W5500

Ethernet Shield S

USER GUIDE

— Release 1.5

Table of Contents

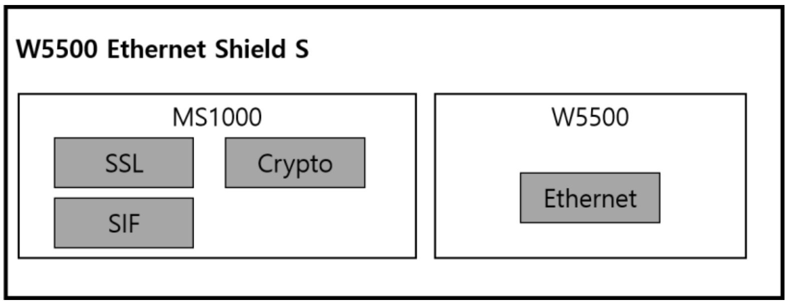
1 Overview

1.1 W5500 Ethernet Shield S

The "W5500 Ethernet Shield S" is a security enhanced version of the "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 MS1000 Secure MCU from eWBM for the security features required to make a secure connection. The MS1000's strong security and high speed HW based crypto functions ensure that all data transferred between the server and a client is protected.



This "W5500 Ethernet Shield S" is Arduino pin-compatible.

1.2 Board Compatibility List

- Arduino UNO (ATmega328P)
- Arduino Mega (ATmega2560)

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

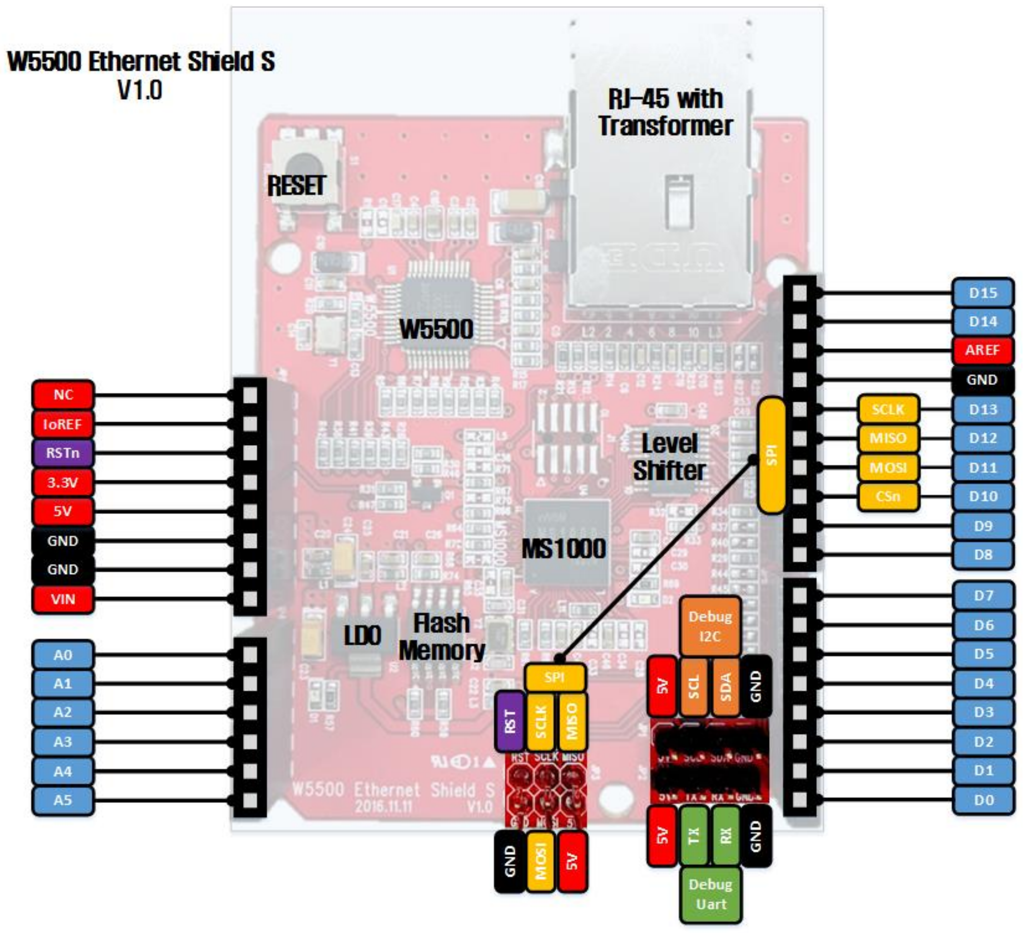


Figure 1 Pin Assignment on Arduino

2.2 HARDWARE CONFIGURATION

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

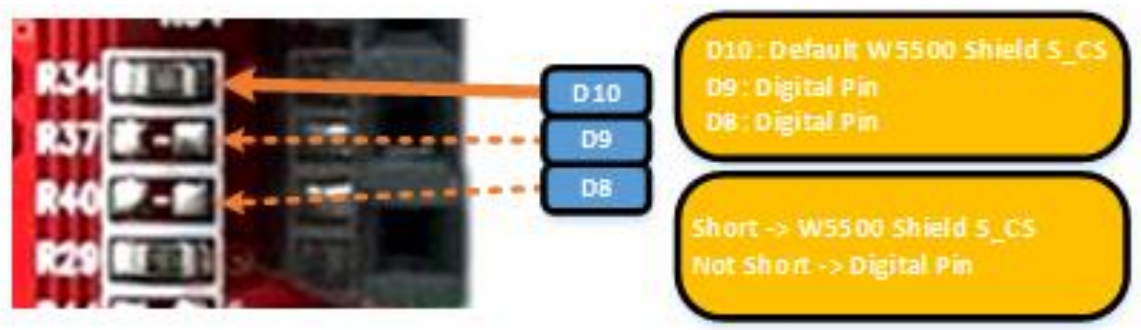


Figure 2 Pin Assignment on Arduino

- To use the W5500 Ethernet Shield S with other modules, you may need to change the 'Chip Select' (S_CS) pin to either D8 or D9.

2.3 Software Features

- W5500 Ethernet Shield S supports SSL/TLS 1.2
- The following table is a list of the support SSL features:

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 Technical Reference

3.1 Block Diagram

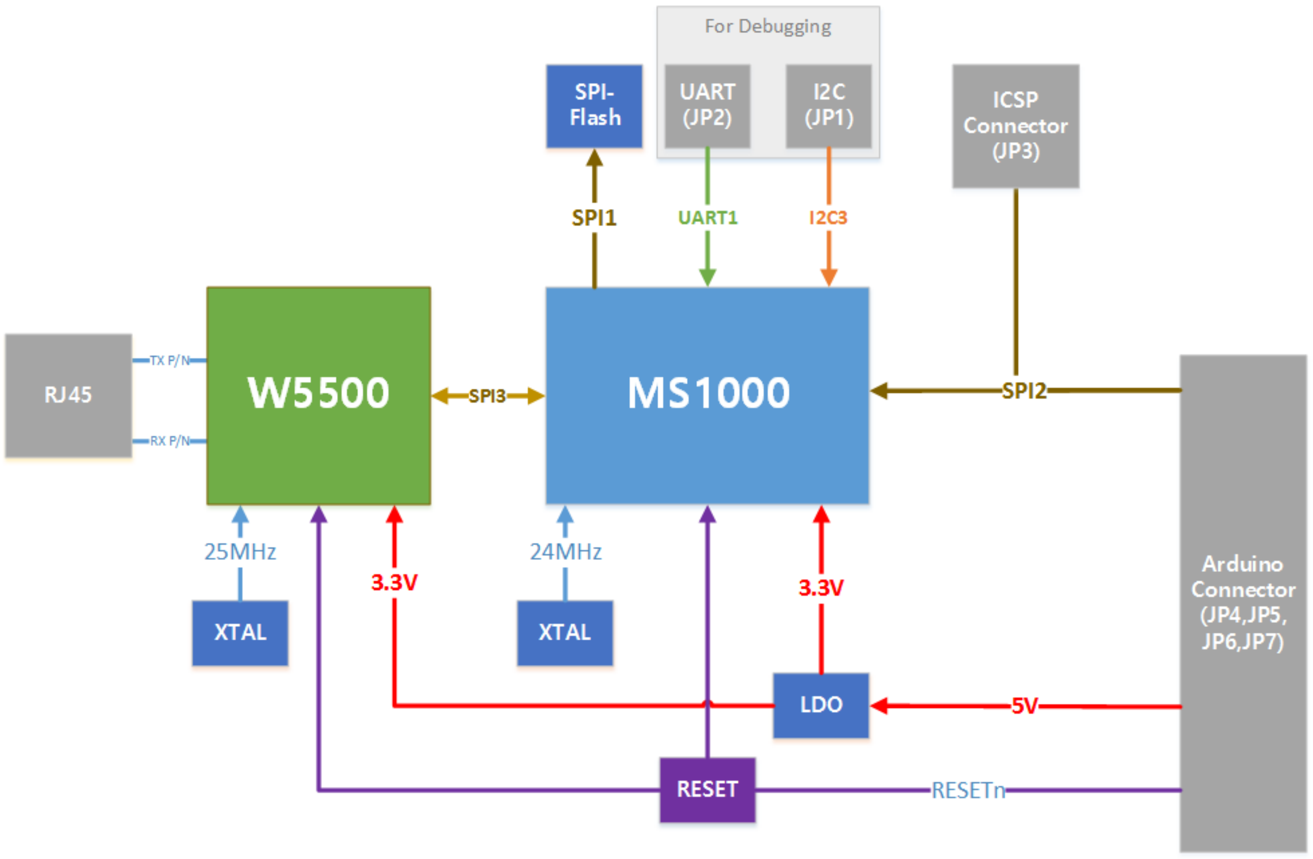


Figure 3 W5500 Ethernet Shield S Block Diagram

3.2 SCHEMATICS

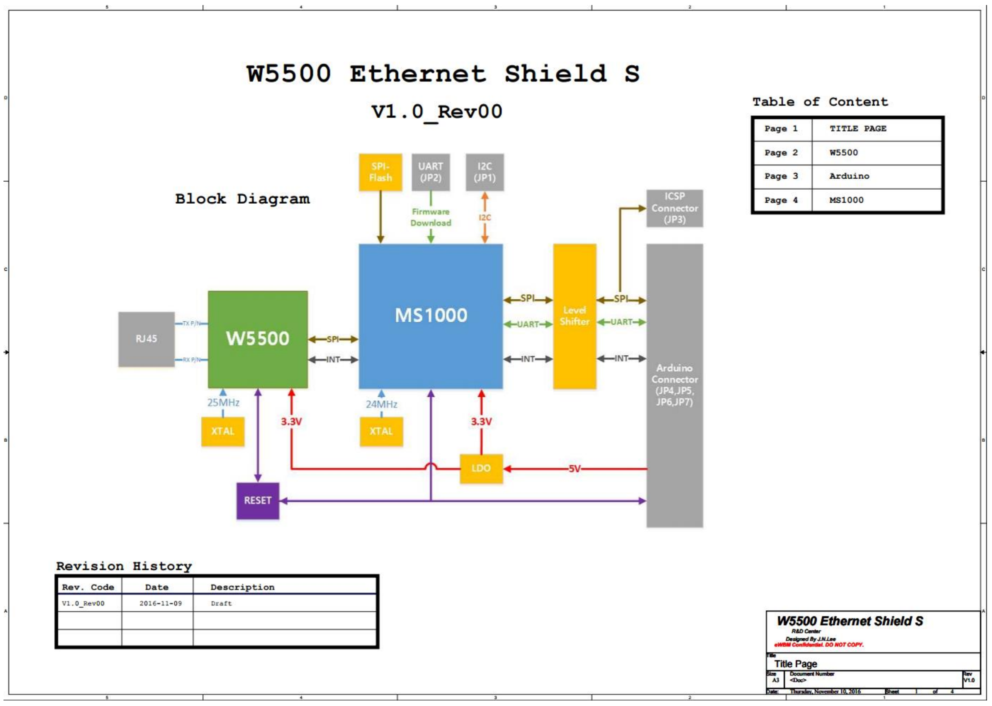


Figure 4 W5500 Ethernet Shield S Schematic (1)

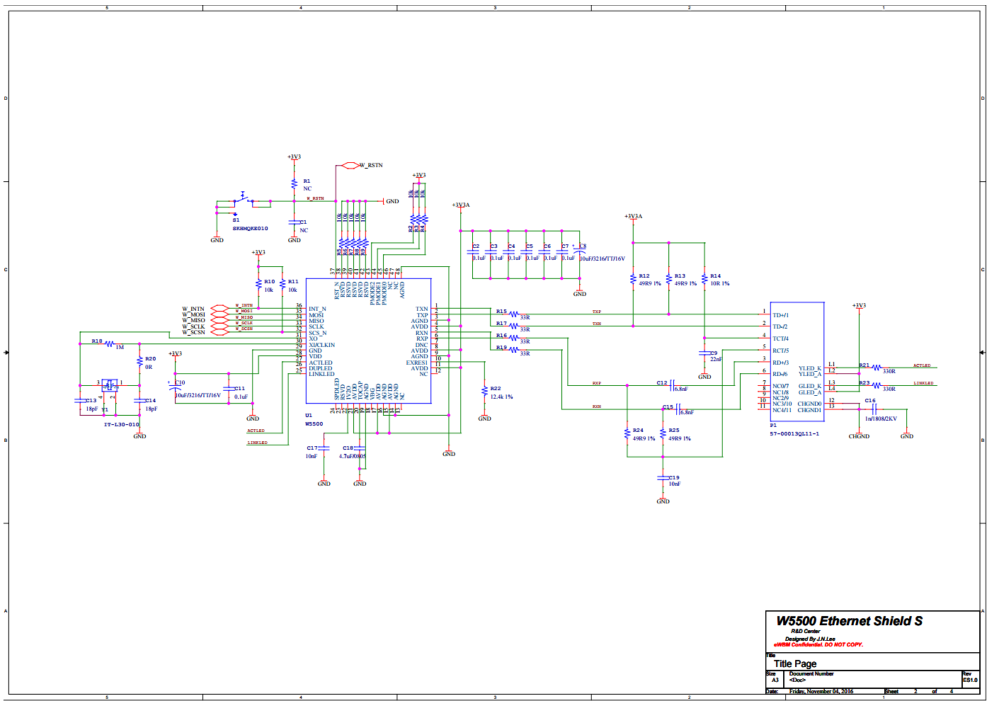


Figure 5 W5500 Ethernet Shield S Schematic (2)

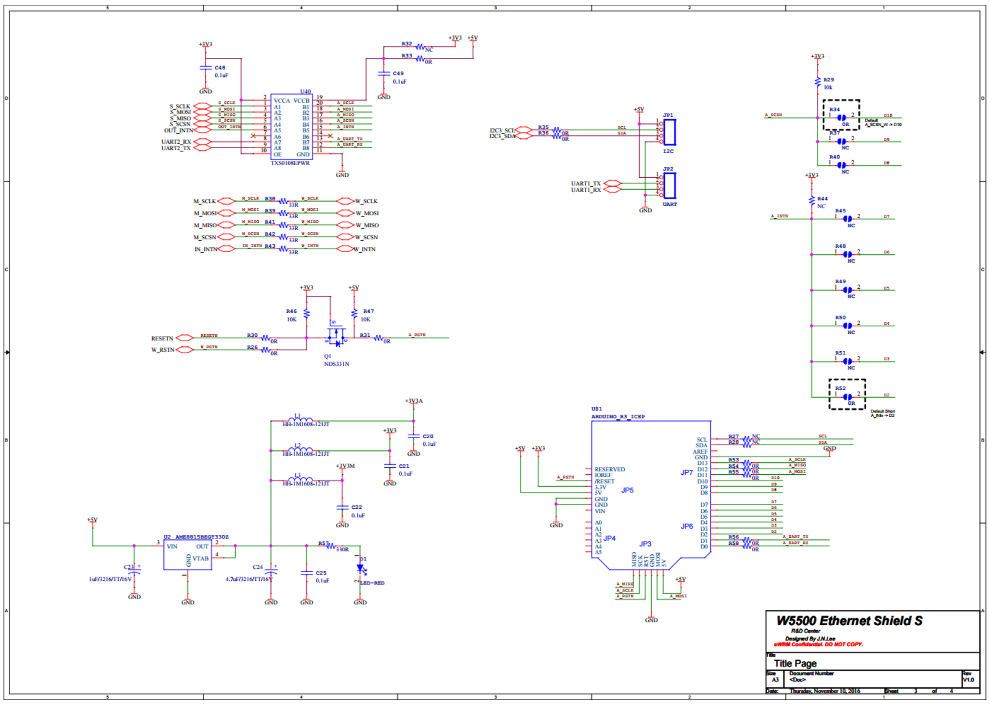


Figure 6 W5500 Ethernet Shield S Schematic (3)

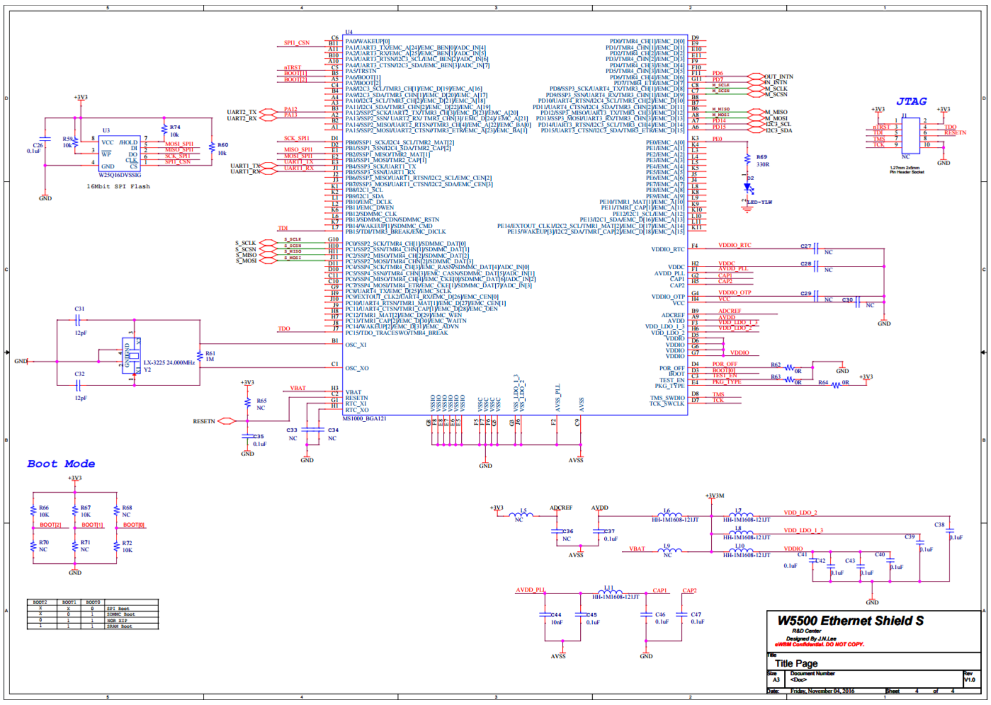


Figure 7 W5500 Ethernet Shield S Schematic (4)

3.3 DIMENSIONS

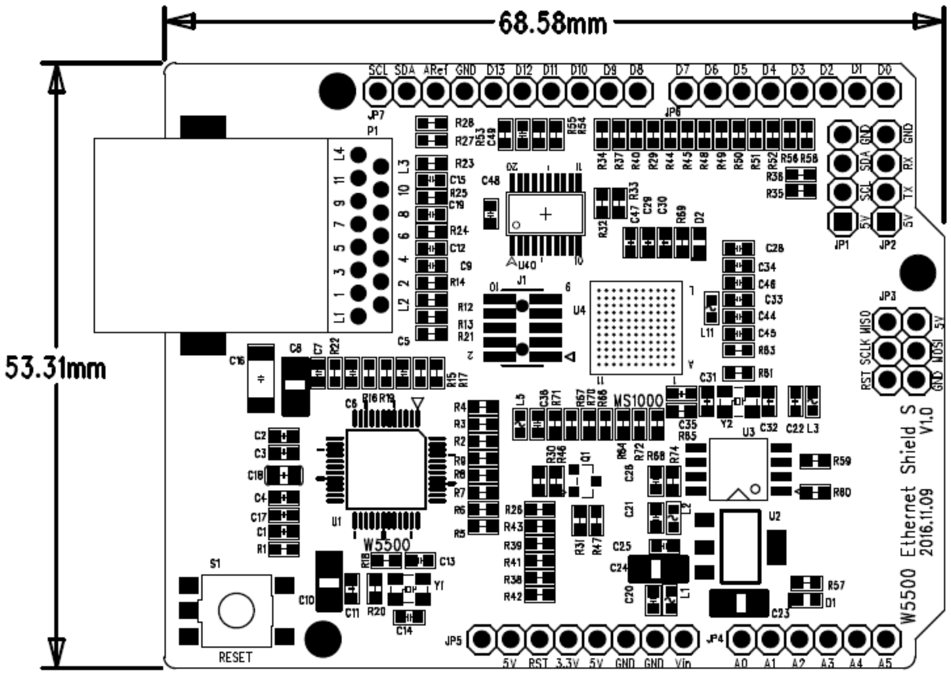


Figure 8 W5500 Ethernet Shield S (Top Side)

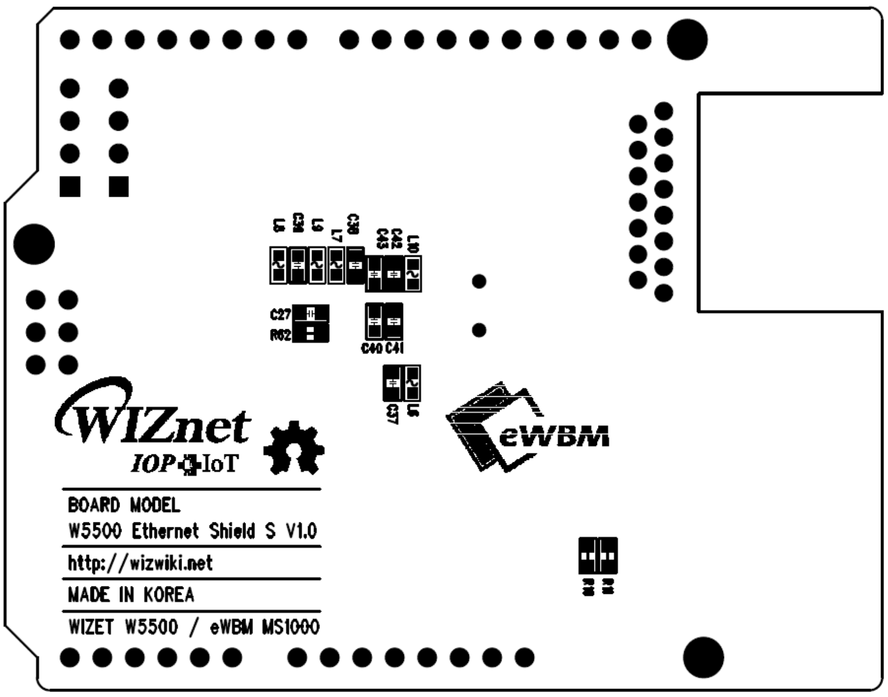
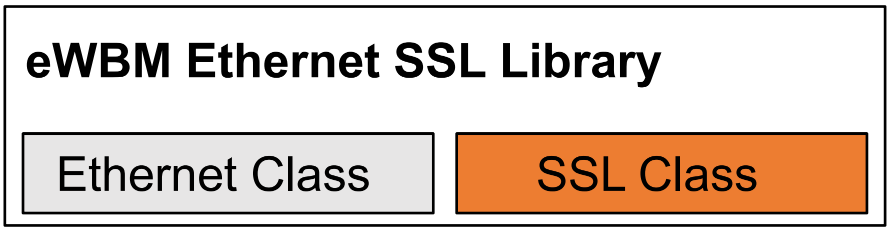


Figure 9 W5500 Ethernet Shield S (Bottom Side)

4 GETTING STARTED

4.1 USING THE ETHERNET SSL LIBRARY FOR ARDUINO UNO



Class	Description
Ethernet Class	Wiz Ethernet library which provides internet connectivity for Arduino boards. For more information on the WIZ Ethernet Library go to: https://github.com/Wiznet/WIZ_Ethernet_Library For the API Guide go to: https://www.arduino.cc/en/Reference/Ethernet
SSL Class	eWBM SSL class which provides SSL connectivity for Arduino boards.

4.1.1 DESCRIPTION OF SSL CLASS

The SSL Class performs the following functions:

- SSL initialize
- Connect to the server
- Send/receive data.

* Notes: eWBM SSL Class only provides SSL Client operation. SSL Server capability is not supported.

4.1.2 SSL CLASS API REFERENCE

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

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 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 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 time buffer len: length of buf
Returns	None

4.2 START GUIDE

4.2.1 INSTALLING THE AUDRINO SOFTWARE (IDE)

Download and install the Arduino Software (IDE) following the instructions on the Arduino website:

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

4.2.2 IMPORTING THE WIZNET ETHERNET SHIELD S LIBRARY

Step 1: Download the W5500 Ethernet Shield S library (EthernetSSL.zip) from:

https://github.com/eWBM/EthernetSSL-library

Step 2: Import the "*EthernetSSL*" library using the .ZIP file by following the instructions on the Arduino website:

https://www.arduino.cc/en/Guide/Libraries

Please refer to "Importing a .zip Library" section.

Notes: If the IDE already contains "Ethernet" library, it must be removed before importing "EthernetSSL".

Step 3: Select the "*EthernetSSL*" under the "Sketch" tab:

"Include Library -> EthernetSSL"

Step 4: After Step 3, "*EthernetSSL*" header files are inserted in the source code automatically by the Arduino IDE.

```
#include <Dhcp.h>
#include <Dns.h>
#include <Ethernet.h>
#include <EthernetClient.h>
#include <EthernetServer.h>
#include <EthernetUdp.h>
#include <SSL.h>
#include <Twitter.h>
#include <util.h>
```

Step 5: *EthernetSSL* library is now ready to be used within the Arduino IDE. The zip file will have been expanded in the libraries folder in the Arduino sketches directory.

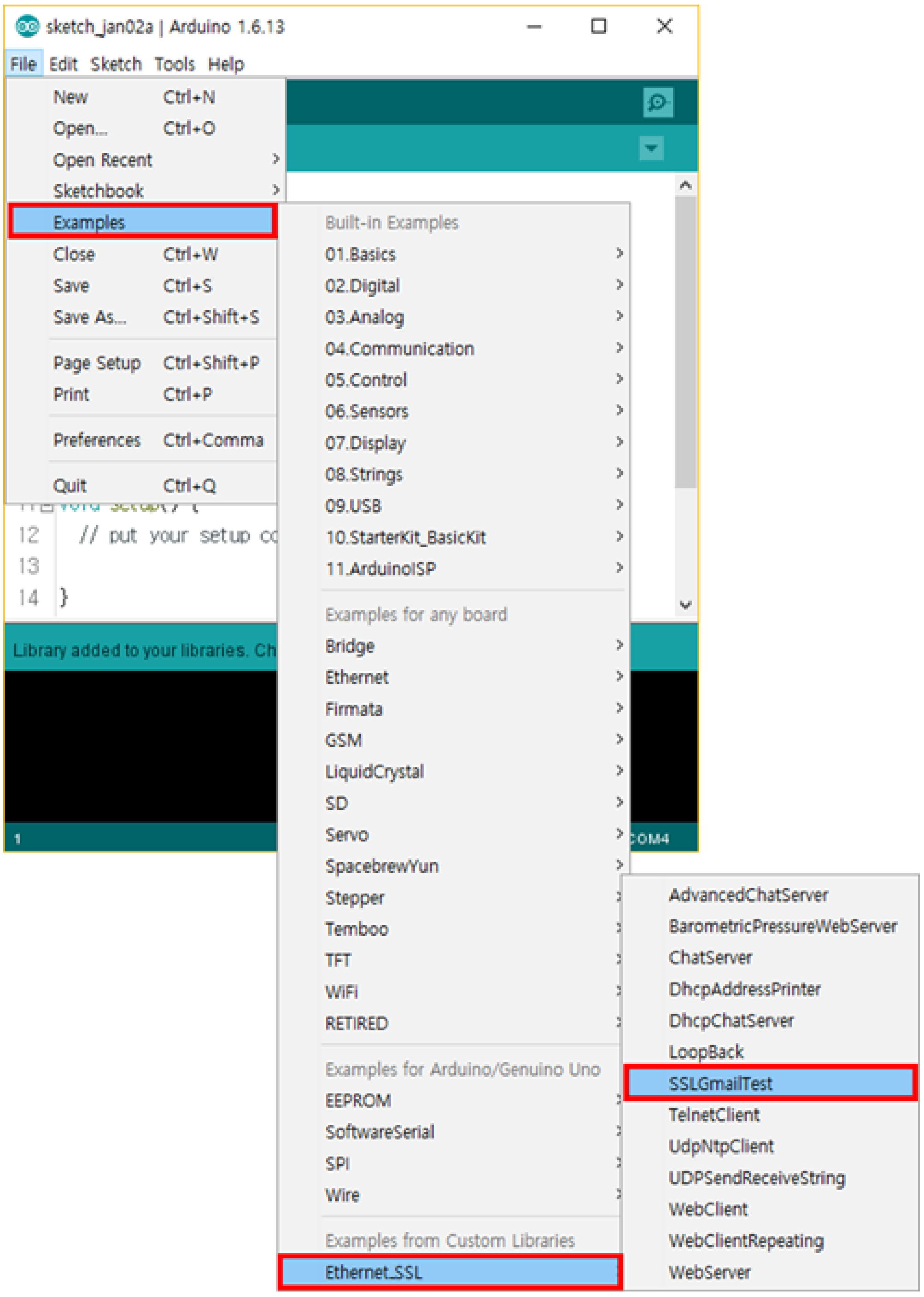
(Default: C:\Users\<User Name>\Documents\Arduino\libraries\EthernetSSL)

4.2.3 STARTING THE WIZNET ETHERNET SHIELD S SSL EXAMPLE

Step 1: Open the Arduino IDE

Step 2: Select the SSL Gmail Test under the "File" tab:

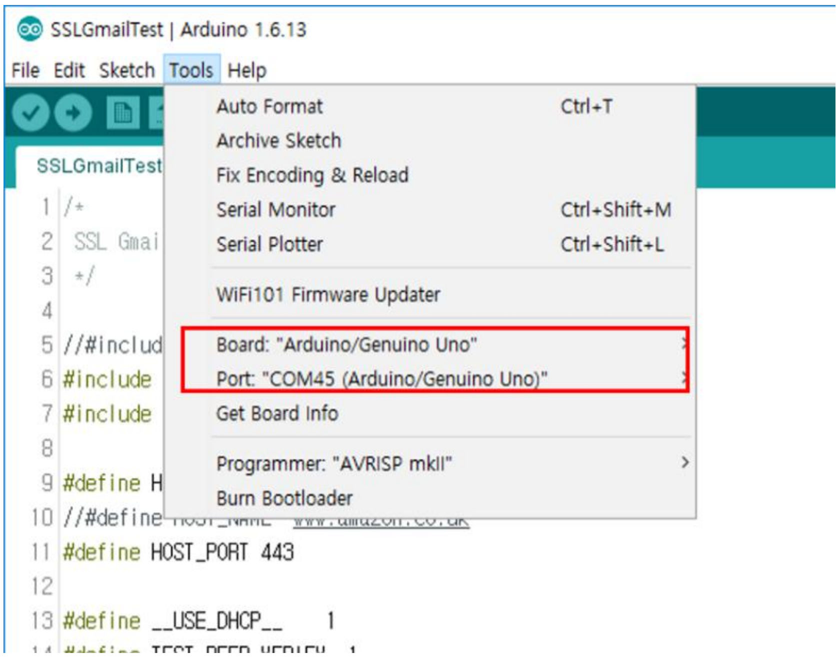
"Example -> Ethernet -> SSLGmailTest"



Step 3: Select the board type and connected COM port under the “Tools” tab:

“board -> Arduino Uno”

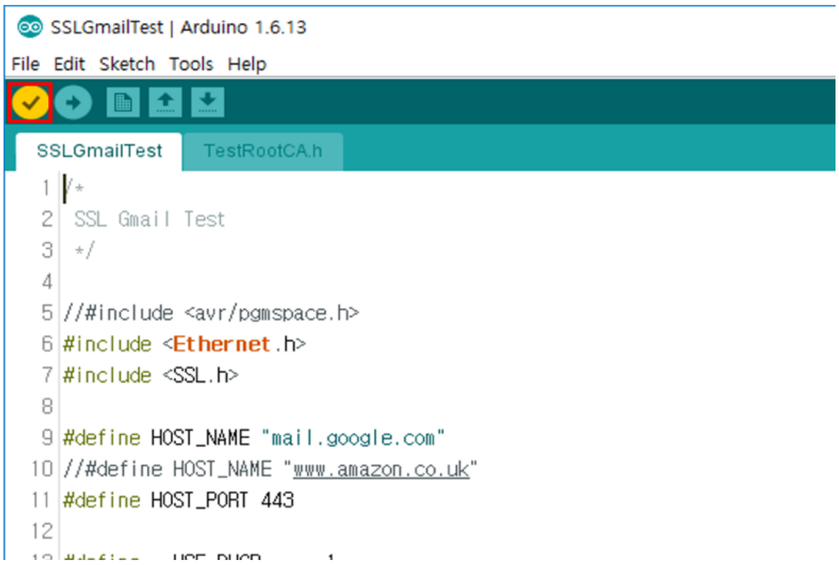
“port -> COMx”


SSLGmailTest | Arduino 1.6.13
File Edit Sketch Tools Help
SSLGmailTest
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

Board: "Arduino/Genuino Uno"
Port: "COM45 (Arduino/Genuino Uno)"

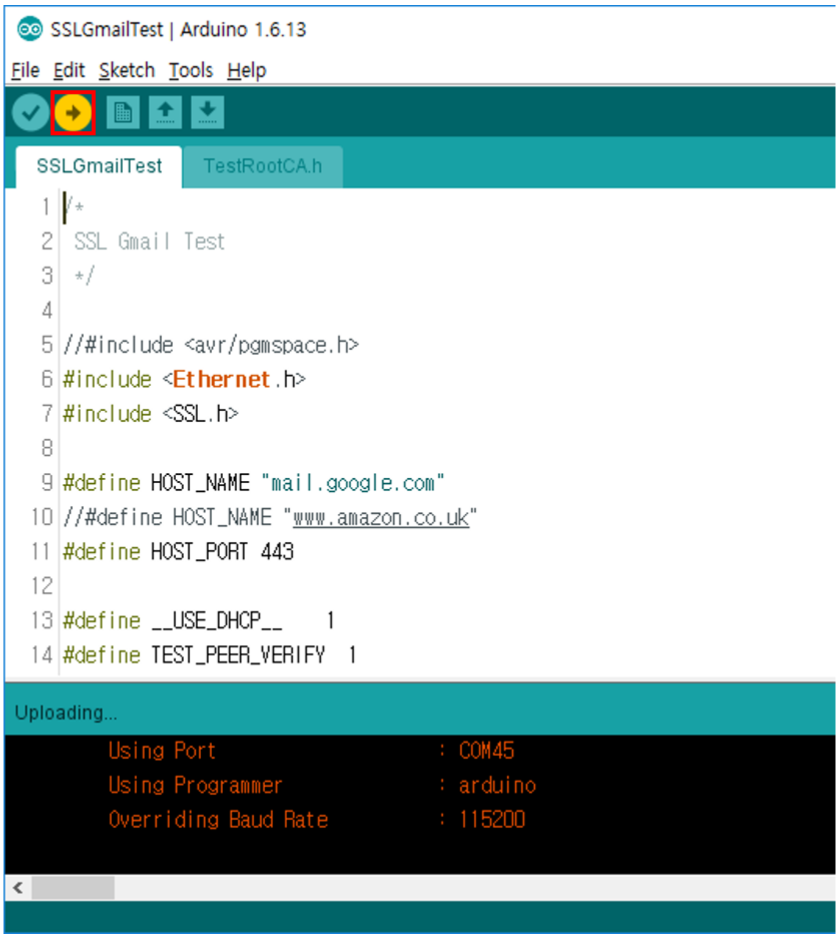
```
/*
 SSL Gmai
 */

//#includ
#include
#include

#define H
//#define
#define HOST_PORT 443

#define __USE_DHCP__     1
```

Step 4: Click “Verify” to check for code errors.



```
/*
 SSL Gmail Test
 */

//#include <avr/pgmspace.h>
#include <Ethernet.h>
#include <SSL.h>

#define HOST_NAME "mail.google.com"
//#define HOST_NAME "www.amazon.co.uk"
#define HOST_PORT 443

```

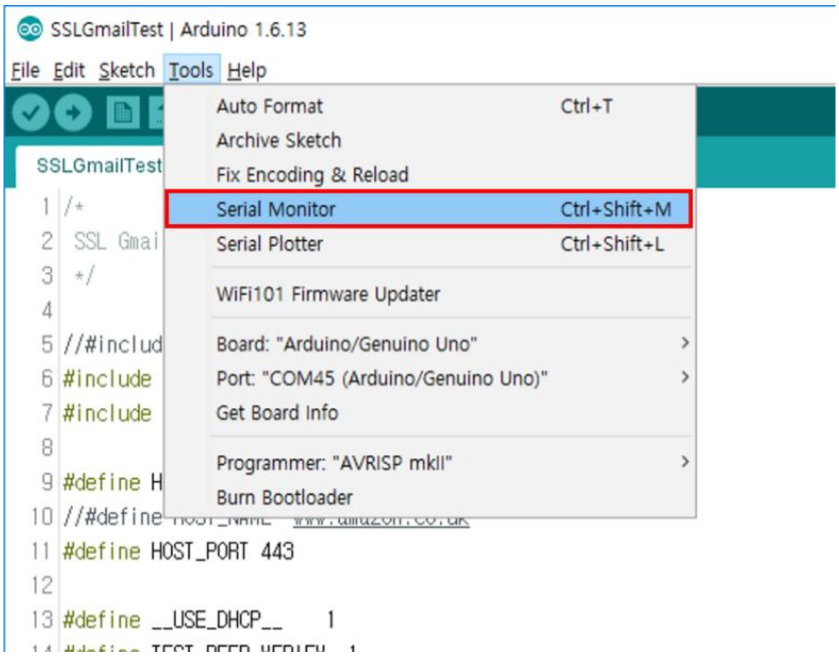
Step 5: Click "Upload" to load the example into the Arduino board.



```
/*
 SSL Gmail Test
 */

//#include <avr/pgmspace.h>
#include <Ethernet.h>
#include <SSL.h>

#define HOST_NAME "mail.google.com"
//#define HOST_NAME "www.amazon.co.uk"
#define HOST_PORT 443

#define __USE_DHCP__      1
#define TEST_PEER_VERIFY  1
```

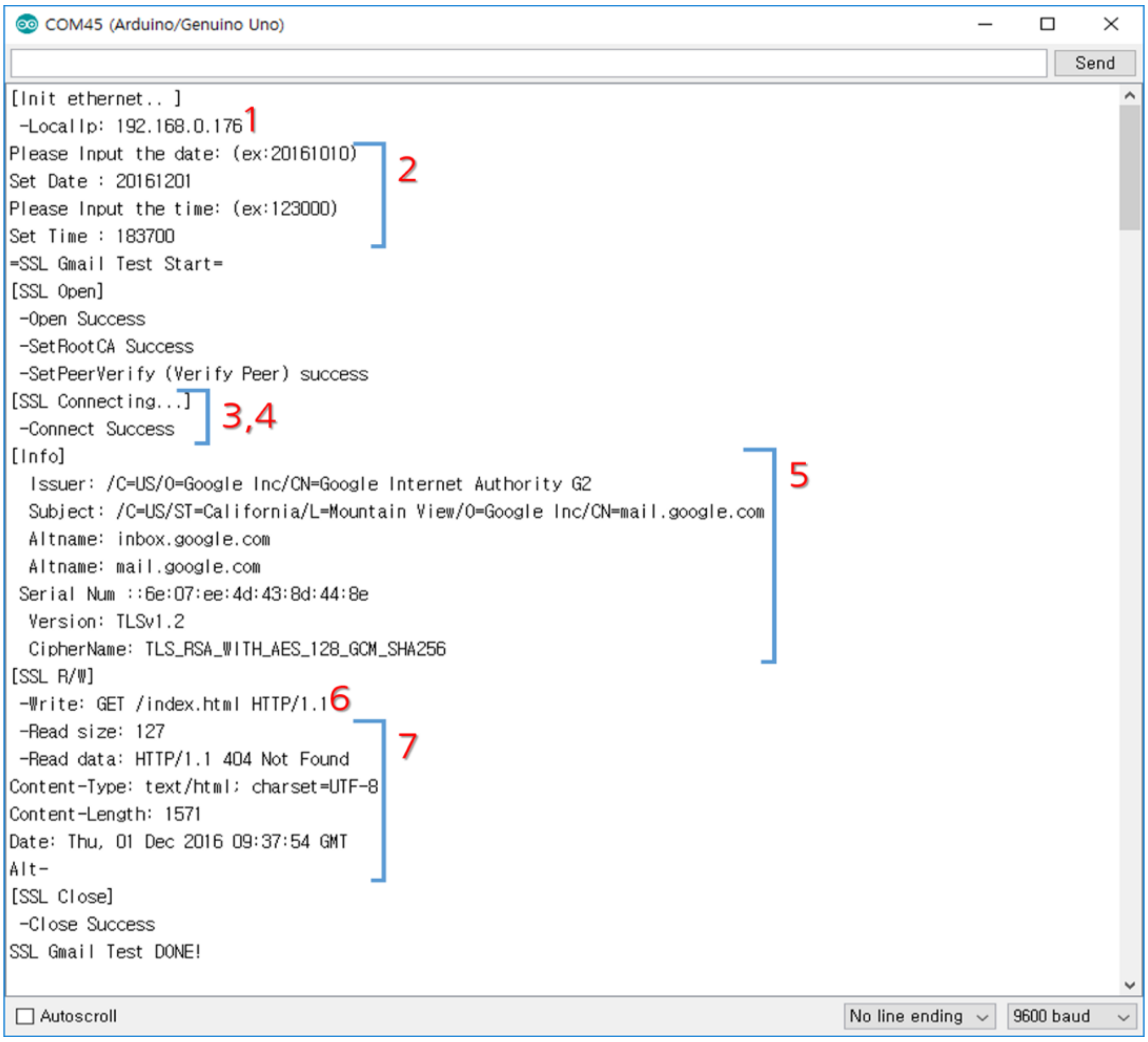
```
         Using Port                    : COM45
         Using Programmer              : arduino
         Overriding Baud Rate          : 115200
```

Step 6: Start the "Serial Monitor" when "Upload" is complete.


SSLGmailTest | Arduino 1.6.13
File Edit Sketch Tools Help
SSLGmailTest
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM45 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

```
/*
 SSL Gmai
 */

//#includ
#include
#include

#define H
//#define
#define HOST_PORT 443

#define __USE_DHCP__      1
```

Step 7: Review the results of the SSL Gmail Test.



```
[Init ethernet.. ]
 -LocalIp: 192.168.0.176
Please Input the date: (ex:20161010)
Set Date : 20161201
Please Input the time: (ex:123000)
Set Time : 183700
=SSL Gmail Test Start=
[SSL Open]
 -Open Success
 -SetRootCA Success
 -SetPeerVerify (Verify Peer) success
[SSL Connecting...]
 -Connect Success
[Info]
  Issuer: /C=US/O=Google Inc/CN=Google Internet Authority G2
  Subject: /C=US/ST=California/L=Mountain View/O=Google Inc/CN=mail.google.com
  Altname: inbox.google.com
  Altname: mail.google.com
 Serial Num ::6e:07:ee:4d:43:8d:44:8e
  Version: TLSv1.2
  CipherName: TLS_RSA_WITH_AES_128_GCM_SHA256
[SSL R/W]
 -Write: GET /index.html HTTP/1.1
 -Read size: 127
 -Read data: HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
Content-Length: 1571
Date: Thu, 01 Dec 2016 09:37:54 GMT
Alt-
[SSL Close]
 -Close Success
SSL Gmail Test DONE!
```

Description:

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

DOCUMENT INFORMATION

Revision History

Revision	Date	Description
1.0	2017/01/04	Official Release

Copyright

Open Source Notice

1. wolfSSL (formerly CyaSSL), yaSSL, wolfCrypt, yaSSH and TaoCrypt software are free software downloads and may be modified to the needs of the user as long as the user adheres to version two of the GPL License.
2. Businesses and enterprises who wish to incorporate wolfSSL products into proprietary appliances or other commercial software products for re-distribution must license commercial versions.
3. eWBM uses wolfSSL under GPLv2 License. eWBM does not have any legal responsibility for redistribution by incorporating it into proprietary appliances or other commercial software products using the Source.