DOC NO:	ECE495-2025G505	**ISSUE DATE:**	05-12-2025
FILE NAME:	g5_assign5_detailed_design.pdf	**VERSION NO:**	0.1
FILING PATH:	https://github.com/AneekMubarak/MobileToolAssistant/blob/main/docs/		
TITLE:	DETAILED DESIGN		
PROJECT:	MOBILE TOOL ASSISTANT		
KEYWORDS:	Detailed Design		

APPROVED BY

Daniel Teng Date
Customer

PREPARED BY:

Aditya Ramakrishnan Date
Design Engineer

Aneek Mohamed Ruksan Date
Design Engineer

Chidubem Emeka-Nwuba Date
Design Engineer

Ethan D'Almeida Date
Design Engineer

REVIEWED BY:

Chandler Janzen Date
Supervisor

REVISION HISTORY

Version No.	Date	Originator	Summary of Changes
1	05-12-2025	A.M.R, A.D, C.E.N, E.D	Document created

Table of Contents

Table of Figures

(A – Aditya Ramakrishnan, B – Aneek Mohamed Ruksan, C – Chidubem Emeka-Nwuba, D – Ethan Peter D'Almeida)

Processor (A: 90% C: 10%)

Purpose:

The Processor Module reads all sensor inputs, determines the robot's motion commands, and sends control signals to the motor drivers. It handles UWB ranging, LiDAR scan processing, battery monitoring, state decisions, and safety shutdown actions at the robot's end.

Inputs:

- UWB Positional Data & Start/Hold Commands
- LiDAR Scan Data
- Battery Voltage/Current

Outputs:

- 4× PWM Motor Signals
- UWB Configuration Commands
- LiDAR Configuration Commands

Parts Required:

- Teensy 4.1 Microcontroller
- Development environment

Motor Control (D: 100%)

Purpose:

The motion control subsystem is responsible for converting the user's position from the User Tracking subsystem into motor drive signals.

Inputs:

- User Position (x, y) from User Tracking subsystem.
- System State (from obstacle detector and from UI)

Parts Required:

- 4x DC Gear Brush Motor (HP25SG-370H-1220-46.8, Manufacturer: 31ZY-2480)
- 4x BTS7960 Driver

Algorithm:

1. Compute distance error using the actual distance from the UWB module and target distance (1m).
2. Compute forward speed using

$$speed = K\_sp * distance\_error$$

3. Compute the turning amount using

$$turn\ amount = K\_turn * angle\_error$$

4. Each side is grouped, so

$$left\ side = speed - turn\ amount$$
$$right\ side = speed + turn\ amount$$

5. Compute PID speed for each motor group
6. Convert PID to PWM and direction signals

Obstacle Detection (C: 90%, A: 10%)

Purpose:

The purpose of the Obstacle Detection subsystem is to detect obstacles in the robot's path from a minimum distance of 1m. This information is then sent to the microcontroller to process and make decisions on whether to keep moving, slow down, or stop.

Outputs:

The LiDAR sensor sends all scan data through its TX pin using a UART interface.

The TX pin outputs:

- Distance measurements for each detected point (in millimeters)
- Angle information for each point (0 to 359 degrees)
- Quality values that show how reliable each measurement is
- Packet flags that mark the start of new scans and help synchronization

Software Algorithm:

1. Initialize the RPLIDAR C1 and start continuous scanning.
2. Collect a full 360-degree scan set containing angle, distance, and quality for each data point.
3. MCU UART receiver parses each packet and extracts valid data points, and filters measurements to only the front sector of the robot
4. Compute the minimum distance in the front sector.
5. Compare the minimum distance to thresholds:
    a. If distance ≤ 1.0m: STOP the robot
    b. If distance is between 1 and 1.5 meters: start SLOWING DOWN (Reduce motor PWM or wheel speed)
    c. If distance > warning distance: continue normal operation.
6. Repeat the process at the scan rate of the lidar.

Parts Required:

1. RPLIDAR C1 lidar sensor
2. 5V regulated power supply for the lidar
3. TX and RX connection cable (to connect to MCU)

Power Management (A: 100%)

Purpose:

The Power Management Module provides the required voltage for the robot from a pair of rechargeable 3S LiPo batteries. It supplies power to the motor drivers, processor, UWB modules, and LiDAR, while protecting the system with a fuse and allowing the processor to monitor battery voltage for low-battery alerts.

Output:

- 12 V to 4× motor drivers.
- 5 V rail to Teensy 4.1 and RPLIDAR C1.
- 3.3 V rail to 4× DWM1000 UWB modules.
- Battery voltage signal
- LED indicator (indicates battery is > 10%)

Parts Required:

- 2 × 3S 12 V 6000 mAh LiPo batteries
- 1 × XT60 female cable connector
- 20A inline fuse
- 5V regulator
- 3.3V regulator

Algorithm:

1. Measure the battery voltage using the ADC.
2. Convert the ADC reading to actual battery voltage.
3. If battery voltage is above 10.5 V, allow normal operation.
4. If battery voltage is between 10.5 V and 10.2 V, trigger a Low Battery warning (flash LED).
5. If battery voltage is below 10.2 V, trigger a Critical Battery state and stop the motors.
6. Repeat this measurement continuously while the system is powered.

Power Calculation:

Motor Power (4 DC Motors) – 12V / 2.2A
Rated electrical power per motor: P = 12 V * 2.2 A = 26.4 W
Assuming 60% load: P = 0.60 * 26.4 W = 15.84 W
Total motor power for 4 motors: P_total = 4 * 15.84 W = 63.36 W

RPLIDAR C1 Power – 5V / 0.26A
P = 5 V * 0.26 A = 1.30 W

Teensy 4.1 Power – 5V/ 0.1 A
P = 5 V * 0.10 A = 0.50 W

UWB Modules (4 x DWM1000) – 3.3V/ 0.16A
Total current for 4 modules: I = 4 * 0.16 A = 0.64 A
Total UWB power: P = 3.3 V * 0.64 A = 2.112 W

Total Average Power:
P_total = 63.36 W + 1.30 W + 0.50 W + 2.112 W = 67.272 W

User Tracking (B:100%)

Purpose

The User Tracking subsystem enables the robot to follow a human operator using a UWB tag. It calculates the user's position and direction relative to the robot and responds to start or hold commands from the tag, ensuring smooth and accurate tracking.

Input

- Configuration commands

Output

- Direction to the user
- Distance to the user
- Start/Hold command status.

Parts Required

a) User Node

1x UWB Transceiver (DWM1000)
1x STM32L0 (handles button input and UWB communication)
2x Pushbuttons (START, HOLD)
1x Rechargeable Li-ion or Li-Po battery (3.7 V, 1000 mAh) for the user node.

b) Robot Node

3x UWB Transceivers (DWM1000)

Algorithm

1. User node monitors START and HOLD buttons.
2. When START is pressed, the tag continuously transmits UWB pulses.
3. Robot node receives pulses on three transceivers, records timestamps, and calculates distances using Time-of-Flight.

4. Two front transceivers provide distance and direction to the user; the rear transceiver determines whether the user is in front or behind the robot.
5. Using the distance from the 2 front transceivers, the MCU will perform 2D trilateration to calculate the position of the user with respect to the robot. The average distance will also be calculated.
6. If HOLD is pressed, the tag stops transmitting and the robot pauses tracking.
7. Process repeats continuously while the system is powered, ensuring smooth real-time tracking.

Appendix

Processor:
Below is the schematic diagram for the Processor and how it connects all the different system blocks.

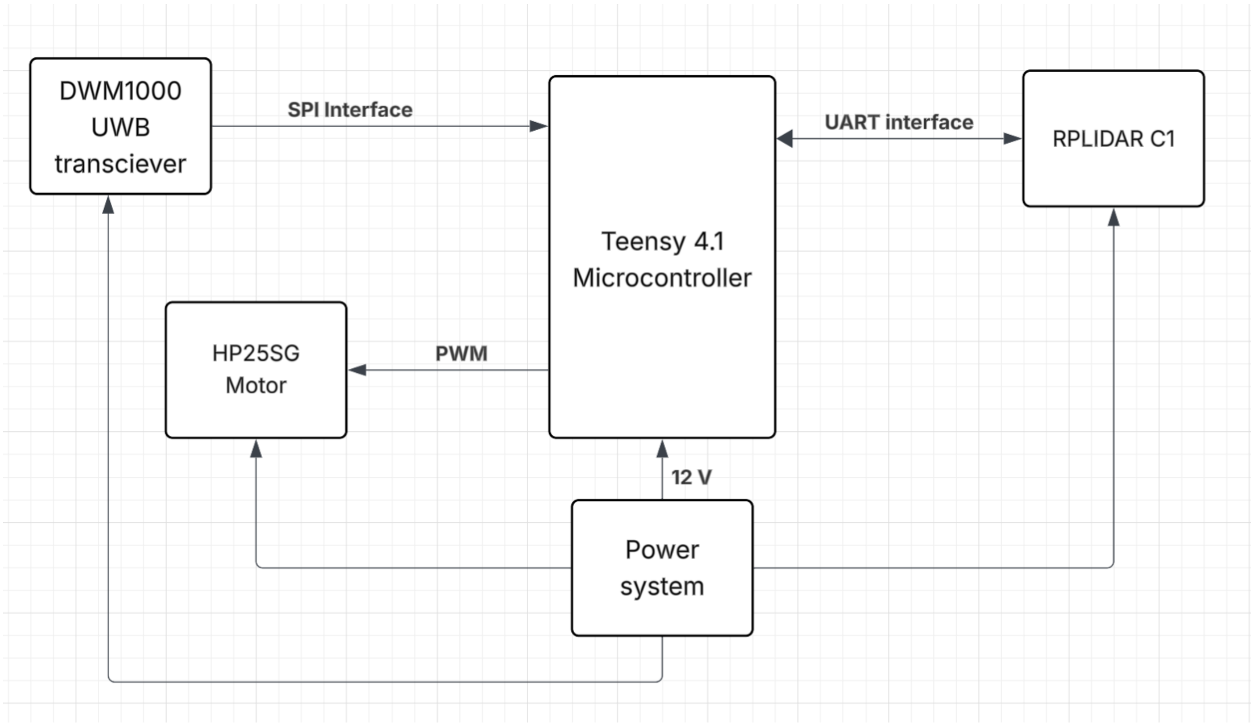

Figure 1: Schematic for Processor Subsystem

<u>Power Management Subsystem</u>:
Below is the schematic diagram for the Power Management System.

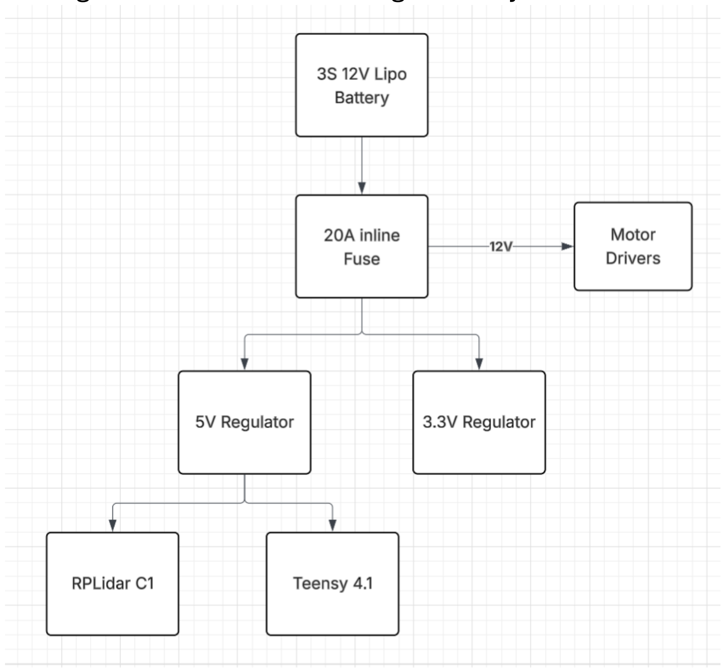

Figure 2: Schematic for Power Management Subsystem

Below is the flowchart diagram describing the software process of the Power Management system:

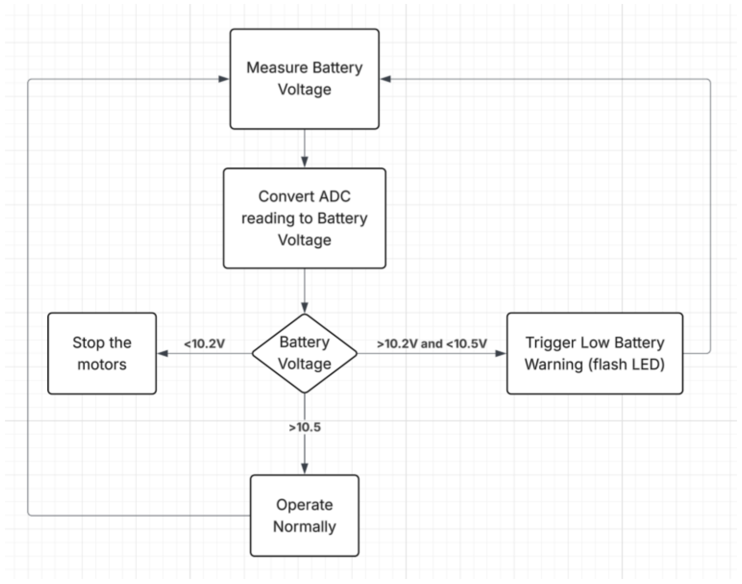

Figure 3: Flowchart for the Power Management Subsystem

Obstacle Detection Subsystem:
Below is the schematic diagram for the Obstacle Detection.

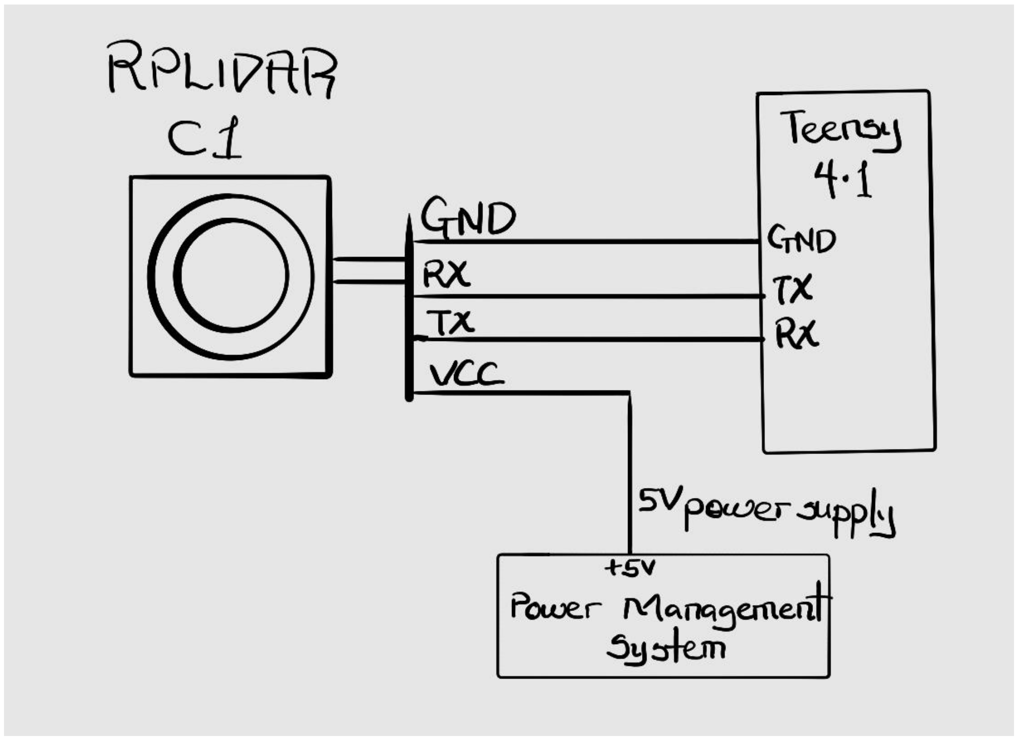

Figure 4: Schematic for Obstacle Detection Subsystem

Below is the flowchart diagram describing the software process of the Obstacle Detection system.

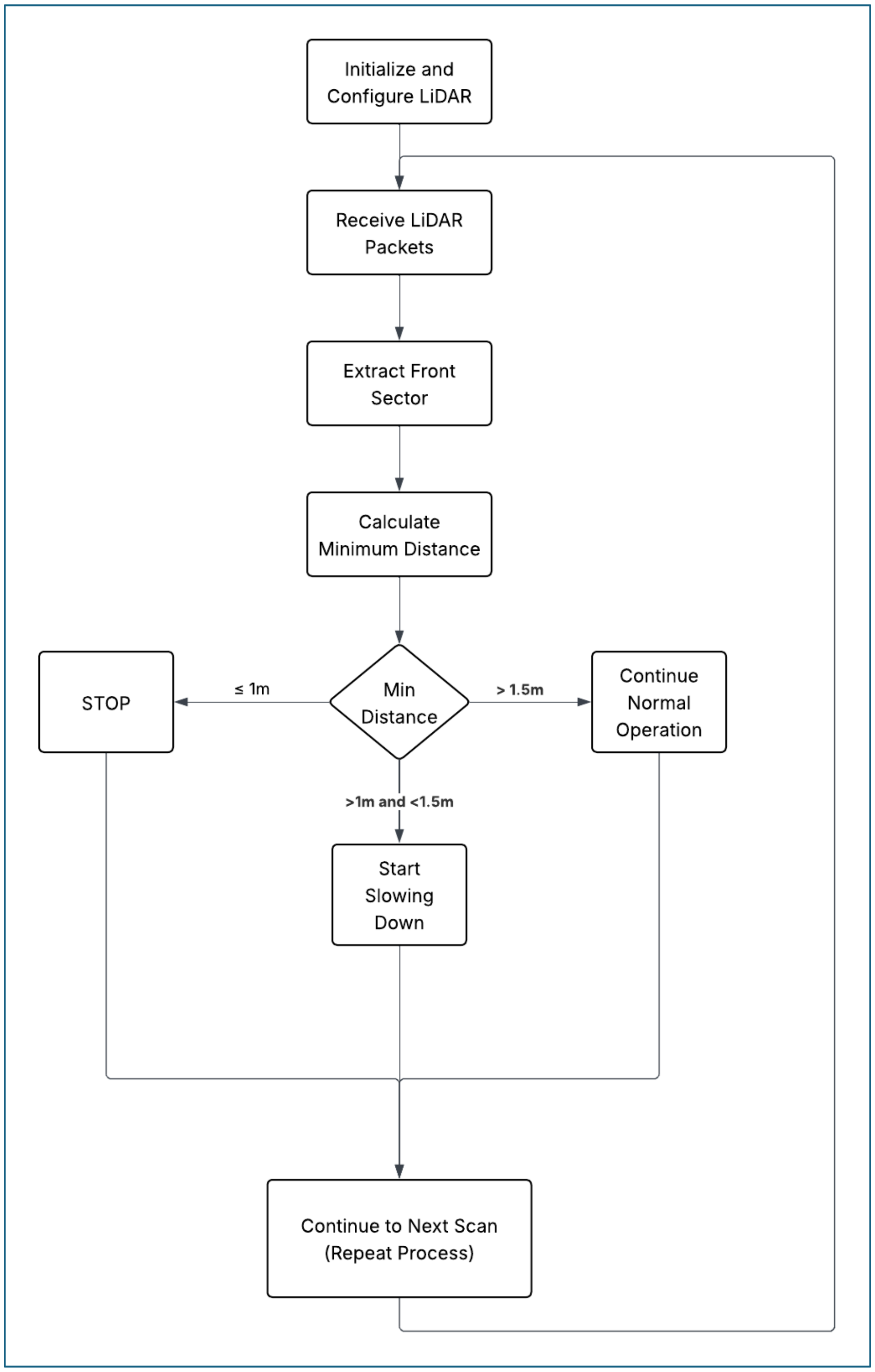
Initialize and Configure LiDAR
Receive LiDAR Packets
Extract Front Sector
Calculate Minimum Distance
Min Distance
≤ 1m
STOP
> 1.5m
Continue Normal Operation
>1m and <1.5m
Start Slowing Down
Continue to Next Scan (Repeat Process)

Figure 5: Flowchart of the Obstacle Detection System

<u>Motion Control Subsystem</u>:
Below is the schematic diagram for the Motion Control System.

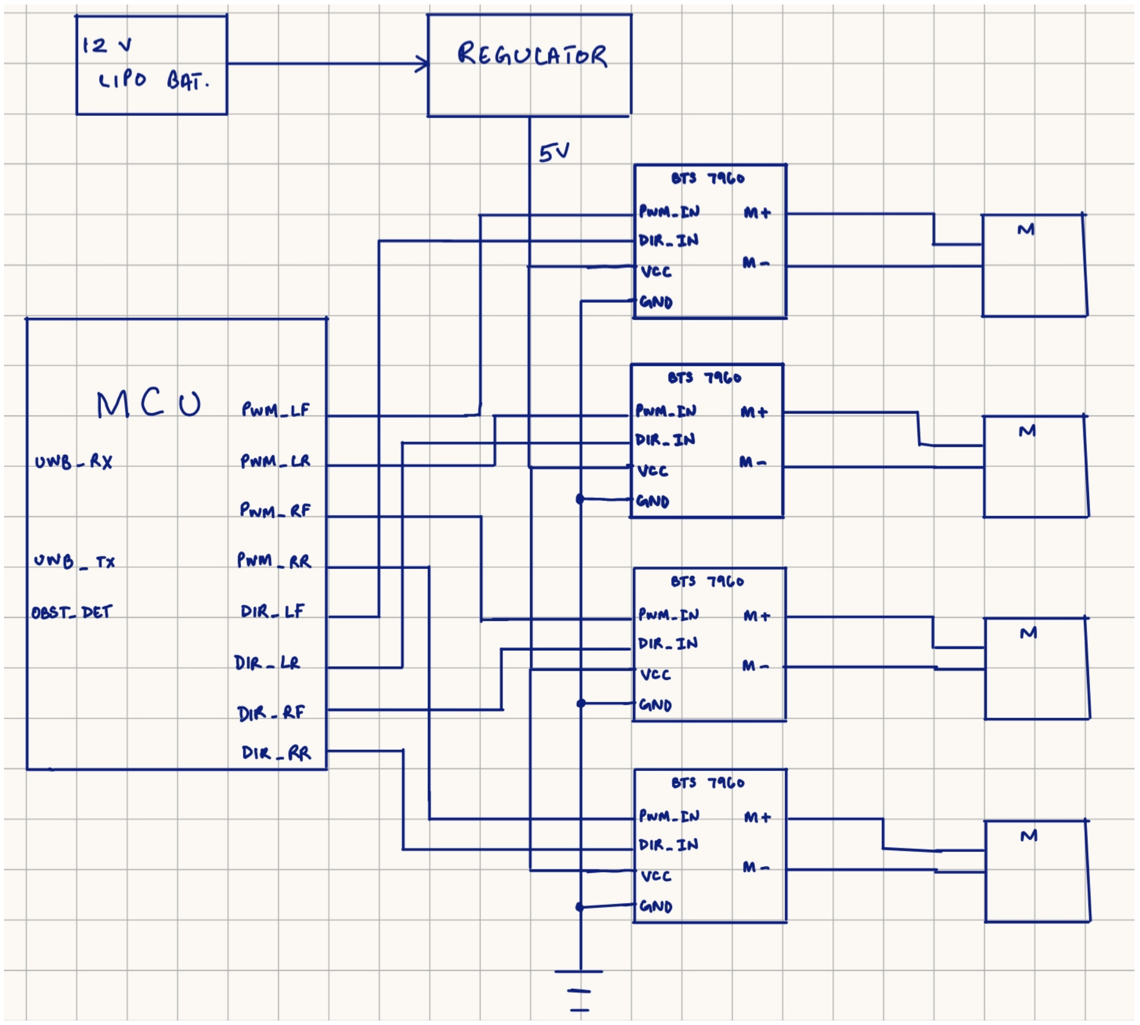

Figure 6: Schematic for Motion Control Subsystem

User Detection Subsystem

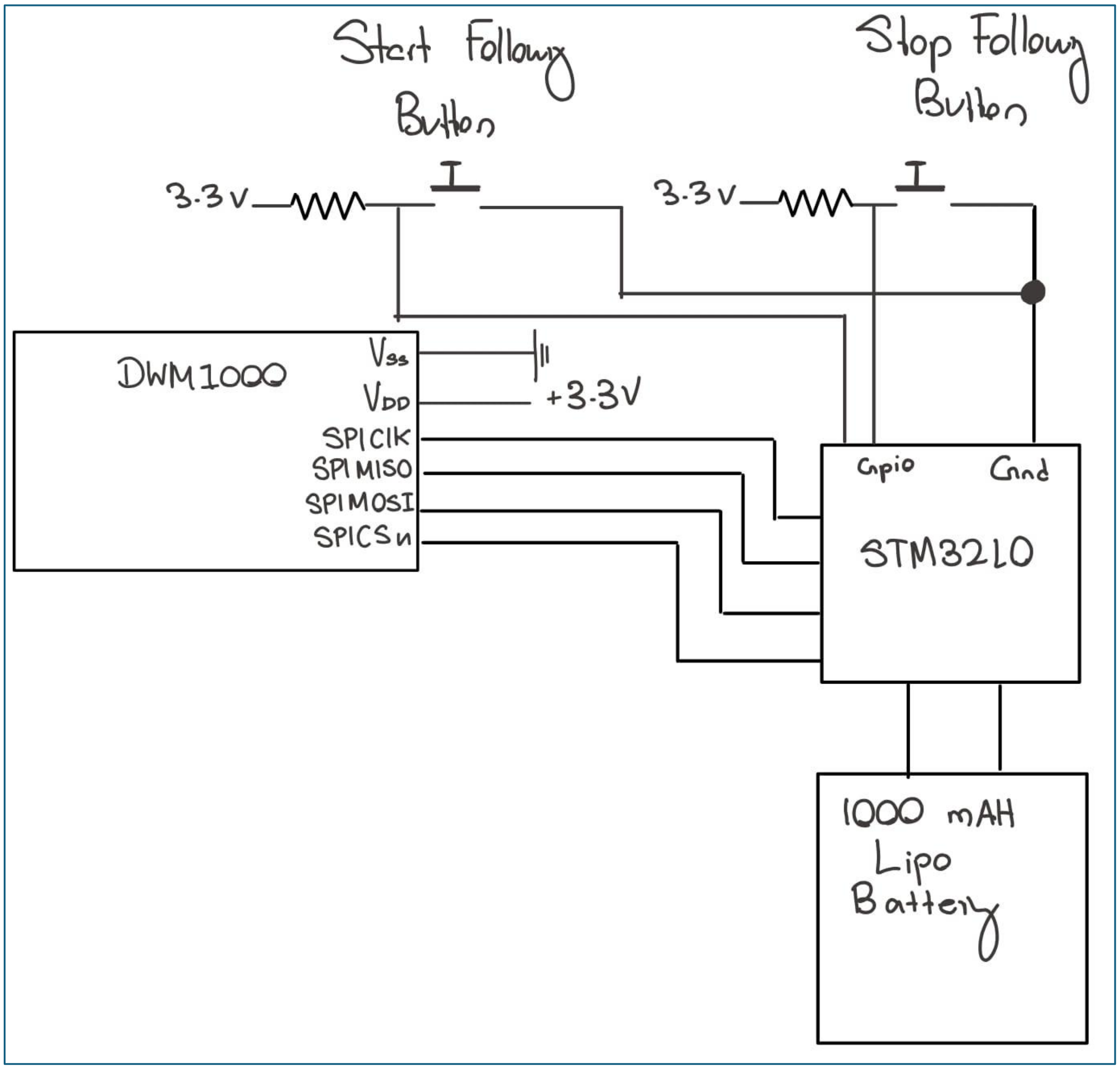

Figure 7: Schematic of User End of the User Detection Subsystem

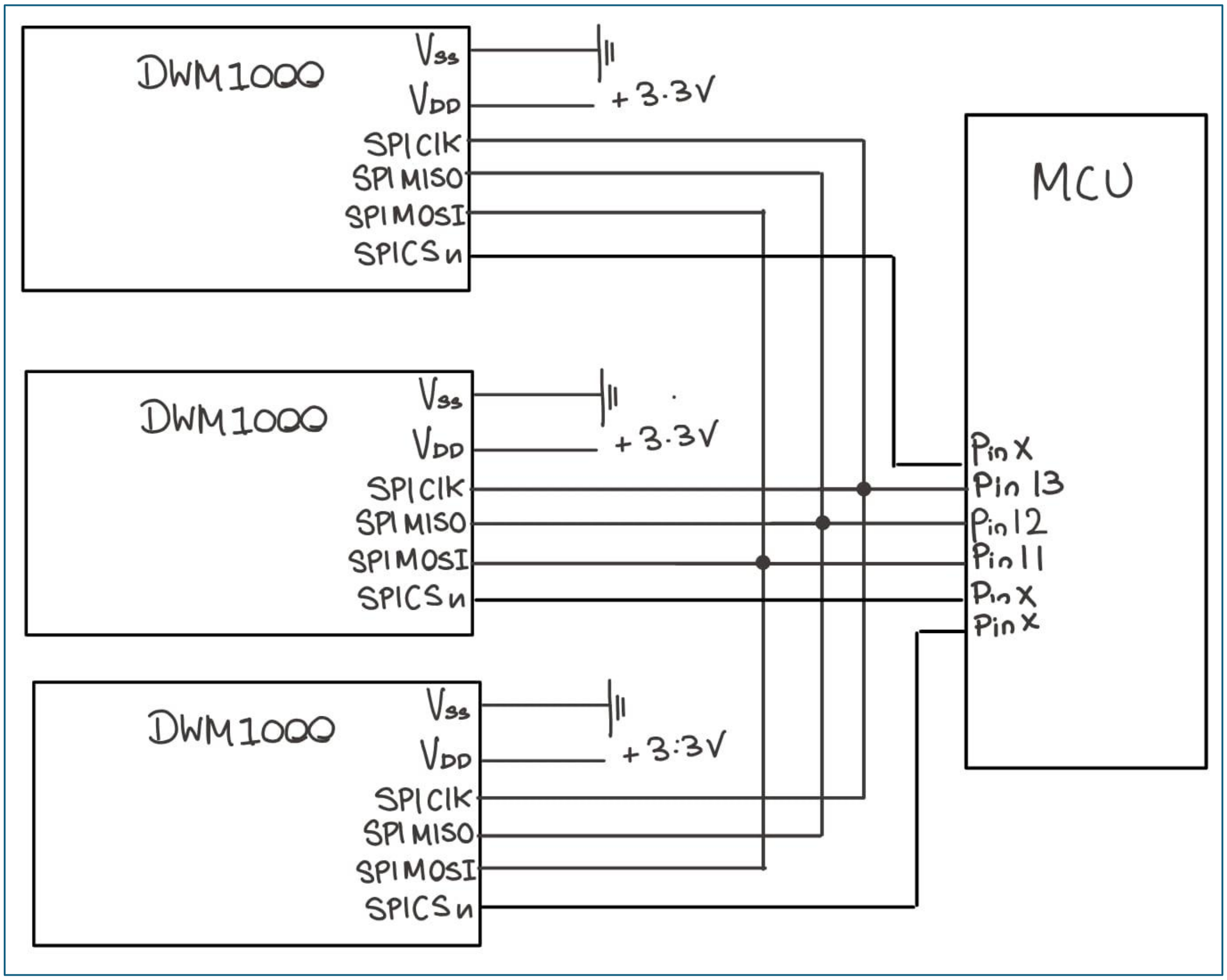

Figure 8: User end of the Robot End of the User Tracking Subsystem