**COMPSYS 301 Report**

Isaac Lee, Oliver Wright, Brian Wei, Ammaar Marzook

*ESCE Department*

*University of Auckland*

**Digital Design**

In this project, our primary focus was the development of our digital design utilizing the PSoC 5 development kit (Kit 059), C programming language, and the PSoC Creator IDE. Our digital design integrated various data resources, such as light sensor circuits, quadrature decoders, and PCB designs, to enhance our robot’s precision navigation on the designated map.

In our digital design approach, one of the objectives was to maximize resource efficiency while ensuring flexibility and performance. During our sensor firmware design, particularly light sensor circuit, we focused on achieving a circuit which offers the flexibility to measure light intensity as needed, demonstrating our adaptable approach.

Additionally, it was important to ensure efficiency in navigation algorithms of our robot, minimizing distance between target locations, while also aiming to optimize food pellet collection. This section delves into the essential findings and design considerations that drove these solutions.

**Design Considerations and Decisions**

1. **Path Finding Algorithm**

When determining the best algorithm to reach our target location efficiently, our group evaluated two options: Breadth-First Search (BFS) and A\* (A-star). Both algorithms had the potential to find the fastest path, but upon closer examination, it became evident that BFS had certain drawbacks, particularly in terms of memory efficiency and processing time.

In contrast, A\* emerged as the more promising choice due to its memory efficiency, allowing it to process the necessary data more quickly for our project. Additionally, A\* held features such as possessing a heuristic-driven approach, which proved advantageous for navigating toward our project’s goals. A\* excelled in traversing the graph efficiently by employing heuristics to estimate the distance of each node from the destination. This heuristic-guided approach ensured that the algorithm consistently made progress towards the destination.

One of A\*’s advantages was its adaptability in handling weighted directed graphs, setting it apart from BFS. This flexibility proved advantageous when dealing with scenarios where the distances between nodes were not uniform, a common occurrence in projected graphs. This attribute allowed us to account for variations in node distances, especially when node dimensions varied in both width and height on projected graphs. A\*’s adaptability in the face of these variations consistently enabled us to select the shortest and most efficient route for our robot’s navigation.

1. **Motor Control and Line tracing**

In our pursuit of precise autonomous line tracing, we focused on optimizing our motor control and line tracing systems. The use of quadrature decoder allowed us to monitor wheel speed and overall distance traveled, ensuring uniformity during straight-line movement to minimize turning error, as any inconsistency between the two wheels would lead to turning errors.

Our motor control system utilizes a PI (Proportional-Integral) controller to regulate the robot's speed. This controller operates with a configurable interrupt, set at approximately 50 milliseconds, allowing us to calculate speed regularly. We considered an alternative approach where we would wait for a specific number of wheel counts to accumulate and then measure the time it took. However, this method had a drawback – it updated speed more slowly during low-speed operations, making the PI controller less responsive. Despite this disadvantage, our chosen approach had a distinct advantage. It ensured that speed calculations consistently consumed a fixed amount of time, providing stability to our PI controller on a reliable timer. This approach involves a trade-off, sacrificing some accuracy at lower speeds, but we found it acceptable for our application.

When exploring methods for accurate map traversal, we examined various approaches, such as drift tracing and employing different operational states. However, these methods had limitations, especially concerning overshooting issues when our sensors deviated from the intended path. This problem stemmed from the placement of the alignment sensors at the robot's center. To address the re-alignment overshooting challenge, we experimented with temporarily boosting the motors after exiting the drift correction state. Regrettably, this solution did not achieve the desired results either.

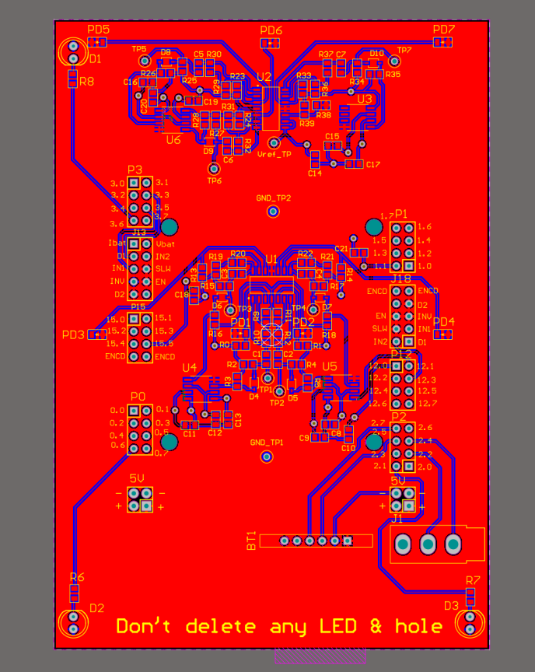
To effectively address these issues, we adopted an error approximation approach coupled with the implementation of a PID controller. This shift proved importance in achieving significantly improved results, aligning with the PID controller's core purpose of achieving precise and stable line tracing.

1. **PCB Design**

For the development of our printed circuit board (PCB), we utilized Altium Designer.

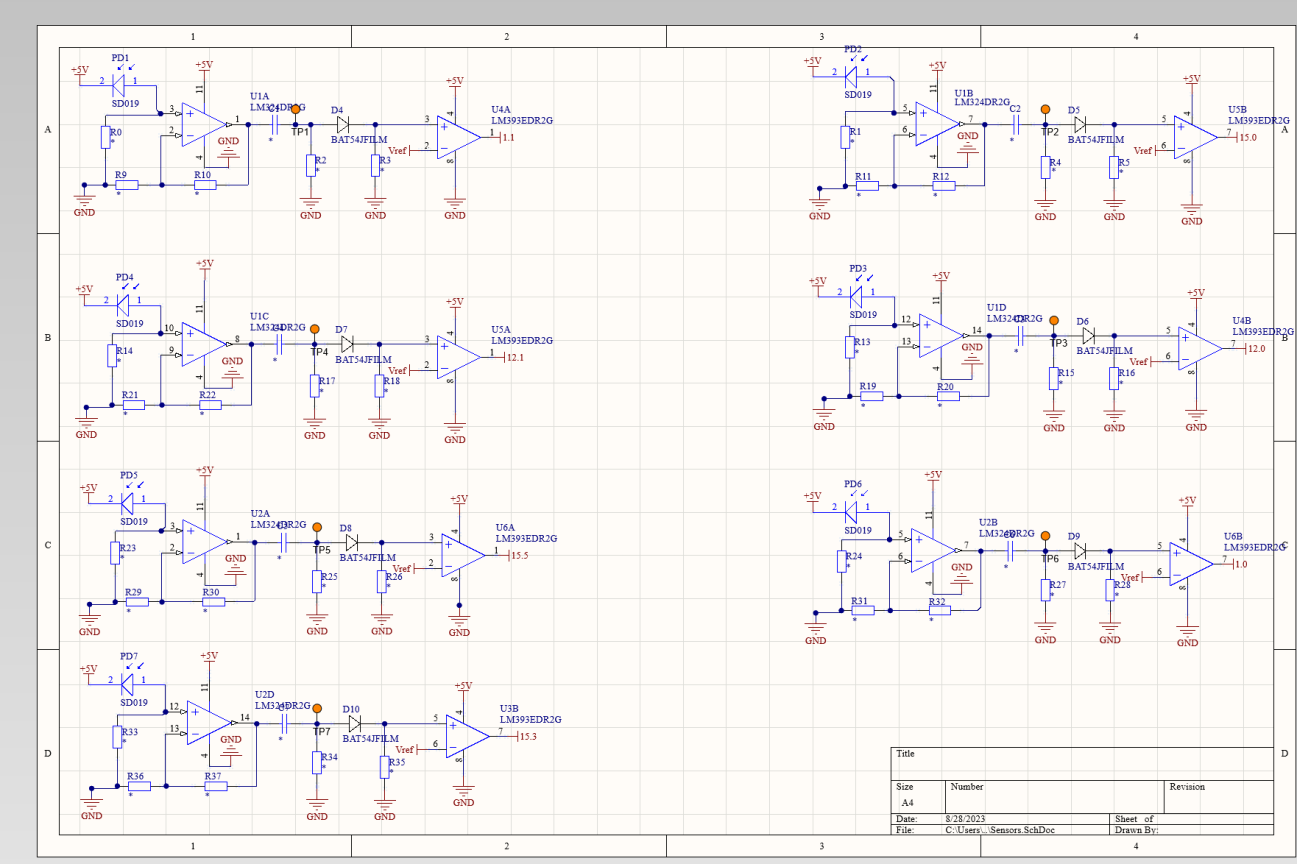
Our robot is equipped with seven photo diodes, each with a specific role. Three front diodes handle initial line detection, as well as the two on the sides aid in turning adjustments to prevent overturning. Among the four middle diodes, two at the centre ensure precise line tracking, while the utmost diodes on both sides of middle diodes server as reference points for lateral adjustments.

We also incorporated LEDs for debugging purposes. These LEDs provide real-time visual feedback, allowing us to monitor the robot’s sensor readings and responses, making troubleshooting and fine-tuning our algorithms more efficient during testing and development.



Figure

1. **Sensor Circuit Design**



Figure

In this project section, we aimed to determine the sensor circuit’s cut-off frequency, primarily for capturing the projector’s 120Hz light signal accurately. Our initial design features a phototransistor, non-inverting amplifier, low-pass filter, and Schmitt trigger. Extensive testing was necessary to ensure its functionality.

The initial change was from realizing that focusing on extracting and processing the processing the projector’s signal, rather than filtering it entirely, would yield better results. This led us to shift from a low-pass filter to a high-pass filter.

Furthermore, we encountered issues with the phototransistor’s inconsistency. Despite the smooth operation initially, it later exhibited erratic behaviour. Recognizing the importance of stability and reliability in our sensor circuit, we explored alternative components and configurations.

Consequently, we opted for a shift from the Schmitt trigger to a comparator for level sensing. The change was due to the realization that the comparator provided a more efficient and time-saving implementation compared to the Schmitt trigger, effectively distinguishing between logic how and low states.

During this process of refinement, we selected the resistor and capacitor values to fine-tune our sensor circuit’s performance. We employed resistors R0 (6.8M Ohms), R9 (2.7M Ohms), R10 (1.5k Ohms), R2 (33k Ohms), and R3 (820k Ohms), with a capacitor C4 (47nF).

**Verification and Testing**

**Conclusion**

In this project, we successfully designed a robot capable of playing a Pacman-like game, collecting food pellets while efficiently navigating a maze. Our work involved creating an analogue circuit that interprets photodiode signals, processed by a PSoC microcontroller. We validated our circuit design through initial LTSpice simulations and then built the final printed circuit board (PCB) using Altium Designer, featuring surface-mount technology (SMT) components.

We also integrated an A\* pathfinding algorithm into our system, enabling the robot to navigate the maze efficiently while optimizing its path for food pellet collection.

Throughout this project, our team acquired valuable skills in PSoC, communication, and other various other aspects essential to engineering and project development, enhancing our capabilities for future endeavours.