### 3.2.4. Motion Subsystem

#### 3.2.4.1. Solution Approach

Motion subsystem consists of motors for driving the robot on the play field, wheels, other assembly parts such as encoders and gearboxes attached to these motors and also driver IC which converts inputs from the Main Processor Subsystem into meaningful inputs for the hardware part of the motion subsystem. Motion subsystem also sends feedback to Main Processor Subsystem.

We chose to implement our motion subsystem via a DC motor as in Figure 3.2.4.1.1. , with the following specifications in Table 3.2.4.1.1 :

Table 3.2.4.1.1 : DC Motor Specifications

|  |  |
| --- | --- |
| Dimensions | 37D x 102L mm |
| Weight | 310 g |
| Nominal Operating Voltage | 12V |
| Free run speed at 12V | 120 rpm |
| Free run current | 400 mA |
| Stall torque | 25 kg.cm |
| Gearbox ratio | 60:1 |

In order to drive these two DC motors, we are using L298N which is our main solution as a motor driver as mentioned in the Conceptual Design Report. The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive two DC motors that have voltages between 5 and 35V, with a peak current up to 2A. We will drive two DC motors using this IC with respect to the pin connections as seen in Figure 3.2.4.1.2.



Figure 3.2.4.1.1 : RW-ML-1333 DC Motor

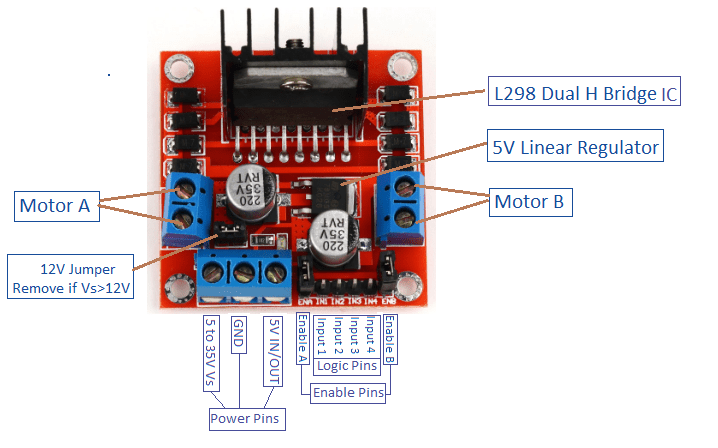


Figure 3.2.4.1.2 : Pin Connections of L298N Driver IC

L298N motor driver uses PWM as input to adjust the speed of the motors by changing the duty cycle. 100 % duty-cycle means full throttle and 0 % duty-cycle means zero-velocity. We are using an Arduino Mega on the robot side which creates these PWM signals with respect to the commands sent from the controller side. We are not implementing speed control directly from the controller side. However, we are implementing smooth start and brake strategies in order to prevent high peak current at the start of the motion and also to protect the mechanics of the DC motors. We can also set the turning direction of the motors by simply setting two pins for each motor either HIGH-LOW or LOW-HIGH. These four pins can be seen in Figure 3.2.4.1.2. as the Logic Pins. This direction control is used for differential drive of the robot which was also mentioned in the Conceptual Design process.

After deciding on our DC motor, we needed to choose an encoder to give feedback to the Main Processor Subsystem. This feedback is vital since even though both of the DC motors are identical there may be a calibration difference between them. Our robot may deviate from its straight path. Therefore, we need to calibrate these two motors by implementing optimization with respect to the encoder counting differences.

Incremental encoders are useful for our purpose. The encoder type we are using, has Hall-effect sensor in it and counts up to 44 for one single turn but it should be multiplied with the gear ratio. Hall effect encoders use magnetic phased arrays that contain hall sensor elements arranged in a pattern to match a magnetic wheel. A signal is produced as the sensor passes over the magnetic field which is then interpolated to the desired resolution. The representation is as in Figure 3.2.4.1.3.

 For wheels we are using two standard wheels and also three ball wheels to stabilize the robot chassis. Wheels-chassis integration is covered in the technical drawings part.

Figure 3.2.4.1.3 : Magnetic Encoder Representation with Hall-effect Sensor

#### 3.2.4.2. Block Diagram and Flowchart

Block diagram of the motion subsystem can be seen in Figure 3.2.4.2.1.



Figure 3.2.4.2.1 : Block Diagram of the Motion Subsystem

Flowchart of the motion subsystem can be seen in Figure 3.2.4.2.2.

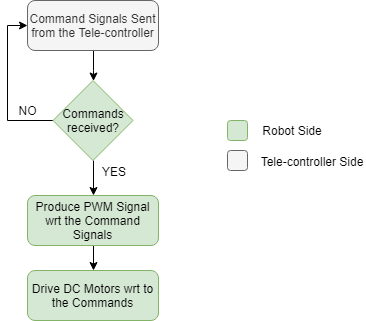


Figure 3.2.4.2.2: Flowchart of the Motion Subsystem

## 3.3. System Level Block Diagram & Flowchart

Block diagram of the overall system can be divided into two parts. First one is the Controller Side and the second one is Robot Side. Block diagrams of these two parts can be seen in Figure 3.3.1 and Figure 3.3.2.

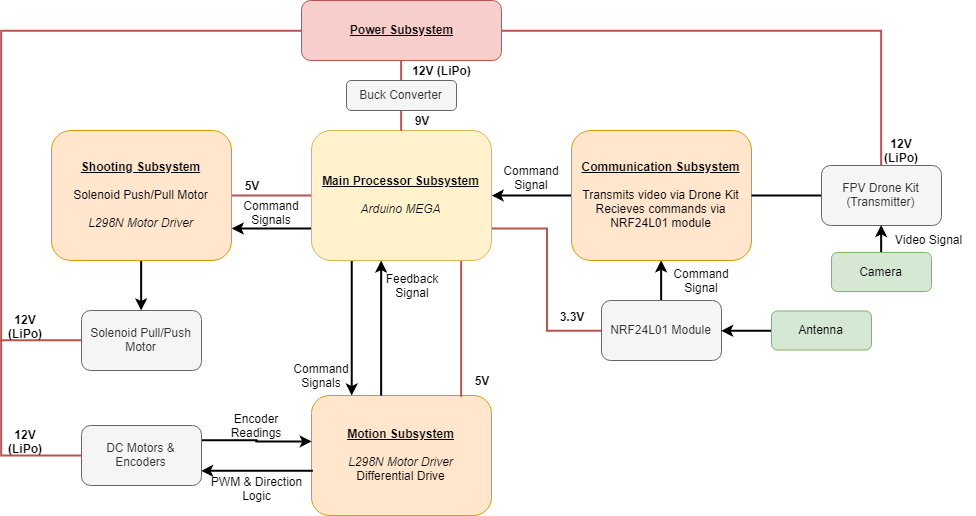


Figure 3.3.1: Block Diagram of the Overall System (Robot Side)

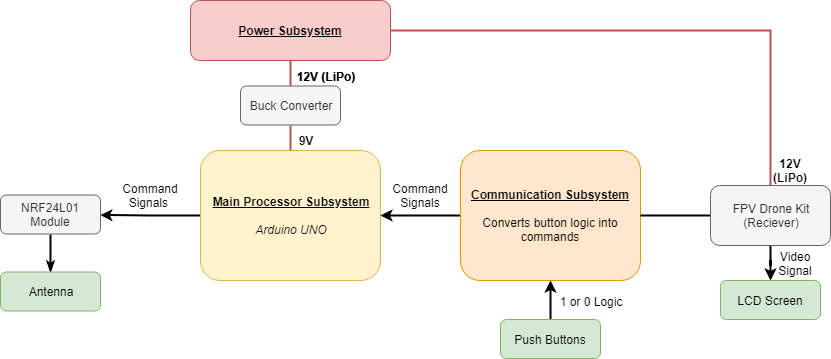


Figure 3.3.2: Block Diagram of the Overall System (Controller Side)

Flowchart of the overall system can be seen in Figure 3.3.3.

## 

Figure 3.3.3 : Flowchart of the Overall System

## 7.4. Motion Subsystem

We executed following test procedures so far:

First of all we drove our DC motors for no-load condition to observe the amount of current it draws. A single DC motor draws 400 mA as mentioned in its datasheet. However, once we integrated the motors on our chassis and observed the amount of current drawn from both of the motors, we saw that it can exceed 2 A. This is because DC motors draw huge amount of current at the start of the motion. In order to prevent these high current peaks we are implementing soft start strategies. This is implemented by software. We will increase the speed incrementally.

We did further tests on the speed of the vehicle with 100% throttle. Considering the length of the half playfield and the time our robot to measure the approximate speed of the robot. The velocity of our robot is approximately 0.45 m/s. Since this velocity is highly enough to satisfy the 20 sec restriction we can consider lowering the speed to increase efficiency of the battey usage.

In addition to these test results we are planning to do tests on speed optimization mentioned above. These adjustments will make our robot more stable and decrease the mechanic strain on the DC motors. We will implement these tests in the playfield.