HARDWARE UNIT TESTS REPORT - IOT PROJECT G05- SMART DOORBELL

|  | **Hardware name** | **Test results** | **Test code** |
| --- | --- | --- | --- |
| 1 | max98357 audio amplifier& Button | After pressing the button the there is a sine wave signal (pure ture) generated at a frequency of 440 Hz.We heard a continuous voice | [audio\_amplifier\_button](https://drive.google.com/drive/folders/1kaOtp-eXC2VN9XXh2f5FNGEeYXXZSa1y)  https://github.com/ferasskhoury23/IOT-project/tree/master/UNITTESTS/audio\_amplifier\_button |
| 2 | HCSR-04 -distance sensor | We measured distance and it has changed as we moved the body .  We used this equation to get the distance output:  “ float distance = (duration \* 0.0343) / 2;”  **\*EXPLANATION ABOUT THE EQUATION IN NOTE 1** | [HCSR04](https://drive.google.com/drive/folders/1o_Ctm7fG41GEnNBKIEZU4HIeAqUz1hO2)  https://github.com/ferasskhoury23/IOT-project/tree/master/UNITTESTS/HCSR04 |
| 3 | 1.3 inch MONOCHROME OLED display 128\*64 with SH1106 controller | We tested the display by running the given test from github .  first we used the i2c scanner to verify the screen address . we got 0x3C . | [display\_128x64\_i2c](https://drive.google.com/drive/folders/1vmAJzGs47JWbXMYi9IO6FXsobpt65o-V)  https://github.com/ferasskhoury23/IOT-project/tree/master/UNITTESTS/display\_128x64\_i2c |
| 4 | INMP411S2 microphone | **PROBLEM:we used the provided tests in the recommended links , and there was no response from the microphone (we think it’s ruined..)** |  |
| 5 | keypad | **We’ve tested the keypad and it worked- after pressing each button in the keypad it appeared on the terminal** | [keypad\_test](https://drive.google.com/drive/folders/1TbpdcYX-cd3OkWaNyCGsgsqI2CB89eNL)  https://github.com/ferasskhoury23/IOT-project/tree/master/UNITTESTS/keypad\_test |
| 6 | esp32-cam | **we succeeded to run a basic we camera server from the examples code in the arduino IDE .  the controller connects to the internet , starts a local server , and we can open the server address (IP) and see a live stream from the camera .**  **read note 7 to learn how to wire esp32-cam** | [CameraWebServer](https://drive.google.com/drive/folders/1LeywfptIGuqeoh5GmzFMj4f8EQ8Un91f)  https://github.com/ferasskhoury23/IOT-project/tree/master/UNITTESTS/CameraWebServer |

**Notes:**

**NOTE 1:**

Variables used for the distance equation:

**duration:**

This is the time (in microseconds) that the ultrasonic pulse takes to travel from the sensor to the object and back.

It is measured using the pulse IN function, which gives the time for which the ECHO pin remains HIGH.

**0.0343:**

This is the speed of sound in air at standard conditions, given in centimeters per microsecond:

Speed of sound=343m/s=0.0343 cm/µs

Speed of sound=343m/s=0.0343cm/µs

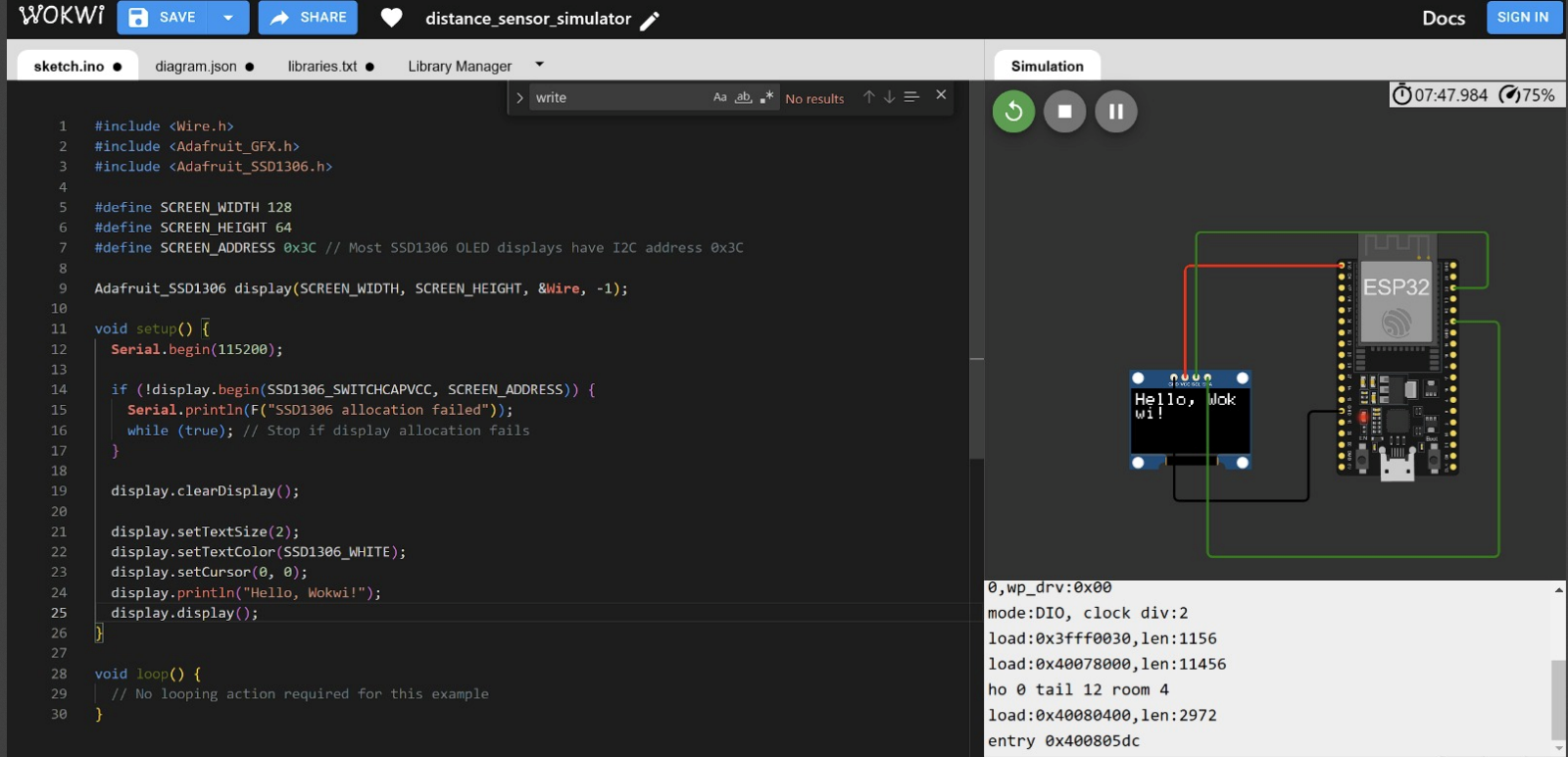
**/ 2:**

The ultrasonic wave travels to the object and back, so the measured time includes the round trip.

Dividing by 2 gives the one-way distance (from the sensor to the object).

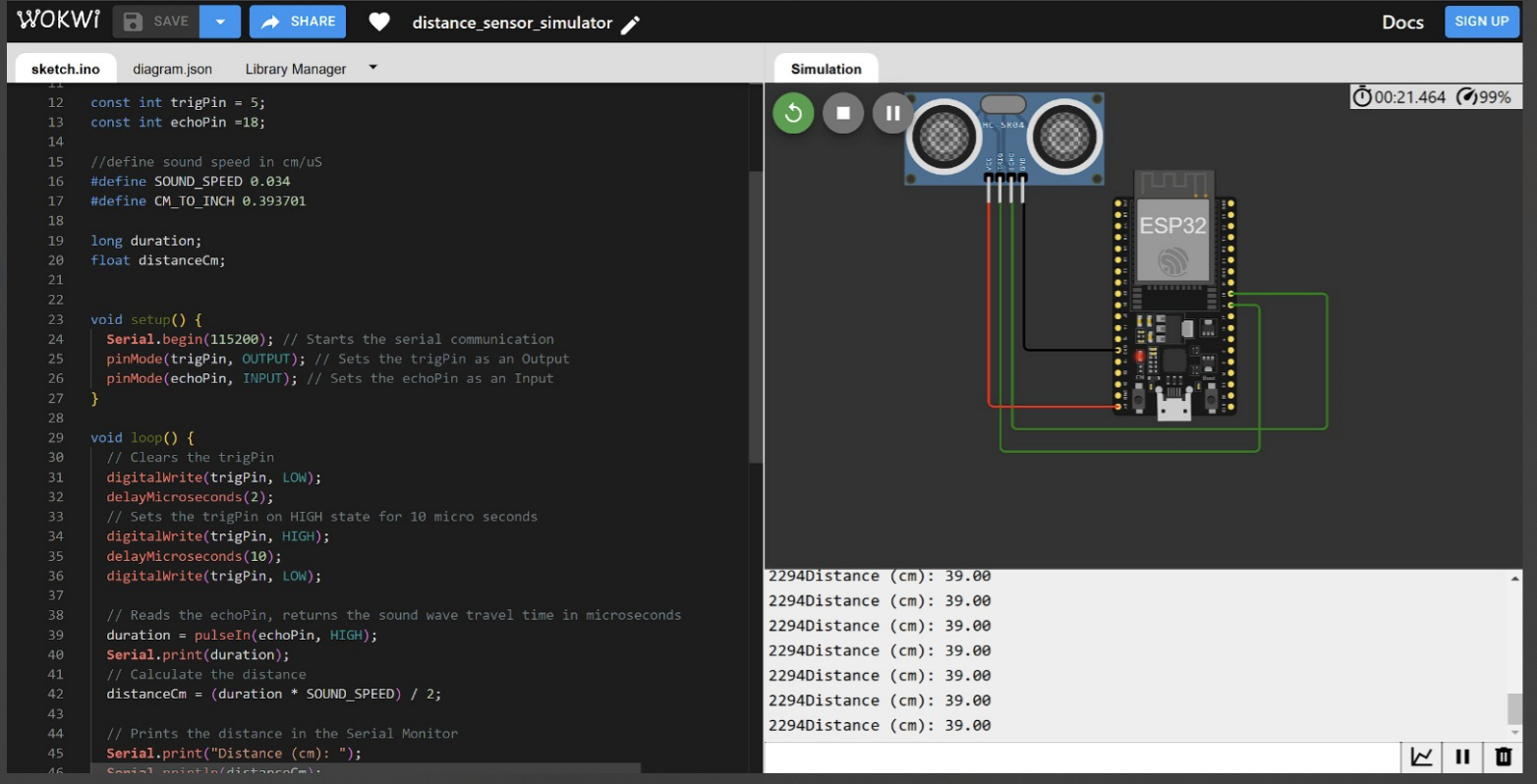
| esp32 | display |
| --- | --- |
| GND | GND |
| 3.3V | VCC |
| D22 | SCK |
| D21 | SDA |

**Note2: display simulation**

****

| esp32 | HCSR-04 |
| --- | --- |
| GND | GND |
| 5V | VCC |
| D18 | echo |
| D5 | trig |

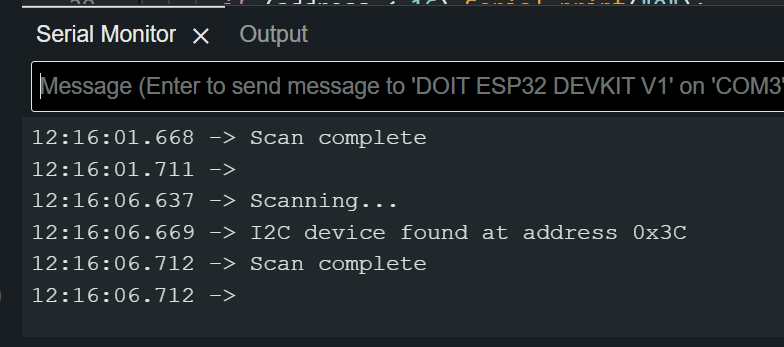
**Note3: distance sensor simulation**

****

**Note4:**

**what is IS2:**I2S (Inter-IC Sound) is a protocol designed specifically for transmitting audio data between digital audio devices. It is commonly used to connect microcontrollers (like the ESP32) with audio devices such as DACs (like the MAX98357), ADCs, or audio codecs.

for example : we used the scanner to detect the display address



**Note5:**

The pulseIn() function reads a HIGH or a LOW pulse on a pin. It accepts as arguments the pin and the state of the pulse (either HIGH or LOW). It returns the length of the pulse in microseconds. The pulse length corresponds to the time it took to travel to the object plus the time traveled on the way back.

**esp32-cam microcontroller :**

**Note6:**

To connect the board in download mode , hold the IOO button and then connect the usb to the computer.

Then after uploading the code , unplug the usb and replug it , so the controller will start running the code that we just uploaded automatically.

When running the CameraWebServer test , make sure to use a 2.4ghz internet connection . and make sure to define the correct camera model , which is CAMERA\_MODEL\_AI\_THINKER .

**Note 7:**

We use FTDI to connect the controller with the computer via usb . the wiring should be as follows :

| esp32-cam | FTDI |
| --- | --- |
| GND | GND |
| 5V | VCC |
| UOR | TX |
| UOT | RX |