

**TERMI** User Guide  
Copyright © **SIGMATONE**  
Version 1.0

## Table of Contents

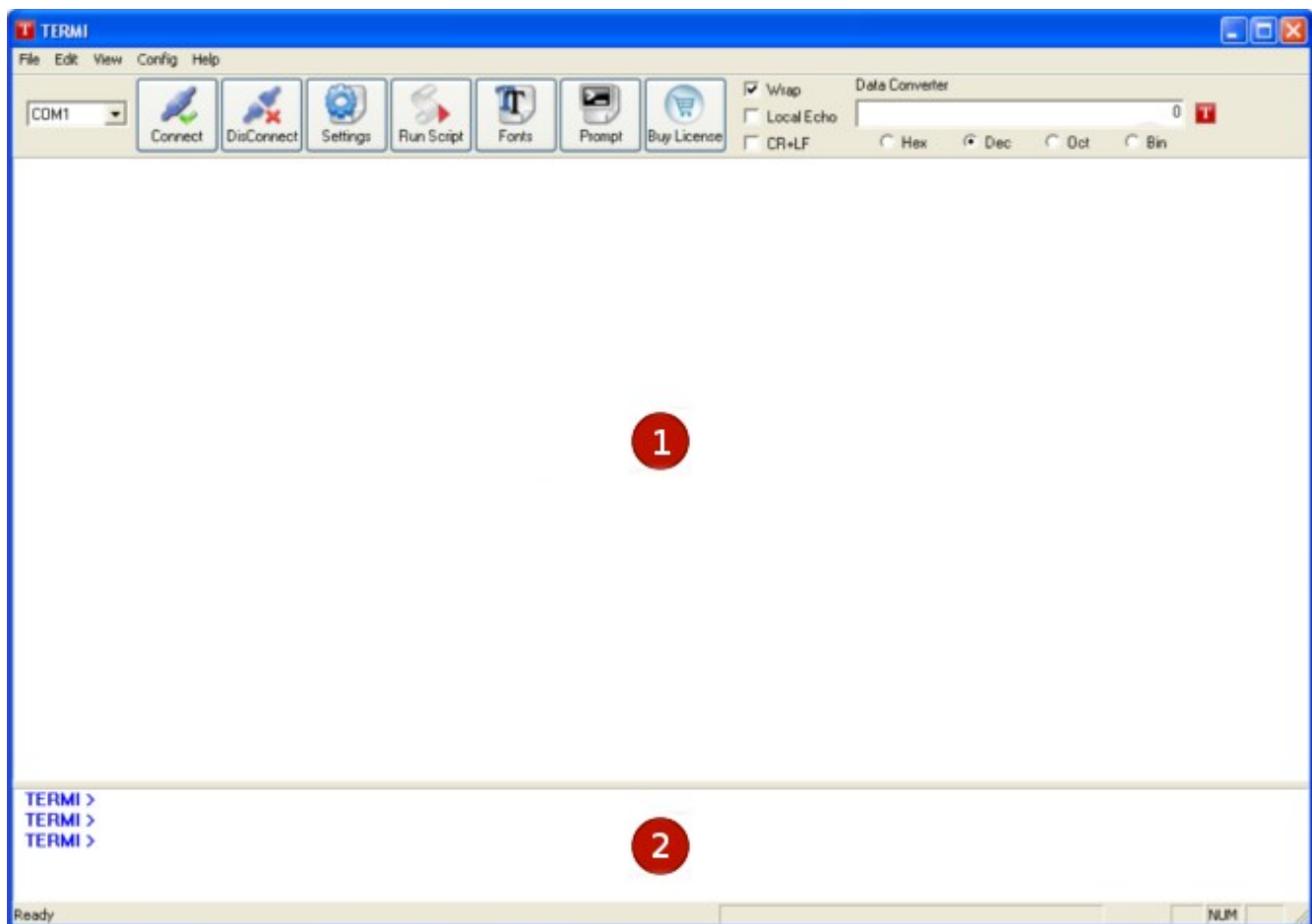
1 System Requirement .....	3
2 User Interface.....	4
3 TERMI Commands.....	8
3.1 SendOut.....	8
3.2 SendOutR.....	8
3.3 SetTermiReadDelay.....	9
3.4 GetTermiReadDelay.....	9
3.5 ResetTermiReadDelay.....	10
3.6 SetReadBufferSize.....	10
3.7 GetReadBufferSize.....	10
3.8 ResetReadBufferSize.....	11
3.9 pwd.....	11
3.10 dir.....	11
3.11 clear.....	12
3.12 cmdhist.....	12
3.13 GetLicensePath.....	12
3.14 GetTermiVersion.....	13
3.15 help.....	13
3.16 exit.....	14
4 Set License Path.....	14

## 1 System Requirement

Minimum System Requirement
<ul style="list-style-type: none"><li>• Windows XP/2000/NT</li><li>• 500Mhz Processor, 256 MB RAM</li><li>• COM or Serial Port</li></ul>

## 2 User Interface

Termi provides two interface pans, pan-1 directly talks to the connected serial-port and pan-2 supports powerful TCL scripting. Pan-2 can also talk to serial port using commands provided by TERMI. These TERMI commands can be used directly as TCL commands for scripting/automation.



## 1) Serial Pan

As shown above, pan-1 is a serial-port pan which gets connected directly to the serial-port. Whatever user types on pan-1 goes directly to the serial port and subsequently all incoming data also gets displayed. This pan is also referred as serial pan in rest of the document.

## 2) TCL Pan

Pan-2 is scripting pan which supports TCL scripting. This pan can send data on serial port using commands like “SendOut” and “SendOutR”.



## 3) Serial-Port-list :

Drop-down list of ports, TERMI supports COM1 to COM8 ports.

## 4) Connect :

Connect button is used to open port selected from serial-port-list. Only one port can be open at a time.

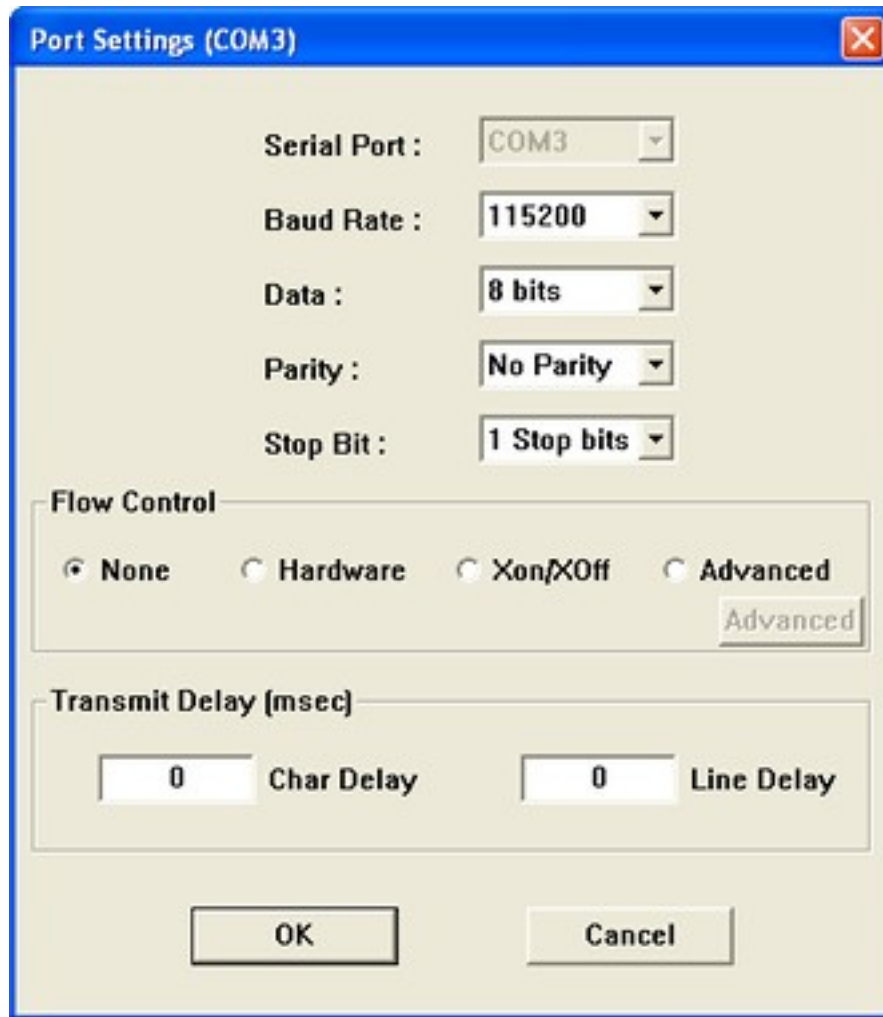
## 5) Dis-Connect :

Use to disconnect connected port.

## 6) Settings :

To set port specific settings. Port settings can be made to opened/connected port only.

Following settings are provided :-



7) Run Script :

To execute TCL script. Use this button to browse and run TCL script.

8) Fonts :

To change serial-pan font.

9) Prompt:

Hide/Show TERMI prompt. If user doesn't need TERMI prompt he may hide it to create more working space for serial-pan.

## 10) Buy License

Takes user directly to the TERMI license purchase page. User may also safely purchase license from [www.sigmatone.com](http://www.sigmatone.com). After successful license purchase, license file is sent to user via email. Valid license allows user to copy and save logs.

## 11) Format Settings :

- a) Wrap : Enables line wrapping on serial-pan
- b) Local Echo : Enables local echoing of transmitted character on serial-pan
- c) New Line : By default CR is new-line character but can be set to CR+LF

## 12) Data Converter :

Allows user to convert values between decimal, octa-decimal, hexadecimal and binary. Very useful feature for embedded developers.

## 3 TERMI Commands

Apart from normal TCL commands TERMI also supports following commands.

### 3.1 SendOut

Sends “data” out to connected serial-port. While sending “data” out, character and line delays from port settings are applied. If data string contains TCL variable, string value of the variable is sent out.

```
TERMI > set $address 20000
TERMI > set $data 1234
TERMI > SendOut “devwrite $address $data”
TERMI >
```

For the above code, data sent to serial port will be “**devwrite 20000 1234**” (without quotes)

Please note that SendOut command only sends data out, it won't read or return anything back.

This command is very efficient if user only wants to send data out without bothering about incoming data.

*Note that both outgoing and incoming data will get displayed on serial-pan.*

```
Syntax : SendOut “string”
Return : None
Example :
TERMI > SendOut “devwrite $d”
```

### 3.2 SendOutR

This command is same as “SendOut” except that it returns incoming data after



command execution.

After sending sending “data” out to the connected serial-port, SendOutR waits for *TermiReadDelay*. While waiting for *TermiReadDelay* received data is collected in ReadBuffer. Collected data is returned as command-return after *TermiReadDelay* expires.

Character and line delays settings are applied while sending data out.

TermiReadDelay and ReadBuffer size can be changed by using *SetTermiReadDelay* and *SetReadBufferSize* commands.

**Syntax** : SendOut “string”

**Return** : Incoming Data

**Example** :

```
TERMI > set $address 20000
```

```
TERMI > SendOutR “devread $address”
```

```
1234
```

```
TERMI>
```

*Note that both outgoing and incoming data will get displayed on serial-pan.*

### 3.3 *SetTermiReadDelay*

Sets TermiReadDelay in milli seconds. SendOutR commands accumulates incoming data for TermiReadDelay duration in ReadBuffer.

**Syntax** : SetTermiReadDelay time-in-mili-seond

**Return** : None

**Example** :

```
TERMI > SetTermiReadDelay 50
```

### 3.4 *GetTermiReadDelay*

Returns current TermiReadDelay.

**Syntax** : GetTermiReadDelay

**Return** : Time in milli seconds

**Example :**

```
TERMI > GetTermiReadDelay
50
TERMI >
```

### 3.5 *ResetTermiReadDelay*

Resets TermiReadDelay to original 200ms.

**Syntax** : ResetTermiReadDelay

**Return** : None

**Example :**

```
TERMI > ResetTermiReadDelay
TERMI >
```

### 3.6 *SetReadBufferSize*

Sets read buffer size, this buffer is used for receiving data in SendOutR command. Buffer size is set as number of characters.

**Syntax** : SetReadBufferSize size-in-no-of-characters

**Return** : None

**Example :**

```
TERMI > SetReadBufferSize 1500
TERMI >
```

### 3.7 *GetReadBufferSize*

Returns current read-buffer size.

**Syntax** : GetReadBufferSize

**Return** : size in number of characters

**Example :**

```
TERMI > GetReadBufferSize
1500
TERMI >
```

### 3.8 *ResetReadBufferSize*

Sets read buffer size to original 1000 characters.

**Syntax** : ResetReadBufferSize

**Return** : None

**Example** :

TERMI > ResetReadBufferSize

TERMI >

### 3.9 *pwd*

Returns present working directory path. *pwd* can be changed using *cd* command.

**Syntax** : pwd

**Return** : present working directory path

**Example** :

TERMI > pwd

C:\Program Files\TERMI

TERMI >

### 3.10 *dir*

Prints files and directory present in *pwd*.

**Syntax** : dir

**Return** : files/directory present in *pwd*

**Example** :

TERMI > dir

mclcfpg.dll

serial\_port.exe

TCL85.DLL

TERMI.exe

TERMI.ini

TERMI >

### 3.11 *clear*

Clears TERMI pan.

**Syntax** : clear

**Return** : None

**Example** :

TERMI > clear

### 3.12 *cmdhist*

Termi stores history of previous commands typed by user, this command prints list of previously typed commands. Previously typed command can be accessed directly by up/down arrow keys.

**Syntax** : cmdhist

**Return** : None

**Example** :

TERMI > cmdhist

dir

pwd

GetTermiReadDelay

### 3.13 *GetLicensePath*

Prints current license path.

**Syntax** : GetLicensePath

**Return** : License path

**Example** :

TERMI > GetLicensePath

C:\TERMI\_LIC\TERMI

TERMI >

### 3.14 *GetTermiVersion*

Prints TERMI version.

**Syntax** : GetTermiVersion

**Return** : version no

**Example** :

```
TERMI > GetTermiVersion
```

```
1.0
```

```
TERMI >
```

### 3.15 *help*

Prints one liner help for each TERMI command.

**Syntax** : help

**Return** : command list and one liner description

**Example** :

```
TERMI > help
```

```
pwd                (prints present working directory)
dir                (prints files/dir present in pwd)
SendOut            (Sends string out to serial port)
SendOutR           (Sends string out to serial port and reads data back)
clear              (clears TERMI window)
cmdhist            (Previous commands history)
SetTermiReadDelay (sets read delay in ms)
GetTermiReadDelay (gets read delay)
ResetTermiReadDelay (resets to original delay value)
SetReadBufferSize (sets read buffer size)
GetReadBufferSize (gets read buffer size)
ResetReadBufferSize (resets to original buffer size)
GetTermiVersion    (prints Termi Version)
GetLicensePath     (prints license file path)
help               (help command)
exit               (Exit TREMI Application)
```

```
TERMI >
```

### 3.16 exit

exits TERMI

**Syntax** : exit

**Return** : None

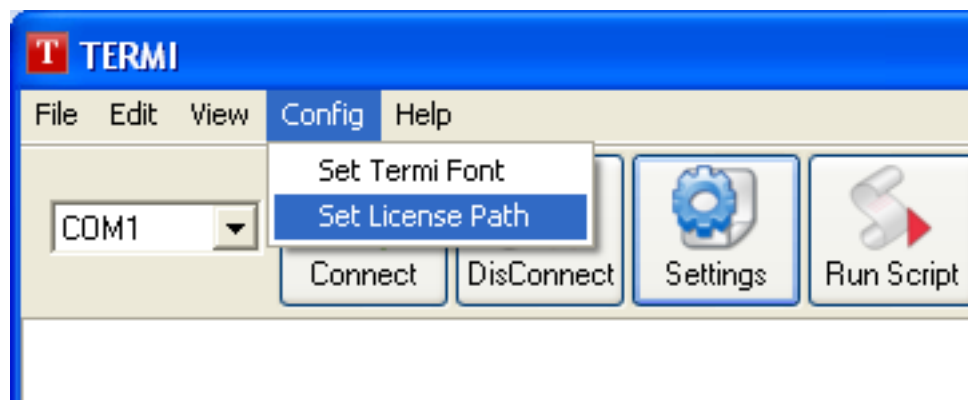
**Example** :

TERMI > exit

## 4 Set License Path

After successful license purchase, user receives termi.lic license file via email. User should keep this license file at some safe location and point TERMI to the license file.

To point TERMI to license file click Config->Set License Path (as shown below)



Browse/select license file and click OK. After license file path is set, user is requested to restart TERMI for changes to take into effect. After restart, the registration information should be updated in the About box (as shown below)

