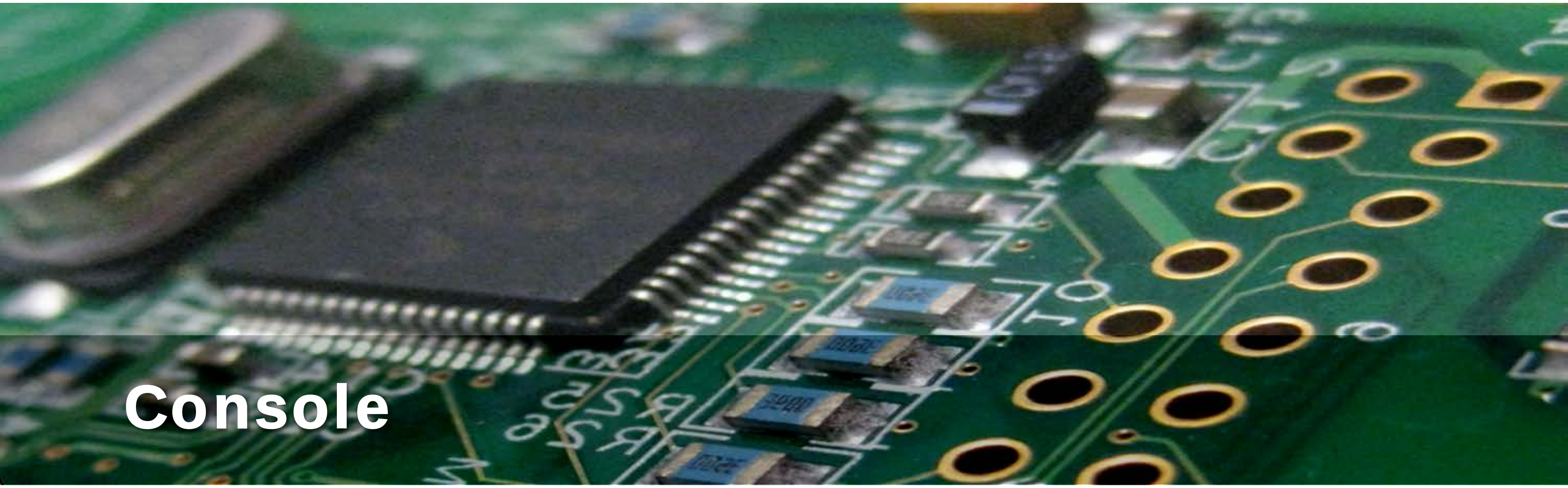
Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Technik & Architektur



Console

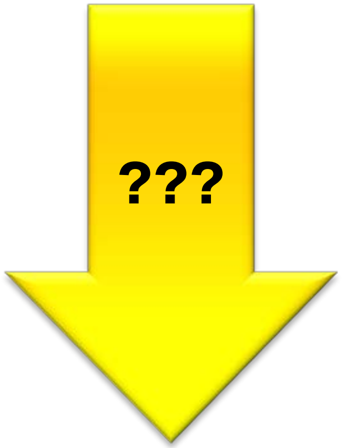
“I do not not like printf()!”

Prof. Erich Styger
erich.styger@hslu.ch
+41 41 349 33 01

Learning Goals

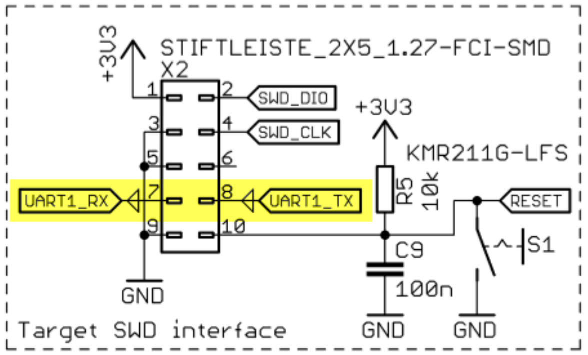
- Problem: Write string for button pressed? Debug messages?

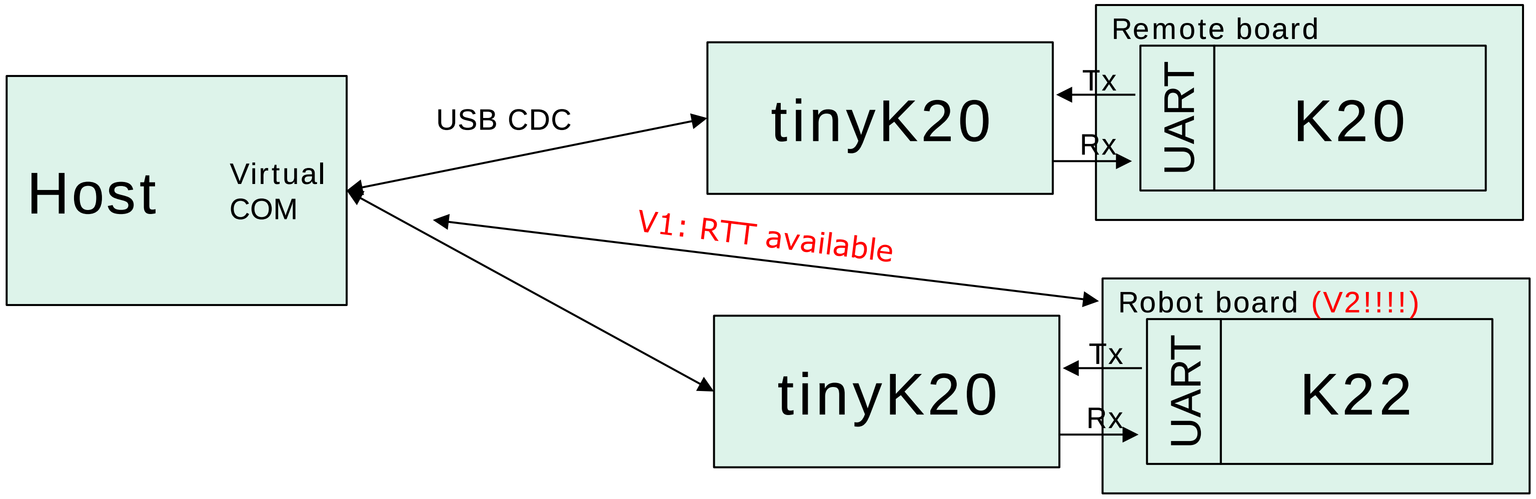
- Goal
  - Console
  - Send character/strings
  - Connection to host
  - Debug/Status messages



Console Hardware Routing

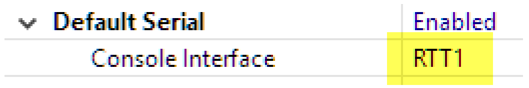
- Console (Terminal) connection to Host
- Using SCI (Serial Communication Interface)
- Robo V2: RX/TX on SWD. V1 ➔ Segger RTT



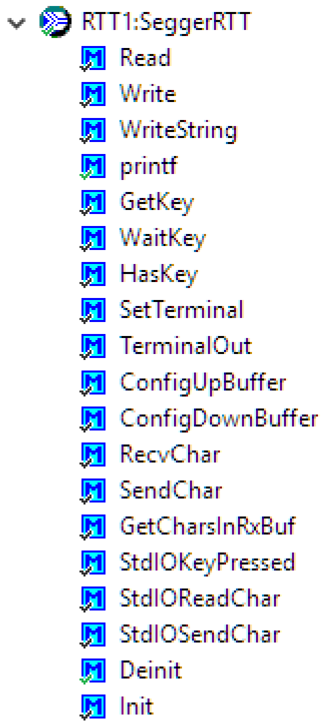


Segger RTT

- ‘virtual’ communication through debug interface
- https://mcuoneclipse.com/2015/07/07/using-segger-real-time-terminal-rtt-with-eclipse/
- Use RTT as default serial
- Client (inside Segger installation)
  - JLinkRTTClient
  - JLinkRTTViewer (GUI, Windows only)



Default Serial	Enabled
Console Interface	RTT1



RTT1:SeggerRTT
- Read
- Write
- WriteString
- printf
- GetKey
- WaitKey
- HasKey
- SetTerminal
- TerminalOut
- ConfigUpBuffer
- ConfigDownBuffer
- RecvChar
- SendChar
- GetCharsInRxBuf
- StdIOKeyPressed
- StdIOReadChar
- StdIOSendChar
- Deinit
- Init

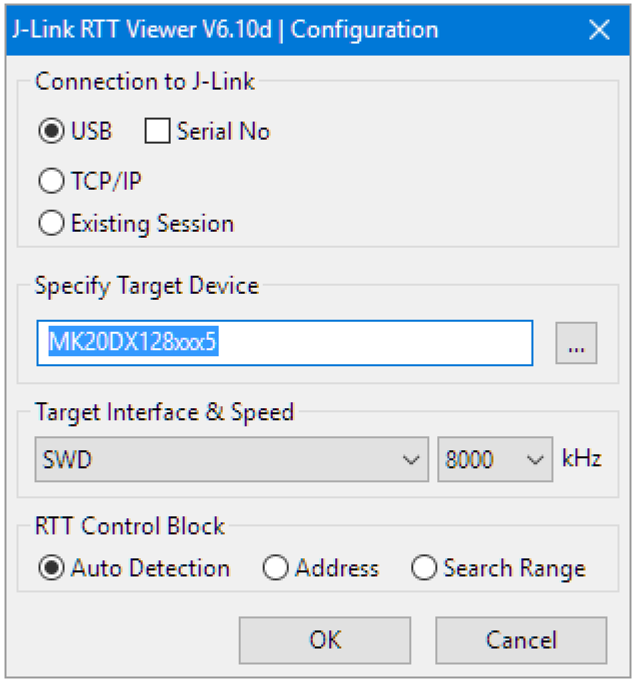
Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Technik & Architektur

Segger RTT Client

- Devices
  - Robo: MK22FX512xxx12
  - Remote: MK20DX128xxx5



J-Link RTT Viewer V6.10d | Configuration

Connection to J-Link

[● USB] ☐ Serial No

○ TCP/IP

○ Existing Session

Specify Target Device

MK20DX128xxx5

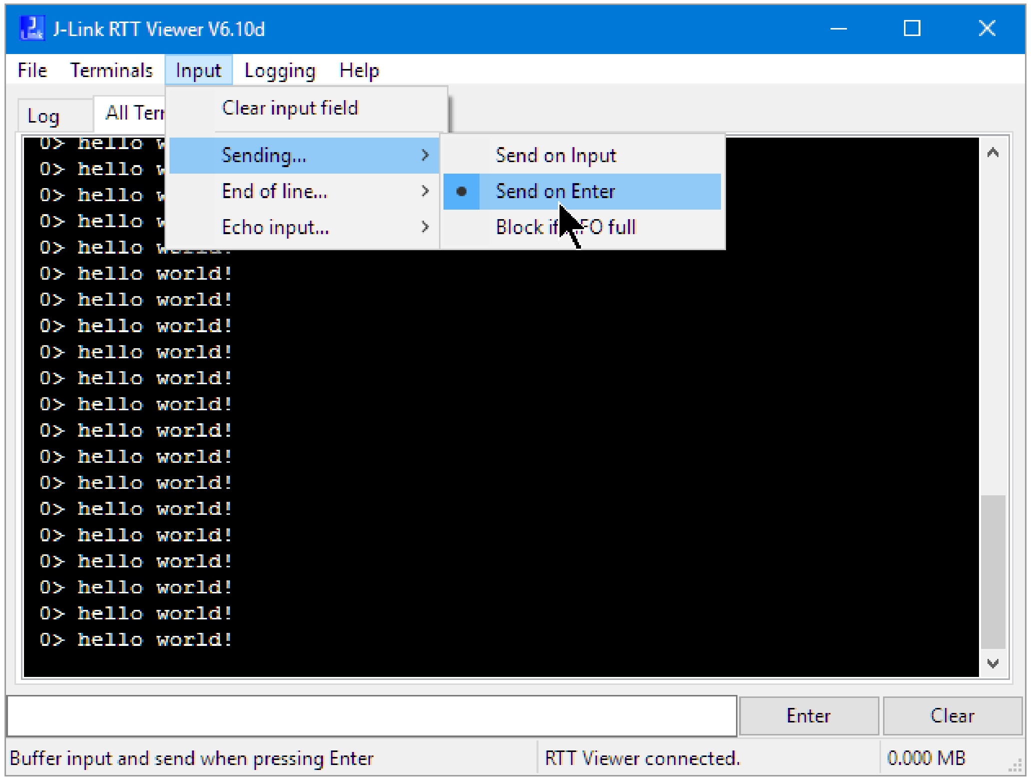
Target Interface & Speed

SWD 8000 kHz

RTT Control Block

[● Auto Detection] ○ Address ○ Search Range

OK Cancel



J-Link RTT Viewer V6.10d

File Terminals Input Logging Help

Log All Terminals

Input: Clear input field / Sending... / End of line... / Echo input...

Sending...: Send on Input / [● Send on Enter] / Block if FIFO full

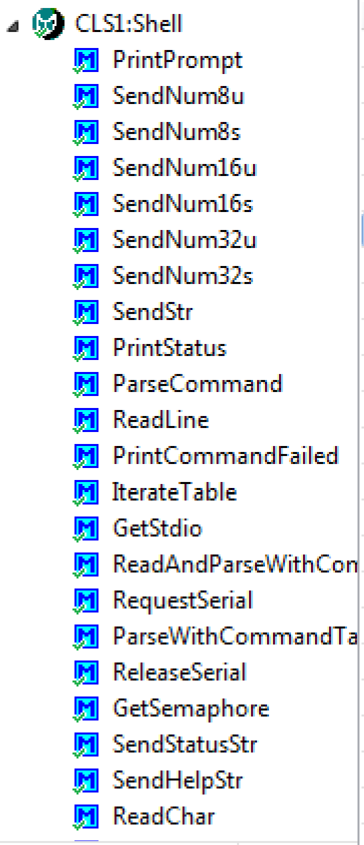
```
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
0> hello world!
```

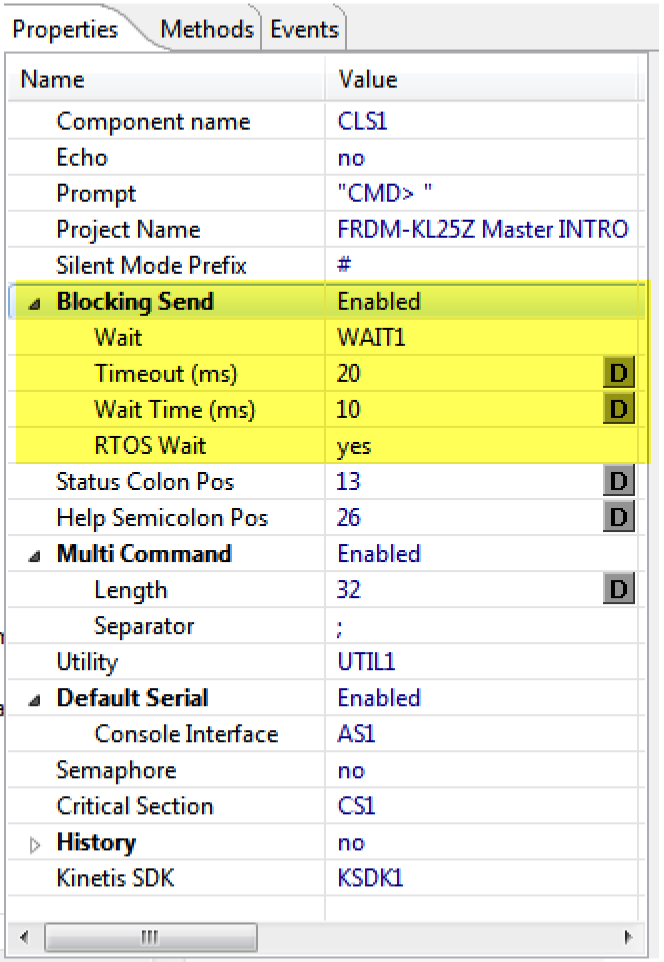
Enter Clear

Buffer input and send when pressing Enter | RTT Viewer connected. | 0.000 MB

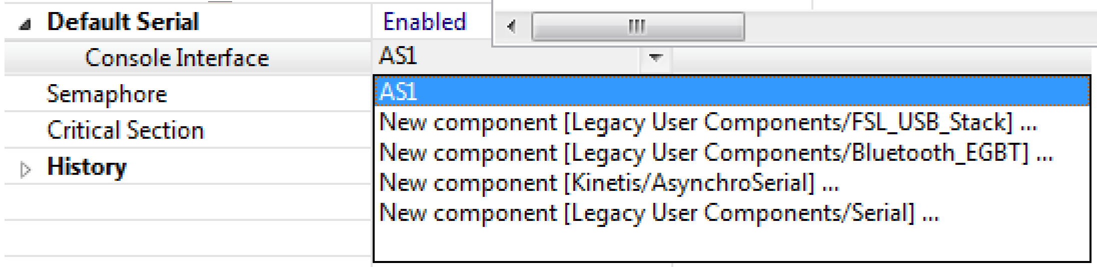
Shell Processor Expert Component

- Console Shell
  - **Serial (SCI / RS-232)**
  - **RTT**
  - **(USB)**
- Uses
  - Wait
  - Utility
  - CriticalSection
- Core of Shell
  - Prompt
  - Status
  - Help
  - Std I/O


CLS1:Shell
PrintPrompt
SendNum8u
SendNum8s
SendNum16u
SendNum16s
SendNum32u
SendNum32s
SendStr
PrintStatus
ParseCommand
ReadLine
PrintCommandFailed
IterateTable
GetStdio
ReadAndParseWithCon
RequestSerial
ParseWithCommandTa
ReleaseSerial
GetSemaphore
SendStatusStr
SendHelpStr
ReadChar


Properties | Methods | Events

Name	Value
Component name	CLS1
Echo	no
Prompt	"CMD> "
Project Name	FRDM-KL25Z Master INTRO
Silent Mode Prefix	#
Blocking Send	Enabled
Wait	WAIT1
Timeout (ms)	20
Wait Time (ms)	10
RTOS Wait	yes
Status Colon Pos	13
Help Semicolon Pos	26
Multi Command	Enabled
Length	32
Separator	;
Utility	UTIL1
Default Serial	Enabled
Console Interface	AS1
Semaphore	no
Critical Section	CS1
History	no
Kinetis SDK	KSDK1



Default Serial	Enabled
Console Interface	AS1
Semaphore	
Critical Section	
History	

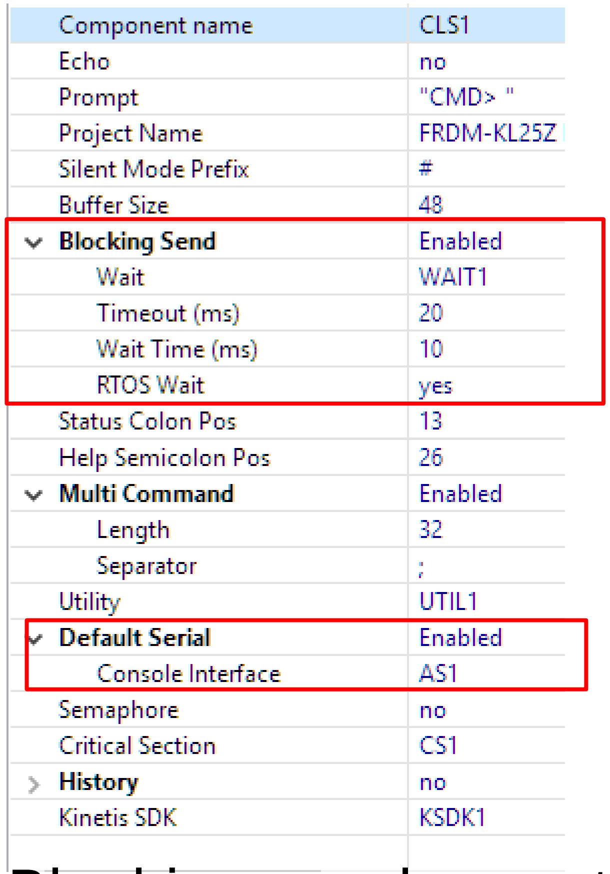
AS1
New component [Legacy User Components/FSL_USB_Stack] ...
New component [Legacy User Components/Bluetooth_EGBT] ...
New component [Kinetis/AsynchroSerial] ...
New component [Legacy User Components/Serial] ...

Lucerne University of Applied Sciences and Arts

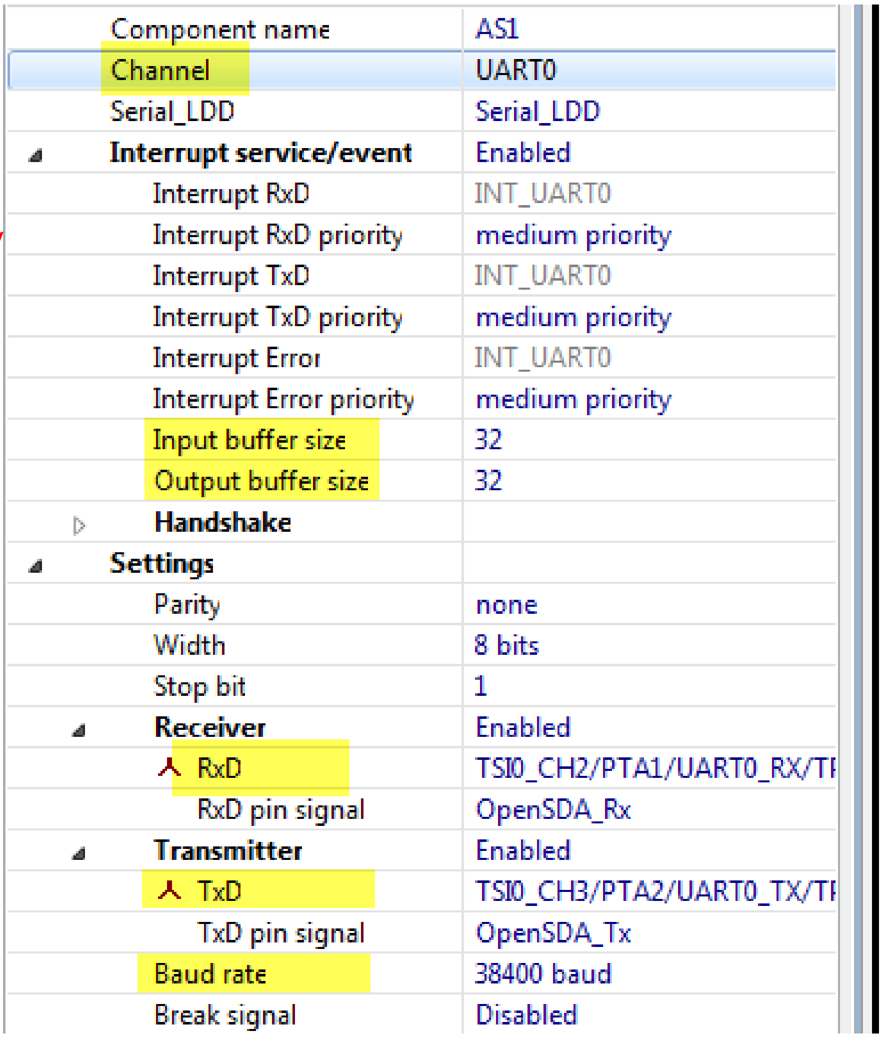
HOCHSCHULE LUZERN

Technik & Architektur

AsynchroSerial UART Interface



Component name	CLS1
Echo	no
Prompt	"CMD> "
Project Name	FRDM-KL25Z
Silent Mode Prefix	#
Buffer Size	48
Blocking Send	Enabled
Wait	WAIT1
Timeout (ms)	20
Wait Time (ms)	10
RTOS Wait	yes
Status Colon Pos	13
Help Semicolon Pos	26
Multi Command	Enabled
Length	32
Separator	;
Utility	UTIL1
Default Serial	Enabled
Console Interface	AS1
Semaphore	no
Critical Section	CS1
History	no
Kinetis SDK	KSDK1

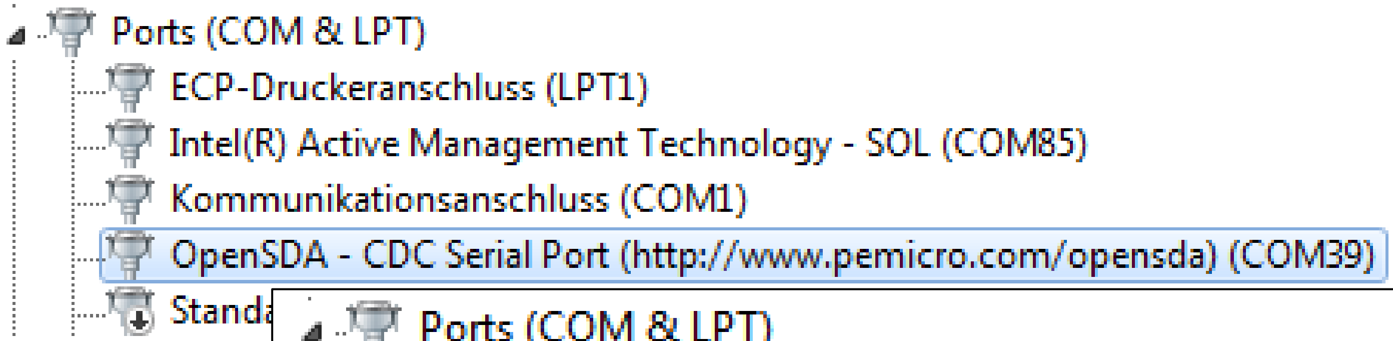


Component name	AS1
Channel	UART0
Serial_LDD	Serial_LDD
Interrupt service/event	Enabled
Interrupt RxD	INT_UART0
Interrupt RxD priority	medium priority
Interrupt TxD	INT_UART0
Interrupt TxD priority	medium priority
Interrupt Error	INT_UART0
Interrupt Error priority	medium priority
Input buffer size	32
Output buffer size	32
Handshake	
Settings	
Parity	none
Width	8 bits
Stop bit	1
Receiver	Enabled
RxD	TSI0_CH2/PTA1/UART0_RX/TI
RxD pin signal	OpenSDA_Rx
Transmitter	Enabled
TxD	TSI0_CH3/PTA2/UART0_TX/TI
TxD pin signal	OpenSDA_Tx
Baud rate	38400 baud
Break signal	Disabled

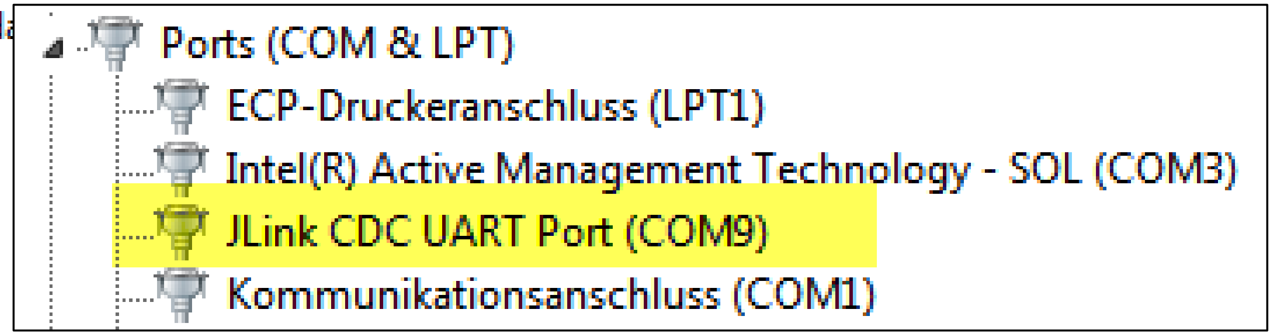
- Blocking send or not
- Channel
- ISR with ring buffer
- UART: RX and TX, Baud

Virtual COM Drivers (Windows)

- COM1 (normal RS-232)
- USB CDC enumerates as virtual COM port
- OpenSDA CDC Serial Port



Ports (COM & LPT)
ECP-Druckeranschluss (LPT1)
Intel(R) Active Management Technology - SOL (COM85)
Kommunikationsanschluss (COM1)
OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM39)
Standa

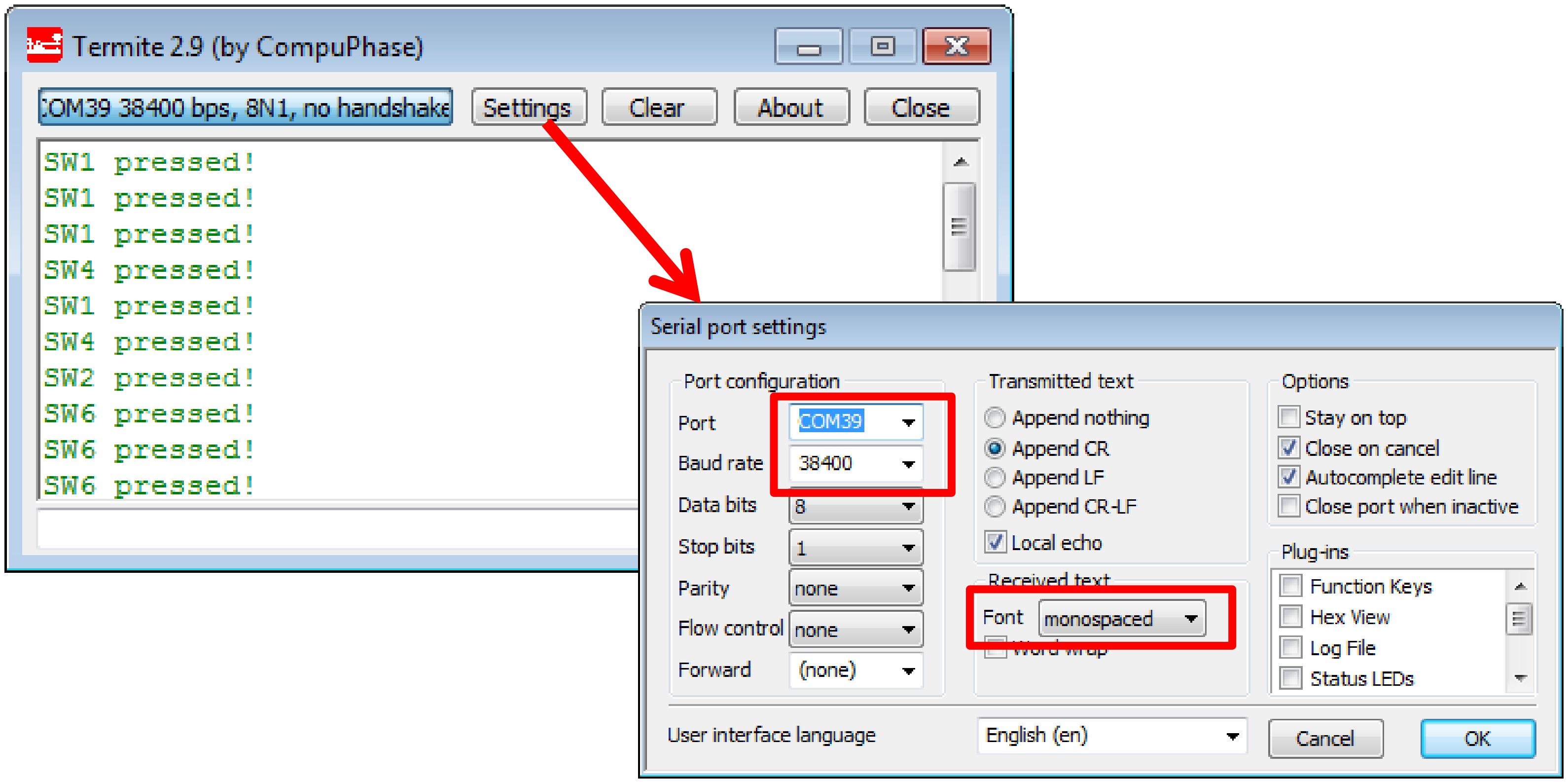


Ports (COM & LPT)
ECP-Druckeranschluss (LPT1)
Intel(R) Active Management Technology - SOL (COM3)
JLink CDC UART Port (COM9)
Kommunikationsanschluss (COM1)

Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

Terminal Program: Termite

- http://www.compuphase.com/software_termite.htm



```
SW1 pressed!
SW1 pressed!
SW1 pressed!
SW4 pressed!
SW1 pressed!
SW4 pressed!
SW2 pressed!
SW6 pressed!
SW6 pressed!
SW6 pressed!
```

Port: COM39
Baud rate: 38400
Data bits: 8
Stop bits: 1
Parity: none
Flow control: none
Forward: (none)

Append nothing ○
Append CR ●
Append LF ○
Append CR-LF ○
Local echo ✓

Font: monospaced

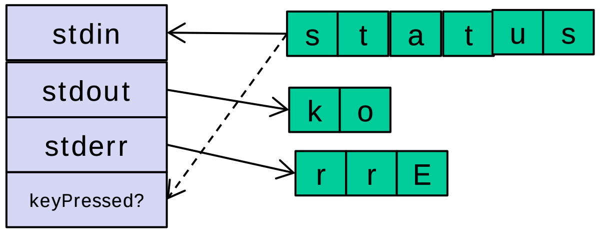
Stay on top ☐
Close on cancel ✓
Autocomplete edit line ✓
Close port when inactive ☐

Problem: Windows USB CDC

- Standard Windows CDC Driver Problem
- Problem if USB CDC COM Port open
    - Device stops communicating
    - Cable gets unplugged
    - Otherwise: COM port is blocked
- Solutions
    - Proprietary Serial driver (mbed.org, N/A)
    - Or:
        1. Have COM port closed (in Terminal Program)
        2. Unplug cable
        3. Plug cable in again
        4. Open COM Port
- Windows 10: much better ☺

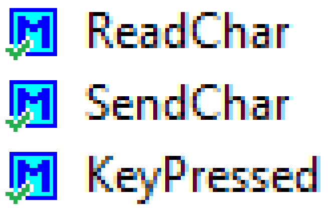
Shell Standard I/O

- I/O structure with callbacks
    - **Stdin**: read char
    - **Stdout**: write char
    - **Stderr**: write char
    - **KeyPressed**: char in stdin?
- Pointer to Functions
- Can be re-assigned
- Re-routing/logging/piping



```
typedef void (*CLS1_StdIO_In_FctType)(uint8_t *); /* Callback for an I/O input function. */
typedef void (*CLS1_StdIO_OutErr_FctType)(uint8_t); /* Callback for an output or error I/O function */
typedef bool (*CLS1_StdIO_KeyPressed_FctType)(void); /* Callback which returns true if a key has been
pressed */

CLS1_ConstStdIOTypePtr CLS1_GetStdio(void) {
  static CLS1_ConstStdIOType CLS1_stdio =
  {
    (CLS1_StdIO_In_FctType)CLS1_ReadChar, /* stdin */
    (CLS1_StdIO_OutErr_FctType)CLS1_SendChar, /* stdout */
    (CLS1_StdIO_OutErr_FctType)CLS1_SendChar, /* stderr */
    CLS1_KeyPressed /* if input is not empty */
  };
  return &CLS1_stdio;
}
```


ReadChar
SendChar
KeyPressed

Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Technik & Architektur

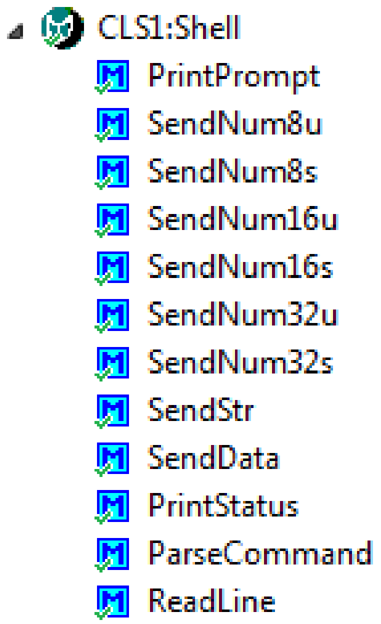
Writing Strings / Numbers

```
CLS1_SendStr("SW2 pressed!\r\n", CLS1_GetStdio()->stdOut);
```

```
void CLS1_SendStr(const uint8_t *str, CLS1_StdIO_OutErr_FctType io)
{
  while(*str!='\0') {
    io(*str++);
  }
}
```

```
void CLS1_SendNum32s(int32_t val, CLS1_StdIO_OutErr_FctType io)
{
  unsigned char buf[sizeof("-1234567890")];

  UTIL1_Num32sToStr(buf, sizeof(buf), val);
  CLS1_SendStr(buf, io);
}
```



CLS1:Shell
- PrintPrompt
- SendNum8u
- SendNum8s
- SendNum16u
- SendNum16s
- SendNum32u
- SendNum32s
- SendStr
- SendData
- PrintStatus
- ParseCommand
- ReadLine

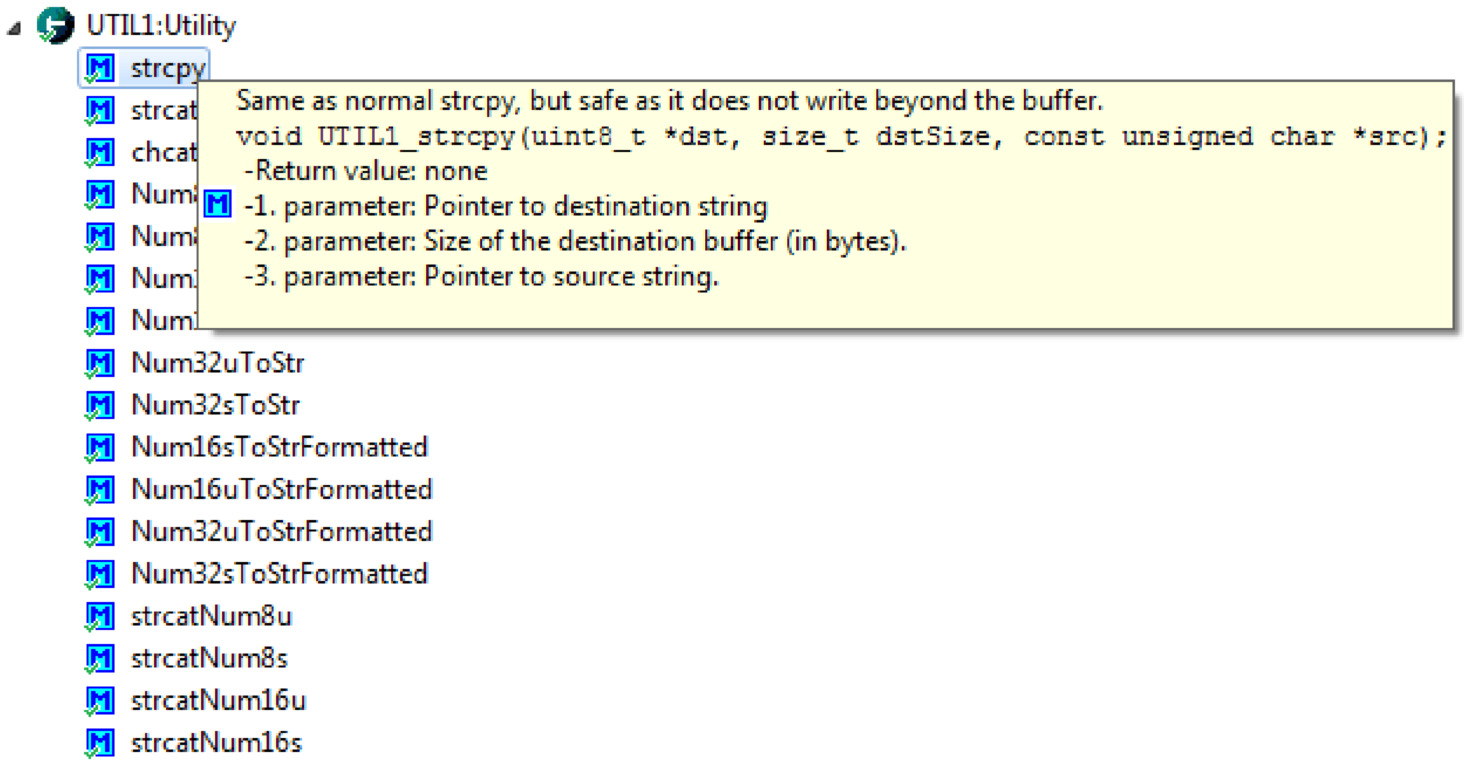
Lucerne University of Applied Sciences and Arts

HOCHSCHULE LUZERN

Technik & Architektur

Utility: Safe String Routines

- Buffer size as parameter
- Unlike normal strcpy(), does *not* cause buffer overflow
- Buffers always zero byte terminated

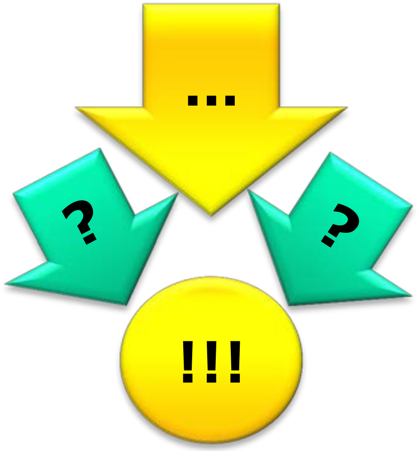

UTIL1:Utility
strcpy
strcat
chcat
Num
Num
Num
Num
Num32uToStr
Num32sToStr
Num16sToStrFormatted
Num16uToStrFormatted
Num32uToStrFormatted
Num32sToStrFormatted
strcatNum8u
strcatNum8s
strcatNum16u
strcatNum16s

Same as normal strcpy, but safe as it does not write beyond the buffer.
```
void UTIL1_strcpy(uint8_t *dst, size_t dstSize, const unsigned char *src);
```
-Return value: none
-1. parameter: Pointer to destination string
-2. parameter: Size of the destination buffer (in bytes).
-3. parameter: Pointer to source string.

Summary

- Problem: Write string for button pressed? Debug messages?

- RS-232/SCI, RTT, USB, ...
  - Bridge
  - Settings
  - Driver structure
  - Standard I/O
- Windows (<10) and USB CDC/COM
- Safe String Utility Functions



Lab: Console

- Add Shell component
- Use Console on host
    - Termite, putty, etc
- Print Messages for key events
- Explore writing numbers, strings, ...