W5500

Ethernet Shield S

USER GUIDE

— Preliminary

스타일 정의: 목차 3: 탭: 47.53 글자, 오른쪽,채움선: …

COPYRIGHT

This datasheet is intended for use for FURTHER ENGINEERING, DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY. It is not a finished product and may not (yet) comply with some or any technical or legal requirements that are applicable to finished products, including, without limitation, directive regarding electromagnetic compatibility, recycling (WEEE), FCC, CE or UL (except as may be otherwise noted on the product). eWBM supplied this product "AS IS," without any warranties, with all faults, at the buyer's and further users' sole risk. The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies eWBM from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge and any other technical or legal concerns.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER USER NOR EWBM SHALL BE LIABLE TO EACH OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

No license is granted under any partner right or other intellectual property right of eWBM covering or relating to any machine, process, or combination in which such eWBM products or services might be or are used.

메모 포함[R1]: Must be replaced

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

TABLE OF CONTENTS

서식 있음: 탭: 6 글자(없음)

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

~~PREFACE~~

메모 포함[YJ2]: Preface needs to be updated for W5500 User Guide

~~This datasheet provides reference information for the MS500 microcontroller, describing the functional blocks of the system-on-chip (SoC) device designed around the ARM® Cortex™-M0 core.~~
~~The purpose of this book is to provide working knowledge of the MS500 microcontroller.~~
~~Using this foundation, the reader will be equipped to understand and implement the MS500 microcontroller features.. For better understanding of MS500, the major technical concepts of the MS500 are selected and introduced gradually over several chapters with preactical examples supporting theory.~~

~~AUDIENCE~~

~~This book is intended for system software developers, hardware designers, and application developers.~~

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

DOCUMENT REVISION AND REFERENCE

Revision History

Revision	Date	Description

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

1 Overview

1.1 W5500 Ethernet Shield S

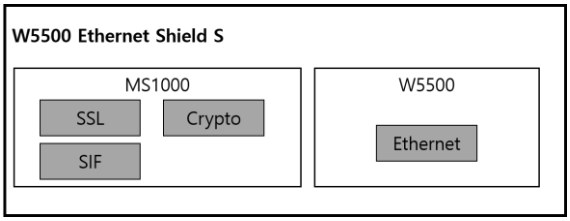
The "W5500 Ethernet Shield S" is ~~the upgraded~~a security enhanced version of the ~~existing~~ "W5500 Ethernet Shield~~,~~" which has been redesigned to include SSL (Secure Sockets Layer) connectivity.

More information on the "W5500 Ethernet Shield" can be found here:
http://wizwiki.net/wiki/doku.php?id=osh:w5500_ethernet_shield:start)

The "W5500 Ethernet Shield S" contains both the W5500 Hardwired TCP/IP chip for network connectivity and the ~~designed using the~~ MS1000 Secure MCU from eWBM for the security features required to make a secure connection~~ and the WIZnet W5500 chip~~. The MS1000's strong security and high speed HW based crypto functions ~~(Visit the link below for further information on the W5500 Ethernet Shield http://wizwiki.net/wiki/doku.php?id=osh:w5500_ethernet_shield:start)~~

~~The W5500 Ethernet Shield S provides all of the existing network functions from the W5500 Ethernet Shield, and specially supports the SSL (Secure Sockets Layer) protocol. It~~ ensure~~s~~ that all ~~the~~ data ~~is passed~~ transferred between the server and a client is protected.

~~In the W5500 Ethernet Shield S, the MS1000 takes the SSL function with the HW security accelerator engine for the best performance.~~



서식 있음: 가운데

This "W5500 Ethernet Shield S" is ~~compatible with the~~ Arduino ~~Platform~~pin-compatible.

1.2 Available Board List

- Arduino Board (e.g. the Uno, Mega etc…)

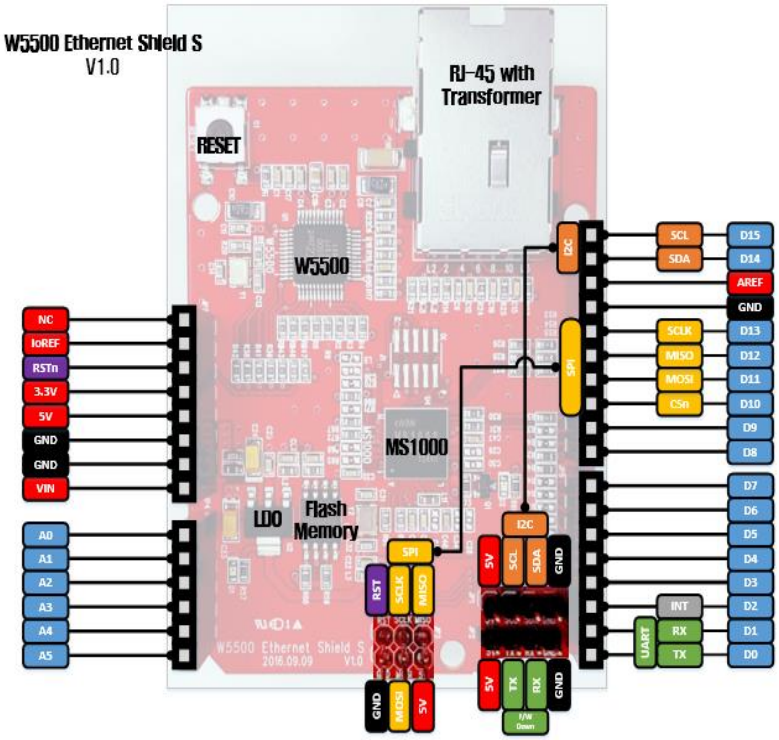
Copyright - © 2016 eWBM Co., Ltd. All rights reserved

2 FEATURES

2.1 HARDWARE FEATURES

- Supports 3.3V
- ARM® Cortex-M3™ MCU with HW Crypto engine (MS1000)
- High Speed Ethernet controller (W5500)
- 10/100 Ethernet PHY embedded.
- Hardwired TCP/IP Protocols: TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE.
- Supports SPI, I2C, UART interface

Copyright - © 2016 eWBM Co., Ltd. All rights reserved



2.2 Hardware Configuration

- MS1000: ARM® Cortex-M3™~~TM~~ based microcontroller with HW crypto engine.
- W5500: Hardwired TCP/IP Ethernet Controller
- RJ-45 with Transformer: Ethernet Port
- I2C: I2C interface
- UART: UART interface
- SPI: SPI Interface

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

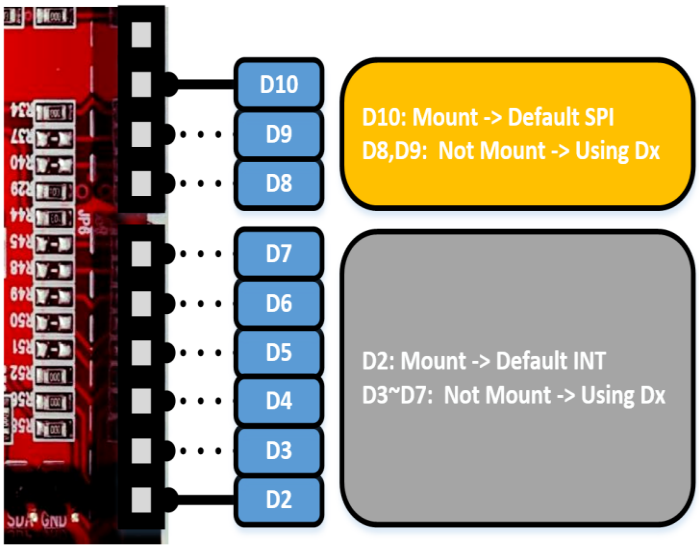


Figure 1 Pins usage on Arduino

2.3 Software Features

- Supports SSL/TLS 1.2
- SSL Specification

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

Category	Description	Comment
Cipher Suit **- Public Key Algorithm**	RSA ECC	TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_128_CCM_8 TLS_RSA_WITH_AES_256_CCM_8 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8
Cipher Suit **- Block/Stream Ciphers**	AES CCM GCM CBC CTR ECB	
Cipher Suit **- Hash Functions**	SHA1 SHA256	
Side of Connection	Client only	
Client Authentication	APIs support	CA certificate load, Certificate/Private Key load

3 SPI OPERATION

3.1 OVERALL SPI INTERFACE

-The SPI interface of the "W5500 Ethernet Shield S" supports a ~~up to a~~clock speed of up to 4MHz in ~~the~~ slave

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

mode.

Function	Interface		GPIO
W5500 Ethernet Shield S	SPI	SCK	PC0
		SSN	PC1
		MISO	PC2
		MOSI	PC3
		OUT_INT	PD6

3.2 SPI TIMING

DATA MODE

The~~re are four combinations of the~~ ~~SCK~~ phase and ~~the~~ polarity of the serial clock (SCK) must be set properly with respect to the serial data. There are 4 combinations which can be selected ~~, which are determined by~~ by the control bits~~:~~ CPHA and CPOL.

Data bits are shifted out and latched in on the opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize

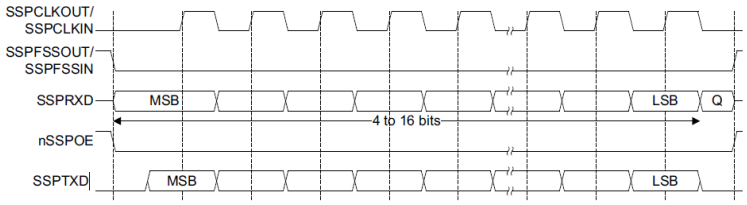
메모 포함[R3]: Why is this here... Get JS to verify if it is correct

- By default, the "W5500 Ethernet Shield S" is set to CPOL = 1, CPHA = 1

CPOL and CPHA Functionality

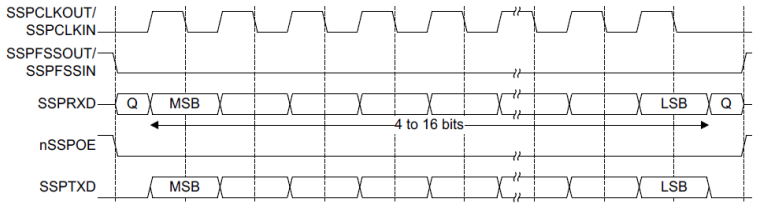
	Leading Edge	Trailing Edge	SPI Mode
CPOL = 0, CPHA = 0	↑		0
CPOL = 0, CPHA = 1		↓	1
CPOL = 1, CPHA = 0	↓		2
CPOL = 1, CPHA = 1		↑	3

SPI Transfer format with CPOL =0, CPHA = 0

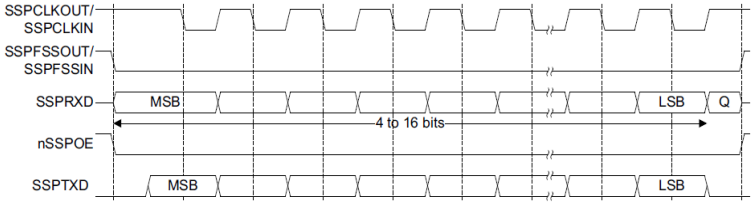


SPI Transfer format with CPOL =0, CPHA = 1

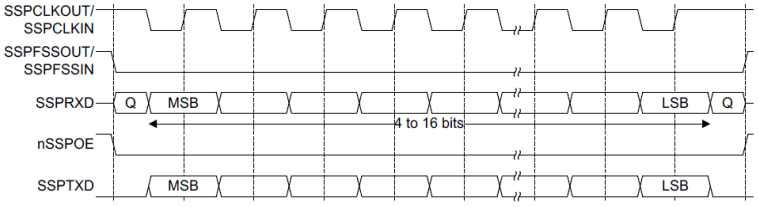
Copyright - © 2016 eWBM Co., Ltd. All rights reserved



SPI Transfer format with CPOL =1, CPHA = 0



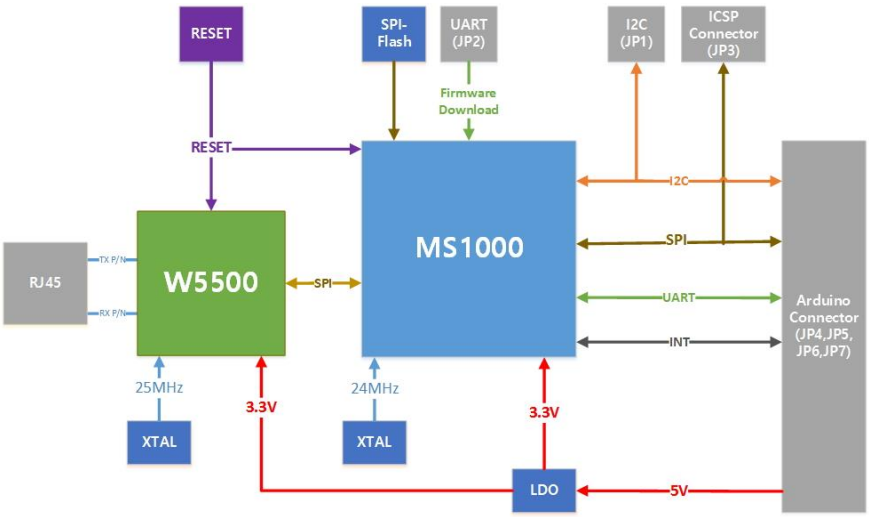
SPI Transfer format with CPOL =1, CPHA = 1



Copyright - © 2016 eWBM Co., Ltd. All rights reserved

4 Technical Reference

4.1 Block Diagram



Copyright - © 2016 eWBM Co., Ltd. All rights reserved

4.2 SCHEMATICS

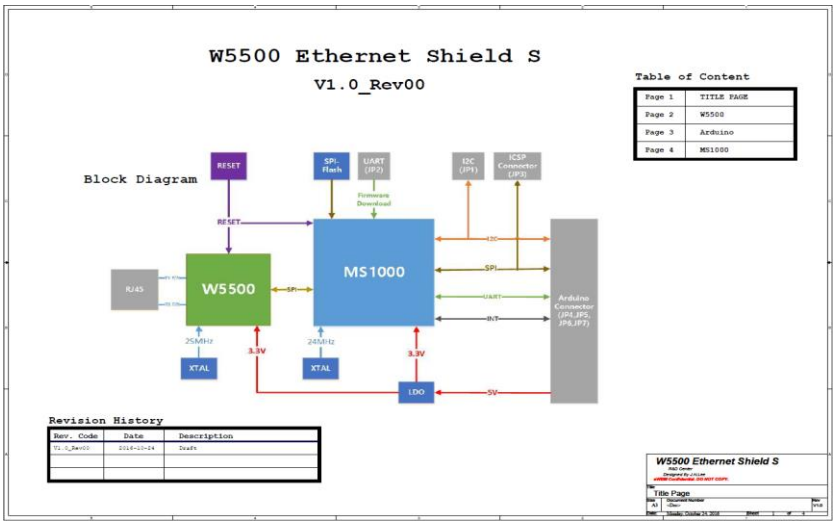


Figure 2 W5500 Ethernet Shield S Schematic (1)

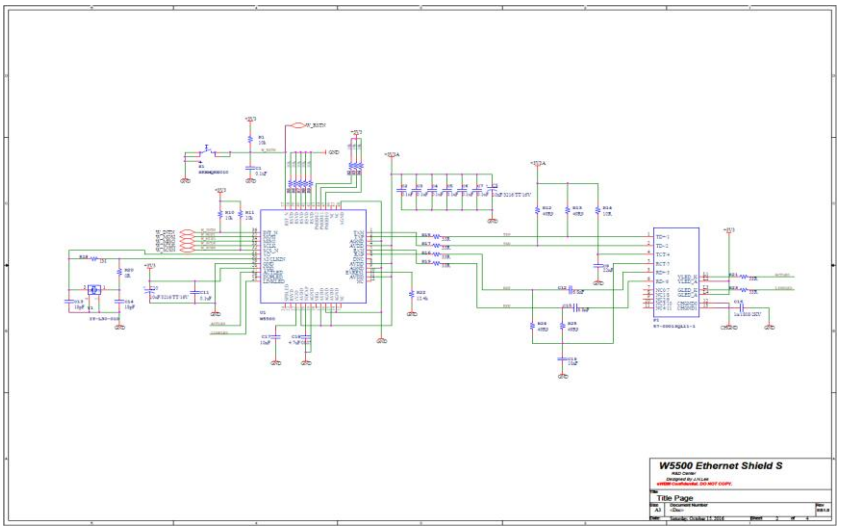


Figure 3 W5500 Ethernet Shield S Schematic (2)

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

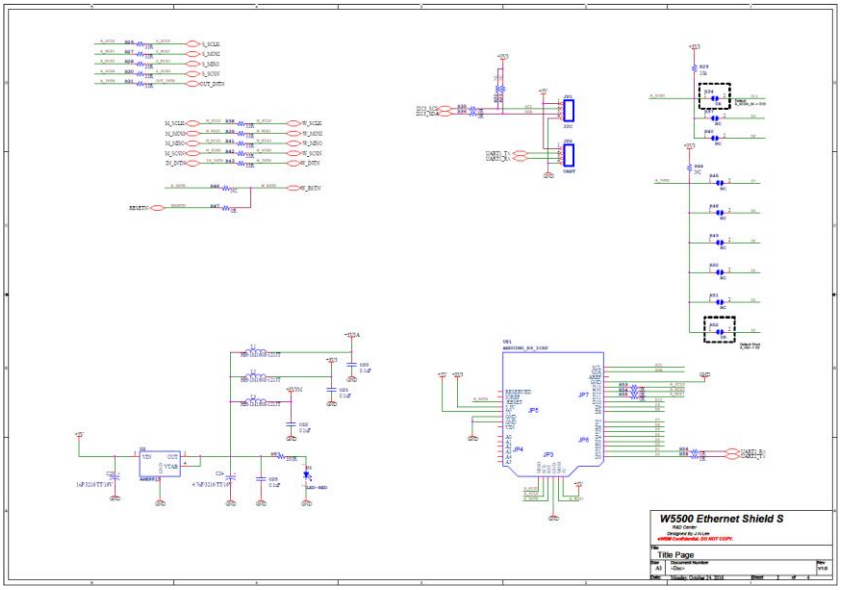


Figure 4 W5500 Ethernet Shield S Schematic (3)

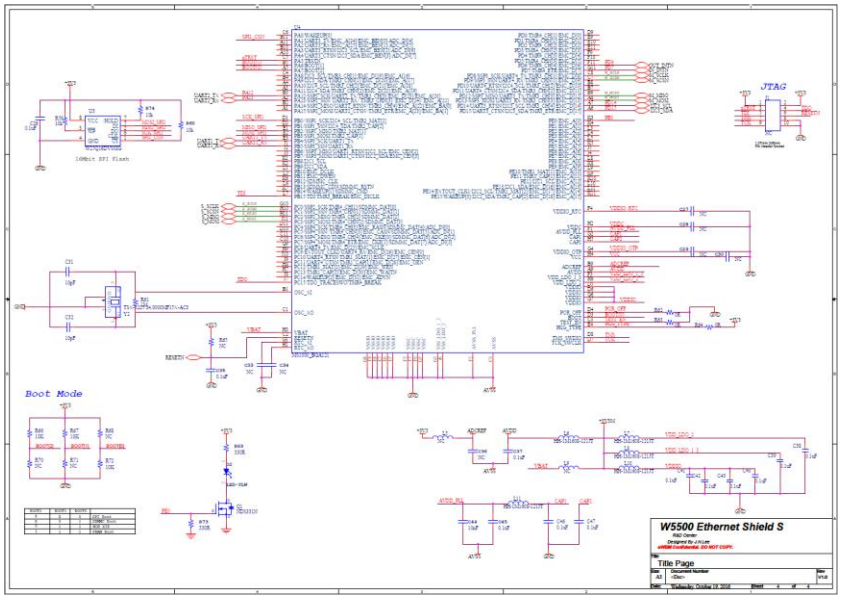


Figure 5 W5500 Ethernet Shield S Schematic (4)

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

4.3 DIMENSIONS

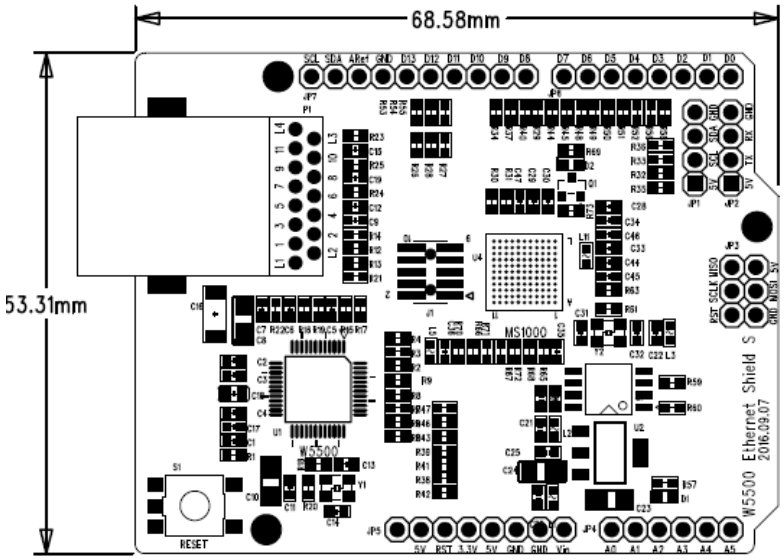
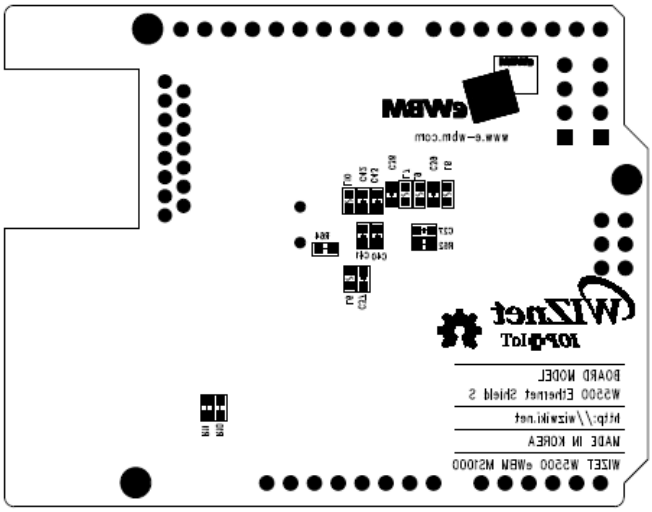


Figure 6 W5500 Ethernet Shield S Dimension(Top Side)

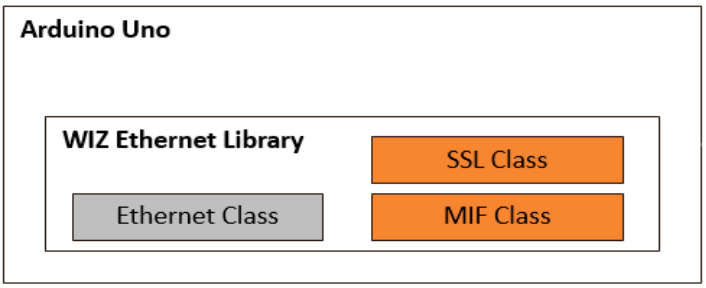


W5500 Ethernet Shield S Dimension

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

5 Getting Started

5.1 Using the WIZ Ethernet Library for Arduino Uno



Class	Description
Ethernet Class	Included Class to the Wiz Ethernet library to support internet in Arduino Uno Refers to the Arduino Ethernet library and API Guide at the follow site. WIZ Ethernet Library: https://github.com/Wiznet/WIZ_Ethernet_Library Arduino Ethernet API: https://www.arduino.cc/en/Reference/Ethernet
SSL Class	Added Class to the Wiz Ethernet library to support SSL in the Arduino
MIF Class	Added Class to the Wiz Ethernet library to communicate with W5500 Ethernet Shield S in the Arduino

5.1.1 DESCRIPTION OF ADDED SSL CLASS AND MIF CLASS

The SSL client performs the following: SSL initialize, connect to the server, and send/receive data.

- Only SSL Client operation (does not work as an SSL Server)
- USE_MS1000_MIF feature is a function for SSL client only on W5500
- When USE_MS1000_MIF feature is Disable, SSL client does not work

5.1.2 API REFERENCE OF SSL CLASS AND MIF CLASS

- SSL CLASS

Open()	
Description	Open of SSL Socket
Syntax	SSLClient.Open()
Parameters	None
Returns	If successful the call will return SSL_SUCCESS

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

Close()	
Description	Close of SSL Socket
Syntax	SSLClient.Close()
Parameters	None
Returns	If successful the call will return SSL_SUCCESS

Connect()	
Description	This function is called on the client side and initiates an SSL/TLS handshake with a server
Syntax	SSLClient.Connect(ip, port) SSLClient.Connect(hostname, port)
Parameters	Ip: connecting to domain ip address hostname: connecting to hostname (ex: www.google.com) port: SSL port
Returns	If successful the call will return SSL_SUCCESS

WriteData()	
Description	This function writes sz bytes from the buffer, data, to the SSL connection, ssl
Syntax	SSLClient.WriteData()
Parameters	buf: data buffer which will be sent to peer size: size, in bytes, of data to send to the peer IsPMEM: the generating data to the Flash (Program) instead of SRAM memory
Returns	If successful the call will return SSL_SUCCESS

ReadData()	
Description	This function reads sz bytes from the SSL session (ssl) internal read buffer into the buffer data. The bytes read are removed from the internal receive buffer.
Syntax	SSLClient.ReadData()
Parameters	buf: data buffer which will be read to peer size: number of bytes to read into data. readsz: getting read size
Returns	If successful the call will return SSL_SUCCESS

SetPeerVerify()	
Description	This function sets the verification method for remote peers and allows a verify callback to be registered with the SSL session. The verify callback will be called only when a verification failure has occurred. If no verify callback is desired, the NULL pointer can be used for verify_callback
Syntax	SSLClient.SetPeerVerify()
Parameters	verify: enable verify
Returns	If successful the call will return SSL_SUCCESS

SetRootCA()	
Description	This function sets a CA certificate buffer into the SSL. It behaves like the non buffered version, only differing in its ability to be called with a

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

	buffer as input instead of a file.
Syntax	SSLClient.SetRootCA()
Parameters	buf: the CA certificate buffer len: size of the input CA certificate buffer IsPMEM: the generating data to the Flash (Program) instead of SRAM memory
Returns	If successful the call will return SSL_SUCCESS

GetVersion()	
Description	This function gets the SSL/TLS protocol version for the specified SSL session using the version as specified by version.
Syntax	SSLClient.GetVersion()
Parameters	buf: the version information buffer len: length of buf
Returns	If successful the call will return SSL_SUCCESS

GetCipherName()	
Description	Retrieves the peer's certificate cipher name
Syntax	SSLClient.GetCipherName()
Parameters	buf: the cipher name buffer len: length of buf
Returns	If successful the call will return SSL_SUCCESS

GetX509IssuerName()	
Description	Retrieves the peer's certificate issuer name
Syntax	SSLClient.GetX509IssuerName
Parameters	buf: the issuer name buffer len: length of buf
Returns	If successful the call will return SSL_SUCCESS

GetX509SubjectName()	
Description	Retrieves the peer's certificate subject name
Syntax	SSLClient.GetX509SubjectName
Parameters	buf: the subject name buffer len: length of buf
Returns	If successful the call will return. SSL_SUCCESS

GetX509NextAltName()	
Description	Retrieves the peer's certificate next altname
Syntax	SSLClient.GetX509NextAltName
Parameters	buf: the next altname buffer len: length of buf
Returns	If successful the call will return SSL_SUCCESS

GetX509SerialNum()	
Description	Retrieves the peer's certificate serial number
Syntax	SSLClient.GetX509SerialNum()
Parameters	buf: the serial number buffer

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

	len: length of buf OutNumSz: getting a length of serial number
Returns	If successful the call will return SSL_SUCCESS

SetDate	
Description	This function sets a date.
Syntax	SSLClient.SetDate()
Parameters	buf: the date buffer len: length of buf
Returns	None

SetTime	
Description	This function sets a time.
Syntax	SSLClient.SetTime()
Parameters	buf: the date buffer len: length of buf
Returns	None

- MIF CLASS

Write()	
Description	This function writes 1 byte to Slave
Syntax	gMIFInfo.Write()
Parameters	w: to write data
Returns	None

Read()	
Description	This function reads 1 byte from Slave
Syntax	gMIFInfo.Read()
Parameters	None
Returns	Read data

WaitCmd()	
Description	This function waits 1 byte command for send to Slave
Syntax	gMIFInfo.WaitCmd()
Parameters	waitcmd: 1byte command
Returns	If successful the call will return 0

StartCmd()	
Description	This function sends 1 byte command to Slave
Syntax	gMIFInfo.StartCmd()
Parameters	cmd: 1byte command ctrlb: distinguish from read commad and write command datalen: read/write data length
Returns	If successful the call will return 0

EndCmd()	
Description	This function indicates the end of command

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

Syntax	gMIFInfo.EndCmd()
Parameters	None
Returns	If successful the call will return 0

Init()	
Description	MIF Class Initialize
Syntax	gMIFInfo.Init()
Parameters	None
Returns	None

WriteData()	
Description	This function writes sz bytes from the buffer, data, to Slave
Syntax	gMIFInfo.WriteData()
Parameters	addr: SSL command set ctrlb: distinguish from read commad and write command pWBuf: data buffer which will be sent to slave len: data buffer size IsPMEM: the generating data to the Flash (Program) instead of SRAM memory
Returns	If successful the call will return 0

ReadData()	
Description	This function reads sz bytes from the Slave.
Syntax	gMIFInfo. ReadData()
Parameters	addr: SSL command set ctrlb: distinguish from read commad and write command pRBuf: data buffer which will be read to slave len: data buffer size
Returns	If successful the call will return 0

IsReady()	
Description	Check the MIF initialize
Syntax	gMIFInfo.IsReady()
Parameters	None
Returns	If MIF initialize successful, the call will return true

5.2 INSTALLING THE ARDUINO SOFTWARE

5.2.1 DOWNLOAD IDE

Download the Arduino IDE ~~at~~ from the Arduino site:

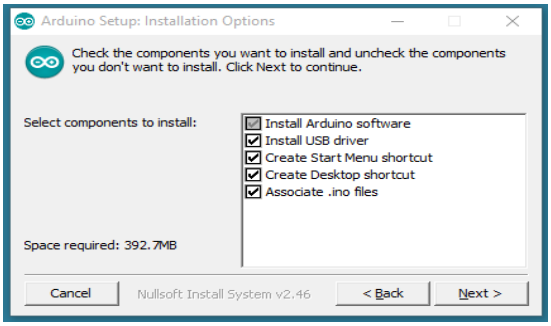
https://www.arduino.cc/en/Main/Software

5.2.2 INSTALL IDE

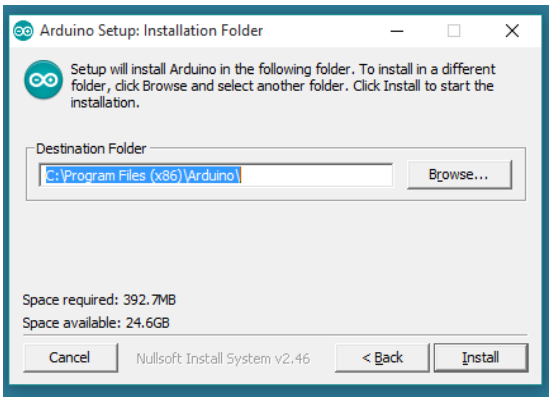
Copyright - © 2016 eWBM Co., Ltd. All rights reserved

When the download completes, proceed with the installation. Please allow the driver installation process.

Step 1: Choose the components to install

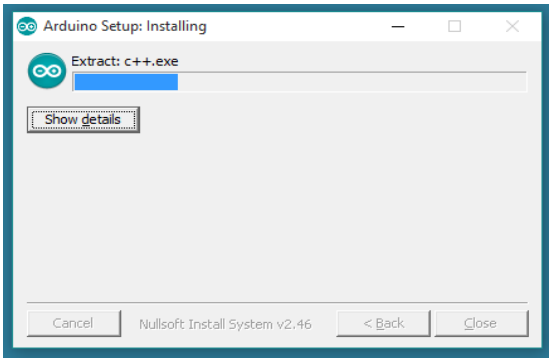


Step 2: Choose the installation directory (it is recommended to keep the default)



Step 3: The process will extract and install all of the required files ~~to properly execute~~for the Arduino ~~Sortware~~ Software (IDE)

Copyright - © 2016 eWBM Co., Ltd. All rights reserved



5.2.3 LAUNCH THE IDE

Double-click the Arduino icon (arduino.exe) created by the installation process.
Open the blink example.

5.3 WIZNET LIBRARY UPDATE

5.3.1 GETTING FOR W5500 ETHERNET SHIELD S LIBRARY

Step 1: ~~Getting the released~~Download the W5500 Ethernet Shield S ~~source~~library from~~.~~:

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

- ~~Base source code~~
  https://github.com/Wiznet/WIZ_Ethernet_Library

서식 있음: Standard, 들여쓰기: 왼쪽: 1.25 cm, 오른쪽: 0 cm

메모 포함[R4]: Need to put the real address of library here

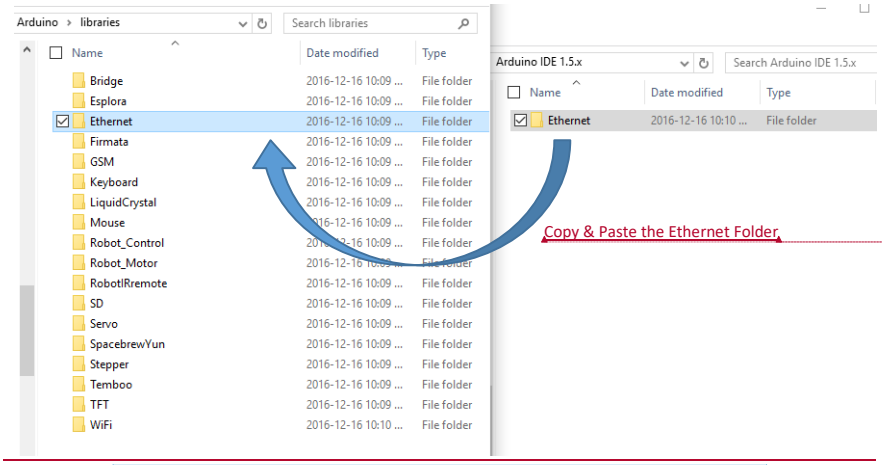
서식 있음: 들여쓰기: 왼쪽: 2.66 cm, 첫 줄: 0.16 cm

5.3.2 LIBRARY UPDATE

Step 1: Unzip the ZIP file

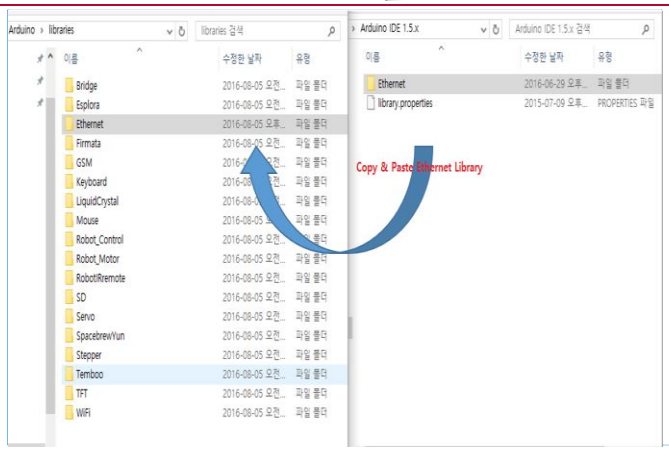
Step 2: Go into "C:\Program Files\Arduino\libraries"

Step 3: Copy & Paste ~~'Arduino~~ "Arduino IDE 1.5.x ~~Folder →~~ \Ethernet\ ~~→ src~~" folder to "C:\Program Files\Arduino\library\Ethernet" folder.



서식 있음: 글꼴: 9 pt, 글꼴 색: 빨강

서식 있음: 글꼴: 9 pt, 글꼴 색: 빨강, 영어(캐나다)



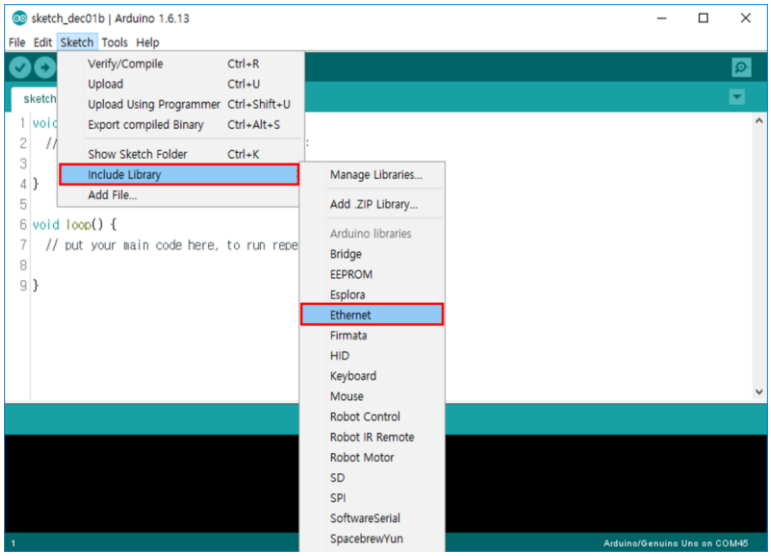
메모 포함[YJ5]: This picture still has Korean

Copyright - © 2016 eWBM Co., Ltd. All rights reserved

5.3.3 LIBRARY IMPORT

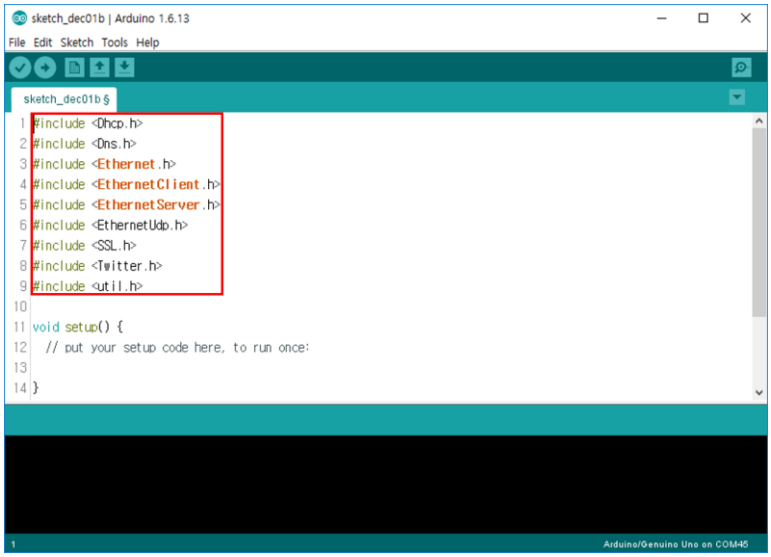
Step 1: Use Arduino IDE by importing the ~~L~~library

Step 2: To use the library of Arduino Ethernet shield, add the header files by selecting "~~Import~~ Include Library > Ethernet" ~~of~~ under the Sketch ~~menu~~tab



▶ Add a header file

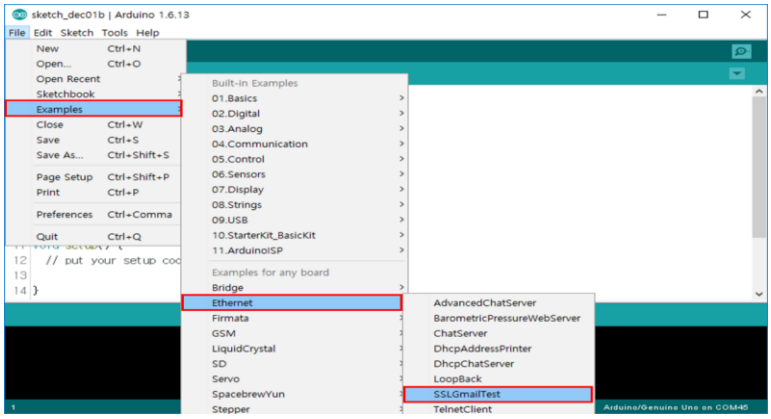
Copyright - © 2016 eWBM Co., Ltd. All rights reserved



5.4 Arduino Example

5.4.1 START ARDUINO (SSL EXAMPLE)

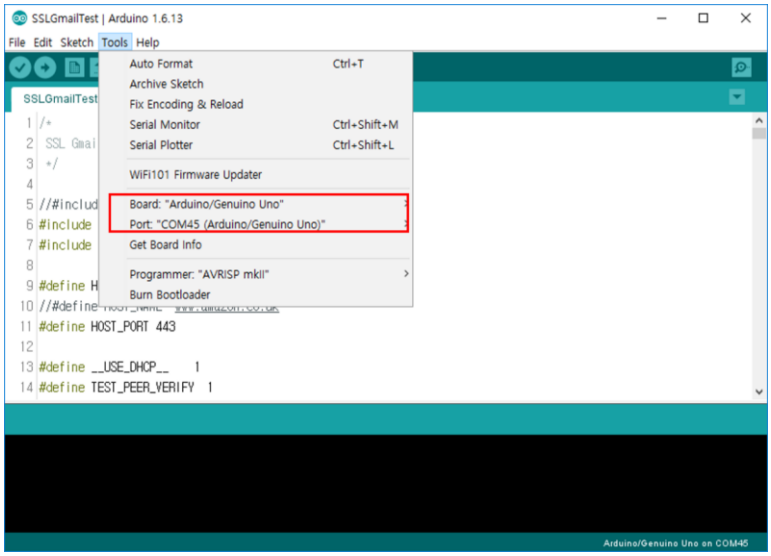
Step 1: Run Arduino
Step 2: under the "File" tab ~~Select~~ select "Example -> Ethernet -> SSLGmailTest"



Copyright - © 2016 eWBM Co., Ltd. All rights reserved

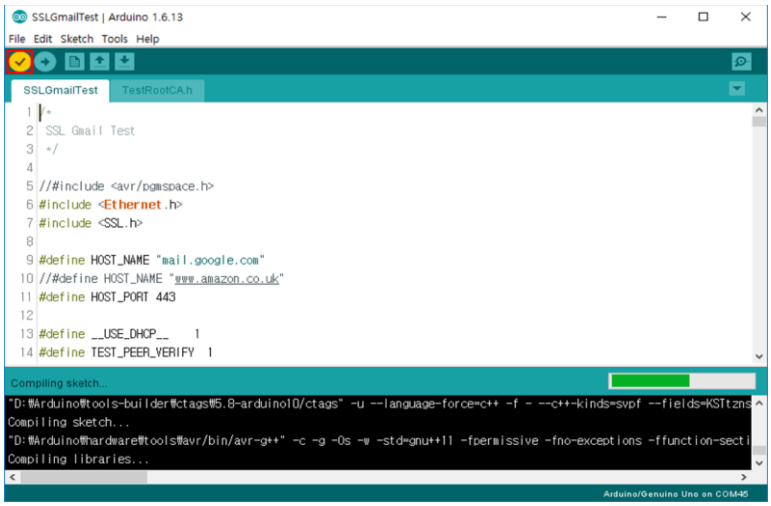
서식 있음: 다음 단락과의 사이에 페이지 나누지 않음, 현재 단락을 나누지 않음

Step 3: A specific setting is needed based on the board type such as Uno, Mega, Due... etc
Tool -> board -> Arduino Uno
Tool -> port -> Comx



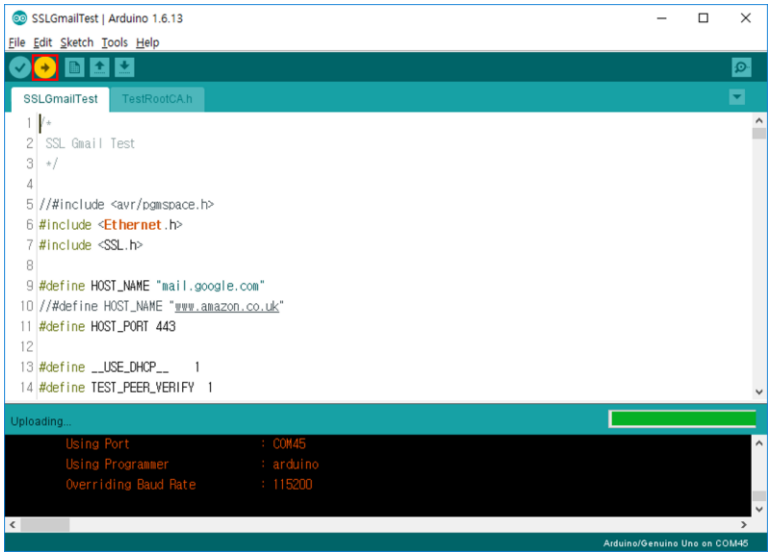
5.4.2 OPERATION ARDUINO

Step 1: Click “Verify” to check for code error



Copyright - © 2016 eWBM Co., Ltd. All rights reserved

Step 2: Then, click “Upload” to upload onto the Arduino board

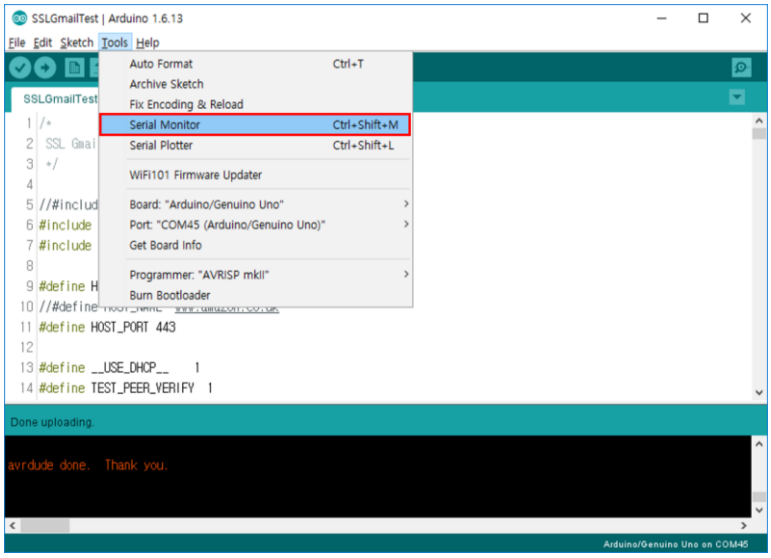


Step 3: Click “Serial Monitor” when “Upload” is completed

▶ Flashing

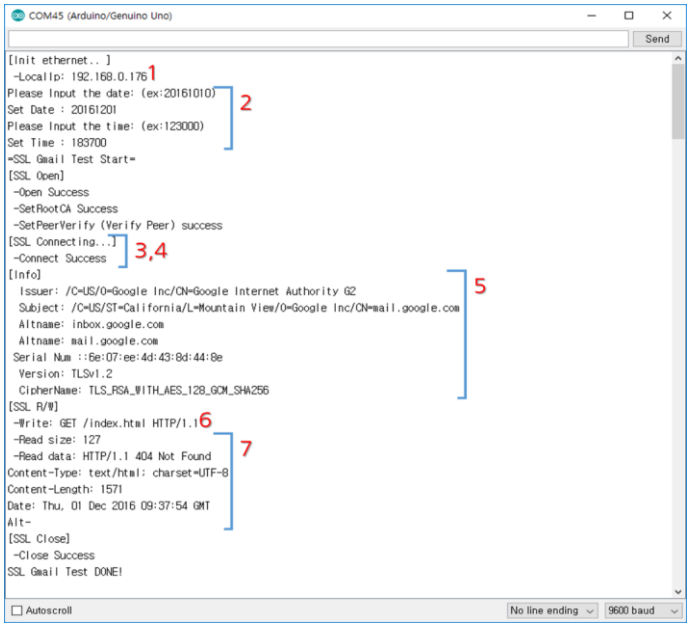
Copyright - © 2016 eWBM Co., Ltd. All rights reserved

▶ Starting "Serial Monitor"



Step 4: Result of Gmail Test

Copyright - © 2016 eWBM Co., Ltd. All rights reserved



Description:

1) DHCP Initialize and Network Configuration (Allocate IP address)
2) Enters a date and time.
3) Receives Gmail IP via DNS SERVER
4) Connects the Gmail server
5) Receives peer info (issuer/subject/altname/serial number)
6) Sends data to SSL.
7) Receives data from server (SSL Version/Cipher Suite/Content type/Content -Length)

Copyright - © 2016 eWBM Co., Ltd. All rights reserved