EBmonitor Host CLI

by Gerard Zagema (EmBitz okt 2025)

EBmon is a lightweight debugging tool for embedded systems, providing real-time monitoring of STDIO-like pipes between a target device and a host. It has been used for many years as a plugin in the EmBitz IDE, allowing developers to inspect and control I/O without halting the target. The CLI tool ebmon is a standalone, bidirectional terminal which is lightning fast because it doesn't require GDB client.

Features

- Bidirectional communication using stdin/stdout pipes
- Optional ELF file integration for debug info about pipes
- Lightweight and lightning-fast standalone
- Unbuffered, buffered and buffered with EOL sending
- Can be used as a replacement for the EBmon plugin in the EmBitz IDE (not alongside it)
- Multiple concurrent EBmon instances are supported by using uniquely renamed pipes
- Not limited to EBlink; works with any suitable non-blocking GDB server
- Available in all packages Linux and windows

Host-Side Tool (`ebmon`)

Usage

ebmon [<elf file name>] [options]

Options

Option	Description
-h, --help	Print this help
-g, --nogui	Disable GUI message boxes (Windows only)
-s, --server <host>[:<port>]	Connect to GDB server (default 127.0.0.1:2331)
-e, --elf <file>	ELF file with debug info about the pipes
-i, --in <name|addr>	Input pipe (default _eb_monitor_stdin)
-o, --out <name|addr>	Output pipe (default _eb_monitor_stdout)
-l, --log <file>	Log file for all target output data
-t, --timeout <s>	Server connection timeout
-v, --verbose <0..8>	Verbose logging level (default 3)

Examples

ebmon myprogram.elf

ebmon myprogram.elf -i myInputPipe -s workbench:4523

ebmon -o 0x20000500

Notes:
- If a pipe variable name is given, an ELF file is mandatory
- The ELF file can be provided as the first argument or via –elf option

Interactive Commands

Once ebmon is running, the following commands are available:

Key	Action
F5	Run program
Shift-F5	Halt program
F1	Restart program
Ctrl-U	Unbuffered output (send directly)
Ctrl-B	Buffered output (send on Enter)
Ctrl-N	Buffered output with EOL termination
Ctrl-E	Local Echo ON/OFF
Ctrl-H	Display this help
Esc	Quit program (also Ctrl-C)

Note: Printing formfeed \f clears the screen.

Target-Side Implementation

The target side uses circular buffers for STDIO-like communication:

```
static char _stdout_buffer[STDOUT_BUFFERSIZE] __attribute__((aligned(4)));
static char _stdin_buffer[STDIN_BUFFERSIZE]   __attribute__((aligned(4)));

typedef struct _std_pipe {
   uint16_t length;
   volatile uint16_t tail;
   volatile uint16_t head;
   void *ptr;
} _std_pipe;

static _std_pipe _eb_monitor_stdout = {STDOUT_BUFFERSIZE, 0, 0, _stdout_buffer};
static _std_pipe _eb_monitor_stdin  = {STDIN_BUFFERSIZE,  0, 0, _stdin_buffer};
```

- _write() writes to _eb_monitor_stdout buffer
- _read() reads from _eb_monitor_stdin buffer
- EBmonitor_flush() clears the selected pipe

Target Library

Include ebmon.c in your embedded project. Make sure STDOUT_BUFFERSIZE and STDIN_BUFFERSIZE are configured appropriately for your application.

License

This project is licensed under the MIT License — see LICENSE for details.

Authors

- Gerard Zagema – Sysdes Company supporting EBmonitor development

History / Background

EBmonitor has been part of the EmBitz IDE plugin ecosystem for many years, providing developers with:
- Real-time monitoring of embedded STDIO pipes
- Minimal runtime footprint on the target
- Requires no hardware resources on the target side