



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Romeo for Edison Controller SKU: DFR0331

Contents

- 1 Introduction
- 2 Specification
- 3 Application
- 4 Pinout Diagram
- 5 Tutorial
 - 5.1 Romeo for Edison Controller Quick Start
 - 5.1.1 A.The Construction of the Hardware Environment
 - 5.1.1.1 Hardware Connection Diagram
 - 5.1.2 B.The Construction of the Software Environment
 - 5.1.2.1 Operating System
 - 5.1.2.2 Compile Environment
 - 5.2 Romeo for Edison Controller Typical Applications
 - 5.2.1 Digital Application
 - 5.2.1.1 A.LED Blink
 - 5.2.1.2 B.Button Test
 - 5.2.1.3 C.Servo Control
 - 5.2.1.4 D.Motor Control
 - 5.2.2 Analog Application
 - 5.2.2.1 A.Analog Read
- 6 Atmega8 Firmware Upgrade
- 7 FAQ
- 8 Development Tools



(/wiki/index.php/File:DFR0331_PD.JPG)

Romeo for Edison Controller

Introduction

Romeo for Intel Edison is a multi-purpose, all-in-one development platform based on Intel Edison and Arduino SoC. It is especially designed to be useful for robotics applications. Romeo for Intel Edison is compatible with Arduino open source platform and Linux, and supports Java and C development environment. Users are able to extend this platform with thousands of existing shields and modules such as switches, sensors, LEDs, servos, motors easily with Romeo for Intel Edison. It can also be used as a standalone communication platform for software like flash, processing, Max/MSP and VVVV. The integrated 2 way DC motor driver and wireless capability allows you to start your project immediately without the need of any additional motor driver or wireless shield. The Romeo for Intel Edison inherits all functions of the Romeo all-in-one controller by integrating powerful functions the Intel Edison board possesses. You can describe it as a control board specially designed for robotics applications, carrying the powerful gene of Intel Edison and being compatible with Arduino.

NOTE: According to the Arduino IDE version update and Intel update, the wiki might be out of date, you could visit FAQ (https://www.dfrobot.com/wiki/index.php?title=Romeo_for_Edison_Controller_SKU:_DFR0350#FAQ) for proper solution.

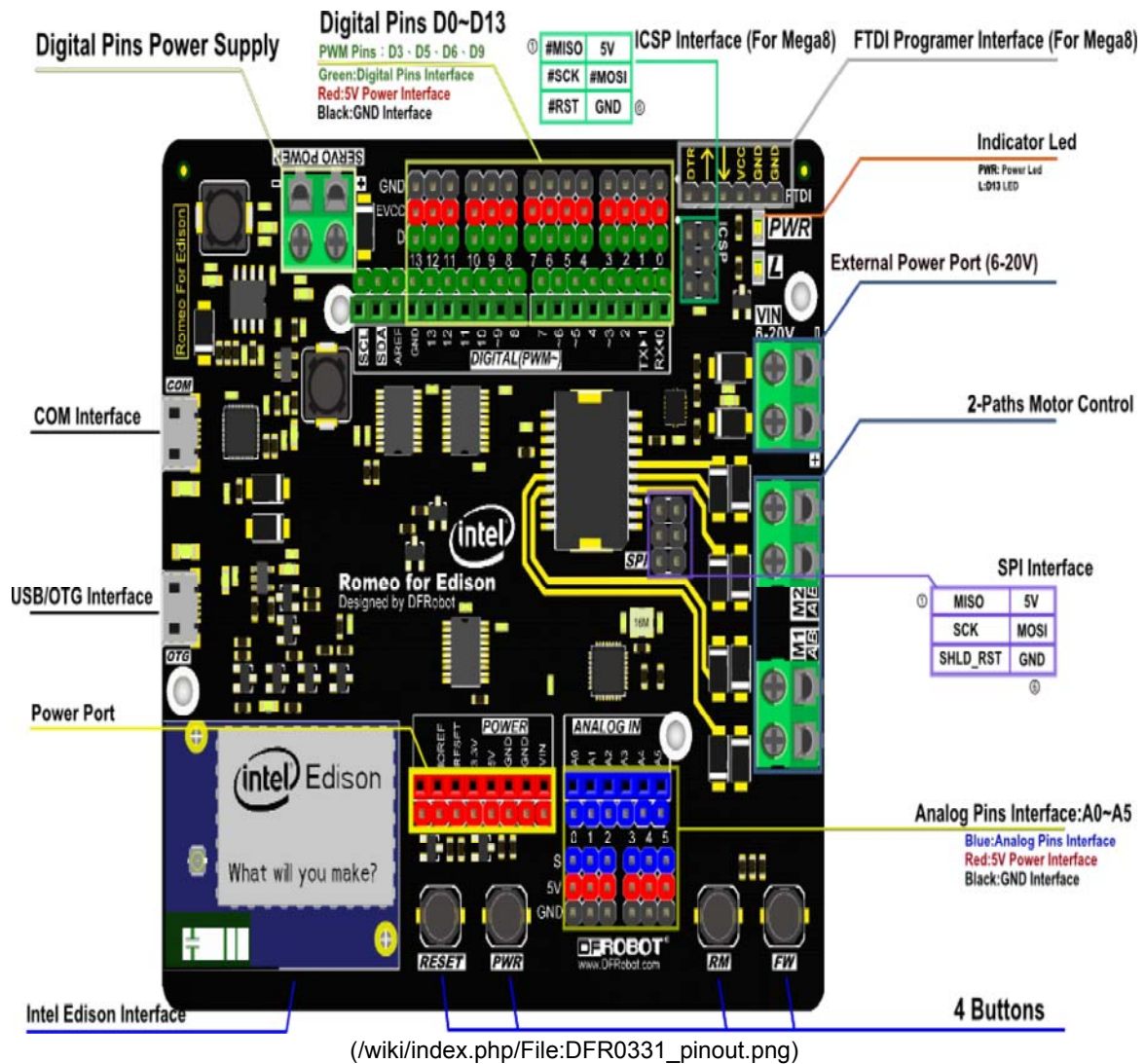
Specification

*Operating Voltage	:5V
*Output Voltage	:5V/3.3V
*Input Voltage(limits)	:6-20V
*Digital I/O pins	:14
*Analog I/O pins	:6
*DC Current per I/O Pin	:10mA
*Motor Driver Constant Current	:2 x 2A
*Size	:100x88x15mm

Application

- Digital I/O pins :D0-D13, A0-A5
- PWM Output interface :4 (Digital pins D3, D5, D6 and D9)
- Support USB power and external power supply switch
- Support SPI program port
- Support 1 x I2C interface
- Support 2 way motor drive (Peak Current: 2A)
- Support Broadcom 43340 802.11 a/b/g/n dual band (2.4G and 5GHz) WiFi
- Support Bluetooth 4.0

Pinout Diagram



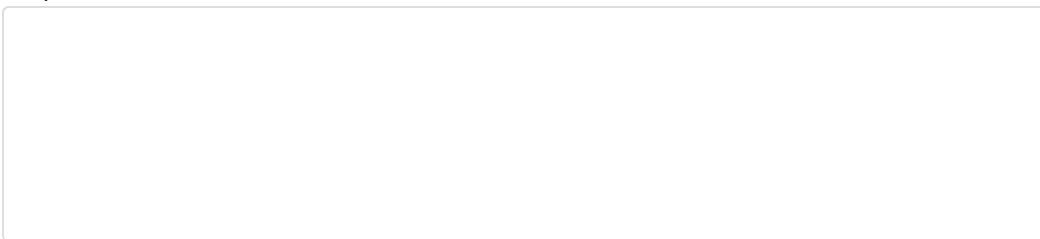
Tutorial

Romeo for Edison Controller Quick Start

A. The Construction of the Hardware Environment

Part List:

- Romeo for Edison x1
- Edison module x1
- micro USB x1
- Computer x1

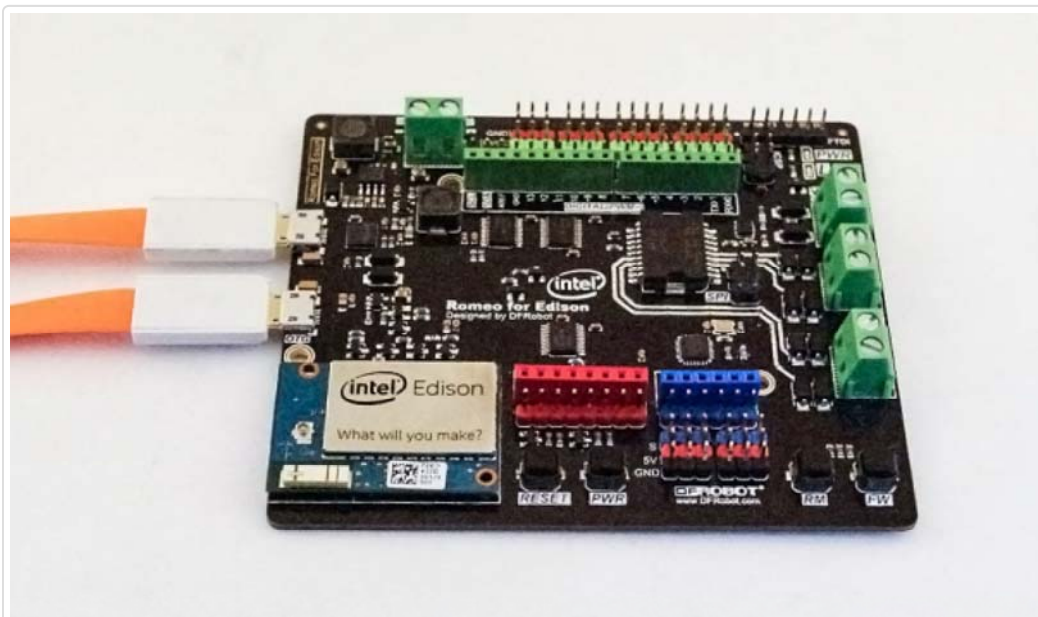




(/wiki/index.php/File:DFR0331_Part_List1.png)

Part List

Hardware Connection Diagram



(/wiki/index.php/File:DFR0331_PC_connection1.png)

Hardware Connection Diagram

B. The Construction of the Software Environment

Operating System

Windows 8.1

Compile Environment

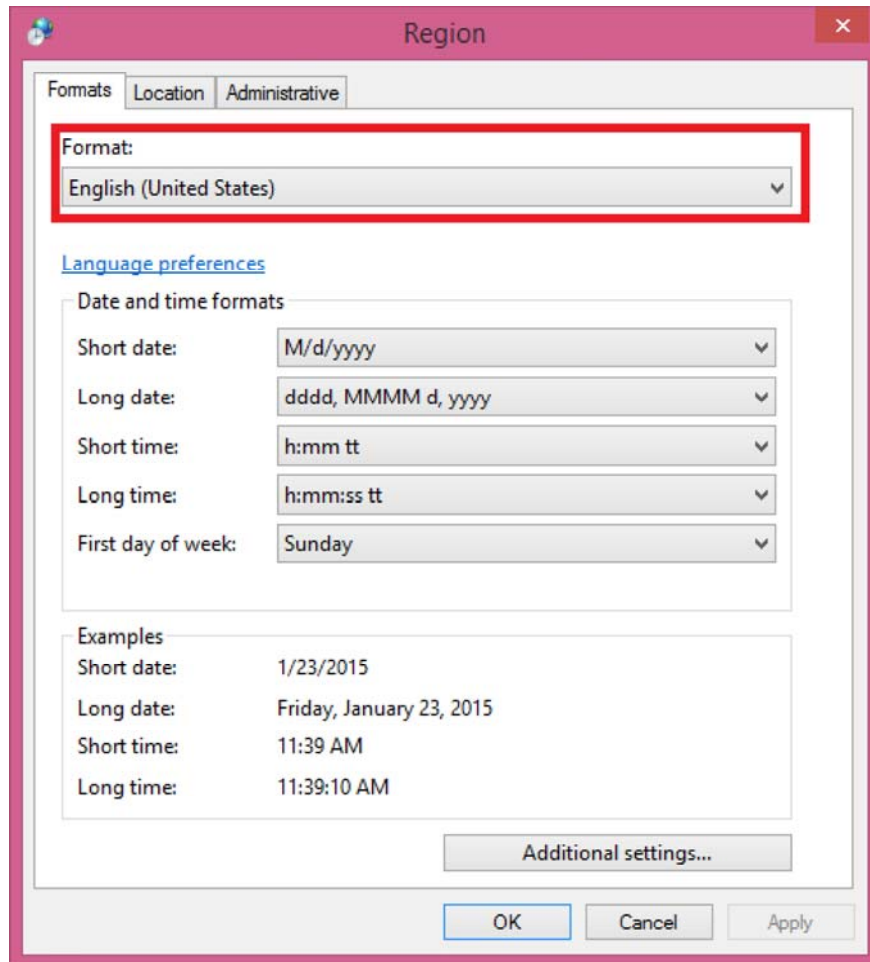
A. Edison Arduino IDE

Download Edison Arduino IDE (Windows version) (<https://communities.intel.com/docs/DOC-23242>)

Extract to your local folder.

NOTE: There is a bug with Edison Arduion IDE, you have to change your PC location to U.S to avoid IDE FC.

Control Panel-->Change data, time or number formats-->Format-->English(United States)



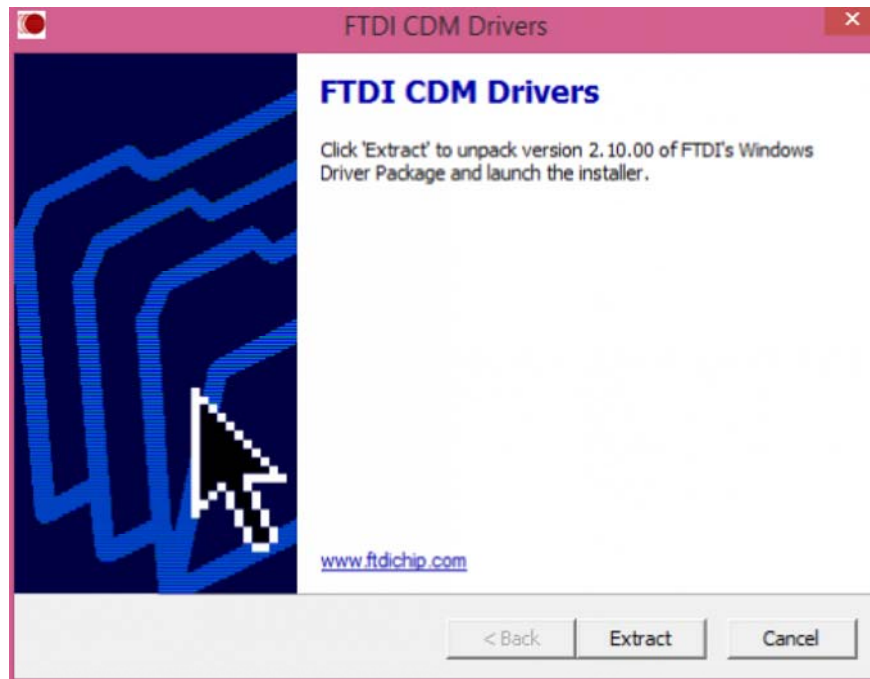
(/wiki/index.php/File:DFR0331_Postition.png)

Open Edison Arduion IDE

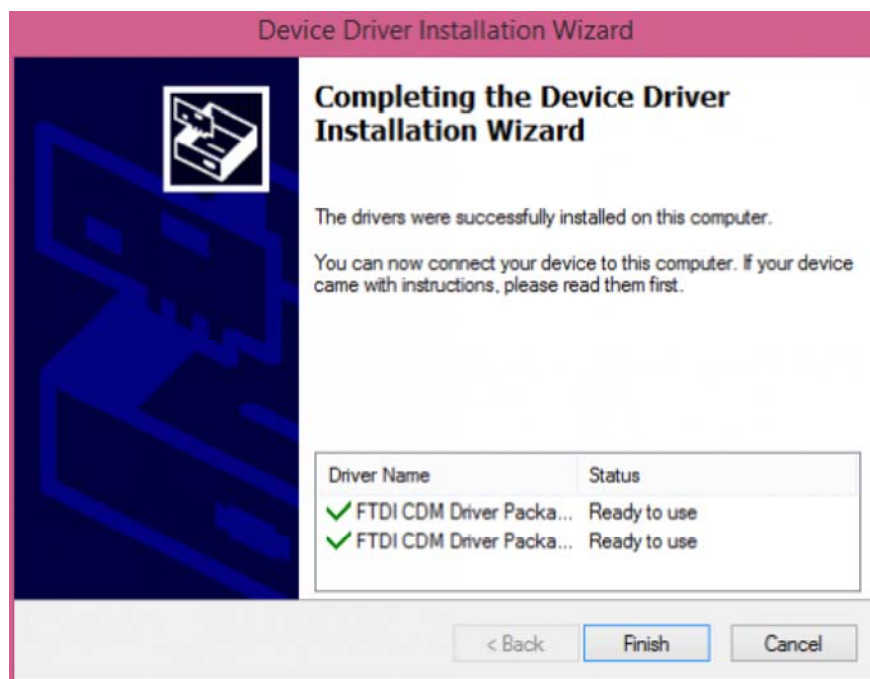
B.Install Driver

Download FTDI drivers (<http://www.ftdichip.com/Drivers/CDM/CDM%20v2.10.00%20WHQL%20Certified.exe>)

Install CDM v2.10.00 WHQL Certified.exe



(/wiki/index.php/File:DFR0331_driver1.png)



(/wiki/index.php/File:DFR0331_driver2.png)

Download Intel Edison Drivers (<https://communities.intel.com/docs/DOC-23242>)

Edison - Software Downloads



由 sjaworski 于 2014-9-5 下午 12:30 创建, 最后由 intel_jorge 于 2014-11-14 下午 5:35 修改。

Package release version Arduino 1.5.3 - Intel 1.0.4

See the [Getting Started Guide](#) for step-by-step information on installing the software.

Download the IDE zip file for your operating system (OS). Each OS zip file includes the latest firmware so you can use the IDE to update your board.

Galileo, Edison

Link	IDE Software	Operating System	Intel Board	File Size	File Type
Download	Arduino Software 1.5.3 - Intel 1.0.4	Linux 32 Bit	Galileo, Edison	145MB	tgz
Download	Arduino Software 1.5.3 - Intel 1.0.4	Linux 64 Bit	Galileo, Edison	149MB	tgz
Download	Arduino Software 1.5.3 - Intel 1.0.4	Mac OSX	Galileo, Edison	112MB	zip
Download	Arduino Software 1.5.3 - Intel 1.0.4	Windows	Galileo, Edison	159MB	7z
Download	Intel customized Arduino IDE 1.5.3 (source code)	N/A	376MB	bz2	

Source files, Image and SDK's. Version: **Rel-1-Maint-WW42**. [Release Notes](#).

Link	Software	Operating System	Board	File Size	File Type	Notes
Download	Edison Yocto complete image	Linux - Yocto	Edison	105MB	zip	
Download	Edison SDK Linux 32	Linux	Edison	380MB	zip	
Download	Edison SDK Linux 64	Linux	Edison	382MB	zip	
Download	Edison SDK Mac OSX	OSX	Edison	347MB	zip	
Download	Edison SDK Windows 32	Windows 32	Edison	421MB	zip	
Download	Edison SDK Windows 64	Windows 64	Edison	425MB	zip	
Download	Edison Linux source files	Linux	Edison	34MB	tgz	
Download	Edison GPL/LGPL source files	Linux	Edison	603MB	tar	

Driver Software.

Link	Software	Operating System	Board	File Size	File Type	Notes
Download	Windows Driver setup 1.0.0	Windows	Edison	9MB	exe	

For older versions of the software and images (WW36), visit this [link](#).

Note: As Intel is committed to providing support primarily for the latest software and images, we offer access to these older versions for your convenience only.

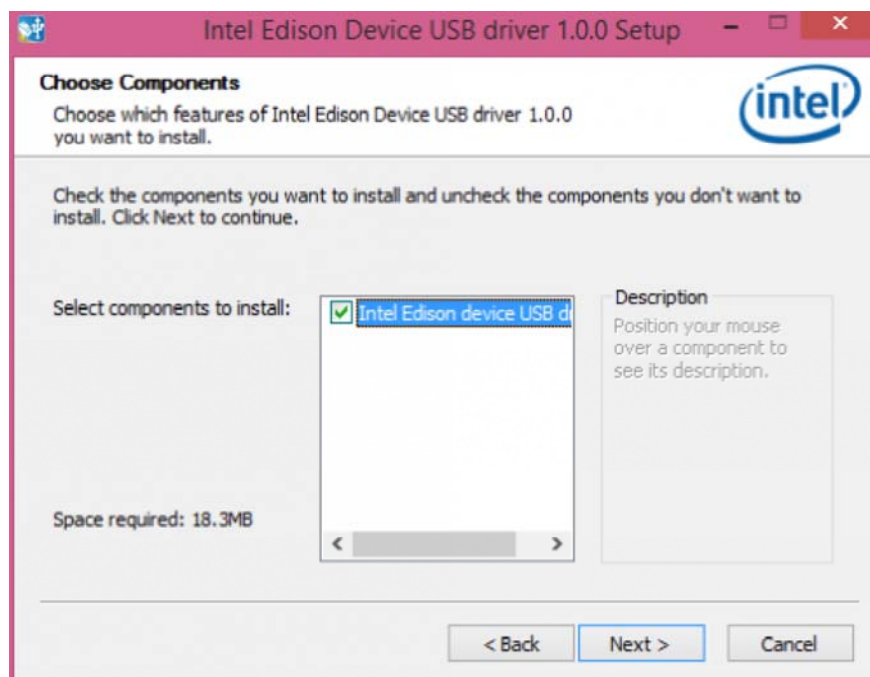
122759 查看  分类:  标签: linux, software, drivers, yocto, edison, arduino sw, arduino software

(/wiki/index.php/File:DFR0331_Edison_driver.png)

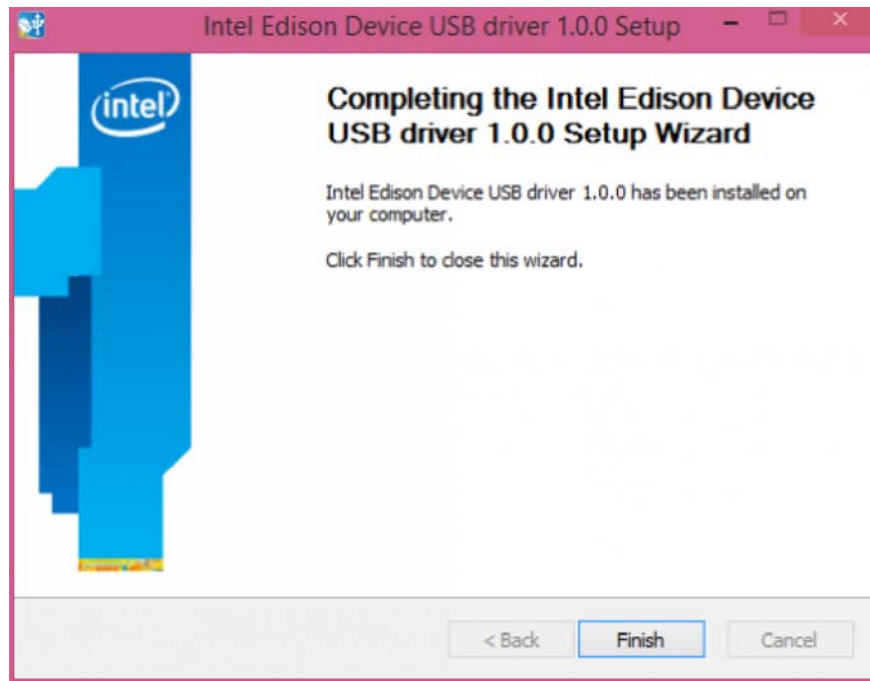
Install Edison Driver



(/wiki/index.php/File:DFR0331_Edison_driver1.png)

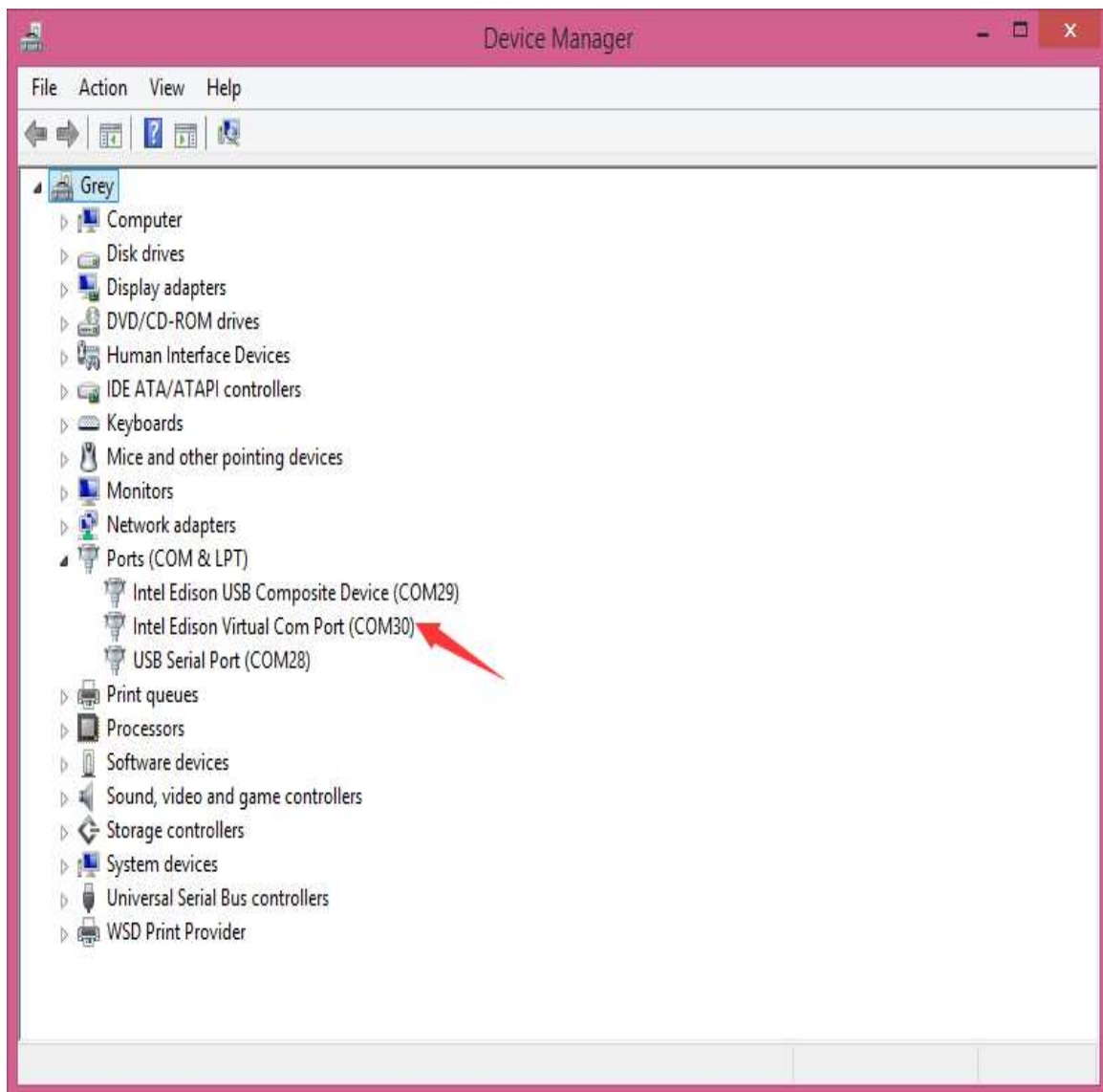


(/wiki/index.php/File:DFR0331_Edison_driver2.png)



(/wiki/index.php/File:DFR0331_Edison_driver3.png)

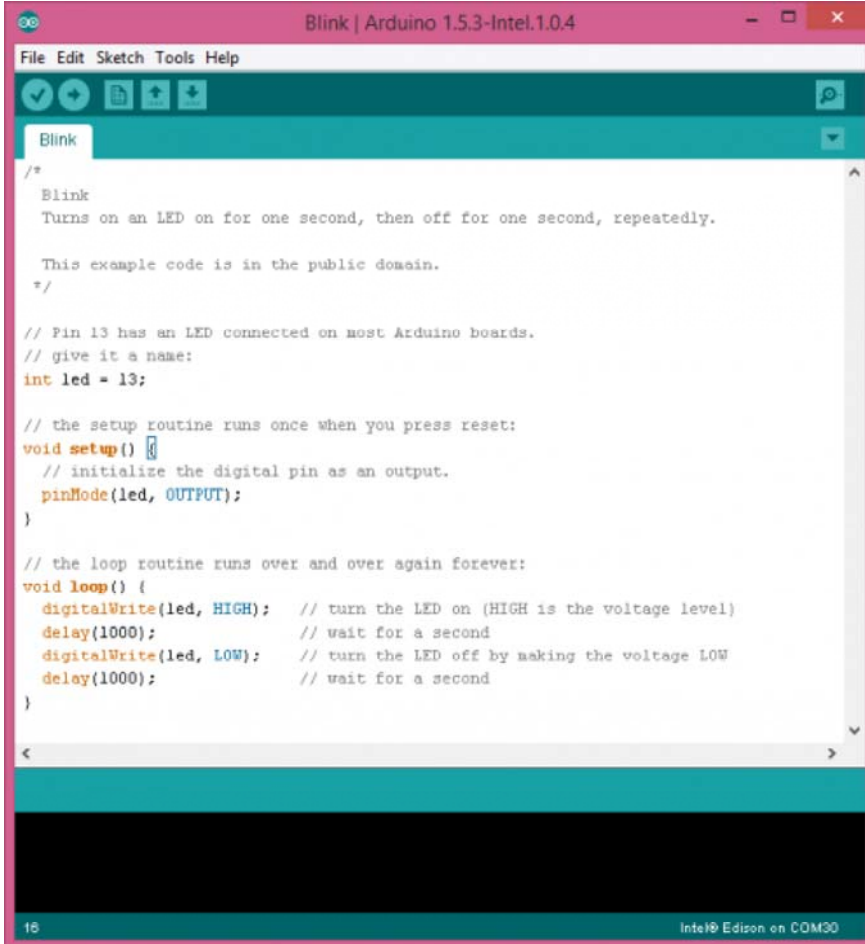
Open the Device Manager, and check Device Driver



(/wiki/index.php/File:DFR0331_Edison_Device_Manager.png)

C.Quick test with the module

Open Edison Arduino IDE and open a sample code "Blink"

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.5.3-Intel.1.0.4". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, opening, and other functions. The main text area contains the following code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

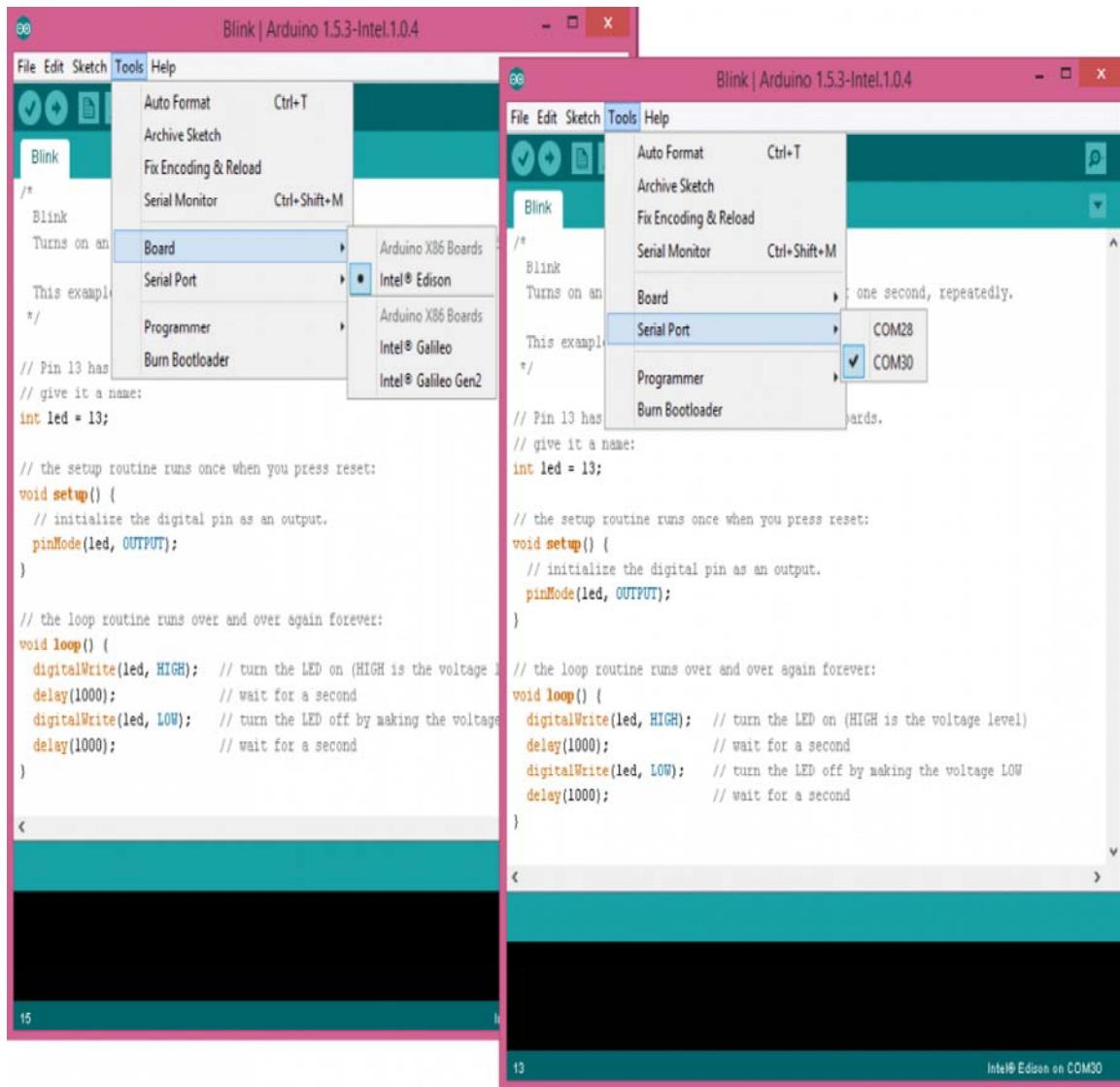
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

The status bar at the bottom of the IDE shows "Intel Edison on COM30".

(/wiki/index.php/File:DFR0331_Edison_Quicktest_Blink.png)

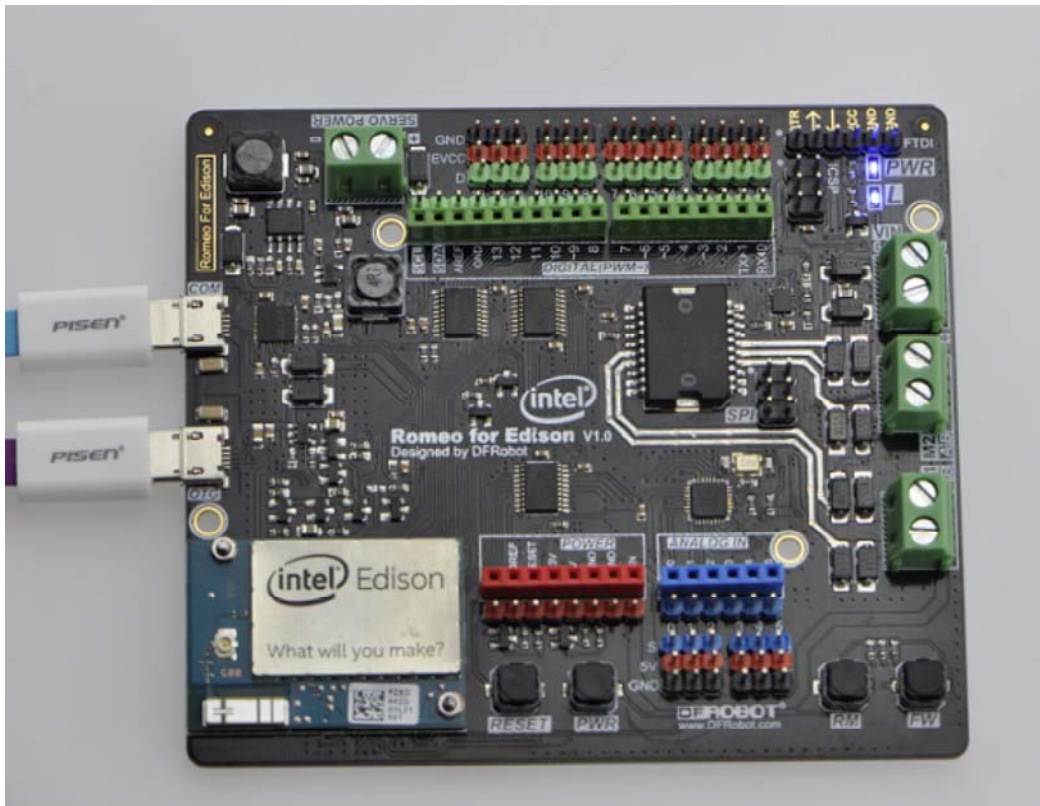
Select Tools-->Board-->Intel Edison

Select Tools-->Serial Port-->COM xx



(/wiki/index.php/File:DFR0331_Edison_Quicktest_Board.png)

When the uploading has been finished, the LED will be flashing once a second.



(/wiki/index.php/File:DFR0331_Edison_Quicktest_Board_Blink.png)

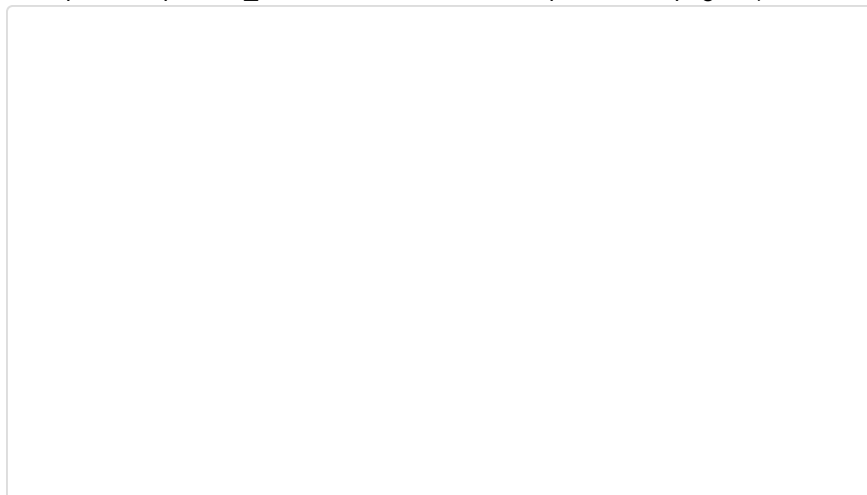
Romeo for Edison Controller Typical Applications

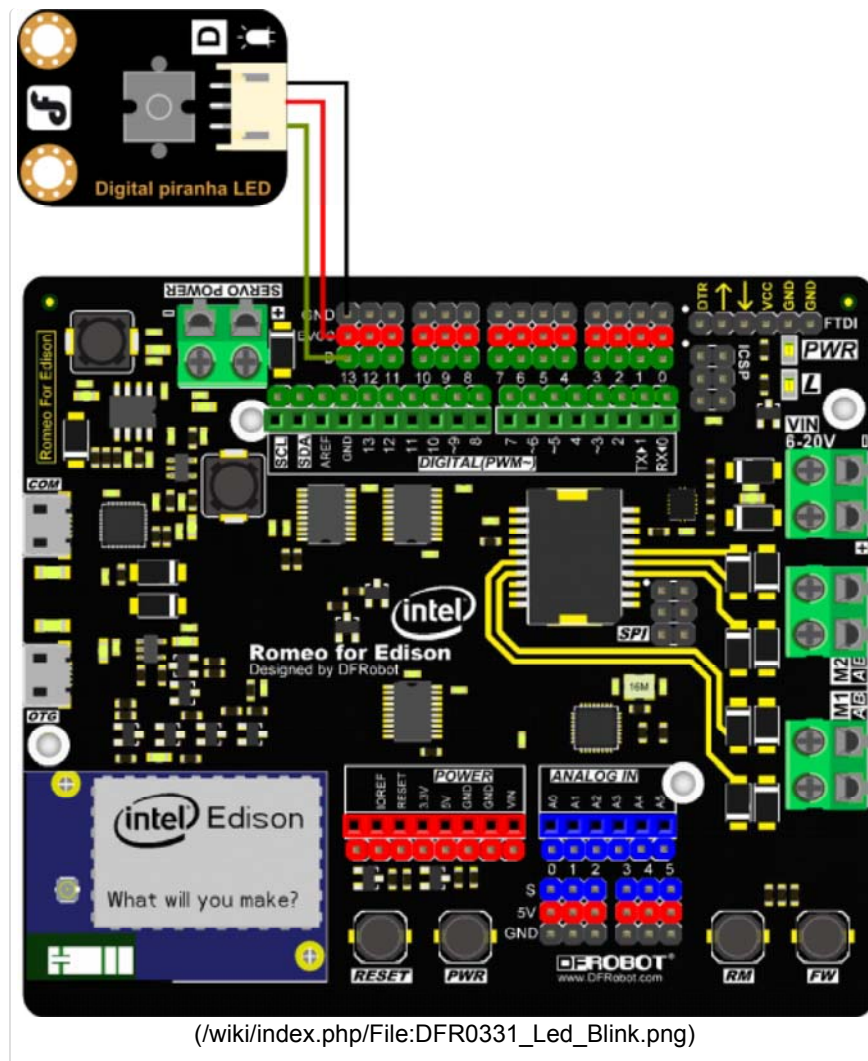
Digital Application

A.LED Blink

Part List and Hardware Connection:

- Micro USB cable x1
- Romeo for Edison Controller x1
- Intel® Edison x1
- Digital Piranha LED light module (https://www.dfrobot.com/index.php?route=product/product&product_id=632&search=led&description=true&page=5) x1





Sample code: Open Arduino IDE, Select Boards --> Intel Edison and COM port

```
int led = 13;

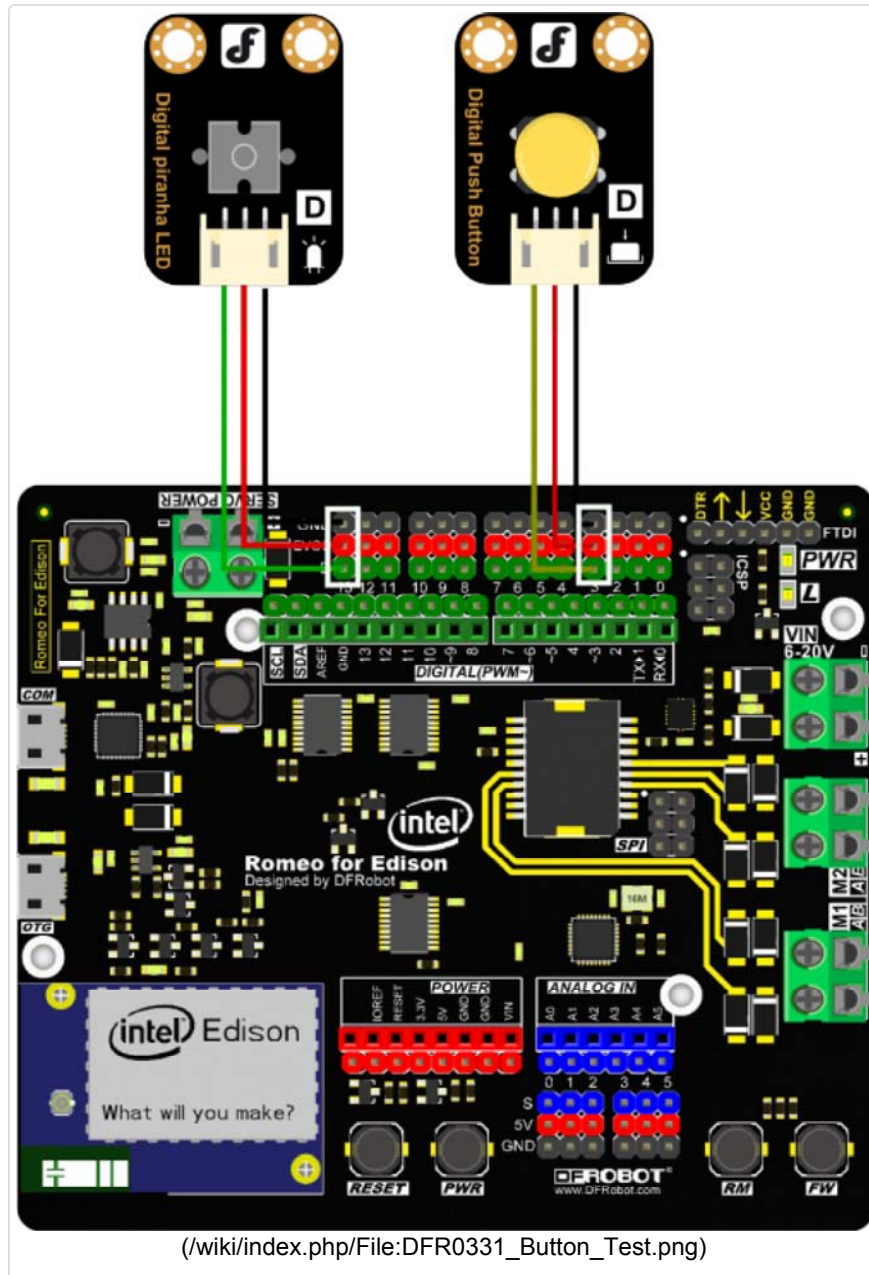
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

B.Button Test

Part List and Hardware Connection:

- Micro USB cable x1
- Romeo for Edison Controller x1
- Intel® Edison x1
- Digital Piranha LED (https://www.dfrobot.com/index.php?route=product/product&product_id=632&search=led&description=true&page=5) x1
- Digital Push Button (https://www.dfrobot.com/index.php?route=product/product&product_id=1095&search=DFR0029&description=true) x1



Sample code: Open Arduino IDE, Select Boards -->Intel Edison and COM port


```
int buttonPin = 2;    // Define Button Pin D2
int ledPin = 13;     // Define LED Pin D13

int buttonState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);    // Define D13 Output Mode
  pinMode(buttonPin, INPUT);  // Define D2 Input Mode
}

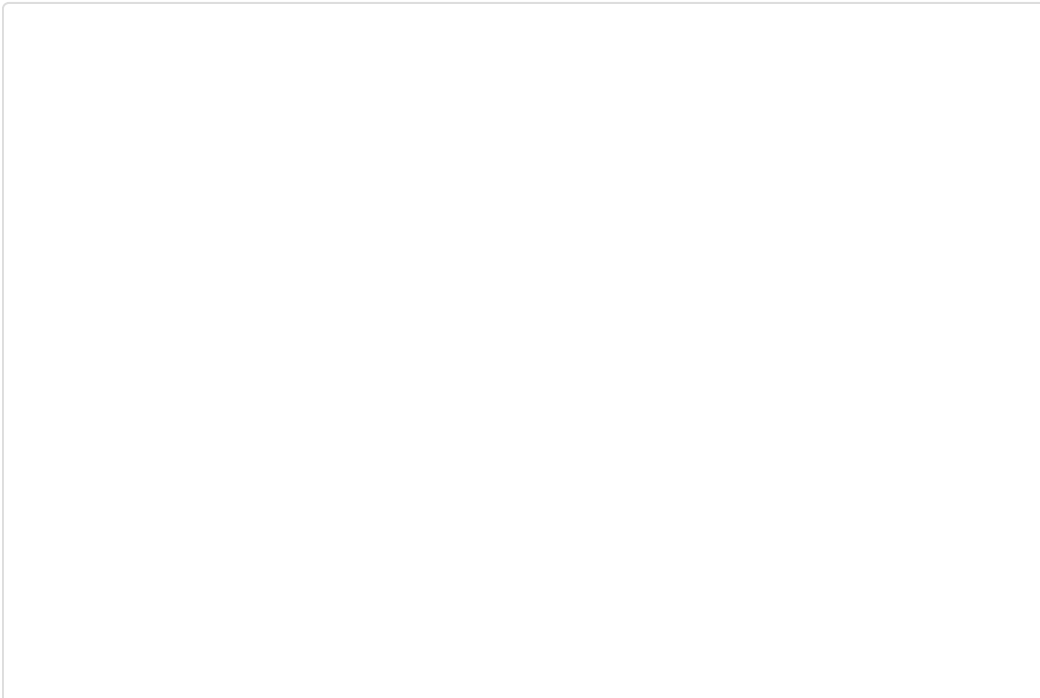
void loop(){
  buttonState = digitalRead(buttonPin); //Read D2 Status

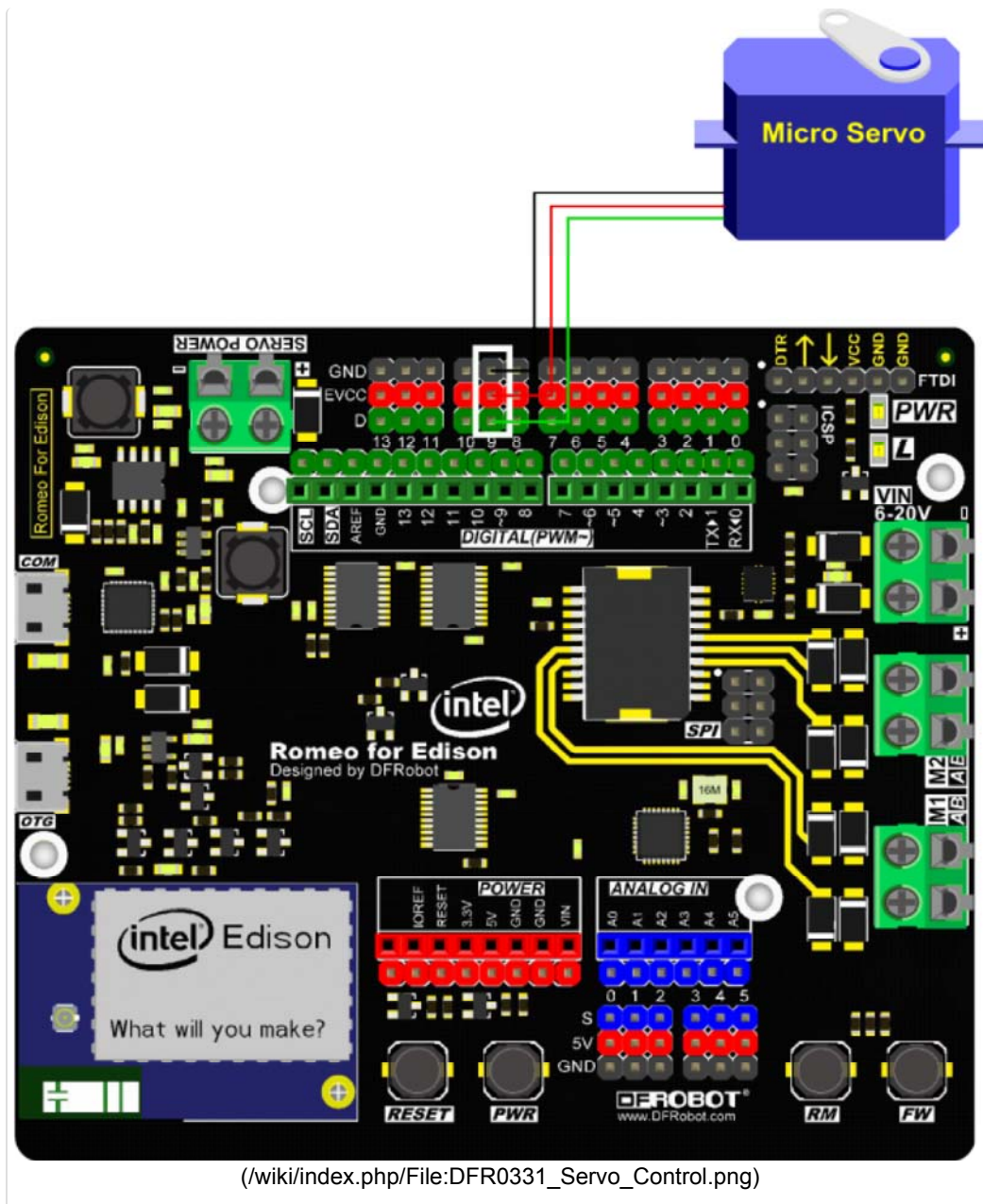
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH); //Turn On LED
  }
  else {
    digitalWrite(ledPin, LOW);  //Turn Off LED
  }
}
```

C.Servo Control

Part List and Hardware Connection:

- Micro USB cable x1
- Romeo for Edison Controller x1
- Intel® Edison x1
- Micro Servo (https://www.dfrobot.com/index.php?route=product/product&product_id=255&search=servo&description=true&category_id=48) x1





Sample code: Open Arduino IDE, Select Boards --> Intel Edison and COM port

```

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
                // a maximum of eight servo objects can be created

int pos = 0;    // variable to store the servo position

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1) // goes from 0 degrees to 180 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable
    'pos'
    delay(15);                       // waits 15ms for the servo to reach the pos
    ition
  }
  for(pos = 180; pos>=1; pos--=1)    // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable
    'pos'
    delay(15);                       // waits 15ms for the servo to reach the pos
    ition
  }
}

```

NOTE: Only PWM interface could control the Servo.

D.Motor Control

Note: We made a lot of update for "Romeo for Edison main board", to manage them easily we have moved it on github.
 NG.ino is ATmega8 firmware; DFRobotEdison is Romeo motor and analog driver library.

New Romeo for Edison library (<https://github.com/ouki-wang/remeo4edison>)

Old Romeo for Edison library

(https://github.com/Arduinolibrary/DFRobot_Intel_Edison/raw/master/DFRobotEdison%20V1.1.zip)

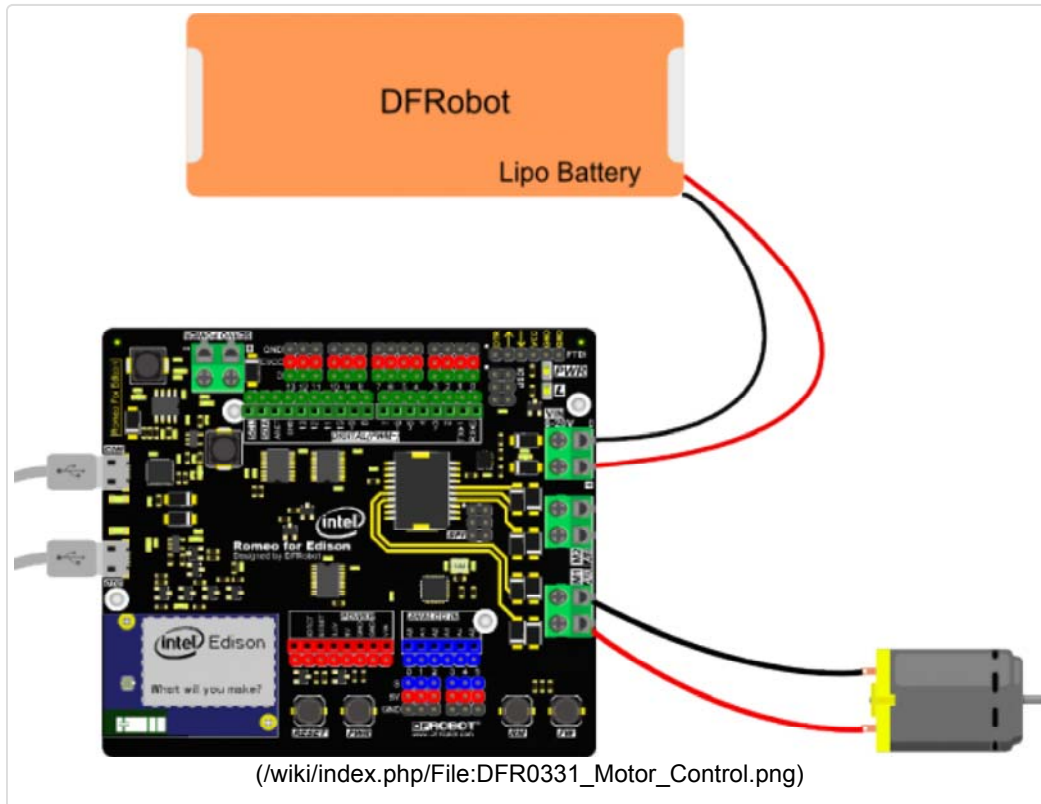
Note:

The product manufactured after Sep. 2015 have burned the newest firmware. Please use the corresponding library to avoid bugs.
 For the older product, you could update ATmega8 firmware to use the newest library.

Leave the downloaded library file under Arduino library folder: e.g. arduino-1.6.3\libraries Directory.

Part List and Hardware Connection:

- Micro USB cable x1
- Romeo for Edison Controller x1
- Intel® Edison x1
- Micro Metal Gear Motor (https://www.dfrobot.com/index.php?route=product/product&product_id=874&search=FIT0302&description=true) x1



Sample code: Open Arduino IDE, Select Boards --> Intel Edison and COM port

```
/*
 *Copyright (c) 2014, Chengdu Geeker Technology Co., Ltd.
 *All rights reserved
 *
 *FileName : Motor.ino
 *File Tag :
 *Summary : moter control example
 *
 *Version : V1.2
 *Author : ouki.wang(ouki.wang@dfrobot.com)
 *Time : 2015.10.10

 *Version : V1.1
 *Author : weibing.deng(15196617738@163.com)
 *Time : 2015.1.7
 *
 *Version : V1.0
 *Author : weibing.deng(15196617738@163.com)
 *Time : 2014.12.22
 */
#include <DFRobot.h>
#include <IIC1.h>
DFRobotEdison Motor1;
DFRobotEdison Motor2;

void setup() {
  Motor1.begin(M1); /*Initialize motor1*/
  Motor2.begin(M2); /*Initialize motor2*/
}

void loop() {
  Motor2.move(); /*motor rotate with the fastest speed*/
  Motor1.move();
  delay(3000);
  Motor2.setDirection(CLOCKWISE); /*Motor M2 clockwise rotation*/
  Motor1.setDirection(CLOCKWISE); /*Motor M1 clockwise rotation*/

  Motor2.setSpeed(100); /*set speed 100 (0-255)*/
  Motor1.setSpeed(100); /*set speed 100 (0-255)*/
  delay(3000);

  Motor2.setDirection(ANTICLOCKWISE); /*Motor M2 anticlockwise rotation*/
  Motor2.setSpeed(200); /*set speed 200 (0-255)*/
  Motor1.setDirection(ANTICLOCKWISE); /*Motor M1 anticlockwise rotation*/
  Motor1.setSpeed(200); /*set speed 300 (0-255)*/
  delay(3000);

  Motor2.stop(); /*stop moter*/
  Motor1.stop(); /*stop moter*/
  delay(3000);
}
```

Analog Application

A.Analog Read

Part List and Hardware Connection:

- Micro USB cable x1
- Romeo for Edison Controller x1
- Intel® Edison x1

NOTE: As Romeo for Edison is using ATmega 8 as Analog pins driver. Please download the DFRobot4Edison library first.
The library is keeping update, so some sample code and picture will be different to the old one.

Sample code: Open Arduino IDE, Select Boards -->Intel Edison and COM port

In this section, please download the DFRobot4Edison library (<https://github.com/ouki-wang/remeo4edison>) first.

```
/*
 *Copyright (c) 2014, Chengdu Geeker Technology Co., Ltd.
 *All rights reserved
 *
 *File Name : analogRead.ino
 *File Tag :
 *Summary : A0-A5 port Analog reading example
 *
 *Version : V1.0
 *Author : weibing.deng(15196617738@163.com)
 *Time : 2014.12.22
 */
#include <DFRobot.h>
#include <IIC1.h>

unsigned int value = 0;

void setup() {
  Serial.begin(9600);
}

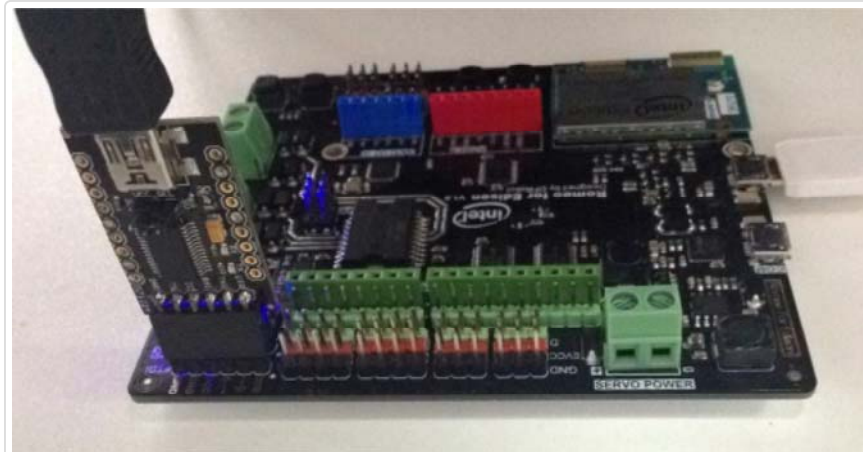
void loop() {
  value = analogRead(A0);
  Serial.print("A0=");Serial.println((int)value);
  value = analogRead(A1);
  Serial.print("A1=");Serial.println((int)value);
  value = analogRead(A2);
  Serial.print("A2=");Serial.println((int)value);
  value = analogRead(A3);
  Serial.print("A3=");Serial.println((int)value);
  value = analogRead(A4);
  Serial.print("A4=");Serial.println((int)value);
  value = analogRead(A5);
  Serial.print("A5=");Serial.println((int)value);

  delay(2000);
}
```

Atmega8 Firmware Upgrade

Tools: FTDI programmer (http://www.dfrobot.com/index.php?route=product/product&product_id=147&search=ftdi&description=true) x1

Download the newest Atmega8 firmware here (<https://github.com/ouki-wang/remeo4edison>) (The newest firmware fixed IIC communication bug)



(/wiki/index.php/File:DFR0331_Atmega8_firmware.png)

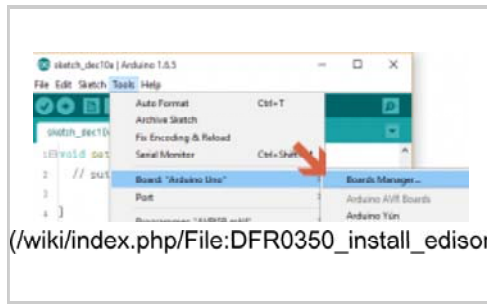
If you find that there is something wrong with Analog input pins and motor driver, please update the ATmega8 firmware. (A more clear tutorial about how to use the FTDI programmer to update ATmega8. (<http://www.dfrobot.com/forum/viewtopic.php?f=5&t=1852>))

1. Plug FTDI module on the Romeo (FTDI module's pin map should be located on 3.3V)
2. Open Arduino IDE, select the right board ATmega8 (It is in the Arduino NG and older——>Processorer "ATmega168"——>"ATmega8")
3. Update the last github firmware.
4. Done!

FAQ

Q1. No Arduino for Edison IDE download link could be found from the link you give, i.e. Intel ... Downloads (<https://software.intel.com/en-us/iot/hardware/edison/downloads>), and after I finished installing all software, I got newest Arduino IDE 1.6, and there is no place for me to choose Intel to upload Blink.ino, what should I do?

A. Welcome ! You can go to **Tools > Board > Boards Manager...**, and find **Intel i686 Edison**, click **Install** to make it. After you install the needed files, you could see Intel choice.



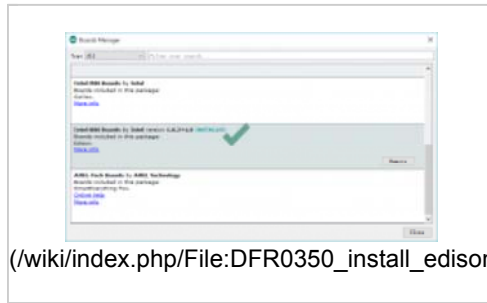
(/wiki/index.php/File:DFR0350_install_edison_in_arduino_1.6_1.png)

Q1. 1 Open Boards Manager



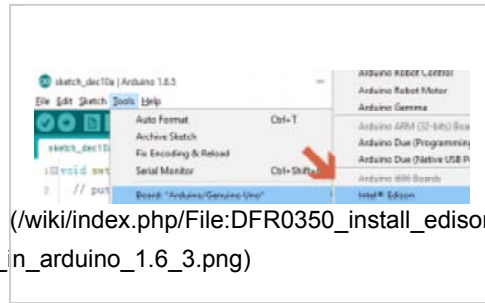
(/wiki/index.php/File:DFR0350_install_edison_in_arduino_1.6_2.png)

Q1. 2 Install i686 Edison



(/wiki/index.php/File:DFR0350_install_edison_in_arduino_1.6_3.png)

Q1. 3 Waiting to install finished



(/wiki/index.php/File:DFR0350_install_edison_in_arduino_1.6_4.png)

Q1. 4 Choose the target board, Intel® Edison

Q2. Win XP/ Win 7 incompatible issue.

A. Intel Edison has a compatible issue on Win 7/Win XP, please upgrade your Operating System to Win 8/10.

Q3. My motors lose control/ Analog reading error/ TWI(IIC) communication not good.

A. It is for the TWI(IIC) issue, please upgrade Atmega8 firmware to the newest version, and use the corresponding newest library.

Q4. I can't debug via COM port without OTG plugged.

A. Internal Edison will detect the value of "VBUS" pin, and it doesn't permit debugging while "VBUS" is LOW. It means you need connect a OTG cable to OTG port.(It doesn't need to connect anything to OTG cable), when you want to debug Edison via the COM port.

Q5. I'm using **Intel XDK** to develop your Romeo board rev 1.0. I have the code from the example library and work perfectly with Intel Edison Breakout Kit (Groove kit) board, but when I run it on Romeo board it's fire error. I checked all the ports and they are mixed. Do you have the table where is the ports?

A. The pin-mapping is different from Eclipse/Intel XDK and Arduino IDE. And you can leave out the mapping problem, just use the table D0-D13,A0-A5 on Romeo for Edison, and use Arduino IDE to develop your program according to our wiki. If you do need the mapping table for Intel XDK/ Eclipse development, please contact Intel official site (<http://www.intel.com/content/www/us/en/support/boards-and-kits.html>) for help. Thanks for your understanding!

For any question/ advice/ cool idea to share with us, please visit **DFRobot Forum**
(<http://www.dfrobot.com/forum/>).

Development Tools

- Romeo for Edison Schematic
(https://github.com/Arduinolibrary/DFRobot_Intel_Edison/blob/master/Romeo%20for%20Edison%20Controller%20Schematic.pdf?raw=true)
- Old version Romeo for Edison library V1.1
(https://github.com/Arduinolibrary/DFRobot_Intel_Edison/raw/master/DFRobotEdison%20V1.1.zip)
- New version Romeo for Edison library V1.2 (<https://github.com/ouki-wang/remeo4edison>)
- Intel Edison Getting Started Tutorial (<https://communities.intel.com/docs/DOC-23147>)
- Edison Related Software Download (<https://communities.intel.com/docs/DOC-23242>)
- Edison Firmware Update Tutorial (<https://communities.intel.com/docs/DOC-23192>)
- Build your Own Edison System Image Tutorial (<https://communities.intel.com/docs/DOC-23159>)
- Intel Edison Introduction (<https://communities.intel.com/docs/DOC-23139>)
- Edison Module Introduction Webpage (<http://www.intel.com/content/www/us/en/do-it-yourself/edison.html>)
- Edison Development Forum (https://communities.intel.com/community/makers/edison/getting-started?_ga=1.101445057.1901666174.1406084086)

→ (/wiki/index.php/File:Nextredirectltr.png)Go Shopping Romeo for Edison Controller(SKU:DFR0331)
(http://www.dfrobot.com/index.php?route=product/product&product_id=1241&search=DFR0350&description=true&category_id=48)

→ (/wiki/index.php/File:Nextredirectltr.png)Go Shopping Romeo for Edison Controller(SKU:DFR0350)
(https://www.dfrobot.com/index.php?route=product/product&product_id=1238&search=DFR0350&description=true)

Categories (/wiki/index.php/Special:Categories):

Product Manual (/wiki/index.php/Category:Product_Manual)

| DFR Series (/wiki/index.php/Category:DFR_Series) | Sensors (/wiki/index.php/Category:Sensors)

This page was last modified on 18 April 2016, at 07:54.

Content is available under GNU Free Documentation License 1.3 or later (<https://www.gnu.org/copyleft/fdl.html>) unless otherwise noted.