

Get started with the Environmental Sensor Board

The Environmental Sensor Board is an add-on board (also known as a pHAT or bonnet) that adds sensing capabilities to your Coral Dev Board or Raspberry Pi projects. It includes a secure cryptoprocessor with Google keys to enable connectivity with [Google Cloud IoT Core](#) services, allowing you to securely connect to the device and then collect, process, and analyze the sensor data.

This page is your guide to get started with the board. You can choose to connect the board to either the [Coral Dev Board](#) or any Raspberry Pi with a 40-pin header. Then you can collect sensor data using our Python API. However, if you plan to process the data from your Environmental Sensor using a cloud backend, then instead try getting started with the Google Cloud tutorial called [Sensor data collection and analytics with Cloud IoT Core](#).

1: Attach the sensor board

Whether you're using the Coral Dev Board or a Raspberry Pi, the Environmental Sensor Board must attach to the 40-pin header so that the sensor board is oriented directly above the host board. If you're using the Dev Board, this requires a header extension so the board can clear the heat sink/fan. For example, see this [extra tall header from Adafruit](#), which gives enough room for the Dev Board fan to breathe.

If you're using a Raspberry Pi, you can now power on your board. The sensor board includes an EEPROM that allows the Raspberry Pi to automatically apply the device tree that configures all the header pins.

If you're using a Dev Board, the device tree is not automatically applied, so you must first power the board and perform the following steps from the Dev Board shell (be sure you're [connected to the internet](#)):

1. Download our device tree for the sensor board and write it to `/boot/coral-enviro-board.dtbo` as follows:

```
2. curl "https://coral.googleusercontent.com/coral-cloud/+refs/heads/master/coral-enviro-devicetree/coral-enviro-board.dtbo?format=TEXT" \
3. | base64 --decode | sudo tee /boot/coral-enviro-board.dtbo > /dev/null
```
4. Add the new device tree to the overlays file:

```
5. sudo sed -i '$s/$/coral-enviro-board/' /boot/overlays.txt
```

This overlay will apply once you reboot your board, but you need to install the Python library first.

2: Install the Python library

On your Dev Board or Raspberry Pi, run the following commands to install the Environmental Sensor Board library and driver:

```
echo "deb https://packages.cloud.google.com/apt coral-cloud-stable main" | sudo tee
/etc/apt/sources.list.d/coral-cloud.list

curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

sudo apt update

sudo apt upgrade

sudo apt install python3-coral-enviro
```

Now reboot to apply the overlay and enable the sensor board driver:

```
sudo reboot now
```

3: Try the demo

We've provided a sample script that prints data from the temperature, humidity, ambient light, and pressure sensors onto the board's OLED display. You can run it as follows:

```
cd /usr/lib/python3/dist-packages/coral/enviro
```

```
# On Raspberry Pi:  
python3 enviro_demo.py
```

```
# On Dev Board (root is required for GPIO control):  
sudo python3 enviro_demo.py
```

Now look at the board's display to see the sensor data.

You can browse the [sample code here](#).

Also take a look at the API documentation for [coral.enviro.board](#) and [coral.cloudiot.core](#).

For more information about connecting your application to the cloud, see the [Cloud IoT Core documentation](#).

Also checkout the complete tutorial about [sensor data collection and analytics with Cloud IoT Core](#).

coral.enviro.board

```
class coral.enviro.board.EnviroBoard
```

An interface for all input and output modules on the Environmental Sensor Board.

| [temperature](#)

Gets the current temperature, in Celsius.

| [humidity](#)

Gets the current relative humidity, in percentage.

| [ambient_light](#)

Gets the ambient light, in lux.

| [pressure](#)

Gets the current atmospheric pressure, in kPa.

| [grove_analog](#)

Gets a raw value from a device connected to the board's analog Grove connector.

Note

ADC is set to +/- 6V range, independent of supply voltage (selected by jumper).

| [display](#)

Gets an instance of [luma.core.device.device](#) representing the board's OLED display.

For example, you can write to the display using [luma.core.render.canvas](#) as follows:

```
enviro = EnviroBoard()

update_display(enviro.display, "Hello world")

def update_display(display, msg):
    with canvas(display) as draw:
        draw.text((0, 0), msg, fill='white')
```