Hardware Requirements Specification

for Just A Really Excessive Defense

System

Version 1.0

Table of Contents

1. Executive Summary

1.1 Project Overview

Just A Really Excessive Defense System, or *J.A.R.E.D.S*, is a fully autonomous sentry turret, with a (modified) dismountable Nerf gun. *J.A.R.E.D.S* is designed to be a versatile weapon capable of covering a wide variety of tasks and combat scenarios. It can be used to fulfill many different roles, from an entertaining robot to a home defense system. The turret can be used to restrict access to private locations, as a security system that deters unwelcome guests, or simply for recreational purposes. This project is targeted to the home defense/security enthusiasts, juveniles who are looking for the extra edge in Nerf wars, and anyone else (13+) looking for a versatile, autonomous sentry gun for personal use.

1.2 Purpose and Scope of this Specification

This project is scoped to satisfy all of the requirements listed below in this document (See section **3. Requirements**), as well as meeting all of the expectations in each section of the course curriculum for TEJ4M. This project is being designed and built for the TEJ4M course at Earl of March Secondary School. The project Manager is Mr. Roller, a computer technology teacher at E.o.M. The design and fabrication processes will be completed independently by myself, and all work will be done from home. Under current circumstances, there will be two months to complete the project, upon the completion of this document until the end of this course, June 20, 2021. All parts and material purchases will be made by myself.

2. Product Description

2.1 Product Context

Currently no other similar commercial products exist on the market. The product is designed to be self-contained and will not rely on any external software or systems to operate. As a result, the product does not interface with any other products or systems.

2.2 User Characteristics

- Staff
    - Public institution staff member
    - Goals to restrict access to certain facilities or just for general control/intimidation of members/students
    - Experience with computers
    - Programming background (Specifically Python), know how to run Python software

    - Knowledge of how to access Raspberry Pi camera stream
    - Ages 24+
- Homeowners/property owners
    - Home defense enthusiasts
    - Goals to add security/alarm system, deterrent for home intruders, monitor camera remotely
    - Prior experience with programming in Python
    - Experience with accessing camera streams
    - Ages 20+
- Juveniles
    - Young people who are looking for a really cool toy
    - Enjoys playing video games, and experience with gaming controllers
    - Some experience with using computers
    - Ages 13-20
- General user
    - Steady income
    - Computer hobbyist/robotics enthusiast
    - No prior experience or knowledge required
    - Often makes impulse purchases
    - Desires product for recreational/personal/other
    - Ages 13+

2.3 Assumptions/Risks

- All hardware components must arrive from retailers on time, otherwise different hardware will need to be utilized for the build
- When available, all hardware components must function as intended (Quality control. E.g. Cooling fans must be operational, servo motors must operate at mentioned stall torque) If function is not desirable, than components must either be replaced or the design will need to be reconsidered and changed to adjust for any inconsistencies
- All equipment and tools are assumed to be available and function properly, and all consumables required for the build will be in abundant supply, otherwise new tools will need to be ordered. (E.g. Soldering irons will be operational, sufficient supply of solder wire, etc.)
- All firmware and software will be compatible and up-to-date (Programming interfaces will be compatible with OS, APIs will be compatible with Arduino IDE version and Python 3.7.3)
- The user will own a PC and have the appropriate version of Python installed to run software to remotely control sentry gun to access all features of the product
- The user will be responsible and use LiPo batteries with appropriate care and use to avoid potential safety hazards

2.4 Constraints

- Webcam module resolution is restricted to 640x480
- Raspberry Pi processing power is limited (1.5Ghz, 2GB SDRAM)
- Servo motors can only support certain loads (mass must be restricted)
- Sentry gun LiPo battery can only supply maximum continuous current of 100A (150A burst)
- Sentry gun shell space is limited
- Arduino Nano only has 32kb flash memory
- Raspberry Pi power source must supply sufficient power (must be rechargeable for portability)
- Materials for frame construction are limited

2.5 Dependencies

- PC bluetooth application must be completed before the implementation of the bluetooth module
- A wireless card with bluetooth is required before bluetooth application can be tested
- 3D modelling software must be downloaded to design and print prototype parts
- Firmware APIs must be downloaded on Raspberry Pi before programming can begin
- Arduino APIs must be installed in Arduino IDE before programming
- A new battery compartment must be designed for the sentry gun before a battery can be housed freely on the rear of the gun
- Sentry gun must be rewired with new hardware and internals before frame hardware can be installed
- A reliable system for mounting/dismounted must be designed and constructed before the sentry gun can be mounted to the frame for installation

3. Requirements

3.1 Functional and Hardware Requirements

Req#	Requirement	Comments	Date Rvwd	Customer Reviewed / Approved
HRS-3.1.1	**Functional Requirements**	General functionality requirements.		
HRS-3.1.1.1	The sentry gun shall be able to operate	Default mode of operation. A webcam module is		

	autonomously to acquire targets.	required for operation. Sentry gun will acquire targets and fire on sight.		
HRS-3.1.1.2	The sentry gun shall be placed on two axes to allow for left-right and up-down movement.	(See section **HRS-3.1.2**)		
HRS-3.1.1.3	The sentry gun will continuously span a 270° field of view until a target is acquired.	For autonomous operation: the sentry gun will passively span the area for new targets.		
HRS-3.1.1.4	The sentry gun shall be dismountable from the frame.	Sentry gun-frame hardware connections will be made using 2.54mm JST connectors.		
HRS-3.1.1.5	The sentry gun shall have a manual mode of operation, controlled via a USB gaming controller.	A Wired USB controller can be connected to Raspberry Pi 4B USB 2.0 Type-A port to override manual operation.		
HRS-3.1.1.6	The turret shall be controllable remotely via a Windows bluetooth application.	Application can be used to switch between operating modes, can also be used to remotely control the turret.		
HRS-3.1.2	**Frame Requirements**	Requirements for the construction of the turret frame.		
HRS-3.1.2.1	The sentry gun shall rotate on two axes for rotation: A swivel axis and a tilt axis.	Axes will be driven with high-torque servo motors. Swivel axis will allow for lateral rotation, tilt axis will allow for		

		vertical alignment.		
HRS-3.1.2.2	Each individual axis shall be controlled with a 35kg/cm digital servo motor, with a control angle of 270°.	Two high-torque servos will be used to allow for lateral and vertical rotation along the axes. Servos will be powered with ~7.4v.		
HRS-3.1.2.3	Space on the frame shall be allocated for a compartment to house microcontrollers and other hardware which will control the frame and sentry gun.	A small compartment to securely house frame electronics and batteries.		
HRS-3.1.2.4	Ball bearings shall be used around each rotational axis.	Ball bearings will reduce servo load and increase speed of rotation.		
HRS-3.1.2.5	The frame shall be constructed with wood.	Lumber will be acquired to construct the frame.		
HRS-3.1.3	**Hardware Requirements**	General hardware and electronic component requirements.		
HRS-3.1.3.1	A webcam module shall be mounted directly above the barrel of the sentry gun unit. Module will be connected directly to a Raspberry Pi 4B via USB 2.0 Type-A for processing.	Video will be processed by firmware on Raspberry Pi. (See section **HRS-3.1.4.1**). A USB Type-A extension cable will be used if needed.		
HRS-3.1.3.2	Raspberry Pi and an Arduino Nano	Serial communication		

	shall be connected with a USB 2.0 Type-A to Mini USB cable for serial communication.	between both microprocessors to interface firmware. (See section **HRS-3.3.1.1**)		
HRS-3.1.3.3	Frame axes shall be driven by an Arduino Nano.	Each servo will be driven separately using digital pins 2 and 3 on the Arduino.		
HRS-3.1.3.4	A 128x64 0.96” OLED display shall be mounted to the sentry gun to display a GUI. SCL and SDA pins will be connected to pins A5 and A4 on an Arduino Nano respectively.	OLED will display the GUI for the sentry gun configuration settings. GUI will be navigated via rotary encoder with a push button. (See section **HRS-3.2.4**)		
HRS-3.1.3.5	A rotary encoder with a push button shall be used to navigate the GUI (**HRS-3.1.3.4**).	(See section **HRS-3.2.4**)		
HRS-3.1.3.6	The sentry gun shall use 3 brushed DC motors for operating firing controls. Motors will be driven with Arduino PWM output pins.	Two motors for flywheels; one motor for pusher motor. Motors will be powered with ~11.1v. Both sets of motors will use Arduino PWM output pins to control output motor RPM.		
HRS-3.1.3.7	2x25mm Cooling fans shall be added to each side of the sentry gun to cool the pusher motor.	Extra cooling will be required for the over-volted DC motor to extend lifespan.		
HRS-3.1.3.8	All stock Hasbro	Internals will be		

	Nerf internals shall be replaced with 16awg wiring and *V-212-1C6* Genuine Omron SPDT switches.	upgraded to improve performance and support increased current flow.		
HRS-3.1.3.9	*IRFZ44N* MOSFETs shall be used to drive the motors.	MOSFETs will be used to drive the DC/Servo motors to prevent Arduino output pins burning out.		
HRS-3.1.3.10	The sentry gun pusher motor shall be wired for motor braking.	Motor braking will allow for the motor to instantly stop after firing, for added controllability.		
HRS-3.1.3.11	A new battery tray shall be designed and constructed using 3D printed parts and acrylic to house the LiPo battery on the rear of the sentry gun.	A compartment will be designed and built to store the 3s LiPo battery for the sentry gun.		
HRS-3.1.3.12	Indicator LEDs shall be implemented on the sentry gun to indicate power. Colours shall be configurable via the GUI.	LED strips will be added to the sides of the sentry gun to indicate power. Colours can be configured by the user via the GUI on the OLED display. (See section **HRS-3.1.3.4**)		
HRS-3.1.3.13	A 12v Piezo buzzer shall play a pre-programmed tune upon start-up when the Arduino receives power.	Start-up audio cue.		
HRS-3.1.3.14	A digital voltmeter shall be mounted	Voltmeter will be implemented to		

	to the side of the sentry gun to indicate the voltage of the battery.	allow user to monitor and ensure safe levels of LiPo battery voltage.		
HRS-3.1.3.15	A 4.0 bluetooth module shall be connected to RX/TX pins on the Arduino Nano to receive bluetooth signals.	A bluetooth card will be required to transmit bluetooth signals from PC.		
HRS-3.1.3.16	A USB Type-A extension with on/off switch will be used between the Raspberry Pi and power source.	A switch will be implemented into the circuit for convenient operation.		
HRS-3.1.4	**Firmware Requirements**	General Firmware and Software Requirements.		
HRS-3.1.4.1	Video processing script shall be written in Python 3.7.3.	Script will be run on Raspberry Pi. OpenCV API will be used to track faces.		
HRS-3.1.4.2	Script shall be programmed to not target certain personnel.	.jpeg files will be uploaded to code to bypass targeting system.		
HRS-3.1.4.3	Script shall serialize data @9600bauds/s to Arduino to drive servo motors, using pyserial v3.5 API.	Serial communication between Python script and Arduino. (See section **HRS-3.1.3.2**)		
HRS-3.1.4.4	The script shall stream webcam video using Flask.	Video can be accessed through an external device for remote operation.		
HRS-3.1.4.5	Arduino shall open	Multiple ports will be		

	multiple serial ports to communicate between Raspberry Pi and bluetooth 4.0 module.	opened to allow communication between multiple serial devices using SoftwareSerial API.		
HRS-3.1.4.6	The python script shall begin running upon start-up of the Raspberry Pi.	The script will run as soon as power is switched on without the need for any user input.		
HRS-3.1.5	**Power Requirements**	Batteries/power sources which will be used to power the hardware in section **HRS-3.1.3**.		
HRS-3.1.5.1	The Raspberry Pi shall be powered with a 20100mAh rechargeable battery.	Battery is charged via USB-C. Raspberry Pi will be powered via USB Type-A.		
HRS-3.1.5.2	The Arduino Nano shall be powered by the Raspberry Pi via Mini USB connector.	USB for serial communication also powers Arduino Nano with +5v and GND wires.		
HRS-3.1.5.3	The sentry gun shall be powered with a 3 cell (11.1v) LiPo battery pack with an XT60 connector.	A high-discharge battery pack will be used to supply sufficient power to the sentry gun. Battery will be balance charged via JST-XH connector.		
HRS-3.1.5.4	The servo motors shall be powered with a 2 cell (7.4v) LiPo battery pack with an XT60 connector.	Servo motors and other electronic frame components will be powered by a separate power source.		

3.2 User Interface Requirements

Req#	Requirement	Comments	Date Rvwd	Customer Reviewed / Approved
HRS-3.2.1	A central power switch (SPST) will deliver power to the Raspberry Pi and all of the frame circuitry. When turned on, the sentry gun will operate autonomously.	Central power switch to enable or disable all of the circuitry and microprocessors on the frame.		
HRS-3.2.2	The sentry gun can be operated manually using a USB gaming controller.	The user can connect a USB device to the Raspberry Pi to manually control the sentry.		
HRS-3.2.3	The sentry gun can be controlled remotely using a bluetooth PC application.	The user can run a python script to either select operating mode or override all of the controls completely from a remote location.		
HRS-3.2.4	The sentry gun shall possess a built-in user-interface to configure weapon settings.	UI will be navigated with a rotary encoder. Configuration settings include, but are not limited to: Motor RPM, fan speed, LED RGB levels.		
HRS-3.2.5	The sentry gun can be operated manually by dismounting the gun from the frame.	The user can disconnect all of the connecting wires and use the physical triggers to fire the gun.		

HRS-3.2.6	A simple boat rocker switch shall be located on the sentry gun to enable firing and manual controls.	A secondary switch must be used to allocate power to the motors to allow firing of the sentry gun.		
HRS-3.2.7	The Arduino shall have a programming/debug Micro USB port for programming the microcontroller (Atmega328) and serial communication.	Windows 10 (64 bit) Arduino IDE installation required.		

3.3 Hardware Interfaces

3.3.1 Internal Interfaces

Req#	Requirement	Comments	Date Rvwd	Customer Reviewed / Approved
HRS-3.3.1.1	The Raspberry Pi shall use USB 2.0 to communicate with the Arduino.	A USB Type-A to Mini USB connector will be used to interface the two microcontrollers. One of the four USB Type-A ports on the Raspberry Pi will be used and the Micro USB programming/debug port on the Arduino Nano.		
HRS-3.3.1.2	The sentry Arduino shall use the I2C protocol to drive the OLED display. (See section **HRS-3.1.3.4**)	Vcc and GND pins will be connected to their respective pins on the Arduino. SCL and SDA pins will be connected to pins A5 and A4 on the		

		Arduino respectively.		
HRS-3.3.1.3	A bluetooth 4.0 module shall be interfaced with the Arduino via UART.	The bluetooth RX/TX pins will be connected to their corresponding pins on the Arduino (pins 0 & 1).		

3.3.2 External Interfaces

Req#	Requirement	Comments	Date Rvwd	Customer Reviewed / Approved
HRS-3.3.2.1	The Raspberry Pi shall process video received from the webcam module via USB.	Video resolution will be 640x480. Frames will be processed and checked for faces.		
HRS-3.3.2.2	A bluetooth 4.0 receiver module shall receive incoming data from a PC.	A PC with WiFi card and bluetooth is required.		
HRS-3.3.2.3	The Raspberry Pi shall receive input and be controlled manually by a gaming controller via USB.	The controller will be connected to one of the four USB type-A ports on the Raspberry Pi.		
HRS-3.3.2.4	The Arduino shall have a Micro USB programming/debug port for uploading code to the microcontroller and accessing the Arduino serial monitor.	Arduino IDE is required.		

4. Requirements Confirmation/Stakeholder sign-off

Meeting Date	Attendees (Name and Role)	Comments

APPENDIX

Appendix A. Definitions, Acronyms, and Abbreviations

A

Autonomous The ability to act independently or perform tasks with a high degree self-governance
Application Programming Interface (API) A software intermediary that allows interactions between different software applications
Arduino A programmable microcontroller

B

Bluetooth A type of short-range wireless technology for data transmission and communication

D

Direct Current (DC) Current which only flows in one direction

F

Flywheel A wheel fixed to a rotating shaft to store rotational energy

G

Ground (GND) A direct connection to the Earth for electrical current to flow
Graphical User Interface (G.U.I) A graphical user interface is a subset of user interface that has interactive graphical components

I

Integrated Development Environment (IDE) An application for software development, with integrated features such as a source code editor, debugging tools, etc.
Inter-integrated circuit (I2C/IIC) A synchronous serial communication bus

J

Just A Really Excessive Defense System (J.A.R.E.D.S) The name of the project outlined above (See pg. 3)

L

Light Emitting Diode (LED) A diode that emits light when voltage is applied in the forward direction
Lithium Polymer (LiPo) A form of Lithium-ion battery technology that uses a gel electrolyte to convert chemical potential energy into electrical potential energy.

M

Milliamp hours (mAh) - Milliamp hour, a unit of measure of power over time
Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) - A type of transistor that uses a metal oxide-semiconductor
Module Multiple electronic circuits placed on a circuit board to provide a specific function

O

Organic Light Emitting Diode (OLED) A type of display that has an organic film which emits light in response to an electric current

P

Personal Computer (PC) A computer for personal use
Protocol An established set of rules that determines how data is transmitted between devices
Pulse Width Modulation (PWM) A technique which generates pulses of varying widths by changing the duty cycle of a pulse, to simulate an analog signal
Python An interpreted, high-level programming language

R

Raspberry Pi A small, single-board computer
Red/Green/Blue (RGB) An additive colour model in the order of Red, Green, Blue
Receive/Transmit (RX/TX) Receiver/transmitter
Rotary Encoder A device that converts the position/motion of a shaft/axle into an analog/digital output signal
Revolutions per minute (RPM) A unit of measurement of rotational speed

S

Script A subset of programming language that runs code with an interpreter
Serial Clock (SCL) The line that carries the clock signal
Serial Communication Data transmission that sends data 1 bit at a time via computer bus
Serial Data (SDA) The line that is responsible for receiving/transmitting data between master/slave devices
Single Pole Dual Throw (SPDT) A type of switch with one input and one output
Single Pole Single Throw (SPST) A type of switch with one input and two outputs

U

Universal Asynchronous Receiver/Transmitter (UART) Used to transmit and receive serial data
Universal Serial Bus (USB) An industry standard method of transferring bits of data between two devices
User Interface (UI) An interface where human-computer interaction occurs

V

Voltage Common Collector (Vcc) Transistor which shares both input and output voltages with the collector lead

Appendix B. References

- Arduino Nano Product Data Sheet
- Flask
- IRFZ44NPbF Product Data Sheet
- I2C Bus Specification
- The Ontario Curriculum, Grades 11 and 12: Technological Education, 2009 (revised)
- OpenCV
- pyserial 3.5
- Raspberry Pi 4 Model B Product Data Sheet
- SoftwareSerial
- UART
- USB 2.0 Specification
- V-212-1C6 Product Data Sheet