Intel® Edison Transportation & Safety Sample

Description

This is a C++ project showing how the Intel® Edison board and the Intel® IoT Analytics Developer Kit cloud website can be used to manage fleet vehicles for a delivery company. The sensors used are:

- A GPS sensor to gather position data of a vehicle and send it to the cloud.
- An infrared reflective sensor to simulate if the back door on the truck is open, indicating the driver is making a delivery. This data is also sent to the cloud.
- An IR distance interrupter placed on the rear bumper of the vehicle, with the LCD visible to the driver to indicate when the vehicle is getting close to a loading dock when backing up.

The source code for this sample is located here:
https://github.com/intel-iot-devkit/iot-devkit-samples/tree/master/kits/transportation-safety

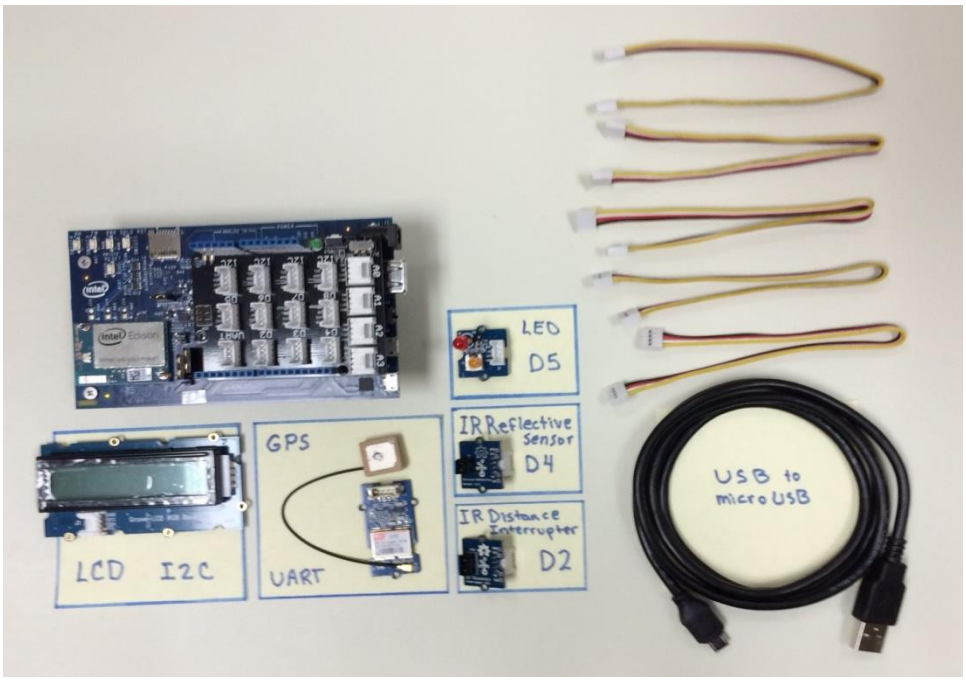
Prerequisites

Complete the Getting Started section on the Intel Developer Zone:
https://software.intel.com/en-us/iot/library/edison-getting-started

Hardware Setup

You will need:

- Intel® Edison kit with Arduino* breakout board
- Grove* – Base Shield
- Grove* – GPS (SEN10752P)
- Grove* – IR Distance Interrupter (SEN09281P)
- Grove* – Infrared Reflective Sensor (WLS07061P)
- Grove* – LCD RGB Backlight (811004001)
- Grove* – LED (COM04054P)
- Grove* – Universal 4 Pin Buckled 20cm Cable (5 PCs pack) (ACC90453O)
- USB to micro USB cable (HOK05173P)
- Two boxes and a CD to trigger the sensors

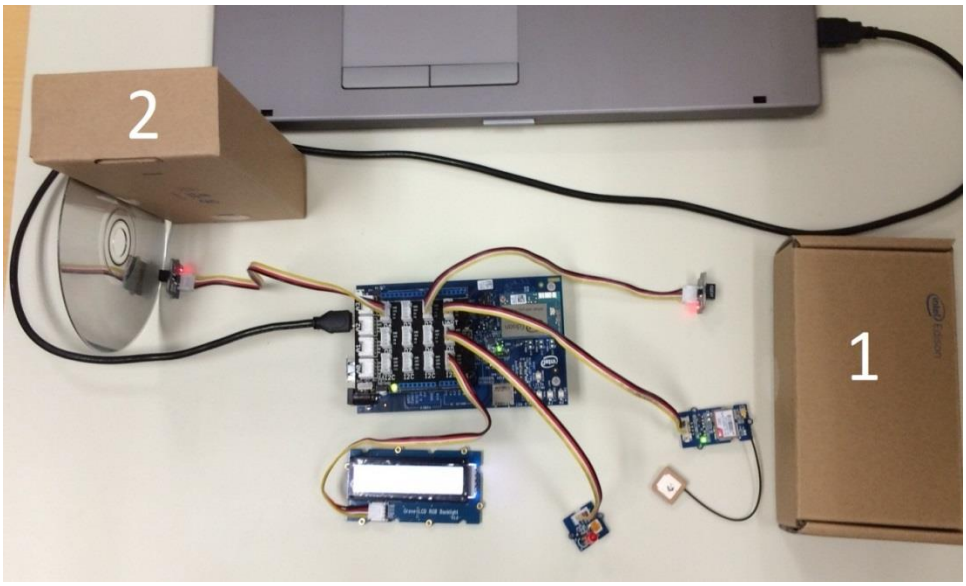


Connect each sensor to the base shield using the 4 pin cables. Be sure to connect each sensor to the appropriate socket on the base shield:

- **Grove – GPS**: UART
- **Grove – IR Distance Interrupter**: D2
- **Grove – Infrared Reflective Sensor**: D4
- **Grove – LCD RGB Backlight**: I2C
- **Grove – LED**: D5

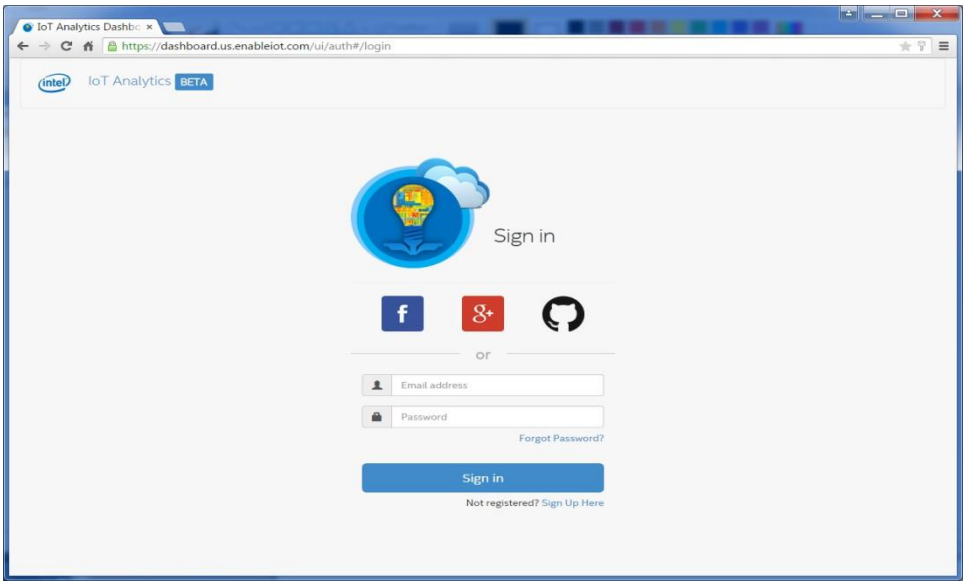
Connect the USB side of the USB cable to the computer and the micro-USB side to the middle port on the Intel Edison board.

In this example, you'll be using two boxes and a CD to trigger the sensors, as shown in the image below. Box 1 is used to trigger the IR distance interrupter. Box 2 with the CD is used to trigger the infrared reflective sensor.



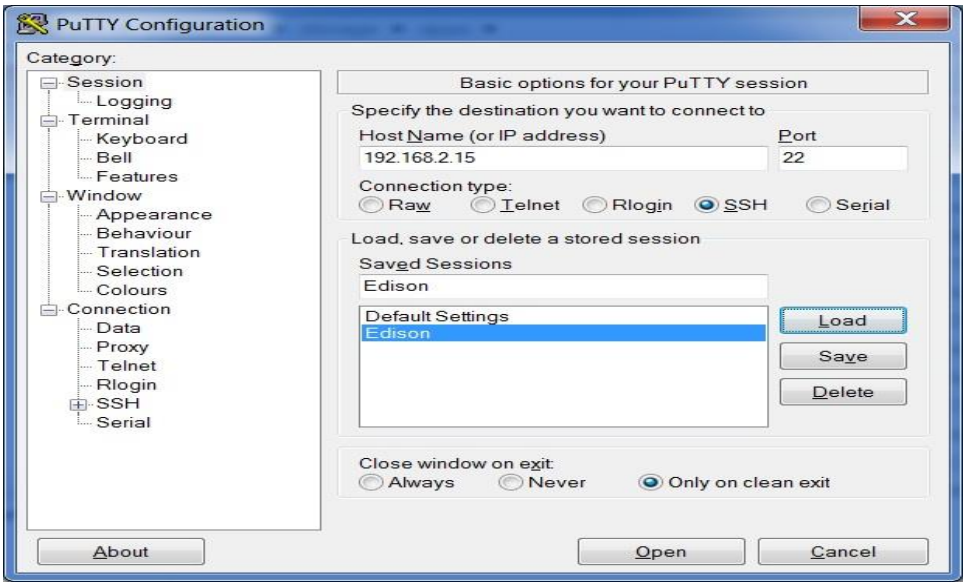
Create an Intel IoT Analytics Account

Go to the Intel IoT Analytics website and create an account:
https://dashboard.us.enableiot.com/ui/auth#/login



Connect the Intel Edison board to Wi-Fi

Create an SSH connection to your Intel Edison board using PuTTY. The default IP Address is 192.168.2.15.



Log in to your Intel Edison board as root. The default password is empty.

Configure the Wi-Fi* connection with the command:

```
root@edison:~# configure_edison --wifi
```

The board searches for available Wi-Fi networks and displays them in a list. Select your wireless network from the list and provide your login information.

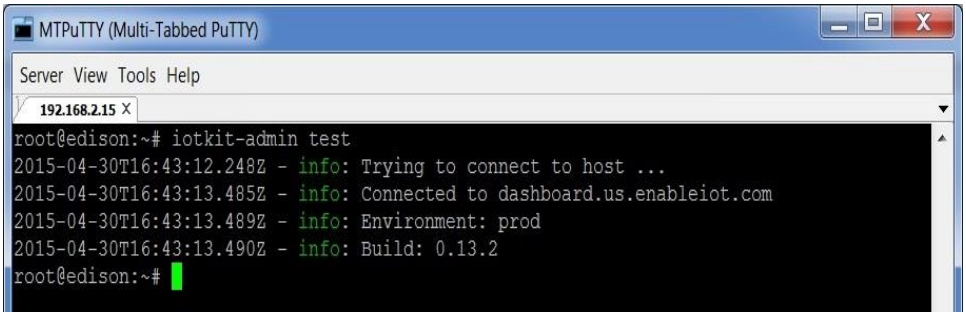
Iotkit-admin and iotkit-agent

The iotkit-admin and iotkit-agent commands are used to send data to the cloud. They come pre-installed on the Intel Edison board and are available by typing either command. Below are the steps used to get this project running. For more detailed information on these commands, visit the wiki: https://github.com/enableiot/iotkit-agent/wiki

Test your internet connection with the command:

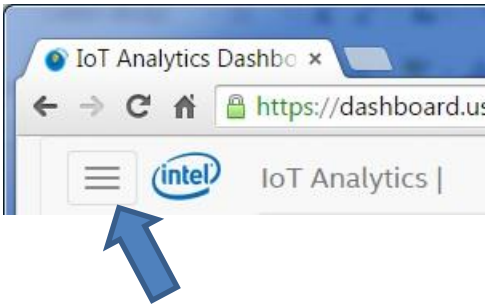
```
root@edison:~# iotkit-admin test
```

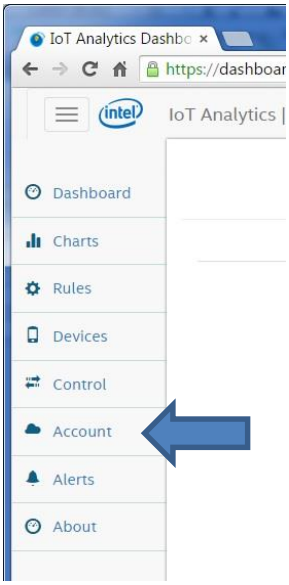
You should see the following information when your test is successful:



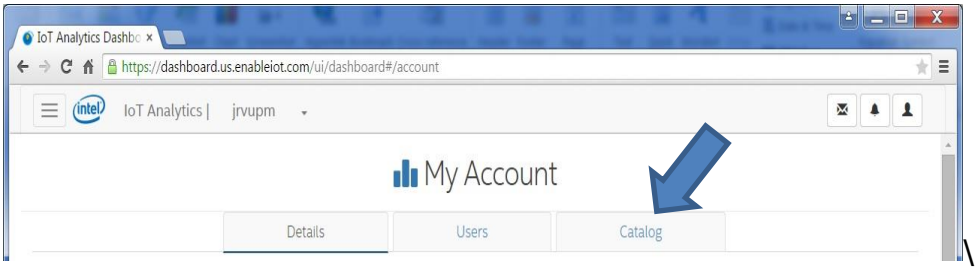
Configure your Intel IoT Analytics Account

Click the menu icon in the upper left corner of the screen and select **Account**.

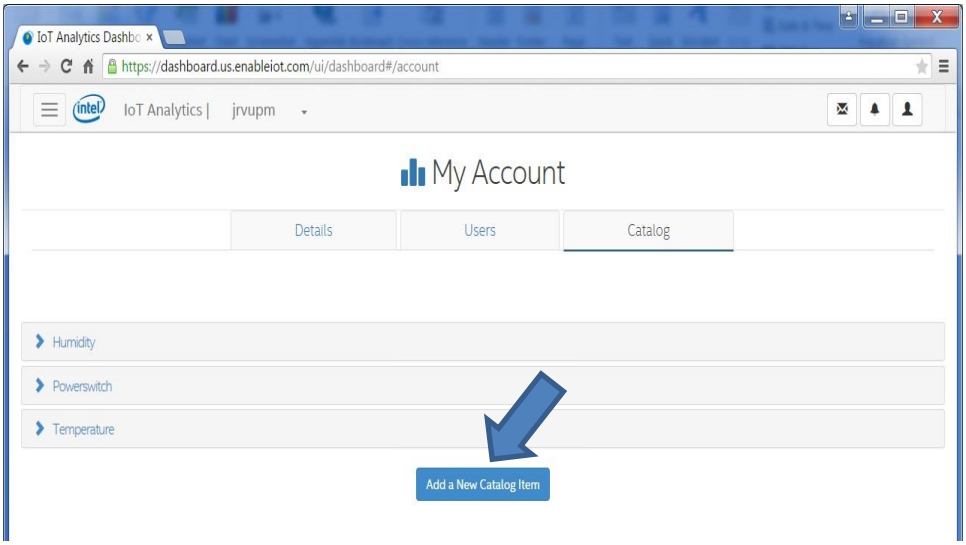




Click **Catalog**.



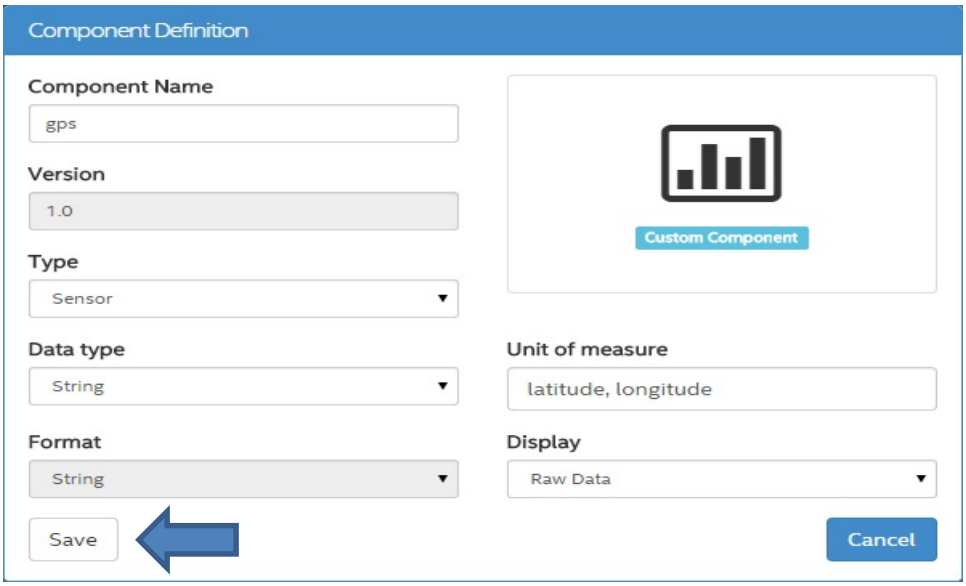
Click **Add a New Catalog Item**.



On the Component Definition page, fill in the fields with the following information:

- **Component Name**: gps
- **Type**: Sensor
- **Format**: String
- **Unit of Measure**: latitude, longitude
- **Display**: Raw Data

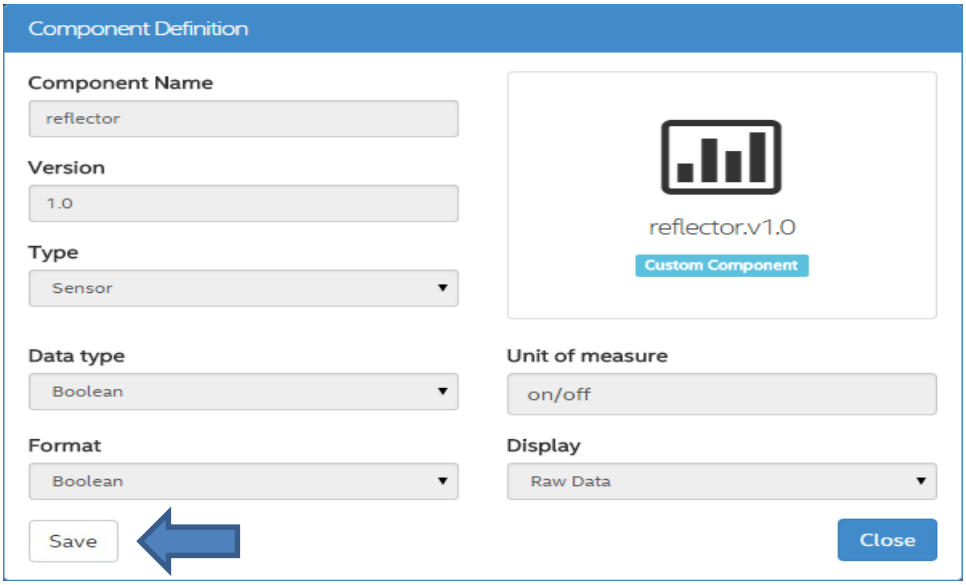
Click **Save**.



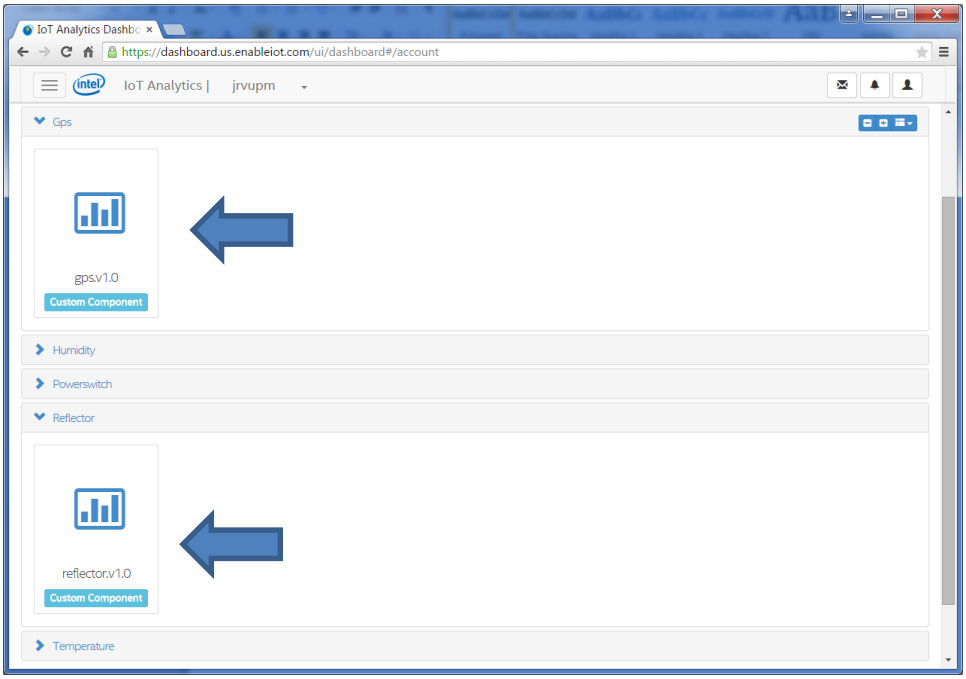
Click **Add a New Catalog Item** again. Fill in the fields with the following information:

- **Component Name**: reflector
- **Type**: Sensor
- **Format**: Boolean
- **Unit of Measure**: on/off
- **Display**: Raw Data

Click **Save**.

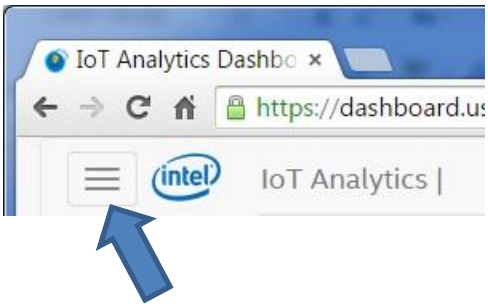


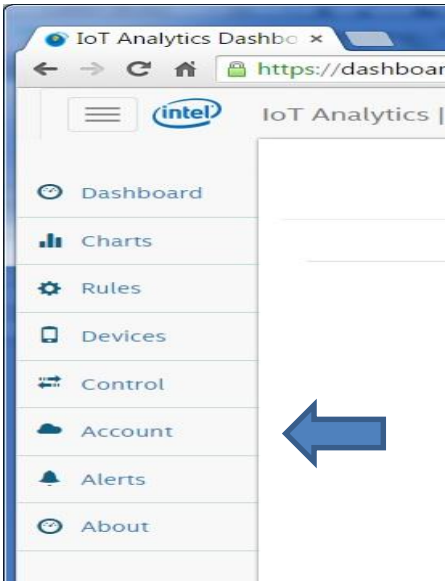
You should now have two new components listed, named gps.v1.0 and reflector.v1.0.



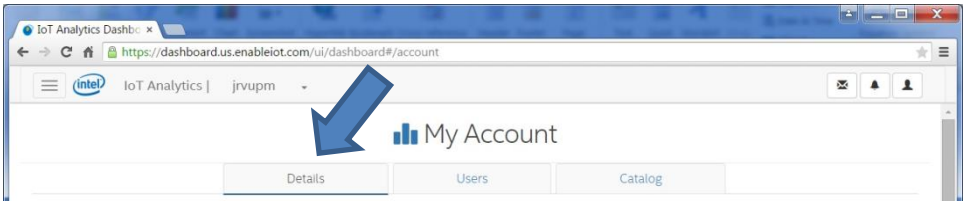
Activate your Intel Edison board with the Intel IoT Analytics Account

Click the menu icon and select **Account**.

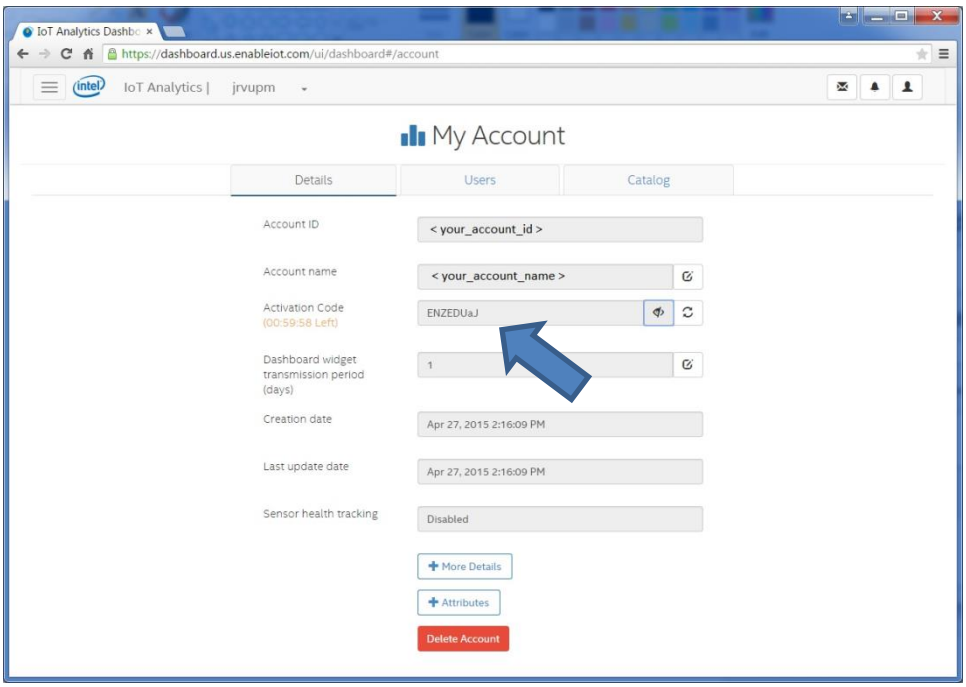




Click the **Details** tab.



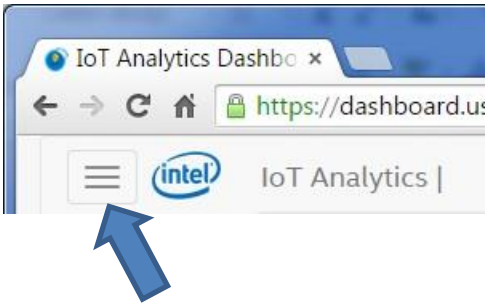
Click the eye icon  to display your activation code in the **Activation Code** field. You also may need to click the refresh icon  to get a new code if yours has expired. Make note of your activation code; you'll need it for the next step.

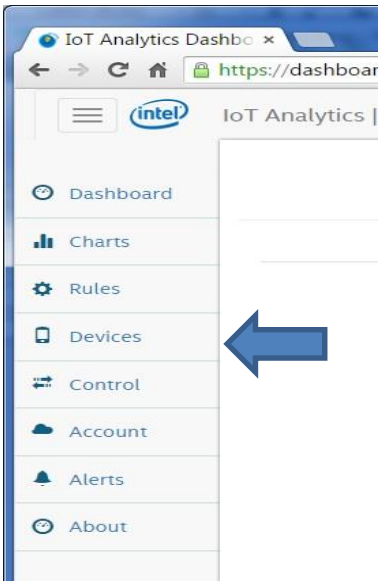


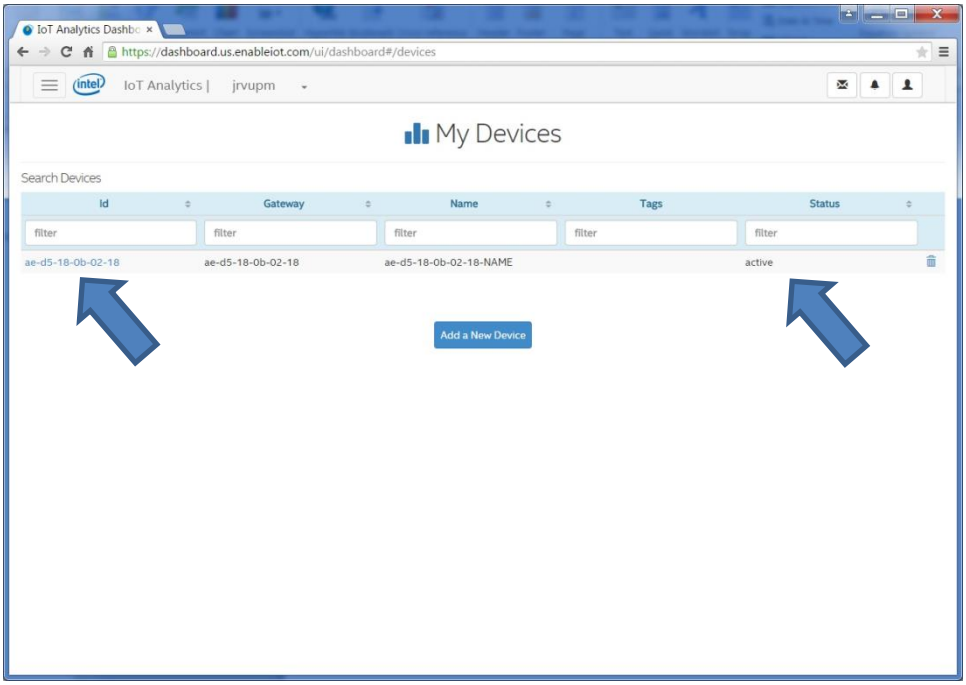
In your PuTTY terminal, type the command to activate your device. Be sure to provide your own activation code, as follows:

```
root@edison:~# iotkit-admin activate ENZEDUaJ
```

Click the menu icon and select **Devices** to confirm that your device has been activated.







Register Components and Send Test Data to your Intel Analytics Account

Register the GPS component with the command:

```
root@edison:~# iotkit-admin register gpsv1 gps.v1.0
```

You should see the confirmation message:

```
root@edison:~# iotkit-admin register gpsv1 gps.v1.0
2015-04-30T18:09:10.499Z - info: Starting registration ...
2015-04-30T18:09:10.533Z - info: Device has already been activated. Updating ...
2015-04-30T18:09:10.541Z - info: Updating metadata...
2015-04-30T18:09:10.580Z - info: Metadata updated.
Attributes sent
2015-04-30T18:09:12.554Z - info: Component registered name=gpsv1, type=gps.v1.0, cid=e6
d73225-247e-4be4-a581-53d69a7101be
```

Register the reflective sensor with the command:

```
root@edison:~# iotkit-admin register reflectorv1 reflector.v1.0
```

You should see the confirmation message:

```
root@edison:~# iotkit-admin register reflectorv1 reflector.v1.0
2015-05-01T04:11:01.001Z - info: Starting registration ...
2015-05-01T04:11:01.035Z - info: Device has already been activated. Updating ...
2015-05-01T04:11:01.043Z - info: Updating metadata...
2015-05-01T04:11:01.082Z - info: Metadata updated.
Attributes sent
2015-05-01T04:11:03.489Z - info: Component registered name=reflectorv1, type=reflector.
v1.0, cid=95bd3855-6a40-4b3d-9792-997bff14e111
```

Send some GPS test data to the cloud with the command:

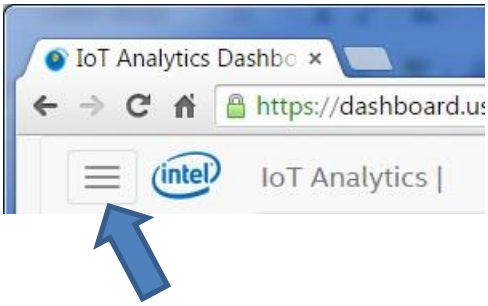
```
root@edison:~# iotkit-admin observation gpsv1 "45.123, -122.345"
```

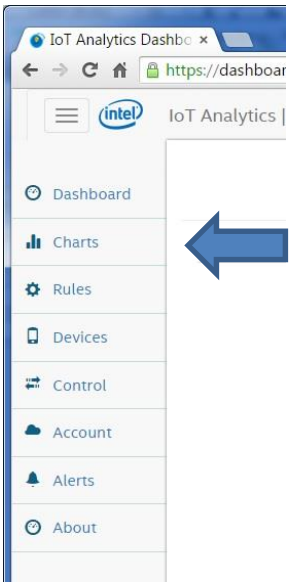
You should see the confirmation message:

```
root@edison:~# iotkit-admin observation gpsv1 "45.123, -122.345"
2015-04-30T19:19:49.305Z - info: Submitting:  n=gpsv1, v=45.123, -122.345
2015-04-30T19:19:50.081Z - info: Response received:  response=none detail, status=0
2015-04-30T19:19:50.086Z - info: Observation Sent response=none detail, status=0
```

Confirm the Test Data Was Uploaded to Your Intel IoT Analytics Account

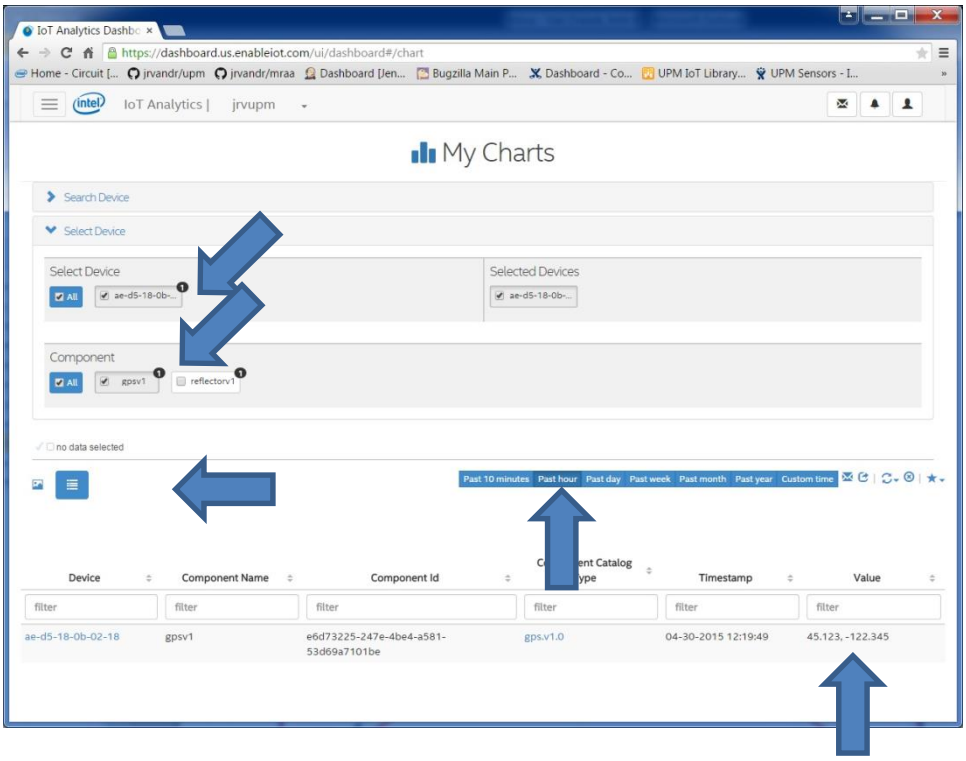
Click the menu icon and select **Charts**.





Select your device and your **gpsv1** component. Click the menu icon under the Selected Device section and choose **raw data**, then **past hour.**

Confirm your test data was uploaded in the Value column, as shown below:



Download the Transportation & Safety Project from GitHub

Go to the main iot-devkit-samples repository on GitHub:

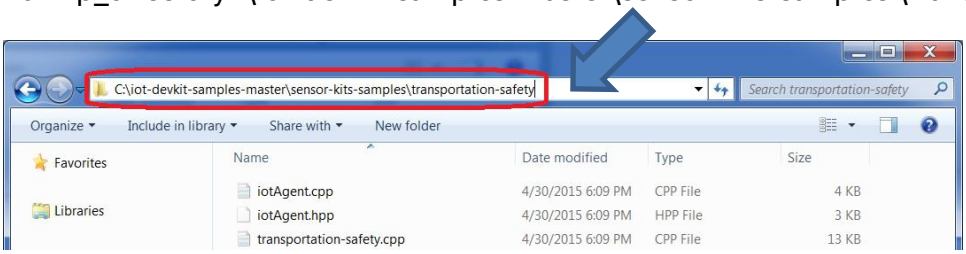
https://github.com/intel-iot-devkit/iot-devkit-samples

Download the zip file from the repository by clicking the **Download Zip** button

or use the direct link in your browser: https://github.com/intel-iot-devkit/iot-devkit-samples/archive/master.zip.

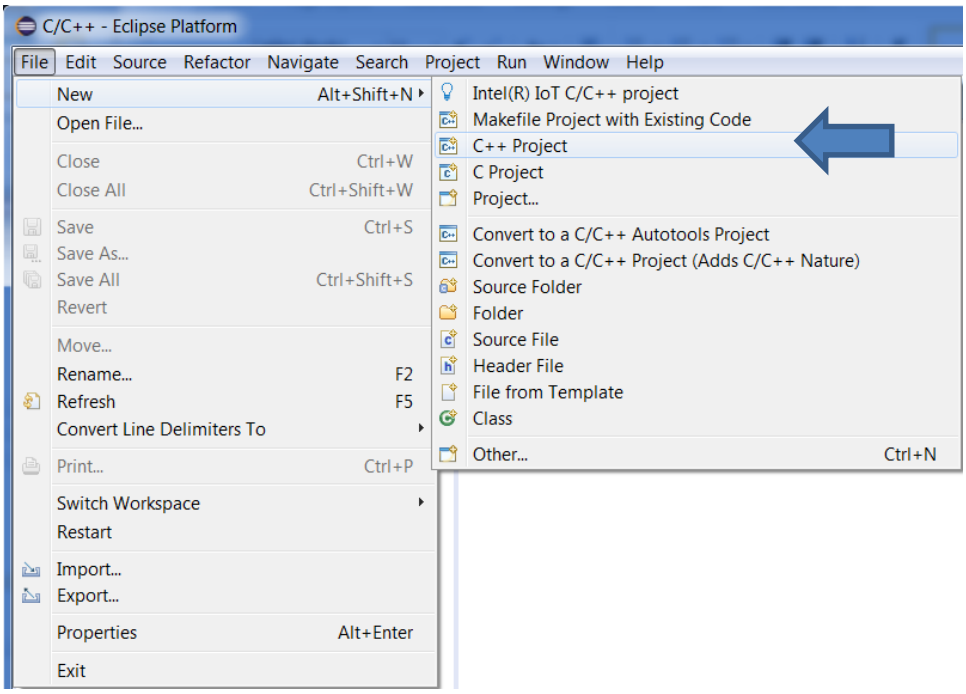
Extract the zip file and find the **transportation-safety** folder, making note of the path:

<unzip_directory>\iot-devkit-samples-master\sensor-kits-samples \transportation-safety

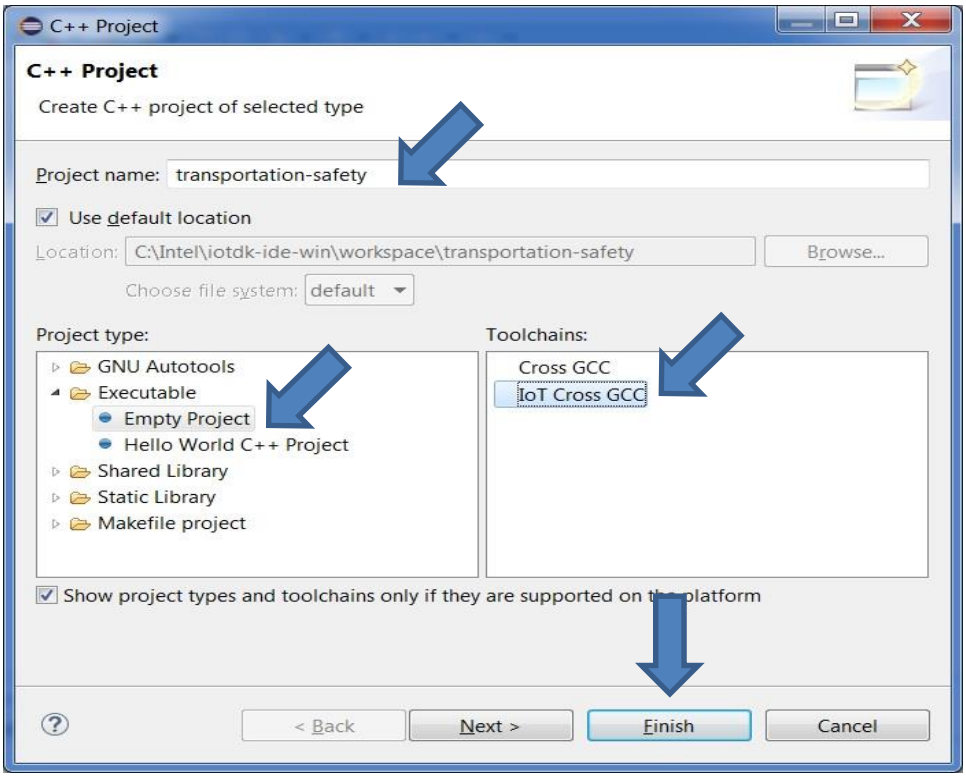


Set up a New Project in Eclipse with the Transportation & Safety Source Code

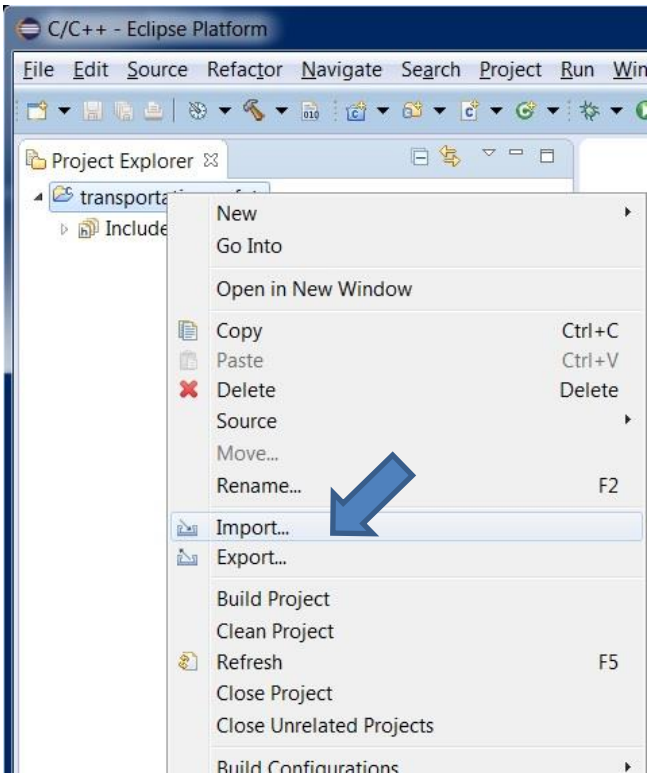
Create a new C++ Project in Eclipse by choosing **File > New > C++ Project**.



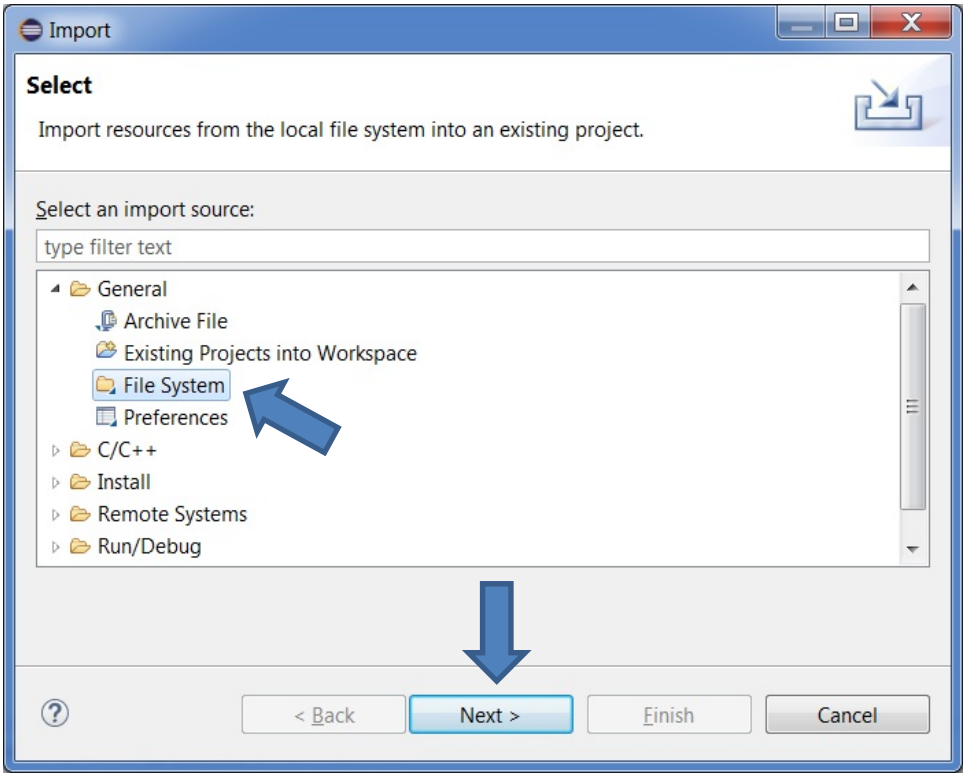
Type `transportation-safety` in the **Project name** field.
In the Project type list, select **Executeable > Empty Project**.
In the Toolchains list, select **IoT Cross GCC**.
Click **Finish**.



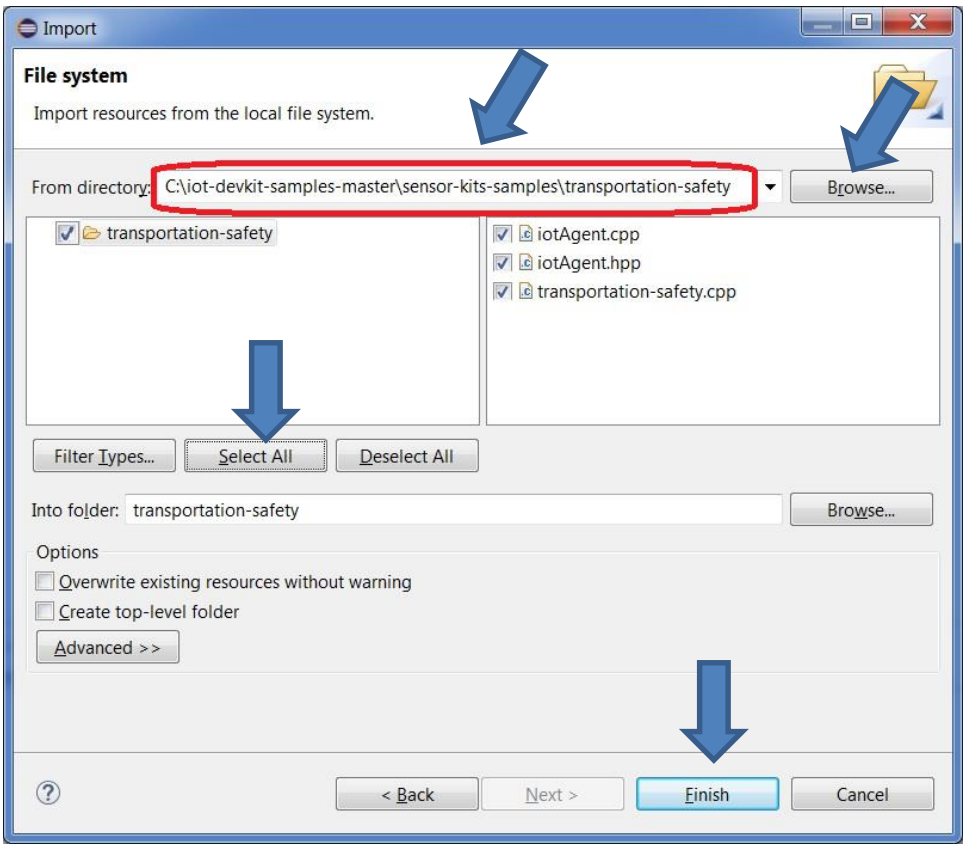
Right-click the **transportation-safety** project in the Project Explorer and select **Import.**



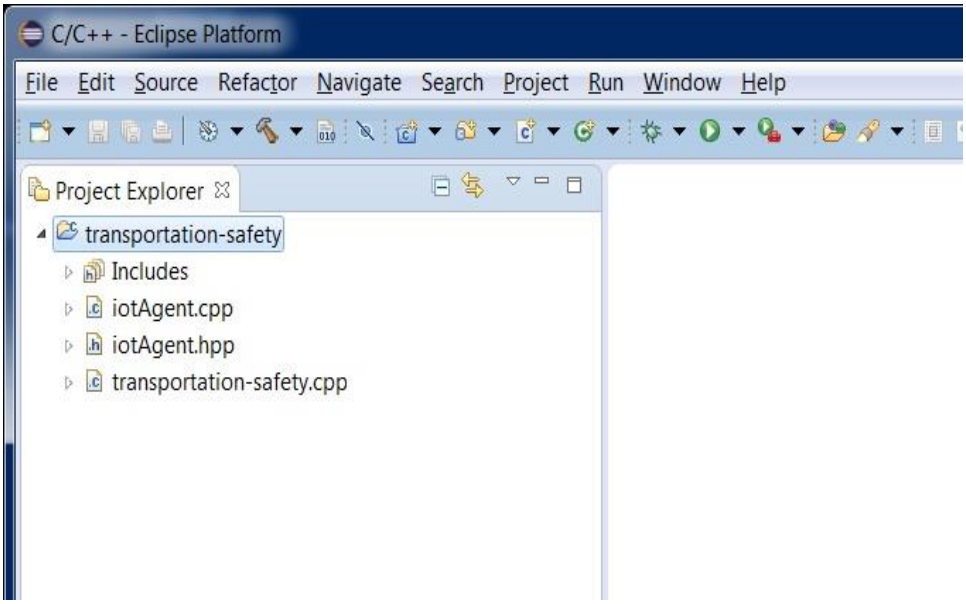
In the Import dialog box, select **General > File System** and click **Next**.



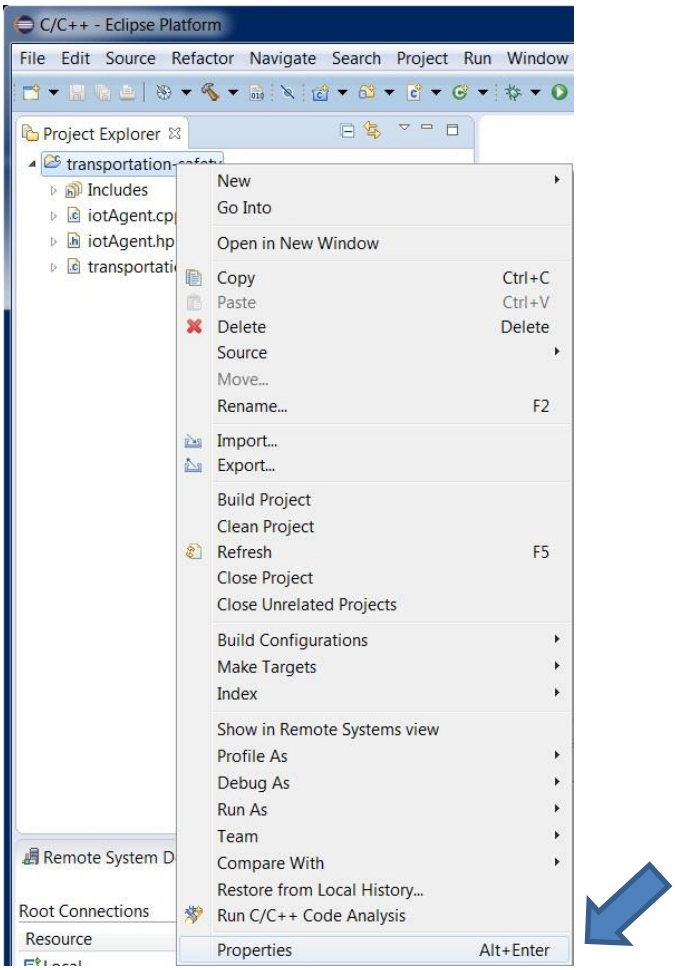
Browse to the **transportation-safety** folder you extracted from the zip file. Click **Select All**, then click **Finish**.



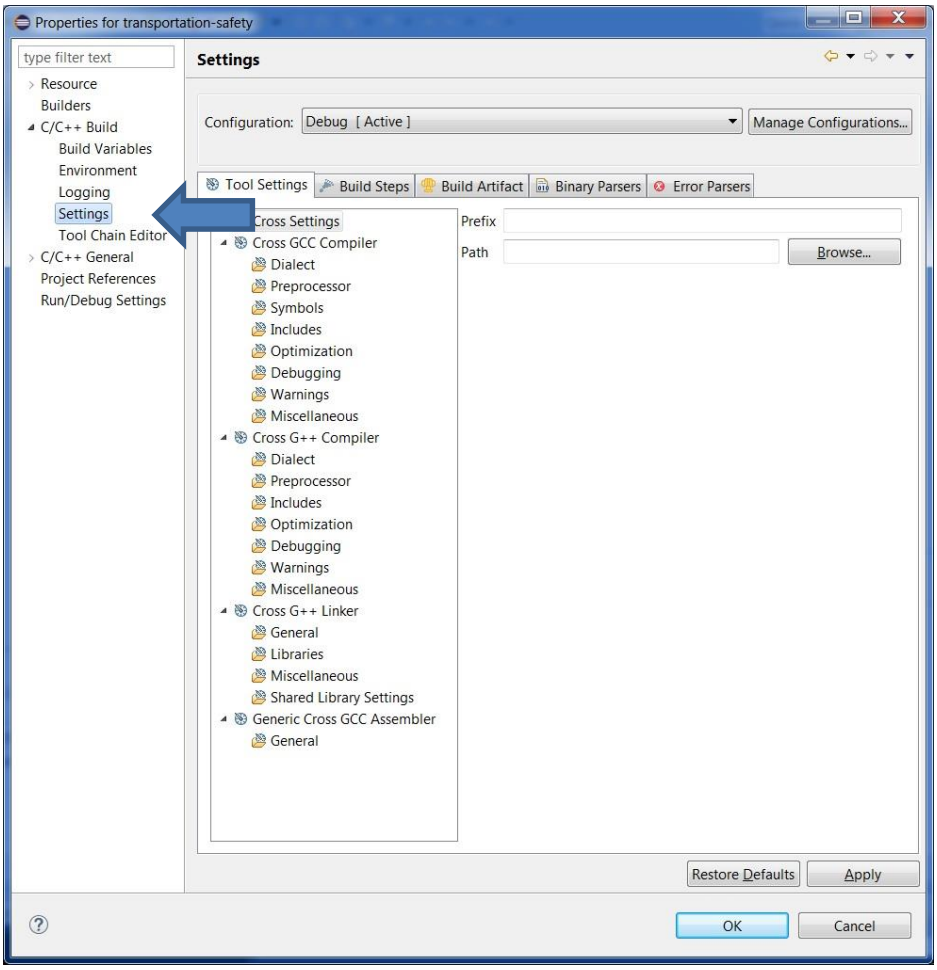
When you're done, your project should look like this:



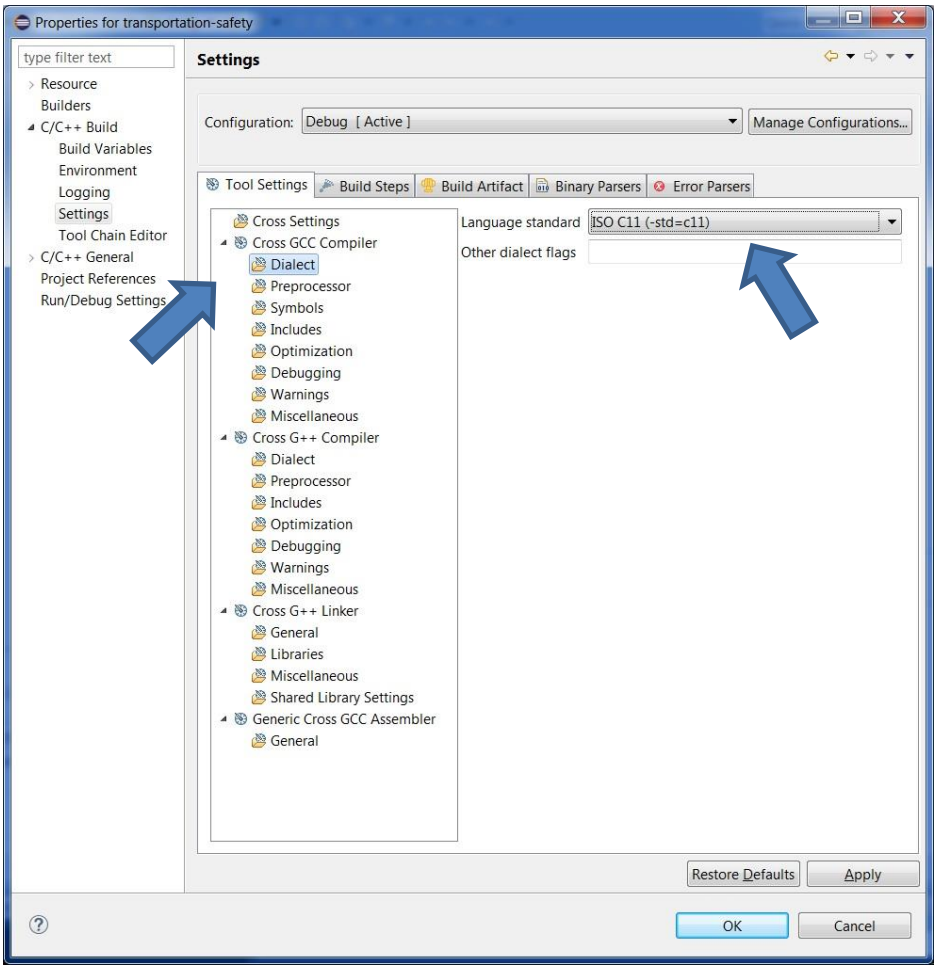
Right-click the **transportation-safety** project in the Project Explorer and select **Properties**.



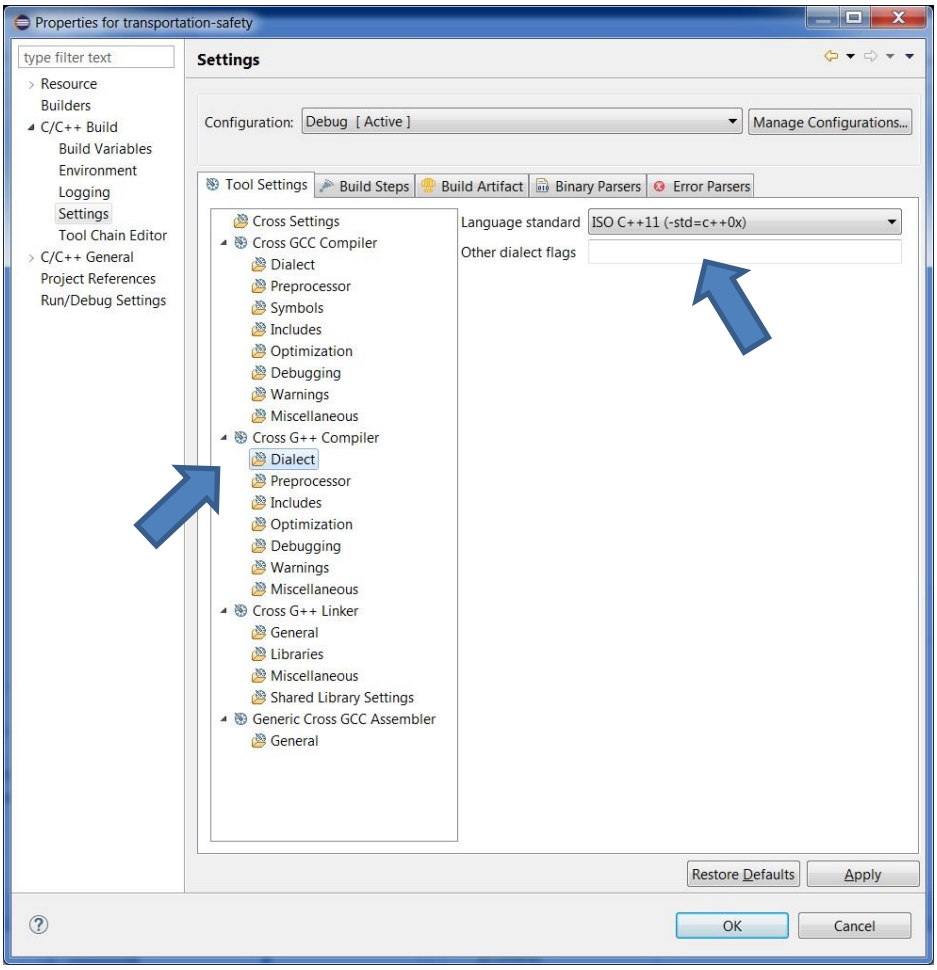
From the list on the left, select **C/C++ Build > Settings**.



From the list on the Tool Settings tab, select **Cross GCC Compiler > Dialect**. From the **Language Standard** drop-down list, select **ISO C11 (-std=c11)**.



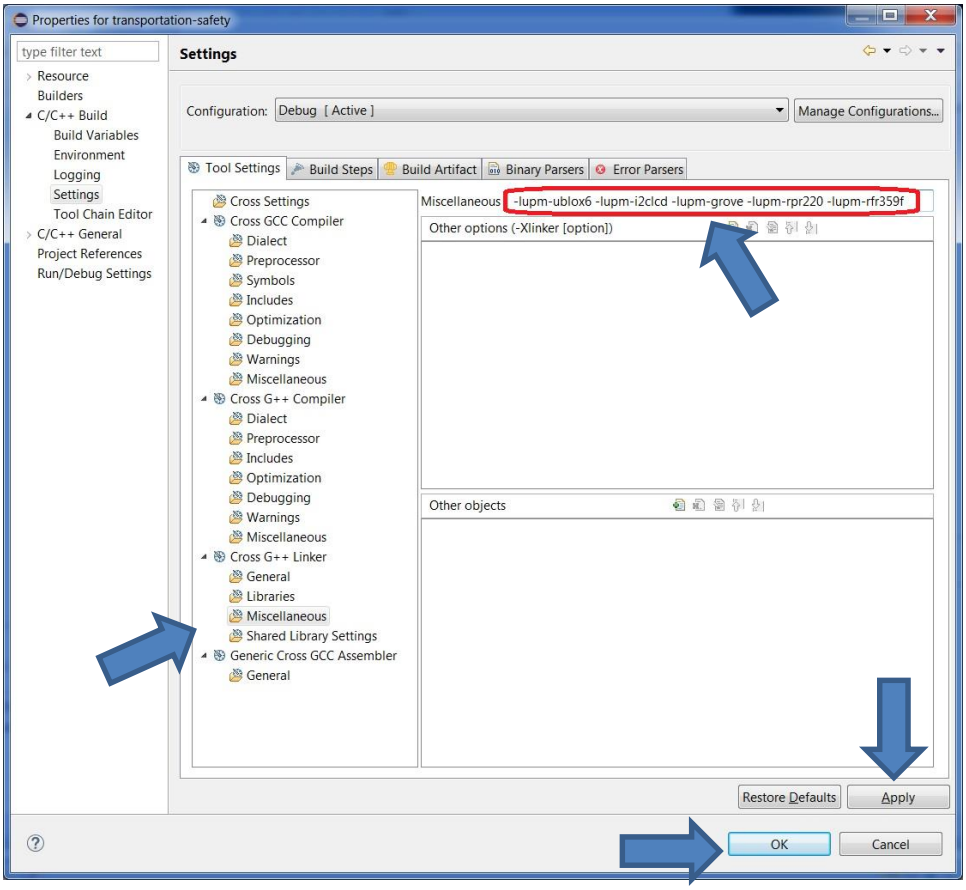
From the list on the Tool Settings tab, select **Cross G++ Compiler > Dialect**. From the **Language Standard** drop-down list, select **ISO C++11 (-std=c++0x)**.



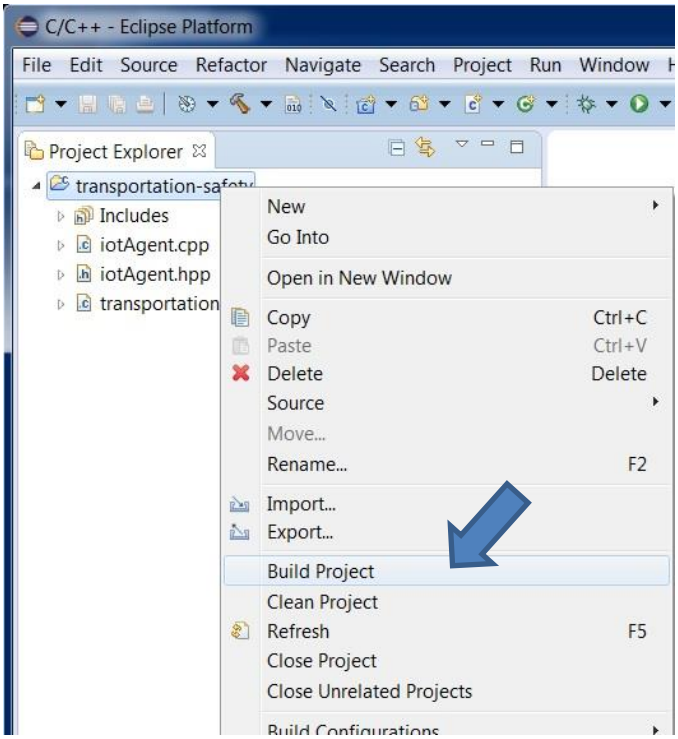
From the list on the Tool Settings tab, select **Cross G++ Linker > Miscellaneous**. In the **Miscellaneous** field, type the following:

```
-lupm-ublox6 -lupm-i2clcd -lupm-grove -lupm-rpr220 -lupm-rfr359f
```

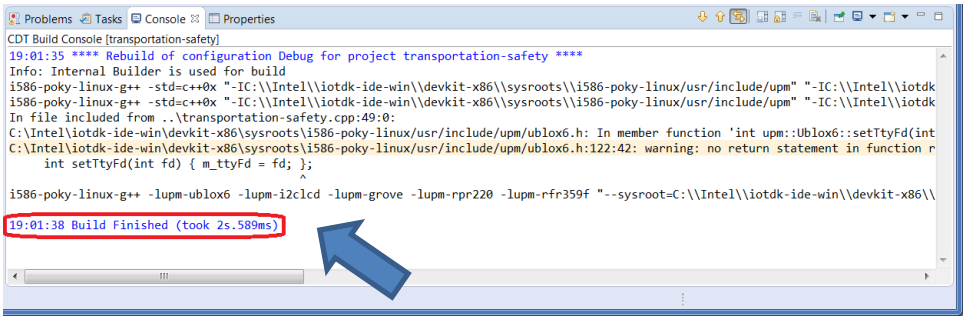
Click **Apply**, then click **OK**.



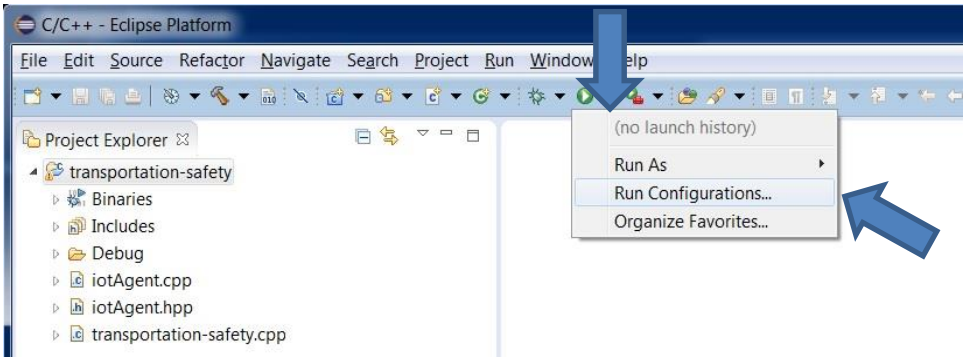
Right-click the **transportation-safety** project in the Project Explorer and select **Build Project**.



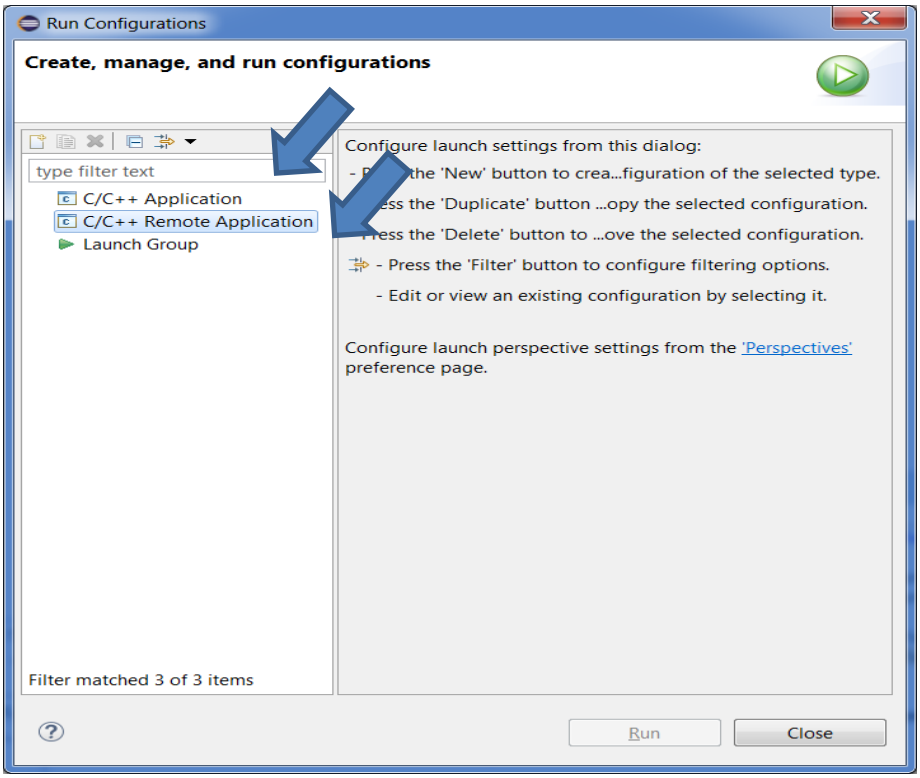
Your project builds and a confirmation message is displayed, as shown below.



From the **Run** drop-down list  , select **Run Configurations**....



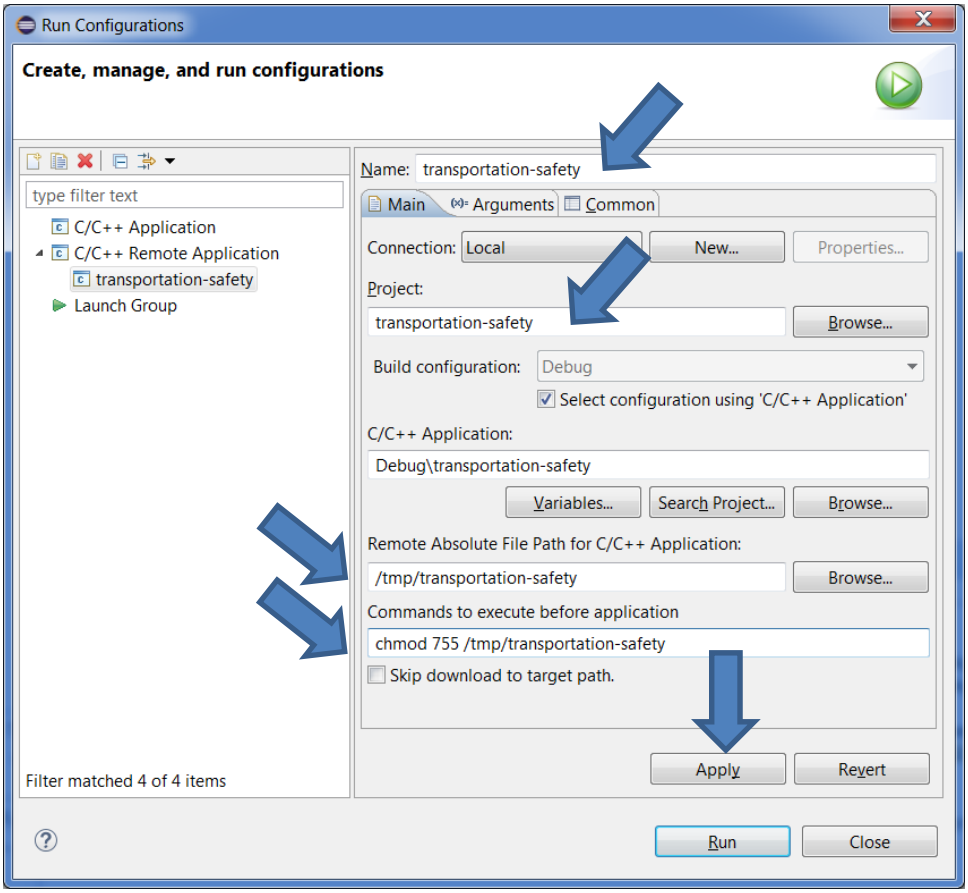
Select **C/C++ Remote Application** and click the **New launch configuration** icon .



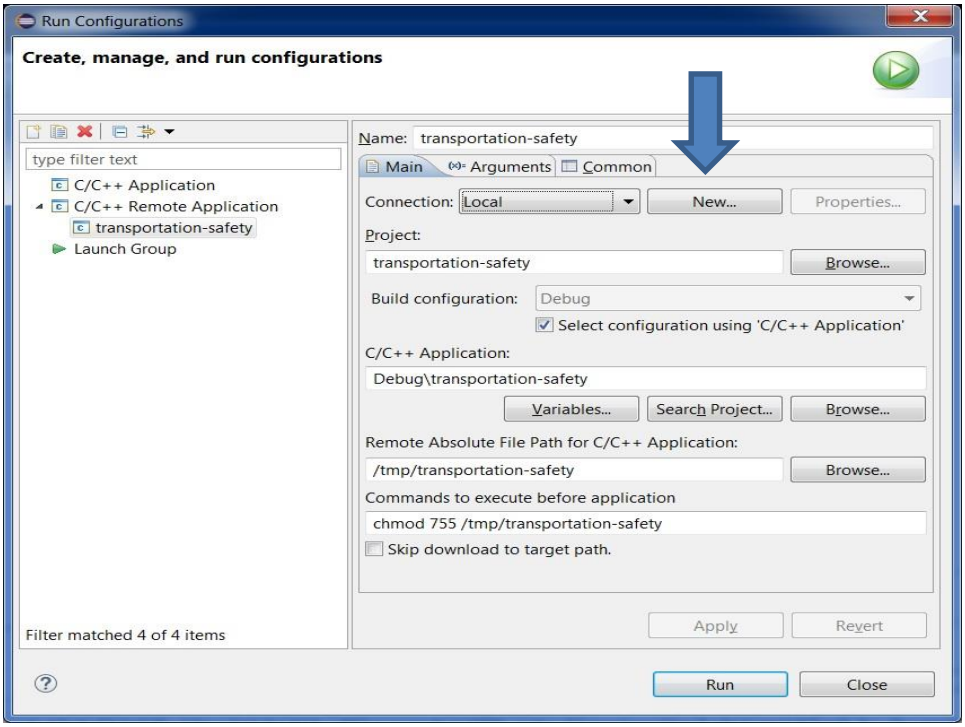
In the Run Configurations dialog box, enter the following information:

- **Name**: transportation-safety
- **Project**: transportation-safety
- **Remote Absolute File Path for C/C++ Applications**: /tmp/transportation-safety
- **Commands to execute before application**: chmod 755 /tmp/transportation-safety

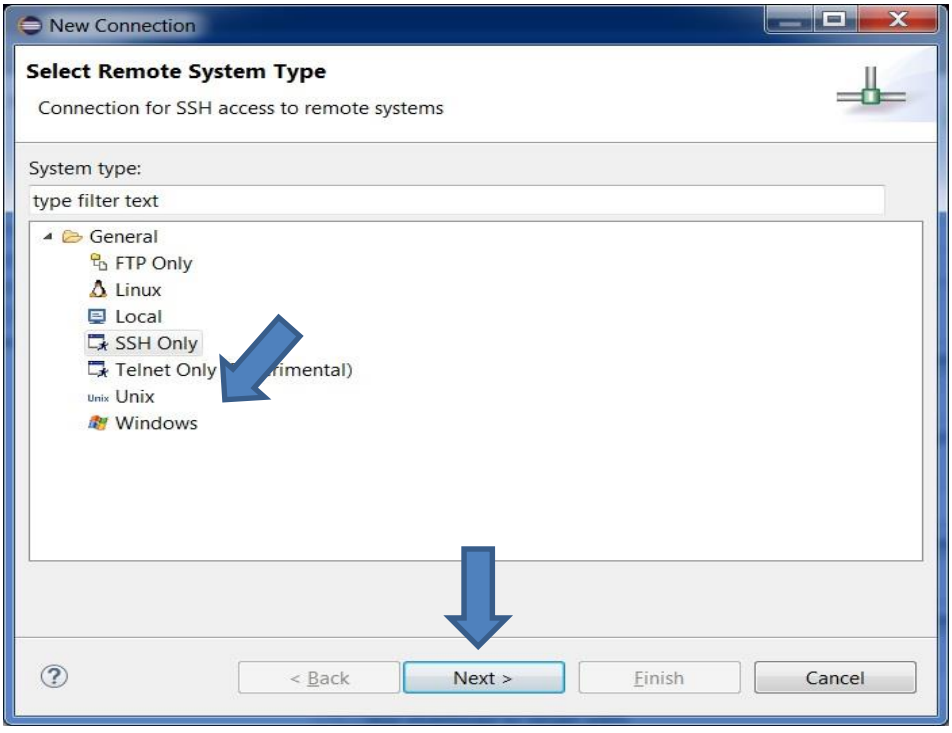
Click **Apply**.



Click **New**.



Select **SSH Only**, then click **Next**.

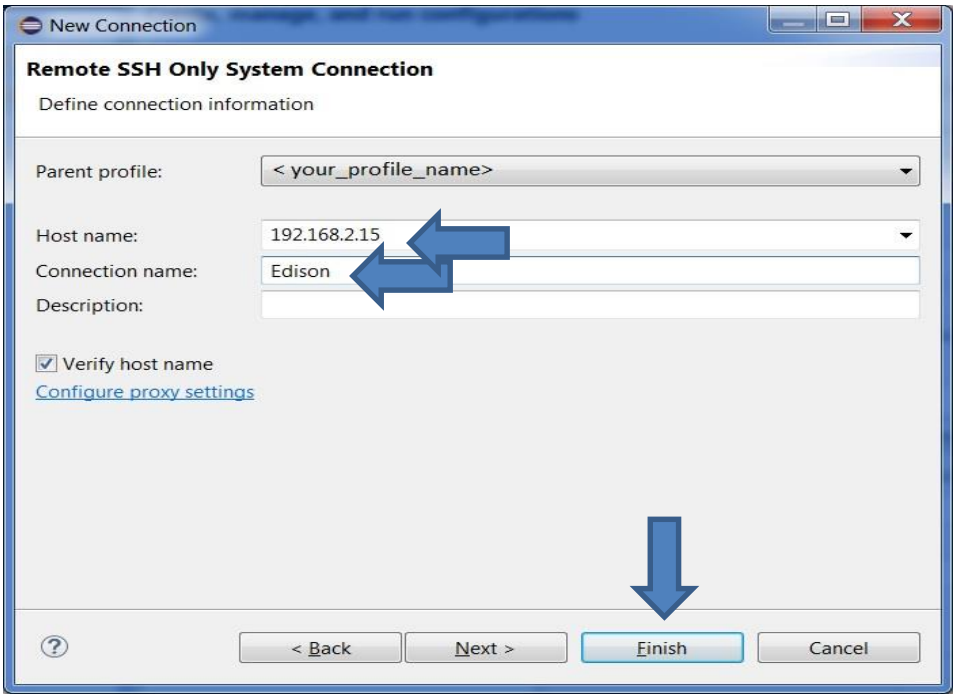


In the New Connection dialog box, enter the following information:

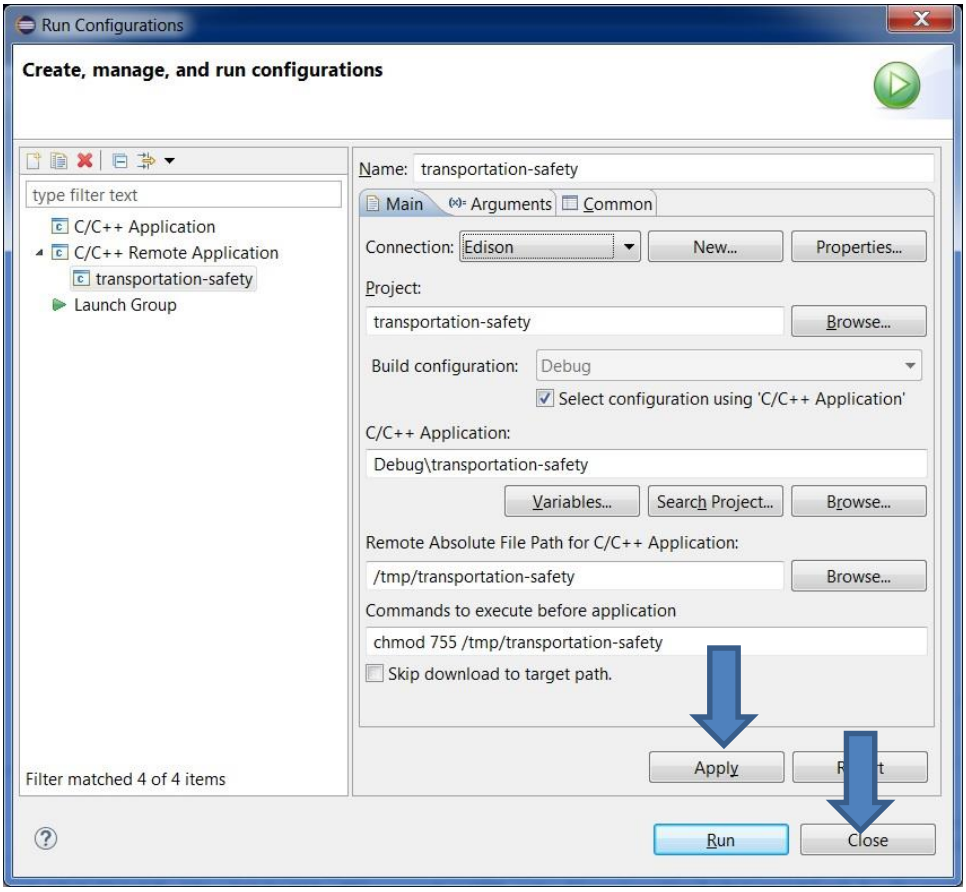
- **Parent Profile**: Keep default

- **Host Name**: 192.168.2.15
- **Connection Name**: Edison

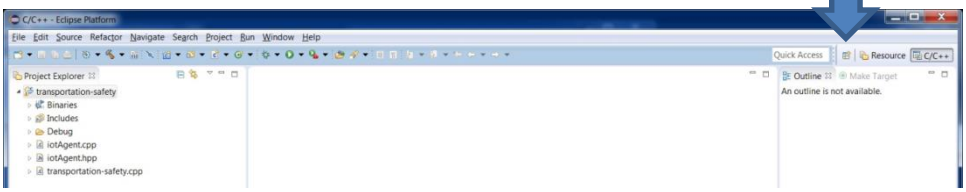
Click **Finish**.



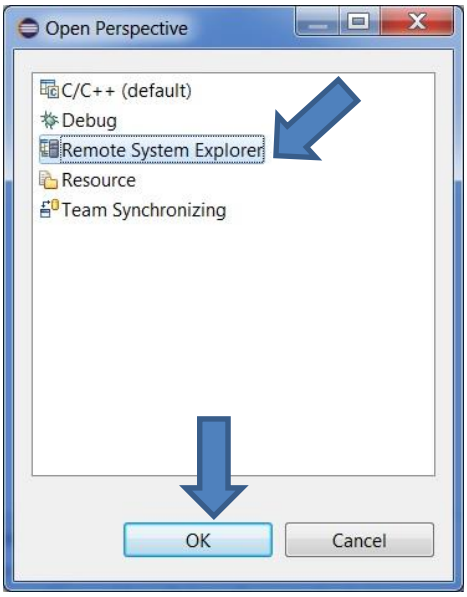
Click **Apply**, then click **Close**.



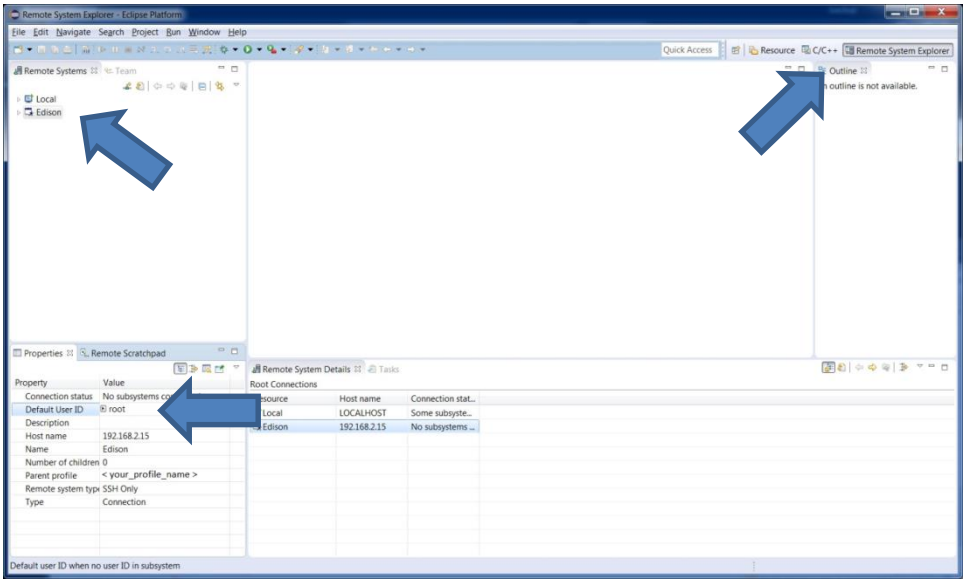
To view the Open Perspectives dialog box, click the icon in t upper-right.



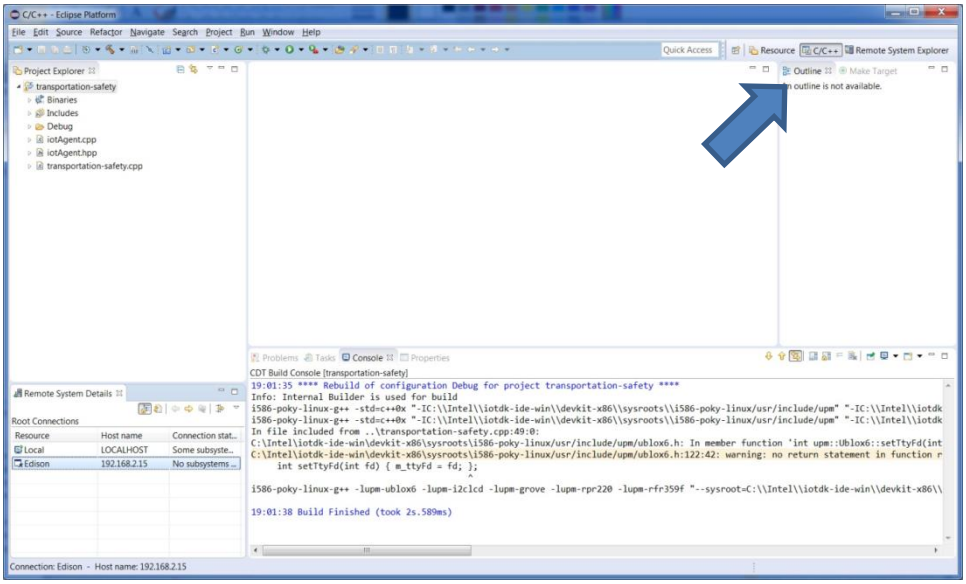
Select **Remote System Explorer** and click **OK**.



Click **Remote Systems Explorer** in the upper right corner.
From the **Remote Systems Explorer** tab, select **Edison**.
In the Properites tab, change the **Default user ID** to root.



Click the **C/C++** perspective button.

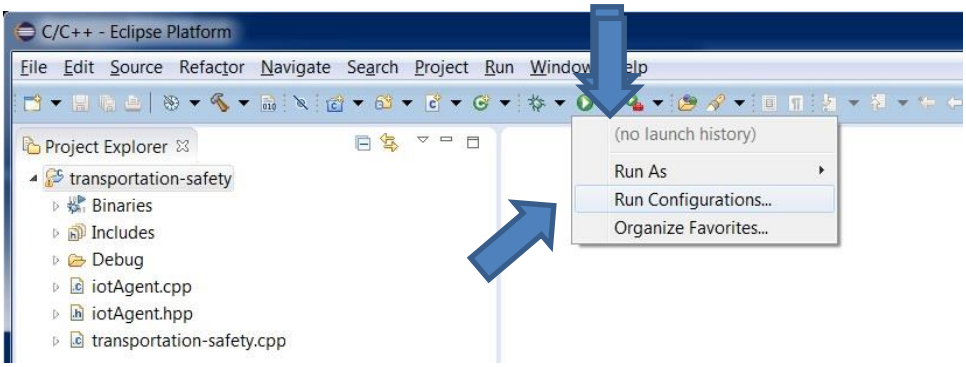


Run the Transportation & Safety Project

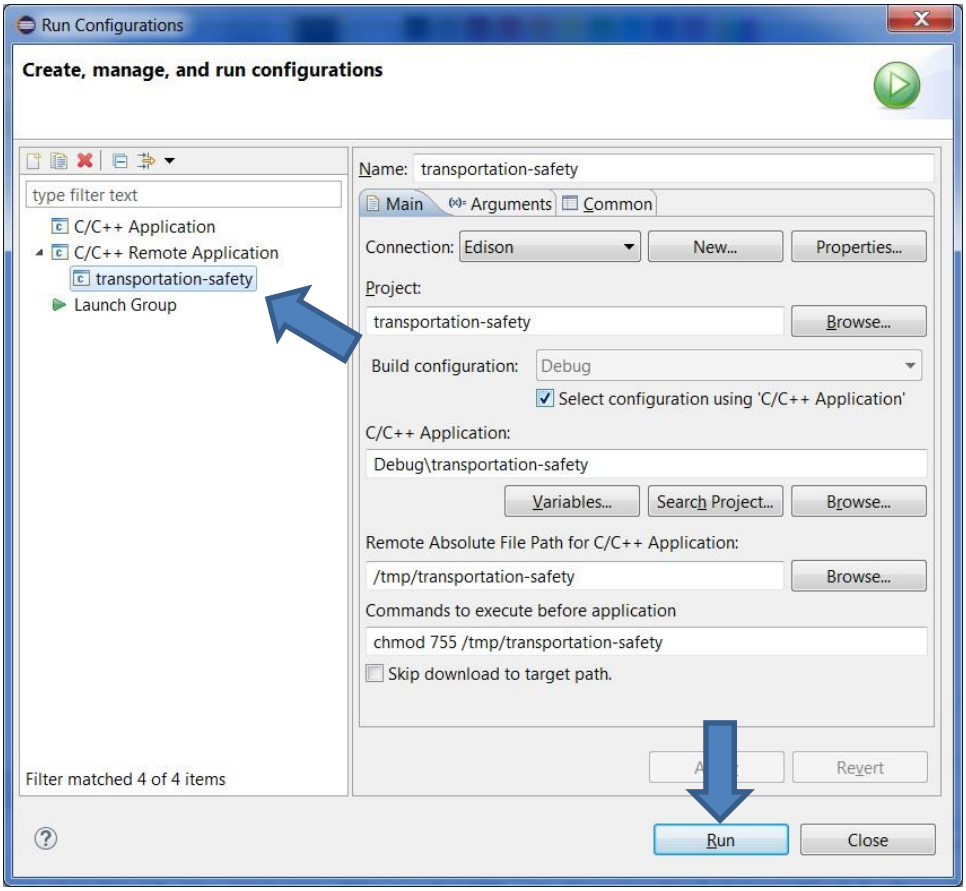
First, make sure to start the iotkit-agent. Open an SSH PuTTY session to your Intel Edison board and start the iotkit-agent with the command:

```
root@edison:~# systemctl start iotkit-agent
```

In the Eclipse project, from the **Run** drop-down list  , select **Run Configurations**....



In the Run Configurations dialog box, select **C/C++ Remote Application > transportation-safety**, then click **Run**.



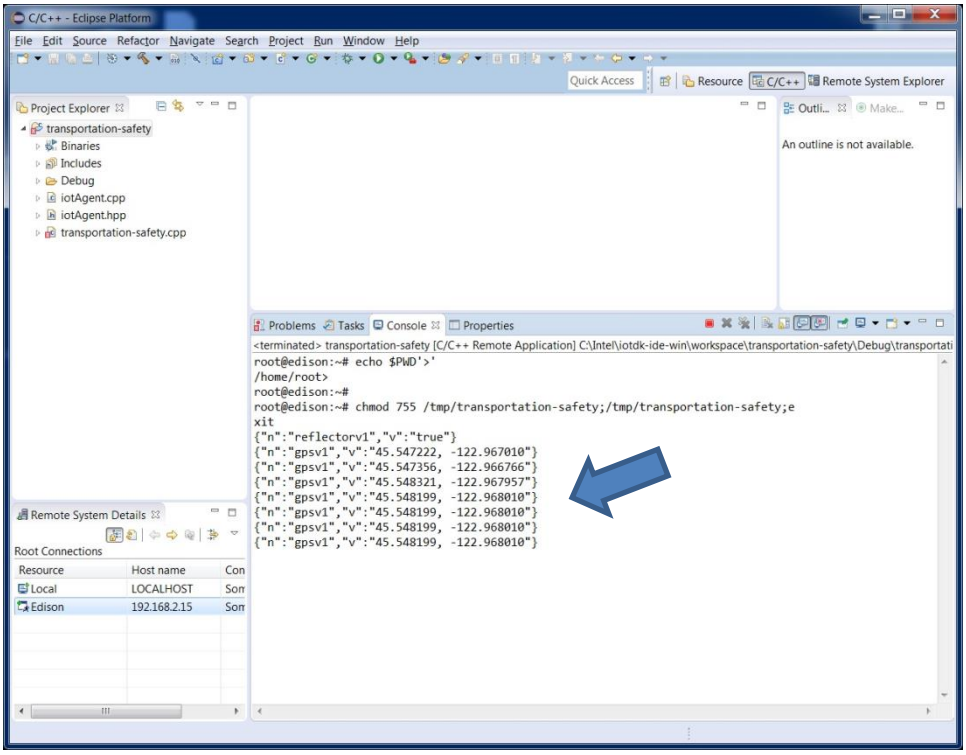
If this is your first time connecting to your Intel Edison board, the Enter Password dialog box opens. Enter the root password for your board, then click **OK**. By default, the password is empty.

GPS Position of a Delivery Vehicle

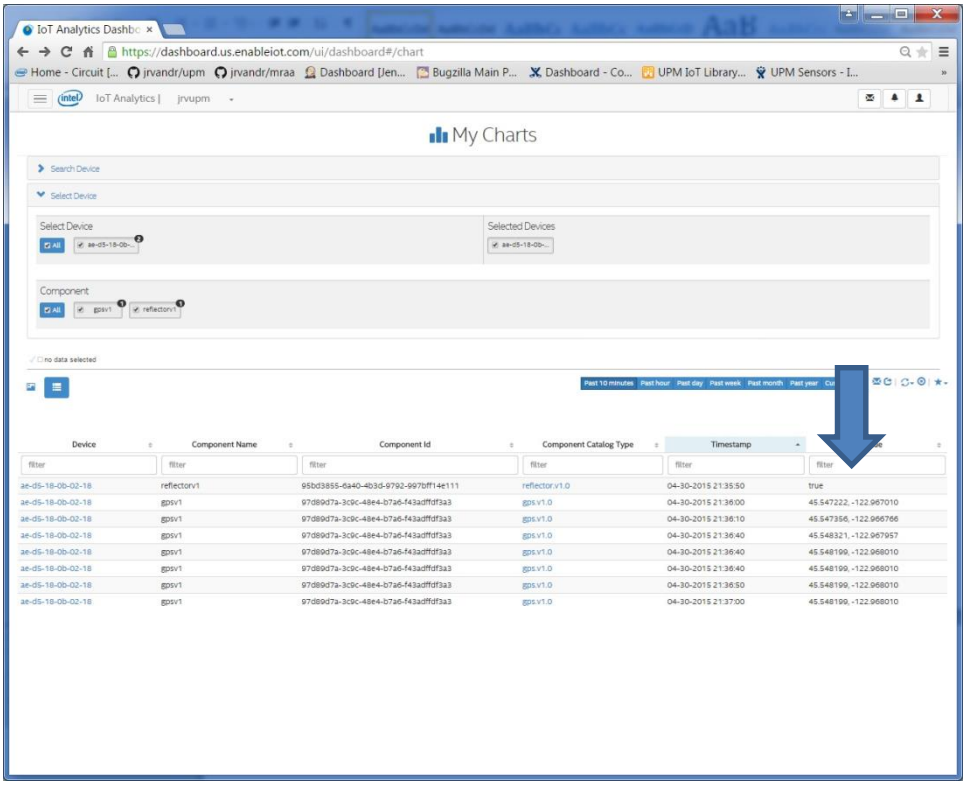
The GPS position of a delivery vehicle can be logged and uploaded for many useful purposes.

- If the driver travels a pre-determined route and the GPS data reports data outside the route, the dispatcher can automatically be notified to contact the driver to see what the problem is. For example, there is a road blocked on a normal route and the deliveries will be late. This information can be used in real-time to inform delivery locations of a delayed delivery.
- The vehicle was stolen and the thief has traveled away from the pre-determined route. The dispatcher can be notified automatically, then report the vehicle as stolen and even provide the location of the vehicle as it changes to aid in its recovery.
- The vehicle was stolen from a parked location in the evening or outside of working hours. A message can be automatically sent to security.
- The driver is lost. The dispatcher can be notified and contact the driver, preventing unnecessary delays.
- The driver made an unscheduled stop for some unknown reason. The dispatcher can be notified and respond accordingly.

View the GPS and infrared reflective sensor data being sent to the cloud in the console window:



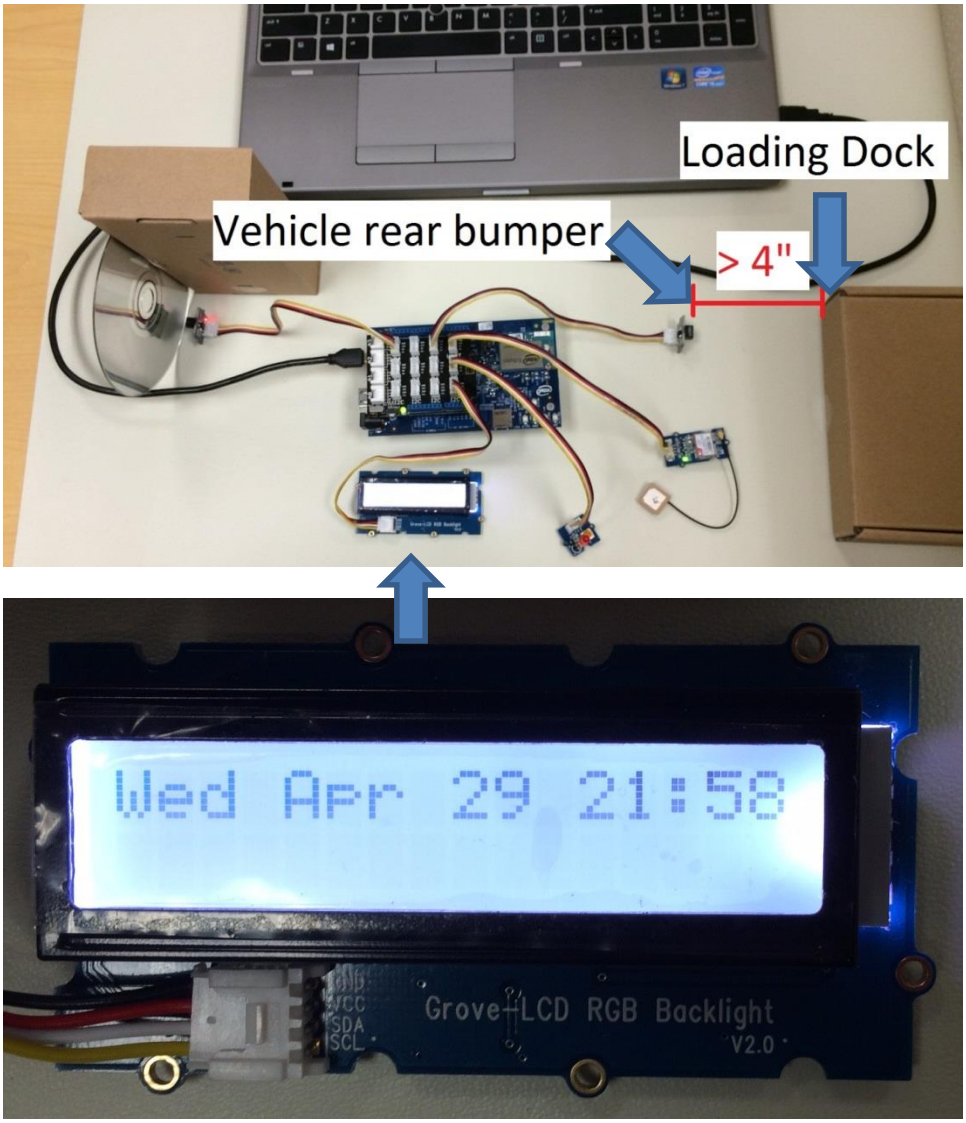
Let's check the Intel IoT Analytics account and see our new GPS data!



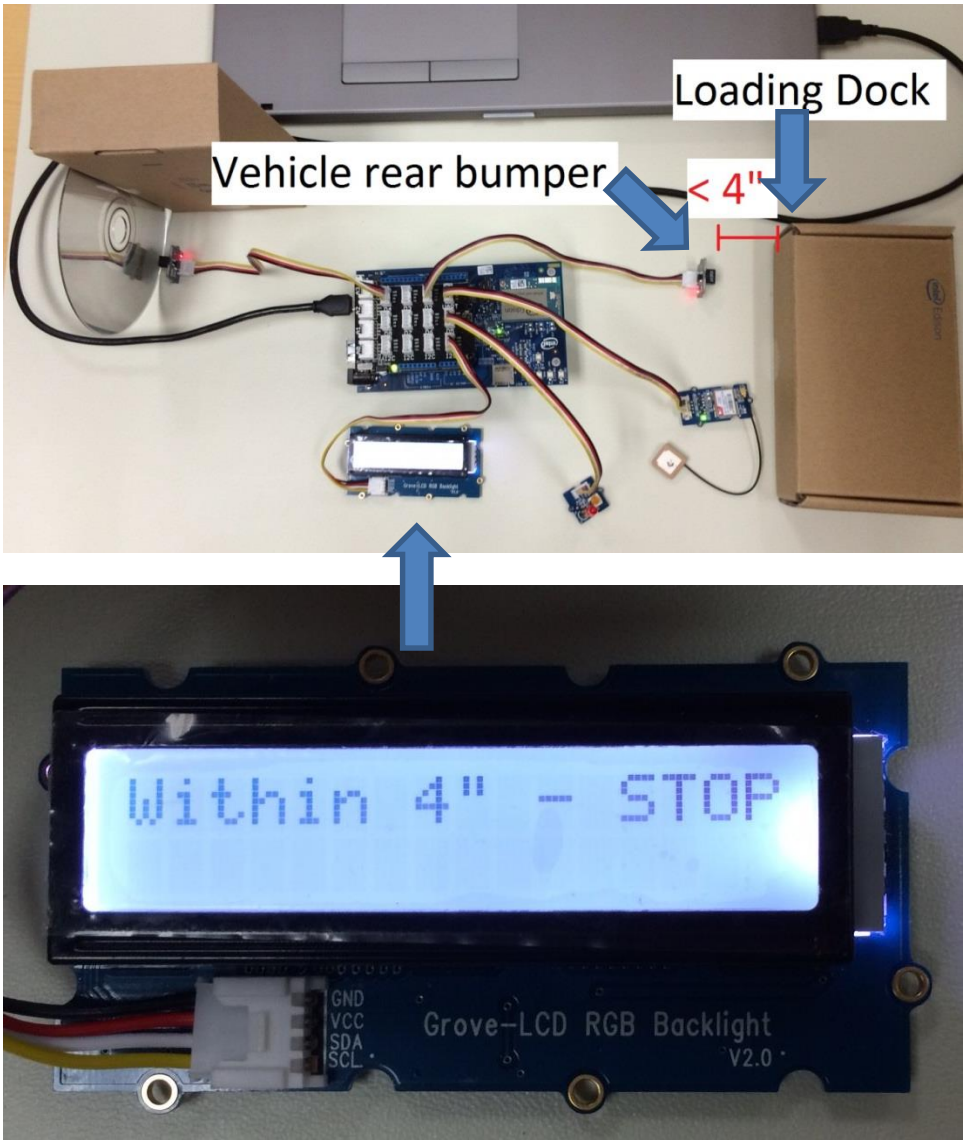
Distance Sensor Monitors Vehicle as it Backs Up to a Loading Dock

In this example, the IR distance interrupter detects objects when they are within 4" of the vehicle bumper. The LCD displays the time until an object is detected.

In the image below, the vehicle bumper is greater than 4" from the loading dock, so the LCD just displays the time.



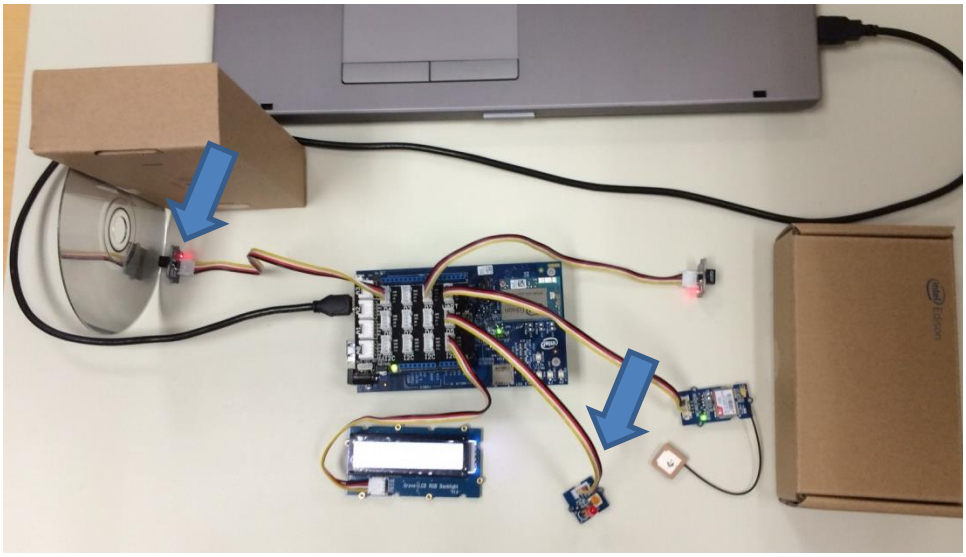
When the distance interrupter determines that the vehicle is within 4” of the loading dock, it prints a message to the LCD.



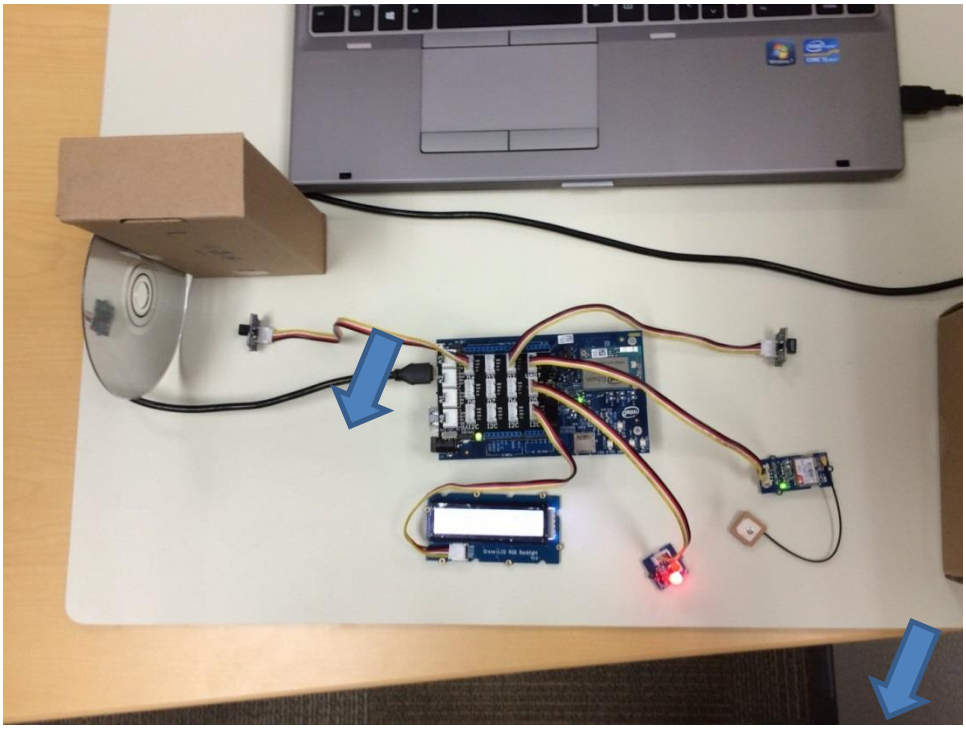
Infrared Reflective Sensor Tailgate Open Detector

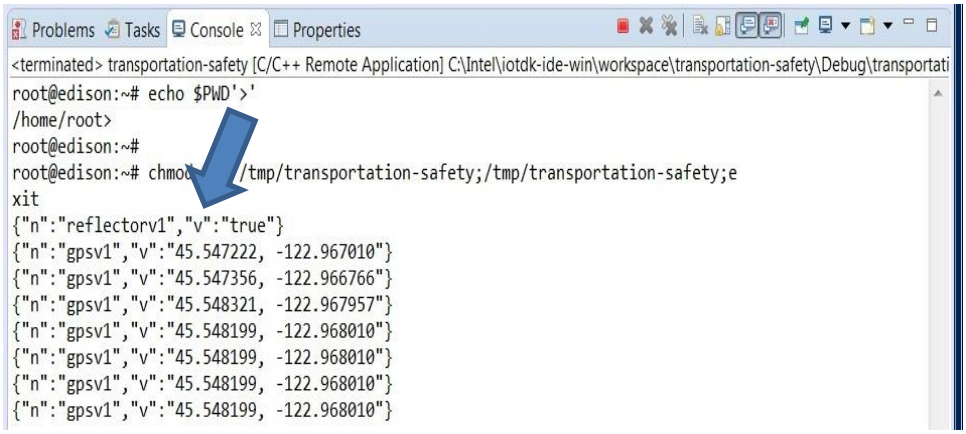
When a delivery truck stops at a loading dock, the tailgate is opened to unload the cargo. After the cargo is unloaded, the tailgate is closed and the driver continues to the next delivery location.

In this simulation, the tailgate is closed when the CD is placed close to the infrared reflective sensor. The small LED on the sensor is turned on and the LED connected to the D5 socket on the base shield is turned off.

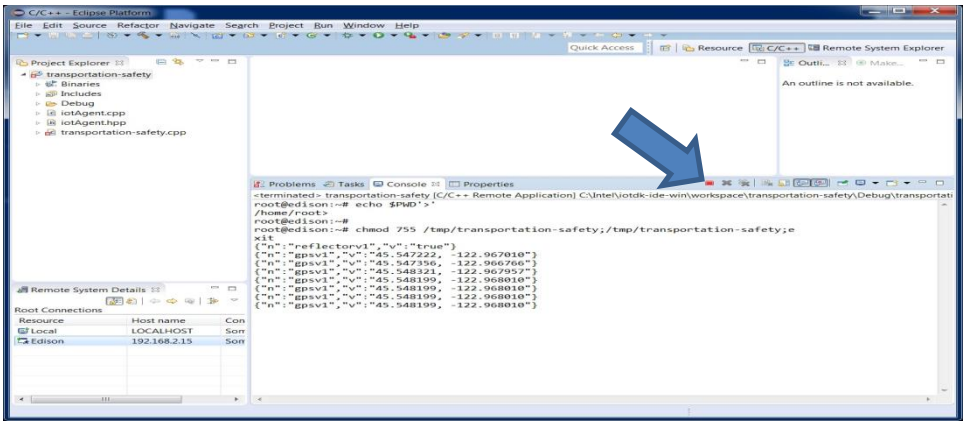


The tailgate is open when the CD is moved away from the infrared reflective sensor, turning off the small LED on the sensor. The LED connected to D5 turns on, warning the driver the tailgate is open. This data is sent to the cloud, as shown in the Eclipse Console window with the following line: {“n”:”reflectorv1”,”v”:”true”}.





Click the Terminate icon ■ to stop the program.



Troubleshooting

To delete a device and re-register it on the Intel IoT Analytics cloud website, do the following:

1) Delete the device from the Intel IoT Analytics website.
2) Refresh the activation code on the website.
3) Enter the following commands in a PuTTY session with your board:

```
root@edison:~# iotkit-admin

root@edison:~# iotkit-admin initialize

root@edison:~# iotkit-admin device-id

root@edison:~# iotkit-admin activate <new_activation_code>

root@edison:~# iotkit-admin register gpsv1 gps.v1.0

root@edison:~# iotkit-admin register reflectorv1
reflector.v1.0
```

```
root@edison:~# iotkit-admin observation gpsv1 "location 1"
root@edison:~# systemctl start iotkit-agent
```

4) Re-run the transportation-safety project to upload your data to the cloud.