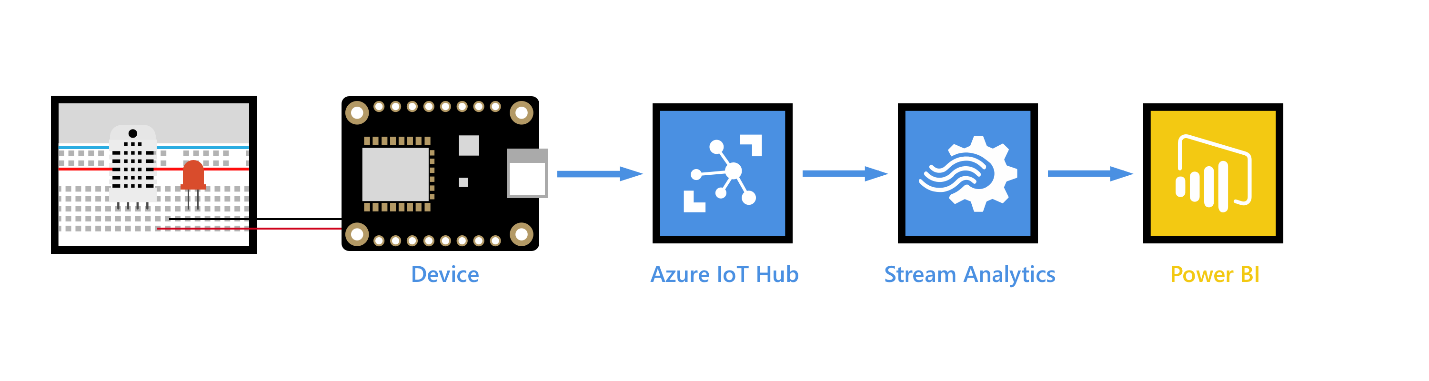
**Visualize Real-Time Sensor Data from Azure IoT Hub using Power BI**

**Steps of the Experiment**

Throughout all the steps, all the other adjustable information we encounter that are not told to be changed, should stay the same as default.

1. Create an IoT hub which is the cloud platform that our device will connect to
2. Register a device for NodeMCU 1.0 (ESP-12E Module) in your IoT hub
3. Setup the circuit
4. Configure NodeMCU 1.0 (ESP-12E Module) with Arduino
5. Run a sample application on NodeMCU 1.0 (ESP-12E Module)to send sensor data to your IoT hub
6. Get your IoT hub ready for data access by adding a consumer group
7. Create, configure and run a Stream Analytics job for data transfer from your IoT hub to your Power BI account
8. Create and publish a Power BI report to visualize the data

**Hardware Used for this Experiment**

Breadboard, 10 k ohm resistor, LED, M/M Jumper wires

NodeMCU 1.0 (ESP-12E Module)

DHT11 temperature and humidity sensor

Micro USB to Type A USB cable

PC running Windows

Wireless Network

**Software Used for This Experiment**

Microsoft Azure

Arduino IDE

Visual Studio Code

Microsoft Power BI

**STEP 1: Create an IoT hub which is the cloud platform that our device will connect to**

1. Sign in to the [Azure portal](https://portal.azure.com/).
2. Select Create a resource > Internet of Things > IoT Hub.
3. In the IoT hub pane, enter the following information for your IoT hub:

* Subscription: Choose the subscription that you want to use to create this IoT hub.
* Resource group: Create a resource group to host the IoT hub or use an existing one. For more information, see [Use resource groups to manage your Azure resources](https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-portal).
* Region: Select the closest location to you.
* Name: Create a name for your IoT hub. If the name you enter is available, a green check mark appears.

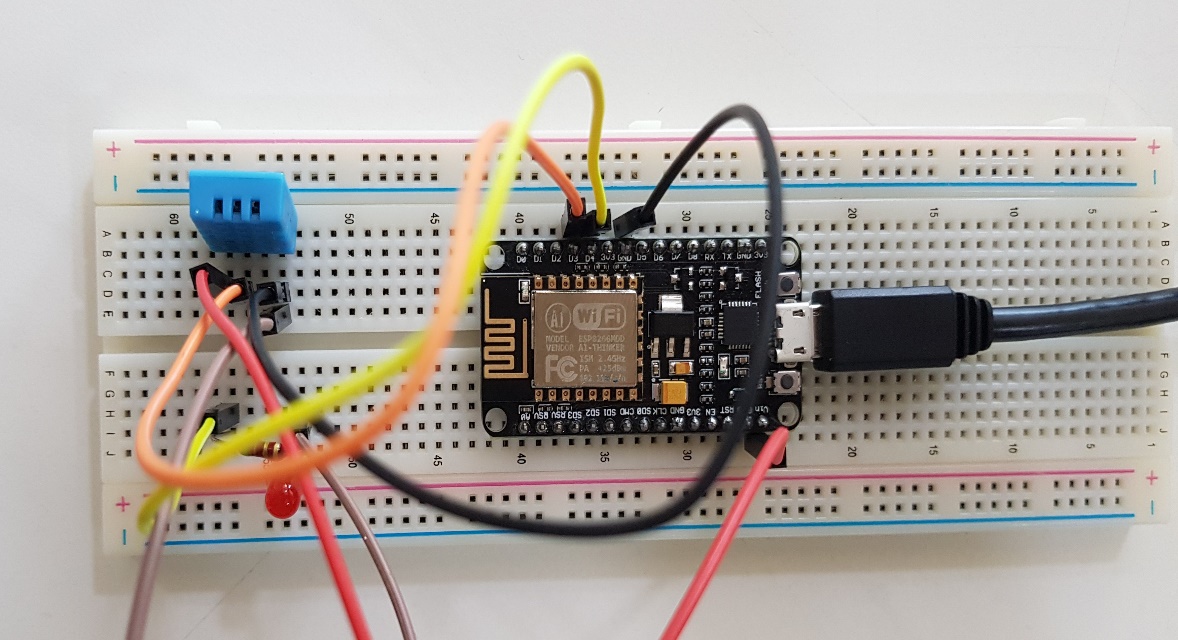
1. Select Next: Size and scale to continue creating your IoT hub.
2. Choose your Pricing and scale tier. Select the F1 - Free tier if it's still available on your subscription.
3. Select Review + create.
4. Review your IoT hub information, then click Create. Your IoT hub might take a few minutes to create. You can monitor the progress in the Notifications pane.

Now that you have created an IoT hub, locate the important information that you use to connect devices and applications to your IoT hub.

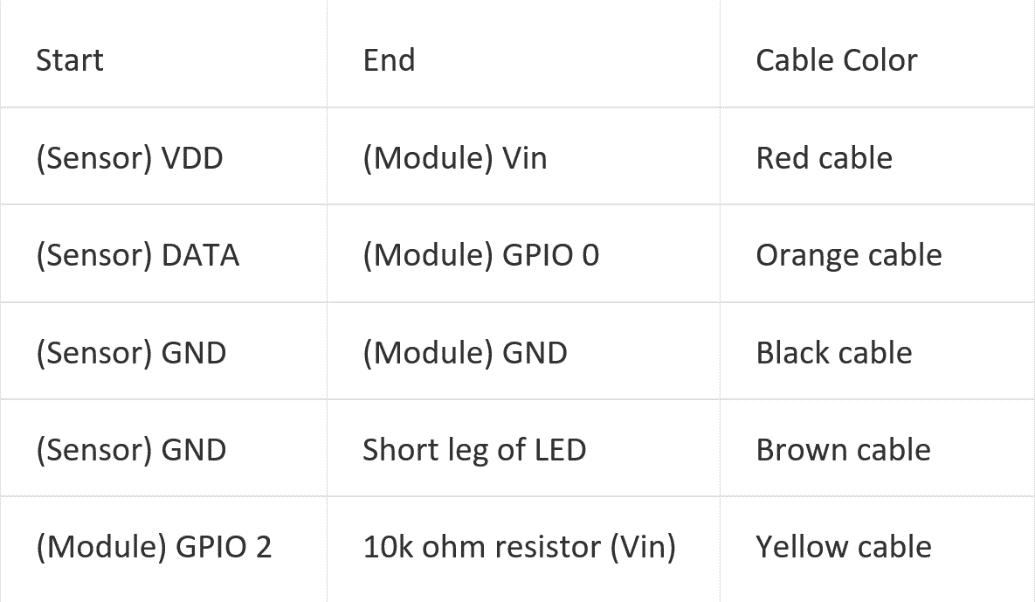
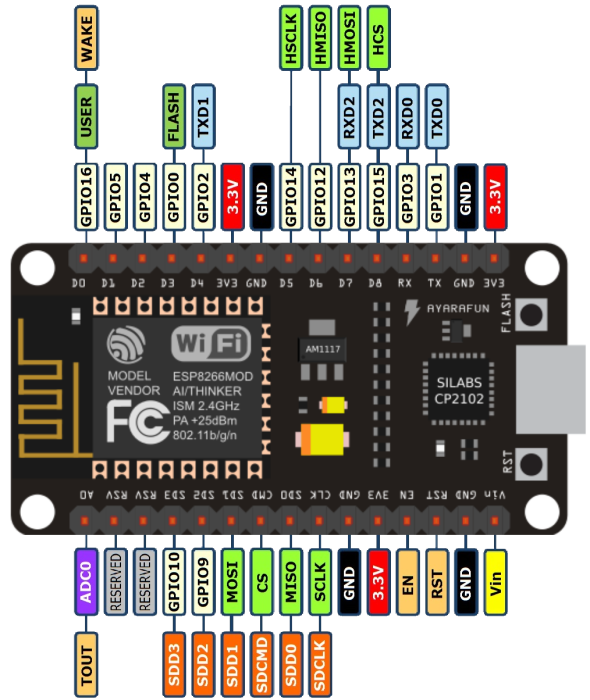
**STEP 2: Register a device for NodeMCU 1.0 (ESP-12E Module) in your IoT hub**

1. In your IoT hub navigation menu, open IoT devices, then click Add to register a device in your IoT hub.
2. Enter a Device ID for the new device. Device IDs are case sensitive.
3. Click Save.
4. After the device is created, open the device from the list in the IoT devices pane.
5. Copy the Connection string---primary key and save it to Notepad to use later.

**STEP 3: Setup the circuit**

****

1. Install the NodeMCU 1.0 (ESP-12E Module) onto the breadboard and make the following connections as shown below in the table.

****

1. Connect long leg of LED in series with 10k ohm resistor.
2. Use the Micro USB to Type A USB cable to connect NodeMCU 1.0 (ESP-12E Module) to your computer.

**STEP 4: Configure NodeMCU 1.0 (ESP-12E Module) with Arduino**

1. Open the Arduino IDE, go to Files and click on Preferences
2. Copy the below code in Additional Boards Manager URLs  
   <http://arduino.esp8266.com/stable/package_esp8266com_index.json>
3. Click OK to close the Preferences Tab
4. Go to Tools and Board, and then select Boards Manager
5. Navigate to esp8266 by ESP8266 Community and install the software for Arduino  
   Go to Board and select NodeMCU 1.0 (ESP-12E Module)
6. Select the correct COM port
7. Make sure that these settings are correct: (Upload Speed: 115200, CPU Frequency: 80MHz, Flash Size: 4M (3M SPIFFS), Upload Using: Serial)

**STEP 5: Run a sample application on NodeMCU 1.0 (ESP-12E Module) to send sensor data to your IoT hub**

1. Download the application from GitHub:

<https://github.com/Azure-Samples/iot-hub-feather-huzzah-client-app.git>

1. In order to assure the workability of the application, download the necessary libraries. Select Sketch > Include Library > Manage Libraries. Type in and download the followings.

* Azure IoT Hub
* Azure IoT Utility
* Azure IoT Protocol MQTT
* Arduino Json

Then, download the DHT Sensor Library from GitHub and simply add it to ‘’libraries’’ folder in ‘’Arduino’’ folder.

<https://github.com/adafruit/DHT-sensor-library>

1. Change a few lines in ‘’configurations.h‘’ to ensure it runs.

* **from** ‘’DEVICE\_ID: Feather HUZZAH ESP8266 Wi-Fi ’’ **to** ‘’#define DEVICE\_ID "NodeMCU 1.0 (ESP-12E Module)"’’
* **from** ‘’#define DHT\_TYPE DHT22’’ **to** ‘’#define DHT\_TYPE DHT11’’
* **from** ‘’#define LED\_PIN 0’’ **to** ‘’#define LED\_PIN 2’’
* **from** ‘’#define DHT\_PIN 2’’ **to** ‘’#define DHT\_PIN 0’’

1. Upload the application to NodeMCU 1.0 (ESP-12E Module)
2. After the upload completes successfully, follow these steps to enter your credentials:

Click Tools > Serial Monitor.

In the serial monitor window, notice the two drop-down lists in the lower-right corner.

Select No line ending for the left drop-down list.

Select 115200 baud for the right drop-down list.

In the input box located at the top of the serial monitor window, enter the following information if you are asked to provide them, and then click Send.

* + Wi-Fi SSID
  + Wi-Fi password
  + Device connection string

**STEP 6: Get your IoT hub ready for data access by adding a consumer group**

Consumer groups are used by applications to pull data from Azure IoT Hub. To add a consumer group to your IoT hub, follow these steps:

1. In the [Azure portal](https://ms.portal.azure.com/), open your IoT hub.
2. In the left pane, click Endpoints, select Events on the middle pane, enter a name under Consumer groups on the right pane, and then click Save.

**STEP 7: Create, configure and run a Stream Analytics job for data transfer from your IoT hub to your Power BI account**

1. In the [Azure portal](https://portal.azure.com/), click Create a resource > Internet of Things > Stream Analytics job.
2. Enter the following information for the job then click Create.

* Job name: The name of the job. The name must be globally unique.
* Resource group: Use the same resource group that your IoT hub uses.
* Location: Use the same location as your resource group.
* Pin to dashboard: Check this option for easy access to your IoT hub from the dashboard.

### To add an input to the Stream Analytics job, follow these steps:

Open the Stream Analytics job.

Under Job Topology, click Inputs.

In the Inputs pane, click Add, and then enter the following information:

* Input alias: The unique alias for the input.
* Source: Select IoT hub.
* Consumer group: Select the consumer group you just created.

Click Create.

### To add an output to the Stream Analytics job, follow these steps:

Under Job Topology, click Outputs.

In the Outputs pane, click Add, and then enter the following information:

* Output alias: The unique alias for the output.
* Sink: Select Power BI.

Click Authorize, and then sign into your Power BI account.

Once authorized, enter the following information:

* Group Workspace: Select your target group workspace.
* Dataset Name: Enter a dataset name.
* Table Name: Enter a table name.

Click Create.

### To configure the query of the Stream Analytics job, follow these steps:

* Under Job Topology, click Query.
* Replace [YourInputAlias] with the input alias of the job.
* Replace [YourOutputAlias] with the output alias of the job.
* Click Save.

### In the Stream Analytics job, click Start > Now > Start. Once the job successfully starts, the job status changes from Stopped to Running.

**STEP 8: Create and publish a Power BI report to visualize the data**

1. Ensure the sample application is running on your device.
2. Sign in to your [Power BI](https://powerbi.microsoft.com/en-us/) account.
3. Go to the group workspace that you set when you created the output for the Stream Analytics job.
4. Click Streaming datasets.
5. Under ACTIONS, click the first icon to create a report.
6. Create a line chart to show real-time temperature over time.

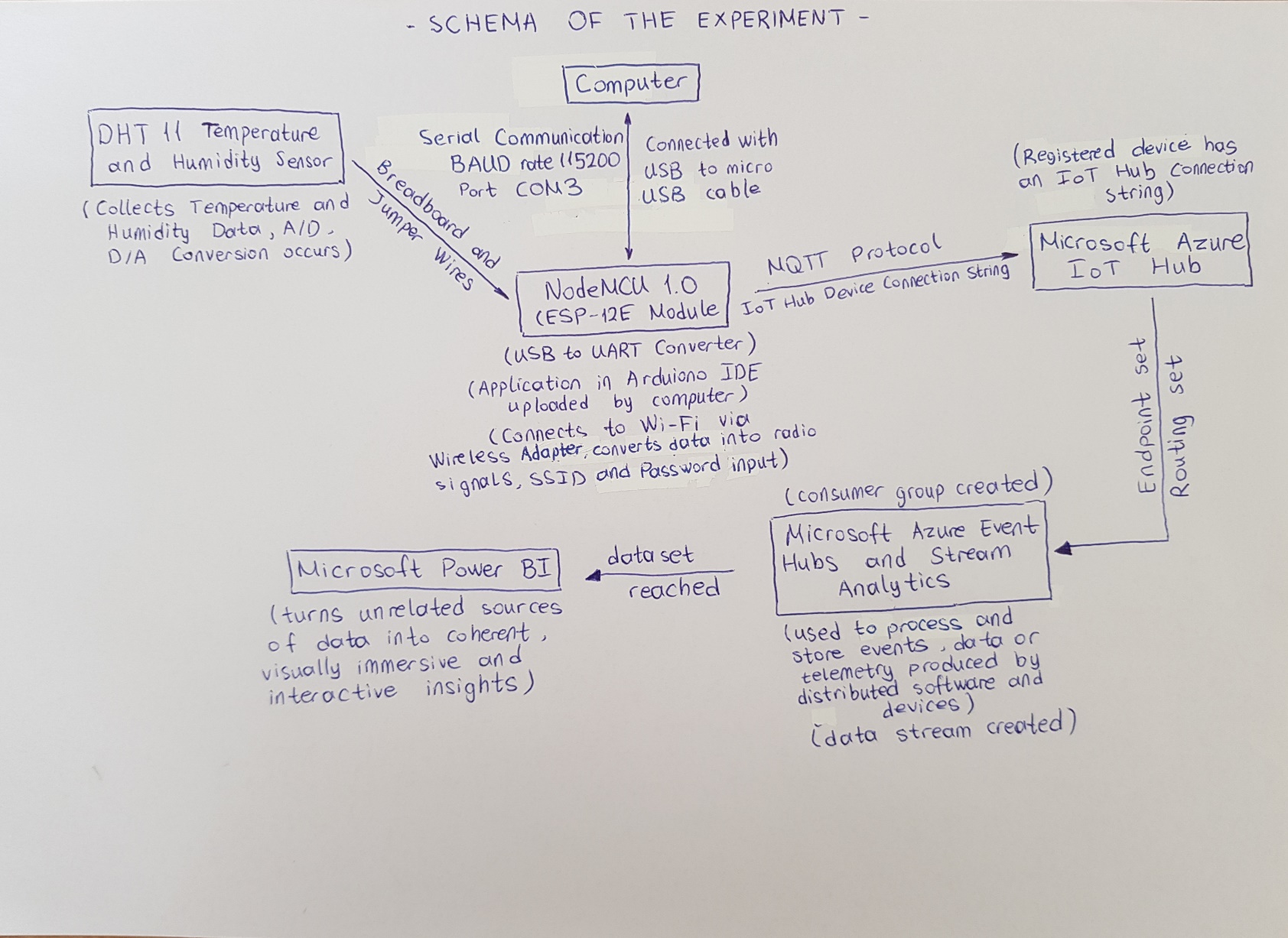
* On the report creation page, add a line chart.
* On the Fields pane, expand the table that you specified when you created the output for the Stream Analytics job.
* Drag EventEnqueuedUtcTime to Axis on the Visualizations pane.
* Drag temperature to Values.

Now a line chart is created. The x-axis displays date and time in the UTC time zone. The y-axis displays temperature from the sensor.

1. Create another line chart to show real-time humidity over time. To do this, follow the same steps above and place **EventEnqueuedUtcTime** on the x-axis and **humidity** on the y-axis.
2. Click Save to save the report.
3. Click File > Publish to web.
4. Click Create embed code, and then click Publish.

You're provided the report link that you can share with anyone for report access and a code snippet to integrate the report into your blog or website.

**You’ve successfully used Power BI to visualize real-time sensor data from your Azure IoT hub.**

**SCHEMA OF THE EXPERIMENT**