Guidelines for external RF front-end on BlueNRG-LP/BlueNRG-LPS/STM32WB0 MCUs

Introduction

The BlueNRG-LP, BlueNRG-LPS, and STM32WB0 series devices are ultralow power Bluetooth® Low Energy SoC devices, which can achieve +8 dBm of output power on an antenna connector. Nevertheless, bluetooth standards allow a maximum output power of +20 dBm (local regulations can still limit the output power to a lower value).

The BlueNRG-LP, BlueNRG-LPS, and STM32WB0 series devices are referred as the *devices* in this document.

Especially for this reason, the *devices* give the possibility to control an external RF front-end, that can increase the output power by using the integrated power amplifier.

In addition to a power amplifier (PA), RF front-ends can usually integrate a low noise amplifier (to improve sensitivity), TX/RX switching circuitry, matching network, and harmonic filters.

1 General information

The STM32WB0 series are Arm® Cortex® core-based microcontrollers.

For more information on Bluetooth®, refer to http://www.bluetooth.com.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

References

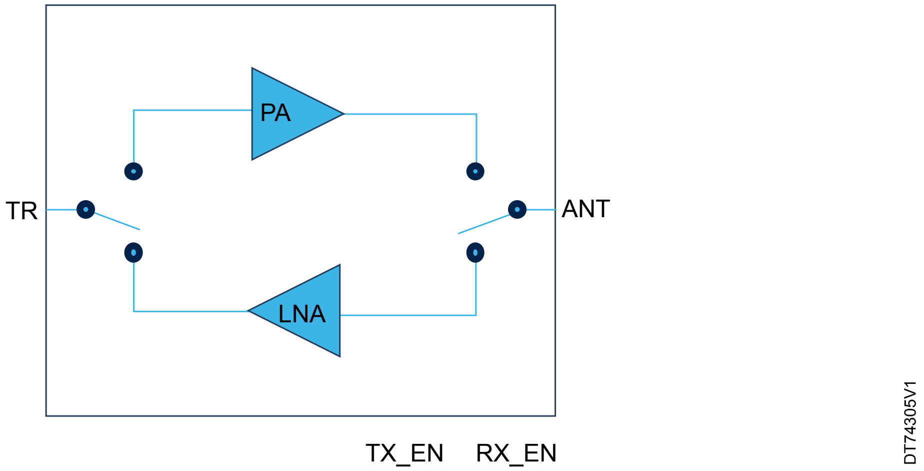
BlueNRG-LP datasheet (DS13282)
BlueNRG-LPS datasheet (DS13819)
BlueNRG-LP reference manual (RM0479)
BlueNRG-LPS reference manual (RM0491)
STM32WB05xZ datasheet (DS14591)
STM32WB05xZ reference manual (RM0529)
STM32WB07xC, STM32WB06xC datasheet (DS14676)
STM32WB07xC, STM32WB06xC reference manual (RM0530)
STM32WB09 datasheet (DS14210)
STM32WB09 reference manual (RM0505)

2 RF control signals

An external front-end usually has at least two pins to control an RF switch, which can connect selectively the antenna either to the output of the PA or to the input of the LNA. The LNA may not be present. In this case, the RF switch can directly connect the TX/RX pin with the antenna.

A simple block diagram example of an external front-end is shown below. There could be other blocks inside the front-end (for example, a harmonic filter on the PA), or a bypass path between the TX/RX port and the antenna.

Figure 1. **Basic block diagram example of a front-end**



To control the external front-end, some signals need to be generated from the SoC. Usually two signals are enough: the former to control the TX path, the latter to control the RX path. The way these signals control the front-end varies from one manufacturer to another.

Moreover, there is always the need to anticipate TX_EN/RX_EN signals before the radio is in transmission or reception state. This is because power amplifiers need time before power is stable and can consume so much current that the *devices* PLL may be destabilized.

Therefore, the radio sequencer generates two signals:

- TX_SEQUENCE, which is raised when the radio sequencer is going to start a transmission, and it is put back to low when internal PA is switched off.
- RX_SEQUENCE, which is raised when the radio sequencer is going to start a reception, and it is put back to low when the radio leaves RX state.

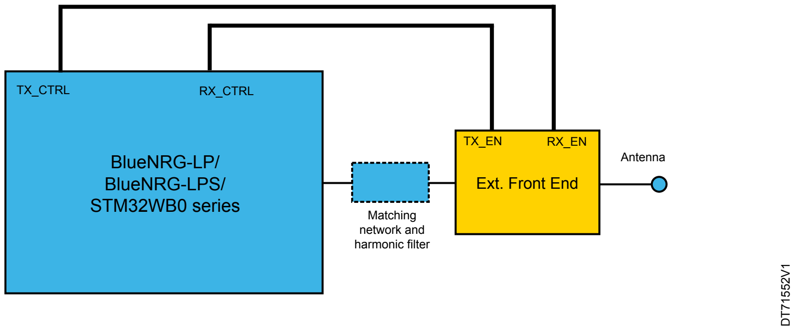
Transition from low to high of TX_SEQUENCE occurs before the first bit is transmitted over-the-air. Also, RX_SEQUENCE transition from low to high occurs before the radio is in reception state. These timings depend on the state (TX or RX) and on PLL calibration whether it is done or not. PLL calibration is done each time a different RF channel is used. Timings are reported in Table 1. These timings are based on hardware requirements and set by the firmware accordingly.

Table 1. **Delay between RF control signals and TX/RX state**

Signal	With PLL calibration	Without PLL calibration
TX_SEQUENCE	118 µs	58 µs
RX_SEQUENCE	116 µs	56 µs

In the Figure 2, a block diagram shows the necessary connections between the BlueNRG-LP/BlueNRG-LPS/STM32WB0 series and the external front-end. TX_CTRL and RX_CTRL are the pins used to control the front-end, and they can be different depending on the used control mode. The matching network and harmonic filter may not be needed if the impedance is already matched and an appropriate harmonic filter is integrated inside the front-end.

Figure 2. Block diagram of connections between the BlueNRG-LP/BlueNRG-LPS/STM32WB0 series and an RF front-end



3 RF control modes

In the BlueNRG-LP/BlueNRG-LPS/STM32WB0 series, there are two options to use TX/RX sequence signals generated by the radio sequencer.

1. Automatic mode, that is, internal radio TX_SEQUENCE and RX_SEQUENCE signals are routed to SoC GPIOs. These signals may not be compatible with the control logic of the front-end
2. Interrupt mode, that is, internal radio TX_SEQUENCE and RX_SEQUENCE signals generate interrupt requests, so that the firmware can act according to the control logic needed by the front-end

3.1 Automatic control mode

In automatic mode, TX_SEQUENCE and RX_SEQUENCE signals can be enabled on some of the *devices* GPIOs.

Table 2. BlueNRG-LP/STM32WB07/STM32WB06 GPIOs connected RF control signals

Signal	GPIO (alternate functions)
TX_SEQUENCE	PA10 (AF2), PB15 (AF1)
RX_SEQUENCE	PA8 (AF2), PA11 (AF2)

Table 3. BlueNRG-LPS/STM32WB05/STM32WB09 GPIOs connected RF control signals

Signal	GPIO (alternate functions)
TX_SEQUENCE	PA10 (AF2), PB14 (AF1)
RX_SEQUENCE	PA8 (AF2), PA11 (AF2)

The advantage of this mode is that TX/RX sequence signals are generated autonomously, without any operation from the firmware. The drawback is in flexibility: the control logic of the front-end must be compatible with the signals generated by the radio sequencer. For instance, a front-end with a control logic as in the table below is directly compatible with TX_SEQUENCE and RX_SEQUENCE signals.

Table 4. Example of a compatible control logic table

Front-end mode	TX_EN	RX_EN
Sleep	0	0
RX	0	1
TX	1	0

3.2 Interrupt control mode

In interrupt mode, the system controller can detect TX_SEQUENCE and RX_SEQUENCE signals coming from the sequencer, which can generate interrupts. To enable TX/RX sequence interrupts, the following code can be used.

```
LL_APB0_EnableClock(LL_APB0_PERIPH_SYSCFG);

LL_SYSCFG_BLERXTX_SetTrigger(LL_SYSCFG_BLERXTX_TRIGGER_BOTH_EDGE, LL_SYSCFG_BLE_TX_EVENT);
LL_SYSCFG_BLERXTX_SetTrigger(LL_SYSCFG_BLERXTX_TRIGGER_BOTH_EDGE, LL_SYSCFG_BLE_RX_EVENT);

LL_SYSCFG_BLERXTX_SetType(LL_SYSCFG_BLERXTX_DET_TYPE_EDGE, LL_SYSCFG_BLE_TX_EVENT);
LL_SYSCFG_BLERXTX_SetType(LL_SYSCFG_BLERXTX_DET_TYPE_EDGE, LL_SYSCFG_BLE_RX_EVENT);
LL_SYSCFG_BLERXTX_EnableIT(LL_SYSCFG_BLE_TX_EVENT|LL_SYSCFG_BLE_RX_EVENT);
NVIC_EnableIRQ(BLE_SEQ_IRQn);
```

The BLE_RXTX_SEQ_IRQHandler can be defined as follows:

```
void BLE_RXTX_SEQ_IRQHandler(void)
{
    if(LL_SYSCFG_BLERXTX_IsInterruptPending(LL_SYSCFG_BLE_TX_EVENT))
    {
        // Set GPIOs to make RF Front End enter TX mode

        LL_SYSCFG_BLERXTX_ClearInterrupt(LL_SYSCFG_BLE_TX_EVENT);
    }

    else if(LL_SYSCFG_BLERXTX_IsInterruptPending(LL_SYSCFG_BLE_RX_EVENT))
    {
        // Set GPIOs to make RF Front End enter RX mode.

        LL_SYSCFG_BLERXTX_ClearInterrupt(LL_SYSCFG_BLE_RX_EVENT);
    }
}
```

Inside the interrupt service routine, any GPIO can be used to drive the RF front-end.

This mode of operation gives a greater flexibility compared to the automatic control mode, since any control logic can be implemented. The drawback of this method is that the control signal may be delayed if a higher priority interrupt is raised in the meantime. It is suggested to use a priority lower than BLE_TX_RX_IRQHandler but higher than other interrupts. Even if the highest priority is assigned to the BLE_TX_RX_IRQHandler, this does not interfere with the BLE_RXTX_SEQ_IRQHandler. In fact, BLE_TX_RX_IRQHandler is executed only at the end of a TX/RX sequence. Hence, the execution of BLE_TX_RX_IRQHandle delay BLE_RXTX_SEQ_IRQHandler only at the end of a TX/RX sequence when the external front-end should be driven to exit RX or TX mode, which is not a critical operation.

Note: *In the context of STM32CubeWB0 the equivalent IRQ handler to be used is:*

`RADIO_TXRX_SEQ_IRQHandler()`

As consequence the IRQ handler must be enabled as follow:

`NVIC_EnableIRQ(RADIO_TXRX_SEQ_IRQn);`

Revision history

Table 5. **Document revision history**

Date	Version	Changes
06-Nov-2020	1	Initial release.
06-Apr-2022	2	Updated Section Introduction, Section 3.1: Automatic control mode and References. Added the BlueNRG-LPS reference throughout the document.
20-Jun-2024	3	Added the STM32WB0 series reference throughout the document.

Contents