AUTO CAT SNACKS DISPENSER

By

Yuhan Bi

Zutai Chen

Natalie Xu

Final Report for ECE 445, Senior Design, Spring 2022

TA: Jeff Chang

4, May 2022

Project No. 39

Abstract

Our product is an auto cat snack dispenser that can play with cats, keep track of cats' exercise, and eventually drop snacks. It is a rechargeable product that can last for more than eight hours. We have designed our project on two PCBs. A base PCB involves an IR, a laser pointer, and a dispenser. A collar PCB involves an IMU. The two parts communicate through Bluetooth modules. Once IR detects cats' approaching, it turns on the laser pointer to play with cats and asks the IMU to start recording cats' exercise. Once enough exercise is recorded, it tells the dispenser to drop snacks for our cats. Most of our subsystems work as designed except for the failure of data transmission between the two Bluetooth modules. In the following sections, we will include how our project is designed, how our final product fulfills the requirements, and conclusions based on our results.

Contents

1. Introduction

Our project is a cat snack dispenser that can keep track of exercises cats have done and play with them at the same time. It is composed of two main parts: a collar unit and a base unit. Our base unit contains an IR that can detect cats' approaching, a laser pointer driven by the motor that can play with cats, and a dispenser also driven by a motor that will later drop snacks for our cats. Our collar unit contains an IMU that can record the amount of exercise being done by the cats. The two parts will communicate with each other through Bluetooth modules. We have designed two PCBs for the two different parts. In the next following sections, we will talk about each subsystem, our success and failures, and future work.

2 Design

Our project is built on two PCBs.

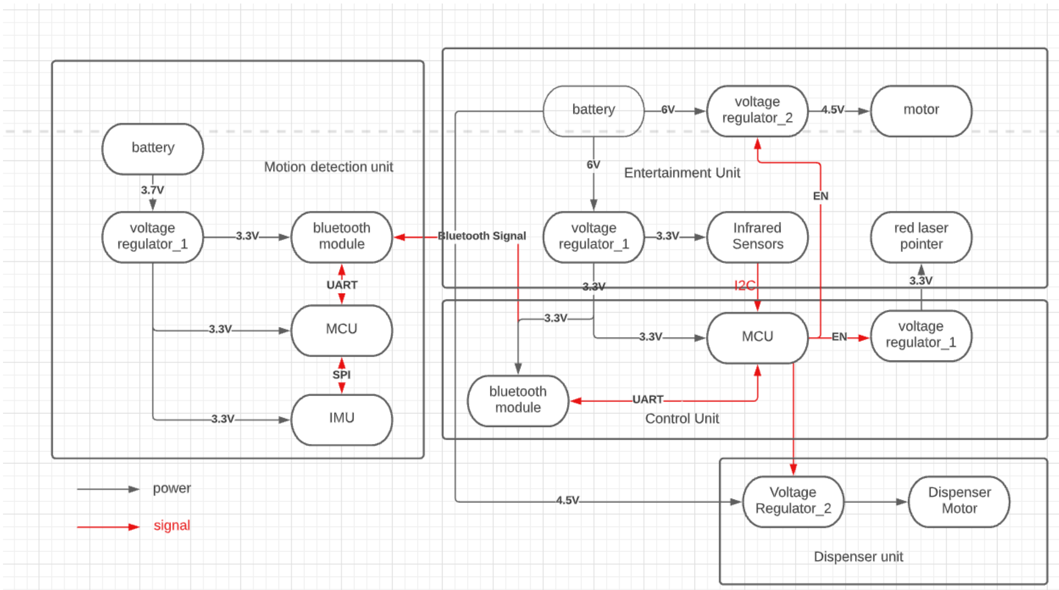

Figure 1. Block Diagram of the Entire System

2.1 Collar PCB

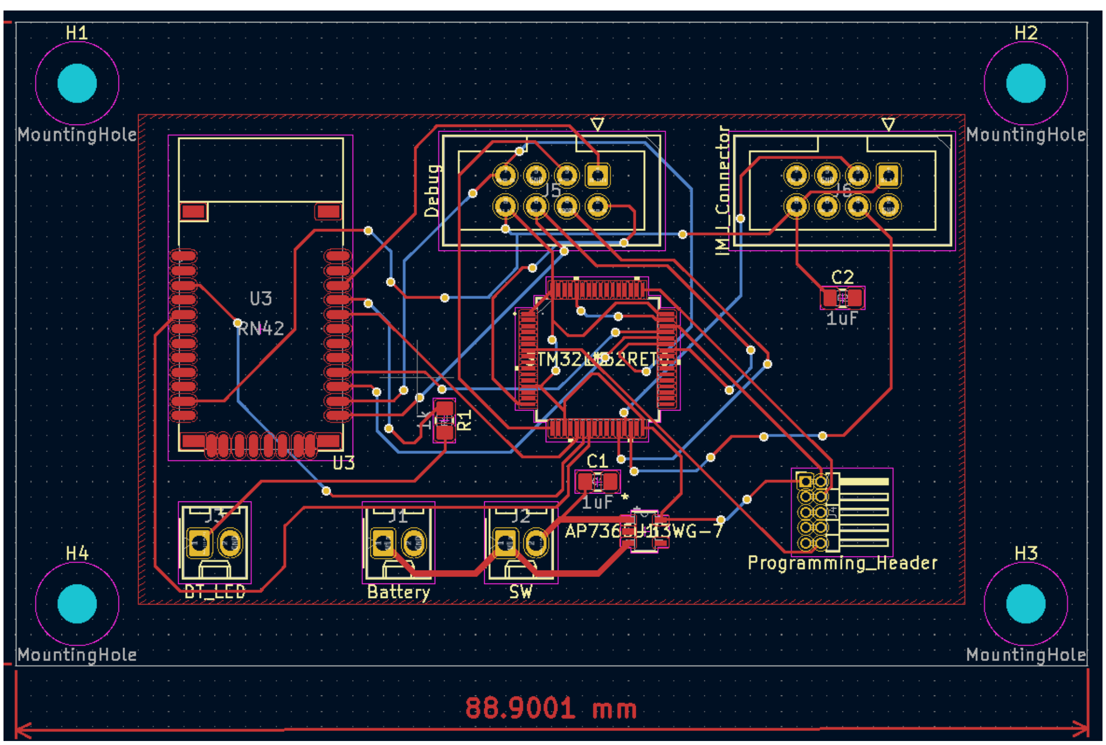

Figure 2. Collar PCB Design

Our collar PCB contains three major components: an MCU, an IMU, and a Bluetooth module. Above we have included a block diagram and a full schematic for this part. Below we will introduce each part in more detail.

2.1.1 IMU

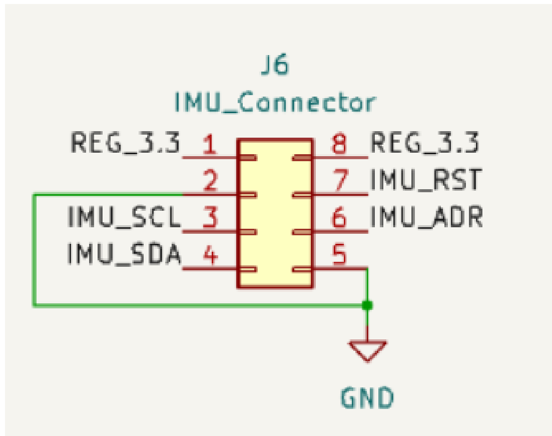

Figure 3. IMU Circuit Schematic

IMU is a hardware part that can contain an accelerometer, a gyroscope, and a gravity sensor. In our design, we will only use the linear acceleration to calculate cats' exercise. Once IMU receives signal from the MCU, it will start to record cats' exercise. BNO055 gives us raw acceleration data using a I2C protocol, and by setting an appropriate delay time, we retrieve the linear acceleration data every two seconds. We set a counter to count the total amount of exercise cats have done. For every time the acceleration reaches the threshold value we set, the counter will increment by one. The equation we use for our counter is:

$$A_{tot} = sqrt(A_x^2 + A_y^2 + A_z^2)$$

It provides the acceleration of the three directions and gives one final total acceleration. We have tested it on our cat and find the threshold value to be about $10.25\ m/s^2$. Since we are coding on STM32Cubeide, we have created a library and a .c file to initialize our IMU to have it work functionally.

2.1.2 MCU

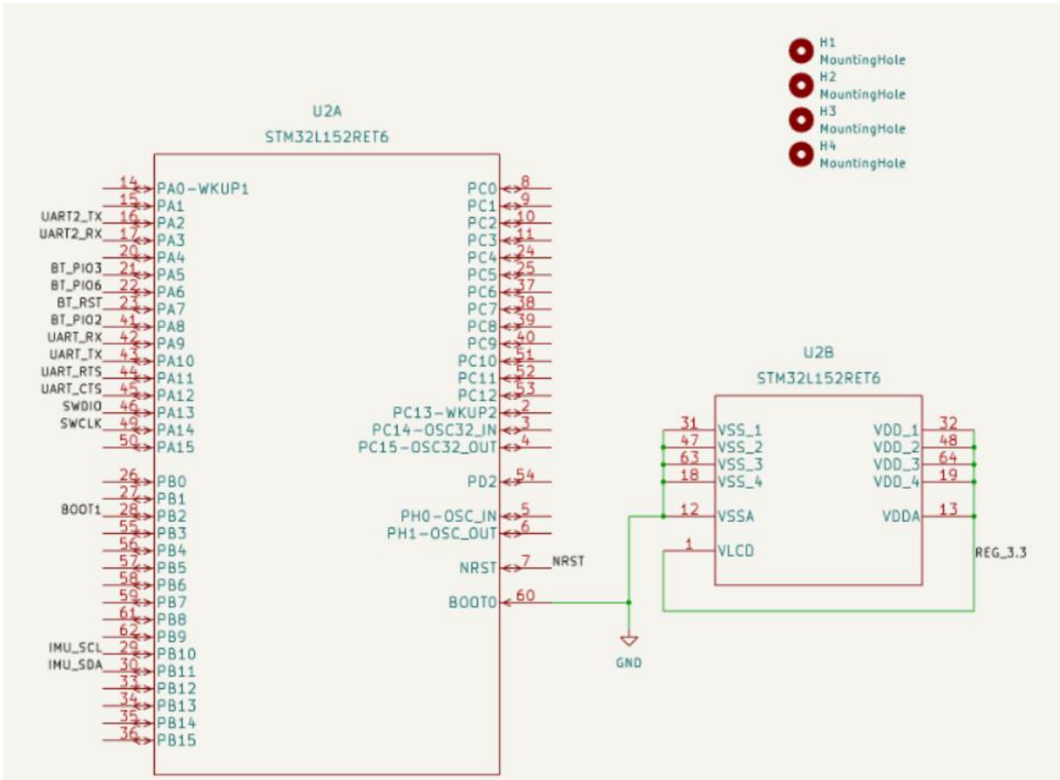

Figure 4. MCU Circuit Schematic

We use STM32L152RET6 as our MCU. The RX and TX pins are used to connect with the Bluetooth module for the UART data transmission, and the SCL and SDA are used to connect with the IMU for the I2C data transmission. We have also left two more RX and TX pins for the serial port, so that we can monitor what is going on inside of the MCU.

2.1.3 Bluetooth Module

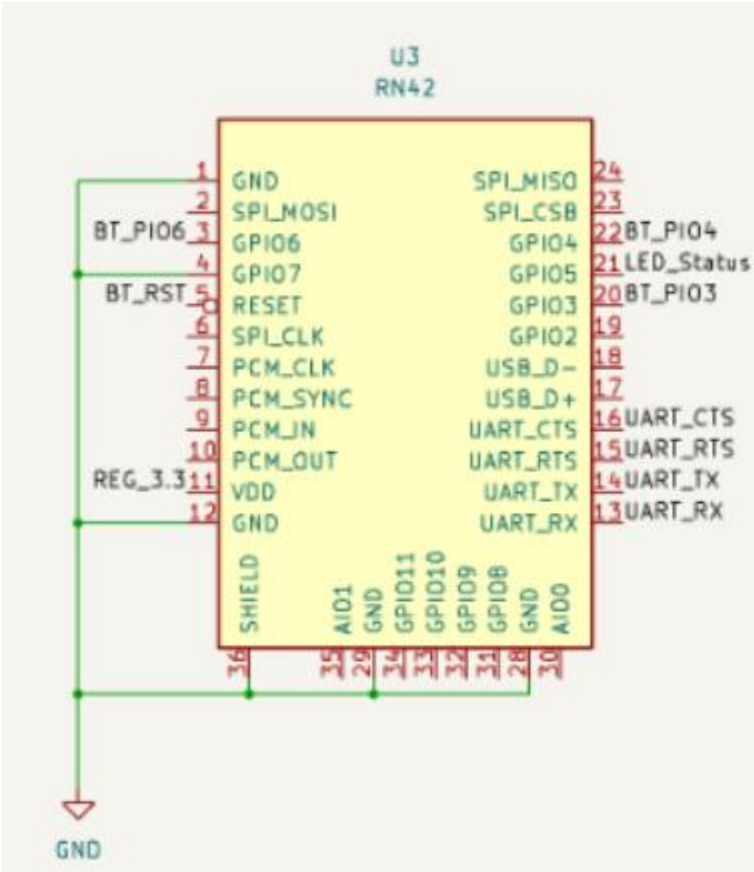

Figure 5. Bluetooth Module Circuit Schematic

The Bluetooth module is used to receive from and send signal to the base unit. Once a signal is received from the other Bluetooth module, it will tell the MCU and IMU start to work. Once enough work has been completed, a signal will be transmitted back to the other Bluetooth module. We designed our project using RN42, but eventually we turn out using HC-05. The reason we decide to switch our Bluetooth module is that when pairing two RN42 together, we will have to disable the authentication, however because we cannot use RN42 with jumper wires, there is no way we can enter the command mode of RN42. On the other hand, since we can use jumper wires to connect the HC-05s on the development board, we can remove their authentication and set the master or slave mode correspondingly using serial communication.

2.2 Base Unit PCB

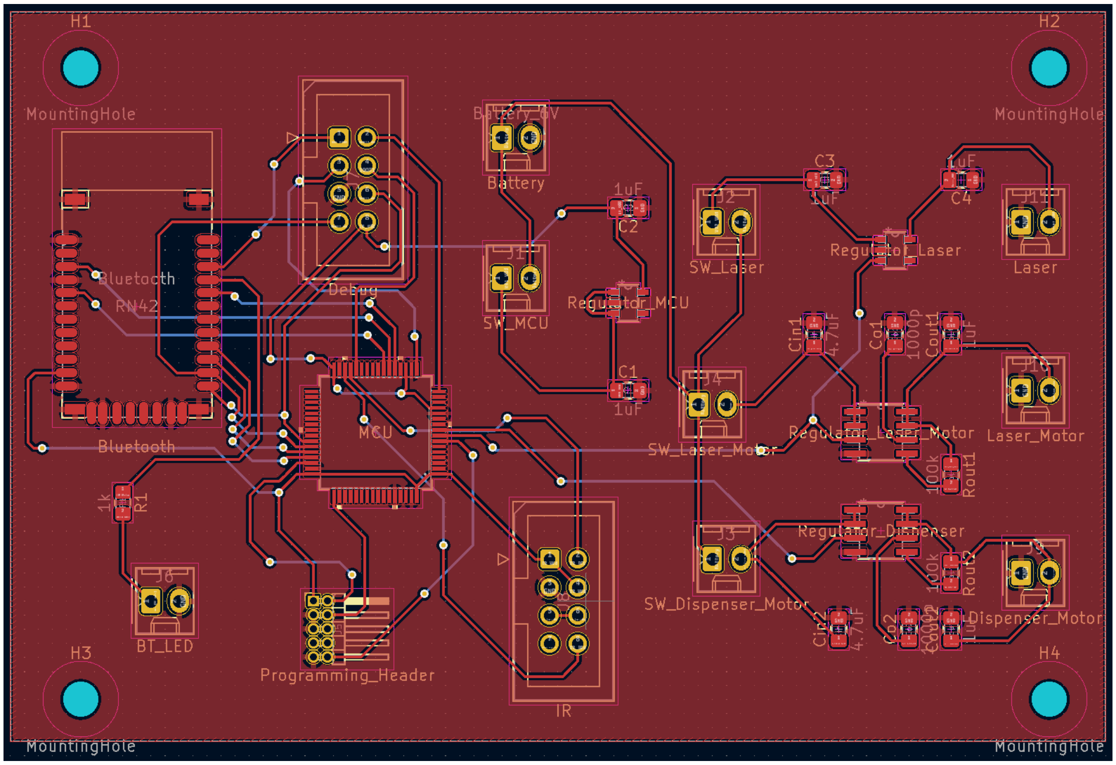

Figure 6. Base Unit PCB

Our base unit PCB contains a MCU, an IR, a laser pointer, two motors, and a Bluetooth module. Above we have included a block diagram and a full schematic for this part. Below we will introduce each part in more detail.

2.2.1 IR Sensor

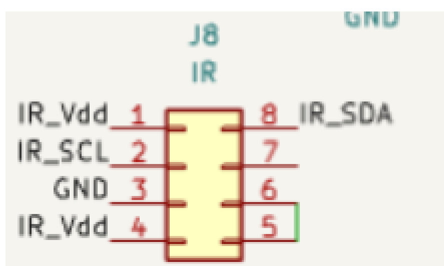

Figure 7. IR Circuit Schematic

IR sensor is used to detect if any cat approaches to the bowl and seeks for food. We use APDS9960 and retrieve its proximity for detection. The basic principle behind APDS9960 is that it contains a LED and a light detector. The LED is always on, and the light detector will continuously check if any source is blocking the light source from the LED. It can provide a raw data from 0 to 255 (8 bits). When nothing is detected, all 8 bits will hold a value of zero. When there is anything being detected, it will give data from 1 to 255 according to the distance between the item and itself. We have also set a threshold value 175, which is about 5cm from the bowl, for our IR sensor. Therefore, it will only send a signal when the cats are truly seeking for food. Since we are using STM32Cubeide to code, we have created an entire library and a .c file to initialize the IR sensor.

2.2.2 MCU

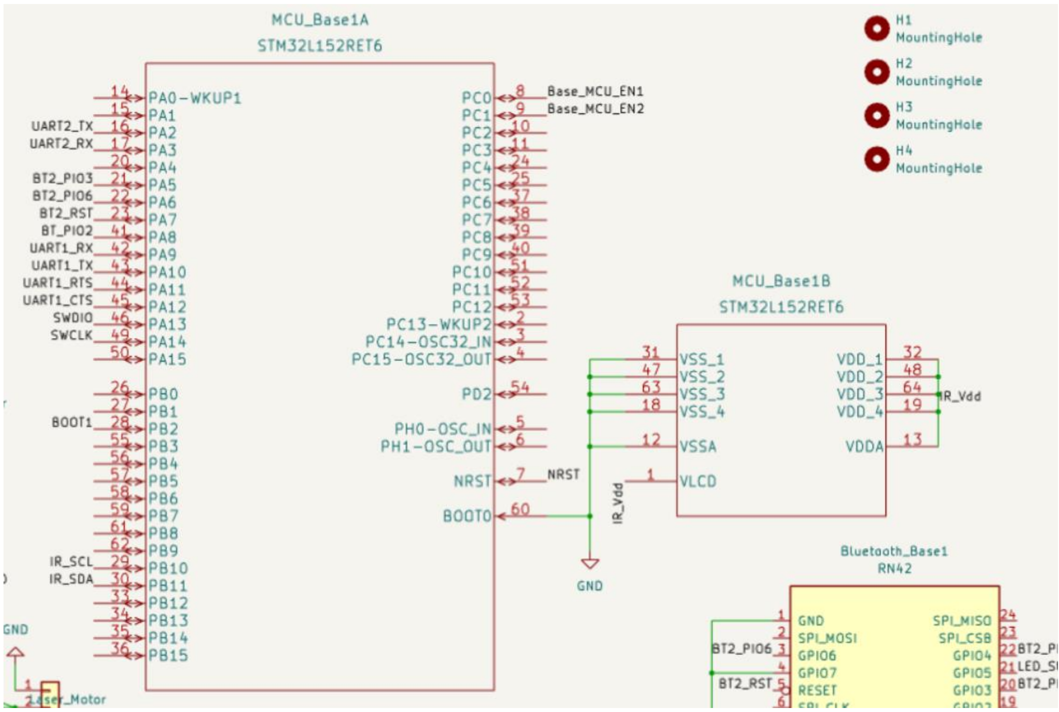

Figure 8. MCU Circuit Schematic

Again, we choose to use STM32L152RET6 for our MCU. The RX and TX pins are used to connect with the Bluetooth module for the UART data transmission, and the SCL and SDA are used to connect with the IR for the I2C data transmission. We have also left two more RX and TX pins for the serial port, so that we can monitor what is going on inside of the MCU. Some other GPIO output pins are also used to trigger the two motors for the laser pointer and the dispenser.

2.2.3 Bluetooth Module

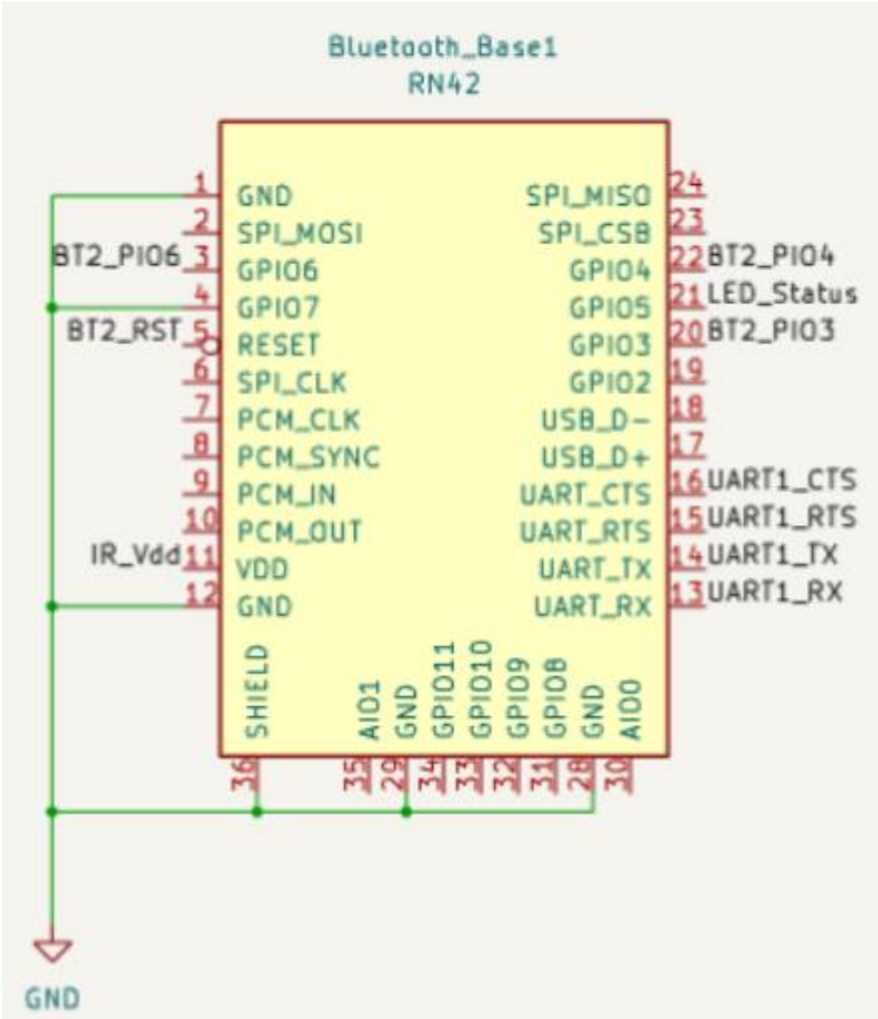

Figure 9. Bluetooth Module Circuit Schematic

The Bluetooth module is used to receive from and send signal to the collar PCB. When IR detects cats' approaching, a signal will be sent to the collar part and asks the IMU to start working. It will also receive signal from the collar part when enough exercise is done by cats, and once the signal is received, it will tell the MCU to turn on the dispenser motor.

2.2.4 Two Motors and A Laser Pointer

Figure 10. Voltage Regulator Circuit Schematic

When IR detects cats' approaching, the laser pointer driven by a motor will both be turned on. They are controlled by the two GPIO output pins on the MCU. When cats have done enough exercise, the dispenser driven by the other motor will be turned on using one GPIO output pin on the MCU.

3. Design Verification

Most of our project works perfectly except the two Bluetooth modules on the PCBs cannot successfully communicate with each other. We have tested all other parts of our design, and they seem to work as expected. IR can detect cats' approaching and turn on the laser pointer, and it can also send a '1' in the simulation. The IMU can work and count on cats' exercise when a '1' is sent to the MCU in the simulation. Once enough exercises have been done, the Bluetooth module can send a '1' back. The dispenser can be turned on when a '1' is entered in the simulation. The Bluetooth modules cannot work because we cannot use jumper wires to remove their authentication. To show the functionalities, we integrate all the parts on the development boards, and with jumper wires, the whole system works just as expected.

3.1 High-level Requirements for the Entire Project

3.1.1 IMU Accuracy

High-Level Requirement: The MCU in the motion detection unit should at least provide an output that achieves 80% accuracy.

Verification: We have connected a LED to our IMU, and the LED will blink every time when the counter increments by one. The counter is designed to increment by one when the cats' acceleration reaches our pre-set threshold value. We tested it on a real cat and found the acceleration to be about $10.25m/s^2$. Some possible inaccuracy may come from when cats are shaking their hair instead of doing exercise, which could have the acceleration to increase very rapidly in a limited time. To avoid this scenario, we used software coding to set a delay time each time after we retrieve data from the IMU. We have tested our product on a real cat, and eventually we found the best delay time to be two seconds. We could easily see if the counter works perfectly by checking if the LED blinks properly. We can see that every time when cats interact with the laser pointer, the LED blinks, and that means our accuracy is 100%.

3.1.2 IMU Low Power Mode

High-Level Requirement: The IMU needs to stay at low power consumption mode (0W power consumption) before the IR detects anything.

Verification: To save energy, we want to make sure that our IMU will not consume too much power when unnecessary. Our BNO055 will automatically enter the low power mode when there is no motion in five seconds. Therefore, to have our BNO055 stay at the low power mode before the IR detects anything, we block the BNO055 to retrieve data through coding. After we initialize our BNO055, it will not work until a signal is received. We achieve this task by writing an "if" statement, and before BT receives anything, the whole IMU will just stay at the low power mode. We also verified this requirement by using an oscilloscope to measure its current in the normal mode and in the low power mode. We found that at a low power mode, the current is about $0.4mA$, and when at a normal power mode, the current is $12.3mA$. By comparing the two results, we can know that the IMU can stay at a low power mode when it is not in use.

3.1.3 System Halt Time

High-Level Requirement: The whole system needs to halt for at least three minutes to start the next activation

Verification: This requirement is also fulfilled by the software code. After the GPIO output pin is on for the dispenser, we set a system halt time to be three minutes. The whole project will just stop there for three minutes, and this can be easily verified by checking if the laser pointer turns on right after the dispenser drops snacks and the IR detects anything.

3.2 High-level Requirements for Each Subsystem

Each subsystem's requirements and verifications are included in the table in Appendix A.

4. Costs

Below we include our cost of each hardware part as well as our labor cost.

4.1 Parts

Table 1 Parts Costs

Part	*Manufacturer*	*Retail Cost ($)*	*Quantity*	*Actual Cost ($)*
MCU (STM32L152RET6)	STMicroelectronics	11.560	3	34.68
Bluetooth (RN42-I/RM)	Microchip Technology	21.900	3	65.70
Bluetooth (HC-05)	HiLetgo	5.990	4	23.96
IR Sensor (APDS9960)	Avago Technologies	7.500	1	7.50
IMU (BNO055)	Bosch Sensortec	34.950	2	69.90
Rechargeable Battery	SparkFun Electronics	4.950	2	9.90
Rechargeable Battery	B B Battery	32.020	1	32.02
1 uF Capacitor	Samsung	0.071	20	1.42
0.47 uF Capacitor	Samsung	0.051	10	0.51
1000 pF Capacitor	Samsung	0.043	10	0.43
1 kOhm Resistor	Stackpole Electronics	0.073	10	0.73
100 kOhm Resistor	Stackpole Electronics	0.360	5	1.80
10-pin Header	CNC Tech	0.640	4	2.56
Laser Diode	Adafruit	5.950	1	5.95
Gear Motor	SparkFun Electronics	4.950	3	14.85
Voltage Regulator	Diodes Incorporated	0.460	3	1.38
Voltage Regulator	Microchip Technology	0.910	3	2.73
Dispenser	M&M'S	19.950	2	39.90
Total				**315.92**

4.2 Labor cost

Based on the 2019-2020 Annual Illini Success Report, the average salary for an electrical engineer graduated from our college is about 36 dollars per hour [1]. Since we all major in electrical engineering, and we plan to work about 8 hours a week, it would result in a total of 2592 dollars for the rest of nine weeks.

Table 2 Labor Costs

	Yuhan Bi	Zutai Chen	Natalie Xu
Major	Electrical Engineering	Electrical Engineering	Electrical Engineering
Total Hours Worked	72	72	72
Multiplier	2.5	2.5	2.5
Labor Force Cost	6480	6480	6480

4.3 Total cost

$$Cost_{tot} = Cost_{labor} + Cost_{parts} = 3 * \$6480 + \$315.92 = \$19755.92$$

4.4 Schedule

Table 3 Example of a Table and Its Title

Week	Natalie	Zutai	Yuhan
1	Complete PCB design and design document.	Complete PCB design and design document.	Complete PCB design and design document.
2	Refine PCB design. Acquire parts needed for the project.	Refine PCB design. Get PCB design approved.	Refine PCB design. Prepare for first round PCB order.
3	Complete PCB board assembly. Work on software code.	Complete PCB board assembly. Bluetooth interface.	Complete PCB board assembly. Power supply and other parts.
4	Complete software code.	Work on driver/BT connection.	Complete PCB board assembly with soldering and mounting.
5	Test software code. Perform full system testing with software.	Test hardware system.	Test hardware system.
6	Complete individual progress report.	Complete individual progress report.	Complete individual progress report.
7	Debug and complete software system testing.	Debug and complete hardware system testing.	Debug and complete hardware system testing.
8	Perform entire system testing with software.	Perform entire system testing with software.	Perform entire system testing with software.
9	Final presentation and final paper.	Final presentation and final paper.	Final presentation and final paper.

5. Conclusion

We learned a plenty of things throughout this project. We learned how to design a PCB, how to integrate parts together, how to create libraries for hardware components, and how to write formal documents. If we could have any chance to re-do the project, we will leave more pins on the PCBs for testing and make better schedules. We will probably work on the software part even before we design our PCB, because some hardware parts are just easier to use than others in terms of coding. Below we will discuss our accomplishment, uncertainties, ethical considerations, and future work in more details.

5.1 Accomplishments

Our project fulfills all high-level requirements we designed, and most subsystems work just as expected. One of our high-level requirements is to make sure that IMU stays at a low power mode before the IR sensor detects anything. This request is completed by the software code, and we will not start to retrieve our data from the IMU before the BT receives any signal. BNO055 will then switch to a low power mode automatically if no data is retrieved in a limited amount of time. The other high-level requirement is to ensure that the IMU to have an at least 80% accuracy. By setting the appropriate threshold acceleration and HAL delay time, our project can have 100% accuracy. To test the accuracy, we have wired up a LED to the collar part so that every time when the counter increments by one, the LED will blink once. The last high-level requirement is to make sure that there is at least a three-minute break between each activation. This is also being done with the software coding. We have set up a system halt time, and the system will just stop once the dispenser drops snacks for the next three minutes. Our IMU can work up to eight hours, and our laser pointer can continuously work for more than two hours.

5.2 Uncertainties

One big uncertainty could be the safety issue. Because our product does not include convenient switches separately, so the only way to power off the system is to disconnect the battery. However, many users, without any background and do not read instruction, may connect the battery inappropriately destroying the whole product. This uncertainty can be fixed by just simply adding switches, however that is still a very big uncertainty for our current product. The other uncertainty is that after cats play with the laser pointer for multiple times, they may just start to lose interest. We have not yet thought about a better solution for this problem except trying to update more functions on our product, but that will also cost owners more money.

5.3 Ethical considerations

Because our collar part involves a large battery, it may give the cats too much weight and eventually cause them to get tired much more easily. We have not yet tested our product on a real cat for a long term, and we cannot guarantee if cats will be happy to wear this product every day. The other ethical consideration is that although our laser pointer is not supposed to harm our cats with the angle we designed, if there are babies under 4 years old play with the laser pointer inappropriately, it may cause some unexpected damages.

5.4 Future work

We will keep working on making our PCB smaller and lighter so that cats can wear the collar. Right now, we have put our collar PCB into a coat because of the huge battery connected to our PCB. We will also add two convenient switches to reset the two PCBs. If possible, we will try to build an app so that the owners can check how much exercise the cats have done in one day.

References

[1] Adafruit BNO055 Absolute Orientation Sensor. Available: https://www.mouser.com/datasheet/2/737/adafruit_bno055_absolute_orientation_sensor-1006287.pdf. [Accessed: March 5,2022]

[2] BL654 Bluetooth Module with NFC. (n.d.) . [Online]. Available: https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl654-series-bluetooth-module-nfc?adgroup=bl654&gclid=Cj0KCQiAjJOQBhCkARIsAEKMtO1Zy_axvInhJjpHQwP_JIdWqZookBmx0zbS74NrFFAdw_RG5eiXDNwaAkeYEALw_wcB&matchtype=%7Bmatchtype%7D&sncid=13&utm_campaign=Bluetooth&utm_medium=cpc&utm_source=google. [Accessed: 05-May-2022].

[3] Bluetooth Uart Service. Microsoft. [Online]. Available: https://makecode.microbit.org/reference/bluetooth/start-uart-service#:~:text=UART%20stands%20for%20Universal%20Asynchronous,by%20a%20physical%2C%20wired%20connection. [Accessed: 05-May-2022]

[4] “Datasheet - STM32L010F4 STM32L010K4 - stmicroelectronics.” [Online]. Available: https://www.st.com/resource/en/datasheet/stm32l010f4.pdf. [Accessed: 05-May-2022].

[5] “Description (pin assignments top view) - diodes incorporated.” [Online]. Available: https://www.diodes.com/assets/Datasheets/AP7366.pdf. [Accessed: 05-May-2022].

[6] E. I. T. S. Services, “Lab,” *Lab :: ECE 445 - Senior Design Laboratory*. [Online]. Available: https://courses.engr.illinois.edu/ece445/lab/index.asp. [Accessed:05-May-2022].

[7] “Huoguo cat shock collar,automatic trainer collar for cats prevent meowing designed,sound vibrate and shock 3 working modes for cats and Kittens Waterproof & Rechargeable,” *Walmart.com*. [Online]. Available: https://www.walmart.com/ip/HUOGUO-Cat-Shock-Collar-Automatic-Trainer-Collar-Cats-Prevent-Meowing-Designed-Sound-Vibrate-3-Working-Modes-Kittens-Waterproof-Rechargeable/558992823. [Accessed: 05-May-2022].

[8] “IEEE code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 05-May-2022].

[9] Inertial navigation article. Vectornav. Available: https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu. [Accessed: 05-May-2022]

[10] K. Magdy, "STM32 HC-05 bluetooth examples master & slave pairing at Commands," *DeepBlue*, 17-Apr-2021. [Online]. Available: https://deepbluembedded.com/stm32-hc-05-bluetooth-module-examples/. [Accessed: 04-May-2022].

[11] "MC7800 - 1.0 a positive voltage regulators - digi-key." [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/ON%20Semiconductor%20PDFs/MC7800(A,AE),NCV7800.pdf. [Accessed: 05-May-2022].

[12] "MCP1727 - 1.5A, low voltage, low quiescent current LDO Regulator Data Sheet." [Online]. Available: https://www.mouser.com/datasheet/2/268/MCP1727_1_5A_Low_Voltage_Low_Quiescent_Current_LDO-2511040.pdf. [Accessed: 05-May-2022].

[13] "SparkFun Electronics." [Online]. Available: https://cdn.sparkfun.com/datasheets/Robotics/SpiderControllerDagu.pdf. [Accessed: 05-May-2022].

[14] University of Illinois, "Illini Success Annual Report 2019-2020," *Box*. [Online]. Available: https://uofi.app.box.com/s/1t8xj69117lrsqm7753ujnrg8yyrtcwn. [Accessed: 05-May-2022].

Appendix A Requirement and Verification Table

Table I. MCU System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. MCU needs to operate at 8MHz for effective data processing.	The MCU should output a system clock signal onto the generic microcontroller clock output pin. Use a generic microcontroller clock output pin to check if the output clock of the MCU meets the requirements.	Yes
2. MCU can give an enable to the voltage regulator connected to the gear motor and the laser pointer attached to it.	Use an oscilloscope to check the voltage across the resistor and see if the MOSFET is operating correctly.	Yes

Table II. Bluetooth System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. The Bluetooth module of the collar needs to be paired with the other Bluetooth module of the base unit.	1). Connect a button to the Bluetooth module of the collar 2). Connect a LED to the Bluetooth of the base unit 3). Check when the button is pressed, if the LED of the other one turns on. 4). The LED on each module flashes every two seconds indicating that they paired successfully.	Yes
2. The two Bluetooth modules can transmit serial data with each other and send to the MCU in UART form.	Tested with Serial.write() and Serial.read() functions on the development board.	Yes

Table III. IR Sensor System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. IR sensors can send a signal to and turn on the MCU.	1). From code set the proximity data threshold to 200 (0-255). Let the IR sensor detect hot sources within 5cm range. 2). Use an oscilloscope and connect it across the MCU to check if it is turned on. 3). Connect it with the laser motor part to check if the laser motor turns on when detecting hot sources.	Yes

Table IV. IMU System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. IMU needs to stay at a low power mode before an enable signal is detected from the IR sensor.	Make sure when no Bluetooth signal is detected, no IMU data is outputted.	Yes
2. The time cats' acceleration reaches the threshold needs to be at least 80% accuracy.	1). Let cats do exercise five times, and check if snacks are dispensed at the base unit. 2). Do this experiment five times to find the average and see if it meets the requirement. Alternative verification: We've connected a LED that blinks every time the threshold value is reached. The accuracy is shown to be 100%.	Yes for alternative verification
3. When cats are shaking their hair, check if the count number increases accordingly.	We retrieve data from the IMU every 2 seconds, so that we can eliminate the noises from shaking.	Yes

Table V. Mechanical System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Once the motor receives the signal from the microcontroller, the dispenser only drops a few snacks every time.	We revised the code to design drop one snack each time as shown in previous videos.	Yes
2. The system should be able to at least operate 8 hours under an average current drain of 833 mA under 6V.	1). Use an oscilloscope to find the current 2). Do calculations based on the current measured to check if it meets the requirements.	Partially since our design has changed.