

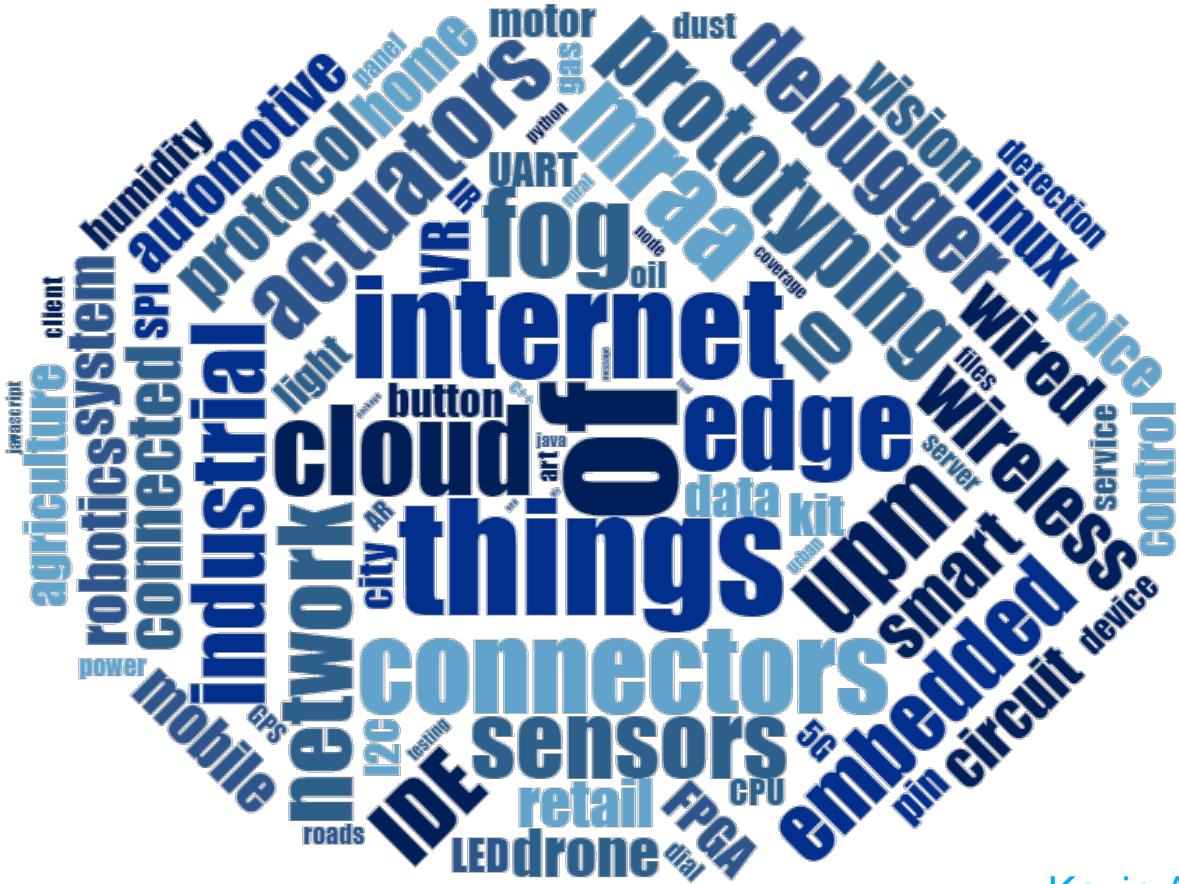


An Introduction to The Internet of Things

where and how to start

November 2017

Mihai Tudor Panu <mihai.tudor.panu@intel.com>



EST. 1999

Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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

Kevin Ashton, P&G



Agenda

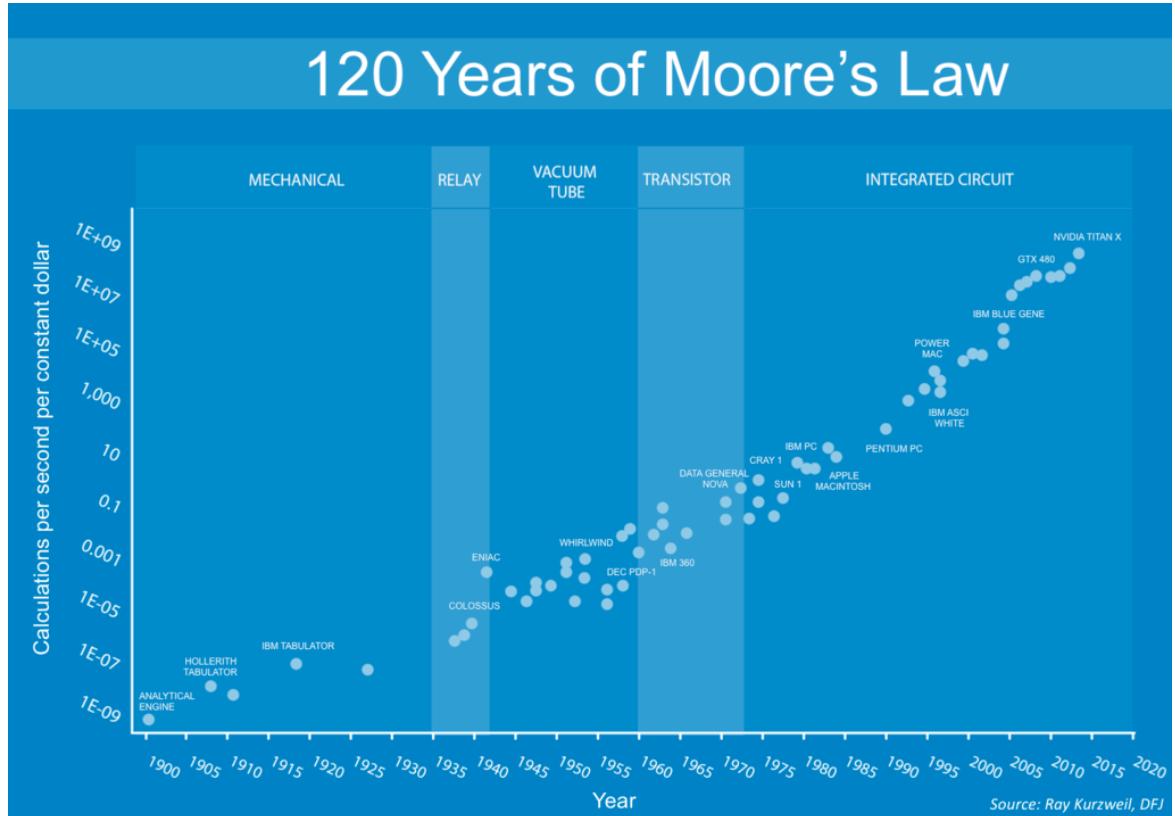
- High level key concepts surrounding IoT
- Easy to use examples and fields they cover
- Developing end-to-end solutions
 - Retail, Automotive, Home & City, Industrial, Computer Vision
- Hardware platforms and tools
- Software to get you going

Concepts

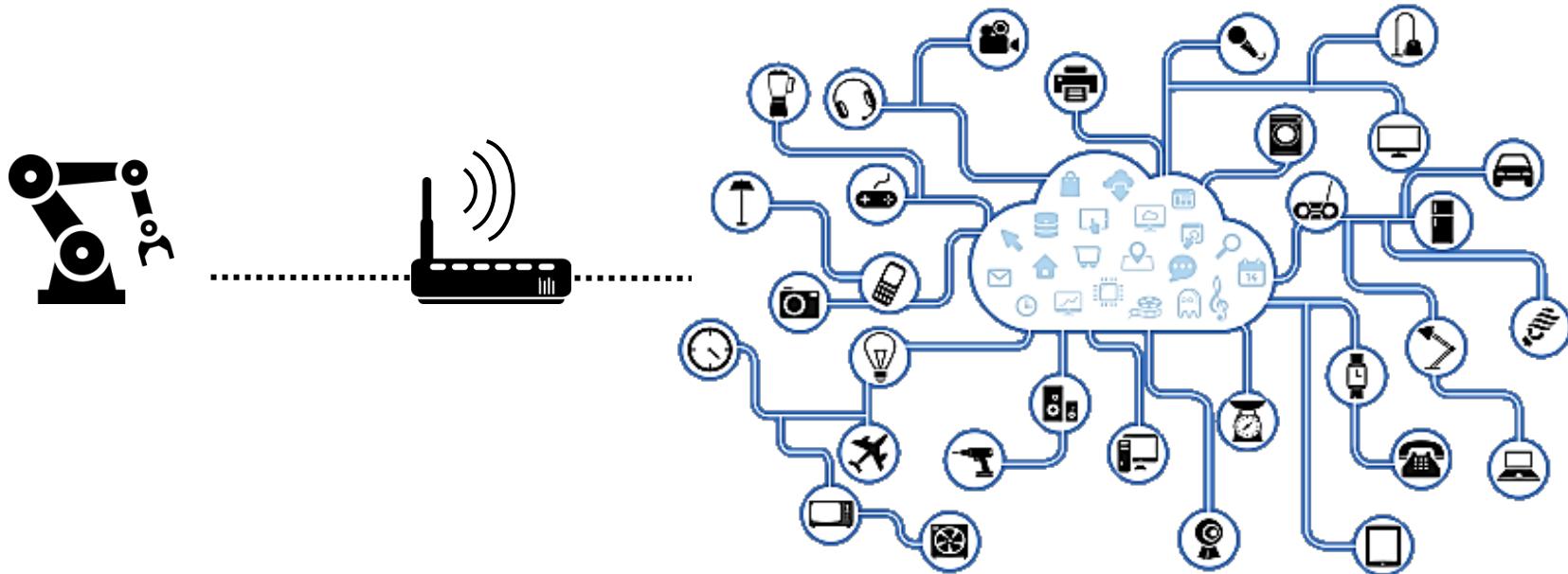
The IoT Enabler: Moore's Law

“The number of transistors in integrated circuits will double every 2 years.”

Gordon Moore,
Intel cofounder

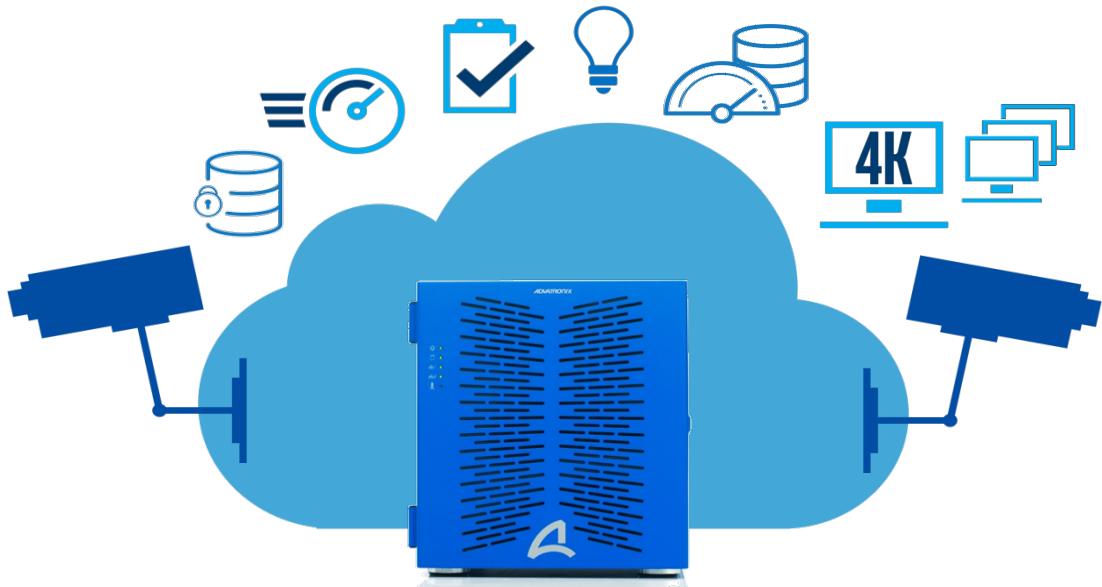


Typical IoT Solution Layout



Edge \leftrightarrow Gateway \leftrightarrow Cloud

More Powerful Edge – Fog



Microservers:

- Can connect to sensors directly and function as an edge devices too
- Plenty of compute power and storage gives more options for data aggregation and processing
- Ability to run and use private cloud solutions locally or distributed across sites
- Enhanced security due to dedicated hardware (e.g. crypto, TPM)
- Custom accelerator cards for advanced algorithms and data processing
- Great fit for Vision, AI, Machine Learning and Big Data

Example applications

How-to Code Samples

18 Complete how-to code samples in multiple programming languages:

- Myriad of IoT starter applications to explore
- Compatible with SeeedStudio, DFRobot sensor kits and many others
- Will run on any board with minimal code changes
- Learn how to use cloud services: AWS*, Bluemix*, Azure*, M2X*, Predix*, SAP*
- Different technologies for sending data: MQTT, REST



<http://github.com/intel-iot-devkit/how-to-code-samples>

These applications are code sample exercises using the Intel IoT Developer Kit, Intel Edison board or Intel IoT Gateway, sensors, actuators, cloud platforms, and APIs.

Add topics

1.661 commits	2 branches	0 releases	16 contributors
Branch: master	New pull request	Create new file	Upload files
chestondev committed on GitHub Merge pull request #312 from intel-iot-devkit/feature/js/sap ...			
access-control	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
air-quality-sensor	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
alarm-clock	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
ble-scan-bracelet	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
close-call-reporter	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
color-match-game	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
docs/cpp	Add missing d3.js license info, and also add back hex2c.sh shell script	10 months ago	
doorbell	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
earthquake-detector	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
equipment-activity-monitor	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
field-data-reporter	Merge pull request #269 from intel-iot-devkit/IanXSmith/patch-37	a month ago	
fire-alarm	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
home-fall-tracker	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
images	Update color match example image with plain Edison board	5 months ago	
line-following-robot	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
plant-lighting-system	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
range-finder-scanner	[py/predix] Clean up various code refactoring.	a month ago	
robot-arm	[py/upm] Update Python samples to use new upm Icd driver modules.	a month ago	
smart-stove-top	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
sound-detector	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
storage-unit-flood-detector	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
watering-system	Merge pull request #312 from intel-iot-devkit/feature/js/sap	21 days ago	
.gitignore	[py/alarm-clock] Implement core alarm functions.	6 months ago	
README.md	Revert "[py/fire-alarm] Initial structure commit"	6 months ago	

How-to Code Samples

Where do they fit:

- Healthcare and assistance
 - Home Fall Tracker
- Security
 - Access Control
- Agriculture
 - Plant Lighting System
 - Watering System
- Robotics
 - Robot Arm
 - Line Following Robot
- Smart home & city
 - Alarm Clock
 - Doorbell
 - Earthquake Detector
 - Fire Alarm
 - Smart Stove Top
 - Storage Unit Flood Detector
 - Air Quality
- Industrial
 - Equipment Activity Monitor
 - Sound Detector
- Wearables
 - BLE Bracelet



<http://github.com/intel-iot-devkit/how-to-code-samples>

Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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



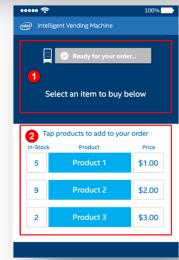
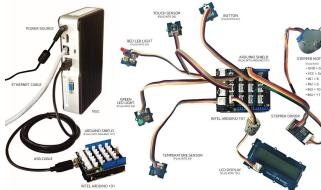
Developing End to end solutions

Intelligent Vending Machine

This retail vertical project monitors the inventory, product sales, and maintenance of a vending machine. The gateway gathers data from a temperature sensor, stepper motor, coil switch, and a product-purchasing application for edge data analytics. The prototype was created using the **Intel® IoT Developer Kit**, **Intel® IoT Gateway Software Suite**, **Grove*** **IoT Commercial Developer Kit**, **Intel® System Studio IoT Edition**, and **Microsoft* Azure*** cloud services and then deployed to an **Intel® IoT Gateway** using industrial sensors and a miniature scale vending machine.

Articles:

- ["The Making Of" Story](#)
- ["How To" Build This Solution](#)
- ["Code" Available on GitHub](#)



Transportation

This connected transportation project monitors the status of a refrigerated trailer. The gateway gathers data from a temperature and magnetic sensor for edge data analytics and monitoring. The prototype was created using the **Intel® IoT Developer Kit**, **Intel® IoT Gateway Software Suite**, **Grove* IoT Commercial Developer Kit**, and **Intel® System Studio IoT Edition**, and then deployed to an **Intel® IoT Gateway** using industrial sensors, **Intel® XDK** and a miniature scale truck trailer.

Articles:

- [The "Making Of" Story](#)
- [How To Build This Solution](#)
- [Code Available on GitHub](#)

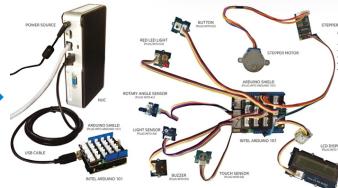


Smart Home

This smart home vertical project monitors the status of a home's front door and garage door for increased security. The gateway gathers data from a doorbell, door lock, stepper motors, and a garage door application for edge data analytics. The prototype was built using the **Intel® IoT Developer Kit**, **Intel® IoT Gateway Software Suite**, **Grove* IoT Commercial Developer Kit**, **Intel® XDK**, and **IBM Bluemix*** and then deployed to an **Intel® IoT Gateway** using industrial sensors and a miniature scale home.

Articles:

- [The "Making Of" Story](#)
- [How To Build This Solution](#)
- [Code Available on GitHub](#)



Environment Monitor

This smart building vertical project monitors air quality and pollutant levels in the surrounding environment. The gateway gathers data from a temperature and humidity sensor, a gas sensor, and a dust particle sensor. The solution was created using the **Intel® IoT Developer Kit**, **Grove* IoT Commercial Developer Kit**, **Intel® System Studio IoT Edition**, and **Amazon* AWS*** cloud services and specialized sensors. It was deployed to a portable enclosure.

Articles:

- ["The Making Of" Story](#)
- ["How To" Build This Solution](#)
- ["Code" Available on GitHub](#)

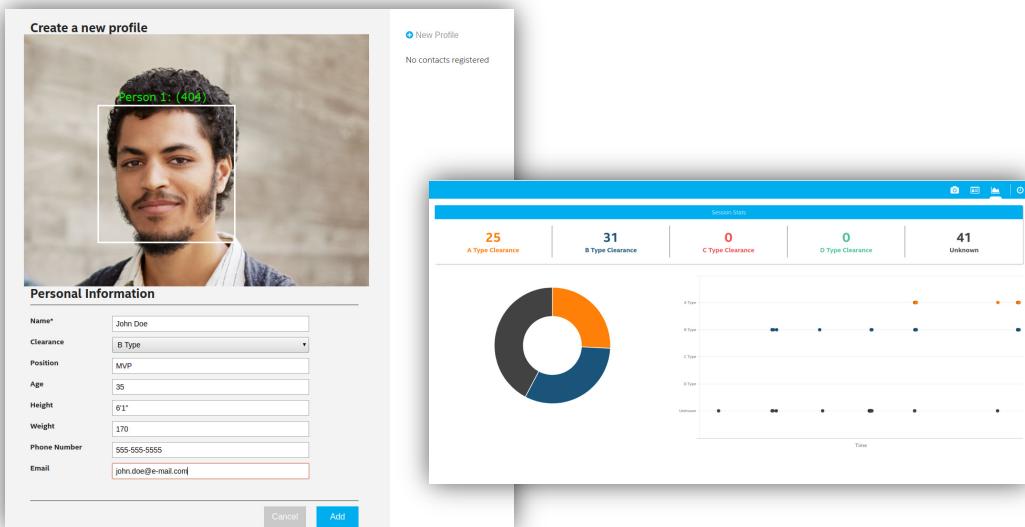


Face Access Control

The Face Access Control application uses facial recognition as the basis of a control system for granting physical access. The application detects and registers the image of a person's face into a database, recognizes known users entering a designated area and grants access if a person's face matches an image in the database.

Articles:

- ["How To" Build This Solution](#)
- ["Code" Available on GitHub](#)

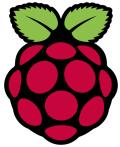


Hardware platforms

So Many Boards

Typical options:

- MCU
- FPGA
- CPU



Raspberry Pi & Compatibles



DE10-Nano



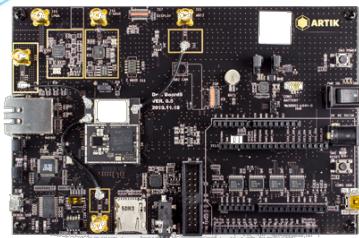
Beaglebone



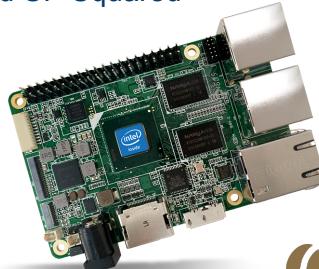
101, UNO, MEGA, Leonardo, Yun, ...



STM32 - Nucleo



UP and UP Squared



Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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

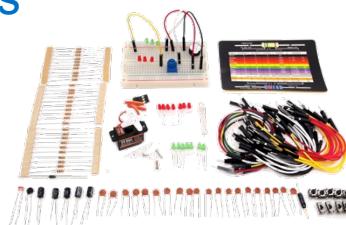
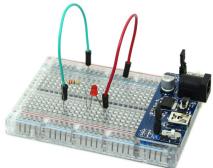


What to Get?

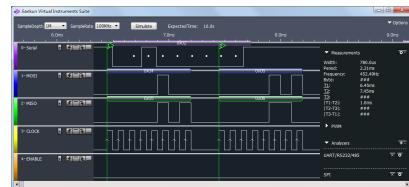
- Does your board come with a power supply?



- Breadboards, wires, cables, circuit parts



- Monitor the signals with a logic analyzer or oscilloscope



- Sensors, actuators, peripherals

Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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



Where to Get it?

- Prototyping – Platforms and Sensors



- Parts for tying it all together



DEVELOPER KITS

How-to Code Samples

- UP Squared IoT Grove Development Kit with GrovePi+ shield
- UP Squared is RPi compatible
- Full kit listing:

What's in the box

UP Squared board (Celeron N3350, 2GB RAM / 32GB eMMC)

Power supply 5V @ 6A

EU/US Power cord

16GB USB memory stick

Micro USB3.0 serial cable

Grove interface board

Grove LCD RGB backlight

Grove light sensor v1.2

Grove button

Grove temperature and humidity sensor

Grove green LED

<http://up-shop.org>



Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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



Reference Implementations

- Start with a prototype, then transition seamlessly to an industrial grade solution that can be taken to market
- Prototypes built with the **Grove* IoT Commercial Kit**, uses **Intel® NUC** as Gateway running **Intel® Gateway Software Suite** and **Arduino 101** as a sensor hub
- What's in the kit?

1 x

[Intel® NUC Kit DE3815TYKE with 4G Memory](#)

Sensors:

- 1 x [Grove* - Button](#)
- 1 x [Grove* – Sound Sensor v1.2](#)
- 1 x [Grove* – Touch Sensor](#)
- 1 x [Grove* – Light Sensor v1.2](#)
- 1 x [Grove* – Temperature Sensor v1.1](#)
- 1 x [Grove* – Rotary Angle Sensor\(P\)](#)
- 1 x [Grove* – Piezo Vibration Sensor](#)

Others:

- 1 x [Grove* Base Shield v2](#)
- 14 x [26AWG Grove* Cable](#)
- 1 x B to A Type USB Cable – 0.5m

Actuators:

- 1 x [Grove* – LCD RGB Backlight](#)
- 1 x [Grove* – Buzzer](#)
- 1 x [Grove* – Red LED](#)
- 1 x [Grove* – Green LED](#)
- 1 x [Grove* – Blue LED](#)
- 1 x [Gear Stepper Motor with Driver](#)



SOFTWARE TO GET YOU GOING

mraa.io/demo

Before You Begin

- Learn how to deploy OS images to devices:
 - Typical media: SD card or USB flash drive
 - Rufus, Win32 Disk Imager, dd (Linux)
- Familiarize yourself with tools that allow remote connections and transfers
 - Serial connections with Putty
 - SCP file transfers
 - VNC works great for desktop access on more powerful IoT platforms
- Find out how to view system logs and install new software on the target
- Create a Github account and start using git to get samples
- Bonus: Visit Docker and embrace virtualization

Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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



Hardware Abstraction (MRAA)

- Standard IO Interface for all IoT Developer Kits
- Supports Intel & non-Intel (community) platforms
- Abstraction APIs:
 - GPIO
 - Analog (AIO)
 - PWM
 - SPI
 - I²C
 - UART



Legacy Intel® Boards

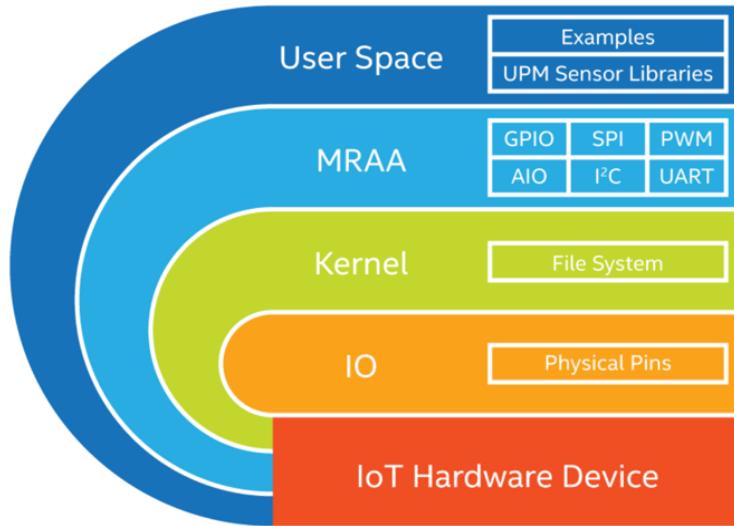


UP & UP2
MinnowBoard

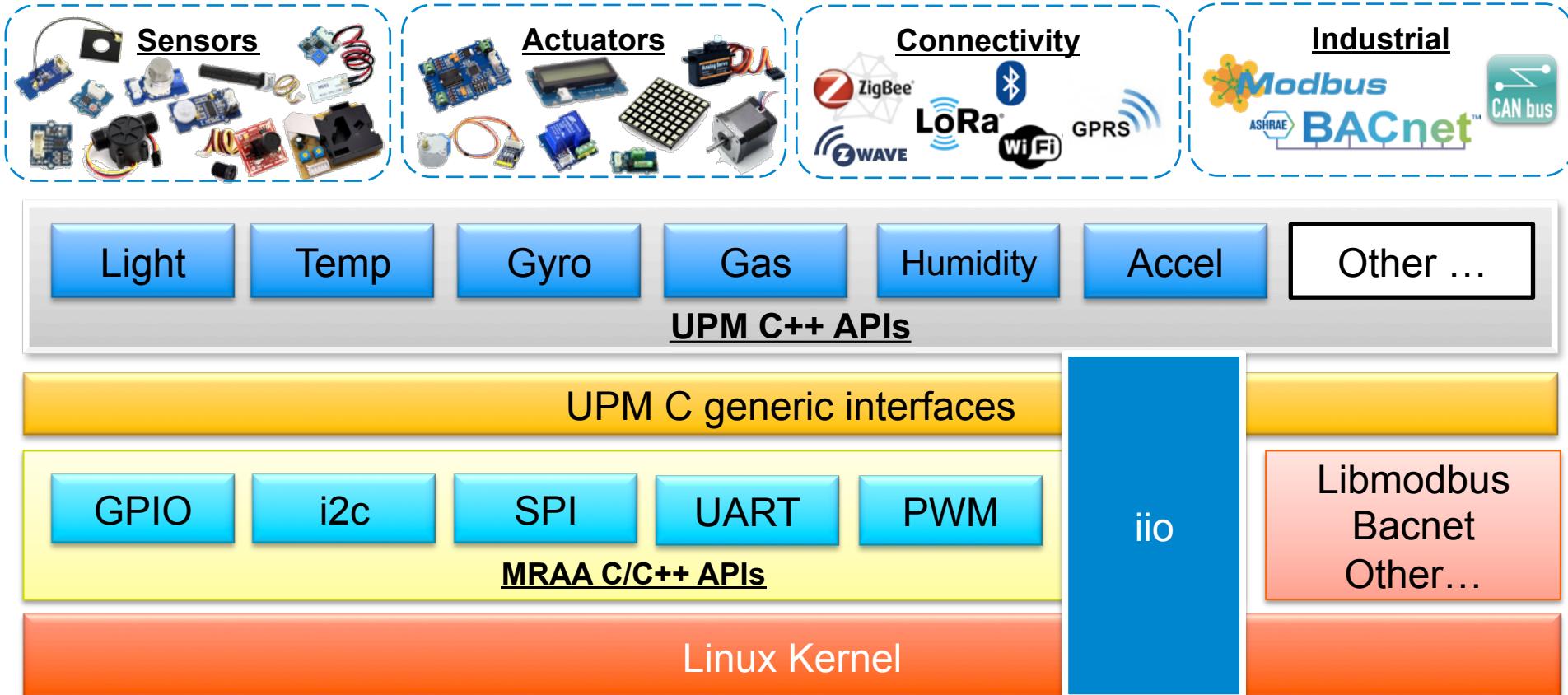


Intel® IoT Gateways
& More

Typical stack on UNIX systems:



Standardized Sensor APIs (UPM)



Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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



GitHub Repositories

github.com/intel-iot-devkit



Code Issues Pull requests Issues Marketplace Gist

Unwatch 112 ⭐ 825 Fork 392

Code Issues Pull requests Projects Wiki Settings Insights

Linux Library for low speed IO Communication in C with bindings for C++, Python, Node.js & Java. Supports generic I/O platforms, as well as Intel Edison, Intel Joule, Raspberry Pi and many more. <http://mraa.io>

Edit

mraa libmraa intel-joule galileo Manage topics

1,549 commits 27 branches 58 releases 79 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

dholiver committed with arroll travis: migrate Travis plans to run in a Docker container

Latest commit 79a3829 days ago

api uart: add a function to query current UART settings 5 days ago

cmake FindNodejs.cmake: added search paths for Ubuntu 16.04 9 months ago

docs mraa: Update to v1.7.0 20 days ago

doxygen2javadoc @ 9cc90b7 doxygen: javadoc add submodule 2 years ago

examples mraa-iuartc: Remove impossible condition with a sprintf 1 day ago

include mraa: match version number in mraa.c with mraa.conf 1 month ago

jststub uart: add tsconsbreak support 24 days ago

src Change messaging to indicate mraaSub is on nvm 11 months ago

tests travis: migrate Travis plans to run in a Docker container 1 day ago

clang-format tests/CMakelists.txt: Fix trying to run tests even if no python inter... 1 day ago

.dockernignore clang-format: run clang-format on C/C++ code 2 years ago

.gitignore travis: migrate Travis plans to run in a Docker container 1 day ago

.gitmodules .gitmodules: add public git url 2 years ago

.travis.yml travis: migrate Travis plans to run in a Docker container 1 day ago

CMakelists.txt mraa: Update to v1.7.0 20 days ago

CONTRIBUTING.md CONTRIBUTING.md: Add author rules and suggestion on filing issues for... 7 months ago

COPYING COPYING: Make it very clear this is an MIT license 1 year ago

This organization Search Pull requests Issues Marketplace Gist

Intel® IoT Developer Kit
IoT Libraries & Code Samples from Intel

Repositories People 48 Teams 17 Projects 0 Settings

Pinned repositories

mraa
Linux Library for low speed IO Communication in C with bindings for C++, Python, Node.js & Java. Supports generic io platforms, as well as Intel Edison, Intel Joule, Raspberry Pi and many more.
C 825 392

upm
UPM is a high level repository that provides software drivers for a wide variety of commonly used sensors and actuators. These software drivers interact with the underlying hardware platform through...
C++ 414 301

Search repositories... Type: All Language: All New

upm
UPM is a high level repository that provides software drivers for a wide variety of commonly used sensors and actuators. These software drivers interact with the underlying hardware platform through calls to MRAA APIs.
C++ 414 301 Updated 5 hours ago

mraa
Linux Library for low speed IO Communication in C with bindings for C++, Python, Node.js & Java. Supports generic io platforms, as well as Intel Edison, Intel Joule, Raspberry Pi and many more.
galileo intel-joule mraa libmraa C 825 392 Updated 6 hours ago

Top languages

C HTML JavaScript C++ Shell

Most used topics

mraa android-things upm

People 48 >

Avatar of a person with a yellow hat and a small robot icon.



This repository Search Pull requests Issues Marketplace Gist

Unwatch 71 ⭐ Unstar 414 Fork 301

Code Issues 32 Pull requests 8 Projects 1 Wiki Settings Insights

UPM is a high level repository that provides software drivers for a wide variety of commonly used sensors and actuators. These software drivers interact with the underlying hardware platform through calls to MRAA APIs.

Add topic

1,809 commits 8 branches 32 releases 50 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

Latest commit 3 days ago

matalkhah05 I2C Removing multiple address calls 29 days ago

cmake: fixed Nodejs detection on Ubuntu systems when using apt install

docs upm: added LIDAR-Lite V3 Optical Distance Measurement sensor library 13 days ago

doxy readme.cpp.mdt: updated example name for iotdk api pages 13 days ago

examples imx: Updated IMX sensor addressing 23 hours ago

include ANDROID: Treat utilities as its own library a month ago

src I2C: Removing multiple address calls 15 hours ago

tests SWIG_JAVA: C++ Interfaces to Java interfaces. Modified one Java Example 3 months ago

.clang-format .clang-format: Add a .clang-format file to UPM 2 years ago

gitignore gitignore: add wildcard so all build/* dirs are ignored 8 months ago

travis.yml travis.yml: use different swig ppa for Travis CI builds 6 months ago

CMakeLists.txt CMakeLists.txt: Hint to libmraa/mrajava 13 days ago

LICENSE LICENSE: Update license to be clearer and add 2016 copyright 6 months ago

README.md README.md: update MMA7660 snippet to reflect current reality 2 months ago

README.md

Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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



Intel Developer Zone - IoT

Jump start your IoT solution with the UP²⁺ Grove* IoT Development Kit.

[Take a Look](#)

DEVELOPMENT AND DEVELOPMENT RESOURCES

[Get Started](#) [Digging Deeper](#) [Going to Production](#)

Recent Updates



Intel® FPGAs
Intel® IoT Gateway Technology
Intel® Quark™ Microcontroller D2000
Intel® Quark™ SE Microcontroller C1000
Intel® System Studio 2018 Beta
Microsoft Azure™ (Cloud)
MinnowBoard Turbot™
Retail
Sensors
Smart Home
Smart Video
Terasic DE10-Nano Kit (FPGA)
UP Squared™ Grove* IoT Development Kit
Wind River Helix® Device Cloud

Build Innovative Solutions

Smart Home

Industries

Intel is powering the Internet of Things for global market opportunities and enabling change across industries.



Smart Video Industrial Automation Retail Smart Home

<https://software.intel.com/en-us/iot>

Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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

 upm Sensor framework for IoT developers

[Find](#) [Join](#) [Documentation](#)

Find your sensor.

Filter

Project Type

- Prototyping
- Industrial
- Medical
- Commercial
- Extended

Categories

- Pressure
- Color Sensor
- Humidity
- Temperature
- Accelerometer

[See more...](#)

Connections

- Gpio
- I2c
- Spi
- AIO
- PWM

[See more...](#)

API for the EMAX/Seeed ES08A Servo (es08a & EMAX 9g) [Learn More](#)



ES08A High Sensitive Mini Servo
Servos
Best for: prototyping & commercial

API for the Grove - LED Bar/My-semi MY9221 12-channel LED driver (my9221, Grove - LED Bar & MY9221 12-Channel LED Driver With Grayscale Adaptive Pulse Density Modulation Control) [Learn More](#)



Display
Best for: prototyping & commercial

API for the Grove Button (Grove Touch Sensor & Grove button) [Learn More](#)



Touch
Best for: prototyping

API for the Grove Circular LED module/My-semi MY9221 12-channel LED driver (my9221, Grove - Circular LED & MY9221 12-Channel LED Driver With Grayscale Adaptive Pulse Density Modulation Control) [Learn More](#)



<https://upm.mraa.io>



Compatibility and Other Tools

Multiple OS support



Multiple language support



Integrated Development Environments
(IDEs)

Build confidence

Easy to use,
fun to learn

Create and debug
full solutions



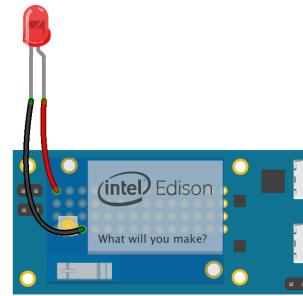
MRAA sample code for led blink

```
int
main(int argc, char** argv)
{
    if (argc < 2) {
        printf("Provide an int arg if you want to flash on something other than %d\n", DEFAULT_IOPIN);
        iopin = DEFAULT_IOPIN;
    } else {
        iopin = strtol(argv[1], NULL, 10);
    }

    signal(SIGINT, sig_handler);

    //! [Interesting]
    mraa::Gpio* gpio = new mraa::Gpio(iopin);
    if (gpio == NULL) {
        return mraa::ERROR_UNSPECIFIED;
    }
    mraa::Result response = gpio->dir(mraa::DIR_OUT);
    if (response != mraa::SUCCESS) {
        mraa::printError(response);
        return 1;
    }

    while (running == 0) {
        response = gpio->write(1);
        sleep(1);
        response = gpio->write(0);
        sleep(1);
    }
    delete gpio;
    return response;
    //! [Interesting]
}
```



```
"use strict";

const mraa = require('mraa'); //require mraa
console.log('MRAA Version: ' + mraa.getVersion()); //write the mraa version to the console

let myLed = new mraa.Gpio(13); //LED hooked up to digital pin 13 (or built in pin on Galileo Gen1 & Gen2)
myLed.dir(mraa.DIR_OUT); //set the gpio direction to output
let ledState = true; //Boolean to hold the state of Led

function periodicActivity() {
    myLed.write(ledState ? 1 : 0); //if ledState is true then write a '1' (high) otherwise write a '0' (low)
    ledState = !ledState; //invert the ledState
}

setInterval(periodicActivity, 1000); //call the periodicActivity function every second
```

Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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

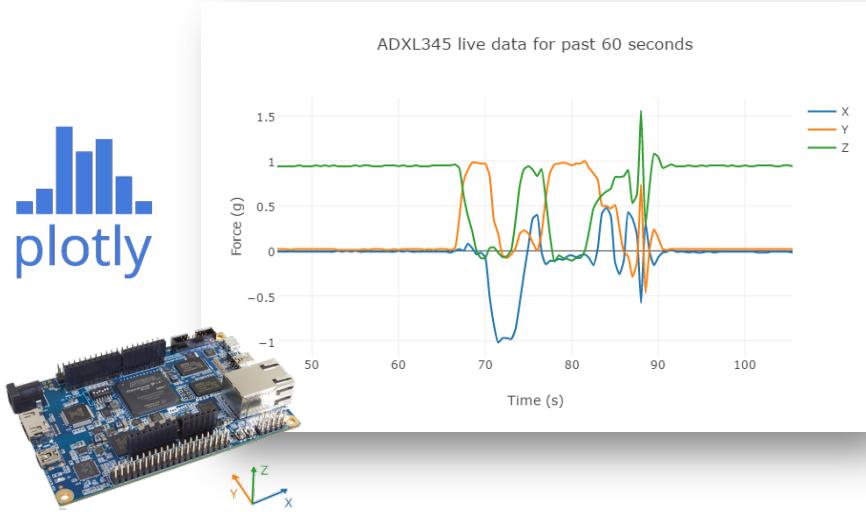


UPM sample code for ADXL345

```
int
main(int argc, char **argv)
{
//! [Interesting]
    int16_t *raw;
    float *acc;

// Note: Sensor only works at 3.3V on the Intel Edison with Arduino breakout
upm::Adxl345* accel = new upm::Adxl345(0);

while(true){
    accel->update(); // Update the data
    raw = accel->getRawValues(); // Read raw sensor data
    acc = accel->getAcceleration(); // Read acceleration (g)
    fprintf(stdout, "Current scale: 0x%2xg\n", accel->getScale());
    fprintf(stdout, "Raw: %6d %6d %6d\n", raw[0], raw[1], raw[2]);
    fprintf(stdout, "AccX: %5.2f g\n", acc[0]);
    fprintf(stdout, "AccY: %5.2f g\n", acc[1]);
    fprintf(stdout, "AccZ: %5.2f g\n", acc[2]);
    sleep(1);
}
//! [Interesting]
return 0;
}
```



```
// Load accelerometer
var adxl345 = require('jsupm_adxl345');

// Instantiate on I2C bus
var adxl = new adxl345.Adxl345(0);

setInterval(function()
{
    adxl.update(); // Update the data
    var raw = adxl.getRawValues(); // Read raw sensor data
    var force = adxl.getAcceleration(); // Read acceleration force (g)
    var rawvalues = raw.getItem(0) + " " + raw.getItem(1) + " " + raw.getItem(2);
    console.log("Raw Values: " + rawvalues);
    console.log("ForceX: " + force.getItem(0).toFixed(2) + " g");
    console.log("ForceY: " + force.getItem(1).toFixed(2) + " g");
    console.log("ForceZ: " + force.getItem(2).toFixed(2) + " g");
}, 1000);
```



Optimization Notice

Copyright © 2016, Intel Corporation. All rights reserved.

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



THANK YOU
Questions?

