* <https://www.circuito.io/app?components=512,9403,11021,13678,164792,398787,398790>
* To link the data of sensors collected using the Arduino code you provided to the Python user interface software, you can use the following steps:
* **Step 1:** Install the necessary Python libraries
* Python
* pip install serial requests pandas
* Use code with caution. [Learn more](https://bard.google.com/faq#coding)
* content\_copy
* **Step 2:** Start the Python user interface software
* Python
* python user\_interface\_software.py
* Use code with caution. [Learn more](https://bard.google.com/faq#coding)
* content\_copy
* **Step 3:** Modify the Arduino code to send the sensor data to the Python user interface software
* // Add the following code to the end of the Arduino setup() function:
* Serial.begin(9600); // Set the serial baud rate to 9600 bps.
* // Add the following code to the Arduino loop() function:
* // Collect the sensor data.
* float temperature = bme680.temperature;
* float humidity = bme680.humidity;
* float pressure = bme680.pressure;
* float air\_quality = bme680.air\_quality;
* float tvoc = bme680.tvoc;
* float mq135\_reading = mq135.read();
* float mq9\_reading = mq9.read();
* float mq7\_reading = mq7.read();
* // Create a JSON object containing the sensor data.
* DynamicJsonDocument jsonDoc;
* jsonDoc["temperature"] = temperature;
* jsonDoc["humidity"] = humidity;
* jsonDoc["pressure"] = pressure;
* jsonDoc["air\_quality"] = air\_quality;
* jsonDoc["tvoc"] = tvoc;
* jsonDoc["mq135\_reading"] = mq135\_reading;
* jsonDoc["mq9\_reading"] = mq9\_reading;
* jsonDoc["mq7\_reading"] = mq7\_reading;
* // Serialize the JSON object to a string.
* char jsonBuffer[512];
* serializeJson(jsonDoc, jsonBuffer, sizeof(jsonBuffer));
* // Send the JSON string to the Python user interface software.
* Serial.println(jsonBuffer);
* **Step 4:** Modify the Python user interface software to receive and process the sensor data
* Python
* import serial
* import json
* # Open the serial port to communicate with the Arduino Uno microcontroller.
* ser = serial.Serial("/dev/ttyUSB0", 9600)
* # Start a loop to continuously receive and process the sensor data.
* while True:
* # Read a line of data from the serial port.
* data = ser.readline().decode("utf-8")
* # Deserialize the JSON string into a Python dictionary.
* sensor\_data = json.loads(data)
* # Process the sensor data.
* # ...
* Use code with caution. [Learn more](https://bard.google.com/faq#coding)
* content\_copy
* **Sample input to the sensors**
* Temperature: 25.0 °C
* Humidity: 50.0 %
* Pressure: 1013.25 hPa
* Air quality: 60 (Moderate)
* TVOC: 100 ppb
* MQ135 reading: 100
* MQ9 reading: 200
* MQ7 reading: 300
* **Sample output**
* {
* "temperature": 25.0,
* "humidity": 50.0,
* "pressure": 1013.25,
* "air\_quality": 60,
* "tvoc": 100,
* "mq135\_reading": 100,
* "mq9\_reading": 200,
* "mq7\_reading": 300
* }