Arduino Project Report

Aumshree P. Shah - Group 19
20231059[a]

(Dated: 18 November 2025)

Synopsis

This report contains the motivation, idea, workings, troubleshooting, and final model of the Arduino project of Group-19.

CONTENTS

[a] aumshree.pinkalbenshah@students.iiserpune.ac.in

I. MOTIVATION AND IDEA

A. Motivation

Although given that the final product should be related to safety and address a safety issue, which my following concept does, I also tried to make something out of the ordinary, which motivates us to learn the workings of Arduino.

B. Concept

My final concept was to make a *Remote Controlled car, equipped with sensors to detect threats and monitor/neutralize them with a coil gun and cameras*. Though I went through many different concepts, like: Water gun with precise aim at fire, Car with automatic fire detection and neutralization, Dog in hostel detection and monitoring air quality levels, I decided to finalize on my former idea due to its complexity, budget constraints, and other reasons.

II. WORKINGS AND TROUBLESHOOTING

A. Finding Resources

The idea of *RC car with Arduino* and *Coilgun controlled by Arduino* separately has been worked on by many people many times. The diagrams and schematics of them, as well as the code, are available on the web and can be found easily, though there isn't much I could find that combines both of these ideas. Hence, although I had many of the resources readily available for my help, learning how it is done and making the changes was indeed very labour-intensive. I have made my own schematics for both the Transceiver and Receiver in the following section.

B. Schematics

First, we shall look at the transmitter schematic as shown in FIG-1.

Here we see the connection from the NRF module to the specific pins is done in such a way that allows the most compatibility. For example, in Arduino Nano, pins D13, D12, and D11 act as SCK, MISO, and MOSI, respectively. Other connections are arbitrary. The Joystick X and Y pins are connected to analog pins. The power is transmitted across the components in the following way:

Battery (6V) $\implies$ Aurduino Nano (5V) $\implies$ NRF Module regulator (3.3V) $\implies$ NRF Module

Now we will look at the receiver schematics shown in FIG-2. Here, although the NRF is connected in the same way to the Arduino, a few connections are made because of pin support; for example, the ENA pin is connected to D9 because only a few D pins support PWM, which the L298 driver requires. The power is transmitted across the components in the following way:

Battery (11.1V) $\implies$ L298N Motor Driver(11.1) $\implies$ 4 x Motors
L298N Motor Driver(5V) $\implies$ Aurduino Nano (5V)) $\implies$ NRF Module w/Regulator

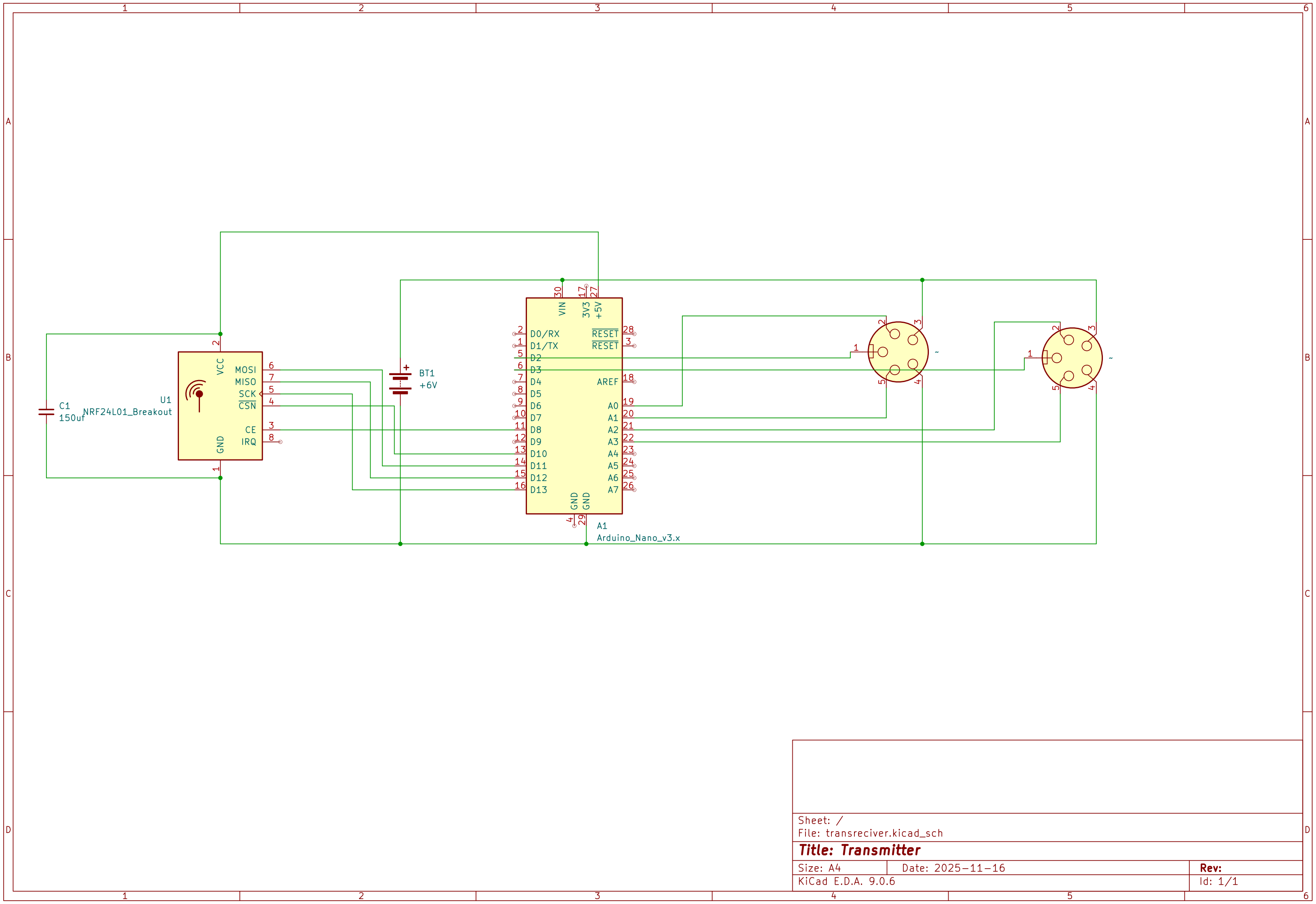


FIG. 1. Tramsitter Schematics

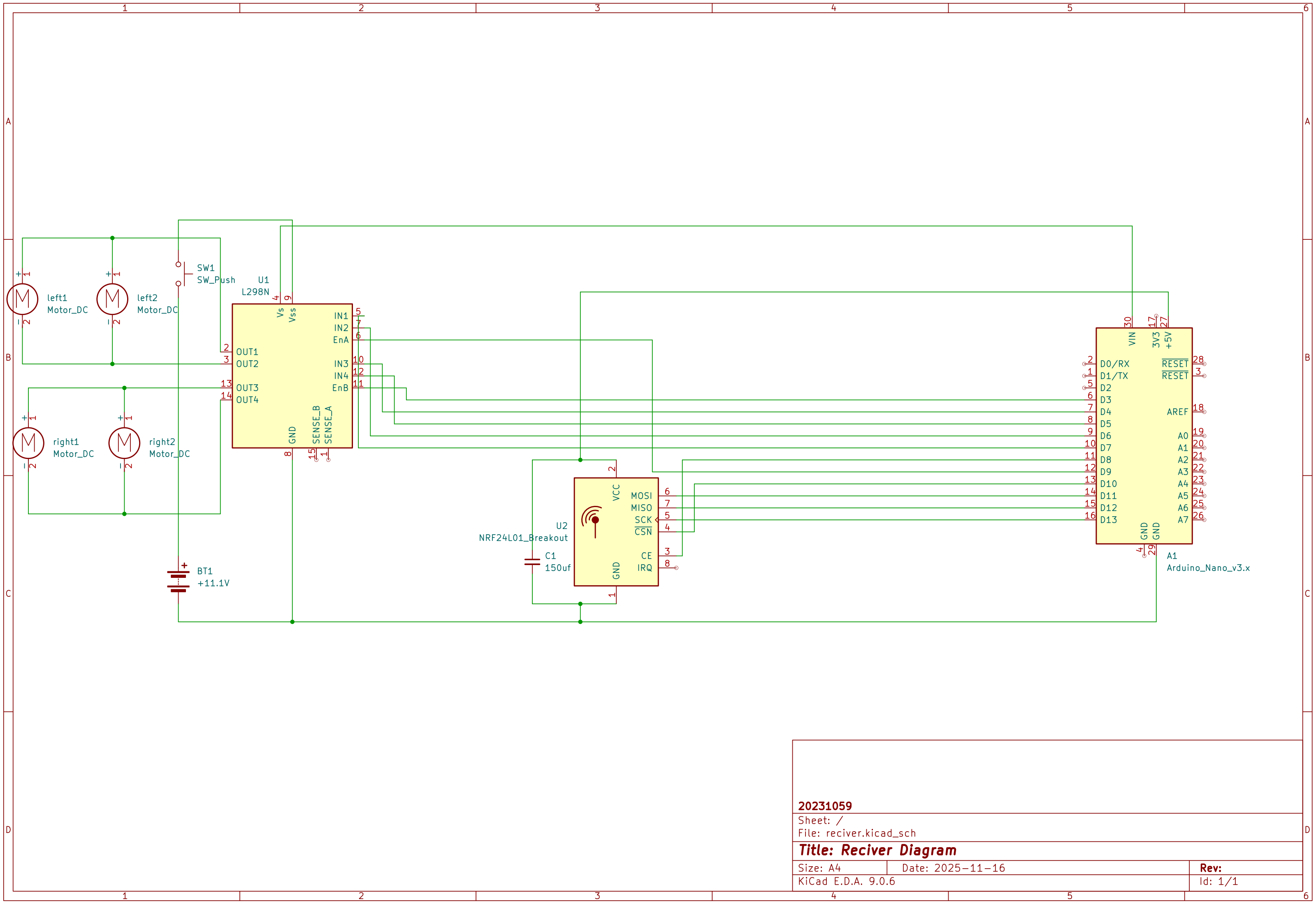


FIG. 2. Reciver Schematics

C. How I made it

Since my main goal was to make a RC-controlled car, my first step was to make two different Arduinos communicate. For the communication, I decided to use **NRF24L01+PA+LNA SMA Antenna Wireless Transceiver communication module** as the price difference between the alternatives v/s this was not much, whereas the range was significantly higher. This module operates at 3.3V, to regulate it, I used **NRF24L01 regulator**, which regulates the logic voltage. I used this regulator after trial and error to regulate the voltage, which will be discussed in a later section.

For the motors, I use **4 x 300RPM PO Motors** since 300 RPM seemed in the proper range for a car with wheels of diameter $\approx$ 7cm. These motors run on 12V.

For the battery, in the transmitter I used **4 x 1.5V alkaline battery** giving 6V, which is given to Arduino V_{in} pin. The Arduino has an internal regulator that regulates it to 5V. The Arduino provides power to the NRF regulator.

In the receiver, I used **3 x 3.7V LiOH battery**, which is rechargeable. The reason for using them will be discussed in the next section. This produces a voltage of $\approx$ 11V, which is given to **L298N Motor Driver**, which runs the motors. The L298N has an internal regulator that regulates the voltage to 5V, which powers the Arduino.

A phone holder is added to the back, which can function as many things.

In the receiver, the NRF is also powered by Arduino through a regulator. Almost all of the connections are made using **Jumper wires of different types** and **Breadboard**. Some do use **High Gauge copper wires** to supply high current to motors.

D. Issues I faced - and how I solved them

1. Battery unable to provide required power

Initially, I used **9V alkaline battery**. A single 9V battery was unable to provide power to the motors, so I used 4 of them in parallel. Although this made the car functional, it had many disadvantages. For example, it was unrechargeable, even 4 of them in series were sometimes not able to provide enough current, and the motors were rated for 12V, so 9V batteries weren't running the motors at maximum capacity.

2. Search for a proper chassis for a car

I needed a strong base for my car. A cardboard couldn't hold its weight. A proper chassis was really hard to find, and after some trial and error, I finally settled on a small black metal box, which I found in a mechanic's box.

3. Regulation of NRF module

This was a really big issue when I first started testing, since to make anything out of this project, I first had to communicate between two Arduinos. The NRF module operates at 3.3V, and the Arduino logic is 5V. I had to use some kind of way to regulate it. First, I used resistor dividers, but it turned out that they caused a high level of signal disturbance. Then I used voltage regulators, these kind of worked, but there was always some kind of issue which made the modules unable to communicate. Also, since the NRF module needs a large supply of current peaks, my initial tests were without parallel capacitors, so no matter how I regulated the voltage, the module was unable to communicate. This was finally resolved by using a dedicated NRF voltage regulator, which was compact, cheap, and did everything we needed.

4. Fused and non-working components

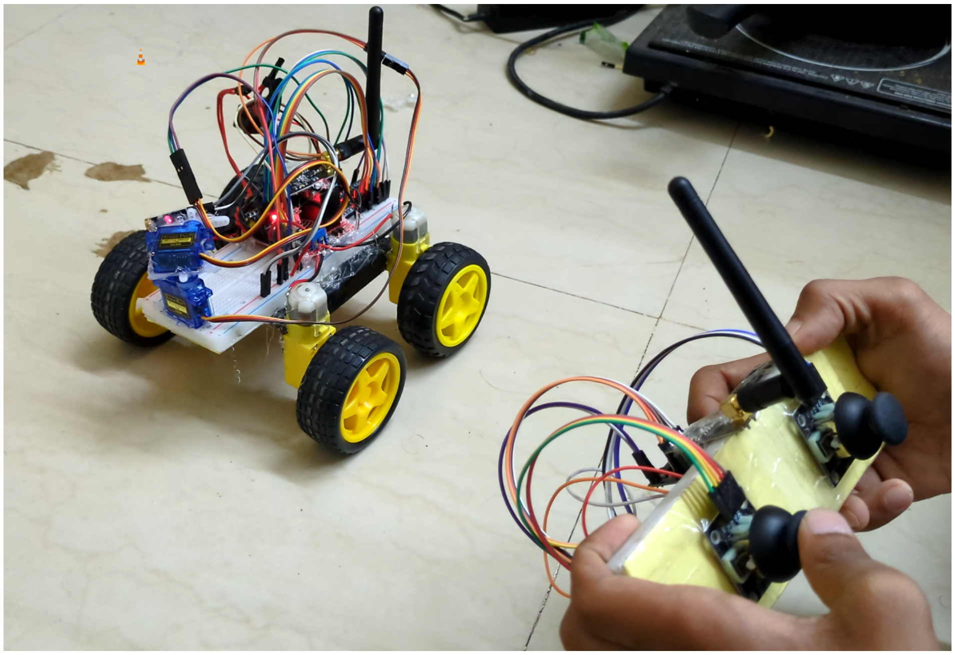
Many of the components got fused while I were working on them, includes but not limited to: Arduino nano, an NRF module, etc. To prevent this from happening on other components, I found the following helpful: Not soldering components directly to the other components, using voltage limiters, reading the datasheet (obvious, but I initially ignored this), and by not shorting the components (obvious, but that's how my nano got fused).

5. Weight, Speed Structure and Battery life

For weight, Speed, and battery life, all these issues were solved by LiOH batteries. To fix the motor ot the chassis, I used Feviquick, which worked really well and was able to carry a good amount of weight on the car. Other components were fixed by hot glue, and the final structure was made after many trials and errors.

III. RESULT AND FINAL PRODUCT

The car and the controller looked like this after completion:



and video's of this running is uploaded on youtube as refrenced in the next refrence section.

REFERENCES

Lauging-Cat-Meme, Aurduino Group-19 Code, https://github.com/LAUGHINGCATMEME/PH3144-Aurduino-Project (2025), [Online; accessed 16-Nov-2025].

Dejan, How to make Aurduino Transmitter and reciver, https://howtomechatronics.com/projects/diy-arduino-rc-transmitter/ (2020), [Online; accessed 16-Nov-2025].

Laughing-Cat-Meme, Demo of RC Car, https://youtu.be/8HlMw1vrnL0 (2025a), [Online; accessed 18-Nov-2025].

Laughing-Cat-Meme, Demo of RC Car - Car POV, youtu.be/yLEg9JWuXcM (2025b), [Online; accessed 18-Nov-2025].