**Intelligent Pet Feeding Machine**

Team: PetLovers

Ruihan Chen ([rhchen@umich.edu](mailto:rhchen@umich.edu)), Yiran Gu ([guyirran@umich.edu](mailto:guyirran@umich.edu)),

Zhiyu Wu ([zhiyuwu@umich.edu](mailto:zhiyuwu@umich.edu)), Zichun Wang ([zichunw@umich.edu](mailto:zichunw@umich.edu))

1. **Customer**

New pet owners who are uncertain about pets’ eating and drinking habits. In addition, pet-keepers who do not stay at home all day who might need the function of automatic feeding, or perhaps zoo staff.

1. **Value**

Our Intelligent Pet Feeding Machine has several functions. One major function is to keep monitoring and recording the amount of water and food consumed by the pet, and another helpful function is to add a proper amount of food and water to the container during meal time. Our device concludes the diet patterns and habits of the pet so that the users can create a customized healthy feeding plan for their pet accordingly.

In the monitoring mode, our device gives warnings to pet owners when pets consume too little water or have abnormal diet patterns. For example, most cats drink little water every day and it’s very hard to tell how much water a cat drinks. However, cats are likely to have health problems such as urinary system disease, kidney disease and constipation if not taking in sufficient water in a long term. If pets drink insufficient water for a long time, the feeding machine will warn the owner and show the history data recorded in the SD card.

In the feeding mode, it is basically used when the owners don’t stay at home with pets, our device automatically adds proper types and amounts of food and water during mealtime. In this way, even if the owners get out for up to 6 days, the pets can still have meals scientifically during this period.

The two functions mentioned above as well as the other functions can work simultaneously.

1. **Approach**

In order to realize the monitoring function, load cells are fixed to the pet’s food bowl and water bowl. The device monitors and records water and food consumption through measured food and water weight. We use another 20-kg load cell to weigh the pet. In this way, the program can tell one pet from another in case the pet owner has multiple pets. It processes data and stores them on an SD card, and gives feeding plans, warnings and suggestions shown on the display. Users can also request taking a look at the recorded data on the display by pressing the corresponding key on the keypad.

To release food to the food bowl, a self-designed food tank is fixed above the food bowl. The tank has 6 grids and the owner can add a desired amount of food, typically one day’s supply. The food tank is connected to a servo, which rotates 60 degrees a time and releases food in one grid. A switch provides feedback to the Nucleo that enough angle has been reached and this will stop the servo. The time interval of releasing food varies, taking the remaining food on the load cell and how much food and water consumed in the last six hours into consideration. The pet owner can also release a portion of food by pressing the button on the controller. A water pump is used to add water to the water bowl and the amount of water added can be set by the pet owner.

Our history data and database is stored in an SD card through the F103C8T6 chip. The Nucleo can access the data by sending requests to the chip. The chip will then enter DMA interrupt, and transmit data back to the Nucleo board. The history data includes temperature and humidity in the past 5 days, which can be translated into freshness of food and water. Our history data keeps track of the weight of the pets, which will then give warning if the pets are overweight or underweight. Data also includes weight on each load cell when a periodic timer interrupt is triggered.

**Major functions of the project**

**Feeding**

The feeding function mainly consists of food feeding and water feeding. Two modes supplement each other: automatic feeding and manual feeding mode.

1. Steering food container
2. Mechanical Structure

The food tank is designed to have 6 inner grids and each holds about a day’s worth of food for pets. The pet owner will be given suggestions to put the pet’s daily consumption of food in each grid. The rotatable food tank is driven by a 360° parallel servo and releases the food in one grid to the food bowl every time.

The container is constructed with a plastic bucket (without bottom) and plastic boards to separate the grids. The foundation is made up of a water tube with a cap and 3 L-shaped stands. The servo is connected to the cap with drilled holes and screws. An acrylic plate with a hole to let the food pass through is fixed on to the cap, it is at the same height with the bottom of the bucket.

When the bucket is moving, the foundation remains still, and food in one grid will be let out through the hole.

1. 360° Parallel servo

We are using timer 4 for PWM control here, and the configuration values are as follows:

prescaler: 19

ARR: 3999

CCR: 318

From the servo manual, the servo will be stopped if \*tim4\_ccr is about 300, and starts moving from 312. After multiple tests, we decide that 318 is the right speed to rotate the container with filled food.

1. Switch and feedback

To stop the servo immediately after a grid of food is released, we use a switch as feedback. With a pull-up resistor, the GPIO output will be pulled low if the switch is connected. As long as the HAL\_GPIO\_ReadPin function detects a high, the \*tim4\_ccr2 will be set to 300 to stop the servo.

On each end of the grid, a plastic stick is placed outside the bucket to trigger the switch. In this way, we can ensure that the container is rotated 60 degrees each time.

1. Water pump

The peristaltic liquid pump transports water from the water tank into the bowl as required. In the automatic feeding mode, it will be stopped if the value of a load cell reaches a previously set value.

1. 12V DC water pump

The speed of water flow is about 0.87 ml/second by measurement.

1. H-bridge and flow control

To control the flow of water, we used an H-bridge to enable and disable the power. Water pump requires a DC power supply of 12V, which is provided from the potential meter and through H-bridge. The enable bit is controlled by GPIO output from the L4R5ZI-P Nucleo board. EN1 connects to and GPIO output by configuration in the .ioc file. An additional 5V comes from the Nucleo board, which is used to control the GPIO pin. Although the water pump is able to pump water in both directions, as we only need to pump water from the tank to the water bowl, EN2 is connected to GND. When EN1 is high, water flows, and when EN1 is low, water flow stops.

1. Water tank

We used a plastic storage box as the water tank.

1. Manual feeding mode

Except for the automatic feeding, the device provides a manually keypad control feeding mode (Key press “1”) ,with the same mechanical structure, manual feeding mode turns the servo and releases the food in the container each time when the customer pushes the button. With the same logic, the water pump can be controlled by Key press “2”, so as the water pump will transport a certain amount of water.

**Processor and Memory**

Everyday, our processor (L4R5ZI-P Nucleo board) will record temperature, humidity, food consumption, water consumption and weight of the pet and create a 32-bit uint8\_t array. The format of the data is (the rest of the bits is space character):

| Bit number | 0 | 1-3 | 5-7 | 9-12 | 14-17 | 19-23 | 30 | 31 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data content | $ | temperature | humidity | Food consumption | Water consumption | Pet weight | \n | \0 |

The data is transferred from our L4R5ZI-P Nucleo board to our F103C8T6 Chip and then written into the SD card. Any valid data array starts with an ‘$’ and ends with “\n\0”.

Everyday, our processor will send a 32-bit uint8\_t array to the F103C8T6 Chip starting with ‘%’ which asks for a warning array. When the F103C8T6 Chip receives a command like that, it reads 5 data arrays for the past 5 days from the SD card and creates a 32-bit uint8\_t warning array after much calculation. The format of the warning array is:

| Bit number | 0 | 1 | 2 | 3 | 4 | 5 | 6-29 | 30 | 31 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data content | $ | T/F | T/F | T/F | T/F | T/F | Space character | \n | \0 |

For the 1st bit, if the food needs to be changed, we put T, otherwise we put F.

For the 2nd bit, if the water needs to be changed, we put T, otherwise we put F.

For the 3rd bit, if the pet is eating too much, we put T, otherwise we put F.

For the 4th bit, if the pet is eating too little, we put T, otherwise we put F.

For the 5th bit, if the pet is drinking too little water, we put T, otherwise we put F.

1. SD card
2. Serves as our database. Records temperature, humidity, food consumption, water consumption and weight of the pet everyday for future analysis and warnings.
3. Read and write SD card by FATFS in CubeIDE. We use f\_puts() of FATFS to write to a specific file in SD card and use f\_gets() to get the last 5 lines of data in the file in SD card.
4. F103C8T6 Chip
5. Communicate with the SD card adapter via SPI.
6. Full-duplex master for SD card adaptor.
7. Communicate with Nucleo board via UART DMA.
   1. Receive commands and daily data from Nucleo board.
   2. Process data read from SD card and send warnings to Nucleo board if necessary.
8. L4R5ZI-P Nucleo board
9. Controls all the other components
10. Processes data from all the sensors
11. Receive warnings from F103C8T6 Chip and show suggestions and warnings on oled display.
12. Use periodic timer interrupts to feed automatically and store data daily by sending data to F103C8T6 Chip and F103C8T6 Chip will store it in SD card.

**Measurement**

Our feeding machine measures the weight of the pet, pet food and water, as well as the ambient temperature and humidity, and calculates food and water consumption and freshness based on measured data.

1. Weight measurement: Load cell and HX711 ADC
2. Nucleo reads HX711 through I2C.
3. Two 1 kg load cells weigh the pet food and water respectively and a 20 kg load cell weighs the pet.
4. When force applied, the metal bar deforms and the resistance changes.
5. 4 wires output to 24-bit ADC HX711 which amplifies the resistance change.
6. HX711 output is linear to weight, using a calibration function to output weight in grams.
7. Take several measurements and calculate the average to improve accuracy.
8. Temperature and humidity measurement: AHT10
9. Transmits to Nucleo through I2C.
10. First 3 bits are raw data for temperature, next 3 bits are raw data for humidity, needs a conversion to output temperature in Celsius and humidity in %.
11. Data processing
12. The pet will not stay on the weighing platform all the time, so only refresh the reading if the 3 consecutive measurements differ less than 10% and the 2nd measurement is displayed because the pet is guaranteed to stay on the platform throughout the whole measurement.
13. To calculate how much a pet eats and drinks everyday, when the day begins, add the weight of remaining food and water to the consumed food / water variables. Each time when pet food or water is added automatically or manually, measure the weight of added food or water and add this weight to the consumed food / water variables. When the day ends, subtract the remaining food and water from the consumed food / water variables.
14. To measure the freshness of the food and water, every day, a proportion calculated based on temperature and humidity will be subtracted from current freshness. When the freshness becomes 0%, a warning will be generated and freshness becomes 100% after the user adds fresh food and water.

**User Interface**

User interface consists of one 4\*4 Matrix Keypad and one 128\*64 OLED display. There are detailed instructions on the display, which show the usage of each button on the keypad. Users send requests to the embedded system through the keypad, and the system responses depending on different kinds of request, maybe displaying response messages on the OLED display. For example, when the user clicks button “4”, it will send a request to the embedded system, and the system responds to display the measured data of each load cell, the temperature and humidity on the display.

(a) 4\*4 Matrix Keypad

1. Connect pins to power (series with a large resistance), then test the output to determine the one-to-one mapping relationship between pins and button switches.
2. Configure four pins which represent columns as GPIO Output, and the other four pins which represent rows as GPIO Input (with external interrupt).
3. If any of the buttons is clicked, it will jump to the HAL\_GPIO\_EXTI\_Callback function. Set a minimum time interval to avoid getting the same press multiple times in the callback function.
4. For any valid click, write to all four GPIO input pins one by one, and for each time, check the value of four GPIO output pins one by one to figure out which button is clicked. Then send a corresponding request to the kernel to handle the request.

(b) 128\*64 OLED

1. Use I2C communication protocol, configure I2C1.
2. Use HAL\_I2C\_Men\_Write to implement the initialization of OLED, setCursor() function to set up a cursor, and UpdataScreen() function to update the display based on values in the register.
3. Use a font library from Github to transfer characters to 2d index, then implement functions WriteChar() and WriteString(), flatten the array of string (in 2d index) into a 1d array register, waiting for UpdateScreen() function call to display the screen.

**Appendix**

Instruction of each button

1- release food

2- release water

3- manually add food

4- display measured data

#- enter/ next page

5- calibration

1- calibrating load cell for food

2- calibrating load cell for water

3- calibrating load cell for pet

#- enter/ next step

0~9- type in number of known weight

\*- backspace

6- display history data

#- previous day

\*- next day

#- enter/ next page

\* - delete

D- go back to menu

1. **Essential System Components**

**Processor:**

1 Nucleo L4R5ZI-P (stock)

1 STMF103C8T6 (order)

**Inputs:**

1 4\*4 Keypad (order)

3 load cell sensors (order)

1 LM335Z thermometer (order)

1 HR202L humidity resistance (order)

1 Switch (stock)

**Outputs:**

1 NHD-C12864A1Z-FSW-FBW-HTT Graphic oled display (stock)

1 H-bridge (stock)

1 Parallax Feedback 360° High-Speed Servo (#900-00360) (stock)

1 Peristaltic liquid pump (order)

**Other:**

1 SD card (stock)

1 SD card adapter (stock)

1 ST Link (order)

1. **Complexity Points**

Load cells - I2C communication (0.5)

Thermometer - I2C communication (0.25)

Hygrometer - I2C communication ( 0.25)

360° High-Speed Servo (0.25)

Water pump (DC motor + H-bridge) (0.25)

Graphics display - I2C communication (0.66)

4\*4 Keypad (0.75)

SD card - SPI communication(0.75)

Blue pill (STM32F103C8T6) - UART DMA communication with Nucleo board (0.25)

I2C interface, SPI interface, UART interface (0.5)

Switch (0.25)

## **Tested Components**

* Pressure sensors: force sensing register, Pololu.
  + It takes too long to stabilize the readings.
  + When the weight is over 2kg, the accuracy decreases, but it is relatively difficult to calibrate.
  + It’s easy to get bent, which influences the accuracy.
* DC motor/ Servo for rotating our device: Normal servos can only turn 180 degrees, which can not meet our requirement of turning a whole cycle for the food container.
* Keypad: Keypad Switch Parallax Inc. 27899PAR-ND.
  + The button control is not sensitive. There are over 50% of times when we try to click the button, but it cannot close the switch under the button.
* Communication directly between Nulceo L4R5ZI-P and SD card adapter: Most tutorials accessible online are based on F-series boards, which have quite different configuration and library functions from L-series boards. After trying on L board for about one week consecutively, we changed to F103C8T6 for FATFS with SPI between F103C8T6 and SD card reader. Additionally, we use UART transmission between the two boards.
* 3D components: The printing time and procedure is quite long for version iteration, while laser cutting parts with transparent acrylic boards can be easy to assemble and cut.

## **Member Task Breakdown**

Ruihan Chen

1. Determined to use I2C to transfer data between ssd1306 oled display and Nucleo and develop some basic functions to use the display including SetCursor, write in char/string and UpdateScreen with Zhiyu Wu.
2. Used AHT10 sensor and I2C to detect the ambient humidity and temperature.
3. Overcame the difficulties of connecting the load cell to hx711 chip and then transferring data back to Nucleo through I2C to let load cell work stably with Zhiyu Wu.
4. Construct the mechanical structure with other team members.
5. Integrated all functions into a main program with Zhiyu Wu.
6. Designed and realized the UI on the display and made use of all keys on the keypad with Zhiyu Wu.
7. Implemented monitoring function, view history data function mostly by myself, and several other functions with Zhiyu Wu.
8. Designed the poster with Yiran Gu.

Yiran Gu

1. Wrote code for 4\*4 matrix keypad myself, by pulling the row to be low one by one and checking the columns. Configured 4 pins as GPIO output and 4 as input.
2. Tried using L4R5Zi-P to communicate with SD card adapter with Zichun Wang, we determined to use the Blue pill (STM32F103 board) to communicate with SD card adapter through SPI; we found FatFs library function in online tutorial, wrote SD card read and write functions
3. Instead of pooling and interrupt, used UART DMA to communicate between the Blue pill and Nucleo board with Zichun Wang; designed the fixed data transmission format with all the other group members; debug the SD card read and write functions and communication between two boards with Zichun Wang, and debug the message with its format problems with ending bit.
4. Contributed to the configuration of DC motor propelled water pump and 360˚ parallel servo with Zichun Wang; built the H-bridge circuit with 12V power and Nucleo board, configured the GPIO pin for Enable pin, and decided the logic of controlling the water pump.
5. Helped contribute to the design and realization of stopping the servo to a certain degree with switch feedback and adjusted the speed of the device using PWM control carefully with Zichun Wang to had it work
6. Designed and constructed mechanical structure with the rest of the team. Drew and cut acrylic boards for food containers, load cells and connecting pieces.
7. Designed poster with Ruihan Chen.

Zhiyu Wu

1. Determined to use I2C to transfer data between ssd1306 oled display and Nucleo and develop some basic functions to use the display including SetCursor, write in char/string and UpdataScreen with Ruihan Chen.
2. Overcame the difficulties of connecting the load cell to hx711 chip and then transferring data back to Nucleo through I2C to let load cell work stably with Ruihan Chen.
3. Configured the inner circuit of the keypad by connecting to power and observing the output signal of each channel.
4. Contributed to the design and realization of stopping the servo to a certain degree using switch.
5. Configured a timer interrupt to send data to the SD card every 24 hours.
6. Contributed to integrating the whole project, including integrating all functions into a main program, and then designed and realized the UI on the display and usage of all keys on keypad with Ruihan Chen; constructed the mechanical structure with all other group members; reinforced all connections in circuit.
7. Recorded a demo video to introduce functions of the project just in case there are unexpected issues.

Zichun Wang

1. After trying hard to use Nucleo L4R5ZI-P to communicate with SD card adapter, determined to use the Blue pill (STM32F103 board) to communicate with SD card adapter through SPI with Yiran Gu; found FatFs library function and wrote SD card read and write functions we needed.
2. Used UART DMA to communicate between the Blue pill and Nucleo board with Yiran Gu; designed the fixed data transmission format with all the other group members; debug the SD card read and write functions and communication between two boards with Yiran Gu.
3. Contributed to the configuration of the water pump and 360˚ servo with Yiran Gu.
4. Helped contribute to the design and realization of stopping the servo to a certain degree and adjusted the device carefully with Yiran Gu to make it work.
5. Designed the mechanical structure of our pet feeding device with other group members; helped construct the mechanical structure with other group members.

### **Weighted Member Contribution**

Ruihan Chen: 25%

Yiran Gu: 25%

Zhiyu Wu: 25%

Zichun Wang: 25%

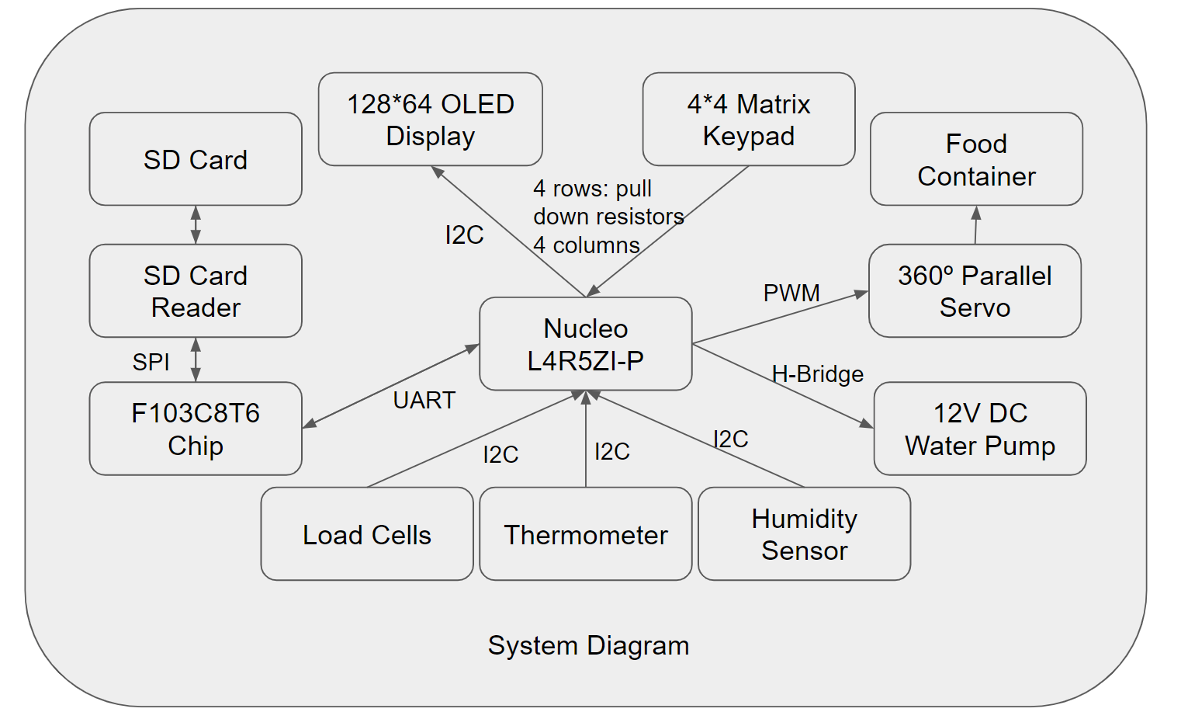


Figure 1: Component Diagram

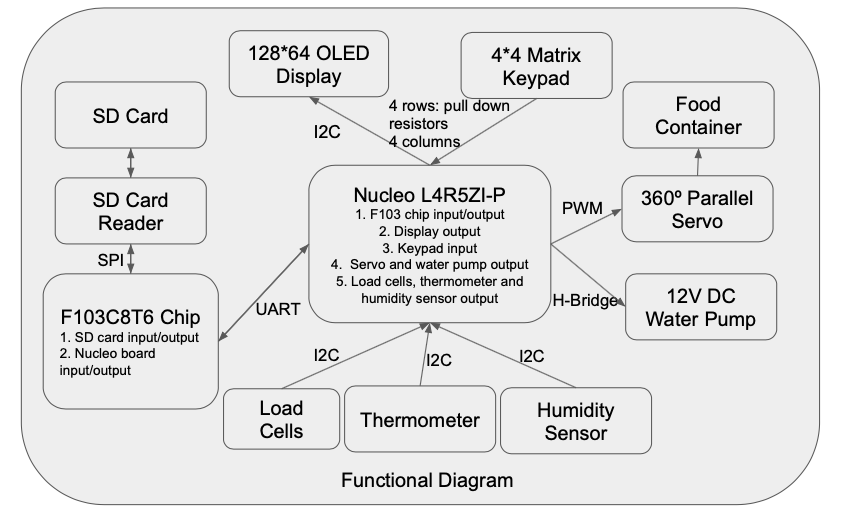


Figure 2: Functional Diagram