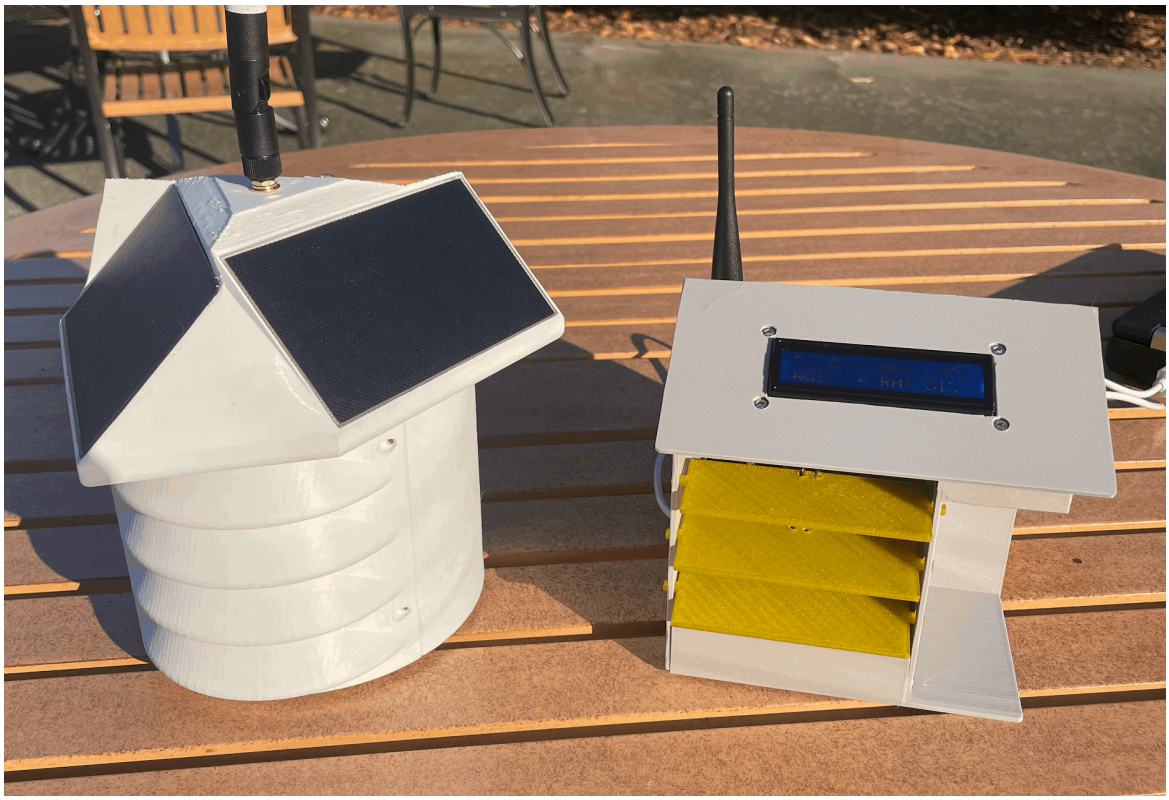
Solar Weather Station & Smart House

Ben Arnett

May 19th, 2025

User Manual



About the system

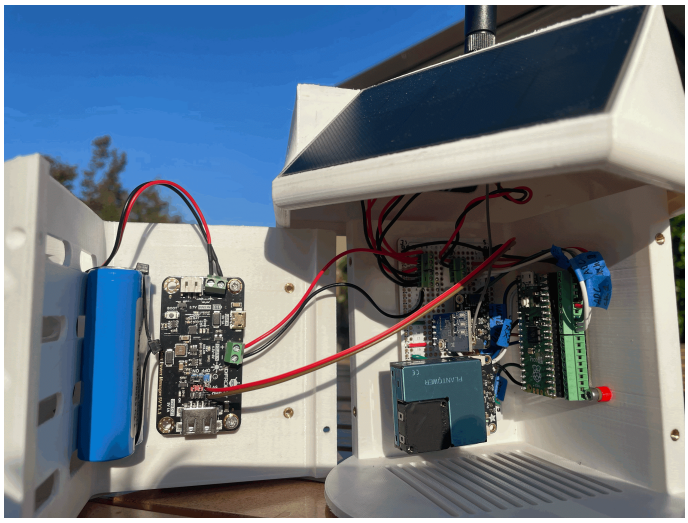
The Solar Weather Station and Smart House are two parts of a single system, with the goal of adjusting Smart Houses' "interior climate" controls based on data from the Weather Station. Both systems are controlled using a Raspberry Pi Pico 2 running CircuitPython. The goal of this project was to make an automated system using LoRa, a long-range low-power wireless data transmission technology.

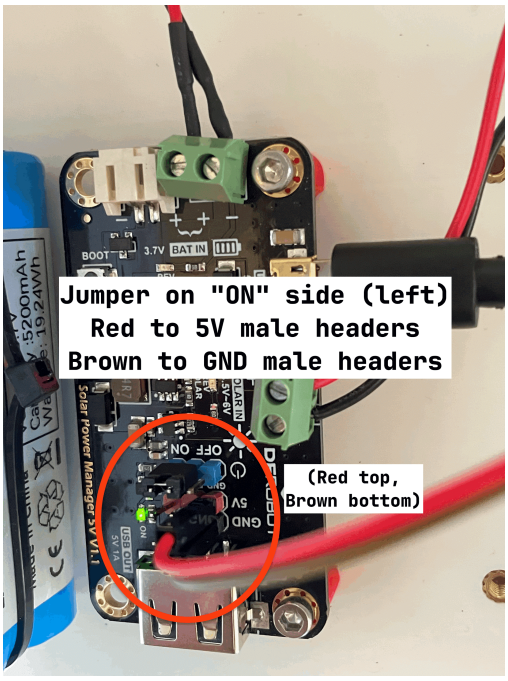
The Solar Weather Station, as the name suggests, consists of various weather data sensors to provide a comprehensive report of current conditions at its location. Data points it can acquire include: temperature, humidity, pressure, light, particulate matter, and AQI (estimated). The power supply of this system consists of three solar panels, a rechargeable battery, and a solar charge controller. In its final software configuration, the device is typically in a light sleep state, waking up every 15 minutes to acquire and send sensor readings before returning to a light sleep state.

The Smart House side contains LEDs, a servo, and a LCD character display. The LEDs simulate interior lighting, the servo opens/closes window shutters, and the LCD displays the latest received weather data.

System Setup / Use

To set up the weather station, first remove the M3 screws securing the body half with side air vents. There will be 4 screws to remove that attach to the other body piece as well as one screw attaching to the top (with the solar panels). Next, loosen (but do not remove) the screw that secures the other body piece to the top. Gently, with the side vents facing the left, tilt the top away from the body half with side air vents, and slowly pull the body halves apart in a manner as if opening a book. Inside, the left half with the side vents will contain the battery and solar charge controller, and the right half (still connected to the top) will contain the sensor array and Pi Pico (see below). On the top right of the sensor board, there will be a 2-pin header with a 2-wire (red and brown), female-female dupont connector coming out of it. Connect the red wire to the +5V pin rail on the solar charge controller and the brown wire to the GND pin rail of the solar charge controller (see below). If the solar power modules' on/off jumper (blue headers) is on the right (off) then move it to the left (on) and the green led indicator should turn on.



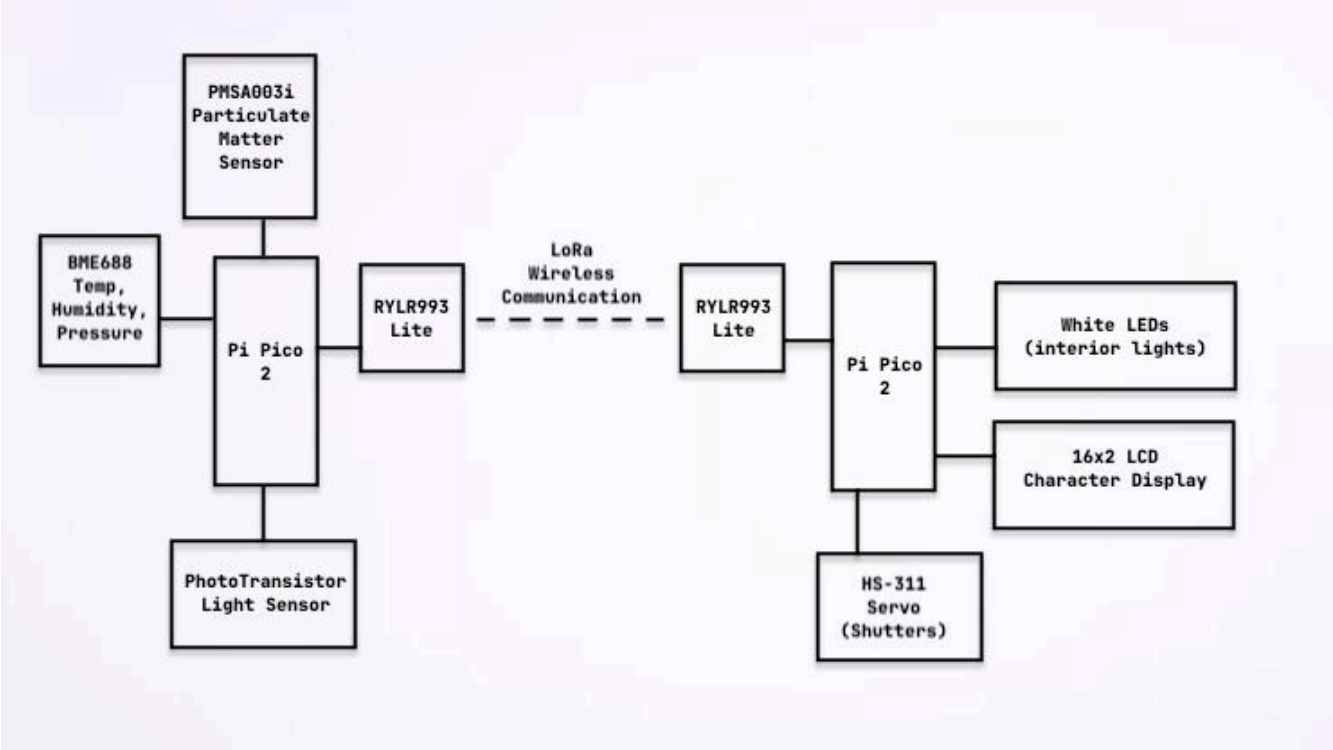


To set up the Smart House, connect a USB type-C cable to the port on the back. When looking at the house with the shutters facing away from you, this port will be on the right side, under & behind the antenna. The USB type-C cable should be connected to a standard 1A, 5V outlet.

Being a fully automated system pair, after both systems are powered on, no further interaction is needed. For best connection between the Weather Station and Smart House, place the Smart House near a window that provides a line of sight to the station. Note: LoRa connection range is affected primarily by obstacles and terrain in between modules i.e. the less stuff in between, the greater the range potential.

Top Level System Overview

Block Diagram:



Inputs / Outputs:

Weather Station:

I/O	Device	Transducer	Use	Interface
I	Phototransistor	Light	Ambient light measure	Analog, 0-3.3V, analogio
I	BME688	Resistive, Current	Temperature, humidity, pressure measurement	Digital, I2C, adafruit_bme680 library
I	PMSA003i	Optical/Laser	Particulate matter sensor > estimated AQI	Digital, I2C, adafruit_pm25 library
O	RYLR993 Lite	Radio Frequency	LoRa Wireless Communication	Digital, UART, AT+ commands

Smart House:

I/O	Device	Transducer	Interface
I	RYLR993 Lite (LoRa)	Radio Frequency	Digital, UART, uart.read()
O	White LEDs	Light	Digital, PWM, pwmio
O	HS-311 Servo	Motion	Digital, PWM, Servo library
O	16x2 LCD Character Display	LCD	Digital, I2C, LCD1602 module

Commands Used:

uart.write() uart.read() adcin.value pms.read() time.sleep() servo_a.angle lcd.set_cursor() lcd.write() lcd.clear() whtled.duty_cycle

Microcontroller Used:

I used the Raspberry Pi Pico 2 for both sides of this system. The primary reason for this was the need for UART protocol support in order to use the RYLR993 Lite LoRa modules. In addition, the Pico has two cores, low power support, and flexible IO configurability. Typically, Pi Picos are programmed using some form of Python. This added some complexity, however, data manipulation is very easy with Python compared to Basic programming on the Picaxe.

External Integrated Circuits / Modules:

- RYLR998 lite LoRa/LoRaWAN module
- DFRobot Solar Power Manager module
- Adafruit BME688 sensor module
- Adafruit PMSA003i particulate matter module

Github Repo:

Fabrication Techniques:

- 3D Printing
- Soldering