

ESPRESSIF
DevCon23

Talking to Cellular Modems

with microcontrollers

David Čermák

Speaker **Intro**

- Embedded software developer
- ESP-IDF project contributor
- Networking and Protocols team manager at Espressif



 **David Čermák**

Talking to Cellular Modems

Analogue modem
with acoustic coupler
and a telephone

Source: Wikipedia

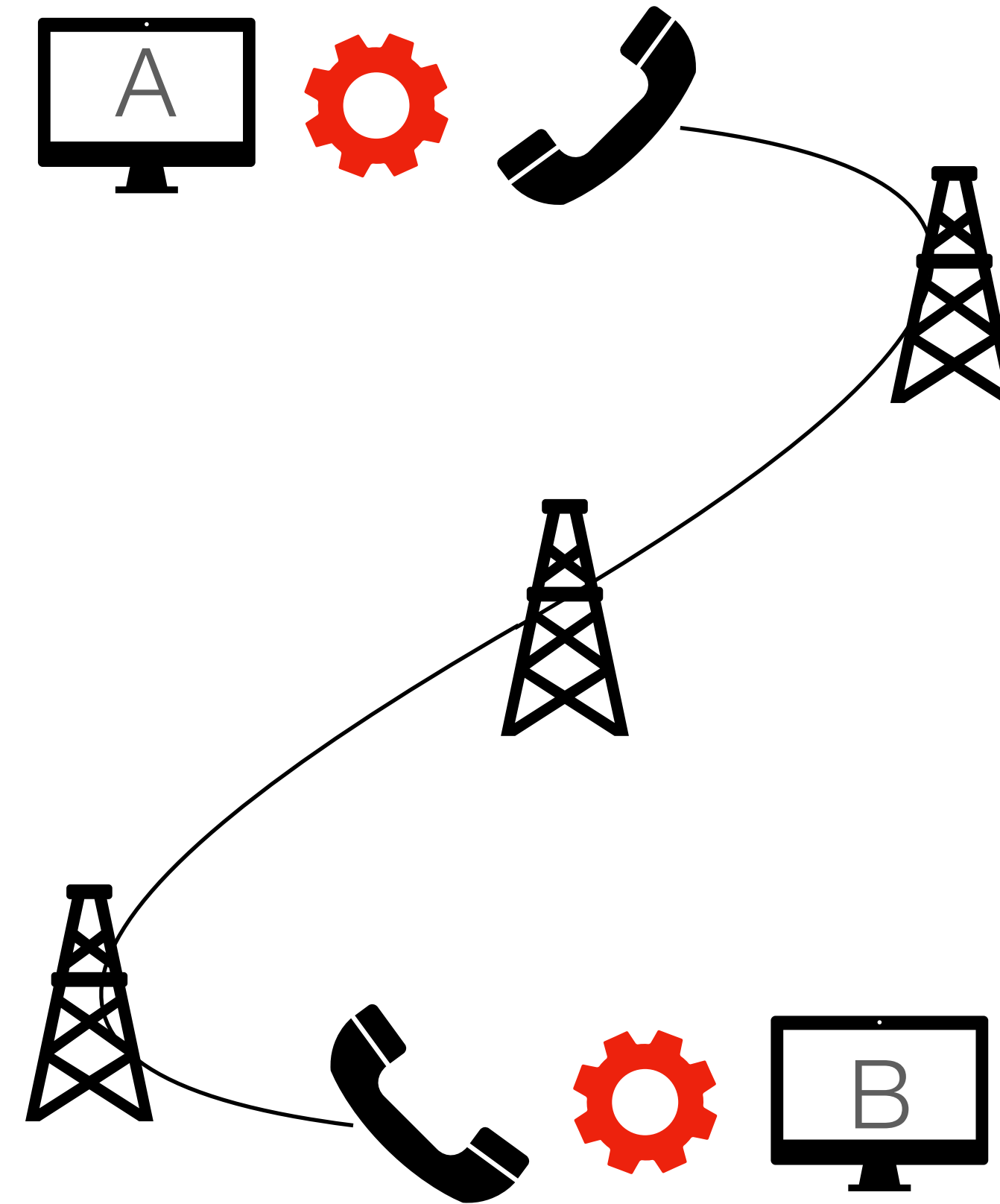
licensed under the [Creative Commons Attribution-Share Alike 2.0 Generic](https://creativecommons.org/licenses/by-sa/2.0/) license, Link: https://en.wikipedia.org/wiki/File:Analogue_modem_-_acoustic_coupler.jpg



Talking to Cellular **Modems**

Modulate

Demodulate

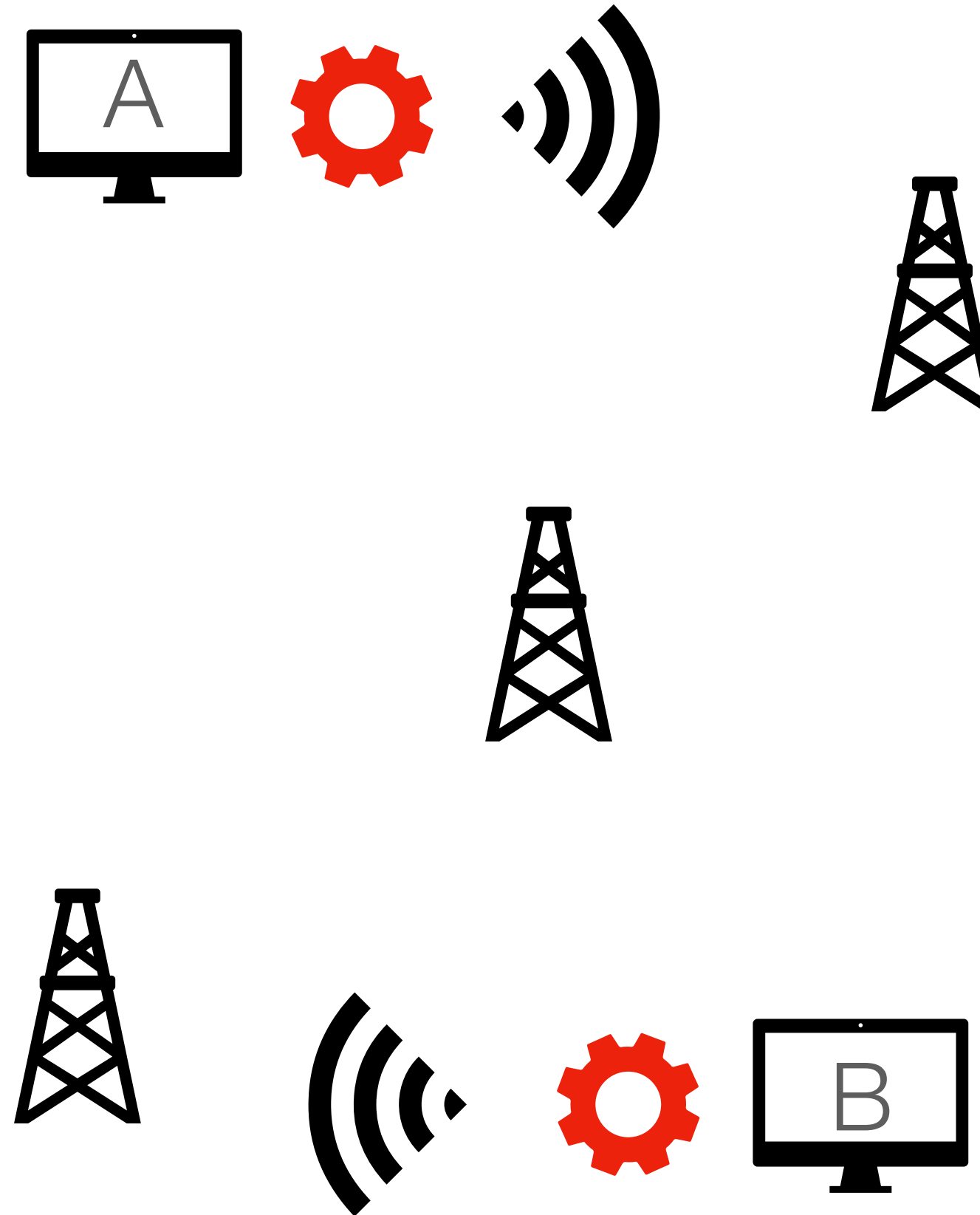


Talking to Cellular Modems

GSM, 2G, 3G

LTE

NB-IoT



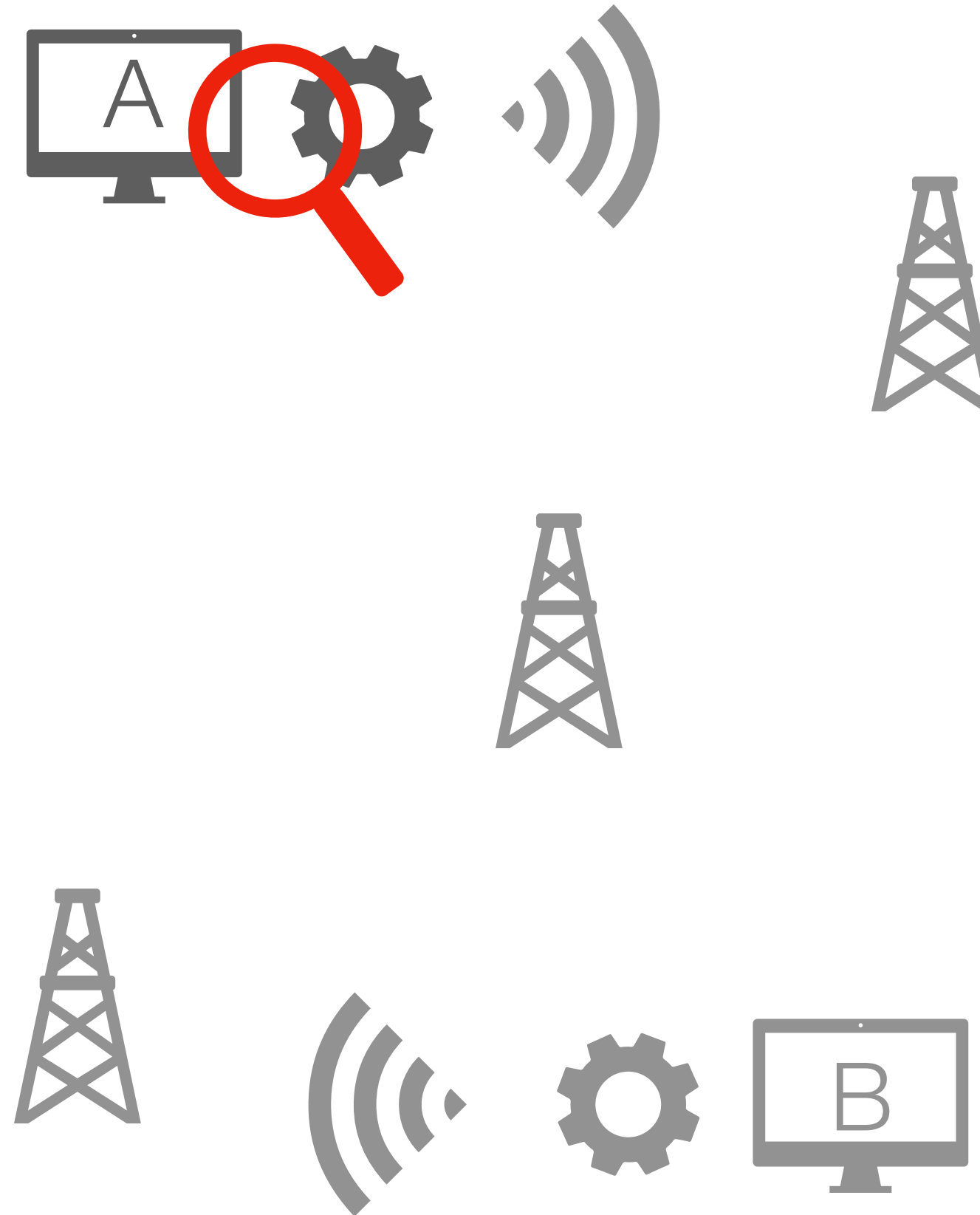
Talking to Cellular Modems

Interface between

- Microcontroller
- Modem

Over UART

With AT commands





CONTENTS

- **Protocols, interfaces**
- **Commands, network**
- **Tips and Tricks**
- **OTA**
- **ESP-IDF solutions**

01



Protocols, interfaces



V.25TER, PPP, MBIM,
GSM 07.10, USB, UART

Most popular modems

Features

Interfaces	Features
UART	Secure cloud services
USB (for diagnostics)	u-blox SARA-R5 Series LTE-M/NB-IoT
DDC (I2C)	Antenna dynamic tuning
USIM	CellTime
GPIO	Ultra low PSM
	TCP/UDP
	HTTP, FTP
	TLS/DTLS
	FW update via serial (FOAT)
	uFOTA
	LwM2M, dynamically loaded objects
	MQTT, MQTT-SN
	CoAP

Most popular modems

Features

Interfaces	Features	Other Features
UART	Secure cloud	Size 26,5x22,5mm <input type="checkbox"/>
USB (for diagnostics)	u-blox SARA-450	Height 2,3 mm <input type="checkbox"/>
DDC (I2C)	Antenna dynamic tuning	Package LGA <input type="checkbox"/>
USIM	CellTime	Interface 3xUART <input type="checkbox"/>
GPIO	Ultra low PSM	Protocol PPP, TCP, UDP <input type="checkbox"/>
	TCP/UDP	Size 24x24mm <input type="checkbox"/>
	HTTP, FTP	Height 3 mm <input type="checkbox"/>
	TLS/DTLS	Package LCC <input type="checkbox"/>
	FW update via serial (FOAT)	Interface UART <input type="checkbox"/>
	uFOTA	Protocol FTP, HTTP, MMS, PPP, SSL, TCP, UDP <input type="checkbox"/>
	LwM2M, dynamically loaded objects	Other Features - <input type="checkbox"/>
	MQTT, MQTT-SN	GNSS - <input type="checkbox"/>
	CoAP	Série SIM800 <input type="checkbox"/>

Most popular modems

Features

Interfaces	Features	Other Features
UART	Secure cloud	
USB (for diagnostics)	u-blox SARA-	
DDC (I2C)	Antenna dynamic tuning	
USIM	CellTime	
GPIO	Ultra low PSM	
	TCP/UDP	
	HTTP, FTP	
	TLS/DTLS	
	FW update via serial (FOAT)	
	uFOTA	
	LwM2M, dynamically loaded ob	
	MQTT, MQTT-SN	
	CoAP	

Size	26,5x22,5mm	<input type="checkbox"/>
Height	2,3 mm	<input type="checkbox"/>
Package	LGA	<input type="checkbox"/>
Interface	3xUART	<input type="checkbox"/>
Protocol	PPP, TCP, UDP	<input type="checkbox"/>

Size	Height	Package	Interface	Protocol
24x24mm	3 mm	LCC	UART	TP, MMS, PPP, SSL, TCP, UDP
			-	-
			-	-
			SIM800	

Product Features

- LTE FDD Cat.4, 3GPP release 9 compliant
- Rx Diversity and MIMO DL 2x2
- SIM application Tool Kit 3GPP TS 51.014
- Serial port multiplexer 3GPP TS27.010
- SMS over IMS
- Built in UDP/TCP/FTP/SMTP stack
- Control via AT commands according to 3GPP TS 27.005, 27.007 and Telit Custom AT commands

Interfaces

- 144-pin LGA Interface
- 10 I/O ports (@1.8V) including multifunctional I/Os
- USB 2.0 HS
- UART
- 1.8 V / 3 V SIM interface
- RF pad, RX Div. & MIMO pad

Talking to modems

Interfaces

- UART
- USB
- I2C, SPI
- ePCI

Protocols

- AT commands (V.25TER and others)
- PPP, TCP, UDP, MQTT, HTTPS, ...
- CMUX (GSM 07.10)
- MBIM

Talking to modems

Interfaces

- UART
- USB
- I2C, SPI
- ePCI

Protocols

- AT commands (V.25TER and others)
- PPP, TCP, UDP, MQTT, HTTPS, ...
- CMUX (GSM 07.10)
- MBIM

02

Commands, network

AT+CPIP

PPP: LCP, IPCP, IPv6CP

Talking to modems

Commands

- Request/Response
- Get/Set properties
- Send text message
- Get NMEA

Network

- Maintain active connection

Talking to modems

Commands

- Request/Response
- Get/Set properties
- Send text message
- Get NMEA

Network

- Maintain active connection

```
/bin/bash
/bin/bash 96x25
Enables PSM on the modem
disable_psm
Disables PSM on the modem
get_psm_cfg
Get PSM config
set_psm_cfg <Requested_Periodic-TAU> [<Requested_Active-Time>]
Set PSM config
<Requested_Periodic-TAU> T3412 Timer in 8-bit format
<Requested_Active-Time> T3324 Timer in 8-bit format
esp> sync
I (514313) modem_console: Sending the initial AT to sync the module...
D (514313) command_lib: generic_command command AT
D (514313) >>: 0x3ffbf4ac 41 54 0d |AT.|
D (514323) <<: 0x3ffbdb1c 0d 0a 4f 4b 0d 0a |..OK..|
D (514333) command_lib: Response:
OK
I (514343) modem_console: OK
esp> get
```


Talking to modems

Commands

- Request/Response
- Get/Set properties
- Send text message
- Get NMEA

Network

- AT commands
- PPP: Collection of Point to Point Protocols

```
exit
  exit the console application

set_deep_sleep <tout>
  Put esp32 to deep sleep
  <tout> TIMEOUT

wakeup_modem
  Wakes up the modem from PSM

enable_psm
  Enables PSM on the modem

disable_psm
  Disables PSM on the modem

get_psm_cfg
  Get PSM config

set_psm_cfg <Requested_Periodic-TAU> [<Requested_Active-Time>]
  Set PSM config
  <Requested_Periodic-TAU> T3412 Timer in 8-bit format
  <Requested_Active-Time> T3324 Timer in 8-bit format

esp> cmd AT+NETOPEN
```

Talking to modems

Commands

- Request/Response
- Get/Set properties
- Send text message
- Get NMEA

Network

- AT commands
- PPP: Collection of Point to Point Protocols

```
exit the console application
set_deep_sleep <tout>
  Put esp32 to deep sleep
  <tout> TIMEOUT
wakeup_modem
  Wakes up the modem from PSM
enable_psm
  Enables PSM on the modem
disable_psm
  Disables PSM on the modem
get_psm_cfg
  Get PSM config
set_psm_cfg <Requested_Periodic-TAU> [<Requested_Active-Time>]
  Set PSM config
  <Requested_Periodic-TAU> T3412 Timer in 8-bit format
  <Requested_Active-Time> T3324 Timer in 8-bit format
esp> set_mode PP
```

03



Tips & Tricks



Simultaneous network and
command interface



Commands and Network

One channel for both network data and command interface (request-response)

01

Switching between PPP and AT mode

02

CMUX: Using channel multiplexer

03

Serve network using AT commands

04

Use two physical channels (in one USB)

Switching between modes

Pluses

- No need for a library
- Supported by most modems
- Most reliable

Minuses

- Switching sequence might differ
- Slow
- Not useful for simultaneous requests

```
exit the console application
set_deep_sleep <tout>
  Put esp32 to deep sleep
  <tout> TIMEOUT

wakeup_modem
  Wakes up the modem from PSM

enable_psm
  Enables PSM on the modem

disable_psm
  Disables PSM on the modem

get_psm_cfg
  Get PSM config

set_psm_cfg <Requested_Periodic-TAU> [<Requested_Active-Time>]
  Set PSM config
  <Requested_Periodic-TAU> T3412 Timer in 8-bit format
  <Requested_Active-Time> T3324 Timer in 8-bit format

esp> cmd
```

Switching between modes

Pluses

- No need for a library
- Supported by most modems
- Most reliable

Minuses

- Switching sequence might differ
- Slow
- Not useful for simultaneous requests

**Network applications that need send commands.
Sleepy applications that wake up, establish connection and go to
sleep again.**

CMUX mode

Pluses

- The most traditional approach
- Simultaneous access to both

Minuses

- Protocol might differ per device
- Need a library

```
set_deep_sleep <tout>
  Put esp32 to deep sleep
  <tout> TIMEOUT

wakeup_modem
  Wakes up the modem from PSM

enable_psm
  Enables PSM on the modem

disable_psm
  Disables PSM on the modem

get_psm_cfg
  Get PSM config

set_psm_cfg <Requested_Periodic-TAU> [<Requested_Active-Time>]
  Set PSM config
  <Requested_Periodic-TAU> T3412 Timer in 8-bit format
  <Requested_Active-Time> T3324 Timer in 8-bit format

esp> set_mode
```

CMUX mode

Pluses

- The most traditional approach
- Simultaneous access to both
- No switching modes

Minuses

- Protocol might differ per device
- Need a library

**Applications that need both commands and network.
Choose the device that supports CMUX.**

Networking with AT

Pluses

- No switching modes
- Available in most devices

Minuses

- Network commands are different
- Security?
- Protocol libraries?

Applications that need commands with occasional networking.

Two physical channels

Pluses

- Best choice if the device supports it

Minuses

- Portability

Devices with USB interface and two channels.

04



Tips & Tricks

Connectivity, OTA

Low power modes

Connectivity & OTA

Flow control

- Use if possible
- Not all devices support it

LCP echo

- Use if possible
- Link protocol to check connectivity

Connectivity & OTA

Two phase OTA

- Check image and pause network
- Bulk erase
- Resume network and OTA

Adjust library lower layers

- Handlers in IRAM
- Buffers
- Baudrate

05

Espressif solutions

ESP-IDF, lwIP

esp_modem

Using IDF only

Using only lwIP PPP

- Simple example
- ~100 lines of code
- [esp-protocols/examples/esp_netif/multiple_netifs/main/ppp_connect_simple.c](https://github.com/espressif/esp-protocols/tree/master/examples/esp_netif/multiple_netifs/main/ppp_connect_simple.c)
(From multiple netif example, which uses Ethernet, WiFi and LTE: [link](#))

Using esp-modem

Features, examples

- CMUX support: [link](#)
- Power save mode example: [link](#)
- Known issues: [link](#)
- TCP client with AT commands: [link](#)
- Linux port: [link](#)
- Modem console: [link](#)

ESP  Registry

Sign in

espressif/esp_modem 1.0.1 v

uploaded 3 weeks ago

esp modem

README

ESP MODEM

latest version 1.0.1

The `esp-modem` component is a managed component for `esp-idf` that is used for communication with GSM/LTE modems that support AT commands and PPP protocol as a network interface.

Links

[Homepage](#)

Supports all targets

License: [Apache-2.0](#)

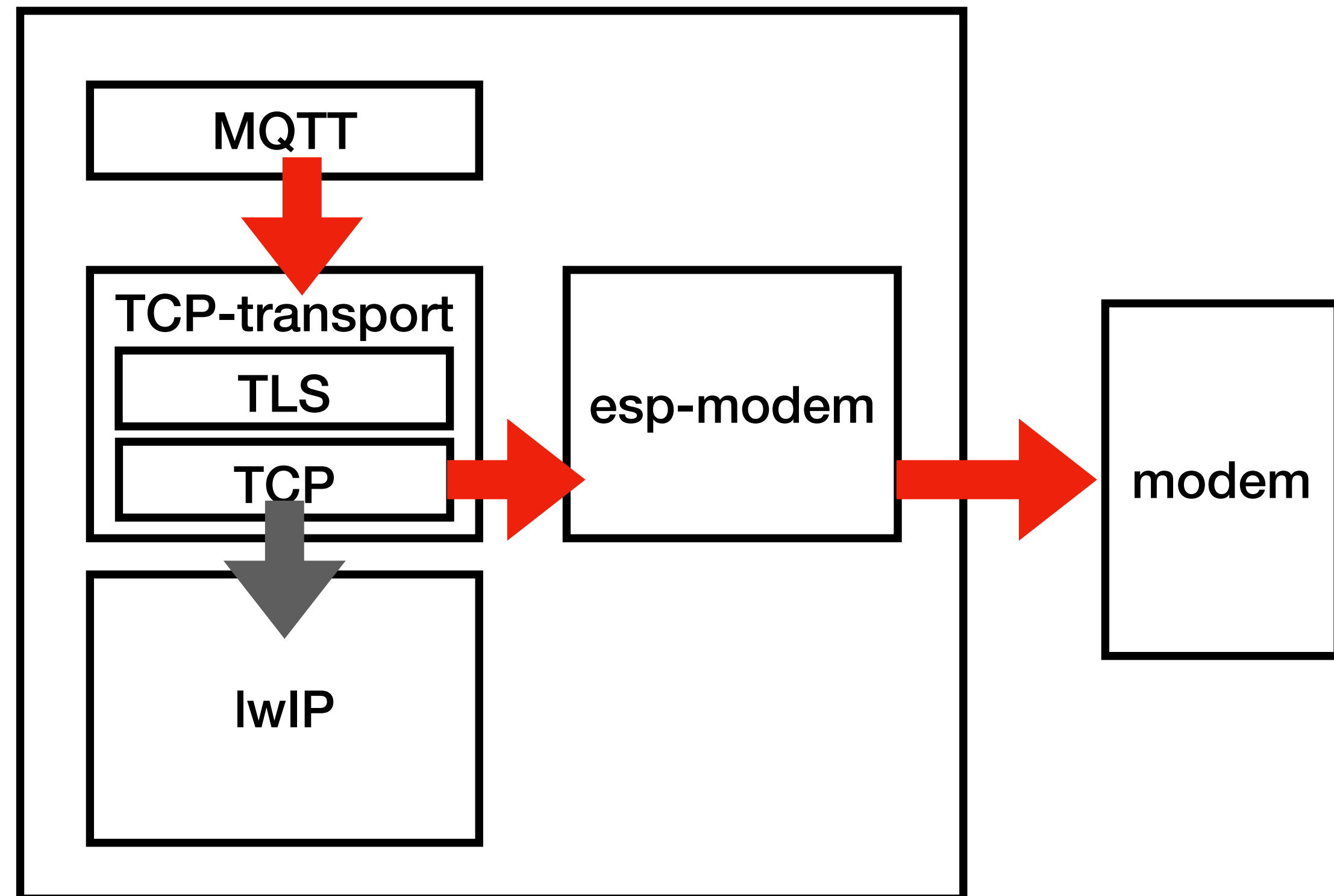
To add this component to your project, run:

```
idf.py add-dependency  
"espressif/esp_modem^1.0.1"
```

or [download archive](#)

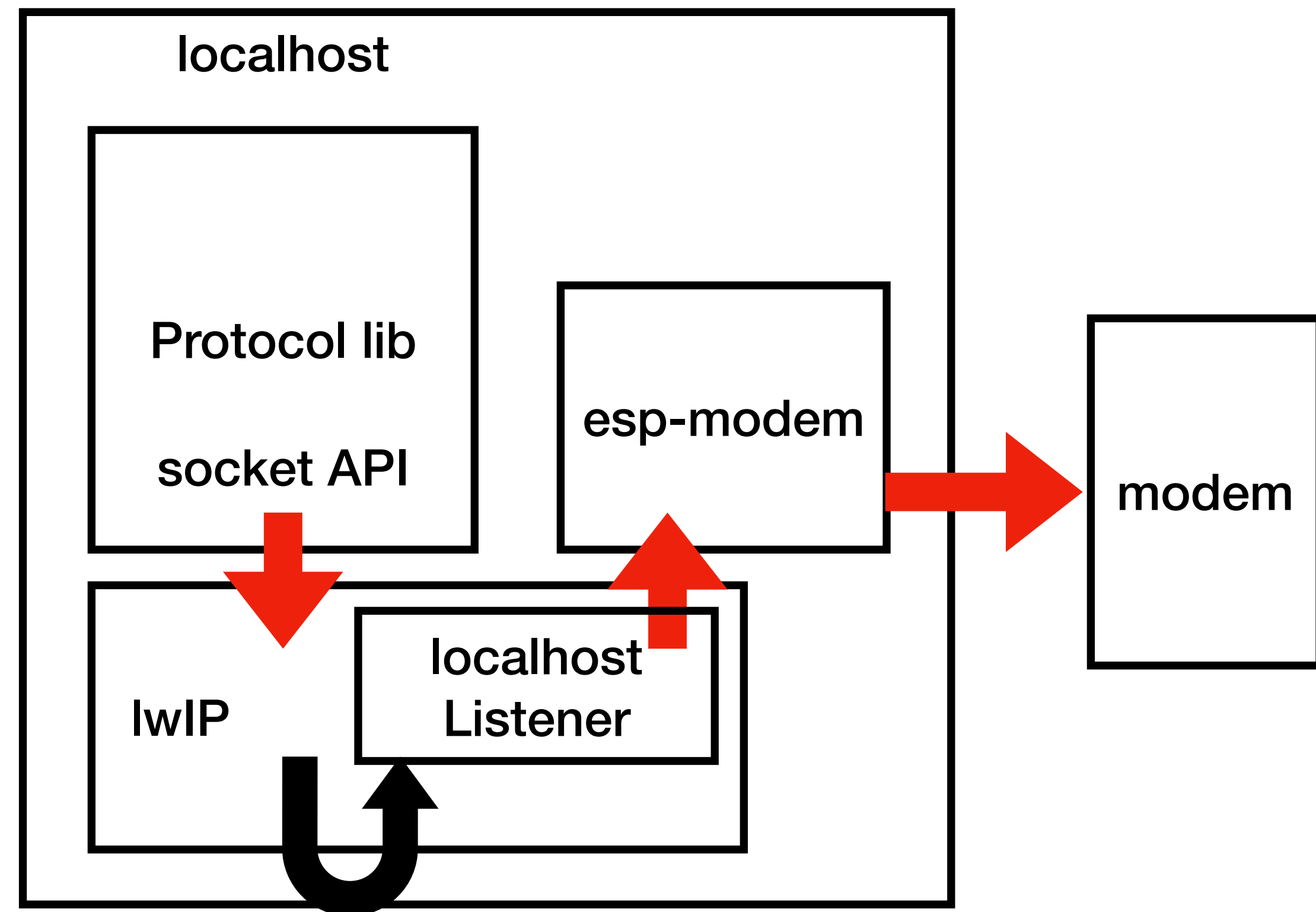
TCP client with AT

- IDF library with TCP transport
- `EXAMPLE_CUSTOM_TCP_TRANSPORT=y`



TCP client with AT

- Any library using
 - Socket API
 - lwIP raw API



Summary

Need commands and network?

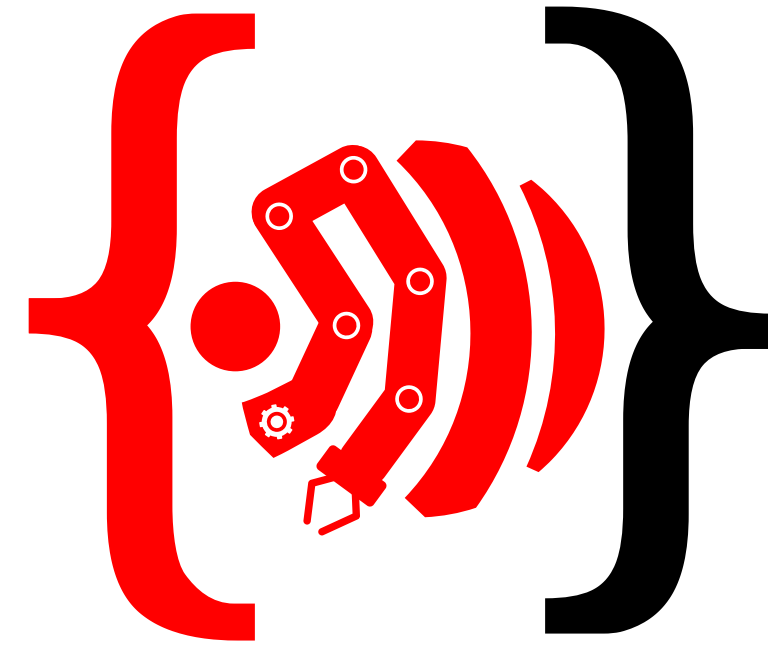
- Use CMUX

Seen buffer overflow?

- Use flow control

Want to check something?

- Experiment with it in modem-console



ESPRESSIF
DevCon23

Thanks for watching !