Documant of how RTT works

Manjiang Cao ⟨*mcao*999@*connect.hkust-gz.edu.cn*⟩

November 19, 2023

1 Overview of RTT principle

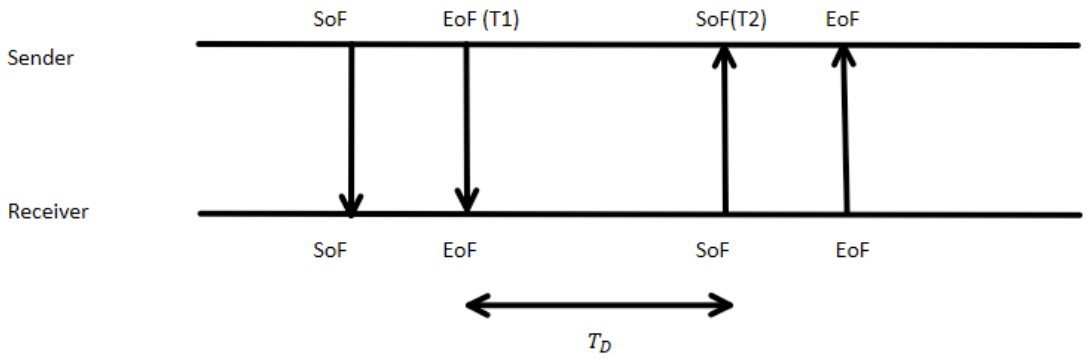

Figure 1: How RTT works

SoF(start of frame): radio start receive or send a packet
EoF(End of frame): radio finish receive or send a packet
Clock frequency:$f = 16Mhz$
Speed of light: $C = 299792458m/s$
distance calculation:

$$\frac{(T2 - T1) - T_D}{2} \times \frac{C}{f}$$

2 Driver implementation

2.1 Timer driver

This part shows how to use high frequency clock on nRF52840 as a timer.

2.1.1 timer_init(void)

Using for initializing the timer.

2.1.2 timer_start(void) and timer_stop(viod)

Using for start or stop the timer.

2.1.3 timer_clear(void)

Using for cleaning timer value, start form 0

2.1.4 timer_capture_now(uint8_t capture_id)

The timer on nRF52840 have 6 registers(CC[0] to cc[5]) to save timevalue, this function can save the current timevalue in a specified(0 to 5) register.

2.1.5 timer_schedule(uint8_t compare_id, uint32_t value)

Save value in a specified register, when timevalue equals to the value will trigger EVENTS_COMPARE.This event can be used to config PPI.In the RTT project, using this function to schedule a exact T_D.

2.1.6 timer_getCapturedValue(uint8_t compare_id)

Using for get captured timevalue from specified register.

2.2 PPI driver

2.2.1 ppi_enable(uint8_t channel) and ppi_disable(uint8_t channel)

Enable or disable specified PPI channel.

2.2.2 ppi_radio_sof_timer_capture(uint8_t channel, uint8_t capture_id)

Using to config a PPI. When radio at start of frame will trigger timer to capture current timevalue. In the RTT project, using for get T_2 at sender.

2.2.3 ppi_radio_eof_timer_capture(uint8_t channel, uint8_t capture_id)

Using to config a PPI. When radio at end of frame will trigger timer to capture current timevalue. In the RTT project, using for get T_1 at sender. At receiver, also used this function to get exact timevalue of end of frame in order to schedule T_D.

2.2.4 ppi_timer_compare_radio_start(uint8_t channel, uint8_t compare_id)

Using to config a PPI. At recevier side, when timer goes to scheduled timevalue will trigger the radio to start send a packet to sender.

3 RTT implementation on nRF52840

If using nRF52840-dk board, check the eui of the board and change the mote_function(). The 01bsp_rtt.c can be divide into two parts: sender_main() and receiver_main(). mote_main() read the eui of board to distinguish which board is sender or receiver and run corresponding function(sender_main() or receiver_main())

3.1 sender_main()

This function is main function of sender.

Firstly, initializing the board and timer, than config PPI using PPI driver and set up UART and radio. After enabling radio, start by a transmit.

The radio have four states. RX at start of frame, RX at end of frame, TX at start of frame, TX at end of frame. For sender, when radio is on TX SoF, do nothing. When radio is on TX EoF, sender need to switch to RX mode and the PPI will capture current timevalue, using $timer\_getCapturedValue()$ to save T1. When radio is on RX SoF, the PPI will capture current timevalue, using $timer\_getCapturedValue()$ to save T2. When radio is on RX EoF, radio already finished receive a packet, according to T1 and T2 compute time interval and transmit time interval and rssi through serial. The sctimer callback function will periodicly start sending a packet to receiver.

3.2 receiver_main()

This function is main function of receiver.

Firstly, initializing the board and timer, than config PPI using PPI driver and set up radio. After enabling radio, set radio state to RX and waiting for packet.

On the receiver side, when radio is on RX SoF do nothing. When radio is on RX EoF, using *timer_schedule*() to schedule T_D than prepare packet and set radio to TX mode, waiting PPI trigger TASK_START event and start to send the packet. When radio is on TX SoF, do nothing. When radio is on TX EoF, set radio to RX mode waiting for next packet.