Table of Contents

1

Introduction

Overview

BWS-136 provides the easiest way to enable serial industrial device with networking capability. BWS-136 converts the serial data to standard TCP/IP protocol therefore the serial device can be accessed everywhere via Internet or Ethernet. In addition, BWS-136 provides an embedded Web server which allows user to save the custom web page therefore user can use a standard Web browser to remote manage the serial device.

Package Check List

BWS-136 is shipped with following items:

1. BWS-136 Module
2. Software CD and Electronic user manual

Product Specifications

LAN

Ethernet: 10/100 Mbps, RJ45 x1

Protection: Built-in 1500V magnetic isolation

Serial

RS-232/422/485: DB9 connector x1

RS-232 : RxD, TxD, RTS, CTS, DSR, DTR, DCD, GND

RS-422: RX+, RX-, TX+, TX-, GND

RS-485: Data+, Data-, GND

Baud Rate: 300~38400 bps

Parity: None, Even, Odd

Data Bits: 7, 8

Stop Bits: 1, 2

Flow Control: RTS/CTS, XON/XOFF

Protection: 15KV ESD

Digital Input/Output

General Purpose DIO x8

DIO0 to DIO5: Programmable Digital I/O (TTL)

DIO6 to DIO7: Programmable Digital I/O (CMOS)

pin 1 : DIO 4
pin 2 : DIO 3
pin 3 : DIO 2
pin 4 : DIO 1
pin 5 : DIO 0
pin 6 : DIO GND
pin 7 : DIO 7
pin 8 : DIO 6
pin 9 : DIO 5

Software

Protocol: TCP, UDP, IP, HTTP, ICMP, DHCP, Telnet
Utility: Java Configuration. Web Configuration, Serial Console, Telnet Console
Operation Mode: TCP Server, TCP Client, Web Control

Power

Power input: 9~40VDC@100mA
Connector: Power Jack or Terminal Block

Environment

Operating Temperature: -10~60°C (14~140°F), 5~95% RH

Storage Temperature: -20~85°C (-4~185°F), 5~95% RH

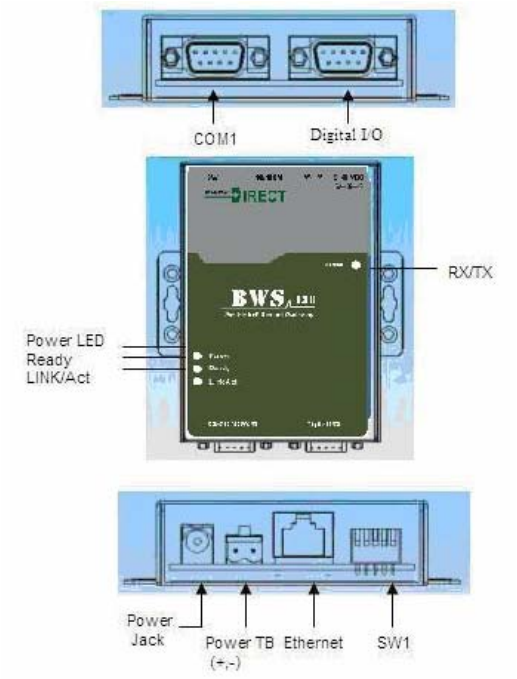
Dimension

108x78x25 mm (HxWxD) without ear

2

Getting Started

Panel Layout



Connecting Power

Connecting 9~40VDC power line with the BWS-136 terminal block or the power jack. If the power is properly supplied, the Power LED will keep solid yellow color.

Connecting Ethernet Port

Connect a RJ45 Ethernet cable to the Ethernet port of BWS-136. The Link/Activity light will keep solid yellow color if Ethernet cable is corrected to the network and this light will keep flashing if there are data transmitted.

Connecting Serial Port

Use a null modem DB9 serial cable to connect a RS232 serial device to BWS-136 serial port. The null modem cable will cross over the RxD to TxD connection between the serial device to BWS-136.

Switch SW1 Settings

Set the SW1 setting to RS-232 mode and pin definition of BWS-136 serial port is as follow:

SW1 Setting:

SW1 Key	1	2	3	4	5
RS-232	ON	OFF	OFF	-	-
RS-422	OFF	OFF	OFF	-	-
RS-485	OFF	ON	ON	-	-
Normal	-	-	-	OFF	OFF
DHCP				OFF	ON
Default				ON	OFF
Console				ON	ON

Normal: BWS-136 is in Normal Operation Mode
DHCP: Network IP Address is assigned by DHCP Server
Default: All the settings are reset to Factory Default.
Console: COM1 serial port is served as console port. User can use ACSII command to configure BWS-136 via COM1 port. Please refer to Appendix I for the ASCII command

ATTENTION

All the settings will not be effective until system reboot by pressing RESET button

LED Status

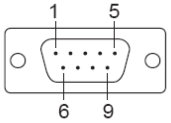
The LED provides the BWS-136 operation information. The

LED status is described as follow:

Power LED: Power LED keeps ON if power (+9VDC to +40VDC) is correctly input to BWS-136.
Ready LED: Ready LED keeps ON when BWS-136 firmware is ready for operation. Ready LED will be flash when BWS-136 in Serial Console mode (SW1 key 4 and key 5 are ON) or Telnet Console mode (Telnet Console port:5001 are connected)
Link/Act LED: Link and Activity LED will turn ON when the Ethernet cable is connected. When there is network data traffic, this LED will be flash.
RX/TX LED: The RX/TX LED is a dual color LED that indicates the serial data traffic. In RS-232 mode, the Yellow LED stands for transmitting data and Green LED means receiving data. In RS-422/485 mode, the Yellow LED stands for receiving data and Green LED means transmitting data.

Serial Port Pin Assignments

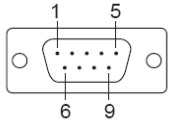
Serial Port COM1 uses a Male DB9 connector and it includes RS-232, RS-422 and RS485 signal and pin assignments are described as follow:



Pin Number	Transmission Signals		
	RS-232	4-wire RS485	RS-485
1	DCD	TxD-	-
2	RxD	TxD+	-
3	TxD	RxD+	Data+
4	DTR	RxD-	Data-
5	GND	GND	GND
6	DSR	-	-
7	RTS	-	-
8	CTS	-	-
9	-	-	-

Digital I/O

Digital I/O uses DB9 connector and the pin assignments are described as follow:



pin 1 : DIO 4
pin 2 : DIO 3
pin 3 : DIO 2
pin 4 : DIO 1
pin 5 : DIO 0
pin 6 : DIO GND
pin 7 : DIO 7
pin 8 : DIO 6
pin 9 : DIO 5

DIO0 to DIO5 are TTL compatible Programmable DIO and DIO 6 to DIO7 are CMOS compatible Programmable DIO. (TTL can pull up +5 V and CMOS can pull up +3 V.)

Factory Default Settings

If you forget your BWS-136 settings, you use SW1 to reset BWSto factory default settings. The factory default settings are:

IP Address: 192.168.16.10

Data Port: 4000

Telnet Port: 5001

Web Port: 80

Baud rate: 19200

Data Format: N,8,1 (Parity, Data bits, Stop bits)

Flow Control: None

Interface: RS-232

Configure BWS-136

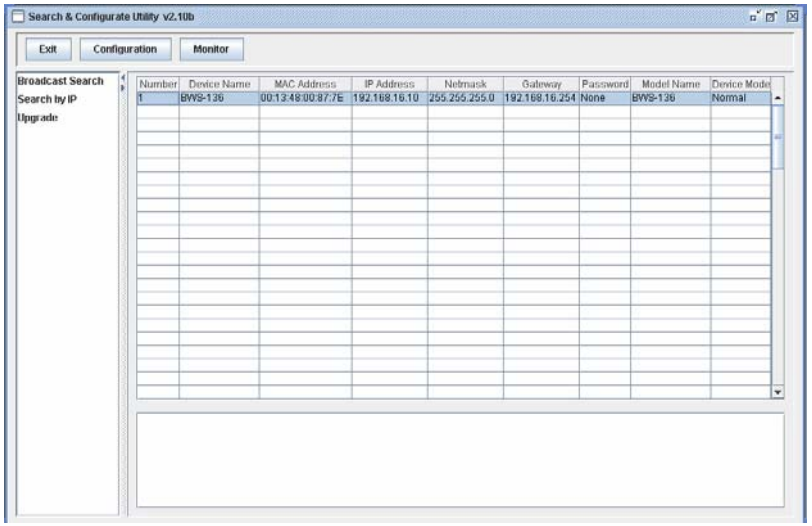
BWS-136 provides three ways to configure the settings. They are:

1. Configuration Utility
2. Web Configuration (Only for change IP address)
3. Telnet Console

3

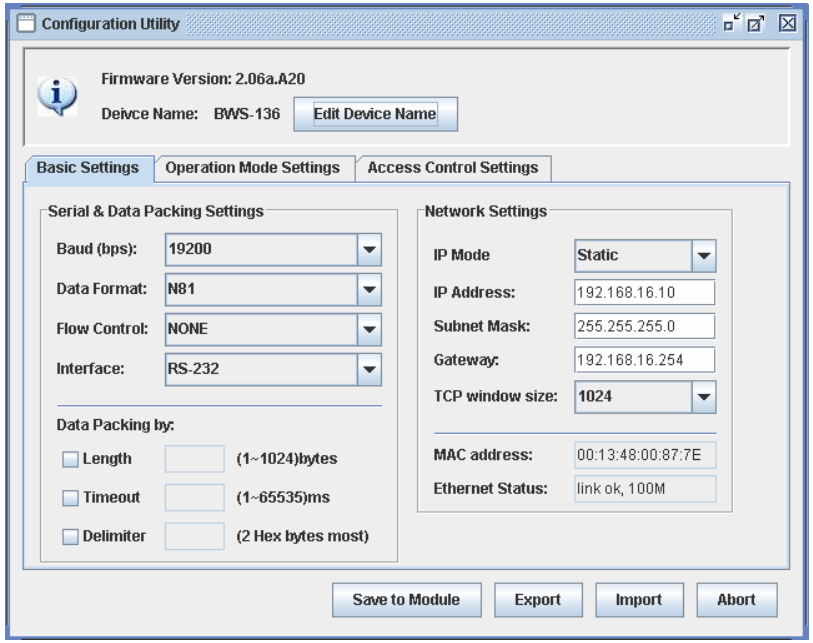
Configuration

Configuration Utility



Once this program started, it will perform a broadcast search to find the BWS device in the network. If you cannot find the BWS module and you know the IP address of the module, you can click **Search by IP** item and specify the IP address to find the BWS-136. The default IP address of BWS-136 is **192.168.16.10.**

Click on the row of the BWS-136 setting, the configuration window will open as follow:



ATTENTION

When the SW1 is set to Console mode, BWS-136 ***cannot be discovered by the Configuration Utility and the configurations are controlled by serial console***.

When enter the configuration mode, the READY LED will be flashing and TCP Data Port will be disabled and data transfer between serial port and Ethernet port will be stopped.

Serial and Data Packing Settings

The group of setting is to configure the serial interface and data packing settings. Data Packing setting is are as follow:

Length: Pack the length of serial data before forwarding data to Ethernet port

Timeout: the period of time to forward data to Ethernet Port

Delimiter: Wait for the Delimiter Character before forwarding data to Ethernet port. The Delimiters are maximum two bytes Hex format ASCII code. If you use Carriage Return (CR) and Line Feed (LF) as delimiters, you can specify 0D0A or 0d0a.

TCP/IP Network Settings

The group of TCP/IP Network settings are configuring the IP Address, TCP port and operation mode as follow:

OpMode: TCP operation mode setting

TCP Port: TCP Port number of BWS-136

Destination IP: The IP address of remote host which BWS-136 will actively connect to (TCP Client mode only)

Connect At: **Startup** means TCP connection is established when system starts (TCP Client mode only)

Timeout: When this option is checked, TCP connection will be disconnected if there is no serial data activity before timeout

IP Mode: Configure the IP Address to be Static IP or Dynamic IP by DHCP

IP Address: Set the Static IP Address

Subnet Mask: Subnet Mask setting

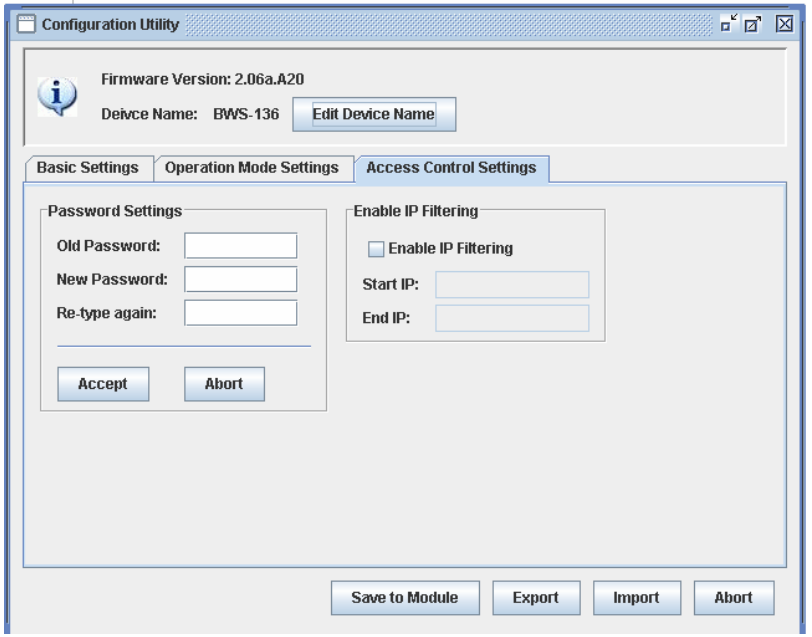
Gateway: Gateway address setting

Device Name Settings

Click Edit Device Name button can edit the device name

Access Control Settings

BWS-136 provides IP address filtering method and password authentication for access control



Enable IP Filtering: When this option is checked, user can specify the starting IP address and ended IP address which are allowed to access BWS-136 TCP port in order to prevent unauthorized access.

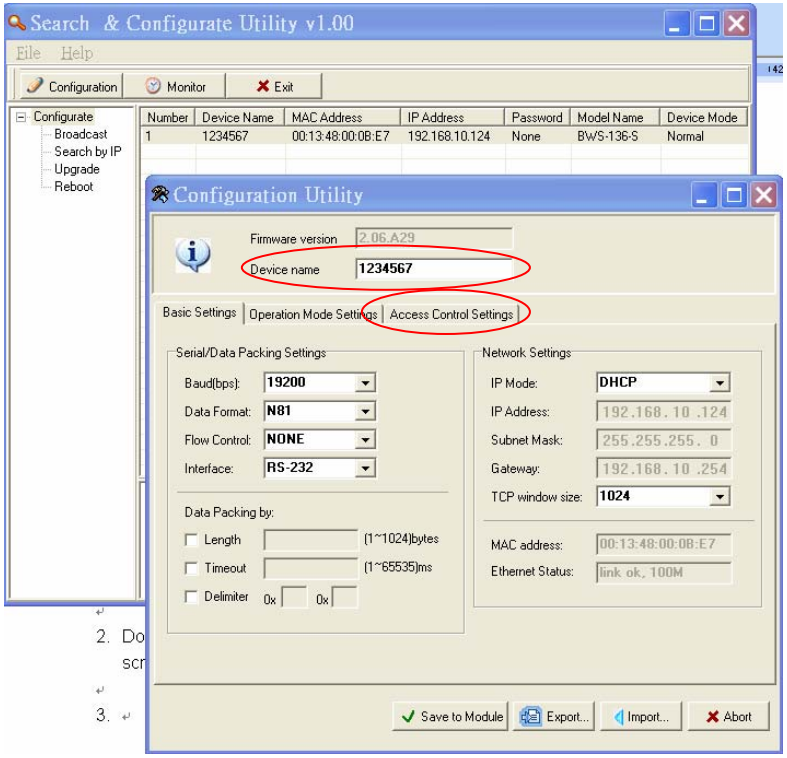
Password Settings: Enter the new password will enable the password authentication. Password is required to login Java Configuration and Web configuration Pages in the next entry.

Auto report:

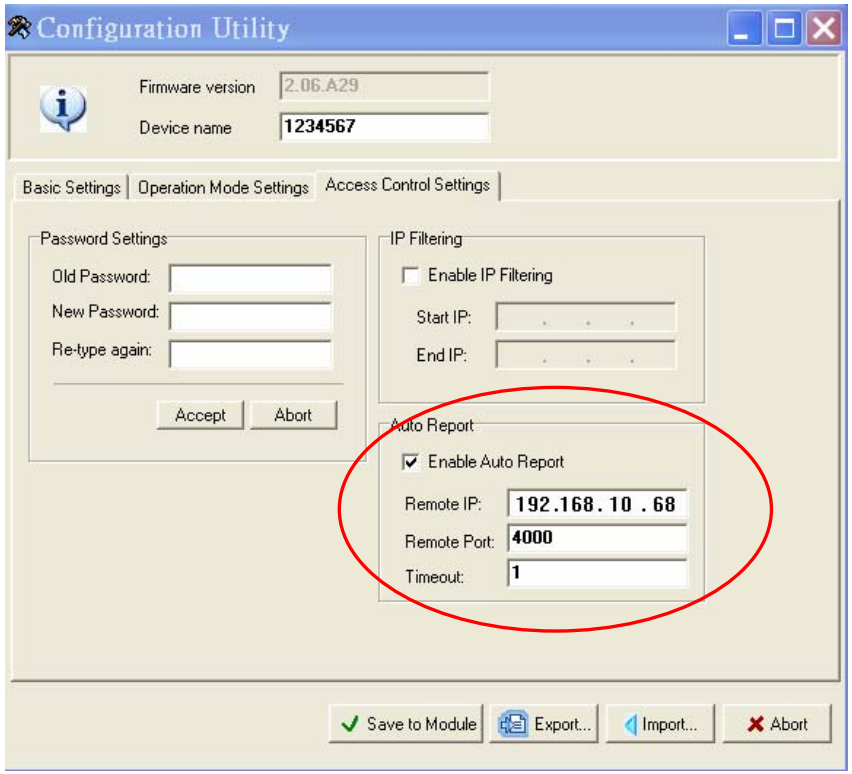
1. Use the broadcast function to find the BWS-136 which you have upgraded with mgr1130 utility.

Notice: You can still configure the new version BWS-136 with the Java configuration utility, but you can't find the new function in java utility. The user interface has been changed for the new function.

2. Double click the device which you choose then it will appear configuration screen. The new function was design in the "Access Control Settings" page. Input the ID information in the "Device name" option.



3. Switch to the "Access Control Settings" page, enable auto report function and input the necessary information.



Notice:

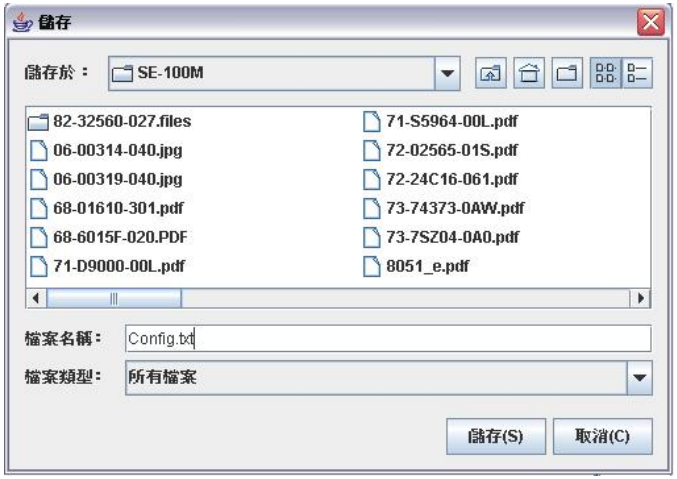
1. **The unit of timeout option is minutes.**
2. **When the BWS-136 have connected the report server and finished the ID information transfer, the BWS-136 will auto break the connection between report server and BWS-136.**

Save Configuration

After the all the settings are configured, click to the **Save to Module button** to save the settings to the module. The new settings will be effective after the reboot of module by clicking the **Reboot Module** button. If the IP address had been modified, please do broadcast search or search by IP again to find the BWS-136 module.

Save Configuration to File

You can also save current configuration to file by click the **Save to File** button. A new window to specify the path of the file will pop out as follow:



The configuration is a text file and uses extension of txt. If you use Text editor to open the configuration command, you will notice that the configuration file contains the ASCII command set of the configuration. You can also edit these commands by the text editor to configure the settings and import it to the BWS-136 by using the Import button.

4

Web Configuration

Overview

BWS-136 can also be configured by Web Browser(only for IP address and serial mode). The build-in Web server and Web configuration pages makes BWS-136 configurable anywhere via a Web browser such as IE and Firefox.

To open the Web configuration pages, you can simply type the IP address to the Web Address input such as 192.168.16.10. The Login windows will ask you enter password if the password option is enabled. After password confirmed, the web configuration page will show up as follow:

Serial/Telnet Command

BWS-136 provides a set of ASCII command to configure BWS through a serial and Ethernet port. User can use serial terminal or Telnet command to configure BWS-136 when Web browser and Java utility are available. In addition, these functions provide user the most convenient way to develop their own configuration utility software simply to use the ASCII command.

Serial Console Mode

To enter the Serial console mode, you can switch SW1 (4~5) to ON position and then serial port will function as a console port. The READY LED will keep flash that indicates BWS-136 is in serial console mode. Please set the serial data setting of the Serial Terminal software (such as Hyper Terminal) to be the same as the BWS-136 serial port. If you forgot the serial port setting, you can use SW1 to reset it to factory default setting as use following setting:

Baud rate: 19200
Data Format: N,8,1 (Parity, Data bits, Stop bits)
Flow Control: None
Interface: RS-232

Now you can use serial terminal software to send the ASCII

command to BWS-136.

In Serial Console Mode, all the network functions will be disabled

Once the configuration is completed, remember to switch SW1 back to Normal operation mode and reboot the module by Reset Command or push reset button.

Telnet Console Mode

BWS-136 uses Port number **5001** as telnet console port. Remember to turn on the LOCALECHO before opening the telnet console port. You must login first before sending command to BWS-136. If password is enabled, you need to use the password to log in. The password is encrypted using Tiny Encryption Algorithm (TEA) and the keys are sixteen bytes with low case character from “a” to “p”. The login command format is as follow:

$LOG[16 bytes of TEA Encrypted Password]

or

$LOG

if there are no password settings

Forgot Password

If you forgot the Password, you can switch SW1 to Factory Default setting mode and use default IP Address **192.168.16.10** and enter the Web console pages. Once the configuration is completed, remember to switch SW1 back to Normal operation mode and reboot the module by Reset Command or push reset button. Password is not required for serial console mode. Therefore you can also use serial console to reset the password.

Command Syntax

The command syntax is
[Delimiter][Command][Carriage Return/**CR**][Line Feed/**LF**]
Delimiter (one byte): The command begins with a delimiter of dollar sign $ and if BWS-136 receives the correct command it will response with an echo which begins with the delimiter of a percentage sign %.
Command: The command are ASCII string which contains three elements: {**R**ead/**W**rite}{Function}{Parameter} as described as follow:

Command Category	**R**	**W**	Function (2 bytes)	Parameters (command related)
Basic	***G***	***S***	***BN***: Device Name	Max. 14 bytes
	G		***BM***: Model Name	Default
		S	***BP***: Password	Max. 8 bytes
	G		***BV***: Firmware Ver..	Default
Serial	***G***	***S***	***SB***: Baud Rate	***1200***
				2400
				4800
				9600
				19200
				38400
	G	***S***	***SD***: Data Format	***N72***
			(Parity,Data,Stop)	***E71***
				O71
				N81
				E72
				O72
				E81
				O81
				N82
	G	***S***	***SF***: Flow Control	***NONE***
				RTS/CTS
				XON/XOFF
	G	***S***	***SI***: Interface	***RS-232***

				RS-422 ***RS-485***
Network	***G***	***S***	***LI***: IP Address	**e.g. *$SLI192.168.2.127***
	G	***S***	***LN***: Netmask	e.g.***$SLN255.0.0.0***
	G	***S***	***LG***: Gateway	**e.g. *$SLG192.168.2.254***
	G	***S***	***LC***: IP Mode	***0***: Static IP ***1***: DCHP
	G		***LM***: MAC	**Default**
	G		***LS***: LAN Status	***Link fail*** ***Link OK,100M*** ***Link OK, 10M***
Operation	***G***	***S***	***OM***: Operation Mode	***TS***:TCP Server ***TC***:TCP Client
	G	***S***	***OC***: TCP connection timeout option	***AC***: always connect ***TO***: Disconnect if timeout
	G	***S***	***OU***: Timeout clock	***0~65535*** (unit: ms) e.g. ***$SOU500***
	G	***S***	***OD***: Data Packing	Three parameters:{ [0]:Delimiter Option [1]: Timeout Option [2]: Length Option }={***1***:Enable,***0***:Disable} e.g. ***$SOD110***
	G	***S***	***OS***: Delimiter	Max. two bytes characters

				e.g. ***$SOS0D0A*** (CR/LF)
	G	***S***	***OT***: Timeout	***0~65535*** (unit: ms) e.g.***$SOT500***
	G	***S***	***OL***: Data Length	***0~1024*** e.g.***$SOL20***
Access Control	***G***	***S***	***OF***:IP Filtering	***0***: Disable ***1***:Enable
	G	***S***	***OI***: Authorized IP	e.g.***$SOI192.168.16.1:192.168.16.9*** (Starting IP:Ended IP)
TCP Server	***G***	***S***	***VP***: Listen Port	***1~65535*** e.g. ***$SVP4000***
TCP Client	***G***	***S***	***CI***: Destination IP	e.g.***$SCI192.168.16.200***
	G	***S***	***CP***: Destination Port	e.g.***$SCP4001***
	G	***S***	***CC***: Connection Logic	***0***: Reserved for PPPoE ***1***:Establish connection when Serial Data in
System		***S***	***YC***: System mode	***R***: Reboot system ***F***: Reset to default setting and reboot ***D***: Set current setting as Default setting
		S	***EI***: Exit console	
DIO		***S***	***DM:*** Set Digital I/O mode	***I:*** Input ***O:***Output

				e.g. ***$SDMI:0,4,5*** set DIO0, DIO4, DIO5 as input ***$SDMO:1,2,3,6,7*** set DIO1,DIO2,DIO3,DIO6,DIO7 as output
	G		***DM:*** Get Digital I/O mode	e.g. ***$GDM*** the response is ***%GDM I,O,O,O,I,I,O,O***
		S	***DOH:*** Set DO channel High Output	e.g. ***$SDOH:1,3,7***
		S	***DOL:*** Set DO channel Low Output	e.g. ***$SDOL:2,6***
	G		***DI:*** Get DI channel status	e.g. ***$GDI*** the response is ***%GDI 0=1,4=0,5=1*** It stands for DIO0=1,DIO4=0,DIO5=1

Remark: Italic font stands for Command String

CR/LF (two bytes): Both command from host and echo from BWS-136 are terminated with a Carriage Return (ACSII code Hex 0d) and Line Feed (ASCII code Hex 0a). Therefore please remember to add CR/LF at the end of command line in the terminal emulation program.

All command strings are in Upper Case

Echo Syntax

After BWS-136 console port received correct command string, it will response Echo String and the Echo Syntax is as follow:
[Delimiter][Function][Status]
Delimiter (One byte): A percentage sign "%"
Function (Two bytes): function string
Status: "OK" or "FAIL"

For example: Set the IP Address by sending command

$SLI192.168.16.100(CR)(LF)

Echo string

%SLIOK(CR)(LF)

means the new IP address is successfully transferred to the module and the new setting will be effective after reboot the module.
To get IP address setting, you can send a command as follow

$GLI

then the Echo string will be

%GLI192.168.16.100

Always remember to reboot the module by send command ***$SYCR*** to use the new configuration

Appendix A

Tiny Encryption Algorithm

1. Example of TEA.C

```
/*****************************************************
```

The Tiny Encryption Algorithm (TEA) by David Wheeler and Roger Needham of the
Cambridge Computer Laboratory. Placed in the Public Domain by
David Wheeler and Roger Needham.
**** ANSI C VERSION (New Variant) ****
Notes:TEA is a Feistel cipher with XOR and addition as the non-linear mixing
functions.
Takes 64 bits of data in v[0] and v[1]. Returns 64 bits of data in w[0] and w[1].
Takes 128 bits of key in k[0] - k[3].
TEA can be operated in any of the modes of DES. Cipher Block Chaining is, for example, simple to implement. n is the number of iterations. 32 is ample,
16 is sufficient, as few as eight may be OK. The algorithm achieves good dispersion after six iterations. The iteration count can be made variable if required.
Note this is optimized for 32-bit CPUs with fast shift capabilities. It can very easily be ported to assembly language on most CPUs. delta is chosen

to be the real part of (the golden ratio Sqrt(5/4) - 1/2 ~ 0.618034 multiplied by 2^32).
This version has been amended to foil two weaknesses identified by David A. Wagner (daw@cs.berkeley.edu): 1) effective key length of old-variant TEA was 126 not 128 bits 2) a related key attack was possible although impractical.

```
*****************************************************/
//#include "xtea.h"
void encipher(unsigned long *const v,unsigned long *const w,
const unsigned long *const k)
{
    register unsigned long
y=v[0],z=v[1],sum=0,delta=0x9E3779B9,n=32;
    while(n-->0)
        { y+= (z<<4 ^ z>>5) + z ^ sum + k[sum&3];
        sum += delta;
        z+= (y<<4 ^ y>>5) + y ^ sum + k[sum>>11 & 3];
        }
    w[0]=y; w[1]=z;
}
void decipher(unsigned long *const v,unsigned long *const w,
const unsigned long *const k)
{
    register unsigned long          y=v[0],z=v[1],sum=0xC6EF3720,
                                        delta=0x9E3779B9,n=32;
    /* sum = delta<<5, in general sum = delta * n */
```

```
  while(n-->0)
        {
              z-= (y<<4 ^ y>>5) + y ^ sum + k[sum>>11 & 3];
        sum -= delta;
              y-= (z<<4 ^ z>>5) + z ^ sum + k[sum&3];
        }
        w[0]=y; w[1]=z;
}
```

2. The TEA.H

```
#ifndef __XTEA_H__
#define __XTEA_H__
void encipher(unsigned long *const v,unsigned long *const w,
const unsigned long *const k);
void decipher(unsigned long *const v,unsigned long *const w,
const unsigned long *const k);
#endif
```