



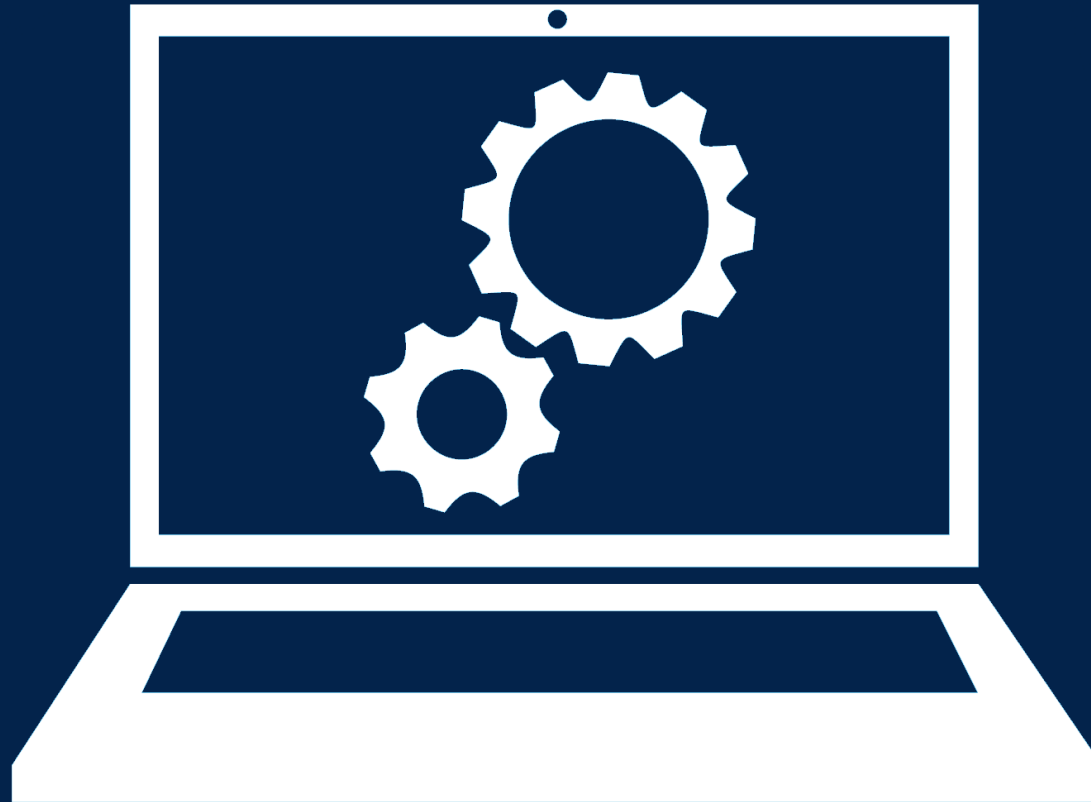
life.augmented

# RT-Thread Demo

Cindy Ge

March 2021

# RT-Thread Demo



RT-Thread (Real Time-Thread), a tiny and elegant IoT operating system, it is an open source, neutral, and community-based RTOS.

## RT-Thread Advantages

- Designed for resource-constrained devices (minimum kernel 1.2KB RAM and 3KB Flash).
- Rich components: VFS, Filesystem (proprietary solution), RS485, FDCAN (see last slide to see supported IPs).
- Easy to use.
- High scalability, easy to tailor and expand.
- Support OpenAMP solution.
- Apache license.

*Now RT-thread is implemented on Cortex-M4 of MP1*

# RT-Thread Development

## RTT source code of STM32MP1:

- <https://github.com/RT-Thread/rt-thread/tree/master/bsp/stm32/stm32mp157a-st-discovery>
- <https://github.com/RT-Thread/rt-thread/tree/master/bsp/stm32/stm32mp157a-st-ev1>

## Development tools:

IAR EW for ARM and ENV tool

## Compiling

The documents about compiling method and how to implement RTT on STM32MP1 can be find from APA MPU Codex wiki.

More RTT information can be learn from RT-Thread official website:

- [https://codex.cro.st.com/plugins/mediawiki/wiki/stm32-mpu-asia-project/index.php?title=RT-Thread\\_is\\_implemented\\_on\\_Cortex-M4\\_of\\_MP1](https://codex.cro.st.com/plugins/mediawiki/wiki/stm32-mpu-asia-project/index.php?title=RT-Thread_is_implemented_on_Cortex-M4_of_MP1)
- <https://www.rt-thread.org/document/site/tutorial/quick-start/introduction/introduction/>

## Supported IPs of STM32MP1:

- NVIC
- WWDG
- RCC
- PWR
- UART/USART
- EXTI
- TIMx/LPTIMx
- I2C/SPI
- ADC/DAC
- GPIO
- IPCC(OpenAMP)
- DMA/DMAMUX
- QSPI(NAND Flash)
- SDMMC(SD Card)
- EMMC
- FDCAN
- SAI
- DCMI
- DFSDM
- SPDIFRx
- CRYIP/CRC/HASH/TRNG
- FMC PSRAM(not yet validated)

One demo *RT-Thread-STM32MP1\_CM4.elf* has been provided, the location is */work/rt-thread/*.

*Deploy to board :*

```
PC: $> scp /work/rt-thread/ RT-Thread-STM32MP1_CM4.elf root@192.168.7.1:/lib/firmware
```

*Check on board :*

```
Board: $> echo RT-Thread-STM32MP1_CM4.elf > /sys/class/remoteproc/remoteproc0/firmware
```

```
Board: $> echo start > /sys/class/remoteproc/remoteproc0/state
```

```
Board: $> stty -onlcr -echo -F /dev/ttyRPMMSG0
```

```
Board: $> cat /dev/ttyRPMMSG0 &
```

```
Board: $> echo "version"> /dev/ttyRPMMSG0
```

```
Board: $> echo "spi_sample 1 2 3 4 5 6 7 8"> /dev/ttyRPMMSG0
```

# Thank you