Edison workshop

Alex Tereschenko <alext.mkrs@gmail.com>

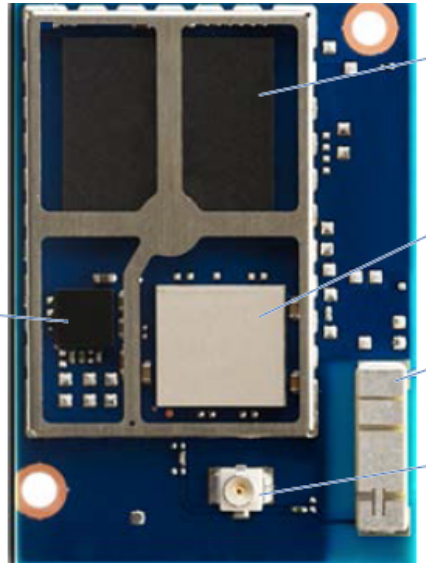
What's Edison

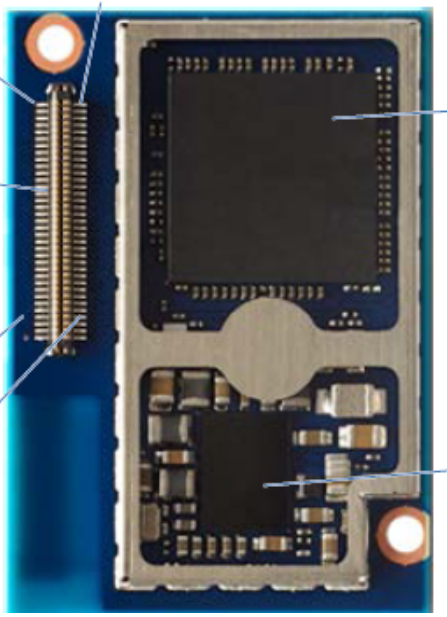
- Intel(r) Edison* Compute Module is a System on Module (SOM), fully-featured computer on a 25mm x 35.5mm board
- You use it with either off-the-shelf or custom designed expansion boards
- It is certified for Wi-Fi and Bluetooth in 68 countries, making it easier to create commercial products/prototypes with the Module as a core

Compute Module HW specs

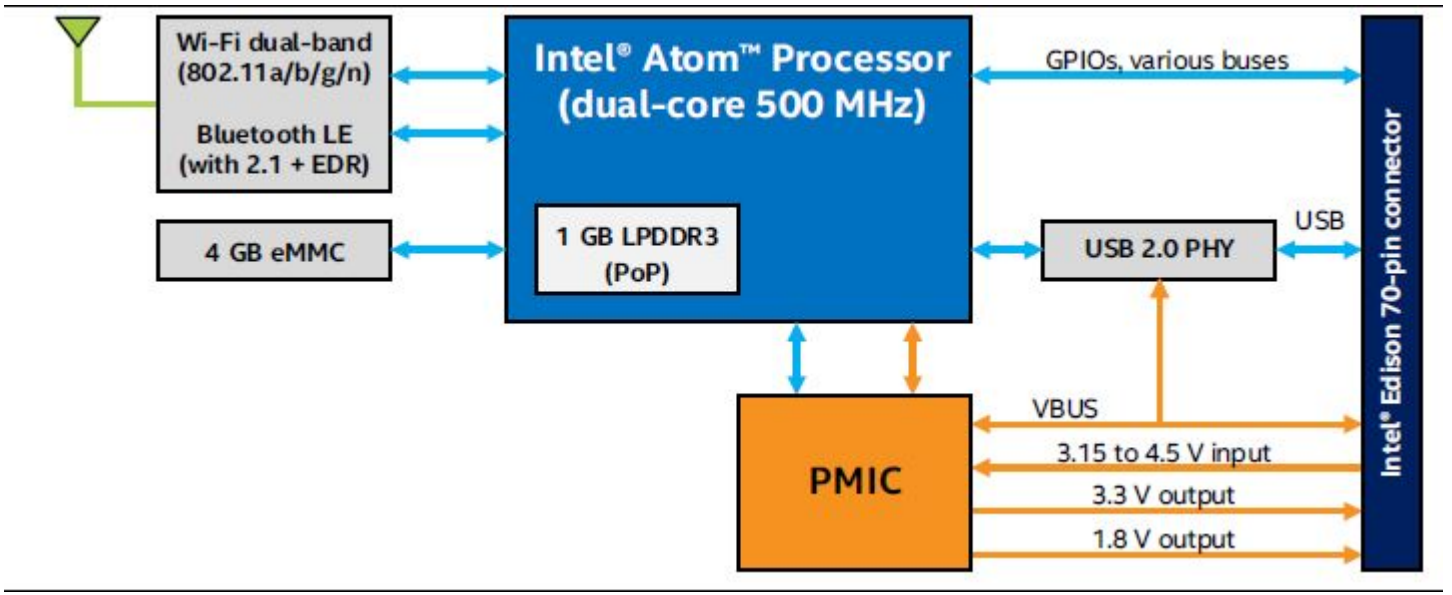
- CPU: 2x500MHz 32-bit Atom™ cores with SSE support
- MCU: 1x100MHz 32-bit Quark™ core
- RAM: 1GB 2-channel LPDDR3
- Flash: 4GB eMMC
- Wireless: dual-band WiFi 802.11 a/b/g/n with onboard antenna, BT 4.0 (including Low Energy)
- Connectivity/buses: SD controller, 2xUART, 2xI2C, 1xSPI, 1xI2S, 12xGPIO (4xPWM), 1xUSB 2.0 OTG, 2x Clock output (32 KHz, 19.2 MHz)
- See more details in Hardware Guides, URLs are on the Links slide

Compute Module outline

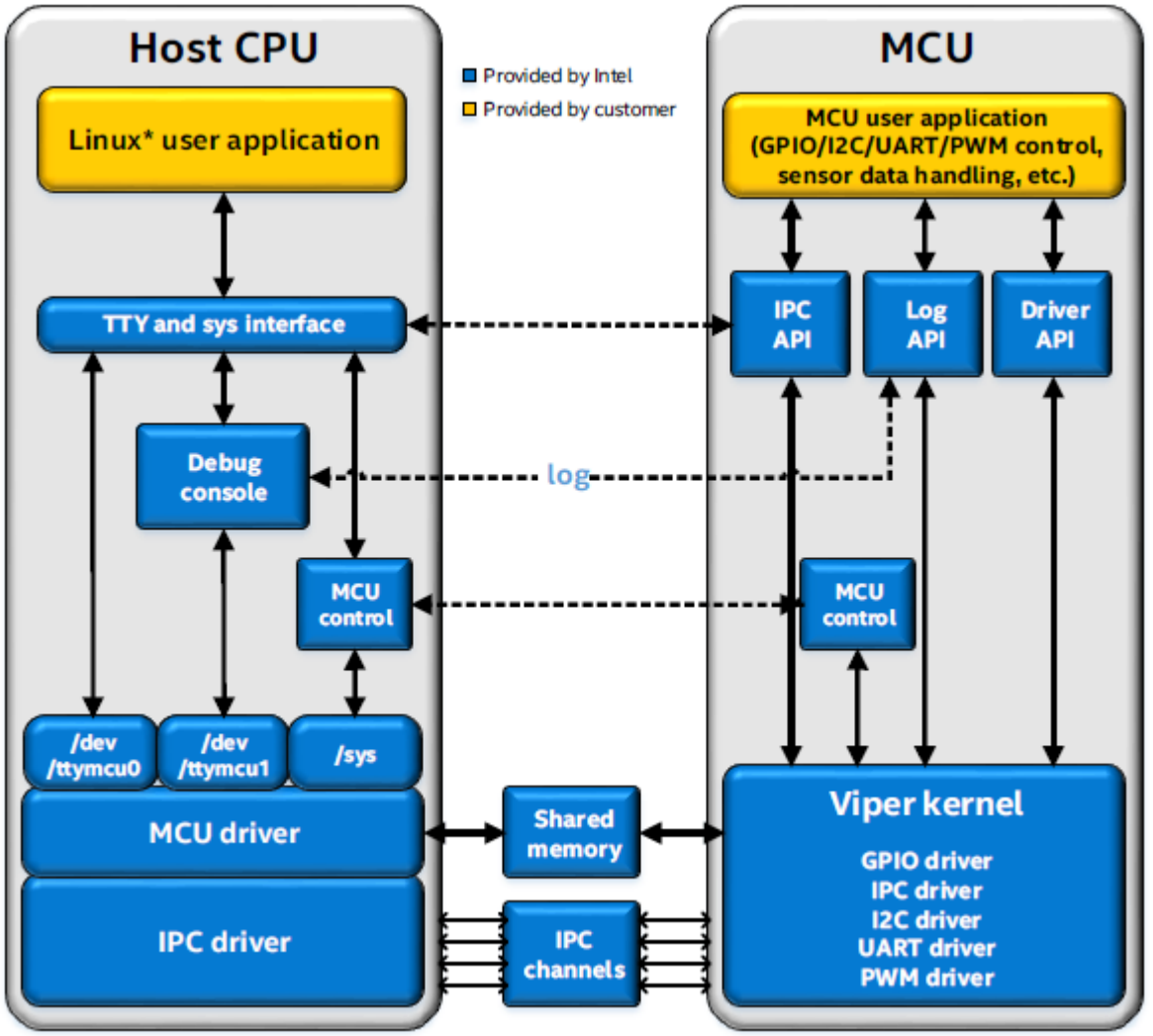




Edison architecture - overall

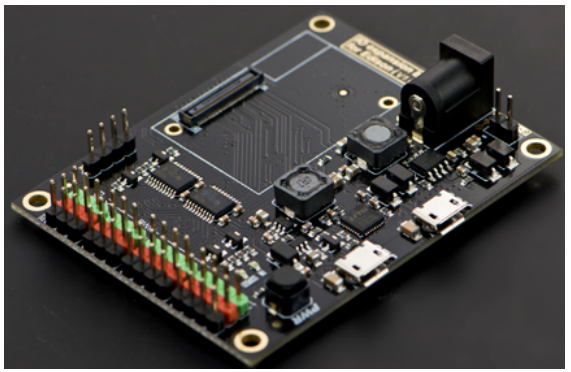


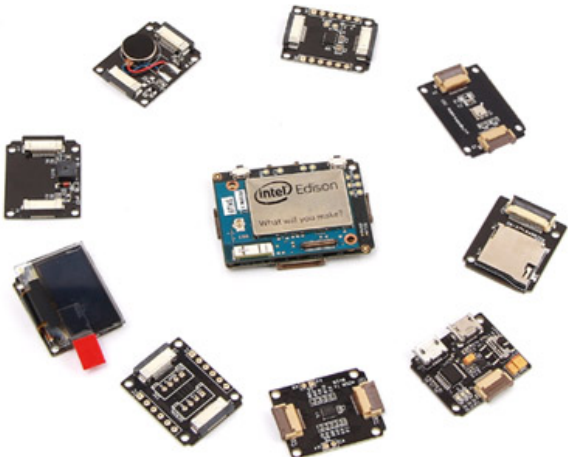
Edison architecture – CPU/MCU

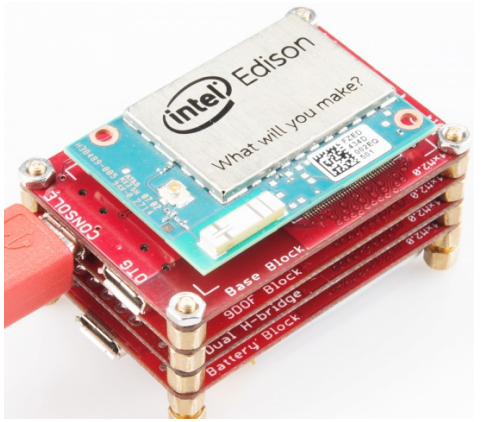


Expansion boards

- Intel(r) Edison Kit for Arduino*
- Intel(r) Edison Breakout Board Kit
- SparkFun Blocks for Edison
- Xadow Wearable Kit
- DFRobot IO Expansion Shield
- + more, just search

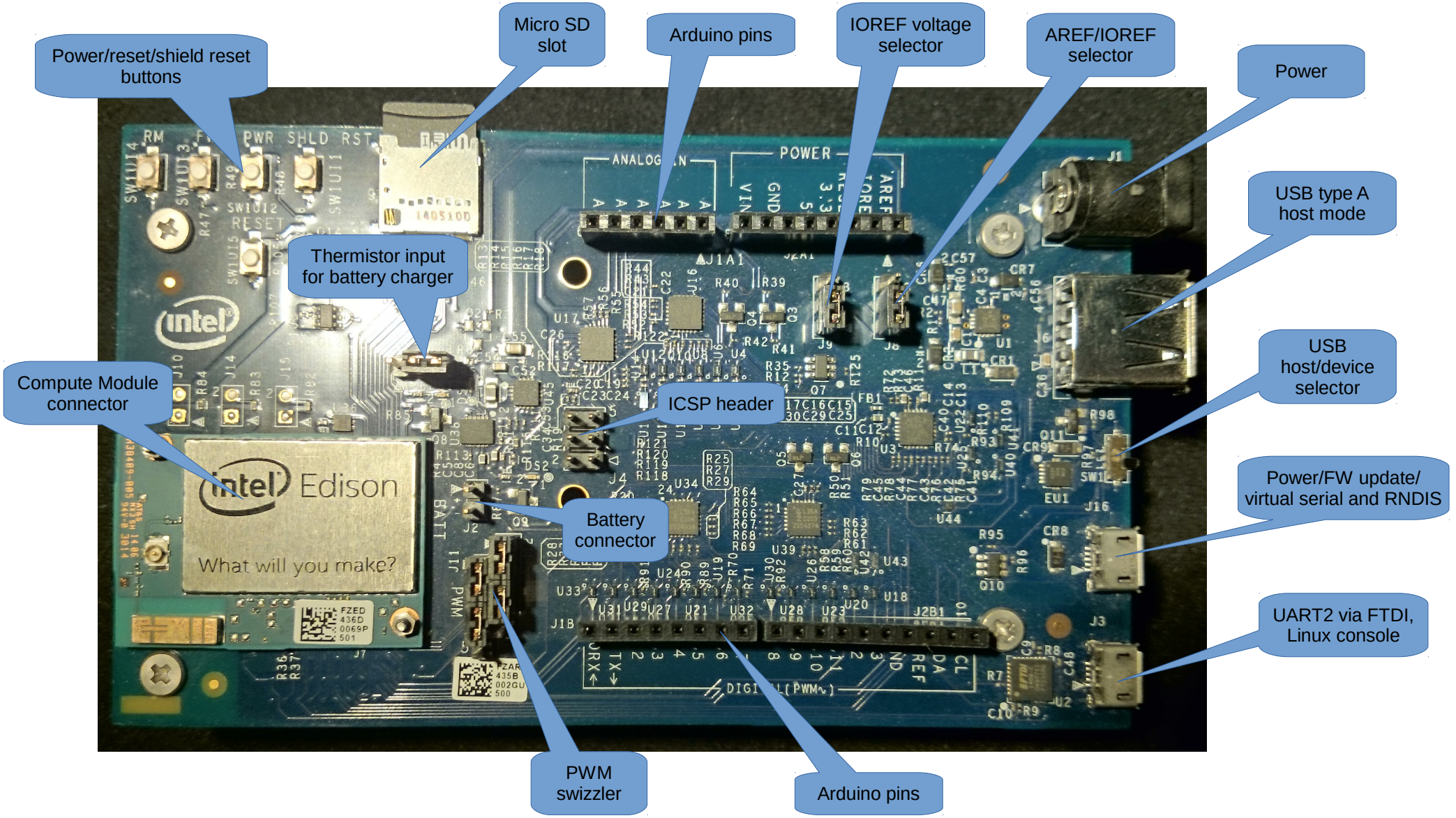




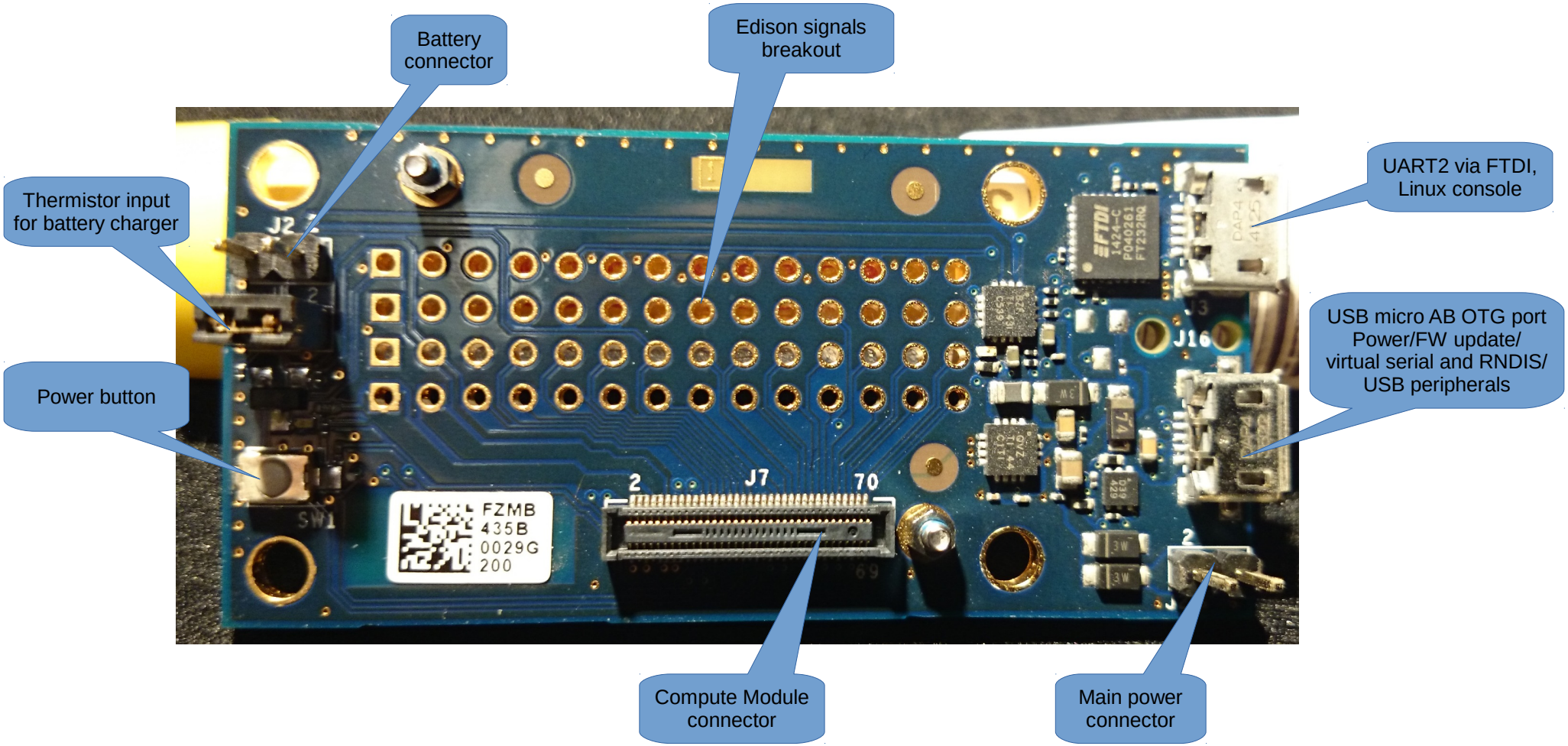


*Other names and brands may be claimed as the property of others.

Expansion boards - Arduino



Expansion boards - breakout



Software environment

- Main OS: Linux-based, Yocto Poky 1.7.2 (kernel 3.10.17)
  - There's an unofficial package repo supported by yours truly, see instructions at http://alextgalileo.altervista.org/edison-package-repo-configuration-instructions.html
- MCU: Viper* RTOS kernel

Programming Edison

- Intel(r) XDK IoT Edition (JavaScript/Node.js)
- Eclipse (C/C++/Java)
- Arduino IDE
- Wyliodrin
- Vi ☺ (C/C++/Java/Python/JS – via mraa & upm)
- MCU is programmed with Eclipse-based MCU SDK using C

Hands-on

Workshop project

Weather station

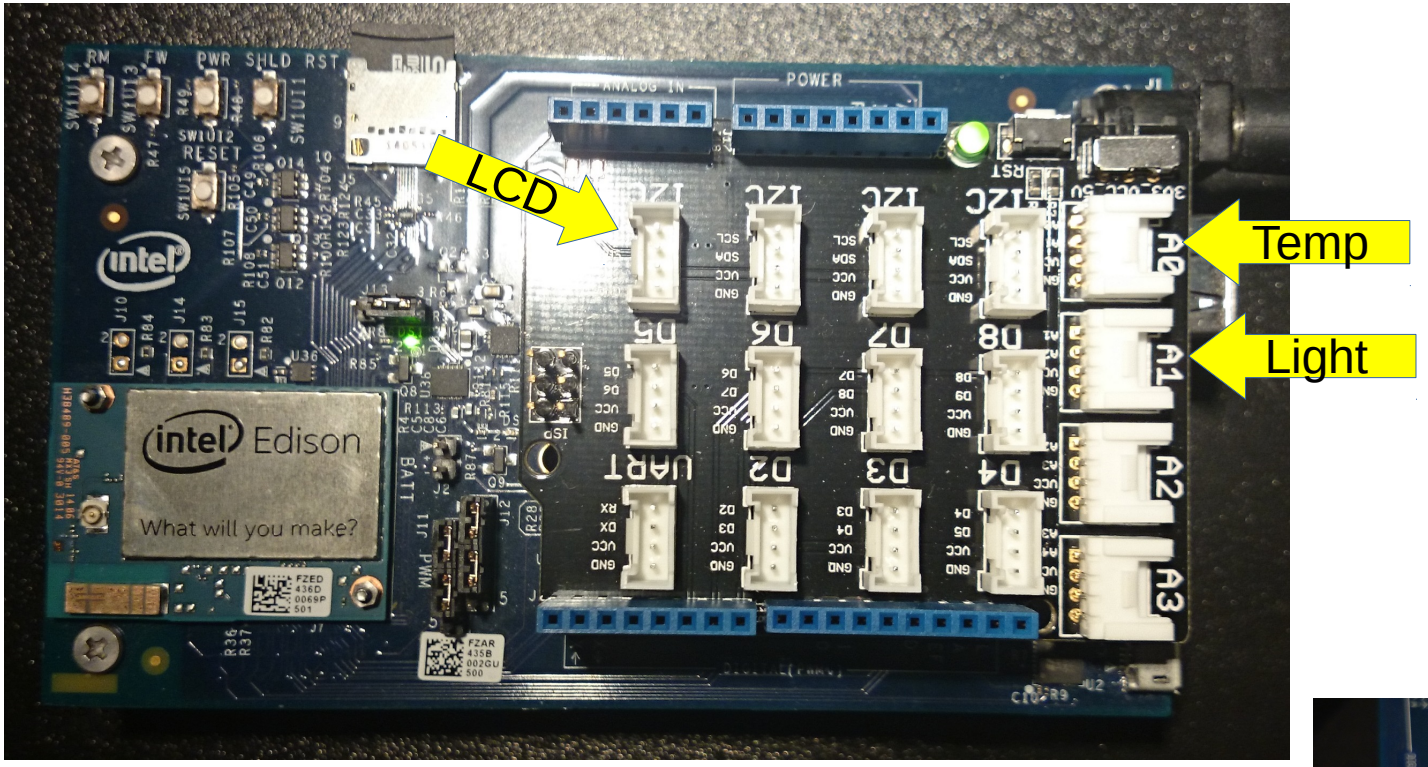
- Getting data from temperature and light sensors
- Displaying data on an LCD screen
- Presenting data through a web interface

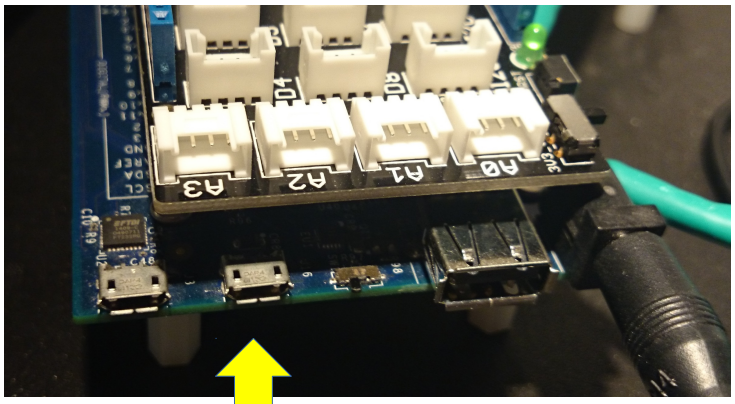
A simple example for you to get the hang of Edison

Environment setup/prework

- Install PuTTY (or your favorite terminal emulator)
- Install WinSCP
- Install Edison drivers from https://software.intel.com/en-us/iot/hardware/edison/downloads (look for “Windows Standalone Driver”)
- Get your favorite text editor (I used vim on the board and Notepad++ on the host)
- Connect Grove Temperature Sensor module to a Base Shield connector A0, Grove Light module to A1, Grove LCD to any connector marked “I2C”
- Connect Edison's microUSB port marked J16 (the middle one) to your computer. Edison should power on and boot and you should see Virtual COM and RNDIS devices created in Device Manager.
- Make sure your computer has an address from 192.168.2.x/24 subnet range assigned to the virtual RNDIS adapter. If it doesn't just configure using standard means or ask the instructor :)

Environment setup – contd.





Implementation

- We'll use MRAA library through UPM modules
- We'll use Python and directly edit files on the device using your preferred text editor and WinSCP
- All sources and slides themselves are available at https://github.com/alext-mkrs/edison-workshop

Implementation – contd.

- Login to your Edison using PuTTY, username is “root”, password is empty. Connect using WinSCP as well – that way you can edit files on the board by just double-clicking them in WinSCP.
- Go into the workshop's directory and reset the state by running the below two commands:

```
cd /home/root/edison-workshop
```

```
git reset --hard
```

- Look into the `/home/root/edison-workshop/code` directory. The sample project is split into four parts, from just sensor data acquisition to a full set. Run `run.py` script in each directory to see how it works. In web UI part open your browser and go to http://192.168.2.15:5000, that is the address and the port for our app running on Edison.
- Create a new directory and start creating your own version, or just go and modify the existing sources to see how it works
- If you want to pursue your own project – go ahead and feel free to use other sensors available

Workshop project walkthrough

Weather station

- **Step 1: getting data from temperature and light sensors**
- Step 2: displaying data on an LCD screen
- Step 3: presenting data through a web interface
- Step 4: complete project

A simple example for you to get the hang of Edison

Workshop project walkthrough

Weather station

- Step 1: getting data from temperature and light sensors
- **Step 2: displaying data on an LCD screen**
- Step 3: presenting data through a web interface
- Step 4: complete project

A simple example for you to get the hang of Edison

Workshop project walkthrough

Weather station

- Step 1: getting data from temperature and light sensors
- Step 2: displaying data on an LCD screen
- **Step 3: presenting data through a web interface**
- Step 4: complete project

A simple example for you to get the hang of Edison

Workshop project walkthrough

Weather station

- Step 1: getting data from temperature and light sensors
- Step 2: displaying data on an LCD screen
- Step 3: presenting data through a web interface
- **Step 4: complete project**

A simple example for you to get the hang of Edison

Things to try further

- Get Bluetooth-enabled sensors and access them using PyBlueZ
- Draw nice charts using something like http://www.highcharts.com/demo/combo-meteogram
- Upload data to the cloud for further analysis and data preservation, e.g. http://enableiot.com or Xively or whatever else.

Links

- Edison home page: http://www.intel.com/content/www/us/en/do-it-yourself/edison.html
- Edison main downloads page: https://software.intel.com/en-us/iot/hardware/edison/downloads
- Edison Compute Module hardware Guide: http://www.intel.com/support/edison/sb/CS-035274.htm
- Edison Kit for Arduino Hardware Guide: http://www.intel.com/support/edison/sb/CS-035275.htm
- Edison Kit for Arduino Schematic: https://communities.intel.com/docs/DOC-23309
- Edison Breakout Board Hardware Guide: http://www.intel.com/support/edison/sb/CS-035252.htm
- Edison Breakout Board Schematic: http://www.intel.com/support/edison/sb/CS-035255.htm
- MCU Application development Getting Started guide: https://software.intel.com/en-us/creating-applications-with-mcu-sdk-for-intel-edison-board
- Mraa library: https://github.com/intel-iot-devkit/mraa
- UPM: https://github.com/intel-iot-devkit/upm

That's it!

Thank you :)