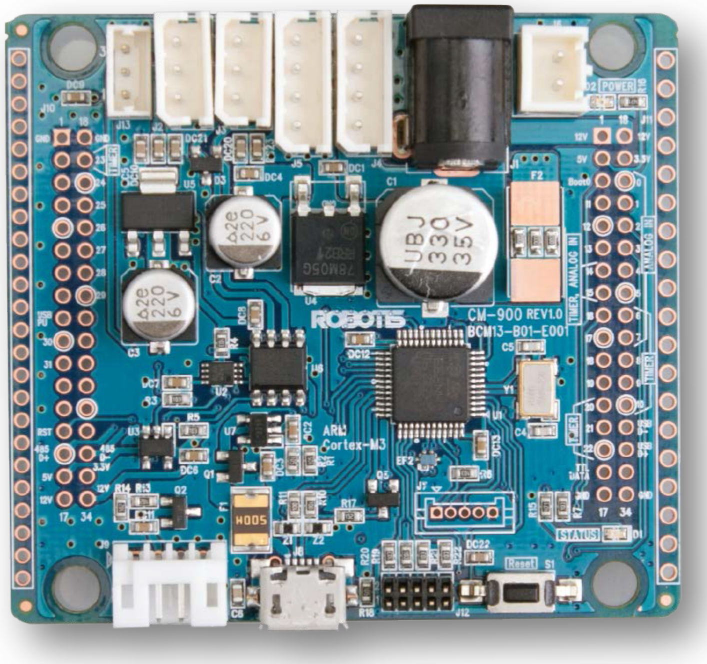
CM-900 Manual



ROBOTIS
#1505 Ace High End No.3, 371-50
Gasandong, Geumcheongu, Seoul, KR
TEL : +82-70-8671-2600
E-mail : contactus2@robotis.com

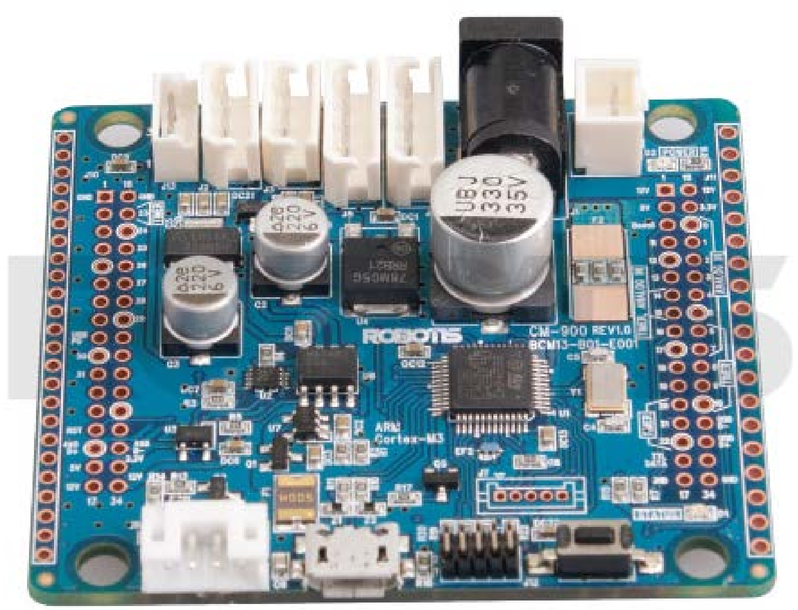
CM-900 Manual

Contents

Overview

CM-900 is an embedded board based on STMicroelectronics' STM32F103C8 Cortex-M3 MCU(Datasheet). The board has 32 pins (16 data pins, 10 analog input pins), 5 dedicated ports for Dynamixel (2 TTL, 2 RS-485, 1 for LX-series); a micro-USB type B port for programming and communications; DC power jack and 2-pin battery connector, reset switch; and a JTAG header.

The CM-900 hardware and software are open-source; support in Windows, Mac OSX, and Linux for convenient and easy development of robots



Note: the CM-900 is not compatible with RoboPlus, use ROBOTIS CM-9 for development.

Ⅰ Sales presentation

1 Getting started with the CM-900

Let’s have a look at the CM-900. With your working PC follow these step by step instructions.

1.1 Windows

1.1.1 Have the CM-900 and USB Cable ready.

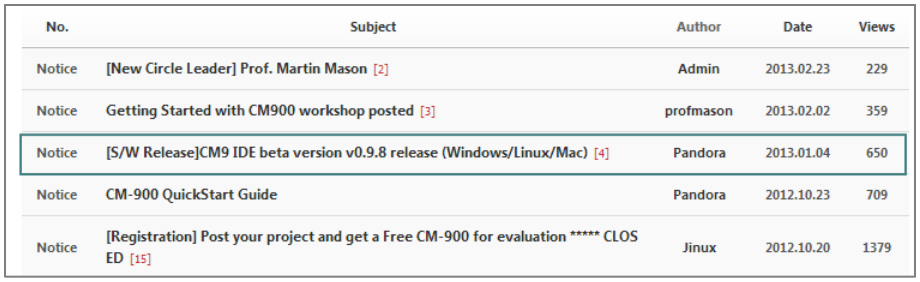
The cable is a type B micro-USB; the same type as with most smartphones.

1.1.2 Download ROBOTIS CM9 development environment

Download the most recent version of ROBOTIS CM9. You can get the most recent version by clicking on the link below.

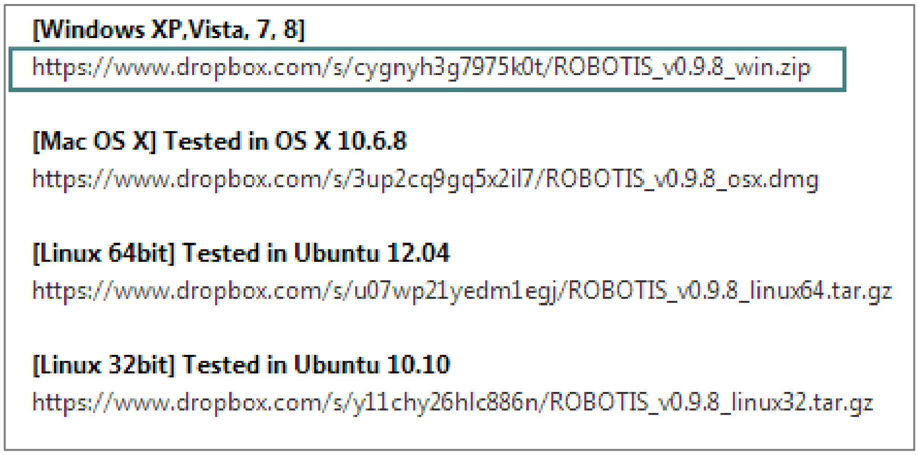
http://www.robotsource.org/xe/Circle_CM9_Developer_World

Look for “Notice” entries.



No.	Subject	Author	Date	Views
Notice	[New Circle Leader] Prof. Martin Mason [2]	Admin	2013.02.23	229
Notice	Getting Started with CM900 workshop posted [3]	profmason	2013.02.02	359
Notice	[S/W Release]CM9 IDE beta version v0.9.8 release (Windows/Linux/Mac) [4]	Pandora	2013.01.04	650
Notice	CM-900 QuickStart Guide	Pandora	2012.10.23	709
Notice	[Registration] Post your project and get a Free CM-900 for evaluation ***** CLOSED [15]	Jinux	2012.10.20	1379

The Windows-version release is shown on the image below; click to download the compressed file.



[Windows XP,Vista, 7, 8]
https://www.dropbox.com/s/cygnyh3g7975k0t/ROBOTIS_v0.9.8_win.zip

[Mac OS X] Tested in OS X 10.6.8
https://www.dropbox.com/s/3up2cq9gq5x2il7/ROBOTIS_v0.9.8_osx.dmg

[Linux 64bit] Tested in Ubuntu 12.04
https://www.dropbox.com/s/u07wp21yedm1egj/ROBOTIS_v0.9.8_linux64.tar.gz

[Linux 32bit] Tested in Ubuntu 10.10
https://www.dropbox.com/s/y11chy26hlc886n/ROBOTIS_v0.9.8_linux32.tar.gz

Download the compressed file on your computer. Decompress the file and run ROBOTIS CM-9; the USB folder (\drivers) will also appear.

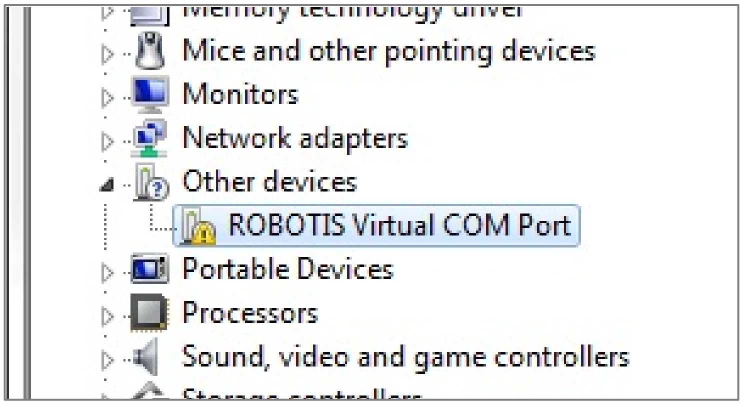
1.1.3 Connect the CM-900 to the PC.
Connect the CM-900 to the PC with the USB cable.

<connect the CM-900 to the PC>

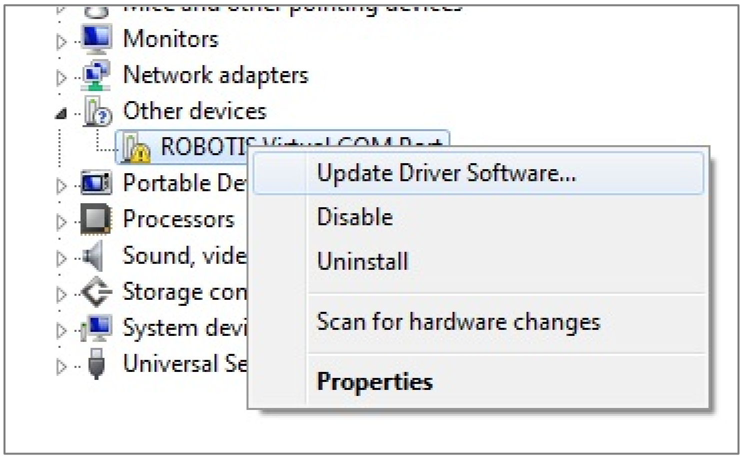
Please refrain from connecting the CM-900 to the PC via USB hub. We recommend connecting the CM-900 to the PC directly. The USB hub may not be able to provide enough electrical current to the CM-900 to properly download programs. Connect the CM-900 to an USB port with guaranteed enough electrical current supply.

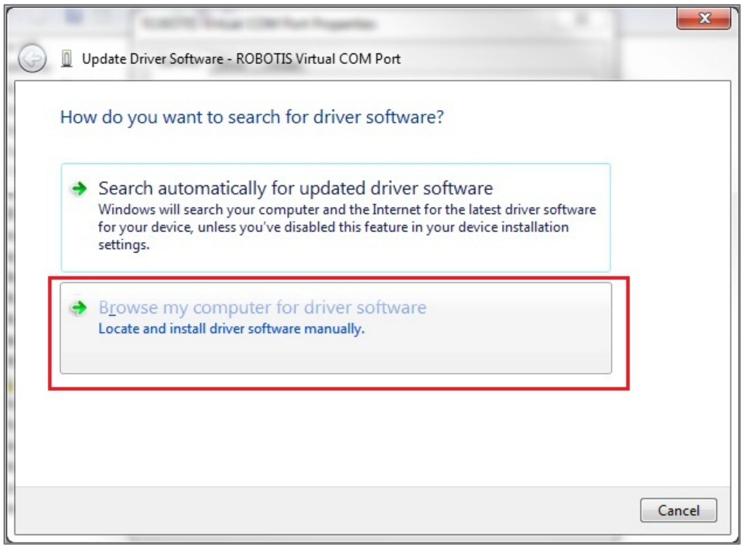
1.1.4 Install device drivers.

The CM-900 will appear as “ROBOTIS Virtual COM Port” when connected to the PC in Windows Device Manager.



Click with the right mouse button -> choose “Update Driver Software”




Update Driver Software - ROBOTIS Virtual COM Port

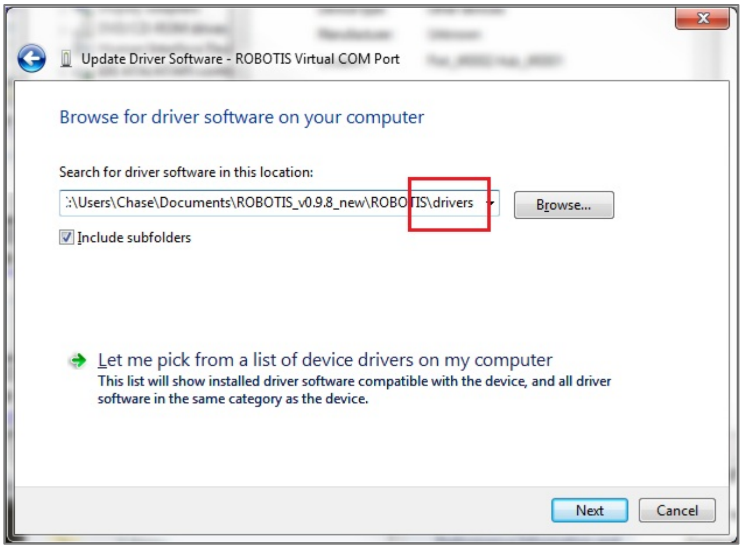
How do you want to search for driver software?

Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.

Select “Browse my computer for driver softeware”.

Choose “Manually search for drivers.”



Update Driver Software - ROBOTIS Virtual COM Port

Browse for driver software on your computer

Search for driver software in this location:

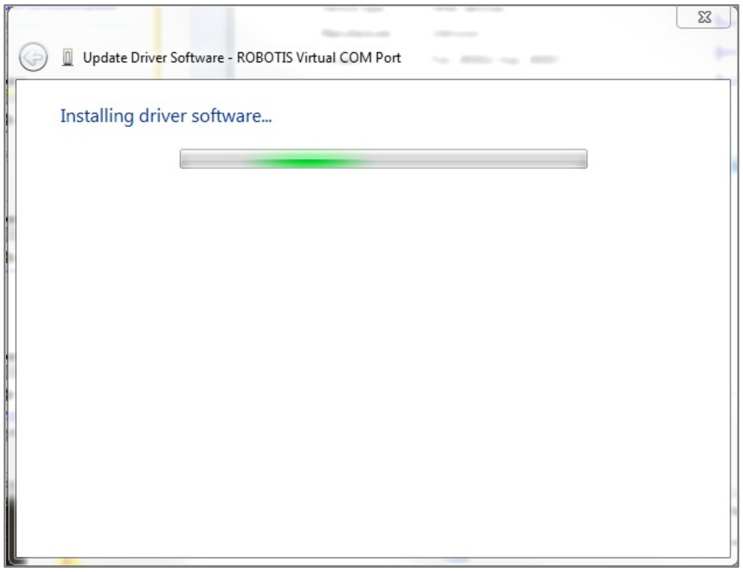
:\Users\Chase\Documents\ROBOTIS_v0.9.8_new\ROBOTIS\drivers

☑ Include subfolders

Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

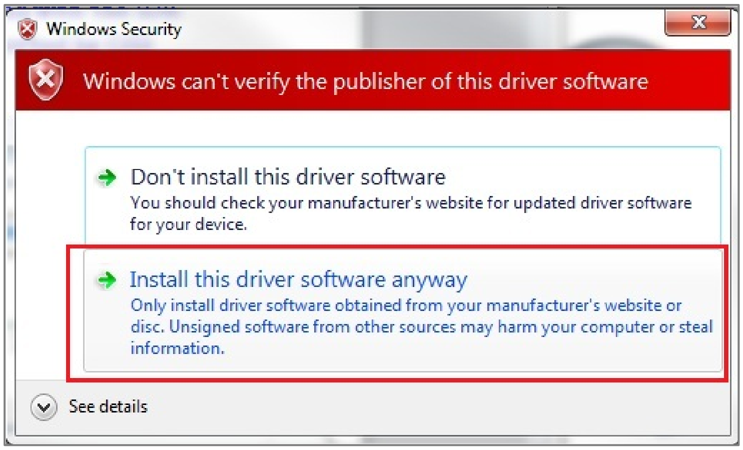
Click on “search” and go to (ROBOTIS\drivers) directory.



Update Driver Software - ROBOTIS Virtual COM Port

Installing driver software...

During driver installation you may encounter the following message; simply click on “install drivers anyways.”



Windows Security

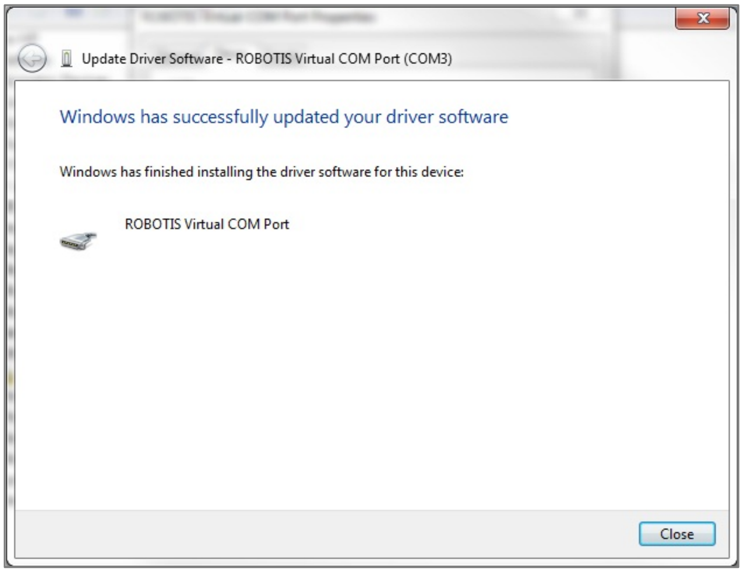
Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

See details

Upon successful installation you will see “successfully updated software driver.”



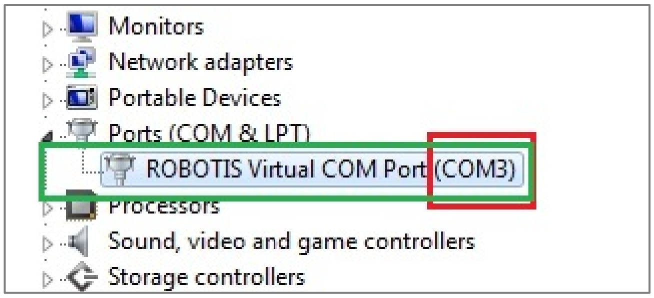
Update Driver Software - ROBOTIS Virtual COM Port (COM3)

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

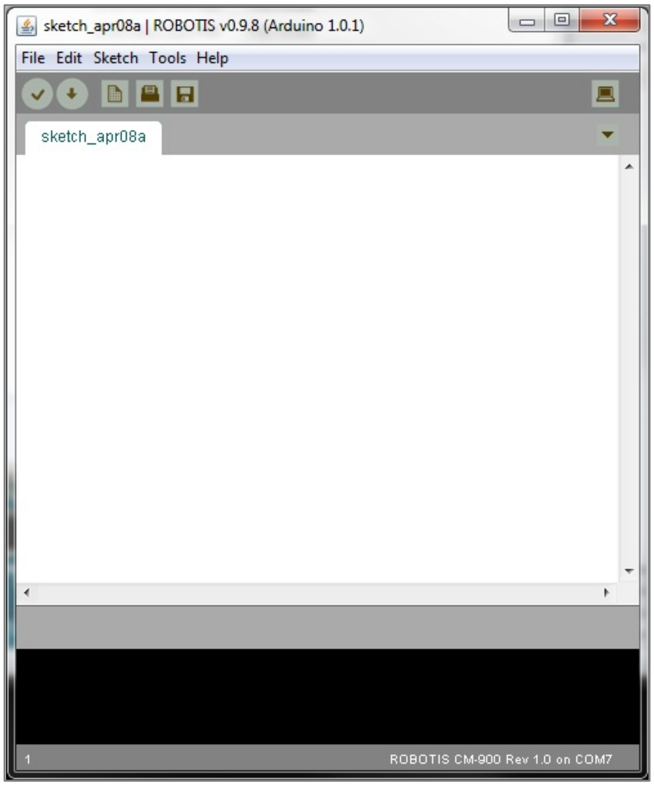
ROBOTIS Virtual COM Port

From the device manager always remember the port number from ROBOTIS Virtual COM Port. If connecting the device to another USB port then the number may change



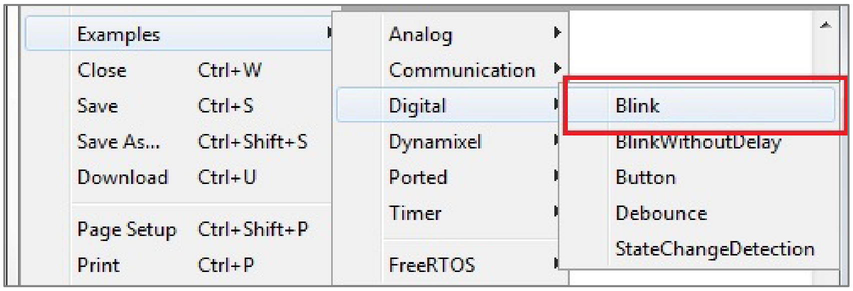
1.1.5 Run ROBOTIS CM-9.

From the decompressed file directory (\ROBOTIS) double-click on ROBOTIS CM-9.exe.


sketch_apr08a | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
sketch_apr08a
1
ROBOTIS CM-900 Rev 1.0 on COM7

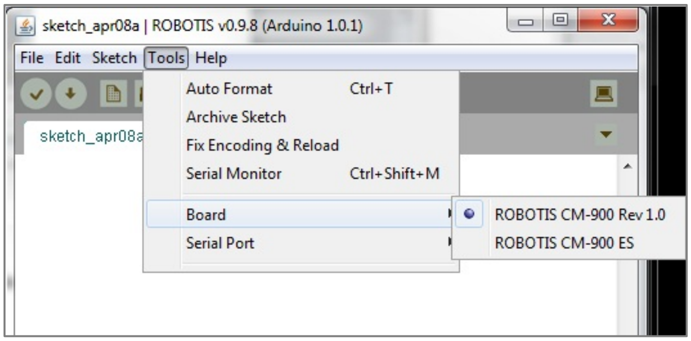
1.1.6 Open the Blink example.

FIle -> Examples -> Digital -> Blink



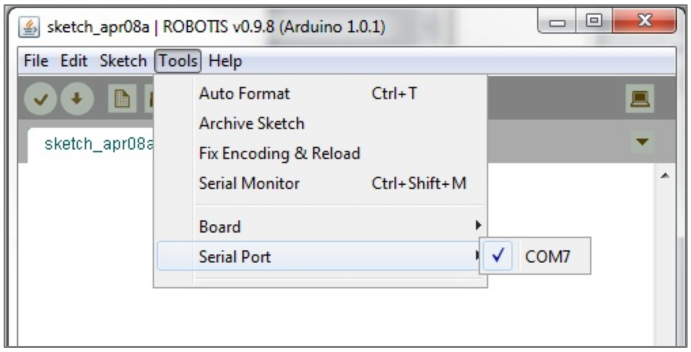
1.1.7 Select a board.

Tools ->Board-> ROBOTIS CM-900 Rev 1.0

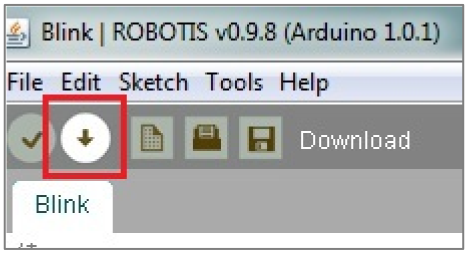


1.1.8 Select serial port.

This is the same number from the Virtual COM device in Windows Device Manager.


sketch_apr08a | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
sketch_apr08a
✓ COM7

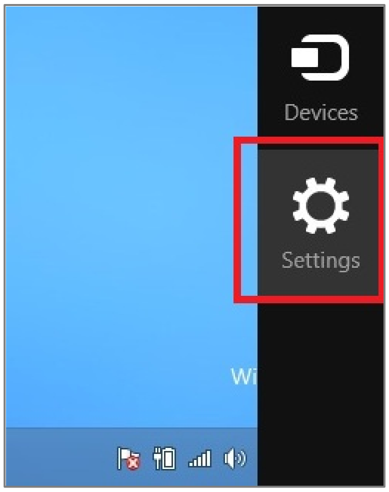
1.1.9 Download the code


Blink | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Download
Blink

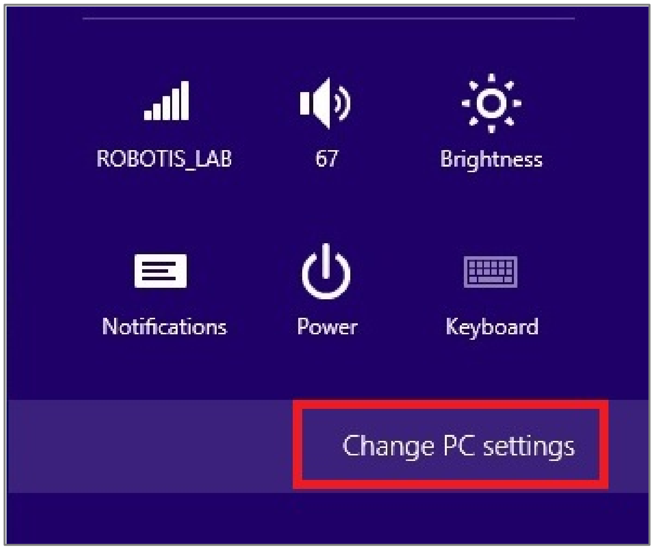
1.1.10 Troubleshooting Windows 8 USB driver installation

Under Advanced options in PC settings select “do not enforce driver signature” then install the CM-900 USB drivers.

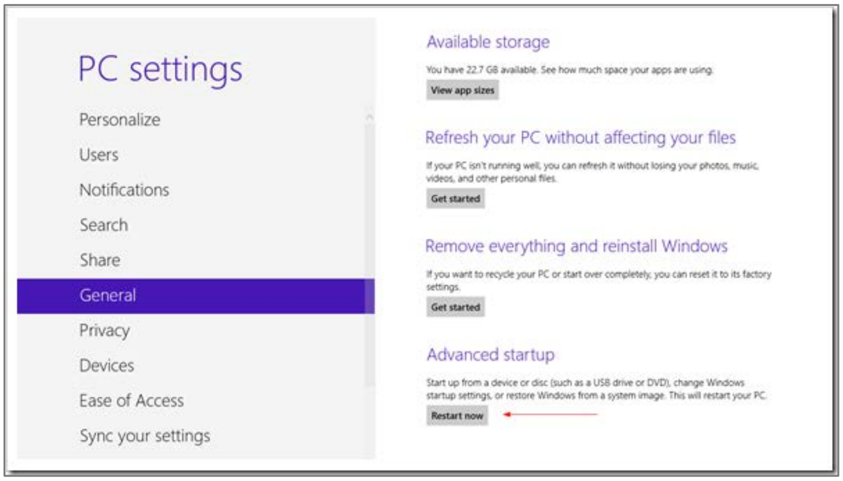
Move the mouse pointer to the upper right side of the screen. When the menu pops click on Settings



Click on Change PC settings located at the bottom right.



Click on General and Select advanced startup (click on Restart now button).



Available storage

You have 22.7 GB available. See how much space your apps are using.

Refresh your PC without affecting your files

If your PC isn't running well, you can refresh it without losing your photos, music, videos, and other personal files.

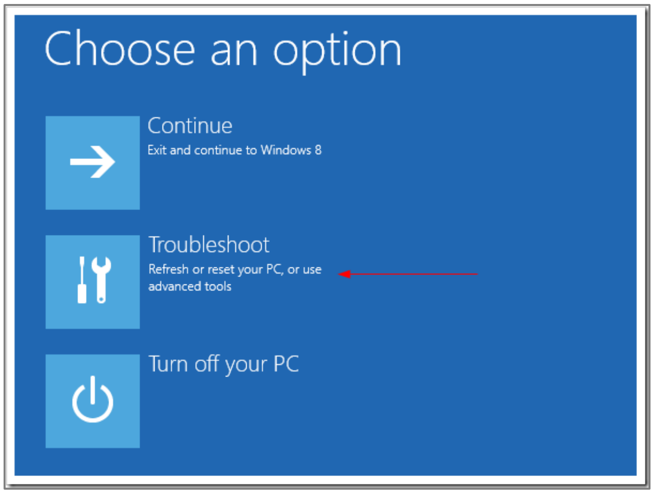
Remove everything and reinstall Windows

If you want to recycle your PC or start over completely, you can reset it to its factory settings.

Advanced startup

Start up from a device or disc (such as a USB drive or DVD), change Windows startup settings, or restore Windows from a system image. This will restart your PC.

Select troubleshoot



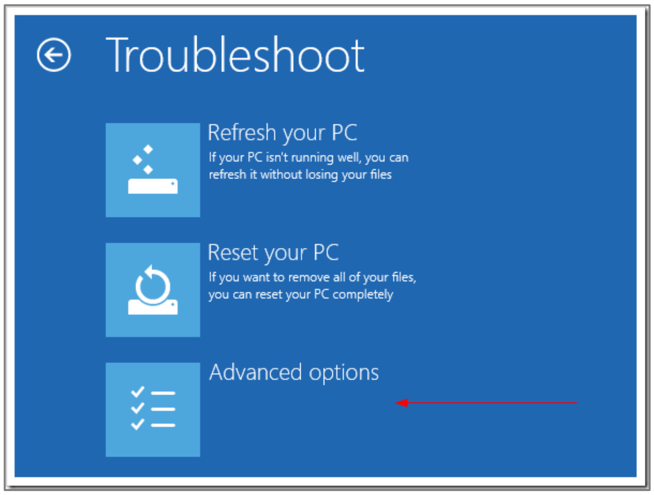
Choose an option

Continue
Exit and continue to Windows 8

Troubleshoot
Refresh or reset your PC, or use advanced tools

Turn off your PC

Select advanced options.



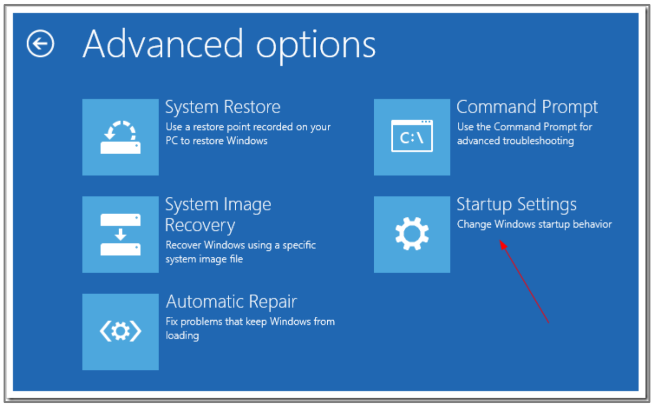
Troubleshoot

Refresh your PC
If your PC isn't running well, you can refresh it without losing your files

Reset your PC
If you want to remove all of your files, you can reset your PC completely

Advanced options

Select startup setup.



Advanced options

System Restore
Use a restore point recorded on your PC to restore Windows

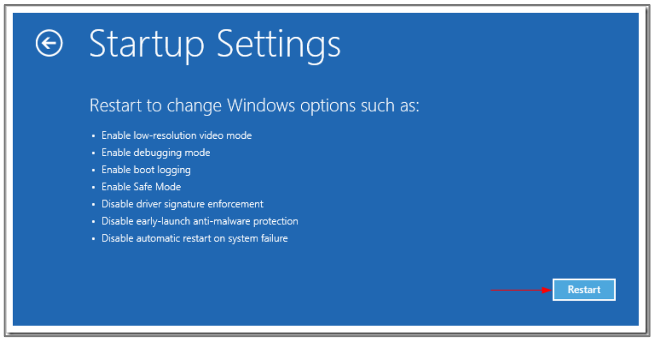
Command Prompt
Use the Command Prompt for advanced troubleshooting

System Image Recovery
Recover Windows using a specific system image file

Startup Settings
Change Windows startup behavior

Automatic Repair
Fix problems that keep Windows from loading

Click on Restart located at the bottom right.

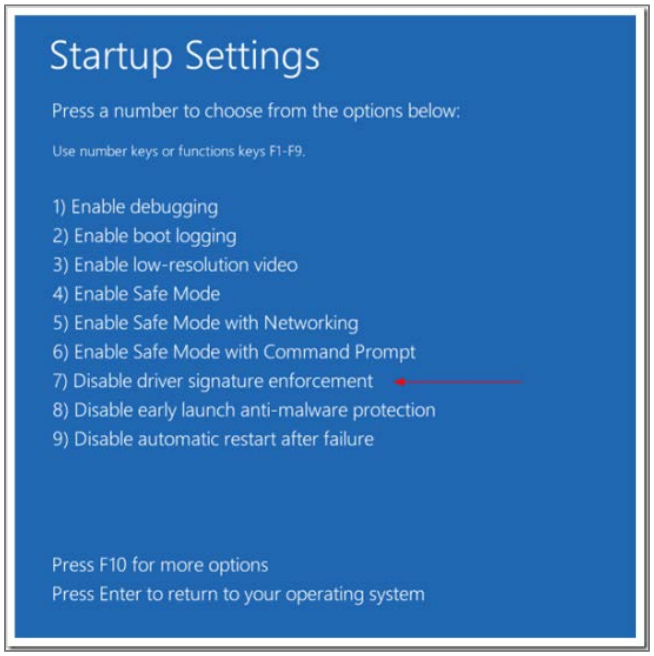


Startup Settings

Restart to change Windows options such as:

- Enable low-resolution video mode
- Enable debugging mode
- Enable boot logging
- Enable Safe Mode
- Disable driver signature enforcement
- Disable early-launch anti-malware protection
- Disable automatic restart on system failure

Click on the 7th option “disable driver signature enforcement.”



Startup Settings

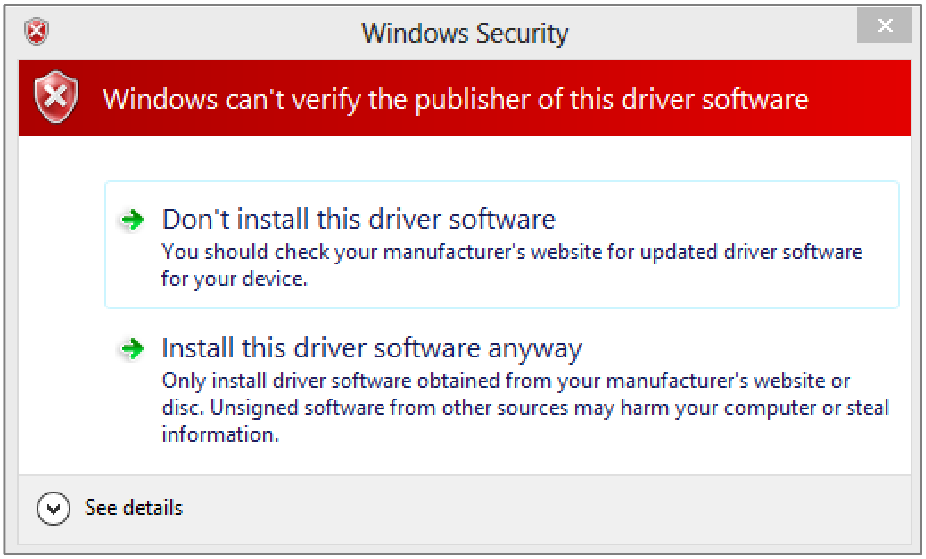
Press a number to choose from the options below:

Use number keys or functions keys F1-F9.

1) Enable debugging
2) Enable boot logging
3) Enable low-resolution video
4) Enable Safe Mode
5) Enable Safe Mode with Networking
6) Enable Safe Mode with Command Prompt
7) Disable driver signature enforcement
8) Disable early launch anti-malware protection
9) Disable automatic restart after failure

Press F10 for more options
Press Enter to return to your operating system

Click on Install this driver software anyway. After restarting the PC connect the CM-900 and install drivers.



Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

1.2 Linux

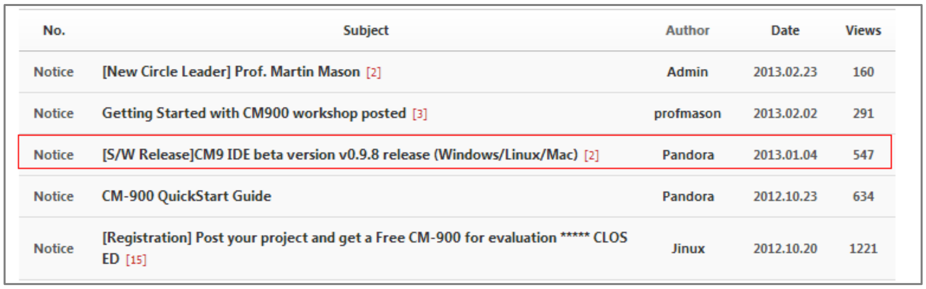
1.2.1 Have the CM-900 and USB Cable ready

The cable is a type B micro-USB; the same type as with most smartphones

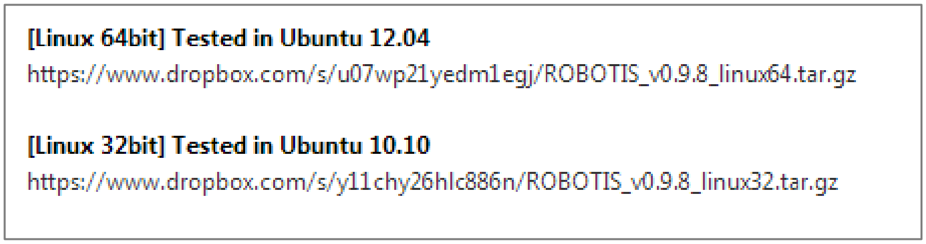
1.2.2 Download the ROBOTIS CM-9 Linux release

Download the 32-bit package for 32-bit versions of your Linux OS; 64-bit package for 64-bit version of Linux.

http://www.robotsource.org/xe/Circle_CM9_Developer_World

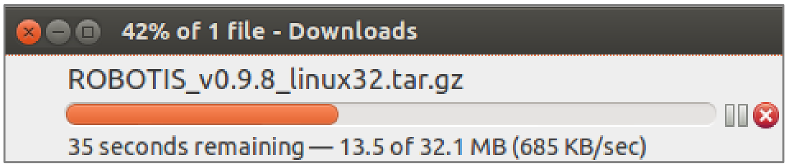


No.	Subject	Author	Date	Views
Notice	[New Circle Leader] Prof. Martin Mason [2]	Admin	2013.02.23	160
Notice	Getting Started with CM900 workshop posted [3]	profmason	2013.02.02	291
Notice	[S/W Release]CM9 IDE beta version v0.9.8 release (Windows/Linux/Mac) [2]	Pandora	2013.01.04	547
Notice	CM-900 QuickStart Guide	Pandora	2012.10.23	634
Notice	[Registration] Post your project and get a Free CM-900 for evaluation ***** CLOS ED [15]	Jinux	2012.10.20	1221



[Linux 64bit] Tested in Ubuntu 12.04
https://www.dropbox.com/s/u07wp21yedm1egj/ROBOTIS_v0.9.8_linux64.tar.gz

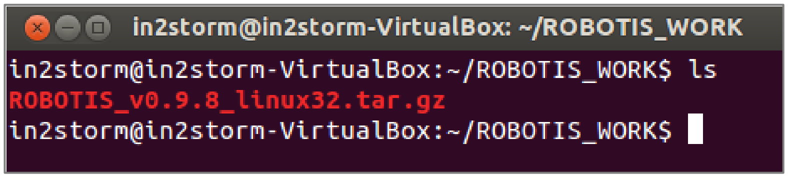
[Linux 32bit] Tested in Ubuntu 10.10
https://www.dropbox.com/s/y11chy26hlc886n/ROBOTIS_v0.9.8_linux32.tar.gz



ROBOTIS_v0.9.8_linux32.tar.gz

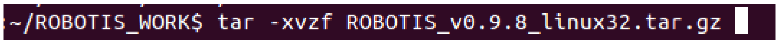
35 seconds remaining — 13.5 of 32.1 MB (685 KB/sec)

After downloading input the following command



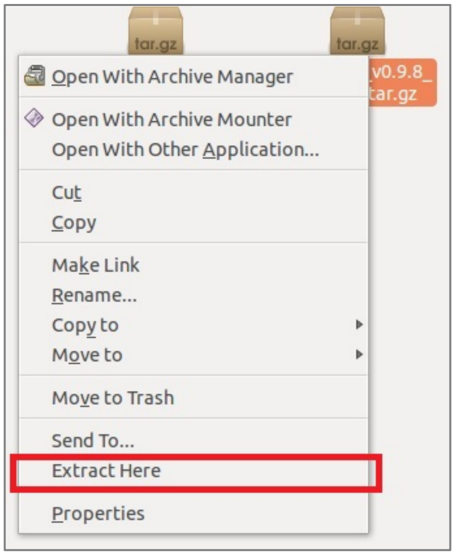
```
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$ ls
ROBOTIS_v0.9.8_linux32.tar.gz
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$ 
```

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

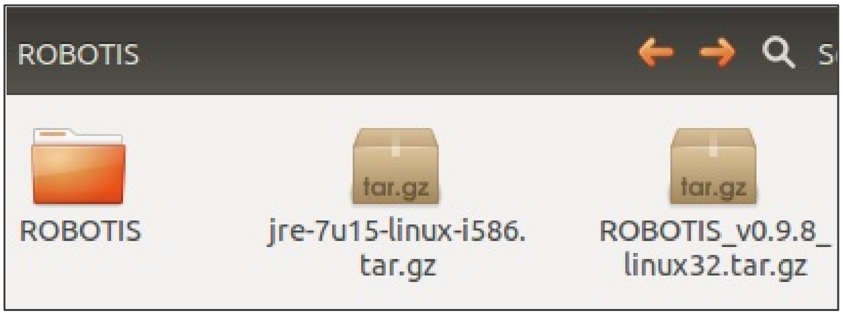


```
:~/ROBOTIS_WORK$ tar -xvzf ROBOTIS_v0.9.8_linux32.tar.gz
```

Or use the right mouse click to decompress the tarball package

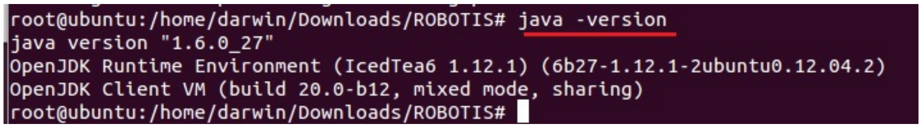


The decompressed file will show a ROBOTIS directory.



1.2.3 Check for JRE installation.

To check input the command java –version.

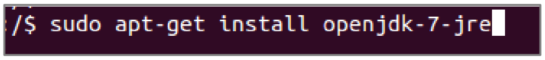


```
root@ubuntu:/home/darwin/Downloads/ROBOTIS# java -version
java version "1.6.0_27"
OpenJDK Runtime Environment (IcedTea6 1.12.1) (6b27-1.12.1-2ubuntu0.12.04.2)
OpenJDK Client VM (build 20.0-b12, mixed mode, sharing)
root@ubuntu:/home/darwin/Downloads/ROBOTIS#
```

If not installed simply get JRE via the apt-get command.

Run openjdk-7-jre-headless.

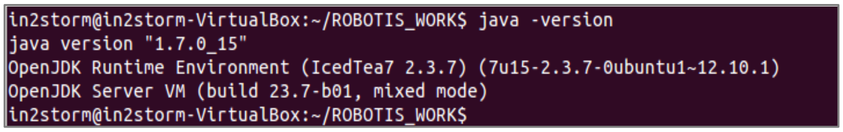
$sudo apt-get install openjdk-7-jre



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

Press the Y key.

Once installation is complete enter the command java –version

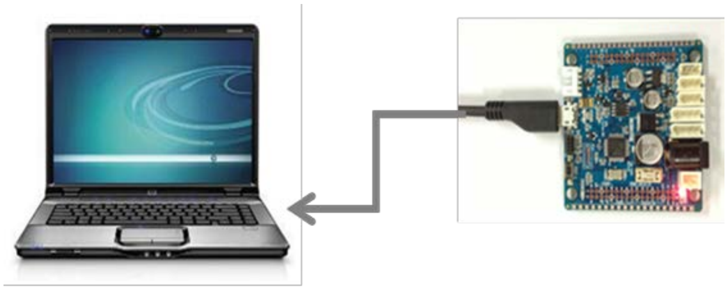


```
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$ java -version
java version "1.7.0_15"
OpenJDK Runtime Environment (IcedTea7 2.3.7) (7u15-2.3.7-0ubuntu1~12.10.1)
OpenJDK Server VM (build 23.7-b01, mixed mode)
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$
```

Upon successful installation of JRE run ROBOTIS CM-9.

1.2.4 Connect the CM-900 to the PC.

Connect the CM-900 to the PC with the USB cable.

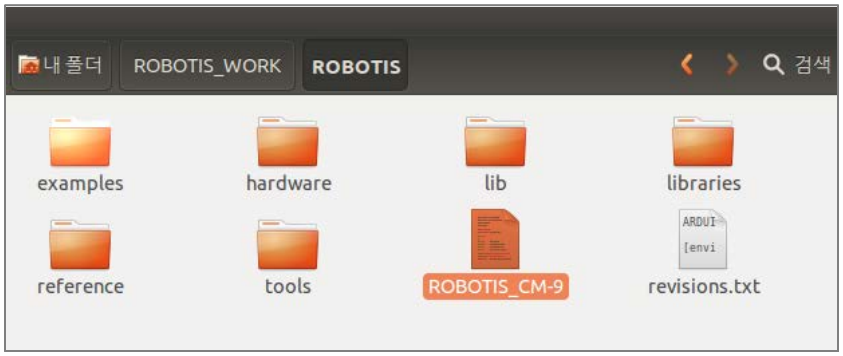


<connect the CM-900 to the PC>

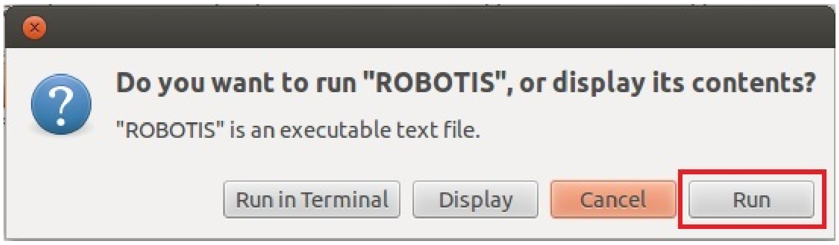
Please refrain from connecting the CM-900 to the PC via USB hub. We recommend connecting the CM-900 to the PC directly. The USB hub may not be able to provide enough electrical current to the CM-900 to properly download programs. Connect the CM-900 to an USB port with guaranteed enough electrical current supply.

1.2.5 Run ROBOTIS CM-9.

From a terminal window go to ROBOTIS directory and enter the command./ ROBOTIS CM-9.

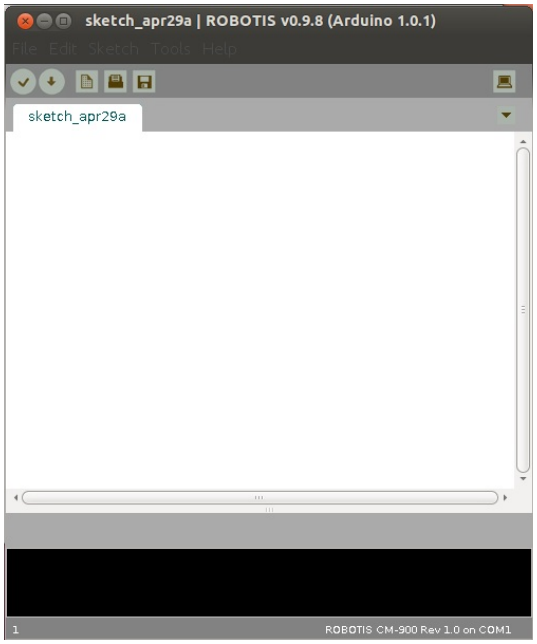


Or double-click on the executable, and click on Run



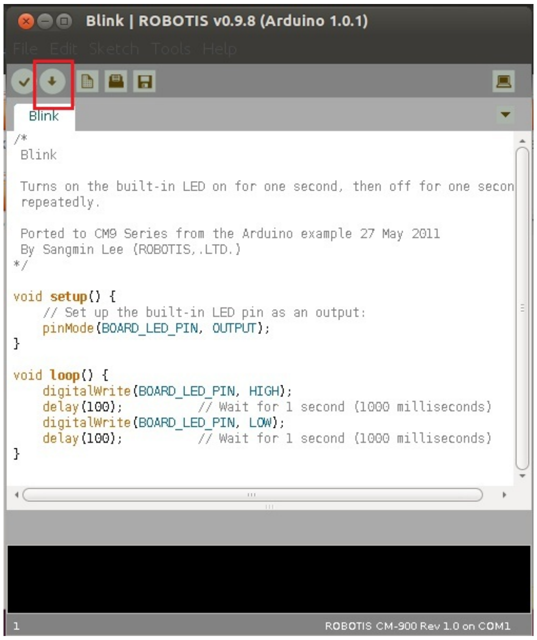
Do you want to run "ROBOTIS", or display its contents?

"ROBOTIS" is an executable text file.



sketch_apr29a | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
sketch_apr29a
1
ROBOTIS CM-900 Rev 1.0 on COM1

1.2.6 Open Blink example

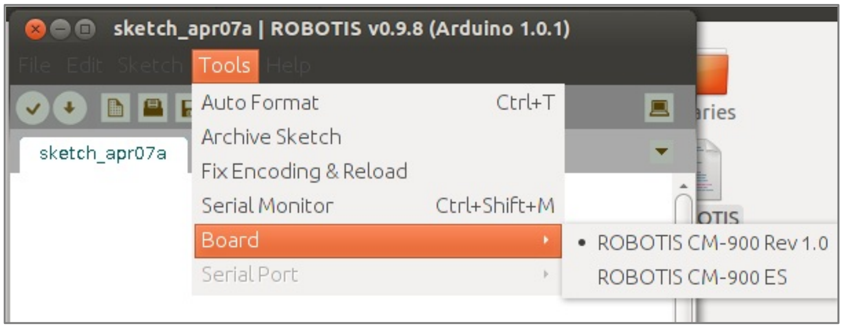


Blink | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Blink
1
ROBOTIS CM-900 Rev 1.0 on COM1

```
/*
 Blink

 Turns on the built-in LED on for one second, then off for one secon
 repeatedly.

 Ported to CM9 Series from the Arduino example 27 May 2011
 By Sangmin Lee (ROBOTIS,.LTD.)
*/

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)
}
```

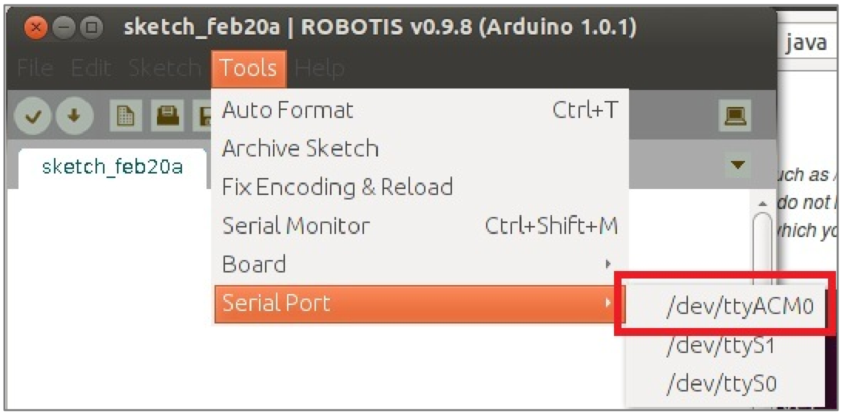
1.2.7 Select a board.

If the CM-900 hardware version is Rev 1.0 or higher then select ROBOTIS CM-900 Rev 1.0.

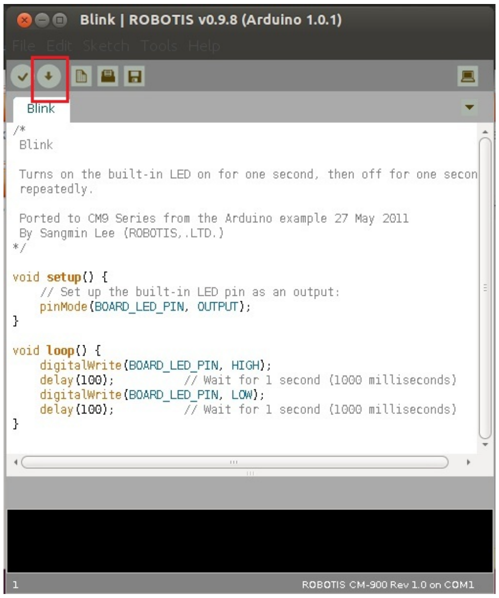


1.2.8 Select serial port.

Select ttyACMX device.



1.2.9 Download the code

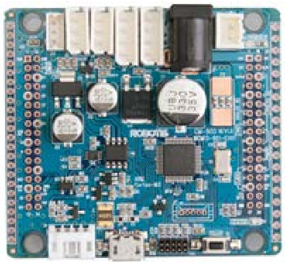

Blink | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Blink
1
ROBOTIS CM-900 Rev 1.0 on COM1

```
/*
 Blink

 Turns on the built-in LED on for one second, then off for one secon
 repeatedly.

 Ported to CM9 Series from the Arduino example 27 May 2011
 By Sangmin Lee (ROBOTIS,.LTD.)
*/

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)
}
```

1.3 Mac OS X

1.3.1 Have the CM-900 and USB cable ready.

The cable is a type B micro-USB; the same type as with most smartphones

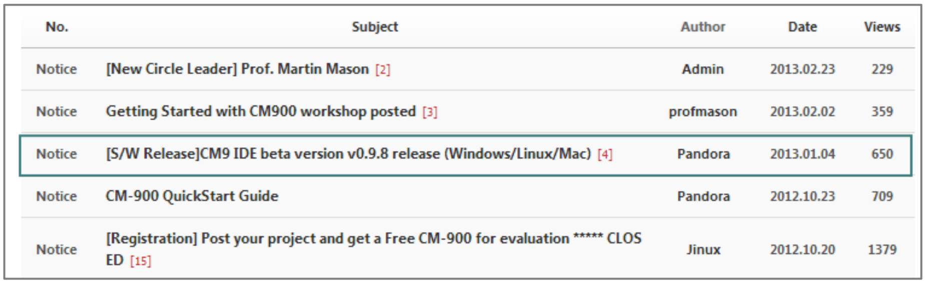


1.3.2 Download ROBOTIS CM9 Mac OS X release

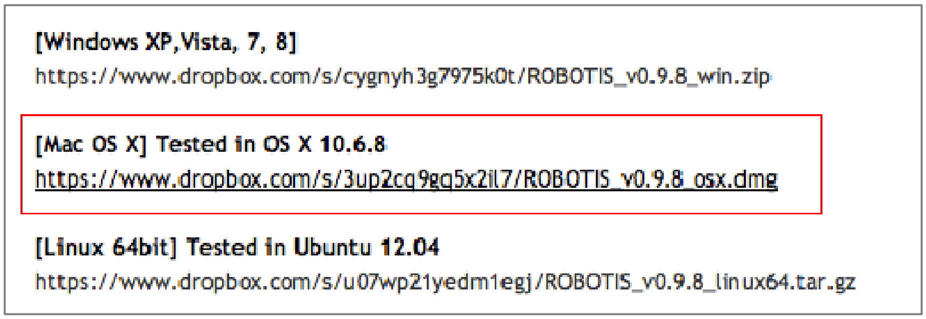
Download the most recent version of ROBOTIS CM9. You can get the most recent version by clicking on the link below.

http://www.robotsource.org/xe/Circle_CM9_Developer_World

Look for “Notice” entries.



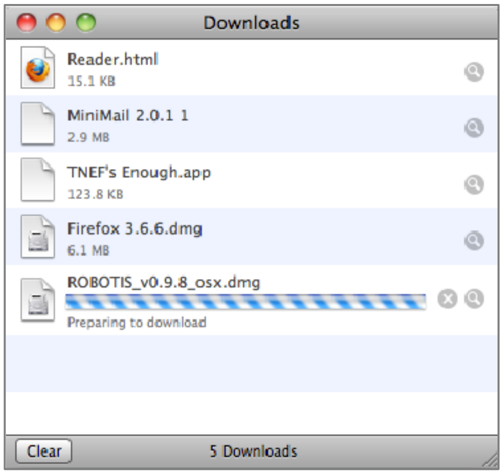
No.	Subject	Author	Date	Views
Notice	[New Circle Leader] Prof. Martin Mason [2]	Admin	2013.02.23	229
Notice	Getting Started with CM900 workshop posted [3]	profmason	2013.02.02	359
Notice	[S/W Release]CM9 IDE beta version v0.9.8 release (Windows/Linux/Mac) [4]	Pandora	2013.01.04	650
Notice	CM-900 QuickStart Guide	Pandora	2012.10.23	709
Notice	[Registration] Post your project and get a Free CM-900 for evaluation ***** CLOSED [15]	Jinux	2012.10.20	1379



[Windows XP,Vista, 7, 8]
https://www.dropbox.com/s/cygnyh3g7975k0t/ROBOTIS_v0.9.8_win.zip

[Mac OS X] Tested in OS X 10.6.8
https://www.dropbox.com/s/3up2cq9gq5x2il7/ROBOTIS_v0.9.8_osx.dmg

[Linux 64bit] Tested in Ubuntu 12.04
https://www.dropbox.com/s/u07wp21yedm1egj/ROBOTIS_v0.9.8_linux64.tar.gz



Downloads

Reader.html
15.1 KB

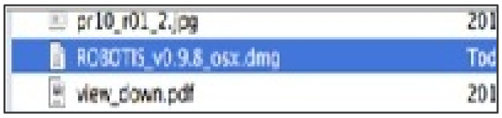
MiniMail 2.0.1 1
2.9 MB

TNEF's Enough.app
123.8 KB

Firefox 3.6.6.dmg
6.1 MB

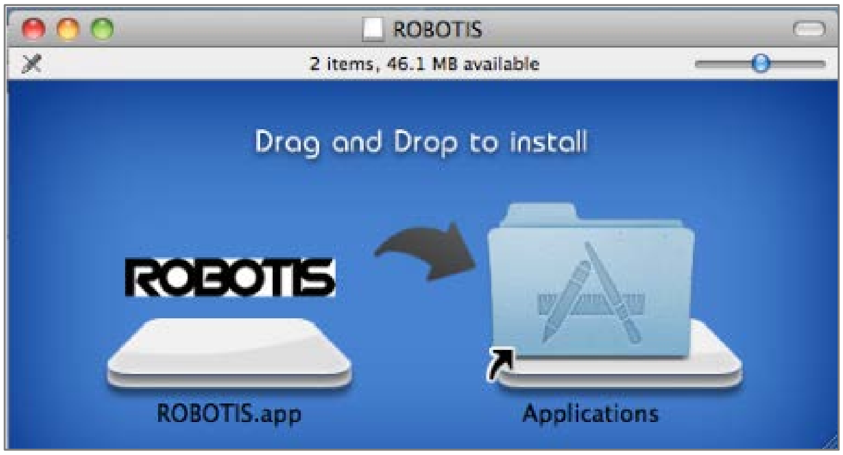
ROBOTIS_v0.9.8_osx.dmg
Preparing to download

Once download is complete double-click on the dmg and mount it.



pr10_r01_2.jpg 201
ROBOTIS_v0.9.8_osx.dmg Tod
view_down.pdf 201

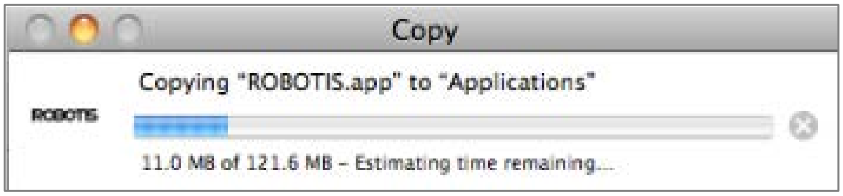
Drag ROBOTIS icon to the Applications folder



ROBOTIS

2 items, 46.1 MB available

Wait until transfer is complete



Copy

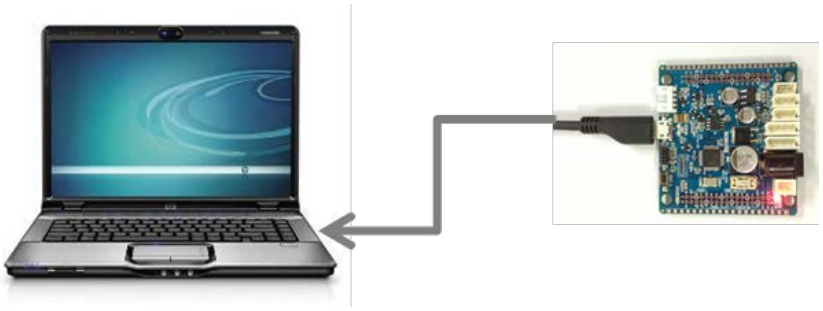
Copying "ROBOTIS.app" to "Applications"

11.0 MB of 121.6 MB – Estimating time remaining...

Once transfer is complete go to the Applications folder and double-click ROBOTIS.app.

1.3.3 Connect the CM-900 to the Mac

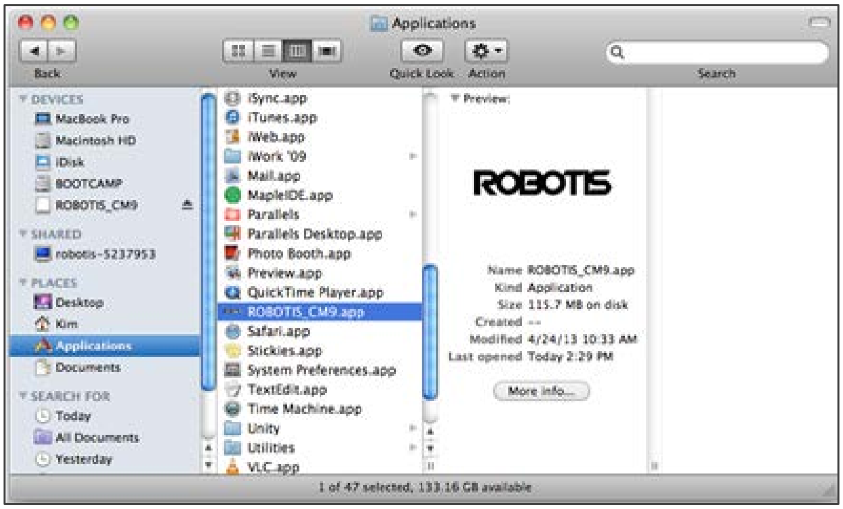
Connect the CM-900 to the Mac with the USB cable.



Please refrain from connecting the CM-900 to the PC via USB hub. We recommend connecting the CM-900 to the PC directly. The USB hub may not be able to provide enough electrical current to the CM-900 to properly download programs. Connect the CM-900 to an USB port with guaranteed enough electrical current supply.

1.3.4 Run ROBOTIS CM9

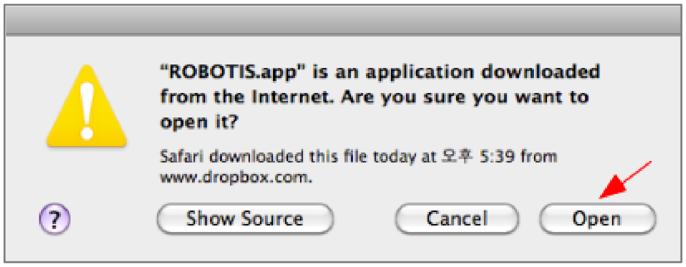
From the Applications folder double-click on ROBOTIS.app.



Name ROBOTIS_CM9.app
Kind Application
Size 115.7 MB on disk
Created --
Modified 4/24/13 10:33 AM
Last opened Today 2:29 PM

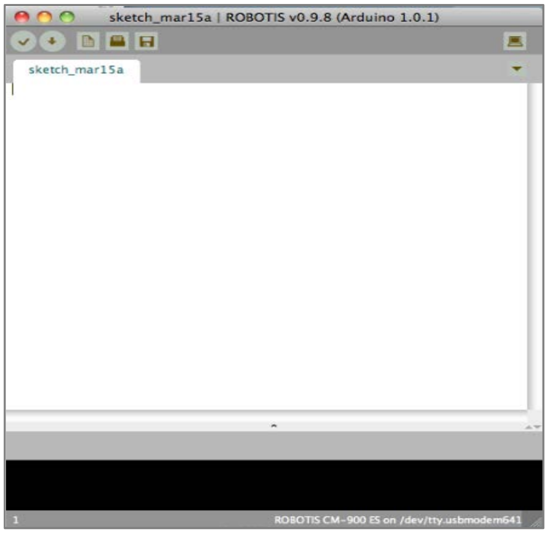
1 of 47 selected, 133.16 GB available

Simply click on open

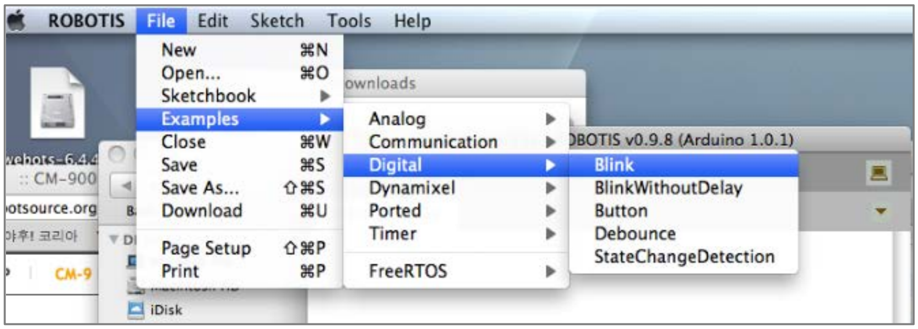


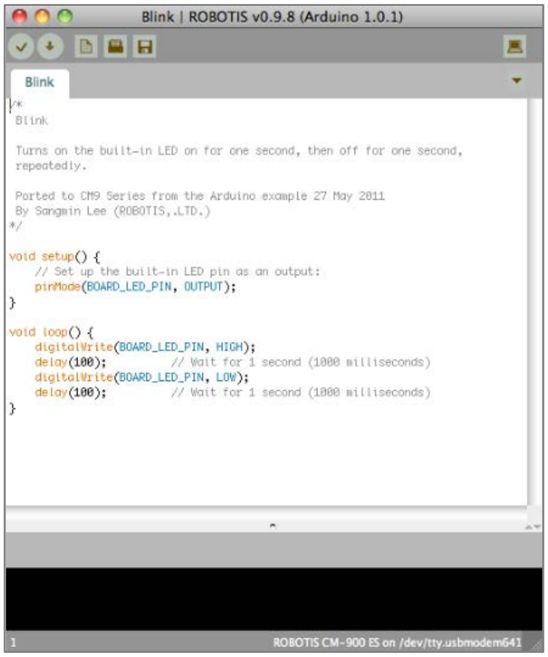
"ROBOTIS.app" is an application downloaded from the Internet. Are you sure you want to open it?

Safari downloaded this file today at 오후 5:39 from www.dropbox.com.


sketch_mar15a | ROBOTIS v0.9.8 (Arduino 1.0.1)
sketch_mar15a
1
ROBOTIS CM-900 ES on /dev/tty.usbmodem641

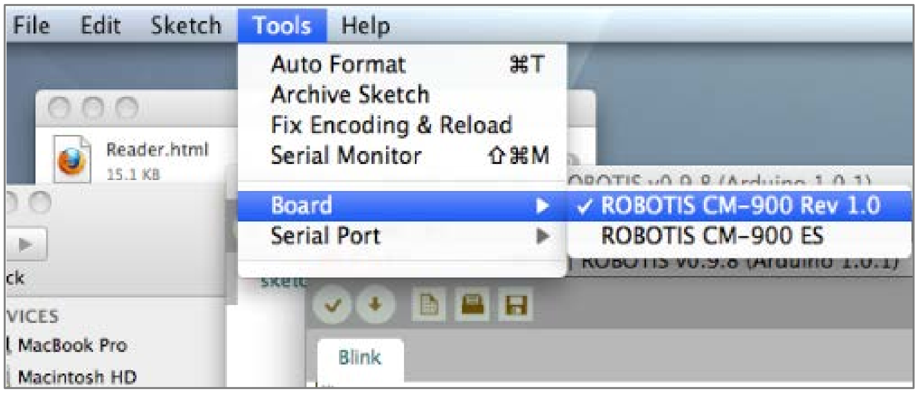
1.3.5 Open the Blink example



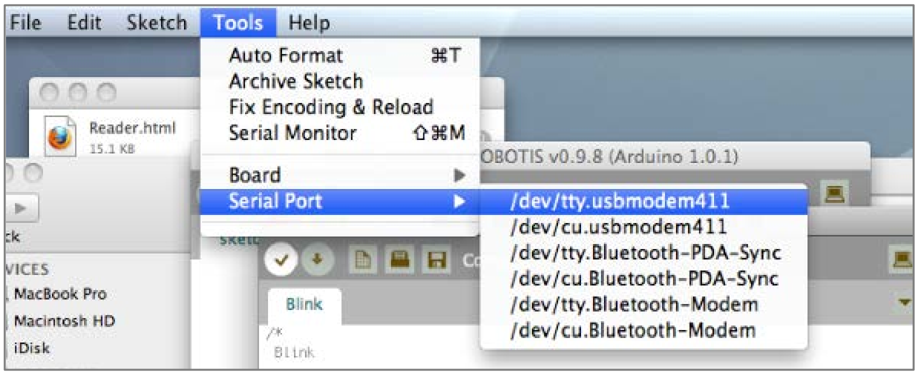

Blink | ROBOTIS v0.9.8 (Arduino 1.0.1)
Blink
1
ROBOTIS CM-900 ES on /dev/tty.usbmodem641

```
/*
 Blink

 Turns on the built-in LED on for one second, then off for one second,
 repeatedly.

 Ported to CM9 Series from the Arduino example 27 May 2011
 By Sangmin Lee (ROBOTIS,.LTD.)
*/

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)
}
```

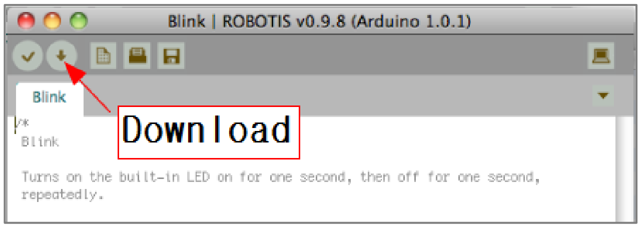
1.3.6 Select the board



1.3.7 Select serial port



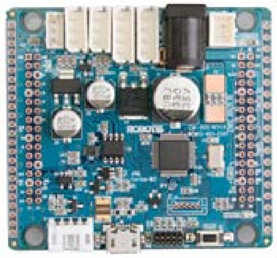
1.3.8 Download the code



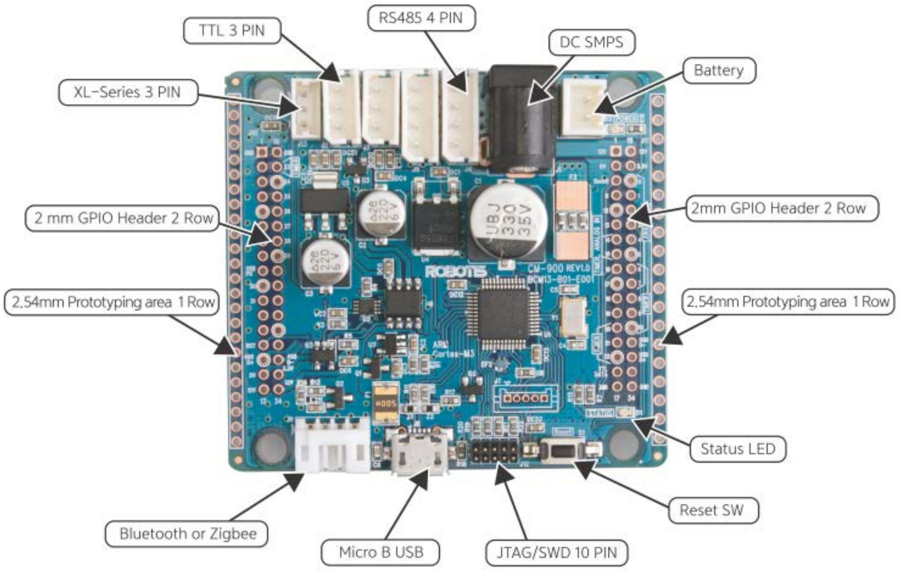
```
/*
Blink

Turns on the built-in LED on for one second, then off for one second,
repeatedly.
```

2 CM-900 hardware

2.1 Illustration of the CM-900



2.2 Parts label



2.2.1 Micro USB (type B): provides the CM-900 with downloading and communication capabilities via PC, in addition to electrical power from USB's 5V. Use the included USB cable or any other conventional USB cable you may likely have lying around.

2.2.2 Reset switch : resets the CM-900 CPU

2.2.3 BT-110/ZIG-110 4PIN : Connect a 4-pin BT-110, BT-210, ZIG-110, or LN-101. This allows to communicate with external peripherals with serial UART communications. The LN-101 is more useful than simple firmware download from the PC.

2.2.4 2mm/2.54 mm prototyping area: located on both sides of the CM-900 board with 2.0mm or 2.54mm pitch to facilitate mounting of other devices.

2.2.5 2 mm GPIO Header : Allows the CM-900's STM32F103C8 CPU to freely interface with external devices.

2.2.6 TTL 3 PIN : connect to Dynamixel via 3-pin cable daisy-chain (TTL communications).

2.2.7 RS485 4 PIN : connect to Dynamixel via 4-pin cable daisy-chain (RS-485 communications).

2.2.8 Power LED : LED on when board is powered on and LED off when board is powered off.

2.2.9 Status LED : CM-900's program verification test LED. Send high/low signals to pin D16 to turn LED on/off.

2.2.10 Battery socket : socket to connect battery.

2.2.11 DC SMPS : jack for 12V SMPS.

2.2.12 XL-Series 3 PIN : Connect to Dynamixel XL-series via 3-pin TTL communications.

2.2.13 JTAG/SWD 10 PIN: JTAG/SWD terminal for other programming features via IAR, Keil.

2.3 Package list

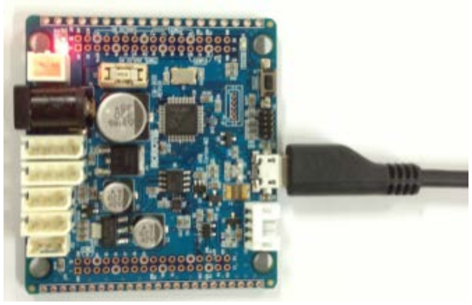
Component		Quan-tity
Controller	CM-900	1
Download	Micro-B-Cable(USB)	1
Manual	User Guide	1

2.4 Product characteristics

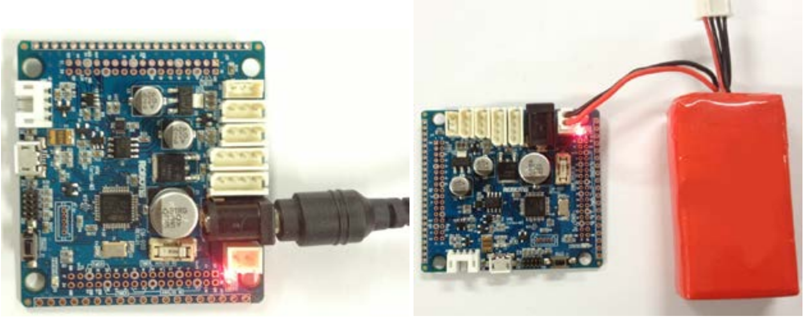
CM-900	
CPU	STM32F103C8 (ARM Cortex-M3)
Op Voltage	5V~24V(USB 5V, DXL 12V, XL-Series 7.4V)
I/O	GPIO 32
Timer	8 (16bit)
Analog In(ADC)	10 (12bit)
Flash	64 Kbytes
SRAM	20 Kbytes
Clock	72Mhz
USB	1 (2.0 FullSpeed)
CAN	1
USART	3
SPI	2
I2C(TWI)	2
Debug	JTAG & SWD
DMA	7ch
3 Pin TTL	2
4 Pin RS485	2
3 Pin XL-Serise	1
SIZE	60mm X 54 mm X 1.6 mm

2.5 Power

2.5.1 Simply connect via USB and the CM-900 is operational.



2.5.2 To drive Dynamixel(s) connect a 12V power supply either via battery or DC power.



2.5.3 If 12V SMPS and battery are connected the CM-900 will draw power only from the 12V SMPS.

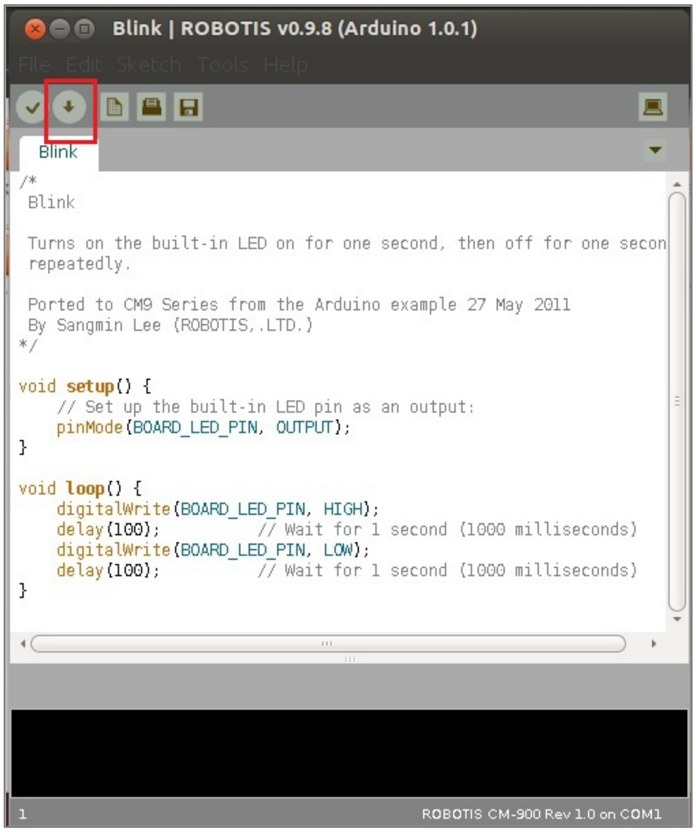
2.5.4 When SMPS or battery and 5V USB are connected then the CM-900 will halt drawing power from USB. USB connection remains intact.

2.6 Operating

2.6.1 When the CM-900 is powered by USB or SMPS/Battery it automatically runs user code 0x08003000.

For programming connect the CM-900 to the PC and run its dedicated software integrated development environment (IDE) to write code, compile and download.

<connect the CM-900 to the PC>

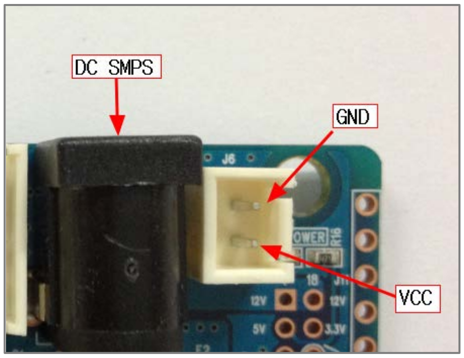

Blink | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Blink
1
ROBOTIS CM-900 Rev 1.0 on COM1

```
/*
 Blink

 Turns on the built-in LED on for one second, then off for one secon
 repeatedly.

 Ported to CM9 Series from the Arduino example 27 May 2011
 By Sangmin Lee (ROBOTIS,.LTD.)
*/

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)
}
```

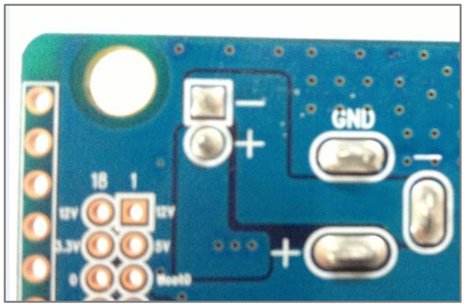
<CM-900's integrated development environment: ROBOTIS CM-9>

2.7 Pin layout information

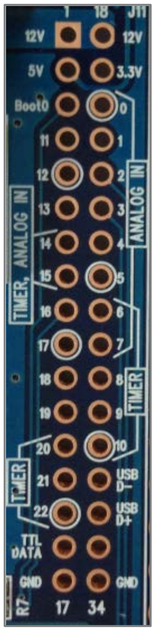
2.7.1 Power port

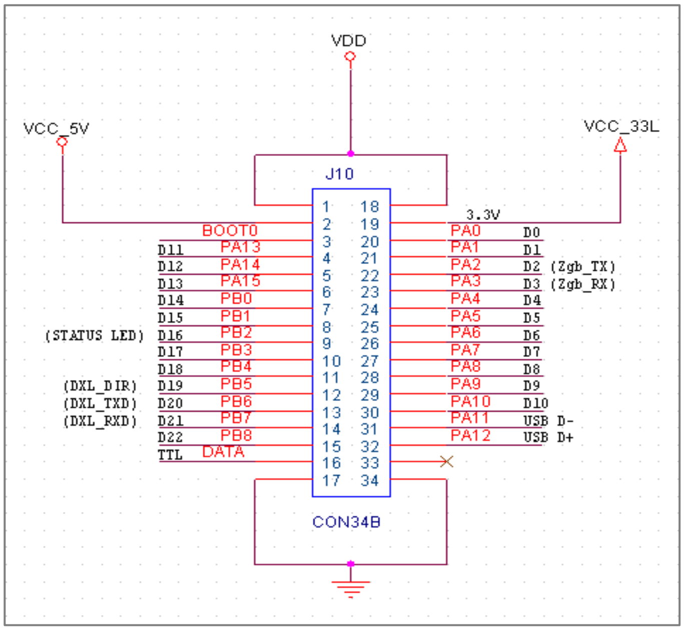


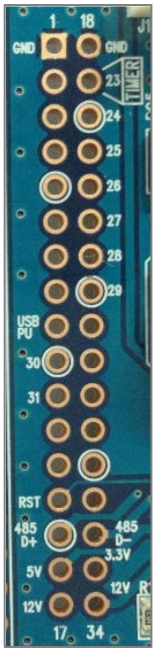
Reverse side also shown

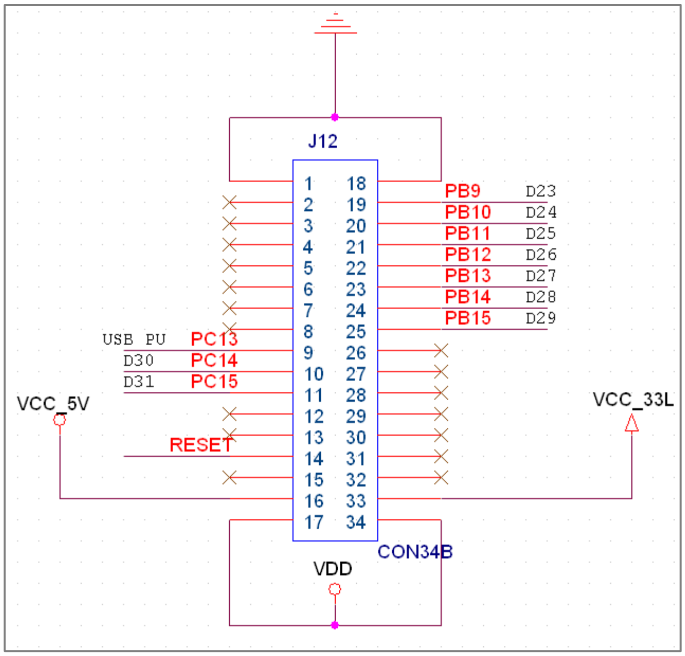


2.7.2 You can see the CM-900's의 GPIO header pin connections to the STM32F103C8 CPU



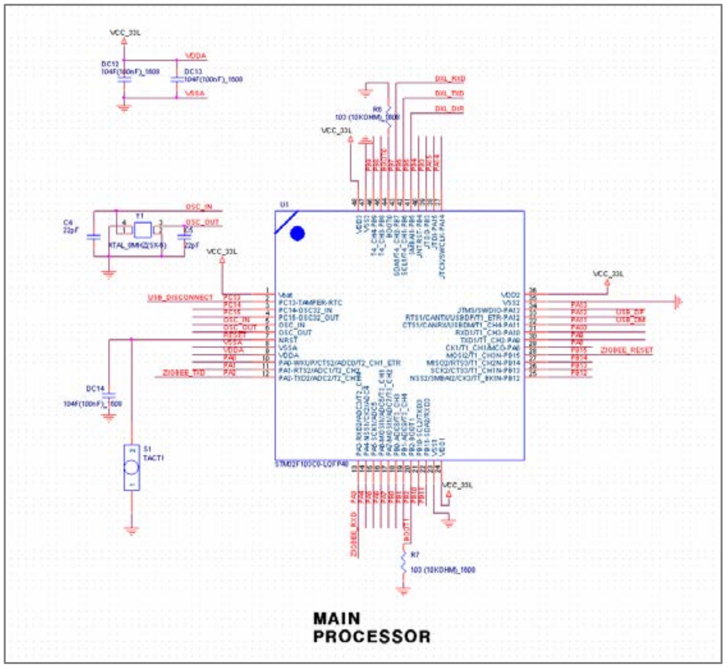






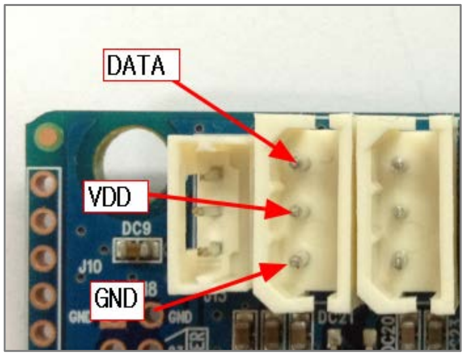
The 'X' mark indicates support for future CM-9 series. VDD is 12V.

Please note Dynamixel-related pins (D6,D7,D19) USB pins (PA11,PA12,PC13) are separately connected.

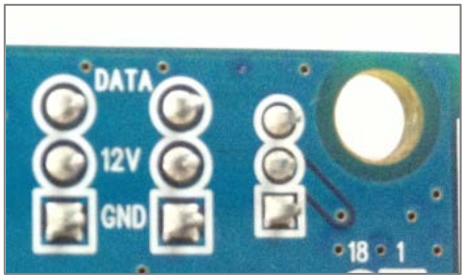


< STM32F103C8(LQFP48 Package) CPU schematic>

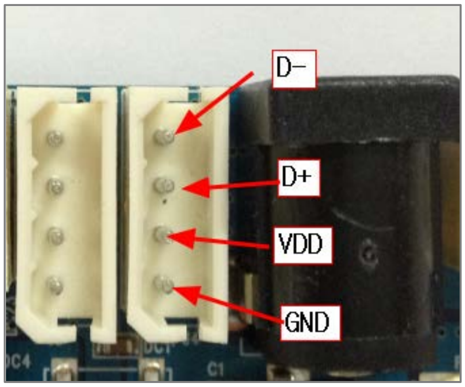
2.7.3 3-pin TTL



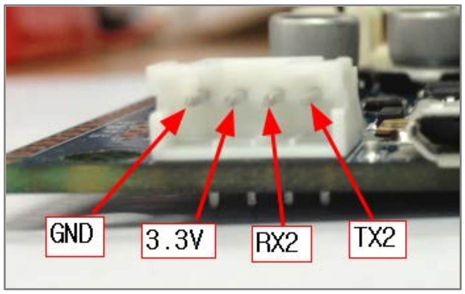
Also shown the reverse side silk screen



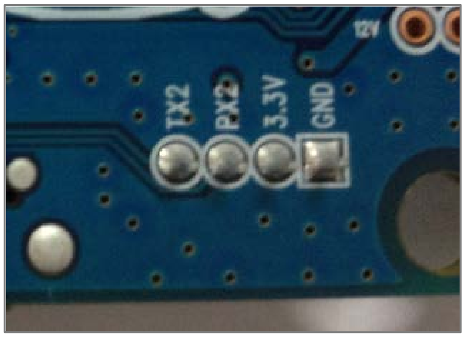
2.7.4 4-pin RS-485



2.7.5 4-pin communications connection



Reverse side



2.8 Schematics & PCB Gerber file (Schematic & Gerber Design)

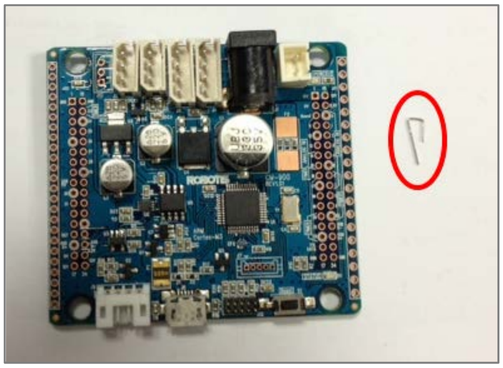
The CM-900 is an open-source embedded board with open hardware and software. The following is a summary of the hardware schematic and PCB Gerber file.

2.8.1 Schematics : CM-900_REV_1.01_Schematic_20121129.pdf

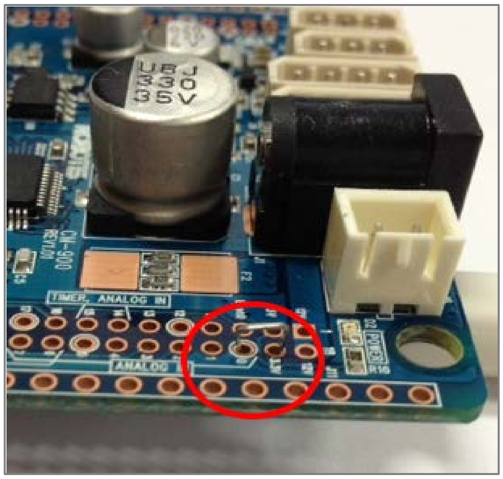
2.8.2 PCB Gerber(EAGLE): **(TBA)**

2.9 Emergency recovery mode

2.9.1 In case the CM-900's USB drivers are not detected nor detected by ROBOTIS CM connect the 3.3V to D0.

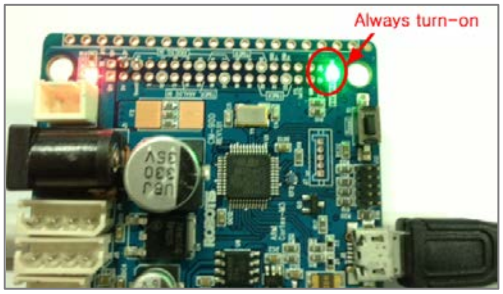


2.9.2 Use a conducting pin (i.e. small wire)

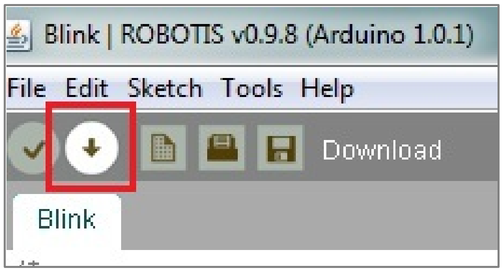


connect 3.3V pin to D0

2.9.3 Connect a USB and verify STATUS LED (on).



2.9.4 Go to: File -> examples -> Digital -> Blink then click on the download button.


Blink | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Download
Blink

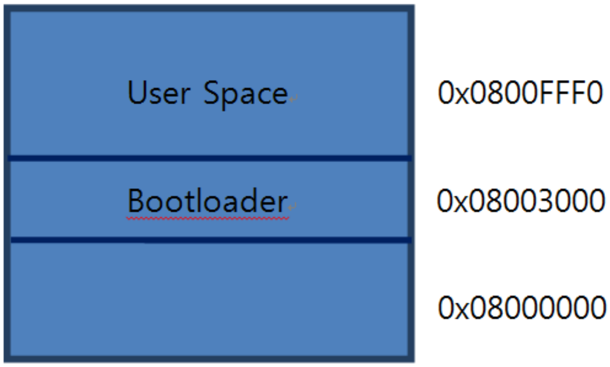
2.9.5 The CM-900 can be programmed via ROBOTIS CM-9.

2.10 CM-900 Flash Memory Map

The CM-900 has 64kbytes of memory available where the bootloader takes up 12kbytes. The remaining is available to the user for programming. The bootloader's binary file begins with 0x08000000.

2.10.1 Bootloader : 0x08000000 ~ 0x08002FFF

2.10.2 User Programming Space : 0x08003000 ~ 0x0800FFF0

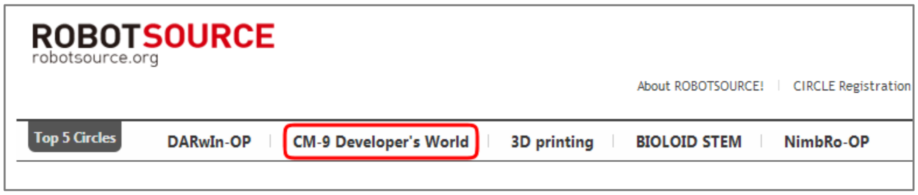


3 CM-900 IDE software

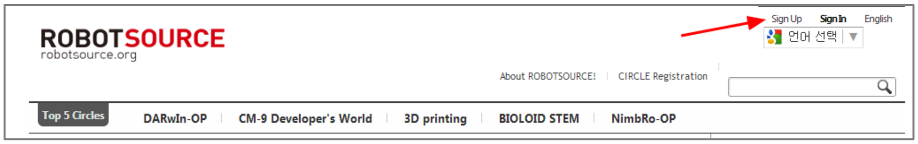
3.1 Download ROBOTIS CM-9

To program the CM-900 you need ROBOTIS CM-9; get it from CM9 Developer's World Circle in BOTSOURCE.

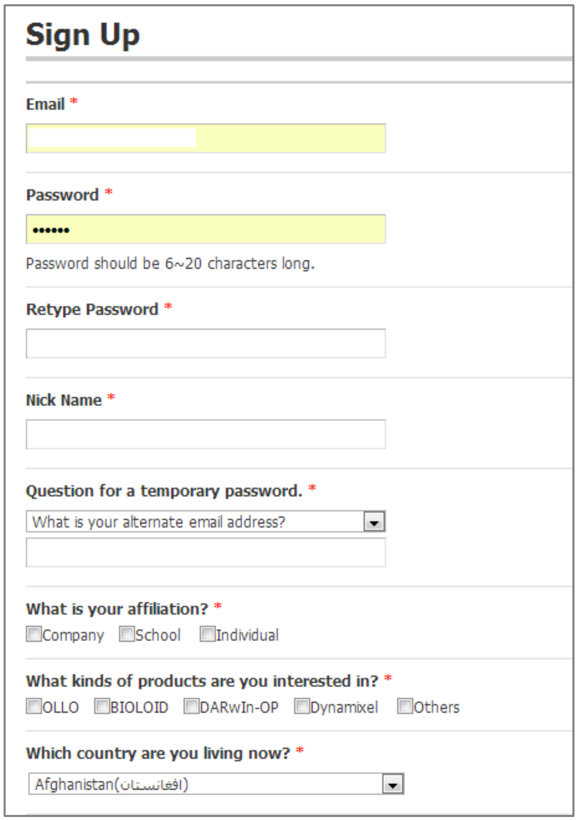
www.robotsource.org



If you have not yet signed up with ROBOTSOURCE we strongly recommend you do so.



Sign Up is a very simple process.



Sign Up

Email *

Password *

••••••

Password should be 6~20 characters long.

Retype Password *

Nick Name *

Question for a temporary password. *

What is your alternate email address?

What is your affiliation? *

□Company □School □Individual

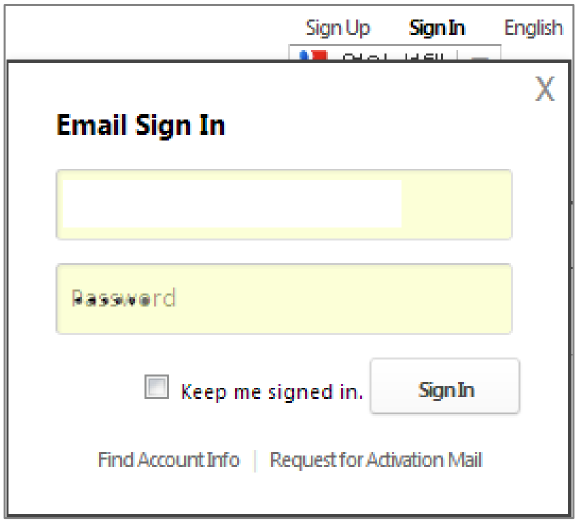
What kinds of products are you interested in? *

□OLLO □BIOLOID □DARwIn-OP □Dynamixel □Others

Which country are you living now? *

Afghanistan(افغانستان)

After signing up log in you can proceed to download ROBOTIS CM-9



Email Sign In

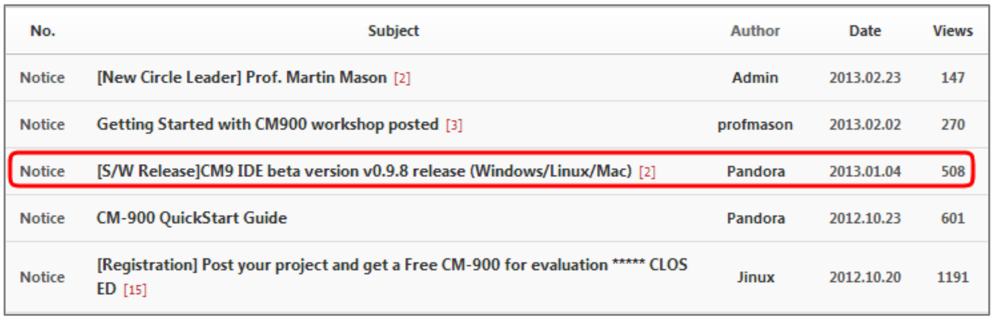
Password

□ Keep me signed in. Sign In

Find Account Info | Request for Activation Mail

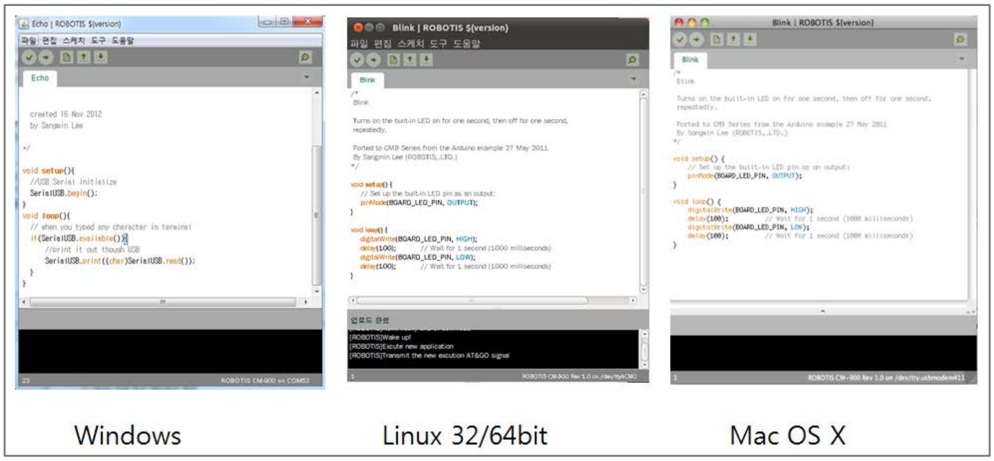
< simply by signing in with your registered email address>

Go to CM-9 Developer's World Circle's Notice and download ROBOTIS CM-9 SW.



No.	Subject	Author	Date	Views
Notice	[New Circle Leader] Prof. Martin Mason [2]	Admin	2013.02.23	147
Notice	Getting Started with CM900 workshop posted [3]	profmason	2013.02.02	270
Notice	[S/W Release]CM9 IDE beta version v0.9.8 release (Windows/Linux/Mac) [2]	Pandora	2013.01.04	508
Notice	CM-900 QuickStart Guide	Pandora	2012.10.23	601
Notice	[Registration] Post your project and get a Free CM-900 for evaluation ***** CLOSED [15]	Jinux	2012.10.20	1191

Click on the link according to your computer OS.


Echo | ROBOTIS $(version)
파일 편집 스케치 도구 도움말
Echo
23
ROBOTIS CM-900 on COM53

```
  created 16 Nov 2012
  by Sangmin Lee

*/

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());
  }
}
```

Blink | ROBOTIS ${version}
파일 편집 스케치 도구 도움말
Blink
업로드 완료
1
ROBOTIS CM-900 Rev 1.0 on /dev/ttyACM0

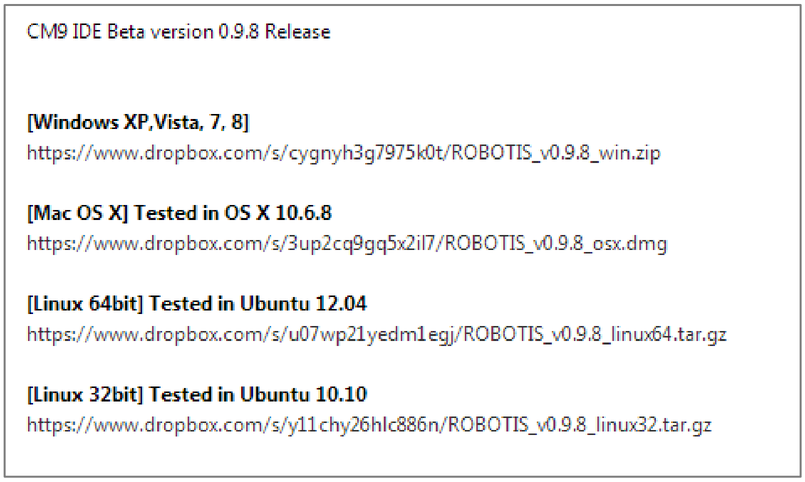
```
/*
 Blink

 Turns on the built-in LED on for one second, then off for one second,
 repeatedly.

 Ported to CM9 Series from the Arduino example 27 May 2011
 By Sangmin Lee (ROBOTIS,.LTD.)
*/

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)
}
```

```
[ROBOTIS]Wake up!
[ROBOTIS]Excute new application
[ROBOTIS]Transmit the new excution AT&GO signal
```

Blink | ROBOTIS ${version}
Blink
1
ROBOTIS CM-900 Rev 1.0 on /dev/tty.usbmodem411

```
/*
 Blink

 Turns on the built-in LED on for one second, then off for one second,
 repeatedly.

 Ported to CM9 Series from the Arduino example 27 May 2011
 By Sangmin Lee (ROBOTIS,.LTD.)
*/

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)
}
```

Windows Linux 32/64bit Mac OS X



CM9 IDE Beta version 0.9.8 Release

[Windows XP,Vista, 7, 8]
https://www.dropbox.com/s/cygnyh3g7975k0t/ROBOTIS_v0.9.8_win.zip

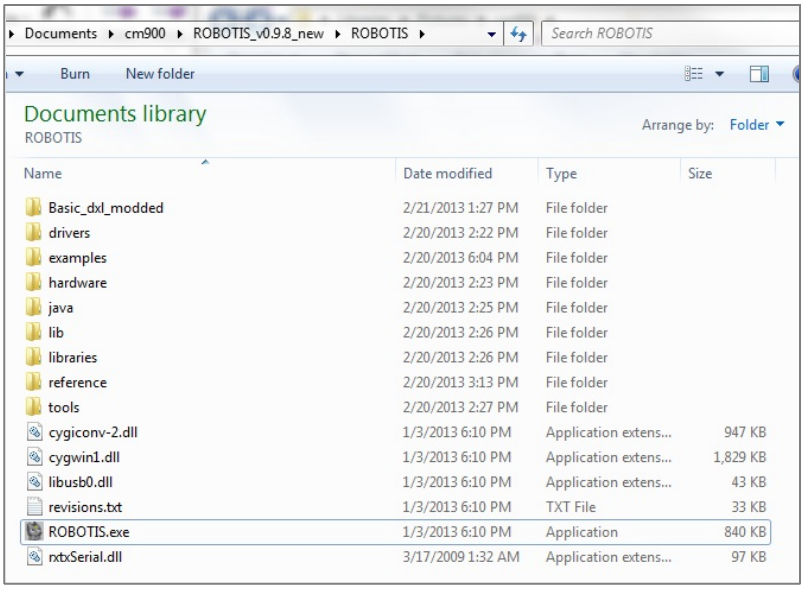
[Mac OS X] Tested in OS X 10.6.8
https://www.dropbox.com/s/3up2cq9gq5x2il7/ROBOTIS_v0.9.8_osx.dmg

[Linux 64bit] Tested in Ubuntu 12.04
https://www.dropbox.com/s/u07wp21yedm1egj/ROBOTIS_v0.9.8_linux64.tar.gz

[Linux 32bit] Tested in Ubuntu 10.10
https://www.dropbox.com/s/y11chy26hlc886n/ROBOTIS_v0.9.8_linux32.tar.gz

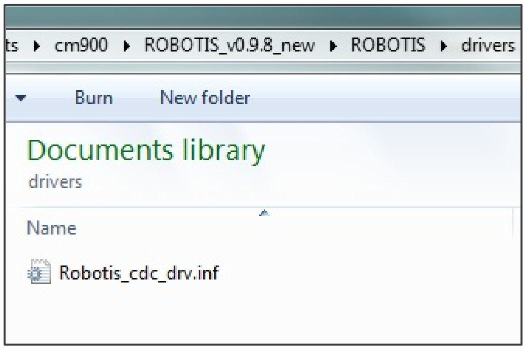
3.2 ROBOTIS CM-9 structure

After decompressing the downloaded file the structure will appear as shown below.



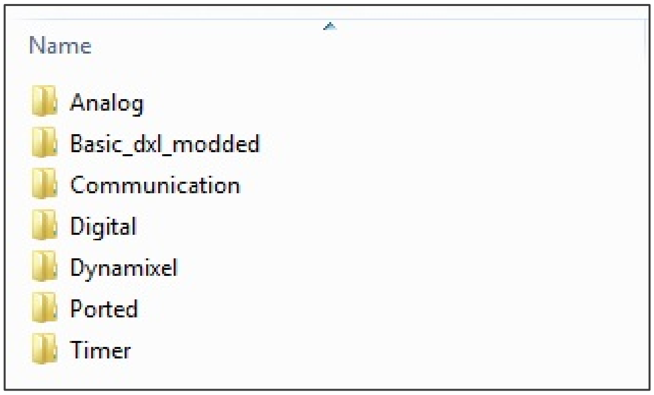
Name	Date modified	Type	Size
Basic_dxl_modded	2/21/2013 1:27 PM	File folder	
drivers	2/20/2013 2:22 PM	File folder	
examples	2/20/2013 6:04 PM	File folder	
hardware	2/20/2013 2:23 PM	File folder	
java	2/20/2013 2:25 PM	File folder	
lib	2/20/2013 2:26 PM	File folder	
libraries	2/20/2013 2:26 PM	File folder	
reference	2/20/2013 3:13 PM	File folder	
tools	2/20/2013 2:27 PM	File folder	
cygiconv-2.dll	1/3/2013 6:10 PM	Application extens...	947 KB
cygwin1.dll	1/3/2013 6:10 PM	Application extens...	1,829 KB
libusb0.dll	1/3/2013 6:10 PM	Application extens...	43 KB
revisions.txt	1/3/2013 6:10 PM	TXT File	33 KB
ROBOTIS.exe	1/3/2013 6:10 PM	Application	840 KB
rxtxSerial.dll	3/17/2009 1:32 AM	Application extens...	97 KB

3.2.1 Drivers : contains the Windows .inf USB drivers



Robotis_cdc_drv.inf

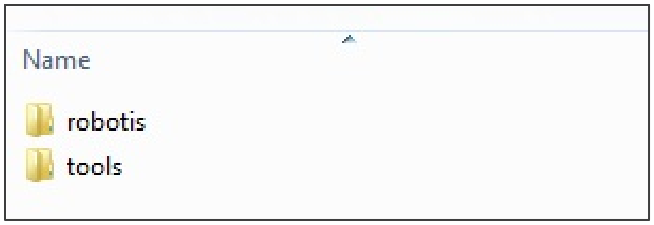
3.2.2 Examples : Contains the files for examples for ROBOTIS CM-9.



Name

- Analog
- Basic_dxl_modded
- Communication
- Digital
- Dynamixel
- Ported
- Timer

3.2.3 Hardware : contains the CM-9-series C/C++ sources + ARM-based compiler

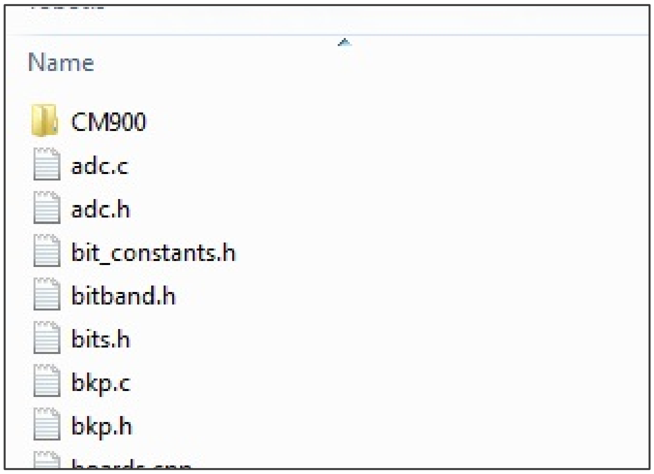


Name

- robotis
- tools

Robotis folder contains the CM-900's API core library

ROBOTIS\hardware\robotis\cores\robotis



Name

- CM900
- adc.c
- adc.h
- bit_constants.h
- bitband.h
- bits.h
- bkp.c
- bkp.h

3.2.4 Java : contains JRE (Java Runtime Environment).

3.2.5 Lib : ROBOTIS CM-9 resources

3.2.6 Libraries : sketch libraries

3.2.7 Reference : CM-9-series data suite and API documentation

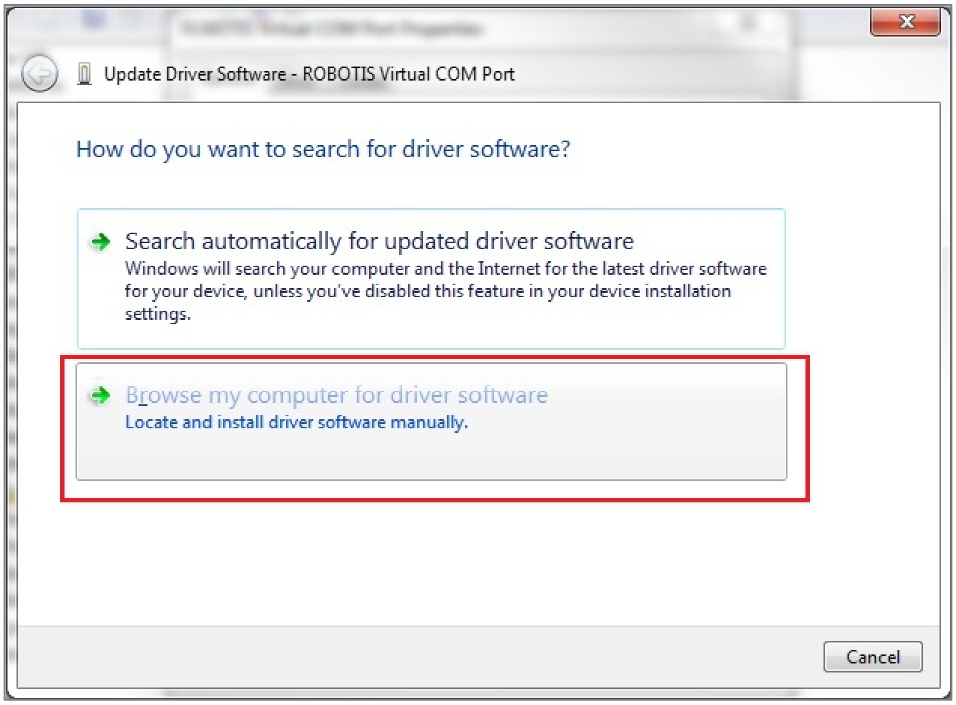
3.2.8 Tools : ROBOTIS CM-9's processing-related tools

3.2.9 ROBOTIS CM-9.exe : ROBOTIS CM-9's executable

3.3 USB drivers installation

The CM-900 USB driver installation is an essential requirement. The following procedure is Windows-specific. Mac OSX and Linux users do not need the following procedure as drivers are already included with the OS.

When the CM-900 is connected to the PC it will appear as ROBOTIS Virtual COM Port in Windows Device Manager. With the right mouse click select update driver software.



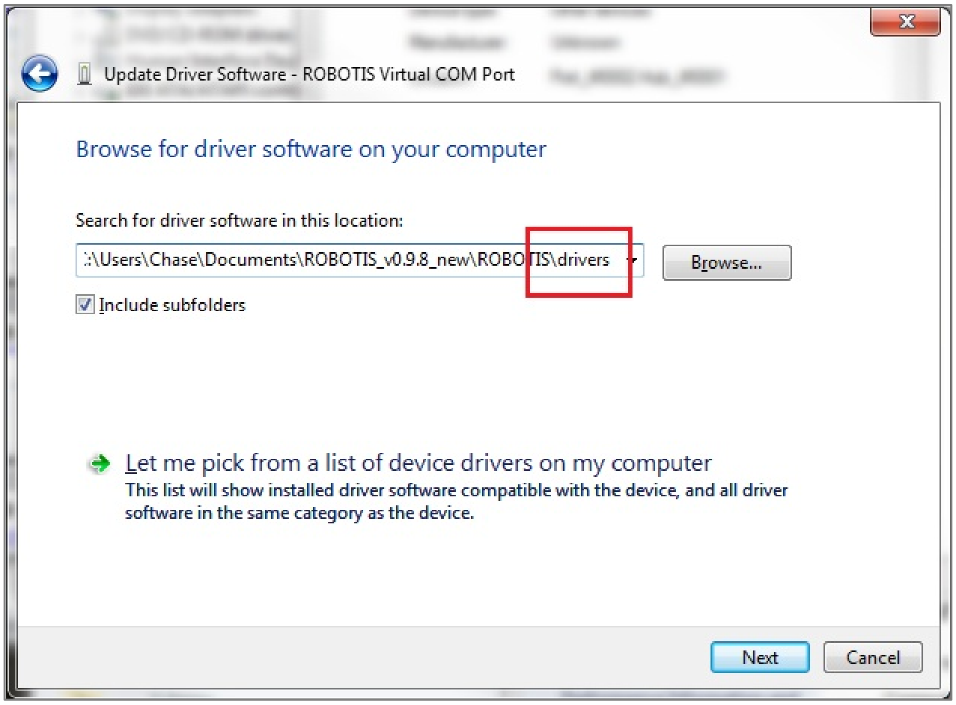
Update Driver Software - ROBOTIS Virtual COM Port

How do you want to search for driver software?

Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.

Pick “browse my computer for driver software”



Update Driver Software - ROBOTIS Virtual COM Port

Browse for driver software on your computer

Search for driver software in this location:

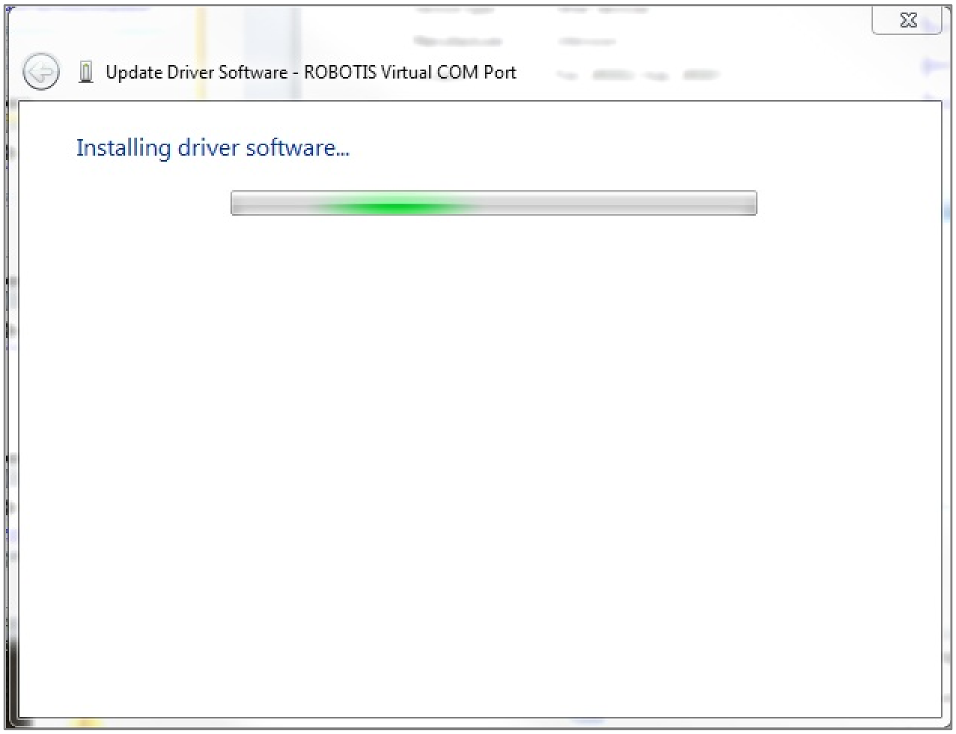
:\Users\Chase\Documents\ROBOTIS_v0.9.8_new\ROBOTIS\drivers

☑ Include subfolders

Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

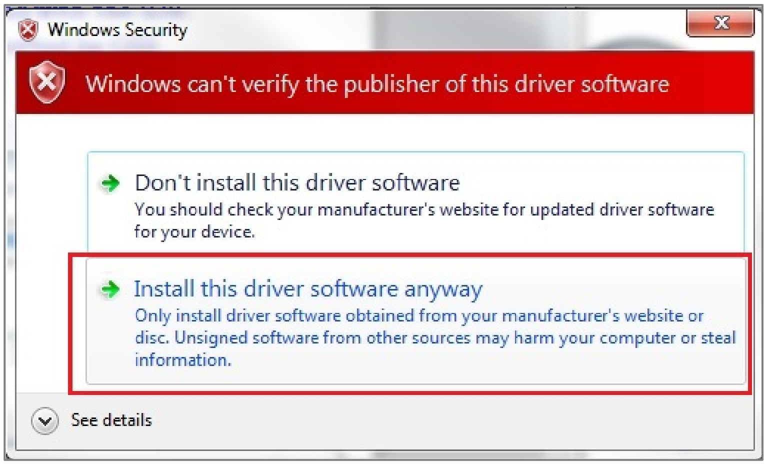
Click on “browse” and select ‘drivers’ folder (from ROBOTIS\drivers).



Update Driver Software - ROBOTIS Virtual COM Port

Installing driver software...

Click on “install this driver software anyway”



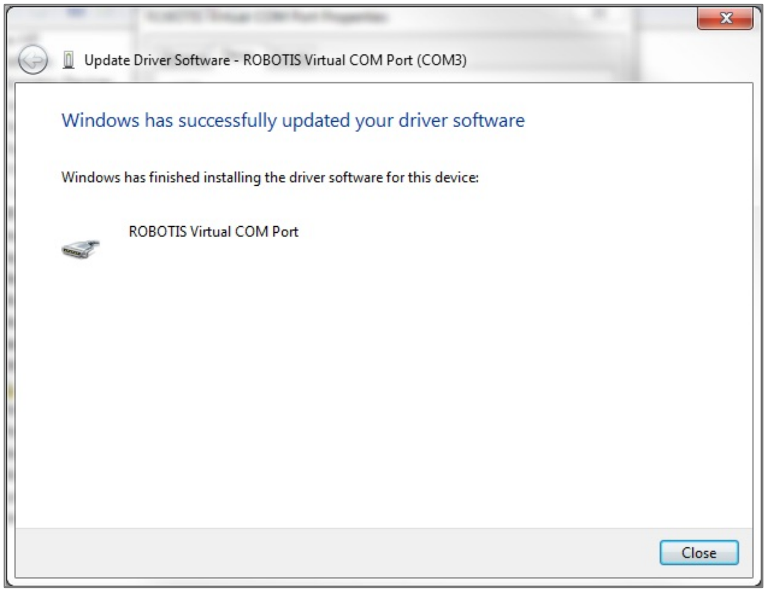
Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Once install is successful a window will appear as illustrated below



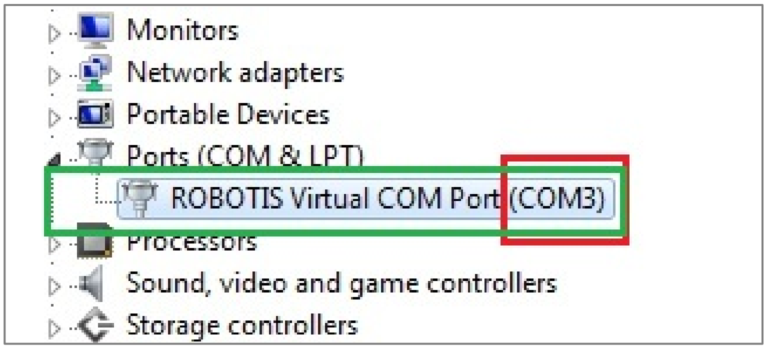
Update Driver Software - ROBOTIS Virtual COM Port (COM3)

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

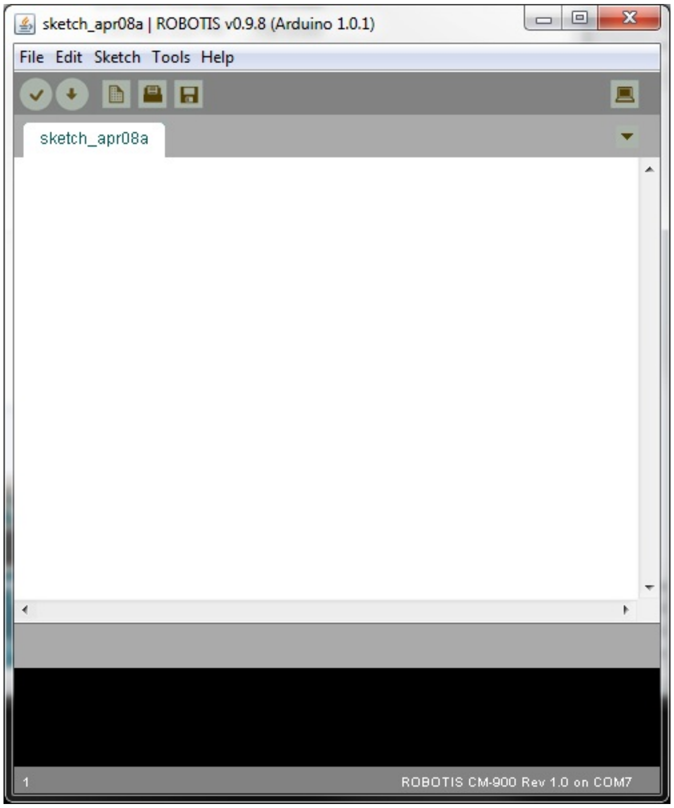
ROBOTIS Virtual COM Port

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



3.4 Software environment setup

After USB driver setup double-click on ROBOTIS CM-9.exe.

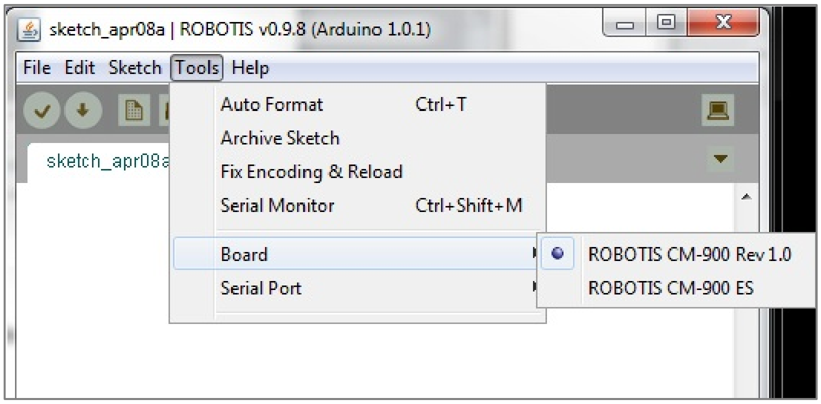

sketch_apr08a | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
sketch_apr08a
1
ROBOTIS CM-900 Rev 1.0 on COM7

<ROBOTIS CM-9 window>

From ROBOTIS CM-9 window you must select a board type and COM number.

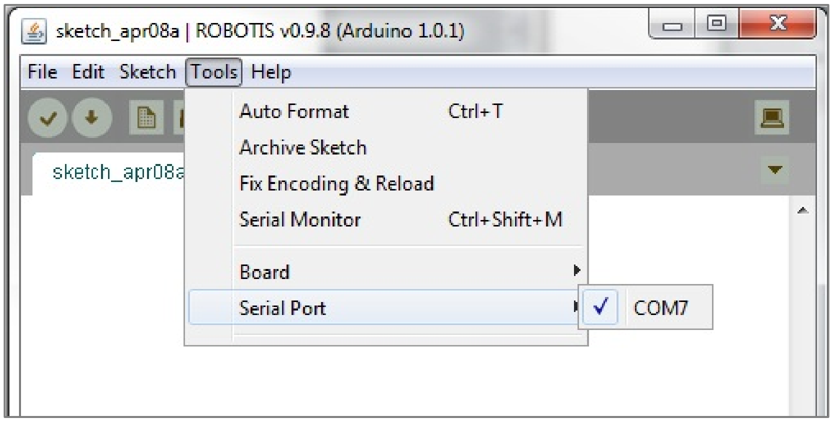
3.4.1 Select a board

Select the matching version of your CM-900 board. In this case select CM-900 REV 1.0 (ROBOTIS CM-900 ES is for previous test versions).

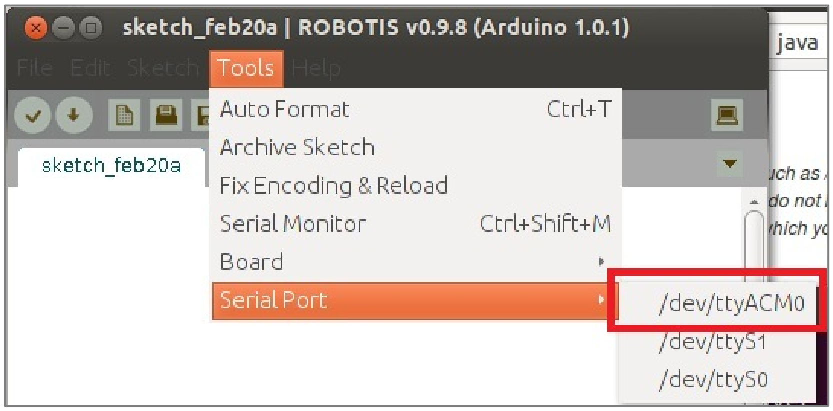


3.4.2 Select serial port

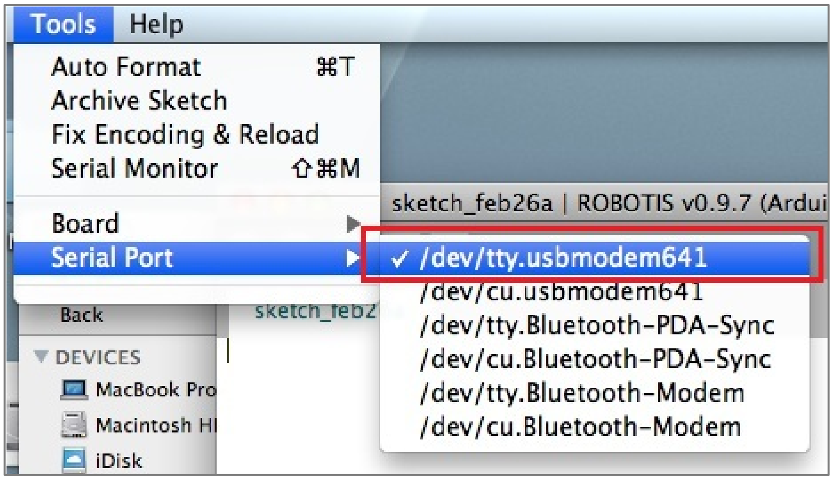
Select the COM port number.


sketch_apr08a | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
sketch_apr08a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
✓ COM7

Linux users select /dev/ttyACMX.

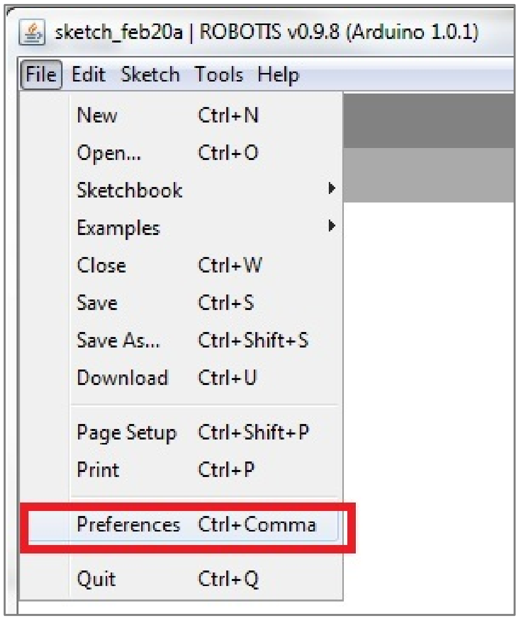


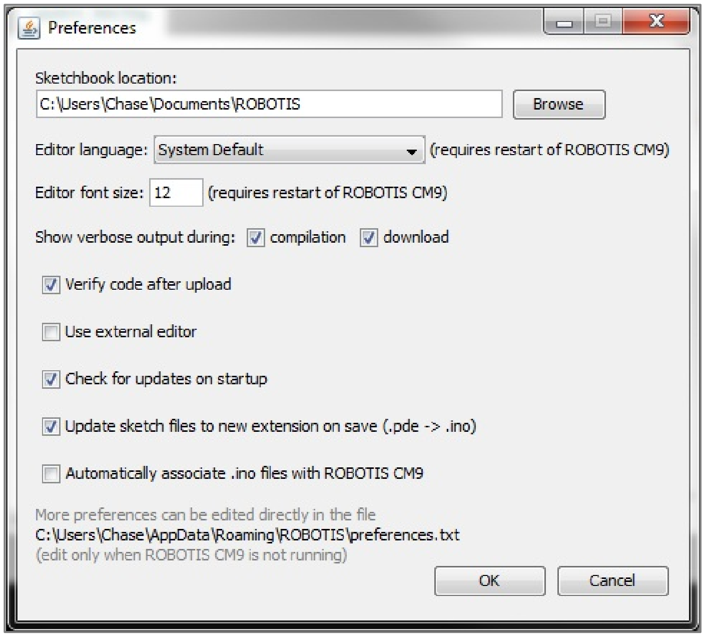
Mac OS X users select tty.usbmodemX11.



3.4.3 Environment setup

Go to File -> Preferences environment to make changes.





Sketchbook location:
C:\Users\Chase\Documents\ROBOTIS

Editor language: System Default (requires restart of ROBOTIS CM9)

Editor font size: 12 (requires restart of ROBOTIS CM9)

Show verbose output during: ☑ compilation ☑ download

☑ Verify code after upload

☐ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

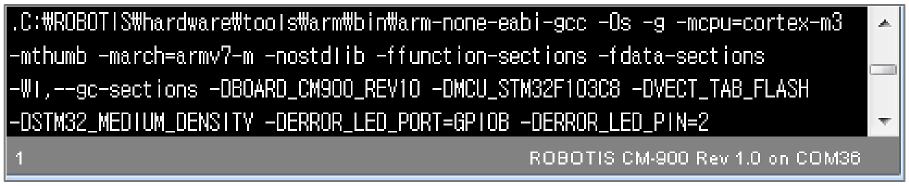
☐ Automatically associate .ino files with ROBOTIS CM9

More preferences can be edited directly in the file
C:\Users\Chase\AppData\Roaming\ROBOTIS\preferences.txt
(edit only when ROBOTIS CM9 is not running)

Sketchbook location: directory for sketch-based projects including examples.

Editor language: change font type.

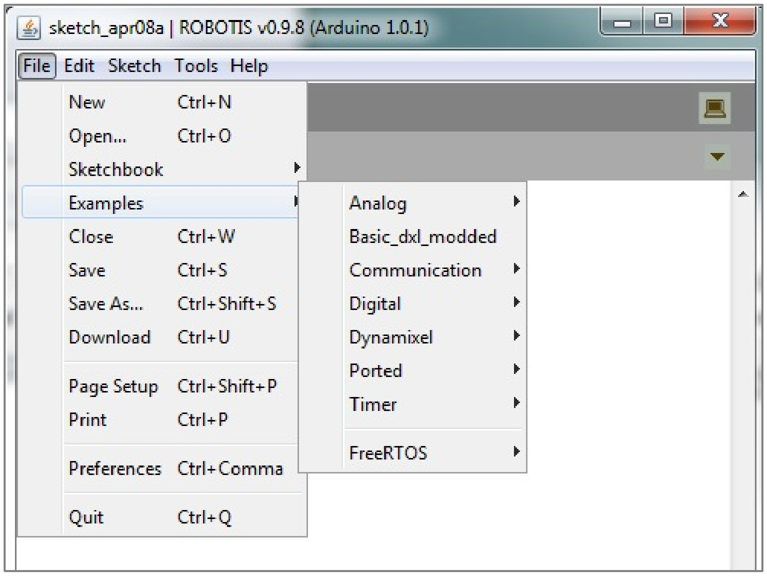
Console window : view the compilation’s output. Check download to download code after compilation.



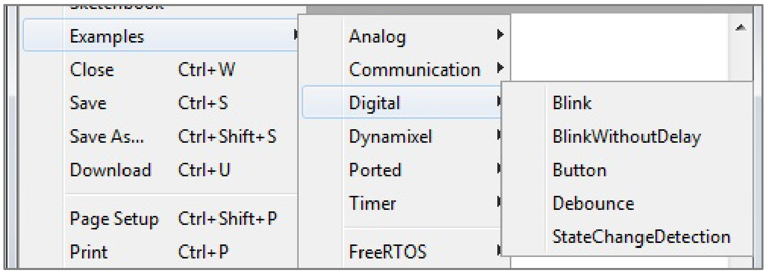
```
.C:₩ROBOTIS₩hardware₩tools₩arm₩bin₩arm-none-eabi-gcc -Os -g -mcpu=cortex-m3
-mthumb -march=armv7-m -nostdlib -ffunction-sections -fdata-sections
-Wl,--gc-sections -DBOARD_CM900_REV10 -DMCU_STM32F103C8 -DVECT_TAB_FLASH
-DSTM32_MEDIUM_DENSITY -DERROR_LED_PORT=GPIOB -DERROR_LED_PIN=2
1                                                ROBOTIS CM-900 Rev 1.0 on COM36
```

3.5 Download examples

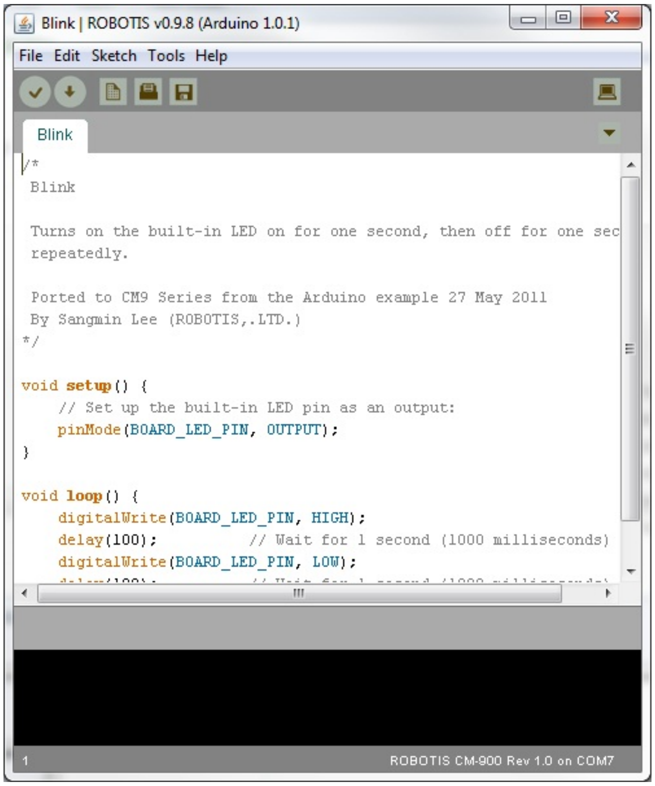
Get ROBOTIS CM-9 example programs from file -> examples.



For example: with Digital I/O open the Blink example, analyze the code then download it to the CM-900. This should help make development easier.



In this Blink example shown simply click on the downwards arrow to download the code to the CM-900.


Blink | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Blink

```
/*
 Blink

 Turns on the built-in LED on for one second, then off for one sec
 repeatedly.

 Ported to CM9 Series from the Arduino example 27 May 2011
 By Sangmin Lee (ROBOTIS,.LTD.)
*/

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);
```

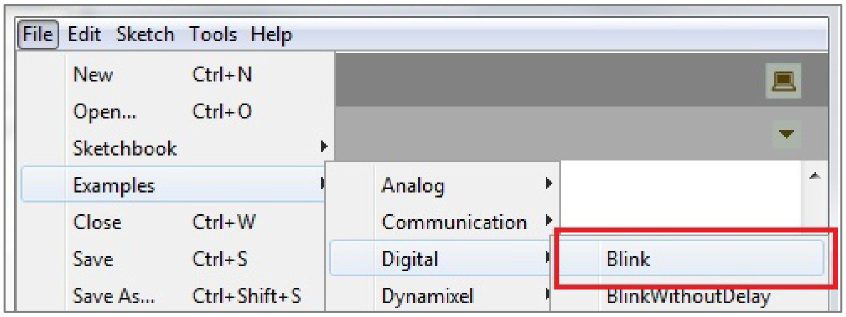
```
1                                ROBOTIS CM-900 Rev 1.0 on COM7
```

The following examples are useful for API reference. Please refer to these when developing the CM-900.

3.6 Blink example

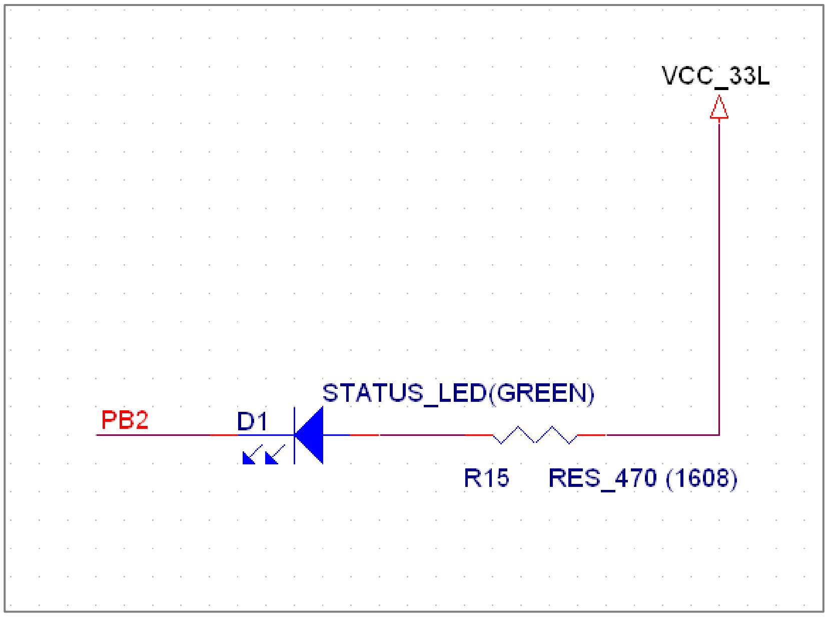
The CM-900 Blink example is a port of Arduino's Blink example.

Go to File -> Examples -> Digital -> Blink



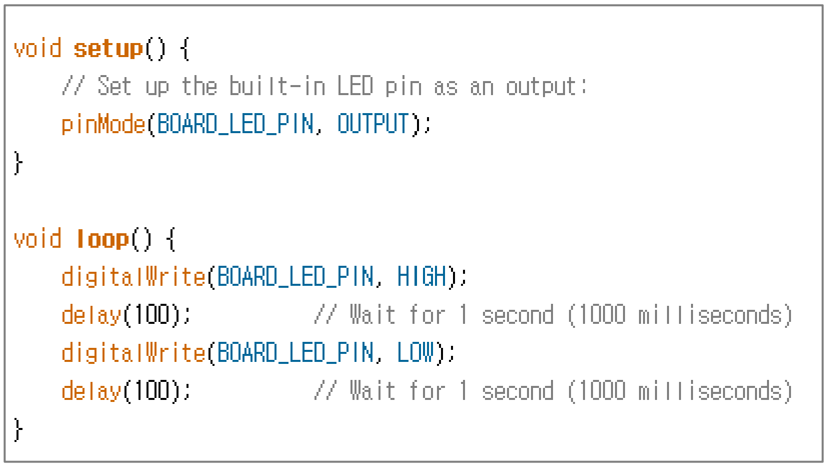
3.6.1 Schematic

The CM-900's status LED connects to the CPU via D16(PB2).



When D16(PB2) is high the LED is off; when low the LED is on.

3.6.2 Sketch code

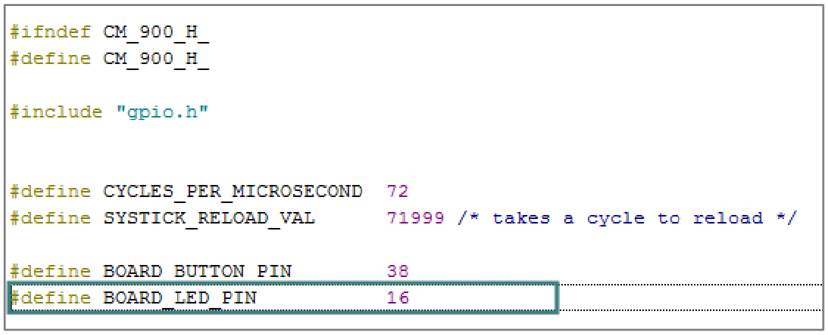


```
void setup() {
    // Set up the built-in LED pin as an output:
    pinMode(BOARD_LED_PIN, OUTPUT);
}

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 D16. This is illustrated in the header file CM-900.h.

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



```
#ifndef CM_900_H_
#define CM_900_H_

#include "gpio.h"


#define CYCLES_PER_MICROSECOND  72
#define SYSTICK_RELOAD_VAL      71999 /* takes a cycle to reload */

#define BOARD_BUTTON_PIN        38
#define BOARD_LED_PIN           16
```

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).

3.6.3 Verify data

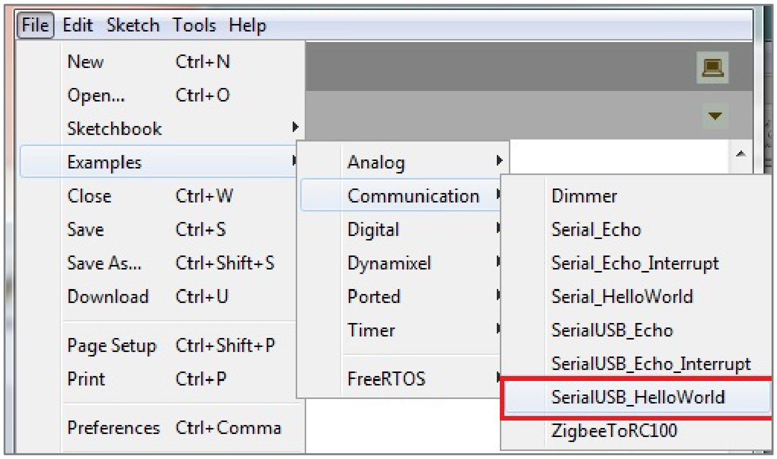
Verify the STATUS LED (on or off)

3.7 SerialUSB_HelloWorld example

This is an example to communicate between the CM-900 and external device (i.e. PC) via USB. Declare SerialUSB instance to enable USB communications.

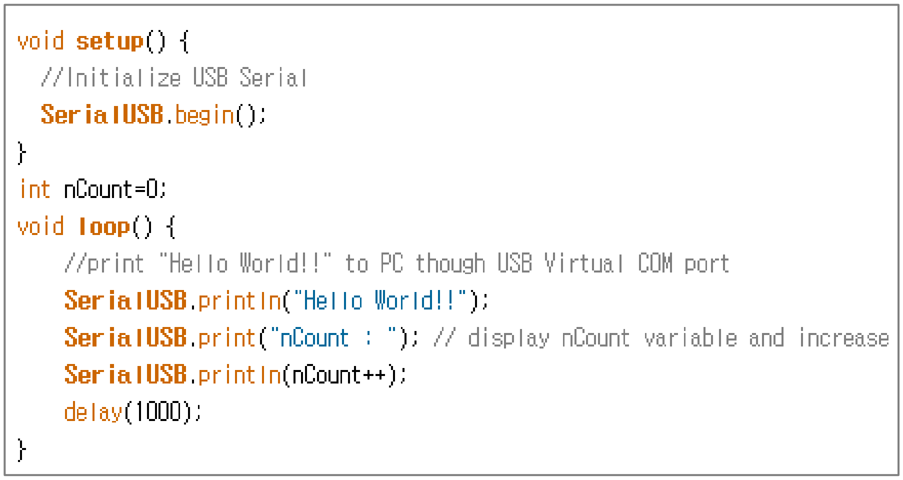
This example show how SerialUSB_HelloWorld communicated with a terminal window (PC).

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



3.7.1 Connect the CM-900 to the PC

3.7.2 Sketch code



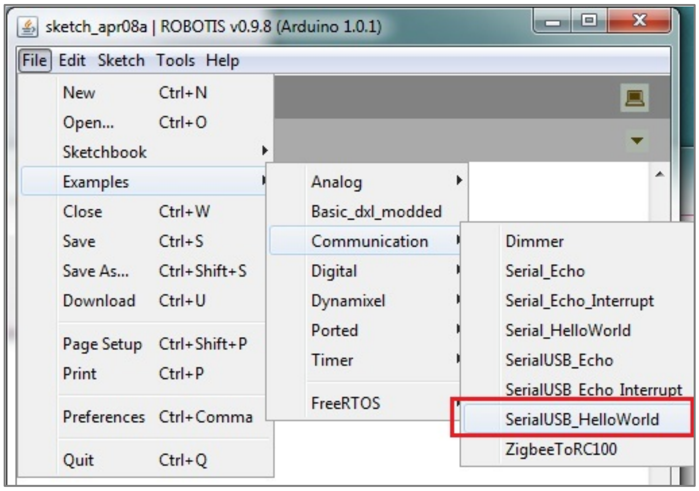
```
void setup() {
  //Initialize USB Serial
  SerialUSB.begin();
}
int nCount=0;
void loop() {
    //print "Hello World!!" to PC though USB Virtual COM port
    SerialUSB.println("Hello World!!");
    SerialUSB.print("nCount : "); // display nCount variable and increase
    SerialUSB.println(nCount++);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

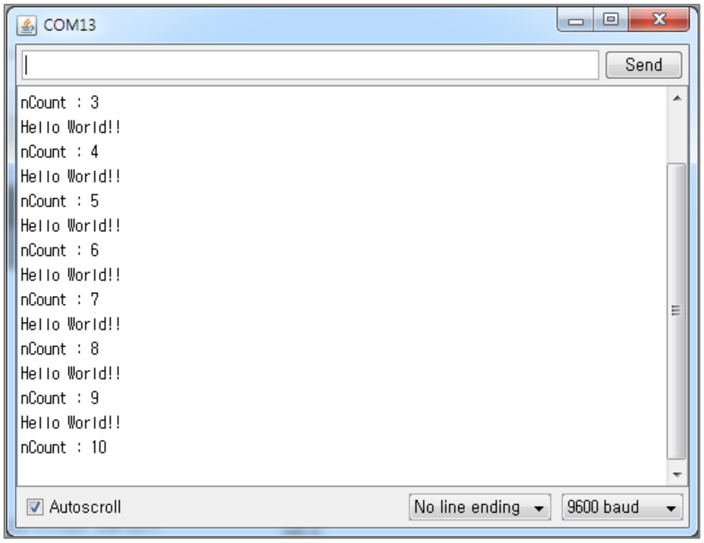
In Loop() with SerialUSB.print() or SerialUSB.println() its possible to get output.

3.7.3 Verify data

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



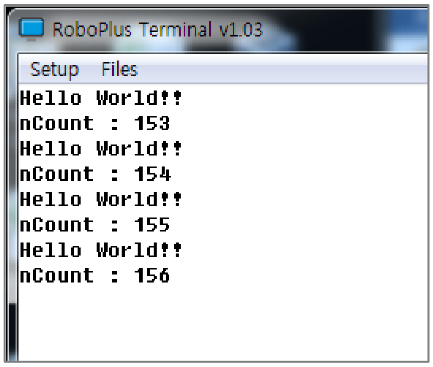
The serial monitor window can be activated by clicking on the laptop icon located on the upper right side.



```
nCount : 3
Hello World!!
nCount : 4
Hello World!!
nCount : 5
Hello World!!
nCount : 6
Hello World!!
nCount : 7
Hello World!!
nCount : 8
Hello World!!
nCount : 9
Hello World!!
nCount : 10
```

☑ Autoscroll No line ending 9600 baud

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



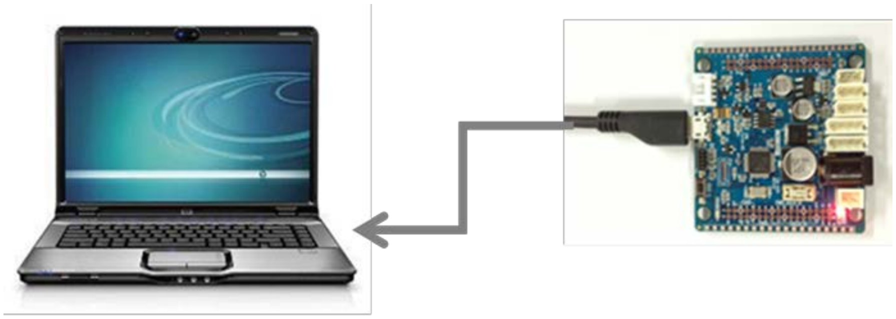
```
Hello World!!
nCount : 153
Hello World!!
nCount : 154
Hello World!!
nCount : 155
Hello World!!
nCount : 156
```

Other terminal window applications are not yet supported.

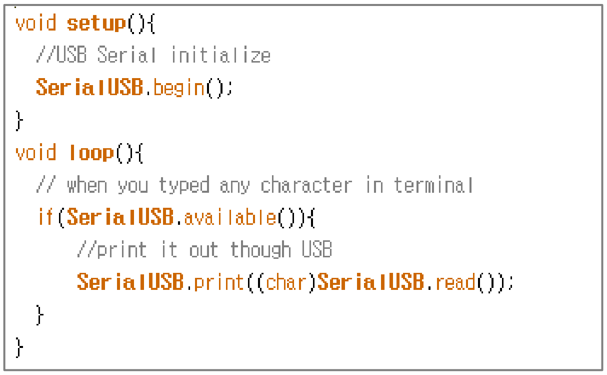
3.8 SerialUSB_Echo example

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

3.8.1 Connect the CM-900 to the PC



3.8.1 Sketch code



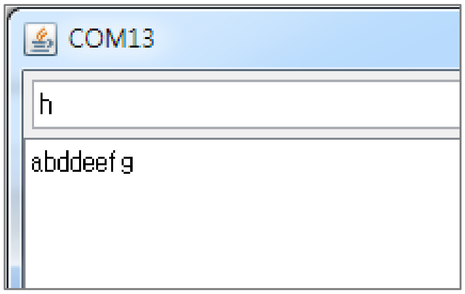
```
void setup(){
  //USB Serial initialize
  SerialUSB.begin();
}
void loop(){
  // when you typed any character in terminal
  if(SerialUSB.available()){
      //print it out though USB
      SerialUSB.print((char)SerialUSB.read());
  }
}
```

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().

3.8.2 Verify data

Use the serial monitor or RoboPlus Termincal to view data. Use the keyboard to input data and the CM-900 returns the same input as output, therefore is an echo.



COM13

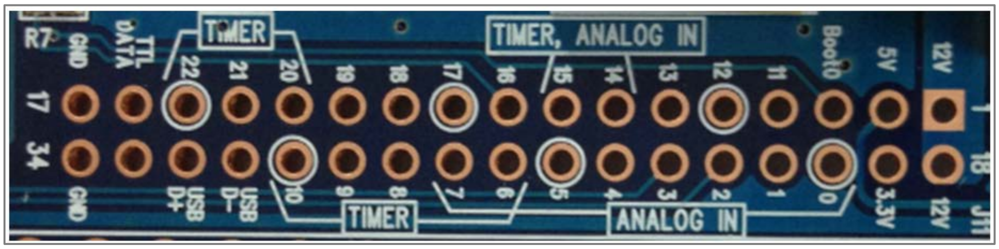
```
h
```

```
abddeefg
```

Any input is returned exactly as output.

3.9 AnalogInSerial example

The CM-900 has a 12-bit resolution ADC with 10 ports. This makes possible to connect multiple devices. The silk screen below shows the available ports.



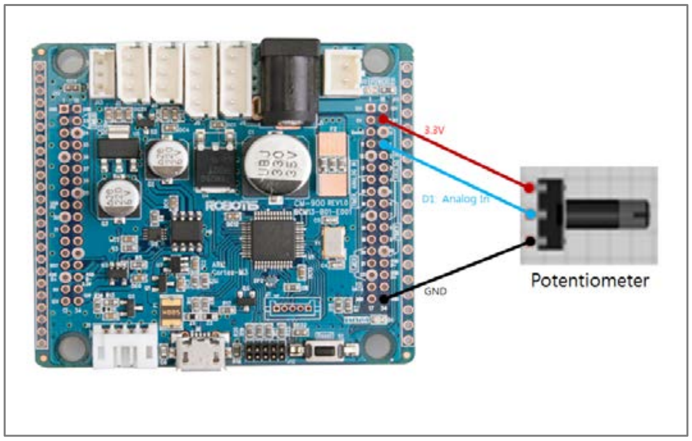
Pins 14 and 15 are for TIMER, ANALOG IN respectively and are duplicated features; with analog input via pinMode() function it is possible to set in analog mode. Input pins 0 through 7, 14 and 15 are available for analog input.

With AnalogInSerial example an analog input received then transmitted via SerialUSB.

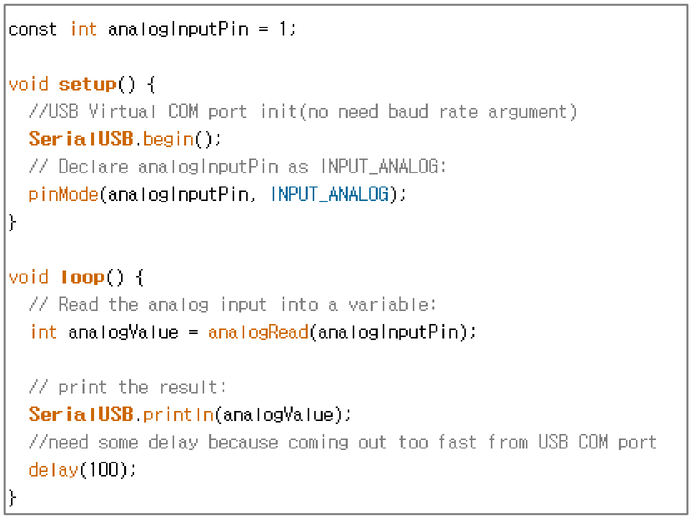
This example is accredited to Tom Igoe for Arduino's board therefore this is an Arduino example.

3.9.1 Schematics

The CM-900 is connected to a variable resistor (potentiometer). The important point is that the maximum allowed input voltage of the CM-900 for analog inputs is 3.3V. The schematic below the variable resistor is implemented to limit the voltage to 3.3V.



3.9.2 Sketch code



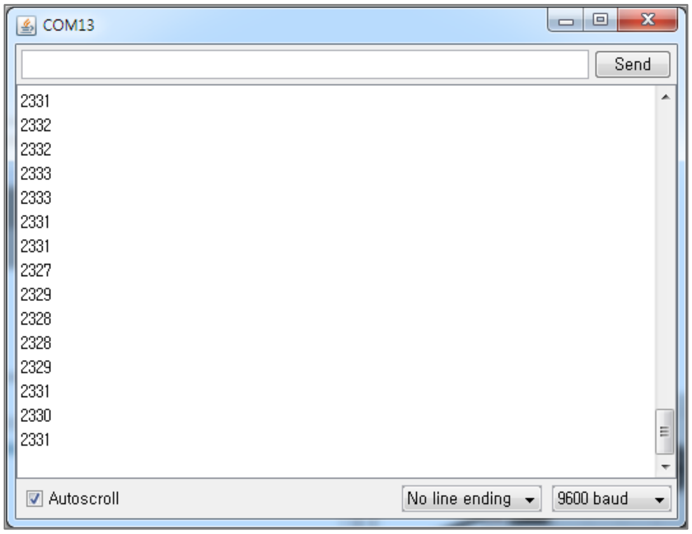
```
const int analogInputPin = 1;

void setup() {
  //USB Virtual COM port init(no need baud rate argument)
  SerialUSB.begin();
  // Declare analogInputPin as INPUT_ANALOG:
  pinMode(analogInputPin, INPUT_ANALOG);
}

void loop() {
  // Read the analog input into a variable:
  int analogValue = analogRead(analogInputPin);

  // print the result:
  SerialUSB.println(analogValue);
  //need some delay because coming out too fast from USB COM port
  delay(100);
}
```

This example shows 2 declarations in setup(). In loop() int analogValue repeatedly looks for analogRead(pin_number) for analog input. The input value is of integer type with 12 bits in range (0-4095).

SerialUSB.println() outputs value(s) from analogValue. If a hexadecimal value output is desired then set SerialUSB.println(analogValue,16), where 16 denotes hexadecimal; for Binary then 2; octal then 8. The default value is in decimal.

3.9.3 Verify data

Open the serial monitor to see output.



```
2331
2332
2332
2333
2333
2331
2331
2327
2329
2328
2328
2329
2331
2330
2331
```

☑ Autoscroll

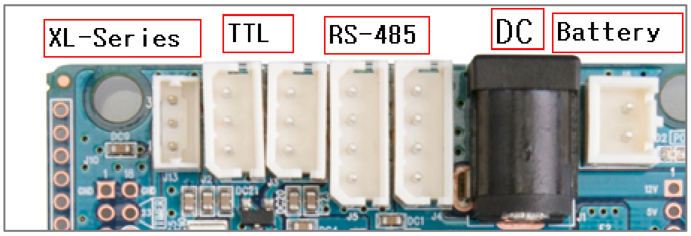
No line ending

9600 baud

3.10 Dynamixel Basic example

The CM-900 includes Dynamixel connectors to facilitate robot development. A pair of 3-pin TTL, a pair of 4-pin RS-485, and a XL-series connector are embedded onto the board. Also, a DC jack and battery connector are also embedded so power can be properly supplied to any connected Dynamixel(s).

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

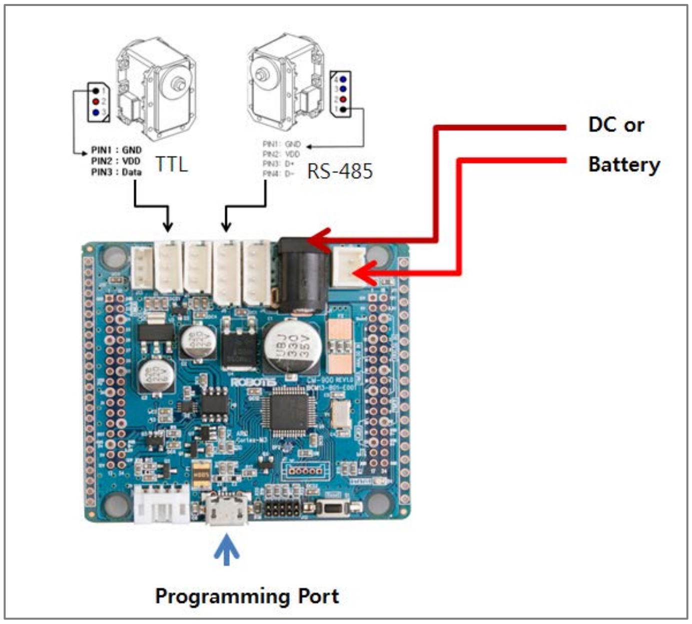


3.10.1 Connecting a Dynamixel

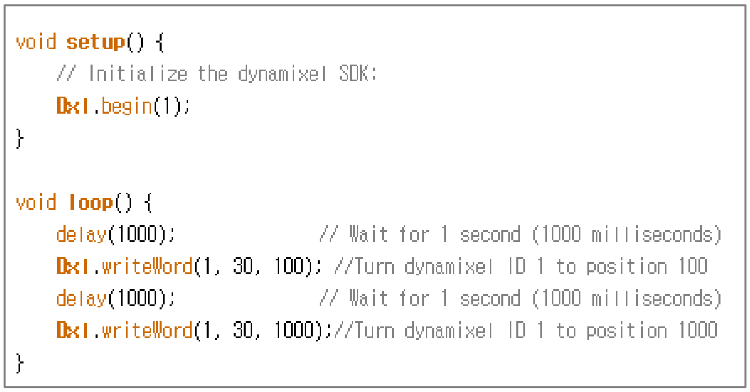
Connect a 3-pin or 4-pin DYnamixel. Connect the SMPS or battery to the CM-900 then run ROBOTIS CM9.

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 CM-900 communicates with Dynamixel serially.



3.10.2 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 CM-900 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.

Note that the actual position varies with different models of Dynamixel. For Dynamixelw with 12-bit resolution (0~4095, 0xFFF) will have a smaller range of motion and reach goal position quicker. For more information on goal position please consult the e-manuals.

3.10.3 Verify data

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

3.11 Dynamixel ReadWrite example

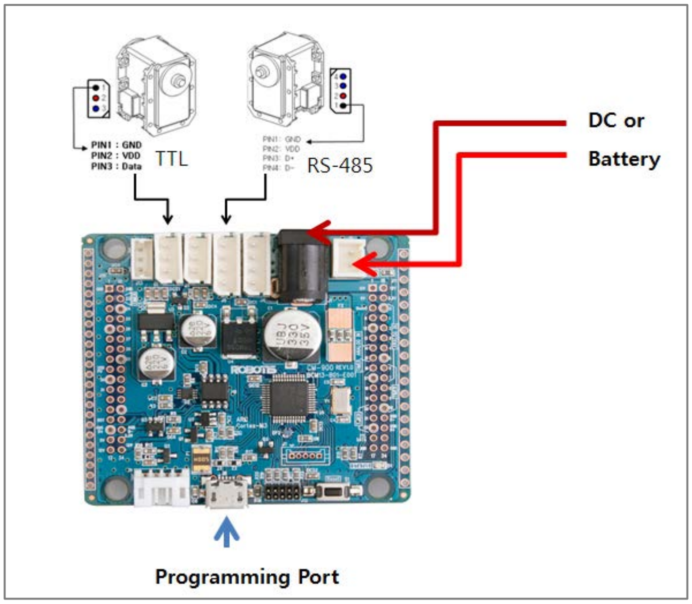
This example shows Dynamixel read/write features. This example checks Dynamixel movement and change of moving (rotating) direction. Once moving is complete position data is then outputted and Dynamixel moves to the next position.

3.11.1 Connect Dynamixel

Connect a 3-pin or 4-pin DYnamixel. Connect the SMPS or battery to the CM-900 then run ROBOTIS CM9.

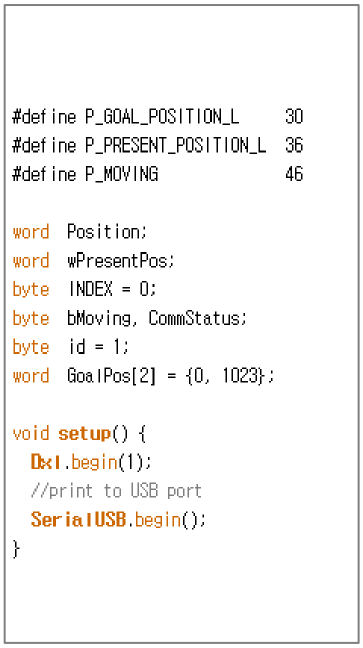
The default values for Dynamixel are 1 for ID and 1 for baud rate (1Mbps). If not, then set said values with Dynamixel Wizard.

The CM-900 communicates with Dynamixel serially.

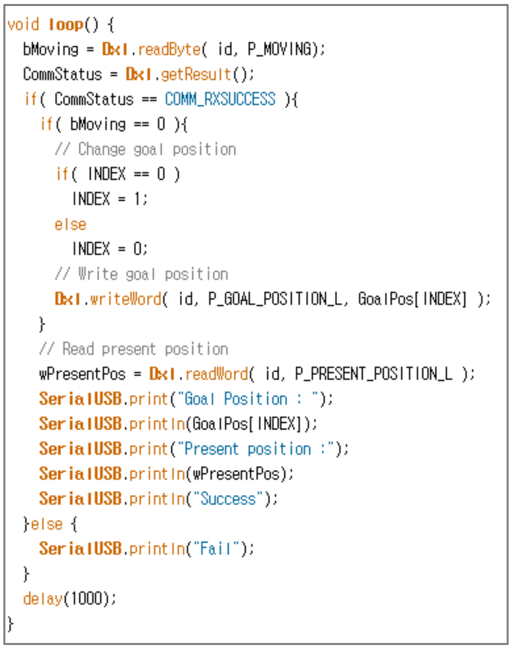


3.11.2 Sketch code

Some of the parameters from Dynamixel control table have been dfined 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);
}
```

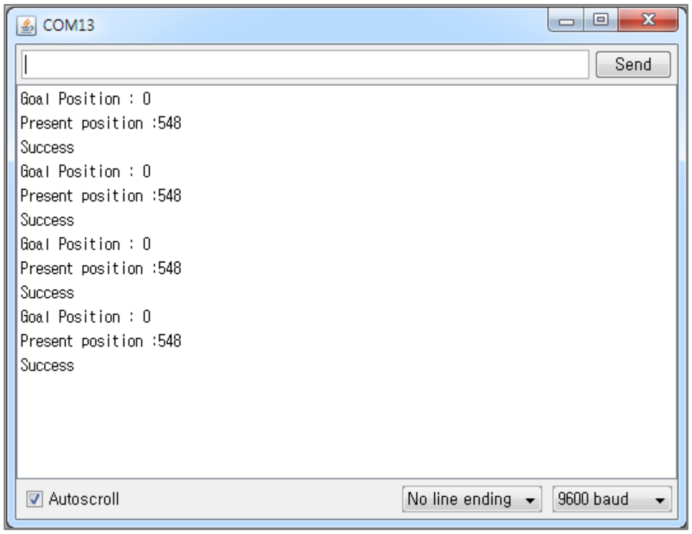
bMoving = Dxl.readByte(id, P_MOVING)

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);

3.11.3 Verify data

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



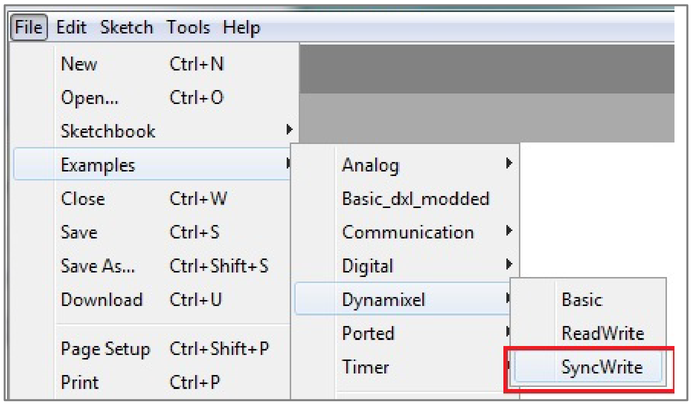
```
Goal Position : 0
Present position :548
Success
Goal Position : 0
Present position :548
Success
Goal Position : 0
Present position :548
Success
Goal Position : 0
Present position :548
Success
```

3.12 Dynamixel SyncWrite example

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

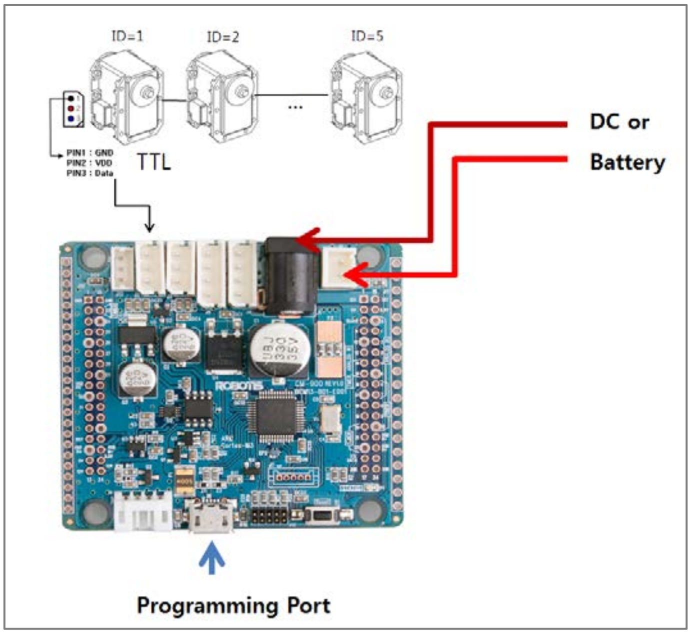
This example shows how to control 5 Dynamixels via Syncwrite packet. For more information on Syncwrite please consult the e-manuals. http://support.robotis.com/ko/e-manual_kor.htm#product/dynamixel/communication/dxl_instruction.htm

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



3.12.1 Connect 5 Dynamixels

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.

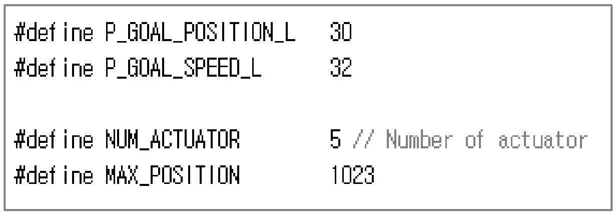


The CM-900 communicates with the Dynamixels serially.

3.12.2 Sketch code

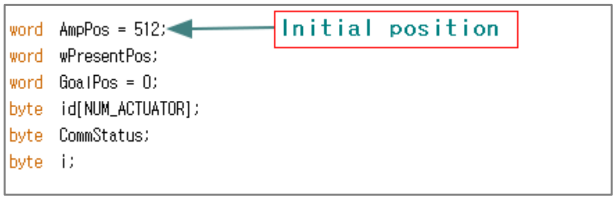
Some of the parameters from Dynamixel control table have been defined in the preprocessor. For more information on Dynamixel control table please consult the e-manuals.

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



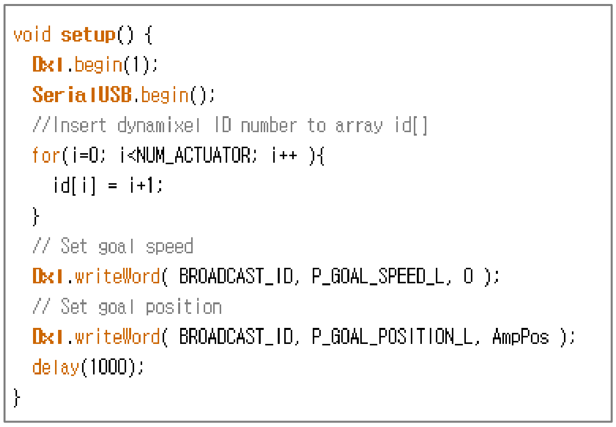
```
#define P_GOAL_POSITION_L   30
#define P_GOAL_SPEED_L      32

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

AmpPos is the initial position of all 5 Dynamixels.



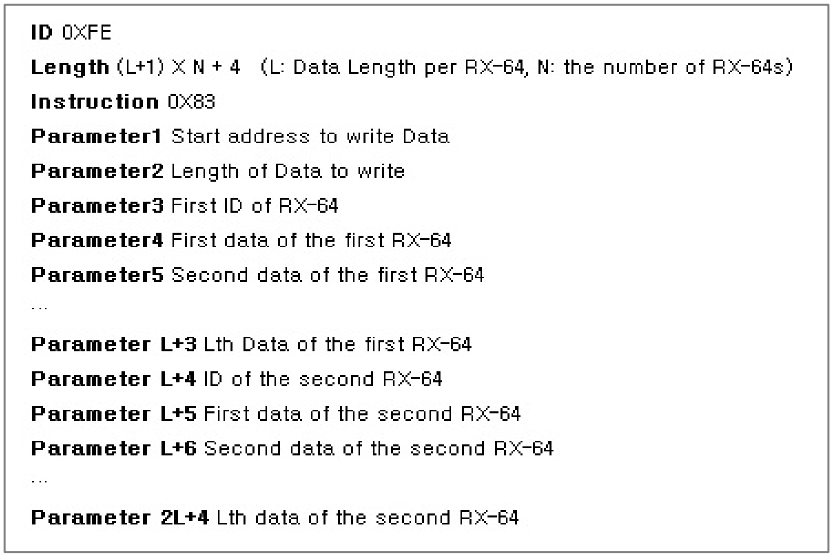
```
word  AmpPos = 512;
word  wPresentPos;
word  GoalPos = 0;
byte  id[NUM_ACTUATOR];
byte  CommStatus;
byte  i;
```

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: Data Length per RX-64, N: the number of RX-64s)

Instruction 0X83

Parameter1 Start address to write Data

Parameter2 Length of Data to write

Parameter3 First ID of RX-64

Parameter4 First data of the first RX-64

Parameter5 Second data of the first RX-64

…

Parameter L+3 Lth Data of the first RX-64

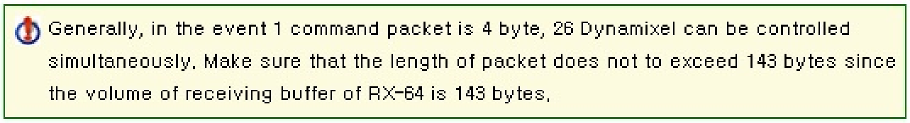
Parameter L+4 ID of the second RX-64

Parameter L+5 First data of the second RX-64

Parameter L+6 Second data of the second RX-64

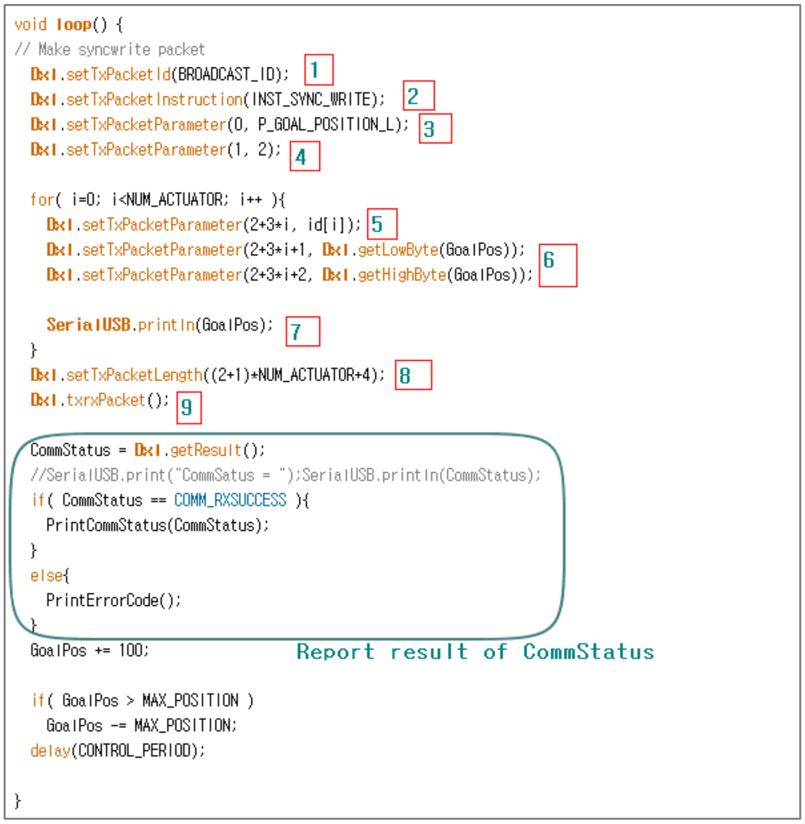
…

Parameter 2L+4 Lth data of the second RX-64



Generally, in the event 1 command packet is 4 byte, 26 Dynamixel can be controlled simultaneously. Make sure that the length of packet does not to exceed 143 bytes since the volume of receiving buffer of RX-64 is 143 bytes.

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);
  Dxl.setTxPacketInstruction(INST_SYNC_WRITE);
  Dxl.setTxPacketParameter(0, P_GOAL_POSITION_L);
  Dxl.setTxPacketParameter(1, 2);

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

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

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

  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

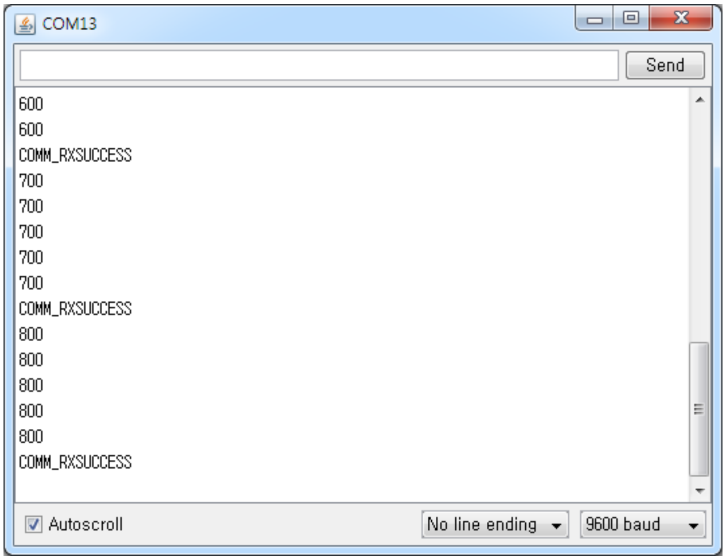
#8: calculates Packet length (see below)

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

Verify data

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



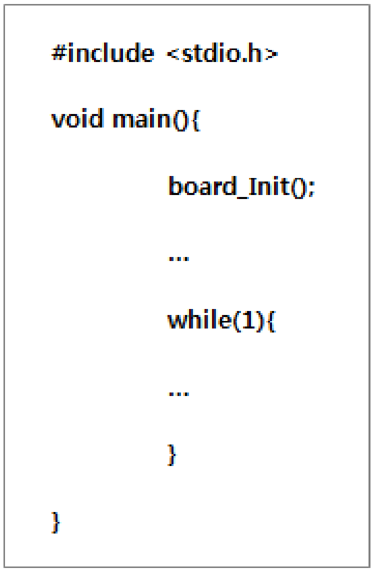
```
600
600
COMM_RXSUCCESS
700
700
700
700
700
COMM_RXSUCCESS
800
800
800
800
800
COMM_RXSUCCESS
```

4 CM-9 API Reference

This CM-900 API Reference documentation has been created by Martin Mason of Mt. San Antonio College of Physics and Engineering. Martin Mason. Thanks to Martin Mason for the development and contribution of the CM-9 series.

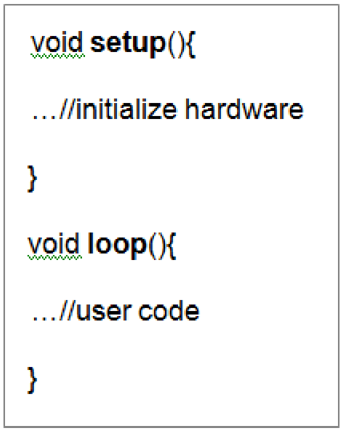
4.1 CM-9 code structure

Usually, firmware codes begin with main.c, or main.cpp with void main().



```
#include <stdio.h>

void main(){

        board_Init();

        ...

        while(1){

        ...

        }

}
```

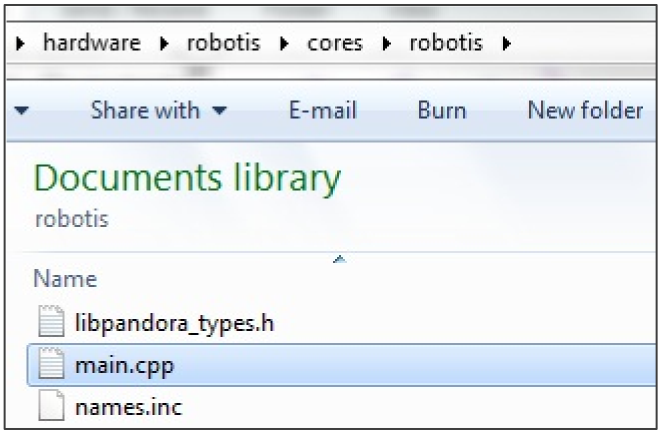
Default hardware initialization is in board_Init() function and code implemented in infinite while or for loops.
This way code structure of the CM-9 can be divided in hardware and parts.
Initialize hardware in setup(){}. Place algorithm under loop(){}.



```
void setup(){

…//initialize hardware

}

void loop(){

…//user code

}
```

Both setup() and loop() reside in main.cpp; the user can create a downloadable binary file by implementing these.

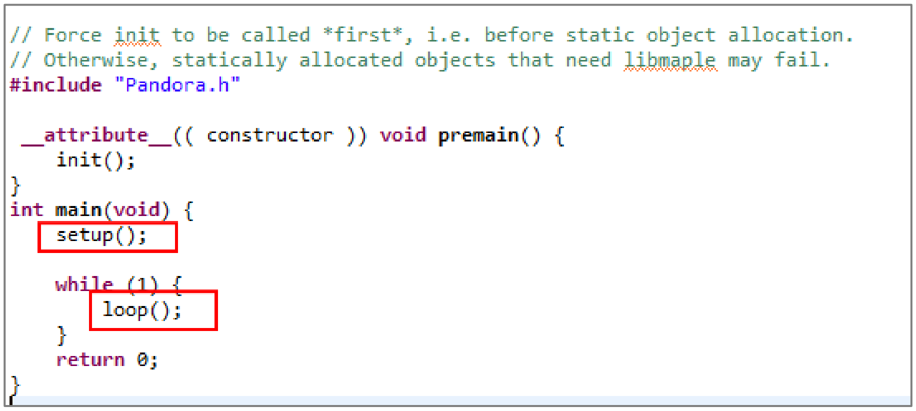
ROBOTIS\hardware\robotis\cores\robotis\main.cpp



Documents library
robotis

Name
libpandora_types.h
main.cpp
names.inc

Open main.cpp…



```
// Force init to be called *first*, i.e. before static object allocation.
// Otherwise, statically allocated objects that need libmaple may fail.
#include "Pandora.h"

 __attribute__(( constructor )) void premain() {
    init();
}
int main(void) {
    setup();

    while (1) {
        loop();
    }
    return 0;
}
```

4.2 Dynamixel API

Use Dynamixel class to control Dynamixel(s). To drive Dynamixel(s) Dxl.begin(baur_rate) is required. Dynamixel class methods are based on Dynamixel SDK.
For more information on Dynamixel please consult the e-manuals.
http://support.robotis.com/

Methods:

Device Control Methods	void begin(int baud)	initialize Dynamixel at set baud rate.
	void end(void)	Pause Dynamixel
High Level Communications	int readByte(int id, int address)	Read 1 byte from Dynamixel
	void writeByte(int id, int address, int value)	Write 1 byte to Dynamixel
	int readWord(int id, int address)	Read 1 word from Dynamixel
	void writeWord(int id, int address, int value)	Write 1word to Dynamixel
	void ping(int id)	Verify Dynamixel connection status
	void reset(int id)	Resets Dynamixel
	int getResult(void)	Get response
	void setPosition(int Servoid, int Position, int Speed)	Set position and velocity of Dynamixel ID

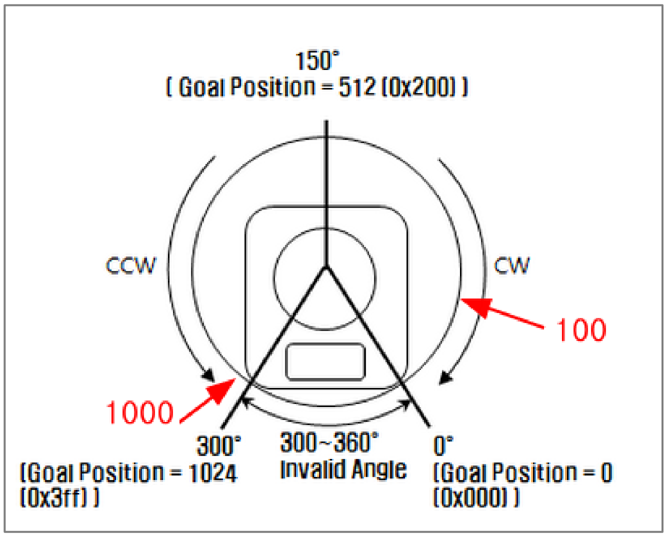
Packet Methods	void setTxPacketId(int id);
	void setTxPacketInstruction(int instruction)
	void setTxPacketParameter(int index, int value)
	void setTxPacketLength(int length)
	int getRxPacketParameter(int index)
	int getRxPacketLength(void)
	int getRxPacketError(int errbit)
Utility methods	int makeWord(int lowbyte, int highbyte)
	int getLowByte(int word)
	int getHighByte(int word)
Low Level Communications	void txPacket(void)
	void rxPacket(void);
	void txrxPacket(void)

For more information on packet-related methods, utility method, low-level methods please consult the e-manuals.
http://support.robotis.com

4.2.1 Getting Started:

The sketch code shown below sets a Dynamixel baud rate to 1Mbps with position switching between value 100 and 1000 with 1000ms pause in between.

```
void setup() {
     // sets Synamixel baud rate to 1Mbps.
     // for values on baud rates visit support.robotis.com
     Dxl.begin(1);
}


void loop() {
     delay(1000);                // wait for 1 second
     Dxl.writeWord(1, 30, 100); //set ID 1to goal position (value 30) of value 100
     delay(1000);                // wait for 1 second
     Dxl.writeWord(1, 30, 1000);// set ID 1to position (value 30) of value 1000
}
```



4.2.2 Code description

Every code requires setup() and loop().

Setup: setup() initializes the CM-900 upon power up or after pressing the reset button. Pin mode setup, device initialization also done in this function.

Dxl.Begin() Dxl assigned instance with begin() method initializes Dynamixel bus. Dxl class instance has more methods other than begin().

Loop: loop() runs setup() and repeatedly runs the CM-900
Dxl.writeword(1,30,100): the first value sets Dynamixel of ID 1. The second value setsgoal position (30 of control table). For more information on control table please visit support.robotis.com.

30 (0X1E)	Goal Position(L)	Lowest byte of Goal Position	RW	-
31 (0X1F)	Goal Position(H)	Highest byte of Goal Position	RW	-

delay (time in milliseconds) : is a set delay time.

4.2.3 Pre-defined constants:

These predefined constants are convenient and make defining unnecessary. Please refer to the predefined constants listed below.

Get Result Flags

Name	DEC
COMM_TXSUCCESS	0
COMM_RXSUCCESS	1
COMM_TXFAIL	2
COMM_RXFAIL	3
COMM_TXERROR	4
COMM_RXWAITING	5
COMM_RXTIMEOUT	6
COMM_RXCORRUPT	7

Instruction Commands

Name	Hex
INST_PING	0x01
INST_READ	0x02
INST_WRITE	0x03
INST_REG_WRITE	0x04
INST_ACTION	0x05
INST_RESET	0x06
INST_DIGITAL_RESET	0x07
INST_SYSTEM_READ	0x0C
INST_SYSTEM_WRITE	0x0D
INST_SYNC_WRITE	0x83
INST_SYNC_REG_WRITE	0x84

Packet Instructions

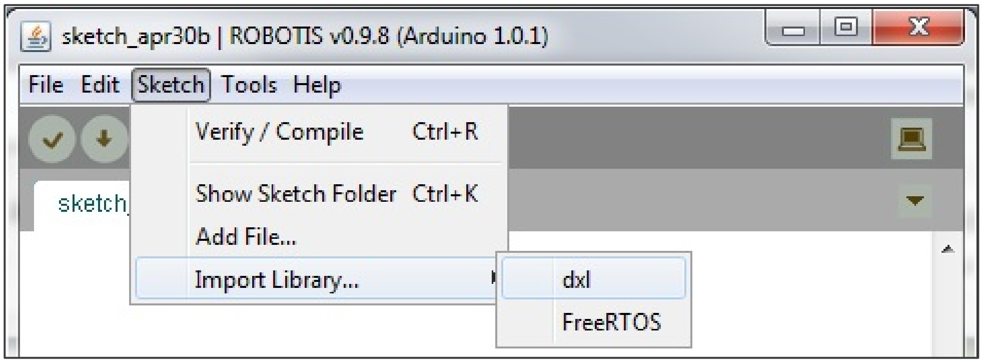
Name	DEC
BROADCAST_ID	254
DEFAULT_BAUDNUMBER	1
ID	2
LENGTH	3
INSTRUCTION	4
ERRBIT	4
PARAMETER	5
MAXNUM_RXPARAM	60
MAXNUM_TXPARAM	150

Error Messages

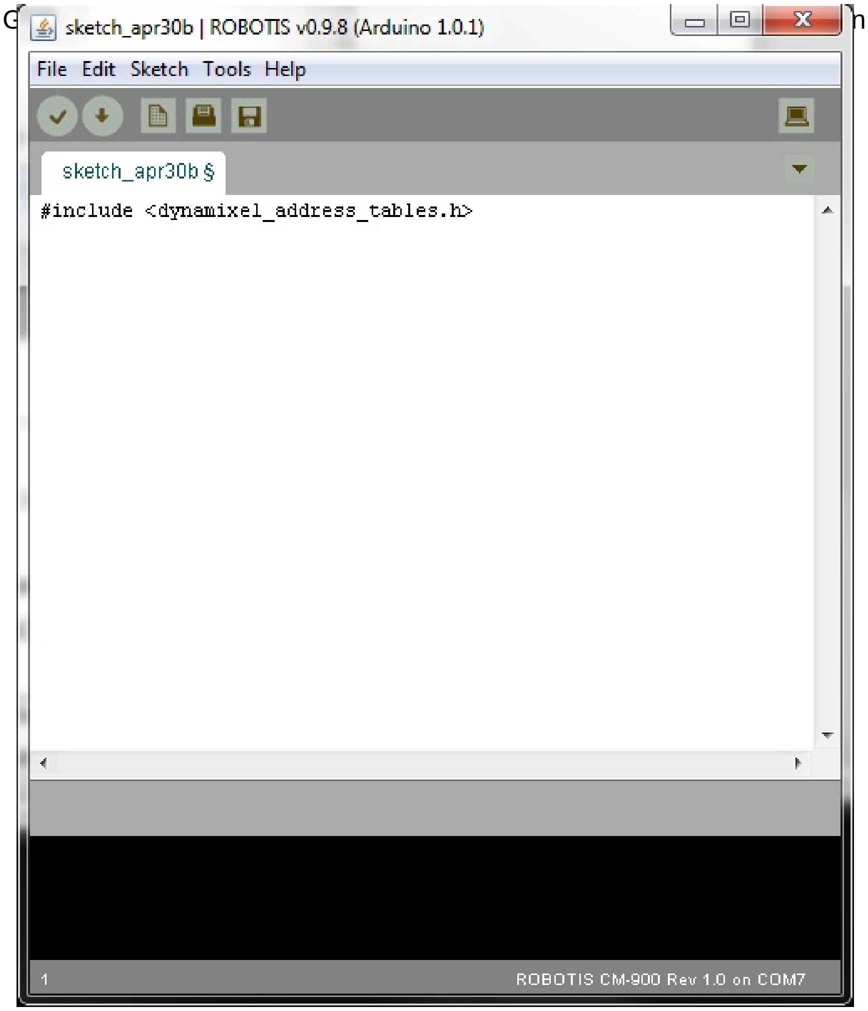
Name	DEC
ERRBIT_VOLTAGE	1
ERRBIT_ANGLE	2
ERRBIT_OVERHEAT	4
ERRBIT_RANGE	8
ERRBIT_CHECKSUM	16
ERRBIT_OVERLOAD	32
ERRBIT_INSTRUCTION	64

From the library folder including the header file dynamixel_address_tables.h provides an abundance of predefined constants. This also includes peripherals from ROBOTIS or third parties, such as HaViMo.

\ROBOTIS\libraries\dxl\dynamixel_address_tables.h


sketch_apr30b | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
Verify / Compile Ctrl+R
Show Sketch Folder Ctrl+K
Add File...
Import Library...
dxl
FreeRTOS
sketch_

G n below).


sketch_apr30b | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
sketch_apr30b §

```
#include <dynamixel_address_tables.h>
```

```
1                                   ROBOTIS CM-900 Rev 1.0 on COM7
```

Function Documentation: refer to ROBOTIS e-Manual v1.11.00

Dynamixel class is assigned in Dxl instance.

Implement Dxl in this form: Dxl.instance(1st value, 2nd value,…).

Ex) Dxl.begin(1); // initialize Dynamixel bus to 1Mbps.

Dxl.writeWord(2,30,512); // sets Dynamixel with ID 2 to goal position of 512.

void begin(int baud);

Initializes Dynamixel bus.

Parameters

- **int** baud

Baud is the value that determines communications speed. The relationship between baud rate and value is 200000 / (Value + 1). The following table lists the values and corresponding baud rates for each value. For example Dxl.begin(34) initializes Dynamixel bus with a baud rate of 57600bps.

Value	Actual BPS	Standard BPS	Uncertainty
0	2000000	2000000	0
1	1000000	1000000	0%
3	500000	500000	0%
4	400000	400000	0%
7	250000	250000	0%
9	200000	200000	0%
16	117647	115200	-2.124%
34	57142	57600	0.794%
103	19230	19200	-.16%
207	9615	9600	-.16%

Default is 1Mbps

Return Values

- A returned value of 1 means success; 0 for failure.

void end(void);

Pauses Dynamixel.

---

int readByte(int id, int address);

Reads 1 byte of address data of Dynamixel. Use from results of communications getResult().

Parameters

- **id**

ID of Dynamized

- **address**

Address from Dynamixel control table. For more information on Dynamixel control table visit support.robotis.com

Return Values

- read data values

Example

```
data = Dxl.readByte( 2, 36 );
if( Dxl.getResult( ) == COMM_RXSUCCESS )
{
    // using data
}
```

void writeByte(int id, int address, int value);

Writes 1 byte in address of Dynamixel control register.

Parameters

- **id**

IDof Dynamixel

- **address**

Address from Dynamixel control table. For more information on Dynamixel control table visit support.robotis.com

- **value**

Written data value

Return Values

- none

Example

```
Dxl.writeByte( 2, 19, 1 );
if( Dxl.getResult( ) == COMM_RXSUCCESS )
{
    // Succeed to write
}
```

int readWord(int id, int address);

A word is comprised of 2 bytes. A word is the control register address of the Dynamixel via its ID. Use this function to get a readout of the control register. For example, a read out of goal position (30 in address table) gives a word readout on for Goal Position (L) (30 in address table) and Goal Position (H) (31 in address table). Use getResult() method where communications is successful.

Parameters

- id

ID of Dynamixel

- address

Address value of Dynamixel from the control table

Return Values

- returns a word (2 bytes).

Example

```
data = Dxl.readWord( 2, 36 );
if( Dxl.getResult( ) == COMM_RXSUCCESS )
{
    // process data here.
}
```

void writeWord(int id, int address, int value);

A word is comprised of 2 bytes. A word is the control register address of the Dynamixel via its ID. Use this function to write the control register. For example, a write goal position (30 in address table) then the writeword writes for Goal Position (L) (30 in address table) + Goal Position (H) (31 in address table). Use getResult() method where communications is successful.

Parameters

- id

ID of Dynamixel

- **address**

Address value of Dynamixel from the control table

- value

Value to be writetn

Return Values

- None

Example

```
Dxl.writeWord( 2, 30, 512 );
if( Dxl.getResult( ) == COMM_RXSUCCESS )
{
    // Write success
}
```

void ping(int id);

Verifies Dynamixel bus for connected Dynamixel. Use Dxl.getResult() for verification.

Parameters

- id

ID of Dynamixel

Return Values

- None

Example

```
Dxl.ping( 2 );
if( Dxl.getResult( ) == COMM_RXSUCCESS )
{
    // verification of ID2 successful
}
```

void reset(int id);

Resets Dynamixel.

Parameters

- id

ID Dynamixel

Return Values

- none

Example

```
Dxl.reset( 2 );
if( Dxl.getResult( ) == COMM_RXSUCCESS )
{
    // reset ID 2
}
```

int getResult(void);

Checks for packet communications result.

Parameters

- None

Return Values

- Returns results. Value types shown below

Value	Meaning

COMM_TXSUCCESS	Instruction packet transmission successful
COMM_RXSUCCESS	Status packet reception successful
COMM_TXFAIL	Instruction packet transmission failed
COMM_RXFAIL	Status packet reception failed
COMM_TXERROR	Instruction Packet transmission error
COMM_RXWAITING	Status Packet reception error
COMM_RXTIMEOUT	Dynamixel not responding
COMM_RXCORRUPT	Status Packet corrupted

Example

```
result = Dxl.getResult( );
if( result == COMM_TXSUCCESS )
{
}
else if( result == COMM_RXSUCCESS )
{
}
else if( result == COMM_TXFAIL )
{
}
else if( result == COMM_RXFAIL)
{
}
else if( result == COMM_TXERROR )
{
}
else if( result == COMM_RXWAITING )
{
}
```

void setPosition(int Servoid, int Position, int Speed);//Created by Martin S. Mason(Professor @Mt. San Antonio College)

Sets Dynamixel position and velocity. This function sets velocity and position registers simultaneously.

Parameters

- **Servoid**

ID of Dynamixel.

- **Position**

Goal Position of Dynamixel

-**Speed**

Goal velocity of Dynamixel (1-1023 range).

Return Values

-none

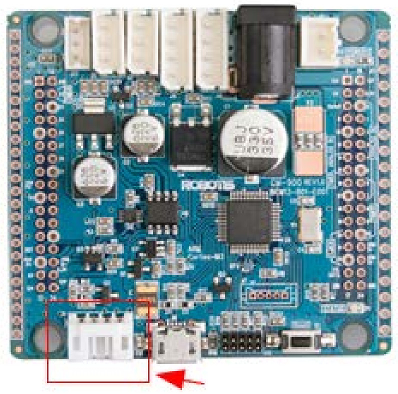
Example

```
result = Dxl.setPosition(3,500,600 );
if( Dxl.getResult( ) == COMM_RXSUCCESS )
{
    // Verify position command has been received
}
```

4.3 Zigbee API

ZigBee is a device allows remote control of the CM900 wirelessly. Simply connect to the CM-900's 4-pin connector and the CM-900. There's no need to implement instances; simply implement the functions listed below.

[BT-110A] or [BT-110A Set] [BT-210], [ZIG-110A Set] or [LN-101

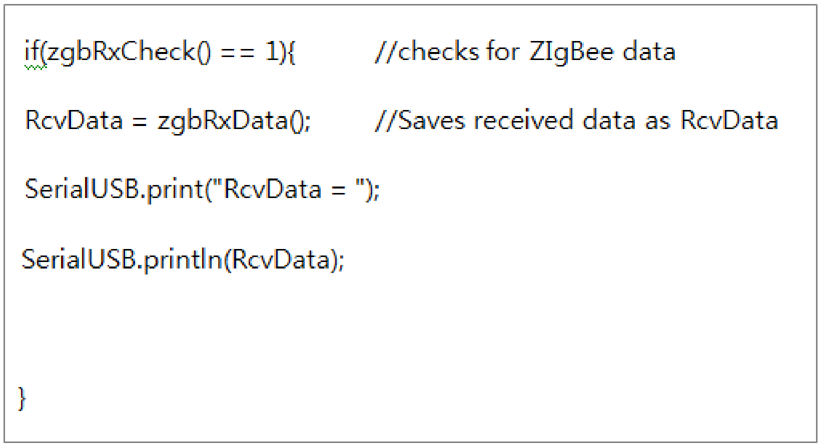


For detailed information about the 4-pin connector refer to the hardware portion of the CM-900

Device Control Methods	int zgbInitialize(int devIndex)	ZigBee device initialized at 57600bps. devIndex default value is 0.
	void zgbTerminate(void)	Halts the device.
Data Methods	int zgbTxData(int data)	Data transmission.
	int zgbRxCheck(void)	Vevifies received data
	int zgbRxData(void)	Returns value upon successful data reception

Example:

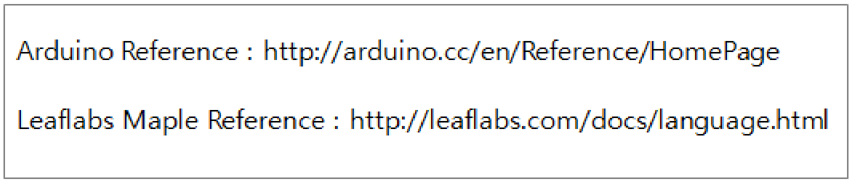
In loop() checks for ZigBee data reception.



```
if(zgbRxCheck() == 1){        //checks for ZIgBee data
RcvData = zgbRxData();        //Saves received data as RcvData
SerialUSB.print("RcvData = ");
SerialUSB.println(RcvData);


}
```

4.4 GPIO

The functions listed below are based on Arduino and Leaflabs. Arduino's Reference are useful because they are relatively easy to understand C/C++ code instead of ARM's codes. Use the links below for more information on Arduino and LeafLabs.



Arduino Reference : http://arduino.cc/en/Reference/HomePage

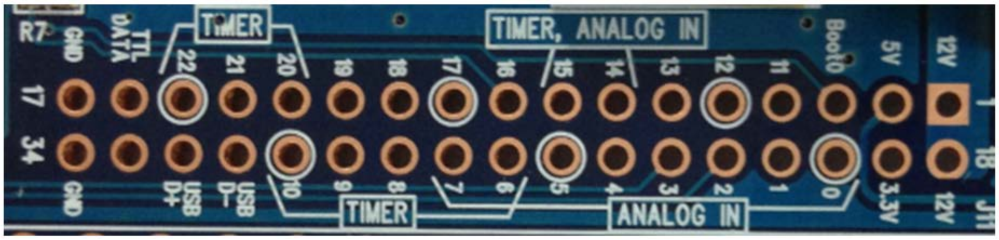
Leaflabs Maple Reference : http://leaflabs.com/docs/language.html

General Methods	pinMode(pin, WiringPinMode mode)	Sets pins for input and output
Digital Methods	int digitalRead(pin)	Reads status of a specific High/Low pin Said pin must be setup as input
	digitalWrite(pin, value)	Writes High/Low to a specific pin Said pin must be setup as output
	togglePin(pin)	Set pin to toggle i.e. switch from Low to High and from High to Low
Analog Methods	int analogRead(pin)	Read pin's analog value Said pin must be setup as analog input
	analogWrite(pin, duty cycle)	Writes analog data to pin pwmWrite() duty cycle (0~65536 range) Duty cycle can be controlled

For digital inputs use pins 0-31 of the CM-900.

Analog IN is engraved in the front portion of the CM-900 PCB. Implement analogWrite(), pwmWrite(), and TIMER on these specified pins.

Pins 6 through 10, 14,15, 20 through 22 are for TIMER and analogWrite(); pins 0 through 7, 14, 15 for TIMER and ANALOG IN. Pins 14 and 15 also available for analogRead().



void pinMode(pin, mode) (adopted from Leaf Labs Maple Documentation)
Changes pin to mode.

Parameters:	• pin - Pin of the CM-900 • mode - Mode listed below for said pin

Values:

- OUTPUT -
  Basic digital output: when the pin is HIGH, the voltage is held at +3.3v (Vcc) and when it is LOW, it is pulled down to ground.

- OUTPUT_OPEN_DRAIN -
  In open drain mode, the pin indicates “low” by accepting current flow to ground and “high” by providing increased impedance.

  An example use would be to connect a pin to a bus line (which is pulled up to a positive voltage by a separate supply through a large resistor). When the pin is high, not much current flows through to ground and the line stays at positive voltage; when the pin is low, the bus “drains” to ground with a small amount of current constantly flowing through the large resistor from the external supply. In this mode, no current is ever actually sourced from the pin.

- INPUT -
  Basic digital input.
  The pin voltage is sampled; when it is closer to 3.3v (Vcc) the pin status is high, and when it is closer to 0v (ground) it is low. If no external circuit is pulling the pin voltage to high or low, it will tend to randomly oscillate and be very sensitive to noise (e.g., a breath of air across the pin might cause the state to flip).

- INPUT_ANALOG -
  This is a special mode for when the pin will be used for analog (not digital) reads. Enables ADC conversion to be performed on the voltage at the pin.

- INPUT_PULLUP -
  The state of the pin in this mode is reported the same way as with INPUT, but the pin voltage is gently “pulled up” towards +3.3v.
  This means the state will be high unless an external device is specifically pulling the pin down to ground, in which case the “gentle” pull up will not affect the state of the input.

- INPUT_PULLDOWN -
  The state of the pin in this mode is reported the same way as with INPUT, but the pin voltage is gently “pulled down” towards 0v.
  This means the state will be low unless an external device is specifically pulling the pin up to 3.3v, in which case the “gentle” pull down will not affect the state of the input.

- INPUT_FLOATING -
  Synonym for INPUT.

- PWM -
  This is a special mode for when the pin will be used for PWM output (a special case of digital output).

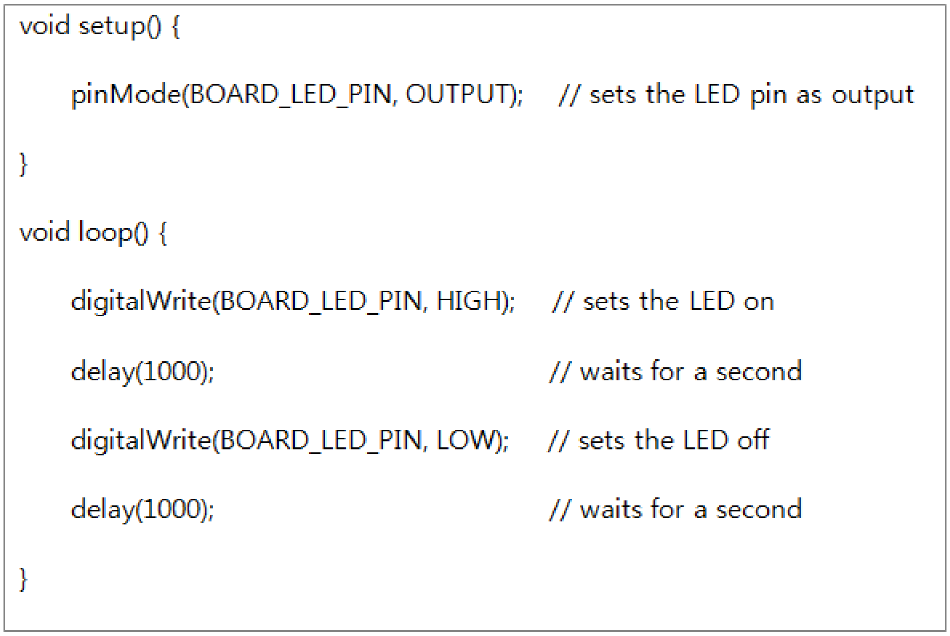
- PWM_OPEN_DRAIN -
  Like PWM, except that instead of alternating cycles of LOW and HIGH, the voltage on the pin consists of alternating cycles of LOW and floating (disconnected).

Discussion

pinMode() is a function in setup() where the pin can be set. The pin must be designated for writing.

Example

pinMode() sets LED to OUTPUT mode. Use the fuction digitalWrite() to write data (high/low). a blinking LED is the result.



```
void setup() {
      pinMode(BOARD_LED_PIN, OUTPUT);    // sets the LED pin as output
}
void loop() {
      digitalWrite(BOARD_LED_PIN, HIGH);    // sets the LED on
      delay(1000);                          // waits for a second
      digitalWrite(BOARD_LED_PIN, LOW);     // sets the LED off
      delay(1000);                          // waits for a second
}
```

uint32 digitalRead (uint8 pin) (adopted from Leaf Labs Maple Documentation)

Reads pin High/Low status. However, said pin must have INPUT_PULLUP or INPUT_PULLDOWN for input. Please refer to pinMode().

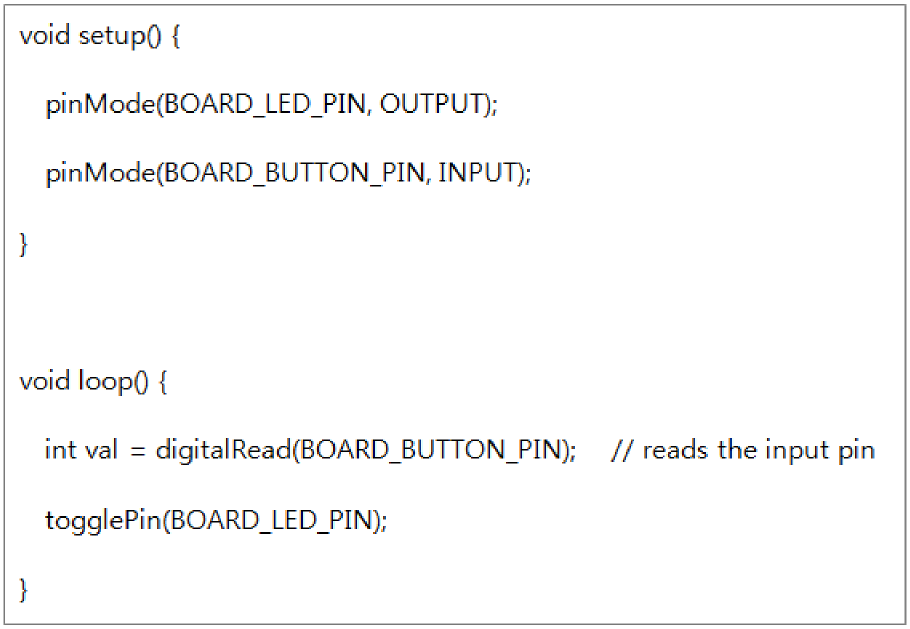
Parameters:	• pin - Declares read pin
Return:	LOW or HIGH.

• Discussion

If actual pin is not connected HIGH or LOW may be read ramdomly

• Example

The following example the LED turns on and off repeatedly with the press of the button



```
void setup() {
  pinMode(BOARD_LED_PIN, OUTPUT);
  pinMode(BOARD_BUTTON_PIN, INPUT);
}


void loop() {
  int val = digitalRead(BOARD_BUTTON_PIN);    // reads the input pin
  togglePin(BOARD_LED_PIN);
}
```

---

void digitalWrite(uint8 pin, uint8 value) (adopted from Leaf Labs Maple Documentation)

Pins outpurs High/Low
However, said pin must have OUTPUT_PULLUP or OUTPUT_PULLDOWN for output. Please refer to pinMode().

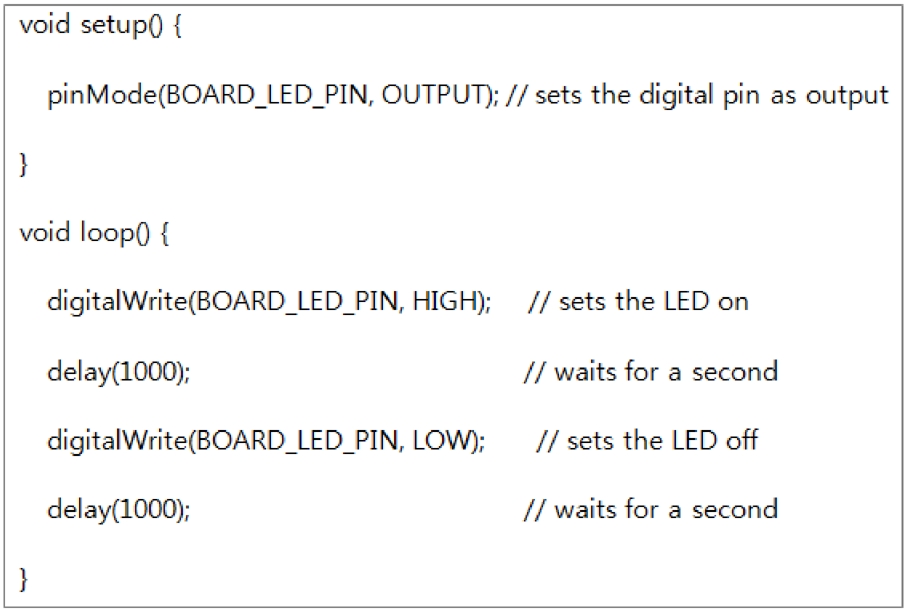
Parameters:	• pin - declares write pin • value - HIGH(1) or LOW (0)

- Discussion

The declared OUTPUT pin outputs 3.3V for HIGH and 0V for LOW.

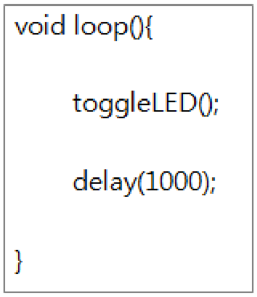
- Example

The following example is an implementation digitalWrite() function from LED Blink example.



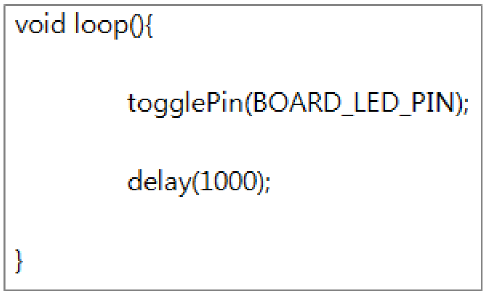
```
void setup() {

   pinMode(BOARD_LED_PIN, OUTPUT); // sets the digital pin as output

}

void loop() {

   digitalWrite(BOARD_LED_PIN, HIGH);    // sets the LED on

   delay(1000);                          // waits for a second

   digitalWrite(BOARD_LED_PIN, LOW);     // sets the LED off

   delay(1000);                          // waits for a second

}
```

The following allow replacement to toggleLED() inside loop(). This function is for the built-in LED.



```
void loop(){

      toggleLED();

      delay(1000);

}
```

Or replaced by togglePin().



```
void loop(){

            togglePin(BOARD_LED_PIN);

            delay(1000);

}
```

unit16 **analogRead**(uint8 pin) (adopted from Leaf Labs Maple Documentation)

Reads pin’s analog value.

This featureis blocked until ADC is converted. Pin mode must be set to INPUT_ ANALOG.

Parameters:	- pin - Analog pin read
Return:	Voltage converted to 12-bit integer (0-4095).

- Discussion

Reads analog value to declared pin. The CM-900 has 16 12-bitchannels. This converts input of 0V to 3.3V to 0 to 4095. There are other factors that affect accuracy and must be taken into account.

To call this fuction pinMode() function must be implementedand ANALOG_INPUT must be set. For more information please check pinMode()..

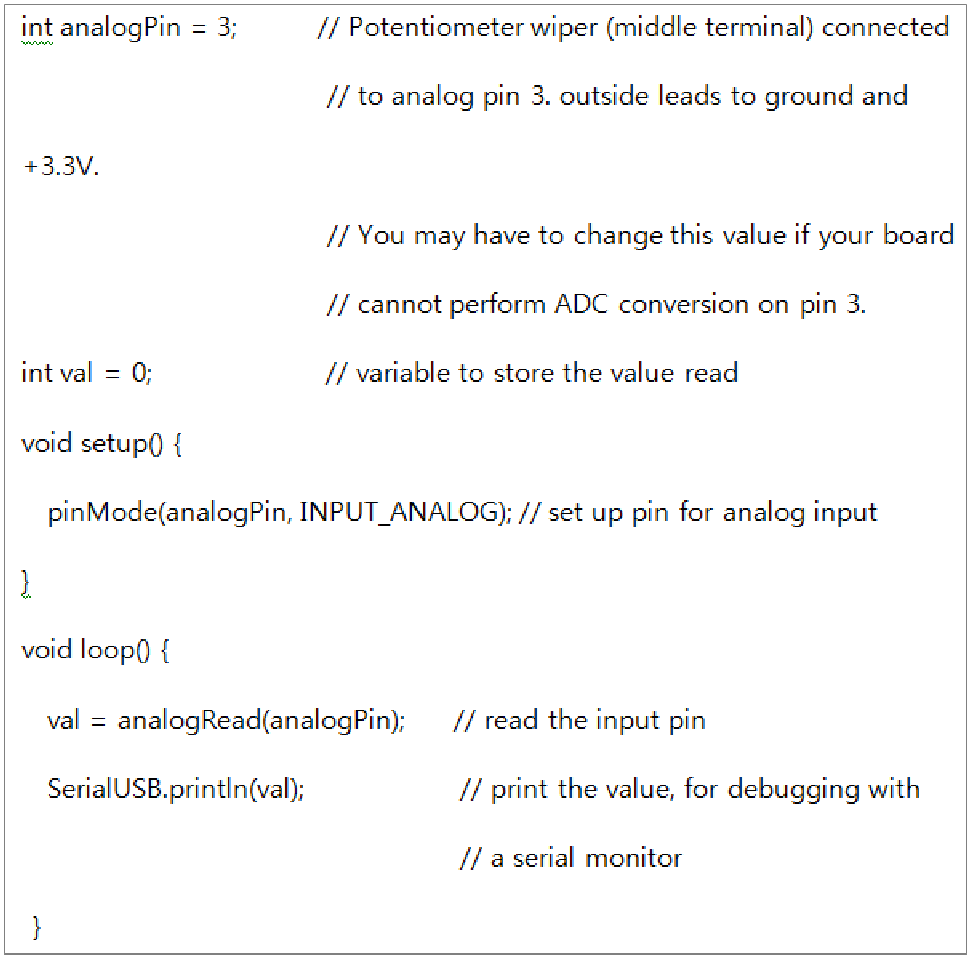
- Parameter Discussion

The number in this function is the analog pin number. These pin numbers are labeled in white on the CM-900's PCB silk screen along with ANALOG IN.

- Note

If a pin is not connected then its readout is not possible.

- Example



```
int analogPin = 3;         // Potentiometer wiper (middle terminal) connected
                            // to analog pin 3. outside leads to ground and
+3.3V.
                            // You may have to change this value if your board
                            // cannot perform ADC conversion on pin 3.
int val = 0;                // variable to store the value read
void setup() {
  pinMode(analogPin, INPUT_ANALOG); // set up pin for analog input
}
void loop() {
  val = analogRead(analogPin);     // read the input pin
  SerialUSB.println(val);           // print the value, for debugging with
                                    // a serial monitor
 }
```

analogWrite(uint8 pin, uint16 duty_cycle) (adopted from Leaf Labs Maple Documentation)

Given the declared duty cycle of the pin a PWM signal is possible. With CM-900 PWM signals can be controlled via duty cycle.

Duty cycle range between 0~65535.

Parameters:

- pin -

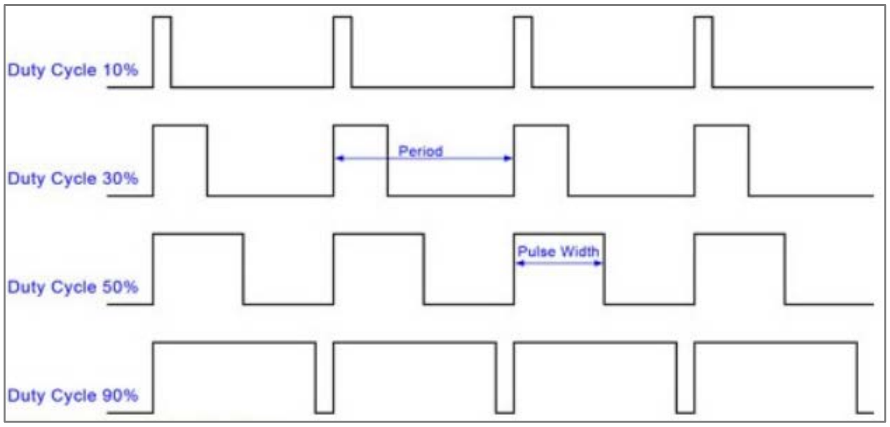
PWM input pin

- duty_cycle -

PWM signal duty cycle (0~65535)

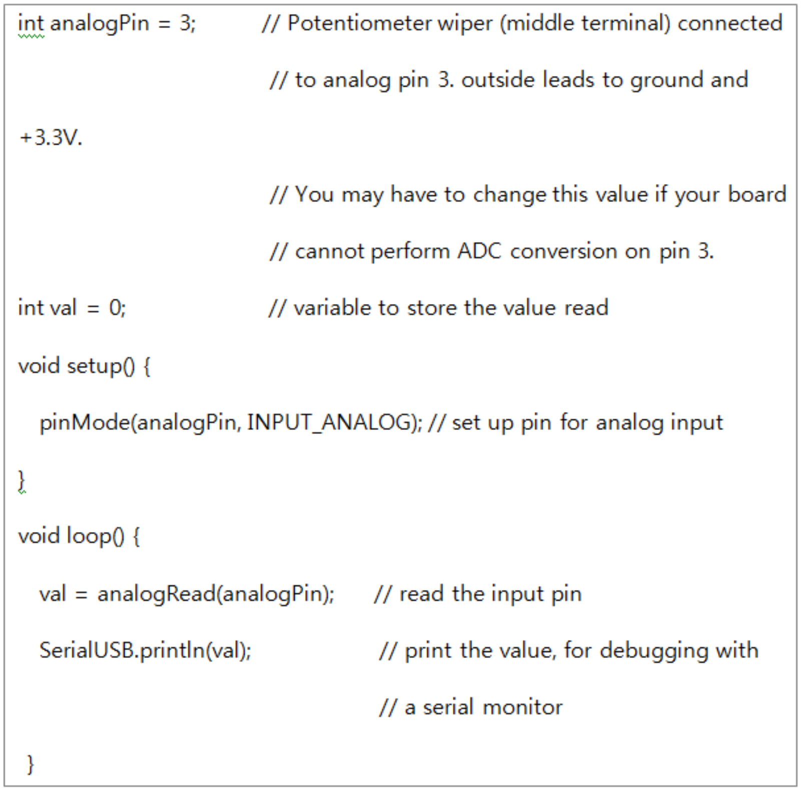
Duty cycle can be controlled in the duty cycle input part of the function.

Values close to 0 the duty cycle is small (small HIGH area). Refer to the diagram below larger duty cycles (larger HIGH areas)



Example

The following is an example read from the potentiometer to control the brightness of the LED.



```
int analogPin = 3;        // Potentiometer wiper (middle terminal) connected
                           // to analog pin 3. outside leads to ground and
+3.3V.
                           // You may have to change this value if your board
                           // cannot perform ADC conversion on pin 3.
int val = 0;               // variable to store the value read
void setup() {
   pinMode(analogPin, INPUT_ANALOG); // set up pin for analog input
}
void loop() {
   val = analogRead(analogPin);     // read the input pin
   SerialUSB.println(val);           // print the value, for debugging with
                                     // a serial monitor
 }
```

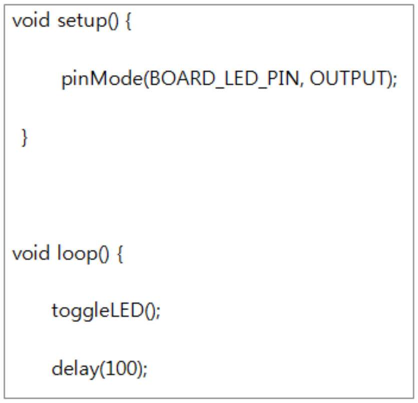
void toggleLED**()** (adopted from Leaf Labs Maple Documentation)

Toggles the CM-900 STATUS LED.

When the LED in on status is called it turns off; when called in off it turns on.

- Example

The following is the Blink example for STATUS LED.



```
void setup() {

      pinMode(BOARD_LED_PIN, OUTPUT);

 }


void loop() {

     toggleLED();

     delay(100);
```

4.5 Interrupt

Interrupt is a feature that allows specific actions to be performed based status return. An interrupt verification code is separately required because it utilizes hardware timer. Despite external devices having its own interrupt, it cannot exceed 16. For example pins 0 through 15 each with its own interrupt event there cannot be any more interrupts in code.

The following functions allow control of interrupts

attachInterrupt(*pin*, voidFuncPtr *handler*, *mode*)	Adds interrupt handler for a specific pin
detachInterrupt(*pin*)	Removes interrupt handler for a specific pin
noInterrupts()	All interrupts disabled
Interrupts()	All interrupts enabled
disableDebugPorts()	JTAG/SWD disable option
enableDebugPorts()	JTAG/SWD enable option

attachInterrupt(uint8 *pin*, voidFuncPtr *handler*, ExtIntTriggerMode *mode*)

Adapted from Maple Documentation

Parameters

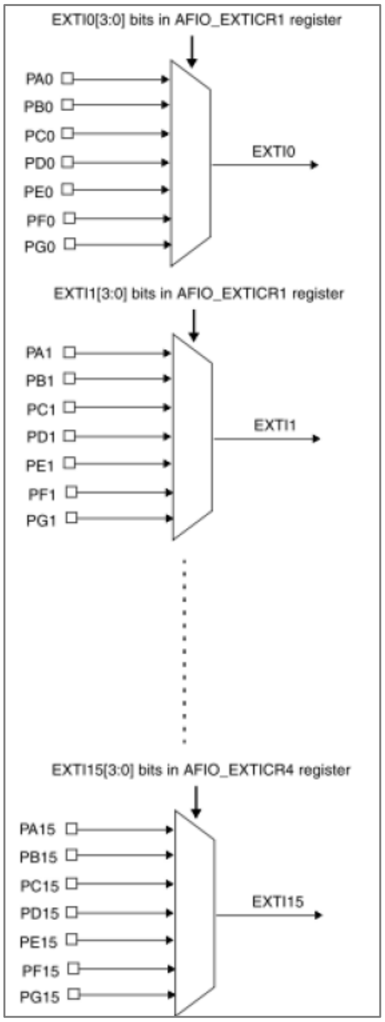
- pin – pin #
- handler – interrupt event pointer
- mode – interrupt form; falling edge or rising edge

mode

interrupt event type

Values:

- RISING -
  Trigger when LOW goes to HIGH
- FALLING -
  Trigger when HIGH goes to LOW
- CHANGE -
  When pins changes HIGH to LOW or LOW to HIGH; event trigger regardles

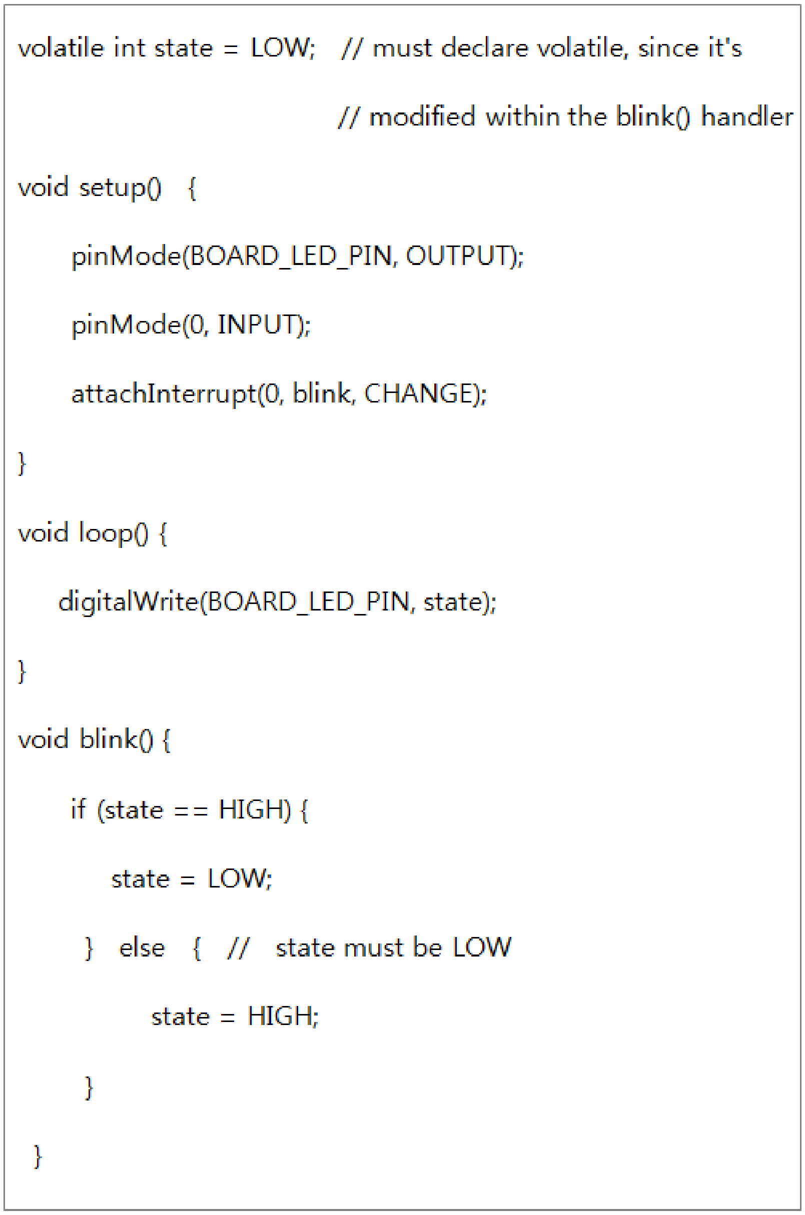

EXTI0[3:0] bits in AFIO_EXTICR1 register
PA0
PB0
PC0
PD0
PE0
PF0
PG0
EXTI0
EXTI1[3:0] bits in AFIO_EXTICR1 register
PA1
PB1
PC1
PD1
PE1
PF1
PG1
EXTI1
EXTI15[3:0] bits in AFIO_EXTICR4 register
PA15
PB15
PC15
PD15
PE15
PF15
PG15
EXTI15

- Discussion

The delay() function cannot be used because interrupt is processed internally. Also values changed to millis () do not increase. Serial data reception process may be lost. To prevent data loss Volatile should be declared globally.

- Example

In this example pin 0 the LED turns on or off when there is a signal change.

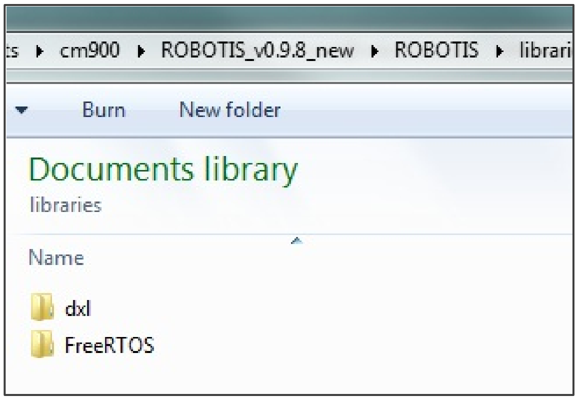


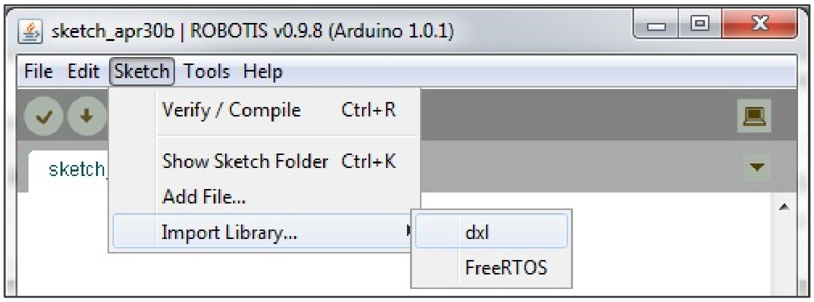
```
volatile int state = LOW;   // must declare volatile, since it's
                            // modified within the blink() handler
void setup()  {
    pinMode(BOARD_LED_PIN, OUTPUT);
    pinMode(0, INPUT);
    attachInterrupt(0, blink, CHANGE);
}
void loop() {
   digitalWrite(BOARD_LED_PIN, state);
}
void blink() {
    if (state == HIGH) {
       state = LOW;
     }  else  {  //  state must be LOW
          state = HIGH;
     }
 }
```

The function blink() is an interrupt handler. When pin 0 inout signal changes blink() gets called to high/low. In turn loop() follows the state (value) and sets the LED on/off.

4.6 User-created API library

When executing ROBOTIS CM-9 libraries from the folder \ROBOTIS\library get carried. This also includes user’s libraries.

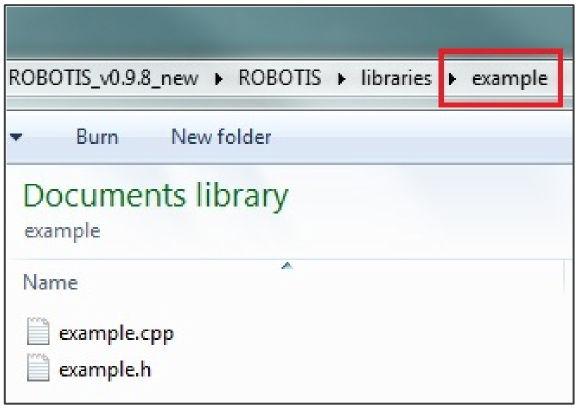




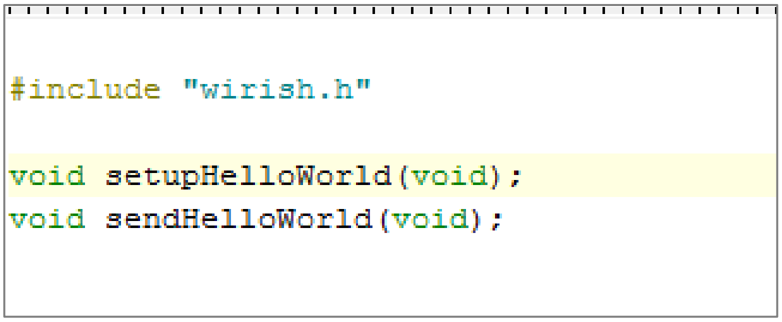
4.6.1 CPP-based library creation

Create file(s) in CPP format; API's in core library.

The following shows user-created example.h header file and example.cpp

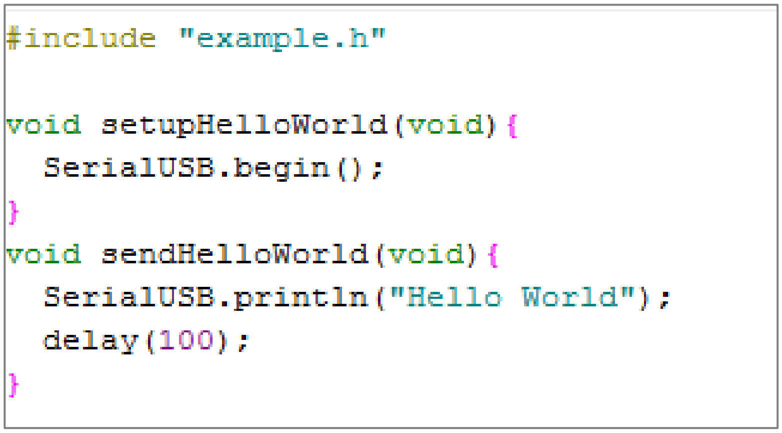


In example.h wirish.h has been declared. This makes APIs from digitalWrite() to Dynamixel API available from the CM-9 core library.



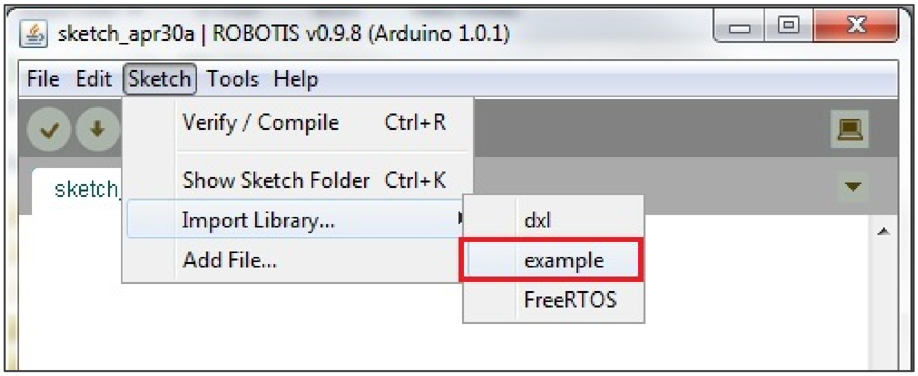
```
#include "wirish.h"

void setupHelloWorld(void);
void sendHelloWorld(void);
```

In example.cpp implement setupHelloWorld() and sendHelloWorld().

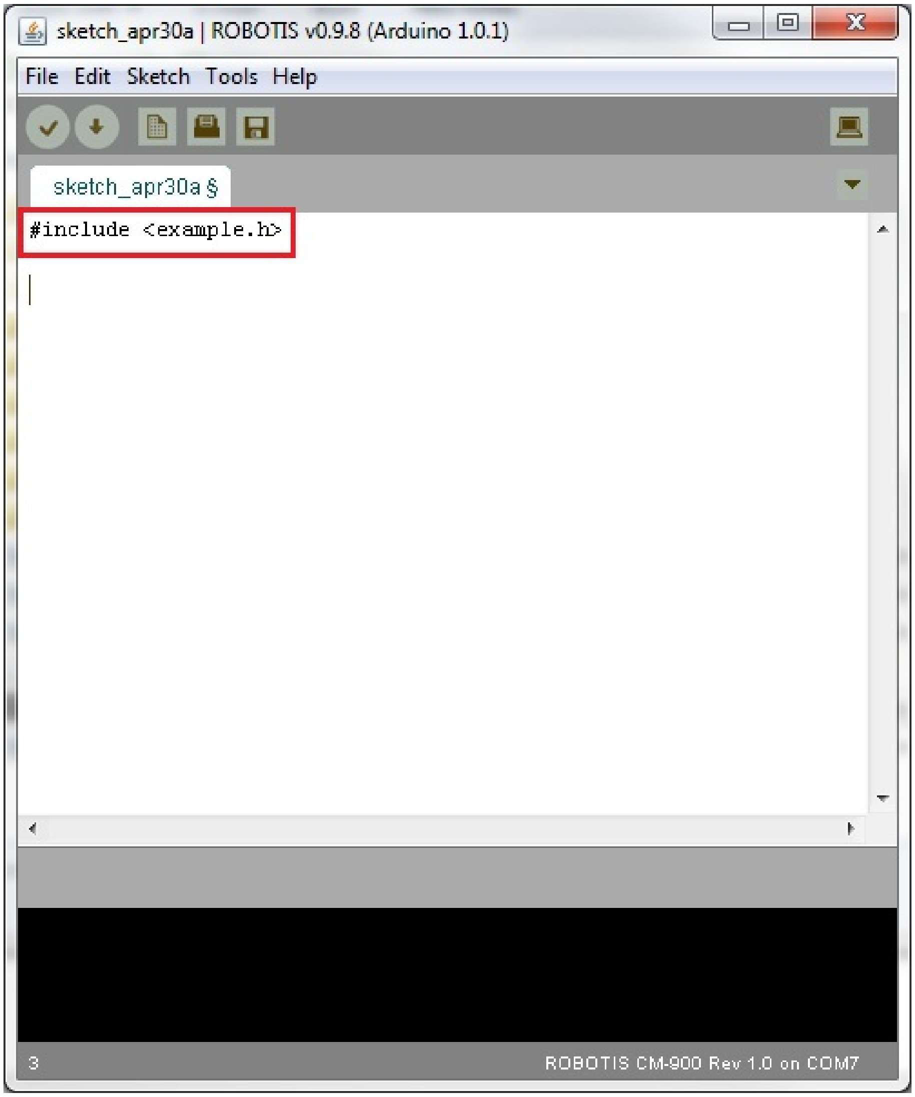


```
#include "example.h"

void setupHelloWorld(void){
  SerialUSB.begin();
}
void sendHelloWorld(void){
  SerialUSB.println("Hello World");
  delay(100);
}
```

The project can be imported as shown below.


sketch_apr30a | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
sketch
Verify / Compile Ctrl+R
Show Sketch Folder Ctrl+K
Import Library...
Add File...
dxl
example
FreeRTOS

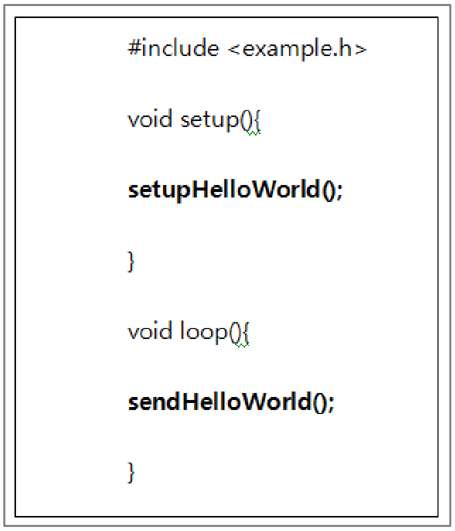
When adding #include <example.h> is automatically included.



sketch_apr30a | ROBOTIS v0.9.8 (Arduino 1.0.1)
File Edit Sketch Tools Help
sketch_apr30a §
3
ROBOTIS CM-900 Rev 1.0 on COM7

```
#include <example.h>
```

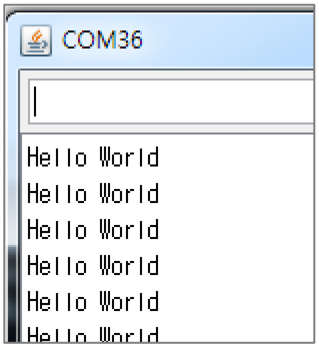
Now both setupHelloWorld() and sendHelloWorld() can be written. Note that in the #include preprocessor the<and>characters refer to ROBOTIS\libraries directory.



```
#include <example.h>

void setup(){

setupHelloWorld();

}

void loop(){

sendHelloWorld();

}
```

<sketch code>

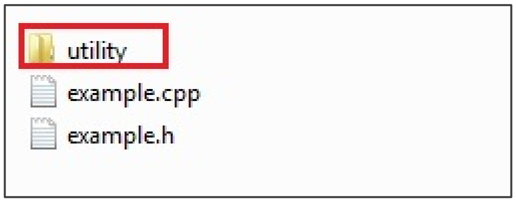
Once compiled and downloaded to the CM-900 the results can be seen on serial monitor



```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

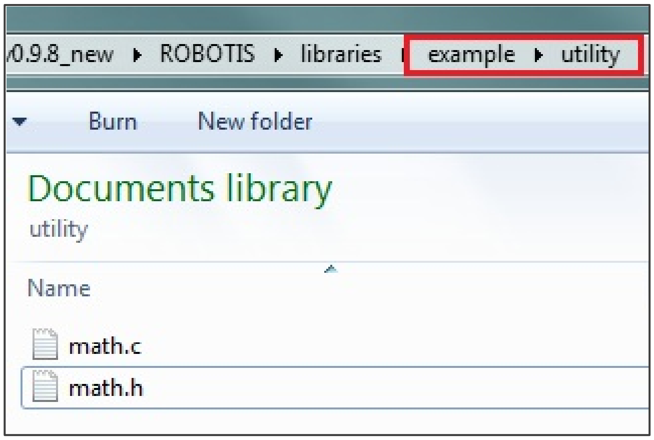
4.6.2 C-based library creation

Most firmware codes are written in C. C is also available for coding with the CM-9. The following shows math.c, math.h files under utility folder under example library.

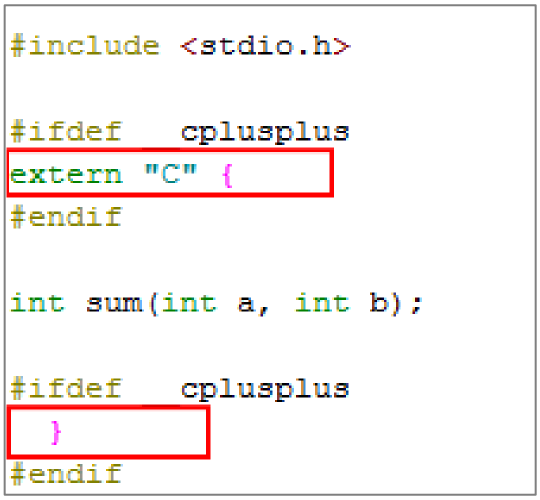


utility
example.cpp
example.h

Both C-based files under utility directory.



The following show the contents of math.c and math.h files.

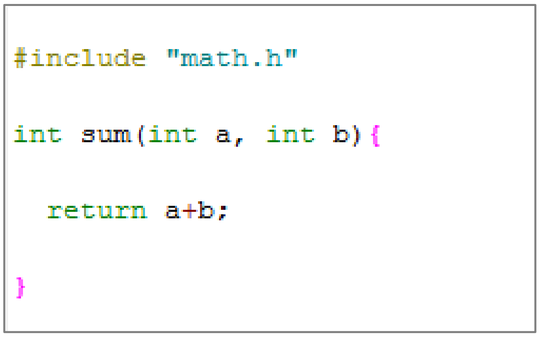


```
#include <stdio.h>

#ifdef   cplusplus
extern "C" {
#endif

int sum(int a, int b);

#ifdef   cplusplus
  }
#endif
```

<contents of math.h>

Let’s look at sum() in ROBOTIS CM-9’s tool chain. Both < > imply looking for a directory called include.

Extern “C” {} required C++-based compilers.

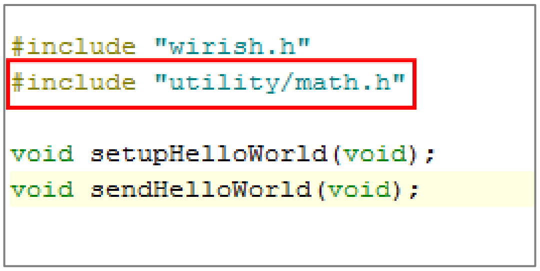


```
#include "math.h"

int sum(int a, int b){

  return a+b;

}
```

<contents of math.c>

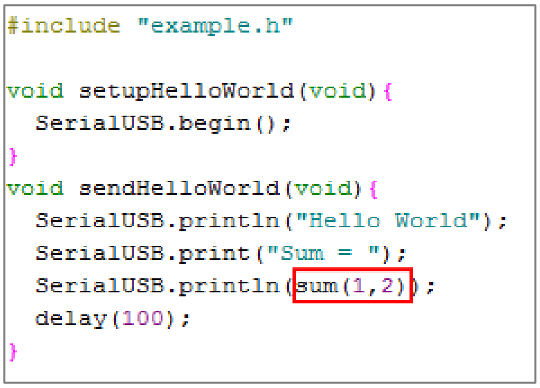
The following show modified example.h and example.cpp.

Declare #include “utility/math.h”



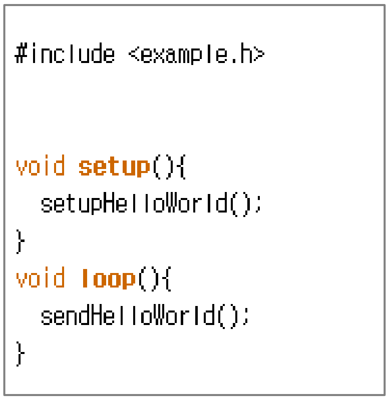
```
#include "wirish.h"
#include "utility/math.h"

void setupHelloWorld(void);
void sendHelloWorld(void);
```

Add sum() function to example.cpp

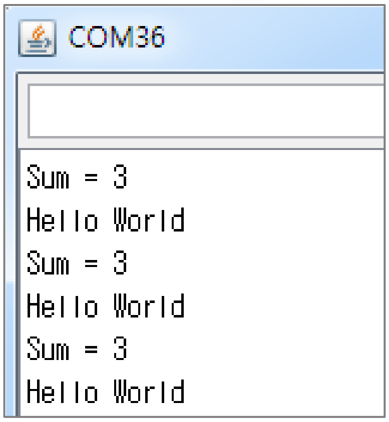


```
#include "example.h"

void setupHelloWorld(void){
  SerialUSB.begin();
}
void sendHelloWorld(void){
  SerialUSB.println("Hello World");
  SerialUSB.print("Sum = ");
  SerialUSB.println(sum(1,2));
  delay(100);
}
```

sendHelloWorld() sketch code outputs sum(1,2) value.



```
#include <example.h>


void setup(){
  setupHelloWorld();
}
void loop(){
  sendHelloWorld();
}
```

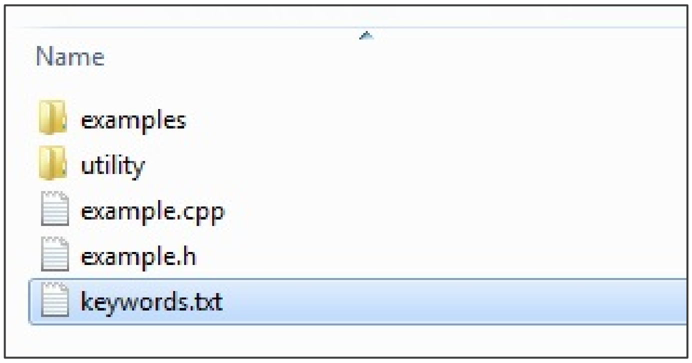


```
Sum = 3
Hello World
Sum = 3
Hello World
Sum = 3
Hello World
```

4.6.3 Syntax library highlights

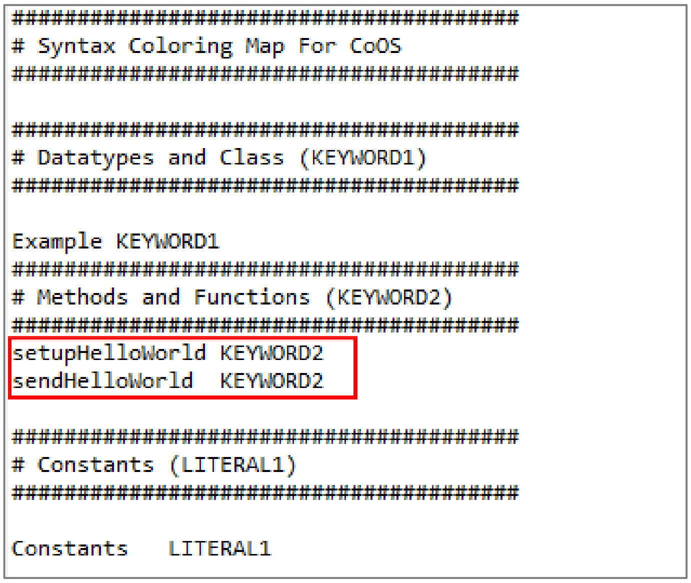
Add keywords.txt file for syntax.

And place it in the same example directory.



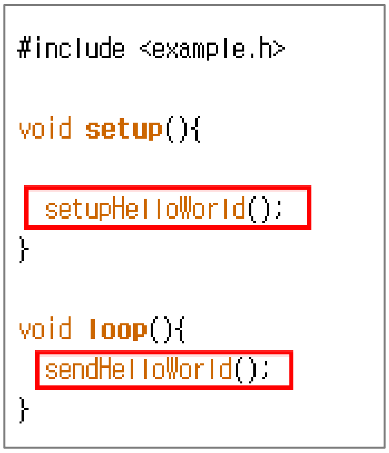
examples
utility
example.cpp
example.h
keywords.txt

Add the contents of the file as shown below.



```
#######################################
# Syntax Coloring Map For CoOS
#######################################

#######################################
# Datatypes and Class (KEYWORD1)
#######################################

Example KEYWORD1
#######################################
# Methods and Functions (KEYWORD2)
#######################################
setupHelloWorld KEYWORD2
sendHelloWorld  KEYWORD2

#######################################
# Constants (LITERAL1)
#######################################

Constants   LITERAL1
```

In ROBOTIS CM-9 look for a color change when inputting setupHelloWorld() and sendHelloWorld().

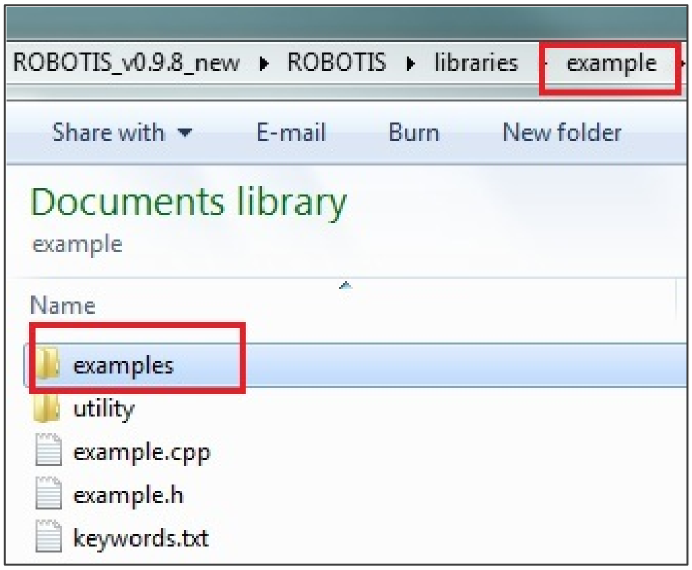


```
#include <example.h>

void setup(){

 setupHelloWorld();
}

void loop(){
 sendHelloWorld();
}
```

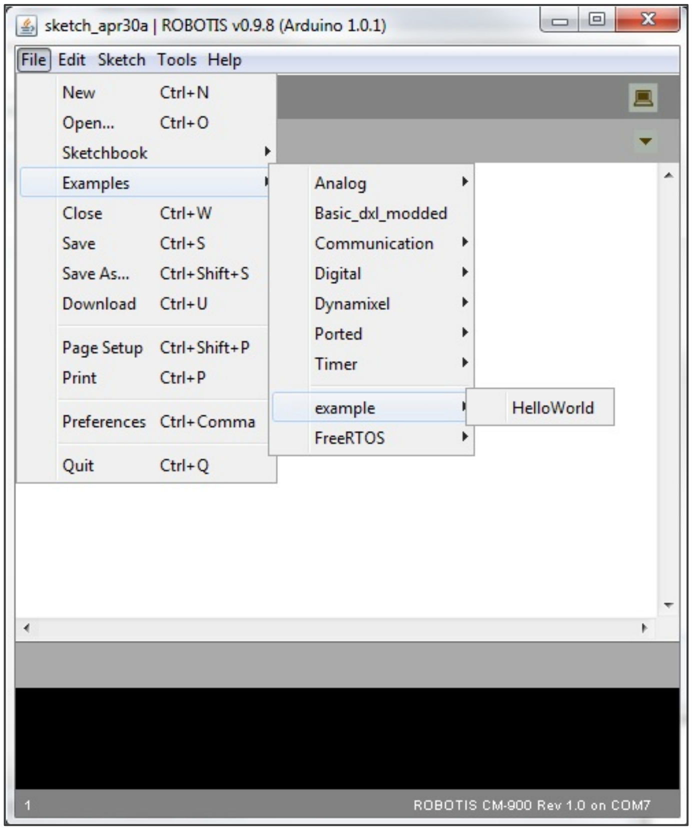
<syntax feature>

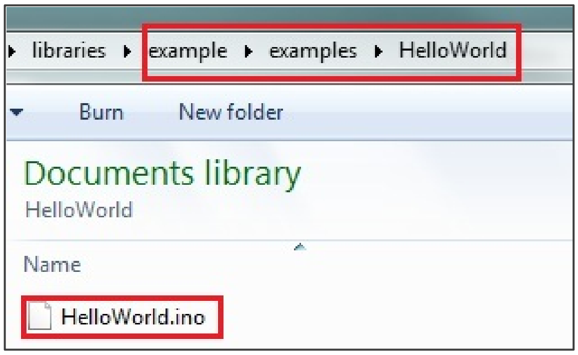
4.6.4 Registering a library

You can register your own library so it can be readily available in the CM-9's menu.



Create a directory named "example" then simply place your sketch code inside the "example" directory. Restart ROBOTIS CM-9 and your custom-created library will shop up in the nemu.





The name in the menu will reflect the name of the sketch code file.

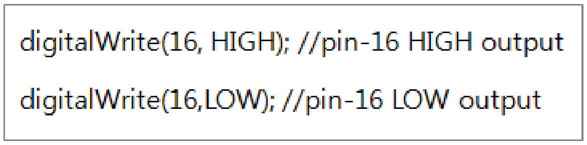
5 Learning (implementing APIs to CM-900)

5.1 Digital I/O

5.1.1 Digital output on pin-16

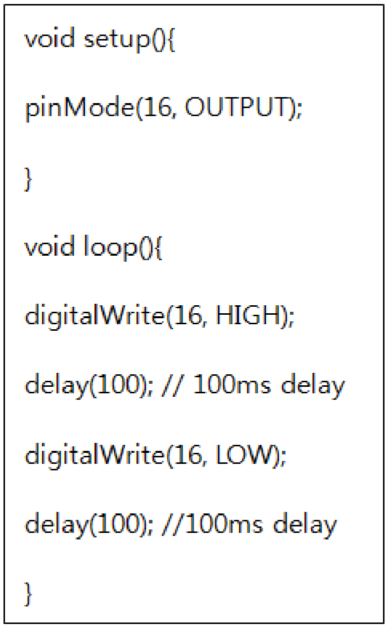
Set pinMode(16, OUTPUT) in setup(); this sets pin-16 to OUTPUT

Declare digitalWrite() to HIGH/LOW.



```
digitalWrite(16, HIGH); //pin-16 HIGH output
digitalWrite(16,LOW); //pin-16 LOW output
```

Pin-16 reads STATUS LED; when HIGH the LED is off; when LOW the LED is on.



```
void setup(){
pinMode(16, OUTPUT);
}
void loop(){
digitalWrite(16, HIGH);
delay(100); // 100ms delay
digitalWrite(16, LOW);
delay(100); //100ms delay
}
```

Blinks in 0.1sec intervals.

5.1.2 Digital input on pin-1

Set pinMode(1, INPUT) in setup(); this sets pin-1 to INPUT.

If external pull-up needed set pinMode(1, INPUT_PULLUP); for pull-down set pinMode(1, INPUT_PULLDOWN).

digitalRead() gets HIGH/LOW value. If pin is not connected then value could be random.

```
int value = digitalRead(1); // read #1, value assigned
```

verify the code.

```
void setup(){
pinMode(1, INPUT);
SerialUSB.begin();
}
void loop(){
        int value = digitalRead(1);
        if ( value == HIGH)
                        SerialUSB.println("HIGH Detected!");
        else
                        SerialUSB.println("LOW Detected!");
        delay(100);
}
```

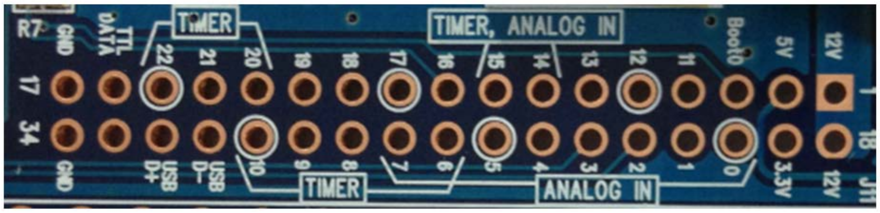
5.1.3 Toggle pin-1

Switch pin-1 from high-to-low then low-to-high.

```
digitalWrite(1, HIGH); //set pin-1 to HIGH
togglePin(1); // switches pin-1 from HIGH to LOW
```

5.2 Analog I/O

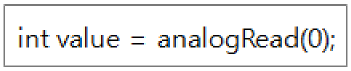
Analog input pins are labeled ANALOG IN on the CM-900's silk screen. Pins 0 through 7,14, and 15 are input pins.



Analog output requires PWM, which is used by TIMER, for analog output.

5.2.1 Analog input on pin-0

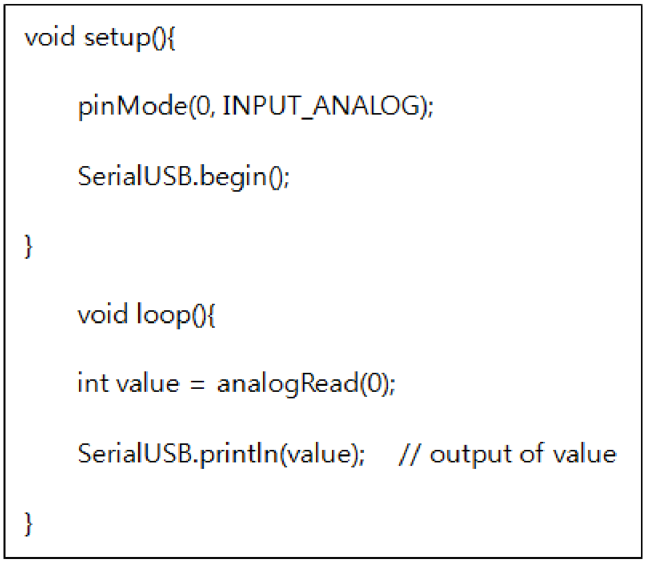
Set pinMode(0, INPUT_ANALOG) in setup(); this sets pin-0 to INPUT_ANALOG.



```
int value = analogRead(0);
```

// pin-0gets analog input, value assigned.

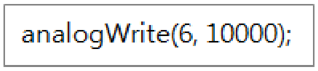
The assigned value gets converted in a 12-bit ADC value (0~ 4095).



```
void setup(){

    pinMode(0, INPUT_ANALOG);

    SerialUSB.begin();

}

    void loop(){

    int value = analogRead(0);

    SerialUSB.println(value);    // output of value

}
```

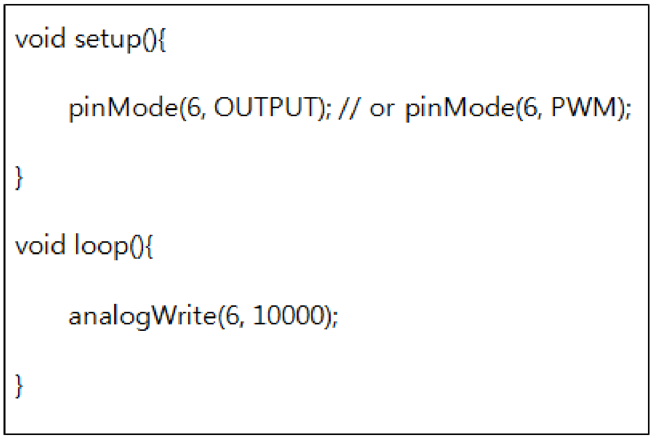
5.2.2 Analog output (PWM) on pin-6

Set pin-6 to pinMode(6, OUTPUT) or pinMode(6, PWM).



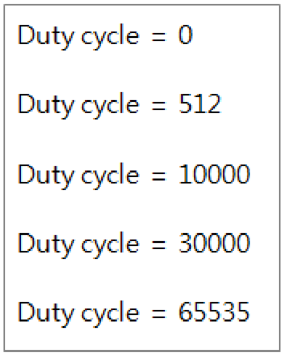
```
analogWrite(6, 10000);
```

Analog output as PWM. PWM's duty cycle is set on the second value (10000). Range is 0~ 65535.



```
void setup(){
        pinMode(6, OUTPUT); // or pinMode(6, PWM);
}
void loop(){
        analogWrite(6, 10000);
}
```

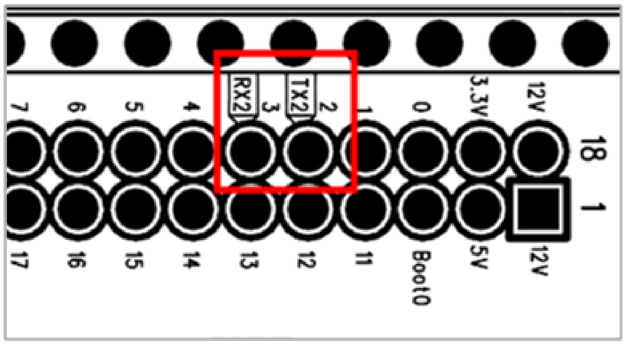
In analogWrite() the second value is PWM's implementation as duty cycle.



```
Duty cycle = 0
Duty cycle = 512
Duty cycle = 10000
Duty cycle = 30000
Duty cycle = 65535
```

5.3 Serial comm

The CM-900 has a total of 3 serial devices. These are USART serial1, serial 2, and serial3. Serial1 is assigned to Dynamixel comm port. Serial2 for 4-pin BT-210, BT-110 devices. To see the serial pins see reverse side of CM-900. Serial1 has TX1 and RX1. Serial2 has TX2 and RX2. Serial3 has TX3 and RX3.



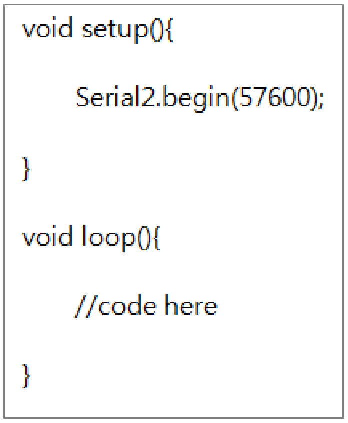
<Serial2>

Serial USB device download is USB communications.

Serial USB devices are controlled via SerialUSB method.

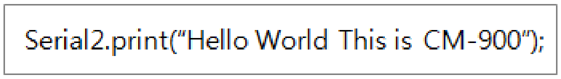
5.3.1 Transmit data via serial device

Initialize device in and run in loop().



```
void setup(){

      Serial2.begin(57600);

}

void loop(){

      //code here

}
```

Data transmission can be outputted with print() and println(). print() has no line brakes while printIn() does.

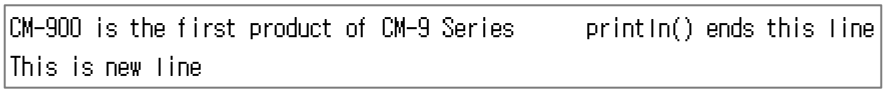


```
Serial2.print("Hello World This is CM-900");
```

"Hello World" is outputted via Serial2(TX2, RX2) device.

```
Serial2.print("CM-900 is the first product of CM-9 Series");

Serial2.println("println() ends this line");

Seirla2.println("This is new line");
```

println() outputs in a new line.



```
CM-900 is the first product of CM-9 Series      println() ends this line
This is new line
```

```
Serial2.print(12);
```

Outputs 12 in decimal (default)

```
int abc = 128;

Seial2.print(abc);
```

Outputs abc as 128

```
Serial2.print(abc, 16);
```

Outputs abc in hexadecimal (0x80)

```
Serial2.print(abc, 2);
```

Outputs abc in binary

```
Serial2.println(3.14);
```

Outputs a double data type and ends line; outputs 2 significant places

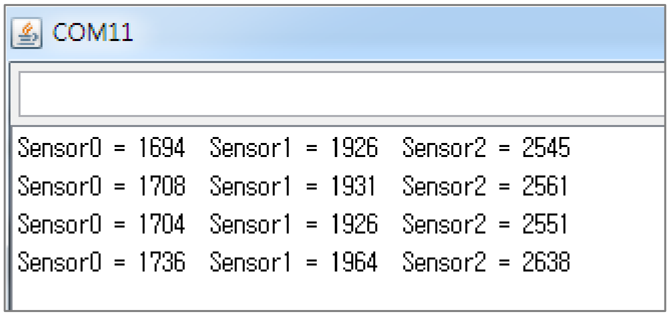
Can output declared double variables.

```
double  var = 1.234;

Serial2.println(var);
```

Input analog values to pin-0, pin-1, pin-2; in turn output via Serial2.

```
int sensorValue0=0;

int sensorValue1=0;

int sensorValue2=0;

sensorValue0 = analogRead(0);

sensorValue1 = analogRead(1);

sensorValue2 = analogRead(2);

Serial2.print(“Sensor0 = “); Serial2.print(sensorValue0);

Serial2.print(“ Sensor1 = “); Serial2.print(sensorValue1);

Serial2.print(“ Sensor2 = “); Serial2.println(sensorValue2);
```

sensorValue2 outputs all 3 pins one line at a time with println().

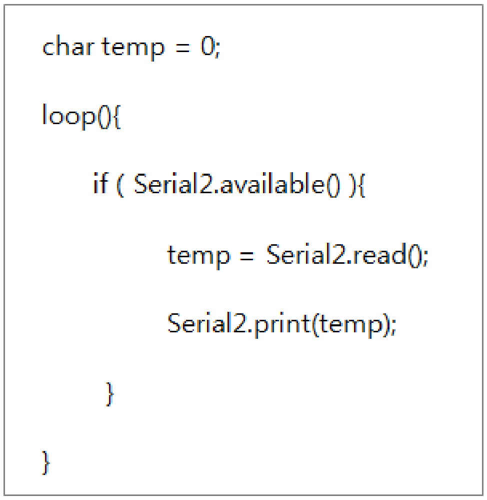


```
Sensor0 = 1694  Sensor1 = 1926  Sensor2 = 2545
Sensor0 = 1708  Sensor1 = 1931  Sensor2 = 2561
Sensor0 = 1704  Sensor1 = 1926  Sensor2 = 2551
Sensor0 = 1736  Sensor1 = 1964  Sensor2 = 2638
```

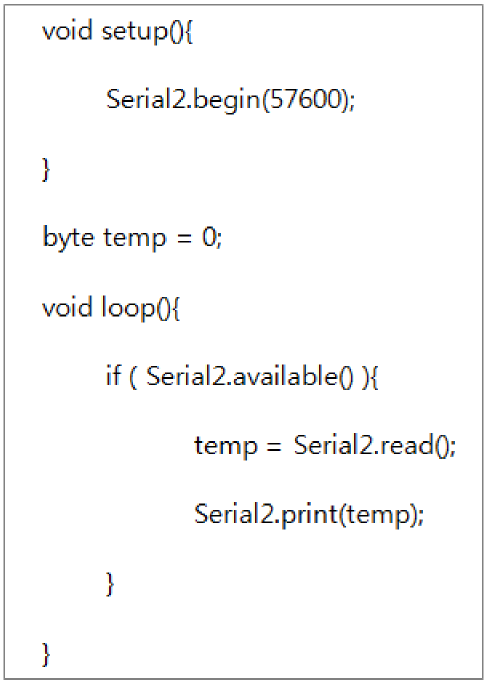
5.3.2 Receive data from serial device

Echo feature can be implemented with serial devices.

Assign temp as char type and save data from Serial2 with read(); use print() to output data for echo purposes.



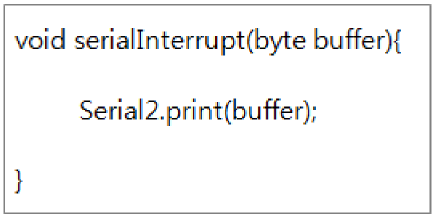
```
char temp = 0;

loop(){

      if ( Serial2.available() ){

               temp = Serial2.read();

               Serial2.print(temp);

        }

}
```



```
void setup(){

        Serial2.begin(57600);

}

byte temp = 0;

void loop(){

        if ( Serial2.available() ){

                    temp = Serial2.read();

                    Serial2.print(temp);

        }

}
```

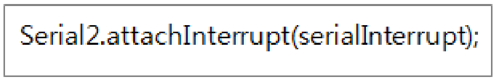
Interrupt implementation

Interrupts from serial devices do not return values. Incoming data can be echoed with print(). This can be implemented without declaring separate prototypes.



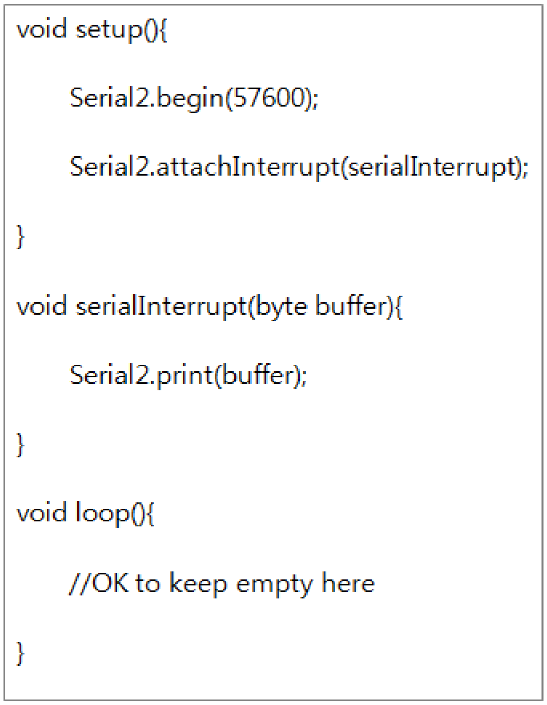
```
void serialInterrupt(byte buffer){
        Serial2.print(buffer);
}
```

serialInterrupt() can be implemented as a pointer in setup()



```
Serial2.attachInterrupt(serialInterrupt);
```

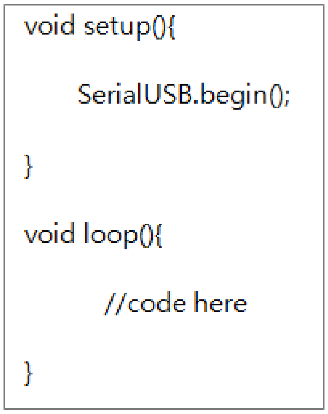
Let’s see code with serial2 device.



```
void setup(){
        Serial2.begin(57600);
        Serial2.attachInterrupt(serialInterrupt);
}
void serialInterrupt(byte buffer){
        Serial2.print(buffer);
}
void loop(){
        //OK to keep empty here
}
```

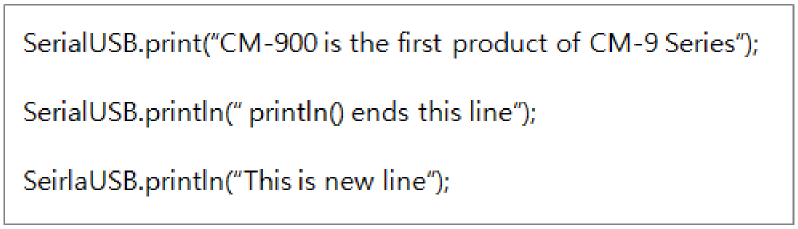
5.3.3 Output data with serial USB device

initialize SerialUSB device in setup(); run code in loop(). There is no need to declare baud rate value.



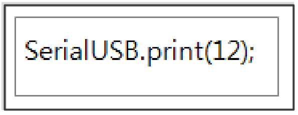
```
void setup(){
      SerialUSB.begin();
}
void loop(){
         //code here
}
```

Use print() and println() for control.



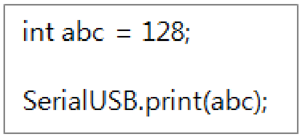
```
SerialUSB.print("CM-900 is the first product of CM-9 Series");
SerialUSB.println(" println() ends this line");
SeirlaUSB.println("This is new line");
```

Output 12 in decimal (default).



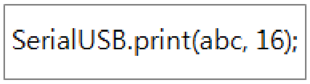
```
SerialUSB.print(12);
```

Output declared int type



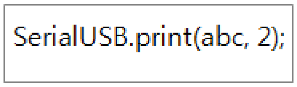
```
int abc = 128;
SerialUSB.print(abc);
```

Output abc in hexadecimal



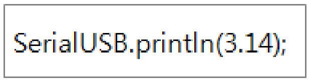
```
SerialUSB.print(abc, 16);
```

abc’s 128 is outputted to hexadecimal (0x80)



```
SerialUSB.print(abc, 2);
```

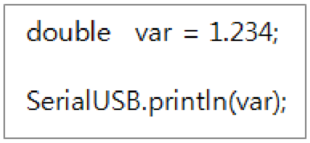
abc outputted in binary



```
SerialUSB.println(3.14);
```

Output of double type; output is 3.14

Declared double type



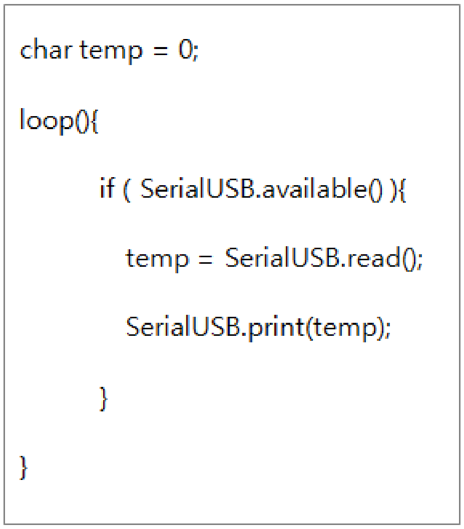
```
double   var = 1.234;

SerialUSB.println(var);
```

Outputs var as is (with 3 significant figures)

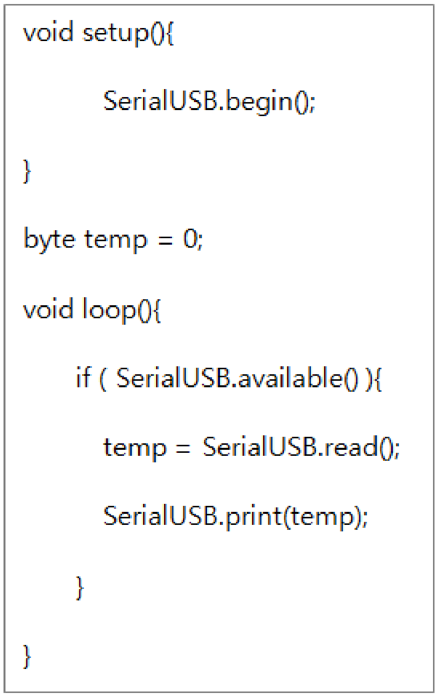
5.3.4 Receive data with serial USB device

Implement echo to serial USB device

Assign temp as char type and save data from serial USB device with read(); use print() to output data for echo purposes.



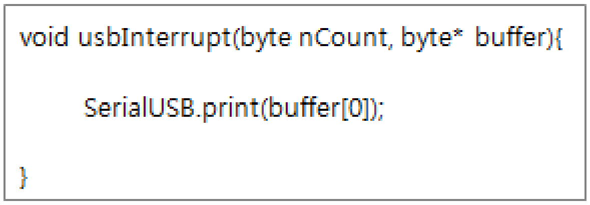
```
char temp = 0;

loop(){

          if ( SerialUSB.available() ){

             temp = SerialUSB.read();

             SerialUSB.print(temp);

          }

}
```



```
void setup(){
        SerialUSB.begin();
}
byte temp = 0;
void loop(){
      if ( SerialUSB.available() ){
         temp = SerialUSB.read();
         SerialUSB.print(temp);
      }
}
```

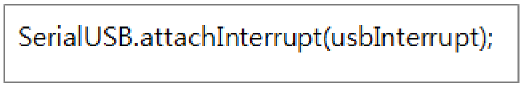
Interrupt implementation

Interrupts from serial USB do not return values byte and *byte types are implemented. Incoming data can be echoed with print(). When data is written to the USB COM port is done 1byte chunks (nCount). Only index 0 of transmitted byte is necessary for echoing.



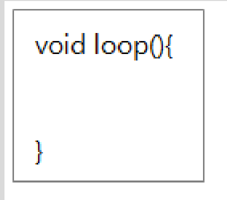
```
void usbInterrupt(byte nCount, byte* buffer){
       SerialUSB.print(buffer[0]);
}
```

Implement usbInterrupt()pointer on setup() through attachInterrupt().



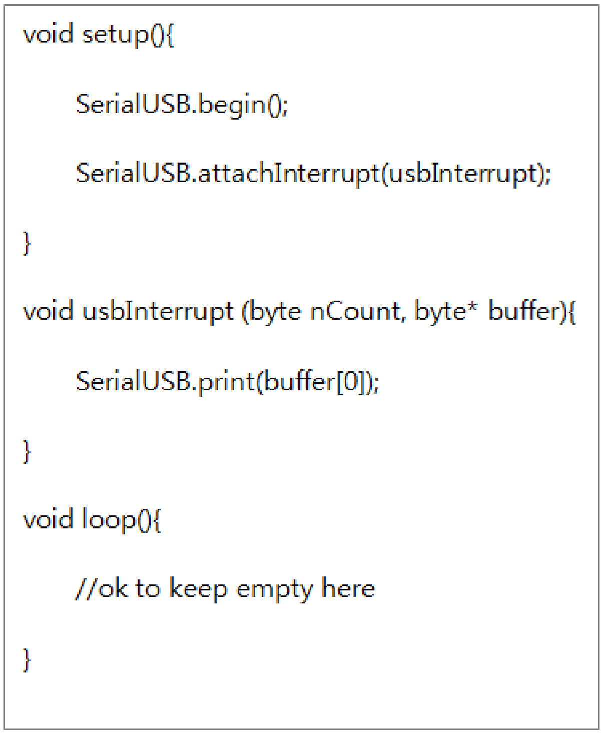
```
SerialUSB.attachInterrupt(usbInterrupt);
```

Its ok to keep loop() empty.



```
void loop(){


}
```

Let’s have a look at SerialUSB device’s interrupt code.



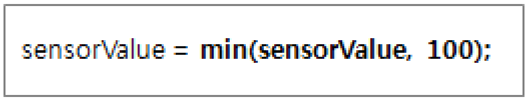
```
void setup(){

       SerialUSB.begin();

       SerialUSB.attachInterrupt(usbInterrupt);

}

void usbInterrupt (byte nCount, byte* buffer){

       SerialUSB.print(buffer[0]);

}

void loop(){

       //ok to keep empty here

}
```

5.4 Math functions

Trigonometric functions can be implemented to ROBOTIS CM-9 without any additional header files.

5.4.1 Basic math functions

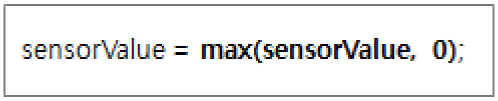
Get analog input and receive a value less than 100.



```
sensorValue = min(sensorValue, 100);
```

min(a,b) only returns values lower than 100. Anything greater than 100 sensorValue does not get assigned.

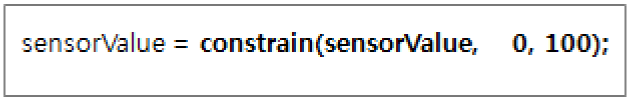
Oppositely the following return values greater than 0.



```
sensorValue = max(sensorValue,  0);
```

max(a,b) only returns values greater than 0. Anything lesser than 0 there is no return.

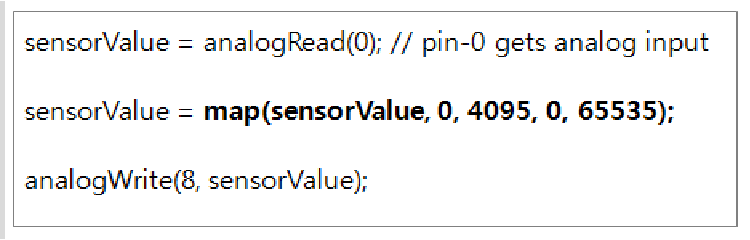
Receive an analog input and get values only between 0 to 100.
constrain(x,a,b) returns x (if x is between and and b).



```
sensorValue = constrain(sensorValue,    0, 100);
```

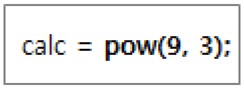
When receiving converter analog values (0~4096) these are mapped 1:1. This is due to PWM having outputs (0~65535).

This can be done with map() function



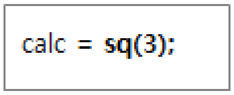
```
sensorValue = analogRead(0); // pin-0 gets analog input

sensorValue = map(sensorValue, 0, 4095, 0, 65535);

analogWrite(8, sensorValue);
```

Calculate 9^3 (nine cube).
Simply implement pow(double x, double y) function



```
calc = pow(9, 3);
```

for squares there's a macro sq(a).
with 3^2

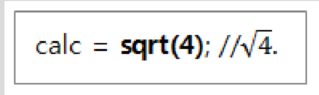


```
calc = sq(3);
```

calc returns 9.

Square roots √‾‾

Simply implement sqrt(double x) function.

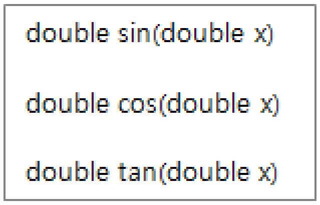


```
calc = sqrt(4); //√4.
```

Calc returns 2.

5.4.2 Output Sin, Cos, Tan

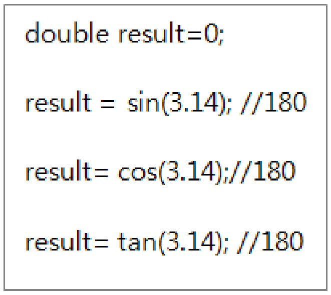
Implement the following functions to obtain sin, cos, and tan.



```
double sin(double x)

double cos(double x)

double tan(double x)
```

where x is in radians.

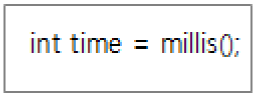
Set a radian value of 3.14



```
double result=0;

result = sin(3.14); //180

result= cos(3.14);//180

result= tan(3.14); //180
```

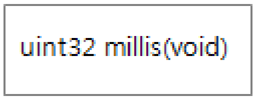
5.5 Time functions

Time unit is in milliseconds



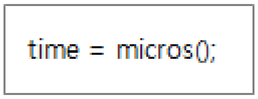
```
int time = millis();
```

The time variable returns millisecond values. Time increases until overflow. Please refer to the millis() function type.



```
uint32 millis(void)
```

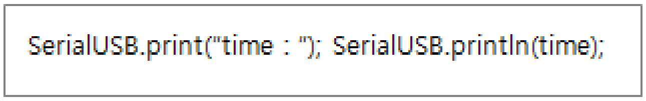
The following has time unit in microseconds.



```
time = micros();
```

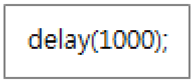
time returns microsecond values. Value increases until overflow (about the 70 min mark) then it resets to 0.

Time variable outputted by SerialUSB device.



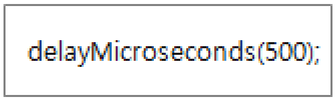
```
SerialUSB.print("time : "); SerialUSB.println(time);
```

Adding delay() to a blinking LED.
The CPU does nothing (remains in standby) for 1 second.
With void delay(unsigned long ms) set a value of 1000 for a delay of 1 second.



```
delay(1000);
```

for reference 1 sec = 1,000 millisecond ,1 millisecond = 1,000microsecond
a short 500us delay.
To implement microsecond delays to the CPU implement
void delayMicroseconds(unsigned int us) function.



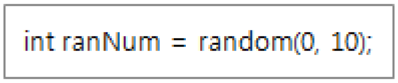
```
delayMicroseconds(500);
```

However, accuracy of the CM-900's CPU(STM32) is not guaranteed with regards to microsecond-type precisions

5.6 Random numbers

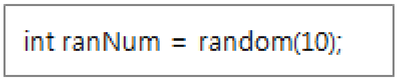
Let’s have 0~10 randomly.

long random(long max) or long random(long min, long max)



```
int ranNum = random(0, 10);
```

there is no need to declared a minimum value; only maximum.



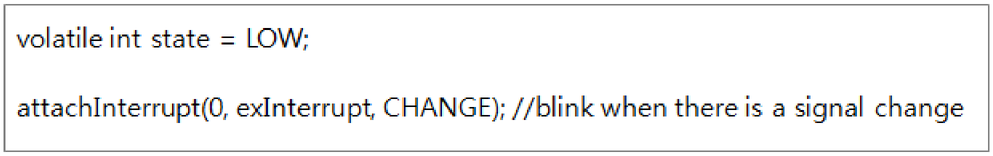
```
int ranNum = random(10);
```

5.7 External interrupt

Have the LED turn on/off when pin-0 gets input signals.

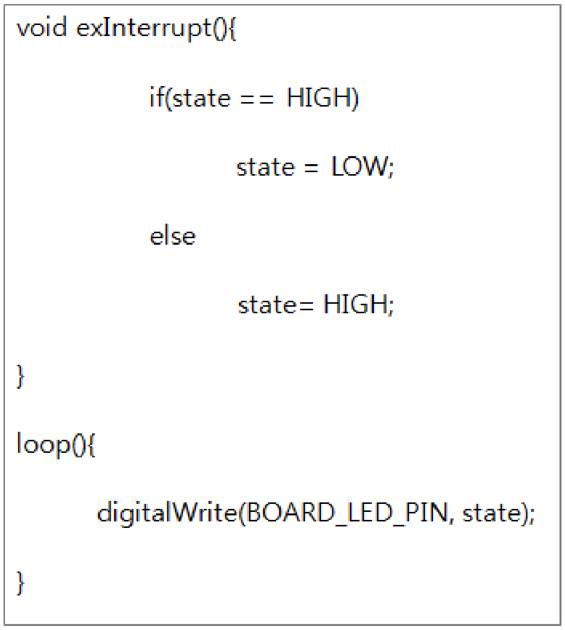
Declare global variables and toggle flags in interrupt routines.

Attach interrupts with attachInterrupt().



```
volatile int state = LOW;

attachInterrupt(0, exInterrupt, CHANGE); //blink when there is a signal change
```

Implement exInterrupt() as void exInterrupt(void) type.



```
void exInterrupt(){

            if(state == HIGH)

                    state = LOW;

            else

                    state= HIGH;

}

loop(){

        digitalWrite(BOARD_LED_PIN, state);

}
```

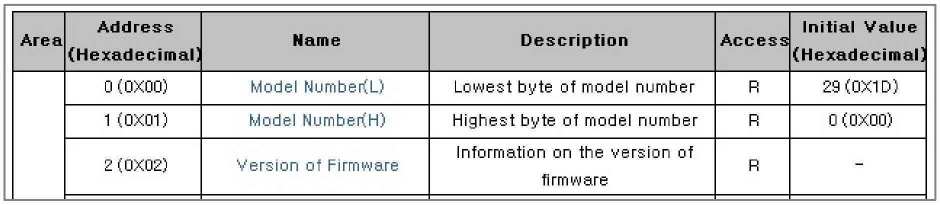
In loop() STATUS LED turns on/off based on state

5.8 Dynamixel

The following example is for ID=1 and baud rate set at 1Mbps [Dxl.begin(1) = 1M bps].

5.8.1 Read the AX-12Afirmware version

The following shows the e-manual's AX-12A control table model number and firmware addresses.



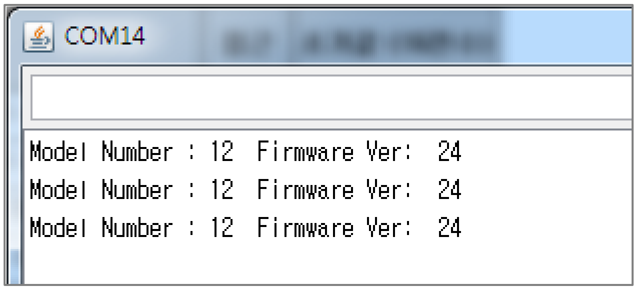
Area	Address (Hexadecimal)	Name	Description	Access	Initial Value (Hexadecimal)
	0 (0X00)	Model Number(L)	Lowest byte of model number	R	29 (0X1D)
	1 (0X01)	Model Number(H)	Highest byte of model number	R	0 (0X00)
	2 (0X02)	Version of Firmware	Information on the version of firmware	R	-

Read ID1's model number (address 0, LSBs portion) and firmware version (address 2).

```
byte nModel = Dxl.readByte(1, 0); // reads model number

byte vFirmware = Dxl.readByte(1, 2); // reads firmware version
```

the following lines are for output.

```
SerialUSB.print("Model Number : ");SerialUSB.print(nModel);

SerialUSB.print(" Firmware Ver : ");SerialUSB.println(vFirmware);
```



```
Model Number : 12  Firmware Ver:  24
Model Number : 12  Firmware Ver:  24
Model Number : 12  Firmware Ver:  24
```

<output>

5.8.2 Read ID1 current temperature

The following shows the temperature address in the control table.

43 (0X2B)	Present Temperature	Current Temperature	R	-

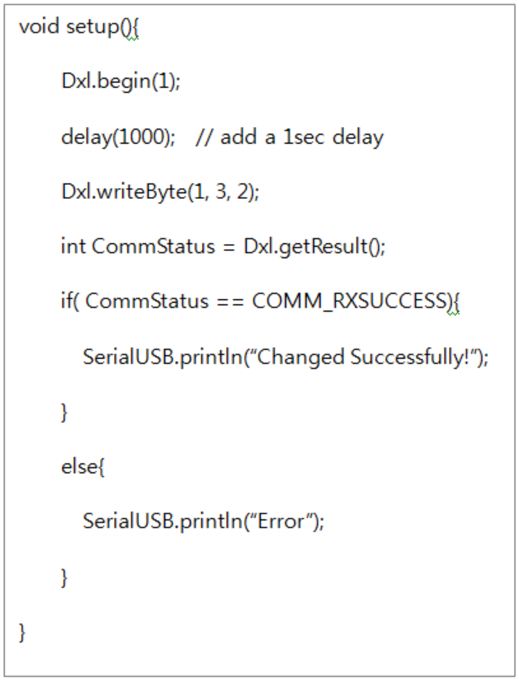
Use readByte() to get data.

```
byte temp = Dxl.readByte(1, 43);

SerialUSB.print("Current Temperature : ");SerialUSB.println(temp);
```

5.8.3 Set the AX-12 to ID2.

Use readWrite() to set address 3.

3 (0X03)	ID	ID of Dynamixel	RW	1 (0X01)



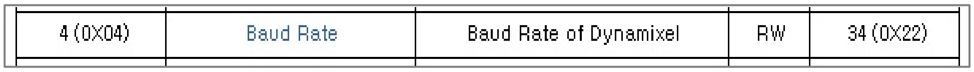
```
void setup(){
    Dxl.begin(1);
    delay(1000);   // add a 1sec delay
    Dxl.writeByte(1, 3, 2);
    int CommStatus = Dxl.getResult();
    if( CommStatus == COMM_RXSUCCESS){
      SerialUSB.println("Changed Successfully!");
    }
    else{
      SerialUSB.println("Error");
    }
}
```

Set ID change in setup(). Always check for communications success.ID 1 is now ID 2.

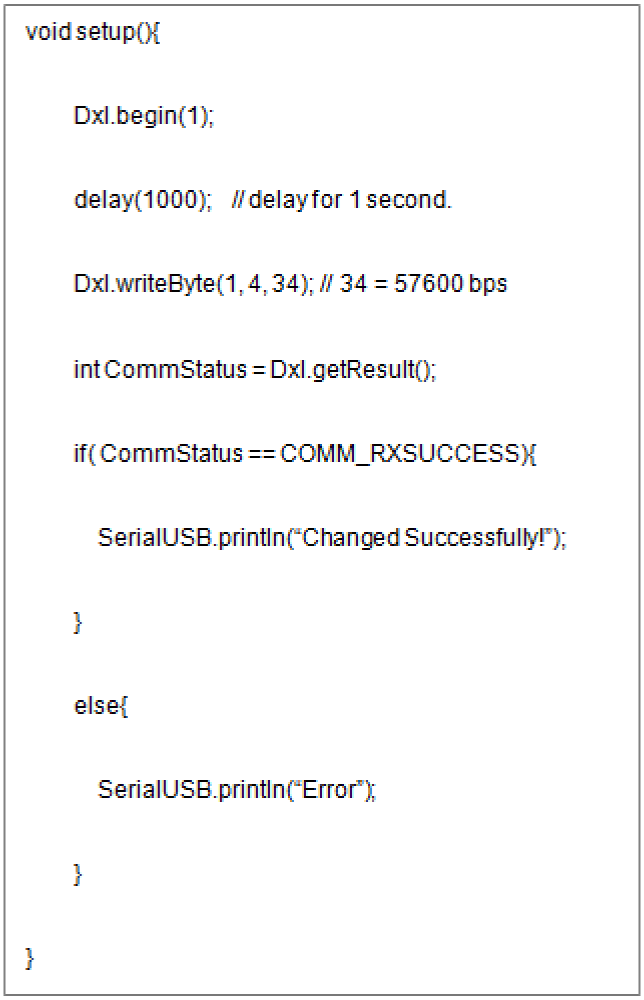
5.8.4 Change baud rate to 57600 bps

To change ID change address 4 (baud rate) via readWrite().

Refer to the index listing the baud rates; 57600 bps has a value of 34.



4 (0X04)	Baud Rate	Baud Rate of Dynamixel	RW	34 (0X22)

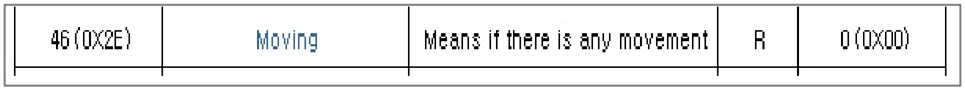


```
void setup(){

    Dxl.begin(1);

    delay(1000);   // delay for 1 second.

    Dxl.writeByte(1, 4, 34); // 34 = 57600 bps

    int CommStatus = Dxl.getResult();

    if( CommStatus == COMM_RXSUCCESS){

      SerialUSB.println(“Changed Successfully!”);

    }

    else{

      SerialUSB.println(“Error”);

    }

}
```

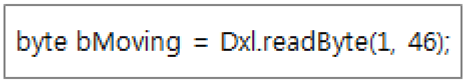
Once baud rate is changed initialize Dynamixel with Dxl.begin(34).

5.8.5 Move (rotate) ID 1

Address 46 (0x2E) of the control table deals with moving aspect.



46 (0X2E)	Moving	Means if there is any movement	R	0 (0X00)



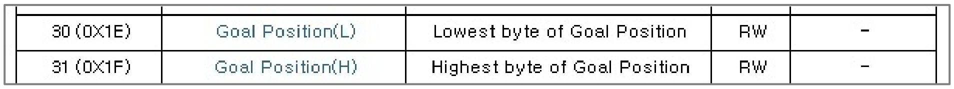
```
byte bMoving = Dxl.readByte(1, 46);
```

When ID1 moves bMoving returns 1; when not 0.

5.8.6 Move the AX-12A to the 150-degree position

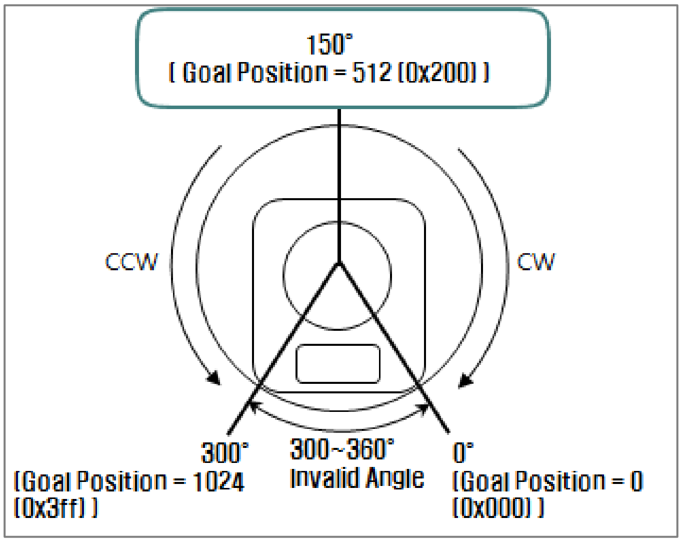
For a goal position of 150 its respective address value must be called (Goal Position L/H).

Goal position is expressed as a word (2 bytes). The table below shows both addresses needed to comprise the Goal Position word. Goal Position (L) (address 30). Use writeWord() to issue position command.



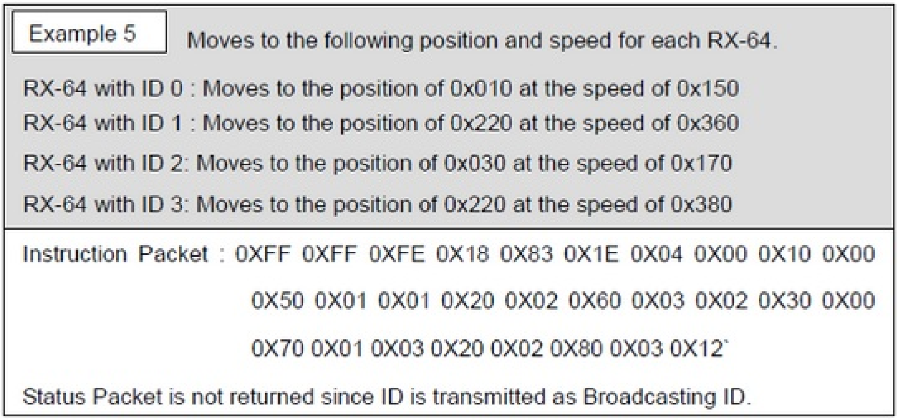
30 (0X1E)	Goal Position(L)	Lowest byte of Goal Position	RW	-
31 (0X1F)	Goal Position(H)	Highest byte of Goal Position	RW	-

Please refer to the diagram below (also found in the e-manuals) with the position with respect to angles.



Use Dxl.getResult() to verify communications.

5.8.7 Set different speeds with several RX-64s



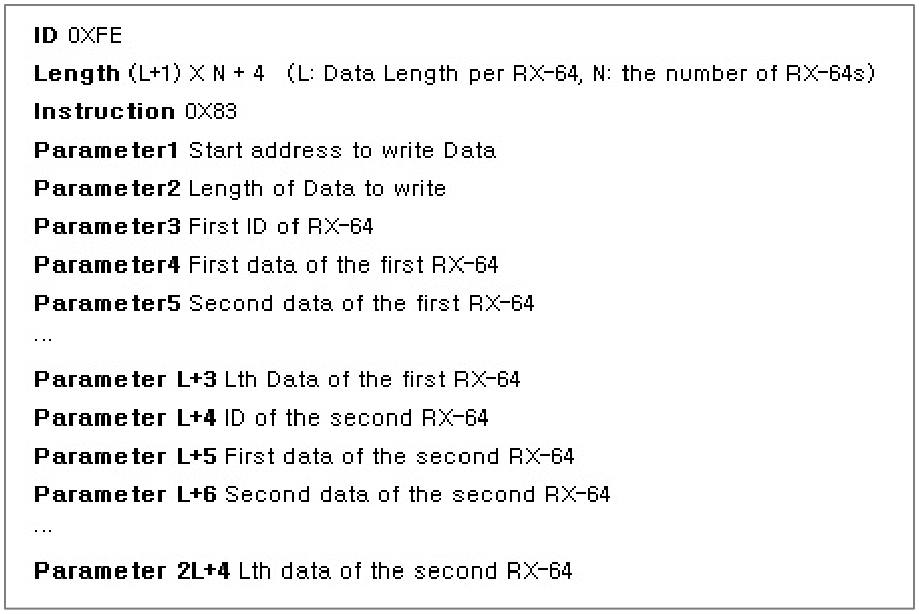
Example 5 Moves to the following position and speed for each RX-64.

RX-64 with ID 0 : Moves to the position of 0x010 at the speed of 0x150
RX-64 with ID 1 : Moves to the position of 0x220 at the speed of 0x360
RX-64 with ID 2: Moves to the position of 0x030 at the speed of 0x170
RX-64 with ID 3: Moves to the position of 0x220 at the speed of 0x380

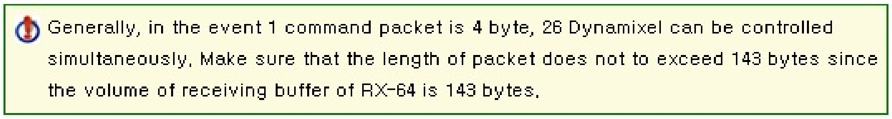
```
Instruction Packet : 0XFF 0XFF 0XFE 0X18 0X83 0X1E 0X04 0X00 0X10 0X00
                     0X50 0X01 0X01 0X20 0X02 0X60 0X03 0X02 0X30 0X00
                     0X70 0X01 0X03 0X20 0X02 0X80 0X03 0X12`
```

Status Packet is not returned since ID is transmitted as Broadcasting ID.

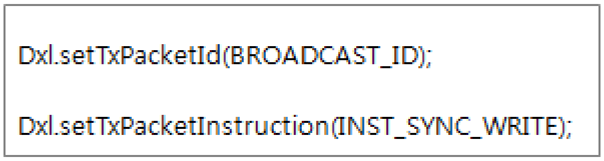
To have several Dynamixels move simultaneously issue the command syncWrite. syncWrite creates a packet then transmits it. Set the packet with setTxPacketXXXX().



ID 0XFE
Length (L+1) X N + 4 (L: Data Length per RX-64, N: the number of RX-64s)
Instruction 0X83
Parameter1 Start address to write Data
Parameter2 Length of Data to write
Parameter3 First ID of RX-64
Parameter4 First data of the first RX-64
Parameter5 Second data of the first RX-64
…
Parameter L+3 Lth Data of the first RX-64
Parameter L+4 ID of the second RX-64
Parameter L+5 First data of the second RX-64
Parameter L+6 Second data of the second RX-64
…
Parameter 2L+4 Lth data of the second RX-64

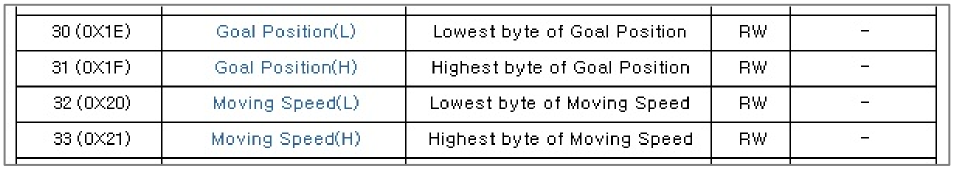


Generally, in the event 1 command packet is 4 byte, 26 Dynamixel can be controlled simultaneously. Make sure that the length of packet does not to exceed 143 bytes since the volume of receiving buffer of RX-64 is 143 bytes.



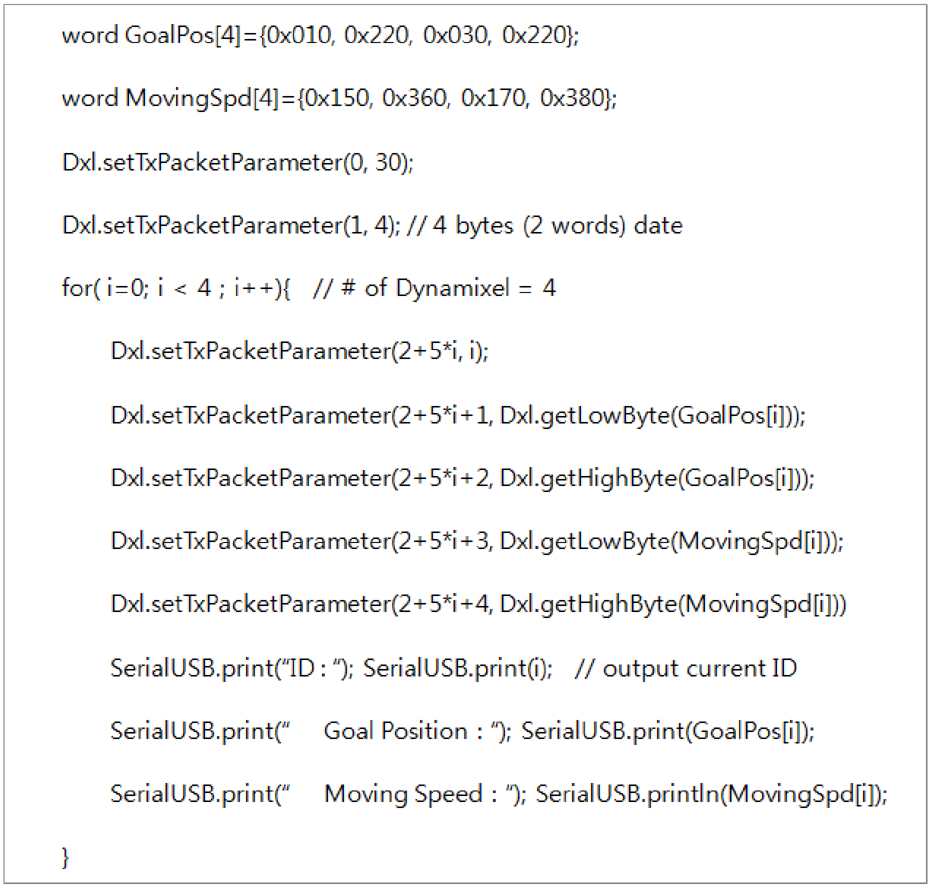
```
Dxl.setTxPacketId(BROADCAST_ID);

Dxl.setTxPacketInstruction(INST_SYNC_WRITE);
```

Set Goal Position and Moving Speed. Dxl.getLowByte() and Dxl.getHighByte() is explicit to high byte (MSBs).

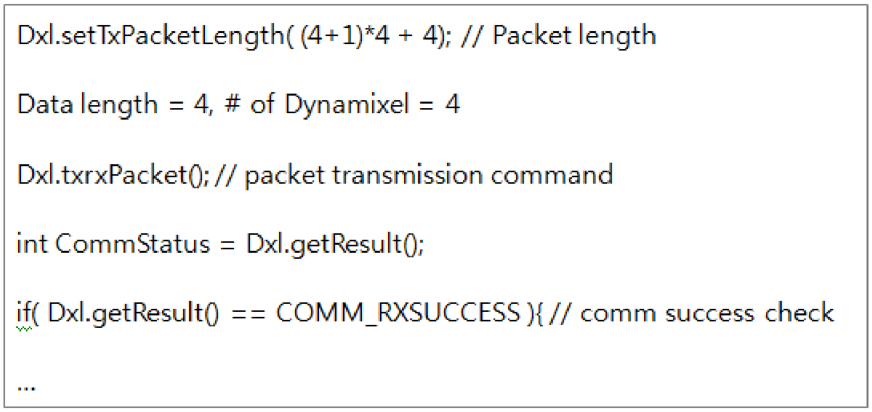


30 (0X1E)	Goal Position(L)	Lowest byte of Goal Position	RW	-
31 (0X1F)	Goal Position(H)	Highest byte of Goal Position	RW	-
32 (0X20)	Moving Speed(L)	Lowest byte of Moving Speed	RW	-
33 (0X21)	Moving Speed(H)	Highest byte of Moving Speed	RW	-

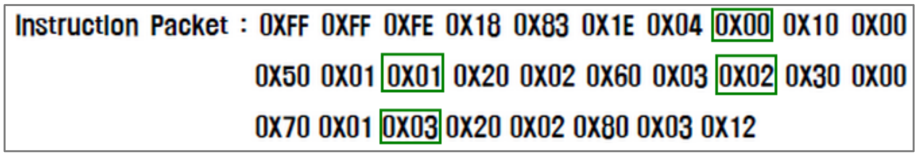
Declare position and velocity values.



```
word GoalPos[4]={0x010, 0x220, 0x030, 0x220};

word MovingSpd[4]={0x150, 0x360, 0x170, 0x380};

Dxl.setTxPacketParameter(0, 30);

Dxl.setTxPacketParameter(1, 4); // 4 bytes (2 words) date

for( i=0; i < 4 ; i++){   // # of Dynamixel = 4

    Dxl.setTxPacketParameter(2+5*i, i);

    Dxl.setTxPacketParameter(2+5*i+1, Dxl.getLowByte(GoalPos[i]));

    Dxl.setTxPacketParameter(2+5*i+2, Dxl.getHighByte(GoalPos[i]));

    Dxl.setTxPacketParameter(2+5*i+3, Dxl.getLowByte(MovingSpd[i]));

    Dxl.setTxPacketParameter(2+5*i+4, Dxl.getHighByte(MovingSpd[i]))

    SerialUSB.print("ID : "); SerialUSB.print(i);   // output current ID

    SerialUSB.print("     Goal Position : "); SerialUSB.print(GoalPos[i]);

    SerialUSB.print("     Moving Speed : "); SerialUSB.println(MovingSpd[i]);

}
```



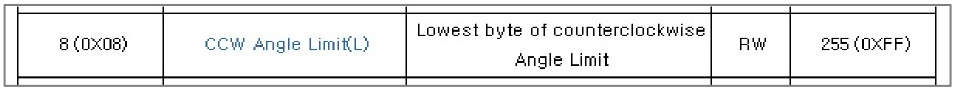
```
Dxl.setTxPacketLength( (4+1)*4 + 4); // Packet length

Data length = 4, # of Dynamixel = 4

Dxl.txrxPacket(); // packet transmission command

int CommStatus = Dxl.getResult();

if( Dxl.getResult() == COMM_RXSUCCESS ){ // comm success check

...
```



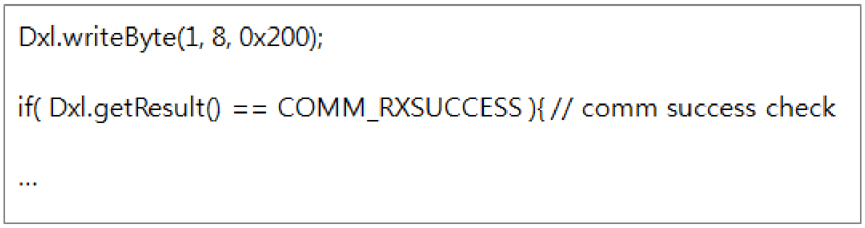
```
Instruction Packet : 0XFF 0XFF 0XFE 0X18 0X83 0X1E 0X04 0X00 0X10 0X00
                     0X50 0X01 0X01 0X20 0X02 0X60 0X03 0X02 0X30 0X00
                     0X70 0X01 0X03 0X20 0X02 0X80 0X03 0X12
```

5.8.8 Limit action between 0~150 degrees

Use CCW Angle Limit 0x3FF to set limit from 300 degrees to 150 degrees. Use writeByte() to send command.



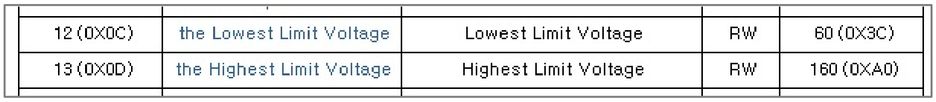
8 (0X08)	CCW Angle Limit(L)	Lowest byte of counterclockwise Angle Limit	RW	255 (0XFF)



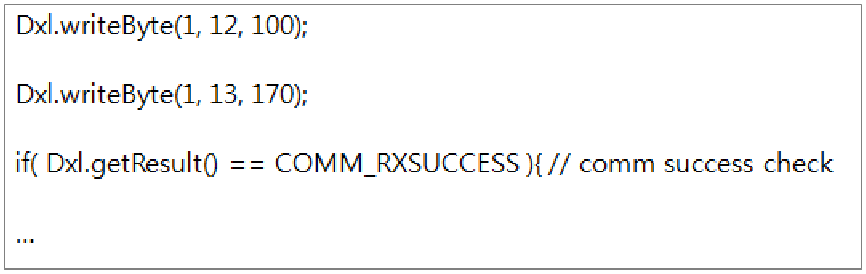
```
Dxl.writeByte(1, 8, 0x200);

if( Dxl.getResult() == COMM_RXSUCCESS ){ // comm success check

...
```

5.8.9 Set input voltage between 10V ~ 17V

10V value is 100(0x64) and 17V is 170(0xAA). Use writeByte() to set commandThe address value from the control table is 12(0x0C) LSBs and 13(0x0D) MSBs.



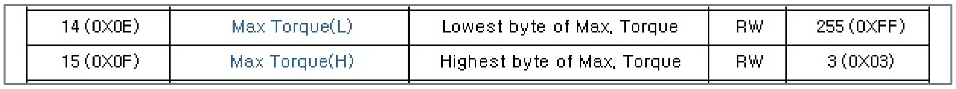
12 (0X0C)	the Lowest Limit Voltage	Lowest Limit Voltage	RW	60 (0X3C)
13 (0X0D)	the Highest Limit Voltage	Highest Limit Voltage	RW	160 (0XA0)



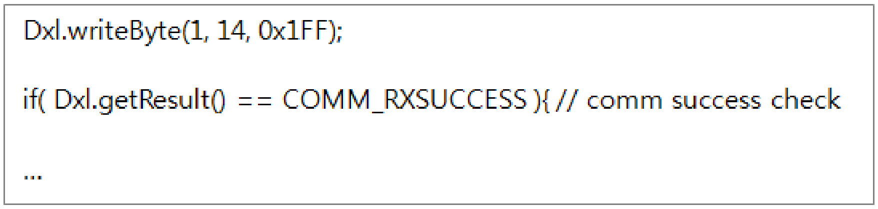
```
Dxl.writeByte(1, 12, 100);

Dxl.writeByte(1, 13, 170);

if( Dxl.getResult() == COMM_RXSUCCESS ){ // comm success check

...
```

5.8.10 Set torque to 50% of max

Set a max Torque (0x3FF) to 50% (0x1FF). Max Torque's LSBs address is 14(0x0E). Use writeByte() to send command.



14 (0X0E)	Max Torque(L)	Lowest byte of Max. Torque	RW	255 (0XFF)
15 (0X0F)	Max Torque(H)	Highest byte of Max. Torque	RW	3 (0X03)

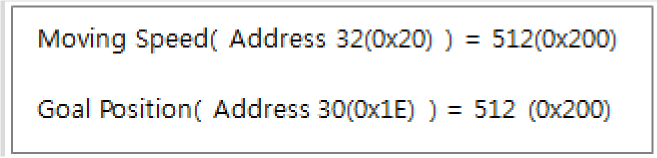


```
Dxl.writeByte(1, 14, 0x1FF);

if( Dxl.getResult() == COMM_RXSUCCESS ){ // comm success check

...
```

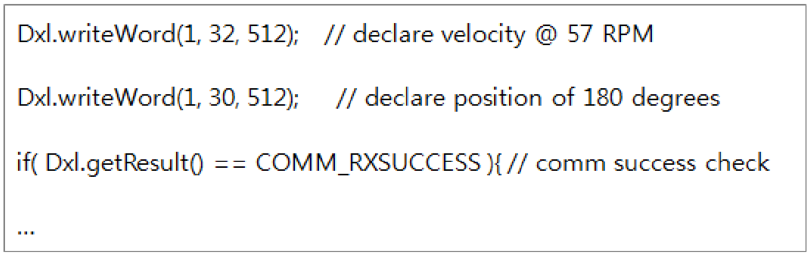
Turn power off then turn it back on to have new torque take place.

5.8.11 Set position of 180 degrees at 57RPM

Declare:



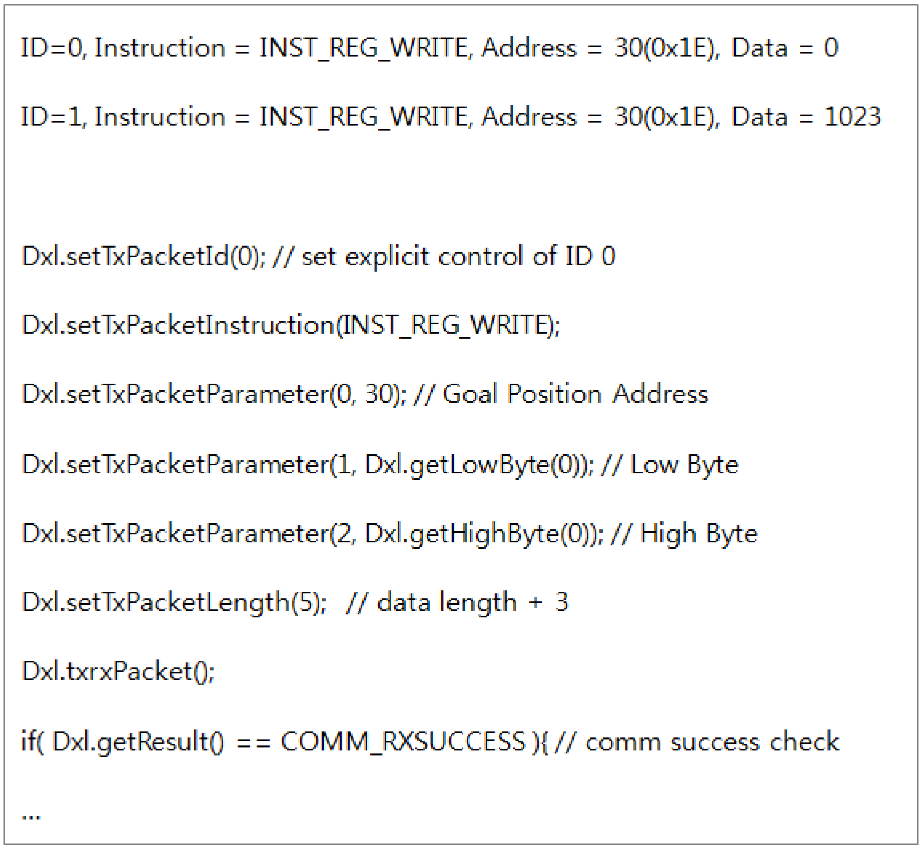
```
Moving Speed( Address 32(0x20) ) = 512(0x200)

Goal Position( Address 30(0x1E) ) = 512 (0x200)
```



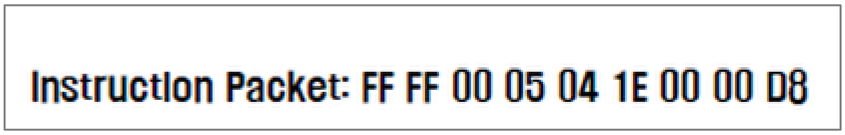
```
Dxl.writeWord(1, 32, 512);   // declare velocity @ 57 RPM

Dxl.writeWord(1, 30, 512);    // declare position of 180 degrees

if( Dxl.getResult() == COMM_RXSUCCESS ){ // comm success check

...
```

5.8.12 Set ID 0 position of 0 and ID 1 of 300 (both must operate simultaneously)

Use Syncwrite and setTxPacketXXX () to create a packet with INST_REG_WRITE and INST_ACTION. For reference 0 degrees is 0 (0x000) and 300 degrees is 1023 (0x3FF).

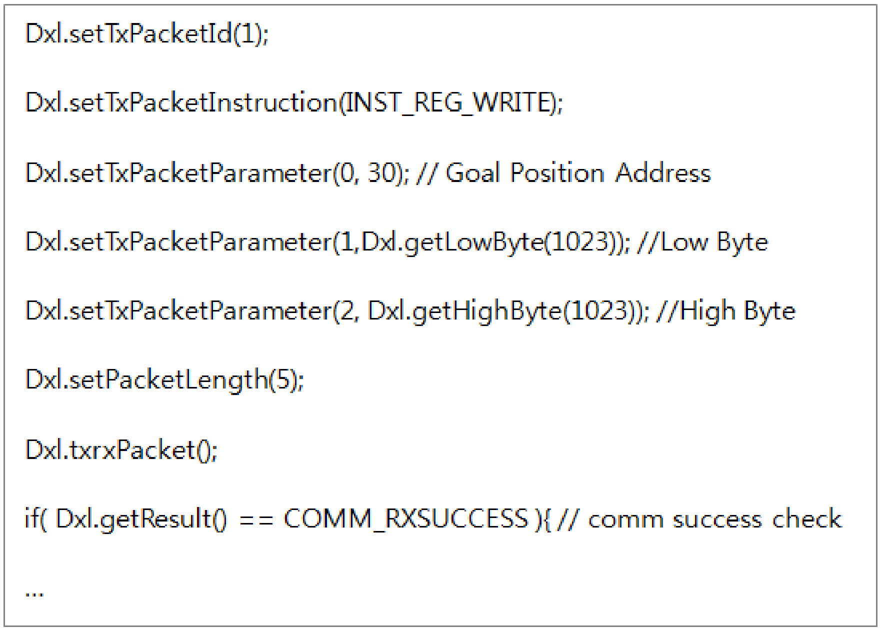


```
ID=0, Instruction = INST_REG_WRITE, Address = 30(0x1E), Data = 0

ID=1, Instruction = INST_REG_WRITE, Address = 30(0x1E), Data = 1023


Dxl.setTxPacketId(0); // set explicit control of ID 0

Dxl.setTxPacketInstruction(INST_REG_WRITE);

Dxl.setTxPacketParameter(0, 30); // Goal Position Address

Dxl.setTxPacketParameter(1, Dxl.getLowByte(0)); // Low Byte

Dxl.setTxPacketParameter(2, Dxl.getHighByte(0)); // High Byte

Dxl.setTxPacketLength(5);  // data length + 3

Dxl.txrxPacket();

if( Dxl.getResult() == COMM_RXSUCCESS ){ // comm success check

...
```

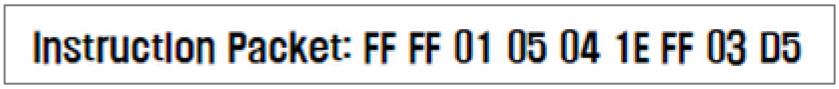


```
Instruction Packet: FF FF 00 05 04 1E 00 00 D8
```

Second Dynamixel packet transmission

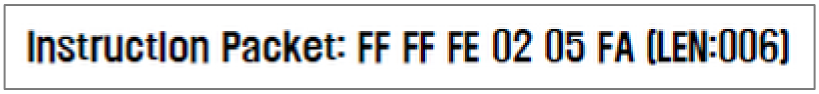


```
Dxl.setTxPacketId(1);

Dxl.setTxPacketInstruction(INST_REG_WRITE);

Dxl.setTxPacketParameter(0, 30); // Goal Position Address

Dxl.setTxPacketParameter(1,Dxl.getLowByte(1023)); //Low Byte

Dxl.setTxPacketParameter(2, Dxl.getHighByte(1023)); //High Byte

Dxl.setPacketLength(5);

Dxl.txrxPacket();

if( Dxl.getResult() == COMM_RXSUCCESS ){ // comm success check

...
```



```
Instruction Packet: FF FF 01 05 04 1E FF 03 D5
```

While ID0 and ID1 are pending INST_ACTION packet is transmitted to run instructions



```
Instruction Packet: FF FF FE 02 05 FA (LEN:006)
```

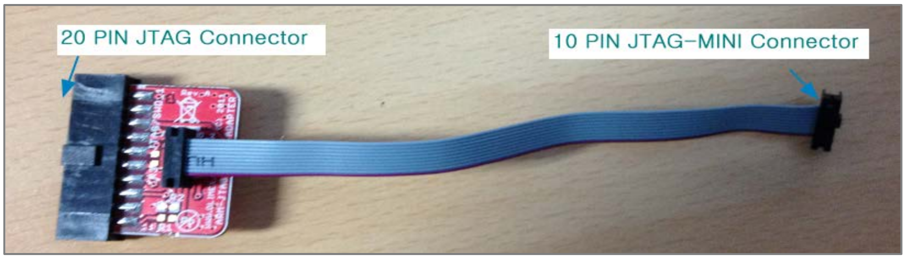
Check for communications success every time a packet is sent.

6 Appendix 1: Download the bootloader with the ST-LINK

6.1 Download the bootloader with the ST-LINK

6.1.1 The CM-900 has a 10-pin JTAG header. Connect a ST-LINK to download a newbootloader.

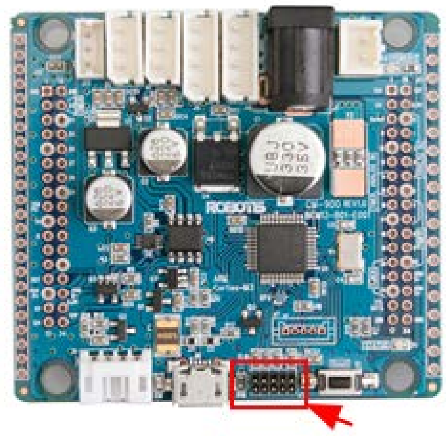
<ST-LINK/V2>



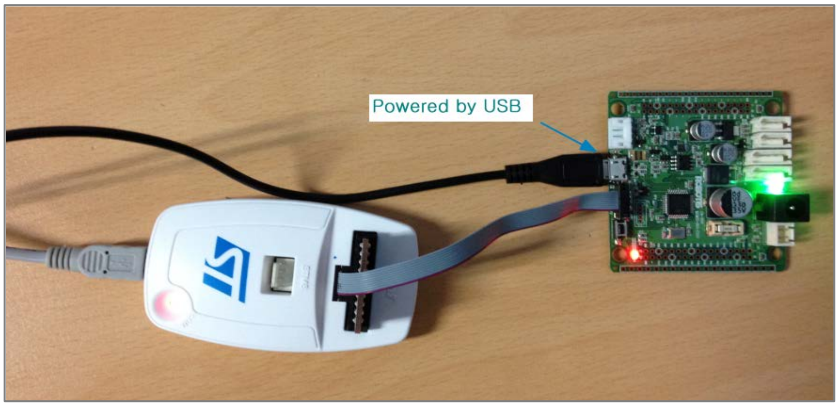
< 20 PIN to 10 PIN Converter >

However, the ST-LINK/V2 needs a 20PIN to 10 PIN JTAG Converter to properly connect to the CM-900.

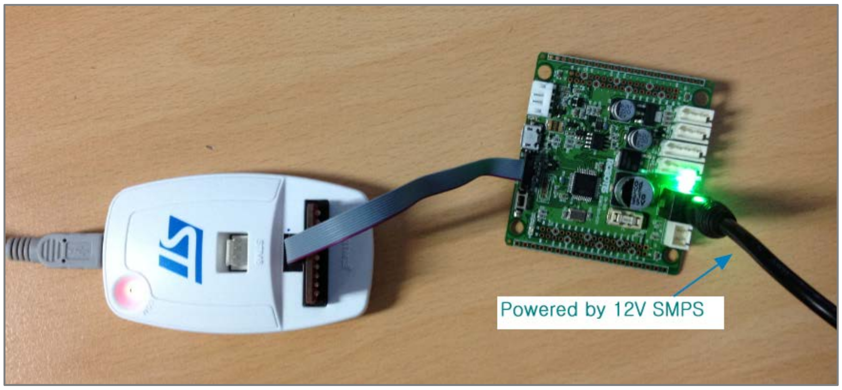
Connect the converter’s 20-pin end to the ST-LINK/V2 and the 10-pin and to the CM-900.



6.2 When connecting the ST-LINK the CM-900 must have power supplied separately by either USB or SMPS/Battery.



<USB-based power>

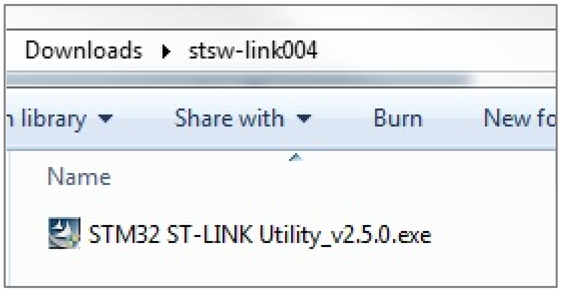


<SMPS-based power>

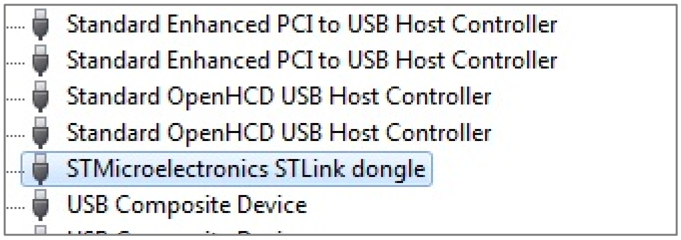
6.3 Download the ST-LINK Utility (drivers included).

http://www.st.com/web/en/catalog/tools/PF258168

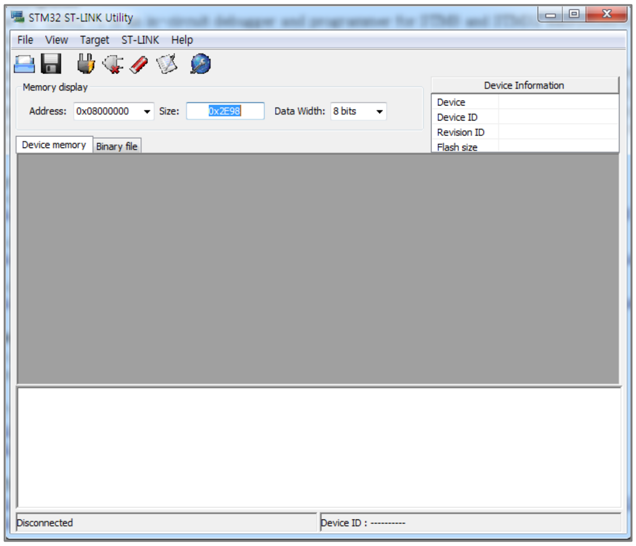
Decompress the file and double-click on STM32 ST-LINK Utility_v2.x.x.exe and install everything, including ST-LINK's driver.



From Windows Device Manager the STLink dongle will show up as STMicroelectronics STLink dongle.


Standard Enhanced PCI to USB Host Controller
Standard Enhanced PCI to USB Host Controller
Standard OpenHCD USB Host Controller
Standard OpenHCD USB Host Controller
STMicroelectronics STLink dongle
USB Composite Device

Run STM32 ST-LINK Utility.

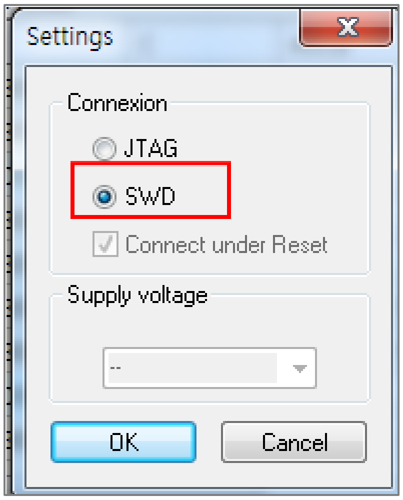


Address: 0x08000000 Size: 0x2E98 Data Width: 8 bits

Disconnected

Device ID : ---------

From Target -> Settings choose SWD(Serial-Wire Debug).



Settings

Connexion

○ JTAG

● SWD

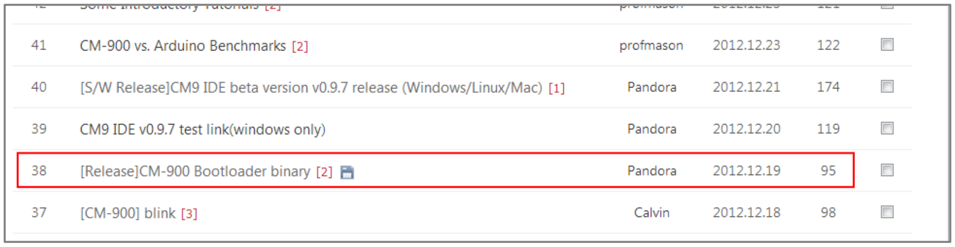
☑ Connect under Reset

Supply voltage

--

6.4 Download the CM-900's bootloader.

http://www.robotsource.org/xe/Circle_CM9_Developer_World

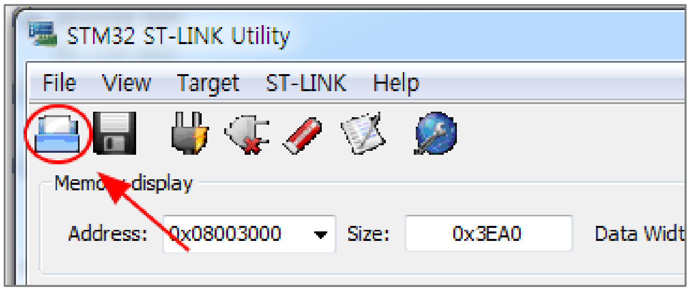


42	Some Introductory Tutorials [2]	profmason	2012.12.23	121	□
41	CM-900 vs. Arduino Benchmarks [2]	profmason	2012.12.23	122	□
40	[S/W Release]CM9 IDE beta version v0.9.7 release (Windows/Linux/Mac) [1]	Pandora	2012.12.21	174	□
39	CM9 IDE v0.9.7 test link(windows only)	Pandora	2012.12.20	119	□
38	[Release]CM-900 Bootloader binary [2]	Pandora	2012.12.19	95	□
37	[CM-900] blink [3]	Calvin	2012.12.18	98	□

For bootloader updates check www.robotsource.org periodically.

Like all CM-900 software the bootloader can also be modified and built. For more information refer to Appedix 2.

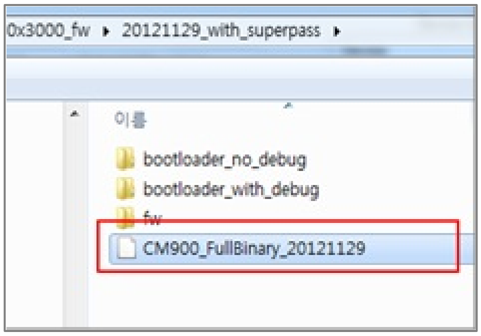
6.5 From File -> Open file select CM900_FullBinary_20XXXXXX.bin (see icon shown below).



STM32 ST-LINK Utility

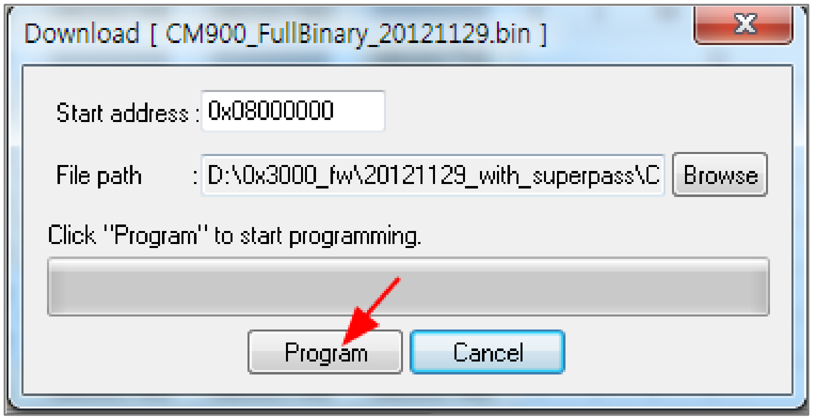
Address: 0x08003000 Size: 0x3EA0 Data Widt

When the window pops up choose the binary file.



0x3000_fw ▸ 20121129_with_superpass ▸
이름
bootloader_no_debug
bootloader_with_debug
fw
CM900_FullBinary_20121129

6.6 Go to Target -> Program or Program & Verify . Leave Start address as is (0x08000000).



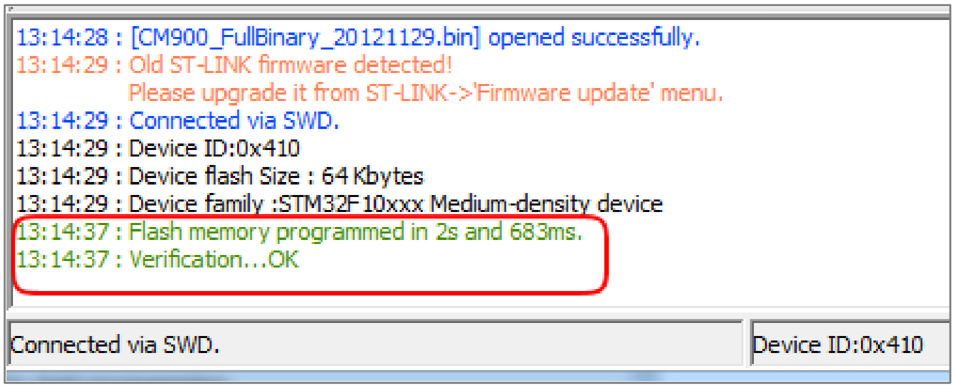
Download [CM900_FullBinary_20121129.bin]

Start address : 0x08000000

File path : D:\0x3000_fw\20121129_with_superpass\C

Click "Program" to start programming.

After integrity check is complete (Verification) you get a verification notification.

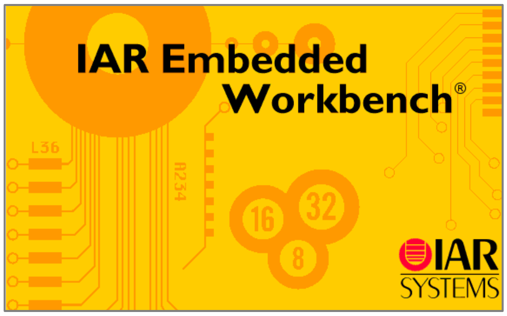


```
13:14:28 : [CM900_FullBinary_20121129.bin] opened successfully.
13:14:29 : Old ST-LINK firmware detected!
           Please upgrade it from ST-LINK->'Firmware update' menu.
13:14:29 : Connected via SWD.
13:14:29 : Device ID:0x410
13:14:29 : Device flash Size : 64 Kbytes
13:14:29 : Device family :STM32F10xxx Medium-density device
13:14:37 : Flash memory programmed in 2s and 683ms.
13:14:37 : Verification...OK
```

Connected via SWD. | Device ID:0x410

Download is now complete. Press the reset button of the CM-900.

6.7 Advanced users can use development environment tools like IAR E/W, Keil uVision, etc; and use the CM-900's JTAG port to edit, doenload, step-by-step debugging, and create custom firmware. However, to use ROBOTIS CM-9 software then the bootloader must be downloaded again..



<IAR Embedded Workbench>

<Keil uVision>

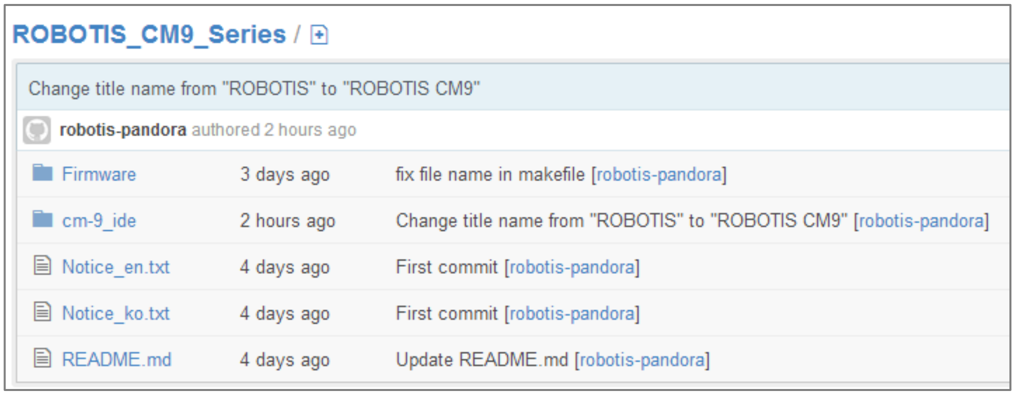
7 Appendix 2 utilizing the source

The CM-900's hardware and software are open-source. You can access the source via Github. You can use the source to develop your own robot development environment and share your code with everybody.

The descriptions below are based on Windows OS but Linux and Mac OSX is also possible with Eclipse.

7.1 ROBOTIS CM9 source location

https://github.com/robotis-pandora/ROBOTIS_CM9_Series.git



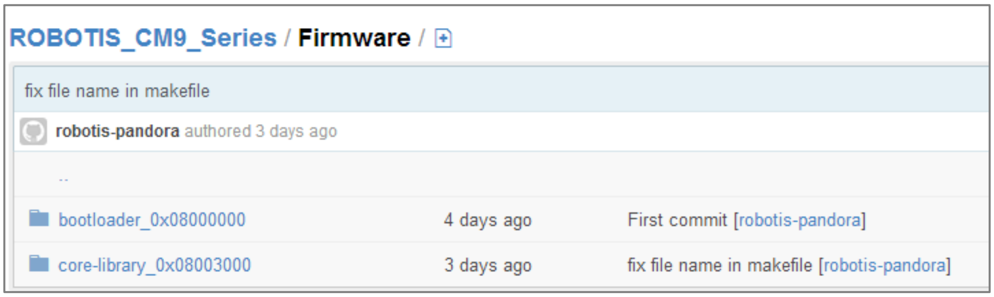
ROBOTIS_CM9_Series /

Change title name from "ROBOTIS" to "ROBOTIS CM9"

robotis-pandora authored 2 hours ago

Firmware	3 days ago	fix file name in makefile [robotis-pandora]
cm-9_ide	2 hours ago	Change title name from "ROBOTIS" to "ROBOTIS CM9" [robotis-pandora]
Notice_en.txt	4 days ago	First commit [robotis-pandora]
Notice_ko.txt	4 days ago	First commit [robotis-pandora]
README.md	4 days ago	Update README.md [robotis-pandora]

7.1.1 Firmware: contains the components for the bootloader. The ROBOTIS CM9's Wiring APIs, compiler and firmware also located in core library. The CM-9's compiler and downloadable firmware located in core-library_0x08003000 directory. The bootloader starts at the flash memory's 0x08000000 location the downloadable firmware starts at 0x08003000.



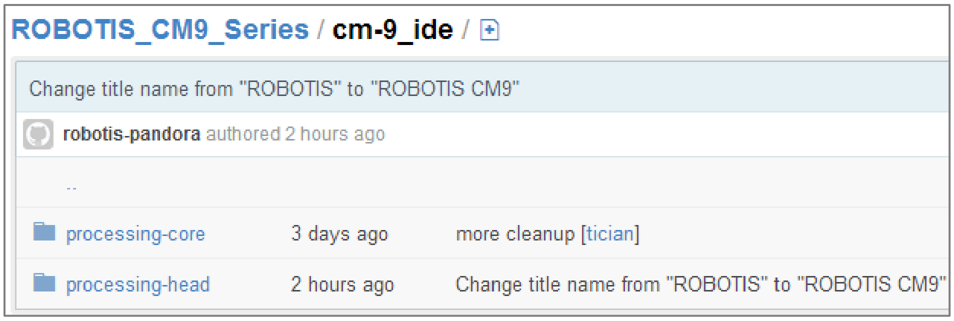
ROBOTIS_CM9_Series / **Firmware** /

fix file name in makefile

robotis-pandora authored 3 days ago

..		
bootloader_0x08000000	4 days ago	First commit [robotis-pandora]
core-library_0x08003000	3 days ago	fix file name in makefile [robotis-pandora]

7.1.2 CM-9_ide: The ROBOTIS CM9 is based on Arduino 1.0.1, where Arduino is based on Processing, hence the reference to Processing-core project. In ROBOTIS CM9's IDE is implemented as Processing-head. Therefore, to develop the IDE modify processing-head



ROBOTIS_CM9_Series / cm-9_ide /

Change title name from "ROBOTIS" to "ROBOTIS CM9"

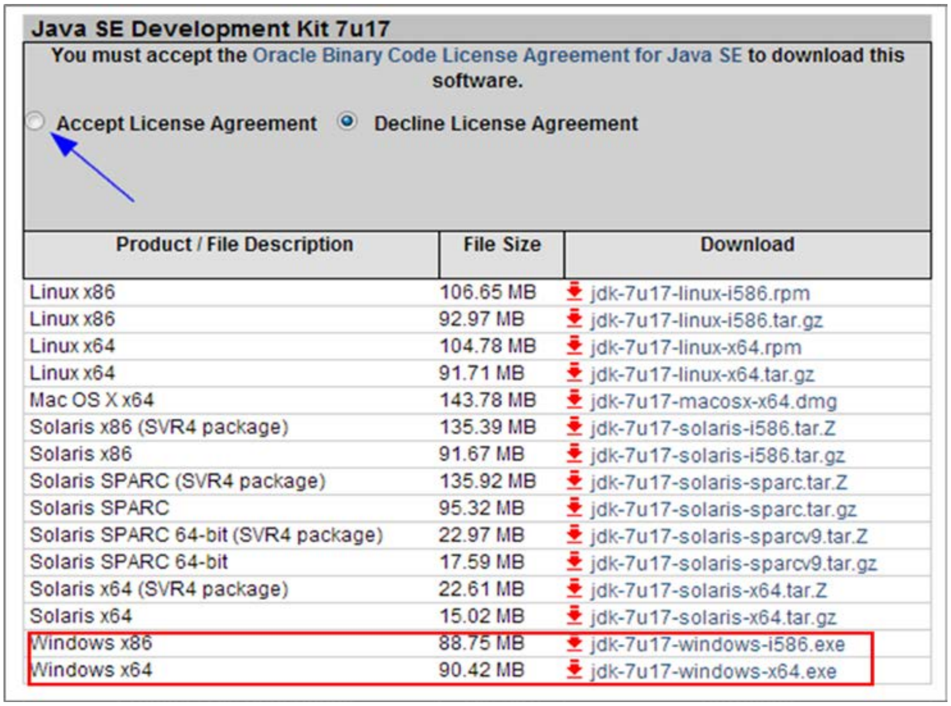
robotis-pandora authored 2 hours ago

..

processing-core	3 days ago	more cleanup [tician]
processing-head	2 hours ago	Change title name from "ROBOTIS" to "ROBOTIS CM9"

7.2 To run Eclipse download and install Java Development kit.

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html



Java SE Development Kit 7u17

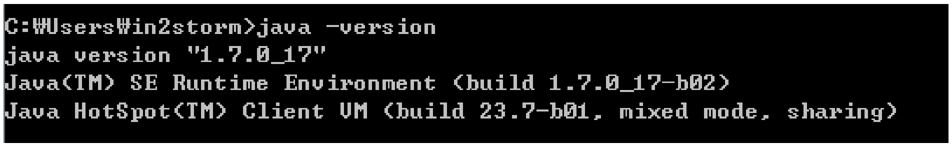
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ● Decline License Agreement

Product / File Description	File Size	Download
Linux x86	106.65 MB	jdk-7u17-linux-i586.rpm
Linux x86	92.97 MB	jdk-7u17-linux-i586.tar.gz
Linux x64	104.78 MB	jdk-7u17-linux-x64.rpm
Linux x64	91.71 MB	jdk-7u17-linux-x64.tar.gz
Mac OS X x64	143.78 MB	jdk-7u17-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.39 MB	jdk-7u17-solaris-i586.tar.Z
Solaris x86	91.67 MB	jdk-7u17-solaris-i586.tar.gz
Solaris SPARC (SVR4 package)	135.92 MB	jdk-7u17-solaris-sparc.tar.Z
Solaris SPARC	95.32 MB	jdk-7u17-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	22.97 MB	jdk-7u17-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.59 MB	jdk-7u17-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	22.61 MB	jdk-7u17-solaris-x64.tar.Z
Solaris x64	15.02 MB	jdk-7u17-solaris-x64.tar.gz
Windows x86	88.75 MB	jdk-7u17-windows-i586.exe
Windows x64	90.42 MB	jdk-7u17-windows-x64.exe

Click on Accept then download and install the appropriate version for your OS.

TO verify proper installation of JDK enter the command as shown below.

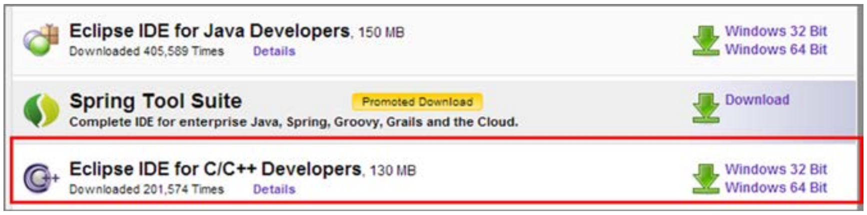


```
C:₩Users₩in2storm>java -version
java version "1.7.0_17"
Java(TM) SE Runtime Environment (build 1.7.0_17-b02)
Java HotSpot(TM) Client VM (build 23.7-b01, mixed mode, sharing)
```

7.3 Importing your project in Eclipse

Use Eclipse's git plug-in for easy acquisition of sources from GitHub. Get C/C++ language type for firmware and Java for IDE. Get the most recent version.

Eclipse download : http://www.eclipse.org/
Get the C/C++ package.



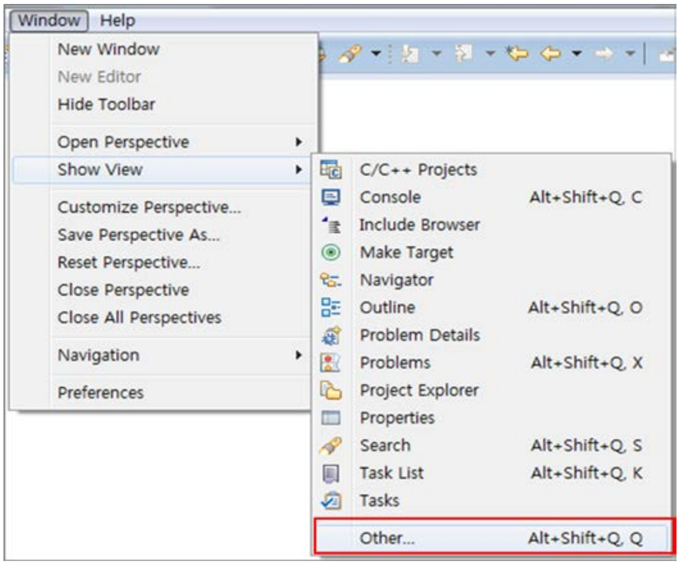
Eclipse IDE for Java Developers, 150 MB
Downloaded 405,589 Times Details
Windows 32 Bit
Windows 64 Bit

Spring Tool Suite Promoted Download
Complete IDE for enterprise Java, Spring, Groovy, Grails and the Cloud.
Download

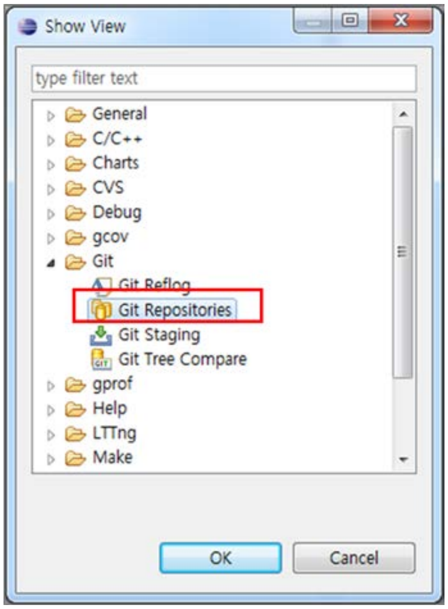
Eclipse IDE for C/C++ Developers, 130 MB
Downloaded 201,574 Times Details
Windows 32 Bit
Windows 64 Bit

In your PC download either 32-bit or 64-bit version. To check Java version enter the command java –version. If the environment version is 32-bit but you got the 64-bit then it won't run and viceversa.

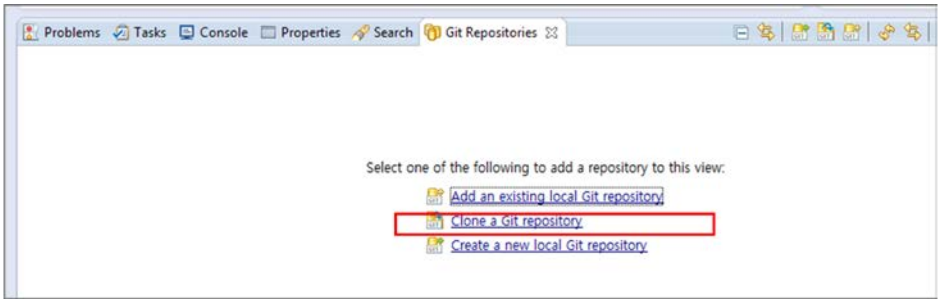
7.3.1 Run the git plug-in from Eclipse.



Go to Window -> Show View -> Other…



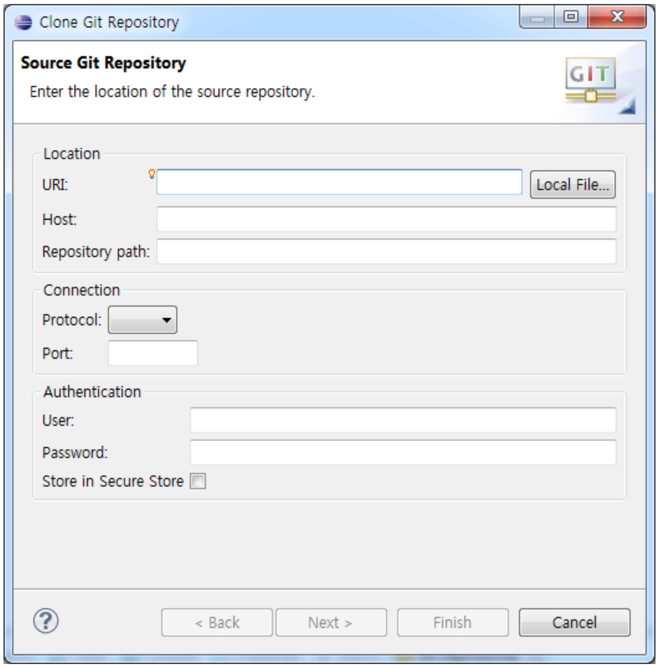
Select **Git Repositories** and the git plug-in will appear in a view window.



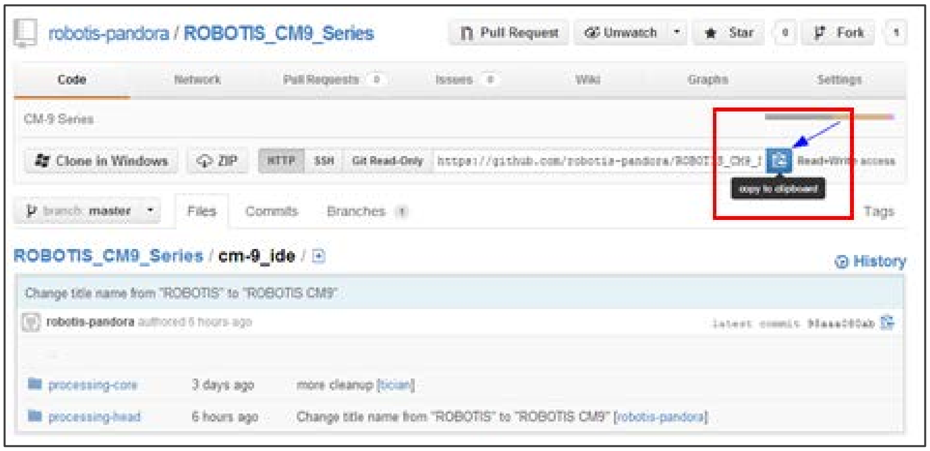
Select one of the following to add a repository to this view:

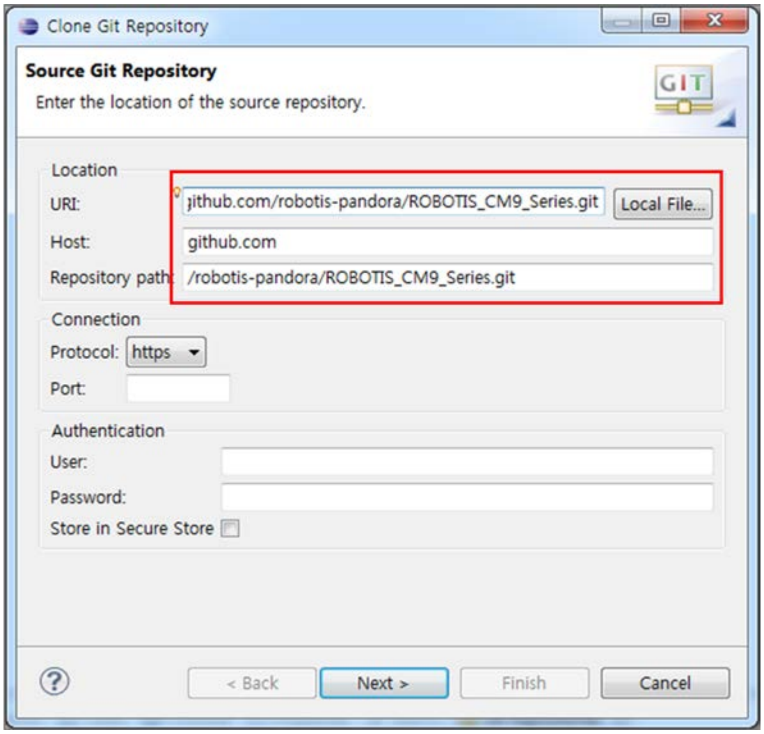
Add an existing local Git repository
Clone a Git repository
Create a new local Git repository

Click on **Clone a Git repository**



Copy the link https://github.com/robotis-pandora/ROBOTIS_CM9_Series.git. The other parts should autofill. Alternately, from the ROBOTIS_CM-9_Series's Github site click on the copy to clipboard icon.





Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI: jithub.com/robotis-pandora/ROBOTIS_CM9_Series.git

Host: github.com

Repository path: /robotis-pandora/ROBOTIS_CM9_Series.git

Connection

Protocol: https

Port:

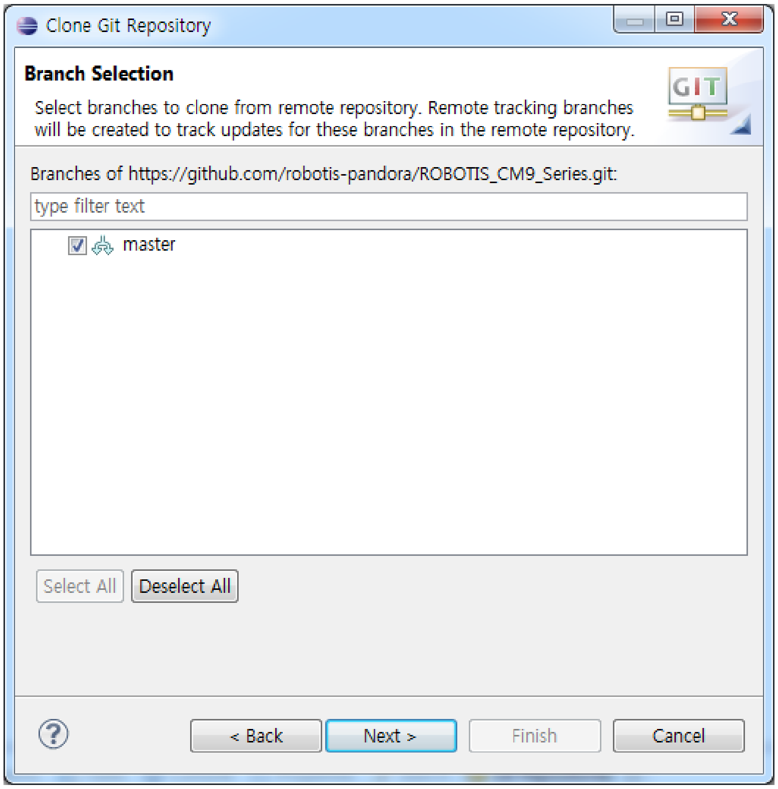
Authentication

User:

Password:

Store in Secure Store □

Click on next



Clone Git Repository

Branch Selection

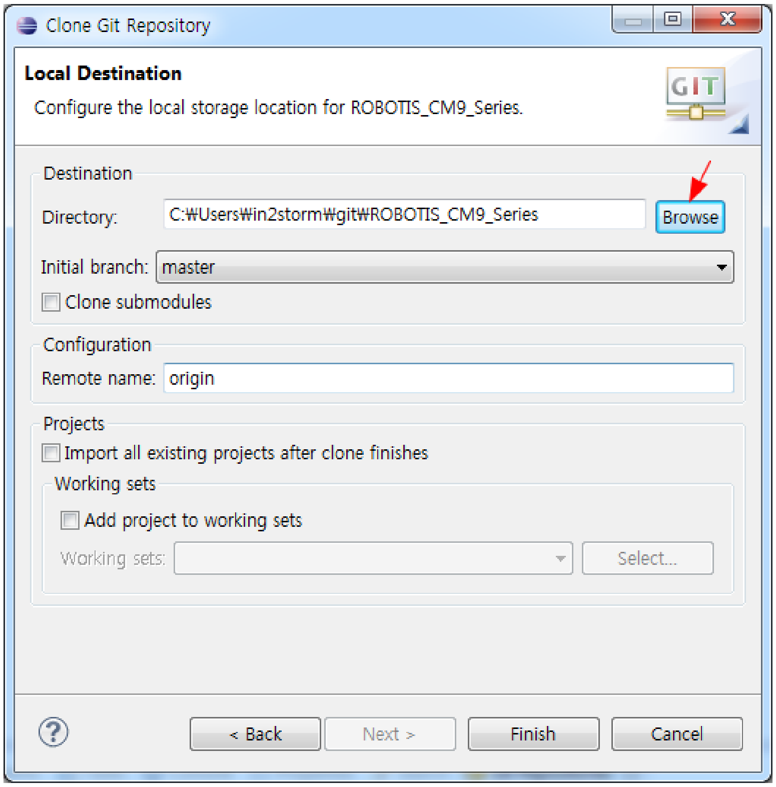
Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.

Branches of https://github.com/robotis-pandora/ROBOTIS_CM9_Series.git:

type filter text

☑ master

Click on next



Clone Git Repository

Local Destination

Configure the local storage location for ROBOTIS_CM9_Series.

Destination

Directory: C:₩Users₩in2storm₩git₩ROBOTIS_CM9_Series

Initial branch: master

□ Clone submodules

Configuration

Remote name: origin

Projects

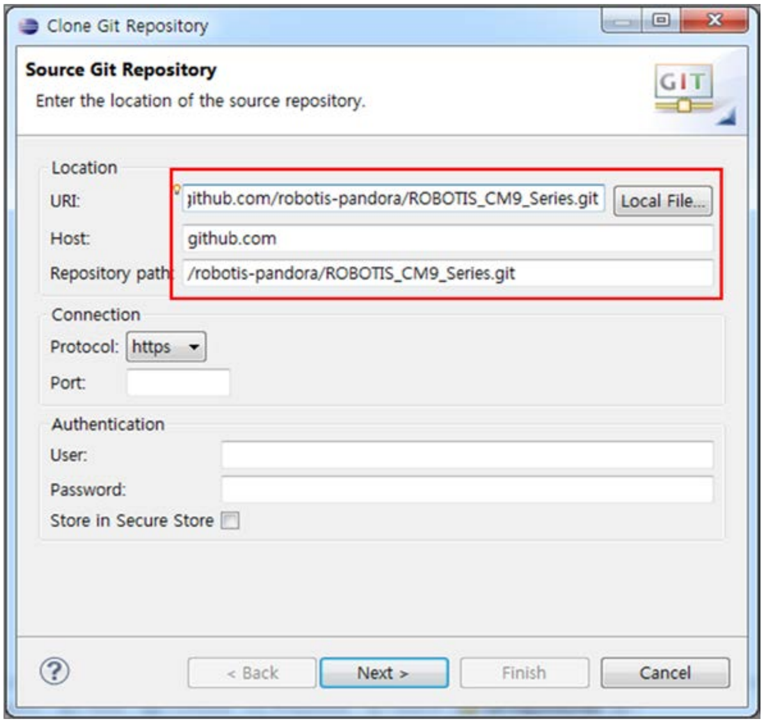
□ Import all existing projects after clone finishes

Working sets

□ Add project to working sets

Working sets:

In Destination select the repository directory.



Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI: ;ithub.com/robotis-pandora/ROBOTIS_CM9_Series.git

Host: github.com

Repository path: /robotis-pandora/ROBOTIS_CM9_Series.git

Connection

Protocol: https

Port:

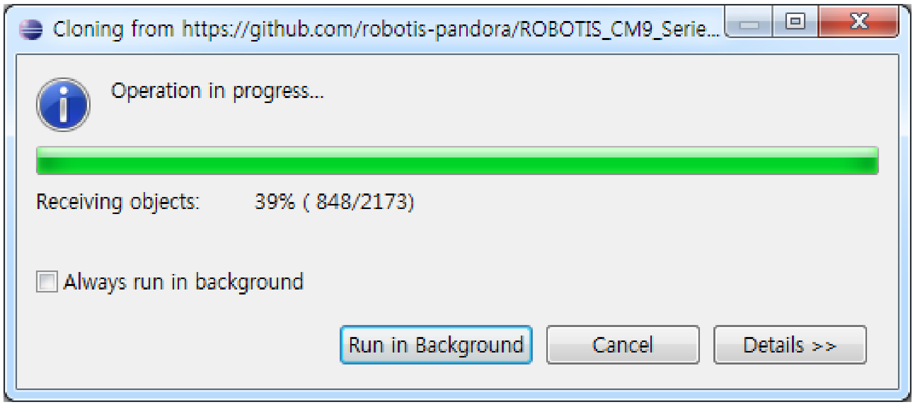
Authentication

User:

Password:

Store in Secure Store □

Click on Finish to finish obtaining the source.



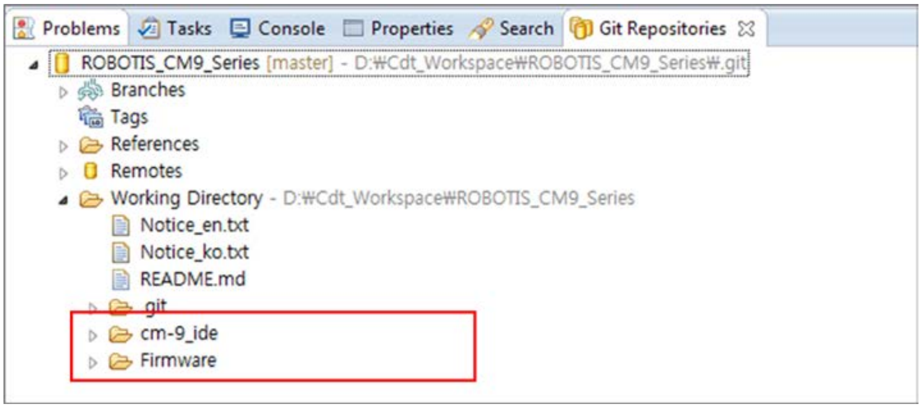
Cloning from https://github.com/robotis-pandora/ROBOTIS_CM9_Serie...

Operation in progress...

Receiving objects: 39% (848/2173)

□ Always run in background

Once cloning is complete a Master branch appears. Expand the node until you see ROBOTIS CM9.

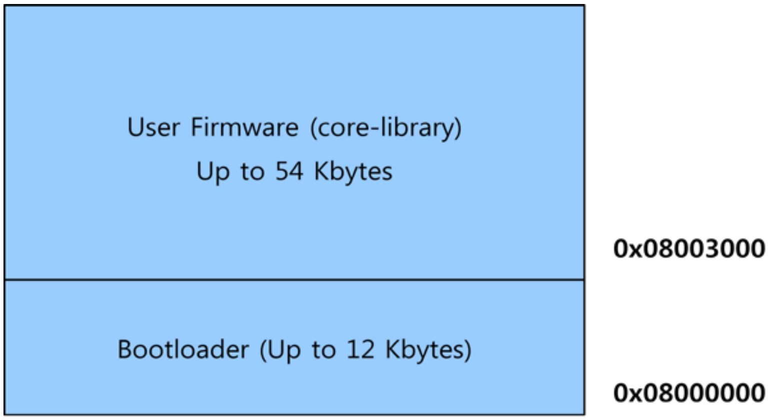


```
ROBOTIS_CM9_Series [master] - D:₩Cdt_Workspace₩ROBOTIS_CM9_Series₩.git
  Branches
  Tags
  References
  Remotes
  Working Directory - D:₩Cdt_Workspace₩ROBOTIS_CM9_Series
    Notice_en.txt
    Notice_ko.txt
    README.md
    .git
    cm-9_ide
    Firmware
```

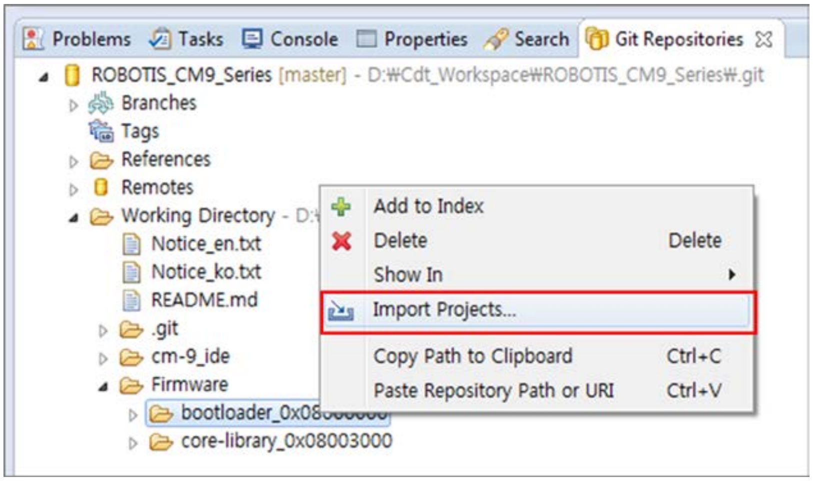
You can develop firmware with Eclipse C/C++ package (firmware folder only). For ROBOTIS CM-9 IDE use Eclipse Java package.

7.3.2 Importing the bootloader project

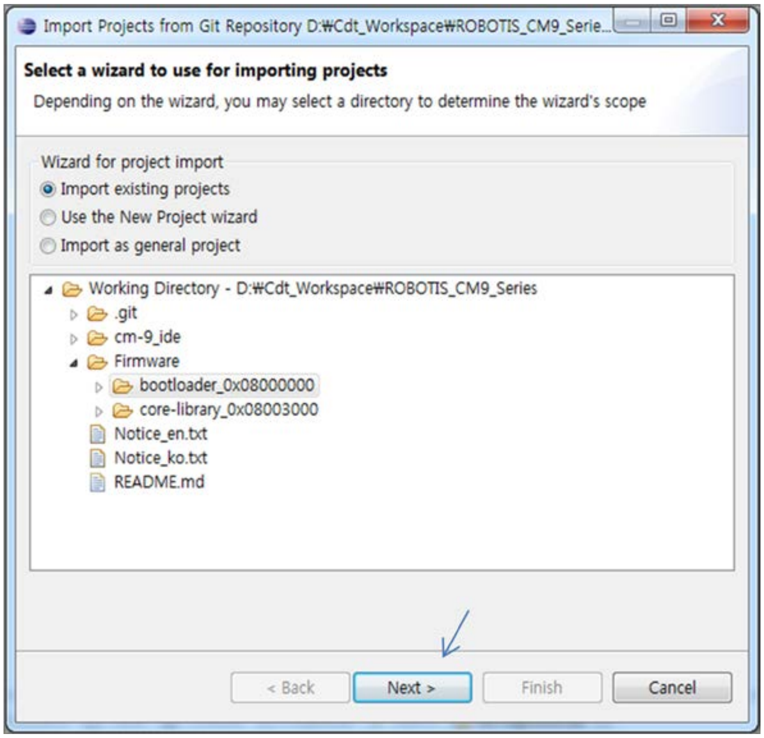
The CM-900 bootloader resides in the CPU's (STM32F103C8) internal flash memory from 0x08000000 and up to 12-kbytes in size. 12 kbytes should not be exceeded otherwise it would take up space from core-library, from 0x08003000, therefore firmware cannot be executed.



In the repository's Working Directory go to Firmware -> bootloader_0x08000000; with the right mouse click select Import Projects…



From **Wizard for project import** select **Import existing projects** and click on **Next**.



Import Projects from Git Repository D:₩Cdt_Workspace₩ROBOTIS_CM9_Serie...

Select a wizard to use for importing projects

Depending on the wizard, you may select a directory to determine the wizard's scope

Wizard for project import

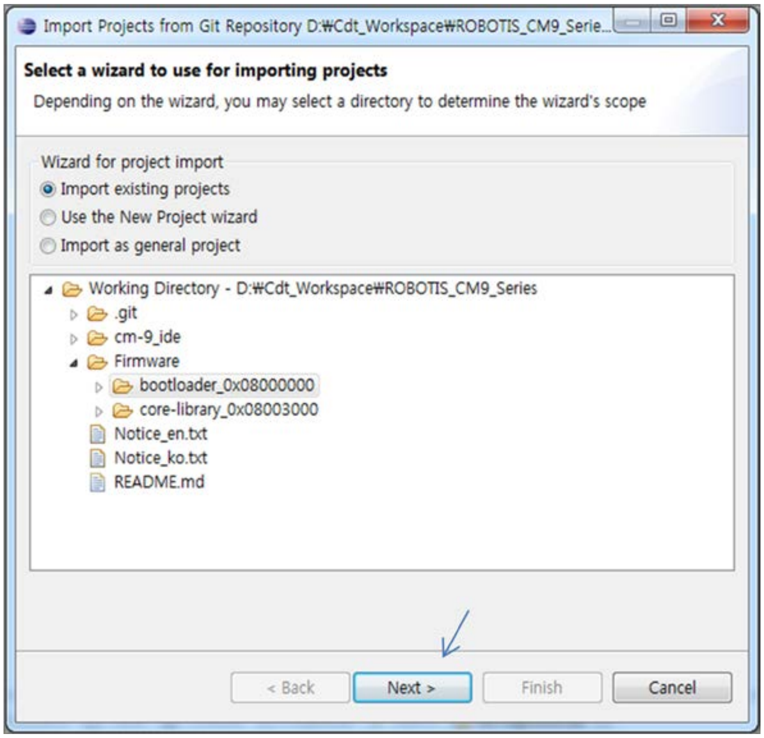
● Import existing projects

○ Use the New Project wizard

○ Import as general project

```
Working Directory - D:₩Cdt_Workspace₩ROBOTIS_CM9_Series
  .git
  cm-9_ide
  Firmware
    bootloader_0x08000000
    core-library_0x08003000
  Notice_en.txt
  Notice_ko.txt
  README.md
```

Click on **Finish** to include the project with Eclipse.



Import Projects from Git Repository D:₩Cdt_Workspace₩ROBOTIS_CM9_Serie...

Select a wizard to use for importing projects

Depending on the wizard, you may select a directory to determine the wizard's scope

Wizard for project import

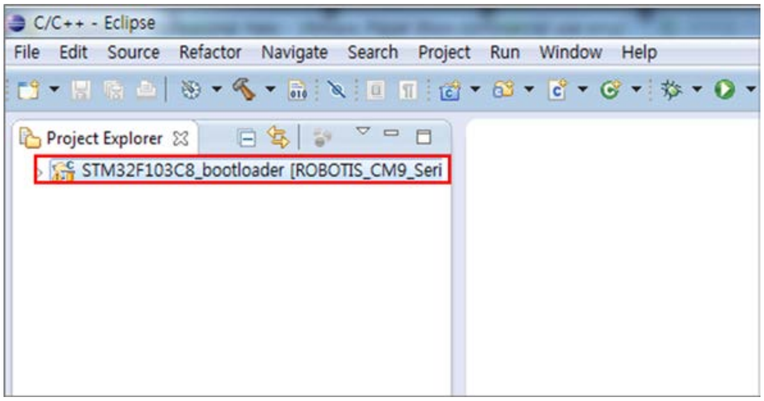
● Import existing projects

○ Use the New Project wizard

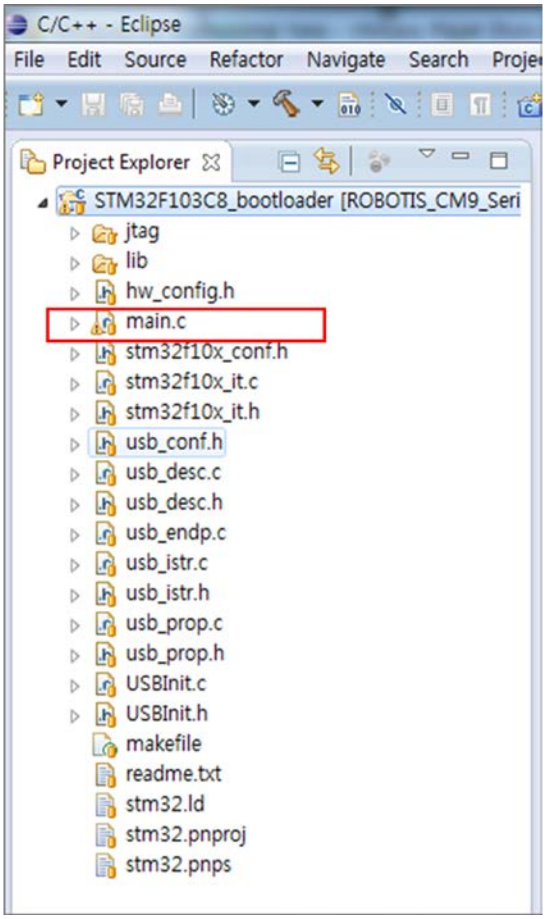
○ Import as general project

```
Working Directory - D:₩Cdt_Workspace₩ROBOTIS_CM9_Series
  .git
  cm-9_ide
  Firmware
    bootloader_0x08000000
    core-library_0x08003000
  Notice_en.txt
  Notice_ko.txt
  README.md
```

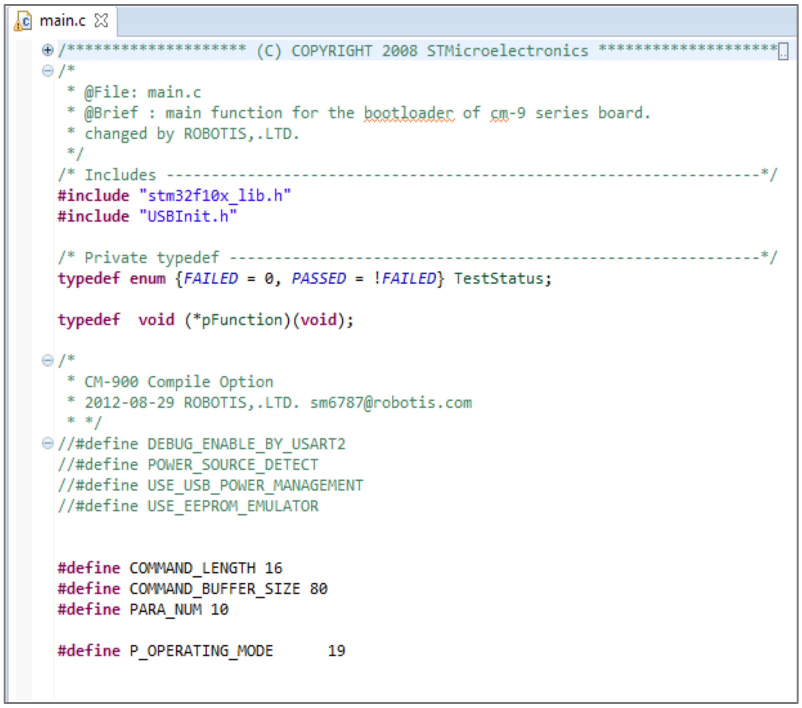
Project Explorer import complete.


C/C++ - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
STM32F103C8_bootloader [ROBOTIS_CM9_Seri

Expand the node and you will see the C files for the bootloader.



The contents of **main.c** shown below.

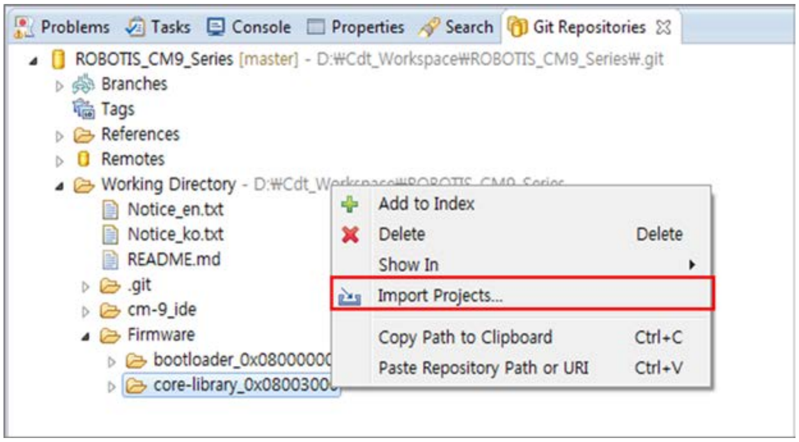


```
/******************** (C) COPYRIGHT 2008 STMicroelectronics ********************
/*
 * @File: main.c
 * @Brief : main function for the bootloader of cm-9 series board.
 * changed by ROBOTIS,.LTD.
 */
/* Includes ------------------------------------------------------------------*/
#include "stm32f10x_lib.h"
#include "USBInit.h"

/* Private typedef -----------------------------------------------------------*/
typedef enum {FAILED = 0, PASSED = !FAILED} TestStatus;

typedef  void (*pFunction)(void);

/*
 * CM-900 Compile Option
 * 2012-08-29 ROBOTIS,.LTD. sm6787@robotis.com
 * */
//#define DEBUG_ENABLE_BY_USART2
//#define POWER_SOURCE_DETECT
//#define USE_USB_POWER_MANAGEMENT
//#define USE_EEPROM_EMULATOR


#define COMMAND_LENGTH 16
#define COMMAND_BUFFER_SIZE 80
#define PARA_NUM 10

#define P_OPERATING_MODE      19
```

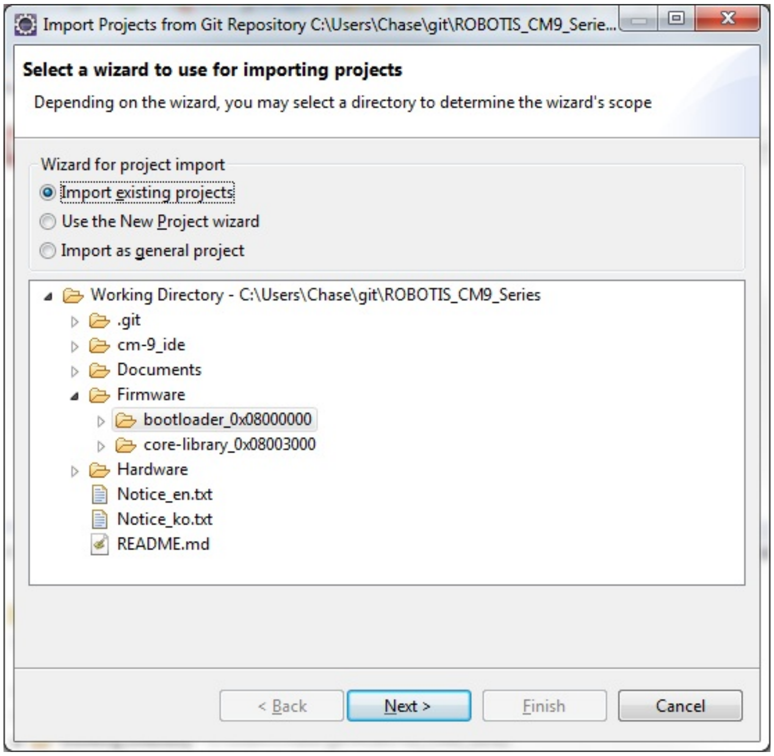
7.3.3 Importing core-library project

The core-library project for the bootloader is on Wiring's variant of C++. Import the core-library project.

From Eclipse's Git Repository select **core-library_0x08003000** and from the pop-up menu select **Import Projects…**



From **Select a wizard…** click on **next**.



Import Projects from Git Repository C:\Users\Chase\git\ROBOTIS_CM9_Serie...

Select a wizard to use for importing projects

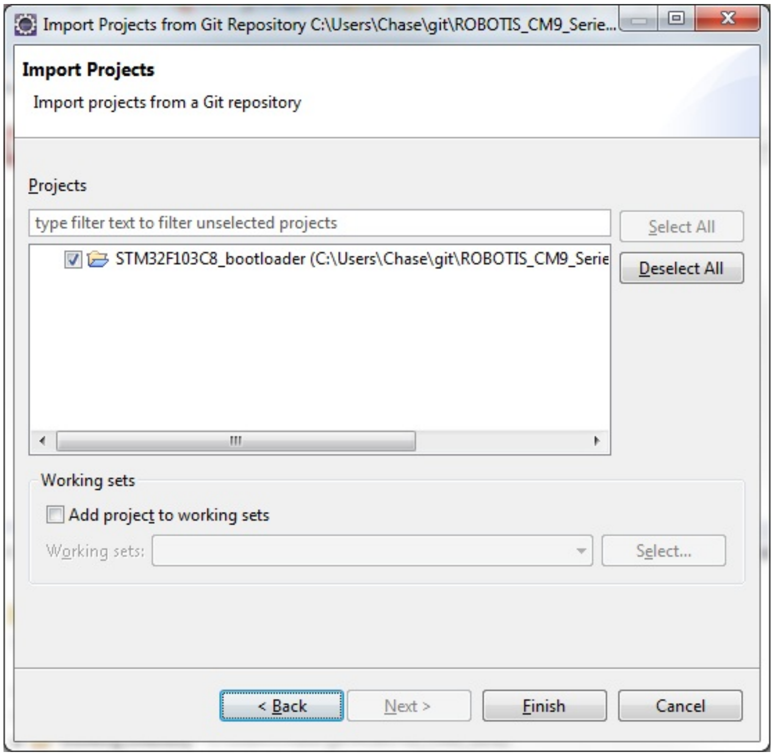
Depending on the wizard, you may select a directory to determine the wizard's scope

Wizard for project import

- ● Import existing projects
- ○ Use the New Project wizard
- ○ Import as general project

```
◢ Working Directory - C:\Users\Chase\git\ROBOTIS_CM9_Series
    ▷ .git
    ▷ cm-9_ide
    ▷ Documents
    ◢ Firmware
        ▷ bootloader_0x08000000
        ▷ core-library_0x08003000
    ▷ Hardware
      Notice_en.txt
      Notice_ko.txt
      README.md
```

From Import Projects make sure you see cm9-core-library then click on finish to finish importing.



Import Projects from Git Repository C:\Users\Chase\git\ROBOTIS_CM9_Serie...

Import Projects

Import projects from a Git repository

Projects

type filter text to filter unselected projects

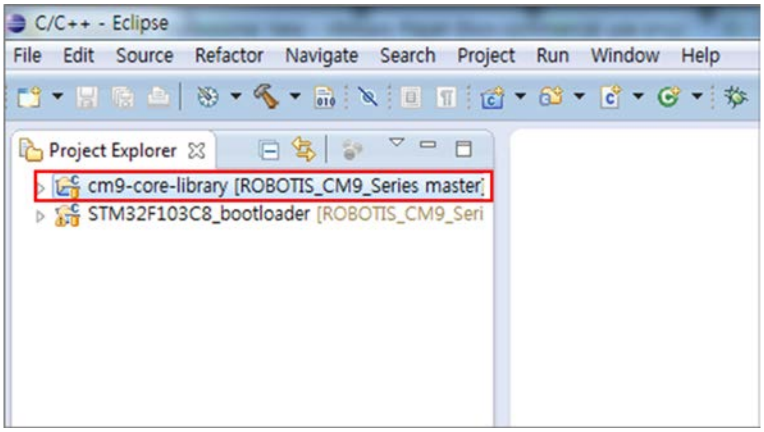
☑ STM32F103C8_bootloader (C:\Users\Chase\git\ROBOTIS_CM9_Serie

Working sets

☐ Add project to working sets

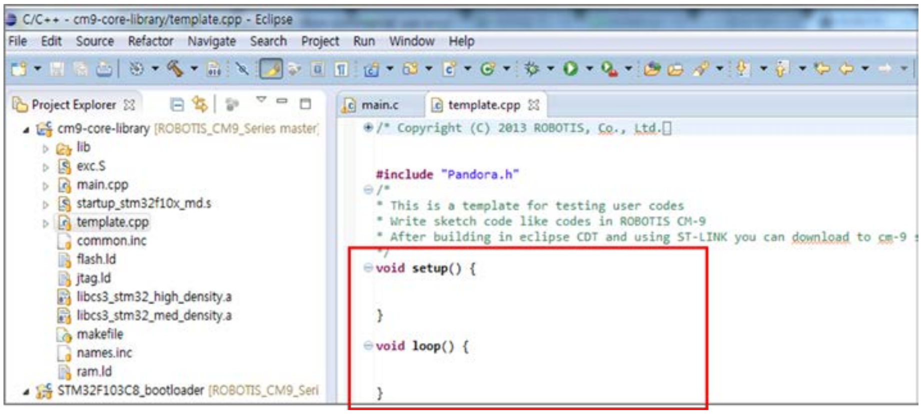
Working sets:

Both bootoader and cm9-core-library projects will appear.


C/C++ - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer

```
▷ cm9-core-library [ROBOTIS_CM9_Series master]
▷ STM32F103C8_bootloader [ROBOTIS_CM9_Seri
```

From the cm9-core-library project open template.cpp and you should notice this is the same from ROBOTIS CM-9's sketch code.


C/C++ - cm9-core-library/template.cpp - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
main.c
template.cpp

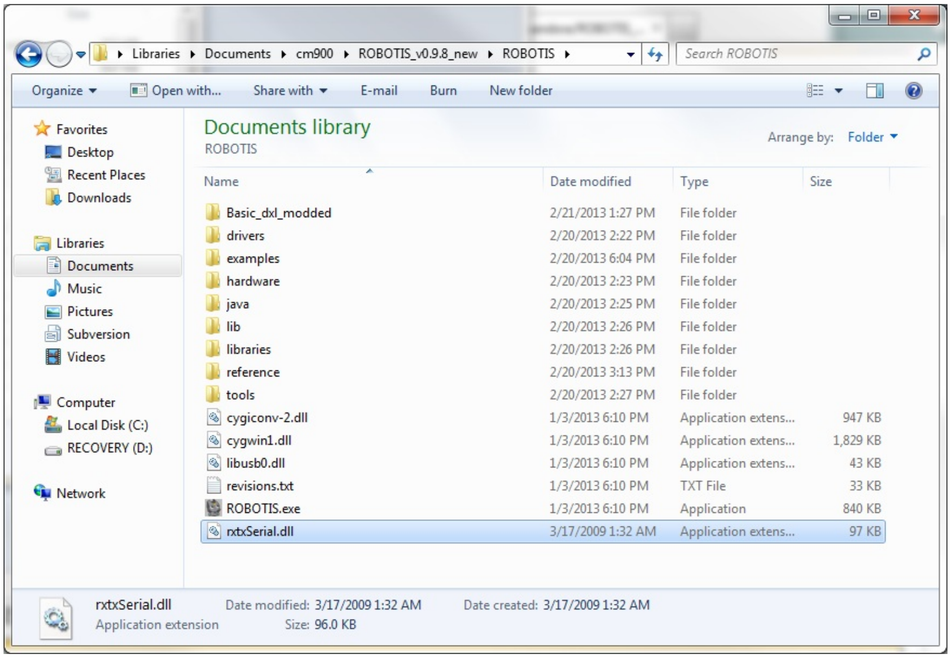
```
◢ cm9-core-library [ROBOTIS_CM9_Series master]
  ▷ lib
  ▷ exc.S
  ▷ main.cpp
  ▷ startup_stm32f10x_md.s
  ▷ template.cpp
    common.inc
    flash.ld
    jtag.ld
    libcs3_stm32_high_density.a
    libcs3_stm32_med_density.a
    makefile
    names.inc
    ram.ld
◢ STM32F103C8_bootloader [ROBOTIS_CM9_Seri
```

```
/* Copyright (C) 2013 ROBOTIS, Co., Ltd.□


#include "Pandora.h"
/*
 * This is a template for testing user codes
 * Write sketch code like codes in ROBOTIS CM-9
 * After building in eclipse CDT and using ST-LINK you can download to cm-9
 */
void setup() {


}

void loop() {


}
```

From template.cpp fill out setup() and loop() functions. This way you can edit robot development environment with Eclipse.

7.4 Register Code Sourcery G++ Lite environment variables

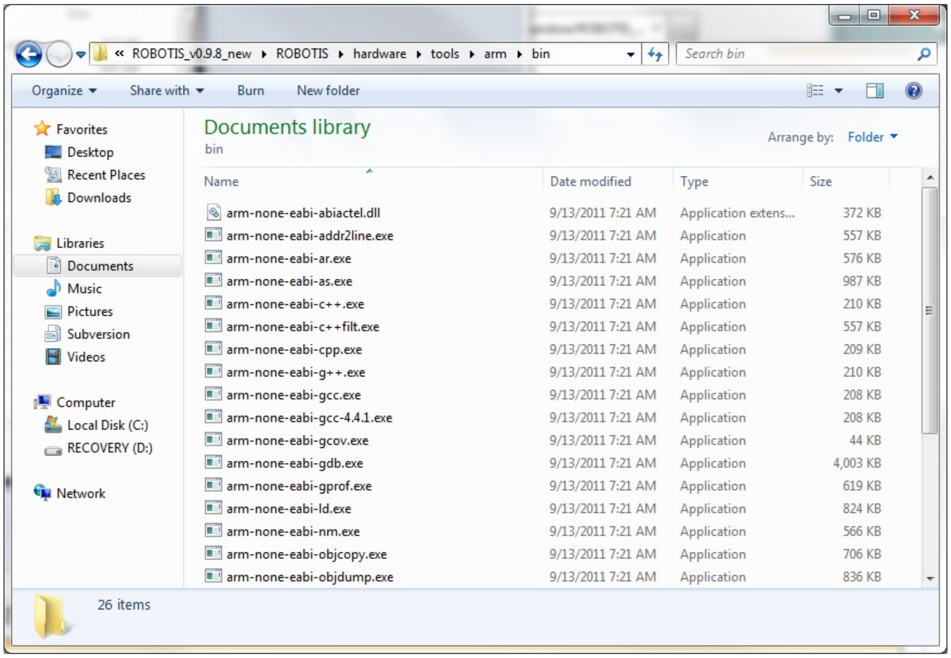
You can build the CM-900's bootloader and firmware with Code Sourcery G++ Lite tool chain. There is no need to download Code Sourcery G++ Lite separately. From the ROBOTIS CM9 folder follow the path below.



Name	Date modified	Type	Size
Basic_dxl_modded	2/21/2013 1:27 PM	File folder	
drivers	2/20/2013 2:22 PM	File folder	
examples	2/20/2013 6:04 PM	File folder	
hardware	2/20/2013 2:23 PM	File folder	
java	2/20/2013 2:25 PM	File folder	
lib	2/20/2013 2:26 PM	File folder	
libraries	2/20/2013 2:26 PM	File folder	
reference	2/20/2013 3:13 PM	File folder	
tools	2/20/2013 2:27 PM	File folder	
cygiconv-2.dll	1/3/2013 6:10 PM	Application extens...	947 KB
cygwin1.dll	1/3/2013 6:10 PM	Application extens...	1,829 KB
libusb0.dll	1/3/2013 6:10 PM	Application extens...	43 KB
revisions.txt	1/3/2013 6:10 PM	TXT File	33 KB
ROBOTIS.exe	1/3/2013 6:10 PM	Application	840 KB
rxtxSerial.dll	3/17/2009 1:32 AM	Application extens...	97 KB

rxtxSerial.dll Date modified: 3/17/2009 1:32 AM Date created: 3/17/2009 1:32 AM
Application extension Size: 96.0 KB

Go to **ROBOTIS\hardware\tools\arm\bin** directory look for arm-none-eabi-XXX.

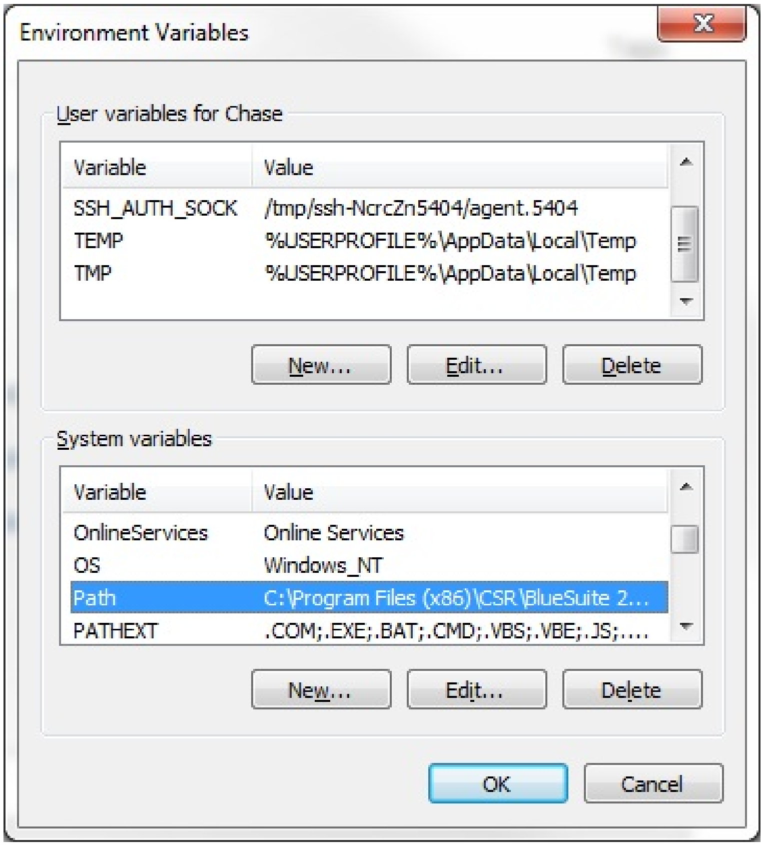


Name	Date modified	Type	Size
arm-none-eabi-abiactel.dll	9/13/2011 7:21 AM	Application extens...	372 KB
arm-none-eabi-addr2line.exe	9/13/2011 7:21 AM	Application	557 KB
arm-none-eabi-ar.exe	9/13/2011 7:21 AM	Application	576 KB
arm-none-eabi-as.exe	9/13/2011 7:21 AM	Application	987 KB
arm-none-eabi-c++.exe	9/13/2011 7:21 AM	Application	210 KB
arm-none-eabi-c++filt.exe	9/13/2011 7:21 AM	Application	557 KB
arm-none-eabi-cpp.exe	9/13/2011 7:21 AM	Application	209 KB
arm-none-eabi-g++.exe	9/13/2011 7:21 AM	Application	210 KB
arm-none-eabi-gcc.exe	9/13/2011 7:21 AM	Application	208 KB
arm-none-eabi-gcc-4.4.1.exe	9/13/2011 7:21 AM	Application	208 KB
arm-none-eabi-gcov.exe	9/13/2011 7:21 AM	Application	44 KB
arm-none-eabi-gdb.exe	9/13/2011 7:21 AM	Application	4,003 KB
arm-none-eabi-gprof.exe	9/13/2011 7:21 AM	Application	619 KB
arm-none-eabi-ld.exe	9/13/2011 7:21 AM	Application	824 KB
arm-none-eabi-nm.exe	9/13/2011 7:21 AM	Application	566 KB
arm-none-eabi-objcopy.exe	9/13/2011 7:21 AM	Application	706 KB
arm-none-eabi-objdump.exe	9/13/2011 7:21 AM	Application	836 KB

26 items

Register this path with Windows environment variables. Once registered do not delete this directory otherwise Eclipse will not be able to build.

Go to Control Panel -> System -> Advanced -> Environment variables.



Environment Variables

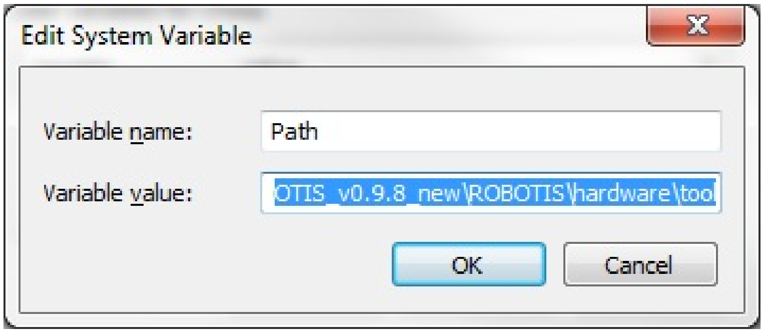
User variables for Chase

Variable	Value
SSH_AUTH_SOCK	/tmp/ssh-NcrcZn5404/agent.5404
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
OnlineServices	Online Services
OS	Windows_NT
Path	C:\Program Files (x86)\CSR\BlueSuite 2...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Enter the path for tool chain.

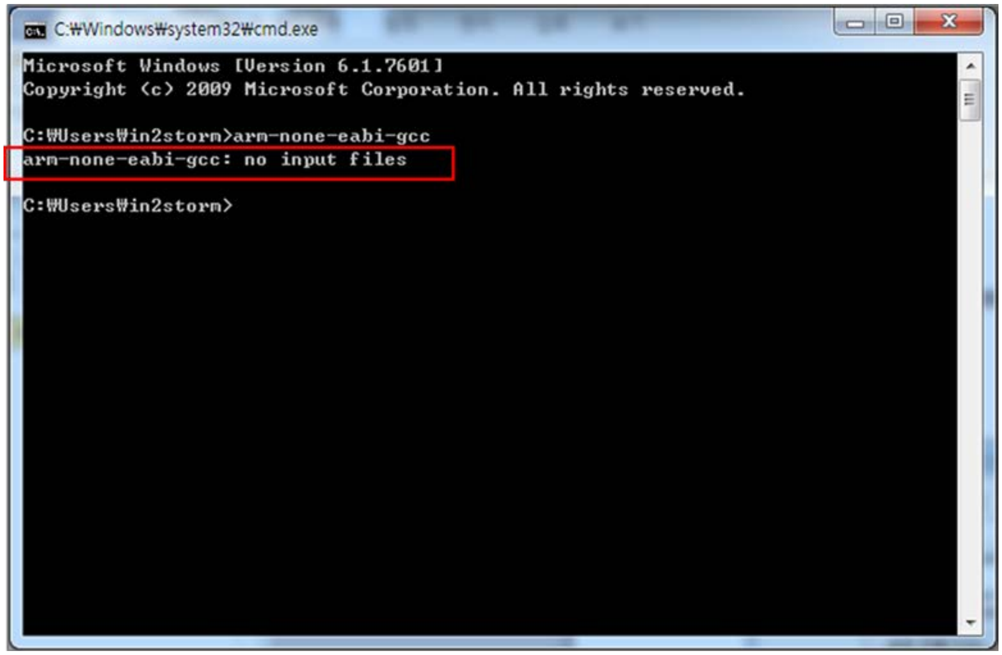


Edit System Variable

Variable name: Path

Variable value: OTIS_v0.9.8_new\ROBOTIS\hardware\tool

rom command prompt enter arm-none-eabi-gcc and you should see a response as shown below.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:₩Users₩in2storm>arm-none-eabi-gcc
arm-none-eabi-gcc: no input files

C:₩Users₩in2storm>
```

7.5 Bootloader build and download

By now you should know how to get ROBOTIS CM-9 Series source via GitHub and register Code Sourcery G++ Lite tool chain environment variables.

Use the ST-LINK to download the build.

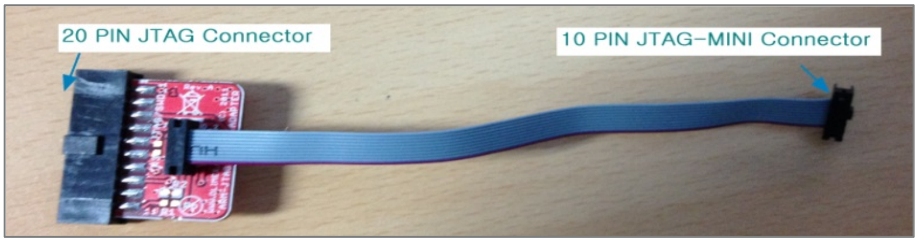
You may use OpenOCD, IAR, or Keil to download. However, the next section is dedicated for ST-LINK utility

From Apendix 1 bootloader download you are required to have the 20-pin to 10-pin converter**.**

The ST-LINK dongle and the converter cable are not included with the CM-900 and must be purchased separately.

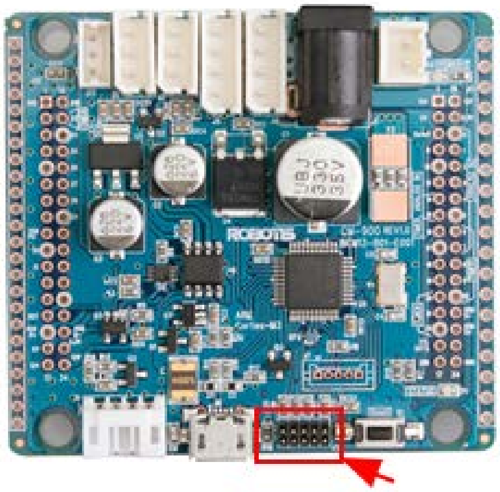
7.5.1 Connect the ST-LINK to the CM-900.

<ST-LINK/V2>

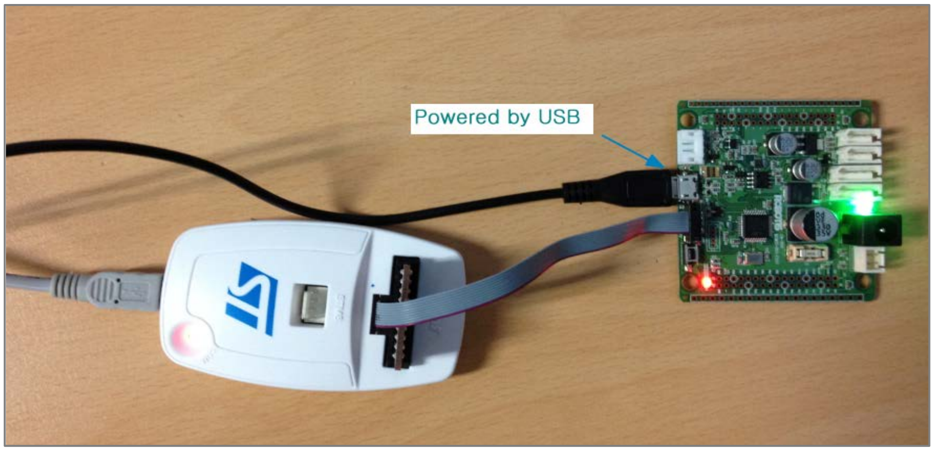


< 20 PIN to 10 PIN Converter >

Connect the 20-pin end to the ST-LINK/V2 and 10-pin end to the CM-900.

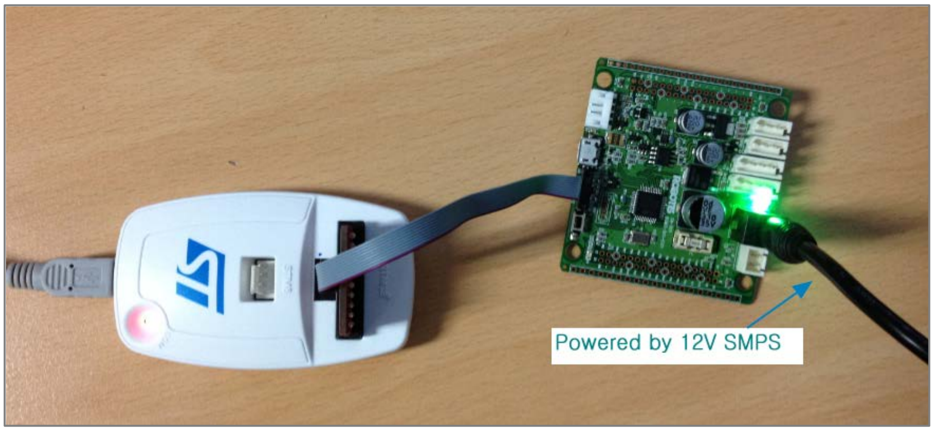


7.5.2 Supply power to the CM-900 separately.



Powered by USB

<Power input via USB>

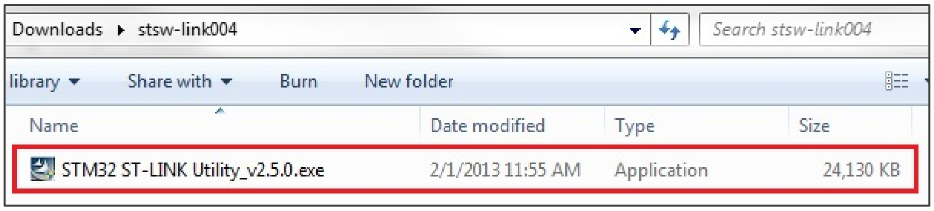


Powered by 12V SMPS

<Power input via SMPS>

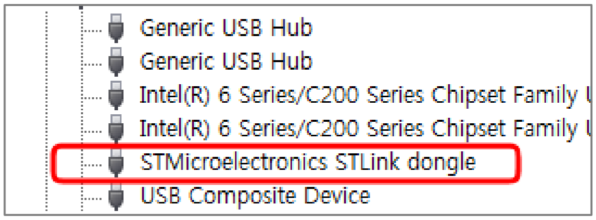
7.5.3 Download the ST-LINK Utility (device drivers included).
http://www.st.com/web/en/catalog/tools/PF258168

Decompress the file and run STM32 ST-LINK Utility_v2.x.x.exe; the ST-LINK driver is also automatically installed.



Name	Date modified	Type	Size
STM32 ST-LINK Utility_v2.5.0.exe	2/1/2013 11:55 AM	Application	24,130 KB

Connect the ST-LINK to the PC via USB..

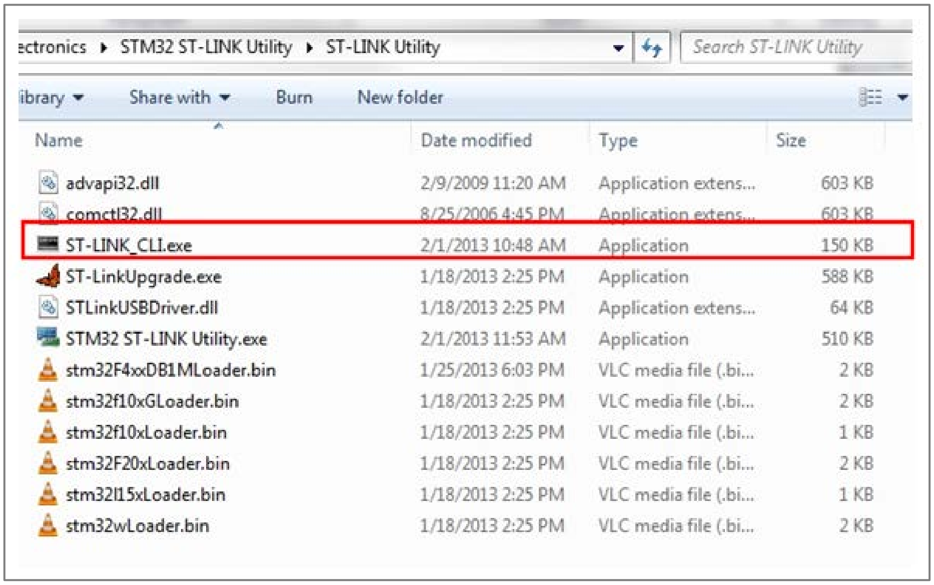


Generic USB Hub
Generic USB Hub
Intel(R) 6 Series/C200 Series Chipset Family U
Intel(R) 6 Series/C200 Series Chipset Family U
STMicroelectronics STLink dongle
USB Composite Device

To verify proper installation from Windows Device Manager go to the Universal Serial Bus controllers -> STMicroelectronics STLink dongle.

7.5.4 Verify ST-LINK_CLI.exe location.

C:\Program Files (x86)\STMicroelectronics\STM32 ST-LINK Utility\ST-LINK Utility

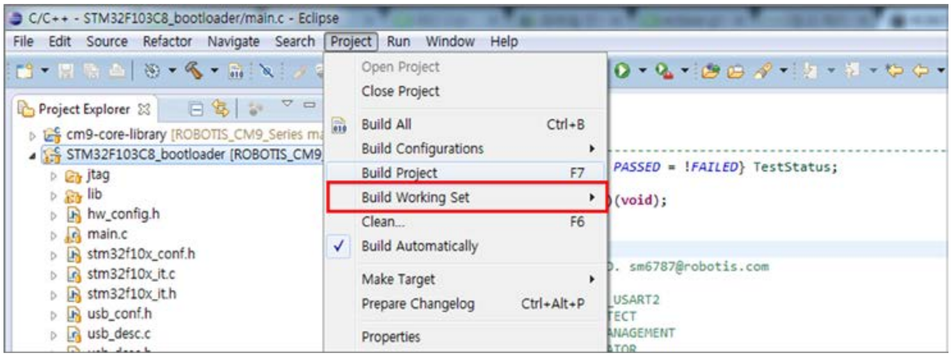


Name	Date modified	Type	Size
advapi32.dll	2/9/2009 11:20 AM	Application extens...	603 KB
comctl32.dll	8/25/2006 4:45 PM	Application extens...	603 KB
ST-LINK_CLI.exe	2/1/2013 10:48 AM	Application	150 KB
ST-LinkUpgrade.exe	1/18/2013 2:25 PM	Application	588 KB
STLinkUSBDriver.dll	1/18/2013 2:25 PM	Application extens...	64 KB
STM32 ST-LINK Utility.exe	2/1/2013 11:53 AM	Application	510 KB
stm32F4xxDB1MLoader.bin	1/25/2013 6:03 PM	VLC media file (.bi...	2 KB
stm32f10xGLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	2 KB
stm32f10xLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	1 KB
stm32F20xLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	2 KB
stm32l15xLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	1 KB
stm32wLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	2 KB

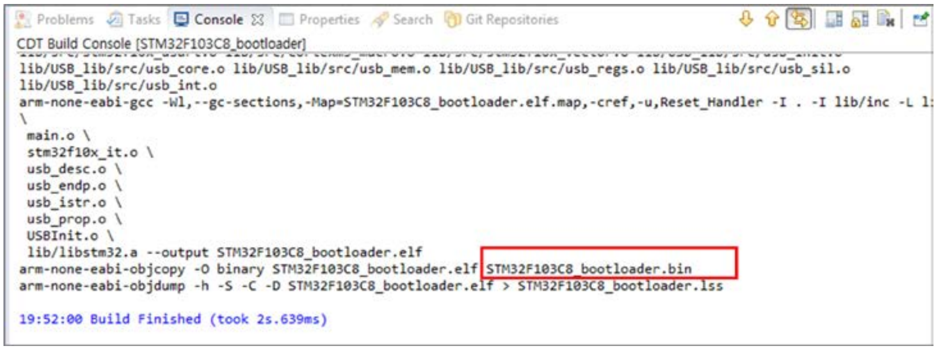
Register this path with Eclipse.

7.5.4 Building the bootloader.

Open any C file from STM32F103C8_bootloader project (say main.c). Proceed to build project.



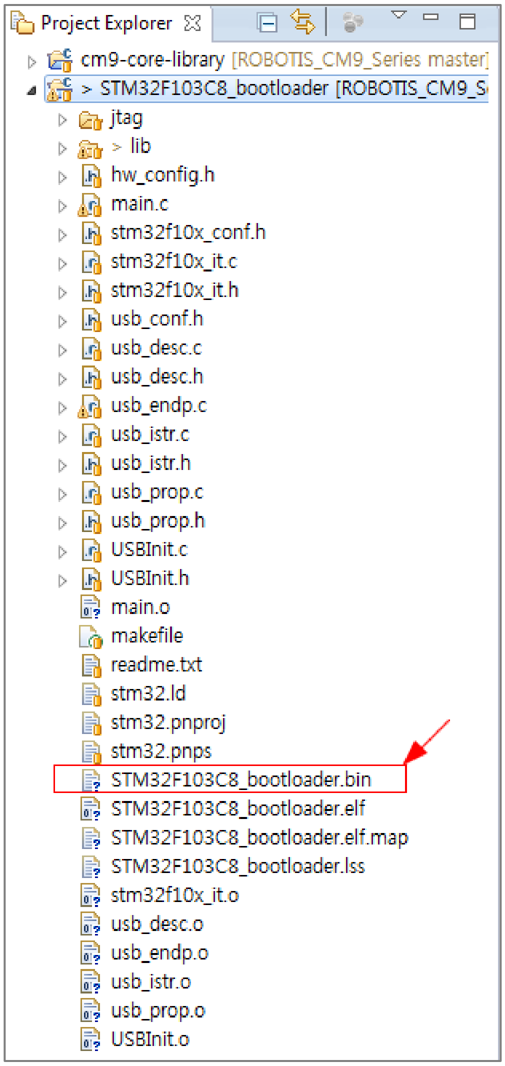
Go to Project -> Build Project to begin build. Upon successful build you should see a message fro m the console as shown below.



CDT Build Console [STM32F103C8_bootloader]

```
lib/USB_lib/src/usb_core.o lib/USB_lib/src/usb_mem.o lib/USB_lib/src/usb_regs.o lib/USB_lib/src/usb_sil.o
lib/USB_lib/src/usb_int.o
arm-none-eabi-gcc -Wl,--gc-sections,-Map=STM32F103C8_bootloader.elf.map,-cref,-u,Reset_Handler -I . -I lib/inc -L l
\
 main.o \
 stm32f10x_it.o \
 usb_desc.o \
 usb_endp.o \
 usb_istr.o \
 usb_prop.o \
 USBInit.o \
 lib/libstm32.a --output STM32F103C8_bootloader.elf
arm-none-eabi-objcopy -O binary STM32F103C8_bootloader.elf STM32F103C8_bootloader.bin
arm-none-eabi-objdump -h -S -C -D STM32F103C8_bootloader.elf > STM32F103C8_bootloader.lss

19:52:00 Build Finished (took 2s.639ms)
```

From the active project explorer (tree nodes) press the F5 key to refresh and **STM32F103C8_bootloader.bin** will appear.

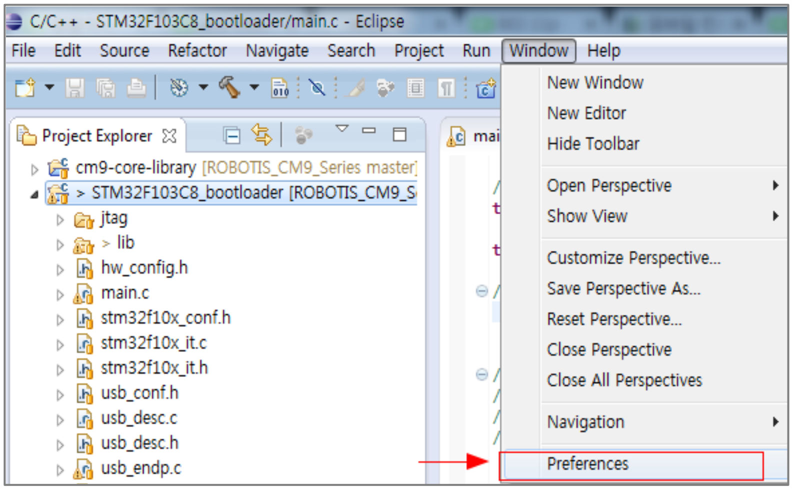


Project Explorer

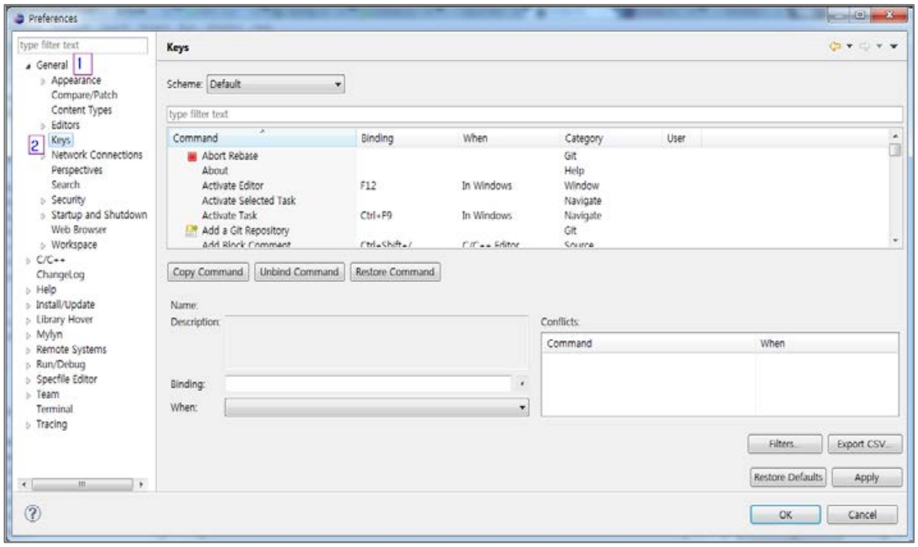
- cm9-core-library [ROBOTIS_CM9_Series master]
- > STM32F103C8_bootloader [ROBOTIS_CM9_S
  - jtag
  - > lib
  - hw_config.h
  - main.c
  - stm32f10x_conf.h
  - stm32f10x_it.c
  - stm32f10x_it.h
  - usb_conf.h
  - usb_desc.c
  - usb_desc.h
  - usb_endp.c
  - usb_istr.c
  - usb_istr.h
  - usb_prop.c
  - usb_prop.h
  - USBInit.c
  - USBInit.h
  - main.o
  - makefile
  - readme.txt
  - stm32.ld
  - stm32.pnproj
  - stm32.pnps
  - STM32F103C8_bootloader.bin
  - STM32F103C8_bootloader.elf
  - STM32F103C8_bootloader.elf.map
  - STM32F103C8_bootloader.lss
  - stm32f10x_it.o
  - usb_desc.o
  - usb_endp.o
  - usb_istr.o
  - usb_prop.o
  - USBInit.o

binary build complete

The following is about Eclipse shortcuts for easier usage.



Go to Window -> Preferences for general preferences.



Scheme: Default

type filter text

Command	Binding	When	Category	User
Abort Rebase			Git	
About			Help	
Activate Editor	F12	In Windows	Window	
Activate Selected Task			Navigate	
Activate Task	Ctrl+F9	In Windows	Navigate	
Add a Git Repository			Git	
Add Block Comment	Ctrl+Shift+/	C/C++ Editor	Source	

Name:

Description:

Binding:

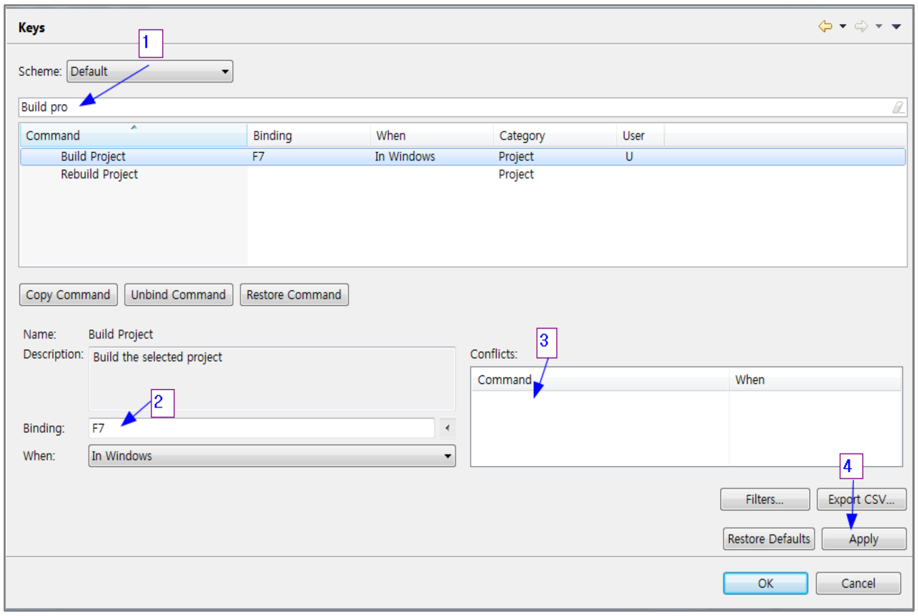
When:

Conflicts:

Command	When

Go to Preferences -> General -> Key.

Enter build project on the search line and the command for build project appears. Click on Binding and press F7 to designate the items. Go to Conflicts to verify of any possible conflicts if none click on Apply.



Scheme: Default

Build pro

Command	Binding	When	Category	User
Build Project	F7	In Windows	Project	U
Rebuild Project			Project	

Name: Build Project

Description: Build the selected project

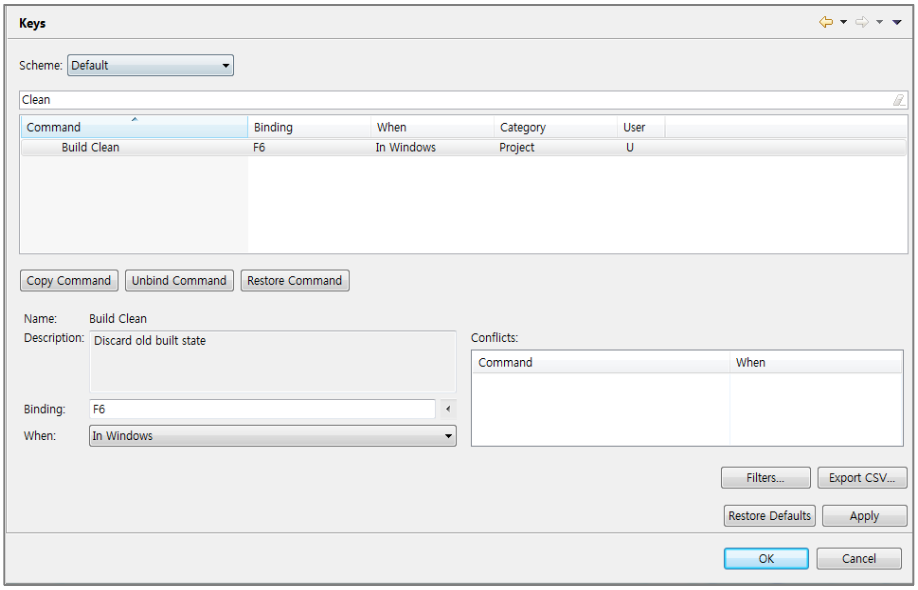
Binding: F7

When: In Windows

Conflicts:

Command	When

Likewise by entering Clean Build Clean items will pop up; press the F6 key for the binding shortcut.



Keys

Scheme: Default

Clean

Command	Binding	When	Category	User
Build Clean	F6	In Windows	Project	U

Name: Build Clean

Description: Discard old built state

Binding: F6

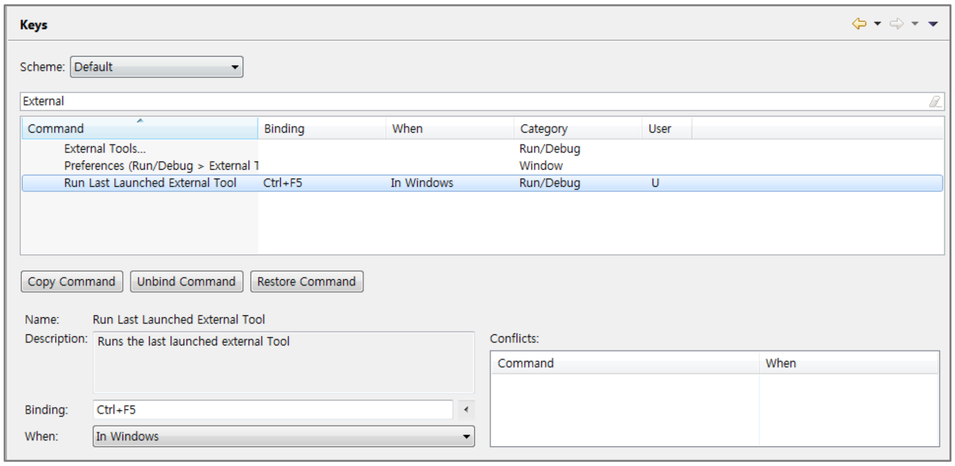
When: In Windows

Conflicts:

Command	When

Afterwards register shortcuts for external tools.

Enter External on search and Run Last Launched External Tool pops up; press Ctrl+F5 for shortcut.



Keys

Scheme: Default

External

Command	Binding	When	Category	User
External Tools...			Run/Debug	
Preferences (Run/Debug > External T			Window	
Run Last Launched External Tool	Ctrl+F5	In Windows	Run/Debug	U

Name: Run Last Launched External Tool

Description: Runs the last launched external Tool

Binding: Ctrl+F5

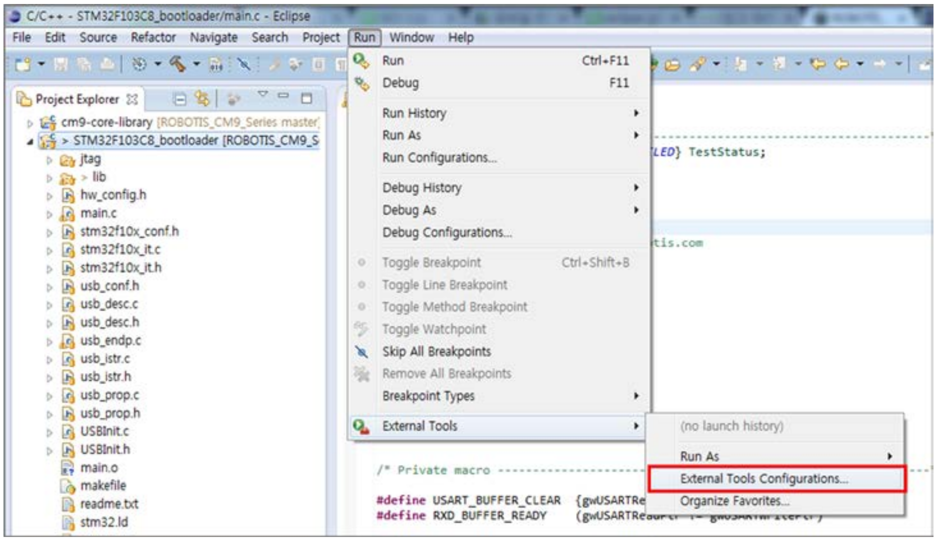
When: In Windows

Conflicts:

Command	When

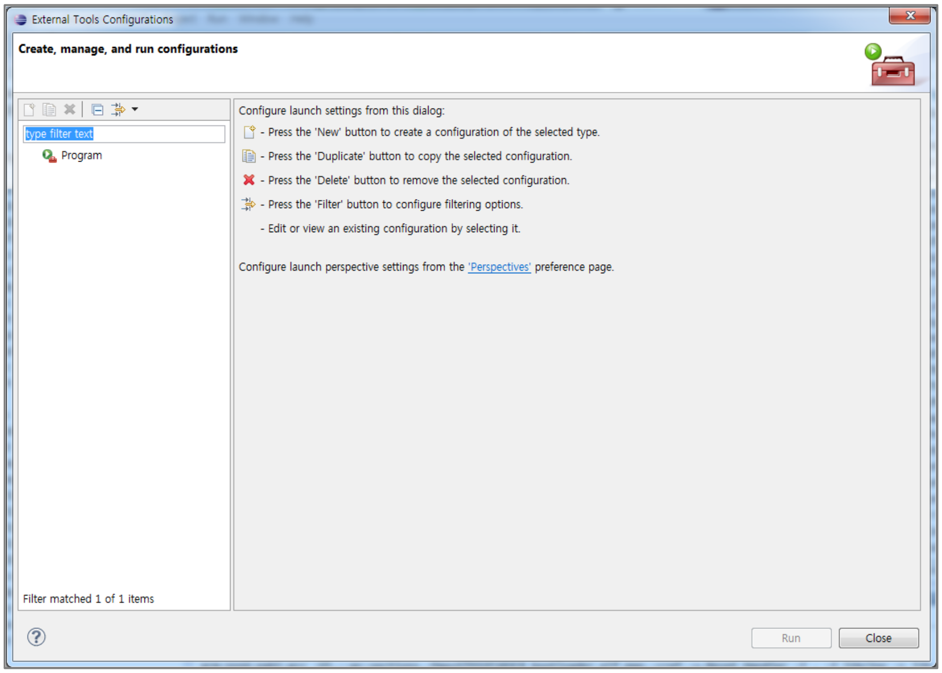
7.5.5 er External Tools with ST-LINK_CLI.exe and download.

Go to Run -> External Tools -> External Tools Configurations…



```
LED} TestStatus;
tis.com
/* Private macro -------------------
#define USART_BUFFER_CLEAR   {gwUSARTRe
#define RXD_BUFFER_READY     (gwUSARTRe
```

You will get a window as shown below.



Create, manage, and run configurations

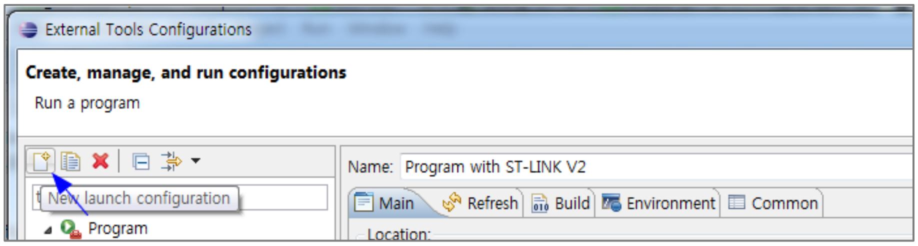
Configure launch settings from this dialog:

- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the 'Perspectives' preference page.

Filter matched 1 of 1 items

Click on New launch configuration icon



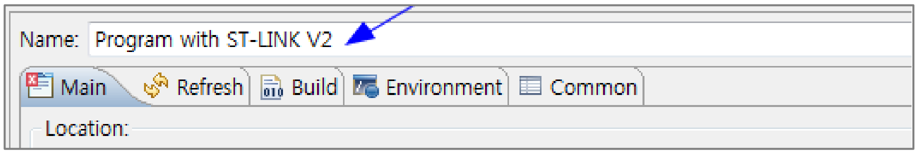
External Tools Configurations

Create, manage, and run configurations

Run a program

Name: Program with ST-LINK V2

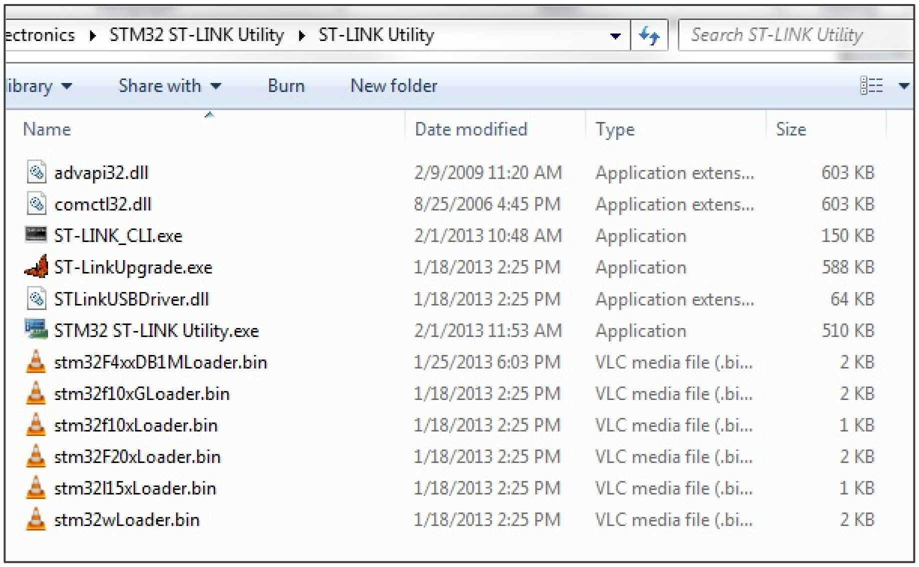
Name : Program with ST-LINK V2.



Name: Program with ST-LINK V2

From the Main tab: click on Browse File System select the path for ST-LINK_CLI.exe.

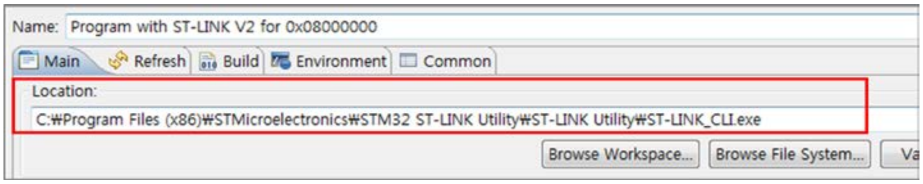
C:\Program Files (x86)\STMicroelectronics\STM32 ST-LINK Utility\ST-LINK Utility\



ectronics ▸ STM32 ST-LINK Utility ▸ ST-LINK Utility

Name	Date modified	Type	Size
advapi32.dll	2/9/2009 11:20 AM	Application extens...	603 KB
comctl32.dll	8/25/2006 4:45 PM	Application extens...	603 KB
ST-LINK_CLI.exe	2/1/2013 10:48 AM	Application	150 KB
ST-LinkUpgrade.exe	1/18/2013 2:25 PM	Application	588 KB
STLinkUSBDriver.dll	1/18/2013 2:25 PM	Application extens...	64 KB
STM32 ST-LINK Utility.exe	2/1/2013 11:53 AM	Application	510 KB
stm32F4xxDB1MLoader.bin	1/25/2013 6:03 PM	VLC media file (.bi...	2 KB
stm32f10xGLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	2 KB
stm32f10xLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	1 KB
stm32F20xLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	2 KB
stm32l15xLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	1 KB
stm32wLoader.bin	1/18/2013 2:25 PM	VLC media file (.bi...	2 KB

Select ST-LINK_CLI.exe.

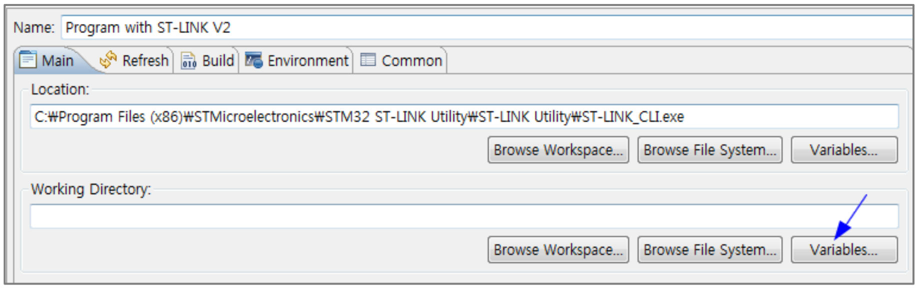


Name: Program with ST-LINK V2 for 0x08000000

Location:

C:₩Program Files (x86)₩STMicroelectronics₩STM32 ST-LINK Utility₩ST-LINK Utility₩ST-LINK_CLI.exe

Assign a working directory. Use Eclipse's common environment variables. Click on Variables…



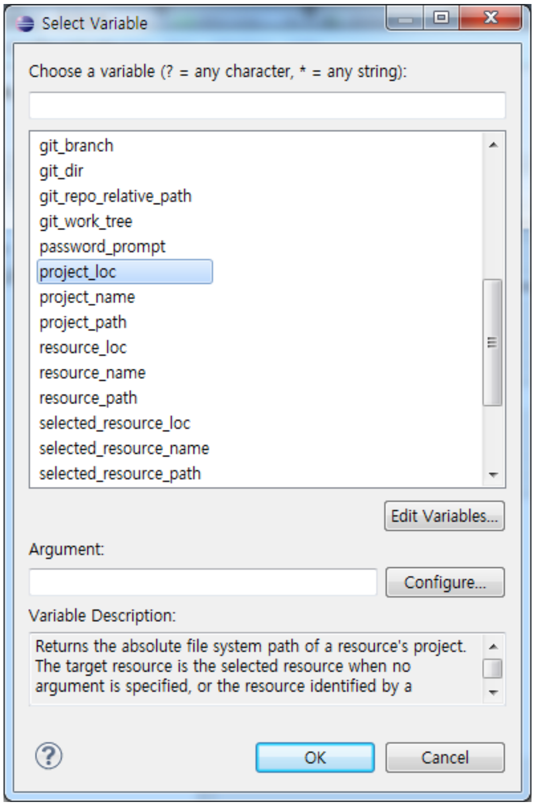
Name: Program with ST-LINK V2

Location:

C:₩Program Files (x86)₩STMicroelectronics₩STM32 ST-LINK Utility₩ST-LINK Utility₩ST-LINK_CLI.exe

Working Directory:

On the current project select ${project_loc}.



Choose a variable (? = any character, * = any string):

git_branch
git_dir
git_repo_relative_path
git_work_tree
password_prompt
project_loc
project_name
project_path
resource_loc
resource_name
resource_path
selected_resource_loc
selected_resource_name
selected_resource_path

Argument:

Variable Description:

Returns the absolute file system path of a resource's project. The target resource is the selected resource when no argument is specified, or the resource identified by a

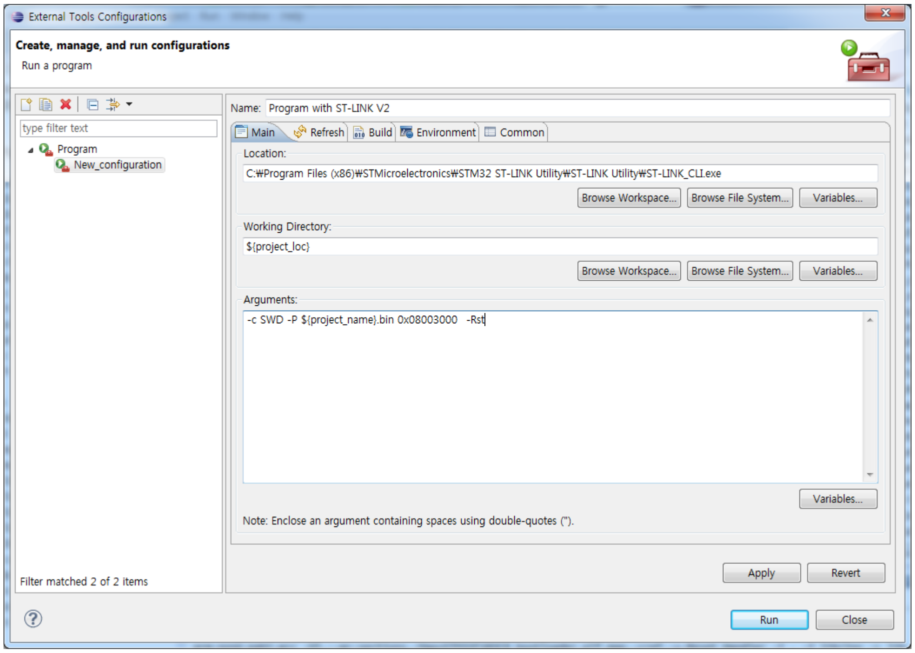
Register with Argument.

ST-LINK_CLI.exe is under Arguments.

Enter -c SWD -P ${project_name}.bin 0x08000000

-Rst. Keep in mind with 0x08000000 location

Click on Apply then click on Run to download.



External Tools Configurations
Create, manage, and run configurations
Run a program
type filter text
Program
New_configuration
Filter matched 2 of 2 items
Main
Refresh
Build
Environment
Common
Browse Workspace...
Browse File System...
Variables...
Note: Enclose an argument containing spaces using double-quotes (").
Apply
Revert
Run
Close

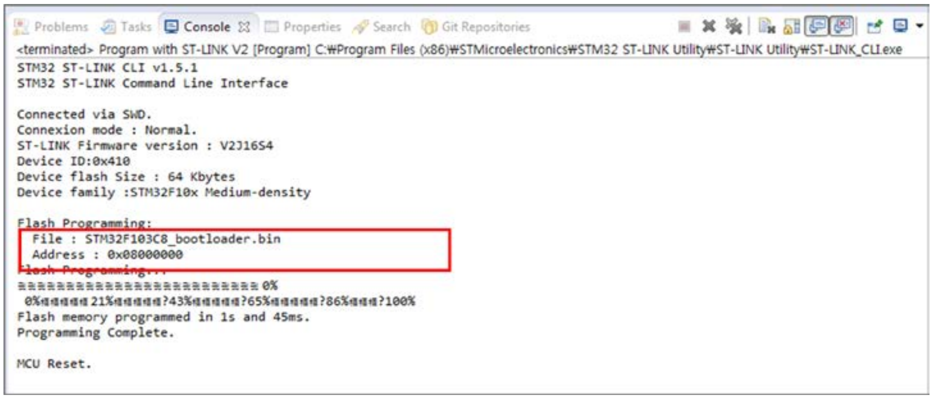
Name: Program with ST-LINK V2

Location:
C:₩Program Files (x86)₩STMicroelectronics₩STM32 ST-LINK Utility₩ST-LINK Utility₩ST-LINK_CLI.exe

Working Directory:
${project_loc}

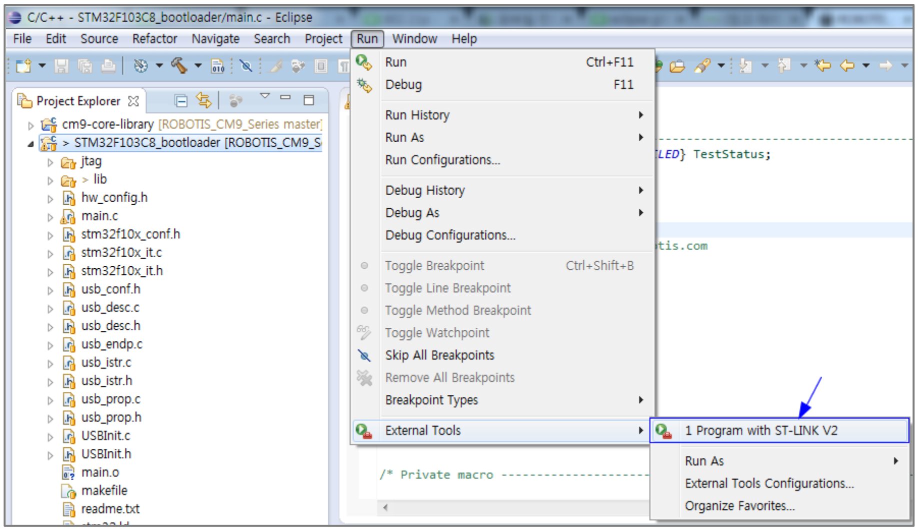
Arguments:
```
-c SWD -P ${project_name}.bin 0x08003000  -Rst
```

The console should output a message as shown below for successful downloa.

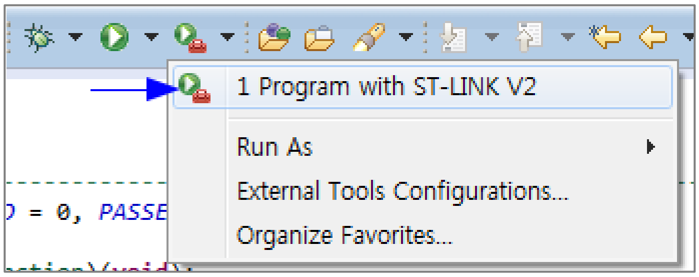


```
<terminated> Program with ST-LINK V2 [Program] C:₩Program Files (x86)₩STMicroelectronics₩STM32 ST-LINK Utility₩ST-LINK Utility₩ST-LINK_CLI.exe
STM32 ST-LINK CLI v1.5.1
STM32 ST-LINK Command Line Interface

Connected via SWD.
Connexion mode : Normal.
ST-LINK Firmware version : V2J16S4
Device ID:0x410
Device flash Size : 64 Kbytes
Device family :STM32F10x Medium-density

Flash Programming:
  File : STM32F103C8_bootloader.bin
  Address : 0x08000000
Flash Programming...
≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡ 0%
 0%▮▮▮▮▮ 21%▮▮▮▮▮?43%▮▮▮▮▮?65%▮▮▮▮▮?86%▮▮▮▮?100%
Flash memory programmed in 1s and 45ms.
Programming Complete.

MCU Reset.
```

Run ST-LINK_CLI.exe from the registered External Tool.



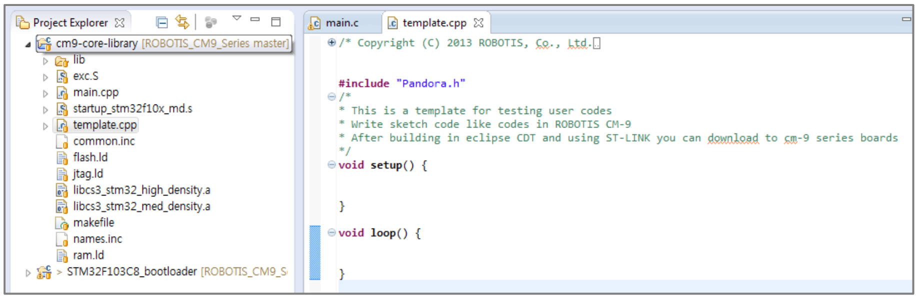
Or press Ctrl + F5.



7.6 core-library project build and download

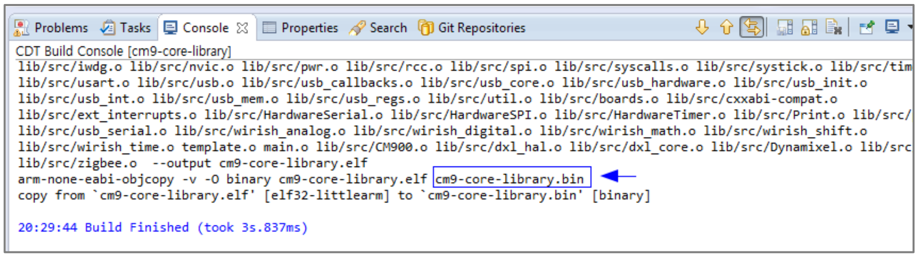
7.6.1 Build the cm9-core-library project

Open template.cpp. Or any other file from the cm9-core-library.


Project Explorer
cm9-core-library [ROBOTIS_CM9_Series master]
lib
exc.S
main.cpp
startup_stm32f10x_md.s
template.cpp
common.inc
flash.ld
jtag.ld
libcs3_stm32_high_density.a
libcs3_stm32_med_density.a
makefile
names.inc
ram.ld
> STM32F103C8_bootloader [ROBOTIS_CM9_S
main.c
template.cpp

```
/* Copyright (C) 2013 ROBOTIS, Co., Ltd.□

#include "Pandora.h"
/*
 * This is a template for testing user codes
 * Write sketch code like codes in ROBOTIS CM-9
 * After building in eclipse CDT and using ST-LINK you can download to cm-9 series boards
 */
void setup() {

}

void loop() {

}
```

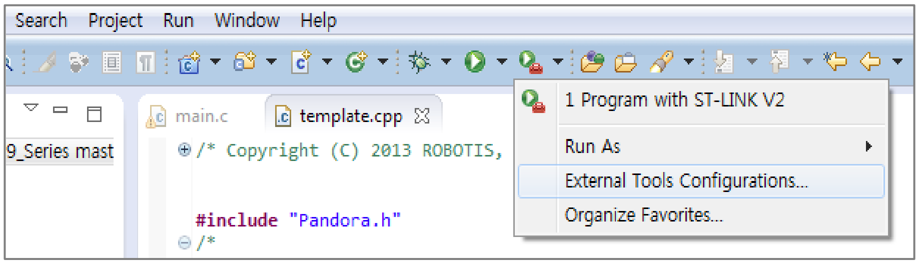
Press F7 for the shortcut or go to Project -> Project Build.



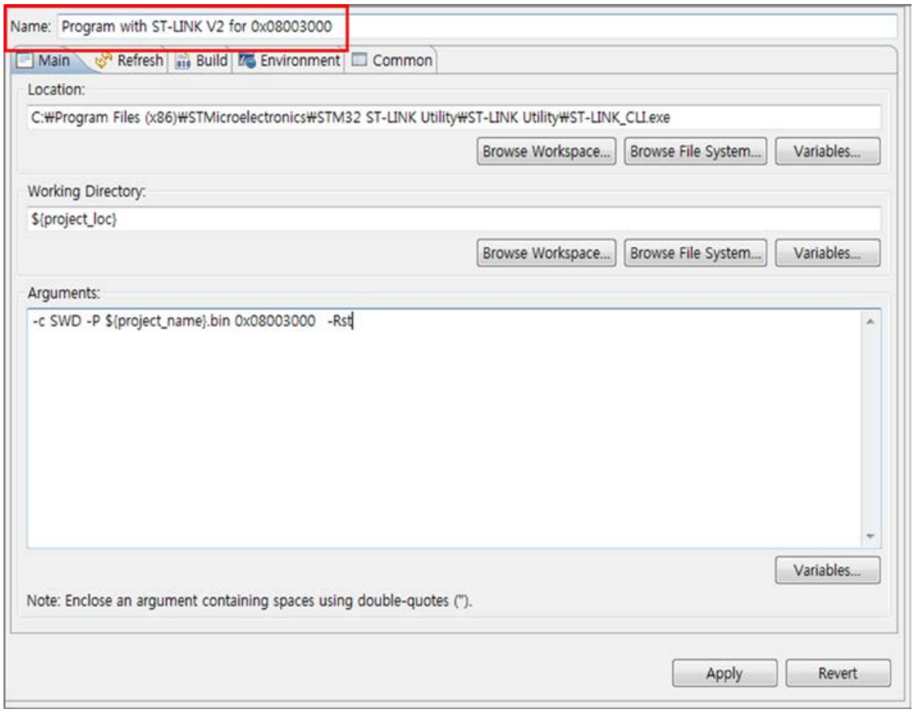
```
CDT Build Console [cm9-core-library]
lib/src/iwdg.o lib/src/nvic.o lib/src/pwr.o lib/src/rcc.o lib/src/spi.o lib/src/syscalls.o lib/src/systick.o lib/src/tim
lib/src/usart.o lib/src/usb.o lib/src/usb_callbacks.o lib/src/usb_core.o lib/src/usb_hardware.o lib/src/usb_init.o
lib/src/usb_int.o lib/src/usb_mem.o lib/src/usb_regs.o lib/src/util.o lib/src/boards.o lib/src/cxxabi-compat.o
lib/src/ext_interrupts.o lib/src/HardwareSerial.o lib/src/HardwareSPI.o lib/src/HardwareTimer.o lib/src/Print.o lib/src/
lib/src/usb_serial.o lib/src/wirish_analog.o lib/src/wirish_digital.o lib/src/wirish_math.o lib/src/wirish_shift.o
lib/src/wirish_time.o template.o main.o lib/src/CM900.o lib/src/dxl_hal.o lib/src/dxl_core.o lib/src/Dynamixel.o lib/src
lib/src/zigbee.o  --output cm9-core-library.elf
arm-none-eabi-objcopy -v -O binary cm9-core-library.elf cm9-core-library.bin
copy from `cm9-core-library.elf' [elf32-littlearm] to `cm9-core-library.bin' [binary]

20:29:44 Build Finished (took 3s.837ms)
```

7.6.2 External Tool Configuration for cm9-core-library

Configure external tools before downloading cm9-core-library.bin. this is due to the different addresses for bootloader and user-created firmware. The CM-900's cm9-core-library begins at the flash memory's 0x08003000 address.



Create new external tools (as shown below).



Name: Program with ST-LINK V2 for 0x08003000

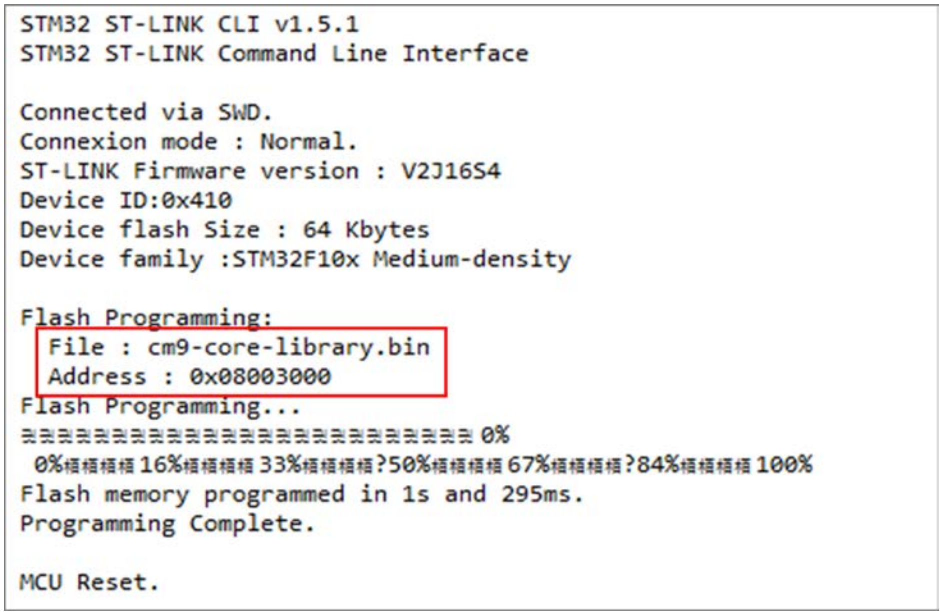
Location:
C:₩Program Files (x86)₩STMicroelectronics₩STM32 ST-LINK Utility₩ST-LINK Utility₩ST-LINK_CLI.exe

Working Directory:
${project_loc}

Arguments:
-c SWD -P ${project_name}.bin 0x08003000 -Rst

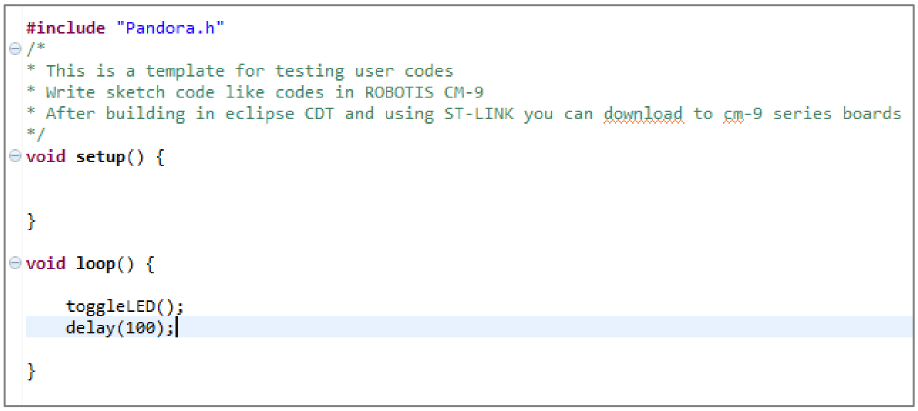
Note: Enclose an argument containing spaces using double-quotes (").

Upon successful download the MCU resets.

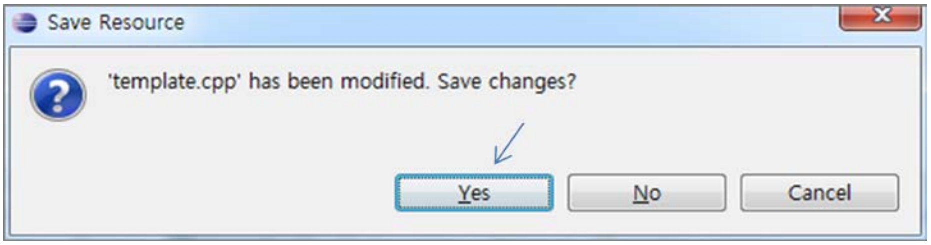


```
STM32 ST-LINK CLI v1.5.1
STM32 ST-LINK Command Line Interface

Connected via SWD.
Connexion mode : Normal.
ST-LINK Firmware version : V2J16S4
Device ID:0x410
Device flash Size : 64 Kbytes
Device family :STM32F10x Medium-density

Flash Programming:
  File : cm9-core-library.bin
  Address : 0x08003000
Flash Programming...
궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂궂 0%
 0%植植植植 16%植植植植 33%植植植植?50%植植植植 67%植植植植?84%植植植植 100%
Flash memory programmed in 1s and 295ms.
Programming Complete.

MCU Reset.
```

In Template.cpp press Ctrl+F5 for shortcut and the CM-900's의 Status LED blinks.

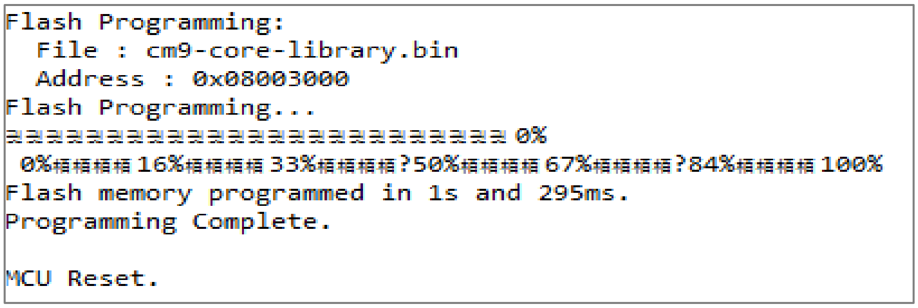


```
#include "Pandora.h"
/*
 * This is a template for testing user codes
 * Write sketch code like codes in ROBOTIS CM-9
 * After building in eclipse CDT and using ST-LINK you can download to cm-9 series boards
 */
void setup() {


}

void loop() {

    toggleLED();
    delay(100);

}
```



Save Resource

'template.cpp' has been modified. Save changes?

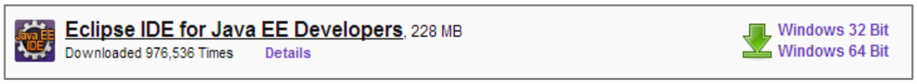


```
Flash Programming:
  File : cm9-core-library.bin
  Address : 0x08003000
Flash Programming...
궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡궡 0%
 0%榴榴榴榴 16%榴榴榴榴 33%榴榴榴榴?50%榴榴榴榴 67%榴榴榴榴?84%榴榴榴榴 100%
Flash memory programmed in 1s and 295ms.
Programming Complete.

MCU Reset.
```

7.7 cm-9_ide source build

7.7.1 Download Eclipse's Java package.

http://www.eclipse.org/downloads/



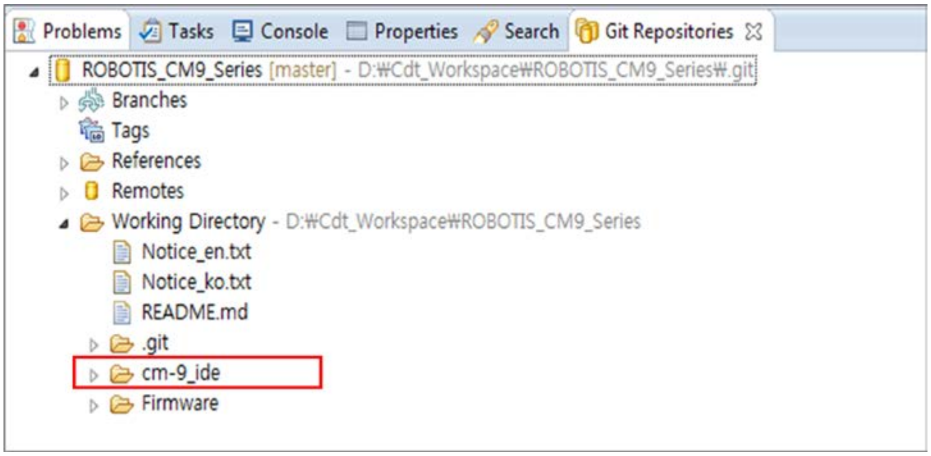
Eclipse IDE for Java EE Developers, 228 MB

Downloaded 976,536 Times Details

Windows 32 Bit
Windows 64 Bit

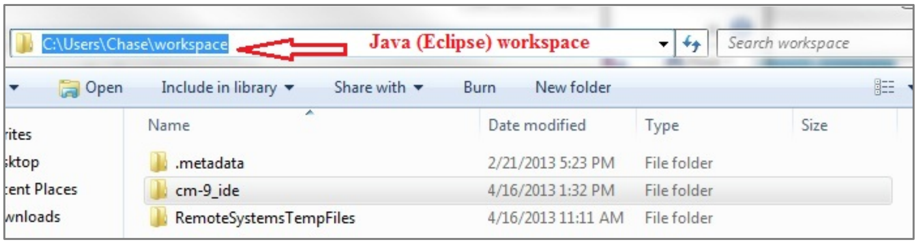
download either 32-bit or 64-bit version according to the version of your computer.

7.7.2 from the workspace import cm-9_ide directory.

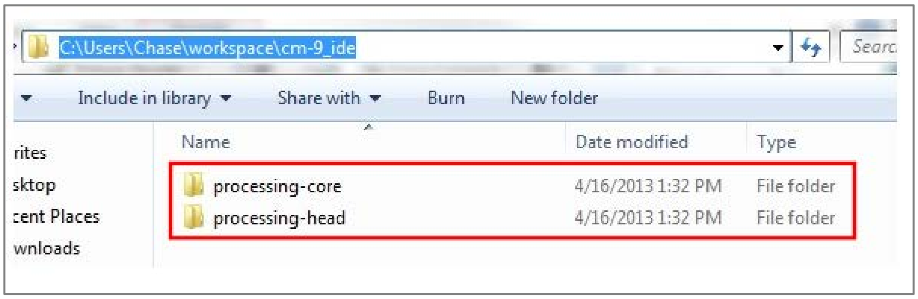


The path, for example, C:\Users\Chase\git\ROBOTIS_CM9_Series\ is the starting point. However, importing cm-9_ide folder from another path is OK.

Copy cm-9_ide folder from your git repository to the Java (Eclipse) workspace

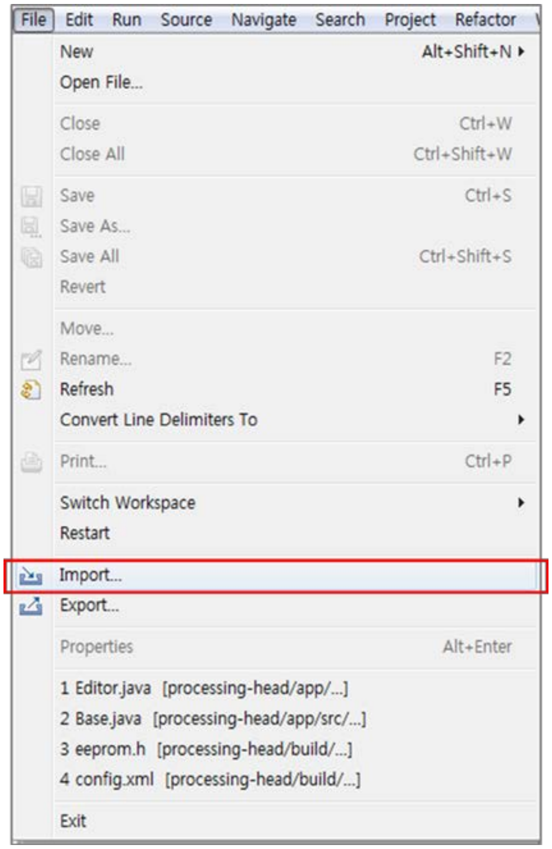


Name	Date modified	Type	Size
.metadata	2/21/2013 5:23 PM	File folder	
cm-9_ide	4/16/2013 1:32 PM	File folder	
RemoteSystemsTempFiles	4/16/2013 11:11 AM	File folder	

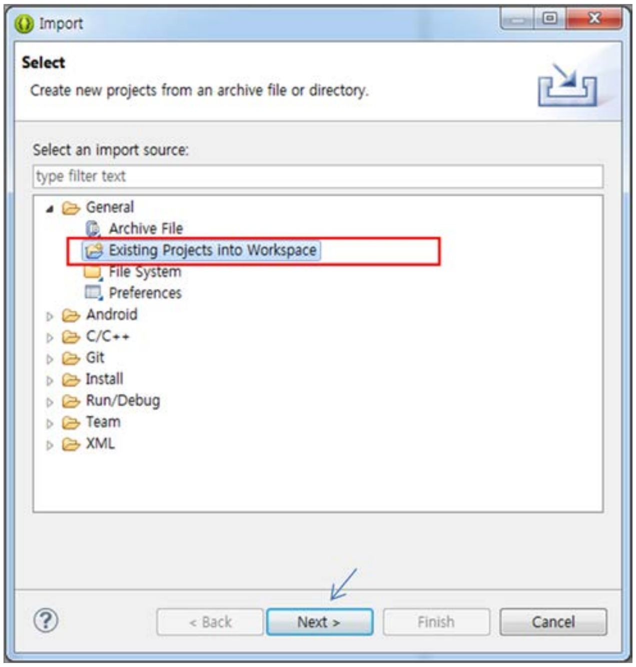


Name	Date modified	Type
processing-core	4/16/2013 1:32 PM	File folder
processing-head	4/16/2013 1:32 PM	File folder

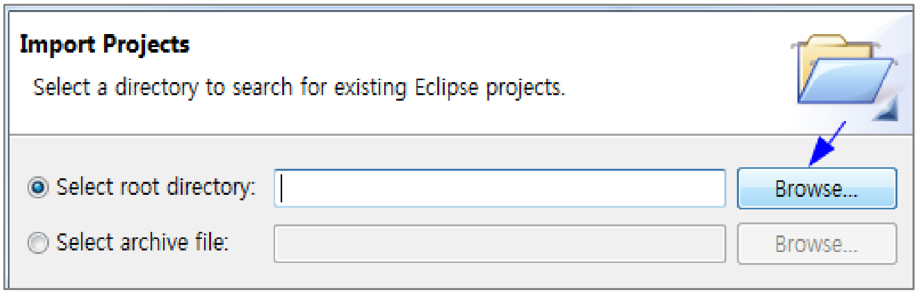
Go to File -> Import.



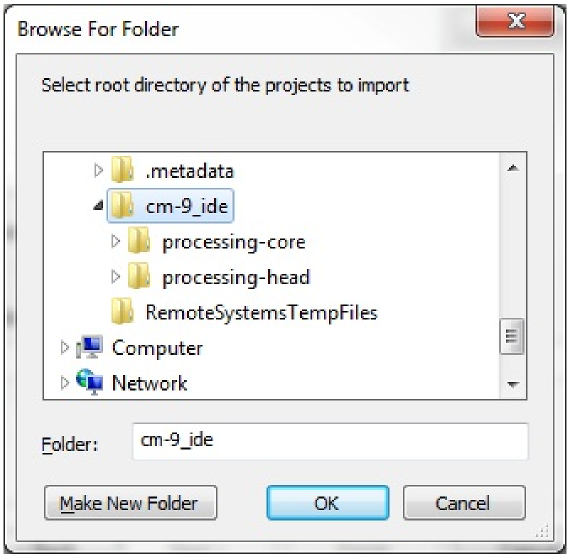
GO to General -> Existing Projects into Workspace; then click on Next



Browse… click on browser and look for the copied folder in the Java workspace.



The copied cm-9_ide folder is assigned.

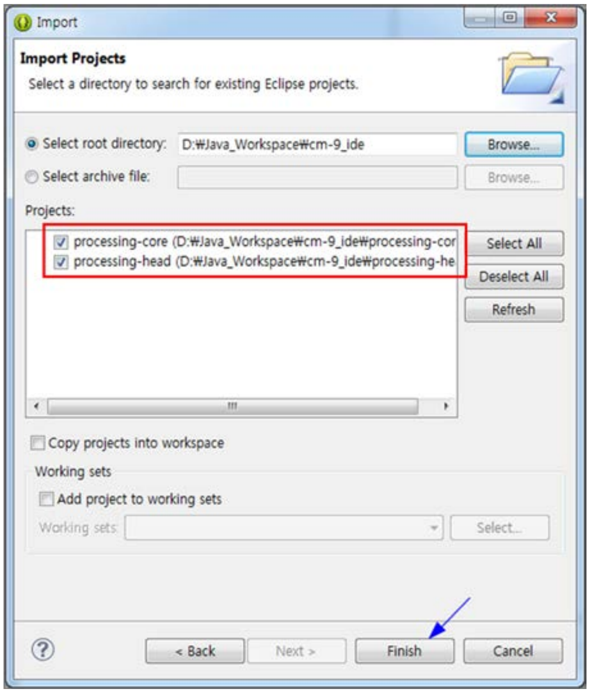


Browse For Folder

Select root directory of the projects to import

Folder: cm-9_ide

Click on OK and you should see 2 projects



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: D:₩Java_Workspace₩cm-9_ide

○ Select archive file:

Projects:

☑ processing-core (D:₩Java_Workspace₩cm-9_ide₩processing-cor

☑ processing-head (D:₩Java_Workspace₩cm-9_ide₩processing-he

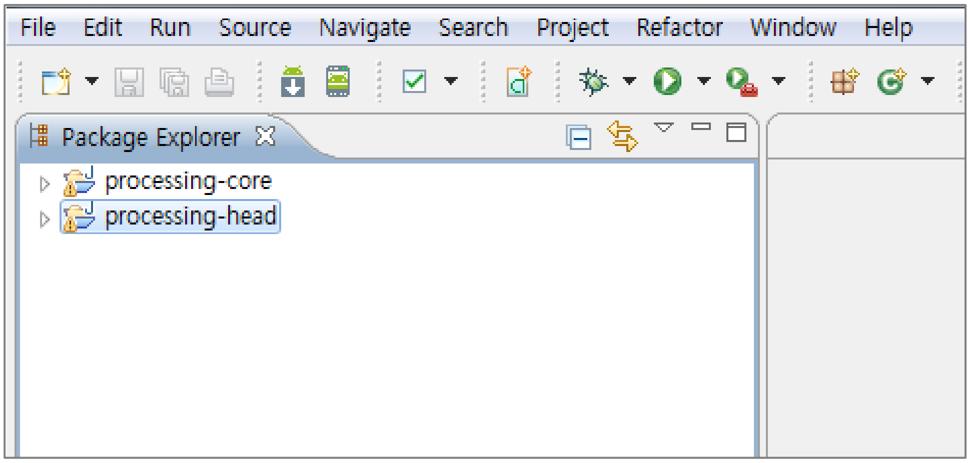
☐ Copy projects into workspace

Working sets

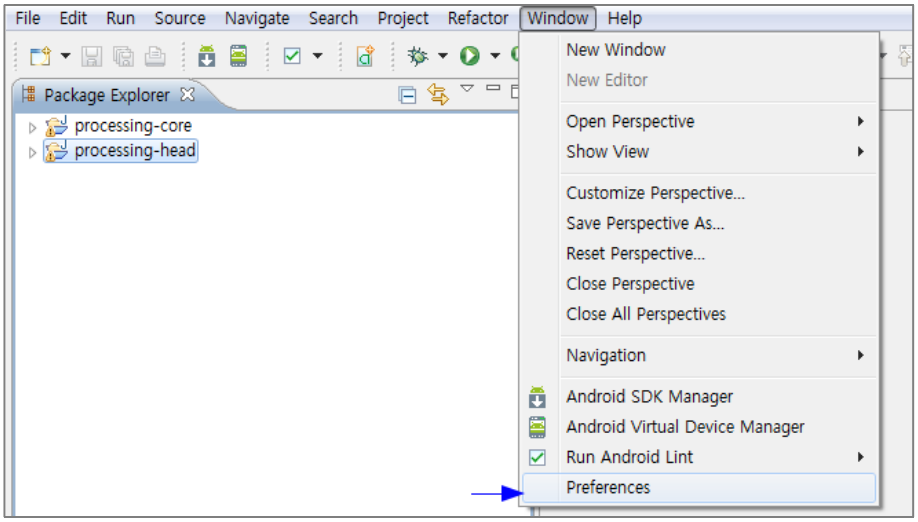
☐ Add project to working sets

Working sets:

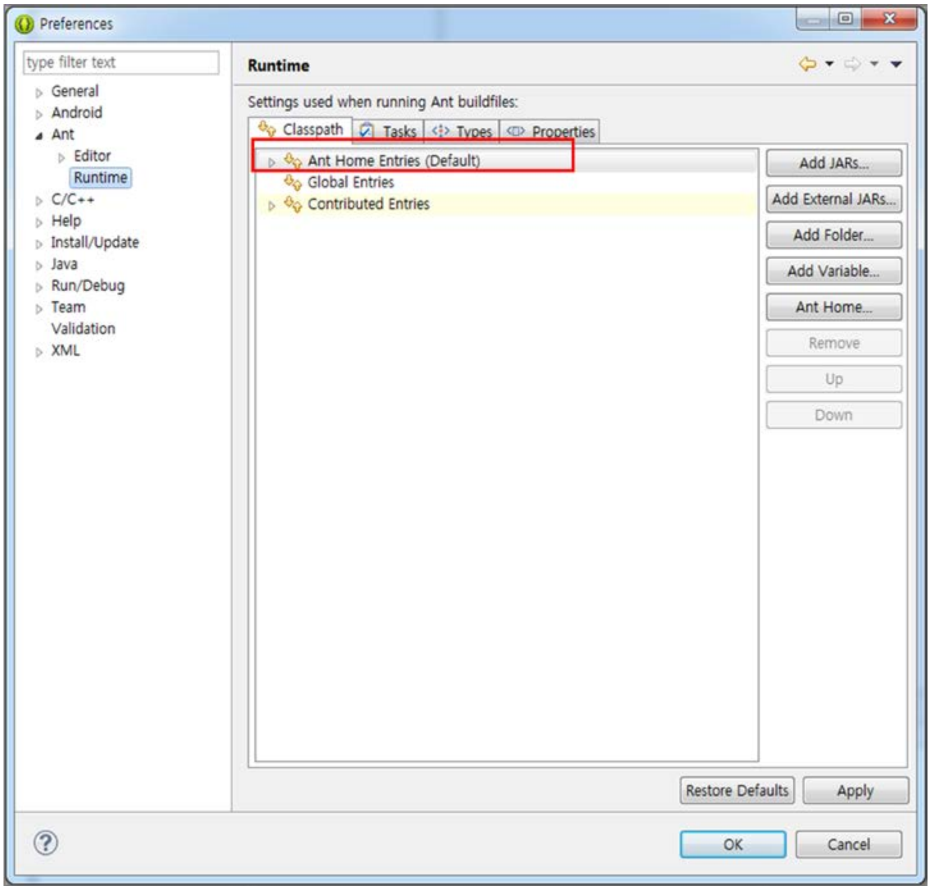
Click on Finish to register both projects.



7.7.3 from Ant tools in runtime options register tools.jar

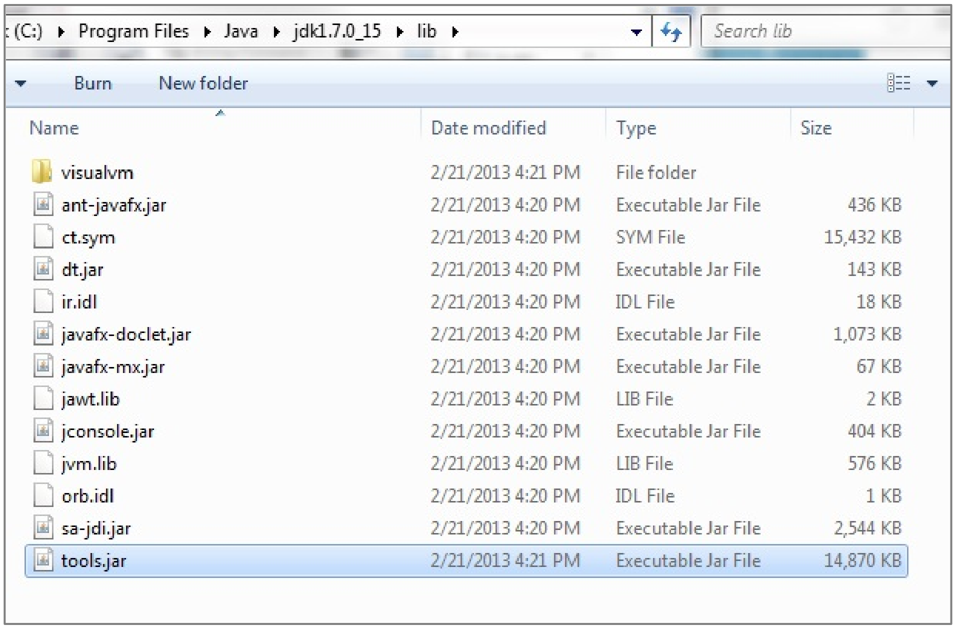


From Preferences -> Ant -> Runtime items click on Classpath tab

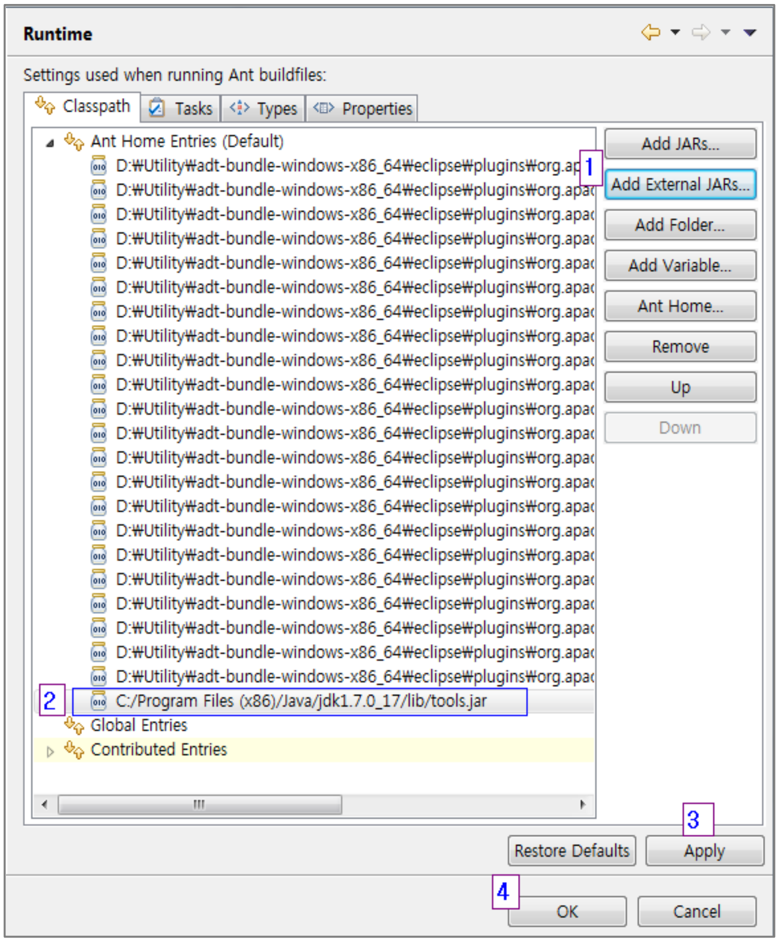


Settings used when running Ant buildfiles:

From Ant Home Entries(Default) add the path to tools.jar by clicking on Add External JARs…



Name	Date modified	Type	Size
visualvm	2/21/2013 4:21 PM	File folder	
ant-javafx.jar	2/21/2013 4:20 PM	Executable Jar File	436 KB
ct.sym	2/21/2013 4:20 PM	SYM File	15,432 KB
dt.jar	2/21/2013 4:20 PM	Executable Jar File	143 KB
ir.idl	2/21/2013 4:20 PM	IDL File	18 KB
javafx-doclet.jar	2/21/2013 4:20 PM	Executable Jar File	1,073 KB
javafx-mx.jar	2/21/2013 4:20 PM	Executable Jar File	67 KB
jawt.lib	2/21/2013 4:20 PM	LIB File	2 KB
jconsole.jar	2/21/2013 4:20 PM	Executable Jar File	404 KB
jvm.lib	2/21/2013 4:20 PM	LIB File	576 KB
orb.idl	2/21/2013 4:20 PM	IDL File	1 KB
sa-jdi.jar	2/21/2013 4:20 PM	Executable Jar File	2,544 KB
tools.jar	2/21/2013 4:21 PM	Executable Jar File	14,870 KB



Settings used when running Ant buildfiles:

Ant Home Entries (Default)

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.ap

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

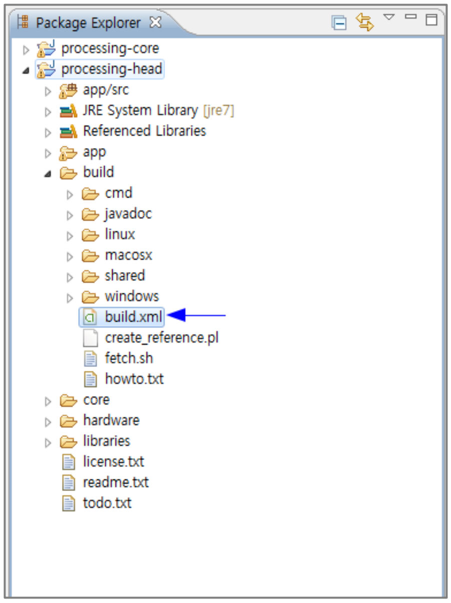
D:₩Utility₩adt-bundle-windows-x86_64₩eclipse₩plugins₩org.apac

C:/Program Files (x86)/Java/jdk1.7.0_17/lib/tools.jar

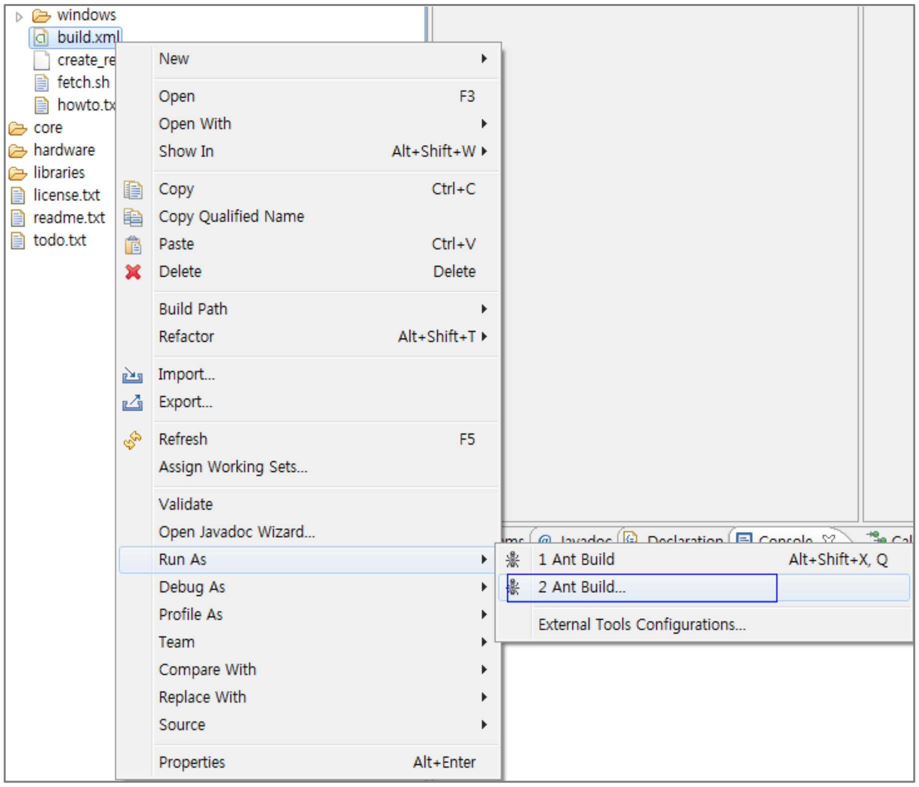
Global Entries

Contributed Entries

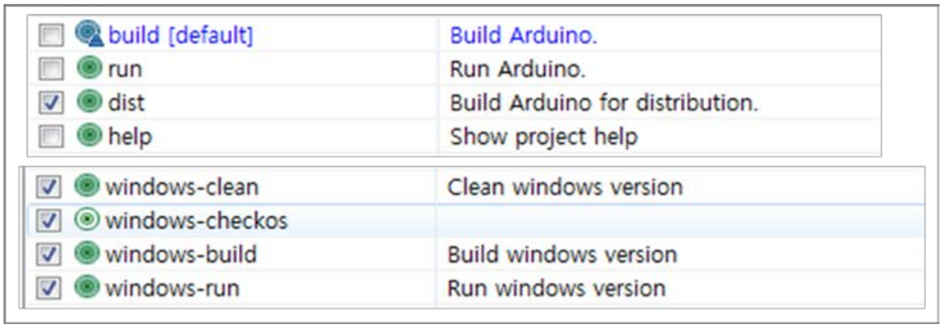
7.7.4 from Project Explorer add build.xml



7.7.5 Right click on the file and go to Run As -> 2 Ant Build…

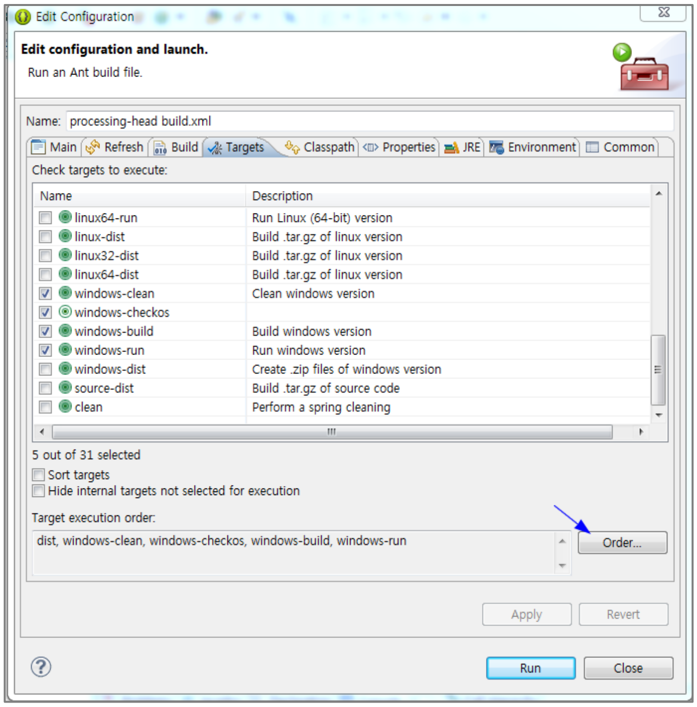


7.7.6 Select all parts



□	build [default]	Build Arduino.
□	run	Run Arduino.
☑	dist	Build Arduino for distribution.
□	help	Show project help
☑	windows-clean	Clean windows version
☑	windows-checkos	
☑	windows-build	Build windows version
☑	windows-run	Run windows version

7.7.7 Control the build sequence as shown below



Edit configuration and launch.

Run an Ant build file.

Name: processing-head build.xml

Check targets to execute:

	Name	Description
□	linux64-run	Run Linux (64-bit) version
□	linux-dist	Build .tar.gz of linux version
□	linux32-dist	Build .tar.gz of linux version
□	linux64-dist	Build .tar.gz of linux version
☑	windows-clean	Clean windows version
☑	windows-checkos	
☑	windows-build	Build windows version
☑	windows-run	Run windows version
□	windows-dist	Create .zip files of windows version
□	source-dist	Build .tar.gz of source code
□	clean	Perform a spring cleaning

5 out of 31 selected

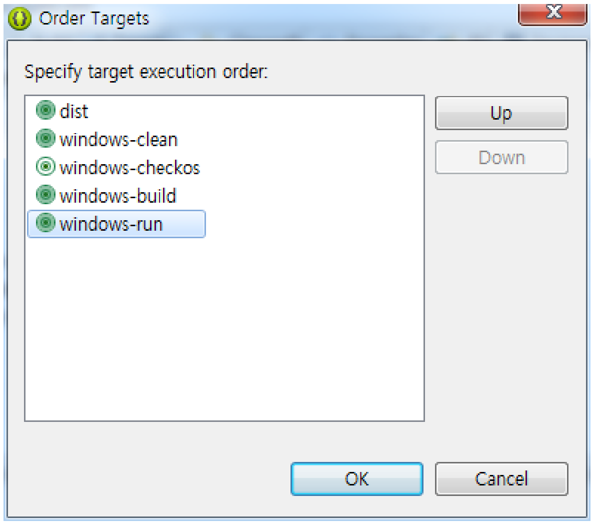
□ Sort targets

□ Hide internal targets not selected for execution

Target execution order:

dist, windows-clean, windows-checkos, windows-build, windows-run

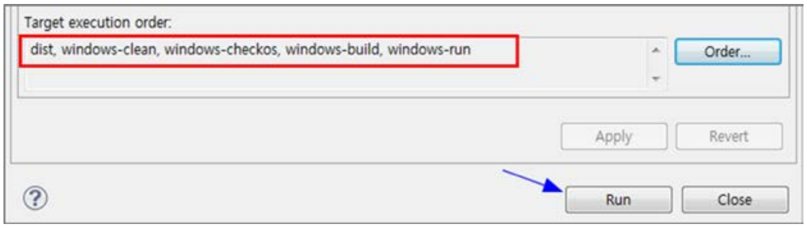
In the pop-up window use the Up/Down to set order.



Specify target execution order:

- dist
- windows-clean
- windows-checkos
- windows-build
- windows-run

Click on OK after setting order

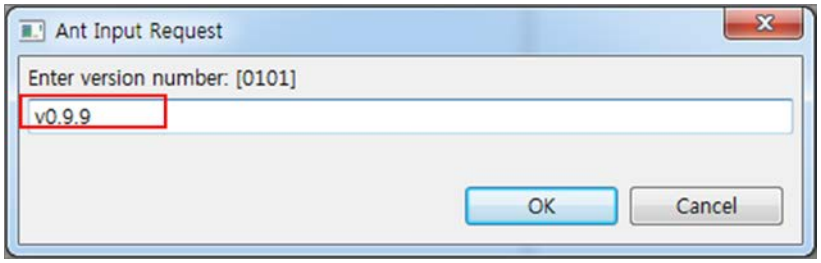


Target execution order:

dist, windows-clean, windows-checkos, windows-build, windows-run

Click on Run to begin build

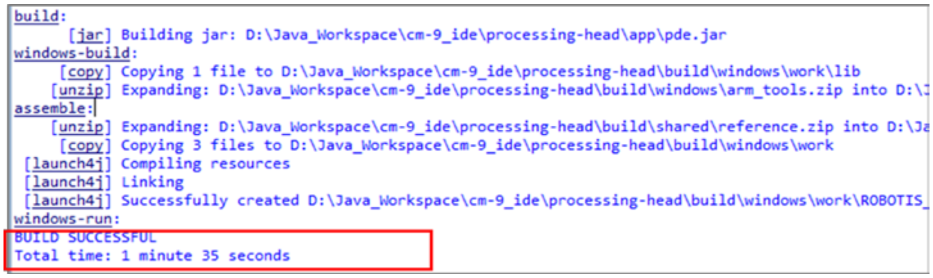
In Ant Input Request pop-up window enter v0.9.9 (that's the version of your generated IDE).


Ant Input Request

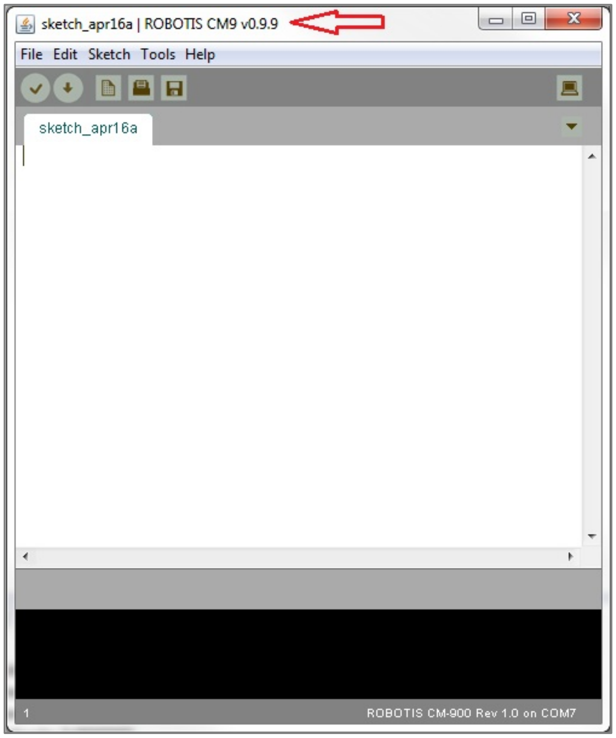
Enter version number: [0101]

v0.9.9

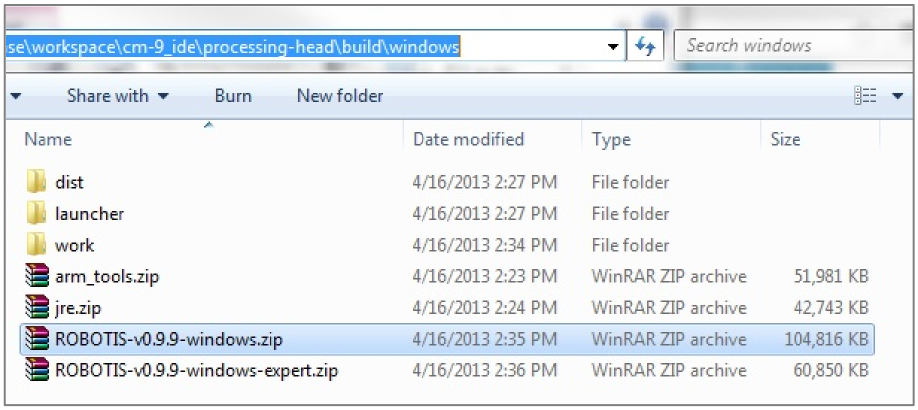
Click on Ok. Build time depends on the computing power of the computer. The approximate build time is about can be from 90 second to as long as 5 minutes.


```
build:
      [jar] Building jar: D:\Java_Workspace\cm-9_ide\processing-head\app\pde.jar
windows-build:
     [copy] Copying 1 file to D:\Java_Workspace\cm-9_ide\processing-head\build\windows\work\lib
    [unzip] Expanding: D:\Java_Workspace\cm-9_ide\processing-head\build\windows\arm_tools.zip into D:\J
assemble:
    [unzip] Expanding: D:\Java_Workspace\cm-9_ide\processing-head\build\shared\reference.zip into D:\Ja
     [copy] Copying 3 files to D:\Java_Workspace\cm-9_ide\processing-head\build\windows\work
 [launch4j] Compiling resources
 [launch4j] Linking
 [launch4j] Successfully created D:\Java_Workspace\cm-9_ide\processing-head\build\windows\work\ROBOTIS_
windows-run:
BUILD SUCCESSFUL
Total time: 1 minute 35 seconds
```

7.7.8 ROBOTIS CM9 launches


sketch_apr16a | ROBOTIS CM9 v0.9.9
File Edit Sketch Tools Help
sketch_apr16a
1
ROBOTIS CM-900 Rev 1.0 on COM7

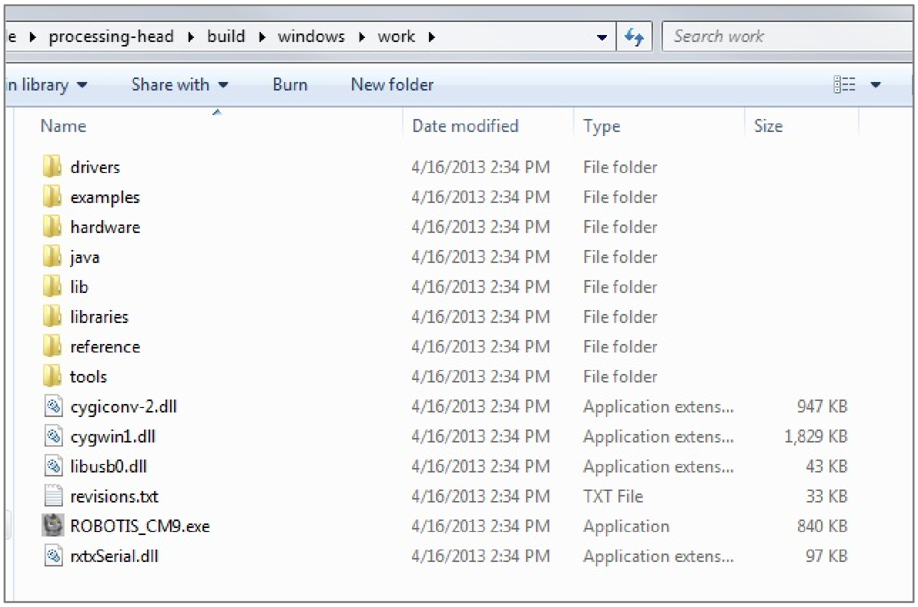
7.7.9 The zip is stored in the set path built on the implemented Java version. For example, ROBOTIS-v0.9.9-windows.zip contains java v0.9.9.



se\workspace\cm-9_ide\processing-head\build\windows

Name	Date modified	Type	Size
dist	4/16/2013 2:27 PM	File folder	
launcher	4/16/2013 2:27 PM	File folder	
work	4/16/2013 2:34 PM	File folder	
arm_tools.zip	4/16/2013 2:23 PM	WinRAR ZIP archive	51,981 KB
jre.zip	4/16/2013 2:24 PM	WinRAR ZIP archive	42,743 KB
ROBOTIS-v0.9.9-windows.zip	4/16/2013 2:35 PM	WinRAR ZIP archive	104,816 KB
ROBOTIS-v0.9.9-windows-expert.zip	4/16/2013 2:36 PM	WinRAR ZIP archive	60,850 KB

The executable is located inside work folder.



e ▸ processing-head ▸ build ▸ windows ▸ work ▸

Name	Date modified	Type	Size
drivers	4/16/2013 2:34 PM	File folder	
examples	4/16/2013 2:34 PM	File folder	
hardware	4/16/2013 2:34 PM	File folder	
java	4/16/2013 2:34 PM	File folder	
lib	4/16/2013 2:34 PM	File folder	
libraries	4/16/2013 2:34 PM	File folder	
reference	4/16/2013 2:34 PM	File folder	
tools	4/16/2013 2:34 PM	File folder	
cygiconv-2.dll	4/16/2013 2:34 PM	Application extens...	947 KB
cygwin1.dll	4/16/2013 2:34 PM	Application extens...	1,829 KB
libusb0.dll	4/16/2013 2:34 PM	Application extens...	43 KB
revisions.txt	4/16/2013 2:34 PM	TXT File	33 KB
ROBOTIS_CM9.exe	4/16/2013 2:34 PM	Application	840 KB
rxtxSerial.dll	4/16/2013 2:34 PM	Application extens...	97 KB