

Prototype Atmospheric Sensor Module

Draft Specification for Prototype Atmospheric Sensor
Module Powered By ESP32S Board

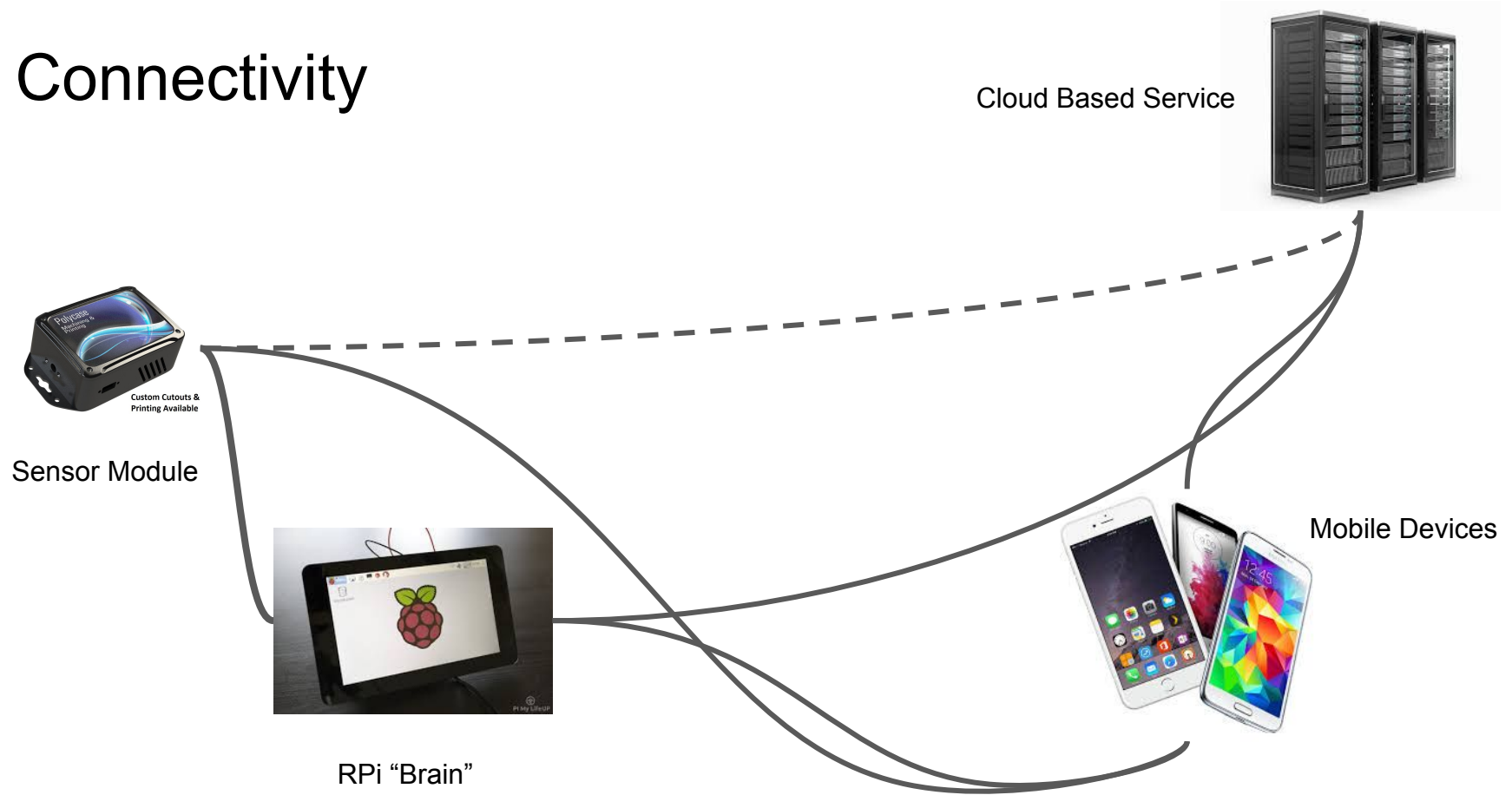
Created: 06/08/2017

Updated: 07/28/2017

Description

The prototype atmospheric sensor module will gather atmospheric/plant growth related data within indoor growing environments. This data will then be transmitted over WiFi to a centralized computer located in the same location, or directly to cloud based storage and visualization platform. The overall aim of this sensor module is to serve as development platform for data collection and sensor evaluation in order to inform the design of the production sensor module.

Connectivity



Supported Hardware/Sensor Modules

- ESP32S - [ESP-32S Development Board](#) (\$10)
- Temp - DHT22 (<https://www.adafruit.com/product/385>) (\$9)
- Humidity - DHT22 (<https://www.adafruit.com/product/385>) (\$9)
- Light level (lux) - TSL2561 (<https://www.adafruit.com/product/439>) (I2C) (\$6)
- RGB sensor - <https://www.sparkfun.com/products/12829> (\$8)
- CO₂ concentration - K30 <https://www.co2meter.com/products/k-30-co2-sensor-module>) (\$90)
- CO₂ concentration MG811 - [eBay Link](#) (\$35)
- OLED Screen - [Amazon Link](#) (\$7)
- Load Cell - <https://www.sparkfun.com/products/10245> (\$10)
- Load Cell Amplifier - <https://www.sparkfun.com/products/13879> (\$10)
- Moisture Sensor - <https://www.sparkfun.com/products/13322> (\$5)
- Waterproof Temperature Sensor - <https://www.adafruit.com/product/381> (\$10)

Enclosure and Battery Holder

- Polycase XR46F - <https://www.polycase.com/xr-46f> (\$10)
- 9V Battery Holder - [Amazon Link](#) (\$5)

Max Total Cost: \$225 + (Assembled Board Cost)

PCB Layout, must fit into XR-46F enclosure

9V Battery Case

2, DHT22 modules

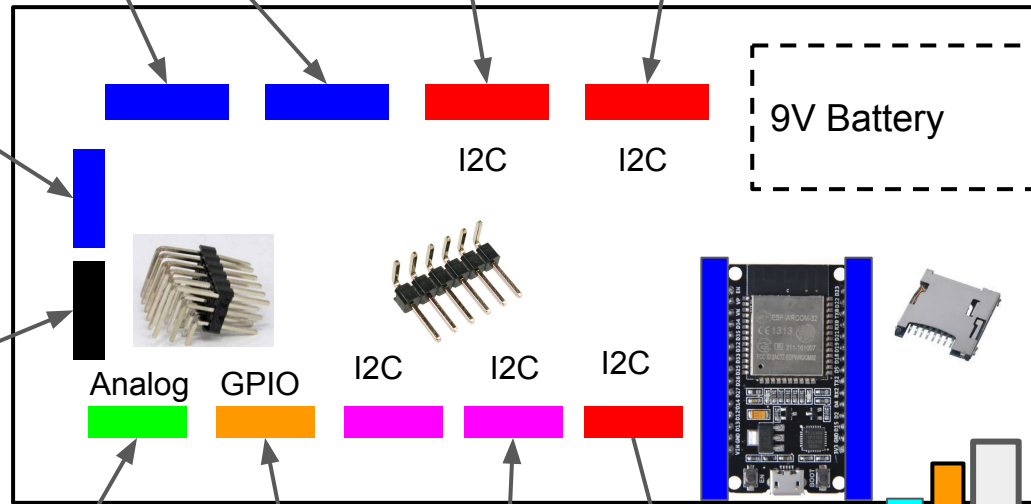


Custom Cutouts & Printing Available



SE-0018

CO₂ sensor is mounted onto case cover



Note:

1. Connect all Sensors/Devices Which support I2C comm over I2C.
2. Battery case will be mounted in enclosure cover.
3. **Blue** - female headers sensor modules directly plugs into
4. **Red** - male I2C header for connecting device using cable
5. **Green** - 4 90 deg analog (g/+5/S) pins protruding from enclosure
6. **Purple** - 2, 90 deg I2C male headers protruding from enclosure for connecting I2C devices.
7. **Orange** - 6, 90 deg 3 pin (g/+5/S) male GPIO headers protruding from enclosure

Bluetooth pairing switch, 3 position power switch, 2.1mm jack for 9VDC/1A

Board Specification (Draft)

1. Have dimensions that fit into XR-46F enclosure
2. Powered by either long lived 9V long lived battery, or 9V, 1A power adapter. The battery will be stored in a case mounted in the enclosure cover
3. Contain 2.1mm jack for 9VDC power supply
4. Contains 3 position slider switch for selecting between Battery/Off/DC power
5. Contain female header pins for mounting ESP32S development board.
6. Contains IC chips to expand analog and GPIO ports of the ESP32S if needed
7. Contains slider switch to turn device on and off when battery powered or plugged in.
8. Contains I2C male headers for light level and RGB sensors
9. Contains 2 female headers for two DHT22 temperature/humidity sensors Two sensors are used to validate results.
10. Contain female header for MG811 CO2 sensor.
11. Contain female header for K30 CO2 sensor. Previous attempts to connect using I2C failed so UART was used. However use of analog output OUT1 should be preferred.
12. Contains 4, 90 deg 3 pin (g/+5/S) male analog headers. One of this headers will be used by the soil moisture sensor
13. Contains 6, 90 deg 3 pin (g/+5/S) male GPIO headers. One of these will be used for the waterproof temperature sensor.
14. Contains 2, 90 deg I2C male headers protruding from enclosure. One of this headers will be used for the load cell/amplifier.
15. Contains male I2C header for connecting OLED screen. The screen will be attached to the enclosure cover
16. Micro SD card socket

ESP32S Draft Firmware Specification

1. Be developed as a project in [Arduino IDE](#) using C source code.
2. Pressing the pairing button to allow mobile devices to connect over bluetooth to enter configuration information which is stored in flash memory.
3. Once a device connects to over bluetooth the following ASCII commands are supported
 - a. Command to configure various settings. The details of those commands are still being developed but will include setting (1) SSID/password, (2) unique name/id, (3) How often sensor reading are taken (3) The URL/access credentials to send sensor data. There could be multiple URLs for sending data to, but the API will be consistent.
 - b. Command to get sensor data. The data will be in JSON format and contains the values for all supported sensors as well as unique ID and battery level. If a sensor is not connected then -1 should be used as a placeholder. The sensor values may be the raw values or preprocessed based on sensor.
 - c. Command to use 4 of the 6 GPIO pins as high/low pins for activating [relays](#)?
4. Send sensor data at specific intervals to URL specified in settings. The response from the server could include supported ASCII commands which should be executed. This is to allow for remote configuration and setting of the 2 GPIO to high or low based on certain sensor readings.
5. Store sensor data on micro SD card if present. This data can be retrieved with ASCII commands.
6. Supports OLED module for displaying sensor data including battery level. The display could be set to cycle through the different readings
7. Supports putting the device and sensor modules to sleep mode to conserve power when battery powered

What's Required and Deliverables

Requirement

- Good written verbal communications skills
- Experience with PCB design
- Experience with firmware programming using C
- Experience working with Arduino IDE

Deliverables

- PCB design which fits into the desired enclosure and has the desired functionality
- 2 prototype devices containing all supported sensors
- 3D file containing cutout information for PCB, sensors and pins
- Files and other information needed to manufacture up to 50 PCB boards
- Well document firmware as Arduino Project