1. OpenCM9.04 Integrated Development Environment (IDE)

① ROBOTIS OpenCM download

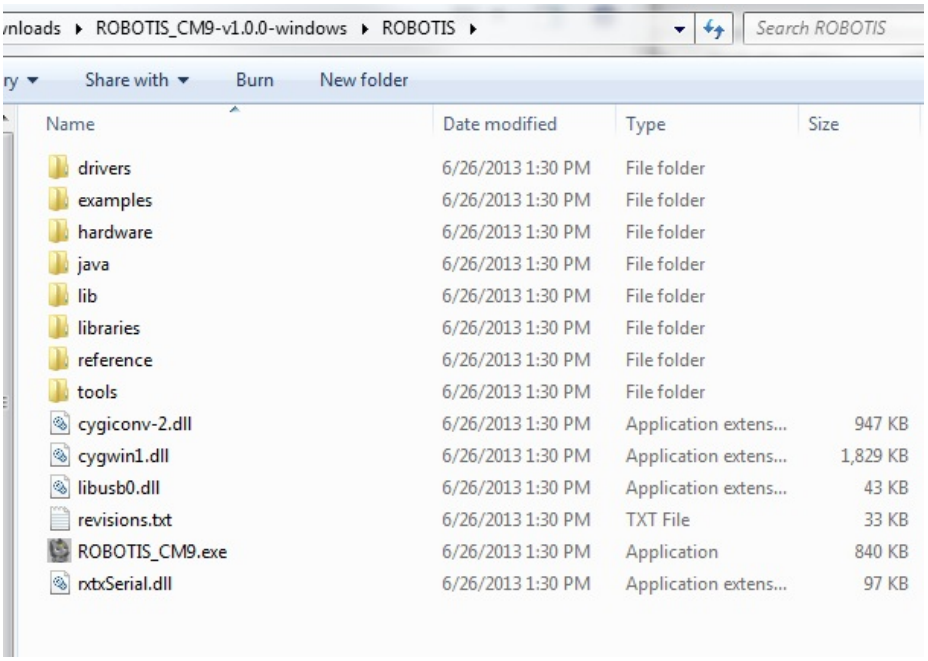
To program the OpenCM9.04 ROBOTIS OpenCM is necessary. ROBOTIS OpenCM can be downloaded from Robotis' e-manuals. More information relating to the OpenCM9.04 series can be found in RobotSource

http://support.robotis.com

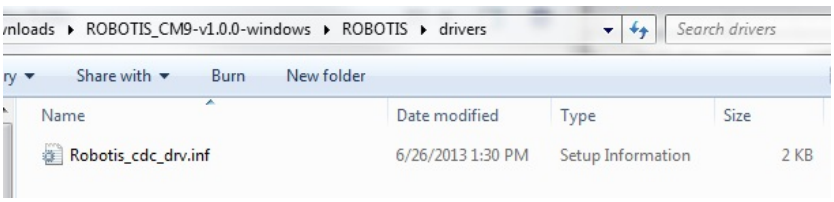
www.robotsource.org

② ROBOTIS OpenCM structure

Download and decompress the file. Once decompressed the structure appears as illustrated below.

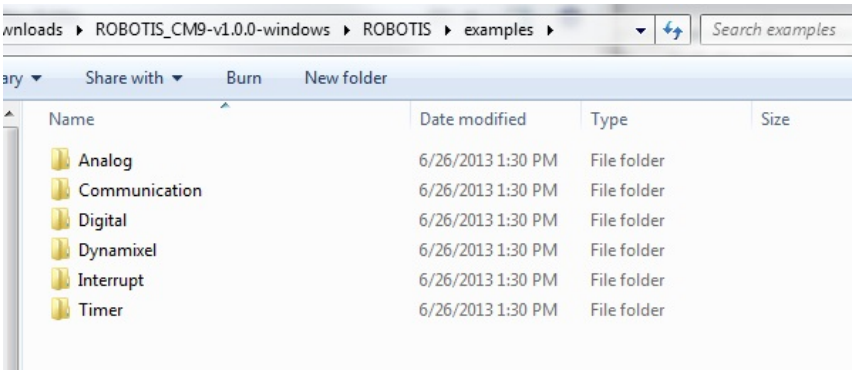


A. Drivers: contains Windows USB drivers (.inf).

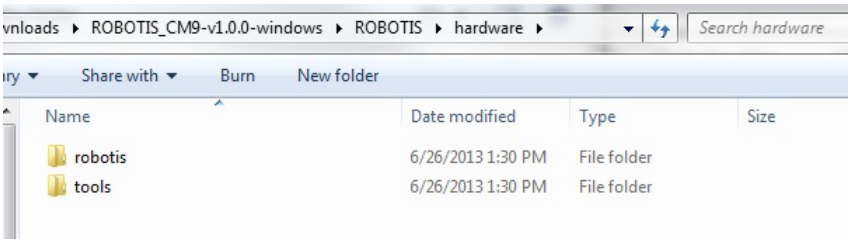


B. Examples: contains ROBOTIS OpenCM's sketch examples from the IDE's

menu.

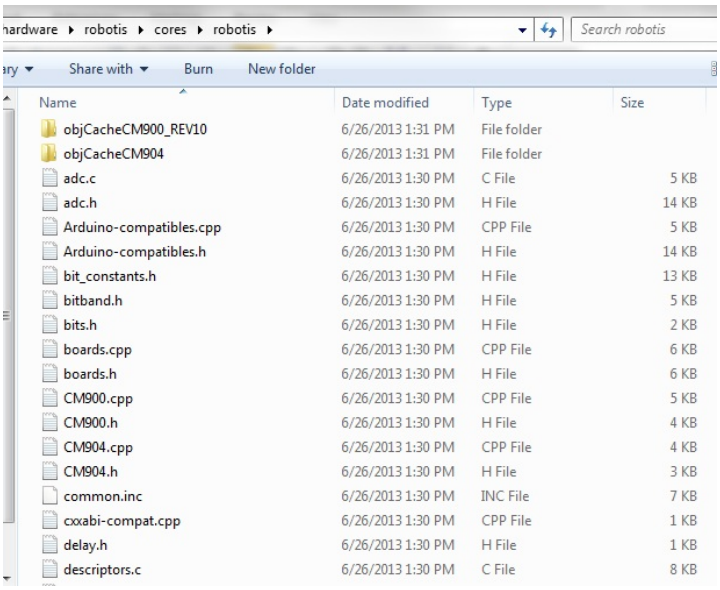


C. Hardware: C/C++ codes and ARM compiler for the OpenCM series.



The robotis directory contains basic OpenCM9.04 APIs

ROBOTIS/hardware/robotis/cores/robotis

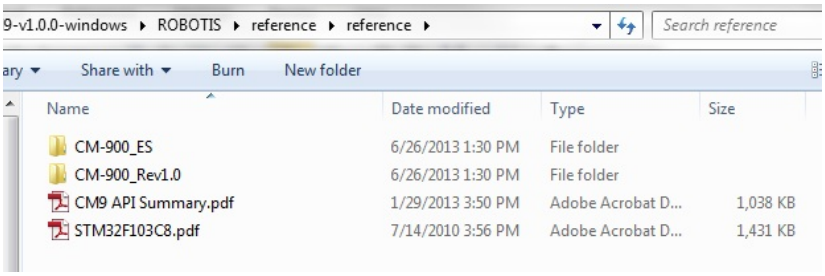


D. Java: contains Windows-based Java Runtime Environment (JRE).

E. Lib: contains ROBOTIS OpenCM resources.

F. Libraries: contains sketch libraries.

G. Reference: contains OpenCM series datasheet and API documentation.



H. Tools: ROBOTIS Processing tool for the OpenCM.

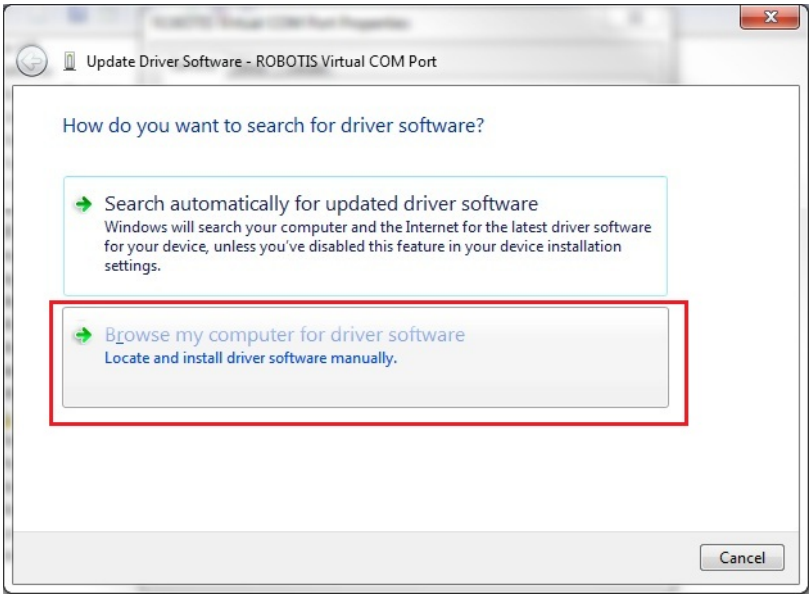
I. ROBOTIS_OpenCM.exe: ROBOTIS OpenCM executable.

The abovementioned structure is Windows-centric package. Linux/Mac OS X packages are listed further in this document.

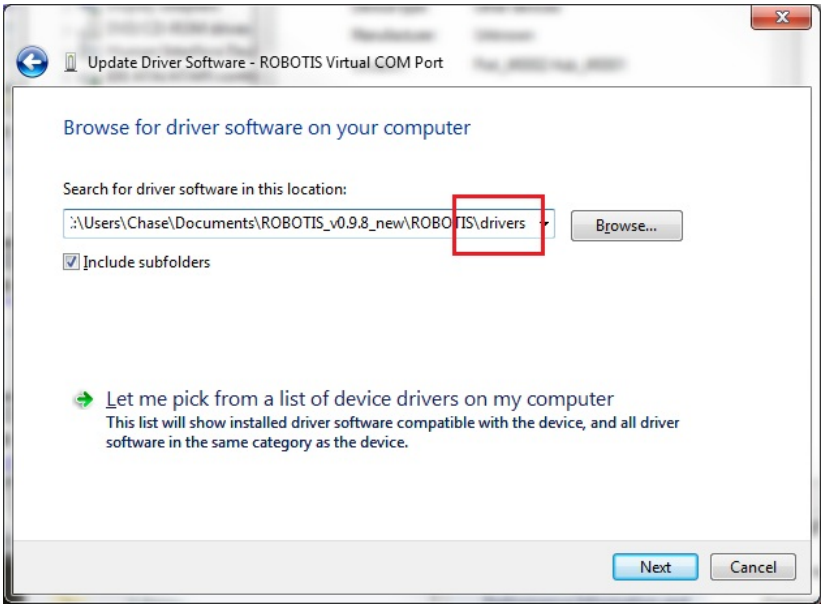
③ USB driver installation

To program the OpenCM9.04 with the ROBOTIS OpenCM USB drivers must be installed. The following procedure is Windows OS specific. Linux and Mac OS X users do not need to follow this procedure.

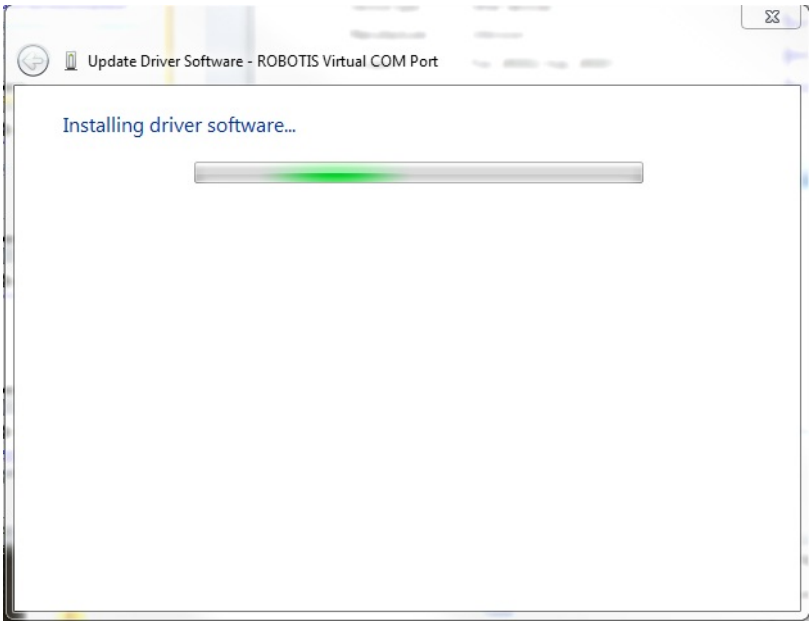
During a link between OpenCM9.04 and PC the device appears as a ROBOTIS Virtual COM Port device. "With the mouse right click select update driver software.



Pick "browse my computer for driver software"

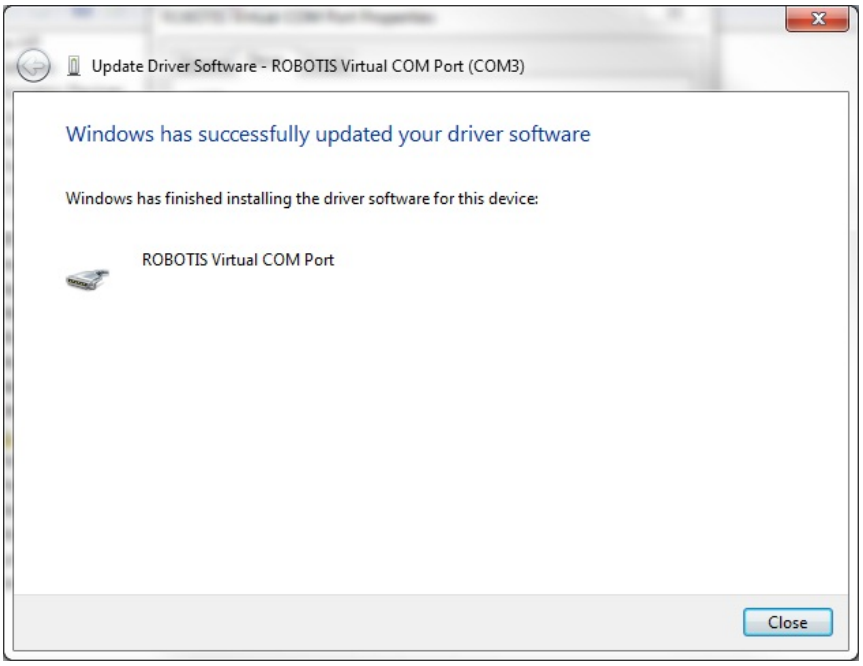


Click on "**browse**" and select "drivers" directory (from ROBOTIS/drivers).



There may be an instance where Windows Security will issue a driver publisher warning If so, simply click on "install this driver software anyways"

Once install is complete a window will appear as illustrated below.

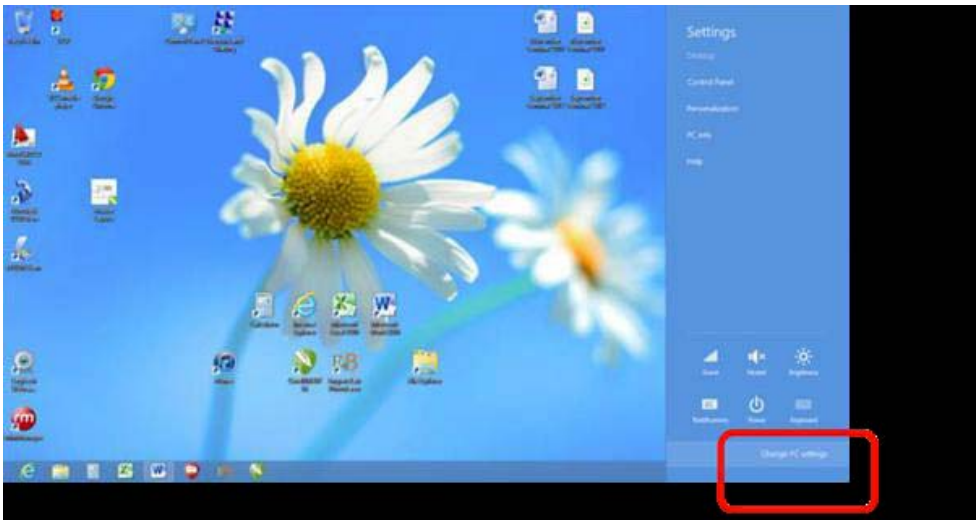


Look for the COM port number under ROBOTIS Virtual COM Port. Remember the number

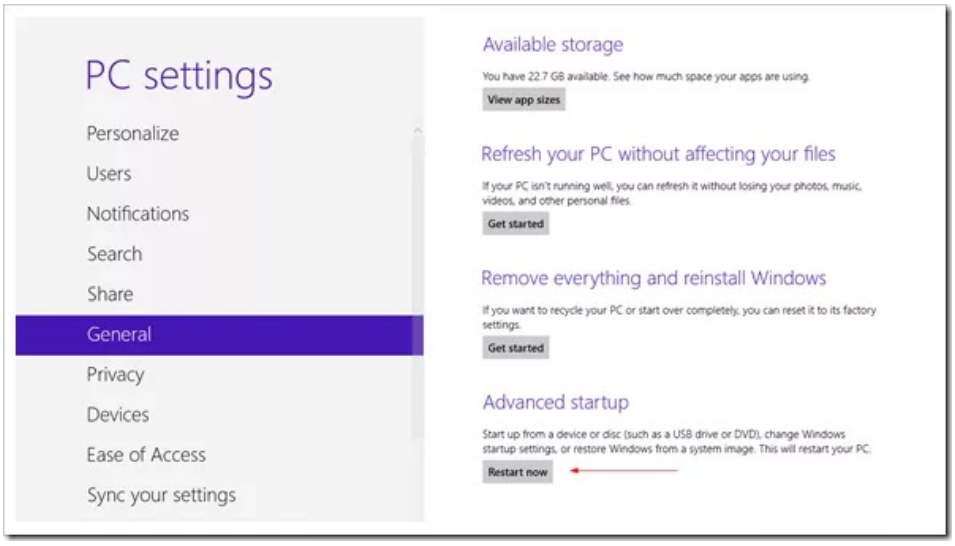
Resolve Windows 8 USB installation issue

Open the Charm Bar (the right side of the screen) and click on "Settings".

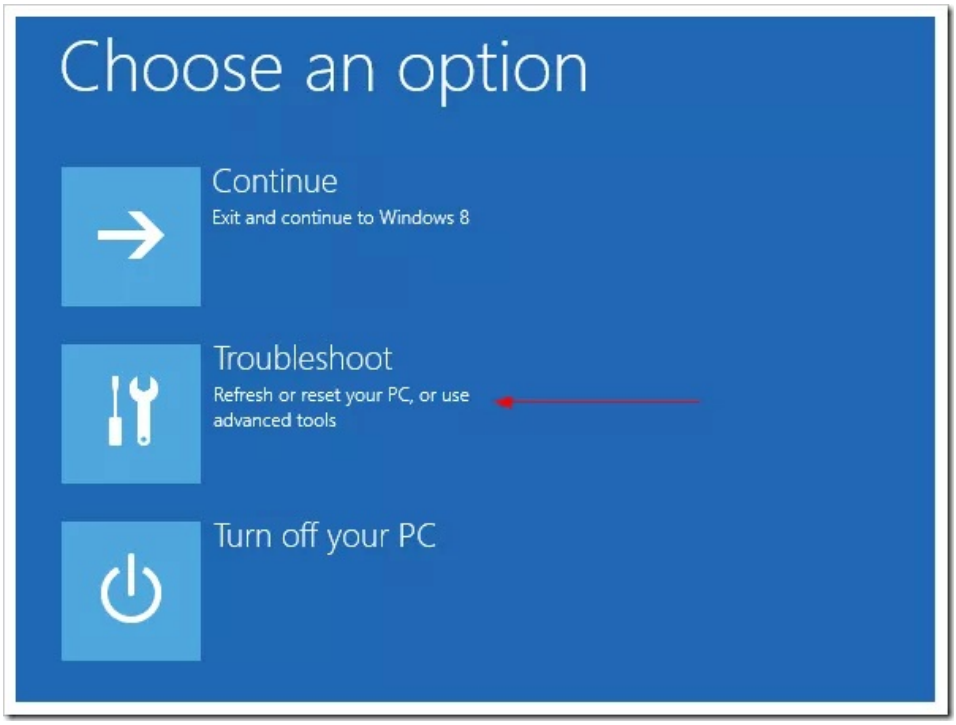
Click on Change PC Settings at the bottom right.



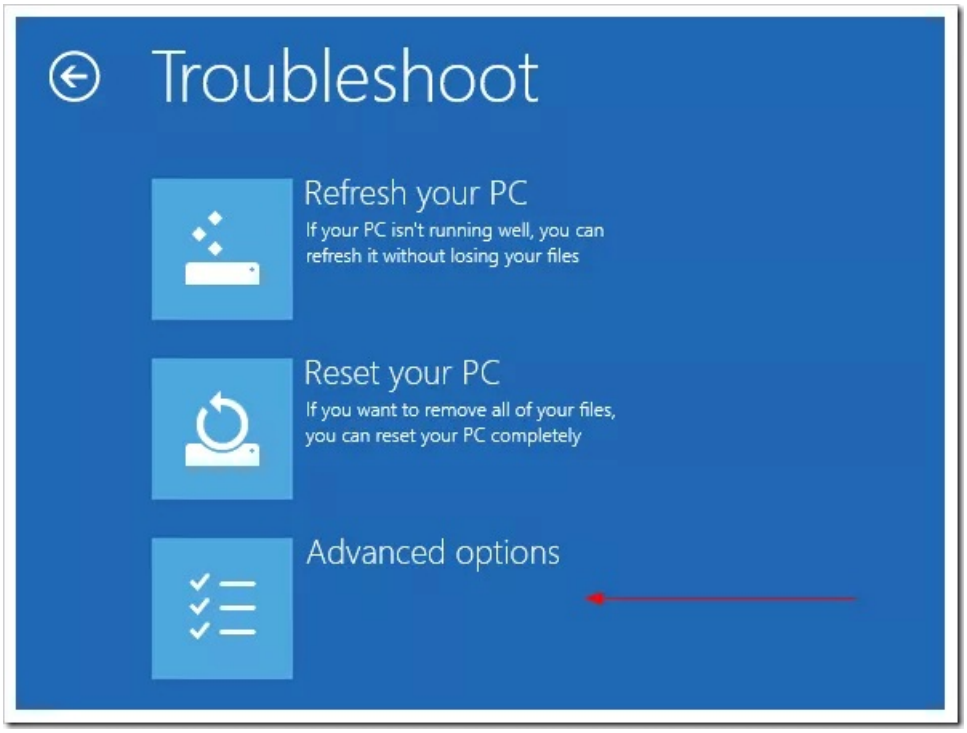
From PC Settings's list on the left side select General.



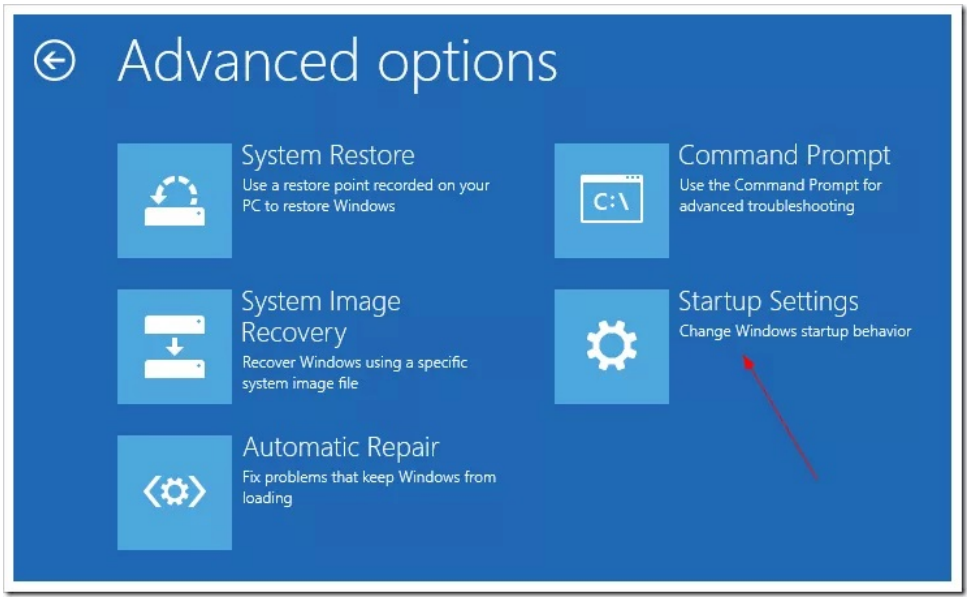
Under the Advanced Start-up option click on the Restart Now button. The PC will reboot and you will be asked to choose an option



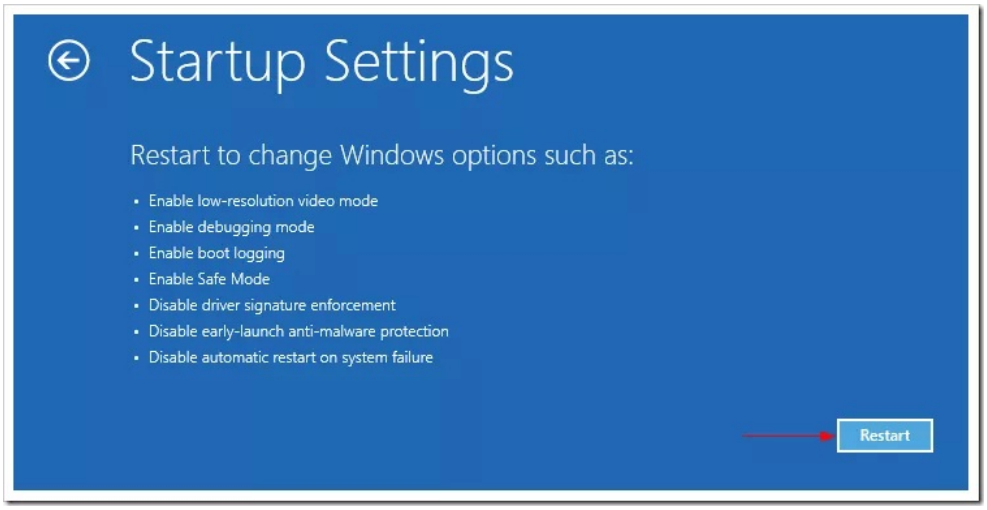
Click on Troubleshoot option



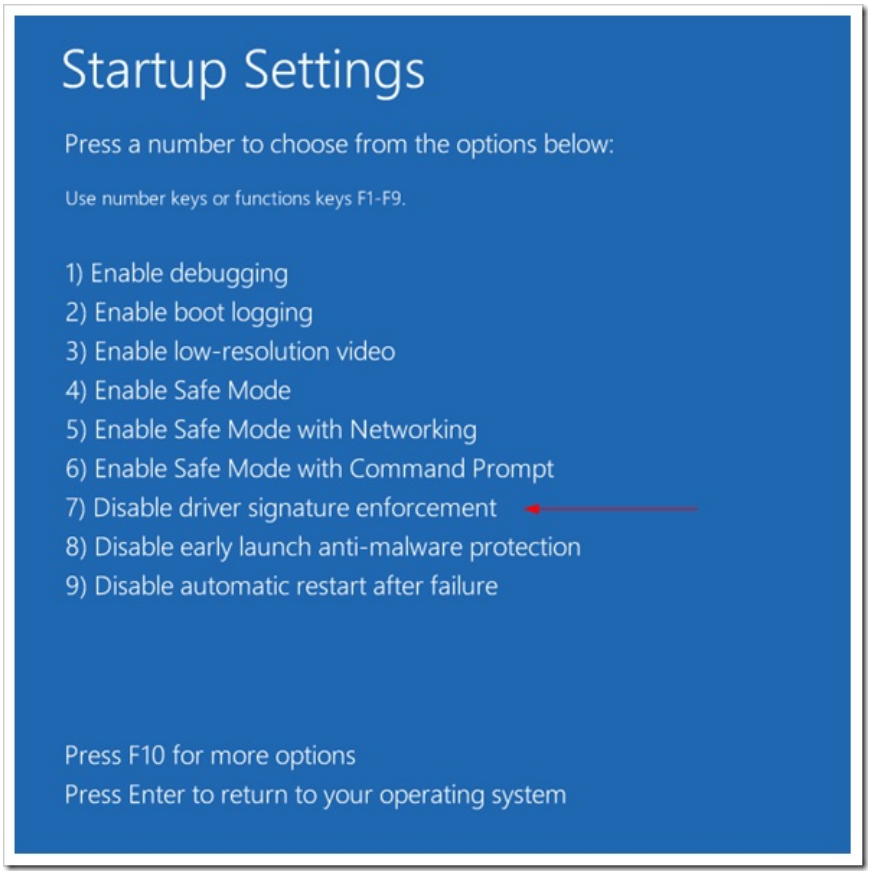
Click on Advanced options.



Click on Startup Settings.



Click on the Restart button located at the bottom right.



Click on Disable driver signature enforcement (the 7th option) by pressing the F7 or number 7 key.

If Windows Security warning appears simply ignore it by clicking on Install this driver software anyway (the bottom choice).

④ Linux

ROBOTIS OpenCM9 software Linux release download

For users of 32-bit Linux OS download the 32-bit package. For 64-bit Linux user download the 64-bit package.

http://www.robotis.com/xe/download/638505

[Windows XP, Vista, 7, 8]
ROBOTIS_v0.9.8_win.zip

[Mac OS X] Tested in OS X 10.6.8
ROBOTIS_v0.9.8_osx.dmg

[Linux 64bit] Tested in Ubuntu 12.04
ROBOTIS_v0.9.8_linux64.tar

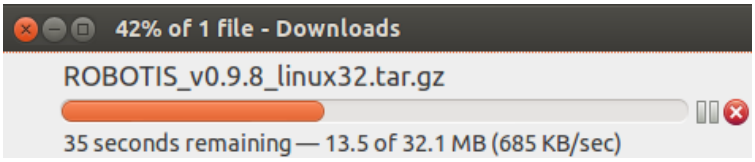
[Linux 32bit] Tested in Ubuntu 10.10
ROBOTIS_v0.9.8_linux32.tar

[Windows XP, Vista, 7, 8]
ROBOTIS_v0.9.8_win.zip

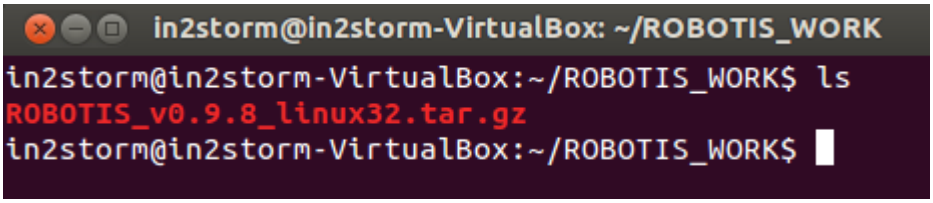
[Mac OS X] Tested in OS X 10.6.8
ROBOTIS_v0.9.8_osx.dmg

[Linux 64bit] Tested in Ubuntu 12.04
ROBOTIS_v0.9.8_linux64.tar

[Linux 32bit] Tested in Ubuntu 10.10
ROBOTIS_v0.9.8_linux32.tar



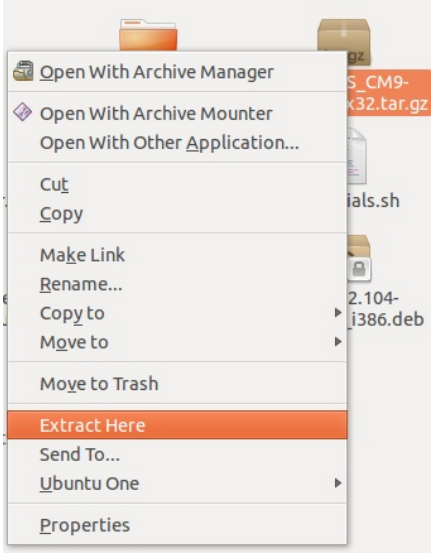
After downloading execute the command below.



~$tar –xvzf ROBOTIS_v0.9.8_linux32.tar.gz

OR

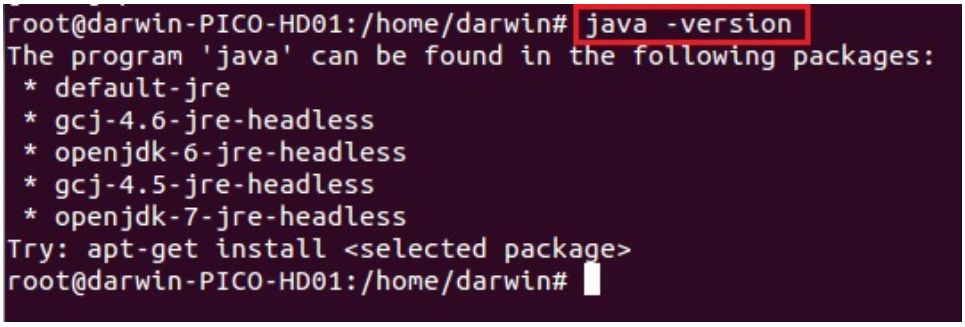
Use the mouse right click and select Extract here to decompress the file



Once decompress a folder named ROBOTIS appears.

A. Verify JRE installation.

For verification input the command as illustrated below "java –version"



If any of the packages shown above are installed, then get the JRE package via apt-get command "sudo apt-get install openjdk-7-jre".

This will install openjdk-7-jre-headless.

```
/$ sudo apt-get install openjdk-7-jre
```

Press the Y key to confirm.

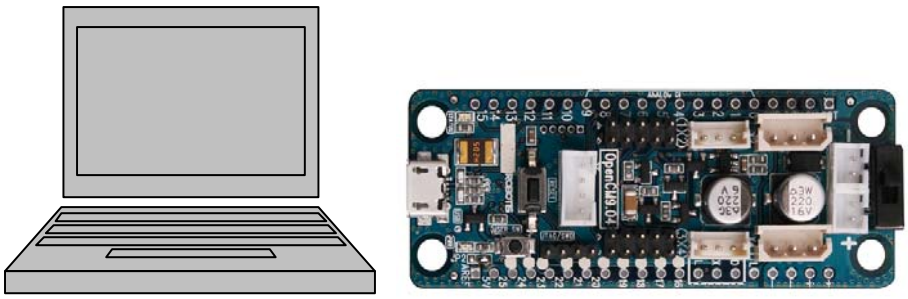
After installation is complete enter "java –version" command for verification.

```
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$ java -version
java version "1.7.0_15"
OpenJDK Runtime Environment (IcedTea7 2.3.7) (7u15-2.3.7-0ubuntu1~12.10.1)
OpenJDK Server VM (build 23.7-b01, mixed mode)
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$
```

Once JRE is installed ROBOTIS OpenCM can be executed.

B. Connect OpenCM-9.04 to PC

Connect the OpenCM9.04 to the PC via USB cable as illustrated below.



<OpenCM9.04 to PC connection>

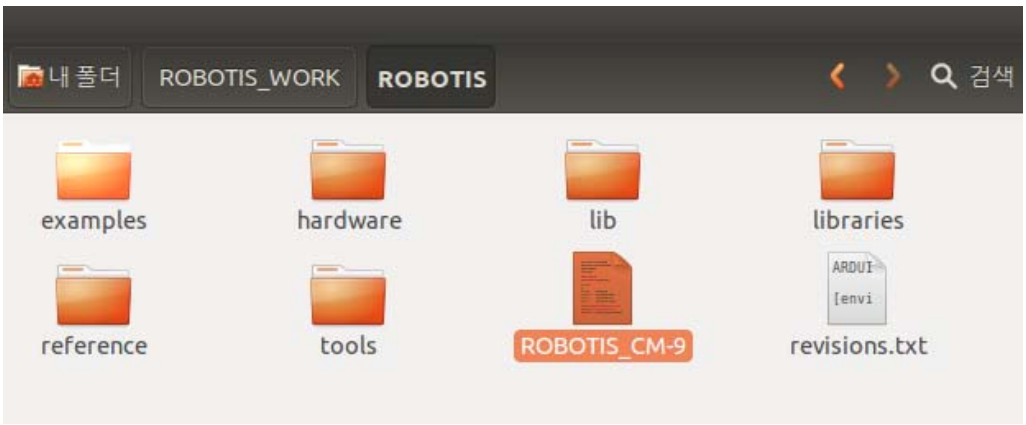
It is recommended to connect the OpenCM9.04 to the PC directly. However, when connecting the OpenCM9.04 to the PC via USB hub USB is it strongly recommended that the hub is powered by a dedicated power supply; this way there is enough power supply to the OpenCM9.04.

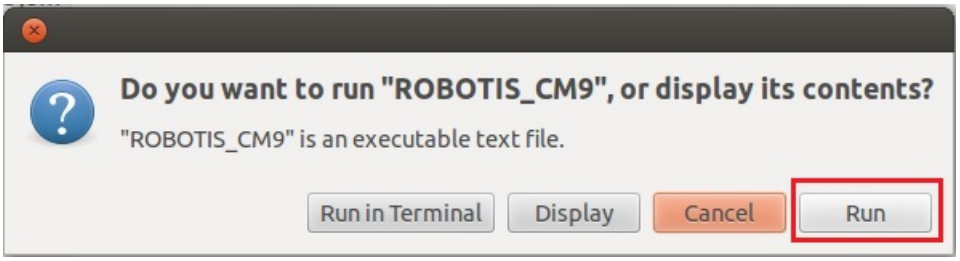
C. Run ROBOTIS OpenCM.

Double–click on ROBOTIS_CM-9

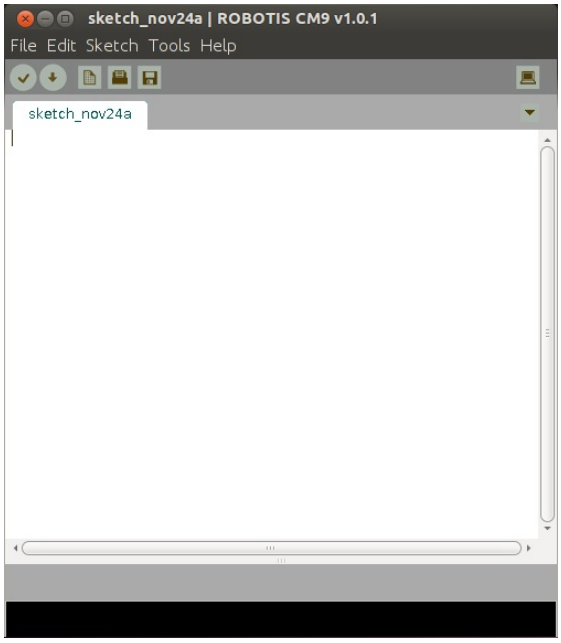
OR

On a terminal window input ./ROBOTIS_CM-9.

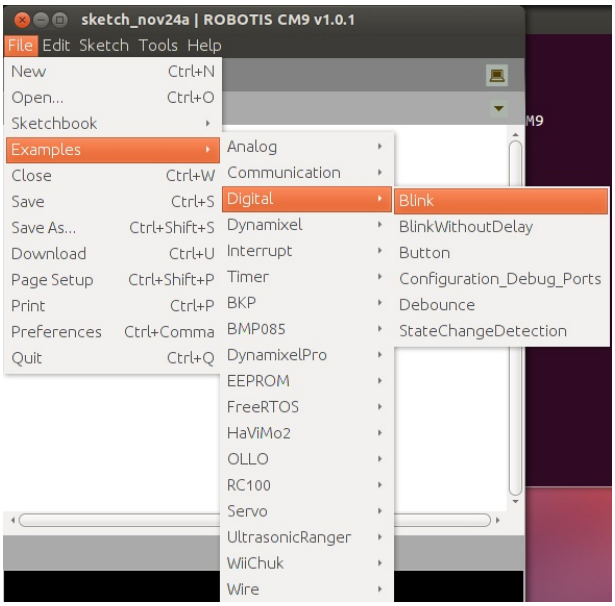




Click on Run

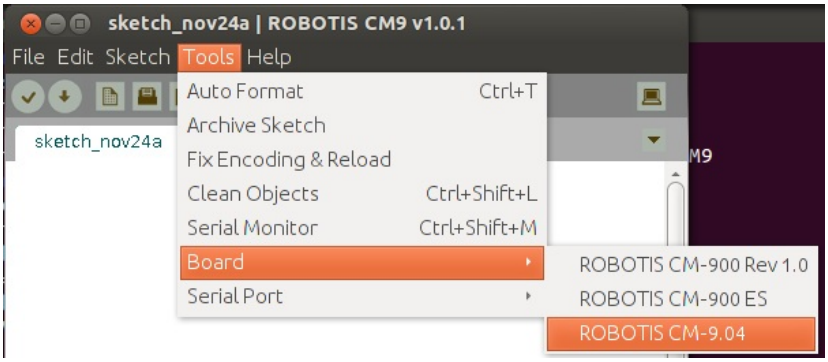


D. Open Blink example



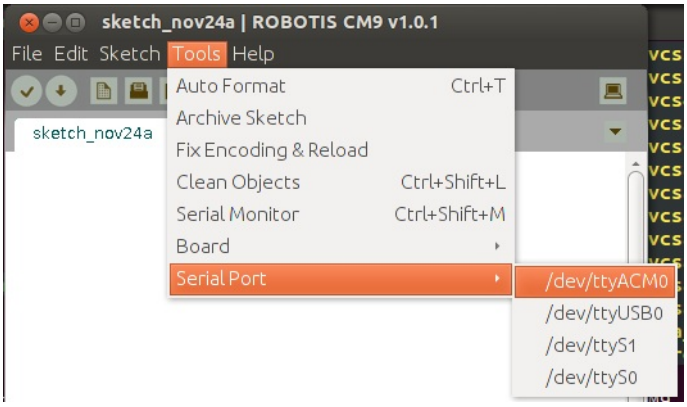
E. Select board.

Select the proper ROBOTIS OpenCM9.04 board.

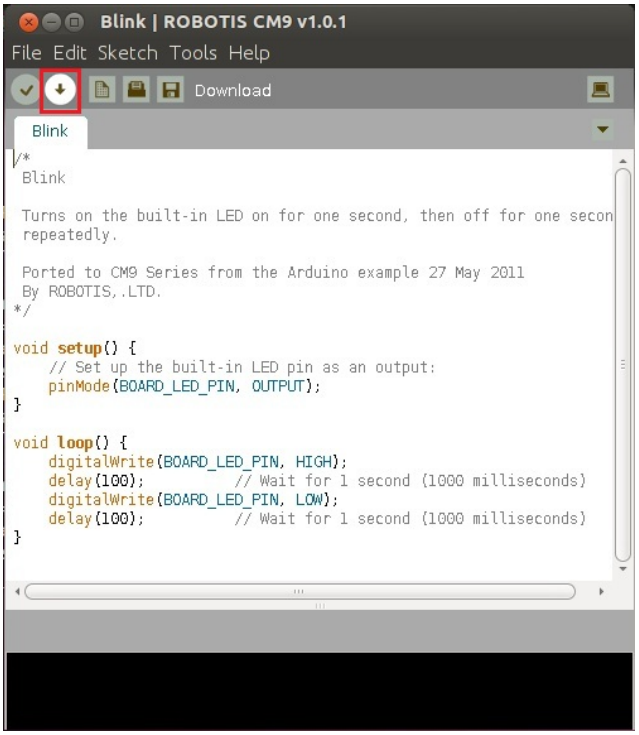


F. Select serial port.

Look for ttyACMX and ensure this option is checked.



G. Proceed to download.



⑤ Mac OS X

A. ROBOTIS OpenCM software Mac OS X release download

Download the Mac OSX package from the e-Manual page.

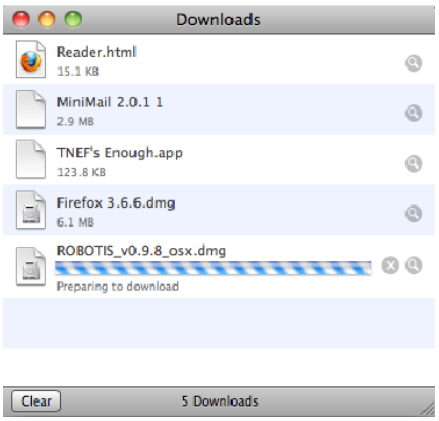
http://www.robotis.com/xe/download/638505

[Windows XP, Vista, 7, 8]
ROBOTIS_v0.9.8_win.zip

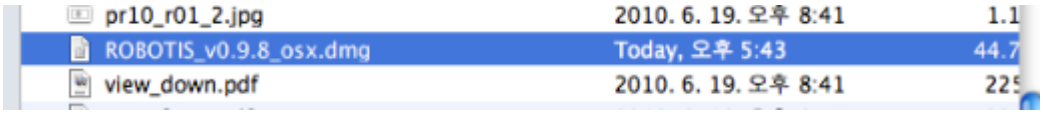
[Mac OS X] Tested in OS X 10.6.8
ROBOTIS_v0.9.8_osx.dmg

[Linux 64bit] Tested in Ubuntu 12.04
ROBOTIS_v0.9.8_linux64.tar

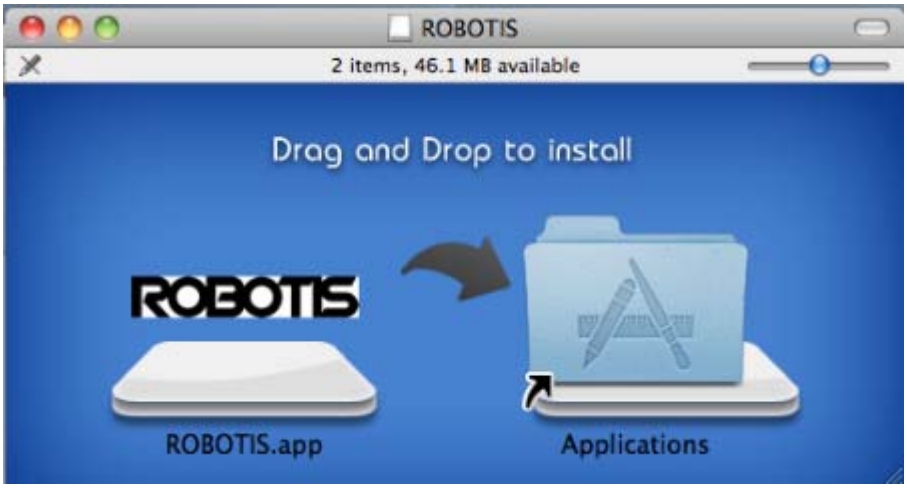
[Linux 32bit] Tested in Ubuntu 10.10
ROBOTIS_v0.9.8_linux32.tar



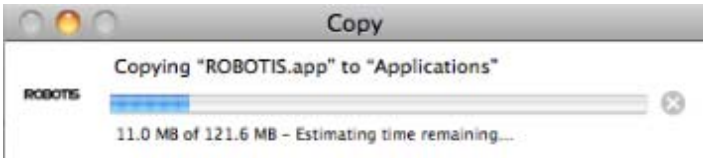
Once package is downloaded click mount the dmg file.



Simply drag the ROBOTIS app to the Applications folder to complete installation.



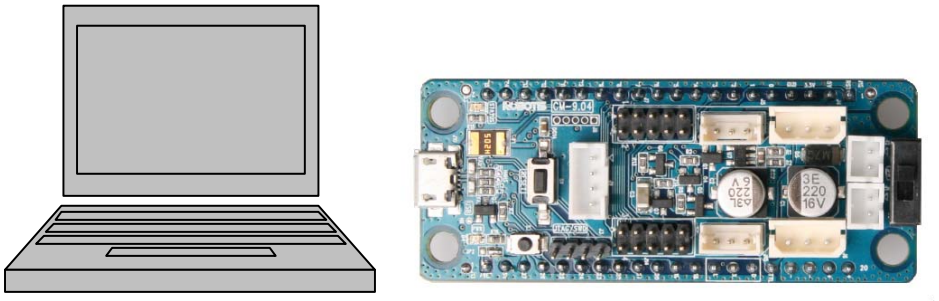
When progress bar is completely filled installation is complete.



Once installation is complete ROBOTIS.app will appears in the Applications folder.

B. Connect the OpenCM9.04 to PC.

Connect the OpenCM9.04 to the PC via USB cable as illustrated below.

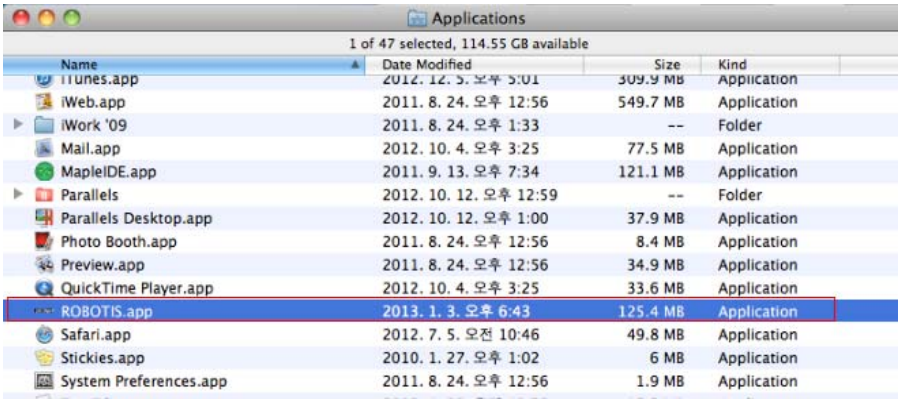


It is recommended to connect the OpenCM9.04 to the PC directly. However, when connecting the OpenCM9.04 to the PC via USB hub USB is it strongly recommended that the hub is powered by a dedicated power supply; this way there is enough power supply to

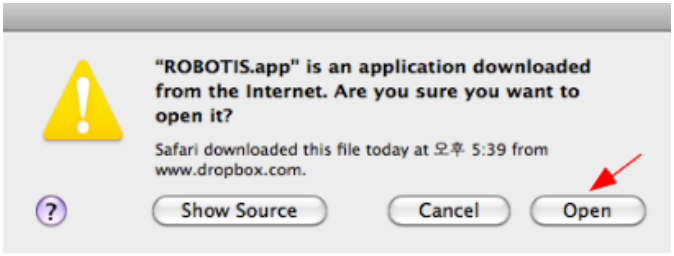
the OpenCM9.04.

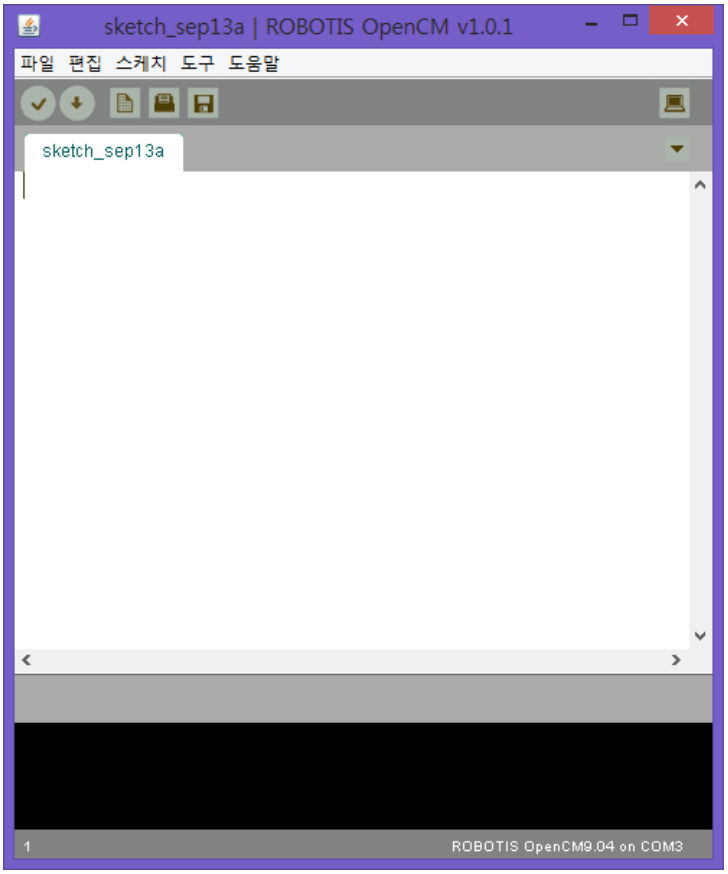
C. Run ROBOTIS OpenCM.

From Finder inside the Applications folder double-click on ROBOTIS.app.

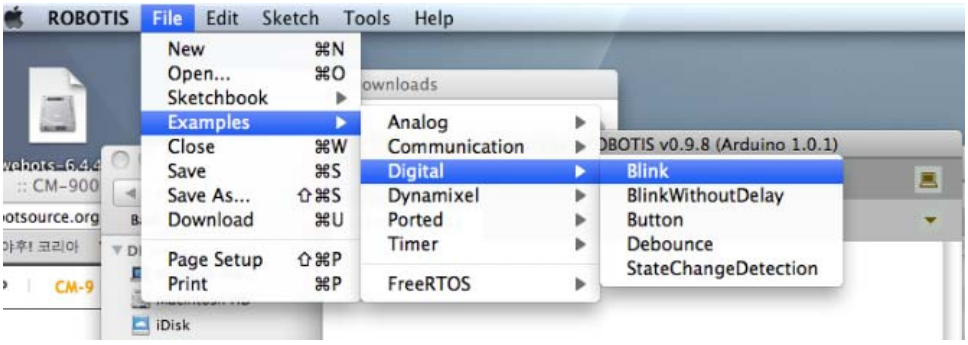


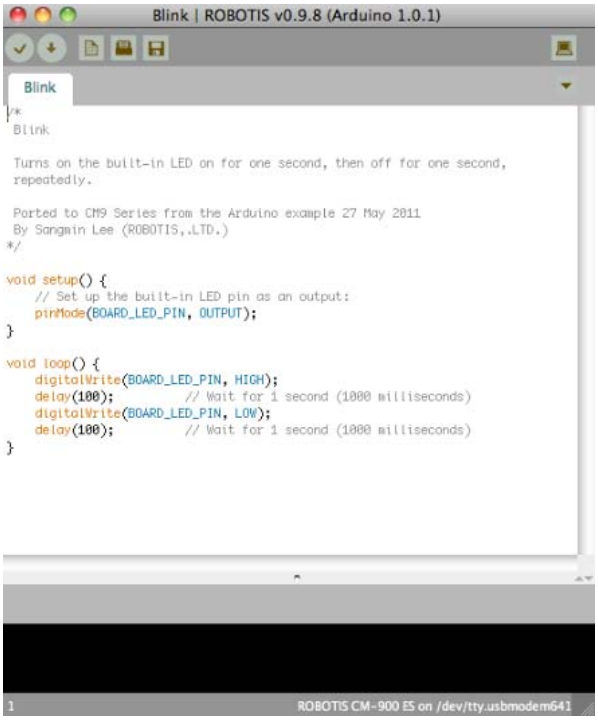
Click on Open



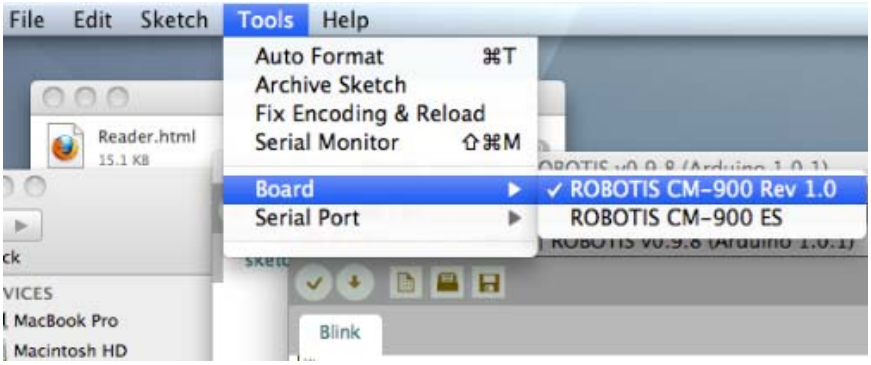


D. Open the Blink example

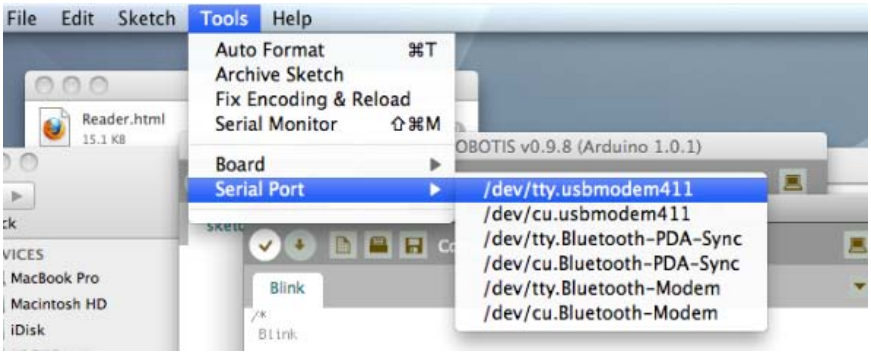




E. Select the proper board

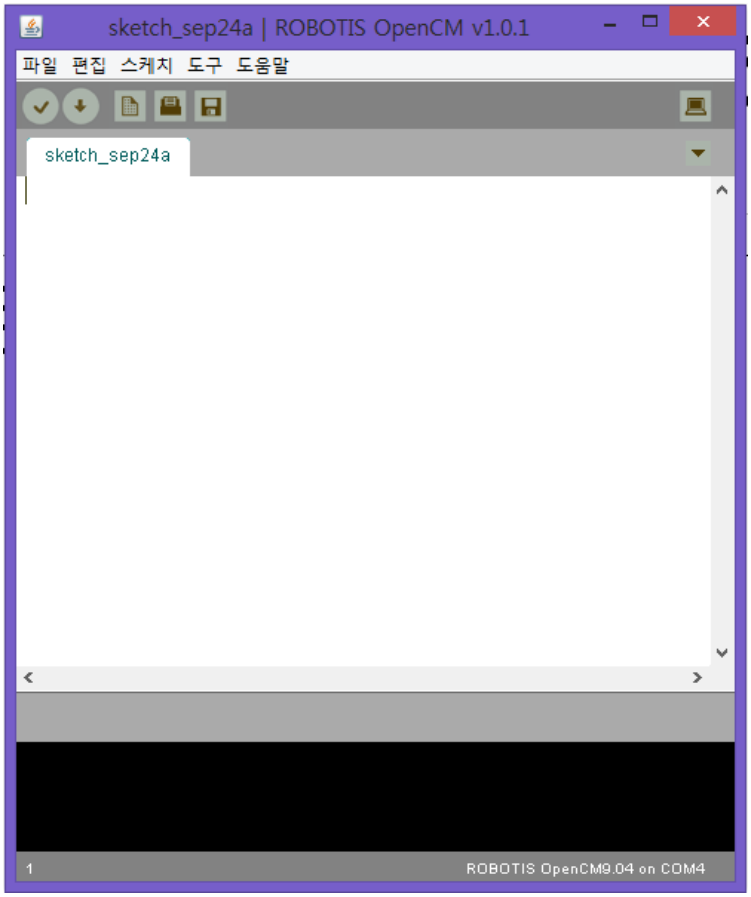


F. Select serial port



⑥ Setup software environment

Once USB drivers are properly set run ROBOTIS_OpenCM.exe. Linux and Mac OS X simply go straight to ROBOTIS OpenCM.

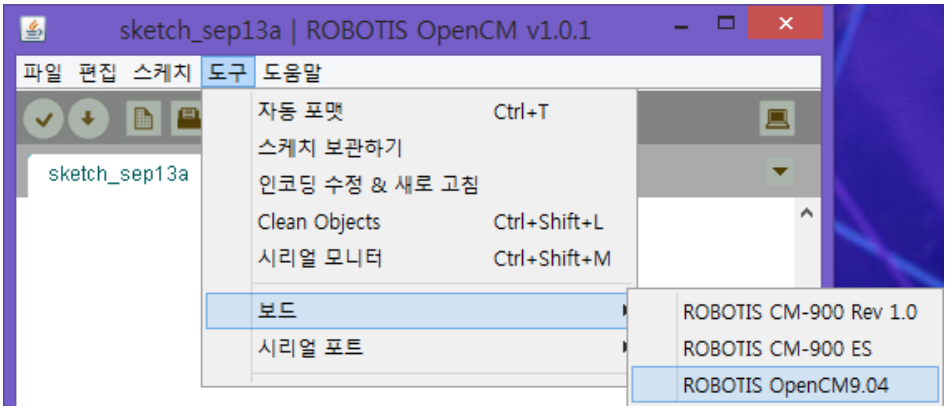


<ROBOTIS OpenCM>

For ROBOTIS OpenCM 2 items must always be checked (1) board type and (2) serial port.

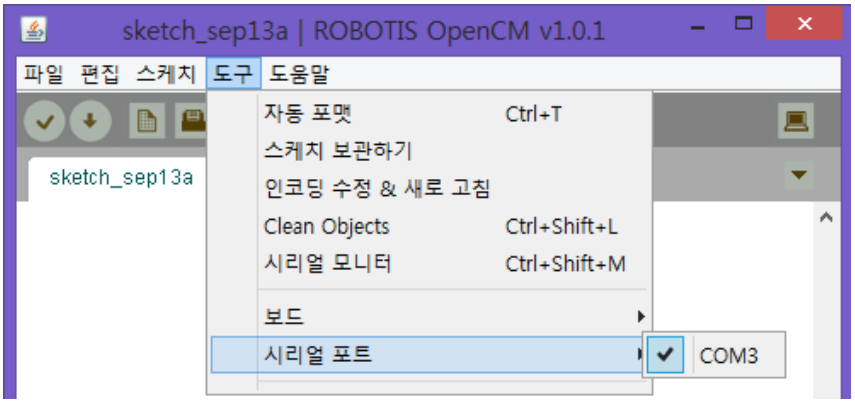
A. Board type

Select the proper OpenCM9.04 hardware version.

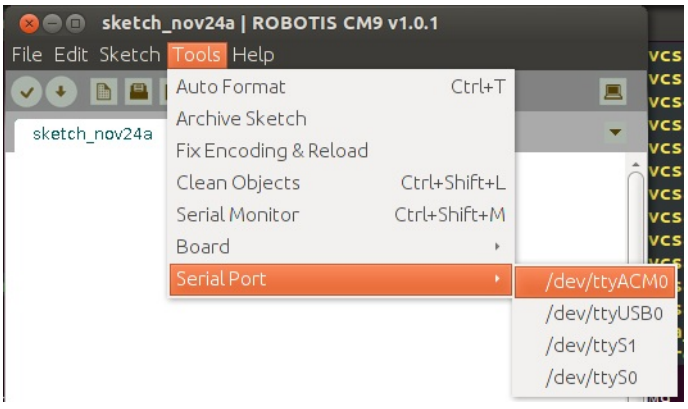


B. Serial port

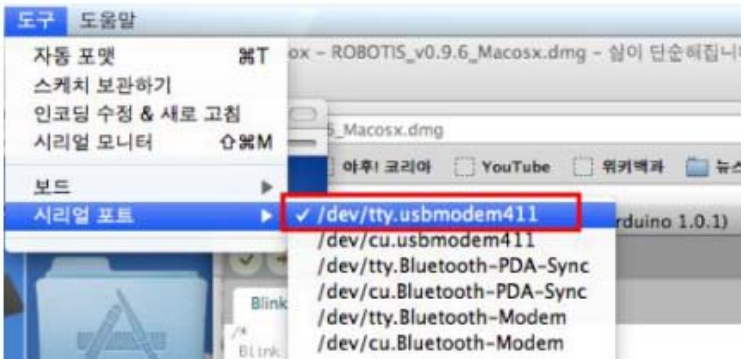
With USB drivers installed select the corresponding COM port number.



For Linux select /dev/ttyACMX.

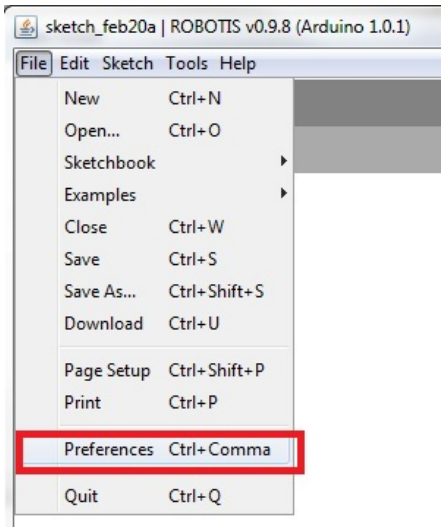


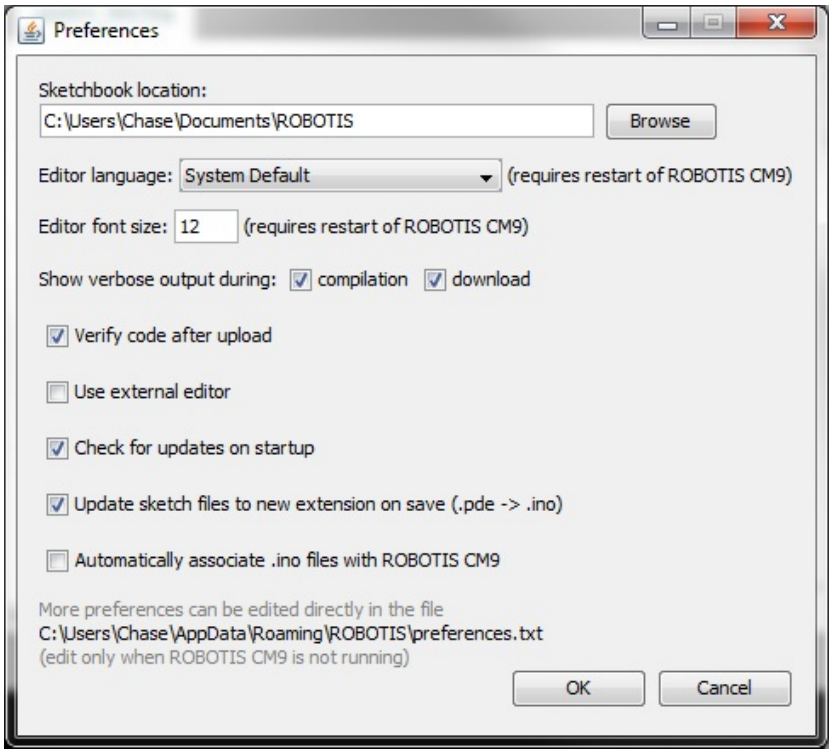
For Mac OS X select tty.usbmodemX11.



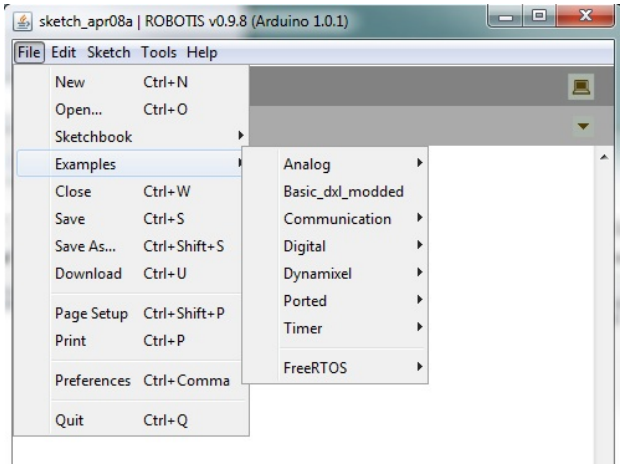
C. Environment setup

Go to File -> Preferences.





i. Sketchbook location: Path for user folder to store project files. Once this is set projects can be accessed directly via File -> Sketchbook.

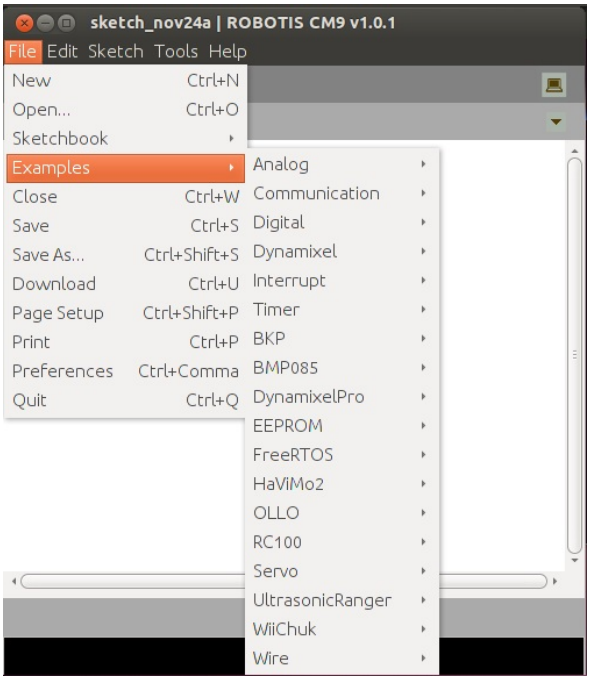


ii. Editor language: change fonts.

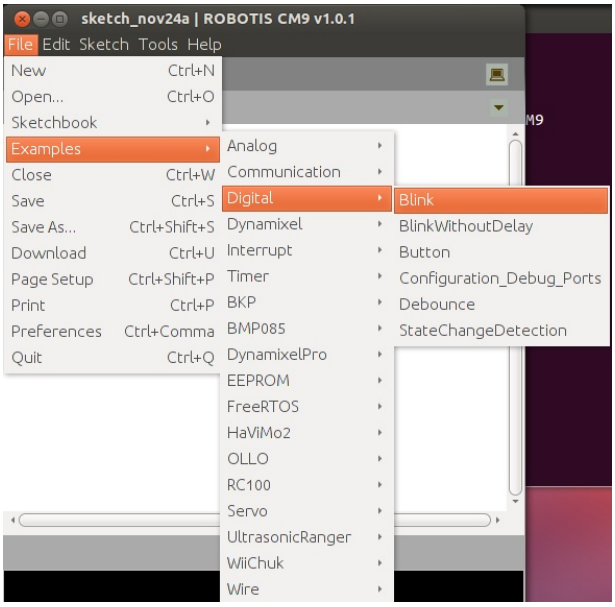
iii. Show verbose output during: allows to see progress during compilation or/and download.

⑦ Download Examples

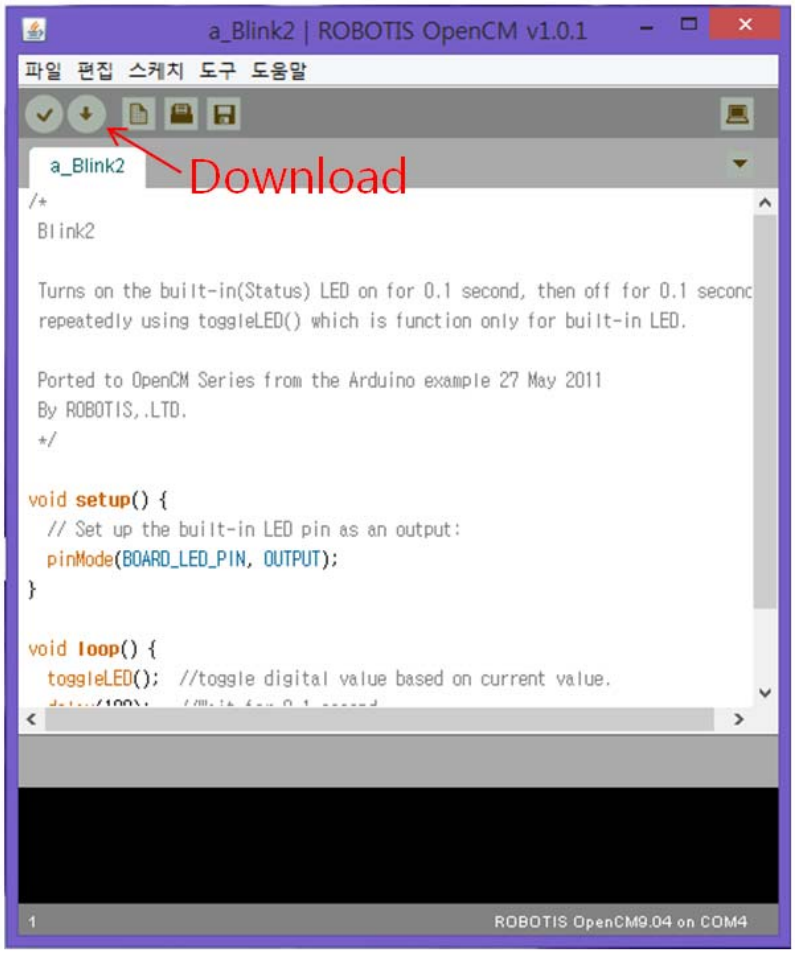
Examples can be easily accessed via File -> Examples.



For example, before opening Digital IO example perform a test download with the blink example to test the OpenCM9.04.



Once Blink is open click on the download arrow.

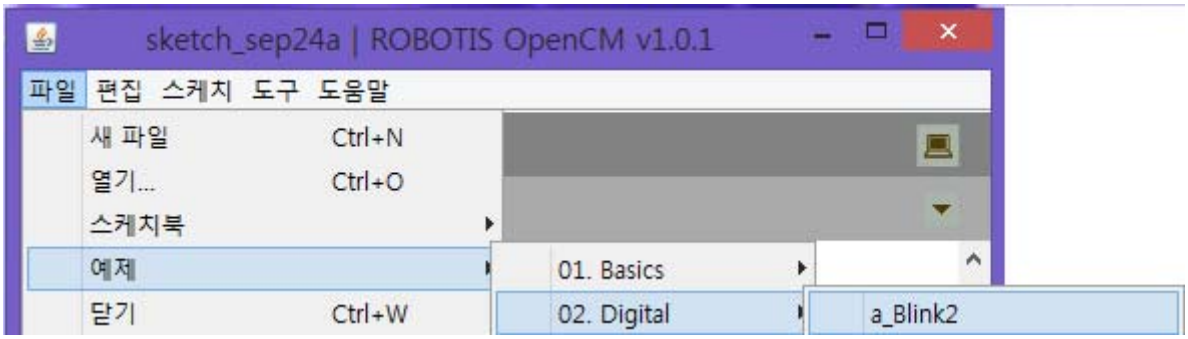


Refer to ROBOTIS OpenCM API Reference documentation (4 pages) and learning "Learning" about the OpenCM (5 pages).

⑧ Blink example

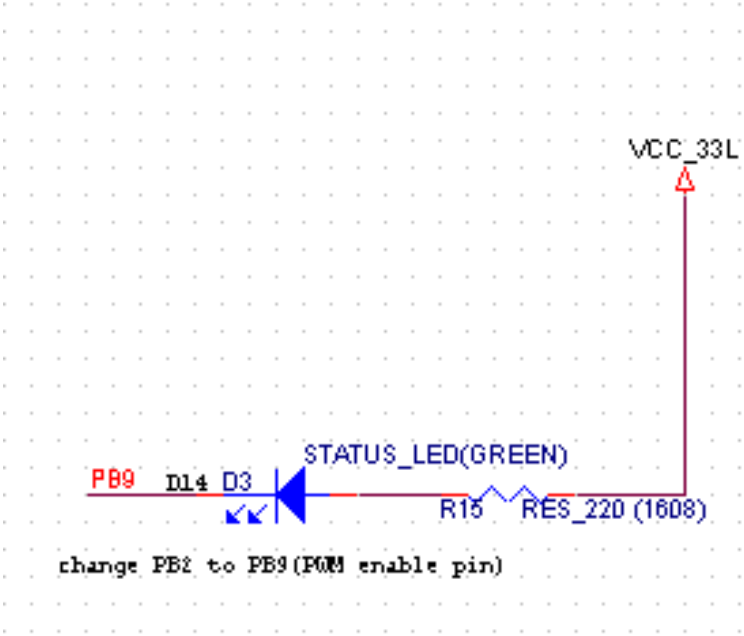
The OpenCM9.04's Blink example is a port of Arduino's Blink example.

Go to File -> examples -> Digital -> a_Blink2.



A. Schematic

The OpenCM9.04 status LED connects to the CPU via D14(PB9).



When D14(PB9) is high the LED is off; when low the LED is on.

B. Sketch code

```
void setup() {
    // Set up the built-in LED pin as an output:
    pinMode(BOARD_LED_PIN, OUTPUT);
}

void loop() {
    digitalWrite(BOARD_LED_PIN, HIGH);
    delay(100);          // Wait for 1 second (1000 milliseconds)
    digitalWrite(BOARD_LED_PIN, LOW);
    delay(100);          // Wait for 1 second (1000 milliseconds)
}
```

The function **pinMode(pin_number, pin_mode)** function is used to unitialize.

Refer to the CM-900 I/O port silk screen; BOARD_LED_PIN is defined for pin D14. This is illustrated in the header file CM-900.h.

ROBOTIS\hardware\robotis\cores\robotis\CM-900.h

ROBOTIS₩hardware₩robotis₩cores₩robotis₩OpenCM

```
#ifndef OpenCM904_H_
#define OpenCM904_H_

//#include "gpio.h"


#define CYCLES_PER_MICROSECOND  72
#define SYSTICK_RELOAD_VAL      71999 /* takes a cycle to reload */
/*
 * [ROBOTIS][CHANGE] CM-900 Do not have built-in button.
 * 2013-04-22
 * */
#define BOARD_BUTTON_PIN        25//38
#define BOARD_LED_PIN           14//16
#define BOOT1_PIN       28 // added
```

The Blink example is a simple high/low signal manipulator with OUTPUT being the output fuction.

Once setup() function has been set you can control the LED with **digitalWrite(pin_number, HIGH/LOW)**를 in the loop() via time with delay(millisecond).

C. Data verification

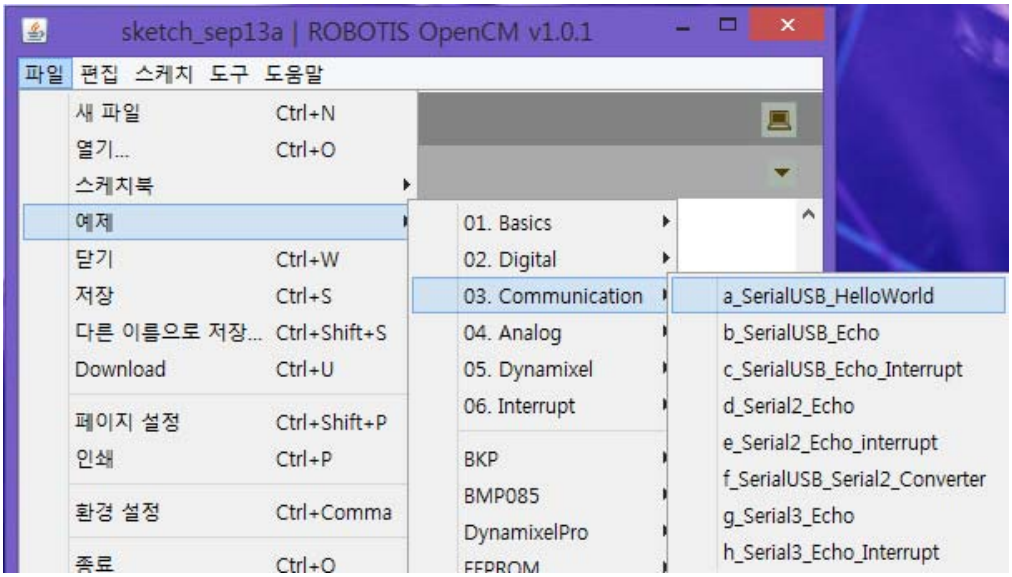
Check whether STATUS LED is on or off.

⑨ SerialUSB_HelloWorld example

This is an example to communicate between the OpenCM9.04 and external device (i.e. PC) via USB. Declare SerialUSB instance to enable USB communications.

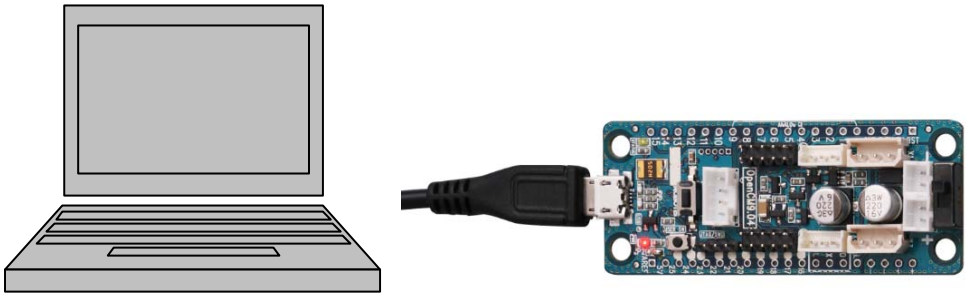
This example show how SerialUSB_HelloWorld communicated with a terminal window (PC).

Go to File -> Examples -> Communication -> SerialUSB_HelloWorld.



A. Connect the OpenCM9.04 to the PC

Connect the OpenCM9.04 to the PC via USB cable as illustrated below.



B. Sketch code

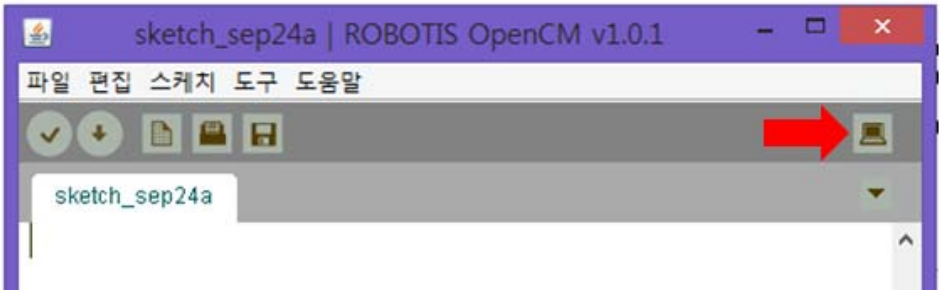
```
void setup() {
  //Initialize USB Serial
  SerialUSB.begin();
}
int nCount=0;
void loop() {
    //print "Hello World!!" to PC though USB Virtual COM port
    SerialUSB.println("Hello World!!");
    SerialUSB.print("nCount : "); // display nCount variable and increase
    SerialUSB.println(nCount++);
    delay(1000);
}
```

Initialize SerialUSB instance in Setup() with begin() method. The void() type returns nothing. Regardless of other serial devices with SerialUSB.begin() method setting the baud rate is not necessary.

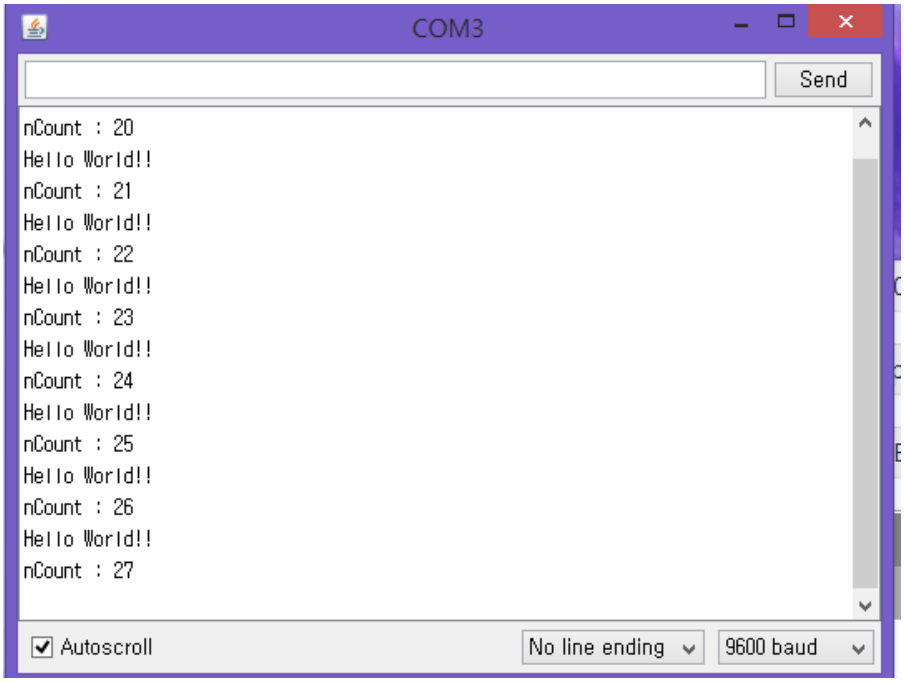
In Loop() with SerialUSB.print() or SerialUSB.println() its possible to get output

C. Verify data

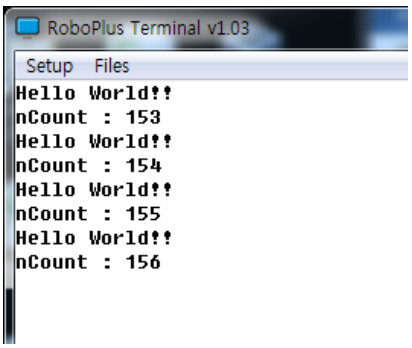
Click on the serial monitor to see output. This is also possible with RoboPlus Terminal.



The serial monitor window can be activated by clicking on the laptop icon located on the upper right side.



The same is possible with RoboPlus Terminal (no need to set baud rate)..



Other terminal window applications are not yet supported.

⑩ SerialUSB_Echo example

SerialUSB_HelloWorld example only showed output SerialUSB_Echo example allows for both input and output.

A. Sketch code

```
void setup(){
  //USB Serial initialize
  SerialUSB.begin();
}
void loop(){
  // when you typed any character in terminal
  if(SerialUSB.available()){
      //print it out though USB
      SerialUSB.print((char)SerialUSB.read());
  }
}
```

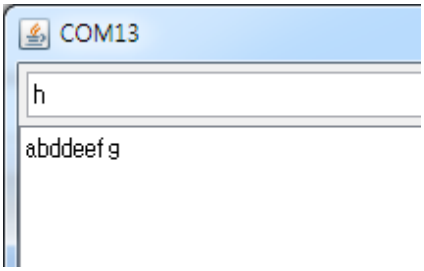
Like SerialUSB_HelloWorld there is no need to set baud rate in SerialUSB.begin().

In Loop() the CPU checks for input repeatedly. In the if clause SerialUSB.avaliable() outputs 0 until the condition is met. Once condition is met SerialUSB.read() sends 1 byte SerialUSB.print().

.

B. Verify data

Use the serial monitor or RoboPlus Termincal to view data. Use the keyboard to input data and the OpenCM9.04.



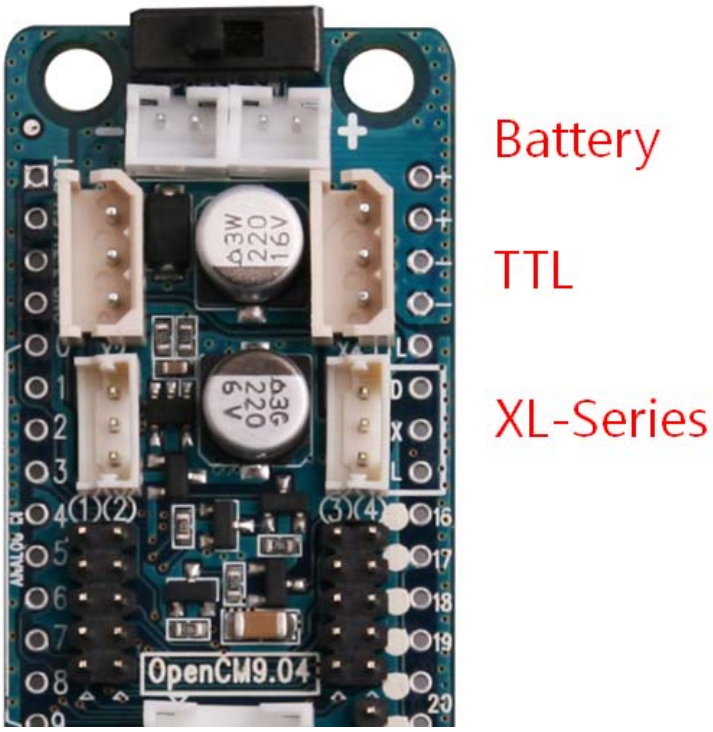
Any keyboard input is returned exactly as output.

⑪ Dynamixel Basic example

The OpenCM9.04 includes Dynamixel connectors to facilitate robot development. A pair of 3-pin TTL, and a pair a XL-series connector are embedded onto the board. Also, a battery with either 12V or 7.4V with the proper connector is also required to power any connected Dynamixel(s).

Dynamixel Basic example is analogous to the Blink example as it switches Dynamixel between one position to another.

.



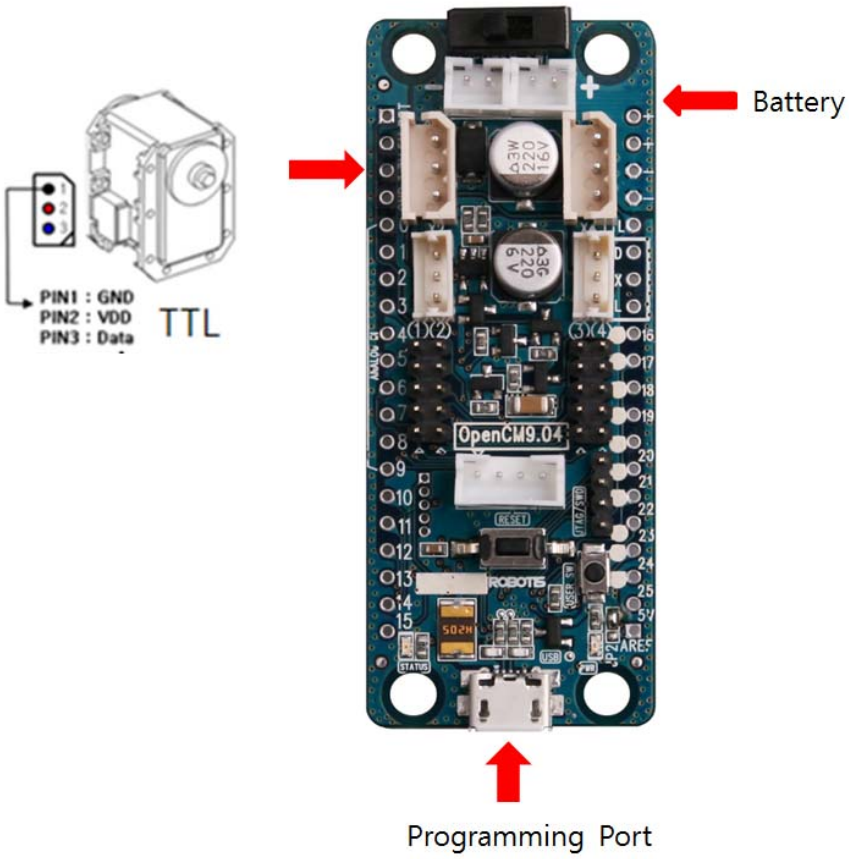
A. Connecting a Dynamixel

Connect a 3-pin Dynamixel. Connect the SMPS or battery to the OpenCM9.04 then run ROBOTIS OpenCM9.

The default values for Dynamixel are 1 for ID and 1 for baud rate (1Mbps). If not, then set said values with Dynamixel Wizard.

The OpenCM9.04 communicates with Dynamixel serially via Serial1 (USART1).

.



B. Sketch code

```
void setup() {
    // Initialize the dynamixel SDK:
    Dxl.begin(1);
}

void loop() {
    delay(1000);              // Wait for 1 second (1000 milliseconds)
    Dxl.writeWord(1, 30, 100); //Turn dynamixel ID 1 to position 100
    delay(1000);              // Wait for 1 second (1000 milliseconds)
    Dxl.writeWord(1, 30, 1000);//Turn dynamixel ID 1 to position 1000
}
```

Dynamixel bus must be initialized. From setup() Dxl.begin(baud_rate) is also initialized. From here any 3-pin or 4-pin Dynamixel device connected to the OpenCM9.04 gets initialized. Baud_rate value of 1 means communications speed is set to 1Mbps. For further information on Dynamixel API please consult the e-manuals.

From loop() with Dxl.writeWord(ID, Address, Value) function set the value for goal position(L) in Address; this corresponds to position portion of Dynamixel, and Value being the value of the position. In this example the position switches

between 100 to 1000 in intervals of 1000ms.

.

C. Verify data

The only way to verify data is to check motor movement visually.

⑫ Dynamixel ReadWrite example

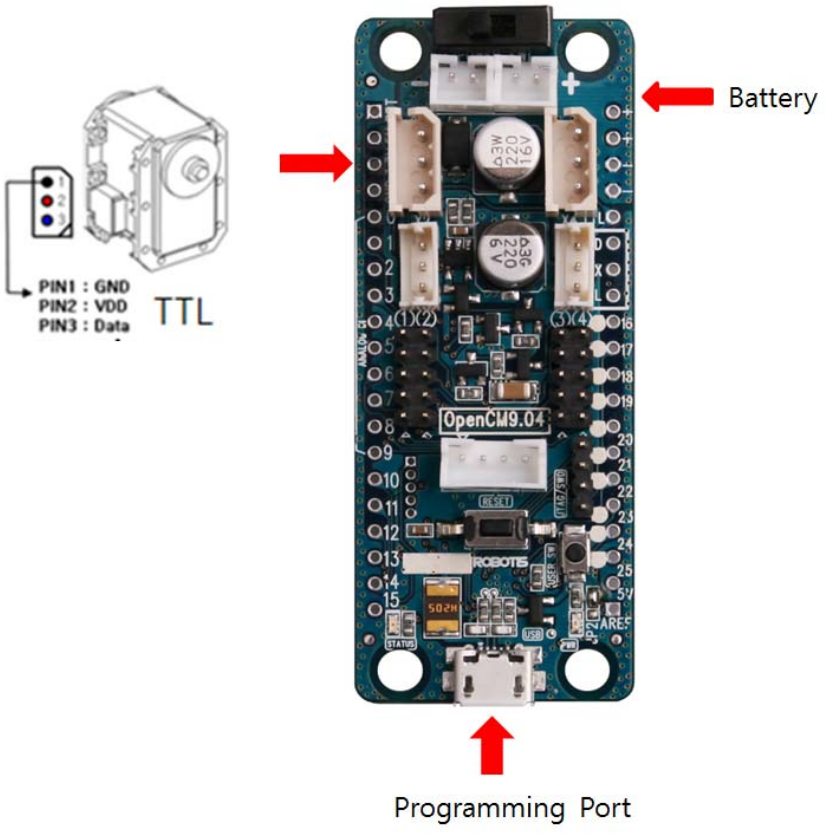
This example shows Dynamixel read/write features. This example checks Dynamixel movement and change of moving (rotating) direction. Once moving is complete position data is then outputted and Dynamixel moves to the next position.

A. Connect Dynamixel

Connect a 3-pin or 4-pin DYnamixel. Connect the SMPS or battery to the OpenCM9.04 then run ROBOTIS OpenCM.

The default values for Dynamixel are 1 for ID and 1 for baud rate (1Mbps). If not, then set said values with Dynamixel Wizard.

The OpenCM9.04 communicates with Dynamixel serially via Serial1 (USART1)



B. Sketch code

Some of the parameters from Dynamixel control table have been defined in the preprocessor for simplicity.

.

```
#define P_GOAL_POSITION_L     30
#define P_PRESENT_POSITION_L  36
#define P_MOVING              46

word  Position;
word  wPresentPos;
byte  INDEX = 0;
byte  bMoving, CommStatus;
byte  id = 1;
word  GoalPos[2] = {0, 1023};

void setup() {
  Dxl.begin(1);
  //print to USB port
  SerialUSB.begin();
}

void loop() {
  bMoving = Dxl.readByte( id, P_MOVING);
  CommStatus = Dxl.getResult();
  if( CommStatus == COMM_RXSUCCESS ){
    if( bMoving == 0 ){
      // Change goal position
      if( INDEX == 0 )
        INDEX = 1;
      else
        INDEX = 0;
      // Write goal position
      Dxl.writeWord( id, P_GOAL_POSITION_L, GoalPos[INDEX] );
    }
    // Read present position
    wPresentPos = Dxl.readWord( id, P_PRESENT_POSITION_L );
    SerialUSB.print("Goal Position : ");
    SerialUSB.println(GoalPos[INDEX]);
    SerialUSB.print("Present position :");
    SerialUSB.println(wPresentPos);
    SerialUSB.println("Success");
  }else {
    SerialUSB.println("Fail");
  }
  delay(1000);
}
```

bMoving = Dxl.readByte(id, P_MOVING) returns a 1 when Dynamixel is moving and 0 when not. If transmission via Dxl.getResult() is successful and

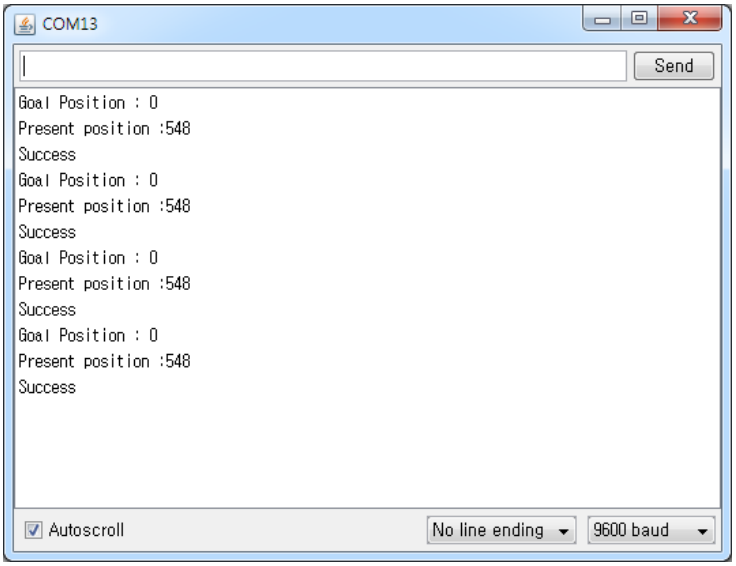
bMoving = 0 the Goal Position's index changes; Dxl.writeWord(id, P_GOAL_POSITION, GoalPos[INDEX]) transmits new data. Value from GoalPos[INDEX] is outputted via USB via the following command

wPresentPos = Dxl.readWord(id, P_PRESENT_POSITION_L);

;

C. Verify data

Open up serial monitor to see output from GoalPos[INDEX] and see position of Dynamixel visually.



⑬ Dynamixel SyncWrite example

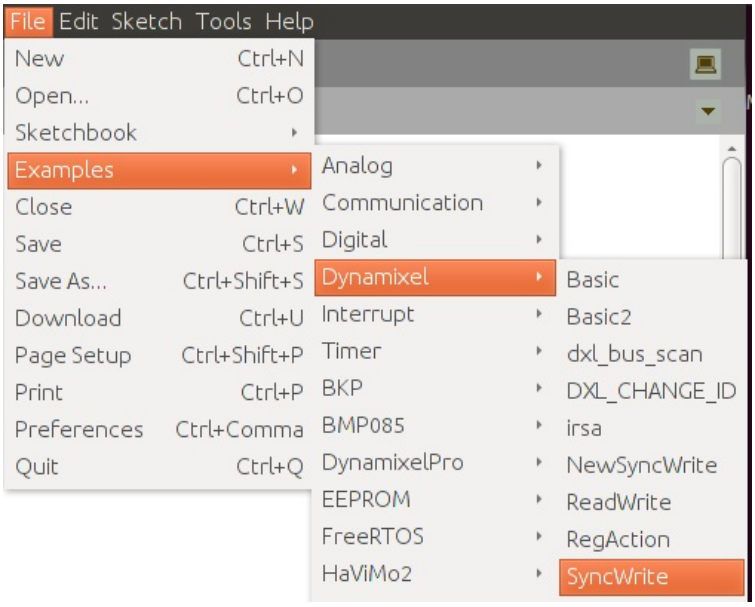
With Dynamixel Broadcast ID its possible to control multiple Dynamixels simultaneously.

This example shows how to control 5 Dynamixels via Syncwrite packet. For more information on Syncwrite please consult the e-manuals.

http://support.robotis.com/ko/e-manual_kor.htm#product/dynamixel/communication/dxl_instruction.htm

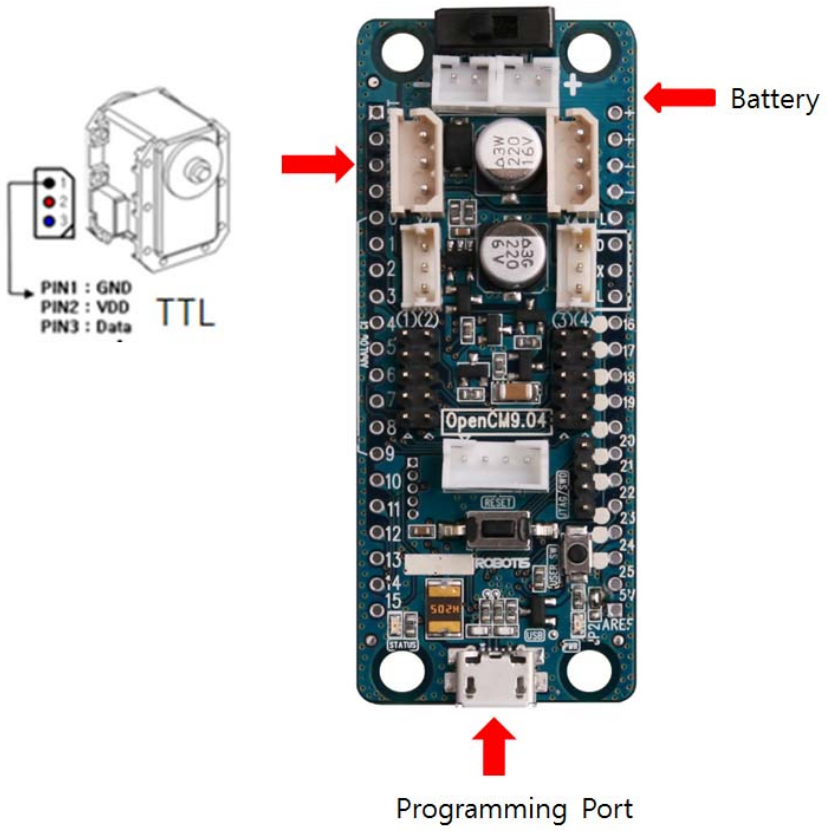
Go to FIle -> Examples -> Dynamixel -> SyncWrite

.



A. Connect Dynamixel

Set ID from 1 to 5 use either 3-pin or 4-pin Dynamixel, or a combination of 5 using both pin types; connect them in any order. Set baud rate to 1Mbps to all 5 Dynamixels.



The OpenCM9.04 communicates with Dynamixel serially via Serial1 (USART1).

B. Sketch code

Some of the parameters from Dynamixel control table have been defined in the preprocessor. For more information on Dynamixel control table please consult the e-manuals.

Note that 1-byte Word LOW (LSBs) is enough for control.

```
#define P_GOAL_POSITION_L    30
#define P_GOAL_SPEED_L       32

#define NUM_ACTUATOR         5 // Number of actuator
#define MAX_POSITION         1023
```

AmpPos is the initial position of all 5 Dynamixels.

```
word  AmpPos = 512;
word  wPresentPos;
word  GoalPos = 0;
byte  id[NUM_ACTUATOR];
byte  CommStatus;
byte  i;
```

Initial position

Dynamixel bus initialized in setup() with Dxl.begin(1) along with SerialUSB.begin().

.

```
void setup() {
  Dxl.begin(1);
  SerialUSB.begin();
  //Insert dynamixel ID number to array id[]
  for(i=0; i<NUM_ACTUATOR; i++ ){
    id[i] = i+1;
  }
  // Set goal speed
  Dxl.writeWord( BROADCAST_ID, P_GOAL_SPEED_L, 0 );
  // Set goal position
  Dxl.writeWord( BROADCAST_ID, P_GOAL_POSITION_L, AmpPos );
  delay(1000);
}
```

In loop() a Syncwrite packet can be divided for Dynamixel communications and output. For packet creation instructions please consult the e-manuals.

ID 0XFE
Length (L+1) X N + 4 (L:RX-64별 Data Length, N:RX-64의 개수)
Instruction 0X83
Parameter1 Data를 쓰고자 하는 곳의 시작 Address
Parameter2 쓰고자 하는 Data의 길이 (L)
Parameter3 첫 번째 RX-64의 ID
Parameter4 첫 번째 RX-64의 첫 번째 Data
Parameter5 첫 번째 RX-64의 두 번째 Data
…
Parameter L+3 첫 번째 RX-64의 L번째 Data
Parameter L+4 두 번째 RX-64의 ID
Parameter L+5 두 번째 RX-64의 첫 번째 Data
Parameter L+6 두 번째 RX-64의 두 번째 Data
…
Parameter 2L+4 두 번째 RX-64의 L번째 Data

일반적으로 동시 제어 가능한 다이나믹셀은, 1개의 명령 패킷이 4바이트인 경우 26개까지 가능합니다.
다이나믹셀의 수신버퍼 용량이 143byte 이므로 Packet의 길이가 143byte를 초과하지 않도록 하십시오.

Please note a word (2 bytes) in a Dynamixel packet includes both High byte (MSBs) and Low byte word (LSBs).

```
void loop() {
// Make syncwrite packet
  Dxl.setTxPacketId(BROADCAST_ID);                                  1
  Dxl.setTxPacketInstruction(INST_SYNC_WRITE);                      2
  Dxl.setTxPacketParameter(0, P_GOAL_POSITION_L);                   3
  Dxl.setTxPacketParameter(1, 2);                                   4

  for( i=0; i<NUM_ACTUATOR; i++ ){
    Dxl.setTxPacketParameter(2+3*i, id[i]);                         5
    Dxl.setTxPacketParameter(2+3*i+1, Dxl.getLowByte(GoalPos));
    Dxl.setTxPacketParameter(2+3*i+2, Dxl.getHighByte(GoalPos));    6

    SerialUSB.println(GoalPos);                                     7
  }
  Dxl.setTxPacketLength((2+1)*NUM_ACTUATOR+4);                      8
  Dxl.txrxPacket();                                                 9

  CommStatus = Dxl.getResult();
  //SerialUSB.print("CommSatus = ");SerialUSB.println(CommStatus);
  if( CommStatus == COMM_RXSUCCESS ){
    PrintCommStatus(CommStatus);
  }
  else{
    PrintErrorCode();
  }
  GoalPos += 100;                    Report result of CommStatus

  if( GoalPos > MAX_POSITION )
    GoalPos -= MAX_POSITION;
  delay(CONTROL_PERIOD);

}
```

#1: Syncwrite Packet set to Broadcast ID.

#2: set Instruction Sync Write (0x83)

#3: Goal Position parameter with value 0.

#4: assign a word (2 bytes) to Goal Position.

#5: Assignment for IDs and Parameters

(data length +1)*(index value i=0,1,2,…) + 2(BROADCAST_ID, INST_SYNC_WRITE)

#6: set word (2 bytes) for Goal Position.

#7: output goal position via USB

#8: calculates Packet length (see below)

Length (L+1) X N + 4 (L:RX-64별 Data Length, N:RX-64의 개수)

#9: the created Packet is transmitted via Dxl.txrxPacket() method

.

C. Verify data

Open the serial monitor to see GoalPos[INDEX] of all 5 Dynamixels.