



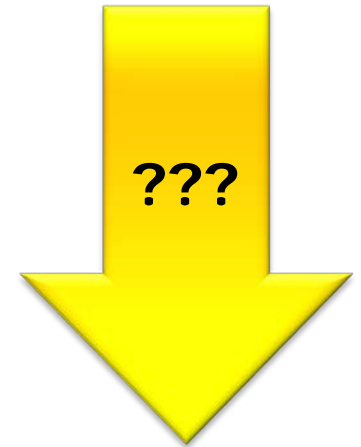
# Console

*"I do not not like printf()!"*

**Prof. Erich Styger**  
[erich.styger@hslu.ch](mailto:erich.styger@hslu.ch)  
+41 41 349 33 01

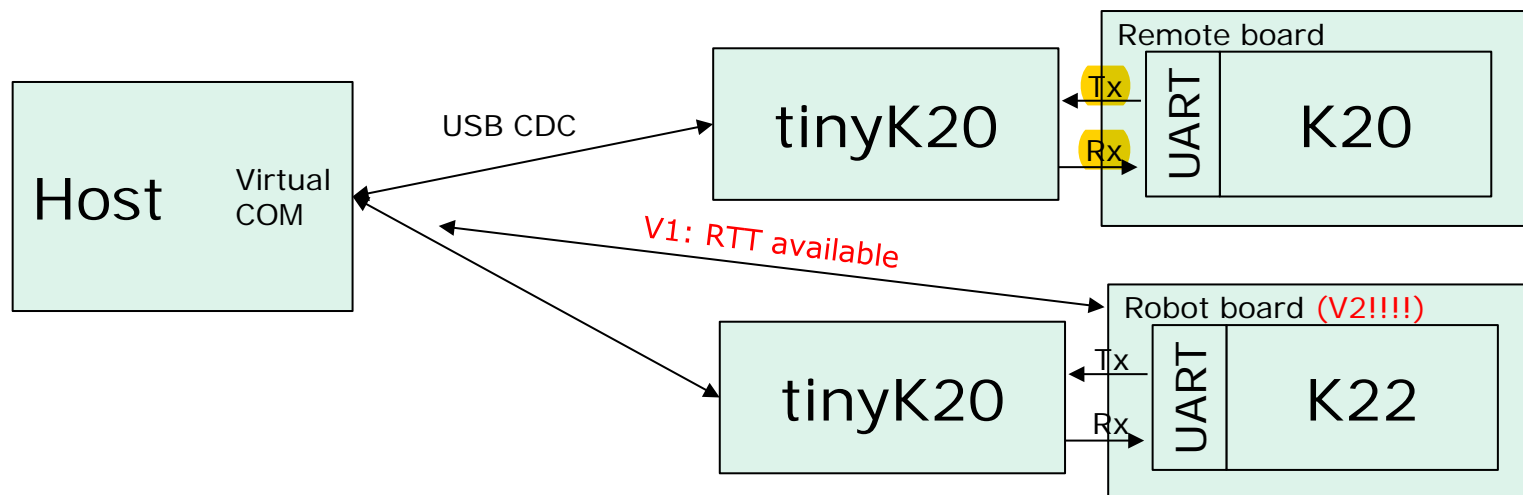
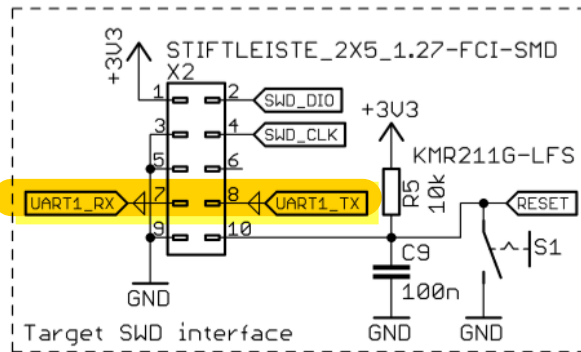
# Learning Goals

- Problem: Write string for button pressed?  
Debug messages?
- Goal
  - Console
  - Send character/strings
  - Connection to host
  - Debug/Status messages



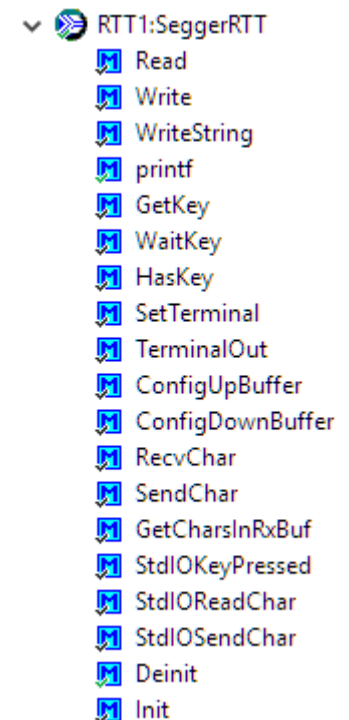
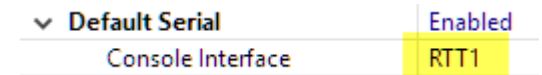
# Console Hardware Routing

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



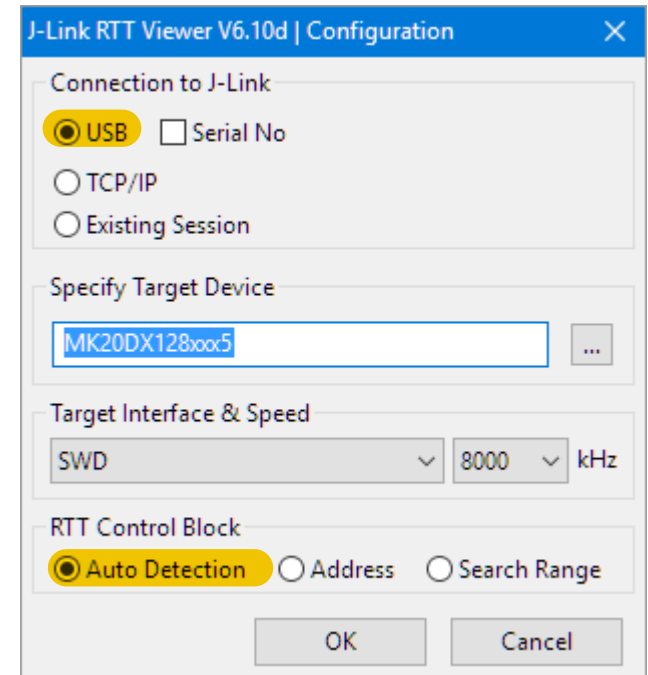
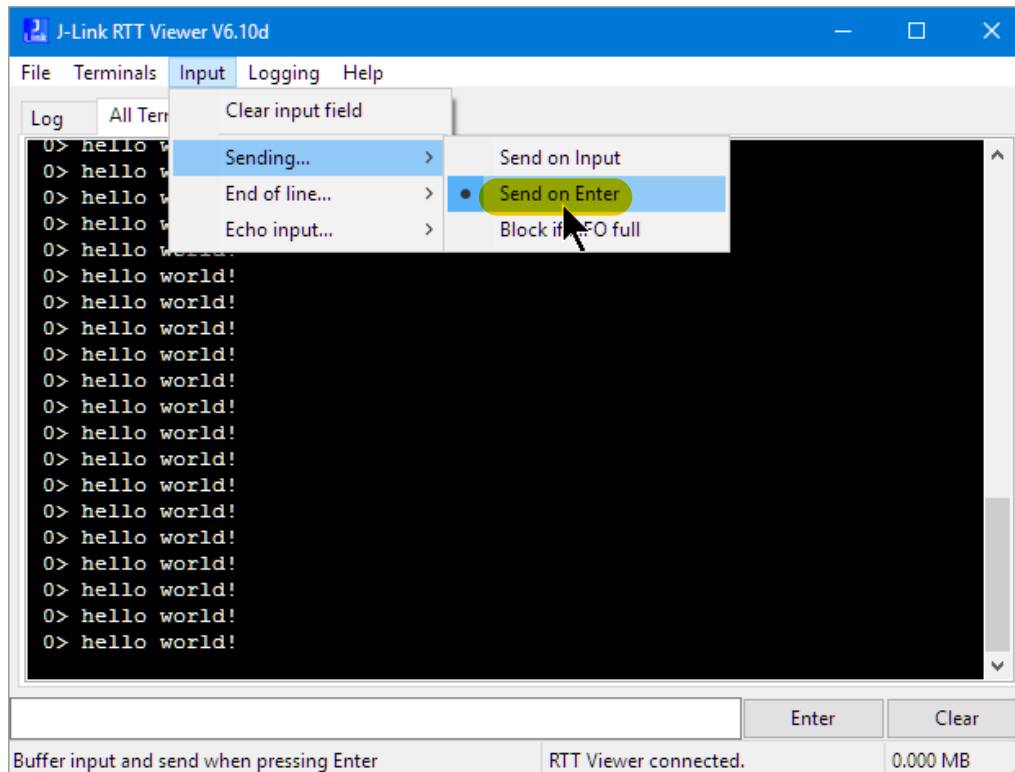
# Segger RTT

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



# Segger RTT Client

- Devices
  - Robo: MK22FX512xxx12
  - Remote: MK20DX128xxx5



# Shell Processor Expert Component

able to talk to different communication devices

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

get different  
communication channel  
->

**CLS1:Shell**

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

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

**Default Serial** Enabled

Console Interface AS1

Semaphore AS1

Critical Section

**History**

New component [Legacy User Components/FSL\_USB\_Stack] ...

New component [Legacy User Components/Bluetooth\_EGHT] ...

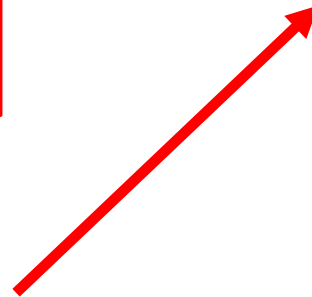
New component [Kinetis/AsynchroSerial] ...

New component [Legacy User Components/Serial] ...

# AsynchroSerial UART Interface

shell component does not implement the

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



Component name	AS1
<b>Channel</b>	UART0
Serial_LDD	Serial_LDD
▼ <b>Interrupt service/event</b>	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
▶ <b>Handshake</b>	
▼ <b>Settings</b>	
Parity	none
Width	8 bits
Stop bit	1
▼ <b>Receiver</b>	Enabled
RxD	TSI0_CH2/PTA1/UART0_RX/TF
Rx pin signal	OpenSDA_Rx
▼ <b>Transmitter</b>	Enabled
TxD	TSI0_CH3/PTA2/UART0_TX/TF
Tx 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)

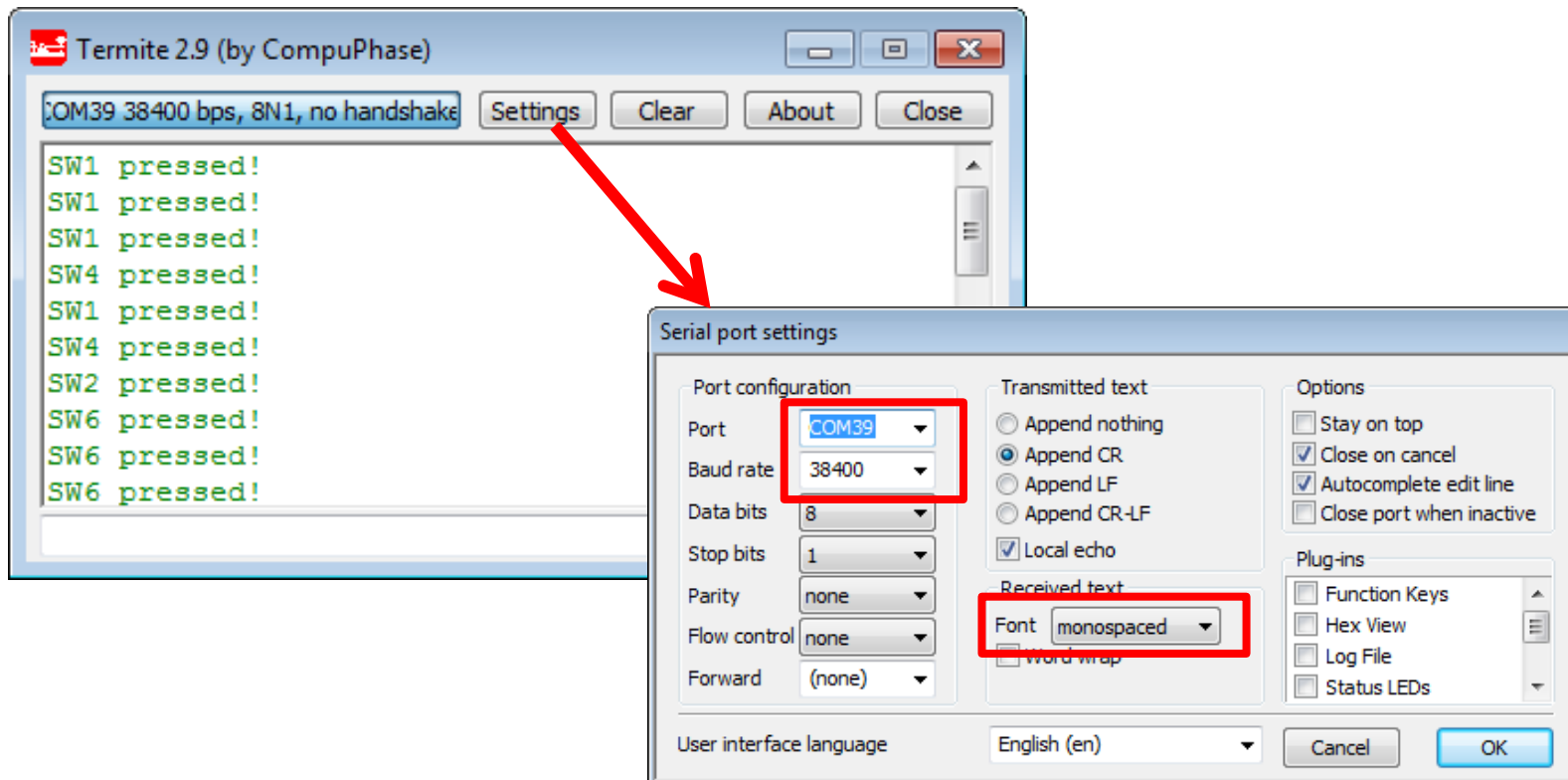
- COM1 (normal RS-232)
- USB CDC enumerates as virtual COM port
- OpenSDA CDC Serial Port





# Terminal Program: Terminate

- [http://www.compuphase.com/software\\_termite.htm](http://www.compuphase.com/software_termite.htm)



## Problem: Windows USB CDC

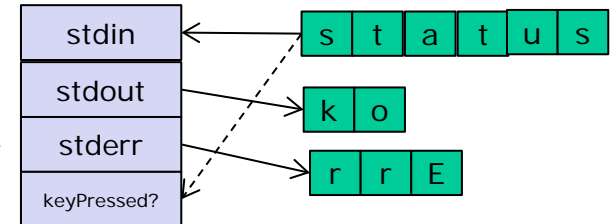
- Standard Windows CDC Driver Problem
- Problem if USB CDC COM Port open
  - Device stops communicating
  - Cable gets unplugged
  - Otherwise: COM port is blocked
- Solutions
  - Proprietary Serial driver (mbed.org, N/A)
  - Or:
    1. Have COM port closed (in Terminal Program)
    2. Unplug cable
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

WINDOWS 7

# 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

byte by byte  
character by character



```
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

# 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);
}
```

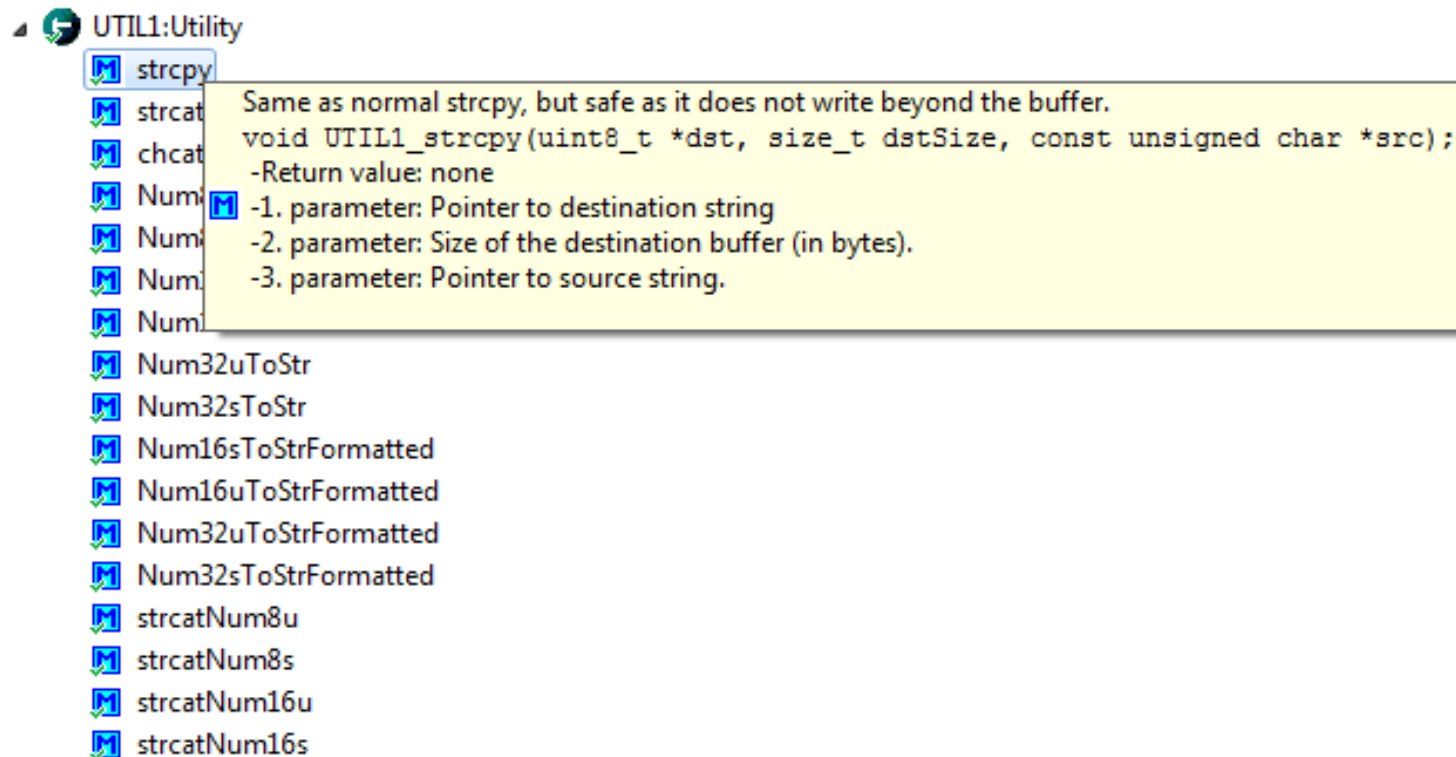


CLS1:Shell

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

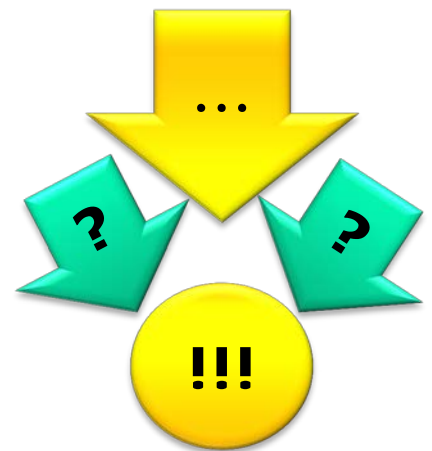
# Utility: Safe String Routines

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



# 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, ...

