

Happy Pet's Testing Plan

Logan Swoyer, Sriram Vishnubhotla

February 27, 2025

1 Testing

1.1 Servo Motor

For the servo it takes input from the MSP-430 as a pulse with PWL signals, this pulse communicates how far to turn the motor. To send the signal we give the pin an integer value that will correspond to the width of the pulse in the duty cycle. This will need to be calibrated so we know the exact angles corresponding to the width of the pulse.

1.2 Buzzer

For testing the buzzer since it is simple we should only have to ensure that the pins are correctly connected to the micro controller. Then once we test we should be able to give a signal to the pin and hopefully hear an output sound.

1.3 Joystick

For the joystick it outputs right-left and up-down information. The right-left values can be interpreted as holding the joystick far right will be a value of 1023 and holding the joystick far left will give a value of 0. Therefore not moving the joystick should give a value close to 511. This logic is also the same for the up-down output, where far up corresponds to 1023 and far down corresponds to 0.

As well as the right-left and up-down outputs, the joystick has a built in select button. This button, if working correctly, should output a 1 or 0 if pressed or not pressed.

1.4 Load Cell Amplifier

For the load cell amplifier we will need to plug in our separate load cell into the amplifier. This is done using the red, black, yellow, green, and white ports, these pins will provide output from the load cell to communicate with the amplifier. Then we should be able to use UART communication to read these values with a serial print and begin the calibration setup. If we are able to read values and

they change when we apply weight to the load cell, than our setup should be correct.

1.5 OLED Display

For the OLED display there is an example we found that provides code for testing the display. Example code Using this code we can initialize the pins and we would use the code in order to write some example characters to the screen.

1.6 Ultrasonic Detector

For the proximity detector it works by sending a pulse, then we check the amount of time for the pulse to come back to sensor. Then we calculate the distance with the formula provided by the datasheet that is $distance(cm) = \mu s / 58$. So in order to test that it is working as expected we can measure out 5cm and place the sensor and an object that distance apart, if the reading is around 5cm we have a working part.

1.7 Wifi Module

In order to get the wifi module to work we need some software to write to it. The software used in an example we found was Toolchain to compile code, CMake and Ninja to build a full Application, and ESP-IDF that contains API for ESP32-C6. Once we have the software installed we can flash the firmware for the ESP-32, then we have to write code for the ESP32 and the MSP-430 in order to communicate between the two. (We are thinking of just using the ESP32 but for the sake of this deliverable we are still including the MSP-430).