Automated Wildlife Watcher

By

Edwin Lu

Kevin Chen

Xu Gao

Final Report for ECE 445, Senior Design, Spring 2023

TA: Mathur Sekar Abhisheka

3 May 2023

Project No. 61

Abstract

Data collection of various wildlife populations have always been insufficient in size or impractical to execute, we seek to address this problem by automating the procedure of data collection with an automatic wildlife watcher. The device can be placed in an area of interest, keep watch with no human input, detect an object entering its designated field of vision, track the object and record its findings. Currently, market products do not facilitate movement of objects within view, this product can be a basis of various other more specialized environments, wildlife species, and other products of similar function.

Contents

1. Introduction 4
    - 1.1 Problem 4
    - 1.2 Proposal 4
    - 1.3 High Level Requirements 4
    - 1.4 Block Diagram and changes 5

2 Design 6
    - 2.1 Visual Aid 6
    - 2.2 Design Procedure 6
        - 2.2.1 Microcontroller Submodule 6
        - 2.2.2 Camera Submodule 7
        - 2.2.3 Gimbal Submodule 7
        - 2.2.4 Power Submodule 8
    - 2.3 Design Details 8
        - 2.3.1 Microcontroller Submodule 8
        - 2.3.2 Camera Submodule 11
    - 2.3.3 Gimbal Submodule 12

3. Design Verification 13
        - 3.1 Microcontroller Submodule 13
        - 3.2 Camera Submodule 13
        - 3.3 Gimbal Submodule 14

4. Costs 16
    - 4.1 Parts 16
    - 4.2 Labor 16

5. Conclusion 17
    - 5.1 Accomplishments 17
    - 5.2 Ethical considerations 17
    - 5.3 Future work 17

References 18

Appendix A Additional Flowcharts 19

1. Introduction

1.1 Problem

Despite interests and concern over climate change and human development, there is actually very little data available about both the diversity and distribution of wildlife insects or avian pollinators. This is especially concerning when considering the myriad number of species that are poorly understood. How many are there? How do they live? What do they eat? What can be done to help further their numbers or have the least negative impact.

It typically takes a lot of time and effort to survey wildlife populations, a more popular approach is to citizen science. By setting up feeding stations or flowering plants in private residences and documenting visiting species, we can gather a more complete picture of the ecological distribution and possible human impact on the local species. But this too is a limited approach as it depends on observers spending time outside and physically observing and document what they saw, a costly and arguably, ineffective method of data collection.

1.2 Proposal

We envision a wildlife camera that can keep a certain area of interest not only for a sustained period of time but also can track and follow any wildlife and zoom in for a more clear image or video of their behavior. This not only can relieve time spent for data gathering but also more precise information about suspected wildlife behavior with minimum human presence and interference, as well as some really nice photos and videos.

1.3 High Level Requirements

- Camera must be able to detect objects entering the 15m * 15m field of vision, 15m away, or 1m * 1m field of vision, 3m away.
- Two servo motors must move in coordination and track objects moving across 3m/s.
- Before the start of each tracking, two servo motors must return to their default positions.
- The software must be able to capture a video of the target.

1.4 Block Diagram and changes

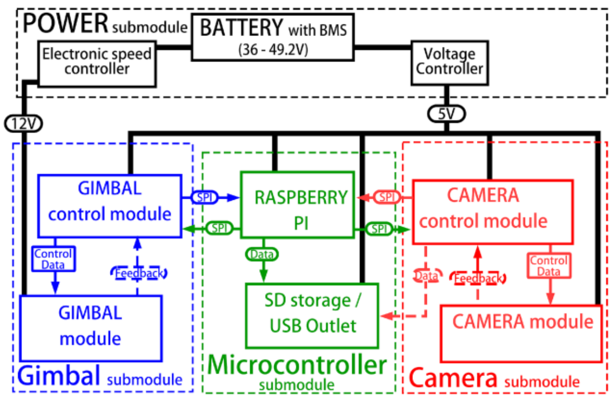
<table>
<tr><th>Block Diagrams</th><th>Description and changes</th></tr>
<tr><td>


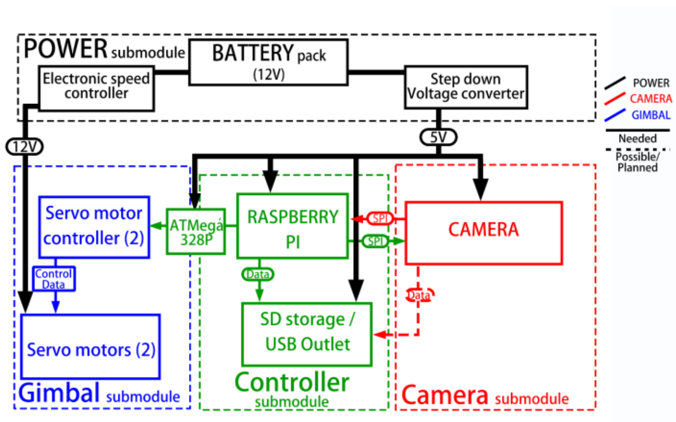
Figure 1: Initial Automated Wildlife Watcher Block Diagram
</td><td>Initial block diagram during the project proposal. There were many connections that were uncertain of (dashed connections), most of the components were undecided as well as the specific values of power supply. The Raspberry Pi was decided to be the only computational component of the entire project.</td></tr>
<tr><td>


Figure 2: Alterated Automated Wildlife Watcher Block Diagram
</td><td>Updated block diagram during the design document revision, peer review, and instructor feedback. Modules were simplified and uncertain components removed, actual microcontrollers were selected and added (ATMega), as well as updated gimbal design.</td></tr>
</table>

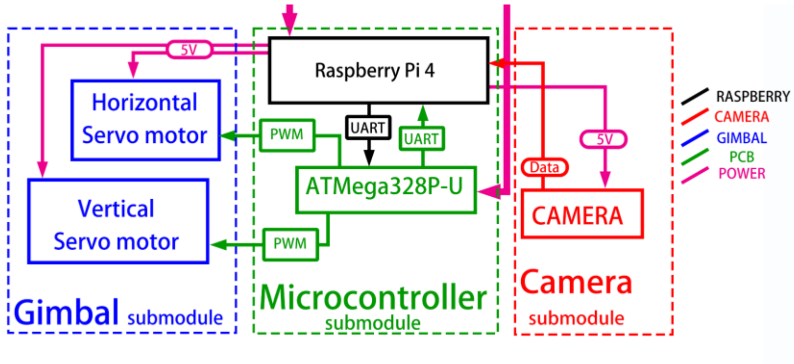
<table>
<tr>
<td>

Figure 3: Final Automated Wildlife Watcher Block Diagram</td>
<td>Final block diagram and project design. Power submodule removed in favor of using external power source, data connections settled with UART and PWMs.</td>
</tr>
</table>

2 Design

2.1 Visual Aid

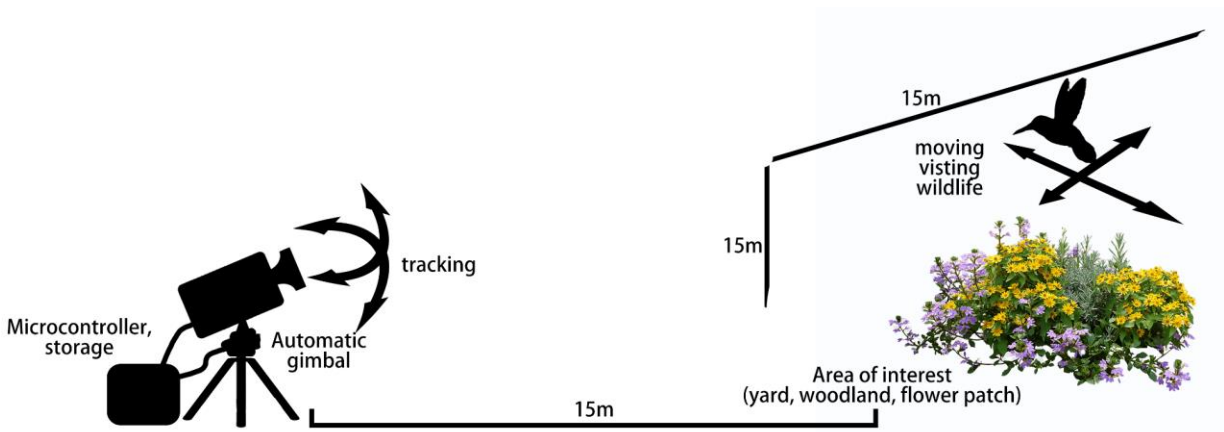


Figure 4: Projected Usage of Watcher System

Figure 5: Actual Physical Product

2.2 Design Procedure

2.2.1 Microcontroller Submodule

The microcontroller, seen in Figure 1, originally was composed of just a Raspberry Pi, as we envisioned it to carry all computational tasks, such as the software of image processing and movement control of the gimbal and other submodules (camera, power).
However that was changed due the course requirements for microcontrollers in the projects as well as difficulties to control the gimbal submodule with solely the Raspberry Pi. After deliberations, a custom made PCB was chosen with the ATMega328P as the microcontroller to control the gimbal submodule. However, due to complications in acquiring the specific ATMega328P chip, a replacement ATMega328P-U chip was used in the final design (see Figure 3), the data connection were codified with pulse wave modulation to the two servo motors directly and UART between the Raspberry Pi and ATMega328P-U.

Furthermore, issues with programming the ATMega328P as well as the PCB design itself could not be ramified in time. As a temporary solution, we replaced the PCB with an Arduino Uno and a breadboard (see 2.3.1 for in-depth details).

2.2.2 Camera Submodule

The camera submodule was conceptually the simplest submodule, as it had a clear function in the project, both as input and feedback interface for the microcontroller submodule, as well as input for the output data (video and images). See in Figure 1, originally, a more complex camera was considered, one requiring a conceptually separated control module for the camera functions, such as zoom. That was quickly changed to the more direct approach of selecting an off-the-shelf webcam that can directly communicate with the microcontroller submodule, as seen in Figure 2. This simplifies the project a great deal, as the camera submodule then only contains a single component without much work.

Since a functional webcam was chosen, it also does not require a separate power supply connection and can be directly supplied from the Raspberry Pi, as seen in Figure 3. Despite the loss of certain functions from the initial requirements, such as automatic zoom and focus, as the camera physically lacked the hardware and the software was not implemented. The changes did finally allow us to create a functional feedback loop of visual input (camera submodule) - computation analysis (microcontroller) - mechanical control (gimbal submodule).

2.2.3 Gimbal Submodule

The gimbal module was the most mechanically complex of the project, as it required explicit hardware functionality for the project to succeed. Initially, a custom made gimbal design was proposed with its own separate control module that accepts data from the Raspberry Pi via SPI.

However, after feedback and availability of the ECE Machine Shop, as well as consideration for the actual functional requirements. It was decided that two off-the-shelf servo motors would suffice as a gimbal submodule, given high-level functionality only requires two-axis of movement in the horizontal and vertical axis. The gimbal is designed to be replaced by two servo motors, with two servo motor controllers that take control signals from the AT Mega 328P chip. At the same time, to determine the high level requirements for the gimbal submodule as well as the altered high level requirements for the field of vision.

$$V_{system} = \omega r \tag{2.1}$$

Equation (2.1) was used to calculate the turn speed the servo motors will need to meet the high level requirement.

During the final construction from the ECE Machine Shop, a tripod was given as a stand for the motors, and given the construction of the servo motors, PWM signals from AT Mega chip can be directly computed by the motors, negating the need for separate controllers. The removal of the power submodule also changed the power supply from the Raspberry Pi.

2.2.4 Power Submodule

The power submodule was initially conceptualized as the project proposal envisioned operation independently from a fixed power supply. However, the actual voltage requirement, runtime, and specific design were all not decided due to other components and hardware were not selected. The general connection had been determined. Certain parts, such as the electronic speed controller, was determined to be needed for the gimbal submodule, as it will require motors; a voltage controller was also suggested for the other parts that may not require a higher voltage.

After deliberations and choosing the other components, a 12V battery pack was selected as an acceptable power supply for the project. The physical battery bracket was supplied by the ECE Machine Shop as extra battery brackets were available, it had 4 slots for AA batteries.

However, during the actual construction from the project, along with complication with the implementation of the other submodules. A direct approach of using an external power supply (off-the-shelf portable power bank, wall outlets) was chosen to simplify the design, and the power supply for most submodules component was supplied from the Raspberry Pi connected to the external power source or directly to the external power source (for the AT Mega 328P).

2.3 Design Details

2.3.1 Microcontroller Submodule

The microcontroller, chosen to be a custom-made PCB with ATmega328P-U and Raspberry Pi, controls both the gimbal and the camera module. The Raspberry Pi is mainly used to contact and control the camera module through the camera's built-in USB channel. ATmega328P is used to communicate with Raspberry Pi and send PWM signals to control the two servo motors in the gimbal submodule. The SD card in the Raspberry Pi will save the data and recorded videos.

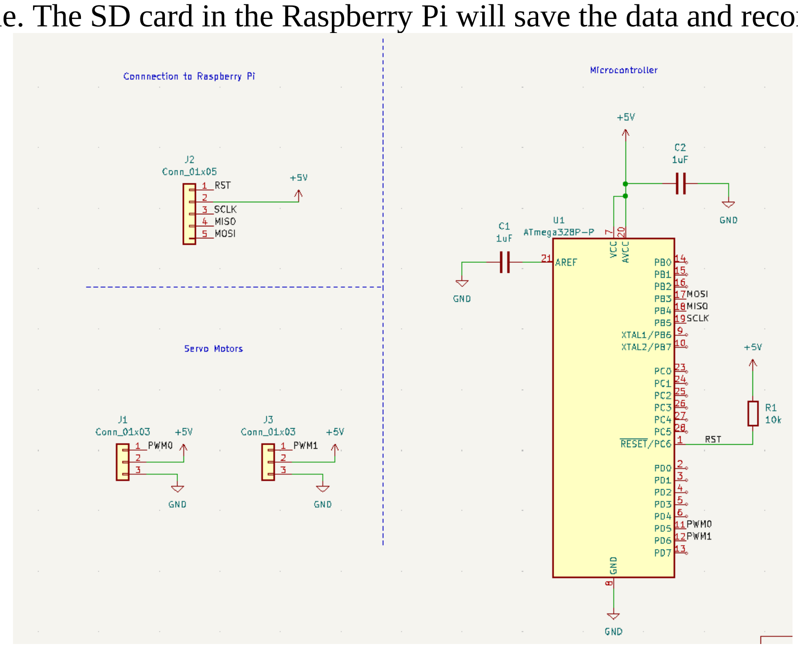


Figure 6: Initial PCB schematic

In the initial design, we decided to program ATmega328P-U by Arduino Uno with the internal 8MHz clock in the ATmega328P-U. However, we failed to program the microcontroller with its internal clock. Due to time issues, we replaced our PCB by Arduino Uno and a voltage divider built on breadboard in order to achieve the functionality of our project.

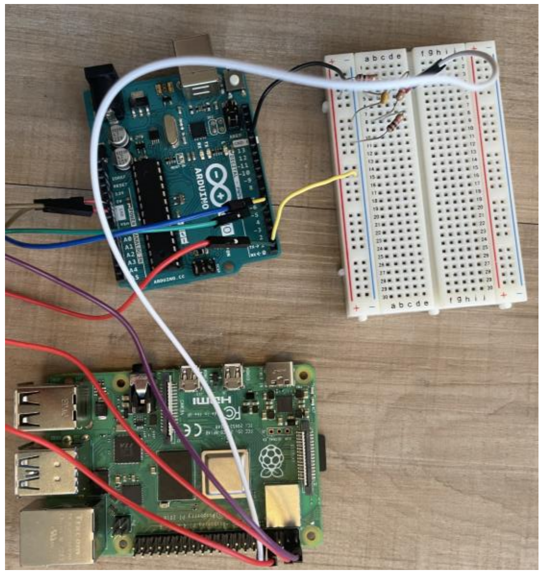


Figure 7: Microcontroller Submodule

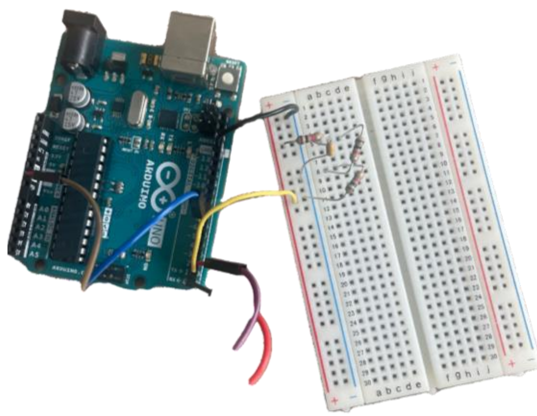


Figure 8: Temporary Solution

In this temporary solution, the main part of our PCB is replaced with Arduino Uno. As mentioned in the procedure, we take UART protocol to build the communication connection between Arduino and Raspberry Pi. Since the GPIO pins on Raspberry Pi can only support 3.3V, we build a simple voltage divider that converts 5V to 3.3V in order to protect the RX pin of the Raspberry Pi. On the Arduino Uno, pins 5 and 6 are used to send PWM signals to control the servo motors in the gimbal submodule.

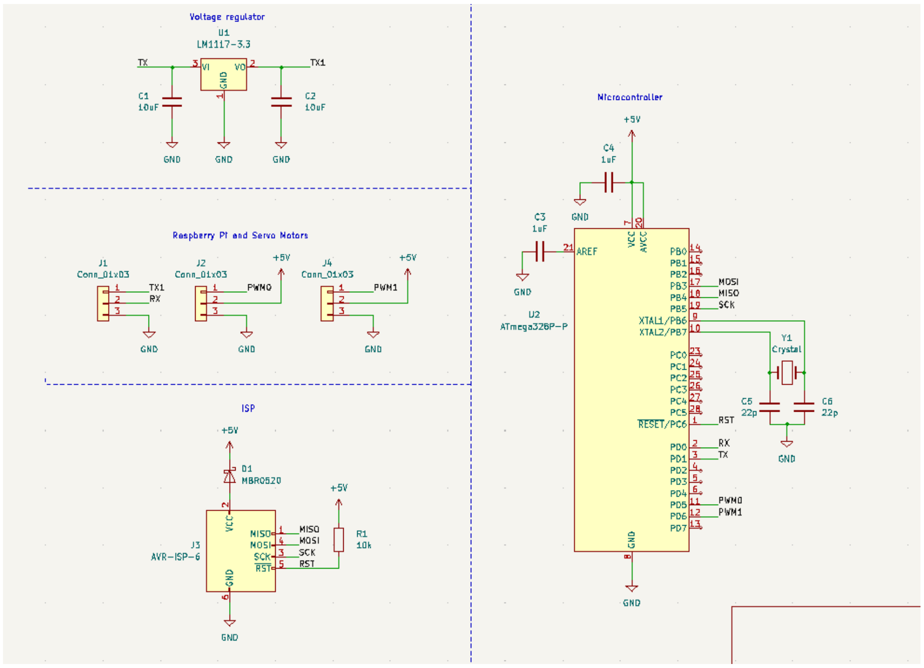


Figure 9: Revised PCB schematic

In the revised PCB schematic, we added a 6-pin ISP connector to the microcontroller in order to program ATmega328P-U by USBasp. We also connect a 16MHz crystal oscillator and two 22pF capacitors to the microcontroller to serve as an external clock. These make up a Minimal USBasp programmable ATMega328P-U circuit. In order to stabilize the output voltage from the voltage divider, we replace the voltage divider built on the breadboard with a Low-Dropout Linear Regulator (LM1117) which can take 5V as input voltage and provide 3.3V as output voltage.

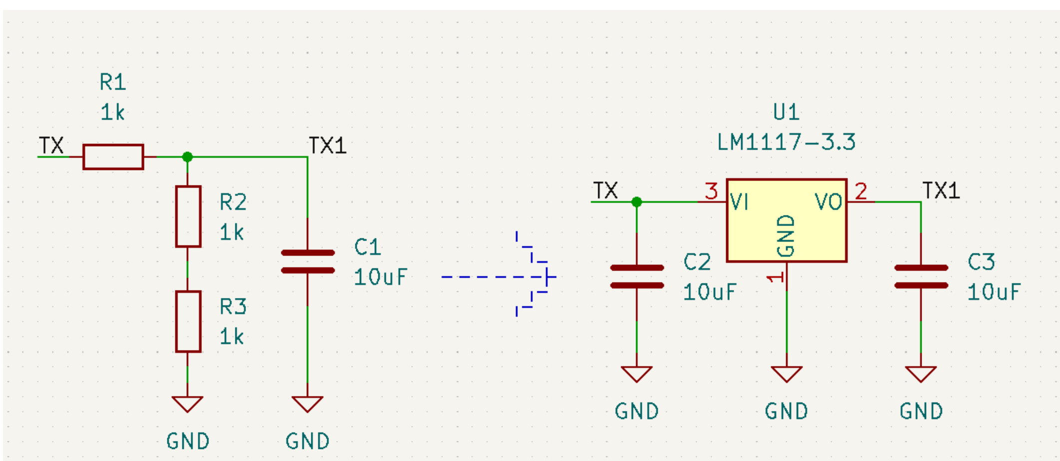


Figure 10: Schematic Change

2.3.2 Camera Submodule

The camera module returns in-time videos to the microcontroller. It will watch over the desired location for object and motion detection, the gimbal submodule will align the camera submodule to certain objects from the signals from the microcontroller, and the camera submodule will record and track objects of interest.

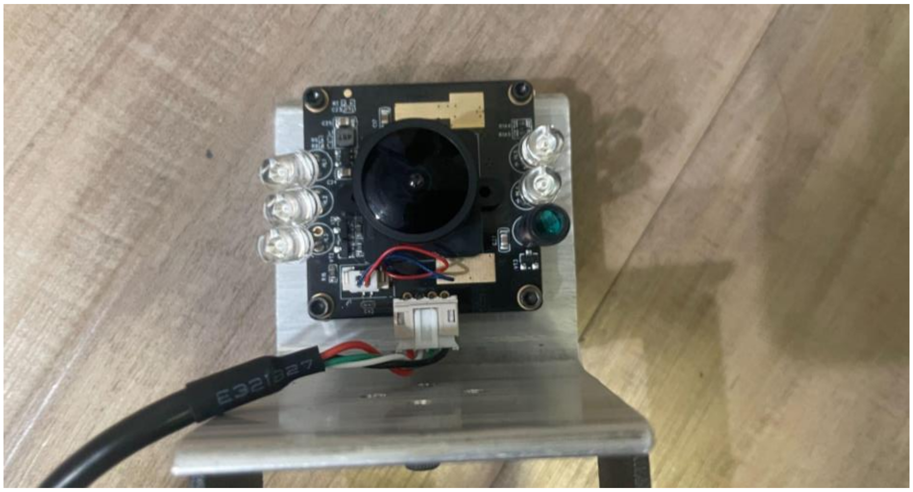


Figure 11: Arducam

2.3.3 Gimbal Submodule

Two stepper motors are used to rotate the camera independently vertically 90 degrees and horizontally 180 degrees. It will support the camera at the designed height to track the target object if it moves within the designed range.

Figure 12: Gimbal Submodule, with the two servo motors as well as the camera submodule

3. Design Verification

3.1 Microcontroller Submodule

Requirements	Verification
When the gimbal needs to move, the microcontroller submodule must communicate with the gimbal system	Ensure the system is operating Move the target object at certain speed no faster than 7 m/s Confirm the gimbal moves when the target object is running out of the camera range
When the camera needs to turn on/off, the microcontroller subsystem must communicate with the camera system	Ensure the system is operating The user should command through the interface to turn off the camera Confirm that the camera turns off
When the recorded image or video needs to be saved, the microcontroller subsystem must give instruction to store data	After the system is operating for a while Turn off the system Check the storage to confirm there are video recordings or images

Table 3.1: Microcontroller Subsystem – Requirements & Verification

The microcontroller did successfully communicate with the gimbal submodule and send the correct signals for the gimbals to move to the directions required for the tracking algorithm.

Due to the nature of the camera submodule being a webcam, it automatically shuts itself off when the software is not in use. The gimbal submodule did communicate with the camera submodule, as the microcontroller facilitates communication between and control of the submodules. User interface would command the system to stop, but it is not reliable and prone with delay from the microcontroller to the gimbal submodule.

Record videos are saved locally in the Raspberry Pi when an external storage device is not present.

3.2 Camera Submodule

Requirements	Verification
When the target object is moving in the camera range, the camera subsystem must detect its motion.	Throw inanimate objects into view Ensure the camera subsystem can detect objects motion
When the target object is too far or too close to the camera, the camera subsystem must zoom in/out at the proper position. Camera can detect	Learn with training and testing sets Place the object at distance of 15 meters and 3 meters from the camera

object entering its the 15m * 15m field of vision, 50m away (or 1m * 1m 3m away)	Ensure the camera subsystem can zoom and focus properly

Table 3.2: Camera Subsystem – Requirements & Verification

The camera successfully detected object motion. However, due to an insufficient amount of testing and a lack of a more robust algorithm, the track algorithm would sometimes mistake its own movement of background objects.

Specific range requirements were not implemented due insufficient testing and a lack of priority for implementation.

3.3 Gimbal Submodule

The gimbal system should provide movement in both the horizontal and vertical axis, to achieve this, a single servo motor (in change of one axis) will be mounted perpendicularly to the mounting tab of another servo motor with a corresponding mounting bracket. Both servo motors will be independently connected to the microcontroller but coordinated to track an object in both axes by the microcontroller. Due to the spatial tolerance, the gimbal should adjust to 23° horizontally and 22° vertically.

Sufficient control system is needed for the servo motor control. A feedback loop of video data from the camera watching the area for motion detection, control system for recognizing the object of motion then tracking it, then feeding signals to the servo motors for tracking the object. The max speed of the target object is 3 m/s, and we also know the radius of our project is about 4 cm. We can use these two values to determine the max angular frequency as seen below.

$$V_{system} = \omega r \quad (3.1)$$

$$\omega = V_{system}/r = (3m/s)\ /\ (4cm)\ =\ 75\ rad/s \quad (3.2)$$

The fastest object that can be consistently detected or tracked is dependent on the speed of the gimbal, which is 0.13s/60° with load.

Requirements	**Verification**
When the target object runs out of the camera range, the motors must be prepared for simultaneous movement to keep the object inside the range. The motors can move the camera to capture object moving at 3 m/s	Ensure the system is operating Move target object at ~ 3 m/s Test to see if the motors can response to keep object inside the camera range
Servo motors can communicate with the	Ensure the system is operating

microcontroller	The motor controllers adjust according to input functions
Servo motors can move according to input parameters	Train with a coordinated poster, adjust to required coordinates when microcontroller requires

Table 3.3: Gimbal Subsystem – Requirements & Verification

The servo motors were operating independently and simultaneously from one another to facilitate movement in both axes.

Servo motors can communicate with the microcontroller as part of the tracking algorithm's functionality, as well as reset to default position under when the microcontroller is reset.

Servo motors did move according to input parameters from the tracking algorithm, however, due to complications with the communication between the Raspberry Pi and the PCB, the servo motors cannot reset to default position every time when the user prompts them.

4. Costs

4.1 Parts

Part	Manufacturer	Retail Cost ($)	Bulk Cost ($)	Actual Cost ($)
Camera	Arducam	34.99	34.99	34.99
Servo Mount Bracket	Elechawk	12.99	12.99	12.99
Servo Motors	Deegoo-FPV	19.99	19.99	19.99
Raspberry pi	GeeekPi	167.00	229.99	167.00
Tripod	QuBona	21.95	21.95	21.95
Micro SD card	SanDisk	11.99	11.99	11.99
HDMI to Micro HDMI adapter	UGreeb	8.99	8.99	8.99
ATMega328P-U	Naughtystarts	25.99	25.99	25.99
SD Card Reader	SmartQ	13.99	13.99	13.99
3 head pins	UNKNOWN	0.11	0.06	0.22
6 head pins	UNKNOWN	0.13	0.07	0.13
Assorted wires	UNKNOWN	6.98	5.48	6.98
Assorted pin connectors	UNKNOWN	12.99	8.59	12.99
HDMI to Micro HDMI adapter	UNKNOWN	6.99	6.99	6.99
USB-A cord	UNKNOWN	6.29	6.29	6.29
Total	$351.48			

Table 4.1: Parts Cost

4.2 Labor

We estimate the worker labor at around $40 per hour to approximate a realistic starting salary for a graduated student with a bachelor's degree in ECE. Each member of the group works an average of 5 hours per week on the project.

$$Labor\ cost\ per\ person\ =\ \$40\ \times\ 5\ \times\ 10 = \$2000 \quad (4.1)$$

$$Total\ labor\ cost = \$2000\ \times\ 3 = \$6000 \quad (4.2)$$

$$Total\ cost = \$6000 + \$351.48 = \$6351.48 \quad (4.3)$$

5. Conclusion

5.1 Accomplishments

We have successfully developed a motion detection and tracking system capable of monitoring objects for short periods of time. This innovation is complemented by the precise, independent control of servo motors, which has greatly enhanced the servos' movement capabilities. Furthermore, We have achieved functional cross-module communication, allowing Raspberry Pi, Arduino Uno, the camera, and servos to work seamlessly together, ultimately leading to more integrated systems.

5.2 Ethical considerations

As UIUC engineering students, we have read and made ourselves familiar with the IEEE code and ACM code. According to part II in the IEEE code[3], we treat everyone fairly and with respect. This group is open to everyone’s ideas and will try our best to avoid injuring others by false actions or any other physical or abuse abuses. In 1.3 of the ACM code[4], we should follow the principle to be honest. Copying other sources without citation is not allowed. In our project, we may use the open source code to help us build our algorithm.

5.3 Future work

In the future, we plan to enhance the capabilities of our project by focusing on several key areas. For the algorithm, we will develop a more robust tracking solution that can adjust for a moving camera and effectively detect moving objects. Additionally, we will train a convolutional neural network (CNN) model on cloud nodes to accurately distinguish between animals and other objects in real-time. For the camera module, we intend to implement zoom functionality and weatherproofing to ensure optimal outdoor performance. Power-wise, an independent power or battery supply will be introduced to increase the system's autonomy. Lastly, we will improve user experience by incorporating remote control through Raspberry Pi's remote desktop using xrdp and developing a more user-friendly interface, making the system more accessible and easy to use.

References

[1] CLIPARTIX. *Hummingbird Clipart*. Accessed: Feb 9, 2023. [Online Image] Available: https://clipartix.com/hummingbird-clipart-image-31490/

[2] PNGITEM. *Garden Flowers png images*. Accessed: Feb 9, 2023. [Online Image] Available: https://www.pngwing.com/en/search?q=garden+Flowers

[3] "IEEE code of Ethics," IEEE. Accessed: Feb 9, 2023. [Online]
Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[4] "ACM Code of Ethics and Professional Conduct," ACM. Accessed: Feb 9, 2023.
[Online]
Available: https://www.acm.org/code-of-ethics

Appendix A Additional Flowcharts

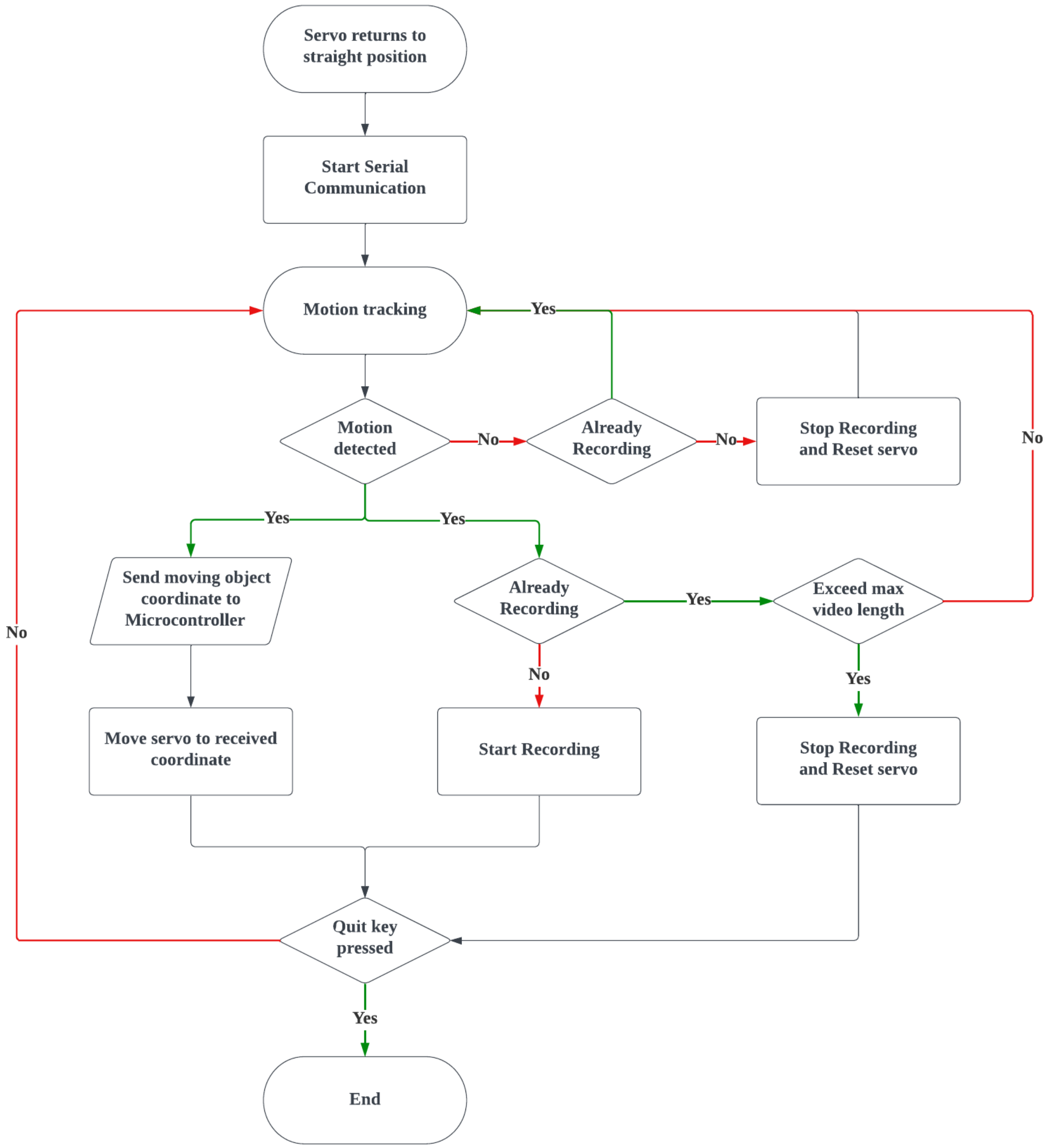


Figure 13: State Machine Flowchart

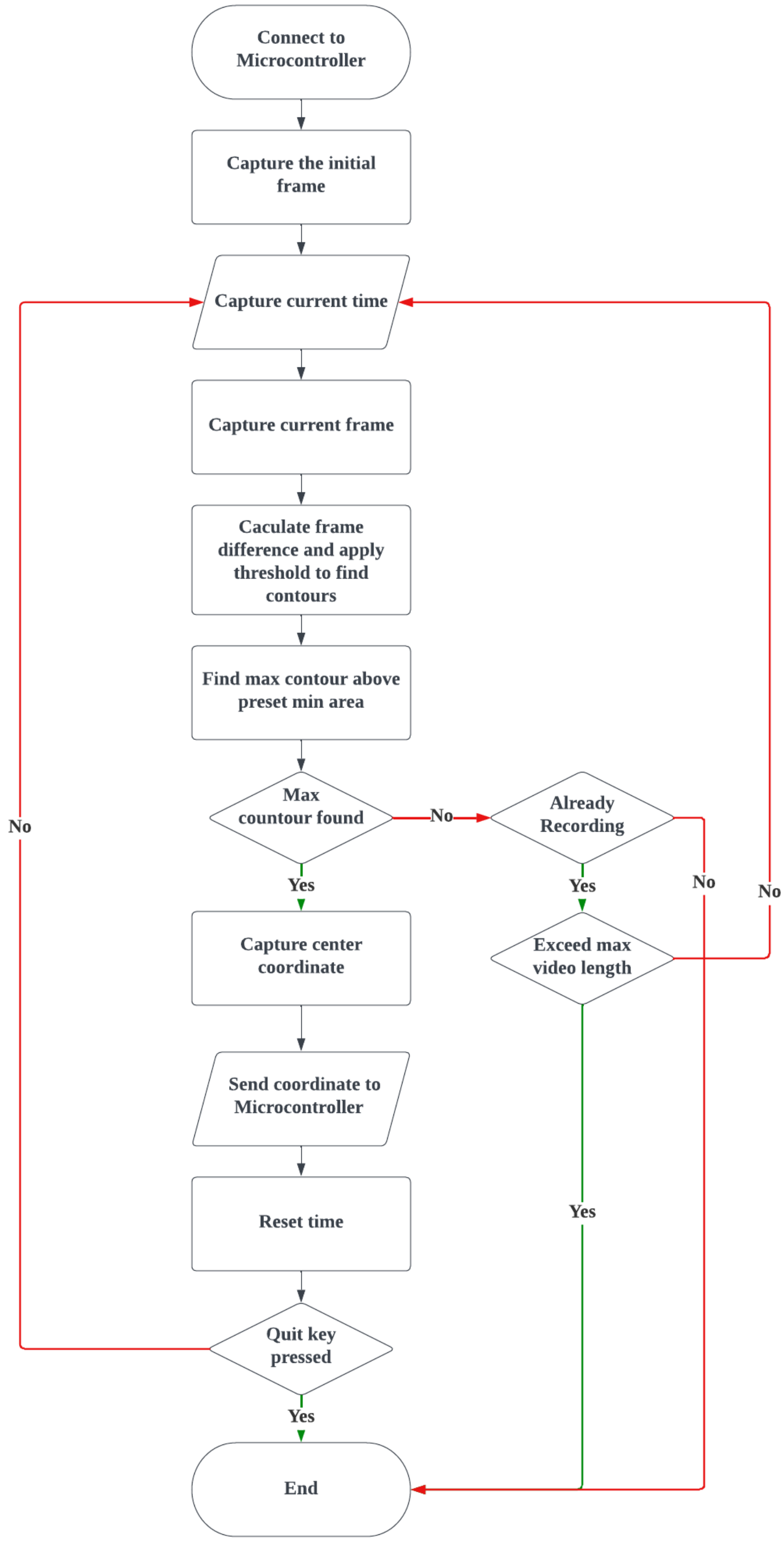


Figure 14:Raspberry Pi (Camera module) Code Flowchart

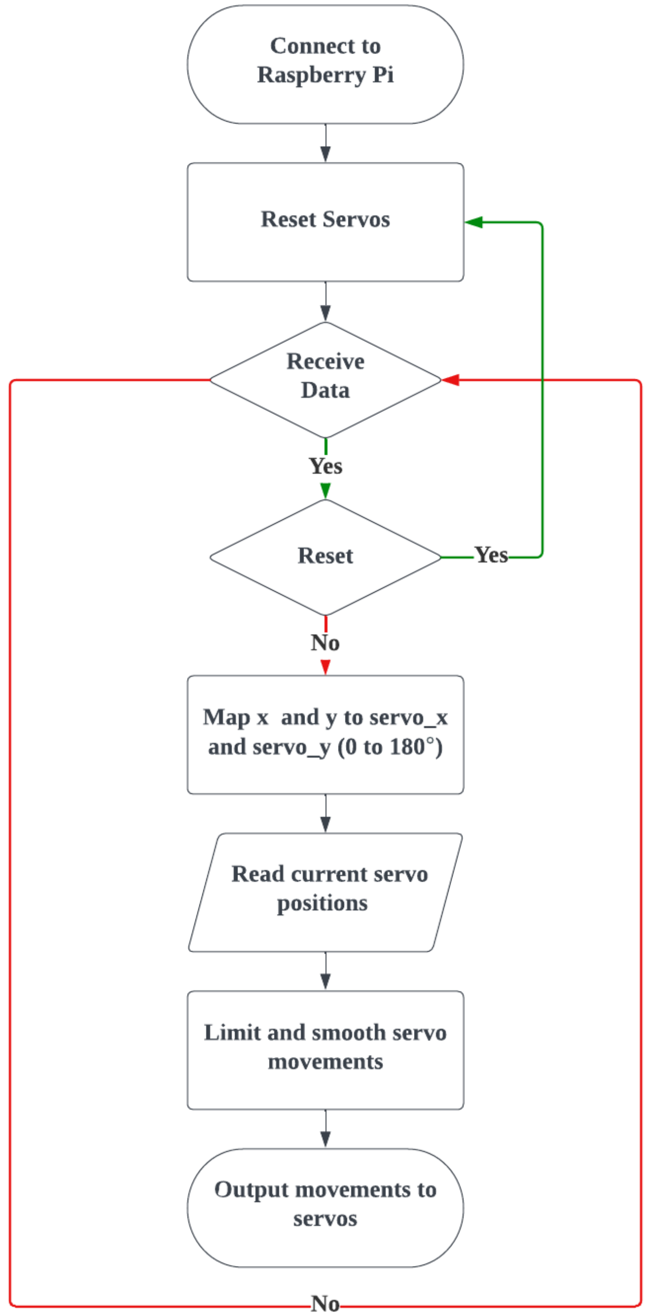


Figure 15: Arduino Uno (Gimbal Control) Code Flowchart