CC1100/CC2500 - Wake-On-Radio

By Siri Namtvedt

Keywords

- *CC1100*
- *CC2500*
- *WOR*
- *Event 0*
- *Event 1*
- *RX timeout*
- *RSSI Threshold*

1 Introduction

The ***CC1100*** and ***CC2500*** both have Wake on Radio (WOR) functionality, which enables the radio to periodically wake up from SLEEP mode and listen for incoming packets without MCU interaction. After a programmable time in RX, the chip goes back to the SLEEP unless a packet has been received. The purpose of this application note is to explain the theory of operation and the different registers involved when using Wake on Radio, as well as highlighting important aspects when using WOR mode. Figure 1 shows the relationship between the WOR events and the different radio states.

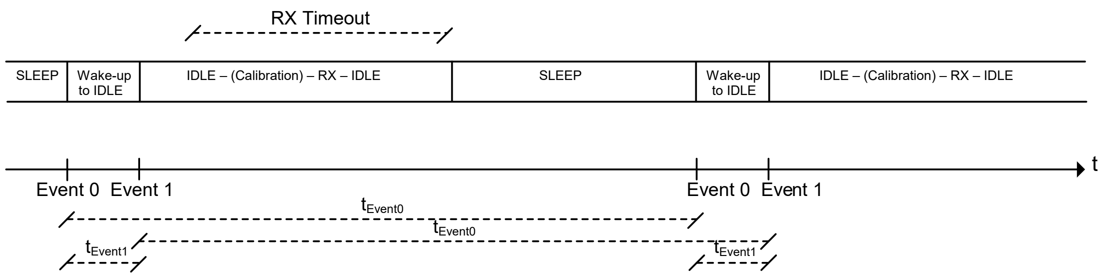

Figure 1. WOR Events and Radio States

Table of Contents

2 Abbreviations

CS	Carrier Sense
MCU	Micro Controller Unit
NA	Not Applicable
PQT	Preamble Quality Threshold
RSSI	Received Signal Strength Indicator
WOR	Wake on Radio
XOSC	Crystal Oscillator

3 Registers

This section covers the theory of operation, the equations governing WOR operation, as well as configuration of the different registers relevant for WOR mode. For more details on the registers described in this sections, please see [1] and [2].

3.1 WOREVT1 and WOREVT0

In SLEEP mode with WOR enabled, reaching Event 0 will turn on the digital regulator and start the crystal oscillator. The time between two consecutive Event 0s is programmed with a mantissa value given by `WOREVT1.EVENT0` and `WOREVT0.EVENT0`, and an exponent value set by `WORCTRL.WOR_RES`. See Equation 1.

$$t_{Event0} = \frac{750}{f_{XOSC}} \cdot EVENT0 \cdot 2^{5 \cdot WOR\_RES}$$

Equation 1. t_{Event0}

Event 0 can be monitored on one of the GDOx pins by setting `IOCFGx.GDOx_CFG` = 0x24. See Figure 2.

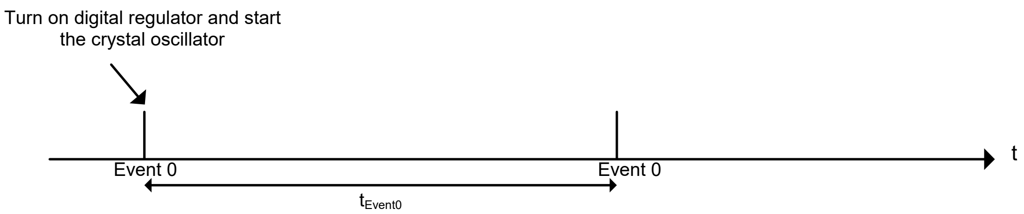

Figure 2. Event 0

Due to a design error related to the WOR timer module, see [3] and [4], the time from the radio enters SLEEP mode until the next Event 0 is programmed to appear (t_{SLEEP}) should not be less than 11.08 ms when using a 26 MHz crystal and 10.67 ms when a 27 MHz crystal is used. If $t_{SLEEP} < t_{SLEEP_{min}}$, there is a chance that the consecutive Event 0 will occur (750·128) / f_{XOSC} s too early.

$t_{SLEEP_{min}}$ can be calculated as showed in Equation 2:

$$t_{SLEEP_{min}} = \frac{750}{f_{xosc}} \cdot 384$$

Equation 2. $t_{SLEEP_{min}}$

The minimum time between two Event 0s ($t_{Event0_{min}}$) depends on t_{Event1} (see 3.2.2), if the PLL is being calibrated or not, and the RX timeout. Example 1 will illustrate these dependencies. By looking at a plot of current consumption vs. time when the radio is configured for WOR mode (see Figure 3), an equation for $t_{Event0_{min}}$ can be found (see Equation 3).

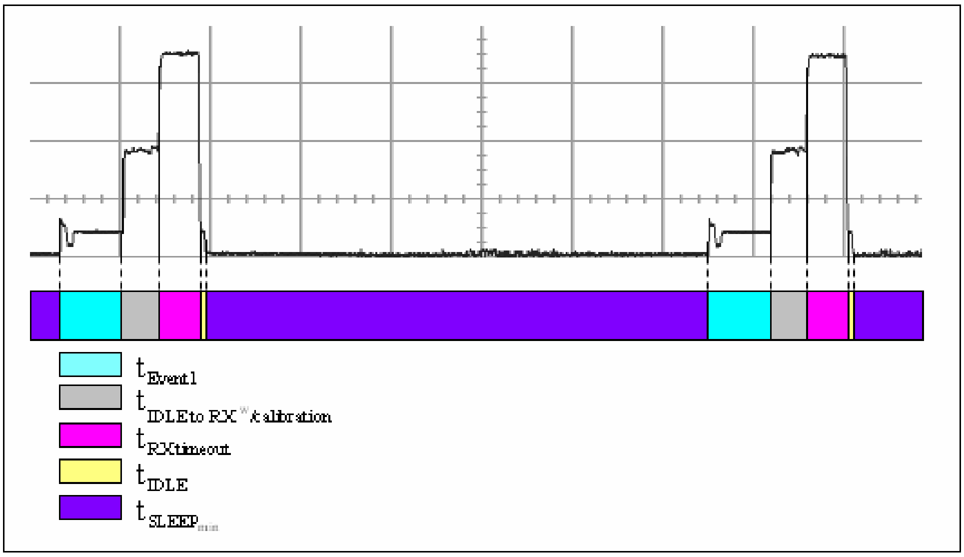

Figure 3. Current Consumption vs. Time

$$t_{Event0_{min}} = t_{Event1} + t_{IDLE\ to\ RX\ ^{w}/Calibration} + t_{RX\ timeout} + t_{IDLE} + t_{SLEEP_{min}}$$

Equation 3. $t_{Event0_{min}}$

$$t_{Event1} = x \cdot \frac{750}{f_{XOSC}}$$

, where x is given by Table 1

Equation 4. t_{Event1}

x	WORCTRL.EVENT1
4	0
6	1
8	2
12	3
16	4
24	5
32	6
48	7

Table 1. *x* Values to use in Equation 4

See section 3.2.2 for considerations that need to be taken into account when programming t_{Event1}.

Example 1:

Assume using the register settings listed in Table 2 (f_{xosc} = 26 MHz).

Register	Value	Comment
MCSM0	0x18	Calibrate when going from IDLE to RX
WORCTRL	0x78	EVENT1 = 7 and WOR_RES = 0
MCSM2	0x01	RX_TIME = 1

Table 2. Register Settings for Example 1

$$t_{Event1} = 48 \cdot \frac{750}{26 \cdot 10^6} = 1.385 \cdot 10^{-3} = 1.385 [ms]$$ (Equation 4 and Table 1)

It takes 809 us to go from IDLE to RX mode with calibration when using a 26 MHz crystal (see [1] and [2]) .

⇒ $t_{IDLE\ to\ RX\ ^{w}/Calibration}$ = 809 us

MSCM2.RX_TIME = 1 and WORCTRL.WOR_RES = 0 ⇒ Duty cycle = 6.25 % (see [1] and [2]). The duty cycle is used to calculate $t_{RX\ timeout}$ (Equation 7).

$t_{IDLE} = 150 us$ (see 3.2.3)

$$t_{SLEEP_{min}} = \frac{750}{26 \cdot 10^6} \cdot 384 = 11.08 \cdot 10^{-3} = 11.08 [ms]$$ (Equation 2)

$$t_{Event0_{min}} = 1.385 \cdot 10^{-3} + 809 \cdot 10^{-6} + (t_{Event0_{min}} \cdot 6.25\%) + 150 \cdot 10^{-6} + 11.08 \cdot 10^{-3}$$
$$\Rightarrow t_{Event0_{min}} = 14.32 \cdot 10^{-3} = 14.32 [ms]$$ (Equation 3)

$$t_{Event0} = \frac{750}{26 \cdot 10^6} \cdot EVENT0 \cdot 2^{5 \cdot 0} \geq 14.32 \cdot 10^{-3} \Rightarrow EVENT0 \geq 497$$ (Equation 1)

EVENT0 = 497 ⇒ WOREVT1.EVENT0 = 0x01 and WOREVT0.EVENT0 = 0xF1.

$$t_{Event0_{min}} = \frac{750}{26 \cdot 10^6} \cdot 497 \cdot 2^{5 \cdot 0} = 14.34 \cdot 10^{-3} = 14.34 [ms]$$ (Equation 1)

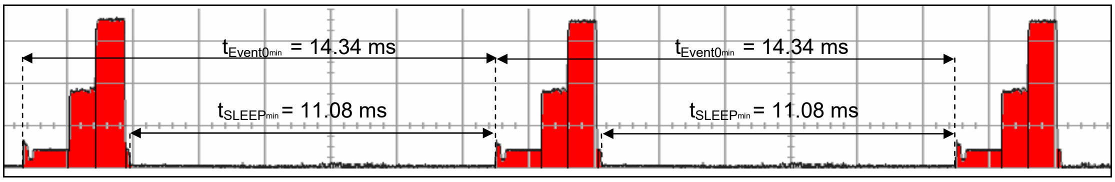

Figure 4. Current Consumption vs. Time (No Packets Received)

Figure 4 shows how the radio will wake up every 14.34 ms when no packets are being received. If a packet is received, the packet will typically be processed by the MCU before the radio is being put back into WOR mode by issuing an SWOR strobe command (see Figure 5).

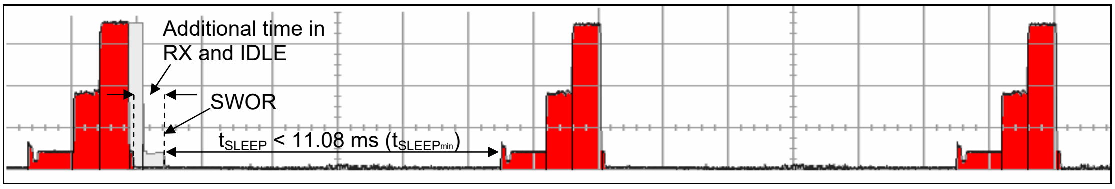

Figure 5. Current Consumption vs. Time (One Packet Received)

When a packet has been received and an SWOR strobe has been issued, t_{SLEEP} becomes less than $t_{SLEEP_{min}}$ and there is a chance that the consecutive Event 0 will occur (750·128) / f_{XOSC} = 3.69·10^{-3} = 3.69 ms too early (see Figure 6). t_{Event0} will in this case be 14.34 ms – 3.69 ms = 10.65 ms.