

Lab #2a: TI Launchpad and M2X

Pre-requisites: Energia IDE, TI LaunchPad CC3200 device and USB cord
Length: 30 minutes

In this exercise:

Setting up the TI LaunchPad CC3200 Device

Obtaining the Baseline Code

Modifying the Code to Send Accelerometer Data

Running the Code

Troubleshooting

Summary

You can use the AT&T M2X Data Service with any device that can make HTTP requests. In this exercise, you will be using the TI LaunchPad CC3200 device that can connect to WiFi. We will provide you with baseline code that contains classes making it easy to communicate with both the device and the M2X service. The code reads data from the device and makes the calls to the M2X service.

In this exercise, you will read accelerometer data from the LaunchPad device and send it to an M2X stream if it calculates that the device is tilted above a certain angle.

Note: If you have trouble getting the device to work properly, see the Troubleshooting section.

Setting up the TI LaunchPad CC3200 Device

Follow these steps to set up the TI LaunchPad CC3200:

NOTE: We'll be using a pre-production version of the Energia IDE. You must use the Energia IDE linked from the directions below, not the release version on the energia.nu website. Check back to the energia.nu site frequently for the production version.

Mac OS X

1. Do **not** connect your CC3200 LaunchPad to your Mac. If you already plugged it into your Mac then unplug it before proceeding to step 2.
2. Download the CC3200 Drivers for Mac OS X: [CC3200 LaunchPad CDC drivers disk image for Mac OS X](http://energia.nu/files/cc3200_drivers_mac.dmg). (http://energia.nu/files/cc3200_drivers_mac.dmg)
3. Double click the file `cc3200_drivers_mac.dmg` to mount the image.
4. Double click the `FTDIUSBSerialDriver_10_4_10_5_10_6_10_7.pkg` and follow the instructions.
5. Reboot your Mac.
6. Connect your CC3200 LaunchPad to your Mac. The CC3200 will be automatically recognized.
7. Now that the drivers are installed, download Energia from here: <http://energia.nu/files/energia-0101E0013b2-macosx.zip>
8. Unzip and run Energia and make sure that you have the right serial port selected using "Tools->Serial Port".

Windows

1. Do **not** connect your CC3200 LaunchPad to your PC. If you already plugged it into your PC then unplug it before proceeding to step 2.
2. Download the CC3200 Drivers for Windows: [CC3200 LaunchPad CDC drivers zip file for Windows 32 and 64 bit](http://energia.nu/files/cc3200_drivers_win.zip) (http://energia.nu/files/cc3200_drivers_win.zip)
3. Unzip and double click DPinst.exe for Windows 32bit or DPinst64.exe for Windows 64 bit.
4. Follow the installer instructions.
5. Connect your CC3200 LaunchPad to your PC. The CC3200 will be automatically recognized.
6. Now that the drivers are installed, download Energia from here: <http://energia.nu/files/energia-0101E0013b2-windows.zip>
7. Unzip and run Energia and make sure that you have the right serial port selected using "Tools->Serial Port".

All Platforms

1. The LaunchPad was designed in such a way that you have to switch between programming and run mode by placing or removing the SOP2 jumper. We have worked around this by (ab)using one of the JTAG pins and doing this for you automatically. The only thing you need is a jumper wire from J8 (emulator side) to SOP2 (CC3200) side. See this diagram: <https://github.com/attm2x/devlab/LaunchPad/cc3200BlueWire.pdf> . Once this is in place you can program the LaunchPad from Energia without having to fiddle with the SOP2 jumper.
2. Attach the USB cord to the device using the mini-USB port.
3. Attach the other end of the USB cord to your computer's USB port. Note that the device is now powered by USB, so you do not have to use a power cord.

More information about the device can be found here:

<http://www.ti.com/ww/en/launchpad/launchpads-connected-cc3200-launchxl.html>

Obtaining the Baseline Code

The TI LaunchPad platform can use the Energia IDE, which is based on the popular Arduino/Wiring framework. If you've used the Arduino IDE, then you'll immediately see the similarities. *(We'll be using a pre-release version of the Energia IDE compiled for this device.)*

1. You will first need to download and install the M2X Client Library for LaunchPad (M2XStreamClient). Clone the repository or download the ZIP file from here: <https://github.com/attm2x/m2x-launchpad-energia>

This library depends on [jsonlite](https://github.com/citrusbyte/jsonlite) (<https://github.com/citrusbyte/jsonlite>), the installation steps are as follows:

2. Clone the [jsonlite](https://github.com/citrusbyte/jsonlite) (<https://github.com/citrusbyte/jsonlite>) repository.

NOTE: Since we are now using the old v1.1.2 API (we will migrate to the new API soon), please use the fork version of jsonlite listed above instead of the original one.

3. Open the Energia IDE, click Sketch->Add File..., then navigate to amalgamated/jsonlite folder in the cloned jsonlite repository. The jsonlite library will be imported to Energia this way.

NOTE: If you cloned the jsonlite library, there will be 3 folders named jsonlite:

jsonlite: the repo folder
 jsonlite/jsonlite: the un-flattened jsonlite source folder
 jsonlite/amalgamated/jsonlite: the flattened jsonlite source

The last one here should be the one to use, the first 2 won't work!

4. Use the instructions outlined in Step 3 above to import the M2XStreamClient library in the current folder.

We will now download the Lab code:

5. From the Energia IDE, click File -> New
6. Navigate to the DevLab github page <https://github.com/attm2x/devlab/LaunchPad> and open the LaunchPad3200DevLab.ino file.
7. Copy this file. Paste into the Energia IDE blank sketch page.

Modifying the Code to Send Accelerometer Data

Let's say your fleet of vehicles are carrying cargo that should be kept flat. You want to collect data when the cargo is tipped above 20 degrees. We've added some code to read the accelerometer values, calculate the angle, and send data to your stream.

Follow these steps to modify the code:

1. Change **<key>** to your M2X key and **<feed>** to your feed ID, and **<stream>** to **"tilt"**, which is the name of the stream that will collect the data.
2. Change **<ssid>** to your wireless SSID, and **<password>** to your wireless password.
3. This will use the accelerometer values to calculate the pitch and roll angles of the device. If the maximum of these two is greater than 20 degrees, it will send the value to the tilt stream. Then, so that it doesn't overwhelm the stream with data, it waits 1 second before taking another accelerometer reading.

```
float x = mySensor.readXData()/65.0;
Serial.print("Accel X: ");
Serial.print(x);

float y = mySensor.readYData()/65.0;
Serial.print(", Y: ");
Serial.print(y);

float z = mySensor.readZData()/65.0;
Serial.print(", Z: ");
Serial.println(z);

// Calculate pitch and roll. Find the maximum tilt angle.
float pitch = atan(x / sqrt(y * y + z * z))*180;
float roll = atan(y / sqrt(x * x + z * z))*180;
float maxTilt =
  max(abs(roll), abs(pitch)) / 3.14159;
Serial.print("pitch: ");
Serial.print(pitch);
Serial.print(" roll: ");
Serial.print(roll);
Serial.print(" maxTilt: ");
Serial.println(maxTilt);

// If the maximum tilt is over 20 degrees, then send
// data to stream
if (maxTilt > 20) {
```

```

int response = m2xClient.post(feedId, streamName, maxTilt);
Serial.print("M2x client response code: ");
Serial.println(response);

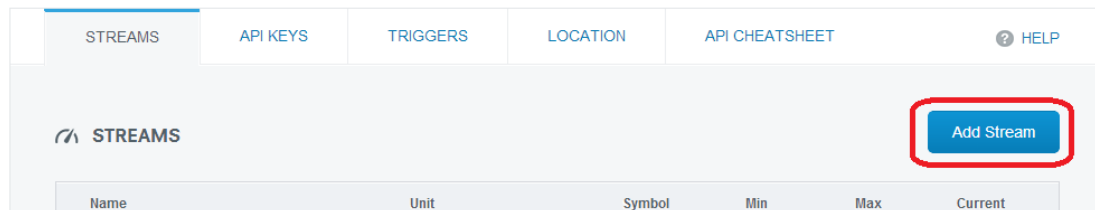
if (response == -1)
  while (1)
    ;
  delay(500);
}

```

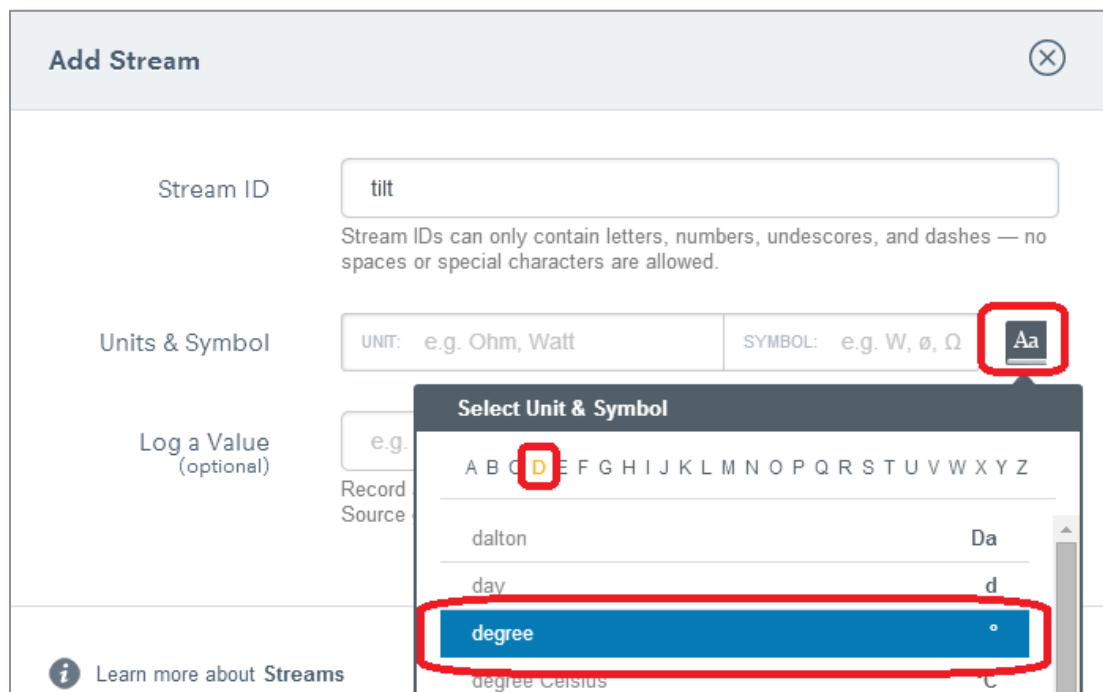
Running the Code

To run the code, first we need to set up a stream called "tilt" to collect the tilt angle data when it is greater than 20 degrees. Follow these steps:

1. Return to your Data Source Blueprint page and open the feed page for the feed you've been using. Click the **Add Stream** button.



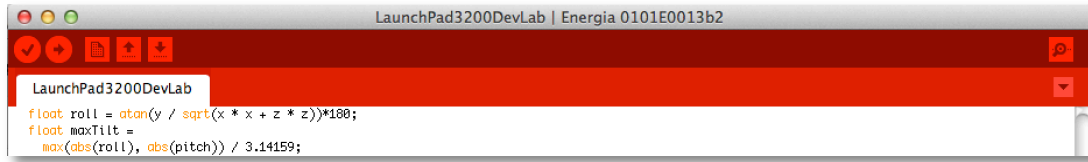
2. Type in "tilt" for the Stream ID. Then click the **Aa** button on the **Units & Symbols** line. Click the letter **D**, and then choose **degree**.



3. Click the **Add Stream** button.

Now return to the Energia IDE and follow these steps to compile your program and load it onto the device.

1. Save the file as “LaunchPad3200DevLab”
2. Click the **Upload** button. (Right arrow at the top of the compiler window.)



3. Once the compiler says “Download Complete...”, click the **Serial Monitor** button at the top right of the Compiler window to see the progress of the device.
4. To run the code, press the reset button, which is a small button at the end of the board with the USB port.
5. Put the device flat on the desk. Wait about 20 seconds for the device to start up and connect to the wireless network. In the meantime, open your “tilt” stream and scroll down to look for data to be collected.
6. You should not be seeing any data on the stream’s graph because the angle is below 20 degrees. Pick up the device and hold it at an angle. After a few seconds, data will start showing up on the graph. Play around with larger and smaller angles and watch for the changes, which will take a few seconds to appear. If you drop below 20°, the data will not be sent to your stream.

Troubleshooting

You may be following the steps in this exercise and find that data is not being sent to your M2X stream. The Energia code contains print lines that can help you troubleshoot and figure out if the problem is in the WiFi connection, the accelerometers, the M2X key, etc.

Click the Serial Monitor button on the Energia compiler window to launch a terminal window to see the debug output from the device.

You can also search the <http://energia.nu/guide/> site for additional troubleshooting information.

Summary

You’ve learned how to import LaunchPad code into the Energia IDE editor and to modify it so that it reads accelerometer data and sends data.