**Name: Shrey Shah SAP ID: 60002200076**

**Experiment No: 1 Date: 14/09/2023**

**Title :** To transmit and receive real time data using XBee module and publish it to Adafruit Cloud using MQTT Protocol.

**Apparatus Required:**

1. XCTU
2. ArduinoIDE with ESP32 packages installed
3. Ultrasoic Sensor
4. Arduino UNO R3
5. ESP32
6. Wires
7. Cables

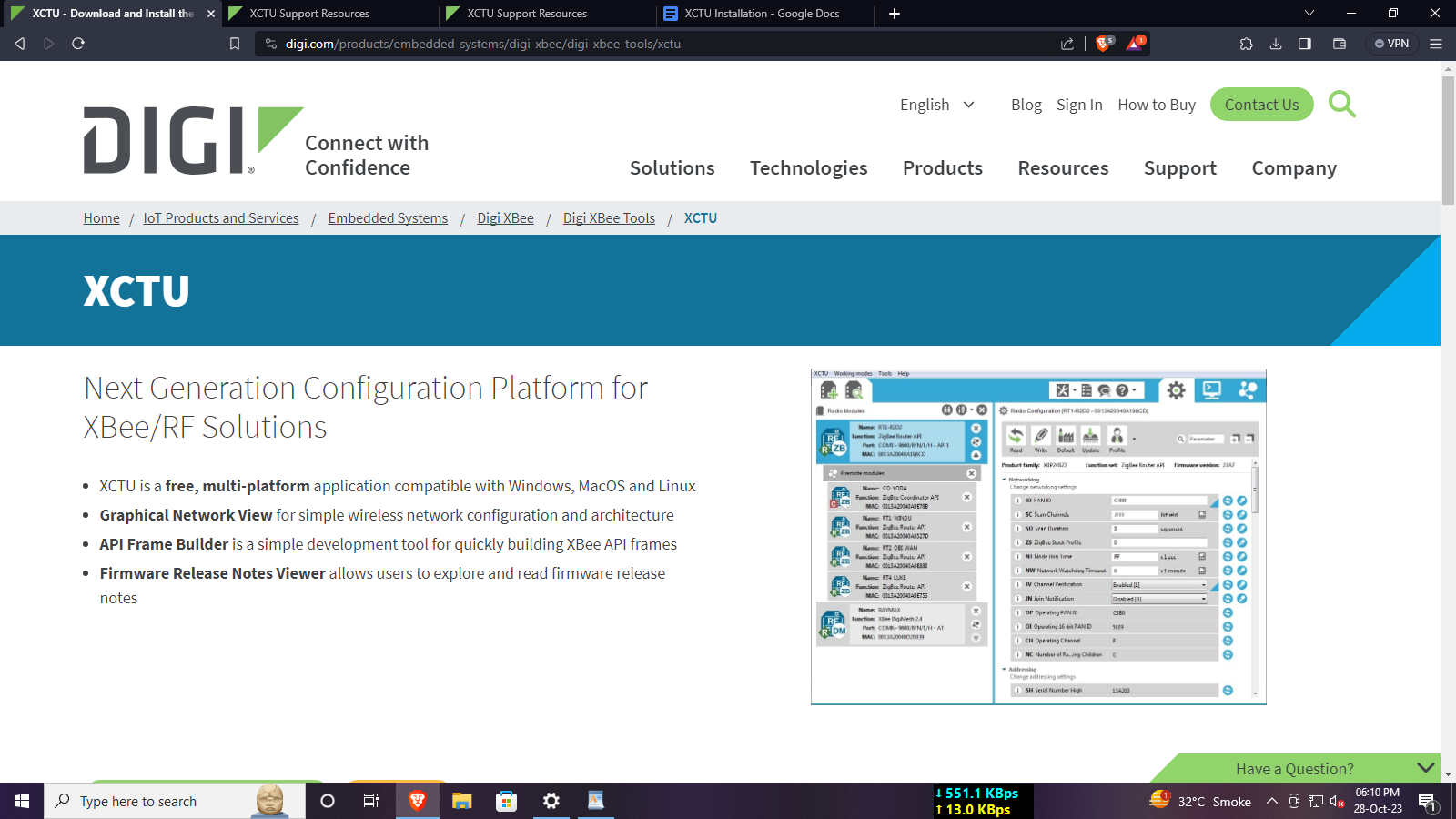
**Learning Objectives** : At the end of this experiment, students will be able to understand:

1. XBee Communication of real time data
2. Publishing the real time data to cloud
3. Data analytics of the published data
4. Implementation of MQTT Protocol

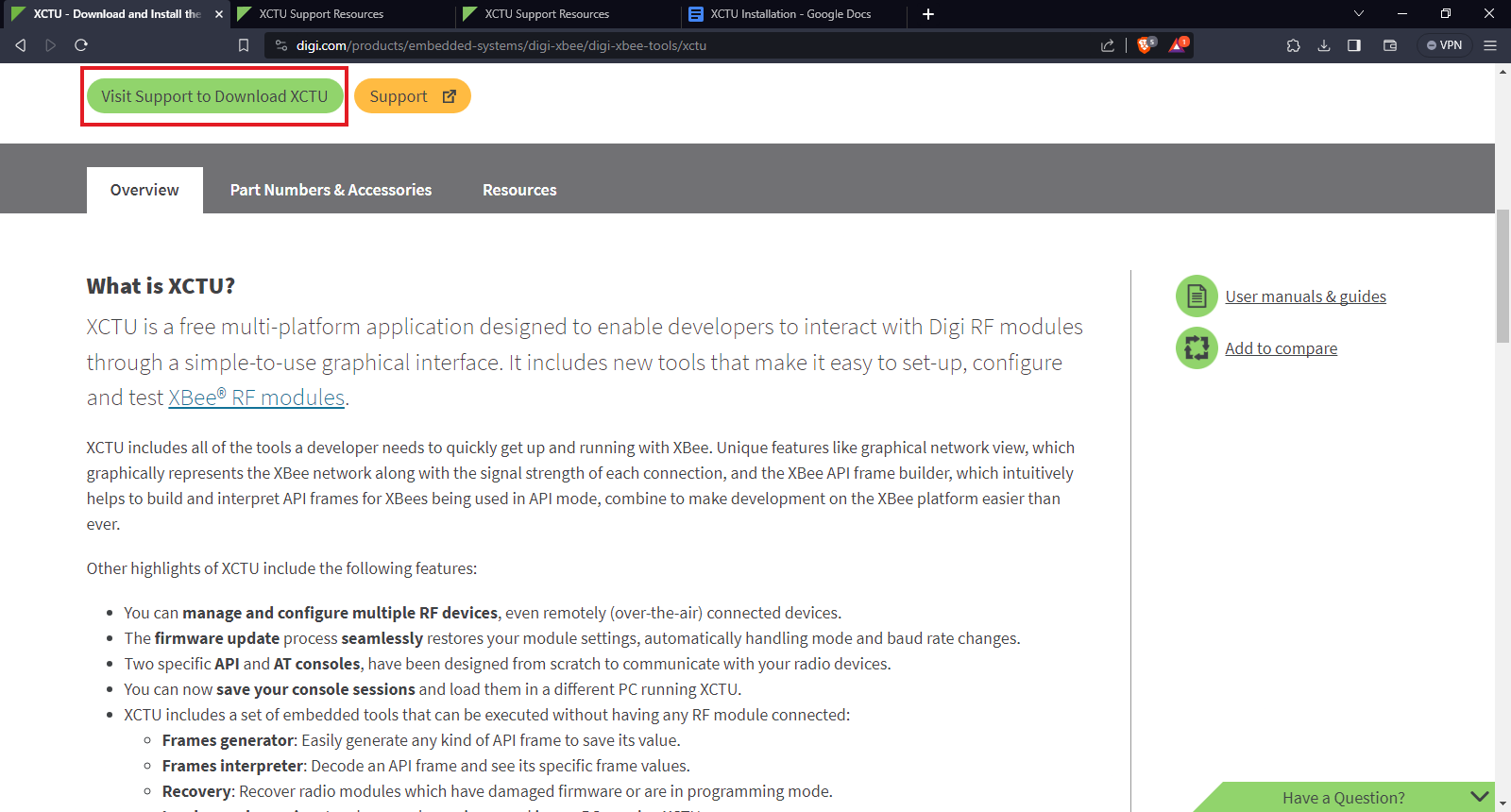
**Procedure :**

1. **Installation of XCTU:**
2. Go to website for downloading XCTU: (Type download XCTU)

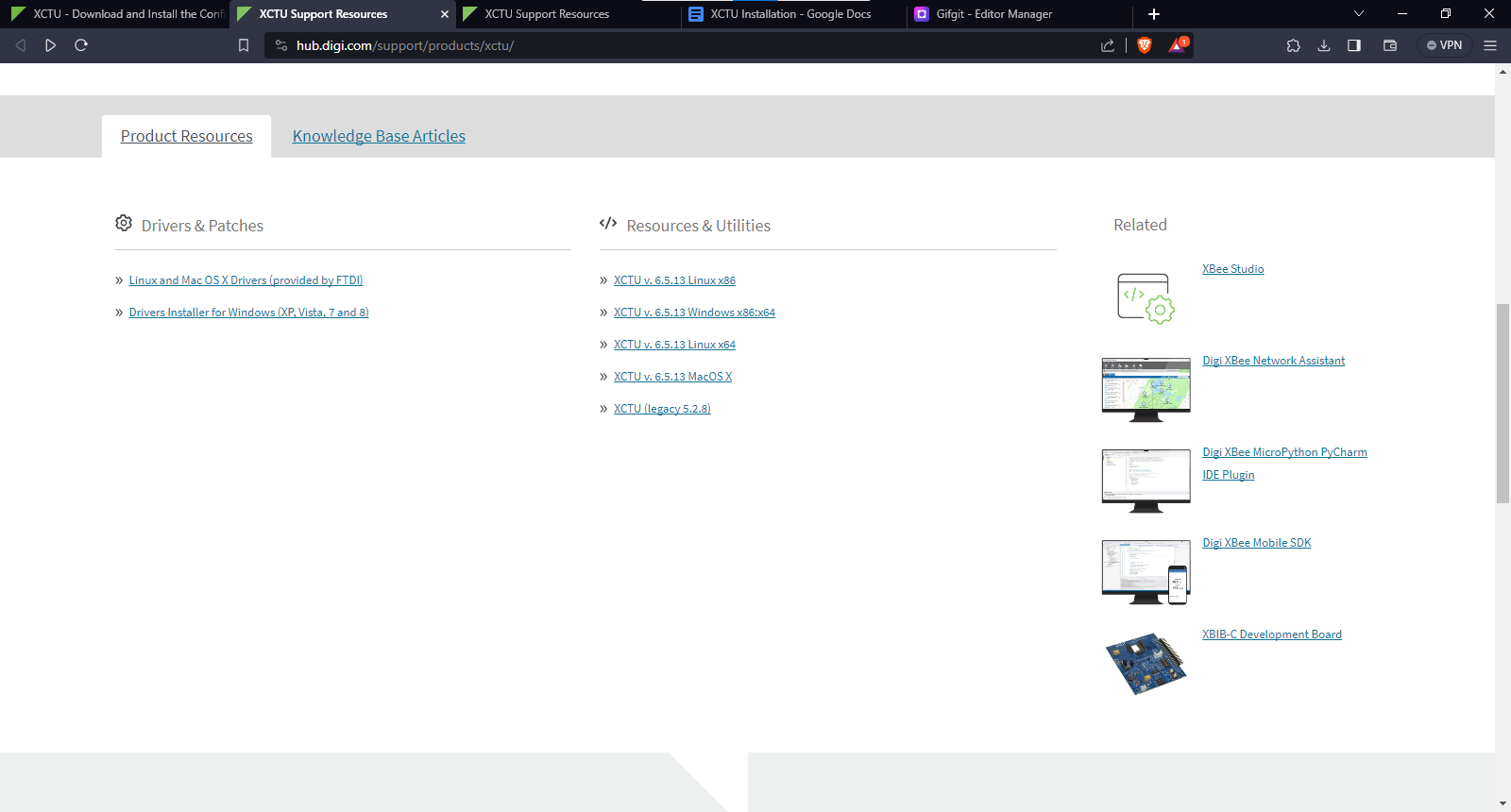
[XCTU - Download and Install the Configuration Platform for XBee/RF Solutions | Digi International](https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu)



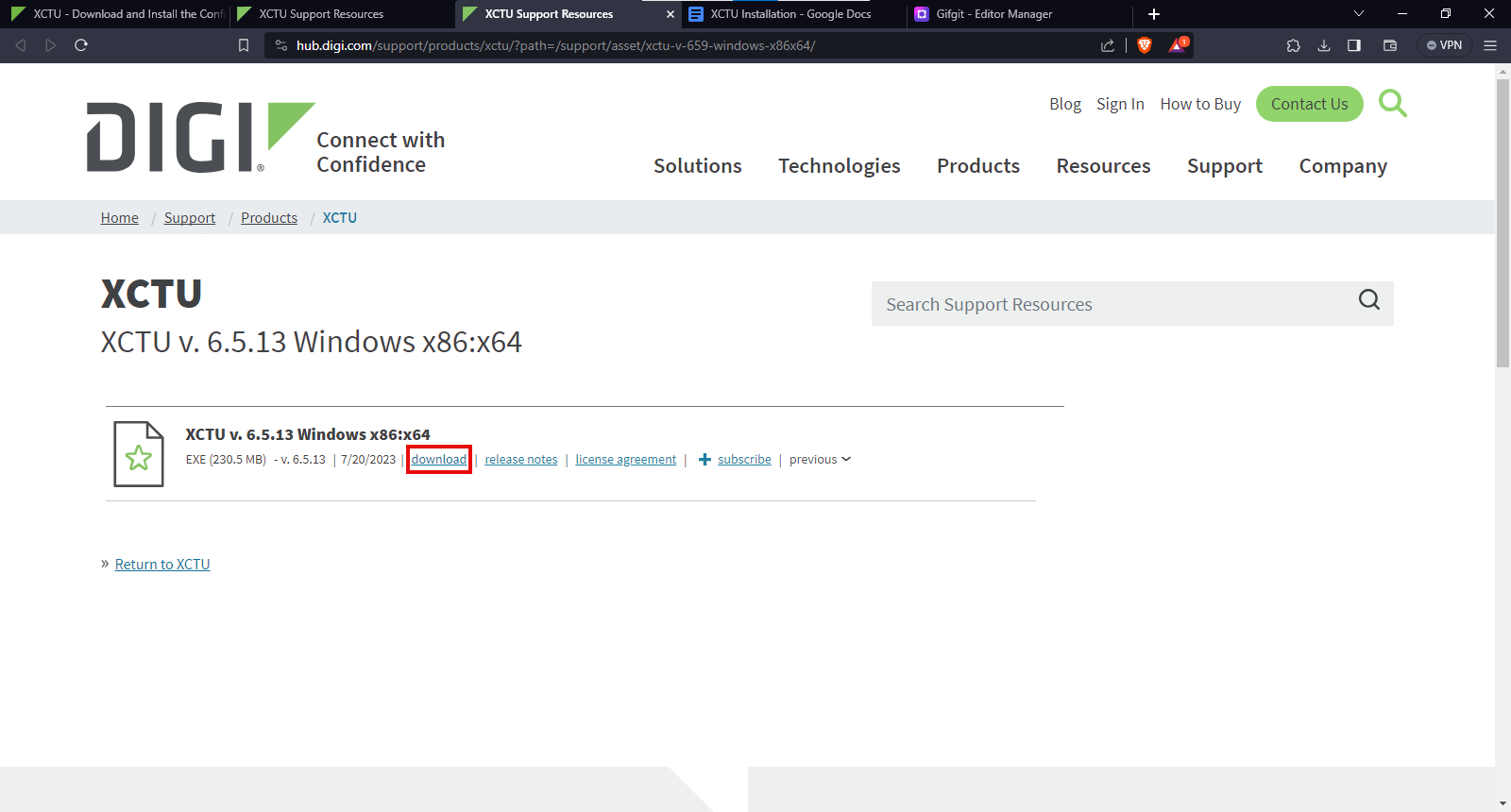
1. Scroll a little down until you see this.



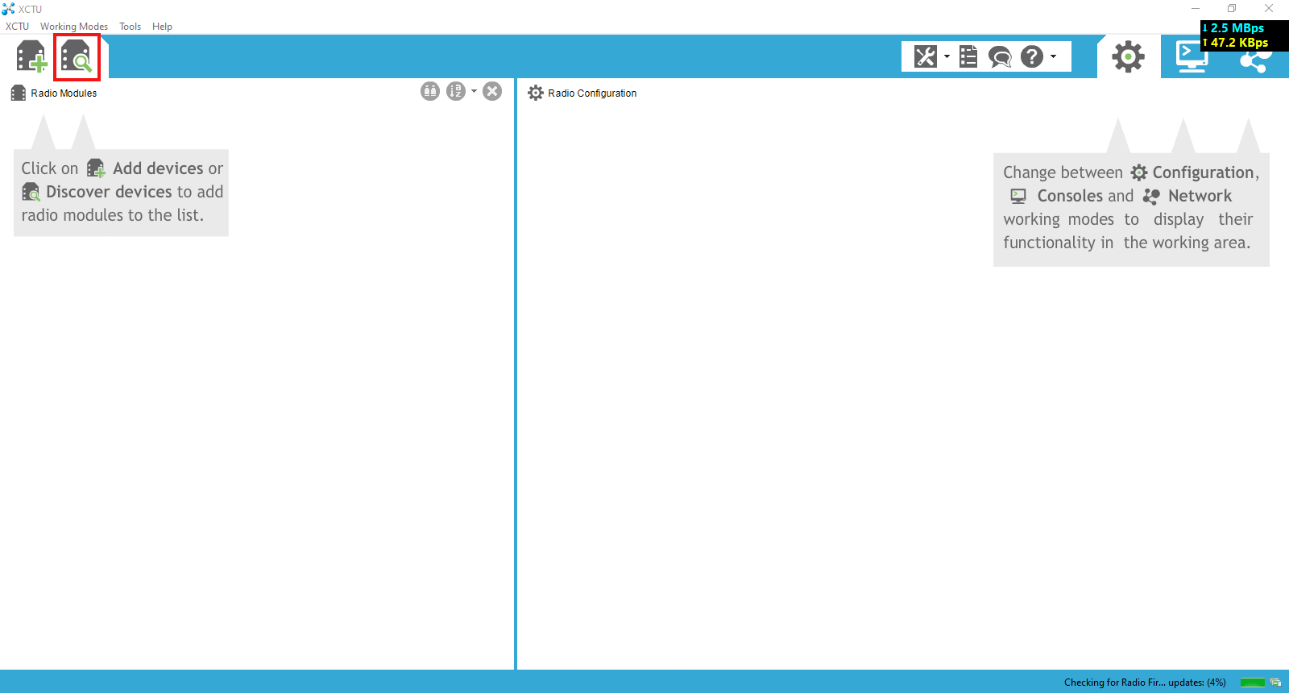
1. Then after redirection scroll down a little bit and select the appropriate choice.



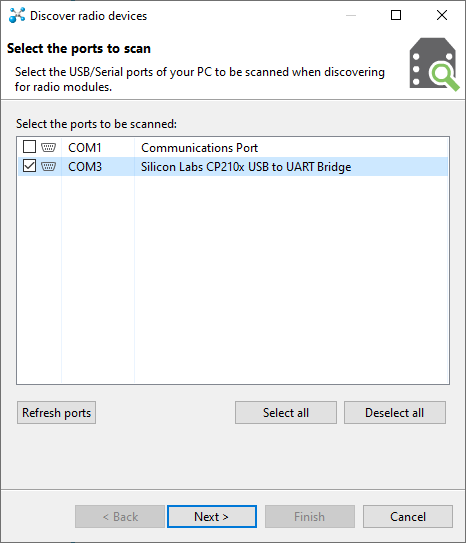
1. After selecting your choice, a window similar to this will appear. Just click on download and wait for it to download.



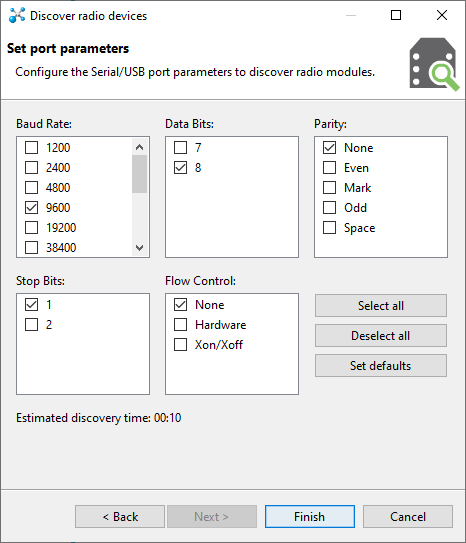
1. After installation, run the application by double clicking it.
2. Run the application in order to configure the XBee Module.
3. Click on the button shown below in order to scan all the XBee Modules connected to your device.



1. Select the appropriate PORT at which the XBee Module is connected. Then click on Next option for further configuration.



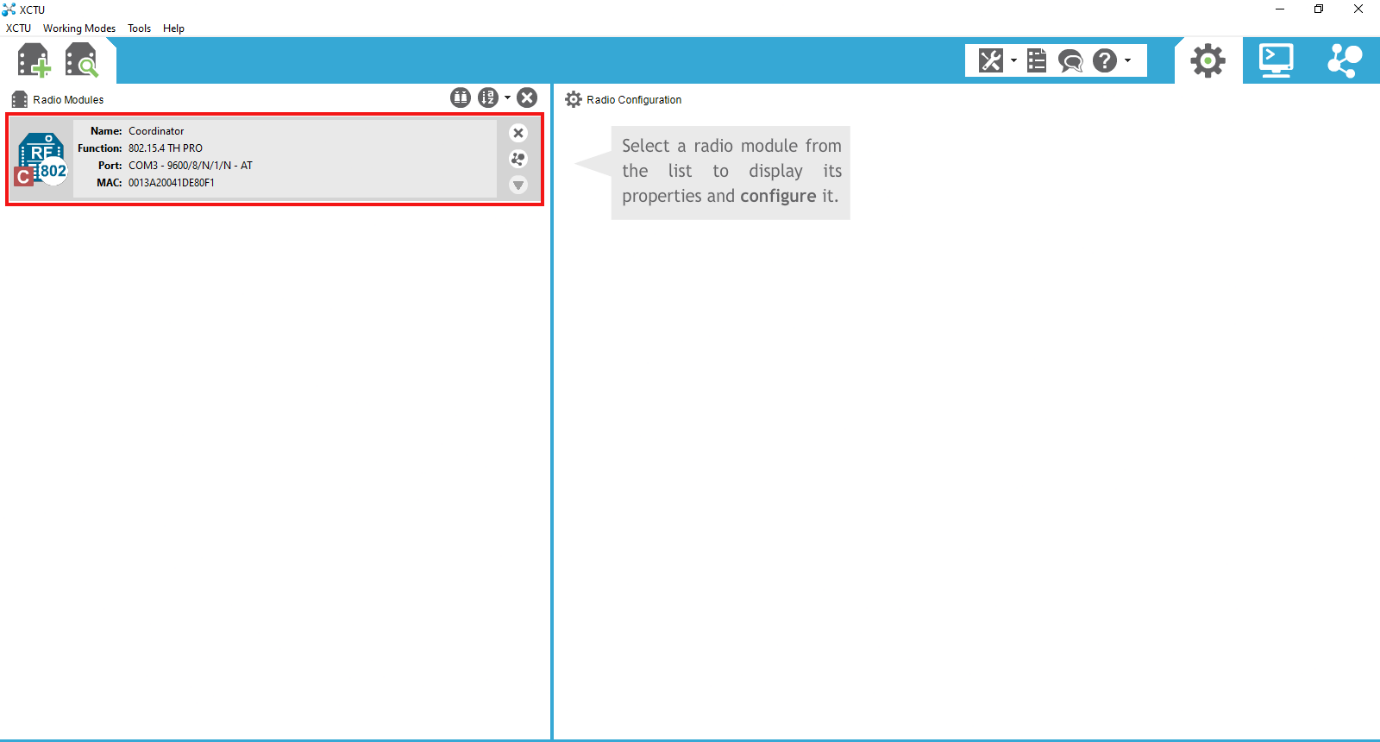
1. Click on the Finish option here. Let the options be set to default configuration.



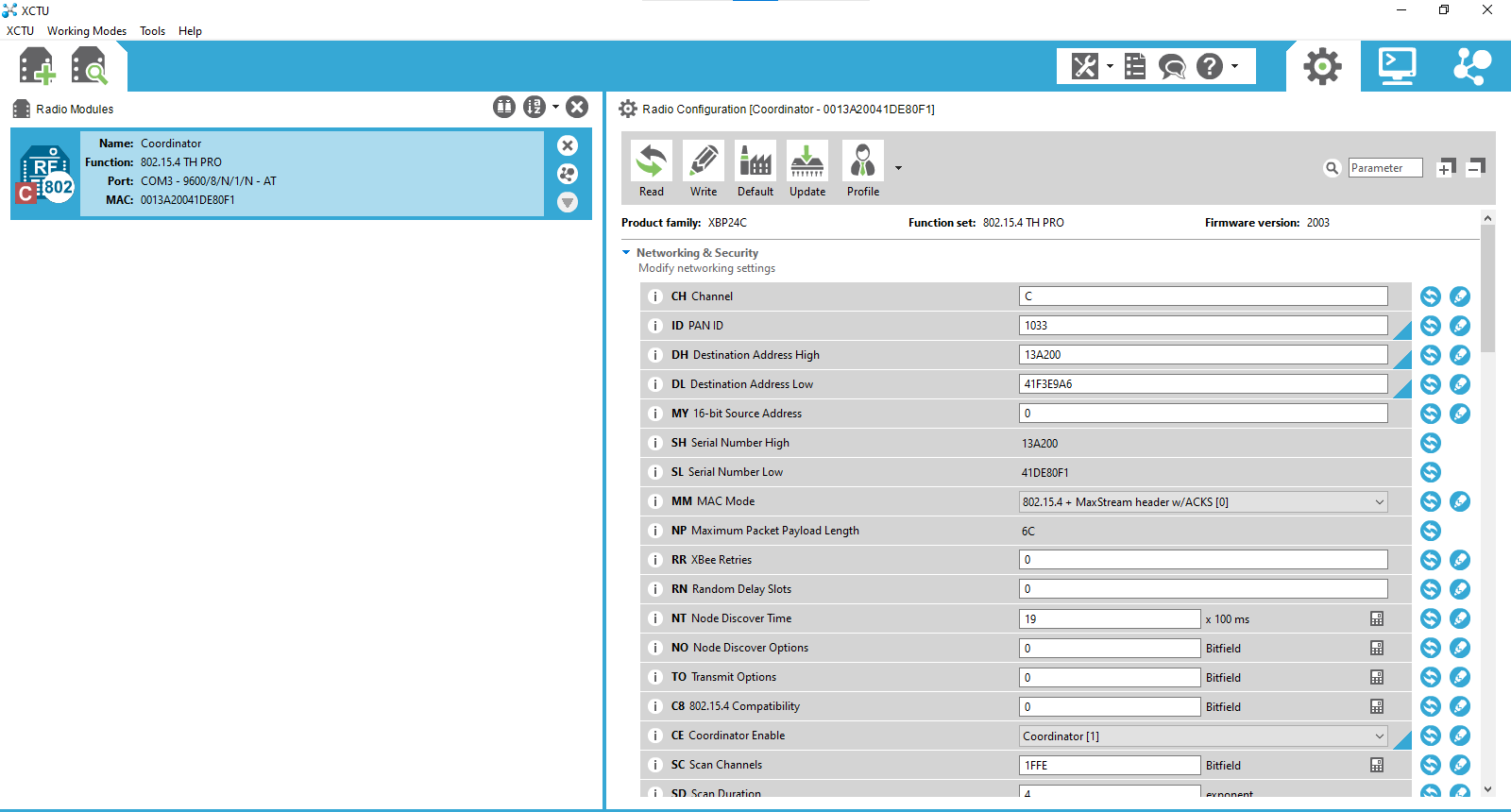
1. Select Add selected devices option in order to further update the required settings.



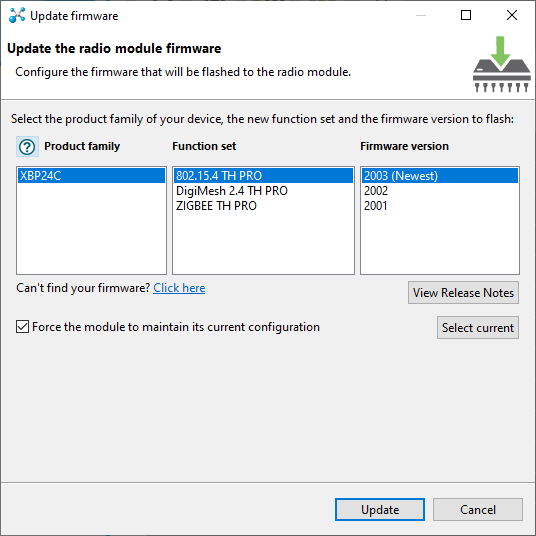
1. Click on the option available on the Left-Hand Side of the window to see all the related settings of the XBee Module.



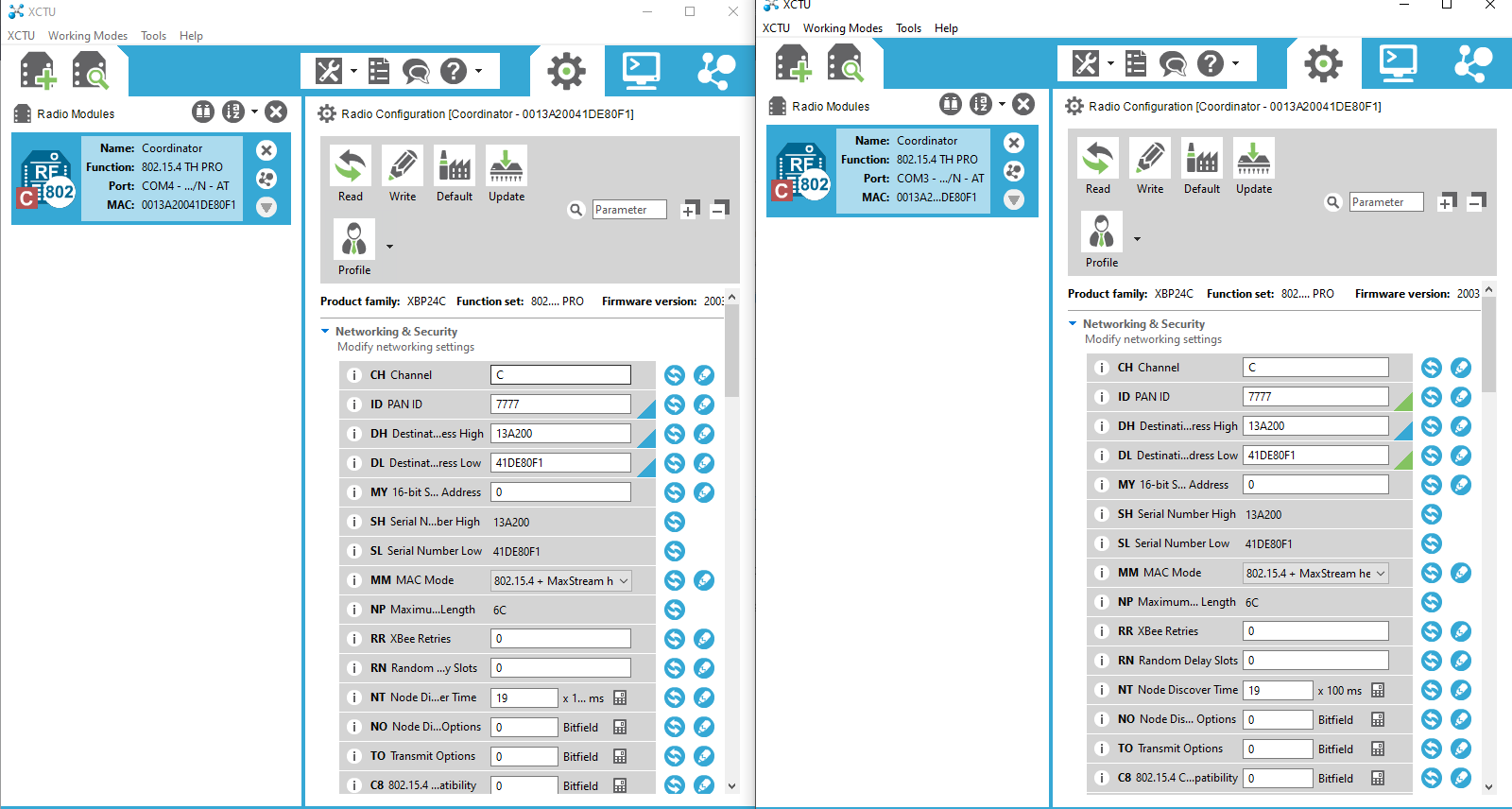
1. You will be able to see a window similar to this.



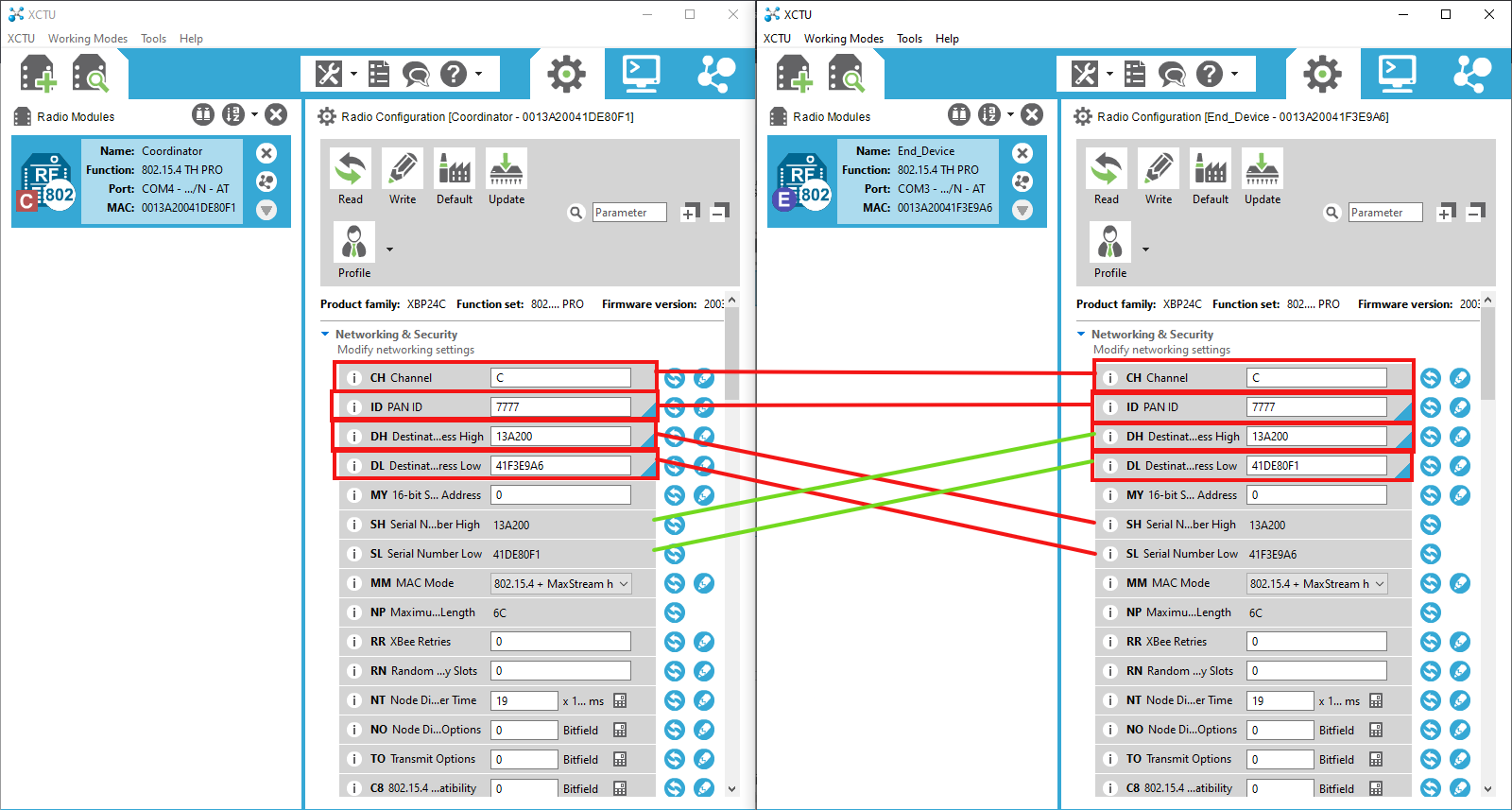
1. As we only need point-to-point communication here, we will select the Update option on the top right and deploy the following configuration and select Yes on subsequent options.



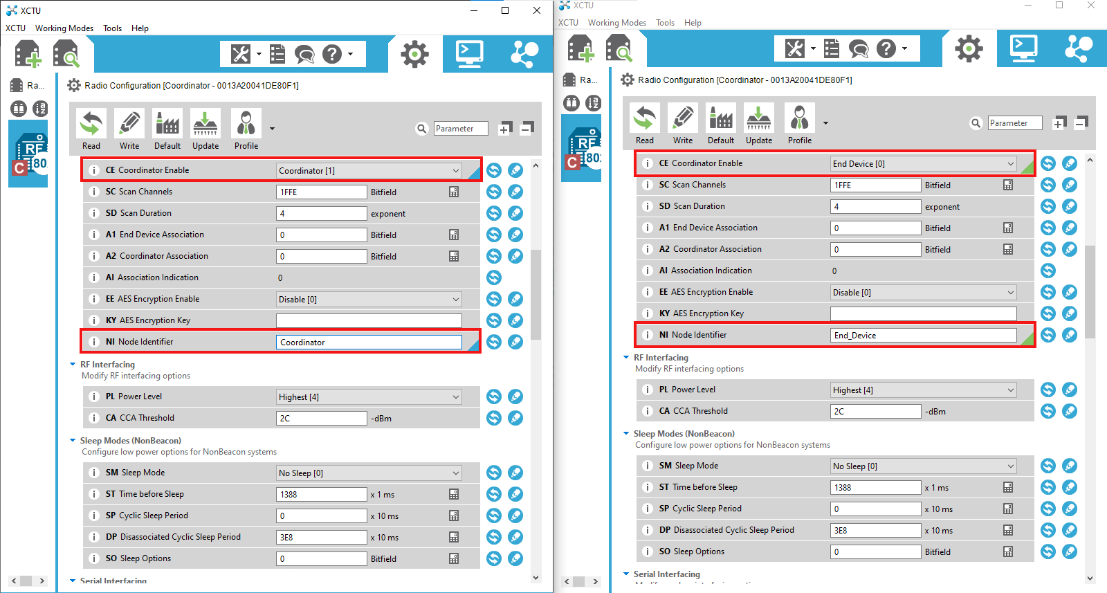
1. Similarly follow the same procedure for the second XBee Module by using another instance of XCTU software. Note that both the XBee Module should run the same function set along with the same firmware version.

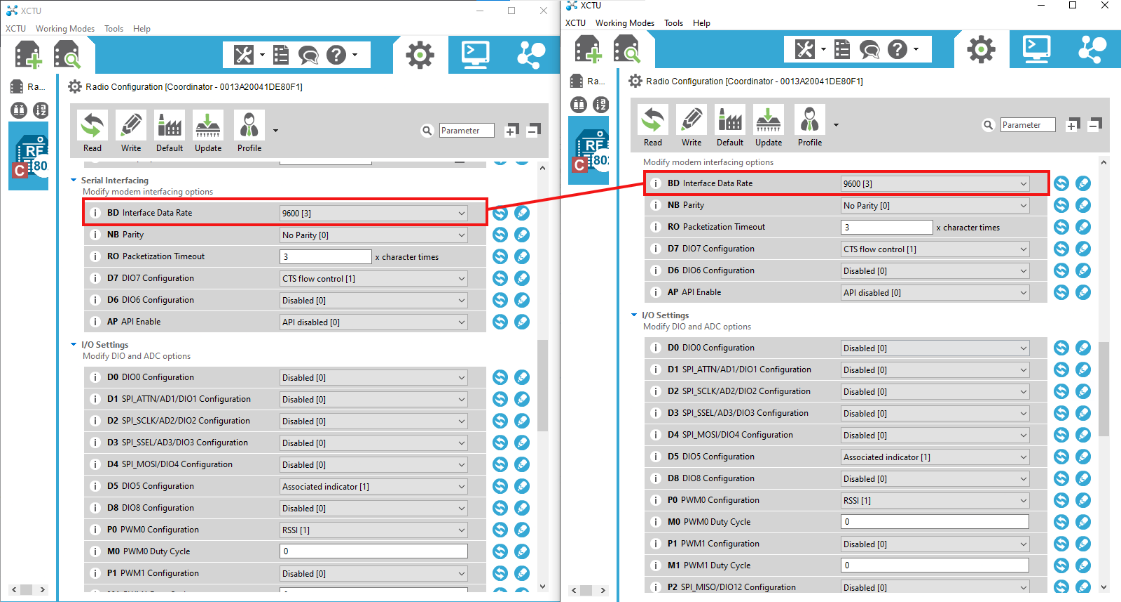


1. Now, we need to configure only 6 options in this case namely: Channel (CH), PAN ID, Destination Address High, Destination Address Low, Coordinator Enable and Baud Rate. We need to note that both the XBee Module should have the same Channel, PAN ID, Baud Rate. Only 2 major differences to be followed is that for a XBee Module, the Destination Address High Should be set to another module’s Serial Number High whereas Destination Address Low should be set to another module’s Serial Number Low. In a given network, only one XBee Module should be set as Coordinator by Coordinator Enable Option and another XBee module has to be set as End Device. We can also configure the MY Source Address in order to communicate with other XBee modules as well by substituting this value in Destination Address Low and setting Destination Address High value as 0.

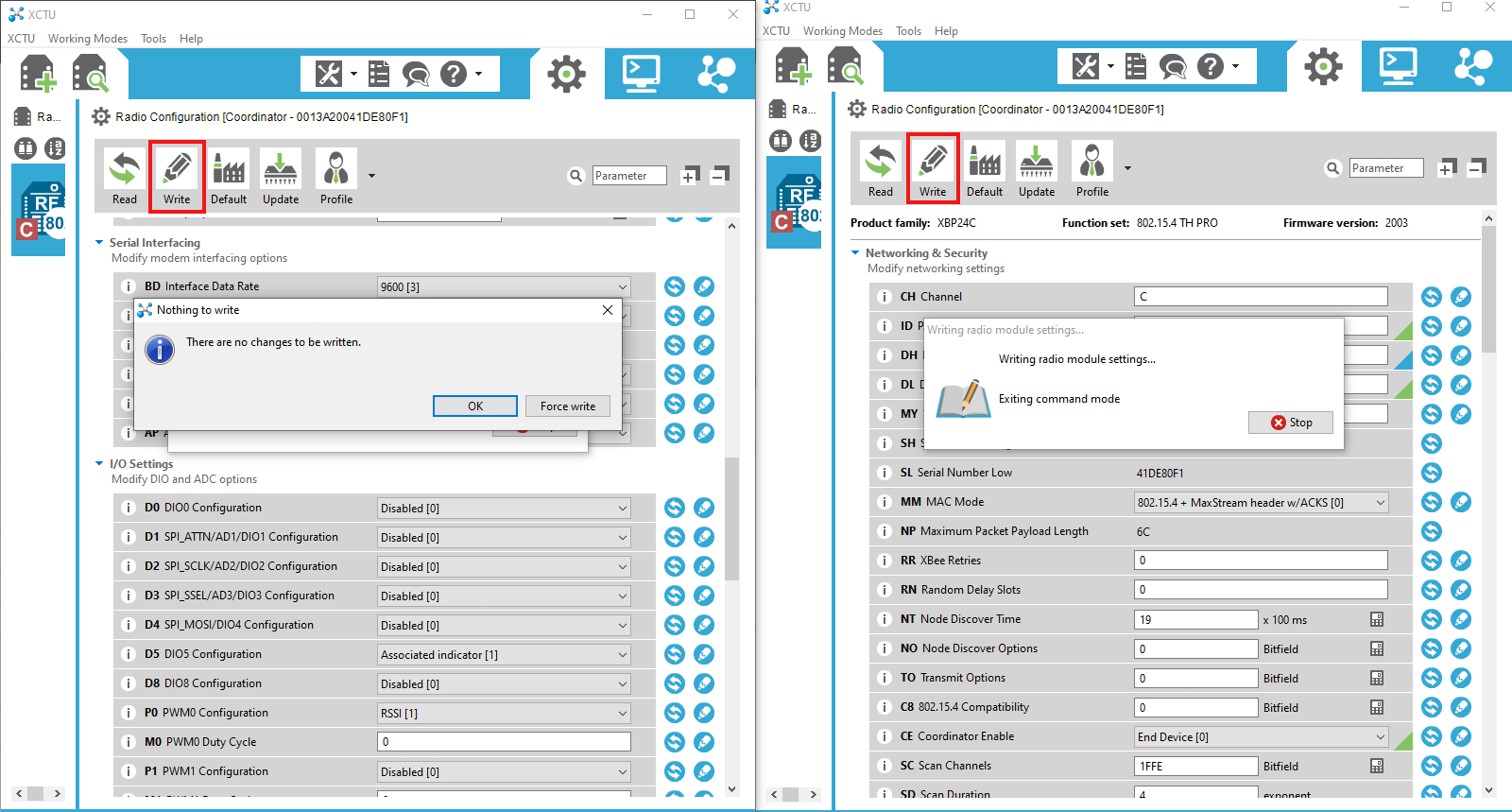


Node Identifiers are used to Name XBee Modules for easier understanding.





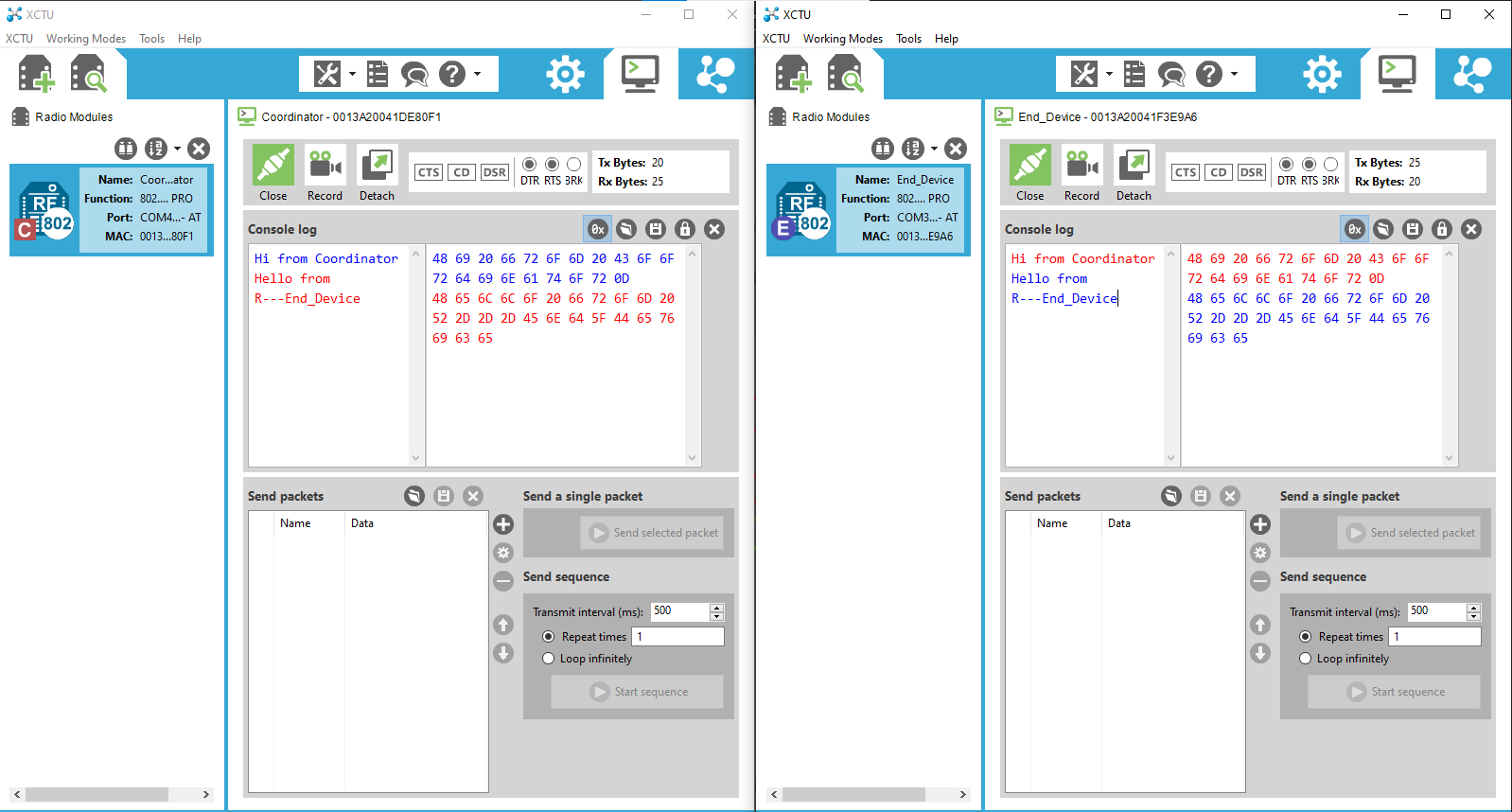
1. Now commit all the changes that you have made by clicking the write button on the top left.



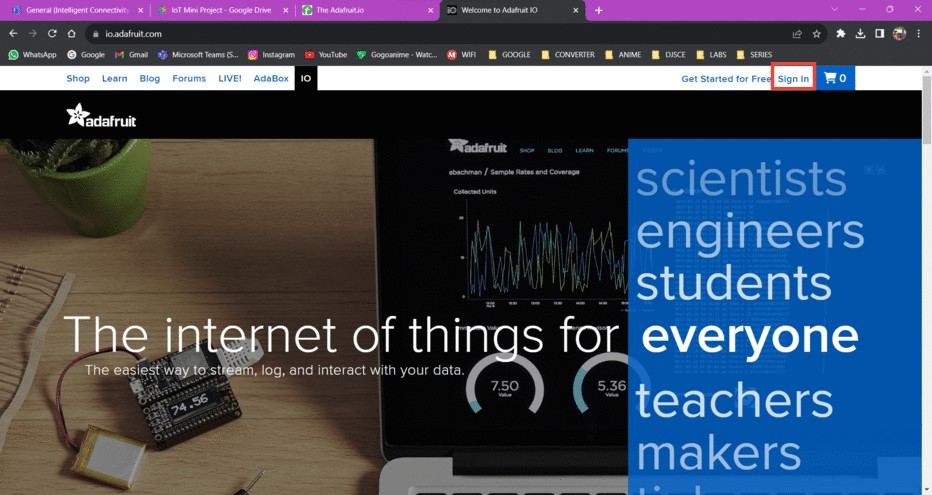
1. Now, in order to test the communication between these configured XBee Modules, Click on the Monitor Icon present on top right and then click on Open as highlighted below. We can also verify that our settings have been written correctly to the modules as we can check their name has changed on the left side.



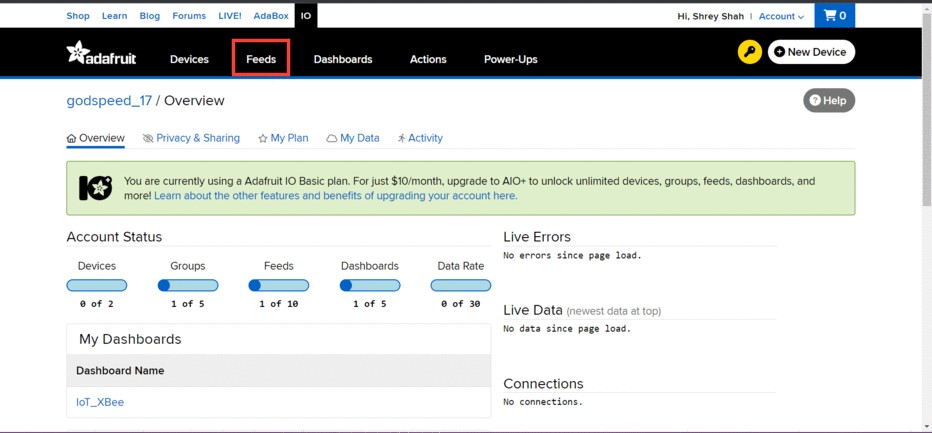
1. Type any message to verify the communication between the configured modules. The following picture indicates that the communication has been established successfully.



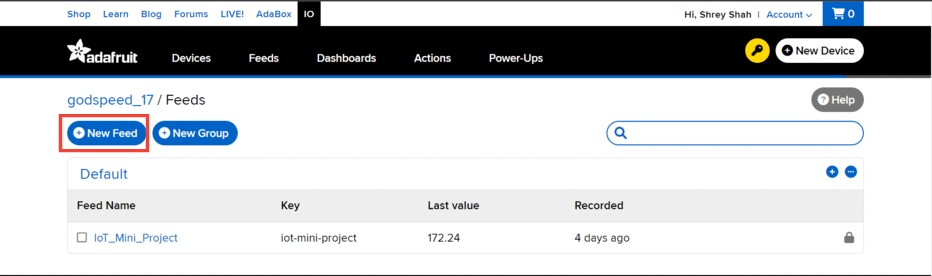
1. **Adafruit Cloud account setup:**
2. Go to [Adafuit Cloud](https://io.adafruit.com/) (io.adafruit.com)



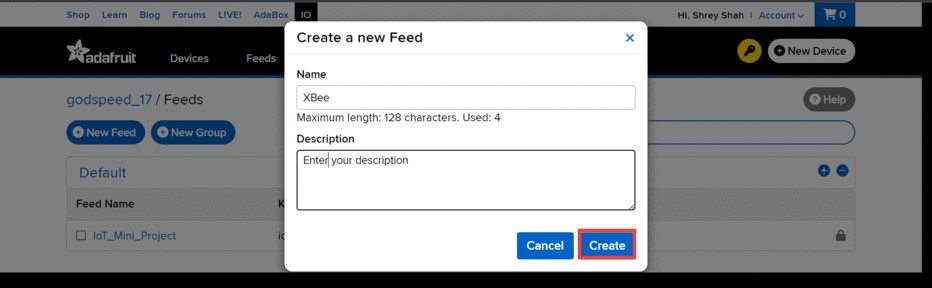
1. Login to your account and go to feeds



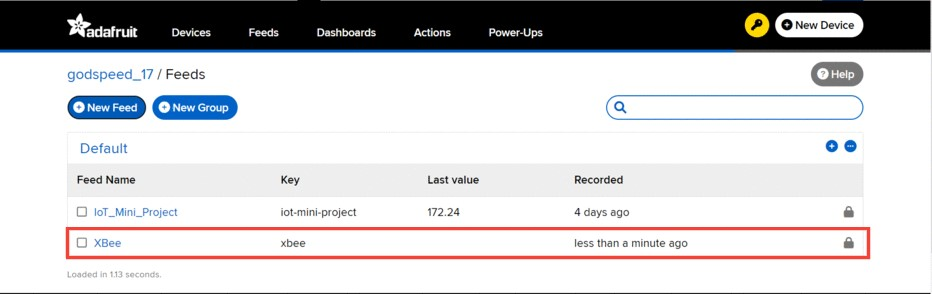
1. Go to Feeds and click on New Feed



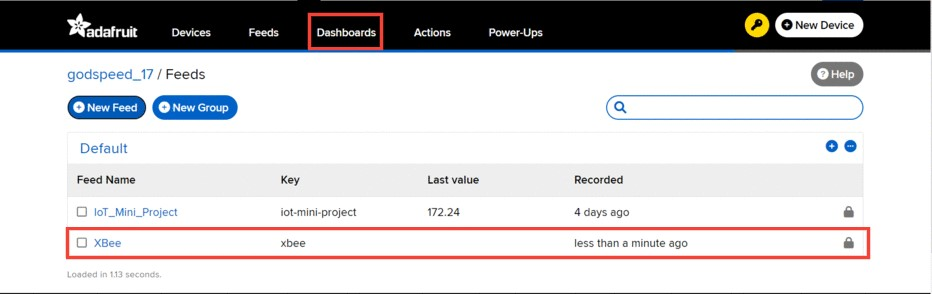
1. Enter your Feed Name and Details if required



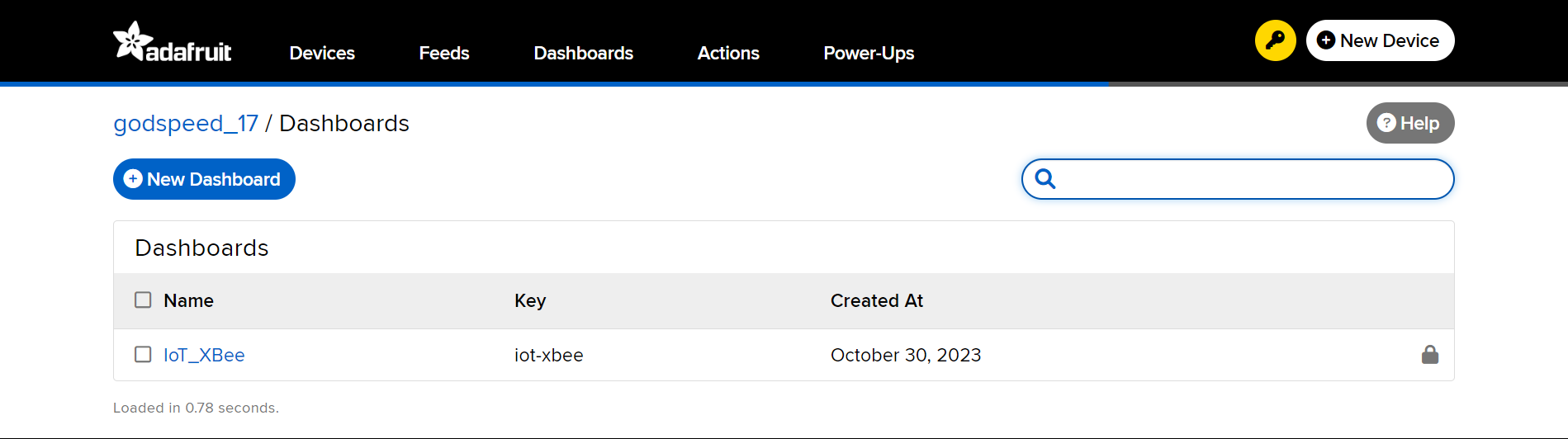
You will now be able to see the new feed created



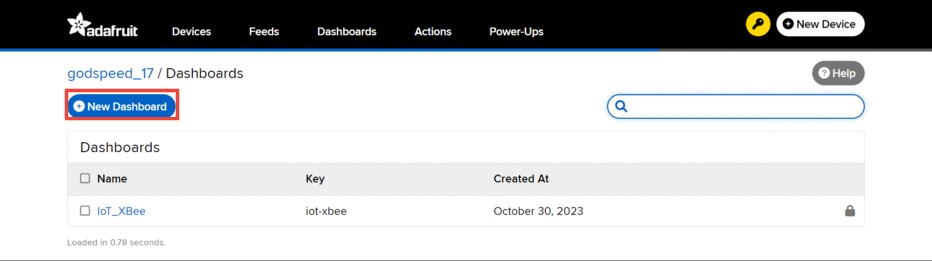
1. Now click on Dashboards



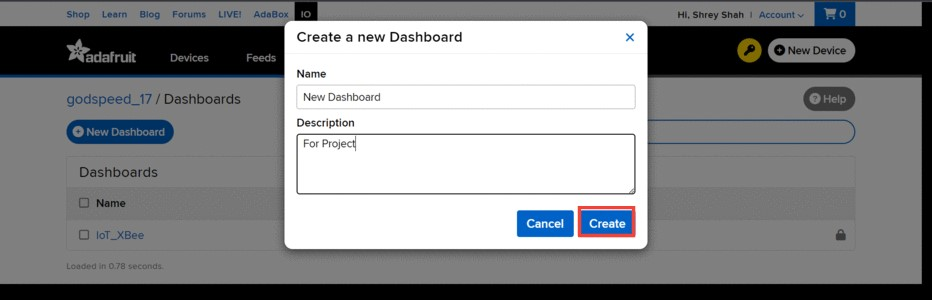
Such an interface will be shown (If you have not created any dashboard previously, it will be blank)



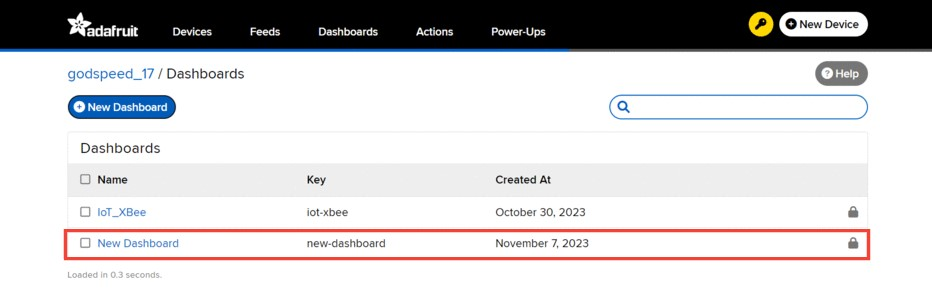
1. Click on New Dashboard



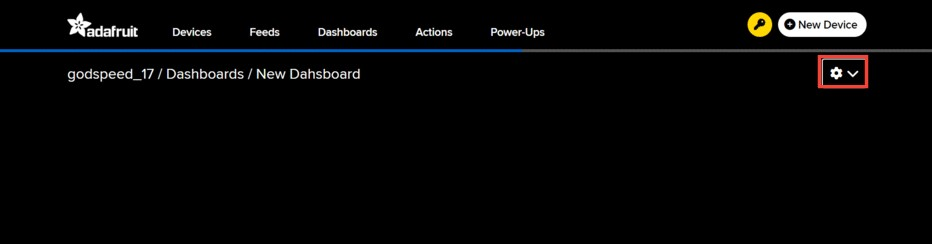
1. Write down your Dashboard name and Project details and click create.

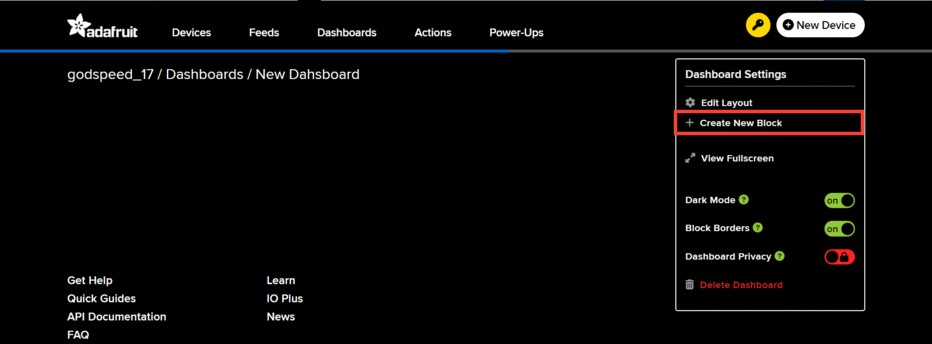


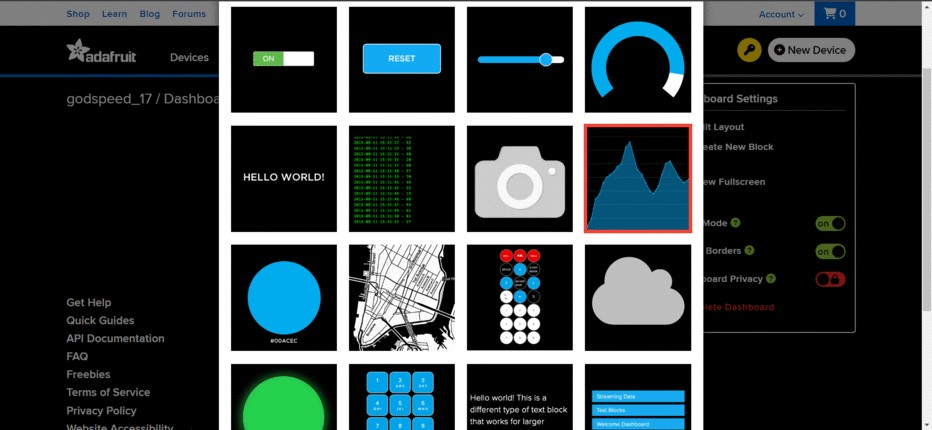
Such an interface will be shown



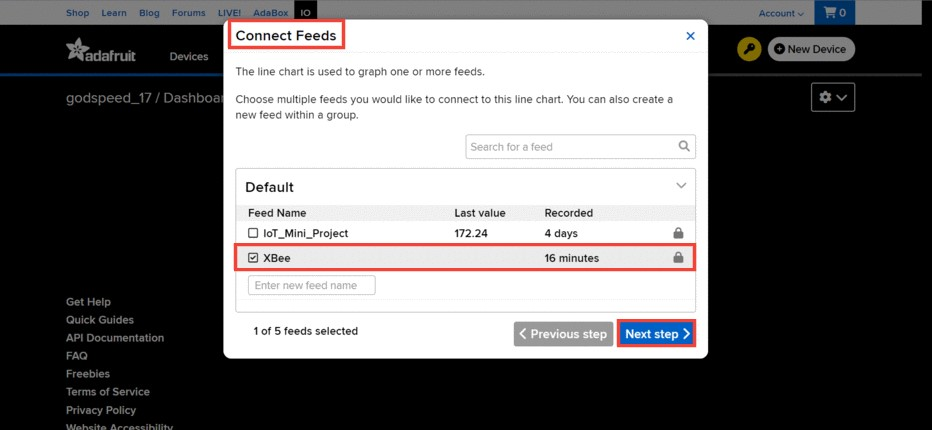
1. Click on the newly created Dashboard and Click on the Settings logo and choose your template. A small example is shown below



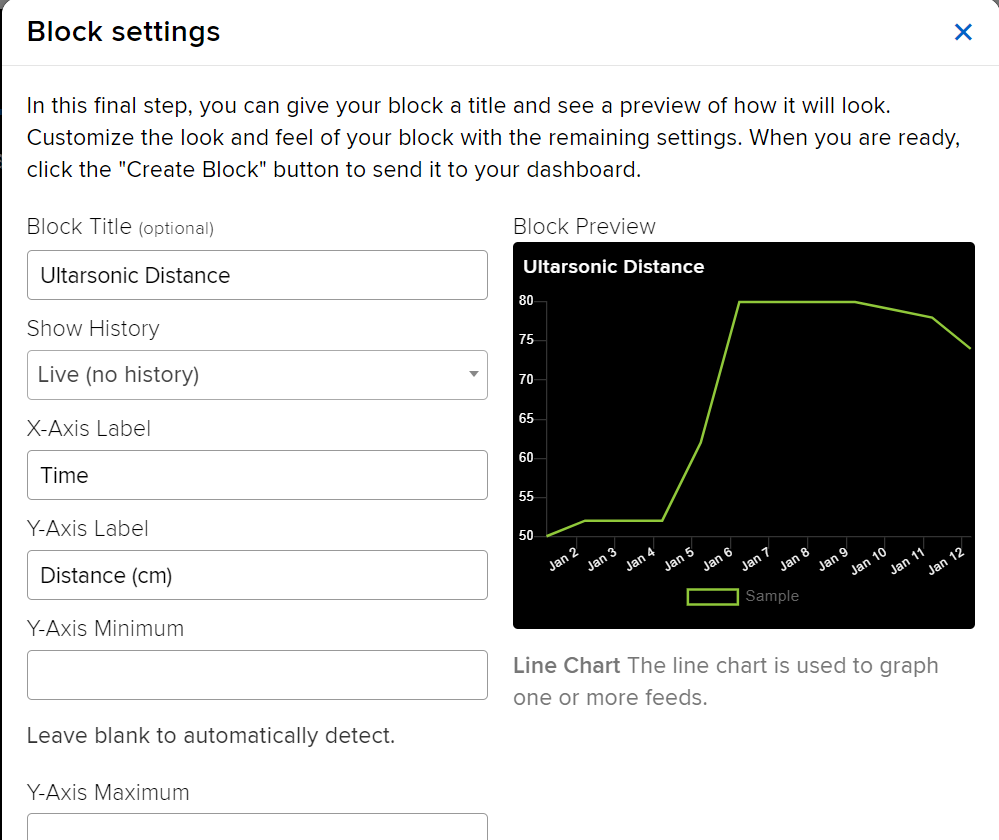


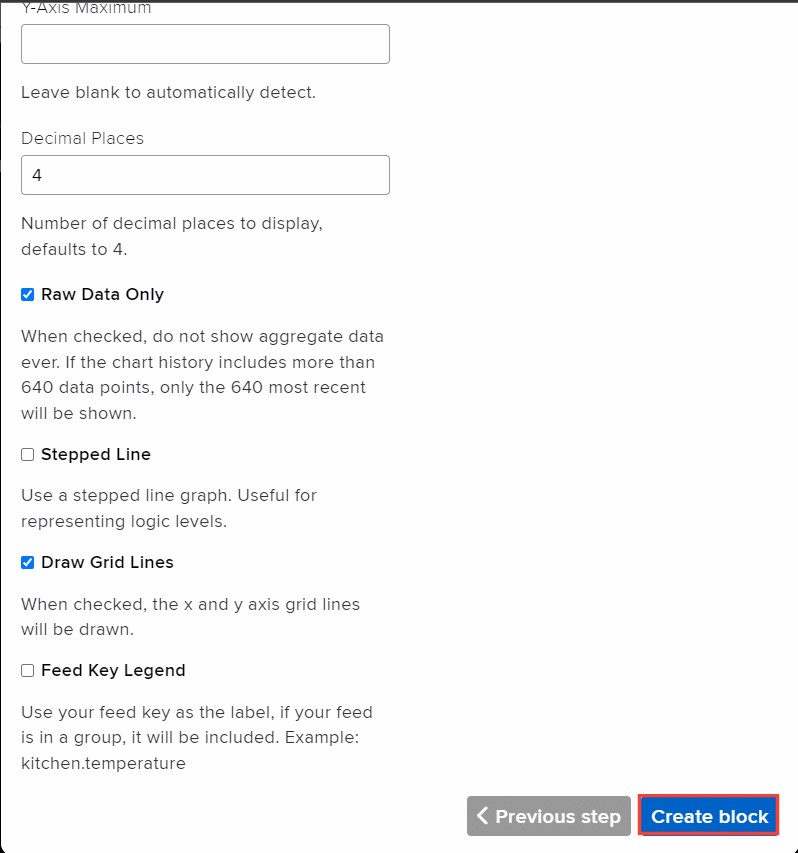


Click on Next Step

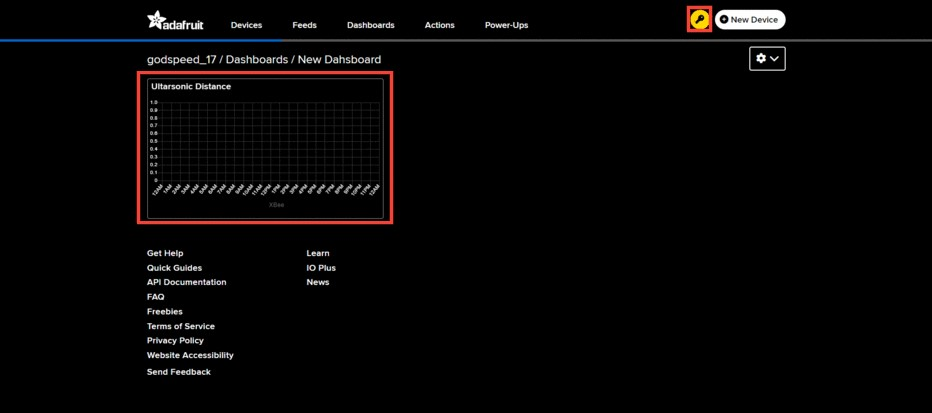


1. Carefully select the actual feed which you created in the above step and the click Next Step. Few configurations will be required for dashboard and few recommended settings are shown below. You can customise this according to requirement

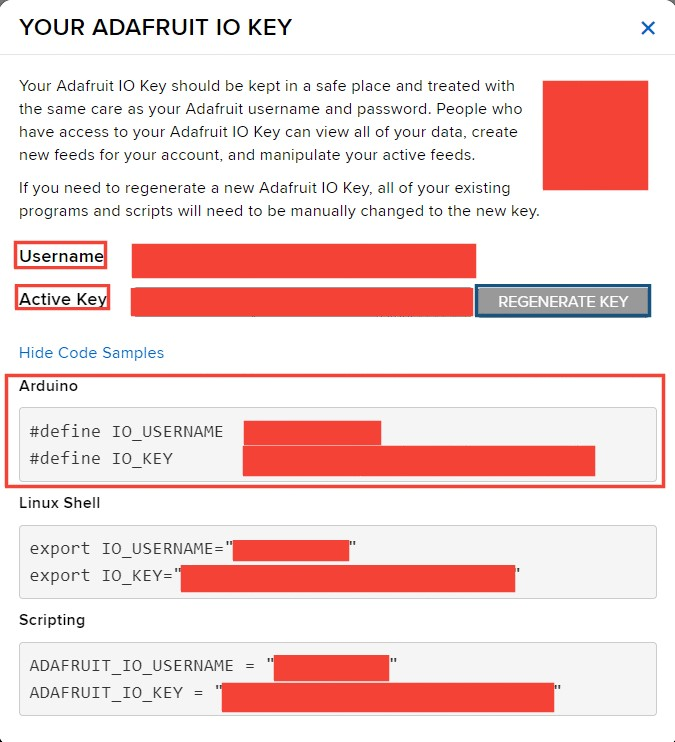




Click on Create block to display the selected settings on the dashboard. The dashboard is shown below

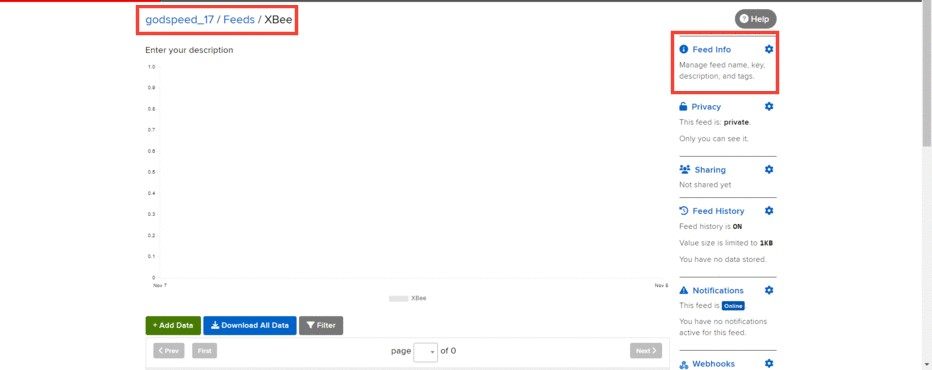


Note that all the requirements about Username, AIO Key while coding the ESP32 can be obtained by clicking the key symbol present on the top right corner.

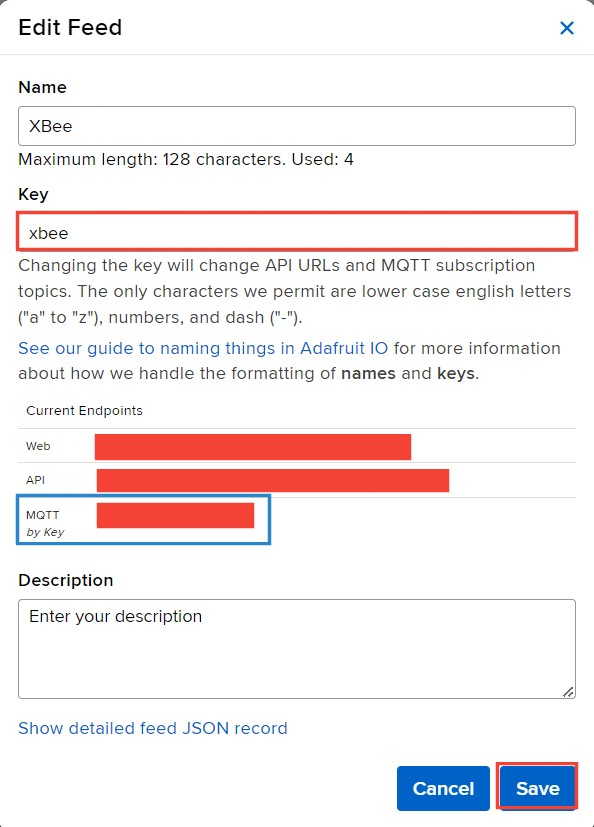


All the details regarding Username and AIO Key can be obtained through here. We can also regenerate AIO Key if needed. This format is readily available for Arduino Framework which is extremely useful in our case.

The rest of the details can be obtained by going to Feeds -> (Your current feed) -> Feed Info



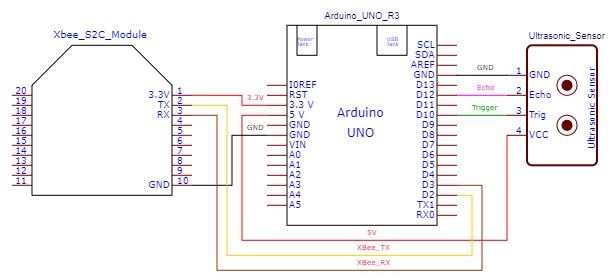
After clicking it, an interface will be shown which is given below



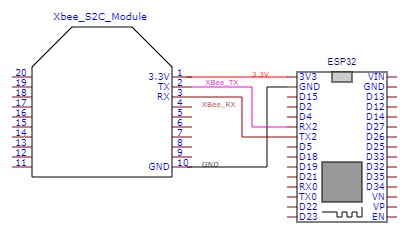
We can see the key value assigned to the feed. MQTT by key is extremely useful while coding as we will be using the MQTT Protocol

1. **Circuit Diagram:**

**Transmitter Side Circuit Diagram:**



**Receiver Side Circuit Diagram:**



1. **Output:**





