

# TEMU

## ***Serial Console UI Model Manual***

Mattias Holm

Version 1.1, 2016-05-12

# Table of Contents

1. Introduction.....

2. Creation.....

3. Configuration.....

4. Attributes.....

4.1. Properties.....

4.2. Interfaces.....

4.3. Ports.....

5. Limitations.....

1

1

1

1

1

1

1

2

*Table 1. Record of Changes*

Rev	Date	Author	Note
1.1	2016-05-12	MH	Auto gen tables.
1.0	2015-10-06	MH	Initial version.

# 1. Introduction

The serial console ui is a simple graphical endpoint for serial traffic that you can connect a device's UART to. It forks of a separate process which display a new window with the serial port output. This window also handles interactive input, meaning that you can for example type commands to a command line interface provided by the software running in the emulated environment.

The console window supports limited VT100 emulation.

# 2. Creation

The ConsoleUI class is defined in libTEMUConsoleUI.so. The constructor takes no parameters.

# 3. Configuration

No specific configuration needed.

# 4. Attributes

## 4.1. Properties

Name	Type	Description
object.timeSource	object	Time source object (a cpu or machine object)
serial	iref	

## 4.2. Interfaces

Name	Type	Description
ObjectIface	ObjectIface	
SerialIface	SerialIface	

## 4.3. Ports

Prop	Iface	Description
serial	SerialIface	serial port

## 5. Limitations

As with all other models, problems not listed here should be reported to Terma as they may indicate bugs in the software.

- The Console UI requires QT 4 to be installed (e.g. with your package manager) and any needed support libraries for QT. Thus the console in particular has a lot of extra dependencies over the rest of the emulator. If you are running this on specific systems and the console does not work, please report this to Terma.
- The console always do VT100 emulation, the emulation cannot be disabled.
- Only partial VT100 support exists. The supported CSIs include colors and cursor movements. Some CSIs may be missing.
- The console does not echo input back automatically, this is typically done by the remote serial end. Consequently, you will not see any characters if you type them in the console and the remote does not echo back.
- The console model will in normal mode poll the input at 100 Hz as the emulator at the moment of writing does not support the injection of asynchronous events. The 100 Hz polling is good enough for interactive use.