ECE445

Design Document

The ILLINI Wagon

Team #39

Ian Watson (ianjw2)
Neha Joseph (nehaej2)
Ramya Reddy (ramyar3)

Table of Contents

1. Introduction

1.1 Problem and Solution

Problem

Bringing in groceries can be challenging for the elderly, and hauling a cooler, chairs, and other gear to the beach can be a hassle. Carrying heavy packages from the car, transporting picnic or camping supplies, and lugging gear for outdoor events can quickly become exhausting. Parents juggling strollers, diaper bags, and toys during outings could use an extra hand, while festival-goers and tailgaters often struggle with chairs, food, and drinks. Hikers and campers could travel farther without the strain of carrying supplies, and warehouse or delivery workers would benefit from extra support moving boxes and equipment. Whether for daily tasks, outdoor adventures, or work-related needs, the Illini Wagon can take the load off, making life more convenient.

While similar products are available on the market today, they are often either too expensive, lack true autonomy, or both, making them less practical for everyday use. Many require manual operation or remote control, which still places a burden on the user. Others rely on computer vision which drives up costs and limits their accessibility. For people who need a simple, affordable, and truly hands-free solution, there are practically zero options available. The Illini Wagon will be both cost-effective and fully autonomous in order to fill this gap, providing a practical tool for a wide range of users.

Solution

Our proposed solution is an autonomous, self-following wagon designed to carry loads for users in both indoor and outdoor environments. The wagon will employ an Ultra-Wideband (UWB) tracking system to accurately follow the user without requiring direct manual control. This system will consist of a UWB Tag and Anchor setup, where two UWB tags will be mounted on the wagon, and the user will carry a remote with an embedded UWB anchor. By utilizing the Two-Way Ranging (TWR) method, the system will determine the user's real-time position and distance through Time of Flight (ToF) calculations.

To enhance navigation stability, the wagon will incorporate additional sensors such as IMUs (Inertial Measurement Units) for motion estimation and ultrasonic or LiDAR sensors to detect obstacles and prevent collisions. A combination of onboard processing and real-time feedback loops will ensure responsive and reliable tracking, even in dynamic environments. This solution provides a hands-free alternative to traditional rolling carts and backpacks, reducing physical strain while seamlessly adapting to urban and campus landscapes.

1.2 Visual Aid

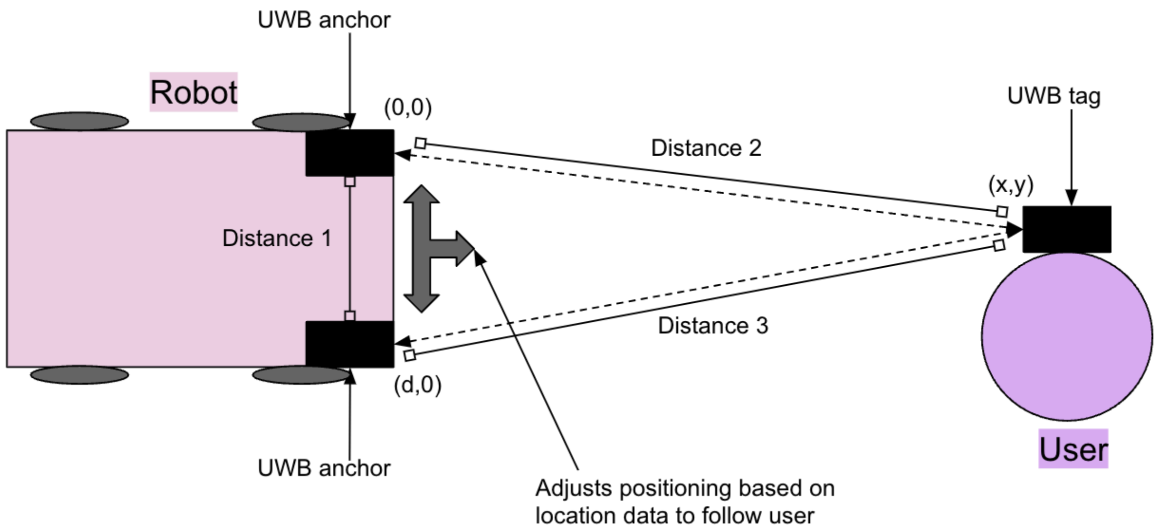

Figure 1: Visual Aid

1.3 High-level Requirements List

1. Wagon can follow user in an open, outdoor space with no obstacles.
2. Wagon is able to carry a load between 10-15 lbs.
3. Wagon is able to maintain a distance of < 2m between itself and user but no less than 0.5 meters from the user.

To demonstrate and test the robot, we will run the robot in the main quad with weighted items. If we are able to accomplish these goals early, we would also like to attempt to implement obstacle detection.

2. Design

2.1 Block Diagram

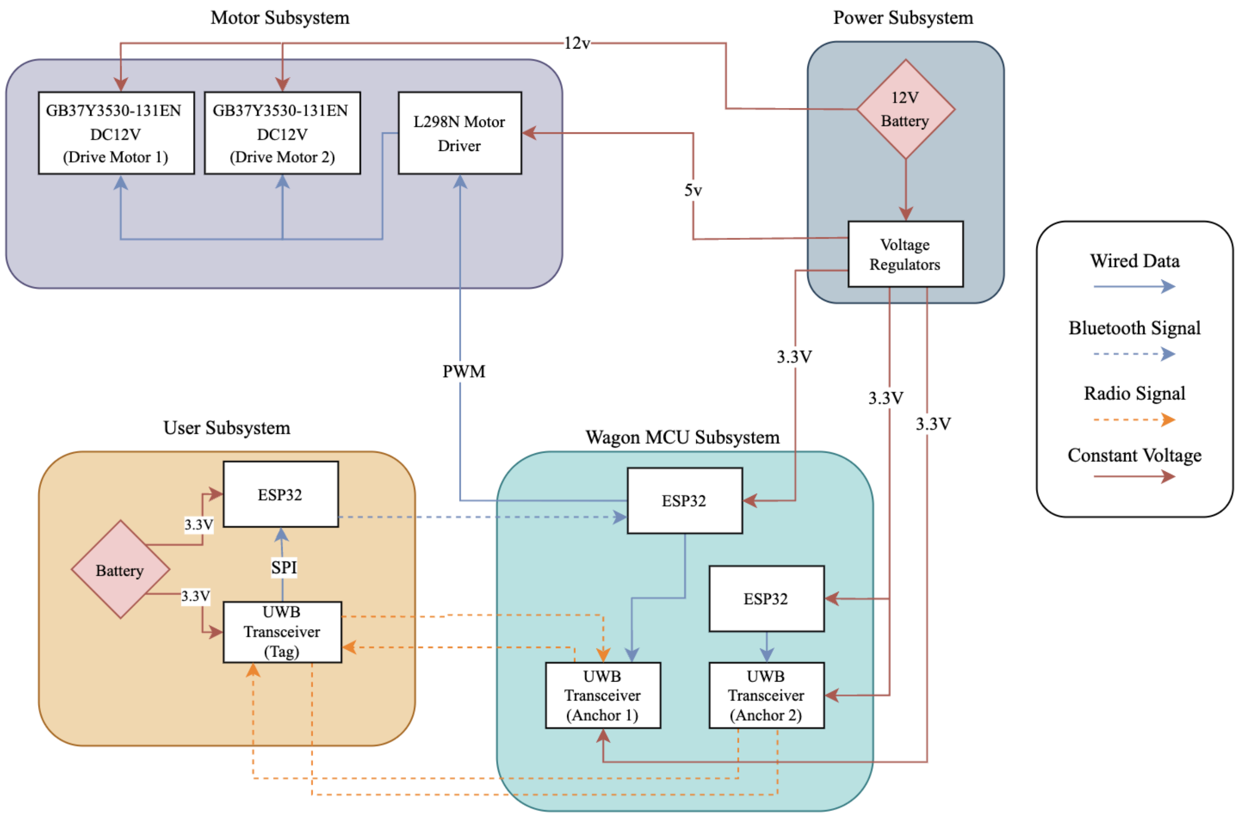

Figure 2: High-level Block Diagram of Entire System

Our design has three main subsystems, each of these having its own power subsystem: motor subsystem, user subsystem, and wagon subsystem. The motor subsystem consists of two 12V DC motors, a motor driver, and an ESP32. The wagon subsystem consists of two UWB transceivers, both acting as anchors, 2 ESP32s, and respective power systems. Each UWB module has its own ESP32 module since they will be on separate PCBs.

The user subsystem consists of a UWB transceiver, which acts as the tag, an ESP32, and a 6 V power source, which is converted down to 3.3 V to supply power to each component. This system will be carried by the user in order for the wagon to follow the user. The tag essentially sends signals to the anchors on the robot via radio waves to calculate the respective distance to each anchor. This data is then sent to the ESP32, which will translate the timestamp information into distance and angle calculations. This information is then sent to the wagon subsystem ESP32 to undergo further processing. Here it is converted into PWM signals which are sent to the motor drivers, which then control the speed of each motor.

2.2 Physical Design

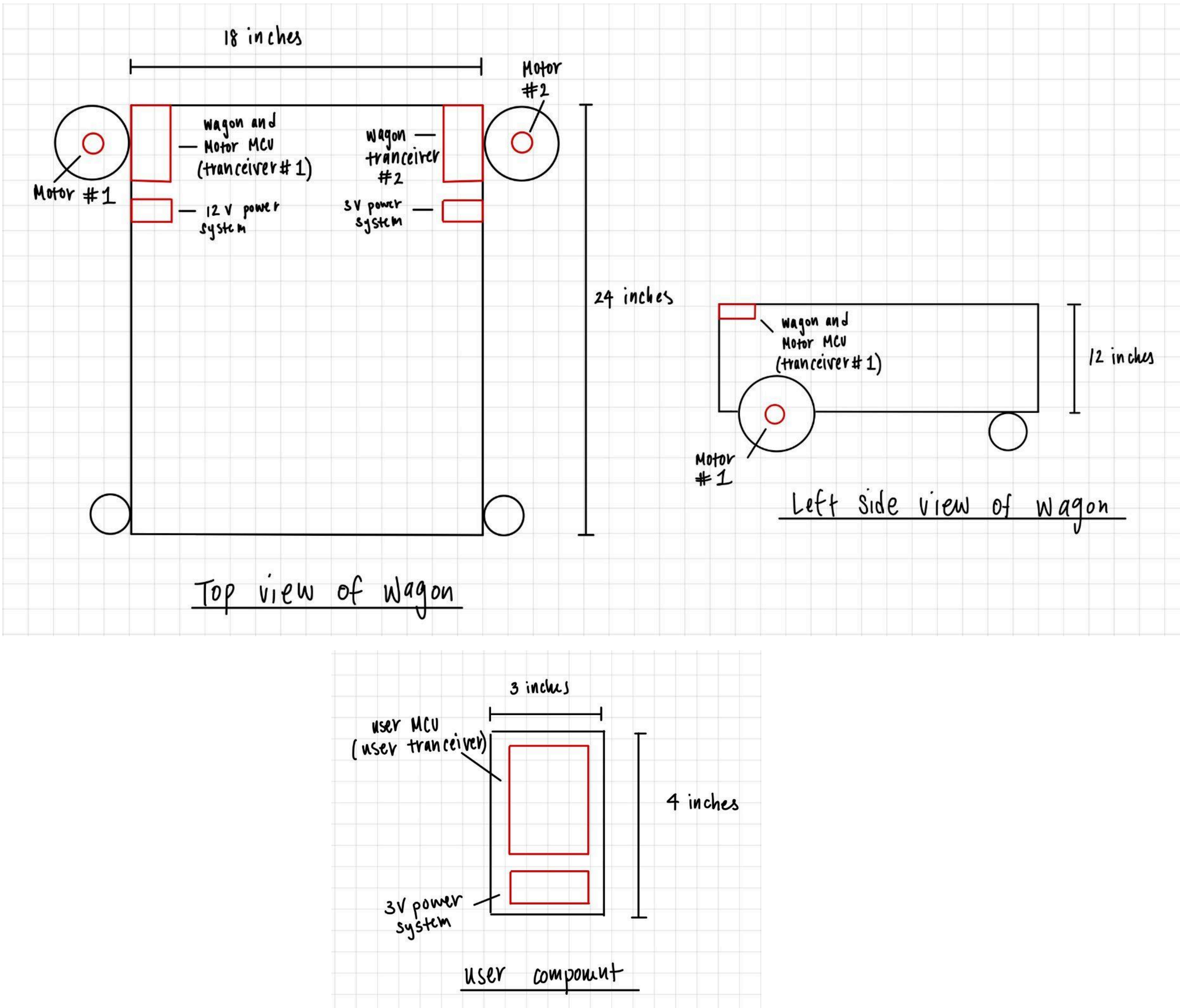

Figure 3: Physical Design of Wagon

This is a rough sketch of the wagon including the PCB placements and the wheel and motor locations. The PCBs (the wagon's motor MCU and the second wagon transceiver) will be enclosed in small boxes and screwed onto the left and right corners of the wagon. The body of the wagon will be made of wood and will measure 18 inches in width, 24 inches in length, and 12 inches in height. The wagon will have an estimated weight of around 15lbs. Motors will be attached to the front wheels, which have a diameter of 14.6cm, with power supplied from the wagon's motor MCU component. The back wheels will be simple soft rubber wheels designed to support the wagon's movement. The user component will be a small enclosed box containing the user MCU PCB and the power supply with the dimension of around 3 inches in width, 4 inches in length, and around 1 inches in height.

2.3 Subsystem Overview

2.3.1 Power Subsystem

High Level Purpose and Description:

Our design has three main power systems, each corresponding to its respective subsystem. The motor subsystem will be powered by a 12 V rechargeable battery. The wagon subsystem will be powered by four 1.5 V AA batteries connected in series, outputting 6 V in total. The user subsystem will also be powered by four 1.5 V AA batteries. We chose to have three separate power systems because each subsystem has its own PCB.

The motor power subsystem is responsible for supplying voltage to one ESP32, one UWB transceiver (DWM1000), the motor driver, two motors, and the USB connection module. To supply the correct voltage to each of these components, we have two systems for stepping down the 12 V source. We first have a buck converter, that steps down 12 V to 5 V. This will supply power to the L298N motor driver chip. We then have a voltage regulator, that steps down the 5 V power supply from the buck converter to 3.3 V. This will power the UWB transceiver, the ESP32, and the CP210N USB to UART master bridge chip.

The user and wagon power subsystems are identical. They are each responsible for supplying voltage to one ESP32 and one UWB transceiver. It will step down the 6V power source to 3.3V utilizing an AP2112K-3.3 fixed voltage converter.

Parts used:

- 12 V 5200mAh rechargeable battery
- 2x 4 AA batteries
- 2x Quad AA battery holder
- LM2596-5.0 Buck Converter
- 3x AP2112K-3.3 voltage regulator

Justifications for Each Component:

<u>12 V 5200mAh Battery:</u>

For the motors in the motor subsystem, we see a minimum start voltage of 1 V, and an ideal running voltage of 12 V. Due to this operational range, we have decided to use a 12 V battery. We have specifically chosen the KBT 12 V 5200mAh battery (1C), as it is able to handle the max current draw of our components. It also has a max continuous discharge rate of 15.6 A (3C), which means it can handle the worst case stall current of 14A total for both motors. These calculations are detailed below.

As determined in our tolerance analysis, we have calculated a max continuous current draw of 2A per motor. Based on documentation we have also found a max current draw of 40mA from the motor driver module, 352mA for the ESP32, and 160mA for the UWB transceiver. This results in a total current draw of 4.552 A.

$$I_{max} = 2A + 2A + 0.16A + 0.04A + 0.352A += 4.552A$$

Since the motors have a max stall current of 7A per motor, this results in a total stall current of 14A. Since we are not straining the motors with a hefty load, the motors may only experience a surge in current requirement upon start up. Assuming the motors experience a surge in current usage for a maximum time of 1 minute, we have calculated the max current draw upon start up to be:

$$Time = \frac{1\,min}{60\,min} = 0.0167\ hours$$

$$I_{drawn} = 14A * 0.0167\ hours = 0.2333Ah$$

$$5.2Ah - 0.2333Ah = 4.96667Ah\ remaining\ battery\ life$$

Based on this information and the max current draw calculated previously, we will have a battery runtime of:

$$Runtime = \frac{Battery\ Capacity\ (Ah)}{Current\ Draw\ (A)} = \frac{4.96667Ah}{4.552\ A} \approx 1.091\ hours \approx 65.46\ minutes$$

Since this wagon is a prototype model, a runtime between 45-60 minutes should suffice, and this battery would provide the system with enough power to fulfill that requirement.

<u>4x AA Batteries:</u>

According to the Duracell datasheet, their Coppertop AA batteries have a voltage of 1.5 V and a total service hours time of approximately 3 hours when current is being drawn at a constant rate of 500 mA. This can be seen from the image taken from the data sheet below. Since both the Wagon and User systems require approximately 3.3 volts and 0.515 A of current, the AA batteries will provide more than sufficient power for our goal runtime between 45-60 minutes.

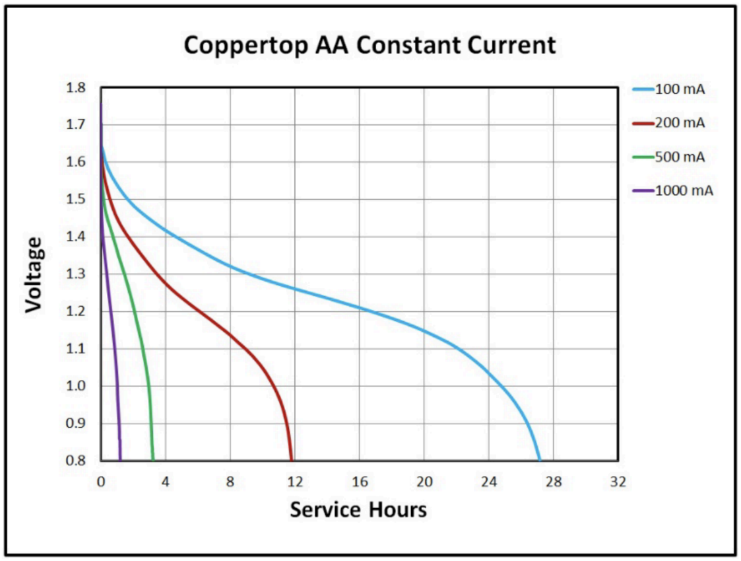

Figure 4: Voltage vs. Service Hours for Duracell Optimum AA Battery Given Constant Current

LM2596-5.0 Buck Converter:

Initially, when designing the motor power system, we debated utilizing a standard 5 V voltage regulator capable of handling a 12 V input. After further research, we determined that the heat dissipation from a 7 V voltage drop utilizing a standard voltage regulator would be too much for our system to handle, especially given that we are using sensitive components such as the DWM1000. To properly handle this high voltage drop, we decided to utilize a step-down buck converter, which generates minimal heat. This specific part we have chosen allows for an output current of 1.4A, which will be enough to support the motor driver logic, max current draw from the ESP32, and the max current draw from the UWB transceiver. The schematic for the buck converter implementation from Texas Instruments documentation can be seen below.

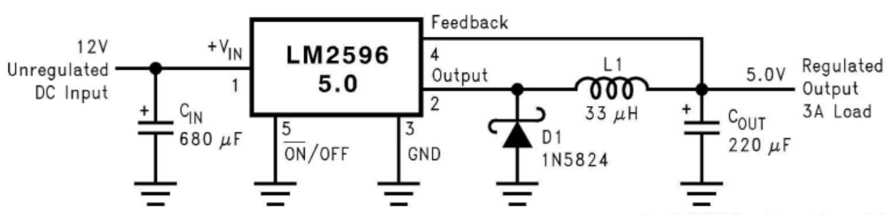

Figure 5: Typical Application Circuit for LM2596-5.0

AP2112K-3.3 Voltage Regulator:

This voltage regulator is used in all three power systems in our design. On the motor subsystem, it will convert the 5 V output from the buck converter to 3.3 V. On the user and wagon subsystems, it will convert the 6 V power source to 3.3 V. We decided to utilize a voltage regulator in both cases since the voltage drops are only 1.7 V and 2.7 V, and will not result in high levels of heat dissipation. This chosen voltage regulator is able to provide an output current of 600 mA, which will be sufficient for the max current draw of both the ESP32 and the DWM1000 modules. The schematic for the voltage regulator implementation from Diodes Incorporated can be seen below.

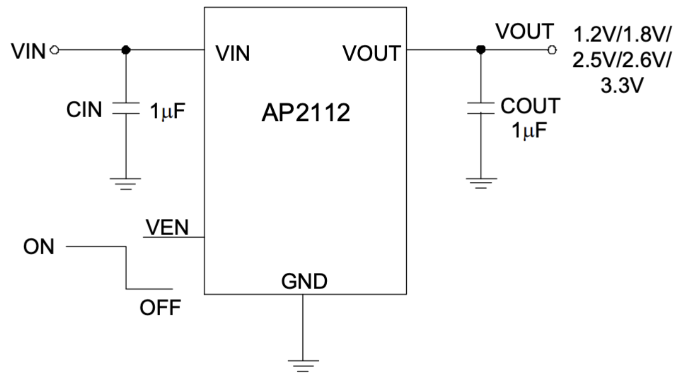

Figure 6: Typical Application Circuit for AP2112

High level requirements table/verifications:

Requirements	Verification
Provide 14 A continuous current for 1 minute.	● Turn off the motors completely, then power them up using the battery pack. ● Ensure that the battery pack can handle any surge in current draw upon start up. ● Ensure voltage provided by the battery stays within 10.5-12 V range after current draw.
Provide a stable voltage source of 5 V to the motor driver module.	● Measure output voltage from 12 V to 5 V buck converter using an oscilloscope to ensure it is providing a steady voltage of 5V±.25 V.
Provide a stable voltage of 3.3 V to the ESP32 and DWM1000.	● Measure output voltage from 5 V to 3.3 V voltage regulator using an oscilloscope to ensure it is providing a steady voltage of 3.3V±.165 V.
Ensure the system can handle the current and voltage requirements of all components for between 45-60 minutes.	● Using an alternate power supply (non-battery), we will measure the current draw using a multimeter for each large component. ● We will then ensure that the total current draw from the system fits within 5.2A, to verify the validity of the chosen 12 V 5200mAh battery supply.

2.3.2 Motor Subsystem

High Level Purpose and Description:
The Illini Wagon's motor subsystem will be responsible for controlling its speed and direction of movement. It features two motors that drive the front wheels. These motors are powered by a motor driver (L298N), which receives signals from the microcontroller (ESP32). The microcontroller processes input from the wagon and user subsystems in order to update, in real time, the values of the PWM signals sent to each motor.

Parts used:
- Non-PCB Components
    - 2x GB37Y3530-131EN (*motor*)
    - 2x 5.75 in. Diameter Wheels
- PCB Components
    - 1x L298N (*motor driver*)
    - 1x ESP32-S3-WROOM-1 (*microcontroller shared with wagon subsystem*)

Justifications for Each Chip:
<u>L298N:</u>
To facilitate the steering of the wagon, we will be utilizing a motor driver. The motor driver will, in layman's terms, increase the voltage of the signal sent from the ESP32 so that it can be sent to the motor.

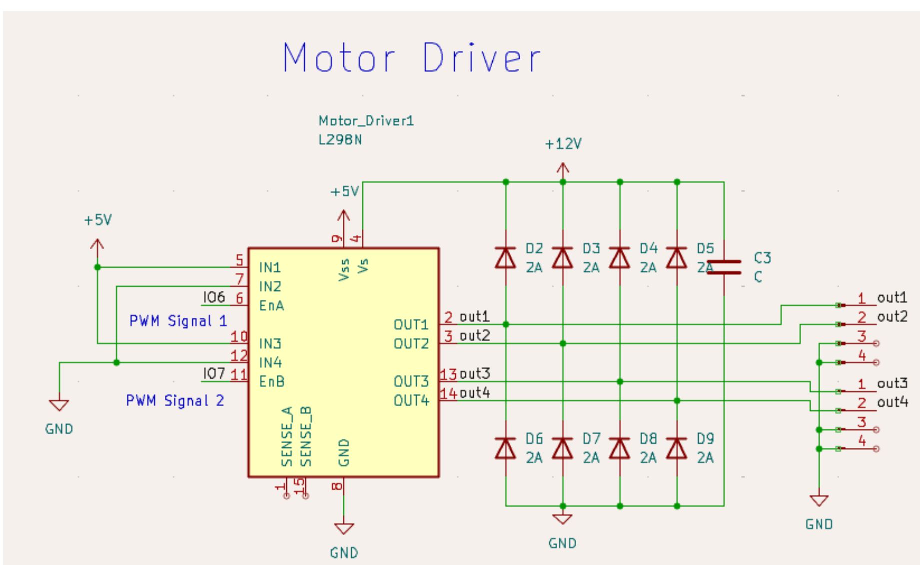

Figure 7: L298N (Motor Driver) Schematic

ESP32-S3-WROOM-1:
This microcontroller will be used in order to generate the PWM signals sent to the motor driver, and will handle the calculations from data being received by the UWB transceiver.

Interactions With Other Systems:
The user and wagon subsystems will work together to provide real-time distance measurements. Utilizing triangulation, we will be able to calculate the angle between the user and the wagon. This information will be sent to the motor subsystem in order to communicate with the motors what the duty cycle values should be.

To steer the wagon, we will be utilizing two distinct Pulse Width Modulation (PWM) signals. When the user is located to the left of the wagon, the value of the duty cycle on the right motor will be greater than the value of the duty cycle of the left motor. When the user is located to the right of the wagon, the value of the duty cycle of the left motor will be greater than the value of the duty cycle of the right motor. To control the speed of the wagon, the distance between the user and wagon will be used as the base value for the duty cycle of each motor. The exact equations that will be used to facilitate this movement are shown below.

Duty Cycle Calculations and Triangulation Method:
The distance between each motor is approximately 0.457 m. Assume the distance from the left motor to the user is x_0 and that the distance from the right motor to the user is x_1. Since we know the length of each side of the triangle, we can use the law of cosines to find the angle of each side of the triangle.

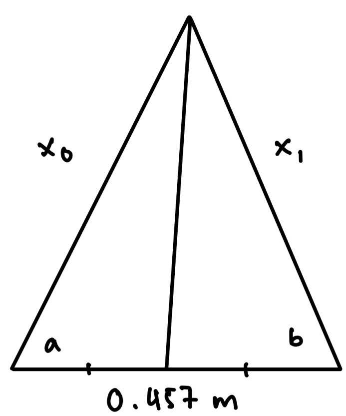

Figure 8: Wagon Steering Geometry

Angle a can be expressed as $cos^{-1}(\frac{x_0^{\ 2}+(0.457)^2-x_1^{\ 2}}{2*x_0*0.457})$

Angle b can be expressed as $cos^{-1}(\frac{x_1^{\ 2}+(0.457)^2-x_0^{\ 2}}{2*x_1*0.457})$

The length median of this triangle can be expressed as $\frac{1}{2}\sqrt{2x_0^{\ 2} + 2x_1^{\ 2} - 0.457^2}$

These values will get updated in real time. The length of the median will be proportional to the base values of the duty cycles. The duty cycle will cap out when the median length exceeds 3 meters. The exact duty cycle values will be chosen later after the wagon is constructed.

As angle a increases, the duty cycle sent to the motor located at angle b will increase, while the duty cycle sent to the motor located at angle a will decrease, and vice versa.

High-Level Requirements Table/Verifications:

Requirement	Verification
RPM of drive motors is proportional to distance from user and falls within 0-200 RPM	● Turn on wagon, then walk away from wagon ● Measure RPM while the user walks away and ensure it increases with distance, and caps out at 200 RPM.
Wagon points to user within 0.15 radians within .5 seconds	● Ensure that each when the user is to the right of the wagon, the left wheel has a higher duty cycle, and vice versa. ● Measure this using an oscilloscope. ● Test the duty cycle values by having wagon operate in controlled environment
Wagon is able to maintain constant distance of 1.3 to 1.7 meters, assuming walking speed	● Bring the wagon to a controlled area, turn on the wagon, and have the user stand close to the wagon to ensure it does not move. ● Have the user walk away slowly and ensure that the wagon is able to track the user within 1.3 to 1.7 meters.
Current supplied to the motor will be between .4A to 7A.	● Have the user within stop distance of the wagon and ensure that the duty cycle is 0, and confirm the wheels are

	not spinning. ● Measure current with current sensor and ensure that it is within 300-400 mA. ● Have user stand far away from wagon, and ensure that the duty cycle is at its maximum ● Measure current again and ensure the value is less than 7A.

2.3.3 User and Wagon Subsystem

High Level Purpose and Description:
The user and wagon subsystems are responsible for communicating with each other to determine and calculate the user's absolute location. The user subsystem consists of a remote carried in the user's pocket, which includes a UWB transceiver and an ESP32 microcontroller. The wagon subsystem consists of two UWB transceivers mounted on either side of the wagon, along with two ESP32 microcontrollers that act as hosts for the UWB transceivers (one of the ESP32 microcontrollers is shared with the motor subsystem). The user and wagon subsystems use the UWB transceivers to implement the TWR (Two-Way Ranging) methodology to calculate the distance between the wagon and the user. TWR is performed by assigning the UWB transceiver on the user subsystem as the tag, and the UWB transceivers on the wagon subsystem as the anchors.

The user subsystem has three main components: a DWM1000 UWB transceiver, an ESP32 module, and a mini power system supplying 3.3V as described in the power system above. The wagon subsystem comprises : two UWB anchors that communicate with the user's UWB tag, two ESP32 microcontrollers, and power sources that will supply 3.3V of power to each component as described in the power subsystem description.

The user subsystem communicates the distance information to the ESP32 on the motor subsystem, which then translates the RPM information to send to the motors. The UWB tag essentially sends and receives signals to the anchors on the robot to calculate distance. The tag initially transmits a message to the anchors, the anchors receive the message and transmit responses back to the tag, and the tag finally receives the response and records the receive timestamp. The tag will store this timestamp data, which will be sent to the ESP32 module through SPI for further processing using the methodology described below. Once this distance calculation has been completed, it will be transmitted to the ESP32 module on the wagon via bluetooth, which will send out information to the motor system.

The following two-way ranging methodology for calculating distance equations were provided through the Decawave DW1000 two-way ranging implementation document [3].

$$TOF = \frac{t_2 - t_1 - t_{reply}}{2}$$

Equation 1: Time of Flight Calculation

t_1: $time\ of\ initial\ transmission$

t_2: $timestamp\ of\ response\ receival$

t_{reply}: $delay\ from\ anchor\ transmitting\ response$

$$distance = C * TOF$$

Equation 2: Distance Calculation

TOF: $calculated\ time\ of\ flight\ value$

C: $speed\ of\ light$

Parts used:

- 3x DWM1000 (*UWB transceivers*)
- 3x ESP32-S3-WROOM-1 (*microcontrollers one shared with motor subsystem*)

Justifications for Each Component:

ESP32-S3-WROOM-1:

We chose the ESP32 microcontroller because it is one of the most widely used microcontrollers, with extensive documentation available online. Additionally, it was a cost-effective choice, as it was available for free through the ECE Service Shop. Finally, the ESP32 provides Bluetooth communication, which aligns perfectly with the communication pipeline we wanted to establish between the user and wagon subsystems.

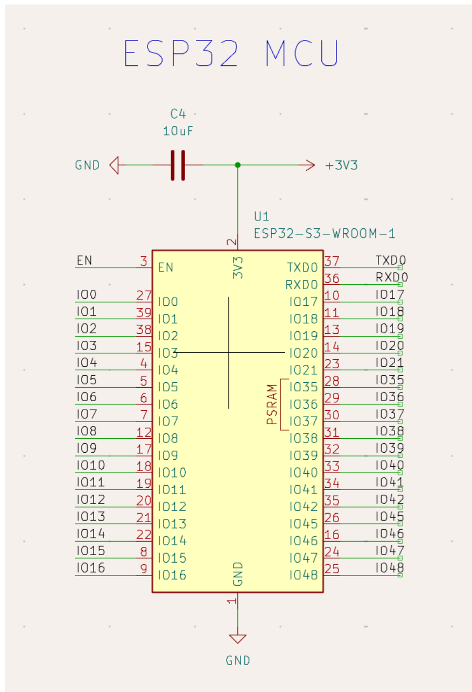

Figure 9: ESP32 MCU Schematic

DWM1000:

Initially, we considered using a GPS system to monitor the user's location and send coordinates to the wagon subsystem to control the wagon's movements. However, we quickly realized that the accuracy of a GPS system (±1.8m) was insufficient for our needs, as we wanted the wagon subsystem to stay within 1-2 meters of the user. We also explored the possibility of using Bluetooth positioning; however, we found it to be both costly and inaccurate. After exploring other sensor options, we decided to use UWB sensors because they offer superior location accuracy (±30 cm). Moreover, the UWB sensors provide little to no delay in positioning compared to other positioning options. Upon further research, we selected the DWM1000 UWB transceivers because they can function as both the anchor and the tag. Additionally, the DWM1000 transceivers were more cost-effective compared to other UWB options.

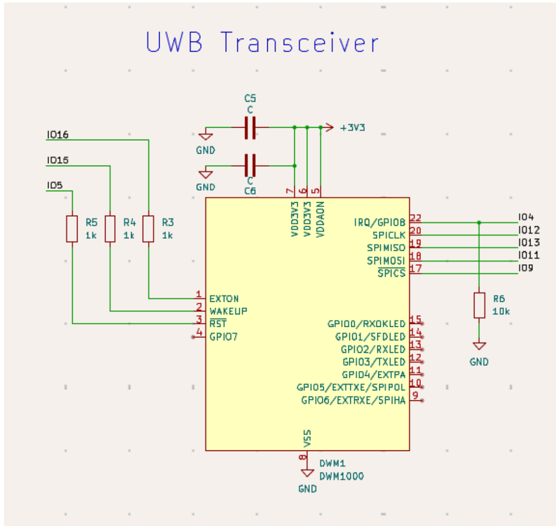

Figure 10: UWB Transceiver Schematic

High level requirements table/verifications:

Requirements	Verification
Transmit data back to the wagon MCU unit on the robot	● We will ensure that we write our code in a way that enables us to see the data that is transmitted as packets back to the wagon system via on screen/on terminal display.
Calculate TOF and distance information through data stored on UWB tag	● We should be able to see the printed values for time stamps/distance on a pop up window or on the terminal via source code. ● We will measure the accuracy of these calculations by comparing the physically measured distance to the calculated distance.
The two UWB anchors should be able to send and receive signals from the user's UWB tag	● Since we are able to display the distance values on the screen, we can also see what information is being transmitted between the UWB modules. ● We can determine the accuracy after computing the distance.

2.4 Tolerance Analysis

One of the key risks to the project's success is whether the Wagon motor and power system can achieve the required speed and carry the expected load at that speed. The motor's ability to handle weight and maintain speed depends on its stall torque and stall current, which are the maximum torque and current that the motor can handle. The motor we selected, the GB37Y3530-131EN, requires a 12V power source, has a stall torque of 45kg-cm and a stall current of 7A. The radius of the wheels that we are using is 7.3cm.

Given that the average human walking speed ranges from 2.5 to 4 mph, and the wagon is expected to carry 10 to 15 lbs of load, we can assess whether the motor can sustain the worst-case scenario.

Let's assume that the robot is placed at 2.5m from the user. In the worst case scenario, the wagon, weighing 15 lbs, carrying 15 lbs of load (15 + 15 lbs = 30 lbs = 13.6 kg), must accelerate from rest to 4 mph (1.8 meters/s) within 2 meters of space.

Acceleration required:

$$V_f^2 = V_i^2 + 2a\Delta x$$

$$1.8^2 = 0 + 2a$$

$$a = 0.81 m/s^2 \Rightarrow 81 cm/s^2$$

Force required:

$$F = ma$$

$$F = 13.6kg * 81cm/s^2 = 11.016N$$

Torque required:

$$T = Fd$$

$$T = 11.016N * 7.3cm = 8.2 \text{ kg-cm}$$

Angular Velocity:

$$\omega = v/r$$

$$\omega = 1.8m/s / 0.073m = 24.66 \text{ rad/s}$$

Power required:

$$P = T\omega$$

$$P = 8.2\text{kg-cm} * 24.66\text{rad/s} = 0.804\text{N-m} * 24.66\text{rad/s} = 19.83W$$

Current required:

$$I = P/V$$

$$I = 19.83W / 12V = 1.653A$$

According to our calculations, the worst-case scenario will require a torque of around 8.2 kg·cm, which is well within the motor's stall torque of 45 kg·cm. Additionally, the required current is around 1.7 A, which is also well within the motor's 7 A stall current. This analysis demonstrates that the selected motor, GB37Y3530-131EN, can handle the worst-case scenario and will be capable of providing stable movement while carrying up to 15 lbs at walking speed, following the user. For this project, we will limit the current to a maximum of 2 A per motor.

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

We have taken the average starting salary of both Electrical and Computer Engineers at the University of Illinois at Urbana-Champaign.

$$Hourly\ Wage\ =\ \$98472.5/52\ weeks/5\ days/8\ hours\ =\ \$47.3425481$$

Assuming our wage is \$47.34/hr and that we each spend 200 hours on this project, the total compensation is calculated as $\$47.34\ *\ 3\ employees\ *\ 200\ hours\ =\ \$28,404$

3.1.2 Parts

Part Number/Description	Manufacturer	Quantity	Unit Price	Total Price	Link
DWM1000	Qorvo	3	$25.00	$70.86	Link
ESP32-S3-Wroom-1	Espressif	3	$6.56	$19.68	Link
12V Metal DC Geared Motor	DF Robot	2	$16.50	$33.00	Link
12V 5200mAh Rechargable Battery	KBT	1	$32.99	$32.99	Link
4x AA batteries	Duracell	2	$4.99	$9.98	Link
AA battery holder	Jameco Electronics	2	$3.04	$6.08	Link
AP2112K-3.3 (voltage regulator)	Diodes Incorporated	3	$0.52	$1.56	Link
LM2674M-5.0 (buck converter)	Texas Instruments	1	$3.97	$3.97	Link
Polarized 100uF	Vishay/Sprague	2	$0.99	$1.98	Link

capacitor 293D107X9016D2TE3					
220 uF (35V) capacitor EEE-FK1V221AV	Panasonic	1	$1.59	$1.59	Link
100 uH Inductor IFDC2020CZER101M	Vishay/Dale	1	$0.61	$0.61	Link
20V 1A Schottky diode CDBA540-HF	Comchip Technology	1	$0.47	$0.47	Link
Fast 2A Schottky diode RBR3MM40ATR	ROHM Semiconductor	8	$0.33	$2.64	Link

From the table above, our total comes out to $185.41.

Many of the smaller PCB components such as resistors and capacitors have not been listed in the final cost, as they are minimal cost components that have been provided to us through the ECE department. Some parts such as the motors, ESP32s, etc. have also been provided to us through the ECE department, but the cost has been listed to give a more accurate total price estimate.

3.1.3 Grand Total

Given the cost of the parts and labour put into the project, we have come up with a grand total of $28,589.41 to complete this project.

3.2 Schedule

Week	Task	Person
2/23-3/1	PCB routing and pcb design review	Everyone
	Motor subsystem pcb/schematic	Ian
	User/power subsystem pcb/schematic	Ramya
	Wagon/motor subsystem pcb/schematic	Neha
3/2-3/8	Add USB to pcb/schematic, order parts, work on design document, breadboard demo, UWB implementation	Everyone
	Look at components for pcb	Ramya
	Look at components for pcb	Neha
3/9 - 3/15	Research UWB, rest of chips, wire USB connection to esp32 on both PCBs	Ian
	Research ESP32, rest of chips, and look at enable pin connection from ESP32 to USB	Ramya
	Research ESP32 and UWB	Neha
3/16 - 3/22 (Spring Break)		
3/23 - 3/29	Solder and test second PCB. Work on third PCB. Design wagon. Use breadboard circuits to collect data for communication between UWBs and ESP32s	Everyone
	Finalize math calculations for triangulation, look into math calculation for turning	Ian
	Look into programming the ESP32 for motor control	Ramya
	Look into enabling communication between 2 tags and 1 anchor	Neha
3/30 - 4/5	Individual progress report, see if any changes need to be made to PCB	Everyone
	Implement code for controlling motors for turning, stopping, etc.	Ian
	Enable system to Wagon system ESP32 bluetooth connection (ESP32 to ESP32 communication), Translate UWB data to information needed for motor system calculations	Ramya
	Implement communication between 2 tags and 1 anchor (UWB to UWB communication)	Neha
4/6 - 4/12	Order PCB fourth round (if necessary), put each PCB component onto the wagon and the user device.	Everyone
	Test turning calculations and make adjustments if necessary, test speed/acceleration	Ian
	Debugging double tag-to-anchor communication, debugging ESP32 bluetooth communication	Ramya
	Debugging double tag-to-anchor communication, debugging ESP32 bluetooth communication	Neha
4/13 - 4/19	Run tests for system in outdoor and indoor areas, team contract assessment, work together to debug any issues	Everyone
4/20 - 4/26	Mock Demo	Everyone
4/27 - 5/3	Final demo and mock presentation	Everyone
5/4 - 5/10	Final presentation and final paper	Everyone

4. Ethics and Safety

Our primary demographic for this project is college students, hence the name 'Illini Wagon.' However, this product can also benefit other groups, including the elderly, individuals with mobility impairments, and families. Although our intention for this project is to create a product that makes carrying things easier, we must acknowledge the safety risks that come with autonomous movement, such as potential collisions or malfunctions.

Ensuring that this product is safe and ethical is one of our top priorities, as it will likely operate autonomously in public spaces. Ethically, the wagon must be designed to respect other pedestrians, avoid creating hazards, and function in a way that does not infringe on privacy or security. Since our design does not include computer vision, the Illini Wagon's navigation must be highly reliable, using sensors and fail-safe mechanisms to prevent collisions. Safety considerations include redundant braking systems, obstacle detection, and emergency stop features to prevent accidents. We will consult the US Department of Transportation's Automated Vehicles Comprehensive Plan[4] in order to make sure that our vehicle complies with this handbook's standards.

In designing the Illini Wagon, we prioritize both ethical responsibility and safety. While our goal is to provide convenience and accessibility, we recognize the importance of minimizing risks and hazards that our product may pose. By adhering to local regulations, implementing safety features, and ensuring through testing, we plan to create a product that is reliable and responsible. Ultimately, our commitment to ethical engineering will help ensure that the Illini Wagon is safe, practical, and beneficial.

5. Citations

[1] STMicroelectronics, “DUAL FULL-BRIDGE DRIVER,” L298N Data Sheet, [Online]. Available: https://www.st.com/resource/en/datasheet/l298.pdf [Accessed: Feb. 18, 2025].

[2] Qorvo, “Getting Back to Basics with Ultra-Wideband (UWB),” Available: https://www.qorvo.com/resources/d/qorvo-getting-back-to-basics-with-ultra-wideband-uwb-white-paper [Accessed: Jan. 30, 2025]

[3] Qorvo, “DWM100 Datasheet,” Available: https://www.qorvo.com/products/d/da007948 [Accessed Feb. 28, 2025]

[4] U.S. Department of Transportation, “USDOT Automated Vehicles Activities,” US Department of Transportation. Available: https://www.transportation.gov/AV [Accessed Feb. 13, 2025]

[5] Espressif Systems, “ESP32-S3-WROOM-1 WROOM-1U Datasheet,” Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf [Accessed Feb. 10th, 2025]

[6] Qorvo, “APS013 APPLICATION NOTE The implementation of two-way ranging with the DW1000, ” Available: https://www.qorvo.com/products/d/da008448 [Accessed Feb. 10th, 2025]

[7] STMicroelectronics, “DUAL FULL-BRIDGE DRIVER,” L298N Data Sheet, [Online]. Available: https://www.st.com/resource/en/datasheet/l298.pdf [Accessed: Feb. 18, 2025].

[8] Handson Technology, “L298N Dual H-Bridge Motor Driver,” User Guide, [Online]. Available: https://components101.com/sites/default/files/component_datasheet/L298N-Motor-Driver-Datasheet.pdf [Accessed: Feb. 18, 2025]

[9] DFRobot. (n.d.). *Metal DC geared motor w/encoder - 12V 251rpm 18kg.CM Wiki*. DFRobot. Available: https://wiki.dfrobot.com/12V_DC_Motor_251rpm_w_Encoder__SKU__FIT0186_ [Accessed: Feb. 12, 2025]

[10] Texas Instruments, “LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator,” Data Sheet, [Online]. Available: https://www.ti.com/lit/ds/symlink/lm2596.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1741287428008 [Accessed: Mar. 6, 2025]

[11] Diodes Incorporated, “600mA CMOS LDO REGULATOR WITH ENABLE,” AP2112 Data Sheet, [Online]. Available: https://www.mouser.com/datasheet/2/115/AP2112-271550.pdf?srsltid=AfmBOoozxiRdWX9KGtM1-j-q3ANoxcZ6SNJQxErg-MNQbqkQ-12IHqVc [Accessed: Feb. 27, 2025]

[12] Duracell, “MN1500 Size: AA Alkaline-Manganese Dioxide Battery,” Data Sheet, [Online]. Available: https://www.duracell.com/wp-content/uploads/2016/03/MN15USCT0122-A.pdf [Accessed: Mar. 6, 2025]

[13] Qorvo, “DW1000 Datasheet,” Data Sheet, [Online]. Available: https://www.qorvo.com/products/d/da007946 [Accessed: Jan. 28, 2025]

[14] Qorvo, “DW1000 User Manual,” User Manual, [Online]. Available: https://www.qorvo.com/products/d/da007967 [Accessed: Jan. 30, 2025]

[15] Last Minute Engineers, (2022, October 16), “In-depth: Interface L298n DC Motor driver module with Arduino” [Online]. Available: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/ [Accessed: Feb. 10, 2025]

[16] Markerfabs, “ESP32 UWB/UWB Pro,” [Online]. Available: https://wiki.makerfabs.com/ESP32_UWB.html [Accessed: Jan. 30, 2025]

[17] Makerfabs, (2023, October 12), “Makerfabs-ESP32-UWB/hardware,” ESP32 UWB .sch file, [Online]. Available: https://github.com/Makerfabs/Makerfabs-ESP32-UWB/tree/main/hardware [Accessed: Feb 23rd, 2025]

[18] IEEE. ”IEEE Code of Ethics.” (2024), [Online]. Available: https://www.ieee.org/%20about/corporate/governance/p7-8.html [Accessed: Feb. 28th, 2025]