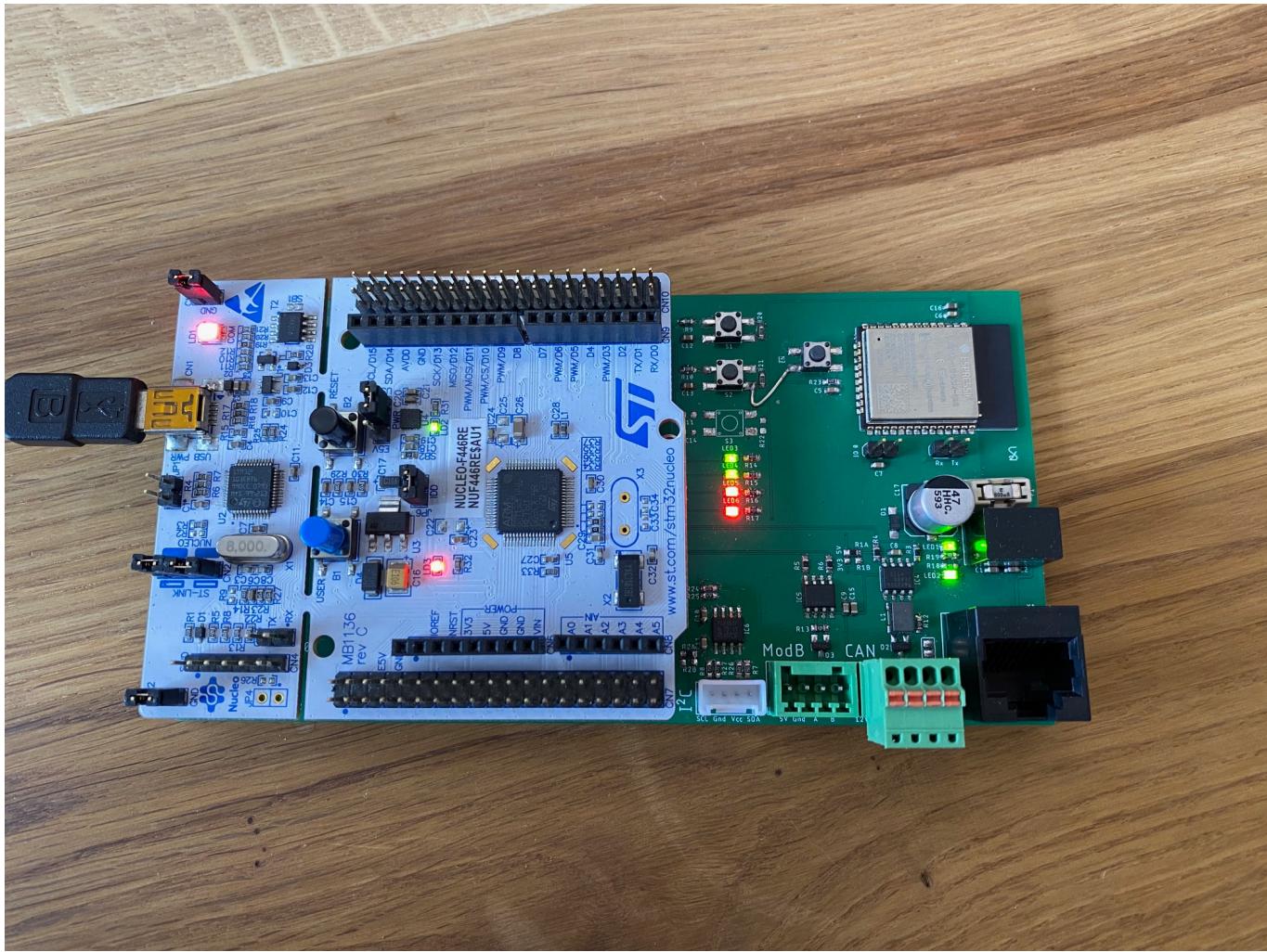


# ESP32 as Wifi Coprocessor with AT-Command-Firmware

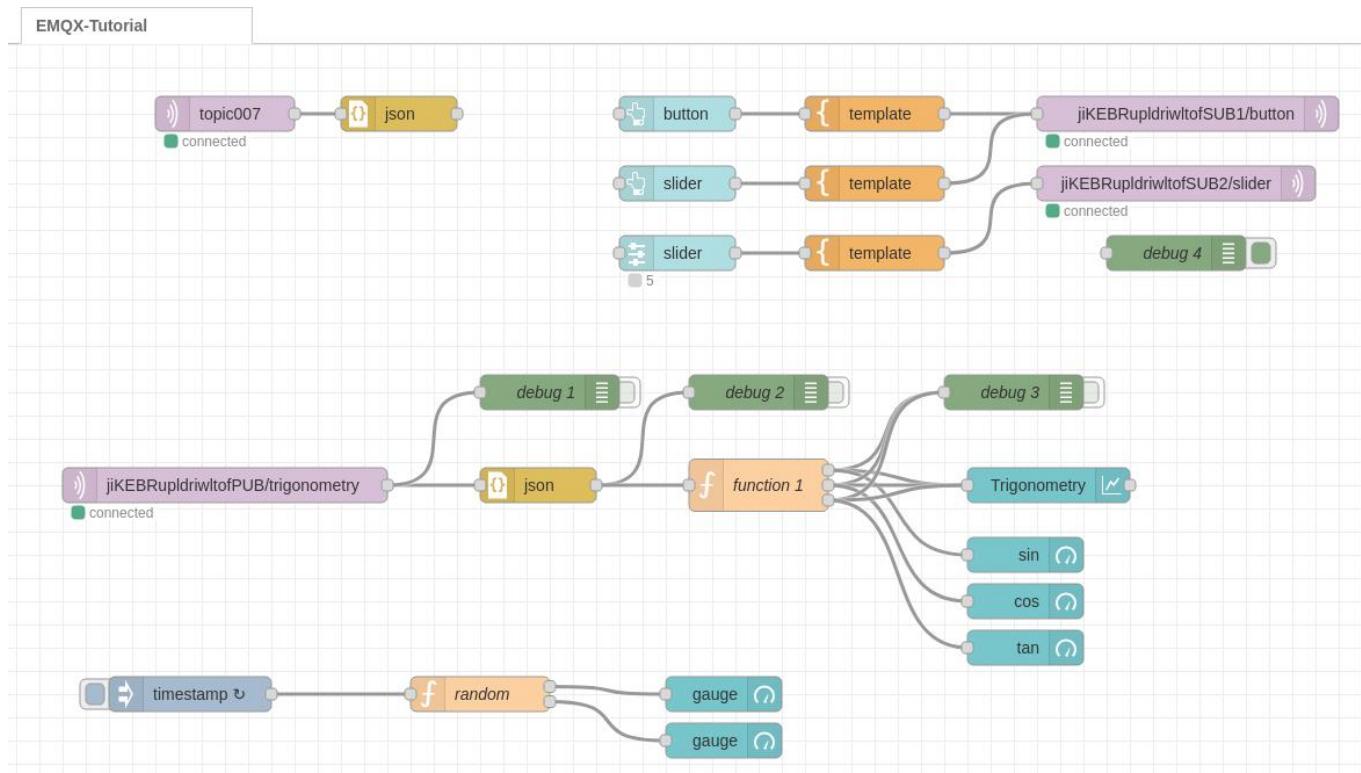
The ESP32 is a very popular microcontroller with integrated Wifi and Bluetooth hardware. The ESP32 is also available as ready to use and pre CE certificated modules with very attractive budget. This approach makes it so successfully, that it has become a serious competitor to the well known Arduino platform.

Many people use a random ESP32 module (e.g. [D1 Mini](#)) and the Arduino IDE or PlatformIO IDE to create wireless projects. But there is another way to bring your application online. The so called **ESP32-AT-Command firmware** is a standalone piece of software running on the ESP32 tuning the ESP32 into a wireless-coprocessor. With only two uart pins it is possible to bring any microcontroller to network. The actual application code runs on your preferred MCU. The key issue is that, the AT-Command handler and the application code **must** be concurrent (deutsch: nebenlaeufig). The application code should not be halted at any time because the AT-Command-handler is waiting for some sort of response due to weak internet connection.

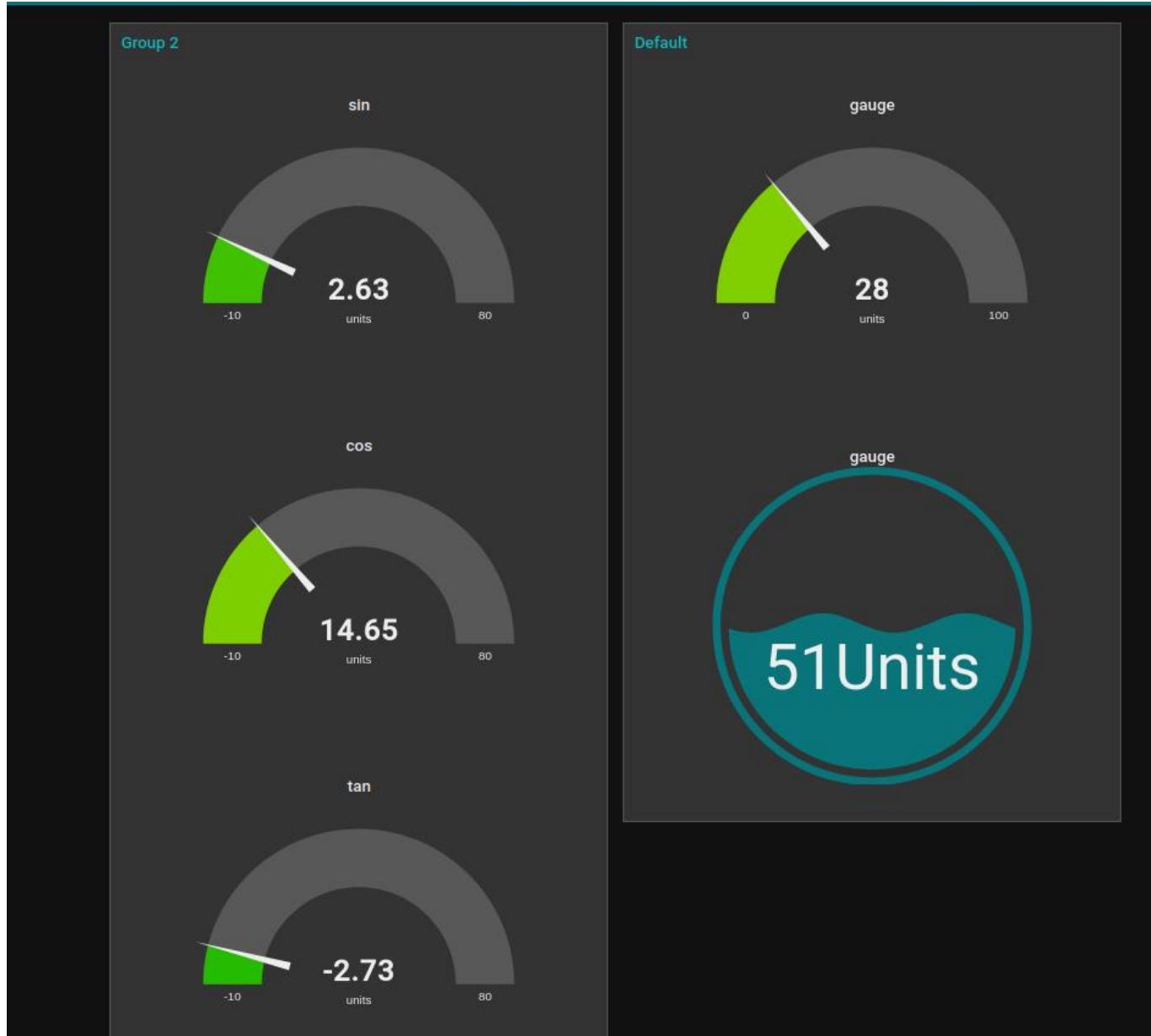
The AT-commands are also used in most of the cellular modules. This little example could be the blueprint for other IoT project. The hardware I used for this Demo is a STM32 Nucleo-F446RE Dev Board attached to a carrier board with a ESP32 soldered on it.



I also used a basic node-red flow to have some 'backend' buttons and diagrams. You could see it as some sort of server or cloud service and the board simulates a IoT-Node.







The Nucleo runs seven noRTOS tasks. One takes care of the connectivity state machine and one sends periodic telemetry data to the server. The rest is demo stuff like flashing a snake LED string or print **hello world** to the terminal.

The screenshot shows a terminal window titled "CuteCom - Default". The window has a menu bar with "Sessions" and "Help", and a toolbar with "Close", "Device: /dev/ttyACM0", and "Settings". The main pane displays a log of serial port activity. The log includes numerous "Online" messages at various timestamps, a "Hello World" message, a telemetry payload message, an AT command response, and several "OK" responses. At the bottom of the window, there are checkboxes for "Clear", "Hex output", and "Logging to: /home/lukas/Fix-LTE-Issue-09-01-2024-2.log", and a status bar indicating the device is an STMicroelectronics STM32 STLink connected via /dev/ttyACM0 at 115200 baud.

```
[14:23:44:567] Online
[14:23:45:083] Online
[14:23:45:563] Online
[14:23:46:061] Online
[14:23:46:557] Online
[14:23:47:055] Online
[14:23:47:553] Online
[14:23:48:051] Online
[14:23:48:549] Online
[14:23:49:047] Online
[14:23:49:544] Online
[14:23:50:042] Online
[14:23:50:043] -- Hello World --
[14:23:50:047] Telemetry [59 bytes] Payload: {"timestamp":1743510169,"sin":-0.96,"cos":14.95,"tan":0.96}
[14:23:50:540] Online
[14:23:51:039] Publishing Raw
[14:23:52:052] -* [debug] - hit needle
[14:23:52:052] AT+MQTTPUBRAW=0,"jKEBRupldriwltofPUB/trigonometry",59,1,0%
[14:23:52:052]
[14:23:52:052] OK
[14:23:52:052]
[14:23:52:052] >+MQTTPUB:OK
[14:23:52:544] Online
[14:23:53:032] Online
[14:23:53:529] Online
[14:23:54:027] Online
[14:23:54:525] Online
[14:23:55:023] Online
[14:23:55:521] Online
[14:23:56:019] Online

```

Device: STMicroelectronics STM32 STLink @ttyACM0 Connection: 115200 @ 8-N-1

The screenshot shows the CuteCom software interface. The title bar reads "CuteCom - Default". The menu bar includes "Sessions" and "Help". The toolbar has "Close", "Device: /dev/ttyACM0", and "Settings". The main window displays a log of serial port activity. The log contains numerous entries, mostly "Online" messages at various times. It also includes several MQTT messages: "+MQTTSUBRECV:0,"jiKEBRupldriwltofSUB1/button",35,this is a test message from button!" and "+MQTTSUBRECV:0,"jiKEBRupldriwltofSUB1/button",35,this is a test message from slider!". The bottom of the window shows buttons for "Clear", "Hex output", and "Logging to: /home/lukas/Fix-LTE-Issue-09-01-2024-2.log". The status bar at the bottom indicates "Device: STMicroelectronics STM32 STLink @ttyACM0 Connection: 115200 @ 8-N-1".

Here is a state diagram of what the AT-Command-Handler state machine is doing. The orange arrows shows the principal direction of states. But the FSM needs to run *wait for response* for each transition to make sure

the *AT-Command* and *AT-Response* handshakes are performed correct.

