

stm32_linux_tutorials

Some good STM32 & Linux examples with really useful documentations for beginners of Embedded Software developers. In this repository, you can learn these stuff step-by-step by really useful and simple examples:

Programming Languages

- * C
- * C++ (basics)
- * Python
- * Linux Bash Script

Concepts

- * RTOS
- * Linux
- * Networks (TCP/UDP)
- * POSIX

Tools

- * Platformio
- * Docker
- * CMake
- * Makefile
- * git

Hardware Requirements

- * STM32 Nucleo board (you can select every type)
- * Raspberry pi (3 or 4)
- * A computer with at least 16GB RAM and 250GB free storage

Hardware Protocols

- * GPIO
- * UART
- * SPI

- * CAN
- * ETHERNET

Roadmap

To accomplish our goal, we will proceed with these tutorials. In each tutorial there will be one or more examples:

- [tutorial00-git-01] Very short introduction to git
- [tutorial01-stm32-pio-01] Writing to GPIO (Blink a LED with STM32)
- [tutorial02-stm32-pio-02] Read from GPIO (Read a push button)
- [tutorial03-stm32-pio-03] External interrupts
- [tutorial04-stm32-pio-04] Very Short introduction to Linux
- [tutorial05-stm32-pio-05] Very short introduction to docker
- [tutorial06-stm32-pio-06] Transmitting message through UART
 - [tutorial06-stm32-pio-06-01] Receiving message in computer via putty
 - [tutorial06-stm32-pio-06-02] Receiving message in computer via docker & linux command
- [tutorial07-stm32-pio-07] Receiving (in polling mode) & Transmitting message through UART
 - [tutorial07-stm32-pio-07-01] Sending & Receiving message in computer via putty
 - [tutorial07-stm32-pio-07-02] Sending & Receiving message in computer via docker & linux command
 - [tutorial07-stm32-pio-07-03] Sending & Receiving message in computer via docker & Python script
 - [tutorial07-stm32-pio-07-02] Sending & Receiving message in computer via docker & C code
- [tutorial08-stm32-pio-08] Receiving (in interrupt mode) & Transmitting message through UART
 - [tutorial08-stm32-pio-08-01] Sending & Receiving message in computer via putty
 - [tutorial08-stm32-pio-08-02] Sending & Receiving message in computer via docker & linux command
 - [tutorial08-stm32-pio-08-03] Sending & Receiving message in computer via docker & python script
 - [tutorial08-stm32-pio-08-04] Sending & Receiving message in computer via docker & C code
- [tutorial09-linux-01] Receiving & Transmitting message to virtual UART in linux via docker

- [tutorial09-linux-01-01] using linux command
- [tutorial09-linux-01-01] using Python + POSIX multithreading (different threads for TX & RX)
- [tutorial09-linux-01-02] using C code + POSIX multithreading (different threads for TX & RX)
- [tutorial09-linux-01-02] using C++ code + POSIX multithreading (different threads for TX & RX)
- [tutorial10-stm32-pio-09] Working with LED Display based on SPI protocol
 - [tutorial10-stm32-pio-09-01] Very short introduction to SPI (Full-duplex & Half-duplex)
 - [tutorial10-stm32-pio-09-02] Implementing SPI and performing self-testing (connecting MISO & MOSI together)
 - [tutorial10-stm32-pio-09-03] Implementing LED Display driver
- [tutorial11-stm32-makefile-01] Converting [tutorial01-stm32-pio-01] example to a makefile project
- [tutorial12-stm32-cmake-01] Converting [tutorial01-stm32-pio-01] example to a CMake project
- [tutorial13-linux-02] Transmitting & Receiving messages through TCP network
 - [tutorial13-linux-02-01] using python
 - [tutorial13-linux-02-02] using C code
 - [tutorial13-linux-02-01] using C++ code