T-EMU: Serial Console Model Manual

© Terma GmbH, Germany, 2015. Proprietary and intellectual rights of Terma GmbH, Germany are involved in the subject-matter of this material and all manufacturing, reproduction, use, disclosure and sales rights pertaining to such subject-matter are expressly reserved. This material is submitted for a specific purpose as agreed in writing, and the recipient by accepting this material agrees that this material will not be used, copied or reproduced in whole or in part, nor its content (or any part thereof) revealed in any manner or to any third party, except own staff, to meet the purpose for which it was submitted and subject to the terms of the written agreement.
1. Introduction

The serial console is a simple endpoint for serial traffic that you can connect a device’s UART to. It echos received data to stdout and optionally logs the data in an unbounded log.

2. Dedicated API

There is a dedicated API for accessing the console log. Note that the functions are defined in libTEMUConsole.so.

```c
#include "temu-c/Models/Console.h"

// These functions are defined in libTEMUConsole.so
uint64_t temu_consoleGetLineCount(void *Con);
const char* temu_consoleGetLine(void *Con, uint64_t Line);
```

3. Creation

The Console class is defined in libTEMUConsole.so. The constructor takes no parameters.

4. Configuration

4.1. Properties

- `config.caretControl`: Will enable handling of caret notation. Handling is primitive and simply omit the caret notation data from stdout. Defaults to on.
- `config.recordTraffic`: Enable traffic recording, the raw data is recorded (unaffected by caretControl). Defaults to off.
lastByte  Contains the last received byte.

outByte  Writing to the outByte property will send the written byte to the serial object (using the serial interface) the console is connected to. It provides a simple way for scripts to send bytes to UARTs.

serial  Interface property that should be connected to a UART or serial model.

queue  Connect to the EventInterface that will drive the console’s time. Normally the CPU or machine object.

4.2. Interfaces

SerialIface  Connect your UART to this interface.

5. Limitations

- The record buffer cannot be cleaned without deleting the console object.

- Caret control only omits caret sequences from being put on stdout (especially nice when booting Linux). It doesn’t act on the sequences in any way at the moment e.g. a delete character will be ignored and not actually delete anything.

- The record buffer will not be checkpointed.