IR REMOTE CONTROL CAR

INTRODUCTION

Infrared radiation (IR) is a subset region of the electromagnetic spectrum that situates between microwaves and visible light.

Infrared radiation consists of two ranges: near infrared light is closest in wavelength to visible light, whereas far infrared light is closer to the microwave region of the EM spectrum. The shorter IR waves are utilized by remote controls. Information is transmitted and received using electromagnetic energy, without wires.

Wireless electronic systems offer a plethora of applications when examined through a pragmatic lens. Mechanical systems become more flexible; often requiring less components for operation with remote controlled electronics. Thus, the following project involves the design and construction of a remote-controlled car.

This report details the implementation of Arduino circuitry and IR signal processing components to transmit, receive and command the dual car motor system.

DEVICE HARDWARE DESIGN & JUSTIFCATION

The full functional hardware block diagram of the IR remote controlled car is displayed below **(Fig. 1)**. The device consists of blocks namely IR transmitter, IR receiver, microcontroller decoder, H-bridge, load driver.

The IR sensor module receives the IR pulses transmitted by the IR remote and converts the signals into corresponding electric pulses. These electric pulses are given to the microcontroller that decodes it into corresponding data bytes. The data bytes are utilized to partake further control decisions. The control output voltage is transmitted to the H-bridge circuit; thus, driving the actual device.

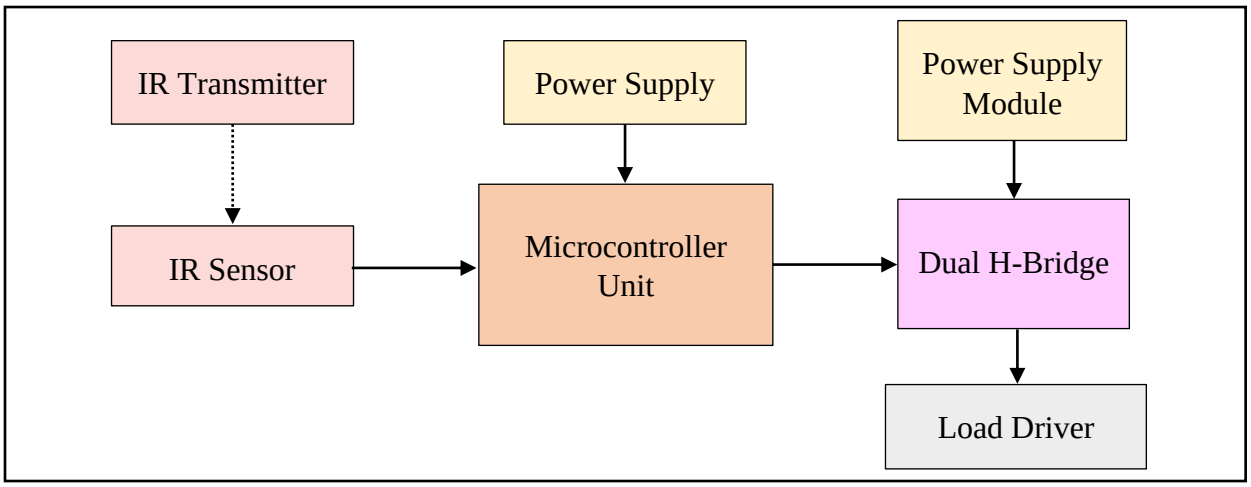

Figure 1: Hardware block diagram

Microcontroller Unit (MCU)

The system circuit schematic **(Fig. 2)** was designed around the Arduino MEGA 2560 REV3 microcontroller. MCU serves as the embedded system platform for hardware and software components. Consists bidirectional I/O digital pins responsible for relaying data from the IR sensor and outputting processed data by means of the Arduino programming software, Sketch. MCU uses calibrated 16 MHz internal crystal oscillator.

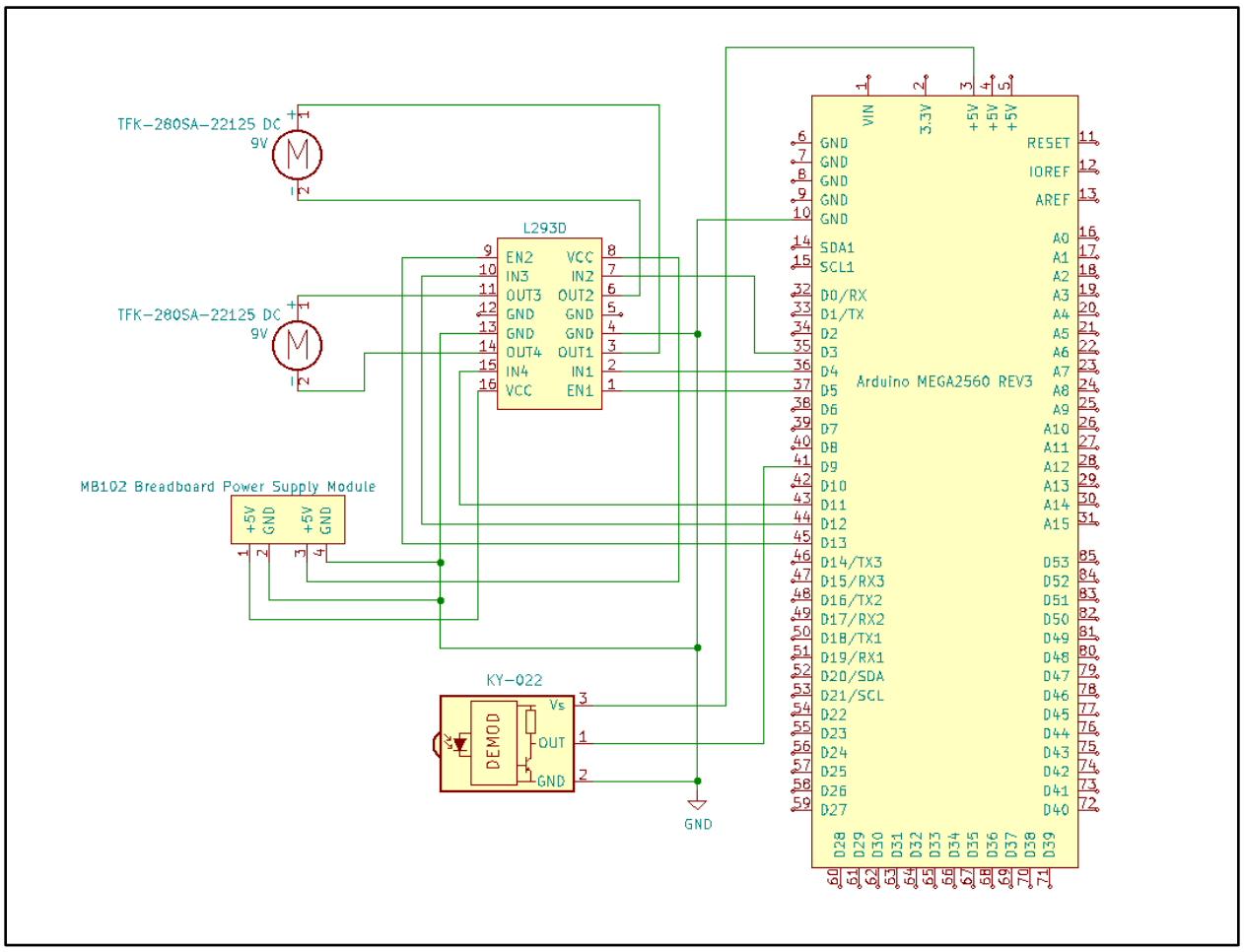

Figure 2: Circuit diagram of system

IR Transmitter

This electronic system used the Sony IR remote which uses 12-bit SIRC protocol **(Fig. 3a)**. The code begins with a header of 2.4ms followed by 7-bit and 5-bit command device address, respective; with LSB transmitted first. Commands are repeated every 45ms upon remote key is held.

The address and commands exist as binary logical ones and zeroes. 600µS (1T) space and 1200µS (2T) pulse form logical one. 600µS space and 600µS pulse form logical zero **(Fig. 3b)**.

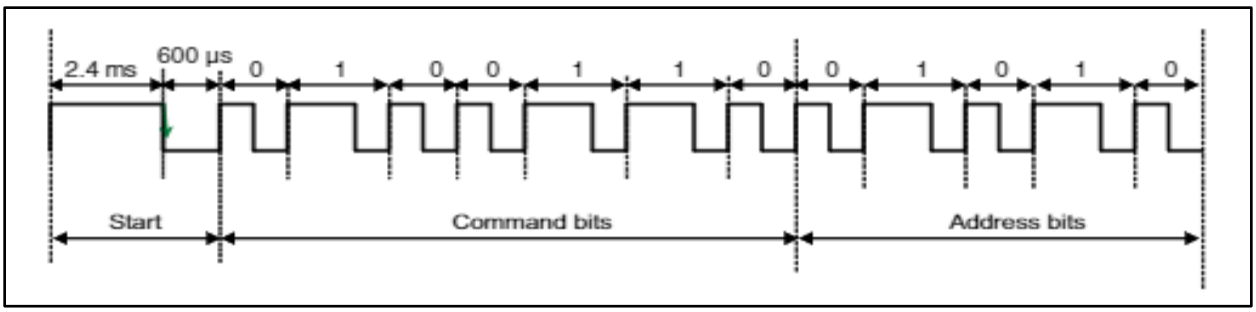

Figure 3a: 12-bit SIRC SONY protocol

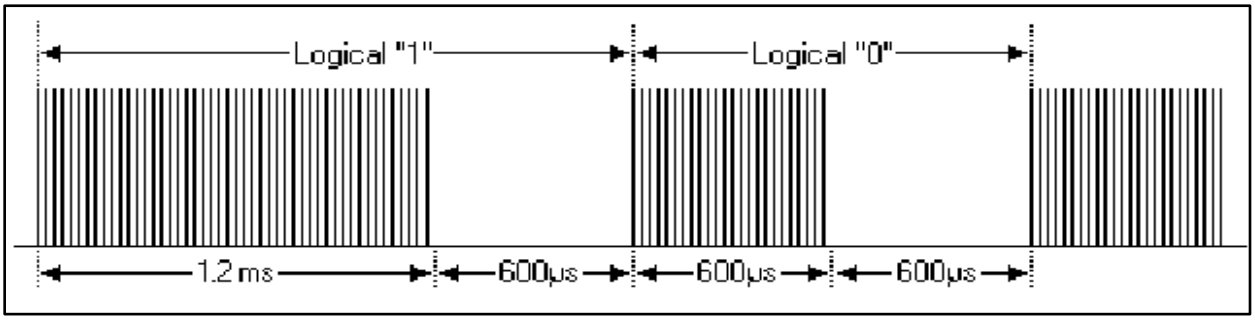

Figure 3b: IR transmitter protocol

IR Sensor

The sensor module, KY-022, demodulates the received IR signal transmitted by the IR transmitter **(Fig. 4)**. This allows the MCU to decode the demodulated frame from the sensor module to corresponding commands and address data bytes (HEX codes). In this way, the decoded data bytes are utilized to conduct control decisions for the H-bridge and load driver.

Figure 4: KY-022 IR sensor

Power Supply Module

A separate MB102 power supply module **(Fig. 5)** was used to power the H-bridge and motors rather than directly supplied by the Arduino itself to address the following issues:

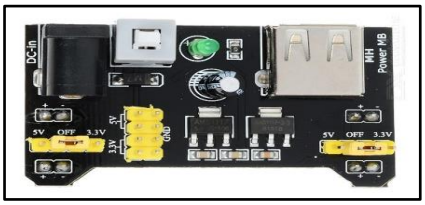

Figure 5: MB102 power module

1. Each pin on an Arduino can handle 40mA. However, the dual motor system current demand may exceed 40mA; hence, risk damaging the Arduino.
2. LDO voltage regulator may overheat and can get destroyed if sourced with high current.
3. Motors have a "blowback" voltage, a back-EMF usually addressed by adding a reverse-biased fast diode, sometimes in addition to a capacitor, across the motor's supply wires. Without such protection, there is high possibility of this voltage destroying the individual GPIO line and more likely, the entire MCU.

Dual H-Bridge

L293D dual H-bridge controls voltage polarity; hence, allowing the motors to drive bi-directionally, clockwise or counter-clockwise **(Fig. 6)**. As such, the dual motor system controlled by the L293D H-bridge allows the vehicle to move freely in a two-dimensional plane.

The interchangeable two combination of open and closed switches yields the two voltage polarities.

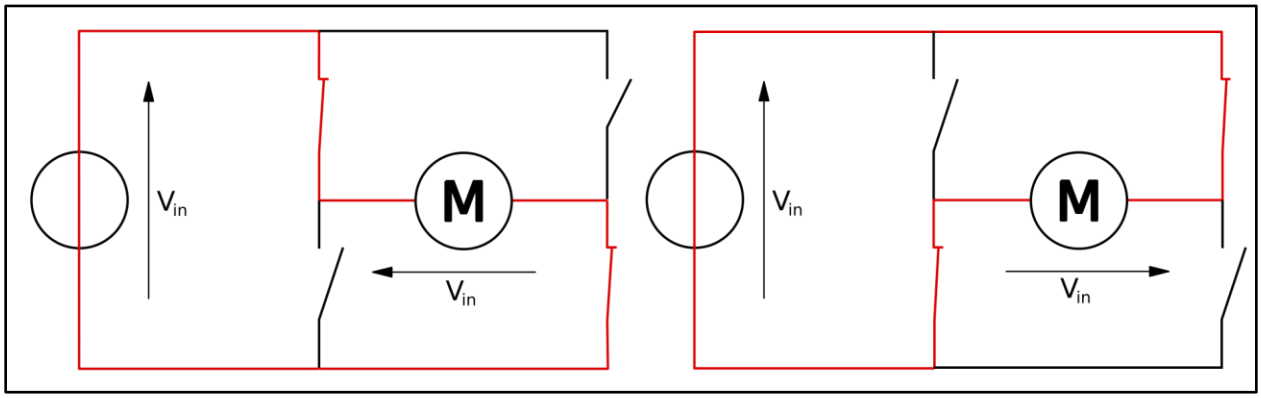

Figure 6: Two basic states (voltage polarity) of an H-bridge

Furthermore, enable pins on the L293D allows the usage of PWM (scale of 0 to 255). In terms of the practical application of this project, the enable pins introduce motor RPM variation; hence, the ability to vary vehicle speed.

Load Driver (Motors)

The dual motor system consists of two TFK-280SA-22125 9V DC motors fixed along the same axle **(Fig. 7)**. Given the no load speed of 7500 ± 10% RPM, the angular speed is adequate in its capability of moving the vehicle.

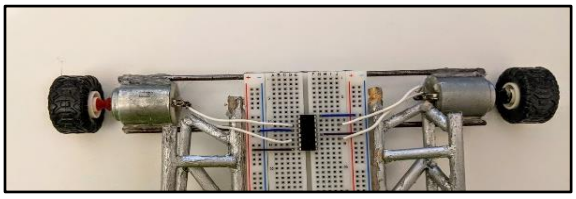

Figure 7: Dual motor system mounted onto vehicle

SOFTWARE DESCRIPTION OF SYSTEM

Below displays the flowchart for the software progression of data processing.

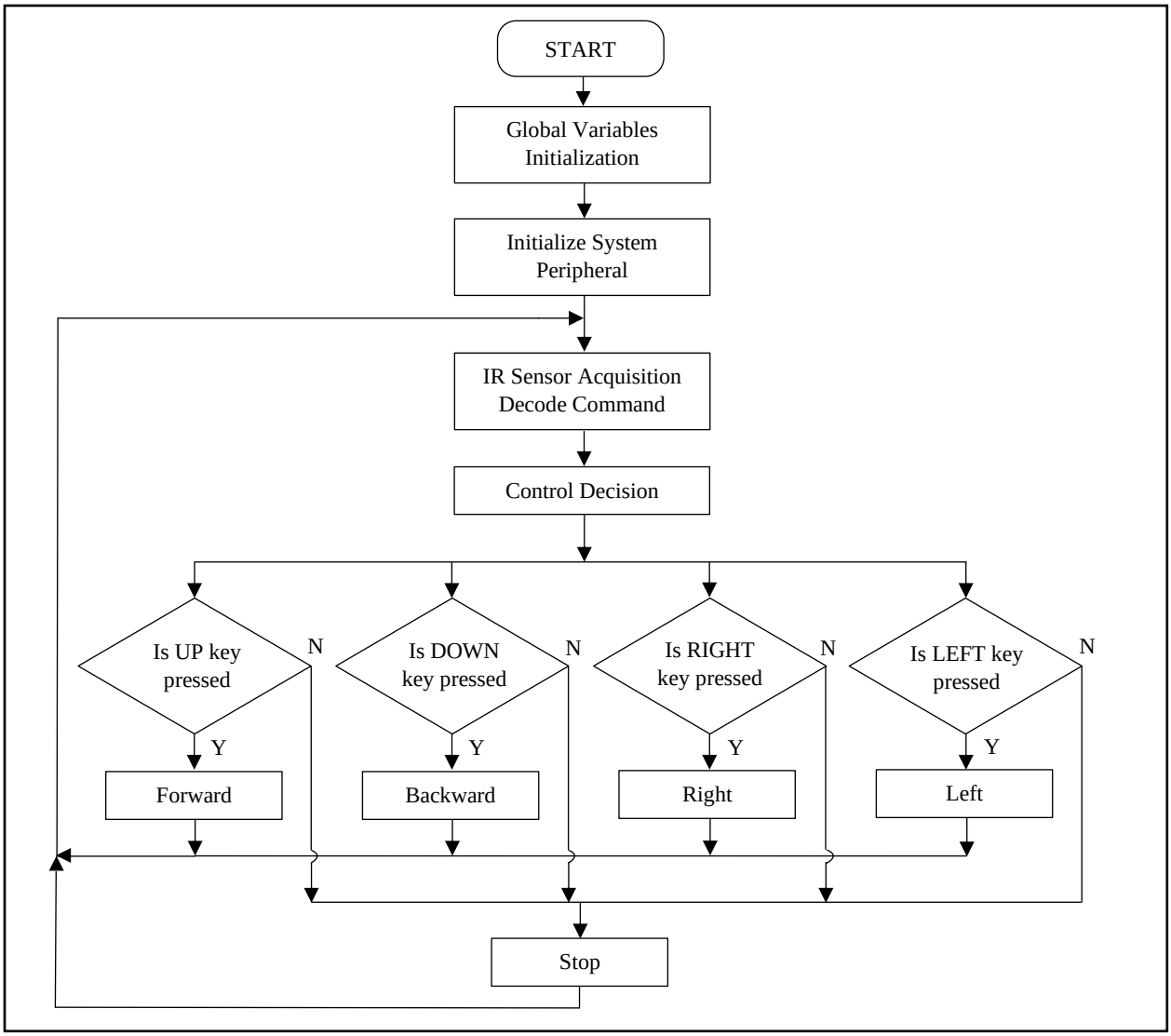

Figure 8: Infrared decoding and command structure flowchart

The full Arduino Sketch program (found on GitHub) is responsible for the following actions: to receive and compute transmitted IR information for motor control.

The example code below states that if the IR remote UP button was pressed, the following commands will transmit to the H-bridge and subsequent motors such that the car will move towards forward.

```
void loop() {
if (results.value == 0x10059) {   /* HEX code for UP button */
       mSpeed = twomSpeed = 255;       /* Set both motor speed (PWM) maximum */

       /* MOTOR 1 */
       analogWrite(speedPin,mSpeed);
       digitalWrite(dir1,LOW);   /* H-bridge switches for clockwise rotation */
       digitalWrite(dir2,HIGH);

       /* Same commands for MOTOR 2 */
       analogWrite(twospeedPin,twomSpeed);
       digitalWrite(twodir1,LOW);
       digitalWrite(twodir2,HIGH);
}
IR.resume(); /* reset receiver and prepare to receive the next HEX code */
}
```

FINISHED PRODUCT

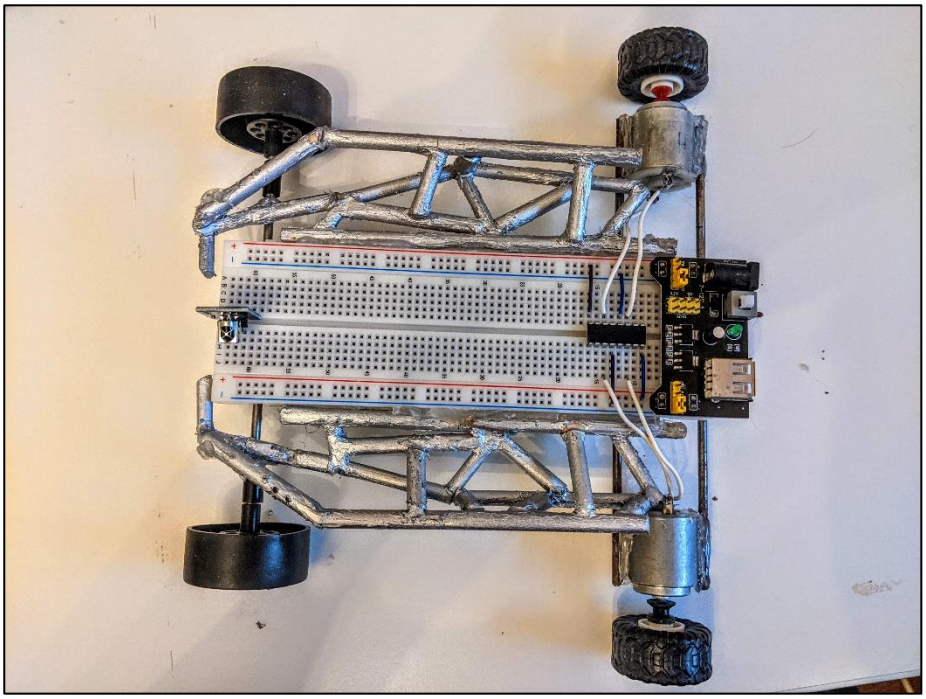

Figure 9a: (Top view) Vehicle chassis without Arduino

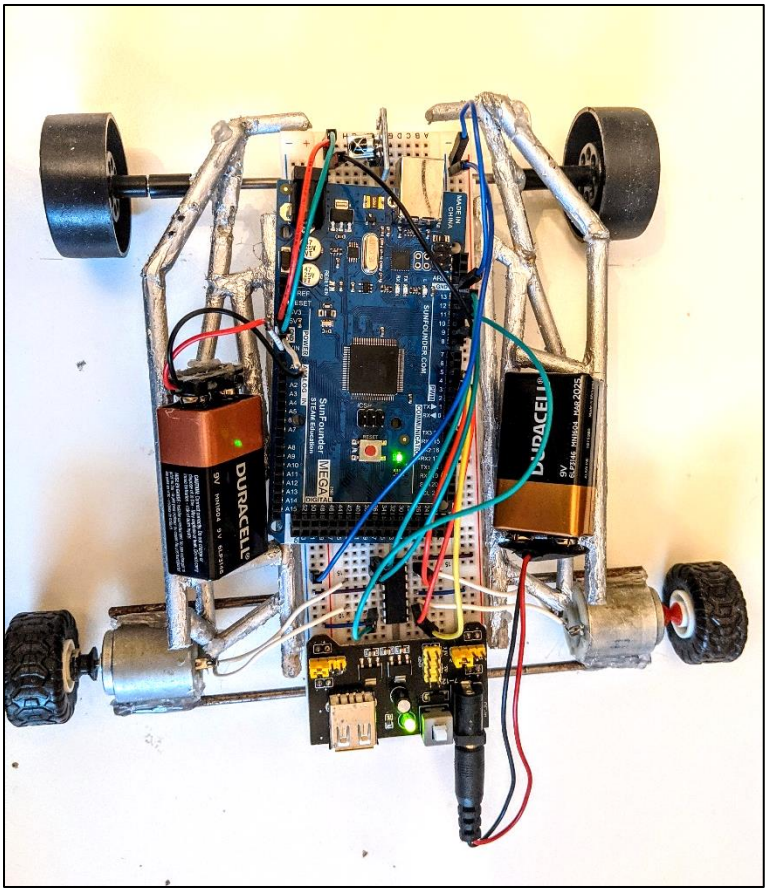

Figure 9b: (Top view) Vehicle chassis with Arduino

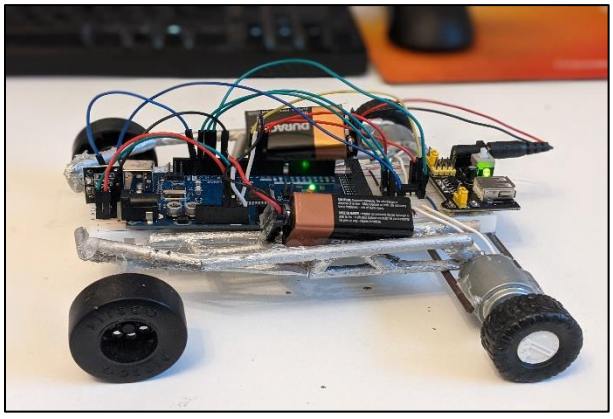

Figure 9c: (Back view)

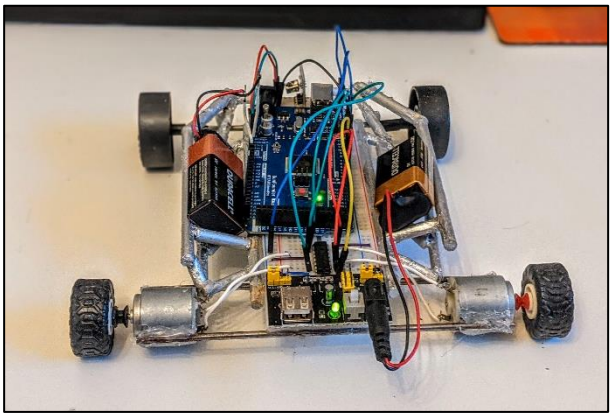

Figure 9d: (Side view)