An Introductory Exploration of Electronics and Robotics

Luke Renchik

5/1/22-8/1/22

This paper is meant to serve as a commentary and abridged history for the creation of Dido The Robot.

The purpose of this project was to build a robot capable of navigating a basic obstacle course, to become more familiar with electronics, electrical instruments, motors, sensors, general engineering principals, and the engineering project workflow.

Preface: Finding Interest in Electronics

During the spring of 22’ I was deep in my coursework for Calculus 1, Trigonometry, and Python Web Development. The challenge of the courses excited me but I found myself yearning to find an application for these newfound skills. I began reading texts on Introductory Physics and Chemistry, the theoretical and mathematical concepts were intellectually stimulating, but I felt the need to get hands-on. I ordered an Arduino Mega kit to begin experimenting with various circuits, sensors, and the Arduino development environment.

My Calculus 1 course was wrapping up in late April 22’ when I decided I wanted to spend the summer exploring the fields of Physics, Electronics, and Engineering.

I enrolled in a pair of introductory physics courses covering Newtonian mechanics and electricity and magnetism, introductory chemistry, as well as a course in C++. To reinforce, strengthen, and extend the lessons I was learning in the classroom I decided to pursue an independent laboratory project.

After a few weeks of researching possible project topics, I had narrowed my interest to exo-skeleton systems, flight, and robots. I ultimately decided on a robot as it was the field that seemed like it could make the biggest impact on humanity.

As I began researching all of the components that are incorporated into building a robot, I was quickly humbled. I had extremely limited experience in the mechanics of a vehicle, the electronics necessary to power motors, and the C++ coding environment. The task ahead would be challenging, but serve as an incredible vessel for learning and growth.

Stage 1: Creating A Toolset

To begin working on this project a workshop equipped with tools and testing equipment is necessary. I have highlighted a few of the crucial tools needed to build the robot below, A full list of the components used can be found in the index.

The first tool necessary is a 3D-Printer, the print used for this project was the Creality Ender 3 v2, a budget-friendly, entry level printer. This should be paired with a CAD software to design parts (this project used a student copy of AutoDesk 360) and Ultimaker Cura 5.0 to convert designs to print files.

Some additional tools used were: a variable temperature soldering iron, nuts, bolts, washers, bearings, DC motors, Servo motors, battery holsters, a digital caliper, and a hot glue gun.

The DC motors require a near 9V charge to run at full power, a rechargeable battery bank and a pair of rechargeable 9 Volt batteries prove to be extremely helpful when testing the project.

Stage 2: The Chassis

The initial chassis design can be found in the index. The design was limited by two logistical challenges listed below.

1.) Designing a layout and mounting system for the necessary components into the pyramidal shell.

2.) Creating a motorized tread system that would attach to the chassis.

Due to the time and experience limitations, designs created by other engineers were used to complete the project.

SMARS Chassis Design

The SMARS Chassis design retained the tread style wheels while creating a lower profile build compared to the initial pyramidal design. It housed its 9V battery below a surface mounted Arduino Uno and a motor controller shield. The pieces in the model were designed to snap together and the build would not require any screws or bolts lowering the upfront cost in materials.

This “snap-together” design became the SMARS’s greatest downfall. The Ender 3 was unable to print the intricate locking mechanism of the treads with enough precision for them to be functional. There was an attempt to bind the treads together by using a spray adhesive when laid in the locking position, the spray adhesive created a gummy film over the treads without giving any additional stability to the locking mechanism, cementing the tread system a failure.

Without the tread system, the two octagonal wheels inhibited the power translated from the battery to the forward progress of the vehicle. Additionally the low profile of the chassis limited the potential for strong DC motors to power the vehicle. This ultimately caused a design shift to the SCRU-FE model chassis.

SCRU-FE v1

After the failure of the SMARS design it was decided that the project would pivot to the SCRU-FE framework. A tri-wheel design that sported a servo mounted ultrasonic sensor located at the top of chassis. The combination of these features make it look distinctly relatable, almost friendly.

The SCRU-FE v1 designs for the rear unpowered wheel and ultrasonic mounting system were both very weak. The rear wheel made of thin plastic wafers created a weakness that could have led to decreased performance and potential structural failures down the road. The ultrasonic mounting bracket suffered from the same weak plastic deficiency. These weaknesses prompted the shift to an alternate build of the model, SCRU-FE v2. A comparison photo of the two chassis can be found in the index.

SCRU-FE v2

The SCRU-FE v2 addressed the weaknesses in the rear unpowered wheel, ultrasonic mounting system, and made improvements to overall build quality for the center chassis.

The DC motors and rear unpowered wheel were assembled using a variety of different length M3 screws and nuts.

The ultrasonic sensor mount and control arm were assembled using a variety of M1 machine screws.

Stage 3: The Electronics

Dido is comprised of an Arduino Uno, a L298n H-Bridge, two DC motors, a HC-SR04 Ultrasonic Sensor, a HXT 900 Servo Motor, a 9V Battery to power the Arudino, and a 4.5V AA battery pack to power the L298n chip.

The Arduino is powered by a 9V rechargeable battery connected through a DC power cable and supplies power to the L298n, Servo Motor, and Ultrasonic Sensor. The Arduino board lacks sufficient power and ground outlets, to solve this a perf board was used to bridge voltage and ground connections to all of the necessary components.

The L298n H-Bridge takes positive and negative connections from both of our DC Motors, A 9V connection to power the motors, a 5V connection to run the internal logic, common ground, and six data cables back to the Arduino. This chip enables the individual motors to have independent speed and spin direction.

The ultrasonic sensor and servo are powered by the Arduino via a perf board that delivers 5v and ground. The echo, trigger, and servo movement data lines are fed into the digital inputs on the Arduino.

Dido wires with excess length were snipped to reduce clutter. Once the desired length was achieved the wires were soldered back together and covered with electrical tape to prevent the shorting of circuits.

Stage 4: Mounting

A major design challenge with Dido was mounting the electrical components to the chassis. In order to turn the Arudino and L298n into a single mountable piece, standoffs were used on the back side of each chip. This caused neither chip to have a favorable mounting orientation.

If the L298n was attached first, the extra distance caused by the heat sink would shift the center of mass and create a torque that would pull dido’s nose to the ground, rendering the rear unpowered wheel useless.

The Arduino would need to be mounted pins down while having no natural board contact or bolt posts to use. Ultimately this ended up being the more favorable orientation, the mounting challenge was solved with the use of two solid rectangle standoffs connecting the Arduino’s component side to the chassis with hot glue. The non-component side of the Arduino is connected to the non-component side of the L298n chip by table-shaped standoffs and hot glue. The perf board ended up glued to the end of the heat sink for ease of access and lack of mounting plan. No overheating problems or negative performance have been noticed from the boards attachment to the heat sink.

Stage 5: The Accessories

Throughout the project there were a number of minor components that were created and used, this section lists those components.

9V Battery Holder

Initially the 9V battery was attached to the chassis with hot glue to create a semi permanent hold. After testing the initial 9V battery no longer had the proper voltage to power all the motors at full speed.

This initial setup was overhauled with a battery cage that allows for the hot swapping of rechargeable batteries to reduce waste and maximize testing uptime. (Index)

Wall-E Eyes

The HC-SR04 ultrasonic sensor provided the perfect opportunity to give the robot a cute set of eyes, modeled after the Disney-Pixar film, Wall-E.

Electrical Tape

The data wires and power connections both caused clutter on the robot, and introduced friction against the tires, this caused Dido to naturally pull to the right, as there was more friction on the right wheel causing it to spin at a slower RPM. Electrical tape was used to minimize contact between the wire and the wheel, but ultimately the friction was never fully removed.

Rear Wheel Weights

While testing, occasionally when Dido came to a stop the forward momentum of the body caused it to tip off of the back wheel and rest upon the front of the PCB. To prevent this five equally spaced weights were added to the rear wheel. These additional weights moved the center of mass backwards helping the quick decelerations faced while testing. Additionally alterations were made to the devices software to help prevent this unwanted consequence.

Stage 6: The Software

The software for Dido was written in C++ within the Arduino development environment. Dido runs a simple movement algorithm powered by six unique functions. Quick descriptions of the functions are listed below, a logical flowchart of the algorithm is supplied in the index.

driveForward(): Set both motors in the forward direction.

driveBackward(): Set both motors in the backwards direction.

coast(): Turn both motors off

spinCounterClockwise(): Set left motor to reverse, right motor to forward for 0.15s.

spinClockwise(): Set right motor to reverse, left motor to forward for 0.15s.

echoWall(): Activate the ultrasonic sensor, calculate the distance to the nearest object, and return that calculation.

Testing

Dido was tested in a small semi-enclosed course. The most challenging aspect of the course was the dead end where Dido must turn around and drive itself out. Video recording of the testing can be found through the link in the title of this section.

Stage 7: Improvements

There are several aspects of this project that could be significantly improved by an updated model. Below is a list of those improvements.

Chassis Mounting System

The Arduino is currently mounted by standoffs hot glued between the board and chassis. While this method of mounting is functional, it is inefficient and messy. In an updated model the chassis would utilize the Arduino Uno’s mounting holes to bolt the board to the chassis.

Wiring Hole in Chassis

The current chassis construction lacks wiring management, causing multiple problems during development, the most pressing concern is that some stray wires create friction with the motors of the vehicle and alter its movement path. The updated model would have a hole within the center mounting plate to run wires internally.

Heat Shrink for Wires

During construction wires were shortened to minimize clutter, electrical tape was used to hold the connection and prevent shorts. In an updated model heat shrinking the wires would provide a stronger bond and greater longevity.

Sensors

IR Receiver - An IR receiver would equip Dido to be loaded with multiple programs which a user could select from with an IR remote.

Bump Sensor - Externally mounted bump sensors would enable Dido to relay when it has run into an obstacle un-noticeable by its ultrasonic sensor.

HC-05 Bluetooth Module - A bluetooth module would allow for live transfer of sensor data and the ability for remote control of Dido.

Arduino Motor Shield

While the L298n H-Bridge was an effective motor controller it hindered the form factor of Dido and proved difficult and inefficient to mount. An Arduino motor shield has an integrated L298P motor controller and attaches nicely to the Arduino Uno. This opens chassis design possibilities by reducing the vertical footprint of the chips.

Moveable Servo

The servo motor intended to move the HR-S04 ultrasonic sensor was rendered ineffective due to cluttered wiring and a protruding L298n chip. A moveable ultrasonic sensor will lead to more intelligent obstacle detection and avoidance.

Robotic Arm

A robotic arm would increase the functionality of Dido allowing it to perform simple tasks like picking up objects or pushing buttons.

A Rear Movement Ball Hotswappable Attachment

A hot swappable rear movement ball that acted similar to a ball computer mouse would provide interesting new movement possibilities due to the increased freedom of mobility. This would only be effective on glossy surfaces that are low friction.

A More Advanced Movement Algorithm

There were instances during testing where Dido required human intervention after becoming stuck on an object it could not detect. This would be corrected with additional sensors and an updated algorithm.

A More Challenging Obstacle Course

The first iteration of the testing course consisted of a small area with a single corner to test Dido’s ability to navigate. An updated course with increased difficulty and multiple objectives would help push the limits of Dido.

Conclusion

The purpose of this project was to build a robot capable of navigating a basic obstacle course, to become more familiar with electronics, electrical instruments, motors, sensors, general engineering principals, and the engineering project workflow.

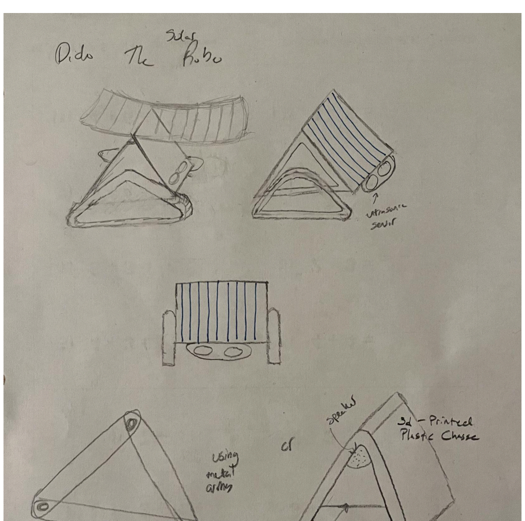
Dido was able to enter a semi-enclosed space, avoid collisions with walls, and find its way out of an enclosed corner. This project teaches how basic electronic devices are powered, how they communicate via data lines, and how to use a microcontroller as a home base for devices and sensors to work through.

By covering topics within mechanical, electrical, and software engineering the Dido project is a great introduction to the fundamental branches of engineering. Dido was able to successfully accomplish its designated tasks, yet there are numerous improvements that could be added with a second iteration of the project demonstrating the importance of prototyping and iteration within the engineering field.

Index.

Bill of Materials

Bill of Materials	
Item	QTY
Servo Motors	1
DC Motor	2
3x AA Battery Holder	1
M3 6mm-20mm Screw & Nut Kit	1
M3 16mm-35mm Screw & Nut Kit	1
M3 0.5mm Stainless Steel Washer Kit	1
60W Adjustable Temperature Soldering Iron	1
M1/1.2/1.4/1.6/1.7 Small Machine Screw Kit	1
Elagoo Uno	1
M/F Breadboard Jumper Wires	10
M/M Breadboard Jumper Wires	30
L298N Motor Driver	1
Perf Board	1
HC-SR04 Ultrasonic Sensor	1
Creality Ender 3 v2	1
Hot Glue Gun	1
Digital Caliper	1
9V Rechargeable Battery	2
Battery Recharging Station	1



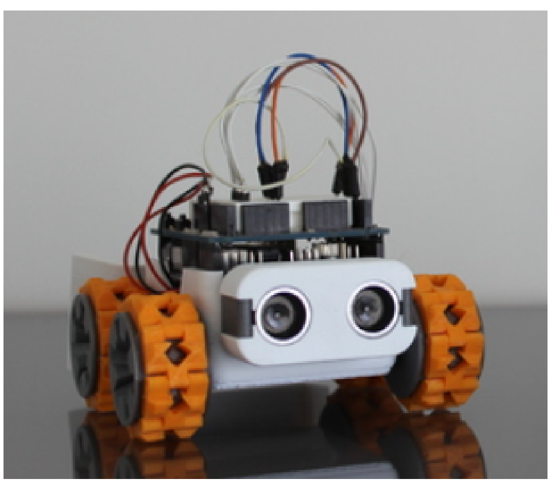
Dido The Solar Robo

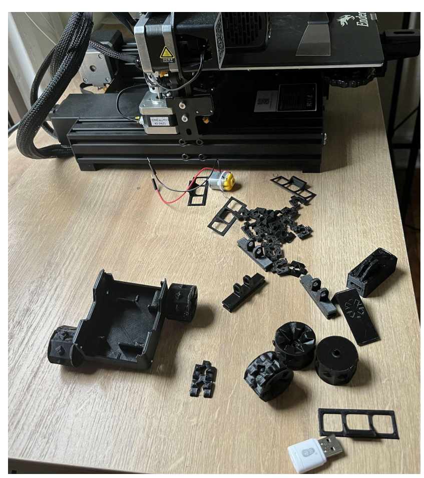
While updating my linkedin I found the [University of Michigan's Solar Car team](), an incredible project with decades of history trying to utilize our planet's most abundant source of energy to power one of the worlds most utilized inventions. Their project was the leading inspiration for the first conceptual design of Dido.

Initial Chassis Design

The pyramidal design critically lacked an attachment mechanism for the DC motors, a space plan for the battery packs, and a logical mounting position for the ultrasonic sensor.

SMARS Chassis Design

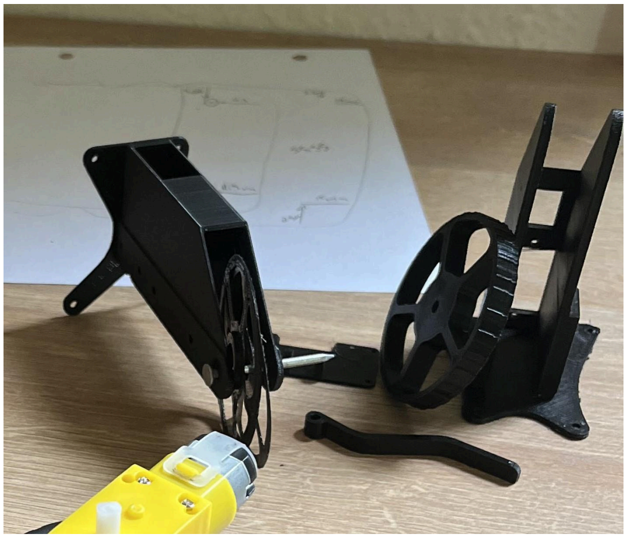




The SMARS chassis design had small linked treads (as pictured) if printed with a low degree of error would link together and form a chain. The Ender 3 was incapable of meeting that error threshold.

SCRU-FE v1 vs. SCRU-FE v2

The difference in build quality between the first and second generation chassis can be seen here, most noticeable is the change in the rear wheel. From two plastic wafers to a much sturdier wagon wheel design.



SCRU-FE v2 Chassis Assembly

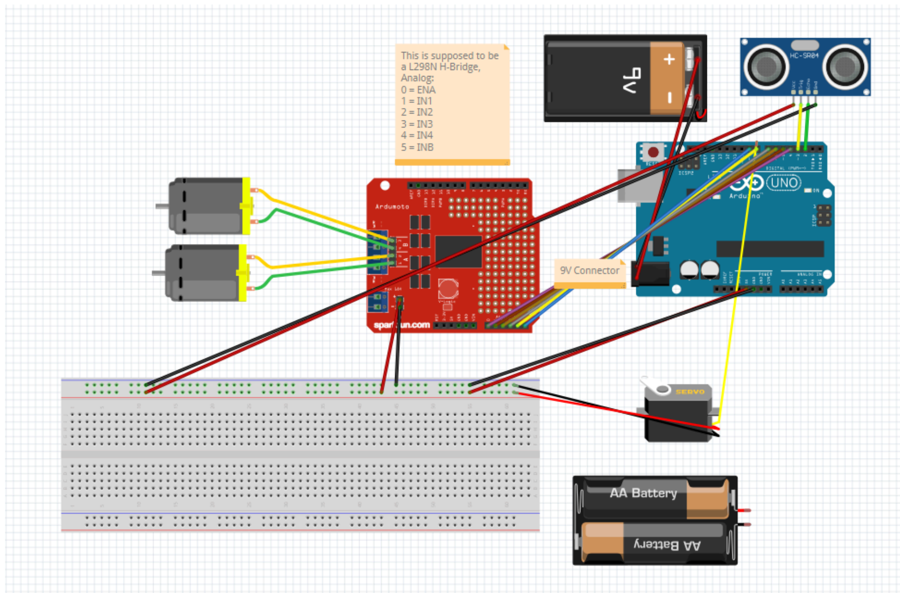
The DC motors and Rear Wheel are secured by M3 Screws with locking nuts. The front slot on the top of the body fits a servo motor which will hold the arm that houses the ultrasonic sensor. The rear hole is left for wiring convenience.

Dido Enjoys A Sunny Afternoon

You can't trap us in the lab ALL day.

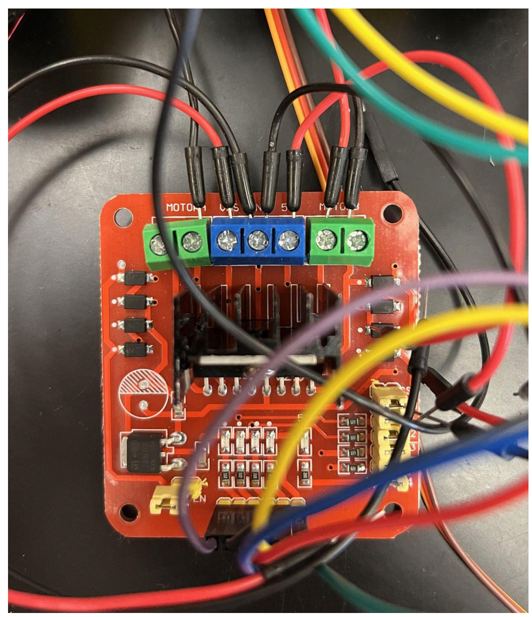
Dido Wiring Diagram

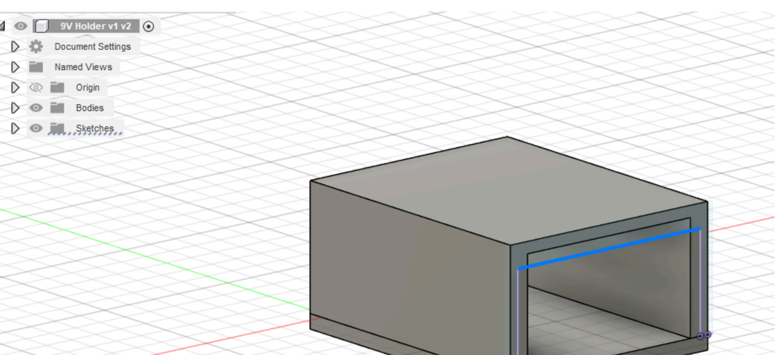
The initial wiring design using program Fritzing.



L298n H-Bridge

The H-Bridge acts as the motor controller for the Arduino Uno.



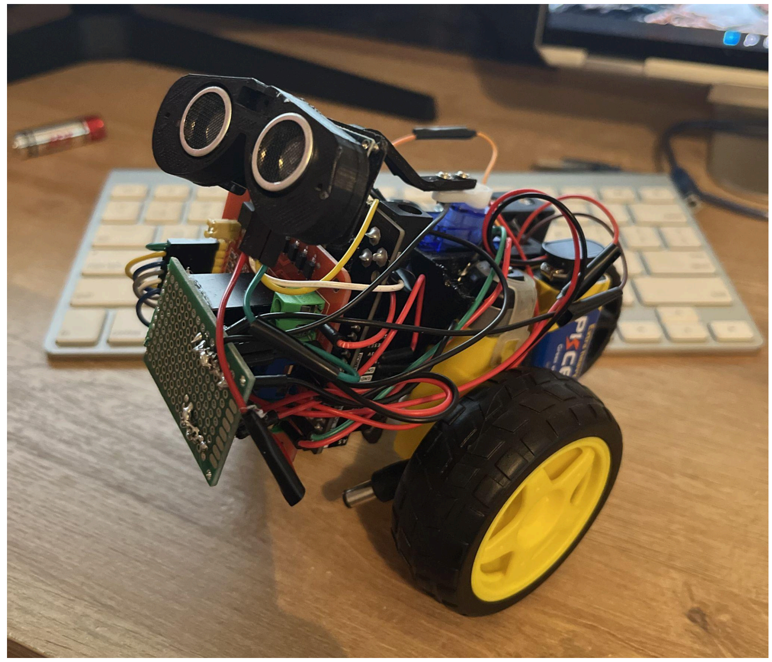


9V Battery Holder

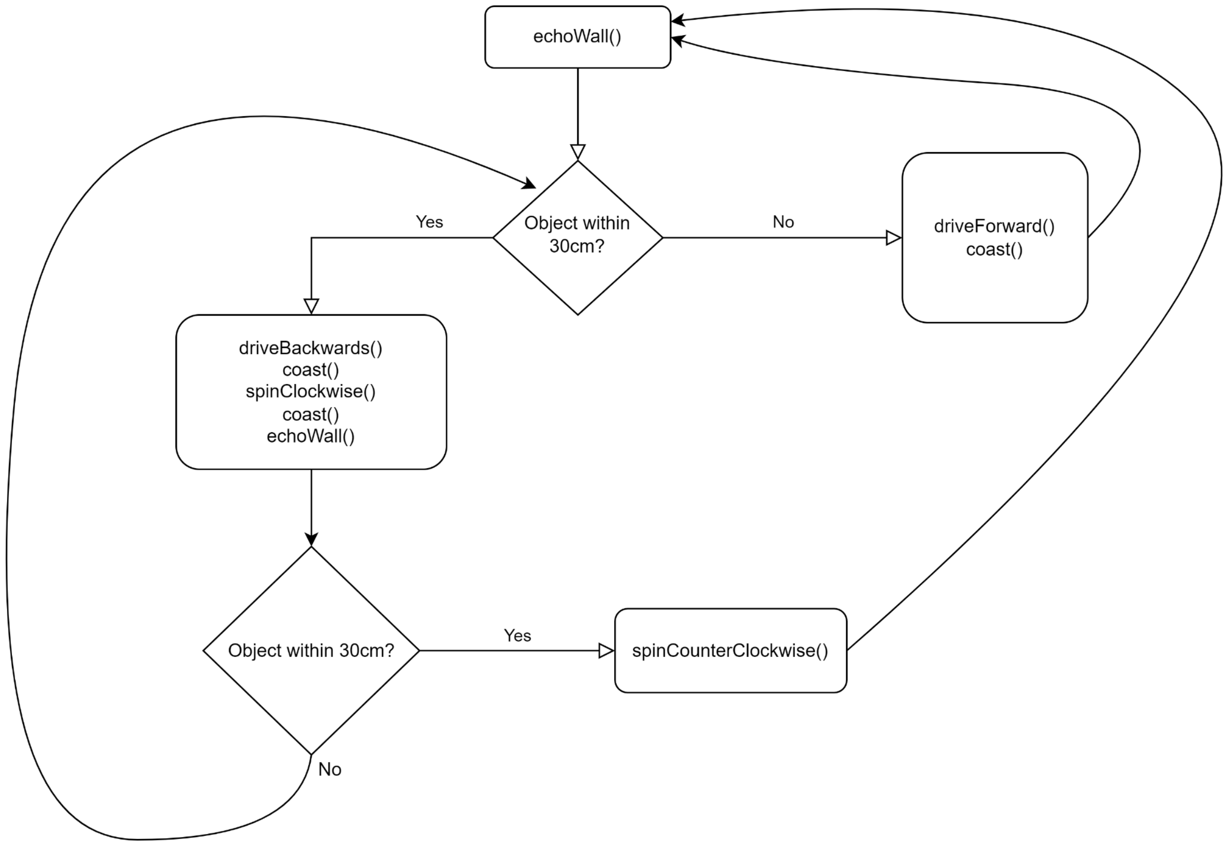
After the initial 9V battery ran out of power, a more sustainable solution was necessary.

Dido the Robot

I kept dreaming of a world I thought I would never see. And then, one day...



Dido Loop Logic


echoWall()
Object within 30cm?
Yes
No
driveForward()
coast()
driveBackwards()
coast()
spinClockwise()
coast()
echoWall()
Object within 30cm?
Yes
spinCounterClockwise()
No