Connecting Your Device to Stream My IoT

In this workshop you will use the Intel Edison board to connect to streammyiot.com in order to send data, visualize data and trigger events. This represents the underpinnings for a majority of IoT use cases.

This tutorial assumes that you are familiar with the Intel Edison board and the Intel XDK IDE. As such this tutorial will **NOT** cover configuration of the Intel Edison or installation of drivers and the XDK development environment.

Pre-requisites

This tutorial assumes you have the following:

- A valid Stream My IoT account: http://streammyiot.com
- An Intel Edison board with internet access
- A development device (e.g. laptop) on the same network as the Intel Edison
- Terminal access to the Intel Edison from your development device
- Intel XDK IoT Edition installed on your development device

If you do not have any of the above please address before proceeding

Tutorial 1: Connecting the Intel Edison to Stream My IoT

1. Initiate a terminal session with your Intel Edison (either through SSH or Serial using your chosen client such as Putty or Screen).

2. Test your connection to the Stream My IoT service using the following command:

```
iotkit-admin test
```

You should see output similar to that below if the connection is successful:

```
root@edison:~# iotkit-admin test
2016-06-03T11:49:25.779Z - info: Trying to connect to host
2016-06-03T11:49:27.194Z - info: Connected to dashboard.us
2016-06-03T11:49:27.199Z - info: Environment: prod
2016-06-03T11:49:27.201Z - info: Build: 0.14.5
```

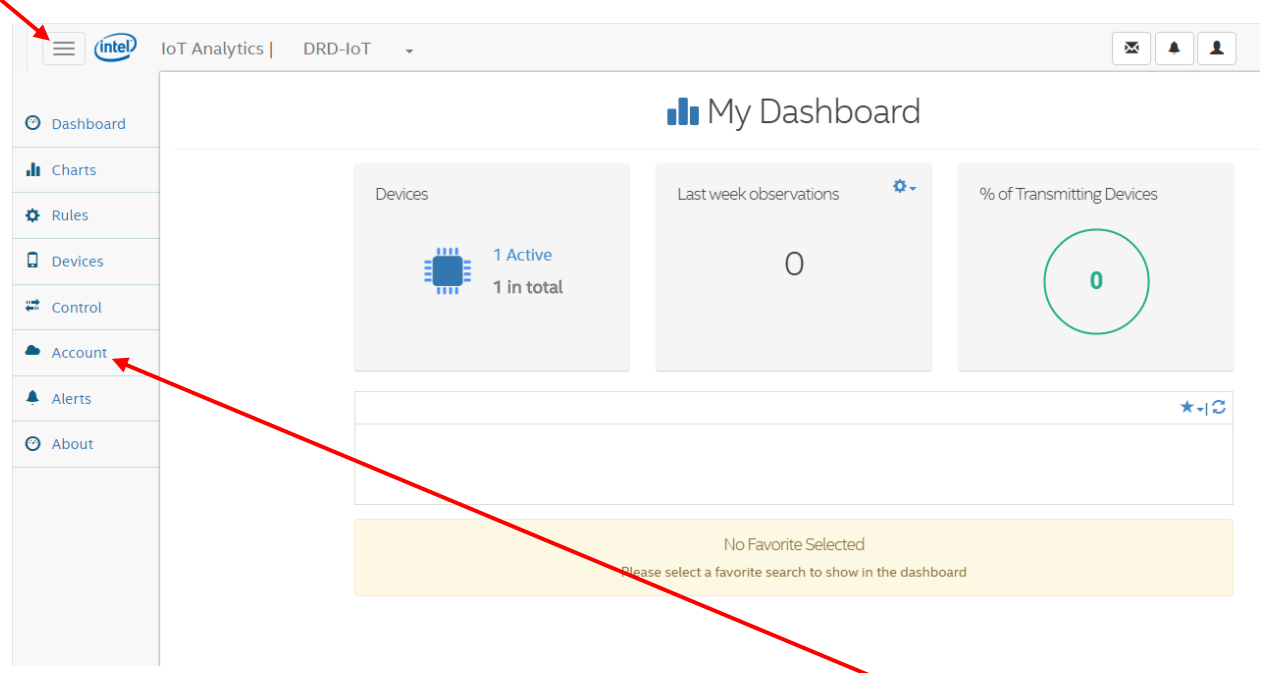
3. Set a permanent device ID for your Intel Edison:

```
iotkit-admin set-device-id {DEVICE_ID}
```

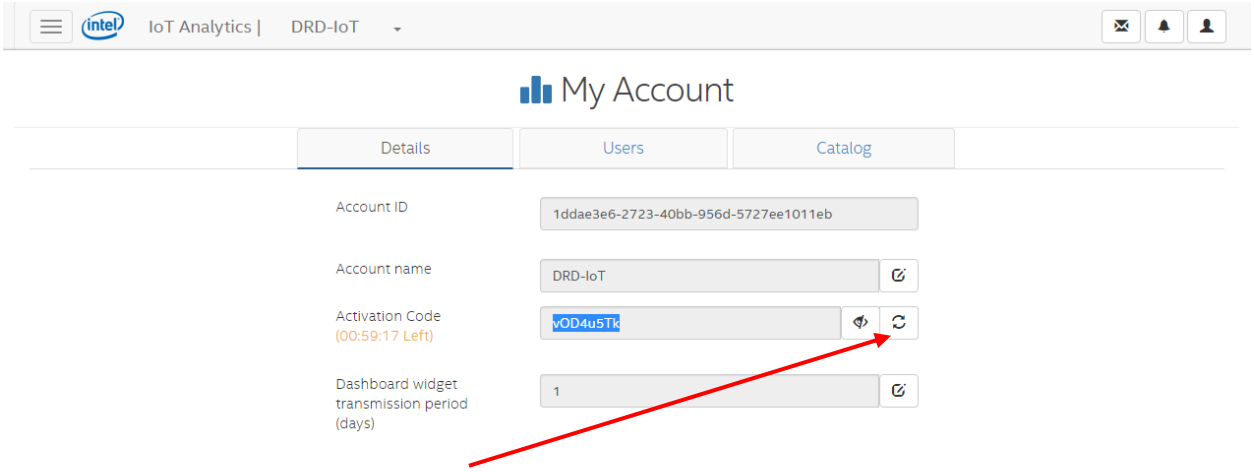
Example:

```
root@edison:~# iotkit-admin set-device-id intel-edison-cloud-01
2016-06-03T12:55:39.566Z - info: Device ID set to: intel-edison-cloud-01
```

4. Activate your Intel Edison in your Stream My IoT account.



Login to your Stream My IoT dashboard in a web browser and select '*Account*' in the left panel.



In the '*Account*' section click the '*refresh*' button to get a new '*Activation Code*'.

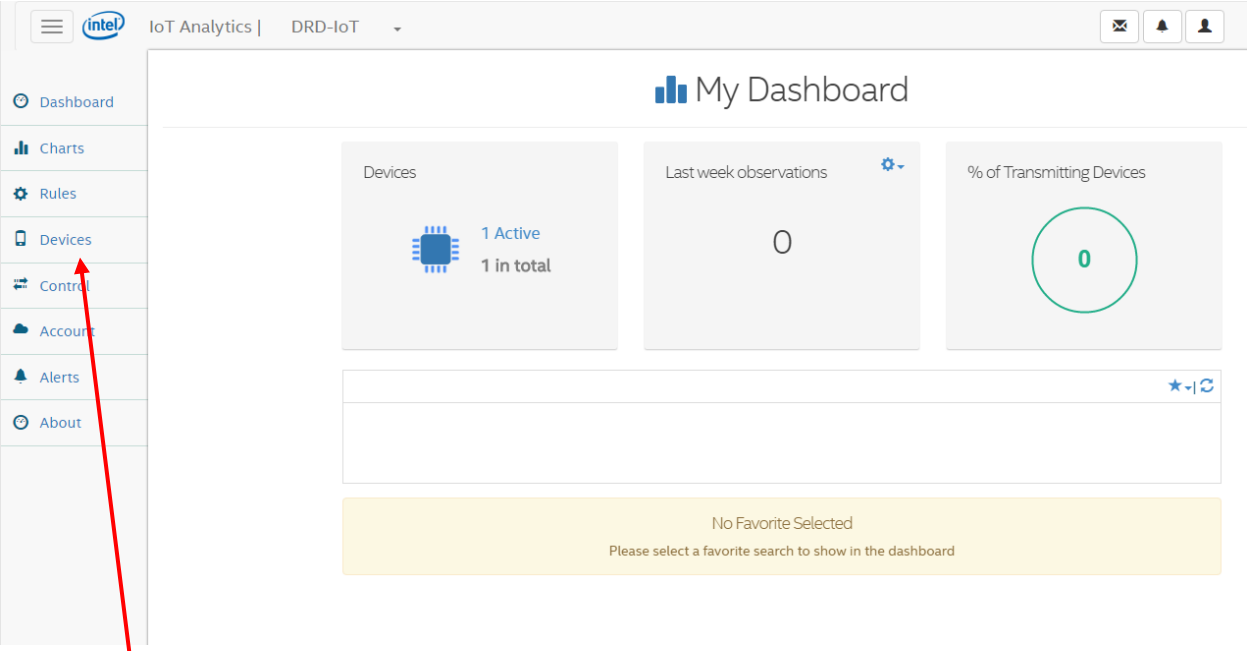
Click the '*View*' button to see the activation code and make a note of it.

Enter the following command in your terminal window to activate your Edison using the activation code:

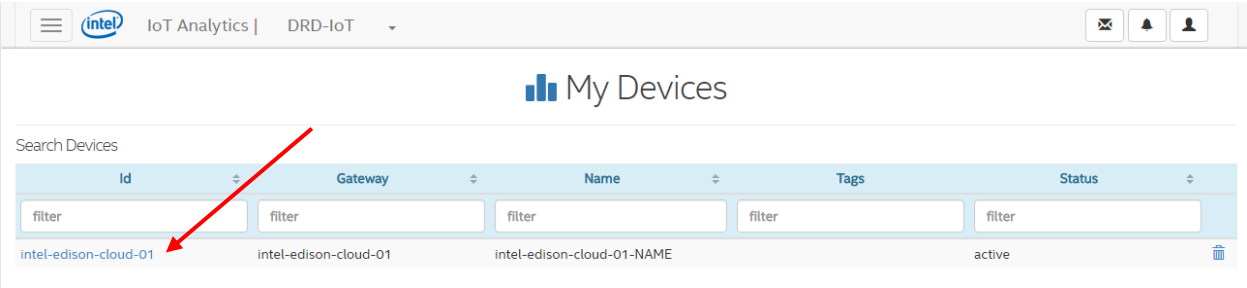
```
iotkit-admin activate {ACTIVATION_CODE}
```

Example:

```
root@edison:~# iotkit-admin activate vOD4u5Tk
2016-06-03T13:16:00.680Z - info: Activating ...
2016-06-03T13:16:03.367Z - info: Saving device token...
2016-06-03T13:16:03.520Z - info: Updating metadata...
2016-06-03T13:16:03.533Z - info: Metadata updated.
```



Select '*Devices*' from the left panel to take you to the device management section of the dashboard.



A device with the ID you specified earlier should now appear in the devices list.

You have successfully connected your Intel Edison to the Stream My IoT service.

Tutorial 2: Sending sensor data to Stream My IoT

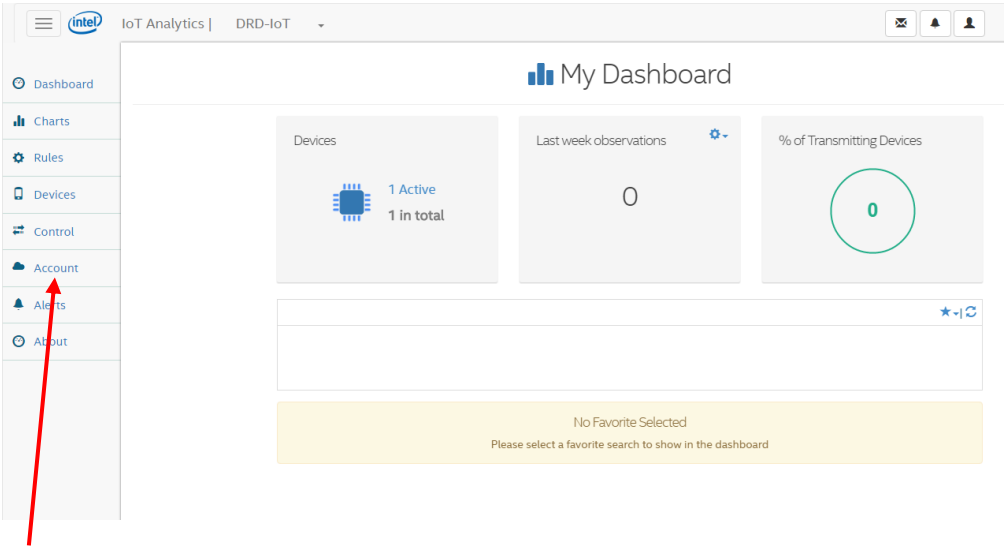
To complete this tutorial you will need the following:

- Grove Base Shield
- Grove LCD RGB Screen
- Grove Light Sensor
- Grove Button
- XDK project template: https://github.com/srware/WHD.global-2017/raw/master/Projects/sensor_actuator_example.zip

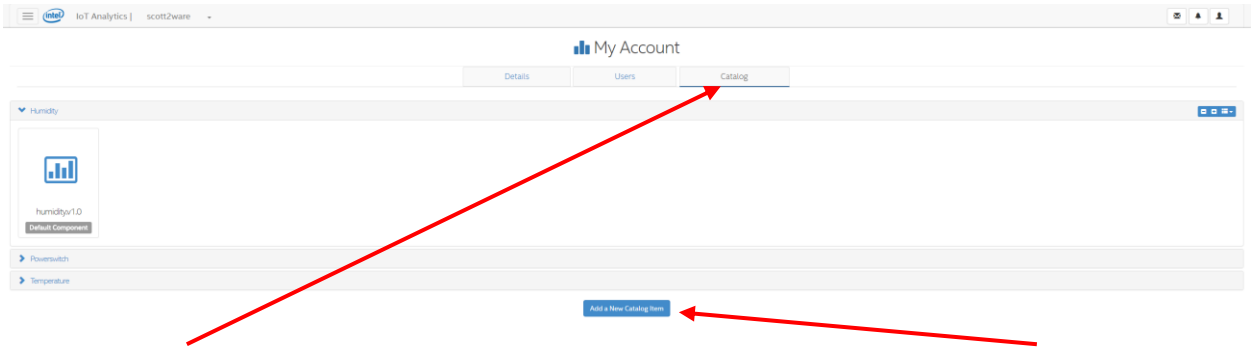
1. Connect the sensors to the base shield as follows using the wires provided:
    - Grove Light Sensor -> A0
    - Grove Button -> D2
    - Grove LCD RGB Screen -> I2C

Ensure the VCC switch on the edge of the base shield is set to 5V

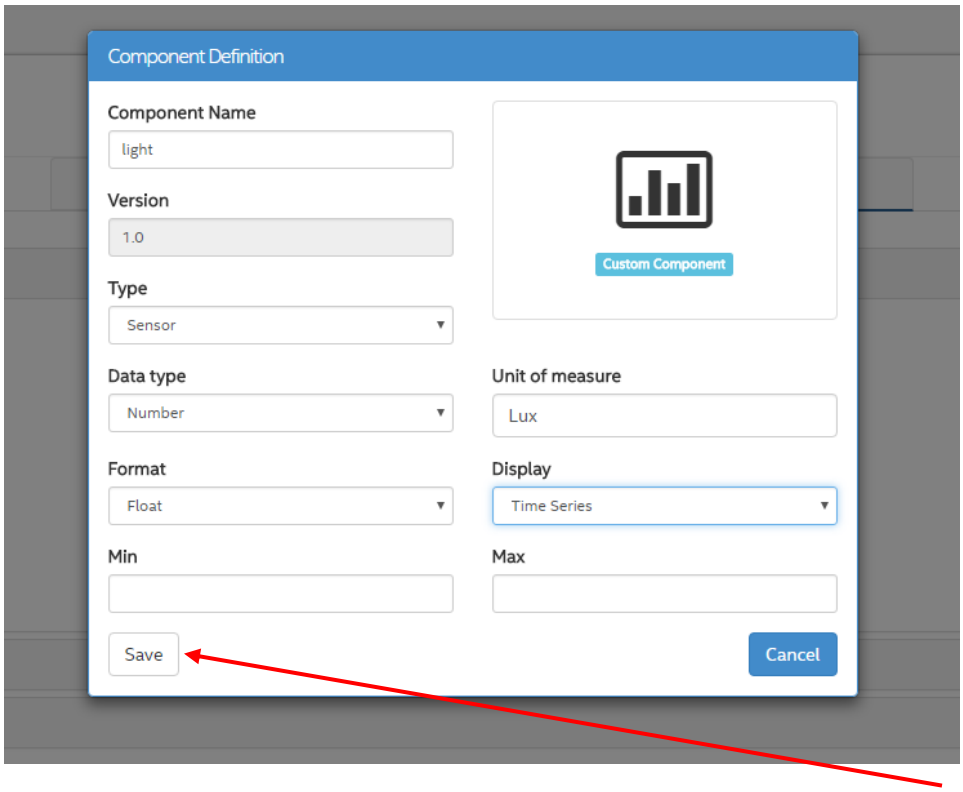
2. Add custom catalog items to Stream My IoT.



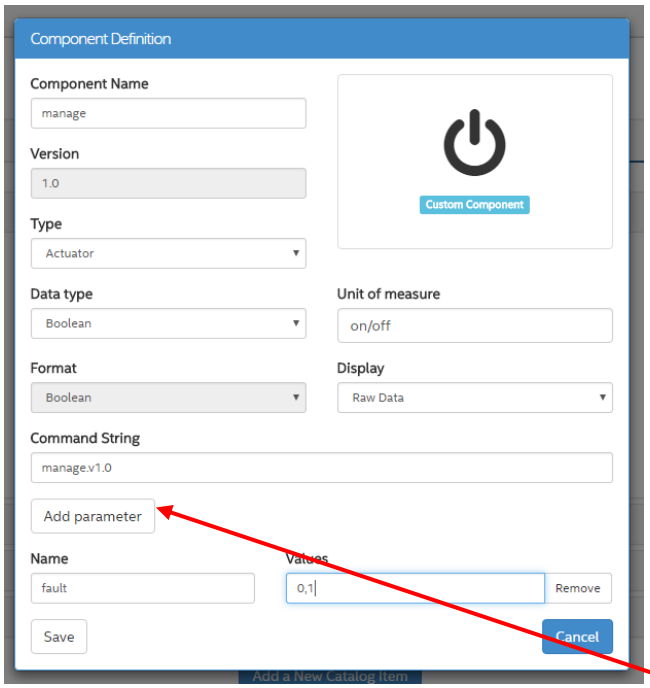
Select '*Account*' from the left panel of your Stream My IoT dashboard.



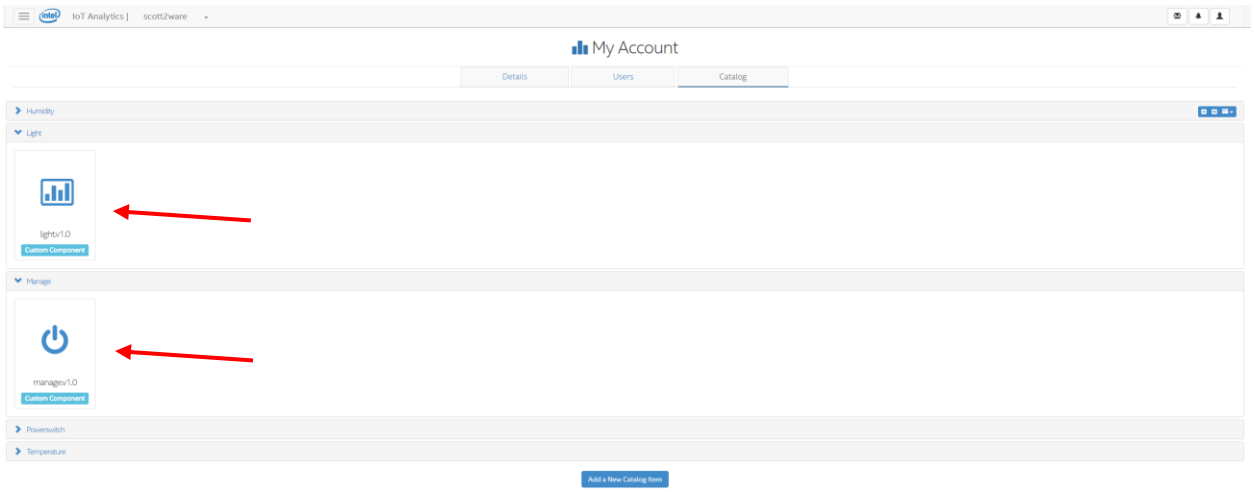
Go to the '*Catalog*' tab to get a list of catalog items available in your account and click '*Add a New Catalog Item*'.



Add a new catalog item called '*light*' using the screenshot above for reference and click '*Save*'.



Add another new catalog item called '*manage*' with the above settings clicking '*Add Parameter*' to add a '*fault*' parameter. When done once again click '*Save*'.

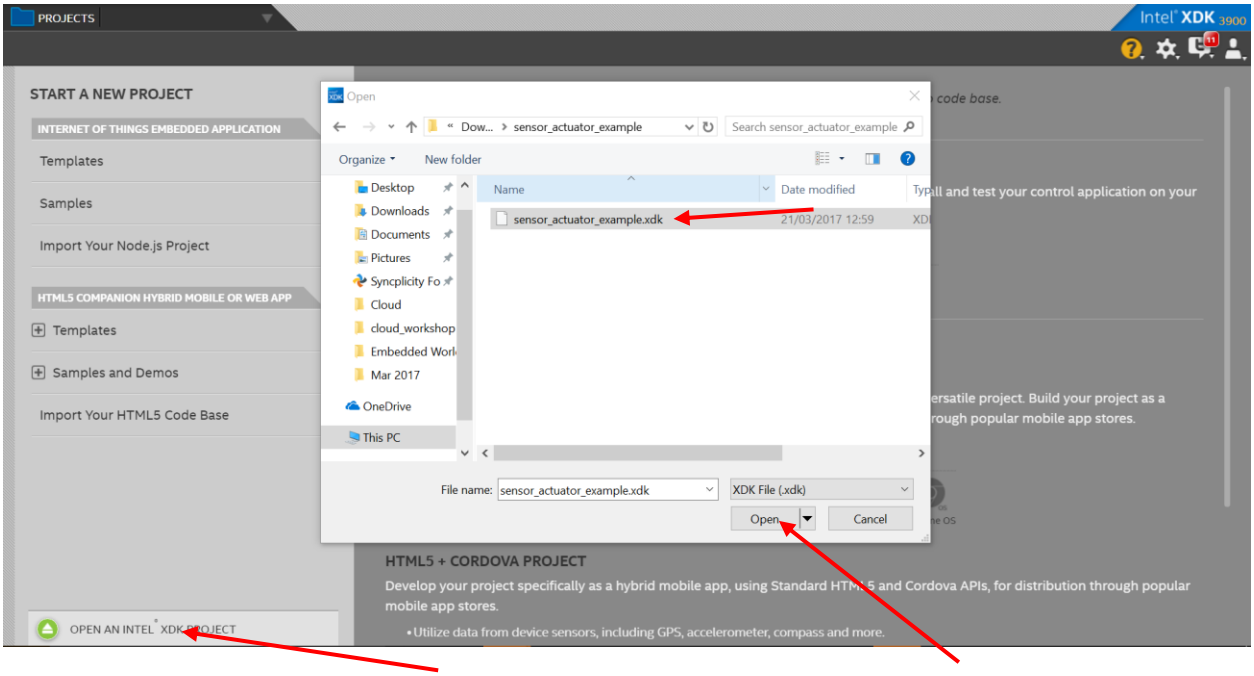


You should now see your new catalog items in the catalog list.

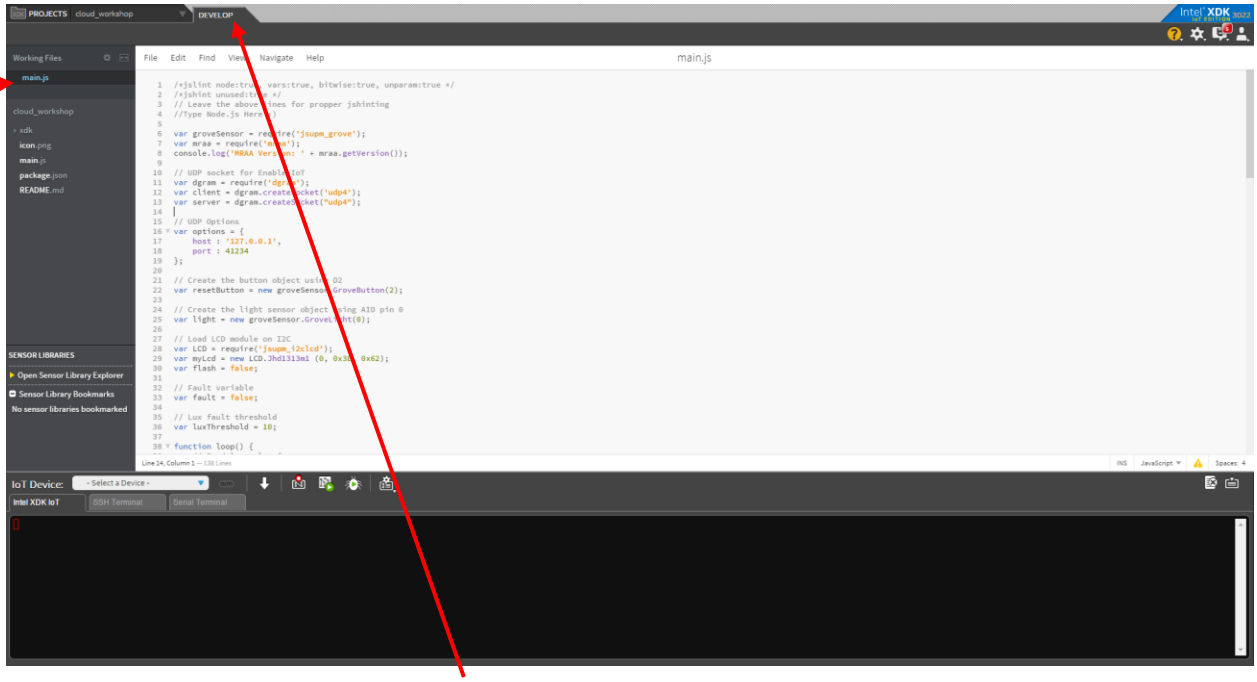
3. Now you have created the correlating catalog items you can register the sensors connected to the Edison on Stream My IoT using the commands below:

```
systemctl restart iotkit-agent

iotkit-admin register light light.v1.0

iotkit-admin register manage manage.v1.0
```

4. Open the Intel XDK IoT Edition on your development device and open the '*sensor_actuator_example*' project you downloaded earlier (if you haven't already done so extract the .zip to your desired location).

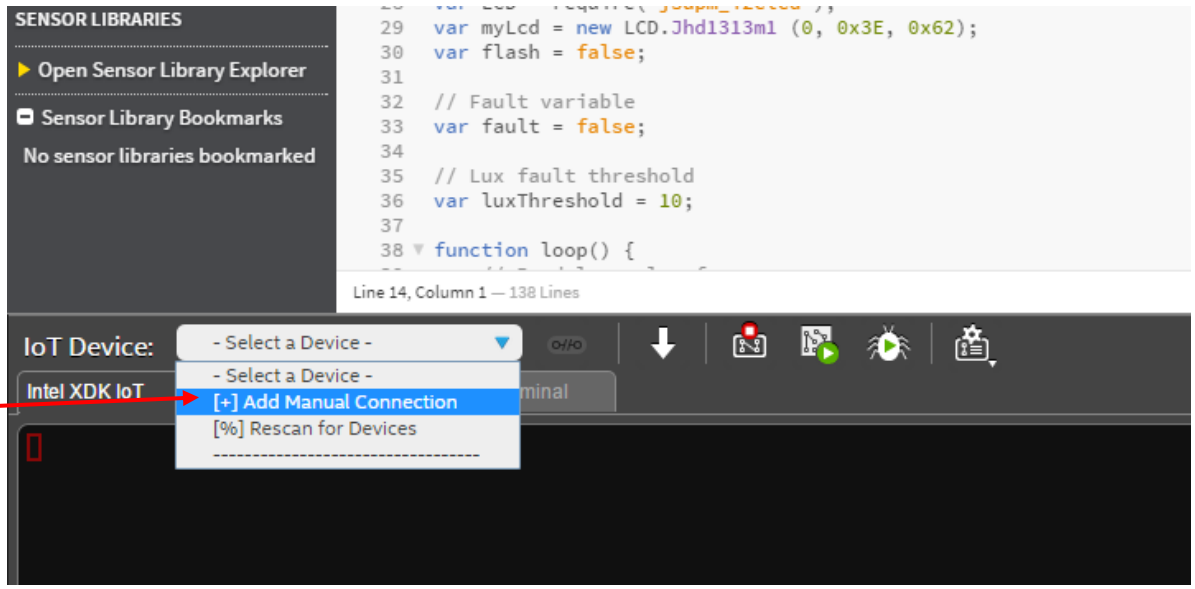


In the XDK interface click '*Open An Intel XDK Project*' and navigate to the '*sensor_actuator_example*' directory. Click on the '*sensor_actuator_example.xdk*' file and click '*Open*'.

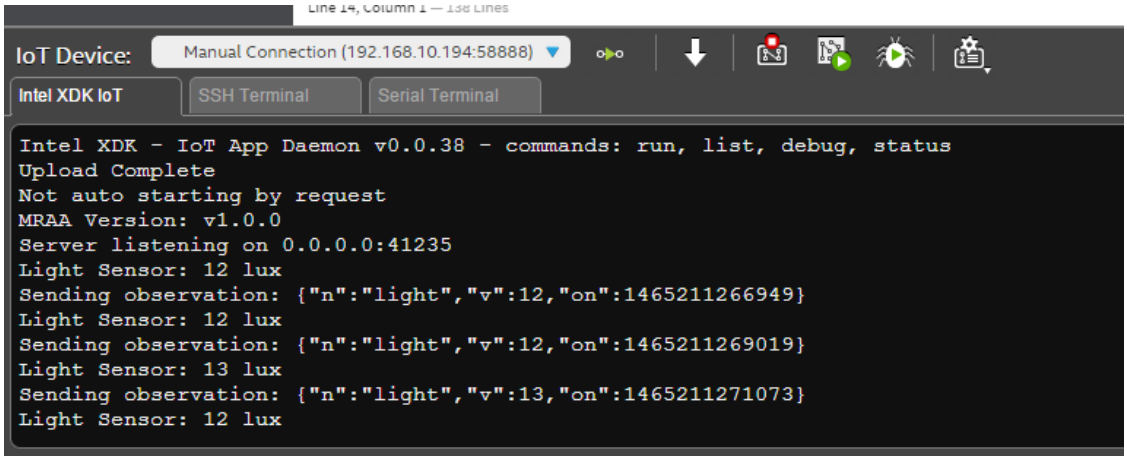


Once the project is open click the '*Develop*' tab in the top bar to take you to the project code. The '*main.js*' file should be loaded by default.

5. Connect XDK to your Edison's IP address. If you are using OS X, Linux or have '*Bonjour*' installed on Windows the Edison should be auto-detected and displayed in the '*IoT Device'* list but when several Edison's are on the same network it is often safest to get your Edison's 'wlan0' address using the '*ifconfig*' command and using the '*Add a manual connection*' option in XDK.



Once you have successfully connected to your Intel Edison click the '*upload*' button to send the project to the board. Once you get an '*Upload Complete*' message in the console click the '*Run*' button to start the application.



If the application is running correctly you should get console output similar to the screenshot above.

Description

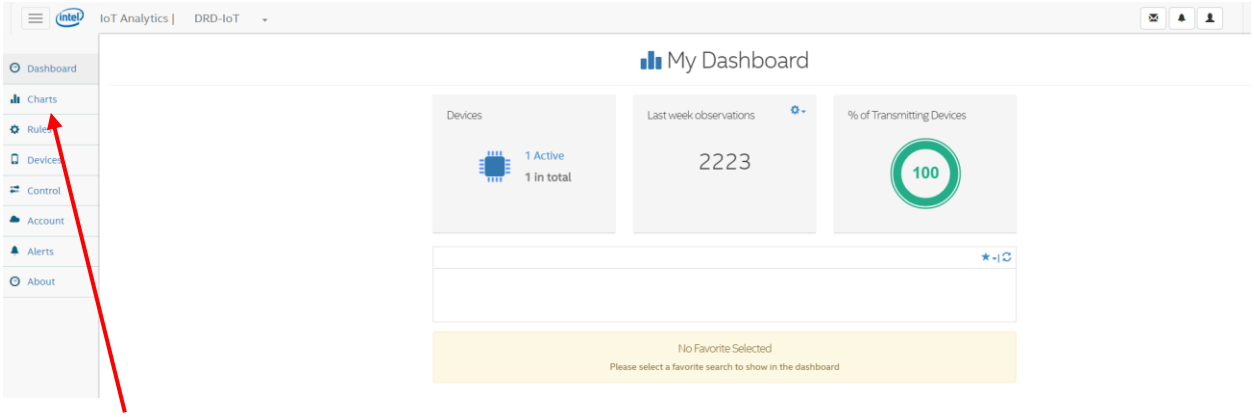
The code in this project grabs a sensor reading from the light sensor and does a number of things. Firstly it checks this value against a threshold (10 Lux by default) and if it is below this value triggers a '*fault*' condition. Secondly this value is uploaded to Stream My IoT and displayed on the LCD screen.

If a fault condition occurs the LCD screen displays a message to indicate this and begins flashing red. A fault condition can be reset with the button connected to the Edison.

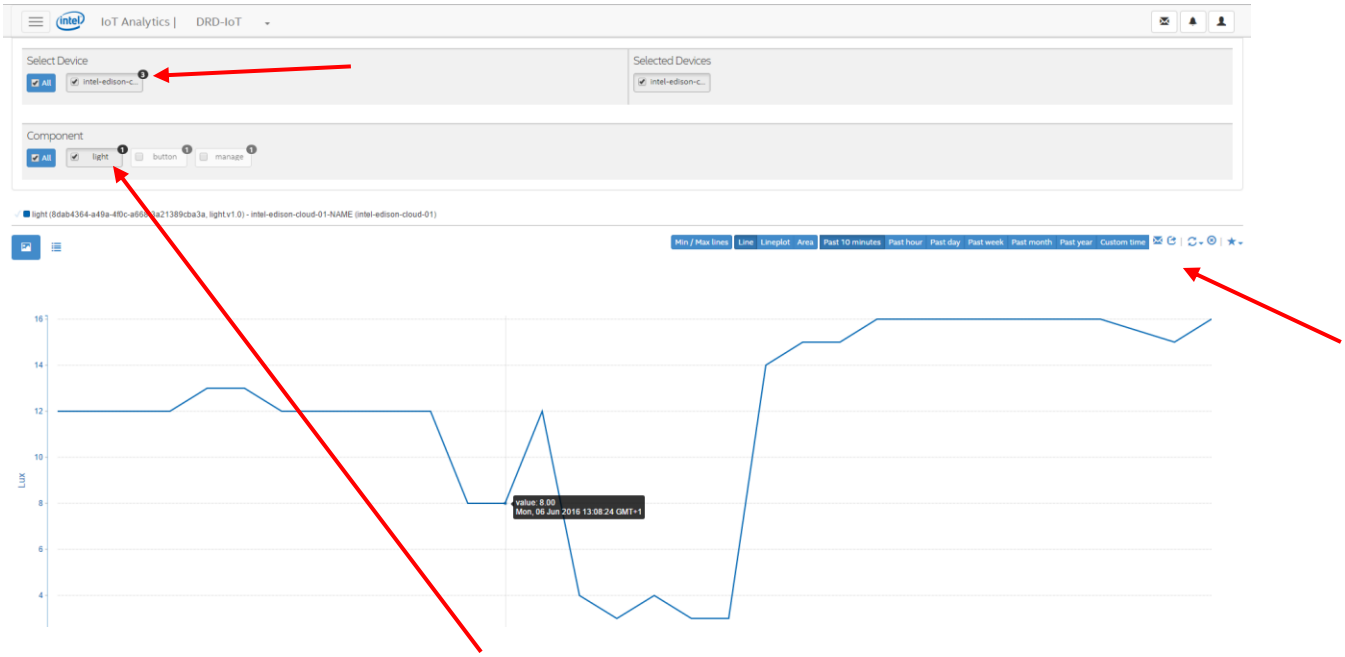
This hardware could be used for street lighting to detect faults. If the light state is 'on' but the lux value is low it is likely that the light has a fault. The screen shows a visual indication to an engineer which light has a fault and once it is fixed they can reset with the hardware button.

By sending the data to the cloud this fault condition can also be picked up by monitoring the data in the Stream My IoT dashboard. In the next tutorial you will setup rules in Stream My IoT to automatically raise an alert when a fault occurs and send an email. You will also setup controls to be able to remotely reset a light when a fault occurs or trigger a fault condition manually.

6. Visualise the light data on your Stream My IoT dashboard.



Select '*Charts*' from the left panel in your Stream My IoT dashboard.

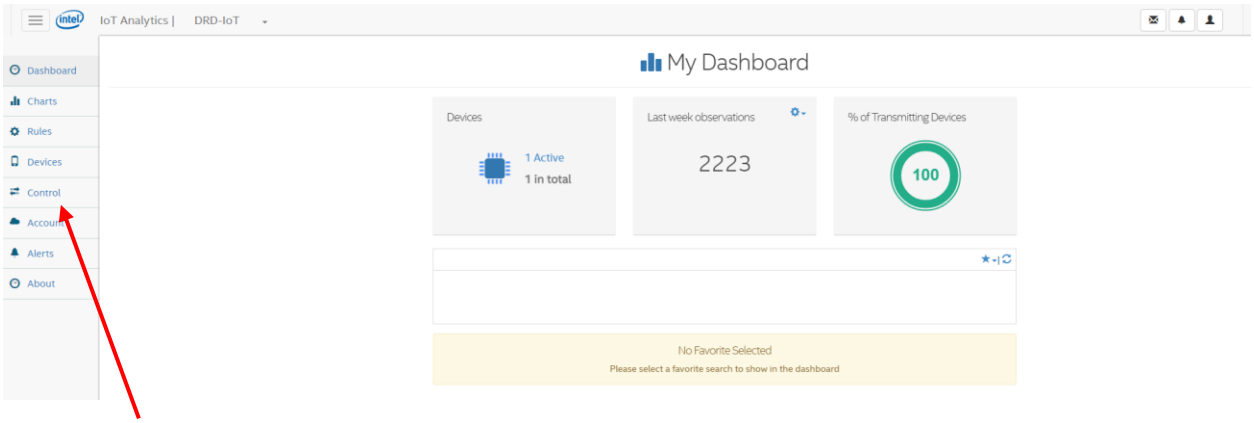


Select your Edison device and select '*light*' from the '*Component*' section. The data will then be displayed in a graph at the bottom of the page. Play with the settings on the right hand side to change the timeline, style and refresh rate of the chart.

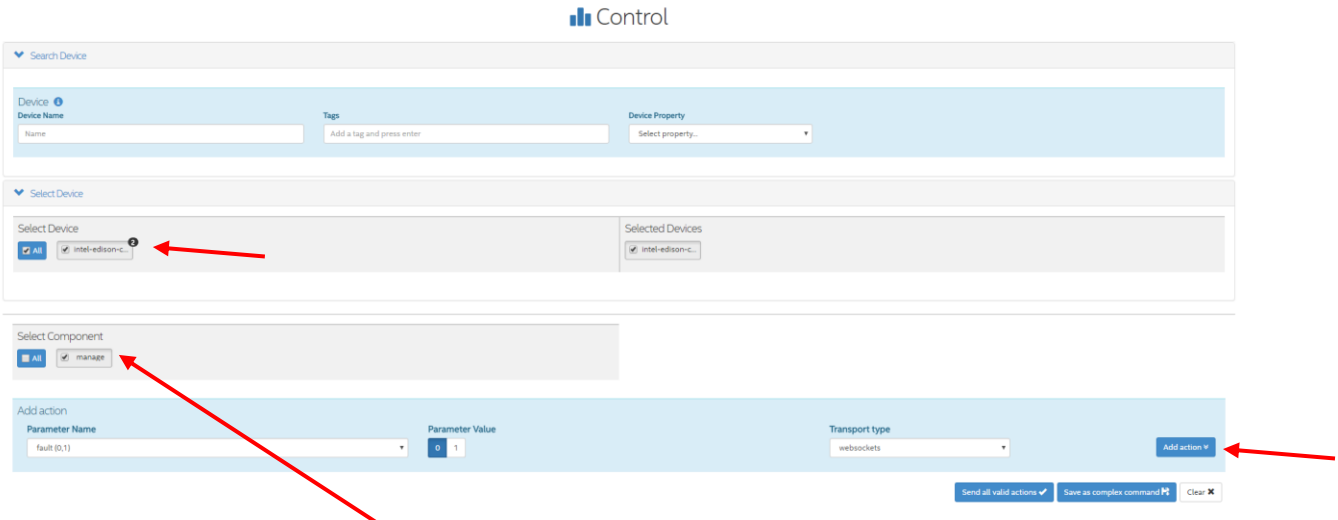
Tutorial 3: Configuring rules and controls in Stream My IoT

Sending data to the cloud is only useful if actions can be triggered as a result. For instance in our scenario we don't want to manually monitor the data from each light to detect if a fault may have occurred, we want this to be detected and brought to our attention automatically. We also want to be able to remotely manage the light which we will cover in this tutorial.

1. Add control commands for your device.



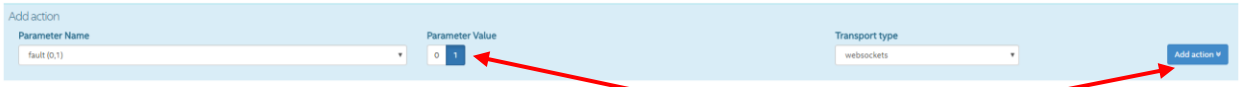
Select '*Control*' from the left panel in your Stream My IoT dashboard.



Select your Edison device and '*manage*' in the '*Select Component'* section.

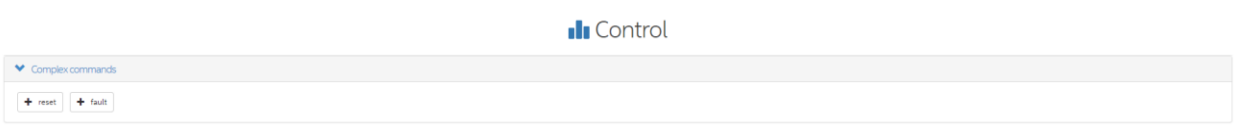
In the '*Add Action*' section select '*fault*' from the '*Parameters*' list, select '*0*' for the value and '*websockets*' for the '*Transport Type*' and click '*Add Action*'. The action will then be shown in the list at the bottom of the page.

Click '*Save as complex command*' and in the dialogue box type '*reset'* as the name and click '*OK*'.



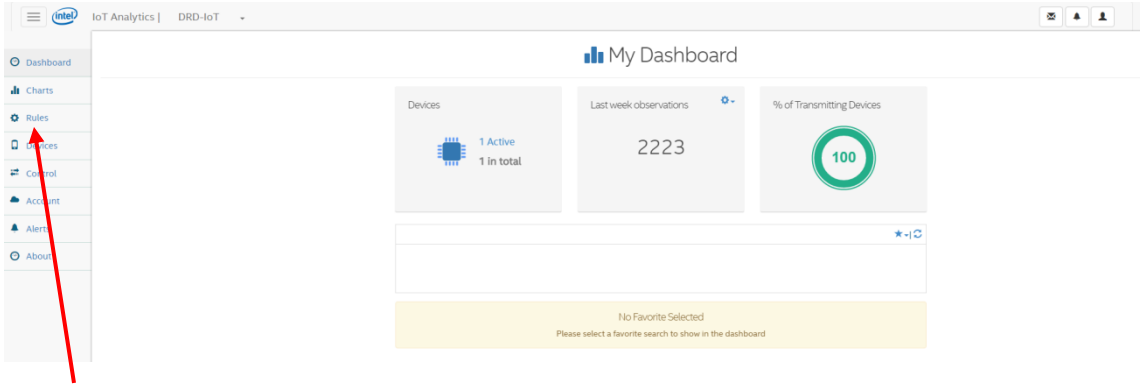
Keeping everything else the same, change the '*fault*' value to '*1*' and click '*Add Action*' again.

As before click '*Save as complex command*' and in the dialogue box type '*fault*' as the name and click '*OK*'.



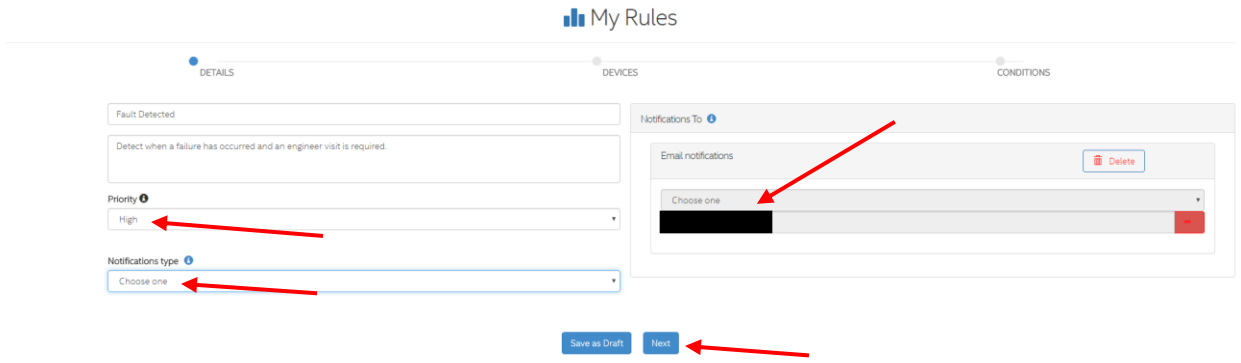
At the top of the page you should now have two '*Complex Commands*' available called '*reset*' and '*fault*'.

2. Add a rule to Stream My IoT to detect a fault.

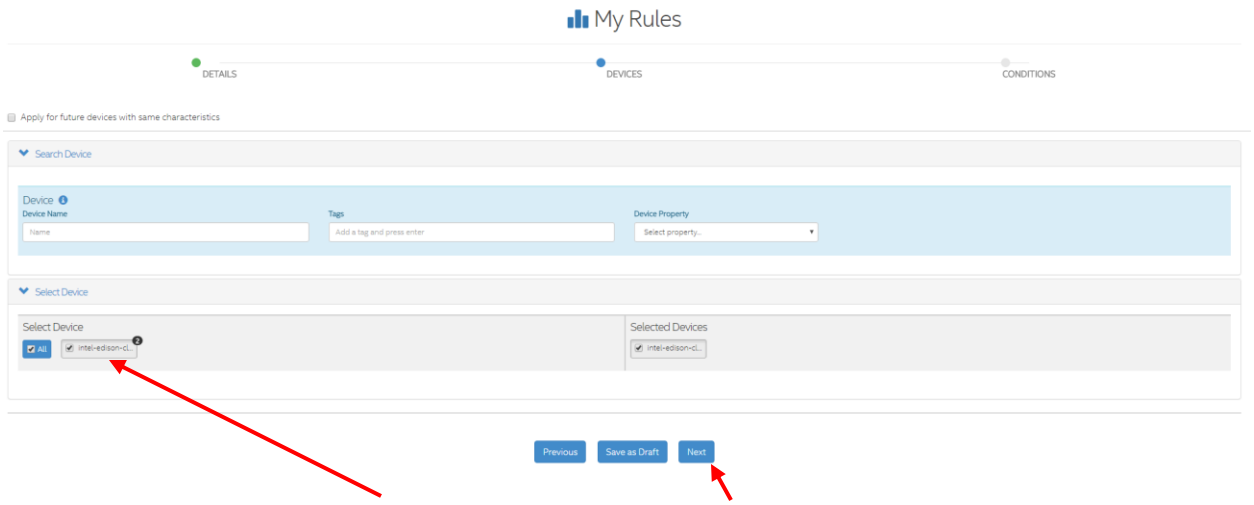


Select '*Rules*' from the left panel in your Stream My IoT dashboard.

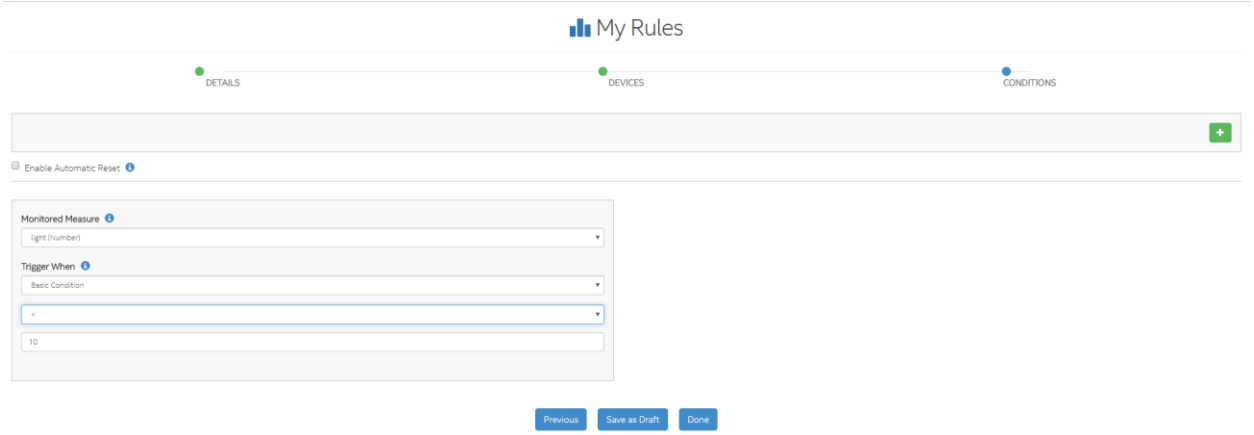
Click '*Add a Rule*' to start configuring a custom rule.



For the '*Details*' section enter a name and description and select '*High*' priority. In '*Notification Type*' select '*Email*' and choose an email address to send alerts to on the right-hand side. Finally click '*Next*'.



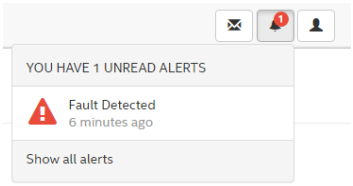
In the '*Devices*' section select your Edison device and click '*Next*' to continue.



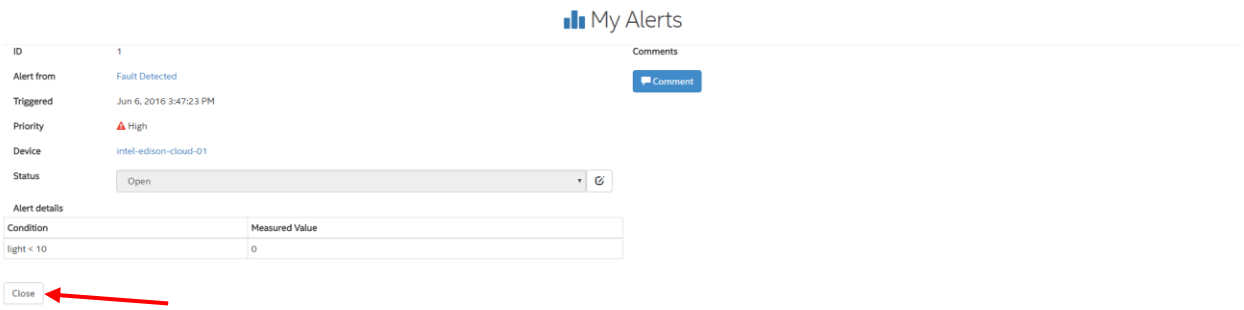
For the '*Conditions*' section select '*light*' as the '*Monitored Measure*' and set a '*Basic Condition*' of *'< 10'*. Finally click '*Done*'.

You have now created a rule which generates an alert when the '*light*' level from the sensor drops below 10 lux which would indicate the light has failed. An email will also be sent to your account. A similar system could be used to automatically notify engineers of a failure for example.

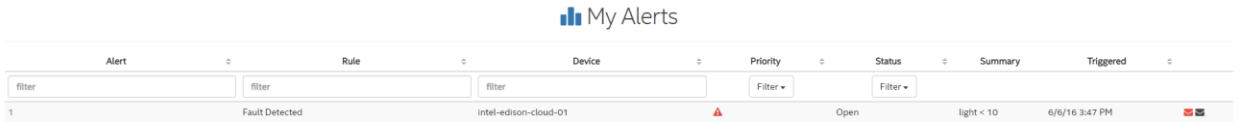
3. Test the system by covering the light sensor with your finger for a few seconds. This should cause a failure condition indicated by the LCD screen flashing red. You will also be able to see this visually in the '*Charts*' section of your Stream My IoT dashboard.



If your rule is working correctly you will also get an alert indicated in the top right corner of your Stream My IoT Dashboard. **Note that the alerts system only refreshes every 5 minutes by default so you will need to be patient!**

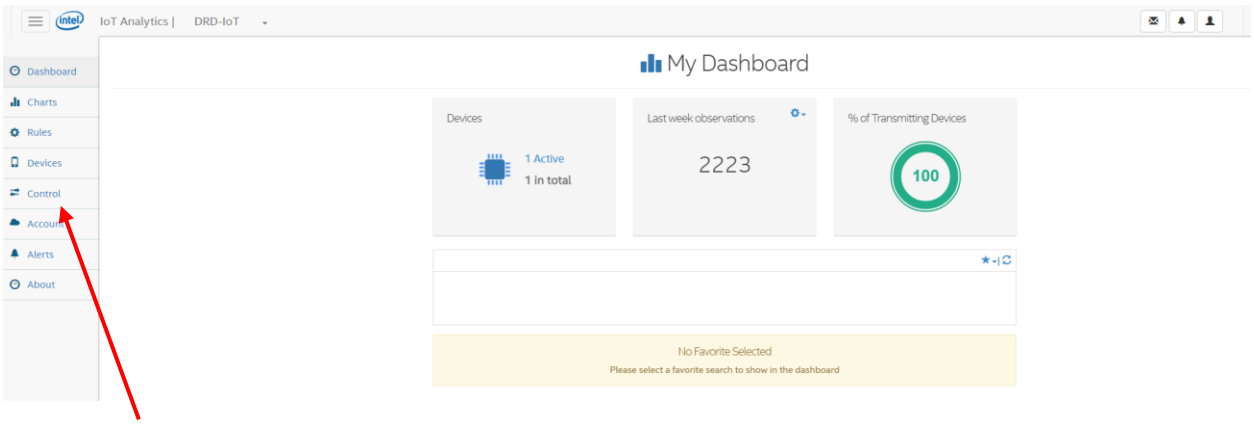


Click on the alert to get more information including the value that was measured and the time it occurred. An alert is essentially an open ticket. Future alerts for the same rule will not be logged until the first is '*Closed*'. To close the ticket click the '*Close'* button.

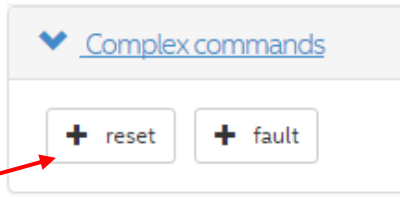


You can also see a list of past alerts by selecting '*Alerts*' from the left panel in your Stream My IoT dashboard.

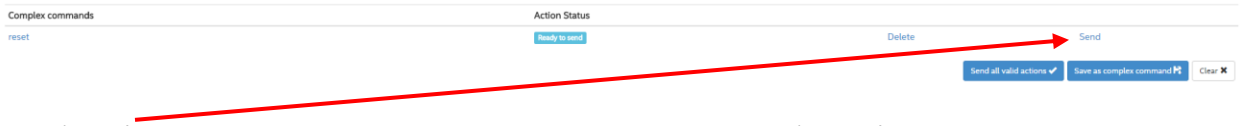
4. Remotely reset the fault condition once the alert or ticket has been closed. As mentioned previously you can use the hardware button to reset the fault condition but you can also do it from your Stream My IoT dashboard.



Select '*Control*' from the left panel in your Stream My IoT dashboard.



Click on the '*reset*' button in the '*Complex Commands*' section to add the action to the list at the bottom of the page.



Click '*Send*' to send the reset command to your Edison device. The '*Action*' status should change to '*Sent*' and the LCD should default back to normal displaying the lux value with a green backlight after a few seconds.

5. You can also remotely trigger the fault condition on your Edison device by selecting the '*fault*' button in the '*Complex Commands*' section and sending this to the device. This could be useful if a member of the public has called in to report a failure which hasn't been automatically detected so the engineer is able to quickly identify which light is affected.

You have successfully completed this section of the workshop. You can now try to extend the existing code for more sensors, following the same procedure to send the sensor data to the cloud and using the rule engine to generate more complex alerts.