

## FAQs for SmartLEWIS MCU family: PMA51xx/71xx

Version 1.2, Nov, 16<sup>th</sup> 2012

History info: V1.2 (Items 12-17 added)

Info: All FAQs are valid for all products of the PMA51xx/71xx family, unless otherwise noted.

### **1. What is the current consumption in case of ASK?**

- The values given in the data sheet are valid for FSK modulation. In case of ASK the current consumption is about 15% to 20% less than in case of FSK.

### **2. Is there a possibility to get the frequency deviation of the 12MHz RC Oscillator from its nominal value?**

The offset of the 12MHz RC oscillator after calibration at RT and 3V supply is stored in memory address 0x58AB. The resolution is 10kHz (e.g. value 0x06 says that at RT and 3V supply the 12MHz frequency will be 12.06MHz).

Unfortunately temperature and voltage drift have no unique behavior (no extrapolation from this RT point possible).

A negative offset (e.g. 11.97MHz) is represented by signed char

FF .. -1 \* 10kHz

FE .. -2 \* 10kHz

FD .. -3 \* 10kHz → 11.97MHz

### **3. Where is the LSB located within the 6 byte ID?**

The 6 byte PMA ID is located at position 0x58B8 to 0x58BD, where 0x58B8 is the LSB and 0x58BD is the MSB.

### **4. In case VCOtuning() function is executed during 1<sup>st</sup> Timer0 interval, then the 1<sup>st</sup> Timer0 interrupt interval shows a different interval compared to following Timer0 intervals. How can I avoid this?**

In general, all house-keeping functions, like VCOtuning(), need to be applied before any RF transmission. The function "VCOtuning()" is using and manipulating the DIVIC register (see PMA71xx\_PMA51xx\_Function\_Library\_Guide). DIVIC register changes the clock of the timers, so this is affecting Timer0 and Timer1. And this means Timer0 and Timer1 should not be used together with function "VCOtuning()" at the same time.

So then either finishing the VCO tuning routine before starting Timer0/1 OR usage of Timer2/3 (can also work on the undivided system clock, see TMOD2 register) are the alternative solutions in that case.

## 5. How can I apply the program code download without usage of Keil?

Recommended procedure for programming your application:

1. Take your HEX code (starting at address 0x4000)
2. In case code does not reach address 0x577C, then fill-up the missing addresses with data 0x00
3. Calculate CRC over the whole Code Sector (address 0x4000 to 0x577C) using CCITT polynom 1021H ( $x^{16}+x^{12}+x^5+1$ ) and 16bit start value 0xFFFF
4. The resulting 16-bit CRC gets added at the end of the HEX code (CRC-High to 577DH and CRC-Low to 577EH)
5. The lockbyte 2 gets added at address 0x577F with a value of D1
6. Now the Code sector is ready for download via I2C "FLASH Write Line" command (including Address, Data and CRC of Each Flash write line, but excluding the PMA71xx/PMA51xx I2C device address)
7. ---
8. This download is described in the I2C Appnote under Chapter 3 and 3.1
9. Here again a CRC is required for each "FLASH Write Line" command

Info: CRC can be calculated using <http://www.lammertbies.nl/comm/info/crc-calculation.html> , when apply following steps:

- Use input type = HEX
- Enter the payload
- Press button "Calculate CRC" → watch the result in the line "CRC-CCITT (0xFFFF)"  
(Info: 0xFFFF is the start value of the 16 bit shift register)

In case the Starter Kit or Evaluation Kit from Infineon are used, the CRC calculation over the whole Code sector can be done via Keil (so point 2 to 9 of previous procedure are automatically done by Keil). Here are the steps to activate this function:

- a. Load your Keil project
- b. Go to "Target Options => Utilities"
- c. Use Target = "Infineon PMA Starter Kit Driver" or "Infineon PMA Eval Kit Driver"
- d. Press the "Settings" button
- e. Under "Flash Download Options" activate the checkmark "Set Lock" for the Code Sector ("Set Lock" for "User Data Sector I" is not required here)
- f. Press OK several times and download the HEX file via Keil
- g. Note: "Set Lock" for the Code Sector means that Lockbyte 2 and CRC is added to the HEX file when downloading (same as in above procedure). So Code sector is locked after download.

**6. What is the reason that Key-Fob output power is lower than expected?**

Less than expected RF-power could be related to:

- mismatch in the RF-PA-to-Antenna matching network
- antenna detuning: due to hand or battery or other objects close to antenna, for instance a metalized plastic shell
- lower as nominal supply voltage: voltage drops by „tired“ batteries, if a significant current, >5mA or so by a BR32xx type is sourced during active TX-frame

**7. Can you show how to decide the value of L3 (in the crystal loop) in PMA Key-Fob?**

L3 is a „choke“ meant to enhance immunity against high-level radiations (external fields). The coils inductance must be small enough not to pull off the crystal frequency on one side, on the other side it must be large enough to represent a high reactance by the potential jamming frequencies. IFX recommends values between 82..120 nH for L3.

**8. Is it OK to connect Vbat pin together with Vreg, VDDD and VDDA?**

This may cause a problem in terms of supply voltage regulator performance. Please separate the raw (unregulated) supply pin Vbat from the regulated supply pins Vreg, VDDD and VDDA.

**9. Can you please give some general layout recommendations?**

The loop between crystal component and PMA crystal oscillator pins needs to be kept very small (for having only a small active area, which can receive undesired signals).

Keep in mind to avoid routing the connection to PA pin close to the crystal or below PMA device.

Connection between Vbat and PA matching (choke – pull-up coil) needs to be kept very short.

Both Vbat (unregulated supply) and Vreg/VDDD/VDDA (regulated supply) need its own blocking capacitor very close to the corresponding pins (for having a good energy buffer close to the pin).

**10. How shall I determine the register value XTCFG.XTDLY?**

First of all the real start-up time of the crystal needs to be measured (Note: small crystal package size, like 3225, typically ends up in larger start-up times). The worst case value of the start-up time needs to be determined via measurements (e.g. over temperature, supply voltage, ..) and with worst case crystal parameters. For this resulting worst case start-up time of the crystal an appropriate value in the register XTCFG.XTDLY needs to be selected (Please select the next higher available register value).

**11. Is it possible to debug in Powerdown mode with USB Starterkit? I cannot catch breakpoint after coming from Powerdown (e.g. due to a button press event). If not possible, what are alternatives?**

Debugging during Powerdown mode is not possible. Also debugging when coming out of Powerdown mode is not possible, because the CPU loses break point information during Powerdown mode (break point info is reset during Powerdown). So it is clear now that breakpoint cannot be caught after button press event.

Alternative solution is to use some PPx output pins, which can be controlled in the source code and therefore can be used as debugging output pins.

General information: Entering Debugging mode is only possible, when PMA code sector is not locked.

**12. I get an unexpected LF receiver behavior with extremely large current consumption of about 400µA, when only using Carrier Detect (CDET) and LF baseband is deactivated. What is the reason and how can I reduce the current consumption to a typical value of 1µA for this case (LF RX ON-OFF timer not used in this case)?**

There are several things that need to be kept in mind:

- A) The Carrier Detector Thresholds need to be re-calibrated from time to time (every 1-2 seconds is recommended) due to temperature and supply drift. This calibration should be initiated by using firmware routing "LFSensitivityCalibration()" (e.g. triggered by the Interval timer).
- B) After LFRX is switched off some time needs to be waited before going to Powerdown (about 1.5 – 2 periods of the 2kHz RC oscillator for synchronization in 2kHz domain). Taking tolerances (1.3kHz – 7% drift due to temp + supply => 1.209kHz) into account results in a required delay of 1.241µs for 1.5 periods. By not incorporating this proper delay, the LF carrier detector can become "uncalibrated" very quickly – the common effect is that the carrier threshold is far too low, so the CDET comparator thinks there's always LF coming in.
- C) When not using LF RX ON-OFF timer, then do not use the LF ON time Extension on Carrier Detect (=> LFCDM.LFOOEXT[7] should be set to ZERO). When an LF-carrier got detected, then the 12MHz RC oscillator starts up. Using LFCDM.LFOOEXT[7]=1 means that the 12MHz RC oscillator stays ON (due to LF ON-time extension due to Carrier detect) in case carrier detector still detects a valid signal (maybe even because of item B).

So applying no delay as described in item B, in combination with LFCDM.LFOOEXT[7] set to 1 (see item C) can explain the additional HiRC (12MHz oscillator) current consumption.

**13. Is there an additional start-up time required for the LF receiver?**

At the LF receiver start-up there is an LF calibration time of 6 periods of the 2kHz RC oscillator required. Including all tolerances this ends up in a LF RX calibration time of 5ms in maximum. After this calibration time a preamble of at least 2ms is required for the receiver chain. In case digital LF baseband mode is activated, then the corresponding processing time for the selected baseband mode needs to be taken into account additionally (keep also TX data rate tolerances in mind).

#### 14. How should I handle configuration and activation of the LF receiver?

It is always a good idea to follow a sequence like this:

- Switch off LF RX
- then configure LFRX
- then switch on LFRX

This is a generally valid sequence.

#### 15. I have used PP0 and PP1 in my application. Do you recommend this?

Please avoid using PP0 and PP1 in the application, as these two pins are also used for selection of operating mode. After a Reset a wrong operating mode can be entered accidentally and normal operation mode is no longer possible. PP0 and PP1 need to be used for programming the PMA, but these 2 pins should not be used for anything else.

#### 16. Can the ADC be used in PMA7107 or PMAx105?

There is no external ADC (sensor interface) functionality available in PMA7107 or PMAx105. But the internal temperature sensor and battery voltage measurement functions are available even in these devices.

#### 17. Debug hint: How can I find out if there is some activity in the PMA device?

There are several debug hints:

- A) It is sometimes helpful to monitor pin 31 (Vreg) during operation – this is the bypass capacitor for the internal core voltage regulator. The voltage will “jump” slightly when the device changes modes from Powerdown to Run, for example. So, you can monitor this pin with an AC coupled oscilloscope channel and look for “jumps” that would indicate mode changes.
- B) It would be helpful to have some sort of ‘pin toggle’ in the wakeup to observe if the IT wakeup is running as expected (saying “I’m alive”). PP2 is perhaps a good candidate – to achieve this, add the following in your source code:
  - a. `P1Out |= 0x04;` // turn PP2 ON during each Wakeup ( “|” .. bitwise OR)
  - b. `P1Out &= ~0x04;` // turn PP2 OFF prior to Powerdown ( “&” .. bitwise AND, “~” .. bitwise NOT)

This toggle function of one of the port pins (PPx) is also a good idea for other debug tests during development phase. In general, it is good practice to disable this debug function in the final product as changing pin levels very rapidly can cause interfering signals (resulting in poor RF and LF performance).