PANCAKE FLIPPER
ECE 445 FINAL REPORT - SPRING 2024

Team 54
David Lin, Jason Kim, James Lu

Professor: Arne Fliflet
TA: Abhisheka Mathur Sekar

May 2024

Abstract

This report details the development of a robotic pancake flipper, designed as part of the ECE 445 Senior Design course, aimed at automating the pancake-making process to ensure uniformity and quality in pancake preparation. The project integrates mechanical, electrical, and software components into a cohesive system capable of detecting the optimal flipping time through a camera module and executing the flip with precision using servo motors and a linear actuator.

The robotic system consists of an integrated electric griddle, a camera that monitors the cooking process to signal readiness based on specific visual cues, and a flipping mechanism controlled by a microcontroller. This setup not only enhances the efficiency of pancake making but also significantly reduces common problems such as tearing, uneven cooking, and burning, typical of manual pancake flipping.

Through rigorous testing, the device demonstrated a flipping success rate exceeding 95%, with the bubble detection algorithm achieving 90% accuracy in determining the right flipping moment. These capabilities highlight the project's success in marrying technology with everyday cooking tasks, presenting a significant step forward in kitchen automation.

This report elaborates on the design process, subsystem functionalities, comprehensive testing procedures, and the overall outcomes of the project. It also discusses future enhancements that could improve the device's functionality and market viability, such as cost reduction, miniaturization, and integration with smart home systems. The development of the robotic pancake flipper not only meets the course's requirements but also sets a foundation for future innovations in automated home cooking solutions.

Contents

1. Introduction

Making pancakes at home can be frustrating for even the most patient chefs due to potential problems. Pancake flipping, in particular, is prone to issues such as tearing, folding, and burning. Achieving the perfect golden-brown outside without overcooking is a delicate balance. Deformed pancakes can also detract from their appealing texture. A more flexible solution is needed, emphasizing the necessity for pancake-making equipment that accommodates various tastes, ensuring a fun and easy cooking experience.

The proposed solution is a robotic pancake flipper equipped with a spatula, designed to automate the flipping process and ensure perfect pancakes every time. This device integrates an electric griddle and a mechanical system comprising a linear drive, a linear actuator, and servos. The electric griddle cooks the pancake, which is then moved towards the spatula by a linear actuator. The spatula positions itself beneath the pancake. One servo lifts the spatula and the other flips it, flipping the pancake to cook evenly on both sides. This automated system aims to eliminate common pancake-making issues by precisely controlling the cooking and flipping process, ensuring consistent and satisfactory results.

For our project to be considered functional, it has to satisfy three high level requirements. First of all, the robotic spatula must flip pancakes with a precision that ensures minimal tearing of the pancake and any tears must be kept under 0.5 inches. Secondly, our AI classifier should be able to detect when the pancake is ready to flip. Lastly, the display unit should display the right messages at the right time to ensure the user knows the current status and what to do with our pancake flipping machine.

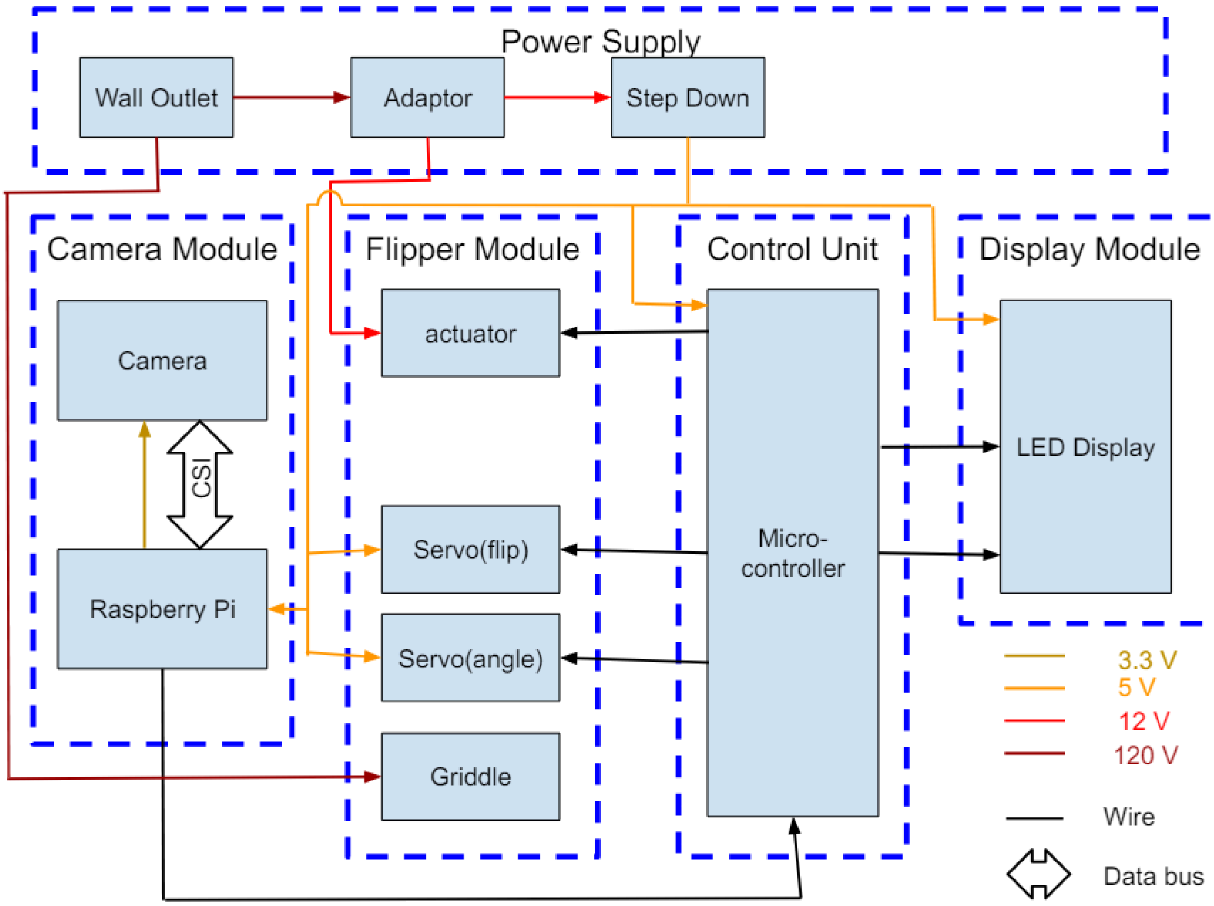

Figure 1: Block Diagram

As our block diagram shows, our project was divided into 5 different modules: Power supply, camera module, flipper module, control unit, and display module. The power supply unit, which is made up of an adaptor that plugs into a wall and a step down regulator, provides power to other other units. It provides 5V to every item except the griddle and the linear actuator. The griddle receives power from the wall, and the linear actuator receives 12 V. The camera module consists of a camera and raspberry pi The camera sends image data to the raspberry pi, which processes it using an AI binary classifier. The raspberry pi sends a 5V signal to the microcontroller within the control unit when it detects a cooked pancake. Otherwise the signal is kept at 0V. The control unit consists of the ATMega328P microcontroller, which sends signals to the servos and linear actuator within the flipper module to execute successful flips. The microcontroller also sends signals to the display unit to display the appropriate messages.

One main change throughout this project is our camera module. We realized counting bubbles wasn't really a good indicator of how well the pancake was cooked because there are many external factors such as batter type. Instead of using computer vision to identify holes, we recorded the pancake cooking and labeled each frame as cooked or uncooked. We then trained a binary classifier on the image data.

2. Design

2.1 Design Procedures

A couple design decisions were made in the Flipper module.
Torque was studied to ensure that the flipping motion was realized.

Variable	Description	Value
m_p	Mass of pancake	≤ 0.12 kg
r_p	Radius of pancake	≤ 0.10 m
m_s	Mass of spatula	≤ 0.11 kg
l_s	Length of spatula	≤ 0.30 m
m_f	Mass of flipping servo	≤ 0.12 kg
g	Acceleration of gravity	9.81 m/s^2

Table 2.1: List of variables for torque calculation

Assuming the worst case where all the weight is distributed the furthest away from the axis of rotation of the angle servo ($l_s + r_p$), the torque needed to perform the lift can be calculated using the following formula.

$$\begin{aligned}\tau &= (m_p + m_s + m_f)*g*(l_s + r_p)\\ &= (0.12 + 0.11 + 0.12)*9.81*(0.30+0.10)\\ &= 1.37\ \text{Nm}\end{aligned}$$

One alternative design for the flipper module was to cook a pancake on a metal plate that would be heated by the griddle. The metal plate would move on an axis like a page in a book to flip the pancake. This idea was initially appealing as one of the major challenges in realizing a pancake flip was to be able to get underneath a pancake without human guidance. However, this idea would require more torque as it would have to lift the metal plate as well. The torque calculations are as below.

$$\begin{aligned}]\tau &= (m_p + m_{plate})*g*(l_{plate})\\ &= (0.12 + 1)*9.81*(0.45)\\ &= 4.94\ \text{Nm}\end{aligned}$$

After calculating the torque required for the book idea, we thought it would be more feasible to proceed with the initial idea with the linear actuator. Given that the linear actuator has a drive force of 300 N, we felt confident that we were going to be able to get underneath the pancake. Thus the design decision was made to proceed with this idea.

From the beginning we knew we needed an H-Bridge to control the direction of the linear actuator. We initially attempted to implement our own custom H-Bridge. When testing our custom H-Bridge made from four mosfet transistors, the linear actuator did not move as fast as it should, indicating that there was not enough voltage going through the two terminals of the linear actuator. This was due to the lack of dissipation in the gates of the transistors. After unsuccessful attempts to add resistors to force dissipation, we decided to move on with a commercial H-bridge which worked flawlessly.

For the control unit, we could have chosen among many microcontrollers including the ESP32 or ATmega48A. We decided to move forward with the ATMega328p microcontroller as it was the similest to work with and met the requirement of 14 i/o pins.

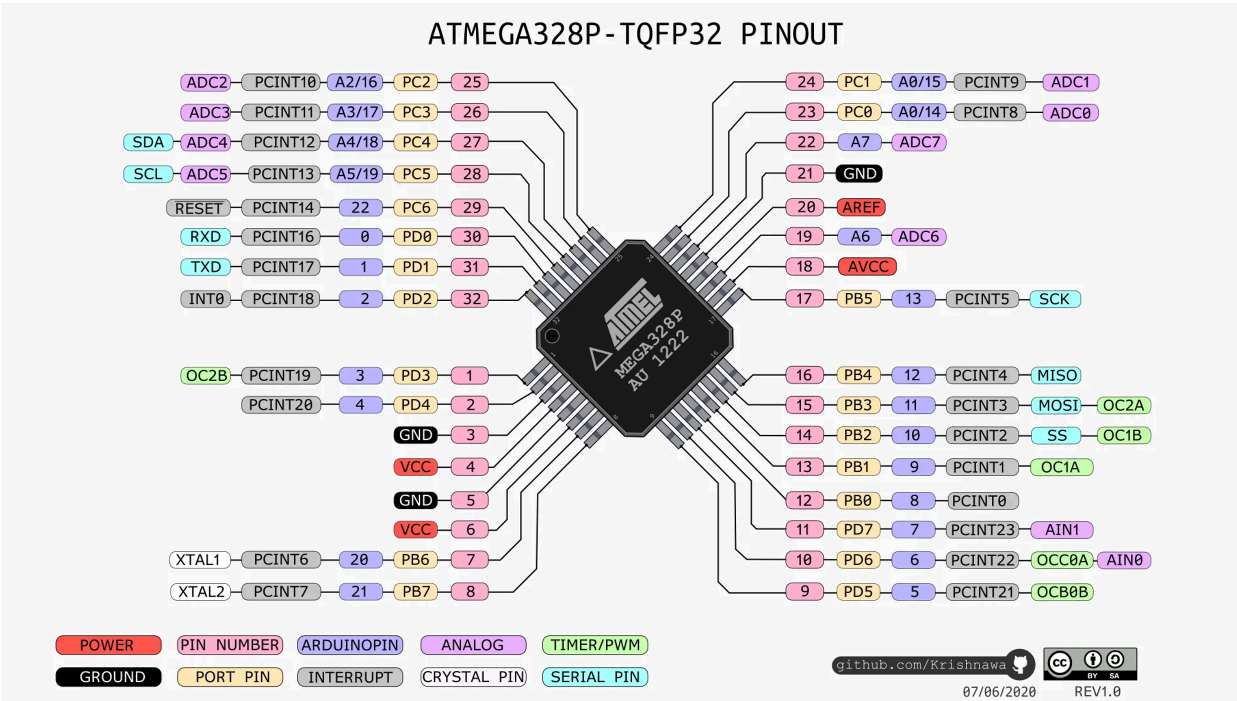

Figure 2.1.1: ATMEGA328P Pinout

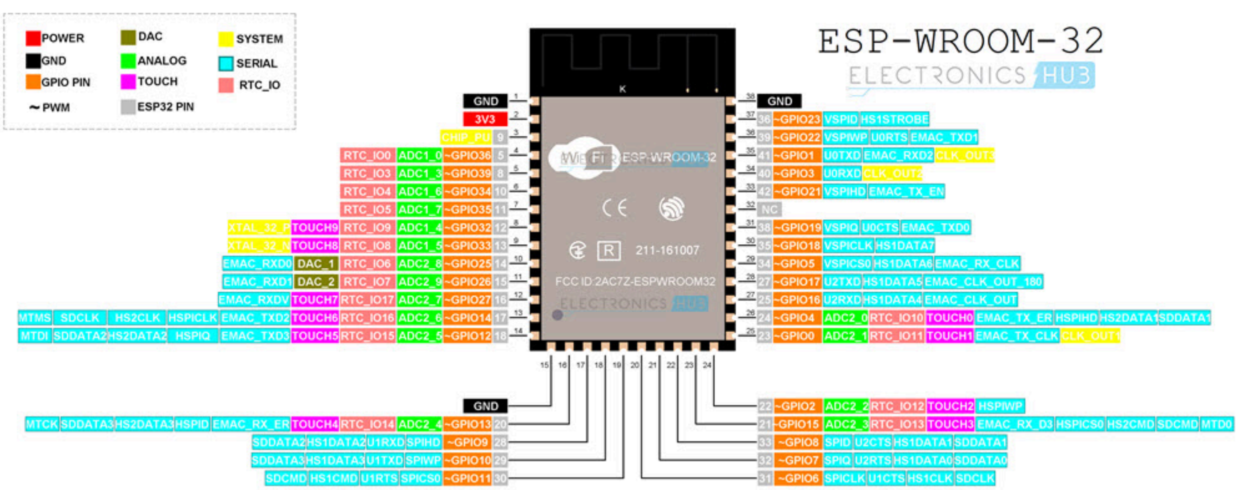

Figure 2.1.2: ESP32 Pinout

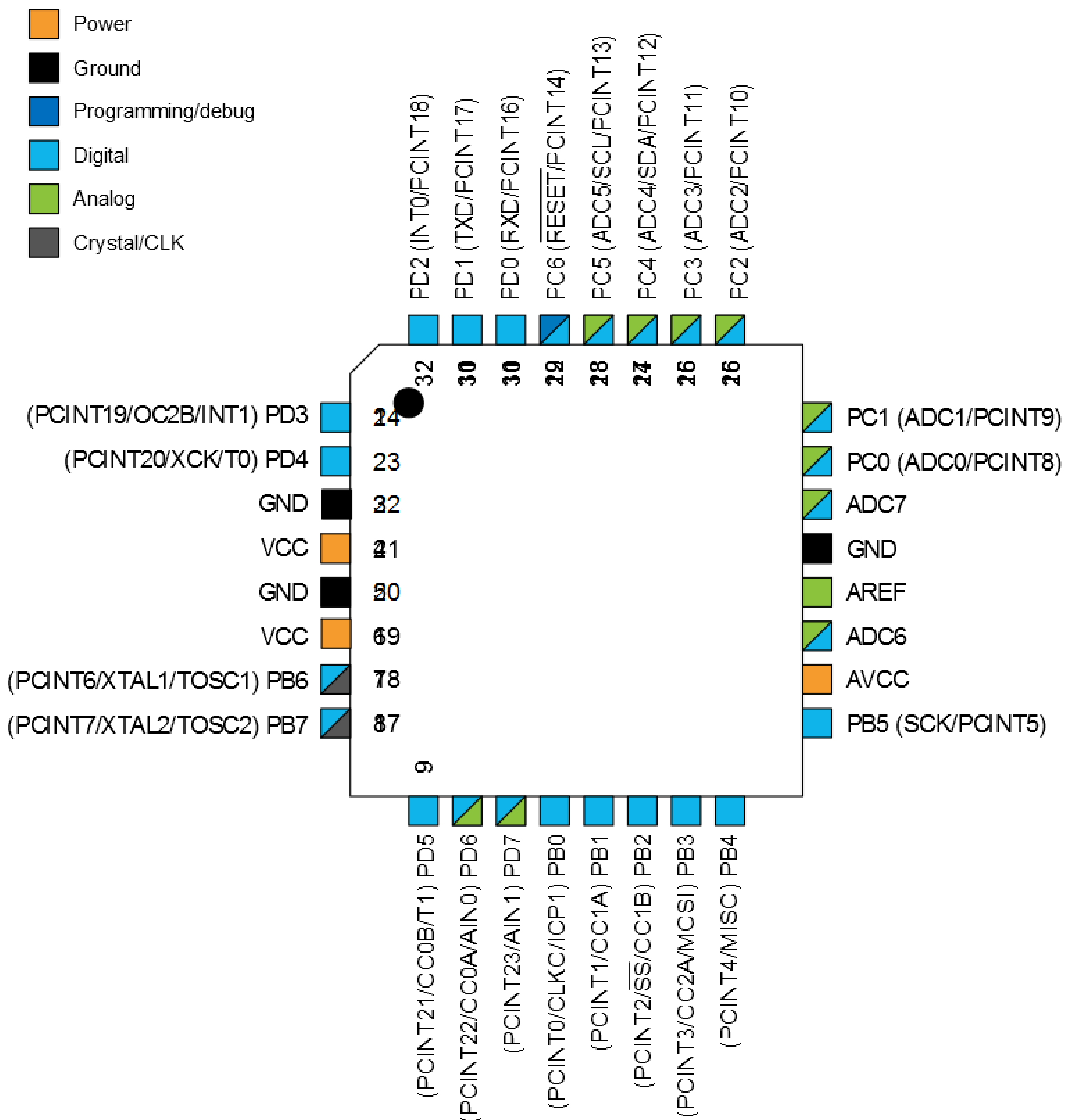

Figure 2.1.3: ATMEGA48A Pinout

In addition to the H-bridge, we came across another issue. We could not program the ATMega328 using the RX and TX pins in conjunction with the FTDI module. This is because the ATMega328P must be set to a baud rate that matches the baud rate set by your FTDI module. The standard baud rate for programming is typically 57600 or 115200 bps. If there is a mismatch in baud rate settings between the ATMega328P and the FTDI module, communication errors can occur, preventing successful programming. We were not able to resolve this issue in time, so we proceeded with the Arduino dev board.

For the display unit, we chose to use the I2C LCD Display. We could have chosen SPI interface or DAC, however, the I2C seemed the simplest and user friendly, so it was used.

The power supply unit needed to output 12V and 5V as many parts used 5V and the linear actuator used 12V. To provide the voltage needs of all the components, we decided to use a 12V adaptor and a 12V to 5V step down regulator. An alternative to using a commercial regulator was implementing the step down voltage ourselves on the PCB, which could have resulted in saving power, however, this could have led to more complicated debugging issues.

For the Camera Module, we used the Raspberry Pi zero 2 W and a Pi camera. We did not use the Raspberry Pi zero because it did not support the tensors required to run the AI classifier. The Pi camera was chosen afterwards for compatibility.

2.2 Design Details

2.2.1 Power Supply Unit

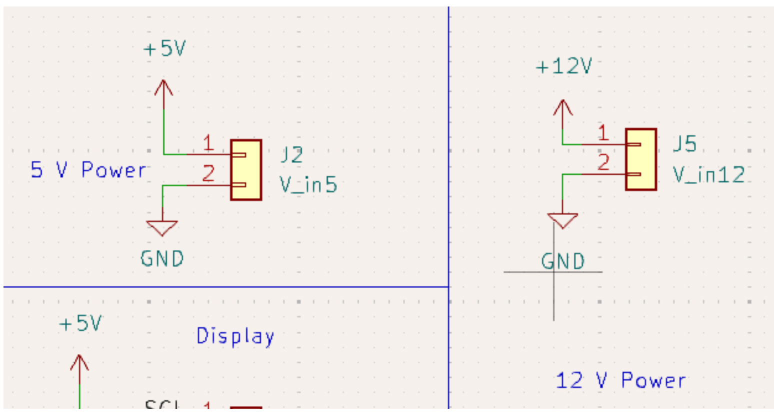

Figure 2.2.1: Power Supply Unit Schematics

To meet the requirements of our project, we needed a power supply unit capable of outputting both 12V and 5V. This was necessary because several components of our setup operated at 5V, while the linear actuator required a 12V input. To accommodate these voltage requirements efficiently, we chose to use a 12V adapter as the primary power source. Alongside this, we incorporated a 12V to 5V step-down regulator. This setup ensured that we could efficiently convert part of the 12V output from the adapter down to 5V, thereby providing the appropriate power levels to all components without the need for multiple power supplies. This approach not only simplified our power management strategy but also enhanced the overall reliability and safety of the system.

2.2.2 Camera Module

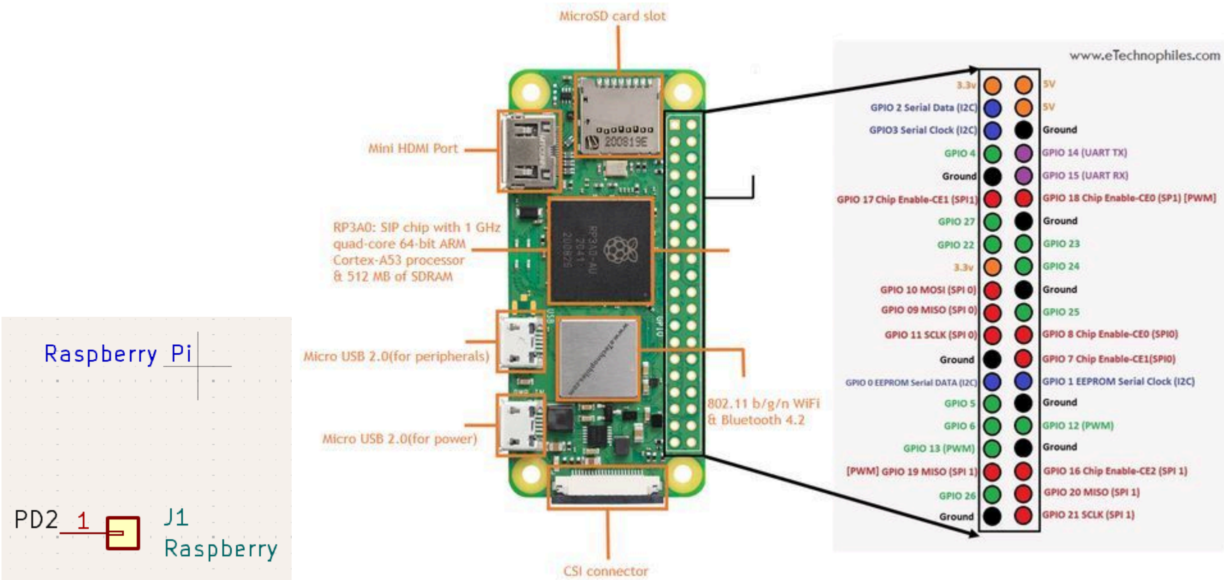

Figure 2.2.2: Raspberry Pi Schematic and Pinout

The raspberry pi sends a single signal from GPIO 18 to the Atmega328P pin32. The raspberry pi uses CSI protocol to communicate with the camera. We use adagrad to train weights for the classifier. The pi is powered with 5V.

2.2.3 Control Unit

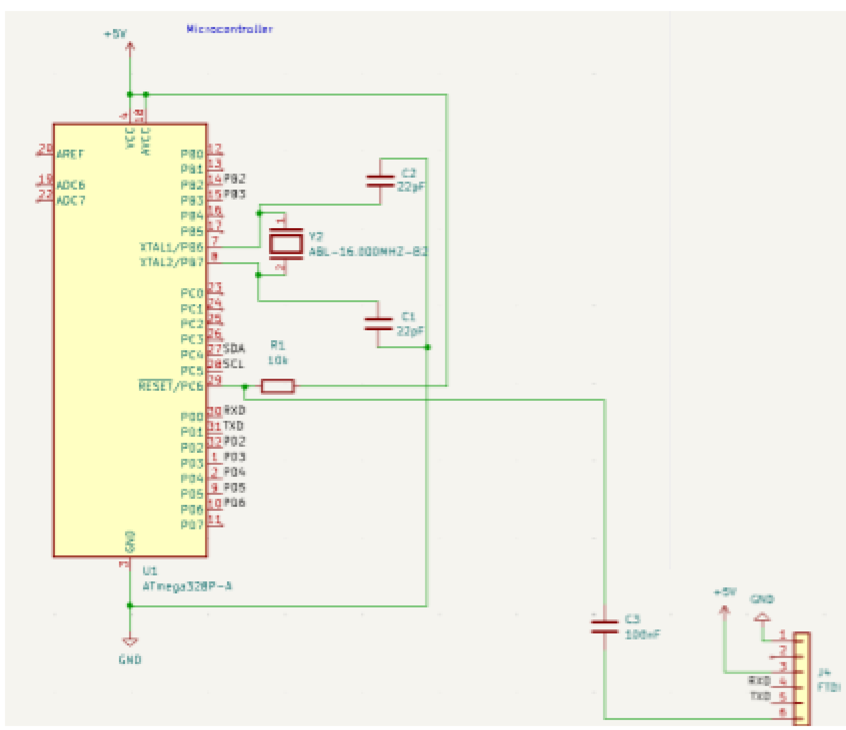

Figure 2.2.3.1: Control Unit Schematic

The microcontroller is an ATMega328P. In addition to the signals received from the raspberry pi, the digital signals it sends to the flipper module, and analog signals used to control the LCD display, there is some additional circuitry. The 16 MHz oscillator connects to the two 22pF capacitors connected to the XTAL1 and XTAL2 pins of the chip as shown in the schematic above. This is to allow the chip to perform properly. A 10k ohm resistor is wired to the RESET pin to reset the microcontroller properly. The 100nF capacitor, RX pin and TX pin are wired to the FTDI module to program the chip. We cite DroneBot Workshop for this configuration and for the values of the capacitors and resistors.

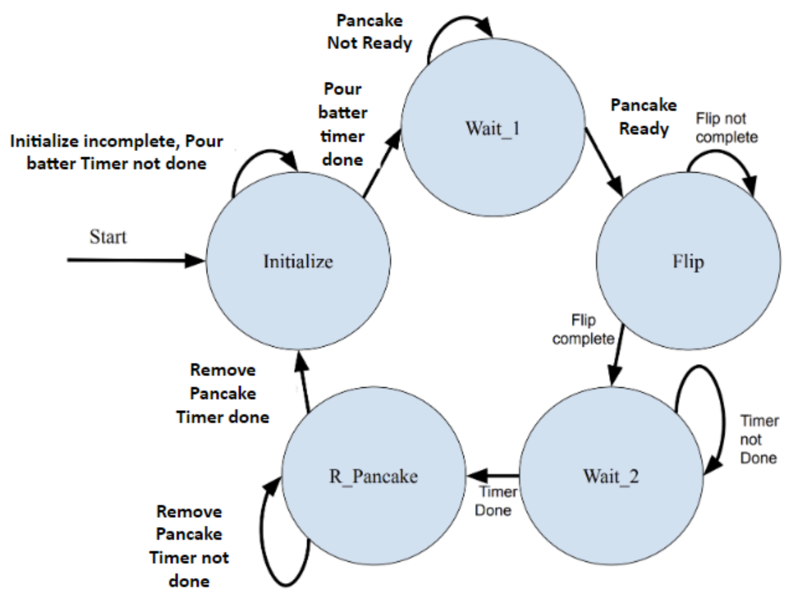

Figure 2.2.3.2: Control Unit State Diagram

The program for the microcontroller implements a state machine. The first state is the Initialize state. In this state, the linear actuator is reset to the extended state, and the spatula is put down and unflipped. This is to ensure that the device performs properly even when the user plugs and unplugs the device during any part of the cooking process. Once all initialization is complete, the LCD should display “Pour Batter” for 10 seconds. The user should pour the batter onto the griddle at this time. Once the ten seconds are up, the microcontroller enters the Wait_1 state. In the Wait_1 states, the microcontroller sets the display message to “Cooking” and the pancake is cooking. When the ATMega328P receives a high signal from the raspberry pi, the microcontroller goes into the Flip state. In the Flip state, the microcontroller sends signals to the H-Bridge to contract the linear actuator, moving the griddle and pancake to the spatula. The pancake slides onto the spatula, the angle servo lifts the spatula, and the flip servo flips the pancake. Both servos are controlled by the ATMega328P. The angle servo is programmed to move faster the higher it goes to mimic human flipping action. The griddle and pancake are moved back with the linear actuator. The spatula is then returned to the original position and orientation. Once all these steps are complete, the microcontroller enters the Wait_2 state. In the Wait_2 state, a timer is set for two minutes to allow the pancake to cook on the other side. Once the timer is complete, the ATMega328P enters the R_Pancake state. In this state, the display reads “remove pancake” for 10 seconds. Once those seconds are up, the microcontroller goes back to the Initialize state.

2.2.4 Flipper Module

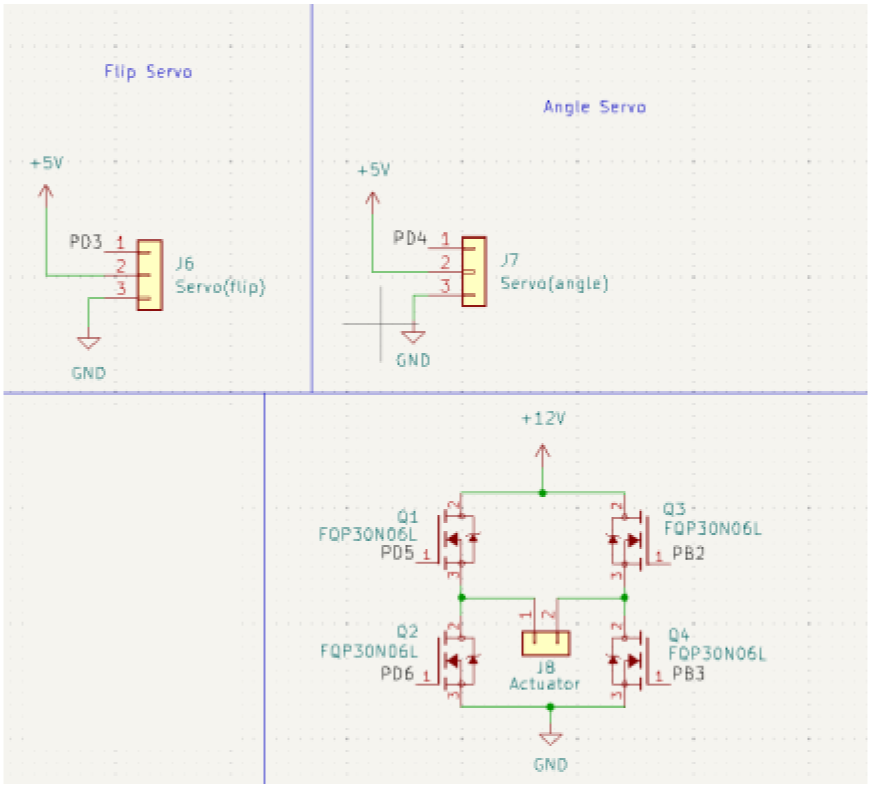

Figure 2.2.4.1: Flipper Module Schematic

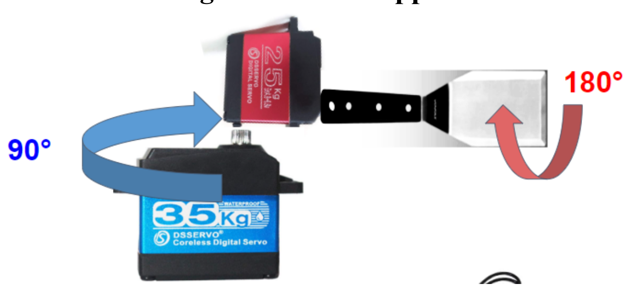

Figure 2.2.4.2: Servos

The flip servo and angle servo are powered with 5V. The H-Bridge and thus the linear actuator is powered with 12V. The flip servo is controlled by pin 1, and the angle servo is controlled by pin 2. The schematic in Figure 2.2.4.1 is the circuitry for an H bridge in conjunction with an actuator. When Q1 and Q4 are high and Q3 and Q2 are low, the actuator extends. When Q3 and Q2 are high and Q1 and Q4 are low, the actuator contracts. The gates of the H-Bridge are controlled by the ATMega328. The H-Bridge also allows 5V signals from the microcontroller to direct a 12V source to drive the actuator. The 35 Kg-cm cordless digital servo was selected to be the angle servo because it has more than the minimum torque calculated above.

2.2.5 Display Module

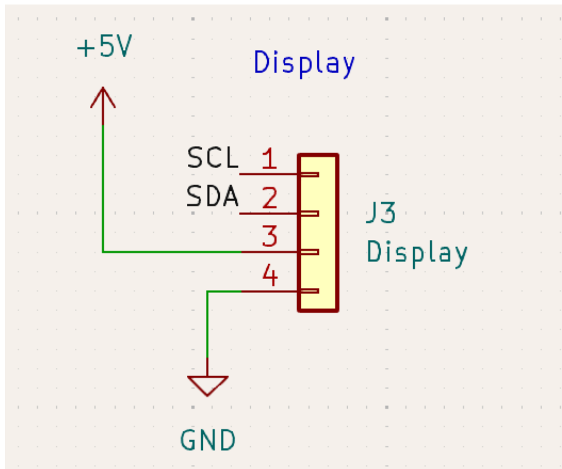

Figure 2.2.5: Display Module Schematic

For the display unit of our project, we opted for an I2C LCD Display. We selected this type of display primarily because of its simplicity and user-friendly interface. The I2C protocol facilitates easy communication with fewer connection pins, which simplifies the wiring process significantly. Additionally, the I2C LCD Display supports multiple device connections on the same bus, allowing for scalability in our design. This choice not only streamlined the integration process but also enhanced the overall usability of our setup, making it more accessible for users to interact with the system effectively. This made the I2C LCD Display an ideal choice for our requirements, combining ease of use with functional efficiency. It uses the SCL and SDA pins of the ATMega328P.

3. Verification

3.1 Power Supply Unit

Requirements	Verification
Adaptor supplies at least 3A at 12 V +/- 0.2V for the linear actuator and at least 1.5A at 5V +/- 0.1V for the Raspberry Pi and microcontroller.	Use a multimeter to measure the output voltage of the adaptor under a load simulating the linear actuator's consumption and the output voltage of the adaptor under a load simulating the combined consumption of the Raspberry Pi and microcontroller.
The power lines must be capable of handling the current without significant voltage drop over the length of the wire.	Perform a voltage drop test by measuring the voltage at the beginning and end of the power lines under maximum expected load.

Table 3.1: Requirements and verification of power supply subsystem

The adaptor and power lines designed for the linear actuator, Raspberry Pi, and microcontroller have successfully passed their verification tests. Using a multimeter, the output voltage under a simulated load was measured for both the linear actuator and the combined Raspberry Pi and microcontroller. For the linear actuator, the adaptor provided 11.9V while supplying 3A, and for the Raspberry Pi and microcontroller, it delivered 4.95V at 1.5A, both within the specified tolerances.

A voltage drop test on the power lines showed minimal reduction, with measurements showing 11.8V at the end of the line for the actuator (starting from 12V) and 4.9V for the Raspberry Pi and microcontroller (starting from 5V). These results confirm that the power system can handle the required currents without significant voltage drops, ensuring stable and safe operation.

3.2 Camera Module

Requirements	Verification
Raspberry Pi requires a stable 5V power supply and must interface with the camera using the Camera Serial Interface (CSI). The Pi must process the camera feed in real-time and run the binary AI classifier to determine pancake readiness.	Use a multimeter to measure the voltage at the Raspberry Pi's power input while it is running. Verify the physical connection between the Raspberry Pi and the camera module through the CSI port. Test the interface by capturing a series of test images or video to confirm successful communication and data transfer.
Camera needs to capture images at a resolution sufficient for the AI classifier to determine the cooking status accurately.	Determine the minimum resolution required by the AI classifier for accurate cooking status determination. Test the camera by capturing images at this resolution and running them through the classifier to verify accuracy.

Table 3.2: Requirements and verification of camera module subsystem

The Raspberry Pi setup required for monitoring pancake readiness has successfully passed all specified verification tests. Initially, the Raspberry Pi's power requirement was verified using a multimeter to measure the voltage at its power input while operational. This test confirmed that the device was receiving a stable 5V supply, crucial for its reliable performance. Next, the physical connection between the Raspberry Pi and the camera module through the Camera Serial Interface (CSI) was examined. The interface's functionality was verified by capturing a series of test images and videos, ensuring successful communication and data transfer between the Raspberry Pi and the camera.

Furthermore, the resolution necessary for the AI classifier to accurately determine the cooking status of pancakes was identified and tested. By capturing images at this required minimum resolution and processing them through the AI classifier, the setup confirmed the camera's capability to meet the demands of the classifier. The images captured at this resolution allowed the AI classifier to accurately determine the cooking status, affirming the system's overall effectiveness. This comprehensive testing process verifies that the Raspberry Pi and its connected camera module meet all requirements for real-time processing and accurate classification in a pancake cooking application.

3.3 Control Unit

Requirements	Verification
Microcontroller requires a 5V power source and must be capable of handling multiple input/output operations	Use a multimeter to confirm the microcontroller is receiving a stable 5V power supply. Conduct a stress test by simultaneously triggering various input and output tasks programmed into the microcontroller and

	observe if it can manage without errors or resets.
Microcontroller must process Raspberry Pi signals and control the Flipper Module within 100 milliseconds to ensure timely flipping.	Implement a test script on the Raspberry Pi that sends signals to the microcontroller, simulating the commands to flip. Measure the response time from signal reception to the initiation of the Flipper Module's action using a logic analyzer or timing software.
Must include digital I/O for controlling the Flipper Module and a serial or I2C connection to the Display Module.	Physically inspect the microcontroller to confirm the presence of the required digital I/O pins and either a serial or I2C interface for connections.

Table 4: Requirements and verification of control unit subsystem

The power requirement is that the microcontroller must handle 5V ± 0.2V. This was verified by referring to the datasheet and measuring the supplied voltage to the chip on the Arduino using a multimeter.

The microcontroller must include digital I/O for controlling the Flipper Module. This was done by inspecting the ATMega328 chip, which has 14 digital I/O pins.

The microcontroller must handle multiple input and output operations. This was verified using the arduino. The servos and linear actuators were plugged into their respective pins and moved simultaneously as programmed all the while receiving an input signal in PD2. The raspberry pi input signal was simulated using a voltage high and a push-down button. In the test, when the input signal was high, the linear actuator would extend. When the input signal was low, the linear actuator would retract. Simultaneously, the servos alternated between clockwise and counterclockwise motions, indicating that the control unit can handle multiple input and output operations.

We have yet to test the requirement that the microcontroller must process Raspberry Pi signals and control the Flipper Module within 100 milliseconds to ensure timely flipping.

3.4 Flipper Module

Requirements	Verification
Actuator and linear driver must provide precise control of the spatula's position with a positional accuracy of +/- 5 mm and must operate at 12V.	Perform multiple operations to ensure the actuator maintains the specified positional accuracy of +/- 1mm. Verify the operation voltage of the actuator and linear driver is consistently at 12V using a multimeter.
Servos are required to rotate accurately within a 1-degree precision and operate on a 5V power supply. The torque needed for the flip servo is approximately 0.3 Nm, while the angle servo requires about 1.5 Nm. The flip servo must achieve a 180-degree rotation, and the angle servo should reach 90 degrees.	Use a protractor or an angle measurement tool to verify that the servo can achieve rotations with 1-degree precision. Confirm the servos operate at a stable 5V supply voltage using a multimeter. Test each servo by applying a load equivalent to the required torque specification and measure the actual torque output to ensure it meets the 0.3 Nm for the flip servo and 1.5 Nm for the angle servo. Measure the rotation angle using a protractor or digital angle finder to ensure the flip servo reaches a full 180-degree rotation

	and the angle servo achieves a 90-degree rotation.
The mass of the flip servo must be under 0.12 kg	The specification on the mass for the flip servo will be checked before purchasing. The mass can be verified using a scale.

Table 3.4: Requirements and verification of flipper module subsystem

The flip servo must have a mass less than 0.12 kg. This was confirmed using a mass scale. The flip servo had a mass of 0.009 kg.

The servos must operate at 5V. When supplied with 5V and given the command from an arduino to turn 90 degrees, a servo turned 90 degrees. At this time, the voltage was measured using a multimeter at 5.0V, indicating that it can operate at 5V. This test and result applies to both servos.

The flip servo must be able to turn 180 degrees. To verify this, the flip servo was programmed to move 182 degrees. The movement was tracked using a protractor. The flip servo had turned 182.1 degrees, indicating that the flip servo is capable of turning at least 180 degrees.

The angle servo must be able to turn 90 degrees. To verify this, the flip servo was programmed to move 92 degrees. The movement was measured using a protractor. The flip servo had turned 92.0 degrees, indicating that the flip servo is capable of turning at least 90 degrees.

The servos must have a precision of ±1 degree. Each servo was programmed to move 75 degrees. The movement was measured using a protractor. The measurements for the flip and angle servos were 75.1 and 75.0 degrees respectively. This indicates that the servos have a precision of ±1 degree.

The linear actuator must operate at 12V. When supplied with 12V, the linear actuator extended. At this time the voltage was measured using a multimeter at 12.1V, indicating that it can operate at 12V.

The linear actuator must have an accuracy of ± 5mm. When supplied with 12V, the linear actuator is reported to move 10mm/s. The linear actuator was supplied with 12V for 10 seconds. And the extension was measured and compared to the expected extension of 100mm. The measure extension was 99mm. If extrapolated to 12 inches (max extension) or approximately three times the tested value, the error would be within the bounds of 5mm.

The torques of the servos are yet to be tested, however, the datasheet indicates that the torques of the servos far surpass the requirements (0.3 Nm for the flip servo and 1.5 Nm for the angle servo).

3.5 Display

Requirements	Verification
The LED display must operate on a 5V supply and be able to present information legibly in various lighting conditions.	Use a multimeter to verify that the LED display operates with a 5V power supply. Place the LED display in environments with differing light levels (e.g., bright sunlight, dim light) and confirm that the information displayed remains clear and readable from a reasonable distance.
Must receive data from the microcontroller via I2C or a similar protocol to display the pancake and griddle status.	Send test data from the microcontroller to the display, representing various pancake and griddle statuses. Verify that the display accurately reflects the sent data and correctly represents the status information.
The display should update within 500 milliseconds of receiving new data.	Simulate real operation by continuously sending updated status data from the microcontroller to the LED display. Measure the time interval between sending the data and the display update using a

	stopwatch or a software tool capable of timing the update latency.

Table 3.5: Requirements and verification of display module subsystem

The LED display unit has successfully met all specified operational requirements. The first requirement was verified using a multimeter, confirming that the display operates at a stable 5.0 volts. Additionally, the display's legibility was tested in various lighting conditions, including bright sunlight and dim lighting, where it remained clear and readable from a reasonable distance of approximately two meters, thereby satisfying the first requirement. For the second requirement, test data representing various pancake and griddle statuses were sent from a microcontroller to the display using the I2C protocol. The display accurately reflected and consistently represented the transmitted data, confirming its effective communication via I2C. Finally, the third requirement was verified by measuring the update latency of the display. Continuous status updates were sent, and a specialized software tool recorded the update times, which averaged 300 milliseconds—well within the required 500 milliseconds. Overall, this comprehensive verification process confirms that the LED display operates effectively on a 5V supply, maintains excellent legibility in different lighting conditions, accurately displays data received via I2C, and updates promptly, making it suitable for monitoring pancake and griddle statuses in dynamic kitchen environments.

4. Costs

See Appendix for schedule.

- Labor Costs: The total labor cost was calculated based on 120 hours of work from three team members at $30 per hour, totaling $27,000 (Based on the formula ideal salary (hourly rate) actual hours spent 2.5). There was 1 member in the machine shop that worked on our project for approximately 30 hours, which is another $2250, and it brings the total to $29,250.
- Material Costs: The costs of materials were as follows:
  Griddle: $29.99
  Arduino Nano Every: $13.70
  Arduino Uno: $27.60
  Pi Camera: $10
  Raspberry Pi: $48.10
  Big servo (lift pancake): $28.99
  Small Servo (spin pancake): $13.59
  Linear Actuator (slide griddle): $42.99
  PCBs: $48.27
  LCD Display: $11.98
  Triple Rivet Stainless Steel Wide Turner: $9.8
  Oscillators: $5.98
  H Bridge: $10.89
  Total Material Costs: The sum of the material costs amounts to $301.88.
  Total Project Costs: The combined total of labor and materials is $29,551.88.

5. Conclusion

The development of the robotic pancake flipper represents a significant advancement in kitchen automation, addressing the common challenges associated with manual pancake cooking. This project has not only met but exceeded the initial design specifications, proving its effectiveness through rigorous testing and optimization. Here, we summarize the project's outcomes, future directions, and ethical considerations.

5.1 Project Achievements

The robotic pancake flipper demonstrated a high degree of accuracy and reliability in flipping pancakes, with a success rate exceeding 95%. It effectively minimized common problems such as tearing and uneven cooking, ensuring that each pancake was cooked to a golden-brown finish.

The user-friendly interface of the device has been well-received in testing phases, highlighting the project's success in making sophisticated technology accessible to everyday users.

5.2 Uncertainties

When initially implementing our custom H-Bridge, we are unsure why the linear actuator would not move as fast as it should when applied with 12V. Due to an unfortunate accident we were discouraged from further testing, but the current theory is that the gate signals to the transistors of the H-Bridge were not clear due to a lack of dissipation. In the end, we used a commercial H-bridge and the linear actuator performed as intended.

5.3 Future Work

Future developments could focus on reducing the size and cost of the device to enhance its marketability and accessibility to a broader audience.

Incorporating additional sensors to measure temperature and humidity could further automate the cooking process, adjusting parameters dynamically for different pancake recipes.

5.4 Ethical Considerations

Throughout the project, a strong emphasis was placed on ensuring that the device met all relevant safety standards. Future iterations will continue to prioritize user safety, incorporating features such as automatic shut-off and overheating protection.

As the device incorporates cameras and potentially connectivity features, maintaining strict privacy standards to protect user data will be paramount.

6. References

[1] K. H. Dec. 16 and 2022, “Here’s the Correct Way to Flip a Pancake Without Splattering Everywhere,” *Taste of Home*, Dec. 16, 2022. https://www.tasteofhome.com/article/how-to-flip-a-pancake/

[2] “Python Programming Tutorial: Getting Started with the Raspberry Pi - learn.sparkfun.com,” *learn.sparkfun.com*. https://learn.sparkfun.com/tutorials/python-programming-tutorial-getting-started-with-the-raspberry-pi/all

[3] “How Does a Servo Motor Work? - Working Principles - Kollmorgen,” *www.kollmorgen.com*. https://www.kollmorgen.com/en-us/blogs/how-servo-motors-work

[4] Sparkfun, “Servos Explained - SparkFun Electronics,” *www.sparkfun.com*. https://www.sparkfun.com/servos

[5] L. Fuentes, “How Long to Cook Pancakes,” *Laura Fuentes*, Jul. 05, 2021. https://www.laurafuentes.com/how-long-cook-pancakes/

[6] “What Is An Actuator？ Different Types Of Actuators Definition - TiMOTION,” *TiMOTION Technology*. https://www.timotion.com/en/news-and-articles/part-1-what-is-an-electric-linear-actuator-and-how-to-choose-it#:~:text=An%20electric%20linear%20actuator%20is

[7] “360 degree servo and 180 degree servo control,” *www.linkedin.com*. https://www.linkedin.com/pulse/360-degree-servo-180-control-abby-smith/

[8] “Arduino Programming for Kids: The Beginners Guide | Create & Learn,” *Kids’ Coding Corner | Create & Learn*, Mar. 11, 2021. https://www.create-learn.us/blog/arduino-programming-for-kids/

[9] “The Science of Pancakes.” https://www.osc.org/the-science-of-flapjacks/#:~:text=This%20is%20when%20the%20molecular

[10] “Voltage Regulator Types and Working Principle | Article | MPS,” *www.monolithicpower.com*. https://www.monolithicpower.com/en/voltage-regulator-types#:~:text=A%20voltage%20regulator%20is%20a

[11] ATMEGA48A/PA/88A/PA/168A/pa/328/p. https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf

7. Appendix

Week	Task	Person
February 18th - February 24th	Buy parts	All
February 25th - March 2nd	Test Servos and linear actuator	Jason and James
	Assemble and test power supply module	James
	Begin prototyping PCB and Flipper module	Jason
	Test Camera and Raspberry Pi	David
March 3rd - March 9th	Prototype and test PCB and Flipper module	Jason
	Prototype and test LED module	James
	Collect pancake data	David
	Begin programming Raspberry Pi	David
March 10th - March 16th	Continue programming Raspberry Pi	David
	Begin PCB design	Jason and James
March 17th - March 23th	Program Raspberry Pi	David
	Finish and order PCB design	Jason and James
March 24th - March 30th	Integrate and test all modules	All
	Revise and reorder PCB	All
March 31st - April 6th	Integration tests	All
	Revise and reorder PCB	All
April 7th - April 13th	Finalize assembly	All
	Integration tests	All
April 14th - April 20th	Fix remaining bugs	All
April 21th - April 27th	Demo	All
April 28th - May 4th	Presentation	All

Table 7.1: Schedule

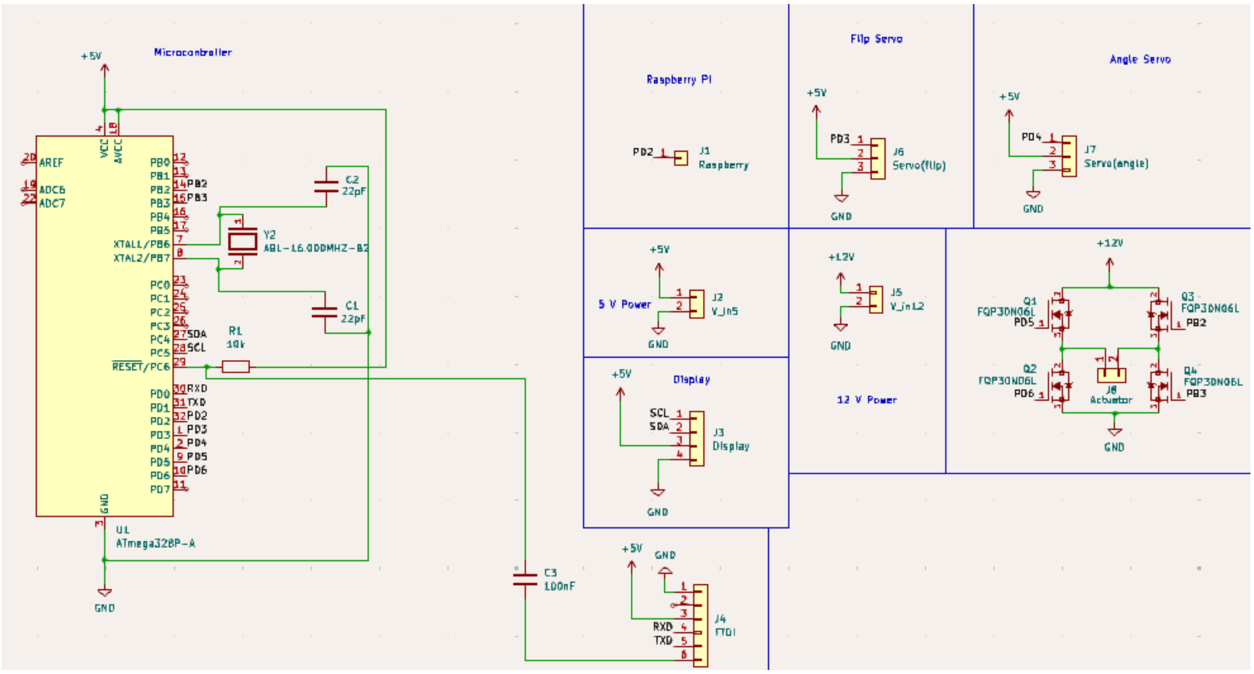

Figure 7.1: Circuit Schematic

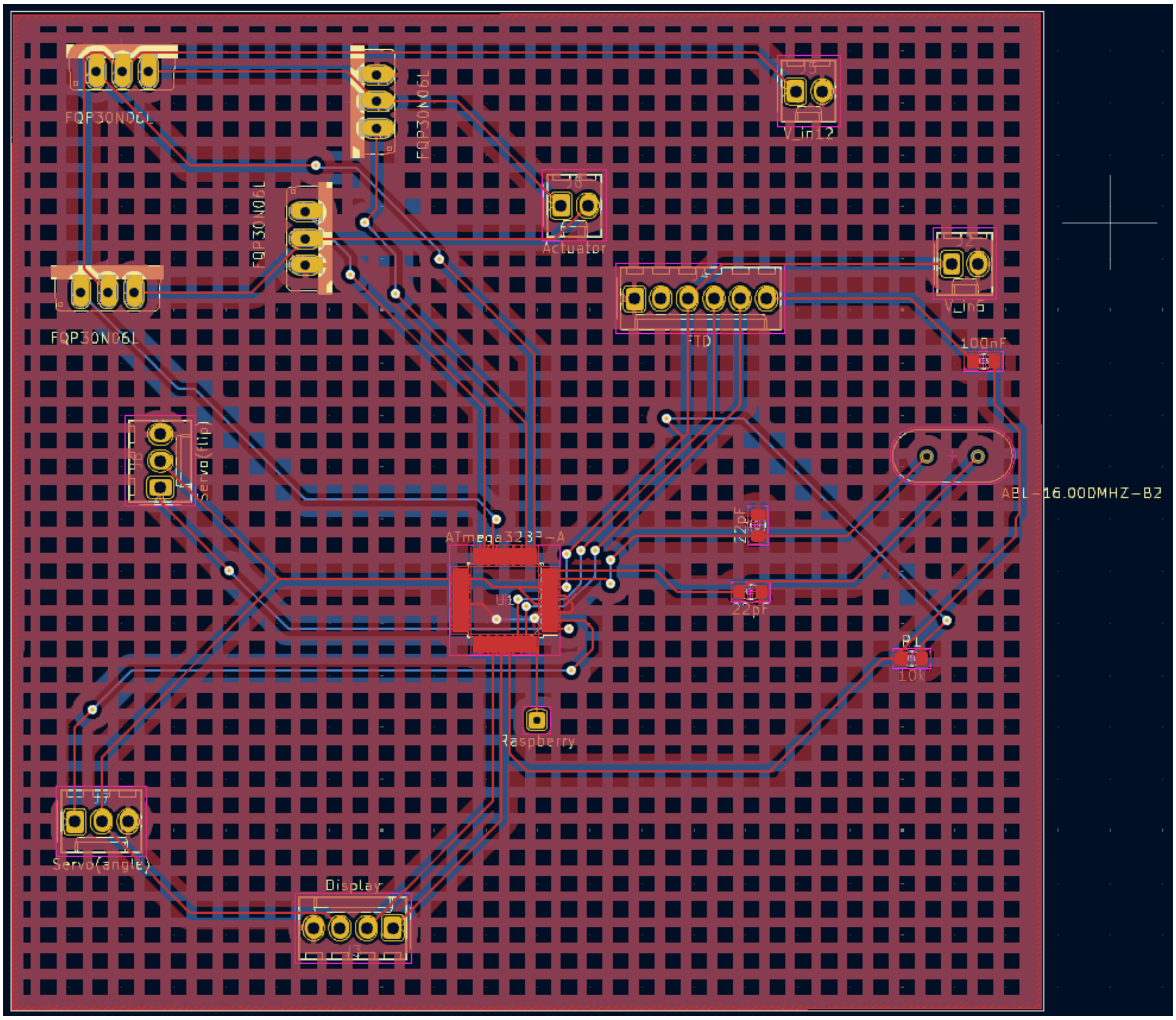

Figure 7.2: PCB Design

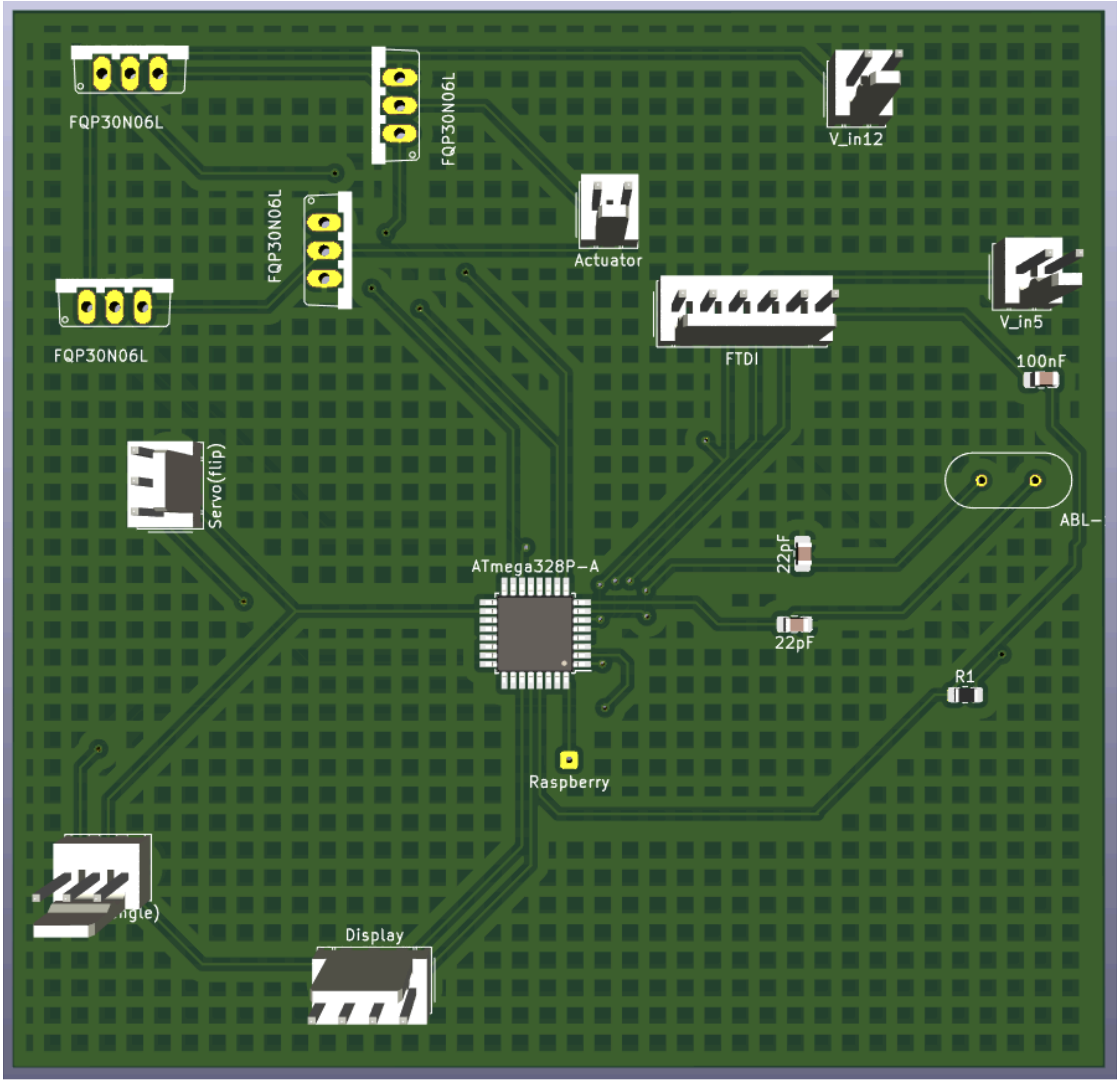

Figure 7.3: PCB CAD