葉 難

Raise your cup, say cheers to the moon, look down on the ground, the shadow is also drinking with me. I'm not a lonely drinker.

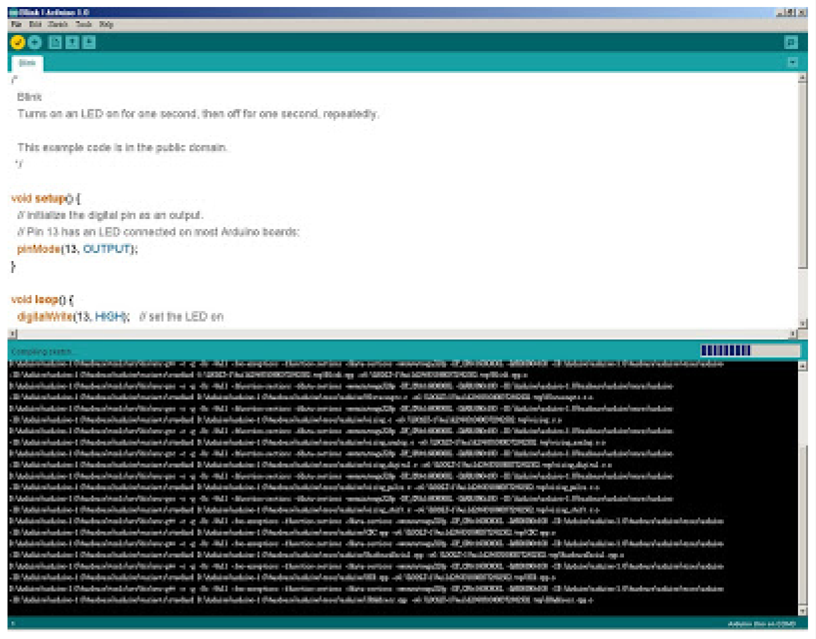
2012/03/19

Compile and Upload Arduino sketches under Windows and Cygwin

My environment is Windows XP and Cygwin. I use Arduino Uno Rev 3，Arduino Software 1.0.

After using Arduino software(IDE) for a while, I felt it's not efficient. Every time I click Sketch-Verify/Compile or File-Upload, it takes a period of time. We can go to File-Preferences and click both "Show verbose output during: compilation and upload", to clearly see what steps it executes.

Besides compiling sketches, it also needs to compile the Arduino core source files in hardware\arduino\cores\arduino. But! It compiles them everytime. Why? Those files are not touched, they don't need recompilation. And If I use Sketch-Verify/Compile to check if the sketch is ok, then use File-Upload to upload to microprocessor, well, the Arduino software does it all over from the very beginning. It's no good.



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

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
```

Anyway, Arduino software(IDE, development environment) is not very smart. I'll introduce how to use make and Makefile to do compilation and uploading in Cygwin command line mode.

SEARCH THIS BLOG

Search

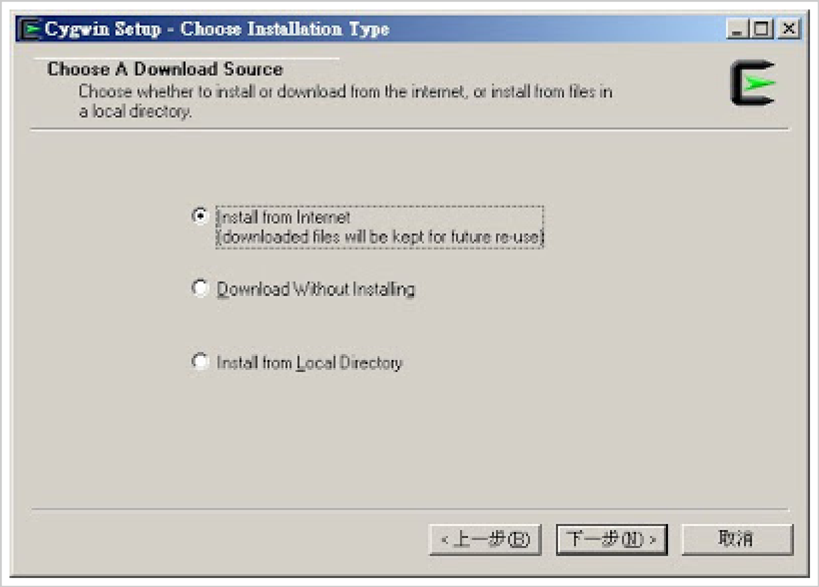
ARCHIVES(BY TIME)

- ▶ 2017 (16)
- ▶ 2016 (73)
- ▶ 2015 (95)
- ▶ 2014 (77)
- ▶ 2013 (88)
- ▼ 2012 (76)
  - ▶ 12 (8)
  - ▶ 11 (5)
  - ▶ 10 (1)
  - ▶ 09 (5)
  - ▶ 08 (11)
  - ▶ 07 (10)
  - ▶ 06 (7)
  - ▶ 04 (3)
  - ▼ 03 (6)
    - Arduino一個好用的計時器程式庫
    - Objective-C與iOS的學習資源
    - 在Windows上寫Objective-C程式（使用GNUstep與Clang、LLVM）
    - Compile and Upload Arduino sketches under Windows ...
    - Arduino小知識：為什麼LED需要串聯的電阻值是220 ohm？

First of all, we still need to install Arduino software. We are going to use the command line mode(terminal) instead of the IDE, but still need various tools and Arduino core source files in it.

Then, install Cygwin. Download Cygwin's setup.exe here. Install it as following.

Launch setup.exe. We will download cygwin and install it from Internet, so choose first option "Install from Internet".



Cygwin Setup - Choose Installation Type

Choose A Download Source

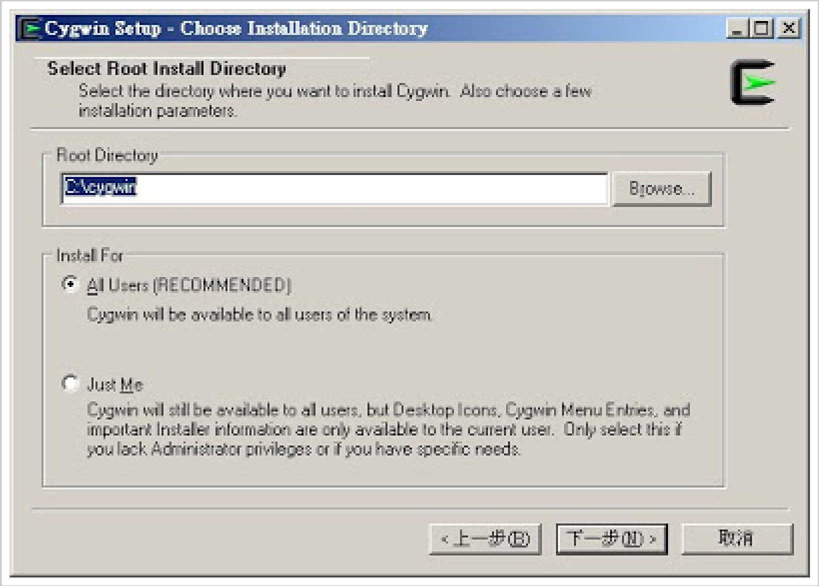
Choose whether to install or download from the internet, or install from files in a local directory.

● Install from Internet (downloaded files will be kept for future re-use)

○ Download Without Installing

○ Install from Local Directory

Specify a directory to install. Default is C:\cygwin. Don't not change it unless you know what you're doing.



Cygwin Setup - Choose Installation Directory

Select Root Install Directory

Select the directory where you want to install Cygwin. Also choose a few installation parameters.

Root Directory: C:\cygwin

Install For

● All Users (RECOMMENDED)
Cygwin will be available to all users of the system.

○ Just Me
Cygwin will still be available to all users, but Desktop Icons, Cygwin Menu Entries, and important Installer information are only available to the current user. Only select this if you lack Administrator privileges or if you have specific needs.

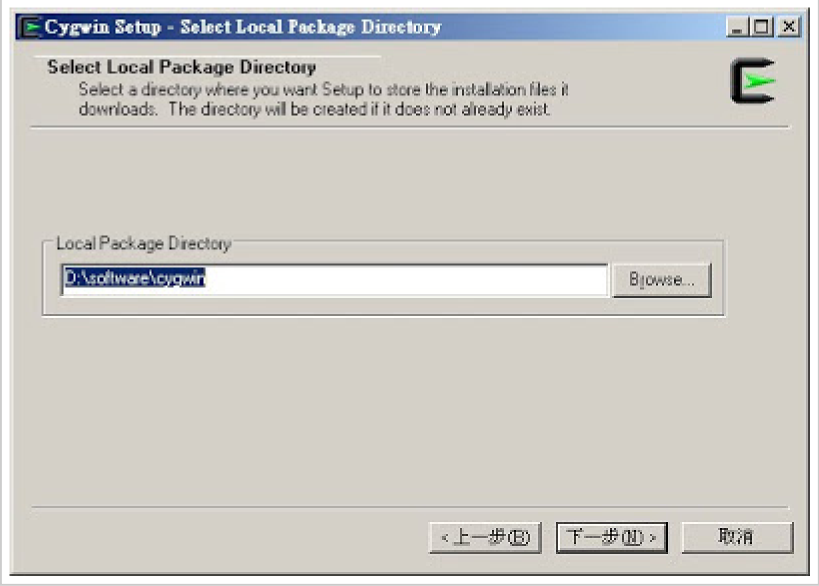
setup.exe will keep the downloaded package in another directory. Please specify some path.

一篇關於Raspberry Pi的報導、錯誤百出

- ▶ 02 (17)
- ▶ 01 (3)
- ▶ 2011 (32)
- ▶ 2010 (28)
- ▶ 2009 (34)
- ▶ 2008 (68)
- ▶ 2007 (105)

ARCHIVES(BY TAGS)

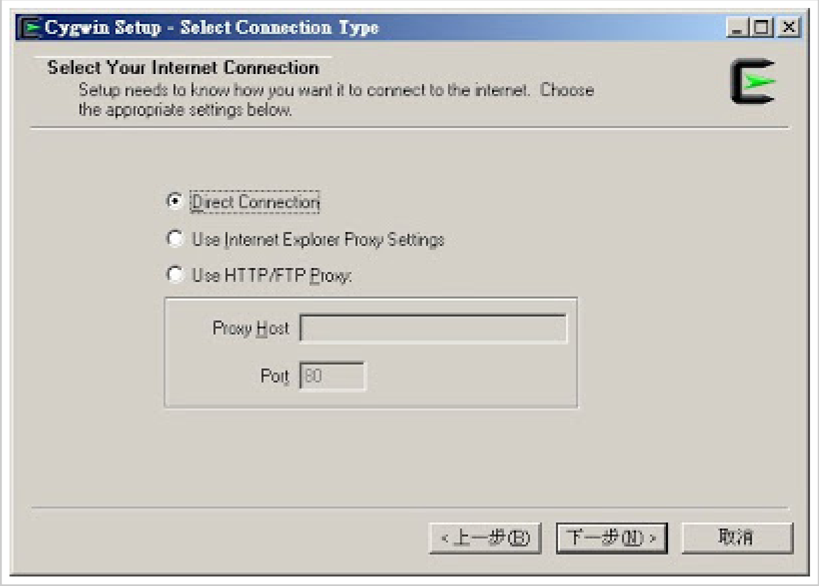
Arduino (96)
Arduino文章列表 (1)
Australia (79)
Australia Working Holiday (1)
Banana Pi (5)
CircuitPython (1)
cocos2d-iphone (4)
ESP8266 (1)
iOS (23)
Java (2)
JavaScript (1)
Learning English (27)
LinkIt ONE (30)
Linux (3)
MicroPython (1)
NodeMCU (1)
programming (26)
Project Euler (5)
Python (18)
Raspberry Pi (192)
UP board (10)
咬一口蘋果 (17)
婚禮喜宴 (3)
懷舊記憶 (12)
旅遊Travel (5)
玩具 (1)
胡言亂語 (14)
遊戲 (3)
閱讀 (36)
電子Maker (26)


Cygwin Setup - Select Local Package Directory

Select Local Package Directory
Select a directory where you want Setup to store the installation files it downloads. The directory will be created if it does not already exist.

Local Package Directory: D:\software\cygwin

Specify internet connection according to your setting.

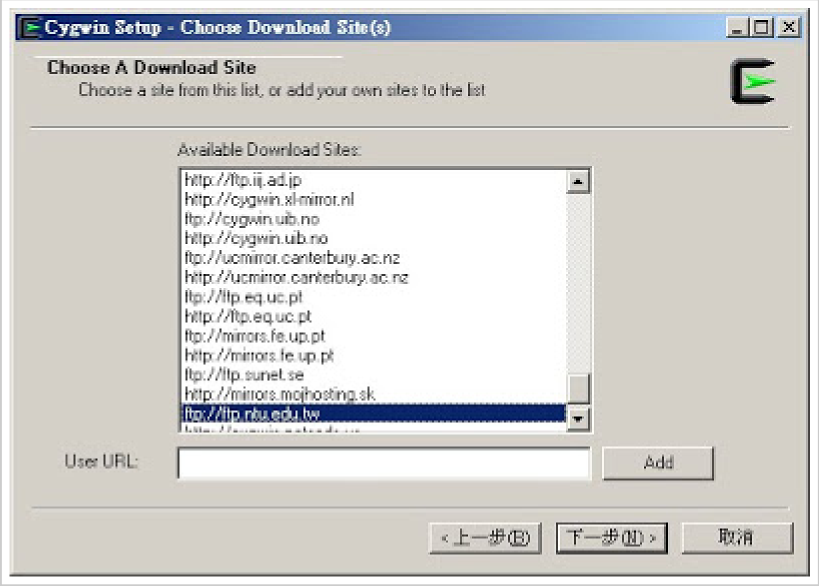

Cygwin Setup - Select Connection Type

Select Your Internet Connection
Setup needs to know how you want it to connect to the internet. Choose the appropriate settings below.

● Direct Connection
○ Use Internet Explorer Proxy Settings
○ Use HTTP/FTP Proxy:

Proxy Host:
Port: 80

Then, choose a server to download. Maybe choose one according to your country.


Cygwin Setup - Choose Download Site(s)

Choose A Download Site
Choose a site from this list, or add your own sites to the list

Available Download Sites:

```
http://ftp.iij.ad.jp
http://cygwin.xl-mirror.nl
ftp://cygwin.uib.no
http://cygwin.uib.no
ftp://ucmirror.canterbury.ac.nz
http://ucmirror.canterbury.ac.nz
ftp://ftp.eq.uc.pt
http://ftp.eq.uc.pt
ftp://mirrors.fe.up.pt
http://mirrors.fe.up.pt
ftp://ftp.sunet.se
http://mirrors.mojhosting.sk
ftp://ftp.ntu.edu.tw
```

User URL:

電影 (3)
電腦與網路 (33)

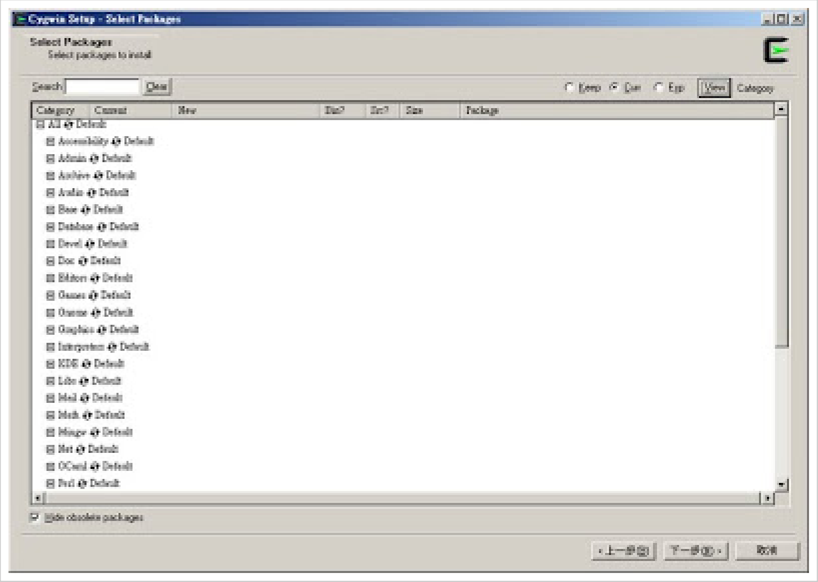
ABOUT ME

YEHNAN

朝曦灑落身 俯仰似流雲

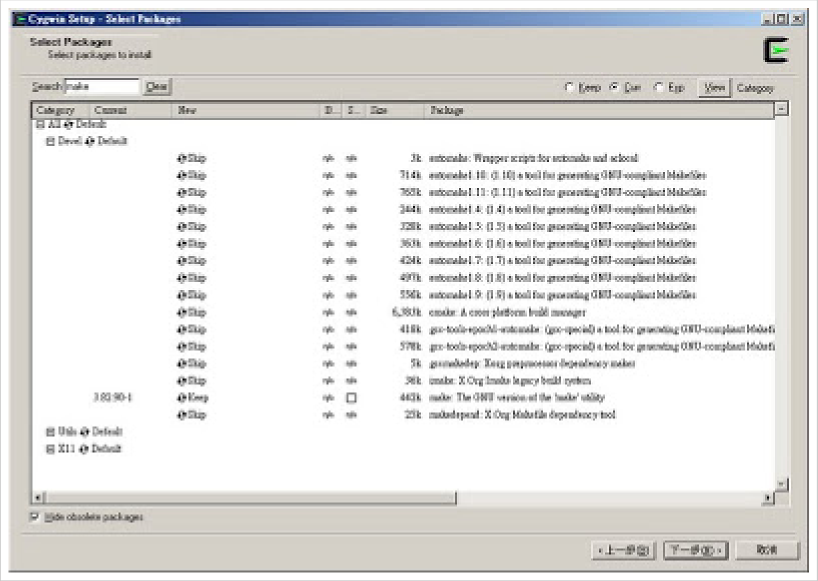
VIEW MY COMPLETE PROFILE

Then choose what packages to install. Defaults won't install "make". So we need to choose it manually.



Category	Current	New	Bin?	Src?	Size	Package
⊟ All ⟳ Default						
⊞ Accessibility ⟳ Default						
⊞ Admin ⟳ Default						
⊞ Archive ⟳ Default						
⊞ Audio ⟳ Default						
⊞ Base ⟳ Default						
⊞ Database ⟳ Default						
⊞ Devel ⟳ Default						
⊞ Doc ⟳ Default						
⊞ Editors ⟳ Default						
⊞ Games ⟳ Default						
⊞ Gnome ⟳ Default						
⊞ Graphics ⟳ Default						
⊞ Interpreters ⟳ Default						
⊞ KDE ⟳ Default						
⊞ Libs ⟳ Default						
⊞ Mail ⟳ Default						
⊞ Math ⟳ Default						
⊞ Mingw ⟳ Default						
⊞ Net ⟳ Default						
⊞ OCaml ⟳ Default						
⊞ Perl ⟳ Default						

Input "make" in Search field. Under "Devel" category, choose "make: The GNU version of the 'make' utility". It should show 3.82.90-1 or similiar version number.



Search: make

Category	Current	New	B...	S...	Size	Package
⊟ All ⟳ Default						
⊟ Devel ⟳ Default						
		⟳ Skip	n/a	n/a	3k	automake: Wrapper scripts for automake and aclocal
		⟳ Skip	n/a	n/a	714k	automake1.10: (1.10) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	765k	automake1.11: (1.11) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	244k	automake1.4: (1.4) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	328k	automake1.5: (1.5) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	363k	automake1.6: (1.6) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	424k	automake1.7: (1.7) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	497k	automake1.8: (1.8) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	556k	automake1.9: (1.9) a tool for generating GNU-compliant Makefiles
		⟳ Skip	n/a	n/a	6,383k	cmake: A cross platform build manager
		⟳ Skip	n/a	n/a	418k	gcc-tools-epoch1-automake: (gcc-special) a tool for generating GNU-compliant Makefi
		⟳ Skip	n/a	n/a	578k	gcc-tools-epoch2-automake: (gcc-special) a tool for generating GNU-compliant Makefi
		⟳ Skip	n/a	n/a	5k	gccmakedep: Xorg preprocessor dependency maker
		⟳ Skip	n/a	n/a	38k	imake: X.Org Imake legacy build system
	3.82.90-1	⟳ Keep	n/a	□	442k	make: The GNU version of the 'make' utility
		⟳ Skip	n/a	n/a	23k	makedepend: X.Org Makefile dependency tool
⊞ Utils ⟳ Default						
⊞ X11 ⟳ Default						

Then download and install. setup.exe might prompt you to install some dependent packages, confirm it.

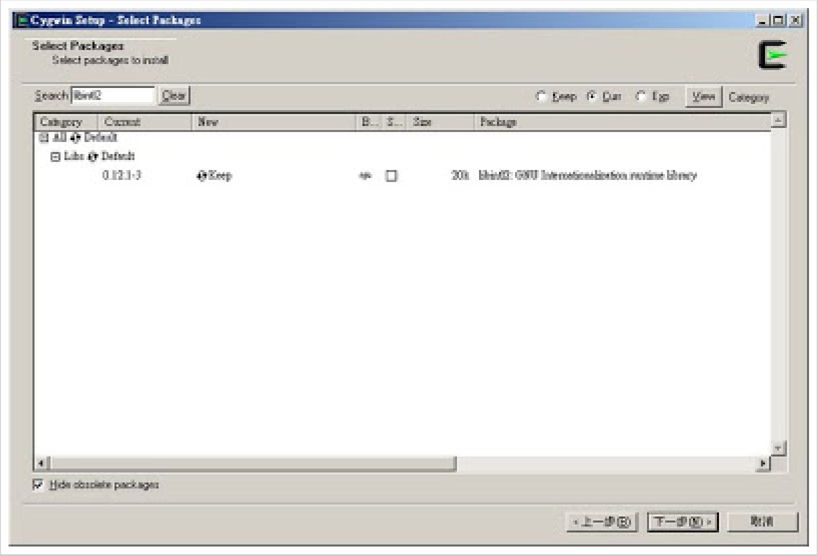
I also install many other packages, so my C:\cygwin is about 700 MB.

However, new version of make can't correctly handle the Makefile I wrote. So we need to revert back to version 3.80. Launch cygwin terminal, do the following steps to change back to 3.80.

```
cd /usr/bin
mv make.exe make_382.exe
wget http://geant4.cern.ch/support/extras/cygwin/make.exe
```

chmod +x make.exe

make 3.80 also need some library, so please launch setup.exe one more time, and install this library "**libintl2".**



Cygwin Setup - Select Packages

Select Packages
Select packages to install

Search libintl2

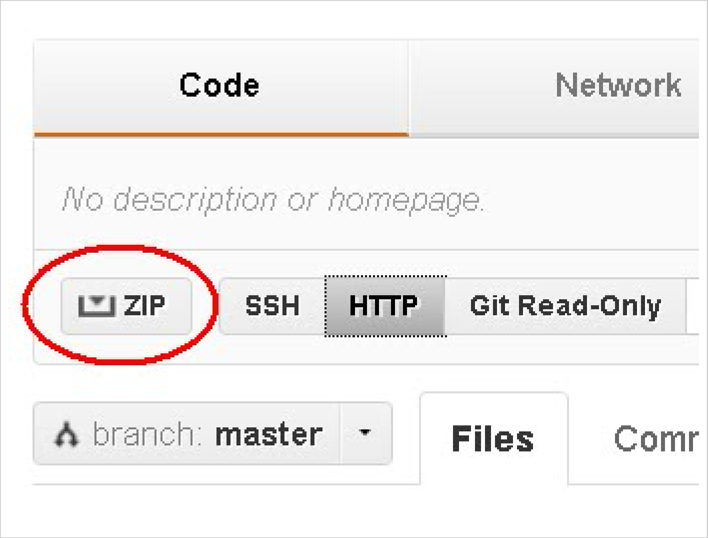
○ Keep ● Curr ○ Exp

Category	Current	New	B...	S...	Size	Package
⊟ All ♻ Default						
⊟ Libs ♻ Default						
	0.12.1-3	♻Keep	n/a	□	20k	libintl2: GNU Internationalization runtime library

☑ Hide obsolete packages

Check it. execute "make --version" in cygwin terminal. It should respond "3.80".

You can download the example sketch using make here. Use ZIP button to download all sketches I wrote. However, you only need the one in the directory "BlinkMakefile"



In the directory BlinkMakefile, there are total 6 files: BlinkMakefile.cpp, foo.c, foo.h, bar.cpp, bar.h, and Makefile。（Other directories contains my other Arduino practices. You can ignore them）

BlinkMakefile.cpp, the code will flash LED of pin 13. The difference is that we need to put#include <arduino.h>at the top of the file. Because, Arduino software does it for us, now we need to do it by ourselves.

For demonstration, I also add a C source file foo.c and its header file foo.h. It contains a function called delayDuration returning 1000 whatever.

BlinkMakefile.cpp uses this value as the time period of lighting LED on.

For demonstration, I also add a C++ source file bar.cpp and its header file bar.h. It contains a class ClassBar, a member function getDelayDuration to return a value. Everytime you called this function, the value will increase 100. First time returns 100, second time returns 200, etc.. BlinkMakefile.cpp will use this value as the time period of lighting LED off. So the off period will become longer and longer.

Now comes to the main player Makefile. Excuse me for not explaining the syntax and meaning of Makefile in detail. I'm not an expert. Following are what you need to modify and adapt.

Originally, Arduino software use .pde or .ino filename extensions. Please change to .cpp.

PROJECT = BlinkMakefile
The filename of the final product files(.elf, .hex. .eep). Change to anything you want, better be no spaces.

ifndef PROJ_SOURCE
PROJ_SOURCE := $(wildcard *.c) $(wildcard *.cpp)
endif。
This will find out all sources file(.c and .cpp). Please put all souce files of your project in the same directory as the Makefile. Don't use sub-directories.

MCU = atmega328p
What microcontroller chip does your board use. Compiler needs this setting to generate correct binary machine code. My board is Arduino Uno Rev3, so uses atmega328p. Both compiling and uploading need this setting.

F_CPU = 16000000L
Frequency of the microcontroller. It seems the same for all boards to maintain backward compatibility, so you probably don't need to change.

ARDUINO_VERSION = 100
The version number of you Arduino software. This will pass into compiler, you can use this in code.

FORMAT = ihex
Output format. Probably no need to change.

PORT = \\.\COM3
Serial port. This is from Tools-Serial Port in Arduino software. You might be curious what is \\.\ for. This is related to some weirdness about argument specification, windows path, etc.. Anyway, just leave it.

UPLOAD_RATE = 115200
Transmission baud rate of the serial port. Normally it is 9600. But my board can use 115200. Uploading needs this setting.

INSTALL_PATH = D:/Arduino/arduino-1.0
This path should point to the top-most directory of the Arduino software. It's where containing the executable file arduino.exe, folder example, folder libraries, etc.. Please change it according to yours.

CORE = $(INSTALL_PATH)/hardware/arduino/cores/arduino
This path should point to the directory of Arduino core source files. It's where containing Arduino.h, wiring_analog.c, wiring_shift.c, etc.. I let it be a path relative to INSTALL_PATH.

BUILTINLIB = $(INSTALL_PATH)/libraries
Path to the Arduino built-in libraries containing EEPROM, Ethernet, LiquidCrystal, etc.. I let it be a path relative to INSTALL_PATH.

SKETCHBOOK = D:/Arduino/sketchbook
The sketchbook path. A path to your sketches/projects.

THIRDPARTYLIB = $(SKETCHBOOK)/libraries
Path to the third-party libraries(the external libraries you installed). I let it be a path relative to SKETCHBOOK.

AVRDUDE_PROGRAMMER = arduino
What programmer to upload the sketches/code/programs. "arduino" means to use the bootloader inside microcontroller to do uploading. According to your board, might be stk500, stk500v1, or stk500v2. Available settings please refer to the document of the tool avrdude which is used to do uploading(programming).

If you need to use the built-in libraries, please modify this line:
BUILTINLIB_ENABLED =
Put the names of libraries after the equal sign. For example:
BUILTINLIB_ENABLED = EEPROM Ethernet Firmata LiquidCrystal
(BUILTINLIB should point to the directory of the built-in libraries)

If you need to use third-party libraries, please modify this line:
THIRDPARTYLIB_ENABLED =
Put the names of libraries after the equal sign. For example:
THIRDPARTYLIB_ENABLED = BOUNCE
(Without saying, you have to download the libary, uncompress, put it in the directory pointed by THIRDPARTYLIB)

Others in the Makefile, they are related to the tool make. Please consult other document if you're interested.

Now, launch cygwin terminal, change working directory to BlinkMakefile. Use the following commands to do compiling and uploading.

First of all:

```
make depend
```

This will find out the dependency relationship among all .c, .cpp, and .h files. And append it at the end of the Makefile.

make

This will compile and link. If succeed, you will get BlinkMakefile.elf, BlinkMakefile.eep.hex, and BlinkMakefile.eep(and other .o files).

If execute make again, it should appear make: Nothing to be done for `all'. Wonderful.

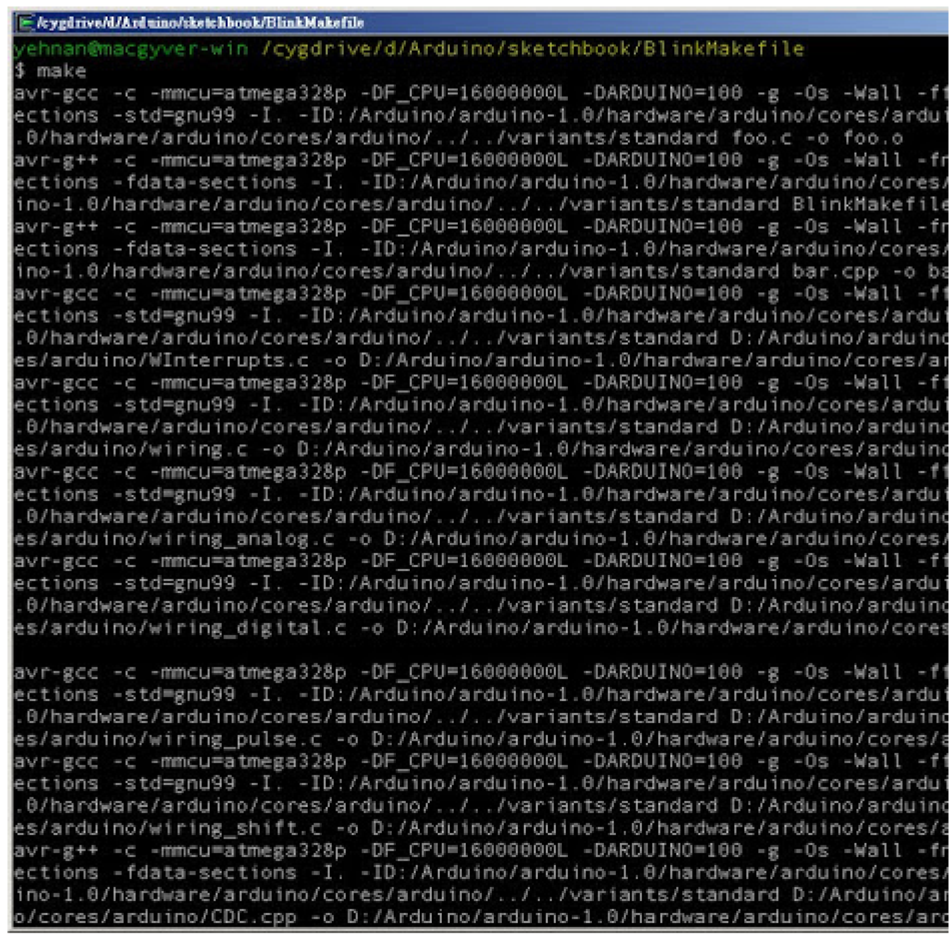
make upload

This will upload the sketch to the board.

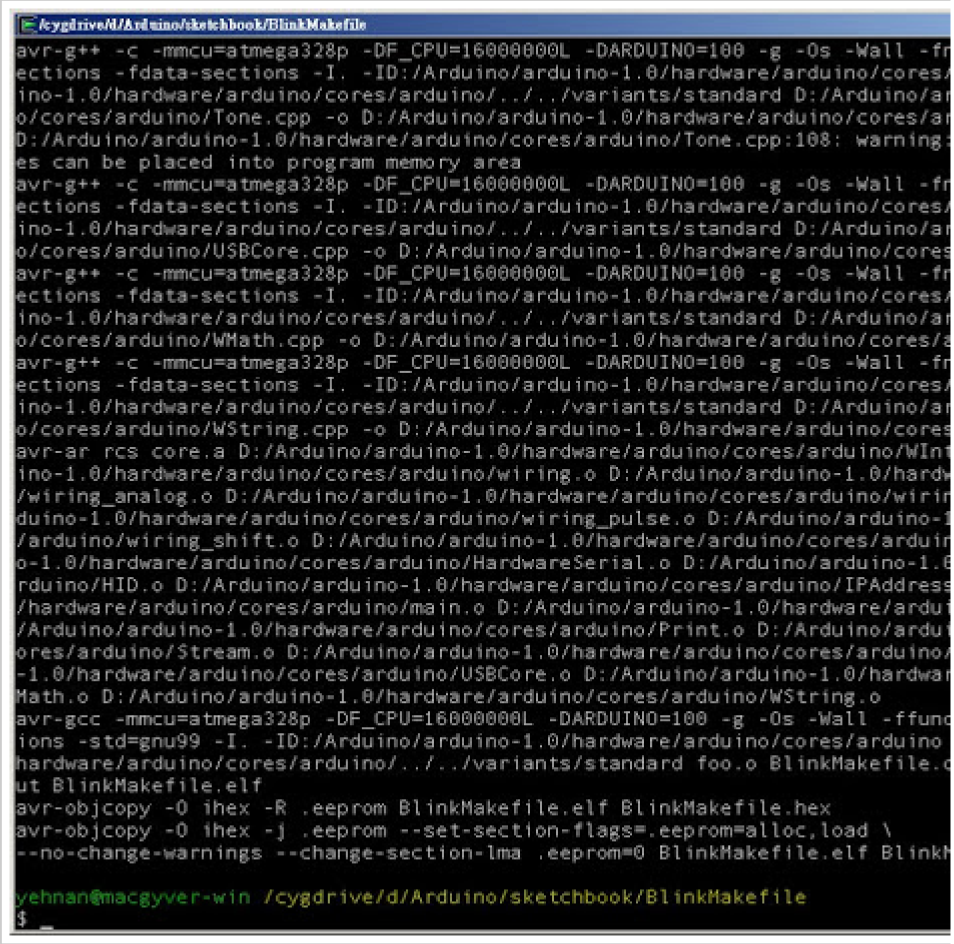
make clean

This will delete all intermediate files. This Makefile will put all .o files and other intermediate files directly in the directories of the source files.

This is the screen snapshot of my executing make.

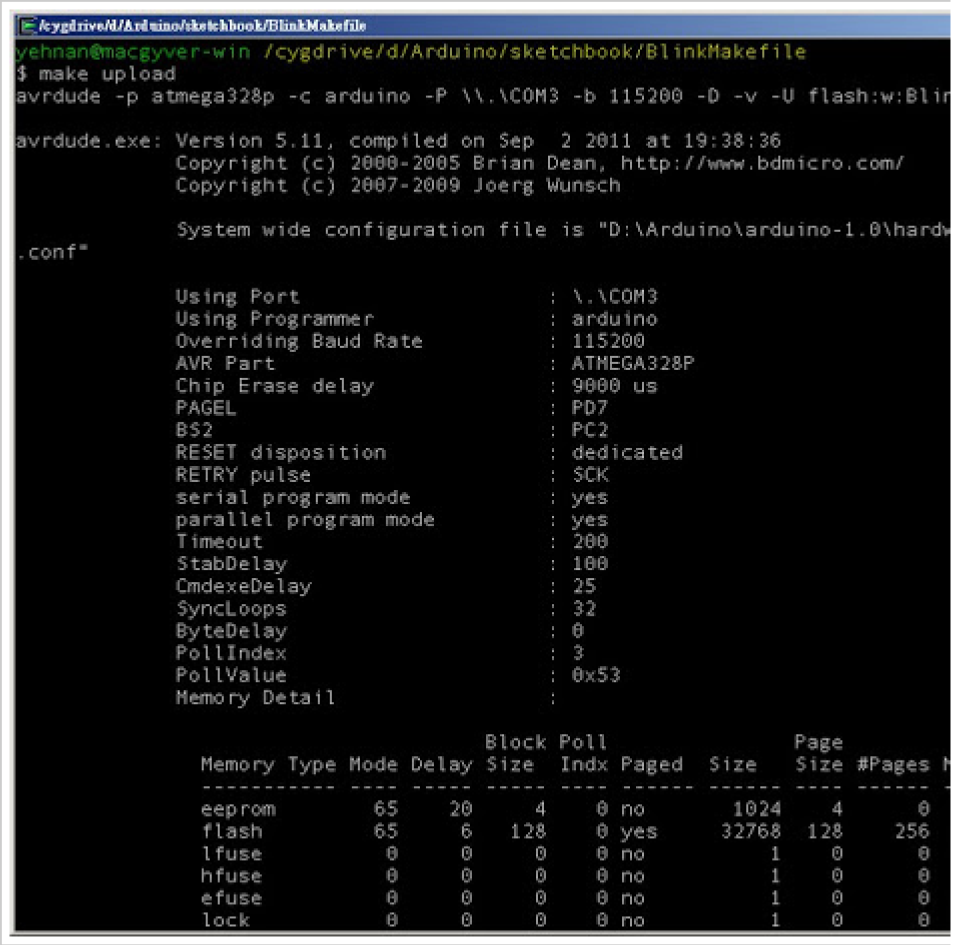


```
/cygdrive/d/Arduino/sketchbook/BlinkMakefile
yehnan@macgyver-win /cygdrive/d/Arduino/sketchbook/BlinkMakefile
$ make
avr-gcc -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ff
ections -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/ardui
.0/hardware/arduino/cores/arduino/../../variants/standard foo.c -o foo.o
avr-g++ -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -fr
ections -fdata-sections -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/
ino-1.0/hardware/arduino/cores/arduino/../../variants/standard BlinkMakefile
avr-g++ -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -fr
ections -fdata-sections -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/
ino-1.0/hardware/arduino/cores/arduino/../../variants/standard bar.cpp -o ba
avr-gcc -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ff
ections -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/ardui
.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/arduinc
es/arduino/WInterrupts.c -o D:/Arduino/arduino-1.0/hardware/arduino/cores/ar
avr-gcc -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ff
ections -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/ardui
.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/arduinc
es/arduino/wiring.c -o D:/Arduino/arduino-1.0/hardware/arduino/cores/arduinc
avr-gcc -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ff
ections -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/ardui
.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/arduinc
es/arduino/wiring_analog.c -o D:/Arduino/arduino-1.0/hardware/arduino/cores/
avr-gcc -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ff
ections -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/ardui
.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/arduinc
es/arduino/wiring_digital.c -o D:/Arduino/arduino-1.0/hardware/arduino/cores

avr-gcc -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ff
ections -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/ardui
.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/arduinc
es/arduino/wiring_pulse.c -o D:/Arduino/arduino-1.0/hardware/arduino/cores/a
avr-gcc -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ff
ections -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/ardui
.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/arduinc
es/arduino/wiring_shift.c -o D:/Arduino/arduino-1.0/hardware/arduino/cores/a
avr-g++ -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -fr
ections -fdata-sections -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/
ino-1.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/ar
o/cores/arduino/CDC.cpp -o D:/Arduino/arduino-1.0/hardware/arduino/cores/arc
```

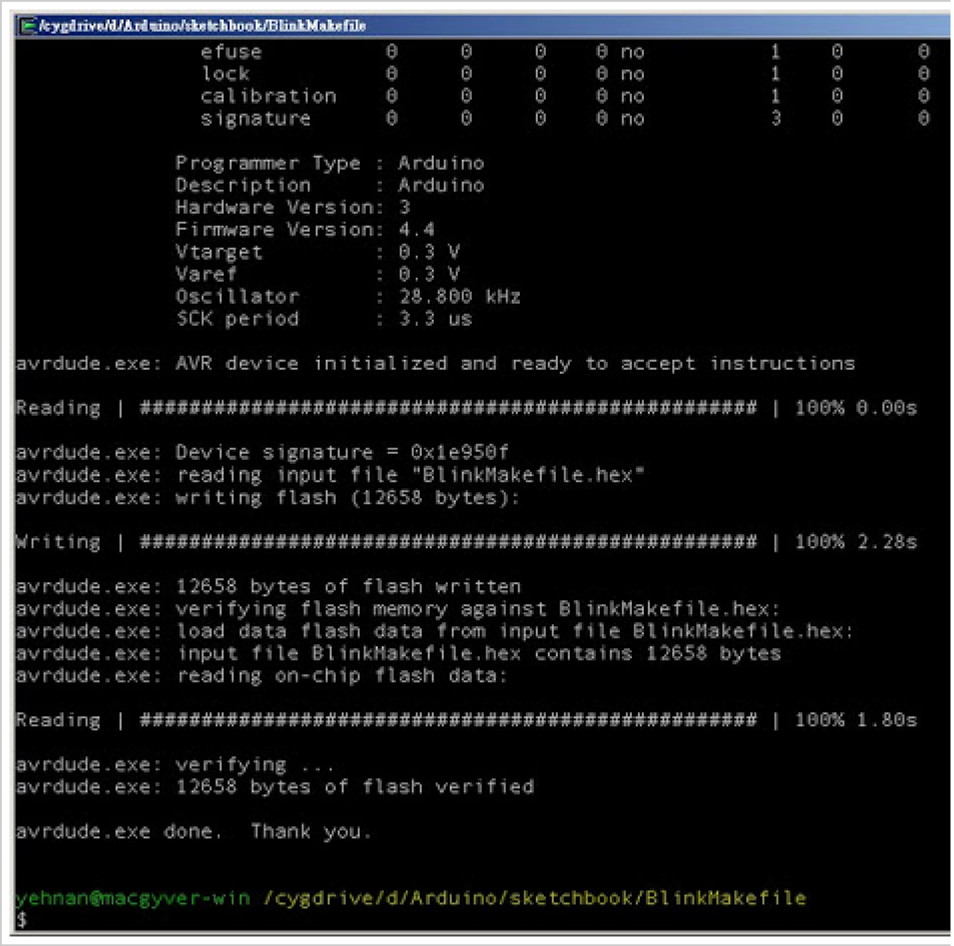


```
avr-g++ -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -fr
ections -fdata-sections -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/
ino-1.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/ar
o/cores/arduino/Tone.cpp -o D:/Arduino/arduino-1.0/hardware/arduino/cores/ar
D:/Arduino/arduino-1.0/hardware/arduino/cores/arduino/Tone.cpp:108: warning:
es can be placed into program memory area
avr-g++ -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -fr
ections -fdata-sections -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/
ino-1.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/ar
o/cores/arduino/USBCore.cpp -o D:/Arduino/arduino-1.0/hardware/arduino/cores
avr-g++ -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -fr
ections -fdata-sections -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/
ino-1.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/ar
o/cores/arduino/WMath.cpp -o D:/Arduino/arduino-1.0/hardware/arduino/cores/a
avr-g++ -c -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -fr
ections -fdata-sections -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/
ino-1.0/hardware/arduino/cores/arduino/../../variants/standard D:/Arduino/ar
o/cores/arduino/WString.cpp -o D:/Arduino/arduino-1.0/hardware/arduino/cores
avr-ar rcs core.a D:/Arduino/arduino-1.0/hardware/arduino/cores/arduino/WInt
ino-1.0/hardware/arduino/cores/arduino/wiring.o D:/Arduino/arduino-1.0/hardw
/wiring_analog.o D:/Arduino/arduino-1.0/hardware/arduino/cores/arduino/wirin
duino-1.0/hardware/arduino/cores/arduino/wiring_pulse.o D:/Arduino/arduino-1
/arduino/wiring_shift.o D:/Arduino/arduino-1.0/hardware/arduino/cores/arduin
o-1.0/hardware/arduino/cores/arduino/HardwareSerial.o D:/Arduino/arduino-1.0
rduino/HID.o D:/Arduino/arduino-1.0/hardware/arduino/cores/arduino/IPAddress
/hardware/arduino/cores/arduino/main.o D:/Arduino/arduino-1.0/hardware/ardui
/Arduino/arduino-1.0/hardware/arduino/cores/arduino/Print.o D:/Arduino/ardui
ores/arduino/Stream.o D:/Arduino/arduino-1.0/hardware/arduino/cores/arduino/
-1.0/hardware/arduino/cores/arduino/USBCore.o D:/Arduino/arduino-1.0/hardwar
Math.o D:/Arduino/arduino-1.0/hardware/arduino/cores/arduino/WString.o
avr-gcc -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO=100 -g -Os -Wall -ffunc
ions -std=gnu99 -I. -ID:/Arduino/arduino-1.0/hardware/arduino/cores/arduino
hardware/arduino/cores/arduino/../../variants/standard foo.o BlinkMakefile.c
ut BlinkMakefile.elf
avr-objcopy -O ihex -R .eeprom BlinkMakefile.elf BlinkMakefile.hex
avr-objcopy -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load \
--no-change-warnings --change-section-lma .eeprom=0 BlinkMakefile.elf BlinkM

yehnan@macgyver-win /cygdrive/d/Arduino/sketchbook/BlinkMakefile
$ _
```

This is the screen snapshot of my executing make upload.



```
yehnan@macgyver-win /cygdrive/d/Arduino/sketchbook/BlinkMakefile
$ make upload
avrdude -p atmega328p -c arduino -P \\.\COM3 -b 115200 -D -v -U flash:w:Blir

avrdude.exe: Version 5.11, compiled on Sep  2 2011 at 19:38:36
             Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
             Copyright (c) 2007-2009 Joerg Wunsch

             System wide configuration file is "D:\Arduino\arduino-1.0\hardw
.conf"

             Using Port                    : \.\COM3
             Using Programmer              : arduino
             Overriding Baud Rate          : 115200
             AVR Part                      : ATMEGA328P
             Chip Erase delay              : 9000 us
             PAGEL                         : PD7
             BS2                           : PC2
             RESET disposition             : dedicated
             RETRY pulse                   : SCK
             serial program mode           : yes
             parallel program mode         : yes
             Timeout                       : 200
             StabDelay                     : 100
             CmdexeDelay                   : 25
             SyncLoops                     : 32
             ByteDelay                     : 0
             PollIndex                     : 3
             PollValue                     : 0x53
             Memory Detail                 :

                                      Block Poll               Page
               Memory Type Mode Delay Size  Indx Paged  Size   Size #Pages M
               ----------- ---- ----- ----- ---- ------ ------ ---- ------
               eeprom        65    20     4    0 no       1024    4      0
               flash         65     6   128    0 yes     32768  128    256
               lfuse          0     0     0    0 no          1    0      0
               hfuse          0     0     0    0 no          1    0      0
               efuse          0     0     0    0 no          1    0      0
               lock           0     0     0    0 no          1    0      0
```



```
               efuse          0     0     0    0 no          1    0      0
               lock           0     0     0    0 no          1    0      0
               calibration    0     0     0    0 no          1    0      0
               signature      0     0     0    0 no          3    0      0

             Programmer Type : Arduino
             Description     : Arduino
             Hardware Version: 3
             Firmware Version: 4.4
             Vtarget         : 0.3 V
             Varef           : 0.3 V
             Oscillator      : 28.800 kHz
             SCK period      : 3.3 us

avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude.exe: Device signature = 0x1e950f
avrdude.exe: reading input file "BlinkMakefile.hex"
avrdude.exe: writing flash (12658 bytes):

Writing | ################################################## | 100% 2.28s

avrdude.exe: 12658 bytes of flash written
avrdude.exe: verifying flash memory against BlinkMakefile.hex:
avrdude.exe: load data flash data from input file BlinkMakefile.hex:
avrdude.exe: input file BlinkMakefile.hex contains 12658 bytes
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100% 1.80s

avrdude.exe: verifying ...
avrdude.exe: 12658 bytes of flash verified

avrdude.exe done.  Thank you.


yehnan@macgyver-win /cygdrive/d/Arduino/sketchbook/BlinkMakefile
$
```

Well, there is a little problem. Originally we can use the Tools-Serial Monitor of the Arduino software. What now?

You can instead use Windows's HyperTerminal or Tera Term or PuTTY、Roger Meier's CoolTerm。

Done.

P.S. I use free version of avast! anti-virus software. It slows down exectuting make, avr-g++, and avr-gcc. So I need to execude those executable files in the avast! settings. (Before 60 seconds, after 5 seconds.)

參考資料：

- Arduino Official website，Arduino Build Process。
- Arduino Playground（wiki），Arduino from the Command Line，Windows command line build。
- Johannes Hoff: Arduino on the command line。
- Martin's Atelier: Arduino from the command line。
- Wikiid: Command line Arduino。
- AVRDUDE - AVR Downloader/UploaDEr，avrdude's arguments。
- avr-gcc、avr-g++、avr-ar、avr-objdump、avr-objcopy、avr-size。
- Dependency Generation with Subdirectories using gcc。
- GNU make, how to determine a path is a file or a directory?。
- GNU Make Manual。

位於 13:47

標籤： ARDUINO

2 COMMENTS:

. 6/9/13 06:21

thanks for the tutorial.
unfortunately this fails for me at "make depend".
i get "avr-gcc command not found" error.
any clues?

Reply

yehnan 6/10/13 22:58

well, maybe put something like the following line
export PATH=$PATH:/cygdrive/d/Arduino/arduino-1.0.5/hardware/tools/avr/bin
in ~/.bash_profile

Reply

Enter your comment...

Comment as: Select profile... ▼

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

##EASYREADMORE##