**Robot Project Notes:**

**Phase 1:**

Dates: 9/1 – 9/29

BOM:

|  |  |  |
| --- | --- | --- |
| **Description** | **Part No.** | **Qty** |
| WWZMDiB L298N Motor Driver Controller Board DC Dual H Bridge Module | B0CR6BX5QL | 1 |
| DC12V Encoder Gear Motor with Mounting Bracket, 65mm Magnetic Motor | B07X7M1LLQ | 2 |
| Arduino UNO R4 WiFi [ABX00087] – Renesas RA4M1 + ESP32-S3, Wi-Fi, Bluetooth, USB-C, CAN, 12-bit DAC, OP AMP, Qwiic Connector, 12x8 LED Matrix | B0C8V88Z9D | 1 |
| Chassis | 3D printed | 1 |
| Wires | Generic | As needed |
| Breadboard |  | 1 |
| 12V DC Wall Wart | Generic | 1 |

Chassis used:

EricBot v3 model (first print)

A grey rectangular object on graph paper

AI-generated content may be incorrect.

EricBot v6 model (second printed version)

A grey rectangular object with holes

AI-generated content may be incorrect.

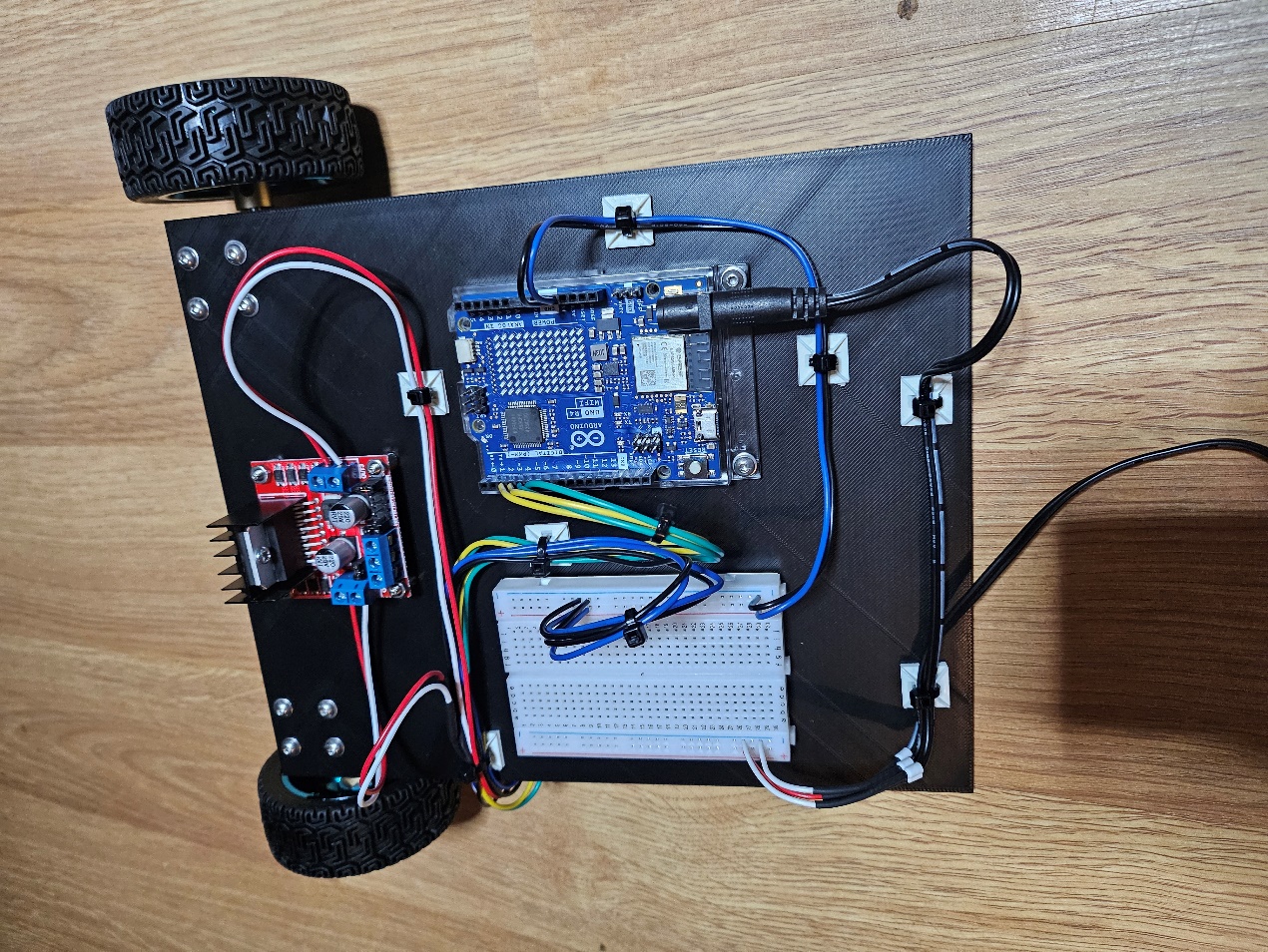
Robot Picture:

Revision 1: Mounted Wheels, fit check some components.

A black rectangular object with wheels and wires

AI-generated content may be incorrect.

Revision 2: Mounted components and wiring



Schematic:

A diagram of a circuit board

AI-generated content may be incorrect.

Code notes:

* VSCode IDE
* Connection to Arduino facilitated by PlatformIO
* Manually write encoder code, must use ISR to function well (polling doesn’t work well)
* Encapsulating ISR’s in Classes using static functions and pointers
* Drive motor using PWM on enaPins on L298N, direction defined by 2 input pins per motor
* Github: <https://github.com/eriico39hi/EricBot>

Goals for this phase:

* Read motor encoder in Arduino console - DONE
* Drive motors through L298N driver PCB’s

References:

* <https://www.reddit.com/r/arduino/comments/ll5avx/is_anyone_using_visual_studio_code_for_arduino/>
* <https://dronebotworkshop.com/platformio/>
* <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>
* <https://forum.arduino.cc/t/using-a-class-member-for-isr-is-it-possible/380127>
* <https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>

**Phase 2:**

9/29 – TBD

ROS2:

Learning ROS2 – Tutorials

~~Configuring environment~~

~~Using turtlesim, ros2, and rqt~~

~~Understanding nodes~~

~~Understanding topics~~

Understanding services

Understanding parameters

Understanding actions

Using rqt\_console to view logs

Launching nodes

Recording and playing back data

Beginner: Client libraries

Using colcon to build packages

Creating a workspace

Creating a package

Writing a simple publisher and subscriber (C++)

Writing a simple publisher and subscriber (Python)

Writing a simple service and client (C++)

Writing a simple service and client (Python)

Creating custom msg and srv files

Implementing custom interfaces

Using parameters in a class (C++)

Using parameters in a class (Python)

Using ros2doctor to identify issues

Creating and using plugins (C++)

Code notes:

* ROS2 and Ubuntu Raspberry Pi install
* VNC configuration – allows for sim windows – using xfce4 as gui

Goals for this phase:

* Communicate Arduino to ROS
* Create Kinematics to define robot in terms on m and deg
* PID tuning

**Phase 3:**

November?

Goals for this phase:

* Motion Program
* Write report