parameters	servo: a variable of type Servo
Syntax	Servo.detach()

Table 4.2: Arduino Servo Library Detach

In order for the board to communicate with the tablet, a second Arduino library must be incorporated. The Arduino Serial library allows transmission between the microcontroller and USB port on the computer which the computer to control the servos.

4.4 Image Capture

The very first step in this motion tracking system is to capture the images that are in the field of view of the turret system. This information is then processed by the image tracking software and used to send control signals to the servo motors as well as targeting data to the user interface so that a user can see what the tracking software can see. This is one of the first decisions that were required out of this project as well as one of the more daunting tasks to accomplish.

4.5 Tracking System

The Processing programming language is used due to its versatility, abundant online support and the fact that the source code is written in Java and not C++. The list of useful libraries that were used for this project included many options that aided in adding more complex abilities to this project. Among the multiple libraries that were used for this project are the JMyron library, the BlobDetection library, and the serial communications library Processing.Serial. When the user selects to run the Processing IDE as well as the associated program files the program must borrow heavily from the predefined libraries in order to function. In the block below sequence of events that takes place after the user initiates the program is illustrated.

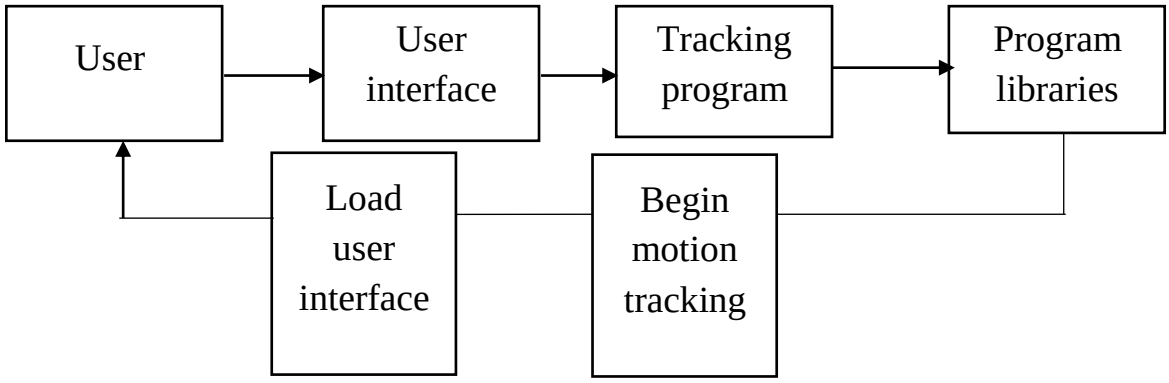

Figure 4.4: Software Sequence of Events block

The Processing based code has multiple functions that it must complete at start up each time that it is run on the user's tablet. The main function calls upon multiple classes that each completes a separate task. The first class that is utilized is the user interface class that loads as soon as the user runs the program. Following this is a class designated to making the connections with the arduino and designating instructions to pin locations. The JMyron library is called to get camera input working and the blob detection library is called to begin the motion tracking portion of the code. Each of these functions and classes are using the information coming from the large amount of required variables. Most variables are partially populated by data received from the video system while the rest are pre-defined.

5.6 Results

Accuracy It is medium in classic system because of impression which effect on the human begins, but the sentry gun doesn't effect. and Safety Because the guard person maybe is exposed to the risk, but sentry gun it is auto and doesn't need for person to shout it. the Cost is less Because the classic system guard needs expenses, regular salary…. but sentry gun doesn't need anything of those. There is no Risk Guard susceptible to the risk, but sentry gun there no risk of guar.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

It is denoted that all results of hardware and software are approximately similar to results in reality. The Sentry Gun System has been controlled automatically through Arduino Uno by building a system of two separated parts, the flexible part depends on high level languages (Proteus and Arduino software). The solid part depends on electronic and mechanical parts. And then connect the solid part with computer via USB to show the status of servo motors which changed due to camera.

5.2 Recommendations

The gun fighter's system has plenty of room for improvement, and, since it has been designed with modifiability and upgradability in mind, it is capable of handling those improvements. Also, the ability to have interchangeable parts is an objective of the ATS so switching to a new and improved part will be relatively easy. The first improvement that should be made is adding the range finder discussed in the features left out section. This would add another factor to the decision of when to attack and when not to attack the targets in the field of vision (this would be based on very specific ranges for warning, engaging, and disengaging).

REFERENCES

[1] Call of Duty Modern Warfare, [Xbox 360 disk], United States: Activision (retail), 5 Nov. 2007.

[2] "Paintball Targeting System." Department of EECS, UCF. Web. 03 Dec. 2011.

[3] "Policy". Arduino.cc. Retrieved 2013-01-18

[4] Edwards, B. "History of Video Calls: From Fantasy to Flops to Facetime." PC World Magazine, June 17, 2010.

[5] Kumagai, Jean. "A robotic sentry for Korea's demilitarized zone." IEEE Spectrum 2007.