Duplo Mindstorm

Prepared by:

Cody Radabaugh

Sam Stage

Dana Clark

Tommy Trinh

Advisor:

Dr. Jamieson

Fall 2024

College of Engineering and Computing

Department of Electrical and Computer Engineering

ECE 448: Senior Design Project I

Table of Contents

Introduction

Project Goal

The goal of this project is to design and develop a servo and sensor system for a Duplo-based robot, with components specifically scaled and prototyped to integrate seamlessly with Duplo blocks. This project requires a combination of 3D printing, modeling, and hardware integration to ensure compatibility and functionality at the Duplo scale. Our focus will be on integrating key components such as stepper motors, H-Bridge motor drivers, Arduino, IR remote, IR sensor, and power supply. These components will be housed in custom 3D-printed Duplo blocks, allowing for modular and versatile robot designs, similar to Minstorms from Lego. We will document the entire design process, including the evaluation of different design choices, and provide demonstrations of various robot configurations. Duplo scaled blocks are a feasible size of block that allows 3D Printed components and reasonably prices robotic components. The combination of hardware and software approach ensures creative solutions for building endless robotic systems.

Lego Mindstorms

Lego Mindstorms, a robotics kit that has hands-on building with programming. Lego created this line in 1998 in collaboration with the MIT Media Lab to develop a robotics kit that would allow users—particularly children—to program their Lego models. These kits are popular in educational settings as well, forming the basis of school programs and global competitions such as the first Lego League.

The most recent iteration, the Mindstorms Robot Inventor 51515, supports programming languages like Scratch and Python through a companion app, making it accessible for users at various skill levels. Although Lego officially discontinued the Mindstorms line in recent years, its influence remains, as it helped popularize STEM-focused educational products and inspired countless enthusiasts to continue innovating within the Lego community.

Duplo Block

Duplo blocks were developed by Lego for young children and are designed to be double the size of standard Lego bricks in all dimensions. This makes them safer and easier for small hands to handle. For example, a basic 2x2 Duplo block measures approximately 32 mm by 32 mm in width and length, and has a height of around 19.2 mm. Duplo blocks are particularly well-suited for 3D printing and modeling robot components. Their larger size and simplified geometry make them ideal for creating modular structures that can house various electronic components such as sensors, motors, and microcontrollers. The uniform design of Duplo blocks also ensures compatibility and ease of integration, allowing for quick assembly and reconfiguration of different robots.

Project Research & Background

Arduino vs Other MicroControllers

The Arduino Uno proved to be a more suitable choice compared to other microcontroller platforms, such as those based on more powerful chips like the Raspberry Pi. The Arduino Uno, with its ATmega328P microcontroller, is optimized for simple to moderate applications, featuring an 8-bit architecture and a 16 MHz clock speed. Its 14 digital I/O pins, including six PWM outputs, provide the necessary functionality for precise motor control and analog signal simulation. Unlike more complex chips that often run full operating systems, the Arduino Uno excels in real-time control applications where low latency and immediate hardware interaction are critical. This makes it ideal for controlling hardware components such as motors and sensors in our project. Additionally, its straightforward power requirements—operating within a 7-12V input range and featuring a built-in voltage regulator—align perfectly with our battery-powered setup, ensuring stable and efficient performance. The simplicity of the Arduino IDE, with its support for C and C++, allowed for rapid development and debugging, while its compact design and compatibility with sensor shields made hardware integration seamless. More powerful platforms would have introduced unnecessary complexity, increased power consumption, and higher resource requirements for tasks that the Arduino Uno can handle efficiently.

Table 1: Pin Plan

Pin	**Component**
Digital 7 S , V , G	*IR Receiver S, + , -*
Digital 5 S	*H Bridge Motor 2 Enable*
Digital 6 S	*H Bridge Motor 1 Enable*
Digital 7 S	*H Bridge Motor 1 IN1*
Digital 11 S	*H Bridge Motor 1 IN2*
Digital 10 S	*H Bridge Motor 2 IN1*
Digital 9 S	*H Bridge Motor 2 IN2*

Sensors

IR Sensor

An IR (Infrared) sensor is a simple tool for detecting objects, measuring distance, or reading signals from the IR remote. The sensor has two main parts: an IR LED that emits infrared light and a receiver (photodiode or phototransistor) that detects reflected IR light. When an object is close, the infrared light bounces back to the receiver, and the sensor sends a digital signal to the Arduino to indicate its presence. Connecting an IR sensor to the Arduino involves wiring the sensor to a digital pin, 5V, and ground while using simple Arduino code to detect HIGH or LOW signals.

Servo/Driver

Stepper Motor/H-Bridge Motor Driver

The stepper motor is a type of motor that works in steps which allows for fine control of speed, acceleration, and position. Typical stepper motors work with a driver that controls the voltage and direction of the current that dictates the speed and direction that the motor is spinning. This allows for great integration in small circuits like the one we are attempting to create. It is fairly straightforward to integrate Arduino and the stepper motors and drivers to spin the motors in a way that works for the scale of our project.

Power Supply

For our power supply, we used parts off of a previous senior design project. The battery packs supply a 12 V DC value which is needed for our motor drivers. We also borrowed their HW-688 power regulator to allow for easy turn on and off by unplugging the barrel jack of the battery pack. The battery packs are rechargeable which is good for our project because of the heavy testing we have to do.

3D Modeling & Printing

3D Modeling

We used Tinkercad for our modeling program. Tinkercad is a web-based software by Autodesk that enables users to design 3D models. It offers a simple drag-and-drop method for creating designs without requiring prior technical skills. You can find downloadable files completed by some of their users. We were able to download a STL duplo block file directly from Tinkercad. Next we edited the downloaded block by inserting new shapes, to either delete or add new sections onto the block, creating our final models.

3D Printing

Creality 3D printers work by creating objects layer by layer using a heated nozzle that melts filament which is fed through an extruder. The printer's software, Creality3D ,slices a 3D model into thin layers, which the machine then follows to build the object from the base up. The nozzle moves along the X and Y axes to shape each layer, while the bed lowers along the Z-axis as the print progresses. Features like automatic bed leveling and cooling fans help ensure consistent quality. The user uploads designs through a hard drive.

Solution Process

3D Modeling

Given the straightforward nature of our 3D modeling requirements, we determined that Tinkercad was the most suitable software for our project. Its intuitive interface and ease of use allowed us to efficiently design and modify our Duplo-compatible components. Once the designs were finalized, we exported them as STL files. All of the design files are on our Github page [1].

Arduino Enclosure

Figure 1: Arduino Enclosure

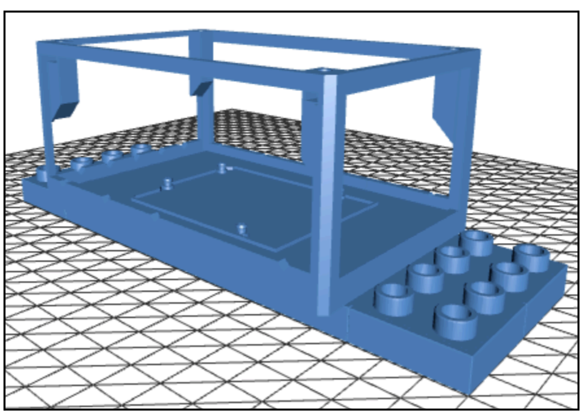


Figure 2: Arduino Enclosure Lid

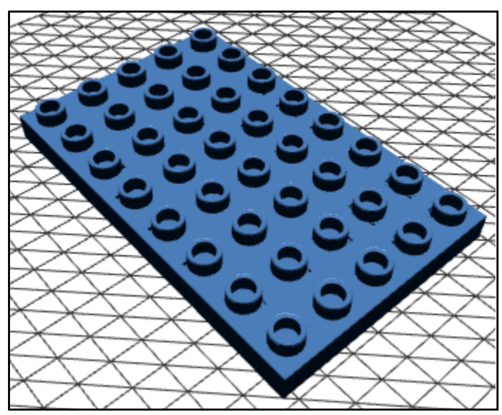


Figure 1 features an enclosure for an Arduino Uno. The design features 2 2x4 duplo pieces on each end of the enclosure. It has pegs in the middle for the arduino to mount on.

Figure 2 displays the arduino enclosure lid. The lid is 5x8 and rests inside the outer framing of the top. The open walls create a lightweight enclosure. The enclosure's base could be expanded to allow for more duplo blocks on the sides.

H Bridge Enclosure

Figure 3: H Bridge Enclosure

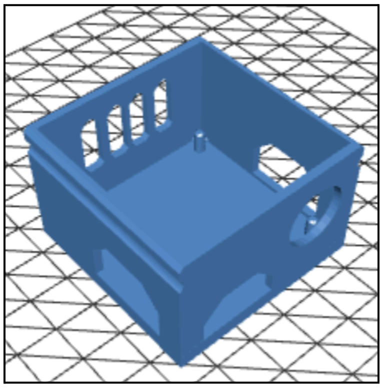


Figure 4: H Bridge Enclosure with Lid

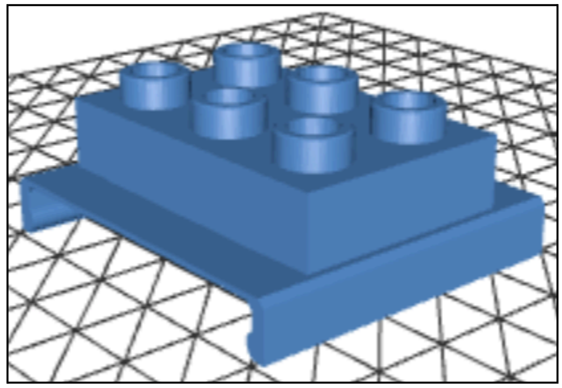


Figure 3 displays the H bridge enclosure (right). The design of the duplo block features holes for the wires to pass through and allow for the cooling of the H Bridge. Figure 4 displays the lid for the H bridge (left). The lid slides onto the H Bridge enclosure with a 2x4 duplo block on the top. This design supports efficient wiring and airflow. However, the holes restrict the wires to only be able to come out of the enclosure from one side and limits freedom.

Battery Enclosure

Figure 5: Battery Enclosure

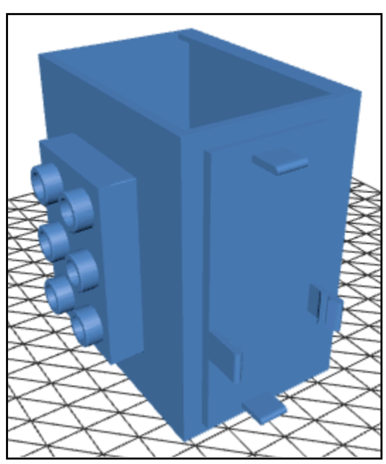


Figure 5 features an enclosure for a 12 V battery pack with a mount for a buck module. The design has a 2x4 duplo block attached to the enclosure with an open end for the battery wires to exit. The side mount holds the buck module in place, allowing the buck module contract with the batteries. The battery pack weight causes the enclosure to struggle staying attached to other duplo blocks. Designing the enclosure to have more duplo pegs can help fix this.

Motor Duplo Mount

Figure 6: Motor Mount

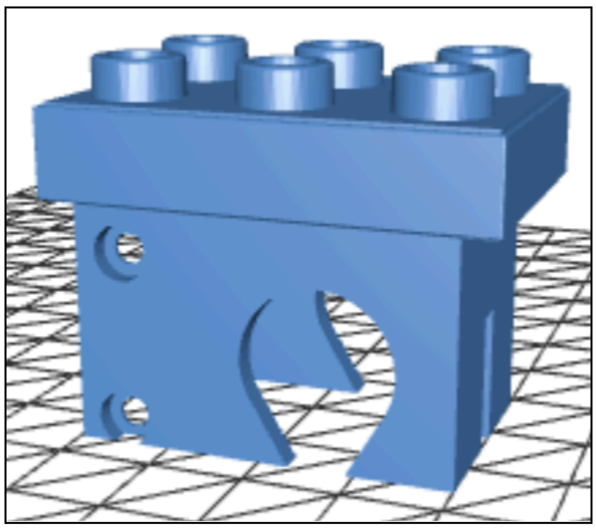


Figure 6 features an enclosure for a Motor. The block has holes for both screw and nuts to secure the motor to the mount. The 2x3 duplo on top makes it easy to connect to the bottom of the arduino enclosure (Figure 1). This model works well with the motor and other duplo blocks.

IR Receiver Duplo Mount

Figure 7: IR Receiver Enclosure

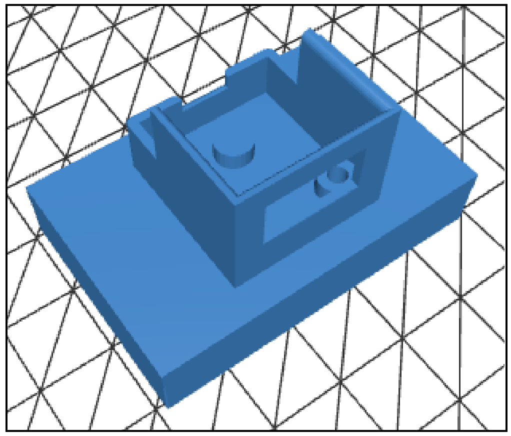


Figure 7 features an enclosure for holding the IR receiver. It is designed with a 2x3 duplo on the bottom to attach to the top of the lid easily. The open top allows for easy connection to the IR remote. This design allows a little movement for the IR receiver which can be improved in the future..

Caster Mount

Figure 8: Caster Mount

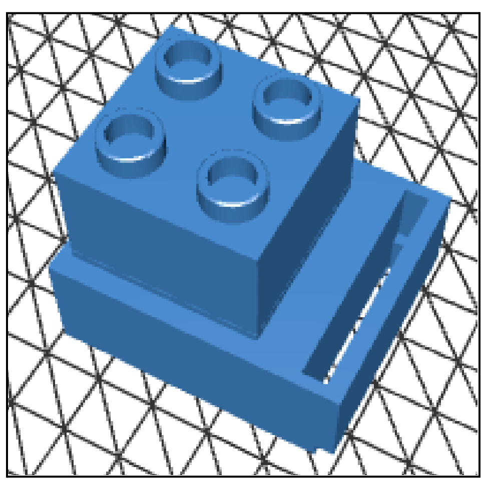


Figure 8 is a mount for the caster wheel. There is a 2x2 duplo block on top of the model. The caster wheel provides for more stability for the robot.

3D Printing

For the printing process, we utilized Creality 3D printers alongside a Bambu P1S printer(provided by Cody). Using Creality's software, we converted the STL files we got from Tinkercad into the file needed to be able to print it out which is called GCODE The GCODE files were then uploaded to a drive to transfer to the 3D printers. We opted for standard black PLA filament, which was readily available to us.

Hardware/Wiring

Figure 9: Arduino Uno Wiring

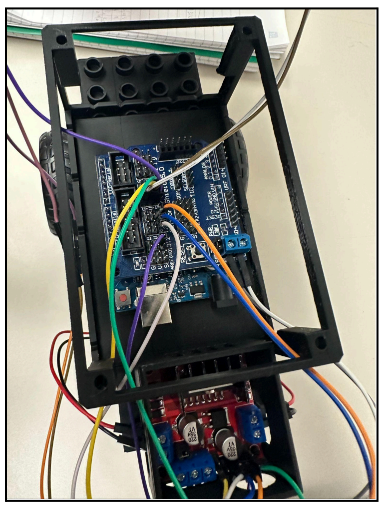


Figure 9 displays the Arduino Uno wiring in the enclosure. The wiring setup consists of multiple wires connected to the input and output pins of the Arduino board, for interfacing with the external components. The pin map can be found in Table 1.

Figure 10: H Bridge Wiring

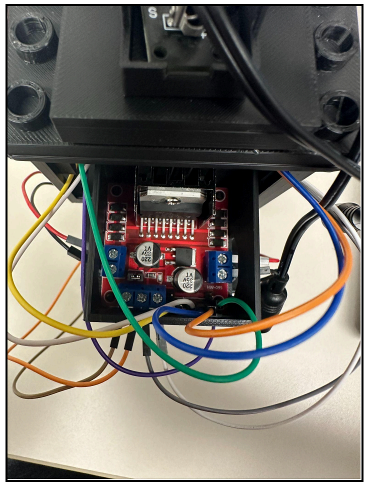


Figure 10 features the wiring for the H Bridge. which is used for controlling the direction and speed of the motors. Various wires are linked to the control signals from the Arduino, allowing it to manage the power flow to a motor in both directions. The H-Bridge allows for bidirectional motor control, enabling the connected motor to rotate forwards or backwards depending on the inputs it receives.

Figure 11: IR Receiver Wiring

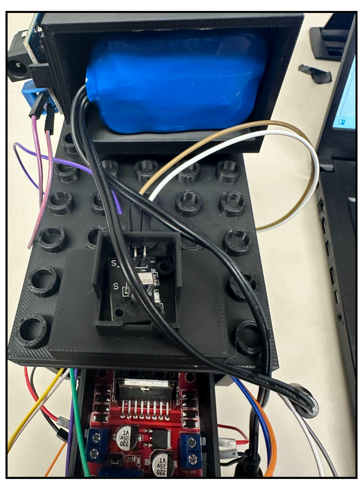


Figure 11 contains an image of the IR receiver wiring. The IR receiver is only connected to the Arduino Uno. This configuration enables the Arduino to receive and process infrared signals.

Figure 12: Buck Module Wiring

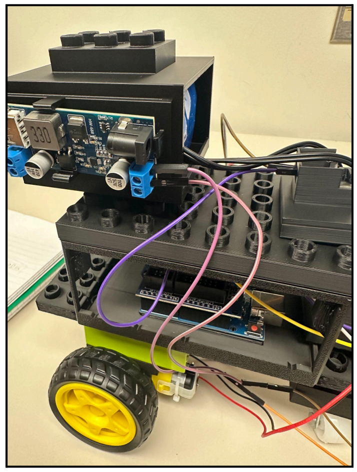


Figure 12 shows the buck module wiring. The buck converter manages the regulated power distribution to ensure the appropriate voltage levels for the connected electronics. The module acts as a switch allowing us to turn the robot on and off by unplugging the barrel jack from the battery pack.

Component Integration

Leveraging our experience from ECE 314, we integrated familiar components efficiently. The Arduino Uno was selected as the main board for its reliable voltage regulation and compatibility with the Arduino IDE. A sensor shield was added to facilitate connections with motor drivers, which control the stepper motors. These motors receive directional signals from the drivers, enabling precise forward and reverse movement. For power, we utilized a 12V battery pack and a power regulator from a previous project. Each component was housed in a custom 3D-printed Duplo block, allowing for modularity and user creativity in assembling different robot configurations.

Code

Figure 9: Demo Code

```
// Cody Radabaugh, Sam Stage, Dana Clark, Tommy Trinh

#include <IRremote.h>

// Defining IR pin and values received from 1-Forward, 2-Backwards, and 3-Stop
// from the IR remote
#define IR_Pin 7
#define FORWARD_COMMAND 24
#define BACKWARD_COMMAND 12
#define STOP_COMMAND 94

// Initializing pins for motor driver
const int RSPD = 120;
const int LSPD = 120;
const int RWhPWMPin = 6;
const int RWhFwdPin = 11;
const int RWhBwdPin = 12;
const int LWhFwdPin = 9;
const int LWhBwdPin = 10;
const int LWhPWMPin = 5;
bool stop = true;

void setup() {
  // Configure motor driver pins as outputs
  pinMode(LWhFwdPin, OUTPUT);
  pinMode(LWhBwdPin, OUTPUT);
  pinMode(LWhPWMPin, OUTPUT);
  pinMode(RWhFwdPin, OUTPUT);
  pinMode(RWhBwdPin, OUTPUT);
  pinMode(RWhPWMPin, OUTPUT);

  // Initialize IR receiver
  IrReceiver.begin(IR_Pin, ENABLE_LED_FEEDBACK);

  // Initialize motor driver pins to low/off state
  digitalWrite(LWhFwdPin, LOW);
  digitalWrite(LWhBwdPin, LOW);
  digitalWrite(LWhPWMPin, 0);
  digitalWrite(RWhFwdPin, LOW);
  digitalWrite(RWhBwdPin, LOW);
  digitalWrite(RWhPWMPin, 0);
}

void loop() {
  // Small delay for stability
  delay(100);

  // If the last command was not stop, continue at the same speed
```

```
  if (!stop) {
    analogWrite(RWhPWMPin, RSPD);
    analogWrite(LWhPWMPin, LSPD);
  }

  // Check if a new IR command is available
  if (IrReceiver.decode()) {
    int current = IrReceiver.decodedIRData.command;

    if (current == FORWARD_COMMAND) {
      // Move forward: Set forward pins to HIGH, backward pins to LOW
      digitalWrite(LWhFwdPin, HIGH);
      digitalWrite(RWhFwdPin, HIGH);
      digitalWrite(LWhBwdPin, LOW);
      digitalWrite(RWhBwdPin, LOW);
      stop = false;
    } else if (current == BACKWARD_COMMAND) {
      // Move backward: Set backward pins to HIGH, forward pins to LOW
      digitalWrite(LWhFwdPin, LOW);
      digitalWrite(RWhFwdPin, LOW);
      digitalWrite(LWhBwdPin, HIGH);
      digitalWrite(RWhBwdPin, HIGH);
      stop = false;
    } else if (current == STOP_COMMAND) {
      // Stop motors: Set speed to 0 and update stop flag
      analogWrite(RWhPWMPin, 0);
      analogWrite(LWhPWMPin, 0);
      stop = true;
    }

    // Resume IR receiver to process the next command
    IrReceiver.resume();
  }
}
```

Figure 9 is our code we used for the demo video [2]. This Arduino program controls a two-wheeled robot using an infrared (IR) remote. It employs the IRremote library to interpret signals received from an IR sensor connected to pin 7 on the Arduino. The robot can execute three commands based on signals from the remote: moving forward, moving backward, or stopping. The motors are controlled through a motor driver module, with specific pins designated for setting motor speed (via PWM) and direction (forward or backward). Default motor speeds are set to a PWM value of 120 for both wheels. The program maintains a "stop" flag to track whether the robot is stationary or in motion. During the setup phase, the motor driver pins are configured as outputs, and the IR receiver is initialized to listen for commands. Initially, all motors are turned off. In the loop function, the program continuously monitors for IR signals. If no new command is received, the robot maintains its current speed and direction. When a new command is detected, the robot reacts by adjusting the motor control pins: enabling forward movement, backward movement, or stopping the motors entirely by setting their speed to zero. After each command is processed, the IR receiver is reset to await further input. This setup enables the robot to be remotely controlled, allowing for basic navigation and movement.

Final Results

For the fall 2024 semester we have completed project research, 3D printing specialized duplo blocks, powering and wiring the robot, writing a simple piece of code for maneuverability, assembly of the duplo blocks in various positions, and documentation. A sample robot demo can be found here showcasing a possible robot configuration that runs the code shown in figure 9.

Figure 10: Full robot

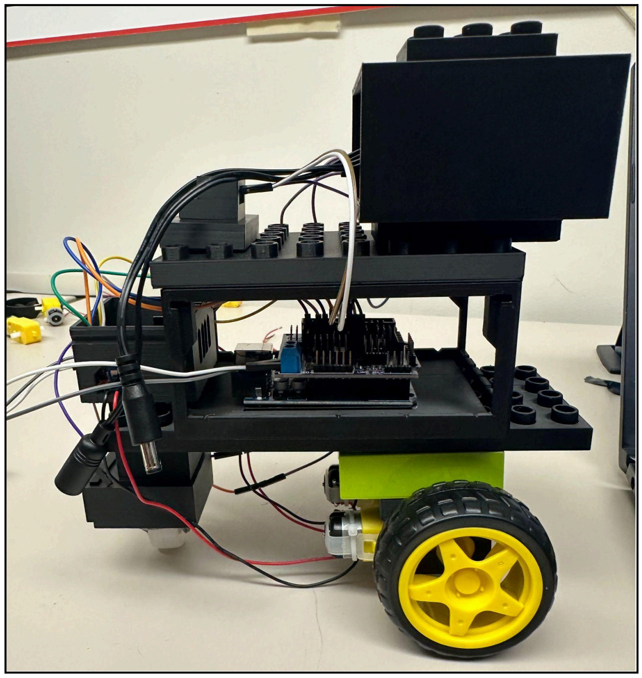


Figure 10 contains a picture of our full robot. The video of the robot [2] features the basic maneuverability of the robot. The design in figure 10 works well with weight distribution. We placed the battery pack directly above the wheels to keep the robot balanced. Everything is placed in wiring distance of the Arduino Uno. We can work on discrete wiring, weight reduction, and adding more components to enhance the robot in the future.

Future Work

With the foundational goals for the first semester successfully achieved, our focus now shifts to expanding the capabilities of the Duplo-based robot. Future work will involve:

Adding More Sensors:

- Touch Sensors: These sensors will provide tactile feedback, enabling the robot to respond to physical interactions, such as stopping when an obstacle is touched.
- Headlights: LED modules integrated into Duplo blocks to improve visibility in low-light environments and signal different states.
- Color Detectors: RGB sensors can enhance the robot's ability to differentiate between colors, opening possibilities for tasks like color-coded sorting or following colored paths.

Enhancing Code:

- Implementing more sophisticated algorithms to handle complex tasks such as obstacle avoidance, autonomous navigation, and task-specific behaviors.
- Developing modular code libraries for easy integration of new sensors and components.

Proof of Concept

Figure 11: Alternate Design Front View

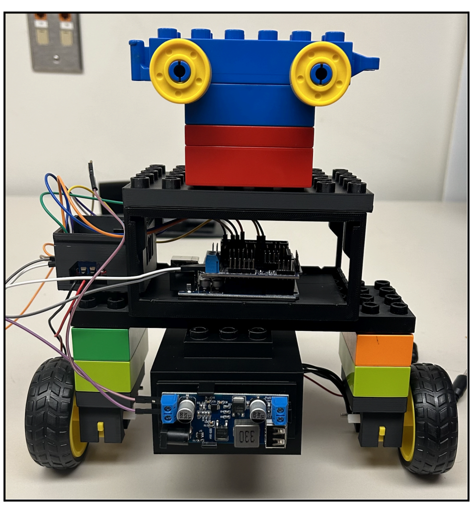
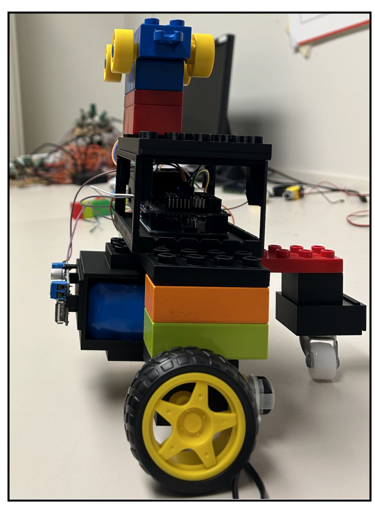


Figure 12: Alternate Design Side View



In figures 11 and 12 we developed a different robot with lego duplo blocks. The design resembles a crab-like toy to appeal to younger generations. We can work on fixing the wiring to be more discrete and easy to move.

Conclusion

While working though our first semester of senior design, we have gone through the main stages of a design process. Starting with an idea, we had to first create goals and a vision of what we wanted to accomplish. Our main goal was to prototype a Duplo version of the Lego Mindstorm including individual components that can be configured however the user wants. After the initial planning stage, we had to research methods of 3D printing and components that would be easily integrated into a circuit. From there we went straight into the design which included multiple designs that were made specifically for each component that we wanted to include with the final product. We then printed these components with Cody’s 3D printer(Bambu P1S). After each component was created, we went into the testing/building stage which included integrating our 3D printed parts with our electrical components. This led us to our final stage of analysis where we fully documented all of our work into a final paper. The design process is a lengthy one, but with this knowledge we can go into the work field with a newfound experience under our belts.

Reflection

This semester's senior design project deepened our understanding of professional and ethical responsibilities as engineers. We kept creativity and freedom at the heart of our project because that's what we believe our project was all about. Building a tool to help children learn the basics of robotics has been an incredibly rewarding experience. We discovered that designing for children can be just as enjoyable and engaging for college students, highlighting the universal appeal of educational tools that are both fun and enriching. Providing educational tools that are fun for all ages is such an important part of society. Starting with parts and pieces that can be put together to do a certain task is cognitively stimulating but also fosters curiosity and problem-solving skills—particularly vital for young minds. As we continue to work on this project through the next semester, we want to emphasize the importance of solving small engineering problems that can carry over to the real world when given small pieces that need to be put together to create something whole and functional.

We recognize the societal impact of our work. Designing this system taught us to think beyond technical details and consider how engineering solutions can influence education and innovation among all age groups. Engineers play a critical role in shaping the world, and our work should be thoughtful and inclusive. In summary, this project provided valuable lessons in ethics, lifelong learning, and societal impact, preparing us to enter the professional world with both technical expertise and a commitment to making a positive difference with the choices we make as engineers.

Works Cited

[1]

"Danaeclark/DuploMindstorm: ECE 448/449 Senior Capstone Project." *GitHub*, github.com/danaeclark/DuploMindstorm. Accessed 4 Dec. 2024.

[2]

"ECE 448: Duplo Mindstorm, Fall 2024 Final Result." *YouTube*, YouTube, www.youtube.com/watch?v=XVqvp0UVz1A. Accessed 4 Dec. 2024.

[3]

"Doblo Factory." *Doblo Factory - EduTech Wiki*, edutechwiki.unige.ch/en/Doblo_factory. Accessed 12 Nov. 2024.

[4]

"Lego Duplo Brick Collection with Parametric Source Files." *Lego Duplo Brick Collection with Parametric Source Files by András Bognár | Download Free STL Model | Printables.Com*, www.printables.com/model/340295-lego-duplo-brick-collection-with-parametric-source. Accessed 12 Nov. 2024.

[5]

"Lego Duplo Compatible Building Block 2x2." *Lego Duplo Compatible Building Block 2x2 by TmnSn | Download Free STL Model | Printables.Com*, www.printables.com/model/137952-lego-duplo-compatible-building-block-2x2. Accessed 12 Nov. 2024.

[6]

"Search for 3D Designs and Circuits." *Tinkercad*, www.tinkercad.com/search?q=Duplo+block&staffPicks=0. Accessed 12 Nov. 2024.

[7]

"Subject & Course Guides: Making from Home Resources: 3D Modeling." *3D Modeling - Making From Home Resources - Subject & Course Guides at Miami University*, libguides.lib.miamioh.edu/make-from-home/tutorials-3D-modeling. Accessed 12 Nov. 2024.