Table of Contents

1. Introduction ......................................................... 1
2. Dedicated API ....................................................... 1
3. Creation ............................................................ 1
4. Configuration ....................................................... 1
5. Attributes ........................................................... 1
   5.1. Properties .................................................... 1
   5.2. Interfaces ..................................................... 2
   5.3. Ports .......................................................... 2
6. Limitations ......................................................... 2
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

config.caretControl can be used to eliminate some VT100 characters that are printed to the console otherwise.

config.recordTraffic can be set to enable data recording in the console model, this data can then be extracted with the API.

5. Attributes

5.1. Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config.caretControl</td>
<td>uint8_t</td>
<td></td>
</tr>
</tbody>
</table>
5.2. Interfaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerialIface</td>
<td>SerialIface</td>
<td></td>
</tr>
</tbody>
</table>

5.3. Ports

<table>
<thead>
<tr>
<th>Prop</th>
<th>Iface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial</td>
<td>SerialIface</td>
<td>serial port</td>
</tr>
</tbody>
</table>

6. 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.