1. Purpose

The creation of the project is to show my ability in allowing people to extract an understanding and innate ability learning the resources given in-depth.

1. In-depth architectural description

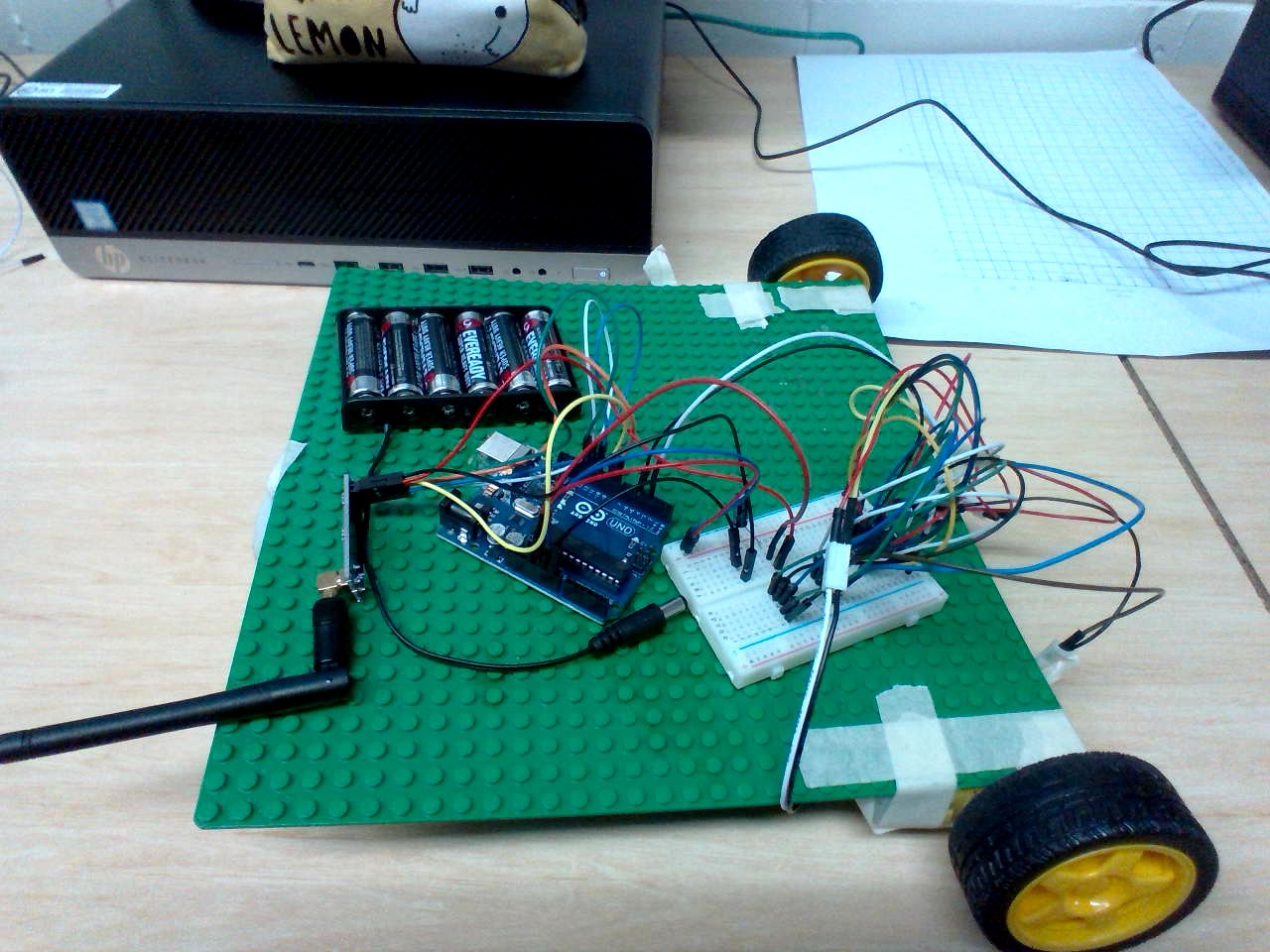
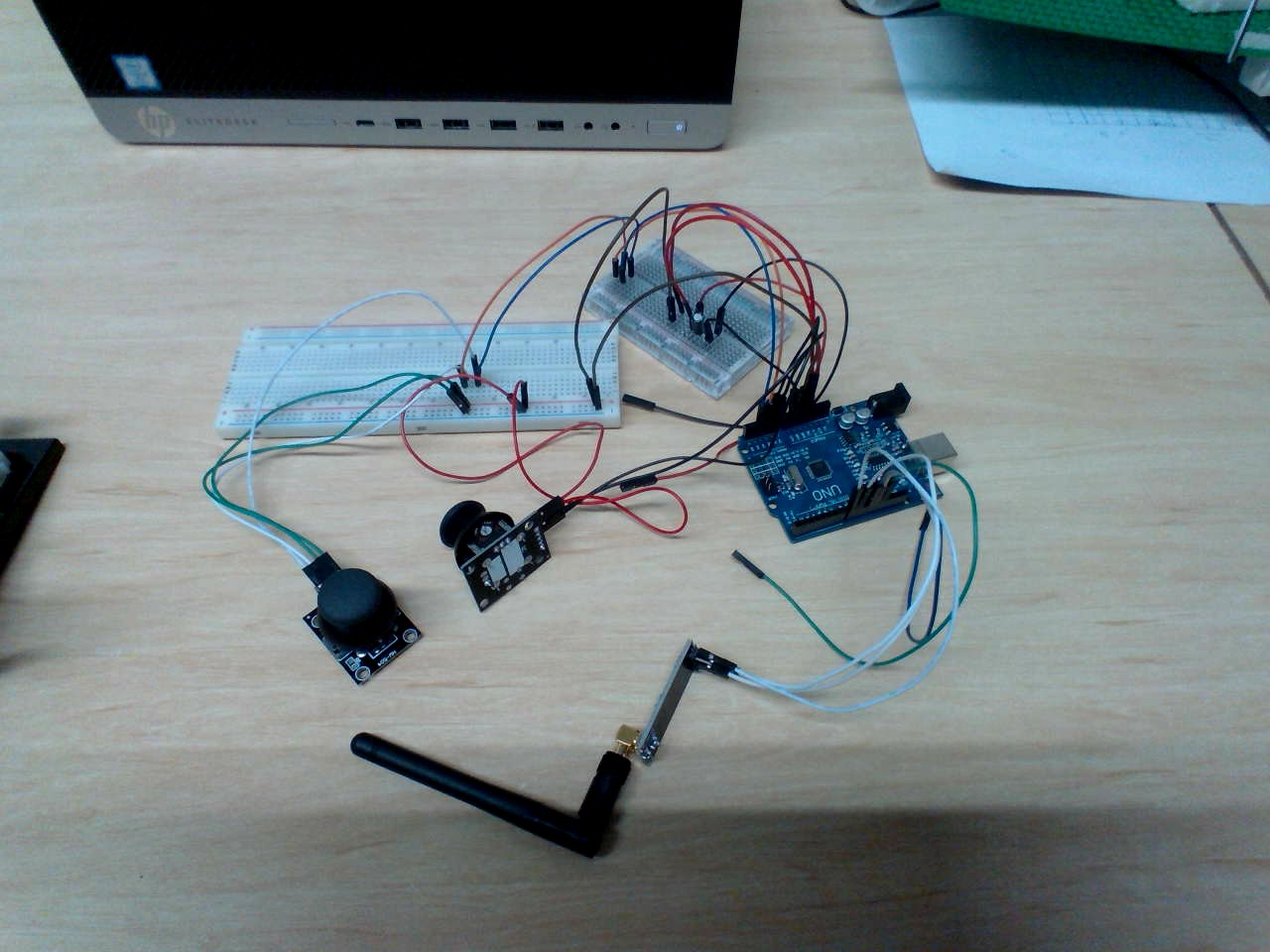
A basic lego build constructed to be capable to hold the Arduino Uno and a battery capacitor. Having the two of the motors a line parallel to each other on the outer rim of the build with the wheels attached. Held in place using rubber bands, holding everything together. Stabilized with additional wheels set on the front and back.

1. Detail design description
   1. Tools/library description

* DC Motor: An electrical machine that is designed to have a rotating point.
  + Utilised to make the vehicle drive around using wheels attached to the DC motor.
* Lego: Plastic building blocks able to craft various designs.
  + Its purpose is to handle creating a basic vehicle that can hold the necessary tools in place.
* Arduino Uno: A microcontroller allows written program to be uploaded, along being one of many different Arduino products.
  + Handles processing the program implemented into the device.
* nRF24L01+ Wireless Module: A data transmitter using frequency to operate wireless.
  + Operates to send information to one microcontroller to another microcontroller.
* H-bridge: An electric circuit capable of switching the direction of the voltage.
  + Handles in flowing voltage through and into the motors to be able to drive it.
* Joystick: An input device of a stuck pivot on the base reporting the direction the stick is currently set.
  + To operate in inputting information from the user base on the direction, mainly in three different speed. Fast, medium and slow.
* Volt capacitor: Stores electric charge when connected to a voltage source.
  + Assist in increasing the amount of voltage flowing through the circuits.
* Library:
  + <SPI.h> : Handles with the SPI communication
  + <nRF24L01.h> : Controls the module
  + <RF24.h> : Control the module
  1. Programming
* RF24 radio(9, 8)
  + Create an object to be able to send signal
* const byte address[6] = "00001"
  + Assign an address for the modules to communicate.
* radio.begin()
  + Initialize the modules to begin to communicate.
* radio.openWritingPipe(address)
  + Sets the modules address to the transmitter.
* radio.stopListening()
  + Initialize the module as a transmitter.
* radio.write(&text, sizeof(text))
  + Send information to the receiver.
* radio.openReadingPipe(0, address)
  + Sets the modules to address to the receiver.
* radio.startListening()
  + Initialize the module as a receiver and receive data.
* radio.available()
  + Checks if the data has arrived, returning TRUE if any data is available in the buffer.
* radio.read(&text, sizeof(text))
  + Reads the data and store it (previously created array).
* analogWrite(3, values[0])
  + Write the analog value (PMW).

Prototype:

Final build:

