

MSR-1

mmWave CO2 Multisensor

- [Setup](#)
 - [Getting Started](#)
 - [Sensor Definitions](#)
 - [Bluetooth Tracking with MSR-1](#)
 - [General Tips](#)
 - [How To Change The Update Frequency Of Sensors](#)
- [Calibrating And Updating](#)
 - [Updating MSR-1 Firmware](#)
 - [How To Tune mmWave Using Home Assistant](#)
 - [How To Tune mmWave Using HLKRadarTool](#)
 - [MSR-1 Temperature & Humidity Offsets](#)
 - [CO2 Calibration For MSR-1](#)
- [Examples](#)
 - [MSR-1 Home Assistant Dashboard Examples](#)
 - [Using MSR-1 Buzzer](#)
 - [MSR-1 Automation Guide](#)
 - [MSR-1 Examples From GitHub](#)
 - [MSR-1 + Alarmo Home Security Install](#)
- [Additional Info](#)
 - [Exposed GPIO](#)
 - [mmWave Presence Sensor Comparison](#)
 - [Adding CO2 To MSR-1](#)
 - [Reducing LD2410 Log Entries In Home Assistant](#)

- [Troubleshooting](#)
 - [Manually Uploading Code Through ESPHome](#)
 - [Putting The MSR-1 In Boot Mode](#)

- [MSR-1 Reviews](#)
 - [Make It Work Tech MSR-1 Review](#)
 - [Smart Home Scene MSR-1 Review](#)
 - [DrZzs MSR-1 Review](#)
 - [BeardedTinker MSR-1 Review](#)
 - [Simon Says Home Assistant MSR-1 Review](#)
 - [MostlyChris MSR-1 Review](#)

Setup

Setup

Getting Started

Please refer to the [general getting started](#) article

Sensor Definitions

Once added to Home Assistant you can configure different settings for your sensor. Below is what each setting does.

Controls

- **RGB Light**
 - A RGB Neopixel. Click on the light bulb to change the color. Click on the toggle to turn on or off

Sensors

- **BME280 Humidity**
 - Humidity reading from the BME280. This is affected by the configuration variable "BME280 Humidity Offset". Changing the offset allows you to dial in the humidity to a known value.
- **BME280 Humidity Calibrated - Removed In v23.11.01.1**
 - ~~We calibrated the BME280 against a lab-rated reference sensor. We set the linear offset values in the YAML and this variable is the output.~~
- **BME280 Pressure**
 - Atmospheric pressure reading from BME280
- **BME280 Temperature**
 - Temperature reading from BME280. This is affected by the configuration variable "BME280 Temperature Offset". Changing the offset allows you to dial in the temperature to a known value.
- **BME280 Temperature Calibrated - Removed In v23.11.01.1**
 - ~~We calibrated the BME280 against a lab-rated reference sensor. We set the linear offset values in the YAML and this variable is the output.~~
- **CO2**
 - True CO2 reading from the SCD40. This will be Unknown if you do not have the CO2 module. This can be calibrated following this guide but does come precalibrated:
[Here](#)
- **ESP Temperature**
 - This is the temperature of the internal ESP chip. Think of it like your measured CPU temp on your PC
- **LTR390 Ambient Light - Removed. Value wasn't useful**
 - Light level measured by LTR390

- **LTR390 Light**
 - Light level measured in lux by LTR390
- **LTR390 UV - Removed. Value wasn't useful**
 - UV level measured by LTR390
- **LTR390 UV Index**
 - UV index measured by LTR390
- **Radar Detection Distance**
 - The last detected distance by the radar. This will stay at the last known value so sometimes can be misleading
- **Radar Move Energy**
 - The amount of movement measured by the LD2410B. Faster movements have higher %
- **Radar Moving Target**
 - Does the radar have a moving target it is tracking
- **Radar Still Distance**
 - The last measured distance of a still target. It will hold the last value so sometimes can be misleading
- **Radar Still Energy**
 - The energy of the current still target
- **Radar Still Target**
 - Does the radar have a still target
- **Radar Target**
 - Does the radar have a still or moving target. Good for triggering automation.
- **Radar Zone 1 Occupancy**
 - This is a configurable zone. Think of zones like distances from the radar unit. Zone 1 might be from 0 cm to 120 cm from the sensor. This is telling you if there is someone in that zone. The zones can be defined in the configuration section with "Radar End Zone 1"
- **Radar Zone 2 Occupancy**
 - Same as zone 1 but just the second zone from the sensor
- **Radar Zone 3 Occupancy**
 - Same as zone 1 & 2 but with the third zone.
- **SCD40 Humidity - Removed In v23.11.01.1**
 - This is the humidity reading from the SCD40. Will be unknown if you do not have the CO2 module
- **SCD40 Temperature - Removed In v23.11.01.1**
 - This is the temperature reading from the SCD40. Will be unknown if you do not have the CO2 module

Configuration

- **BME280 Humidity Offset**
 - Allows you to adjust the reading of the "BME280 Humidity" sensor to match a known value

- **BME280 Temperature Offset**

- Allows you to adjust the reading of the “BME280 Temperature” sensor to match a known value

- **CO2 Calibration Number**

- See calibrating CO2: [Here](#)

- **ESP Reboot**

- Performs a restart of the sensor

- **Factory Reset Radar**

- Sets the radar's move thresholds back to their original values from the manufacturer

- **g0-g8 Move & Still Threshold**

- Please refer to the radar tuning guide: [Here](#)

- **Radar Control Bluetooth**

- This allows you to turn on the LD2410's Bluetooth. This allows you to connect to the HLK Radar phone app if you wanted to upload new firmware to the radar unit (Not the MSR-1 in general, just the radar chip)

- **Radar Distance Resolution**

- Best to keep on 0.75m in many cases. If you change it to 0.25m the first few gates become very very sensitive and the maximum detection distance shrinks a lot.

- **Radar Zone 1 Start**

- **This sets the starting distance for Zone 1 in cm. This is the distance from the sensor to the start of Zone 1**

- **Radar End Zone 1**

- This defines “Zone 1” of the radar. It is a distance from the sensor that specifies what “Zone 1” is. It connects to the “Radar Zone 1 Occupancy” sensor. So if this number is set to “10” that means from 0 to 10 centimeters from the sensor is zone 1.

- **Radar End Zone 2**

- Same as Zone 1. This defines where zone 2 ends

- **Radar End Zone 3**

- Like Zone 2, this defines where zone 3 ends

- **Radar Max Move Distance**

- Maximum distance gate for movement detection. Value between 2 and 8 inclusive

- **Radar Max Still Distance**

- Maximum distance gate for still detection. Value between 2 and 8 inclusive. Defaults to 8.

- **Radar Timeout**

- The time in seconds that the radar's presence will stay high after the target is lost.

- **SCD40 Humidity Offset - Removed In v23.11.01.1**

- Allows you to adjust the reading of the “SCD40 Humidity” sensor to match a known value.

- **SCD40 Temperature Offset - Removed In v23.11.01.1**

- Allows you to adjust the reading of the “SCD40 Temperature” sensor to match a known value

Setup

Bluetooth Tracking with MSR-1

Please refer to the [general Bluetooth tracking](#) article

General Tips

MSR-1

[apollo automation-MSR-1-presence-multisensor-PCB-components-layout-1024x768.jpg.webp](#)

(Picture from [Smart Home Scene](#))

Light Sensor (LTR-390UV)

[Light holes.png](#)

- When mounting the MSR-1 be sure to position the device so that the two large holes are not covered
 - This allows more light to enter and will ensure better accuracy
- The onboard RGB LED will trigger the light sensor
 - Be cognizant of this when making automations based on light/LUX

Mounting

[ID2410 Zone map.png](#)

[ID2410 Gates.png](#)

[ID2410_mounting_hor_1.jpeg](#)

Gate and FOV Visualization

[ID2410 table.png](#)

[MSR-1 radar map.png](#)

[Radar gates Colored.png](#)

- FOV angle -60 to 60
- Gate images above are using a Radar Distance Resolution of 0.75m
- **Gates are pre-defined by the radar module and are in meters (m)**

- **Zones are user-configurable and are in centimeters (cm)**

Gate, Zones and RR.png

Image not found or type unknown

Increased ESP Temperature

If you are experiencing higher than normal ESP temperatures ~140+ degrees F then changing the wifi power save mode option might help decrease the temperature. Here is the link to the ESPHome WiFi Component [Power Save Mode](#).

Power Save Mode

The WiFi interface of all ESPs offer three power save modes to reduce the amount of power spent on WiFi. While some options can reduce the power usage of the ESP, they generally also decrease the reliability of the WiFi connection, with frequent disconnections from the router in the highest power saving mode.

NONE (least power saving, Default for ESP8266)

LIGHT (Default for ESP32)

HIGH (most power saving)

```
wifi:  
  # ...  
  power_save_mode: none
```

The code above can be added to the devices .yaml through the ESPHome addon edit button.

(Thank you for the suggestion, Brian!)

(Referenced from ESPHome website)

Setup

How To Change The Update Frequency Of Sensors

[How To Change The Update Frequency Of Sensors](#)

Calibrating And Updating

Updating MSR-1 Firmware

Please refer to the [general firmware update](#) article

How To Tune mmWave Using Home Assistant

If you're experiencing false triggers, we recommend using Radar Engineering Mode (REM) to monitor the gate energy and adjust the gate threshold to eliminate them.

Here is a quick introduction video of the Id2410b gates and zones.

<https://www.youtube.com/embed/6VrTfaFyMPk?si=pk62XBQ98m8E5uYu>

Here is a video of how to tune the mmWave sensor using radar engineering mode,

https://www.youtube.com/embed/w_Gq62Edsnc

1. Open Home Assistant
2. Navigate to Settings>Devices & services>ESPHome>Select the MSR-1
3. Scroll down to the Configuration section
4. Turn on Radar Engineering Mode (REM)

[Configuration.png](#)
Image not found or type unknown

5. Scroll down to the Diagnostic section and you will see that REM shows the move and still energy for gates 0-8

[Diagnostic.png](#)
Image not found or type unknown

6. The gates are different distances from the sensor

[Id2410 table.png](#) [MSR-1 radar map.png](#)
Image not found or type unknown Image not found or type unknown

7. Moving the gate still and move threshold slider to the right increases the amount of energy needed to trigger the sensor. Do this if you want the gate to be less sensitive.

[High Threshold.png](#)
Image not found or type unknown

8. Moving the gate still and move threshold slider to the left decreases the amount of energy needed to trigger the sensor. Do this if you want the gate to be more sensitive.

[Low Threshold.png](#)
Image not found or type unknown

Alternate Method

Thanks to [MakeltWorkTech](#) for this method!

"I ended up maxing out the sliders on all gates and then bringing them down just enough to pick up human presence. Definitely easier than using the LD2410B app."

Example

- You may have an open-concept kitchen and living room, and you want the MSR-1 to activate your under-cabinet lights only when you're in the kitchen.

Here's how:

1. Stand in the desired trigger locations.
2. Observe the gate energy.
3. Adjust the gate threshold slider to the right, increasing the energy required to trigger the mmWave sensor. This ensures that your kitchen lights only come on when you're actually in the kitchen, not just walking by in the living room. Also, you can lower the gate threshold in the kitchen by moving the slider to the left. This makes the mmWave sensor more sensitive, even when standing still. This way, you avoid having the lights go off while reading a recipe or doing the dishes.

- From [clarinetJWD](#), "Engineering mode is what I was missing at first. I did exactly that, and now detection is basically perfect! My under cabinet lights now come on to max brightness whenever someone is in the kitchen, and return to their previous value when they leave. Next up is replacing the really flaky motion switch in my garage so it stops shutting off when I'm doing a project at the workbench! Thanks, these are really good."

References

- <https://youtu.be/dAzHXpP3FcI?t=431>
- <https://community.home-assistant.io/t/ld2410-esphome-tips/477058/316>
- <https://www.youtube.com/watch?v=I212Lvo1R6s>

How To Tune mmWave Using HLKRadarTool

Auto-Calibration (Average, Maximum, and Intelligent)

1. Download the HLKRadarTool app for either [Android](#) or [Apple](#)
2. Ensure the mmWave radar you want to tune has Radar Control Bluetooth turned on. Home Assistant > Settings > Devices & services > ESPHome Devices > Select Device > Scroll down and toggle on Radar Control Bluetooth

[Radar Control Bluetooth.png](#)

3. Open the app and select your device

[Find Device.png](#)

4. Turn on Engineering Mode

[EM.png](#)

5. Select More and then select Parameter settings

[More and PS.png](#)

6. Select Detect background noise

[DBN.png](#)

7. Insert Delay detection and Detection time values. (For iPhone users Delay detection cannot be 0). Then select Start. This will give us our reference values so we can auto-calibrate the mmWave sensor.

[DD, DT and Start.png](#)

8. Select Back to navigate to the Parameter settings. Then select one of the auto-calibration buttons.

Average

- Sets the gate sensitivity to the average move and still energy

[Average.png](#)
Image not found or type unknown

Maximum

- Sets the gate sensitivity to the maximum move and still energy

[Max.png](#)
Image not found or type unknown

Intelligent

- Sets the gate sensitivity equal to or just above the maximum move and still energy

[Intelligent.png](#)
Image not found or type unknown

9. Now your mmWave should be tuned to your environment!

Manual Calibration

1. Follow steps 1-5 above

2. Select Motion or Static sensitivity, change it to your desired value, and select Set

[Manual.png](#)
Image not found or type unknown

3. Now your mmWave should be tuned to your environment!

MSR-1 Temperature & Humidity Offsets

BME280 & SCD40 Sensors: Overcoming Temperature & Humidity Reading Challenges

The BME280 and SCD40 sensors are known for their precision in measuring temperature and humidity. However, like all sensor systems, they can sometimes provide inaccurate readings due to various factors. In the case of the BME280 and SCD40 sensors, one significant challenge arises from the heat produced by the ESP chip, which can alter the environment inside its enclosure, thereby skewing the readings.

Adjusting the BME280 and SCD40 Temperature Offset

For users who want to fine-tune their sensors, the BME280 and SCD40 Temperature Offset entities can be manually adjusted to match the conditions in their home. The offset values are subtracted from the raw temperature & humidity readings in the firmware to update the sensor readings in the home assistant entity. For example: `scd40_temperature` entity = raw scd40 temperature reading - `scd40_offset`.

By default, these offsets are preset to values based on our NIST certified thermometer, it's important to note that these values are calibrated for our environment. They might not be accurate for all settings. Therefore, by using a reference thermometer, users can adjust the difference between the readings to get a more accurate representation.

Users will also notice the `bme280_humidity_calibrated`, `scd40_humidity_calibrated`, `bme280_temperature_calibrated`, and `scd40_temperature_calibrated` entities. These values utilize the linear filter in the ESPHome firmware to adjust the readings based on our collected data. Again, due to environmental differences, these might not always be precise.

Modeling the Relationship Between Sensors

Another approach to getting accurate readings is to model the relationship between the ESP temperature and the other sensors compared against a reference temperature. This can be achieved by creating a template sensor in Home Assistant that employs a decision tree or our regression coefficients.

Here's an example that can be added to a configuration YAML:

```
sensor:
  - platform: template
    sensors:
      estimated_reference_temperature:
        friendly_name: "Estimated Temperature"
        unit_of_measurement: '°C'
        value_template: >
          {{
            0.3228 * states('sensor.apollo_msr_1_a79e24_esp_temperature') | float +
            0.8702 * states('sensor.apollo_msr_1_a79e24_scd40_temperature') | float +
            -0.1285 * states('sensor.apollo_msr_1_a79e24_bme280_temperature') | float +
            -0.0491 * states('sensor.apollo_msr_1_a79e24_scd40_humidity') | float +
            0.0851 * states('sensor.apollo_msr_1_a79e24_bme280_humidity') | float +
            -17.4840
          }}

      estimated_reference_humidity:
        friendly_name: "Estimated Humidity"
        unit_of_measurement: '%'
        value_template: >
          {{
            -1.2468 * states('sensor.apollo_msr_1_a79e24_esp_temperature') | float +
            -2.1959 * states('sensor.apollo_msr_1_a79e24_scd40_temperature') | float +
            2.9604 * states('sensor.apollo_msr_1_a79e24_bme280_temperature') | float +
            0.2380 * states('sensor.apollo_msr_1_a79e24_scd40_humidity') | float +
            1.8283 * states('sensor.apollo_msr_1_a79e24_bme280_humidity') | float +
            40.1686
          }}
```

For users who want to gather their own data with a reference sensor and MSR-1 within their home, we recommend logging data to a CSV using the following YAML entries:

```
#Configuration.yaml
notify:
  - platform: file
    name: sensor_csv_log
```

```
filename: /config/sensor_log.csv
```

```
timestamp: False
```

```
#Automations.yaml
```

```
- id: log_sensor_data_to_csv
```

```
alias: "Log Sensor Data to CSV"
```

```
trigger:
```

```
platform: time_pattern
```

```
minutes: "/1" # Log data every minute to match reference sensor
```

```
action:
```

```
- service: notify.sensor_csv_log
```

```
data_template:
```

```
message: >
```

```
{{ now().strftime('%Y-%m-%d %H:%M:%S') }},
```

```
{{ states('sensor.apollo_msr_1_a79e24_esp_temperature') | default('NA') }},
```

```
{{ states('sensor.apollo_msr_1_a79e24_scd40_temperature') | default('NA') }},
```

```
{{ states('sensor.apollo_msr_1_a79e24_bme280_temperature') | default('NA') }},
```

```
{{ states('sensor.apollo_msr_1_a79e24_scd40_humidity') | default('NA') }},
```

```
{{ states('sensor.apollo_msr_1_a79e24_bme280_humidity') | default('NA') }},
```

```
{{ states('number.apollo_msr_1_a79e24_bme280_temperature_offset') | default('NA') }},
```

```
{{ states('number.apollo_msr_1_a79e24_scd40_humidity_offset') | default('NA') }},
```

```
{{ states('number.apollo_msr_1_a79e24_bme280_humidity_offset') | default('NA') }},
```

```
{{ states('number.apollo_msr_1_a79e24_scd40_temperature_offset') | default('NA') }},
```

```
{{ states('sensor.apollo_msr_1_a79e24_uptime') | default('NA') }}
```

The Interrelation of Temperature and Humidity

It's important to understand that temperature and humidity share an interdependent relationship. When the air temperature rises, its capacity to hold moisture increases, which can decrease relative humidity levels. Conversely, when the temperature falls, the air's capacity to hold moisture decreases, leading to increased humidity. This relationship plays a significant role in how sensors detect and interpret readings, making it even more crucial to ensure accuracy.

Advanced Accuracy with GPIO

For those seeking the highest accuracy, an advanced solution is available. The exposed GPIO on the back of the board can be utilized to solder a temperature probe. This modification can dramatically improve both temperature and humidity readings, providing data that's as accurate as possible.

Calibrating And Updating

CO2 Calibration For MSR-1

Please refer to the [General CO2 Calibration](#) article

Examples

MSR-1 Home Assistant Dashboard Examples

Radar Scatter Plot With Apex Charts For MSR-1

[JPE Apex Charts 3.png](#)
Image not found or type not shown

```
type: custom:apexcharts-card
header:
  show: true
  title: MSR-1 Radar Distance
  show_states: true
  colorize_states: true
apex_config:
  legend:
    position: top
    horizontalAlign: center
    offsetX: -80
    offsetY: 0
experimental:
  brush: true
brush:
  selection_span: 4h
graph_span: 24hr
chart_type: scatter
series:
  - entity: sensor.apollo_msr_1_a79e14_radar_detection_distance
    name: Detection
    stroke_width: 1
    color: green
    show:
      in_brush: true
      extremas: true
```

```
- entity: sensor.apollo_msr_1_a79e14_radar_moving_distance
  name: Moving
  stroke_width: 1
- entity: sensor.apollo_msr_1_a79e14_radar_still_distance
  name: Still
  stroke_width: 1
```

CO2 Charts

Home Assistant Sensor

[CO2 Graph.png](#)
Image not found or type unknown

```
graph: line
type: sensor
entity: sensor.apollo_msr_1_a79e38_co2
detail: 2
name: Bedroom CO2
hours_to_show: 24
```

Custom: Plotly Graph Card

[Plotly CO2 Graph.png](#)
Image not found or type unknown

```
type: custom:plotly-graph
entities:
  - entity: sensor.apollo_msr_1_a79e38_co2
hours_to_show: 24
refresh_interval: 10
title: Bedroom CO2
```

Home Assistant Gauge

[CO2 Gauge.png](#)
Image not found or type unknown

```
type: gauge
entity: sensor.apollo_msr_1_a79e38_co2
needle: true
unit: ppm
min: 0
max: 3000
severity:
```


green: 0
yellow: 1000
red: 2000

Using MSR-1 Buzzer

This guide will walk you through using the buzzer on your MSR-1. You can also call the below service through automations in Home Assistant.

1. Open Home Assistant and navigate to developer tools in the bottom left

[Screenshot 2023-11-01 at 10.52.38 PM.png](#)

2. Click on services

3. Choose your MSR-1's buzzer (You can type in buzzer to find it easier)

4. Put a rtttl tone into the the string box

- Mario:

```
smb:d=4,o=5,b=100:16e6,16e6,32p,8e6,16c6,8e6,8g6,8p,8g,8p,8c6,16p,8g,16p,8e,16p,8a,8b,16a#,8a,16g,16e6,16g6,8a6,16f6,8g6,8e6,16c6,16d6,8b,16p,8c6,16p,8g,16p,8e,16p,8a,8b,16a#,8a,16g,16e6,16g6,8a6,16f6,8g6,8e6,16c6,16d6,8b,8p,16g6,16f#6,16f6,16d#6,16p,16e6,16p,16g#,16a,16c6,16p,16a,16c6,16d6,8p,16g6,16f#6,16f6,16d#6,16p,16e6,16p,16c7,16p,16c7,16c7,p,16g6,16f#6,16f6,16d#6,16p,16e6,16p,16g#,16a,16c6,16p,16a,16c6,16d6,8p,16d#6,8p,16d6,8p,16c6
```

- Cantina:

```
Cantina:d=4,o=5,b=250:8a,8p,8d6,8p,8a,8p,8d6,8p,8a,8d6,8p,8a,8p,8g#,a,8a,8g#,8a,g,8f#,8g,8f#,f.,8d.,16p,p.,8a,8p,8d6,8p,8a,8p,8d6,8p,8a,8d6,8p,8a,8p,8g#,8a,8p,8g,8p,g.,8f#,8g,8p,8c6,a#,a,g
```

- Huge list: <https://picaxe.com/rtttl-ringtones-for-tune-command/>
- This is a good site to test rttl tones: <https://adamonsoon.github.io/rtttl-play/>

5. Click on call service button in bottom right

Image not found. Type: unknown

Screenshot 2023-11-01 at 10.57.33 PM.png

Examples

MSR-1 Automation Guide

<https://www.youtube.com/embed/YbPrqlh1afU?si=Or35UY7WegrNy9Jz>

Examples

MSR-1 Examples From GitHub

<https://github.com/ApolloAutomation/MSR-1/tree/main/Examples>

Examples

MSR-1 + Alarmo Home Security Install

My grandmother wanted a security system that would alert her when there was movement in her home after she had gone to bed.

Additional Info

Additional Info

Exposed GPIO

[Screenshot 2023-11-18 at 6.33.00 PM.png](#)

The back of the Apollo MSR-1 has exposed GPIO for a variety of uses.

From left to right:

- SLC (Typo, should be SCL)
 - Used for I2C
- IO 5
 - GPIO 5 of ESP32-C3-Mini
- 5V
 - Used for powering the device, or powering a external sensor
- GND
 - Ground
- 3V3
 - Used to power external sensors. Can not power the MSR-1 from this
- SDA
 - Used for I2C

mmWave Presence Sensor Comparison

Link to excel file: [mmWave Presence Sensor Excel Spreadsheet](#)

Scroll table to right to see others

Column1	Apollo MSR-1	Apollo MSR-1 CO2	Everythi ng Presence One Kit	Everythi ng Presence Lite Kit	Screek Human Sensor 1U	Screek Human Sensor 2A	Aqara FP1	Aqara FP2	Seeed Studio mmWave Kit
Price (USD)	\$ 34.99	\$ 54.99	\$ 78.00	\$ 37.00	\$ 25.99	\$ 29.99	\$ 49.99	\$ 82.99	\$ 26.99
Motion Detectio n									
mmWave	LD2410B	LD2410B	SEN0395	LD2450	LD2410C	LD2450	BGT60TR 13C	IWR6843	Human Static Presence Module Lite
Detectio n Distance (m)	6	6	9	6	6	6	5	6 to 8	5

Detection Zones	3 zones and 9 gates (adjustable and distance based)	3 zones and 9 gates (adjustable and distance based)		x		x	x	x	
Target Tracking				x		x	x	x	
PIR			Panasonic						
Connection Method									
Wi-Fi	ESP32-C3-MINI	ESP32-C3-MINI	ESP32-WROOM-32	ESP32-WROOM-32	ESP32-S2	ESP32-C3		ESP32-WROOM-32U	ESP32-C3
Zwave							x		
Zigbee							x		
Bluetooth	ESP32-C3-MINI	ESP32-C3-MINI	ESP32-WROOM-32	ESP32-WROOM-32	ESP32-S2	ESP32-C3		ESP32-WROOM-32U	ESP32-C3
ESPHome	ESP32-C3-MINI	ESP32-C3-MINI	ESP32-WROOM-32	ESP32-WROOM-32	ESP32-S2	ESP32-C3			ESP32-C3
Case / Stand									
Case Included	x	x	\$ 8.00	x	x	x	x	x	x
Cable				x	x	x	x	x	x
Stand	\$4.99 (Optional)	\$4.99 (Optional)	Included with case purchase	x			x	x	
Outlet Mount	\$6.99 (Optional)	\$6.99 (Optional)							
Outlet Mount w/ Wall Plug	\$19.99 (Optional)	\$19.99 (Optional)							

Adding CO2 To MSR-1

<https://www.youtube.com/embed/gxVCZyOQjJs?si=-b2uhOKwxXw6DswV>

1. Unplug your MSR-1 from power and remove the back

[IMG_3784.jpeg](#) type unknown

2. Remove the board from the case and orientate it like the below image. In the bottom right there is a black connector. There is a x in the lower right corner of the connector

[IMG_3788.jpeg](#) type unknown

3. Orientate the CO2 module so the white x on the module is also in the lower right and the gold dot is in the upper left as seen below

[IMG_3787.jpeg](#) type unknown

4. Align the connector on the back of the CO2 module with the connector on the MSR-1. Once aligned, push down so the CO2 module fully seats itself

[IMG_3789.jpeg](#) type unknown

5. Put your MSR-1 back in the case and slide the back cover on. Please refer to our [CO2 calibration guide](#) to calibrate the CO2 module.

Reducing LD2410 Log Entries In Home Assistant

1. Open your Home Assistant config.yaml
2. Add in this code but use your MSR-1's name

```
logbook:  
  exclude:  
    entities:  
      - binary_sensor.hallway_motion_home_security_motion_detection
```

Troubleshooting

Manually Uploading Code Through ESPHome

If your device becomes unresponsive and you've exhausted the other troubleshooting methods you can upload a fresh set of firmware by following the below guide. The utility does need to be run from Chrome or Edge.

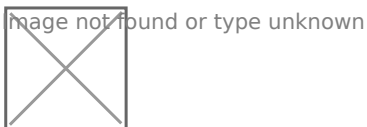
If your device has already been connected to Home Assistant previously please refer to [Removing Device From Home Assistant](#) first before proceeding

1. Plug your MSR-1 into your computer with a quality USBC cable that supports data transfer
2. Navigate to our installer page and click connect [** Install Page **](#)
3. Select your Apollo device, it will show with a similar name to the one below, and click connect. If you aren't sure which device it is, you can unplug the MSR-1 and see which disappears.

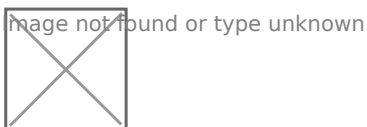
[ComSelection.png](#) image not found or type unknown

If no device shows, click cancel and then install the recommended driver that shows on the popup. If you have installed the driver, tried different cables, and it still won't work refer here for putting the MSR-1 in bootloader mode and then retry step 3. [Putting MSR-1 In Boot Mode Document](#)

4. Choose to install the new firmware



6. Wait for the installer to finish

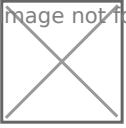


7. After finishing, check for the Apollo hotspot and connect. This might not show if you previously had the MSR-1 connected to your wifi

8. Log into Home Assistant and go to the ESPHome addon check to see if you can adopt the device.

If you encounter the below error, please complete the [Putting MSR-1 In Boot Mode Document](#) and go back to step 3.

image not found or type unknown



Putting The MSR-1 In Boot Mode

This will cover how to put the MSR-1 into boot mode. This is sometimes needed for uploading new firmware if the device is struggling.

Boot Button Only

1. Unplug the device
2. See the photo below to help access the boot button through the top of the case
3. Use a pin to press and hold the boot button, while holding the boot button plug it back into your computer/power and then release the boot button

[MSR-1 Top Shot Boot Button.jpg](#)

Image not found or type is unknown

4. Continue with [uploading the firmware document](#)

Boot and Reset Buttons

1. Unplug the device
2. Slide the back of the case off
3. Remove the device from the case
4. Plug it back into your computer
5. Press and hold the boot button, while holding it press and release the reset button, then release the boot button

[MSR-1_Buttons.jpg](#)

Image not found or type is unknown

6. Continue with [uploading the firmware document](#)

MSR-1 Reviews

MSR-1 Reviews

Make It Work Tech MSR-1 Review

<https://www.youtube.com/embed/Kt1FpRM8R18?t=1882s>

Apollo MSR-1 Review starts at [31:22](#).

MSR-1 Reviews

Smart Home Scene MSR-1 Review

[Smart Home Scene MSR-1 Review](#)

MSR-1 Reviews

DrZzs MSR-1 Review

<https://www.youtube.com/embed/YGNCw1nFwjg?t=1193>

Apollo MSR-1 review starts at [19:53](#).

MSR-1 Reviews

BeardedTinker MSR-1 Review

BeardedTinker YouTube Video review - MSR-1

<https://www.youtube.com/embed/vz1oWhPolwA?si=yan8znUNyBJjo1i5>

MSR-1 Reviews

Simon Says Home Assistant MSR-1 Review

<https://www.youtube.com/embed/M-HAuPH9KOk?si=u4klEOWg9cltZc0r>

MSR-1 Reviews

MostlyChris MSR-1 Review

<https://www.youtube.com/embed/u8q3IMkolc?si=Smxevzkj-goobTfN>