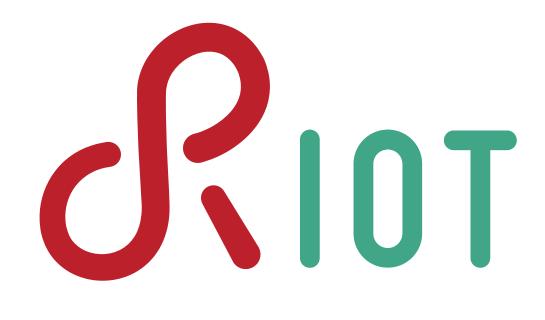
Shells on Constrained Devices Using CoAP to explore new UX paradigms for embedded shells

Bennet Blischke 11.07.2024



What is a shell?

- The interface between User and Operating System
 - Command Line Interface (CLI)
 - Graphical User Interface (GUI)

They form the outer shell of the computing system

image: Flaticon.com



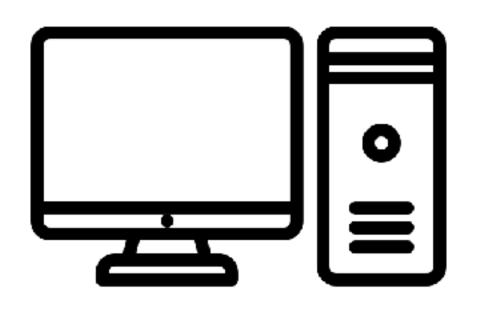
What is a shell?

- The interface between User and Operating System
 - Command Line Interface (CLI)
 - Graphical User Interface (GUI)

• They form the outer *shell* of the computing system

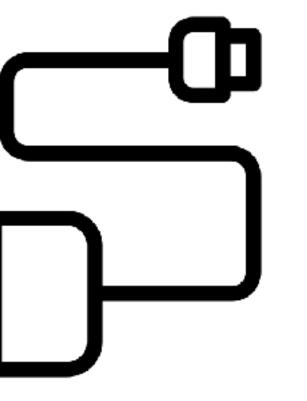
image: Flaticon.com

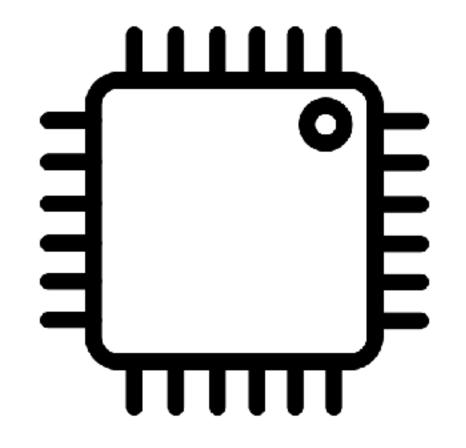




Programmer's computer

Typical Connection: UART over USB



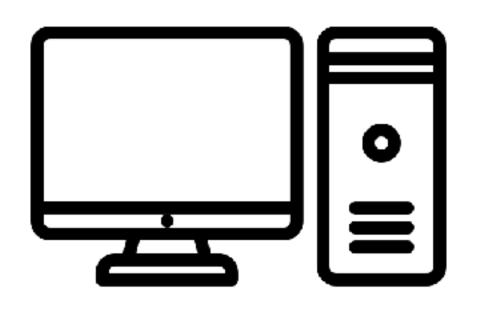


Embedded device

Exchange of ASCII characters

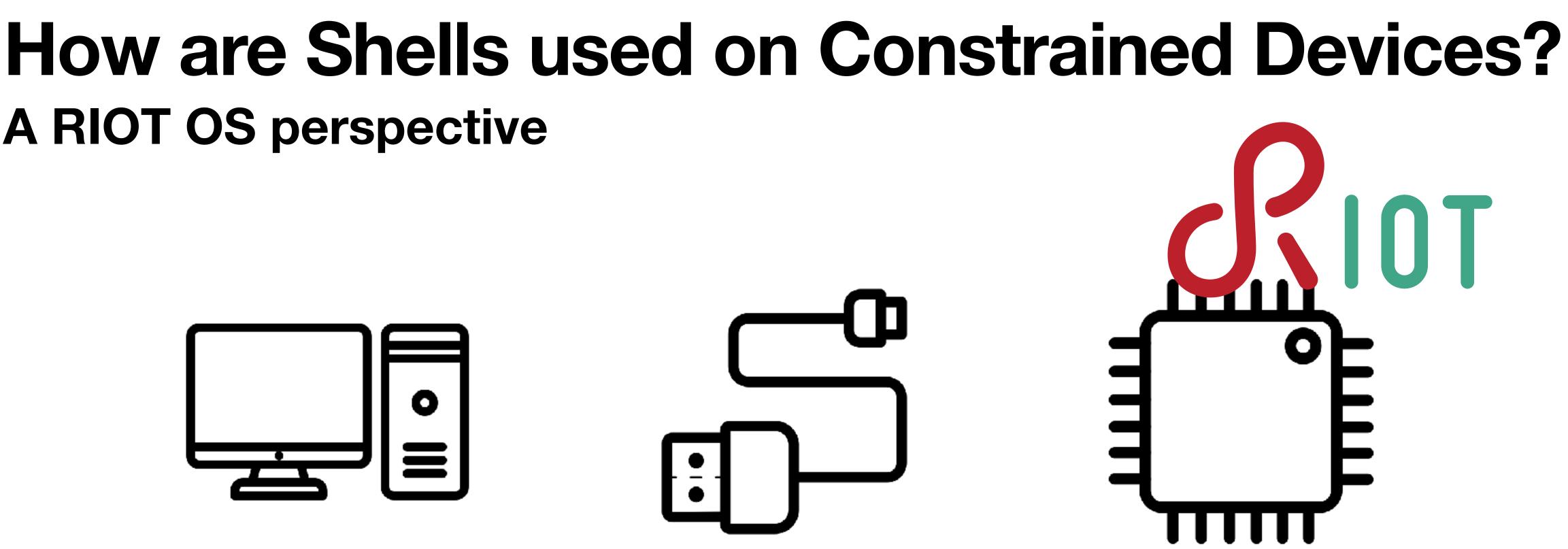


A RIOT OS perspective



Programmer's computer

Typical Connection: UART over USB



Embedded device

Exchange of ASCII characters



teufelchen@teufelchen-OptiPlex-7070:~/Programming/RIOT/examples/default\$ make term



teufelchen@teufelchen-OptiPlex-7070:~/Programming/RIOT/examples/default\$ make term /home/teufelchen/Programming/RIOT/dist/tools/pyterm/pyterm -p "/dev/ttyACM0" -b "115200" Twisted not available, please install it if you want to use pyterm's JSON capabilities 2024-07-09 11:05:16,417 # Connect to serial port /dev/ttyACM0 Welcome to pyterm! Type '/exit' to exit.



```
teufelchen@teufelchen-OptiPlex-7070:~/Programming/RIOT/examples/default$ make term
/home/teufelchen/Programming/RIOT/dist/tools/pyterm/pyterm -p "/dev/ttyACM0" -b "115200"
Twisted not available, please install it if you want to use pyterm's JSON capabilities
2024-07-09 11:05:16,417 # Connect to serial port /dev/ttyACM0
Welcome to pyterm!
Type '/exit' to exit.
help
2024-07-09 11:05:20,257 # help
                                             Description
2024-07-09 11:05:20,260 # Command
2024-07-09 11:05:20,263 # -----
2024-07-09 11:05:20,268 # ifconfig
                                             Configure network interfaces
2024-07-09 11:05:20,272 # nib
                                             Configure neighbor information base
                                             interact with layered PM subsystem
2024-07-09 11:05:20,277 # pm
2024-07-09 11:05:20,283 # ps regular
                                             Prints information about running threads.
2024-07-09 11:05:20,286 # reboot
                                             Reboot the node
                                             interact with sensors and actuators using SAUL
2024-07-09 11:05:20,292 # saul
                                             Sends a custom string as is over the link layer
2024-07-09 11:05:20,298 # txtsnd
                                             Prints current RIOT_VERSION
2024-07-09 11:05:20,302 # version
```

```
>
```



```
teufelchen@teufelchen-OptiPlex-7070:~/Programming/RIOT/examples/default$ make term
/home/teufelchen/Programming/RIOT/dist/tools/pyterm/pyterm -p "/dev/ttyACM0" -b "115200"
Twisted not available, please install it if you want to use pyterm's JSON capabilities
2024-07-09 11:05:16,417 # Connect to serial port /dev/ttyACM0
Welcome to pyterm!
Type '/exit' to exit.
help
2024-07-09 11:05:20,257 # help
                                             Description
2024-07-09 11:05:20,260 # Command
2024-07-09 11:05:20,263 # -----
                                             Configure network interfaces
2024-07-09 11:05:20,268 # ifconfig
2024-07-09 11:05:20,272 # nib
                                             Configure neighbor information base
                                             interact with layered PM subsystem
2024-07-09 11:05:20,277 # pm
                                             Prints information about running threads.
2024-07-09 11:05:20,283 # ps regular
2024-07-09 11:05:20,286 # reboot
                                             Reboot the node
                                             interact with sensors and actuators using SAUL
2024-07-09 11:05:20,292 # saul
                                             Sends a custom string as is over the link layer
2024-07-09 11:05:20,298 # txtsnd
2024-07-09 11:05:20,302 # version
                                             Prints current RIOT VERSION
> ps regular
2024-07-09 11:05:25,739 # ps regular
                                               stack (used) (free) base addr |current
2024-07-09 11:05:25,746 # name
2024-07-09 11:05:25,752 # isr stack
                                                  512 ( 224) ( 288) |0x2000000|0x20000154
2024-07-09 11:05:25,758 # main
                                                 1536 ( 648) ( 888) 0x20000434 0x200007cc
2024-07-09 11:05:25,764 # pktdump
                                                 1472 ( 176) ( 1296) 0x200019c4 0x20001ed4
2024-07-09 11:05:25,770 # 610
                                                  960 ( 216) (
2024-07-09 11:05:25,776 # ipv6
                                                  960 ( 440) (
2024-07-09 11:05:25,782 # udp
                                                  448 (196) (252) |0x200023d0|0x200024cc
2024-07-09 11:05:25,788 # nrf802154
                                                  896 ( 292) ( 604) |0x200012e4|0x200015a4
>
```

744) |0x20001fcc|0x200022cc 520) |0x20000a9c|0x20000cec



```
teufelchen@teufelchen-OptiPlex-7070:~/Programming/RIOT/examples/default$ make term
/home/teufelchen/Programming/RIOT/dist/tools/pyterm/pyterm -p "/dev/ttyACM0" -b "115200"
Twisted not available, please install it if you want to use pyterm's JSON capabilities
2024-07-09 11:05:16,417 # Connect to serial port /dev/ttyACM0
Welcome to pyterm!
Type '/exit' to exit.
help
2024-07-09 11:05:20,257 # help
2024-07-09 11:05:20,260 # Command
                                             Description
2024-07-09 11:05:20,263 # -----
                                             Configure network interfaces
2024-07-09 11:05:20,268 # ifconfig
                                             Configure neighbor information base
2024-07-09 11:05:20,272 # nib
                                             interact with layered PM subsystem
2024-07-09 11:05:20,277 # pm
                                             Prints information about running threads.
2024-07-09 11:05:20,283 # ps regular
2024-07-09 11:05:20,286 # reboot
                                             Reboot the node
                                             interact with sensors and actuators using SAUL
2024-07-09 11:05:20,292 # saul
                                             Sends a custom string as is over the link layer
2024-07-09 11:05:20,298 # txtsnd
                                             Prints current RIOT_VERSION
2024-07-09 11:05:20,302 # version
> ps regular
2024-07-09 11:05:25,739 # ps regular
                                               stack (used) (free) base addr |current
2024-07-09 11:05:25,746 # name
2024-07-09 11:05:25,752 # isr stack
                                                  512 ( 224) ( 288) |0x2000000|0x20000154
                                                 1536 ( 648) ( 888) 0x20000434 0x200007cc
2024-07-09 11:05:25,758 # main
2024-07-09 11:05:25,764 # pktdump
                                                 1472 ( 176) ( 1296) 0x200019c4 0x20001ed4
2024-07-09 11:05:25,770 # 6lo
                                                  960 ( 216) ( 744) |0x20001fcc|0x200022cc
2024-07-09 11:05:25,776 # ipv6
                                                                520) |0x20000a9c|0x20000cec
                                                  960 ( 440) (
2024-07-09 11:05:25,782 # udp
                                                  448 (196) (252) |0x200023d0|0x200024cc
2024-07-09 11:05:25,788 # nrf802154
                                                  896 ( 292) ( 604) |0x200012e4|0x200015a4
> reboot
2024-07-09 11:05:35,533 # reboot
2024-07-09 11:05:35,542 # main(): This is RIOT! (Version: 2024.07-devel-135-g46924-feat/slipconf)
2024-07-09 11:05:35,543 # Welcome to RIOT!
>
```



Shortcomings

- ROM / RAM overhead due to string-parsing on the constrained device
- It's complicated to exchange binary data via an ASCII stream
- No correlation between received text and origin
 - First thread starts to write
 - Another thread interrupts & writes its own message
 - First thread continues and finishes its message
 - The re out of order sult is hard to read and

Shortcomings

- Long running and highly asynchronous tasks block the shell
- Difficulties in unit-/automated testing of shell components
- Signalling of success and failure of commands

The Issue with String Parsing **ROM / RAM overhead due to string parsing on the constrained device**

- Strings require a lot of ROM, one byte per character
 - That just was 53 bytes!
- Even worse, when formatting strings. Example:
 - printf("Year %d", year);
 - 7 bytes for the format string
 - 9 bytes for the result "Year 2024"

The Issue with String Parsing ROM / RAM overhead due to string parsing on the constrained device

- We do a lot with strings in our shells
- A command takes strings as parameter: ifconfig(int argc, char *argv[])
- Now, parse "ifconfig 7 set channel 0x21"
- A command outputs its result by printing formatted strings
 - Try printing a table of running threads

How do We Improve? **Format Strings**

- Deferred Formatting:
 - 1. Assign all format strings a short ID
 - 2. Only compile the ID into the firmware
 - 3. When printing, send the ID + format values as it
 - 4. On the host, look up corresponding strings for received IDs
 - 5. Format and print on the host side



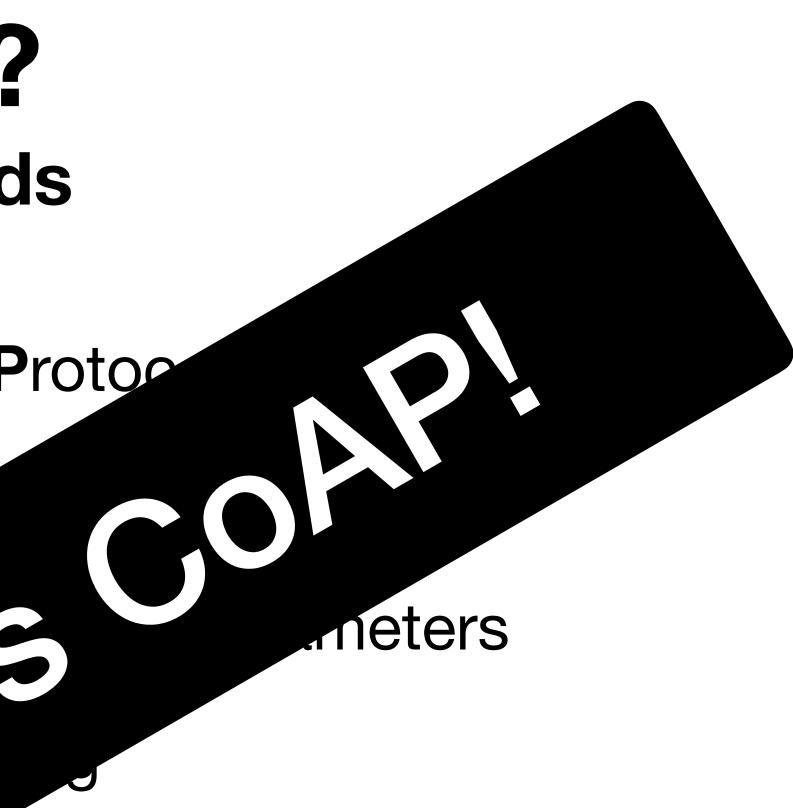
For an example see: https://defmt.ferrous-systems.com/

How do We Improve? **Correlating output to commands**

- We need a lightweight Application Protocol
- Suitable for Constrained devices
- Allows to issue a command with optional parameters
- Provides request & response matching
- Can handle errors:
 - On the host side e.g. "invalid parameter for command"
 - On the IoT side e.g. "failed to enable SPI"

How do We Improve? **Correlating output to commands**

- We need a lightweight Application Protoc
- Suitable for Constrained devices
- Allows to issue a comman
- Provides request
- Can handle er
 - On the host lacksquare
 - On the IoT side e.g. "failed to enable SPI"

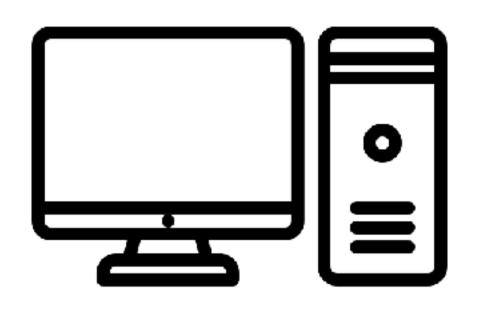


g. "invalid parameter for command"

Even More CoAP Benefits!

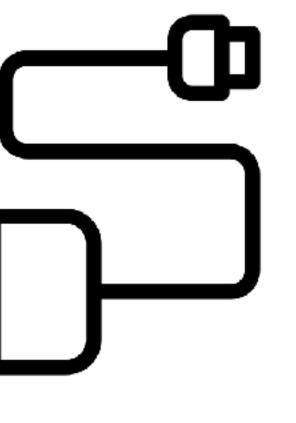
- Completely asynchron: Request now, get the response later!
- With CoAP Observe, we can have timer, events and interrupts in our shell
- Big binary payload? CoAP block-wise transfer saves the day
- Robust ecosystem due to related standards:
 - Discover shell commands via /.well-known/core
 - Describe commands using CoRE Link Format

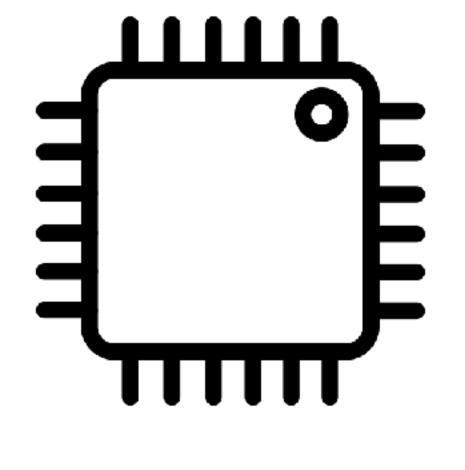
CoAP transports via UDP... ... but we don't have UDP, not even IPv6!



Programmer's computer

Typical Connection: UART over USB





Embedded device

SLIP Serial Line IP

- Very old RFC from 1988
- Short framing and escaping protocol
- Easily send IP frames over serial / UART

oi JART

SLIP **Serial Line IP**

- Very old RFC from 1988
- Short framing and escaping protocol
- Easily send IP frames over serial / UART

Can we speak CoAP over UDP over IP via SLIP?

SLIP **Serial Line IP**

- Very old RFC from 1988
- Short framing and escaping protocol
- Easily send IP frames over serial / UART

- Can we speak CoAP over UDP over IP via SLIP?
 - Yes!
 - But that's a lot of overhead, we just want CoAP

- New draft from 2019

Using an UART interface for diagnostics, configuration, and packet transfer

Builds on top of SLIP, but adds multiplexing for diagnostic and configuration

- New draft from 2019
- **Diagnostic** messages are just UTF-8 strings

Using an UART interface for diagnostics, configuration, and packet transfer

Builds on top of SLIP but adds multiplexing for diagnostic and configuration

• Neat! We can use that for backwards compatibility to the existing shell!

- New draft from 2019
- **Diagnostic** messages are just UTF-8 strings
- Configuration messages are CoAP packages!

Using an UART interface for diagnostics, configuration, and packet transfer

Builds on top of SLIP but adds multiplexing for diagnostic and configuration

• Neat! We can use that for backwards compatibility to the existing shell!

This also solves the issue of finding the IP addresses of both participants

- New draft from 2019
- **Diagnostic** messages are just UTF-8 strings
- Configuration messages are CoAP packages!
- And if we want, we can still exchange IP frames

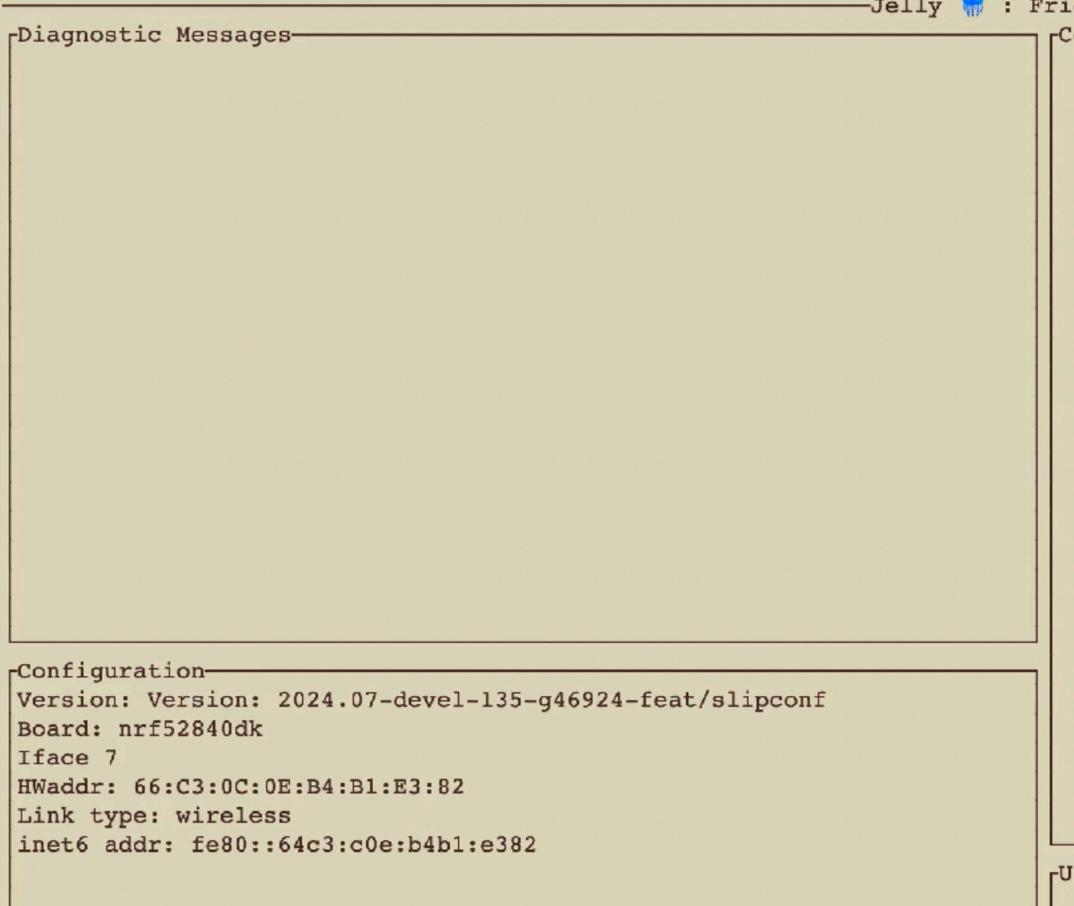
Using an UART interface for diagnostics, configuration, and packet transfer

Builds on top of SLIP but adds multiplexing for diagnostic and configuration

• Neat! We can use that for backwards compatibility to the existing shell!

This also solves the issue of finding the IP addresses of both participants

Let's prototype! What does a SLIPMUX client look like?



-Jelly 🖬 : Friendly SLIPMUX for RIOT OS-

User Input-



Backwards compatible: help

	Jelly	ANAb ANAb	: F	ri
[Diagnostic Messages				Г
Configuration-			_	
Version: Version: 2024.07-devel-135-g46924-feat/slipcon	f			
Board: nrf52840dk				
Iface 7				
HWaddr: 66:C3:0C:0E:B4:B1:E3:82 Link type: wireless				
inet6 addr: fe80::64c3:c0e:b4b1:e382				L
				г٦
				ł
				V

iendly SLIPMUX for RIOT OS-Configuration Messages------



Backwards compatible: help

	Jelly 📅 : Fr
Diagnostic Messages- help	
Command	Description
ifconfig	Configure network interfaces
nib	Configure neighbor information base
pm	interact with layered PM subsystem
ps	Prints information about running threads.
reboot	Reboot the node
saul	interact with sensors and actuators using SAUL
txtsnd	Sends a custom string as is over the link layer
version	Prints current RIOT_VERSION
>	
-Configuration	
	24.07-devel-135-g46924-feat/slipconf
Board: nrf52840dk	24.07-devel-155-940524-lede/Stipconi
Iface 7	
HWaddr: 66:C3:0C:0E:	B4:B1:E3:82
Link type: wireless	
inet6 addr: fe80::64	c3:c0e:b4b1:e382

endly SLIPMUX for RIOT OSonfiguration Messages------

[User Input-



New: Direct CoAP Requests GET /.well-known/core

	Jelly 🚻 :	FT:
Diagnostic Messages- help Command	Description	
ifconfig	Configure network interfaces	
nib	Configure neighbor information base	
pm	interact with layered PM subsystem	
ps	Prints information about running threads.	
reboot	Reboot the node	
saul	interact with sensors and actuators using SAUL	
txtsnd	Sends a custom string as is over the link layer	
version	Prints current RIOT_VERSION	
>		
-Configuration-		
-	24.07-devel-135-g46924-feat/slipconf	
Board: nrf52840dk	24.07-devel-155-940524-read/strpcom	
Iface 7		
HWaddr: 66:C3:0C:0E:	B4:B1:E3:82	
Link type: wireless		
inet6 addr: fe80::64	c3:c0e:b4b1:e382	L
		1
		L
		$-\overline{\checkmark}$

iendly SLIPMUX for RIOT OS-Configuration Messages------

Jser Input-/.well-known/core



New: Direct CoAP Requests GET /.well-known/core

	Jelly 🔤 :	Fri
Diagnostic Messages- help		
Command	Description	
ifconfig	Configure network interfaces	
nib	Configure neighbor information base	111
pm	interact with layered PM subsystem	
ps	Prints information about running threads.	
reboot	Reboot the node	111
saul	interact with sensors and actuators using SAUL	
txtsnd	Sends a custom string as is over the link layer	
version	Prints current RIOT_VERSION	111
>		
Configuration		
-	024.07-devel-135-g46924-feat/slipconf	
Board: nrf52840dk		
Iface 7		
HWaddr: 66:C3:0C:0E:	:B4:B1:E3:82	
Link type: wireless		
inet6 addr: fe80::64	4c3:c0e:b4b1:e382	
		Γ ^U

endly SLIPMUX for RIOT OSonfiguration Messages-<- Req(Get /.well-known/core)[0x0001]--> Res(Content/ApplicationLinkFormat)[0x0001] </sha256> </riot/ver> </riot/value> </riot/board> </echo/></.well-known/ifconfig> </.well-known/core> </config/ps> </shell/nib> </shell/ifconfig> </shell/txtsnd> </shell/pm> </shell/ps> </shell/saul> </shell/version>

</shell/reboot>

ser Input-



Backwards Compatible: ps

T-11--

-Diagnostic Messages-								
ifconfig	Configure	e 1	networ]	٤.	interfa	ces		
nib	Configure	e ı	neighbo	or	inform	ation base		
pm	interact	W	ith lay	/e	red PM	subsystem		
ps	Prints in	nfo	ormatio	on	about	running three	eads.	
reboot	Reboot th	he	node					
saul	interact	W	ith ser	ns	ors and	actuators a	using SAUL	
txtsnd	Sends a	cu	stom st	er.	ing as	is over the	link layer	
version	Prints c	ur	rent RI	01	r_versi	ON		
> ps								
name	stack	(used)	(free)	base addr	current	
isr_stack	512	(224)	(288)	0x20000000	0x20000154	
main	1536	(668)	(868)	0x2000035c	0x200006f4	
pktdump	1472	(176)	(1296)	0x20002f44	0x20003454	
610	960	(216)	(744)	0x2000354c	0x2000384c	
іриб	960	(440)	(520)	0x20000a9c	0x20000cec	
udp	448	(196)	(252)	0x20003950	0x20003a4c	
Slipmux CoAP server	1024	(548)	(476)	0x20003f24	0x200041f4	
slipdev	896	(236)	(660)	0x20001854	0x20001b14	
nrf802154	896	(316)	(580)	0x200012e4	0x200015a4	
>								

Configuration Version: Version: 2024.07-devel-135-g46924-feat/slipconf Board: nrf52840dk Iface 7 HWaddr: 66:C3:0C:0E:B4:B1:E3:82 Link type: wireless inet6 addr: fe80::64c3:c0e:b4b1:e382 endly SLIPMUX for RIOT OSonfiguration Messages-<- Req(Get /.well-known/core)[0x0001]--> Res(Content/ApplicationLinkFormat)[0x0001] </sha256> </riot/ver> </riot/value> </riot/board> </echo/></.well-known/ifconfig> </.well-known/core> </config/ps> </shell/nib> </shell/ifconfig> </shell/txtsnd> </shell/pm> </shell/ps> </shell/saul> </shell/version>

</shell/reboot>

User Input-



Commands via CoAP: ps

	Jelly 🖬 : H	Friendly SLIPMUX for RIOT OS
[Diagnostic Messages-		Configuration Messages
ifconfig	Configure network interfaces	
nib	Configure neighbor information base	
pm	interact with layered PM subsystem	
ps	Prints information about running threads.	.well-known/ifconfig
reboot	Reboot the node	.well-known/core
saul	interact with sensors and actuators using SAUL	
txtsnd	Sends a custom string as is over the link layer	
version	Prints current RIOT_VERSION	
> ps		
name	stack (used) (free) base addr current	
isr_stack	512 (224) (288) 0x2000000 0x20000154	
main	1536 (668) (868) 0x2000035c 0x200006f4	
pktdump	1472 (176) (1296) 0x20002f44 0x20003454	
610	960 (216) (744) 0x2000354c 0x2000384c	
ipv6	960 (440) (520) 0x20000a9c 0x20000cec	
udp	448 (196) (252) 0x20003950 0x20003a4c	Command: ps-
Slipmux CoAP server	1024 (548) (476) 0x20003f24 0x200041f4	name stack used free start SP
slipdev	896 (236) (660) 0x20001854 0x20001b14	isr_stack 512 224 288 0x2000000 0x20000154
nrf802154	896 (316) (580) 0x200012e4 0x200015a4	main 1536 668 868 0x2000035c 0x200006f4
>		pktdump 1472 176 1296 0x20002f44 0x20003488

Configuration Version: Version: 2024.07-devel-135-g46924-feat/slipconf Board: nrf52840dk Iface 7 HWaddr: 66:C3:0C:0E:B4:B1:E3:82 Link type: wireless inet6 addr: fe80::64c3:c0e:b4b1:e382

Command: ps-									
name	stack	used	free	start	SP				
isr_stack	512	224	288	0x20000000	0x20000154				
main	1536	668	868	0x2000035c	0x200006f4				
pktdump	1472	176	1296	0x20002f44	0x20003488				
610	960	216	744	0x2000354c	0x20003868				
іриб	960	440	520	0x20000a9c	0x20000cd8				
udp	448	196	252	0x20003950	0x20003a80				
Slipmux CoAP server	1024	548	476	0x20003f24	0x20004134				
slipdev	896	236	660	0x20001854	0x20001b1c				
nrf802154	896	316	580	0x200012e4	0x2000155c				

User Input-



Backwards Compatible: reboot

	Jelly 📅 : 🔅	Friendly SLIPMUX for RI	IOT OS-				
[Diagnostic Messages-		Configuration Message	es				
ps	Prints information about running threads.	<pre></pre>					4
reboot	Reboot the node	<pre></pre>					
saul	interact with sensors and actuators using SAUL						
txtsnd	Sends a custom string as is over the link layer	<pre>/.well-known/ifco</pre>	onfig>				
version	Prints current RIOT_VERSION	<pre><!--.well-known/core</pre--></pre>	9>				
> ps		<pre></pre>					
name	stack (used) (free) base addr current	<pre></pre>					
isr_stack	512 (224) (288) 0x2000000 0x20000154	<pre></pre>					
main	1536 (668) (868) 0x2000035c 0x200006f4						
pktdump	1472 (176) (1296) 0x20002f44 0x20003454						
610	960 (216) (744) 0x2000354c 0x2000384c	<pre></pre>					
ipv6	960 (440) (520) 0x20000a9c 0x20000cec	<pre></pre>					
udp	448 (196) (252) 0x20003950 0x20003a4c	<pre></pre>					
Slipmux CoAP server	1024 (548) (476) 0x20003f24 0x200041f4	<pre></pre>					
slipdev	896 (236) (660) 0x20001854 0x20001b14						
nrf802154	896 (316) (580) 0x200012e4 0x200015a4	Command: ps					
> reboot		name	stack	used	free	start SP	
main(): This is RIOT	(Version: 2024.07-devel-135-g46924-feat/slipconf	isr_stack	· · · · · · · · · · · · · · · · · · ·			0x2000000000x20000154	
Welcome to RIOT!		main	1536	668	868	0x2000035c 0x200006f4	
>		pktdump	1472	176	1296	0x20002f44 0x20003488	
		J 610	960	216	744	0x2000354c 0x20003868	
[Configuration		j ipv6	960	440	520	0x20000a9c 0x20000cd8	
Version: Version: 202	udp	448	196	252	0x20003950 0x20003a80		
Board: nrf52840dk	Slipmux CoAP server	1024	548	476	0x20003f24 0x20004134		
Iface 7		slipdev	896	236	660	0x20001854 0x20001b1c	
HWaddr: 66:C3:0C:0E:1	34:B1:E3:82	nrf802154	896	316	580	0x200012e4 0x2000155c	
Link type: wireless							

THUE ALBON WITCHORD inet6 addr: fe80::64c3:c0e:b4b1:e382

User Input-



Commands via CoAP: reboot Async!

	Jelly 📊 :	Friendly SLIPMUX for R	IOT OS			
[Diagnostic Messages-		Configuration Message	es			
saul	interact with sensors and actuators using SAUL	.well-known/ifco</td <td>onfig></td> <td></td> <td></td> <td></td>	onfig>			
txtsnd	Sends a custom string as is over the link layer	.well-known/core</td <td>e></td> <td></td> <td></td> <td></td>	e>			
version	Prints current RIOT_VERSION					
> ps						
name	stack (used) (free) base addr current	<pre></pre>				
isr_stack	512 (224) (288) 0x2000000 0x20000154					
main	1536 (668) (868) 0x2000035c 0x200006f4					
pktdump	1472 (176) (1296) 0x20002f44 0x20003454					
610	960 (216) (744) 0x2000354c 0x2000384c					
ipv6	960 (440) (520) 0x20000a9c 0x20000cec					
udp	448 (196) (252) 0x20003950 0x20003a4c					
Slipmux CoAP server	1024 (548) (476) 0x20003f24 0x200041f4					
slipdev	896 (236) (660) 0x20001854 0x20001b14	Command: ps-				
nrf802154	896 (316) (580) 0x200012e4 0x200015a4	name	stack used	free s	tart SP	
> reboot		isr_stack	512 224	288 0	x20000000 0x20000154	
main(): This is RIOT!	(Version: 2024.07-devel-135-g46924-feat/slipconf	main	1536 668	868 0:	x2000035c 0x200006f4	
Welcome to RIOT!		pktdump	1472 176	1296 0:	x20002f44 0x20003488	
> main(): This is RIO	T! (Version: 2024.07-devel-135-g46924-feat/slipco	610	960 216	744 0:	x2000354c 0x20003868	
Welcome to RIOT!		іруб	960 440	520 0:	x20000a9c 0x20000cd8	
>		udp	448 196	252 0:	x20003950 0x20003a80	
		Slipmux CoAP server	1024 548	476 0:	x20003f24 0x20004134	
[Configuration		slipdev	896 236	660 0:	x20001854 0x20001b1c	
Version: Version: 202	4.07-devel-135-g46924-feat/slipconf	nrf802154	896 316	580 0:	x200012e4 0x2000155c	
Board: nrf52840dk						
Iface 7		<pre>_ Req(Get /shell/re</pre>	eboot)[0x000]	3]		
HWaddr: 66:C3:0C:0E:B	4:B1:E3:82	Awaiting response				
Link type: wireless						
inet6 addr: fe80::64c	3:c0e:b4b1:e382					
		User Input-				



That was more than just a Shell!

- Because CoAP is not build for human consumption but for machines, we can do cool things now:
 - Auto detect and auto complete available shell commands
 - Automagically fetch and display useful debugging informations such as version and IP addresses of the constrained device
 - Digest binary payloads: present them human readable, save to disk, ...
 - Chain multiple shell commands, passing their I/O into each other

Conclusion & Outlook

- It is feasible to use CoAP as a basis for a modern shell
- Greatly improved usability
- But there is so much more todo!

 - Interoperability between "new" and "old" shell
 - Security aspects: Opening the shell to the network
 - ROM increase due to CoAP vs. decrease due to reduced string parsing



Structured parameter and return values of commands, e.g. CBOR

Bonus Slide: Early Overhead Estimations Comparing the old `ps` vs. the new CoAP `ps`

	Jelly 😽 : Friendly SLIPMUX for RIOT OS	
[Diagnostic Messages-	Configuration Messages	_
ifconfig	Configure network interfaces //	
nib	Configure neighbor information base //riot/board>	
pm	interact with layered PM subsystem //echo/>	
ps	Prints information about running threads. //well-known/ifconfig>	
reboot	Reboot the node .well-known/core	
saul	interact with sensors and actuators using SAUL //config/ps>	
txtsnd	Sends a custom string as is over the link layer // /////////////////////////////////	
version	Prints current RIOT_VERSION	
> ps		
name	stack (used) (free) base addr current //	
isr_stack	512 (224) (288) 0x2000000 0x20000154	
main	1536 (668) (868) 0x2000035c 0x200006f4	
pktdump	1472 (176) (1296) 0x20002f44 0x20003454	
610	960 (216) (744) 0x2000354c 0x2000384c //shell/reboot>	
ipv6	960 (440) (520) 0x20000a9c 0x20000cec	
udp	448 (196) (252) 0x20003950 0x20003a4c Command: ps	
Slipmux CoAP server	1024 (548) (476) 0x20003f24 0x200041f4 name stack used free start SP	
slipdev	896 (236) (660) 0x20001854 0x20001b14 isr_stack 512 224 288 0x2000000 0x20000154	
nrf802154	896 (316) (580) 0x200012e4 0x200015a4 main 1536 668 868 0x2000035c 0x200006f4	
>	pktdump 1472 176 1296 0x20002f44 0x20003488	
	960 216 744 0x2000354c 0x20003868	417

Configuration Version: Version: 2024.07-devel-135-g46924-feat/slipconf Board: nrf52840dk Iface 7 HWaddr: 66:C3:0C:0E:B4:B1:E3:82 Link type: wireless inet6 addr: fe80::64c3:c0e:b4b1:e382

Command: ps-						
name	stack	used	free	start	SP	
isr_stack	512	224	288	0x20000000	0x20000154	
main	1536	668	868	0x2000035c	0x200006f4	
pktdump	1472	176	1296	0x20002f44	0x20003488	
610	960	216	744	0x2000354c	0x20003868	
ipv6	960	440	520	0x20000a9c	0x20000cd8	
udp	448	196	252	0x20003950	0x20003a80	
Slipmux CoAP server	1024	548	476	0x20003f24	0x20004134	
slipdev	896	236	660	0x20001854	0x20001b1c	
nrf802154	896	316	580	0x200012e4	0x2000155c	

User Input-



Bonus Slide: Early Overhead Estimations Comparing the old `ps` vs. the new CoAP `ps`

		Jelly 🈽 : 1	Friendly SLIPMUX for RI	OT OS-				
Diagnostic Messages ifconfig nib pm ps reboot saul txtsnd version > ps name isr_stack main pktdump 6lo ipv6 udp	Old `ps` interact with sensors and actuat Sends a custom string as is over Prints current RIOT VERSION stack (used) (free) base a 512 (224) (288) 0x2000 1536 (668) (868) 0x2000 1472 (176) (1296) 0x2000 960 (216) (744) 0x2000 960 (440) (520) 0x2000	ise m threads. fors using SAUL the link layer	<pre>Friendly SLIPMUX for RI Configuration Message</pre>	s nfig> >	v `ps`			
Slipmux CoAP server slipdev nrf802154 > Configuration Version: Version: 20 Board: nrf52840dk Iface 7 HWaddr: 66:C3:0C:0E: Link type: wireless	1024 (548) (476) 0x2000 896 (236) (660) 0x2000 896 (316) (580) 0x2000 024.07-devel-135-g46924-feat/slipe	03f24 0x200041f4 01854 0x20001b14 012e4 0x200015a4	name isr_stack main pktdump 6lo ipv6 udp Slipmux CoAP server slipdev nrf802154	512 2 1536 6 1472 1 960 2 960 4 1024 2 896 2		0x2000000000 0x2000035c 0x20002f44 0x2000354c 0x2000354c 0x20003950 0x20003950 0x20003f24 0x20001854 0	x20000154 x200006f4 x20003488 x20003868 x20003868 x20000cd8 x20003a80 x20003a80 x20004134 x20001b1c	

inet6 addr: fe80::64c3:c0e:b4b1:e382

User Input-



Bonus Slide: Early Overhead Estimations Comparing the old `ps` vs. the new CoAP `ps`

Jelly 😽 : Friendly SLIPMUX for RIOT OS			
<pre>pm ps reboot saul txtsnd version > ps name isr_stack main pktdump 6lo ipv6 udp Slipmux CoAP server slipdev</pre>	Transmits: ~700 bytes ROM: ~150 byte code ROM: ~140 byte strings Interact with sensors and actuators using SAUL Sends a custom string as is over the link layer Prints current RIOT VERSION stack (used) (free) base addr current 512 (224) (288) 0x20000000 0x20000000 0x20000154 1536 (668) (868) 0x2000035c 960 (216) (744) 0x2000354c 960 (216) (744) 0x2000354c 960 (440) (520) 0x20000354c 960 (216) (744) 0x2000354c 960 (236) (660) 0x2000354c <th><pre>onfiguration Messages</pre></th> <th>SP 0x20000154</th>	<pre>onfiguration Messages</pre>	SP 0x20000154
<pre>> Configuration Version: Version: 2024.07-devel-135-g46924-feat/slipconf Board: nrf52840dk Iface 7</pre>		pktdump147217612960x20002f4461o9602167440x2000354cipv69604405200x20000a9cudp4481962520x20003950Slipmux CoAP server10245484760x20003f24slipdev8962366600x20001854	0x200006f4 0x20003488 0x20003868 0x20000cd8 0x20003a80 0x20004134 0x20001b1c 0x2000155c

Link type: wireless inet6 addr: fe80::64c3:c0e:b4b1:e382

User Input-



Thank you for having me!