

Overview of the Intel® Internet of Things Developer Kit

A decorative graphic consisting of several horizontal blue lines of varying thickness, with small circles at the ends, resembling a circuit board or data lines.















Daniel Holmlund – Software Engineer, IoT Developer Program, Intel Corporation

Stewart Christie – Community Manager, IoT Developer Program, Intel Corporation

Agenda

- Development Kits Introduction
- Intel® Hardware for Internet of Things (IoT)
- Intel Developer Program for IoT
- Software for IoT
- Demos

Intel Development Kits for Internet of Things

	Intel® Development Kit for IoT software.intel.com/IoT		Intel® Gateway Solutions Development Kits				3rd Party ODM Dev Kits
	Galileo	Edison	DK 50 Series	DK 100 Series	DK 200 Series	DK 300 Series	
Compute	Intel® Quark™ SoC X1000	Intel® Atom™ SoC Tangiers	Intel Quark SoC X1000	Intel Quark SoC X1020D	Intel Quark SoC X1020D	Intel Atom Processor E3826	IBM® Bluemix AT&T®
Kit Contents	Board, Cables, Starter kit, power supply		Board and Power Supply Only	Board, radio(s), chassis, power supply	Board, radio(s), chassis, power supply	Board, radio(s), chassis, power supply	      
Cloud	<ul style="list-style-type: none"> • IOT Cloud Agent: IoT Analytics Platform as a Service • Mashery API Network 		Enabled with third party Cloud solutions				 
Software Components:	<ul style="list-style-type: none"> • Arduino • JavaScript* • WRS VxWorks* • Intel System Studio 	Visual prog* C++ Python* Sensor & Actuator lib	Wind River Linux* OS, Intelligent Device Platform XT, Wind River Workbench Tools, McAfee Embedded Control				    
Software License	Mix of Open source & non-production licenses		Non-production, 6 Month Term License w/ support from Intel	Commercial license			
Support	Community	Community	Intel Application Design Center	Intel Application Design Center	Intel Application Design Center	Intel Application Design Center	
Cost	Free at Roadshows; \$100 online	Free at Roadshows \$120 online	\$200	\$3000	\$3000	\$3000?	
Target Markets	Innovation vehicle	Innovation vehicle	Innovation vehicle	Industrial & Energy	Transportation	Multi Vertical	
Availability	Beta Launch	Beta Launch	Launched	Launched	Launched	WW35-38	

GIGABYTE™

iEi®

SUPERMICRO

avalue Technology Inc.

bcm

EUROTECH

kontron

ECS ELITEGROUP

ADVANTECH

Enabling an Intelligent Planet

ADI ENGINEERING

congatec

NEXCOM

Portwell

ADLINK TECHNOLOGY INC.

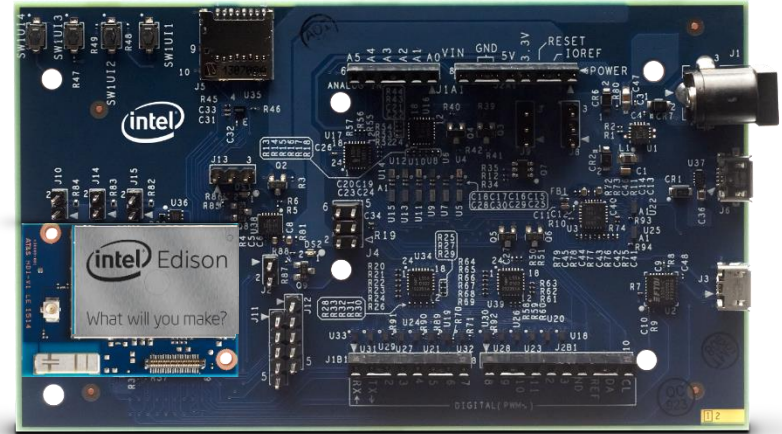
Intel® Hardware for the Internet of Things

IDF14

Intel® Edison – Arduino* Development Board

Board I/O: Compatible with Arduino* Uno (except only 4 PWM instead of 6 PWM)

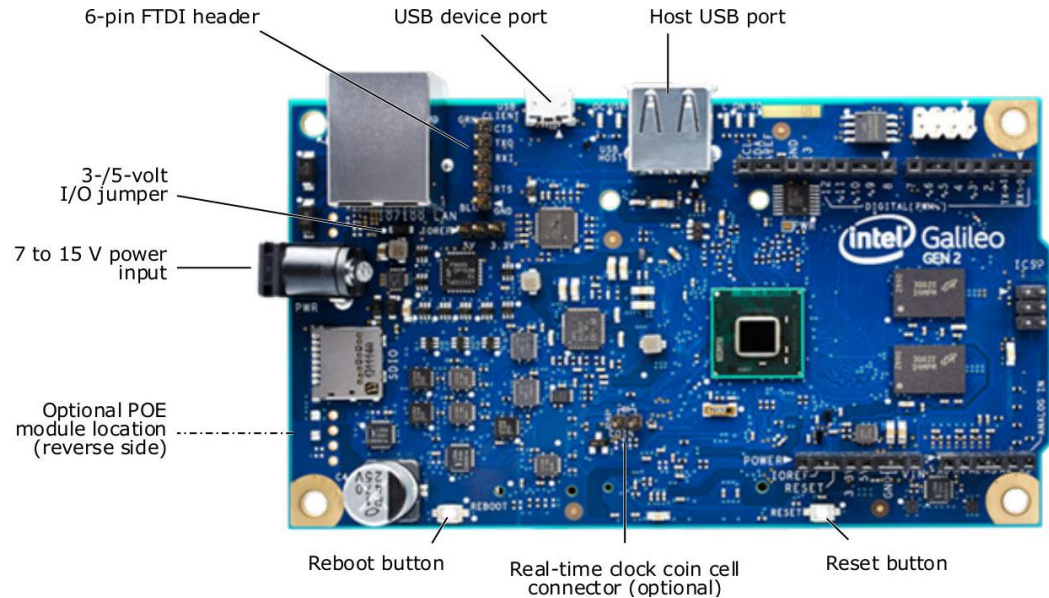
- 20 digital input/output pins including 4 pins as PWM outputs
- 6 analog inputs
- 1 UART (RX/TX)
- 1 I2C
- 1 ICSP 6-pin header (SPI)
- Micro USB device connector OR (via mechanical switch) dedicated standard size USB host Type-A connector
- Micro USB device (connected to UART)
- SD Card connector
- DC power jack (7V – 15V DC input)



Intel® Galileo Development Board – Gen 2

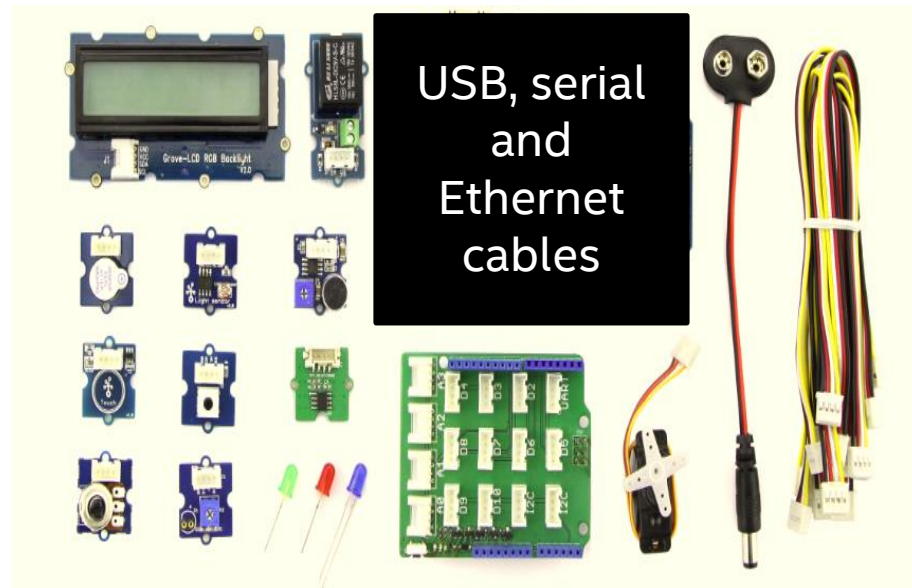
Board I/O:

- Mechanically compatible with Arduino* Uno
- 20 digital input/output pins including 6 pins as PWM outputs
- 6 analog inputs
- 2 UART (RX/TX)
- 1 I2C
- 1 ICSP 6-pin header (SPI)
- USB device connector (Host)
- Micro USB device connector (client)
- SD Card connector
- DC power jack (7V – 15V DC input)



Grove Starter Kit Plus - IoT Version

- | | |
|----|-----------------------------------|
| 1 | Base Shield v2 |
| 2 | Grove - Buzzer V1.1 |
| 3 | Grove - Button |
| 4 | Grove-LED v1.3 |
| 5 | Grove - Sound Sensor_V1.2 |
| 6 | Grove - Rotary Angle Sensor |
| 7 | Grove-Touch Sensor |
| 8 | Grove - Smart Relay |
| 9 | Grove-Light Sensor |
| 10 | Grove - Temperature Sensor_V1.1 |
| 11 | 26AWG Grove Cable |
| 12 | Mini Servo |
| 13 | 9V to Barrel Jack Adapter - 126mm |
| 14 | DIP LED Blue-Blue |
| 15 | DIP LED Green-Green |
| 16 | DIP LED Red-Red |
| 11 | Grove - LCD RGB Backlight |



Development Kit Hardware Bill of Materials

Starter IoT Development Kit to be given away for all roadshow attendees

Arduino Board

- 2nd Generation Intel® Galileo Board
- Intel® Edison Board
- Power Supply Unit



Essential Components

- Black Ethernet Cable
- Micro USB Cable
- 8GB Micro SD Memory Card
- USB to 6 –pin FTDI Serial Cable



Sensor + Actuators

- Grove Starter Kit Plus
 - Intel® IoT version

Roadshows are two day events located at a hackerspace with access to prototyping tools and materials.

Extra Roadshow items

- PCI Express* Wi-Fi* + Bluetooth® cards
- Air Quality, Compass, Gyro and Acceleration sensors
- Extra servos, and actuators
- Cables and connectors

Intel® Developer Program for Internet of Things

Intel® Developer Program for Internet of Things

A comprehensive developer program for hobbyists, students and entrepreneurial developers with outreach, training and tools required to rapidly develop, test and deploy applications for the Internet of Things (IoT).

Program includes:

- Developer kit based on Intel® Galileo and Edison Technology with package of easy to use hardware, software tools, libraries and cloud services
- 10 City IoT Roadshow distributing 1,000 kits
- On-line community for learning, building, sharing

Join the community today at software.intel.com/loT



Get the Intel IoT Developer Kit (Beta)

Boards, sensors and software tools to get your dev environments ready for coding.

[Get the Hardware](#)
[Download the Software](#)



Get up and Running

Dive into starter guides and sample projects to help get your development environment up and talking to your hardware.

[Getting Started Guides](#)
[Code Samples for Sensors and More](#)



Get Help

Need help with code or hardware? Have a comment? Just want to see what everyone is talking about?

[IoT Forum](#)
[Browse our Library](#)

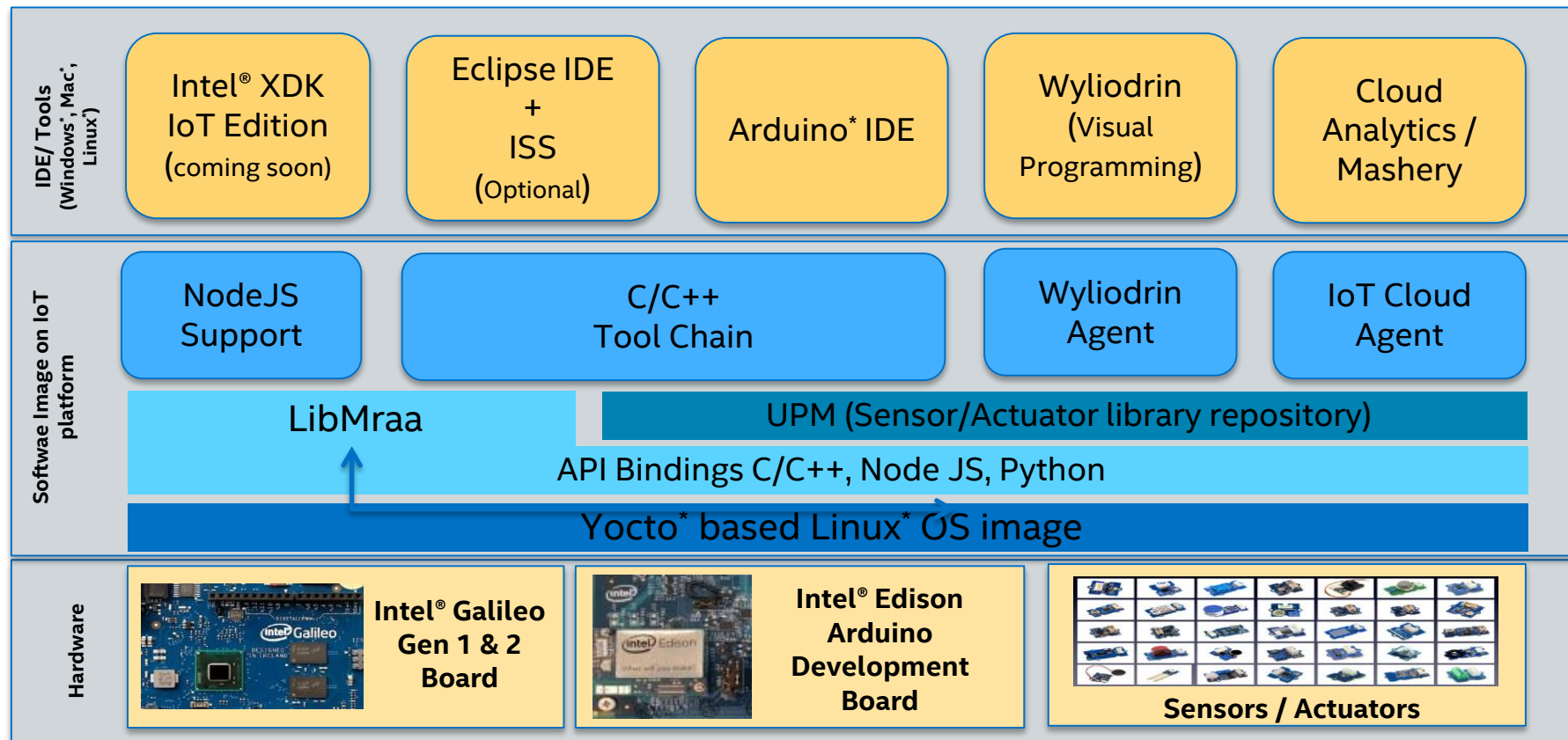
Get Inspired



Target Audience and Developer Path

	Arduino* Developer	Visual Programming (Galileo Only)	JavaScript* Developer	C / C++ Developer	Wind River (Galileo only)
Target Audience	Maker	Beginner	Intermediate	Advanced	Advanced Professional
OS / Boot Image	Yocto* Linux SPI	Yocto Linux	Yocto Linux	Yocto Linux	VxWorks*
IDE Dev Env	Arduino IDE Windows*/ Mac*/ Linux*	Wylodrin Web	XDK Win/ Mac/ Linux	Eclipse Win/ Mac / Linux	WR Eclipse Win / Linux
Programming Language	Arduino Sketch C++	Visual Python	JavaScript (Node JS)	C/ C++	C / C++
Tools/ Libraries	Arduino Libraries	Wylodrin	XDK	ISS	Work Bench / ISS
Cloud	IoT Cloud Analytics	IoT Cloud Analytics Widget	IoT Cloud Analytics Mashery/ 3 rd party	IoT Cloud Analytics Mashery/ 3 rd Party	WR Cloud

Intel® Developer Kit for IoT Components

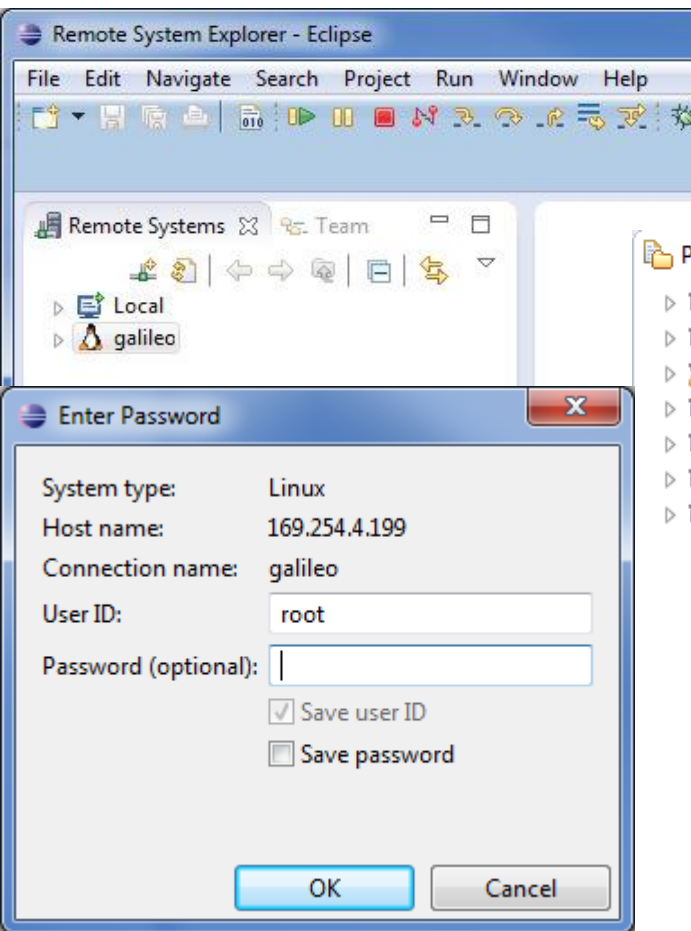


Intel® Developer Kit for IoT Components

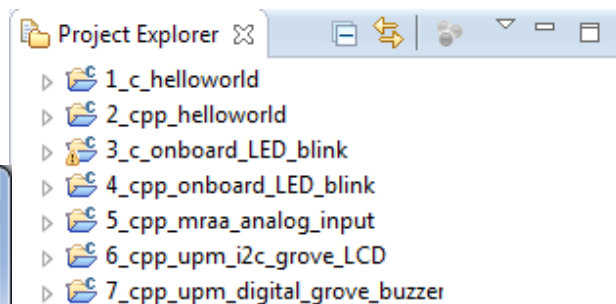
A complete solution for creating IoT solutions targeted for Intel® platforms such as Intel® Galileo board and Intel® Edison board.

- Multiple IDEs (Intel® XDK, Eclipse, Wylidrin, Arduino*)
 - Cloud analytics & data management
 - Mashery IoT Restful APIs
- Multiple Programming Languages (JavaScript*, C/C++, Arduino Sketches, Visual programming)
- APIs that shields hardware complexity (LibMraa)
 - Sensor libraries with API bindings (UPM)
- Open & Standard – Yocto* Linux* based OS
 - Supports full range of Linux tools and libraries
- Full x86 support (Scale from Intel® Quark™ to Intel® Core™)
 - Intel Galileo (Gen1/ Gen 2), Intel Edison

Developer Kit C++ Tools (beta)



- One IDE for all three supported boards
- Ready to run sample code

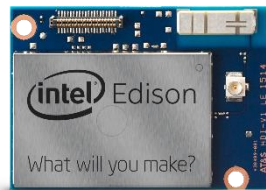
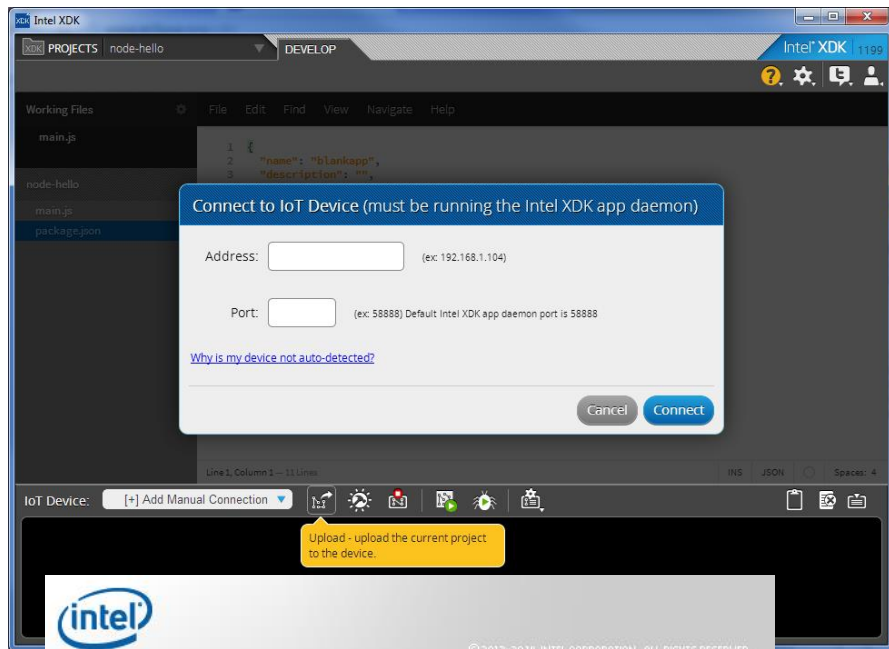


- SSH Connection to developer board for remote GDB server
- Eclipse tools also work for Arduino* Sketch code build and debug

Developer Kit JavaScript* Tools (coming soon)

JavaScript* I/O Libraries for Intel® Galileo technology

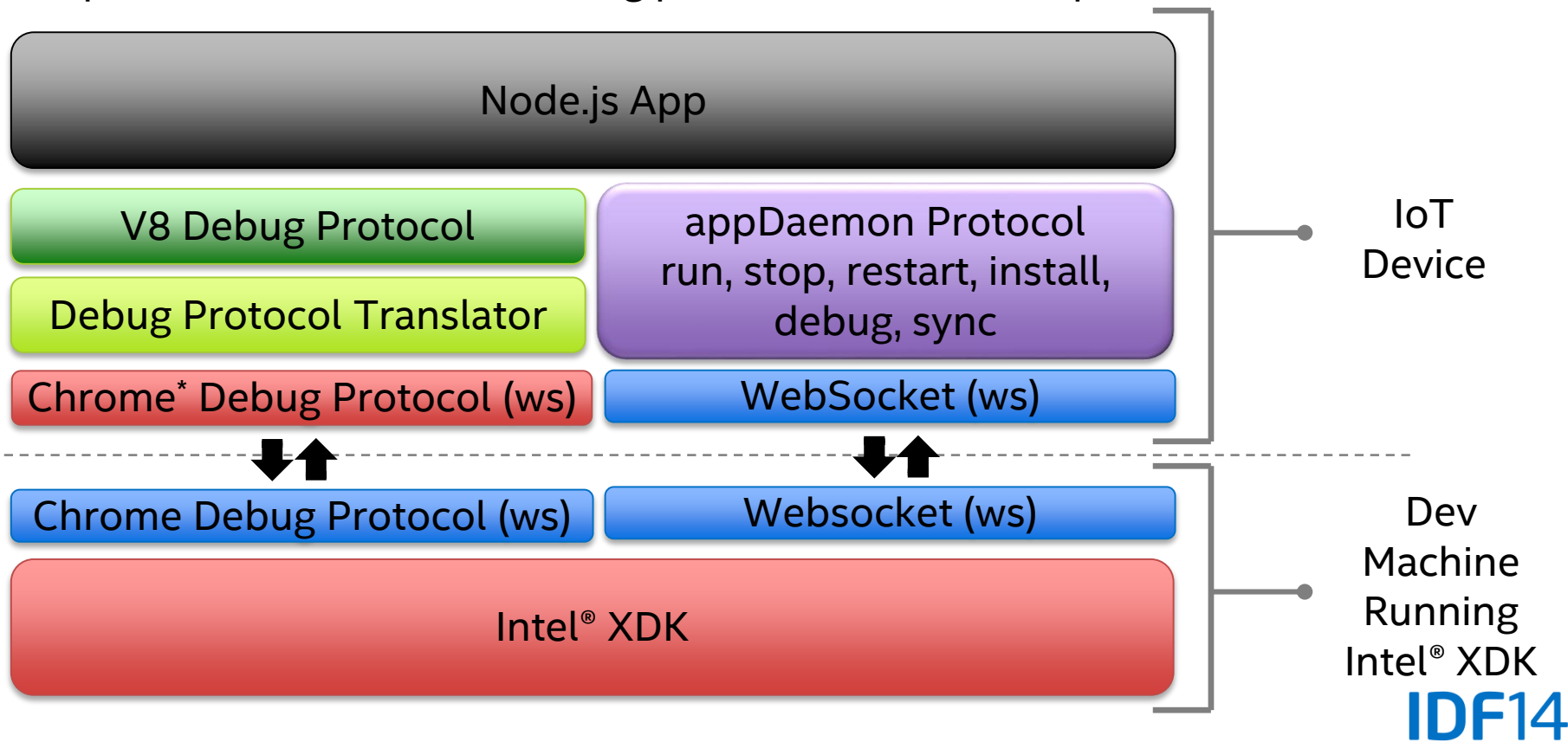
- **mraa** – open source library for Intel® Galileo/Edison technologies used API similar to the *MBED api (Intel® IoT Devkit)
- **io-js** – open source library for Intel Galileo/Edison technologies used API similar to the Arduino* API (Intel Labs China)
- **johnny-five** – Open source I/O library with plugins for various boards including Intel Galileo (Gen2 demo upstairs)
- **on/off** – open source i/o library that can be used to manipulate 'raw' GPIO/PWM via sysfs interface
- **i2c** – node.js i2c library (uses sysfs)



coming soon.

Intel® XDK IoT Device Daemon Overview

Separate web-sockets for debug protocol and daemon protocol



Developer Kit Python* Tools

UNLOCK THE FULL FEATURES OF 



WITH  Wylidrin
IDEAS IN REAL TIME

Visual
Programming



python

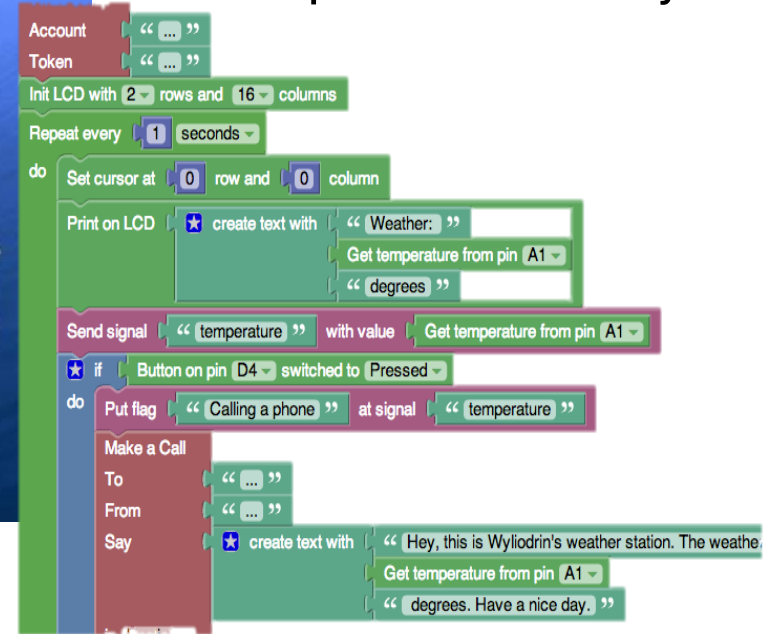


C++



ARDUINO

GUI based online Python*
development from Wylidrin



Wylidrin agent pre-installed in lot
Development Kits SD image.

(Intel® Galileo technology only for now)

Program Overview

The Intel Developer Program for the Internet of Things

A comprehensive program for hobbyists, students, and entrepreneurial developers that delivers everything a developer needs to quickly turn ideas into projects for the Internet of Things

The program includes:

Intel® Development Kit for IoT - The first of several Intel-based IoT development kits, create products using C/C++ and JavaScript*. (beta available now)

Cloud Infrastructure - End-to-end solutions for makers and developers prototyping, experimenting and testing new IoT solutions.

IoT Developer Zone - A new on-line software community to explore, build and share IoT solutions. Join the community at software.intel.com/iot.

Academic Program - Intel is working with top universities and their professors to enable curriculum and inspire students to develop innovative apps on Intel's Internet of Things technologies.

10 City Roadshow + more - A worldwide series of local hackathons and meet-ups that will bring developers together to showcase and share ideas. Intel IoT Development Kits will be provided to participating developers.

Software for IoT

Overview of “libmraa” and “libupm”

Purpose of Each Library

Libmraa

Defines the capabilities of GPIO pins,
Autodetection of Gen 1, Gen 2 and
Edison DVT

```
/**< Pin Valid */  
/**< General Purpose IO */  
/**< Pulse Width Modulation */  
/**< Faster GPIO */  
/**< SPI */  
/**< I2C */  
/**< Analog in */
```

Libupm

Defines the interaction of high level
sensors.

Buttons, Buzzers, LCD, LED, Light
Sensor, Servos, Rotary Angle Sensor,
Relays, Sound Sensors, temperature
sensors ... more are added continually.

Identifying Your Platform

```
/**
 * MRAA supported platform types
 */
typedef enum {
    MRAA_INTEL_GALILEO_GEN1 = 0, /**< The Generation 1 Galileo platform (RevD) */
    MRAA_INTEL_GALILEO_GEN2 = 1, /**< The Generation 2 Galileo platform (RevG/H) */

    MRAA_UNKNOWN_PLATFORM = 99 /**< An unknown platform type, typically will load INTEL_GALILEO_GEN1 */
} mraa_platform_t;

/**
 * Get platform type, board must be initialised.
 *
 * @return mraa_platform_t Platform type enum
 */
mraa_platform_t mraa_get_platform_type();
```

Identifying your target platform
allow platform specific action to
occur.

Platform Specific Actions

```
mraa_platform_t platform = mraa_get_platform_type();
mraa_gpio_context gpio;
char board_name[] = "Some weird devboard that isn't recognised...";
int ledstate = 0;

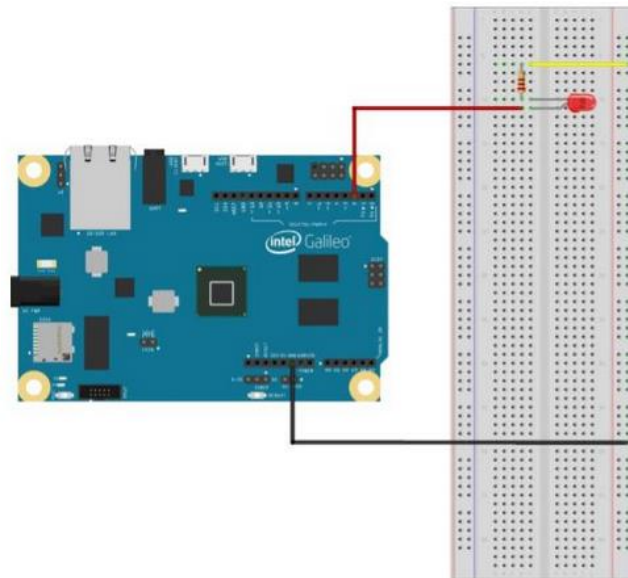
switch (platform) {
    case MRAA_INTEL_GALILEO_GEN1:
        strcpy(board_name, "Intel Galileo Gen1");
        gpio = mraa_gpio_init_raw(3);
        break;
    case MRAA_INTEL_GALILEO_GEN2:
        strcpy(board_name, "Intel Galileo Gen2");
        gpio = mraa_gpio_init(13);
        break ;
    default:
        gpio = mraa_gpio_init(13);
}
```

Using libmraa for the Blinky Light Demo

```
#include <stdbool.h>
#include "mraa.h"

int main() {
    mraa_gpio_context gpio; //create a GPIO structure/context
    gpio = mraa_gpio_init(2); // Initialize pin 2
    mraa_gpio_dir(gpio, MRAA_GPIO_OUT); // Set direction to OUTPUT
    bool ledState=true; //To store the LED state, starting true (HIGH)

    while(true){
        mraa_gpio_write(gpio, ledState?1:0); // Writes into GPIO
        //accordingly to the state
        // true = 1, false = 0
        ledState=!ledState; // changes state by negating
        sleep(1); //waits one second
    }
}
```

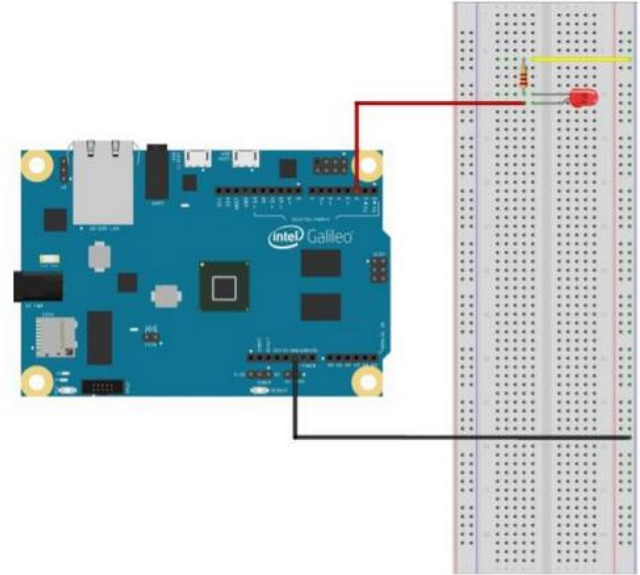


Using libupm for the Blinky Light Demo

```
#include <unistd.h>
#include <iostream>
#include "grove.h"

int
main(int argc, char **argv)
{
    ///! [Interesting]
    upm::GroveLed* led = new upm::GroveLed(2);
    std::cout << led->name() << std::endl;
    for (int i=0; i < 10; i++) {
        led->on();
        sleep(1);
        led->off();
        sleep(1);
    }
    ///! [Interesting]

    return 0;
}
```



Libmraa Error Codes...

```
/**
 * MRAA return codes
 */
typedef enum {
    MRAA_SUCCESS
    MRAA_ERROR_FEATURE_NOT_IMPLEMENTED
    MRAA_ERROR_FEATURE_NOT_SUPPORTED
    MRAA_ERROR_INVALID_VERBOSITY_LEVEL
    MRAA_ERROR_INVALID_PARAMETER
    MRAA_ERROR_INVALID_HANDLE
    MRAA_ERROR_NO_RESOURCES
    MRAA_ERROR_INVALID_RESOURCE
    MRAA_ERROR_INVALID_QUEUE_TYPE
    MRAA_ERROR_NO_DATA_AVAILABLE
    MRAA_ERROR_INVALID_PLATFORM
    MRAA_ERROR_PLATFORM_NOT_INITIALISED
    MRAA_ERROR_PLATFORM_ALREADY_INITIALISED

    MRAA_ERROR_UNSPECIFIED
} mraa_result_t;

= 0, /**< Expected response */
= 1, /**< Feature TODO */
= 2, /**< Feature not supported by HW */
= 3, /**< Verbosity level wrong */
= 4, /**< Parameter invalid */
= 5, /**< Handle invalid */
= 6, /**< No resource of that type avail */
= 7, /**< Resource invalid */
= 8, /**< Queue type incorrect */
= 9, /**< No data available */
= 10, /**< Platform not recognised */
= 11, /**< Board information not initialised */
= 12, /**< Board is already initialised */

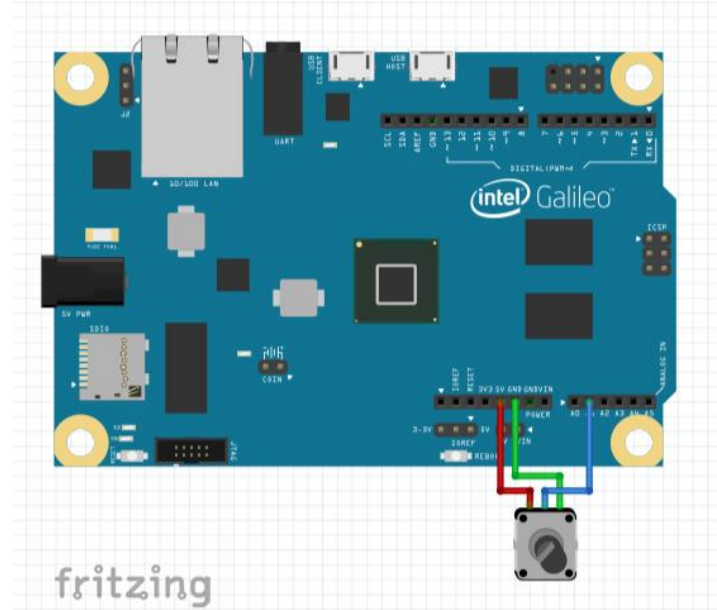
= 99 /**< Unknown Error */
```

Read a Potentiometer

```
#include "mraa.h"

int main ()
{
    uint16_t adc_value = 0;
    mraa_aio_context a1;
    a1 = mraa_aio_init(1);

    for(;;) {
        adc_value = mraa_aio_read(a1);
        fprintf(stdout, "Input A1 %X - %d\n",
            adc_value, adc_value);
    }
    mraa_aio_close(a1);
    return MRAA_SUCCESS;
}
```



Write to an LCD

```
#include "jhd1313m1.h"

int
main(int argc, char **argv)
{
    // 0x62 RGB_ADDRESS, 0x3E LCD_ADDRESS
    upm::Jhd1313m1 *lcd = new upm::Jhd1313m1(0, 0x3E, 0x62);
    lcd->setCursor(0,0);
    lcd->write("Hello World");
    lcd->setCursor(1,2);
    lcd->write("Hello World");
    lcd->close();
}
```



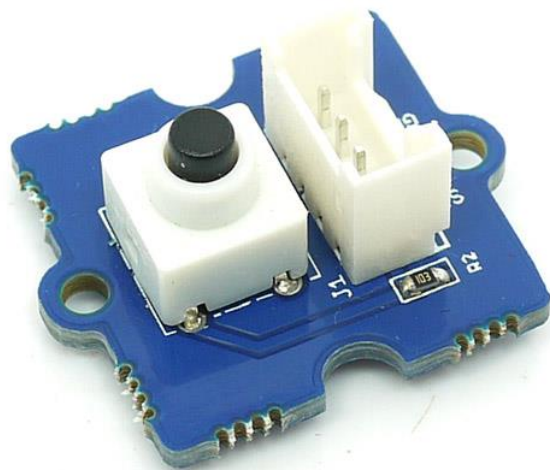
Buttons

```
#include "stdio.h"
#include "mraa.hpp"

void main() {
    //create GPIO and initiate it with pin 8
    mraa::Gpio gpio = new mraa::Gpio(8);

    //set direction to input
    gpio.dir(mraa::DIR_IN);

    //set a loop to read the input
    while(1) {
        fprintf(stdout, "Gpio is %d\n", gpio.read());
        sleep(1);
    }
    delete gpio;
}
```



Mini Servo

```
#include <unistd.h>
#include <iostream>
#include "es08a.h"
#include <signal.h>
#include <stdlib.h>

int
main(int argc, char **argv)
{
    /// [Interesting]
    upm::ES08A *servo = new upm::ES08A(5);
    servo->setAngle (180);
    /// [Interesting]

    servo->setAngle (90);
    servo->setAngle (0);
    servo->setAngle (90);
    servo->setAngle (180);

    std::cout << "exiting application" << std::endl;

    delete servo;

    return 0;
}
```



Temperature Sensor

```
#include <unistd.h>
#include <iostream>
#include "grove.h"

int
main(int argc, char **argv)
{
    ///! [Interesting]
    upm::GroveTemp* s = new upm::GroveTemp(0);
    std::cout << s->name() << std::endl;
    for (int i=0; i < 10; i++) {
        std::cout << s->value() << std::endl;
        sleep(1);
    }
    ///! [Interesting]

    return 0;
}
```



Demo C++ Development

Demo Intel® XDK

Final Reminders

- Come to an Internet of Things Roadshow
- Download the software pieces you need, buy a sensor kit
- Build a project, publish on Instructables
- Learn more at the following IDF classes, and the Makerspace zone on 3rd floor
- Learn more at the Intel Developer Zone at <http://software.intel.com/iot>

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>

Intel, Core, Atom, Quark, Look Inside and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Intel is under license.

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

Copyright ©2014 Intel Corporation.

Legal Disclaimer

Software Source Code Disclaimer: Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the second quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “believes,” “seeks,” “estimates,” “may,” “will,” “should” and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel’s actual results, and variances from Intel’s current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be important factors that could cause actual results to differ materially from the company’s expectations. Demand for Intel’s products is highly variable and, in recent years, Intel has experienced declining orders in the traditional PC market segment. Demand could be different from Intel’s expectations due to factors including changes in business and economic conditions; consumer confidence or income levels; customer acceptance of Intel’s and competitors’ products; competitive and pricing pressures, including actions taken by competitors; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel operates in highly competitive industries and its operations have high costs that are either fixed or difficult to reduce in the short term. Intel’s gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; and product manufacturing quality/yields. Variations in gross margin may also be caused by the timing of Intel product introductions and related expenses, including marketing expenses, and Intel’s ability to respond quickly to technological developments and to introduce new products or incorporate new features into existing products, which may result in restructuring and asset impairment charges. Intel’s results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Intel’s results could be affected by the timing of closing of acquisitions, divestitures and other significant transactions. Intel’s results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues, such as the litigation and regulatory matters described in Intel’s SEC filings. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel’s ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel’s results is included in Intel’s SEC filings, including the company’s most recent reports on Form 10-Q, Form 10-K and earnings release.

Backup

IoT Devkit Comparison

	Intel® Edison Unified OS Image	Intel® Galileo Gen 1/Gen 2 Independent OS Image
OS/ Image	✓ Software package for Edison	✓ EGLibC OS Image (Yocto* 1.6)
LibMraa/ UPM	✓ Access to low-level I/O + Sensor libs -Included in Edison Software stack	✓ Access to low-level I/O + Sensor libs
C/C++ (Eclipse)	✓ Eclipse IDE (64-bit) for C/C++ dev on Windows*, Linux*, Mac*	✓ Eclipse IDE (64-bit) for C/C++ dev on Win, Linux, Mac
JavaScript* (XDK)	✓ Node JS Support. Included in Edison Software stack. Coming Soon (Sept 30 th)	✓ Node JS Support Coming Soon (Sept 30 th)
Visual (Wylodrin)	✗ Future	✓ Wylodrin Component
Arduino*	✓ Included in Edison Software stack	✓ Multi-lib support for Arduino
VxWorks*	✗ Future (TBD)	✓ VxWorks + Work bench
Intel System Studio	✓ Advanced optimization tools for IoT	✓ Advanced optimization tools for IoT
IoT Cloud Analytics	✓ IoT Cloud Analytics component - Included in Edison S/W stack	✓ IoT Cloud Analytics component