Task 3 - Power Management System

Okay first things first, what do we have to build?
It's basically just a robotic arm that can move around.

What do we have?

Let's see what we're working with.

Components	**Quantity**
Brushed DC Geared Motors	4
BLDC Motors	2
Servo Motors	3
NEMA Stepper Motor	1
Raspberry Pi 5	1
RPI Camera Module 3	1
Ldrobot D500 LiDAR Kit	1
ESP32	1

Our objective

- Schematic - Basically what goes where
- Appropriate Battery Selection
- Safety Features (including a kill switch of course)
- Power Distribution Analysis

Schematic - What goes where

Motor Specifications

We first focusing on the Torque specifications of all the motors to figure what motors to use in locomotion vs the robotic arm itself.

Components	**Quantity**	**Rated Torque**
Brushed DC Motors	4	11 kg-cm
BLDC Motors	2	0.573 kg-cm
Servo Motors	3	28.8 to 35 kg-cm
NEMA Stepper Motor (2 Phase)	1	2.9 kg-cm

Calculating that rated torque for the ECO II Series 2207 BLDC Motor was not that straightforward.

The torque of a BLDC motor can be calculated using the formula,

$$\text{Torque} = K_t \times \text{Current}$$

where K_t is the torque constant, which is related to the kV rating.

First, we convert the kV rating from RPM/Volt to the SI unit rad/s/Volt,

$$K_v(\text{SI}) = 1700 \times \frac{2\pi}{60} \text{ rad/s/V}$$
$$K_v(\text{SI}) = 177.89 \text{ rad/s/V}$$

The torque constant K_t is the reciprocal of K_v in SI units,

$$K_t = \frac{1}{K_v(\text{SI})} = \frac{1}{177.89} = 0.00562 \text{ Nm/A}$$
$$\text{Rated Torque} = K_t \times \text{Rated Current}$$
$$\text{Rated Torque} = 0.00562 \text{ Nm/A} \times 10 \text{ A}$$
$$\text{Rated Torque} = 0.0562 \text{ Nm} = 0.0562 \text{ kg-cm}$$

Motor Selection

Locomotion: An obvious choice for the locomotion would be the 4 Brushed DC Geared Motors which provide enough torque for the load movement and is pretty

suitable in a 4-wheel drive configuration.
If we were to choose some other motor, say the servo motor because of its much higher torque, then we would have to make a tricycle drive which is certainly not suitable for high loads and stability.

Base joint actuator: The base motor has to overcome the frictional force and in addition, has to handle the angular acceleration of the whole robotic arm.

$$\tau_{\text{base}} = \tau_{\text{friction}} + I\alpha \tag{1}$$

where I is the moment of inertia and α is the angular acceleration at that instant about the axis.
Assuming we'll keep α minimal and proper lubrication of the motors (avoiding τ_{friction}), the torque required is quite minimal. On the other hand the precision required is massive which can only be provided by the NEMA Stepper motor. Lesser torque but the precision, boosted by software microstepping, is more suited to the base motor.

Shoulder joint actuator: Here we define some terminologies. Let the Load mass be m, link lengths be L_1, L_2, L_3, link masses be M_1, M_2, M_3, end effector mass be M_0 (including the motors for the end effector). This will be clearer to the reader by the figure below. The actuators themselves have masses A_1, A_2, A_3 and A_0 as the end effector.
Coming back to the shoulder joint actuator A_1, the maximum torque required is quite huge. It can be calculated by taking into account the maximum extension case.

$$\tau_{\text{shoulder}} = \frac{M_1 g L_1}{2} + A_2 g L_1 + M_2 g \left(L_1 + \frac{L_2}{2} \right) + A_3 g \left(L_1 + L_2 \right) + M_3 g \left(L_1 + L_2 + \frac{L_3}{2} \right) + A_0 g \left(L_1 + L_2 + L_3 \right)$$

This is quite a huge quantity (the most torque in this whole project in fact) and can only be supported by a servo motor. The choice we make for the shoulder joint is the Pro-Range OT5330M Servo motor.

Elbow joint actuator: For the elbow joint actuator A_2, the maximum torque in maximum extension case is given by,

$$\tau_{\text{elbow}} = M_2 g \left(L_1 + \frac{L_2}{2} \right) + A_3 g \left(L_1 + L_2 \right) + M_3 g \left(L_1 + L_2 + \frac{L_3}{2} \right) + A_0 g \left(L_1 + L_2 + L_3 \right)$$

This is lesser than the torque required in case of the shoulder joint actuator but it is still a huge quantity and can also only be supported by a servo motor. The choice we make for the elbow joint is also the Pro-Range OT5330M Servo motor.

Wrist joint actuator: For the wrist joint actuator A_3, the maximum torque in maximum extension case is given by,

$$\tau_{\text{wrist}} = M_3 g \left(L_1 + L_2 + \frac{L_3}{2} \right) + A_0 g \left(L_1 + L_2 + L_3 \right)$$

This is lesser than the torque required in case of the shoulder and elbow joint actuator but it is still a higher quantity and can also only be supported by a servo motor. The choice we make for the wrist joint is also the Pro-Range OT5330M Servo motor.

End effector actuators: These actuators dont need *that* much torque as the other joints but have to be more versatile. We may need to use it as a gripper or say as a drill-ish machine which require precision/speed at once. Considering this and also the fact that the only motors which are left are the BLDC motors, we're just gonna use the ECO II Series 2207 BLDC Motors.

Motor Controllers and Drivers

First off, our motor controller choice is pretty obvious. It's the ESP32 microcontroller. Because of its computation restrictions we can't really use it as the main driver with the object detection with lidars and camera on it, but motor driving signals are easy to compute.
On top of this, motor drivers for all the motors except the servos (which don't really require a driver) are a necessity. The schematic explaining what im talking about is given below. The COMMS go to Raspberry Pi.

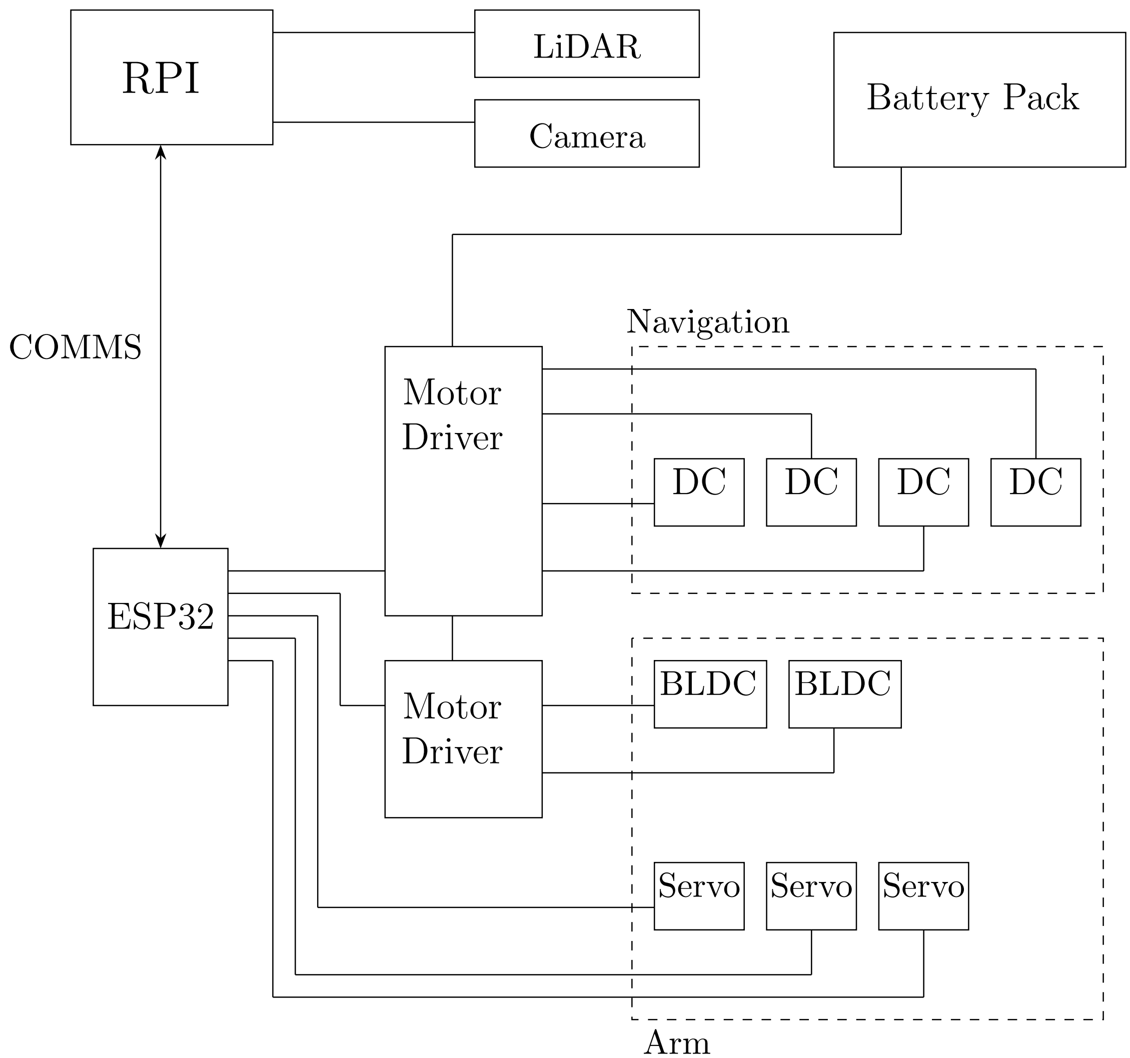
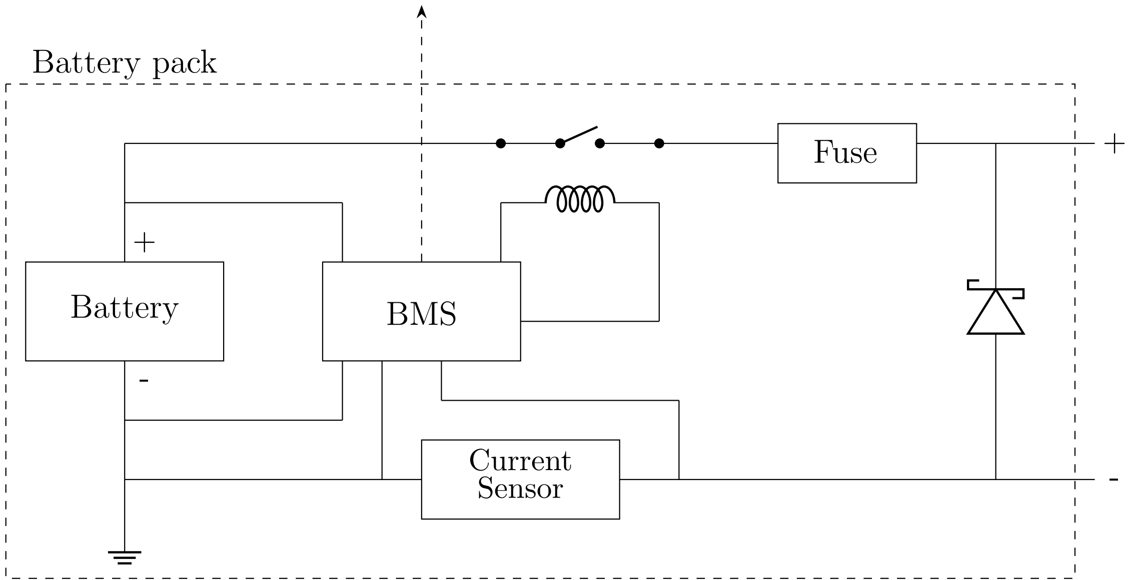


Figure 1: Overview Schematic

Battery Pack

The battery pack consists not merely the battery but the safety and protection systems in place to avoid, well, catastrophes. In place is a **BMS** (Battery management system), **Fuse** to avoid short circuits, **Fool's diode** to prevent reverse current, and a **Master Kill Switch** for turning off the whole bot. The battery pack schematic is given below.



The Fool's diode will help in case of reverse currents as all the reverse current will flow through the diode instead of the actual load. We are gonna use a Schottky diode for this purpose due to its high reverse bias breakdown voltage (about 50 to 60 V which is more than enough).
What battery are we gonna choose? Well we haven't calculated the power specifications yet (which will we will do somewhere down the line), but one thing's sure. It's going to be around 12 V nominal voltage battery (or we could stoop down do 11.1 V if we like 3S LiPo Batteries).

Powering the RPI and ESP32 - Voltage Regulators

Since we're using a 12 V battery, it will blow up our Raspberry Pi and ESP32 instantly. We obviously don't want that, so we use voltage regulators which in this case, we will be using a **Buck Converter** to bring down the voltage to 5V for the RPI, ESP32, LiDAR and Camera Modules. In addition to that, we would also need a 7.4 V power line for the Servo motors to optimally run them for which will be using a different buck converter.

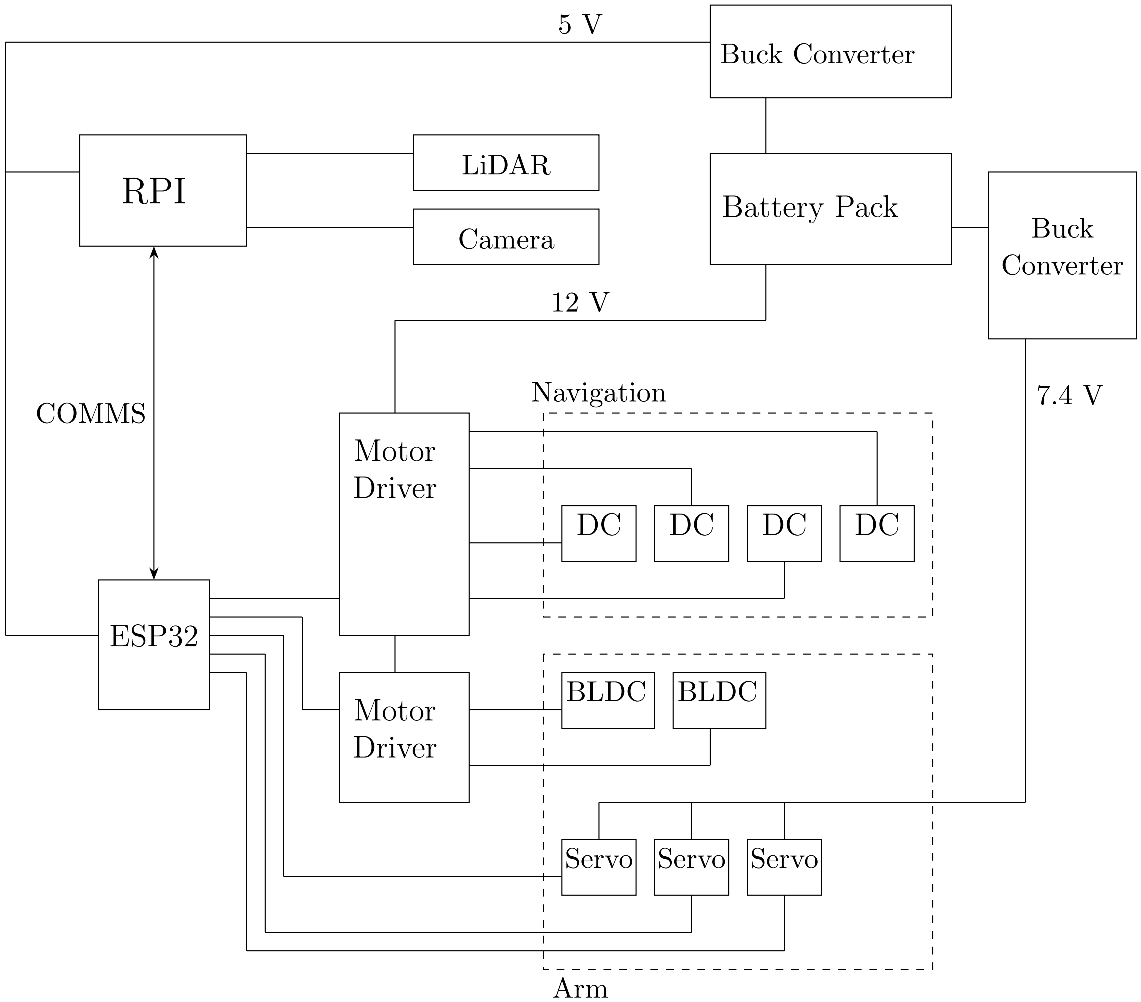


Figure 2: Final Schematic

Power Distribution and Component Selection

lala

Components	**Quantity**	**Rated Voltage**	**Stall Current**
Brushed DC Motors	4	12 V	15 A
BLDC Motors	2	11.1 to 22.2 V	36 A
Servo Motors	3	4 to 8.4 V	3.8 A
NEMA Stepper Motor (2 Phase)	1	12 V	0.5 A × 2 = 1 A