| **Group** 04 Bing Bong |  |
| --- | --- |
| **Major:** | **Team members:** |
| CEG/CS | Megan Cruz |
| CS | Eric Foy |
| EE | Joshua Moses |
| CS | Anthony Curry |
| *CEG* | Samuel Medley |

**Design Objectives - As Built**

Bing Bong is a garbage bin device that follows a person and helps the user hold garbage. An Android app sends Bing Bong commands and receives notifications. Bing Bong knows when the trash can it carries is full.

Objective 1: When activated on the app, Bing Bong moves and follows the person in front of Bing Bong.

Objective 2: Bing Bong avoids obstacles while moving.

Objective 3: Bing Bong detects when the trash can it carries is full.

Definitions:

Obstacles = Any item that prevents Bing Bong from having a clear path to move through.

Trash = Items that are put inside of Bing Bong

Trash Can = A bin that stores the trash for a household before taking it to the curb.

Full = Bing Bong is full when it hits a certain weight or overflows.

Person = A human as identified by the image recognition software.

| **Group** 04 Bing Bong |  |
| --- | --- |
| **Major:** | **Team members:** |
| CEG/CS | Megan Cruz |
| CS | Eric Foy |
| EE | Joshua Moses |
| CS | Anthony Curry |
| *CEG* | Samuel Medley |

Design Assumptions - As Built

Assumption 1: Bing Bong will only be used inside of a residence.

Assumption 2: Bing Bong will not hold liquids.

Assumption 3: The user provides electrical power sufficient to operate Bing Bong.

Assumption 4: There will be an unobstructed path for Bing Bong to follow.

Assumption 5: The user will provide their phone device to use the app

Assumption 6: The user will provide trash bags for Bing Bong’s integrated trash can.

Assumption 7: Bing Bong will not be used by anyone with reduced physical, sensory, or mental capabilities.

Definition:

Power sufficient to operate Bing Bong: 40Wh or less.

| **Group** 04 Bing Bong |  |
| --- | --- |
| **Major:** | **Team members:** |
| CEG/CS | Megan Cruz |
| CS | Eric Foy |
| EE | Joshua Moses |
| CS | Anthony Curry |
| *CEG* | Samuel Medley |

Design Requirements - As Built

Req No. Requirement

1. Once Bing Bong enters an area for autonomous motion, Bing Bong moves without guidance from a user.

1.1 If a payload weighing 0.5kg or less is loaded into Bing Bong, Bing Bong must move while carrying the payload.

1.1.1 Bing Bong must include a storage compartment for an optional payload.

1.2 Bing Bong must follow a person until the garbage can has been filled when that is Bing Bong’s designated task.

1.2.1 Bing Bong must detect when the onboard garbage can is full.

1.2.2 Bing Bong must identify a person.

1.5 Bing Bong must recognize when a person is nearby.

1.6 Bing Bong must avoid obstacles.

2. Bing Bong must have wireless capabilities.

3. Bing Bong must communicate with the Trash App.

3.2 The Trash App must tell Bing Bong to turn off when prompted by the user.

3.3 The Trash App must tell Bing Bong to take out the trash when prompted by the user.

3.4 The Trash App must tell Bing Bong to follow the person in front of it when prompted by the user.

Definitions:

Trash App: The app that is used to communicate instructions to Bing Bong.

Full: Bing Bong is full when it hits a certain weight or overflows.

Autonomous Movement: Ability to move around without external assistance.

Person: A human as identified by the image recognition software.

Nearby: Within 7 feet away from Bing Bong or less.

Obstacles: Any item that prevents Bing Bong from having a clear path to move through.

Trash: Items that are put inside of Bing Bong

User: A person who is using the Trash app.

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

**Design Constraints - As Built**

| Const No. | Constraint |
| --- | --- |
| 1 | Bing Bong shall fit into a box no larger than 45cm x 45cm x 60cm. |
| 2 | Bing Bong’s weight must not exceed 20 kg before any trash is put in it. |
| 3 | Bing Bongs' required power shall not exceed 10Ah. |
| 4 | Bing Bong shall comply with ANSI/RIA R15.08-1-2020 American National Standard for Industrial Mobile Robots |
| 5 | Bing Bong shall comply with IEEE 802.11 |
| 6 | All parts used must be sourced from WSU-approved vendors. |
| 7 | Cost of parts shall not exceed $300. |
| 8 | Bing Bong’s speed shall not exceed 50 cm/s. |
| 9 | Bing Bong must have at least 2 sensors. |
| 10 | Bing Bong must have at least 3 actuating capabilities. |

Definitions:

Trash: Items that are put inside of Bing Bong

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

**Design Standards - As Built**

| Stand. No. | Standard |
| --- | --- |
| 1 | Any C++ code incorporated within the implementation of Bing Bong must conform to the ANSI/ISO C++11 standard or newer. |
| 2 | Any Python 3 code incorporated within the implementation of Bing Bong must conform to PEP 8 – Style Guide for Python Code. |
| 3 | Bing Bong’s electric motors must conform to the IEC 60034-2-1 standard. |
| 4.1 | Bing Bong must conform to the Raspberry Pi 5 Networking IEEE 802.3af-2003 PoE standard. |
| 4.2 | Bing Bong’s USB-C connection must conform to the IEC 62680-1-3 standard. |
| 4.3 | Bing Bong must conform to the Raspberry Pi 5 Arm Cortex-A76 64-bit CPU IEC 61508-2:2010 and IEC 61508-1:2010 standards. |
| 5 | Bing Bong must conform to the ANSI/NEMA C18 - Safety Standards for Primary, Secondary, and Lithium Batteries |
| 6 | Bing Bong must conform to the ISO 376 - Load Cell Calibration Standard for the weight sensor. |

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

**Design Functionality - As Built**

1. Bing Bong is a robot that encourages cleanliness.
2. Bing Bong is a robot that holds trash, it uses AI to follow you around while you clean and holds anything you want to discard.
3. Bing Bong will detect when its trash can is full.
4. Bing Bong does not require assistance from the user to navigate its environment.
5. The associated app can be used to send commands to Bing Bong telling it to start following the user and take out the trash.

Definitions:

Full: Bing Bong is full when it hits a certain weight or overflows.

Trash: Items that are put inside of Bing Bong

Trash can: Object capable of holding trash.

Garbage can: A larger trash can located inside the house.

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

**Design Impact - As Built**

1. Cultural

Bing Bong helps to make the process of cleaning up trash easier. This helps to promote a culture of cleanliness.

2. Economic

Bing Bong's economic impact is expected to be minimal. There may be a negative effect on some labor segments, but robots and automation increase productivity, lower production costs, and create new jobs in the tech sector [8].

3. Environmental

Bing Bong keeps the environment you’re in clear of trash, this helps protect wildlife, plant life, waterways, soil, and ecosystems from toxic and dangerous chemicals and materials they should not be introduced to [2].

4. Global

Bing Bong uses components sourced from multiple countries around the world. This allows Bing Bong’s parts to be affordable and keep the cost of Bing Bong low. Sourcing Bing Bong’s parts from around the world can also pose obstacles such as legal and regulatory issues and currency fluctuations [1].

5. Public health

Bing Bong helps facilitate trash cleanup which leads to a clean working environment. Many cleaning robots have even been found to have positive health benefits, such as reducing allergies [9].

6. Public safety

Bing Bong follows a person. This creates a tripping hazard. A warning label will be posted on the product and Bing Bong will detect when someone gets too close to it and play a sound. [3]

7. Public Welfare

Bing Bong encourages cleanliness which may encourage other positive behaviors. [4]

8. Social

Bing Bong helps you keep your house clean and a clean house encourages you to be more social.

[1]<https://www.gep.com/blog/strategy/global-sourcing-challenges-guide#:~:text=Despite%20the%20potential%20benefits%20of,regulatory%20issues%20and%20currency%20fluctuations>.

[2]<https://www.camelbak.com/blog-why-picking-up-litter-matters.html#:~:text=If%20we%20spent%20just%20one,shouldn't%20be%20introduced%20to.>

[3] Roomba Owner's Manual, <https://prod-help-content.care.irobotapi.com/files/i_series/i5/og/add_en-GB.pdf>

[4]<https://www.verywellmind.com/how-mental-health-and-cleaning-are-connected-5097496>

[7]<https://extension.usu.edu/mentalhealth/articles/the-mental-benefits-of-decluttering>

[8]<https://www.investopedia.com/articles/markets-economy/091316/3-ways-robots-affect-economy.asp>

[9]<https://living-smarter.com/the-truth-about-roomba-and-cancer-is-there-any-link/>

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

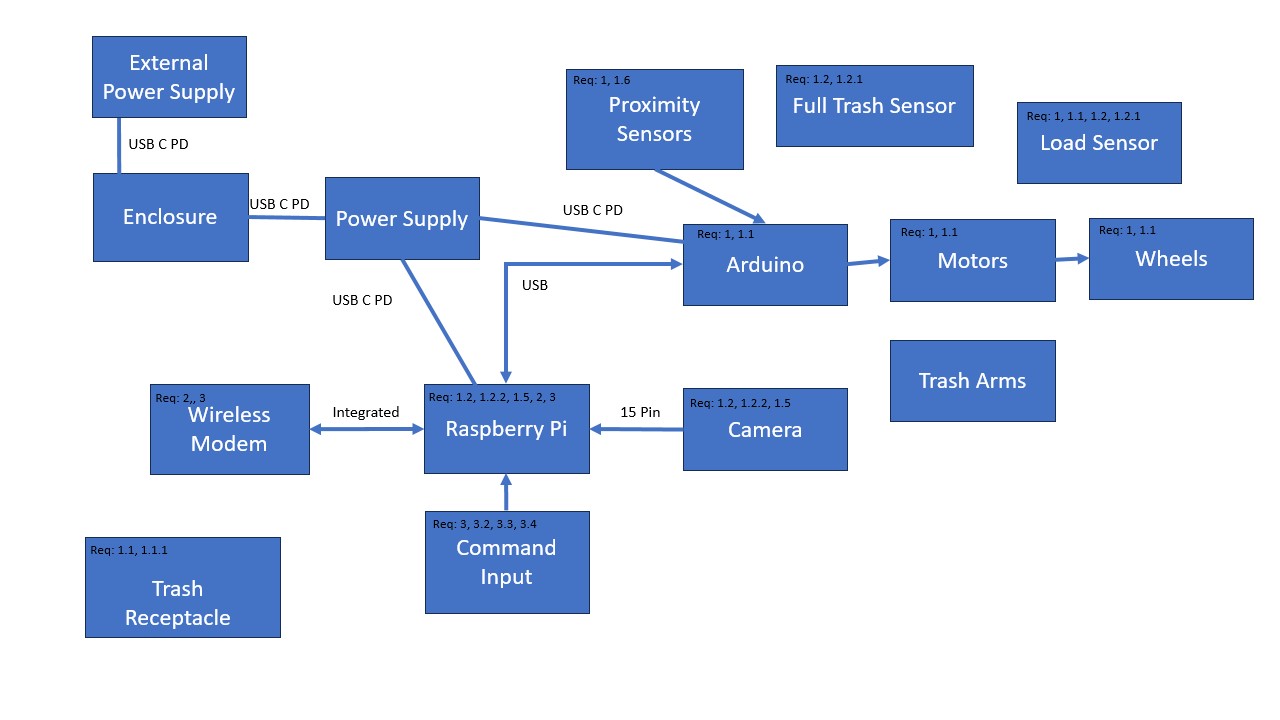
CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

System Architecture - As Built

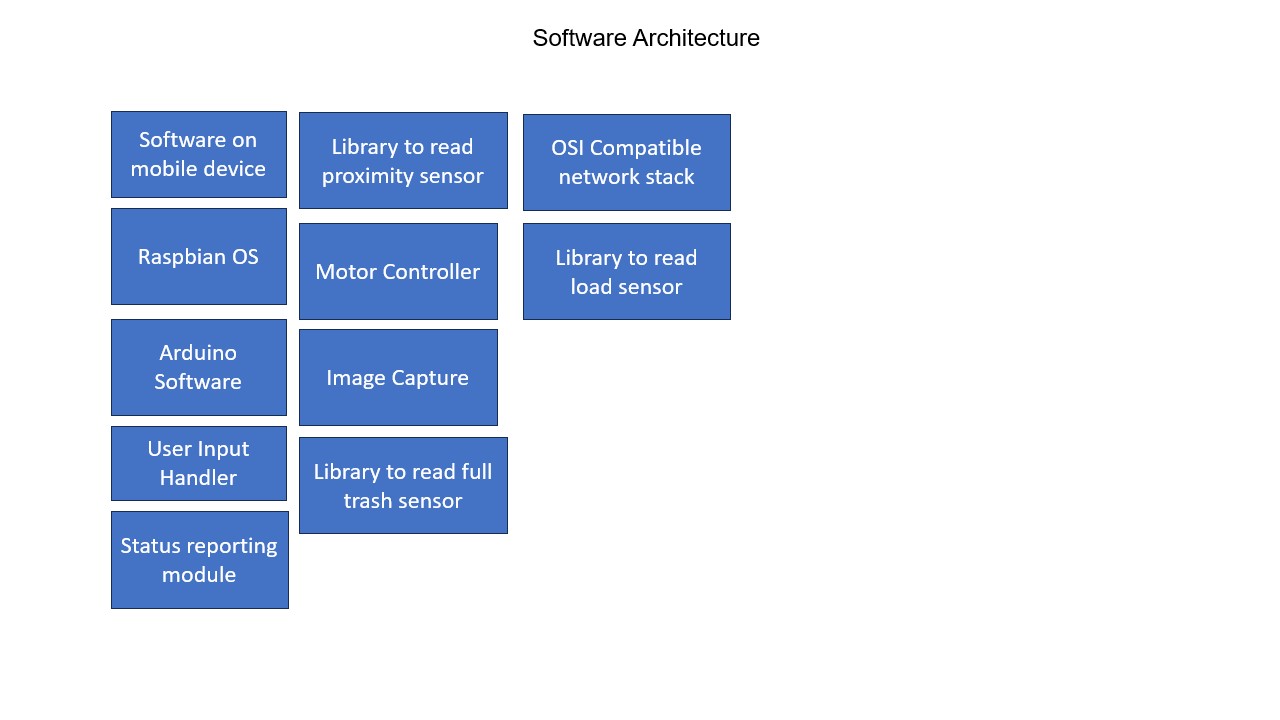
Hardware Architecture - As Built



Hardware Functional Block Definitions

1. Enclosure: Provides physical support for all internal components including microcontrollers and power supply. Includes a port for an external power supply to charge the power supply.
2. Camera: Used for AI detection, informs Bing Bong when it detects a person.
3. Load Sensor: Detects the weight of trash in the trash container.
4. Wireless Modem: Communication with the app for commands and notifications.
5. Proximity Sensors: Sensors that detect Bing Bong’s proximity to nearby obstacles.
6. Arduino: Microprocessor for reading sensors and controlling motors.
7. Raspberry Pi: General purpose computer for using machine learning model to follow a person.
8. Motors: Motors will be used for the wheels. The motors will convert our power from the power supply into torque.
9. Trash Arms: Use energy from motors to empty the trash container.
10. Wheels: Wheels use motor energy to spin, allowing movement.
11. Power Supply: Internal battery that provides power to Bing Bong’s other internal components. This allows Bing Bong to run without a connection to an outlet.
12. External Power Supply: Provides power to the internal power supply.
13. Command Input: Virtual buttons on the android app. If pressed, the respective command is sent to the Raspberry Pi which prompts the Raspberry Pi to execute said command.
14. Full Trash Sensor: Edge sensor that detects when the trash is overflowing. This will also be a proximity sensor.
15. Trash Receptacle: Container that holds the trash.

Software Architecture - As Built



Software Functional Block Definitions

1. Software on Mobile Device: Application on a mobile device to allow the user to send Bing Bong commands
2. Raspbian OS: A Linux operating system for the Raspberry Pi
3. Arduino Software: Arduino libraries such as pulseIn are used by sensors
4. User Input Handler: Allows the user to communicate with Bing Bong
5. Motor Controller: PWM pins on Arduino
6. Image Capture: Camera used by the Raspberry Pi
7. Library to Read Proximity Sensor: Suitable Arduino library and functions
8. Library to Read Load Sensor: Suitable Arduino library and functions
9. Library to Read Full Trash Sensor: Suitable Arduino library and functions
10. OSI Compatible Network Stack: Raspbians implementation of the OSI stack for Wi-Fi communication.
11. Status Reporting Module: Part of the mobile app for notifications

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

**System Design - As Built**

Hardware design - As Built

1. Enclosure
   1. Needs to be smaller than the size constraints.
   2. Needs to be large enough to hold the hardware.
2. Camera
   1. At least a 2-megapixel camera used for object recognition of a person.
3. Load Sensor
   1. Needs to measure up to at least 1 kg +- 0.05 kg.
4. Wireless Modem
   1. The one built into the microprocessor.
5. Proximity Sensors
   1. Chose ultrasonic
   2. Detect objects at least 0.5 inches close to 5 feet away.
6. Microcontroller:
   1. AVR microcontroller.
7. Microprocessor
   1. Needs to be capable of running TensorFlow.
8. Motors
   1. Operating voltage must range from 6V to 12V.
9. Trash Arms
   1. Made of plastic.
10. Wheels
    1. It needs to be strong enough to function properly under a load of at least 25 kg.
11. Power Supply
    1. Needs to supply at least 5V power.
    2. Needs to supply at least 1 amp current.
    3. Needs to last at least 1 hour supplying 1 amp of current.
12. External Power Supply
    1. Needs to supply at least 5V power.
    2. Needs to supply at least 1 amp current.
13. Command Input
    1. Buttons on the app that send commands to the microcontroller.

Software design - As Built

1. Software on Mobile Device
   1. Android
2. Raspbian OS
   1. A suitable OS for the microprocessor will need to be able to install TensorFlow libraries.
   2. It does not need a GUI as we won’t have a screen.
3. Arduino Software
   1. Use built-in libraries and their respective functions or manipulate the hardware registers.
4. User Input Handler
   1. Take user input and send the information to complete the task given.
5. Motor Controller
   1. PWM on the Arduino or more abstracted functions.
6. Image Capture
   1. Drivers are compatible with the operating system.
   2. The library is available in the same programming language used in the main program.
7. Library to Read Proximity Sensor
   1. pulseIn from standard library
8. Library to Read Load Sensor
   1. HX711 library
9. Library to Read Full Trash Sensor
   1. pulseIn from standard library
10. OSI Compatible Network Stack
    1. The library is installed with the microprocessor. No need to use another.
11. Status Reporting Module
    1. Java libraries that hook into the app.

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

**HW/SW Design - As Built**

HW - As Built

|  | Design Options - Cameras |
| --- | --- |
| Criteria | *Option 4*  ***IMX519 Autofocus Camera*** |
| *Resolution*  *Req 1, 1.6* | 16 MP |
| *Connector*  *Req 1.2.2* | FPC |
| *Operating Voltage*  *Const 3* | 3 V |
| *Price*  Const 7 | $25 |

For the cameras, we used the IMX519 Autofocus cameras. This option was within our budget with the resolution we needed to detect a person with better accuracy. This sensor is also ideal because it interfaces directly with the Raspberry Pi connector.

|  | Design Options - Proximity Sensors |
| --- | --- |
| Criteria | *Option 1*  *HC-SR04* |
| Working Voltage  Const 3 | 5V |
| Min Range  Req 1.6, 1.2 | 2cm |
| Max Range  Req 1.6, 1.2 | 4cm |
| Price  Const 7 | $9 |

For the proximity sensor, we used the HC-SR04. due to its range. Bing Bong will need to detect objects close and far and the HC-SR04 gives us the largest range. This sensor is also ideal since it has a digital output.

|  | Design Options - Load Sensor |
| --- | --- |
| Criteria | *Option 1*  *(SEN-10245)* |
| *Weight Capacity*  *Req 1.1* | 50kg |
| *Operating Voltage*  *Const 3* | 10V |
| *Price*  Const 7 | $4.50 |

For the load sensor we used the SEN-10245 sensor, this sensor detects a larger load than we anticipate Bing Bong will carry and is also cheaper than our other options. With the required HX711 this sensor is ideal as it has digital output.

|  | Motors |
| --- | --- |
| Criteria | *Option 2*  *(Smart Car Parts Kit)* |
| *Torque*  *Req 1.2, 1.3.1, 1.4.1, 1.6* | .26kg. cm |
| *Votlage range*  *Const 3* | 12V DC |
| *Speed*  *Req 1.2, 1.6, 1.4.1* | 4310 rpm |
| *Price*  Const 7 | $0 Included with wheels |

For the motors we have decided to use the DC motor smart car part kit for the wheels due to the motor being in the right voltage range and its speed.

|  | Design Options - Single board computer |
| --- | --- |
| Criteria | *Option1*  *Raspberry Pi 5* |
| System  Req 1, 1.2.2, 1.4.1, 3.1, 3.2 | *Ubuntu 23.10 Desktop and server* |
| Video support  Req 1.2.2, 1.4.1, 1.6 | VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.2 |
| Processor  Req 1.2.2, 1.4.1, 1.6 | Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, with Cryptographic Extension, 512KB per-core L2 caches, and a 2MB shared L3 cache |
| Price  Const 7 | $60 each |

We used the Raspberry Pi 5. We chose the Raspberry Pi 5 because it has more capabilities than our other options while still being within our budget.

|  | Design Options - Microcontrollers |
| --- | --- |
| Criteria | *Option 2*  *Arduino Mega 2560 R3* |
| Working Voltage  Const 3 | 5V |
| Chip  Req 1, 1.3.1, 1.4.1, 1.6 | ATmega2560 |
| Price  Const 7 | $49 |
| Memory  Req 1, 1.4.1, 1.2.1, 1.3 | 8kB SRAM, 256kB flash, and 4kB EEPROM |
| Communication  Req 2.1, 3, 1.3, 3.1, 3.2 | UART, IC2, and SPI |
| Special Features  Req 1, 1.4.1, 1.6 | Large form factor and serial ports |

We used the Arduino Mega. Arduino Mega meets our requirements and it is well known throughout our team.

|  | Design Options - Wireless Module |
| --- | --- |
| Criteria | *Option 3*  *(Built-in Raspberry-PI 5)* |
| *Communication Range*  *Req 2* | 91.44m |
| *Frequency*  *Req 2* | 2.4-2.5GHz |
| *Operating Voltage*  *Const 3* | 5.1v |
| *Price*  *Const 7* | $0.00(Comes included with the $60.00 raspberry pi) |

We used the built-in Raspberry Pi wireless module. The built-in Raspberry Pi wireless module comes with the Raspberry Pi, meets our requirements, and is a tool our team is familiar with.

|  | Design Options - Wheels |
| --- | --- |
| Criteria | *Option 3* *Smart Car Parts Kit* |
| *Price*  *Const 7* | $18 |
| *Voltage (DC)*  *Const 3, Req 1, 1.1, 1.2, 1.3.1, 1.4.1, 1.6* | 3V-6V |
| *Current*  *Req 1, 1.1, 1.2, 1.3.1, 1.4.1, 1.6* | 150mA-200mA |
| *Speed*  *Req 1, 1.1, 1.2, 1.3.1, 1.4.1, 1.6* | 90 rpm - 200 rpm |

We used the Smart Car Parts Kit. This kit is the cheapest and comes with 4 wheels, 4 motors, and 2 motor controllers.

|  | Design Options - Internal Power Supply (Battery) |
| --- | --- |
| Criteria | *Option 3*  *SunFounder - Raspberry Pi Power Expansion Board* |
| ***Price***  *Const 7* | $29 |
| ***Voltage - Out***  *Const 3* | 5V |
| ***Current - Out***  *Const 3* | 3A |
| ***Chemistry***  *Req 4.0* | Lithium-Ion |

We used SunFounder - Raspberry Pi power expansion board. This option met all of our requirements.

|  | Design Options - Filament for enclosure and arms |
| --- | --- |
| Criteria | *Option1*  *Polycarbonate* |
| Tensile Strength  Req 1.4 | *67.57 MPa* |
| Temp resistant  Req 1.4 | 100°C |
| Flexural strength  Req 1.4 | 1880 MPa |
| Price  Const 7 | $28.99 1kg spool |

For the filament, we used the polycarbonate option due to its strength, temperature resistance, and price.

|  | Design Options - External Power Supply |
| --- | --- |
| Criteria | *Option 4*  ***12V Wall Adapter*** |
| Battery Type  Const 5, 3 | *any* |
| Output  Const 3 | *12V* |
| *Price*  Const 7 | $6.50 |

We used a12V Wall Adapter. It allows us to use any battery type while providing a solid high voltage for a cheaper price than the other options.

SW - As Built

|  | Software on the mobile device |
| --- | --- |
| Criteria | Option 1  Android |
| *IDE*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | Android Studios |
| *Cost*  Const 7 | free |
| *Available libraries*  *Req 2, 3* | Mqtt communication |
| *Phones that support the software*  *Req 3, 3.1, 3.2, 3.3, 3.4* | Samsung phones, google pixels |

We used Android, option 1. Android is a larger platform with more libraries and documentation. It is also free and we are able to test our software on our own devices.

|  | Single board computer |
| --- | --- |
| Criteria | *Option 1*  *Raspberry Pi OS* |
| *Access*  *Req 2, 3* | All open source |
| *Size*  *Const 7* | 8G |
| *Available libraries*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | Runs Arduino IDE, more libraries connecting Arduino and Raspberry Pi |

We used Raspberry Pi OS, option 1. The Raspberry Pi OS has a lot more compatible libraries for the Arduino and is completely open source.

|  | Arduino Software |
| --- | --- |
| Criteria | *Option 1*  *Arduino IDE 2.2.1* |
| *Compatible* | Uno Rev3 |
| *Languages*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | C++ |
| *Price*  Const 7 | $0 |
| *Access*  *Req 2, 3* | Open source |

We used Arduino IDE 2.2.1, option 1. Arduino IDE 2.2.1 meets all of our requirements and is software that our team is familiar with.

|  | User input handler | |
| --- | --- | --- |
| Criteria | Option 1  Android Studio | Option 3  Single-board computer scripts |
| *Language*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | Java, C++, Go | Python is preferred, everything available |
| *IDE*  *Req 2.1, 3.1, 3.2, 3.3, 3.4* | Yes | No |
| *Wireless*  *Req 2, 3* | Yes | Yes |

For handling user input on the receiving side, we used a single-board computer (option 3). The single-board computer scripts are able to communicate with the trash app while the other options are not as easy to implement.

For handling user input on the transmitting side, we used Android Studio (option 1). Arduino studio is going to be used to create the trash app and within the app, it will be able to handle user input.

|  | Status reporting module | |
| --- | --- | --- |
| Criteria | Option 1  Android Studio | Option 3  Single-board computer scripts |
| *Language*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | Java, C++, Go | Python is preferred, everything available |
| *IDE*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | Yes | No |
| *Wireless*  *Req 2, 3* | Yes | Yes |

For handling status reporting on the transmitting side, we used single-board computer scripts (option 3). The single-board computer scripts are able to communicate with the trash app while the other options are not as easy to implement.

For handling status reporting on the receiving side, we used Android Studio (option 1). Arduino studio is going to be used to create the trash app and within the app, it will be able to handle status reporting.

|  | Image capture |
| --- | --- |
| Criteria | Option 1  Tensorflow Lite |
| *Price*  Const 7 | free |
| *Language*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | Python |

We used Tensorflow Lite because our team is most familiar with it.

|  | Library to read proximity sensor |
| --- | --- |
| Criteria | Option 3  Arduino Standard Library |
| *# of Function Calls*  *Req 2.1, 3.1, 3.2, 3.3, 3.4* | Few |
| *Size*  *Const 7* | 0B |
| *Language*  *Req 1.2.1, 1.4.1, 1.6* | C++ |

We used the Arduino standard library. We need two proximity sensors. One for detecting a full load and one for obstacle avoidance.

|  | Library to read load sensor |
| --- | --- |
| Criteria | Option 1  bogde/HX711 |
| *Size*  *Const 7* | 19KB |
| *Language*  *Req 2, 2.1, 3.1, 3.2, 3.3, 3.4, 3.5* | C++ |

We used HX711. The HX711 library meets all of our requirements.

Group 04 Bing Bong

Major: Team members:

CEG/CS Megan Cruz

EE Joshua Moses

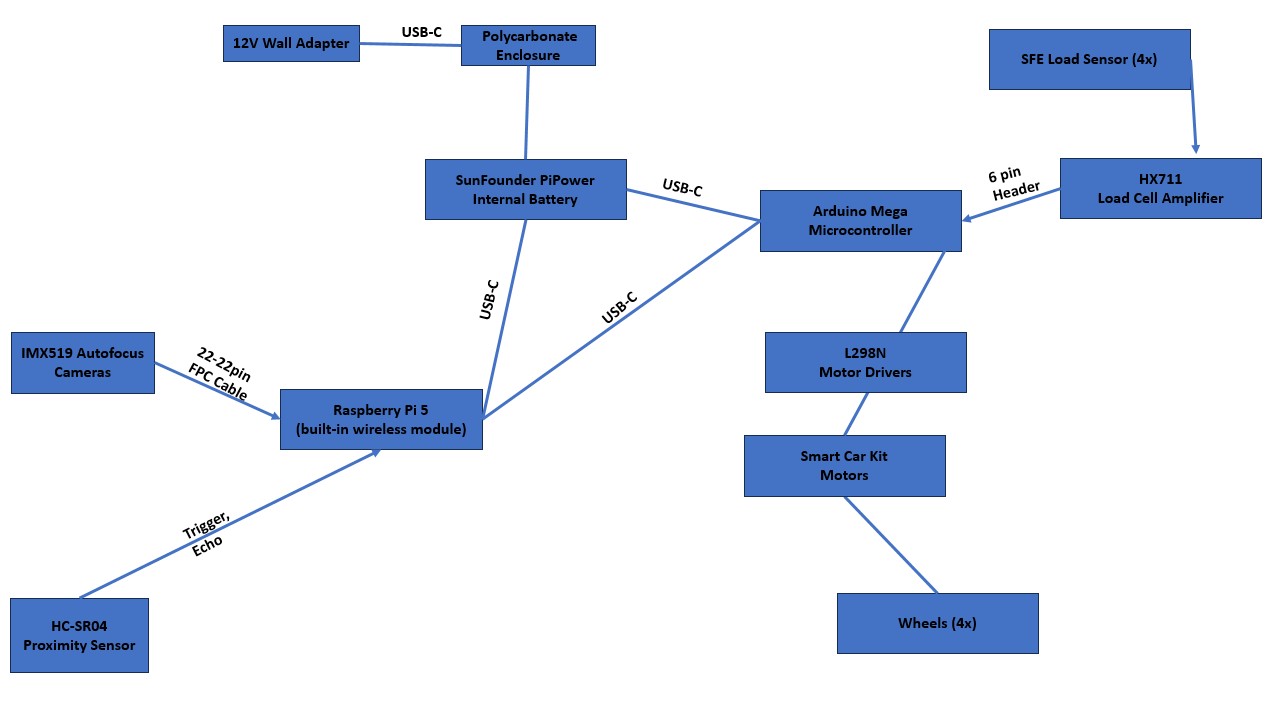
CS Eric Foy

CS Anthony Curry

CEG Samuel Medley

**Implementation - As Built**

**Interface Control - As Built**



**Budget - As Built**

| **Name** | **Vendor** | **Cost** |
| --- | --- | --- |
| Raspberry Pi 5 | Sparkfun | $80 without shipping |
| Arduino Mega | Amazon | $22.56 |
| IMX519 Autofocus camera | RobotShop | $24.99 |
| HC-SR04 proximity sensor | Amazon | 2 × $6.99 |
| 4 x SEN-10245 - Load Sensor | Amazon | Included with HX711 |
| 1 x HX711 - Load Sensor Amplifier | Amazon | $9.66 |
| 4 x TT Motor DC Motors | RobotShop | Included with Smart Car Parts Kit |
| SunFounder PiPower - Internal Battery | Amazon | $32.24 |
| Polycarbonate - Filament for | Amazon | $29.99 |
| 12V Wall Adapter | Sparkfun | $6.50 |
| Smart Car Parts Kit (4 Wheels, 4 Motors, and 2 Motor Controllers) | RobotShop | $18 |

**Schedule - As Built**

**Overview of Tasks - As Built:**

AI, 3D printing - Megan

Movement, Proximity - Eric

Trash - Josh, Sam

App - Anthony, Eric, Megan

**Breakdown of Tasks - As Built:**

GUI app or web application

* Downloaded android studios
* Created accounts
* Started with hello world template
* Added buttons for different Bing Bong actions
* Completed look of GUI
* Demonstrated the GUI portion of the app

App wireless connected to Bing Bong

* Picked what type of connection will be used
* Setup the wireless module on raspberry pi
* Demonstrated the wireless module working on the raspberry pi
* Created a function that allows the connection between the app and raspberry pi by using the IP address
* Demonstrated the connection of the app and raspberry pi by plugging in the IP address

App Commands

* App GUI has clickable buttons.
* Programmed the button to tell Bing Bong to turn off.
* Programmed the button to tell Bing Bong to take out the trash.
* Programmed the button to tell Bing Bong to follow the person in front of it.

Raspberry pi set up

Bing Bong designed (where do motors go.. Raspberry pi… Arduino etc)

* Created sketch of where part will be placed on Bing Bong
* Built out of tensorflow and camera

Movement of Bing Bong

* Proximity sensor for object avoidance
  + Received part proximity sensor.
  + Hooked proximity sensor up to the Arduino.
  + Programmed proximity sensor to detect distance from object.
  + Programmed detected distance to update the motors to avoid close objects.
* Motors and wheels assembled
  + Received part motors.
  + Received part wheels.
  + Hooked wheels up to the motors.
  + Hooked motors up to the motor controller.
  + Hooked motors with wheels up to a simple bed.
  + Connected motors to the Arduino digital pins.
* Programmed DC motors
  + Programmed motors to go forward.
  + Programmed motors to go backwards.
  + Programmed motors to stop.
  + Programmed motors to turn.

Trash

* Detected when full load
  + Received load sensors.
  + Connected load sensors to the load sensor amplifier.
  + Connected load sensors to the Arduino.
  + Communicated between the load sensor and the Arduino
* Detected when full amount with proximity sensor - programmed trash proximity sensors
  + Received part proximity sensor.
  + Hooked proximity sensor up to the Arduino.
  + Programmed proximity sensor to detect a close object that stays there. That will indicate that the trash is full.

CAD design of case for Bing Bong

* Dimensions of Bing Bong after being built
* Prototyped Bing Bong enclosure
* Printed first run of enclosure
* Examined enclosure
* Made adjustments for final enclosure
* Printed second enclosure

Connected Raspberry Pi to Arduino

* Decided to use either USB.
* Programmed sending data from the Raspberry Pi to the Arduino.
  + Tested that this data is received on the Arduino
* Programmed sending data from the Arduino to the Raspberry Pi
  + Tested that this data is received on the Raspberry Pi
* Software on the Raspberry Pi to handle data received from the Arduino.
* Software on the Arduino to handle data received from the Raspberry Pi.

Created AI model

* Set up Tensorflow Lite
* Created AI models
* Trained AI Model

Connected the AI model to the motor controls using USB serial.

**Critical Milestones - As Built:**

Orders sent to be approved **12/8/2023**

Orders approved **2/8/2024**

Ordered parts **2/21/2024**

Received ordered parts **3/19/2024**

Demonstrated first data communication between Raspberry Pi and Arduino **1/31/2024**

Demonstrated first data communication between Raspberry Pi and App **3/13/2024**

Bing Bong hardware assembled **3/4/2024**

Demonstrated autonomous movement **2/19/2024**

Demonstrated the app turns Bing Bong off. **4/19/2024**

Demonstrate Bing Bong following a person. **4/14/2024**

Demonstrated the app tells Bing Bong to start following the user. **4/19/2024**

Bing Bong completed. **4/20/2024**

**Risk Analysis - As Built**

**Risk 1:** Snow days. (Likelihood: 1 of 5, Severity: 3 of 5)

**Mitigation:** We ensured that there is extra time in the schedule to make up for a few missed days. (Likelihood: 1 of 5, Severity: 1 of 5)

**Risk 2:** Delayed delivery of parts for the project. (Likelihood: 3 of 5, Severity: 4 of 5)

**Mitigation:** We ordered parts as soon as possible to avoid this problem. This did happen. They forgot to order our parts until the end of February. We received them at the end of March. (Likelihood: 5 of 5, Severity: 2 of 5)

**Risk 3**: Team members being frequently tardy. (Likelihood: 2 of 5, Severity: 3 of 5)

**Mitigation:** We sent Discord messages to tardy group members. (Likelihood: 1 of 5, Severity: 2 of 5)

**Risk 4:** CAD design of Bing Bong takes too long. (Likelihood: 3 of 5, Severity: 3 of 5)

**Mitigation:** Used a wooden framework for a week but the CAD design did not take too long. (Likelihood: 2 of 5, Severity: 1 of 5)

**Risk 5:** If a task is proving difficult or if the task is taking too long for a given team member. (Likelihood: 3 of 5, Severity: 4 of 5)

**Mitigation:** Added team members to the task. (Likelihood: 2 of 5, Severity: 2 of 5)

**Risk 6:** The project is not on schedule. (Likelihood: 1 of 5, Severity: 4 of 5)

**Mitigation:** We ensured that there was extra time in the schedule to ensure that we finished on time. (Likelihood: 1 of 5, Severity: 2 of 5)

**Risk 7:** Parts arrive broken or we break them. (Likelihood: 2 of 5, Severity: 3 of 5)

**Mitigation:** While we waited for them to be re-ordered we worked on the software portions of our project. (Likelihood: 2 of 5, Severity: 1 of 5)