

TPMS

Tire Pressure Monitoring Sensor

SP37

Application Note

LF Application Note
Revision 1.0, 2011-12-05

Edition 2011-12-05

**Published by
Infineon Technologies AG
81726 Munich, Germany
© 2012 Infineon Technologies AG
All Rights Reserved.**

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (www.infineon.com).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision History

Page or Item	Subjects (major changes since previous revision)
Revision 1.0, 2011-10-10	
	Initial Version

Trademarks of Infineon Technologies AG

AURIX™, C166™, CanPAK™, CIPOS™, CIPURSE™, EconoPACK™, CoolMOS™, CoolSET™, CORECONTROL™, CROSSAVE™, DAVE™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, HITFET™, HybridPACK™, I²RF™, ISOFACE™, IsoPACK™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OptiMOS™, ORIGA™, PRIMARION™, PrimePACK™, PrimeSTACK™, PRO-SIL™, PROFET™, RASIC™, ReverSave™, SatRIC™, SIEGET™, SINDRION™, SIPMOS™, SmartLEWIS™, SOLID FLASH™, TEMPFET™, thinQ!™, TRENCHSTOP™, TriCore™.

Other Trademarks

Advance Design System™ (ADS) of Agilent Technologies, AMBA™, ARM™, MULTI-ICE™, KEIL™, PRIMECELL™, REALVIEW™, THUMB™, μVision™ of ARM Limited, UK. AUTOSAR™ is licensed by AUTOSAR development partnership. Bluetooth™ of Bluetooth SIG Inc. CAT-ig™ of DECT Forum. COLOSSUS™, FirstGPS™ of Trimble Navigation Ltd. EMV™ of EMVCo, LLC (Visa Holdings Inc.). EPCOS™ of Epcos AG. FLEXGO™ of Microsoft Corporation. FlexRay™ is licensed by FlexRay Consortium. HYPERTERMINAL™ of Hilgraeve Incorporated. IEC™ of Commission Electrotechnique Internationale. IrDA™ of Infrared Data Association Corporation. ISO™ of INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. MATLAB™ of MathWorks, Inc. MAXIM™ of Maxim Integrated Products, Inc. MICROTEC™, NUCLEUS™ of Mentor Graphics Corporation. Mifare™ of NXP. MIPI™ of MIPI Alliance, Inc. MIPS™ of MIPS Technologies, Inc., USA. muRata™ of MURATA MANUFACTURING CO., MICROWAVE OFFICE™ (MWO) of Applied Wave Research Inc., OmniVision™ of OmniVision Technologies, Inc. Openwave™ Openwave Systems Inc. RED HAT™ Red Hat, Inc. RFMD™ RF Micro Devices, Inc. SIRIUS™ of Sirius Satellite Radio Inc. SOLARIS™ of Sun Microsystems, Inc. SPANSION™ of Spansion LLC Ltd. Symbian™ of Symbian Software Limited. TAIYO YUDEN™ of Taiyo Yuden Co. TEAKLITE™ of CEVA, Inc. TEKTRONIX™ of Tektronix Inc. TOKO™ of TOKO KABUSHIKI KAISHA TA. UNIX™ of X/Open Company Limited. VERILOG™, PALLADIUM™ of Cadence Design Systems, Inc. VLYNQ™ of Texas Instruments Incorporated. VXWORKS™, WIND RIVER™ of WIND RIVER SYSTEMS, INC. ZETEX™ of Diodes Zetex Limited.

Last Trademarks Update 2011-02-24

Table of Contents

1	Introduction	7
2	LF Receiver	8
2.1	Voltage Divider	8
2.2	Attenuator for AGC	8
2.3	RSSI Generator	8
2.4	Data Filter and Data Slicer	9
2.5	Data Decoder and Baud Rate Generator	9
2.6	Carrier Detector	10
2.7	Carrier Detector Filter	10
2.8	RC-Oscillator	11
2.9	On-Off-Timer and LF-Receiver	11
2.10	LF-Baseband	11
3	LF Operating Modes	12
4	LF-Telegram	13
5	Example Code	13
5.1	Example 1 - Carrier Detection Mode	13
5.2	Example 2 - Telegram Detection Mode	15
5.3	Example 3 - Mixed Mode Operation	16
5.4	Example 4 - On-Off-Timer usage with Telegram Detection and Carrier Detection Modes	20
5.5	Example 5- On-Off-Timer usage with Mixed Mode	21
5.6	Example 6 - LF Data Reception	22
6	Measurement setup considerations	23
6.1	Current consumption monitoring	23
6.2	LF sensitivity measurement	24
7	LF antenna design	25
8	Appendix: Complementary C definitions	27

List of Figures

Figure 1	SP37 block diagram	7
Figure 2	LF-Receiver Block Diagram	8
Figure 3	LF telegram. The telegram is modulated on a 125 kHz carrier (shaded areas)	13
Figure 4	Program Flow for carrier detection mode.....	13
Figure 5	Example 1 code for Carrier Detection Mode	14
Figure 6	Telegram Detection Mode.....	15
Figure 7	Example 2 code for Telegram Detection Mode.....	16
Figure 8	Mixed Mode.....	17
Figure 9	Example 3 code for Mixed Mode.....	18
Figure 10	Timing of Mixed Mode wakeup	19
Figure 11	Example 4 code for On-Off-Timer with Telegram Mode	20
Figure 12	Example 5 code for On-Off-Timer with Mixed Mode.....	21
Figure 13	On-Off-Timer combined with Telegram Detection Mode	22
Figure 14	Example 6 code for Telegram Mode with data reception.....	23
Figure 15	Circuit for current monitoring. The DC gain is 200	24
Figure 16	Measurement setup for LF sensitivity measurement	24
Figure 17	LF antenna circuit. Left: ideal circuit. Right: real circuit with parasitic elements	25
Figure 18	Complementary C definitions	27

List of Tables

Table 1	Important SFRs associated with Voltage Divider	8
Table 2	Important SFRs associated with Attenuator for AGC.....	8
Table 3	Important SFRs associated with Data Filter and Data Slicer	9
Table 4	Important SFRs associated with Data Decoder and Baud Rate Generator.....	9
Table 5	Important SFRs associated with Carrier Detector.....	10
Table 6	Important SFRs associated with Auto-Calibration function.....	10
Table 7	Important SFRs associated with Carrier Detector Filter.....	11
Table 8	Important SFRs associated with On-Off-Timer and LF-Receiver	11
Table 9	Important SFRs associated with LF-Baseband.....	11
Table 10	LF Operating Modes	12

1 Introduction

The SP37 is a highly specialized and optimized pressure sensor for automotive tire pressure monitoring applications (TPMS). The SP37 contains all of the essential building blocks for a complete TPMS wheel module; only a small number of external components are required. Figure 1 shows a block diagram of the SP37. The device incorporates an 8051-compatible microcontroller which can execute user application code from 6 kByte of on-chip flash memory. Many functions of the device are controlled by Special Function Registers (SFR) which may be manipulated by user code. The SFRs will assume default values upon device reset. In this document only those registers and bits are discussed that need to be changed. However, in the explanation of the LF receiver some SFRs that do not need to be changed are listed for the sake of completeness.

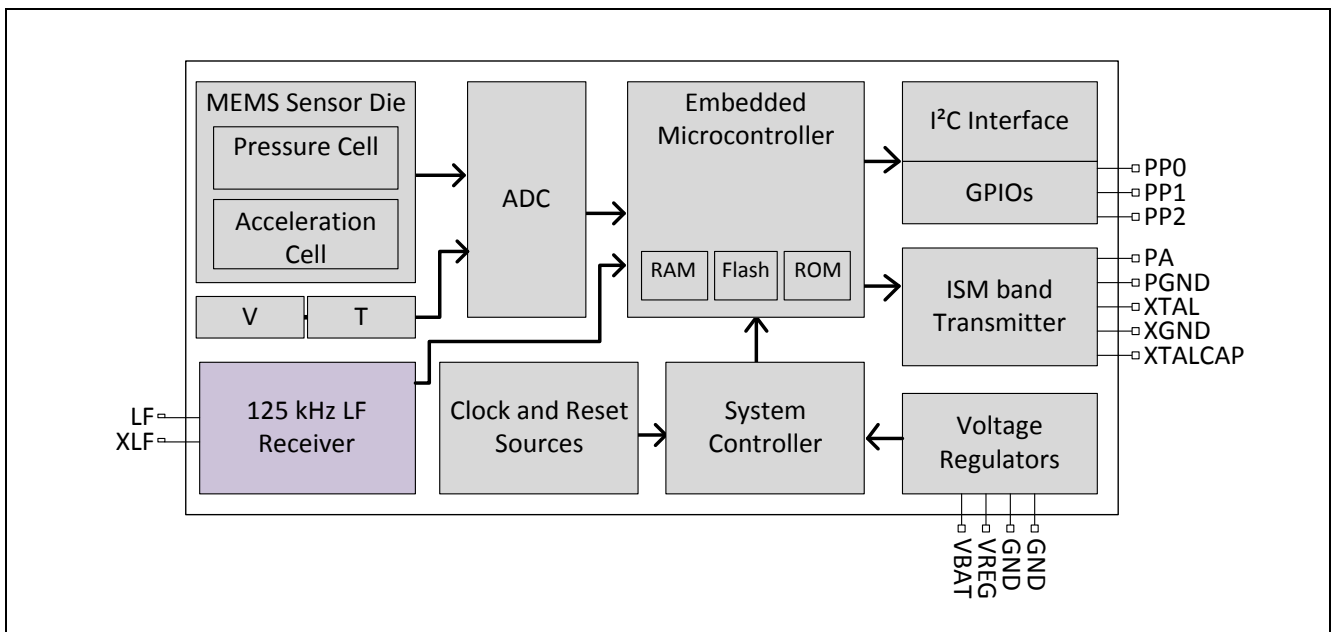


Figure 1 SP37 block diagram

The scope of this application note is the LF Receiver block and how to apply it with respect to lowest possible power consumption. The purpose of the LF interface is to allow a bidirectional communication with the wheel modules, mainly for the following reasons:

- Triggering a pressure measurement (pressure on demand function)
- Triggering the transmission of a unique ID number (wheel localization feature)
- Triggering of operation modes, e.g. diagnosis modes for production and maintenance
- Update of user configuration data, e.g. frequency of pressure telegram transmission

The power consumption of the wheel module is crucial for its lifetime. Hence the communication from the vehicle electronics to the wheel modules cannot be accomplished by RF because the power consumption of an RF receiver is relatively high. An LF receiver can meet the very low power consumption requirements. In contrast, the communication from wheel module back to the vehicle electronics is best accomplished by RF since the power consumption of an RF transmitter is much lower than of an LF transmitter. There are several configuration options for the LF receiver, controlled via SFR, that determine the following system parameters:

2.3 RSSI Generator

This circuit provides an analog signal which varies logarithmically with the amplitude of the 125 kHz input signal.

2.4 Data Filter and Data Slicer

These blocks form a ASK demodulator. The Data Filter is a low pass filter that reduces the bandwidth of the RSSI signal. The Data Slicer is an averaging type that converts the filtered signal into a digital signal that can be processed by the digital baseband circuit.

Important SFRs associated:

Table 3 Important SFRs associated with Data Filter and Data Slicer

Register name <bit number>	Bit name	Function	Reset Value
LFRXS <5>	LFRAX	Output of the data slicer (digital signal), Input of the Data Encoder(read only)	undefined

2.5 Data Decoder and Baud Rate Generator

This digital circuit decodes the Manchester coded LF telegram. It inspects the digital output of the Data Slicer, recognizes the synchronization pattern, decodes the Manchester coded data, detects wakeup pattern matching, and extracts data bytes from the bit stream (see also section LF-Telegram). Wakeup bits in the SFR WUF are set as soon as synchronization pattern or wakeup pattern are recognized. Note that all bits in the WUF register can be masked with a corresponding bit in the WUM register.

Table 4 Important SFRs associated with Data Decoder and Baud Rate Generator

Register name <bit number>	Bit name	Function	Reset Value
LFDIV<5:0>	LFDIV	LF Baud rate generator division factor.	17 _H
LFPCFG<0>	PSEL	Pattern Select mode: 0 = Wakeup on Pattern P0 only, 1 = Wakeup on both Pattern P1 and Pattern P0	0 _B
LFPCFG<1>	PSL	Wakeup Pattern Length: 0 = 8 bit sequence, 1 = 16 bit sequence	0 _B
LFPCFG<4>	SYNM	LF Synchronizer Mode: 0 = Sync and Wakeup Pattern match, 1 = Sync match only	0 _B
LFP0H, LFP0L	LFCODEP0	Wakeup Pattern P0 (16 bit)	FFFF _H
LFP1H, LFP1L	LFCODEP0	Wakeup Pattern P1 (16 bit)	FFFF _H
LFRXC<1>	SYNCIND	1 Indicates sync match, remains set as long as valid Manchester data is detected	0 _B
LFRXD<7:0>	LFRXD	LF Receiver Data (byte)	00 _H
LFRXS<0>	LFDATA	LF serial decoded data (bit)	0 _B
LFRXS<1>	LFBP	Indicates available data bit in LFDATA	0 _B
LFRXS<2>	LFOV	1 = LFDATA overwrite condition	0 _B
LFRXS<3>	LFDP	Indicates available data byte in LFRXD (cleared upon read of LFRXD)	0 _B
LFRXS<4>	LFDOV	1 = LFRXD overwrite condition	0 _B
LFRXS<6>	DECERR	1 = Manchester decode error detected	0 _B
WUF<2>	LFPM0	LF Pattern 0 Match Wakeup	0 _B
WUM<2>	LFPM0_MASK	1 = disable LFPM0 Wakeup	1 _B
WUF<3>	LFPM1	LF Pattern 1 Match Wakeup	0 _B
WUM<3>	LFPM1_MASK	1 = disable LFPM1 Wakeup	1 _B
WUF<4>	LFSY	LF Synchronization Match Wakeup	0 _B
WUM<4>	LFSY_MASK	1 = disable LFSY Wakeup	1 _B

2.6 Carrier Detector

This circuit detects the presence of an LF carrier. The sensitivity of the LF Carrier Detector depends on the settings of the Carrier Threshold, the Voltage Divider and the AGC circuit. The SP37 provides three calibrated sensitivity levels for carrier detection. In order to meet these levels appropriate register settings are determined during production and stored in flash memory at the following memory locations:

- 0x5810: LF-sensitivity = 0.33 ... 3.35 mVpp
- 0x580F: LF-sensitivity = 2 ... 11 mVpp
- 0x580E: LF-sensitivity = 10 ... 50 mVpp

The SFR LFRX0 has to be loaded in user code with the content of one of the three memory locations in order to select the corresponding sensitivity level.

There is a hard-wired function implemented for auto-calibration of the carrier detector threshold. This function automatically shifts the minimum detection threshold above noise level in order to prevent unintended wakeups. If enabled, this function carries out the threshold calibration every time the LF-Receiver is switched on (see section On-Off-Timer and LF-Receiver). After auto-calibration the threshold should be frozen by setting the bit LFENFCTC. Otherwise the threshold follows the average of the input LF signal, resulting in unwanted low LF sensitivity.

Table 5 Important SFRs associated with Carrier Detector

Register name <bit number>	Bit name	Function	Reset Value
LFRXS<7>	CDRAW	Output of the Carrier Detector, Input for the Detector Filter	undefined
LFRX0<1:0>	SELIN	Antenna voltage divider factor	00 _B
LFRX0<3:2>	ATR	AGC Threshold	10 _B
LFRX0<7:4>	CDETT	Carrier Detector Threshold Level	0011 _B

Table 6 Important SFRs associated with Auto-Calibration function

Register name <bit number>	Bit name	Function	Reset Value
LFCDM0<2>	LFENFCTC	1 = freeze threshold after calibration. 0 = do not freeze	0 _B
LFCDM0<3>	LFENCDCAL	Enable LF Carrier Detect Calibration. 1 = enabled	0 _B
LFCDM0<5:4>	DYNTR	Carrier Detector Dynamic Threshold. Set to 01 _B	00 _B
LFRX1<5:4>	ATC	Auto-calibration Time. Use 10 _B for telegram detection and 01 _B for carrier detection.	00 _B
LFRXC<7>	CDRECAL	Restart Carrier Detect Recalibration. Set 1 for calibration start. Bit will be reset automatically.	0 _B

2.7 Carrier Detector Filter

This block rejects short carrier bursts in order to reduce wakeups caused by noise. If the Detector Filter is enabled the carrier burst must be a minimum width before the carrier detector wakeup bit is set. As for the sensitivity, there are predefined filter setting which can be loaded from flash memory into SFR LFCDFLT:

- 0x580D: Detector Filter Time = 62...240µs
- 0x580C: Detector Filter Time = 500 ... 800µs
- 0x580B: Detector Filter Time = 800 ... 1150 µs

Note: In order to disable the filter LFCDFLT has to be loaded with 0x00.

Table 7 Important SFRs associated with Carrier Detector Filter

Register name <bit number>	Bit name	Function	Reset Value
LFCDFLT<6:0>	CDFT	Carrier Detector Filtering Time	00 _H
LFCDFLT<7>	CDFM	Reserved, must be 0	0 _B
WUF<5>	LFCD	LF Carrier Wakeup Bit	0 _B
WUM<5>	LFCD_MASK	LF Carrier Wakeup Bit	0 _B

2.8 RC-Oscillator

This block provides the LF baseband circuit with a 90 kHz system clock.

2.9 On-Off-Timer and LF-Receiver

This timer allows operating the LF receiver in a polled mode for further reduction of current consumption in power-down mode. The On-Off Timer has a programmable time-base and adjustable On- and Off- times. A ROM-Library is available to calibrate the time-base of the On-Off-Timer to 50ms. This function writes a suitable calibration value into the SFR LFOOTP. If another time-base value is desired LFOOTP may be changed in user code, where the time-base is nominally (LFOOTP+1)/2000Hz. Note that the 2000Hz RC oscillator is not calibrated and does vary with temperature and supply voltage.

The actual on-time is determined by the lower nibble of the SFR LFOOT (ONTIM):

$$\text{On-time} = \text{Int}(\text{LFOOT}/4+1) * (\text{ONTIM}+1) * \text{time-base} / (\text{LFOOTP} + 1) \approx (\text{ONTIM}+1) * \text{time-base} / 4.$$

The actual off- time is determined by the higher nibble of the SFR LFOOT (OFFTIM):

$$\text{OFF-time} = (\text{OFFTIM}+1) * \text{time-base} * 4$$

Table 8 Important SFRs associated with On-Off-Timer and LF-Receiver

Register name <bit number>	Bit name	Function	Reset Value
LFOOT<3:0>	ONTIM	On/Off Timer on-time: 0000 _B : 12.5 ms ... 1111 _B : 200 ms	0000 _B
LFOOT<7:4>	OFFTIM	On/Off Timer off-time. 0000 _B : 200 ms ... 1111 _B : 3.2 s	0000 _B
LFOOTP<7:0>	LFOOTP	LF ON/OFF Timer Precounter	64 _H
LFRXC<0>	LFONIND	LF receiver ON/OFF Indicator. Can be used for indication of the ON/OFF timer duty cycle. 1 = LF receiver is on.	0 _B
LFRXC<2>	ENLFRX	0 Disable LF Receiver 1 LF Receiver state is determined by On/Off Timer	0 _B
LFRXC<3>	ENOOTIM	0 Disable On/Off Timer 1 Enable On/Off Timer	0 _B

2.10 LF-Baseband

In order to reduce power-down current the LF-Baseband can be switched off while the LF analog FE remains powered. In this mode the SP37 cannot analyze the content of a LF telegram and will always be woken up if the LF-carrier is strong enough.

Table 9 Important SFRs associated with LF-Baseband

Register name <bit number>	Bit name	Function	Reset Value
LFRXC<5:4>	LFBBM	00 _B Disable LF baseband (e.g. LF Carrier Detect only) 01 _B Enable LF baseband (e.g. LF Carrier Detect and/or LF Telegram) 1X _B reserved, do not use	00 _B

3 LF Operating Modes

The LF operating modes discussed in this application note are listed in the following table. The table also shows typical current consumption and pros and cons for each mode. Please be aware that the current consumption only applies for power-down where the LF receiver is listening. During normal mode (execution of user code) the current consumption is considerably higher. However, current consumption in power-down mode determines battery lifetime because in a typical TPMS application the wheel module is 99% of its time in power-down.

It is important to note that if many unintended wakeup events per hour occur due to interference sources the power consumption of the device may increase considerably. This is particularly critical for the Carrier Detection Mode and the Mixed Mode, where user code execution is triggered by just the presence of an LF-carrier. *Hence these modes cannot be recommended generally for all applications.* In fact it must be made sure on system level that a certain number of unintended wakeups per hour is not exceeded (see also calculations in section "Example 3 - Mixed Mode Operation"). The probability of false wakeups can be lowered by enabling the Carrier Detector Filter. However, this increases current consumption to a level similar as in Telegram Detection Mode. So, for low current consumption in an environment with high interference level the Telegram Detection Mode with On-Off-Timer enabled is recommended.

Table 10 LF Operating Modes

LF Operating Mode	Typ. Current	LF analog front end	LF Baseband	Comment
Telegram Detection (TD)	4.4 μ A	on	on	+ selective-wake up of modules + highest possible LF sensitivity + 100% listening time - high current consumption
Carrier Detection (CD)	3.1 μ A (Carrier Detector Filter disabled!)	on	off	+ 100% listening time + lower current consumption - no selective wakeup of wheel modules - lower LF sensitivity - wakeup of CPU by interference sources possible
Mixed Mode (TD + CD)	3.1 μ A (Carrier Detector Filter disabled!)	on	off	+ selective-wake up of modules + 100% listening time + lower current consumption - lower LF sensitivity - wakeup of CPU by interference sources possible
On-Off-Timer plus TD	<1 μ A (average)	6% on-time	6% on-time	++ lowest current consumption + selective-wake up of modules + highest possible LF sensitivity - reduced LF- listening time, causes wakeup delay

4 LF-Telegram

LF Telegrams must start with a preamble in order to establish the LF Data Slicer threshold in the LF Analog Front End. The preamble must have a 50% duty cycle. The minimum length of the preamble is 2ms. The preamble is followed by the sync pattern, see Figure 3. Following the sync pattern are an optional 8 or 16 bit long wake up ID and an arbitrary number of data bytes. Wakeup ID and data bytes are Manchester encoded. The default bit time of SP37 is $t_{bit}=256\mu s$ (baud rate = 3906/s).

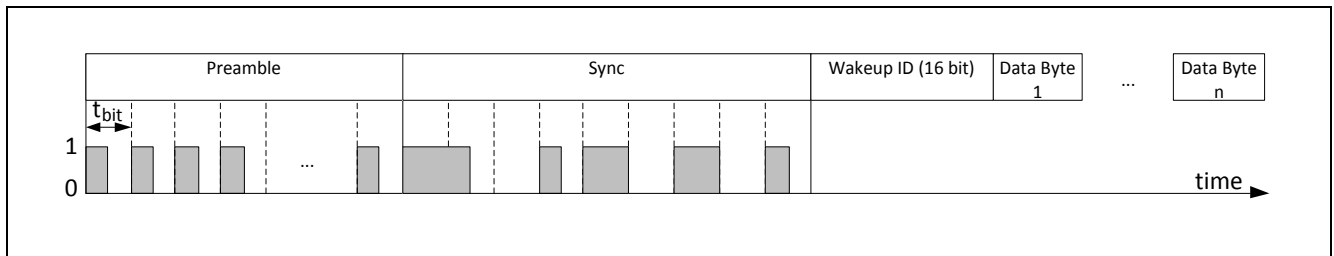


Figure 3 LF telegram. The telegram is modulated on a 125 kHz carrier (shaded areas)

5 Example Code

The code examples are consecutive with increasing complexity and they are based on each other. Hence it is recommended to start reading this chapter from the beginning. Please also note the hints in the appendix before trying the example code provided in this application note.

5.1 Example 1 - Carrier Detection Mode

Figure 4 shows the program flow of a wakeup on carrier detection. Basically the user code is entered on each device reset or wakeup event. Hence there must be a determination of the wakeup cause at the beginning of the code. Wakeups events cause dedicated flags in the SFR WUF to be set. By analyzing the contents of WUF, the wakeup source can be identified.

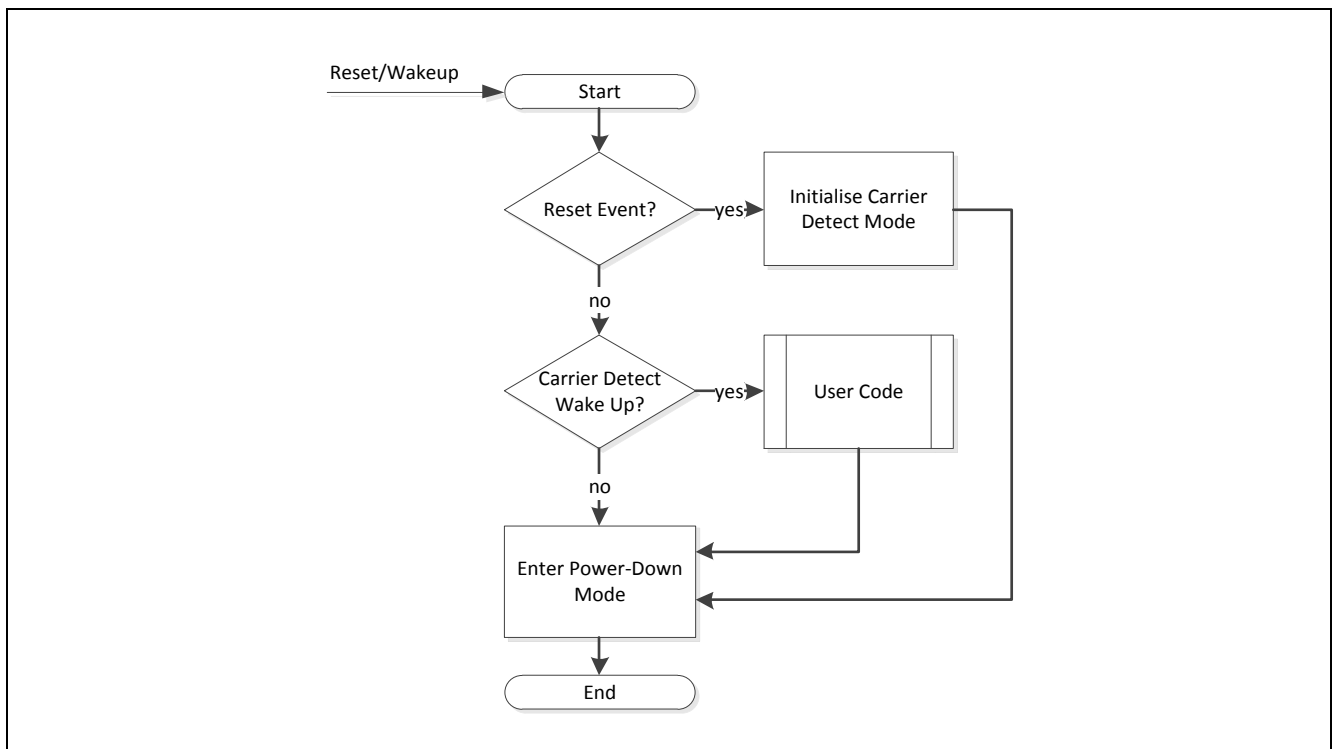


Figure 4 Program Flow for carrier detection mode

The source code for this example is given in Figure 5. The LF Carrier Detector is initialized after device reset. The actual user code is executed only after a carrier detection wakeup. In this example the Detector Filter is disabled (LFCDFLT = 0x00;). The filter may be enabled by using one of the following assignments LFCDFLT = CBYTE[0x580D]; or LFCDFLT = CBYTE[0x580C]; or LFCDFLT = CBYTE[0x580B]; (see the Carrier Detector Filter description). Using the Carrier Detector Filter will increase noise immunity by rejecting short LF 'burst noise' pulses, at the expense of slightly higher current consumption because the Carrier Detector Filter is part of the LF-baseband circuit and thus the RC-oscillator of the LF-baseband remains active.

The carrier detection threshold is determined by the assignment LFRX0 = CBYTE[0x5810];. This setting selects the lowest specified carrier detection threshold (most sensitive setting). See the Carrier Detector description for more details.

The SP37 includes an Interval Timer which periodically, every 0.5 s by default, generates a non-maskable wakeup event. An Interval Timer wakeup is indicated by WUF<0> being set upon wakeup. In this code example the device returns to power-down immediately after interval-timer wakeup, because the corresponding bit (WUF<0>) does not trigger any action. However, frequent Interval Timer wakeups make it difficult to measure the power-down current. The interval timer cannot be disabled so in order to overcome this problem the Interval Timer wakeup interval is increased to approx. 2 min by the assignment ITPR=0x00; (details see specification).

The auto-calibration function for the minimum carrier detection threshold is enabled by the assignment LFCDM0 = 0x1C;. Without this assignment the auto-calibration is disabled, resulting in a lower detection threshold but with higher risk of unwanted wakeups.

```
void main (void)
{
    unsigned char store_wuf;

    store_wuf = WUF;
    if (store_wuf == 0x00){
        LFCDM0 = 0x1C;
        LFRX0 = CBYTE[0x5810];

        LFRX1 = 0x10;
        LFCDFLT = 0x00;

        WUM &= ~(0x20);
        LFRXC = 0x04;
        ITPR=0x00;
    }
    if (store_wuf & 0x20){
        // *Place your user code here*
        RS232_Init(PP2,PP1);
        printf("\r\nCarrier Detected");
        RS232_UnInit(PP2,PP1);
    }
    Powerdown();
}
```

//Load store_wuf with WUF. This action clears WUF.
//Reset value of WUF is 0x00
//Enable auto-calibration & freeze threshold
//Load Carrier Detector Threshold from flash
// location 0x5810 = 0.33 to 3.35 mVpp
//Choose auto-calibration time for carrier detection
//Disable Carrier Detector Filter for lowest current
// consumption in power-down.
//Set wakeup mask for carrier detection wakeup
//LF Baseband disabled, LF-Receiver enabled
//Set Interval Timer to approx. 2 min
//
//For demonstration purposes the predefined RS232
// functions are used to send out a string via
// RS232 Interface. Pin PP2 is used as TX, PP1 as RX
//
//This ROM-library function switches device into
// power-down mode.

Figure 5 Example 1 code for Carrier Detection Mode

5.2 Example 2 - Telegram Detection Mode

Figure 6 shows the program flow for Telegram Detection Mode. The structure is the same as that of the Carrier Detection Mode code example. There are two important differences within the initialization phase: The LF-Receiver is configured for telegram reception, and the LF Baseband Baud Rate generation is calibrated.

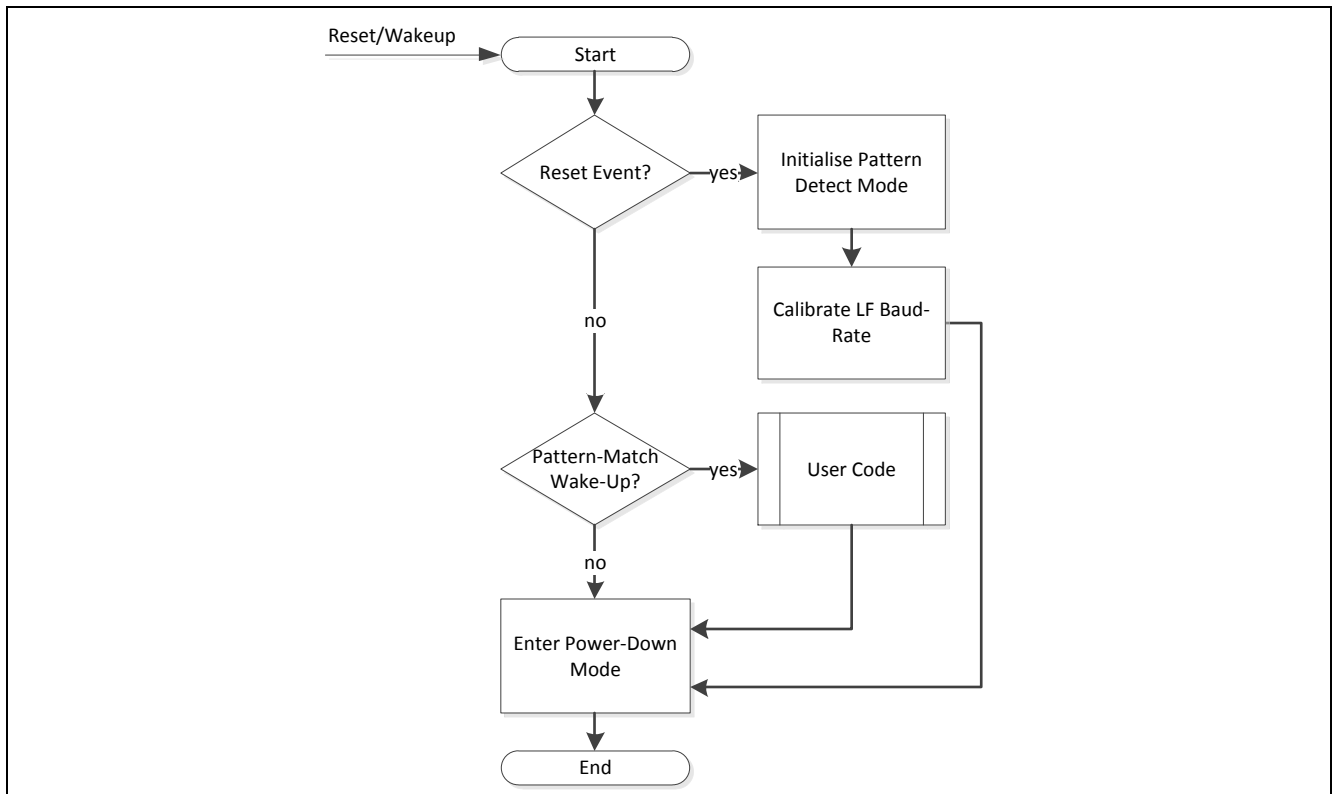


Figure 6 Telegram Detection Mode

A predefined ROM-Library function is used for calibration of the LF Baud rate to 3906 bit/s. This function uses the RF-quartz oscillator as a reference. Hence the quartz needs to be switched on before calibration by calling the ROM-Library function `StartXtalOsc(40)`; (see code listing in Figure 7). The parameter 40 defines a delay time of $40 \times 42.67\mu\text{s}$ before the next command executed in order to let the oscillator stabilize. The function call `StopXtalOsc()`; stops the RF oscillator. There are up to three wakeup events available if a pattern match is detected (see Data Decoder section for more details). In the code example (see code listing 2) the LF Pattern 0 Match Wakeup is used and SFR LFPCFG is configured accordingly. Pattern P0 can be defined arbitrary by setting the SFRs LFP0H and LFP0L. In this example it is set to 0x1234, and Pattern P1 is not used. Each Pattern has its own wakeup event flag, so the Wakeup Mask register (SFR WUM) must be configured to enable Pattern P0 Wakeup. Furthermore, the baseband is switched on by setting bit 4 in SFR LFRXC.

```
void main (void)
{
    unsigned char store_wuf;

    store_wuf = WUF; //Load store_wuf with WUF. This action clears WUF.
    if (store_wuf == 0x00){ //Reset value of WUF is 0x00
        LFP_CFG = 0x02; //Use 16Bit pattern P0 for wakeup
        LFP0H = 0x12; LFP0L = 0x34; //Definition of P0 high byte and low byte
        LFRX1 = 0x20; //Choose auto-calibration time for telegram detection
        WUM &= ~(0x04); //Enable Pattern Match Wakeup
        LFRXC = 0x14; //LF-Receiver including LF Baseband enabled
        ITPR=0x00; //Set Interval Timer to approx. 2 min
        StartXtalOsc(40); //Start RF quartz oscillator and wait 40x42.67µs
        LFBaudrateCalibration(3906); //Calibrate LF Baud Rate to 3906
        StopXtalOsc(); //Stop RF quartz oscillator
    }
    if (store_wuf & 0x04){
        // * Place your user code here *
        RS232_Init(P2,P1); //For demonstration purposes the predefined RS232
        printf("\r\nPattern Detected"); // functions are used to send out a string via
        RS232_UnInit(P2,P1); // RS232 Interface. Pin P2 is used as TX, P1 as RX
    }
    Powerdown(); //This ROM-library function switches device into
                // power-down mode.
}
```

Figure 7 Example 2 code for Telegram Detection Mode

5.3 Example 3 - Mixed Mode Operation

Figure 8 shows the flow of an example of the LF Receiver in Mixed Mode. In this mode, the LF Base Band is switched off in power-down and only turned on after LF Carrier Detection. As soon as a carrier is detected the device starts user code execution. The user code first switches on the LF Base Band and then starts checking for a pattern match event. Hence this mode combines low power-down current with Pattern Detection. Since the current consumption during user code execution is high (typ. 1.4 mA) a timeout loop is used to return to power-down mode soon, if no pattern match occurs. If a pattern match is detected the actual user code will be executed.

Figure 9 shows the code that demonstrates how to use the Mixed Mode. In the initialization section actually both, the pattern match and the carrier detection wakeup are configured. However, since the LF baseband is turned off before entering power-down the pattern match wake up is inactive in this phase. Timer 0 is used as time-out timer. In the code example a time-out of 25 ms is chosen. The time-out duration must be greater than the sum of preamble length, sync length and matching pattern length (see also Figure 10).

As mentioned before it is important to consider that in a noisy environment the carrier detection threshold needs to be high enough in order to prevent frequent unintended wakeups. Otherwise there will be no more power saving advantage of the Mixed Mode compared to Telegram Mode. At the worst the average current consumption in Mixed Mode becomes even higher than in Telegram Mode.

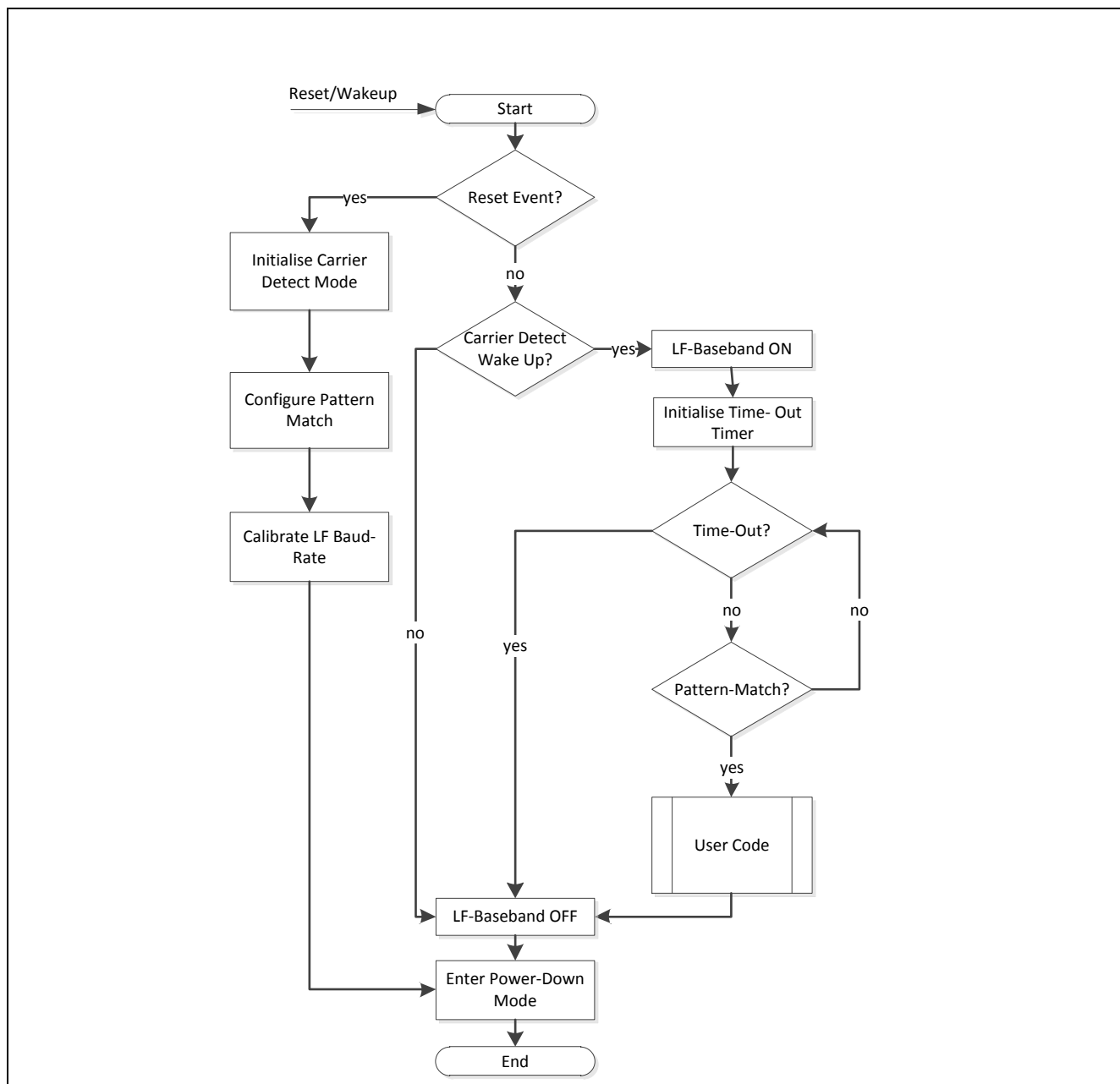


Figure 8 Mixed Mode

```
void main (void)
{
    unsigned char store_wuf;
    unsigned char pattern_match;

    store_wuf = WUF;
    if (store_wuf == 0x00){
        LFCDM0 = 0x1C;
        LFRX0 = CBYTE[0x5810];
        LFRX1 = 0x10;
        LFCDFLT = 0x00;
        WUM &= ~(0x24);
        ITPR=0x00;
        LFPFCFG = 0x02;
        LFP0H = 0x12; LFP0L = 0x34;
        StartXtalOsc(40);
        LFBaudrateCalibration(3906);
        StopXtalOsc();
    }
    if (store_wuf & 0x20){
        LFRXC = 0x14;
        TORUN = 0;
        TMOD = 0x51;
        TH0 = 0x92; TL0 = 0x7C;
        TORUN = 1;
        do {
            store_wuf = WUF;
            if (store_wuf & 0x04){
                pattern_match=1;
            }
        }while ((!TOFULL)&&!(store_wuf & 0x04)); //Exit loop if time out or pattern match
        RS232_Init(PP2,PP1);
        if (pattern_match==1){
            // *** Place your user code here ***
            printf("\r\nP0 Match");
        }
        else printf("\r\nCarrier Detected");
        RS232_UnInit(PP2,PP1);
    }
    LFRXC = 0x04;
    pattern_match=0;
    Powerdown();
}
```

Figure 9 Example 3 code for Mixed Mode

The following formula calculates the number N of unintended wakeups per hours for which current consumption of Mixed Mode and Telegram Mode becomes equal:

$$N = \frac{3600s(I_{powerdown}^{TD} - I_{powerdown}^{MM})}{t_{timeout} I_{normalmode}} \quad (1)$$

A typical value for N is

$$N = \frac{3600s \cdot (4.4\mu A - 3.1\mu A)}{25ms \cdot 1.4mA} = 133 \quad (2)$$

In praxis it should be taken care for N being much smaller than this value by proper choice of carrier detection threshold.

Regarding telegram length the same applies for Mixed Mode and Telegram Mode. In both modes the preamble is needed for adapting the data detection threshold to the received carrier amplitude. For this purpose a minimum preamble length of 2ms is specified.

In Mixed Mode the preamble is also used for Carrier Detection Wakeup. In order to be still able to detect the telegram after switching on the baseband, the minimum length of the preamble must be greater than the sum of carrier detection latency, wakeup time and baseband settling time.

The carrier detection latency is in the order of 200µs (if detection filter is off); the wakeup time is about 1ms. The baseband settling time is less than 500µs. Hence a preamble length of 2ms is sufficient for Mixed Mode operation.

Figure 10 shows the oscillograph curve of a wakeup telegram (blue) along with the SP37 current consumption (red), where the SP37 is programmed with the code of Figure 9. (See also section "Hardware Considerations" for how to monitor the SP37 supply current). The carrier detection latency and the wakeup time can be clearly observed in the oscillogram, whereas the baseband settling time is not visible. However, the baseband settling time can be estimated by variation of the preamble length. The oscillating power consumption at the end is due to the RS232 operation (transmission of the string "P0 Match"). Furthermore it can be seen in Figure 10 that the RS232 transmission starts right after the pattern P0 = 0x1234 has been recognized. Note that the two data bytes following the wakeup pattern P0 are ignored in this example.

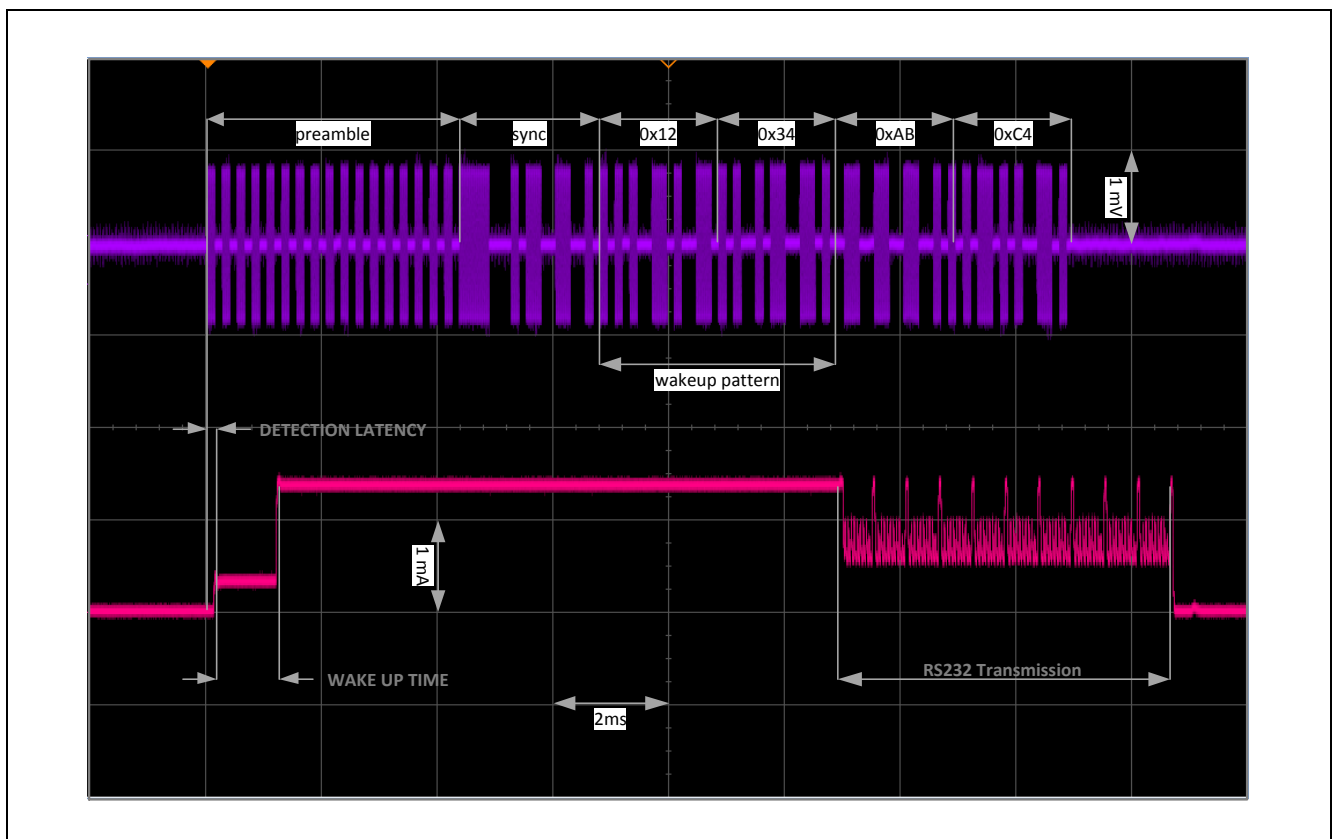


Figure 10 Timing of Mixed Mode wakeup

5.4 Example 4 - On-Off-Timer usage with Telegram Detection and Carrier Detection Modes

A considerable reduction of current consumption in power-down can be achieved by only periodically activating LF and using the receiver in a polled fashion. The SP37 LF receiver supports this scenario with its built-in LF On-Off-Timer. This timer periodically switches on and off the LF-Receiver independently of the CPU. See section On-Off-Timer and LF-Receiver for the definition of On- and Off-time. Listing 4 shows the corresponding code. Compared to the Telegram Detection Mode without On-Off-Timer, there are only a few additional steps that must be taken. First, the ON- and OFF-time must be defined by setting SFR LFOOT. Second, the time base of the On-Off-Timer must be defined by setting the precounter register (SFR LFOOTP). This should be done automatically by using the ROM library function "IntervalTimerCalibration" which calibrates the register LFOOTP for a time base of exactly 50ms. This function also calibrates the Interval Timer (see section Example 1 - Carrier Detection Mode). The function argument (here 2) adjusts the Interval Timer time base to 500ms but has no effect on the On-Off-Timer time base. Finally, the LF ON-OFF Timer must be activated before entering power-down by setting the corresponding bit LFRXC<3> (ENOOTIM = 1;). This is necessary because ENOOTIM is cleared automatically each time a sync or pattern match wakeup occurs.

```
void main (void)
{
    unsigned char store_wuf;

    store_wuf = WUF;                //Load store_wuf with WUF. This action clears WUF.
    if (store_wuf == 0x00){          //Reset value of WUF is 0x00
        LFPCFG = 0x02;              //Use 16Bit pattern P0 for wakeup
        LFP0H = 0x12; LFP0L = 0x34; //Definition of P0 high byte and low byte
        LFRX1 = 0x20;               //Choose auto-calibration time for telegram detection
        WUM &= ~(0x04);             //Enable Pattern Match Wakeup
        LFRXC = 0x14;               //LF-Receiver including LF Baseband enabled
        ITPR=0x00;                  //Set Interval Timer to approx. 2 min
        LFOOT = 0x22;               //ON-Time = 37.5 ms , OFF-Time = 600ms
        StartXtalOsc(40);           //Start RF quartz oscillator and wait 40x42.67µs
        LFBaudrateCalibration(3906); //Calibrate LF Baud Rate to 3906
        IntervalTimerCalibration(2); //Calibrate On-Off-Timer precounter to 50ms
        StopXtalOsc();              //Stop RF quartz oscillator
    }
    if (store_wuf & 0x04){
        // * Place your user code here *
        RS232_Init(PP2,PP1);        //For demonstration purposes the predefined RS232
        printf("\r\nPattern Detected"); // functions are used to send out a string via
        RS232_UnInit(PP2,PP1);      // RS232 Interface. Pin PP2 is used as TX, PP1 as RX
    }
    ENOOTIM = 1;                  //Activate On-Off-Timer
    Powerdown();                    //This ROM-library function switches device into
    // power-down mode.
}
```

Figure 11 Example 4 code for On-Off-Timer with Telegram Mode

The disadvantage of switching the LF-receiver is a reduction of LF-listening time. Hence, for safe wakeup the telegram needs to be repeated several times, i.e. a burst of repeated telegrams must be transmitted. There are two conditions for safe wakeup:

1. The telegram burst must be longer in duration than the sum of the LF On-Off-Timer 'ON' and 'OFF' durations.
2. The time of two subsequent telegrams (including any pause in-between telegrams) must be shorter than the LF ON-OFF Timer 'ON' time minus the LF receiver settling time.

It is important to know that the effective listening time is shorter than the On-time due to the LF-receiver settling time of maximal 5.8 ms (see SP37 specification).

The oscillogram in Figure 13 shows these timing requirements, wherein the blue curve is the telegram burst and the red curve represents the SP37 supply current. The resolution of the current curve is high enough to detect the increase of supply current when the LF-Receiver is switched on.

The burst in Figure 13 is about 650ms long which is longer than the sum of ON- and OFF-time of 637.5ms, i.e. condition 1 is fulfilled. The listening time is approximately 37.5ms - 5.8 ms = 31.7ms. The telegram lasts 13ms, the pause 4ms. Hence also condition 2 is fulfilled, because $2 \times 13\text{ms} + 4\text{ms} = 30\text{ms} < 31.7\text{ms}$.

5.5 Example 5- On-Off-Timer usage with Mixed Mode

Usage of the On-Off-Timer with Mixed Mode is basically the same as with TD- or CD-Mode. However, reactivation of the timer after wakeup is different because in Mixed Mode the LF Base Band is switched independently from the LF-receiver and internal clock synchronization processes are necessary. Figure 12 **Error! Reference source not found.** shows how to use the On-Off-Timer with Mixed Mode.

Before entering power-down the LF receiver must be switched off by clearing the bit ENLFRX (LFRXC &=0xFB;). Then, after a 128µs delay by calling the ROM library function Wait100usMultiples(1), the LF-receiver is reconfigured with the assignment LFRXC = 0x0C;. This procedure allows internal synchronization and assures reactivation of the On-Off-Timer after wakeup.

```
void main (void)
{
    unsigned char store_wuf;
    unsigned char pattern_match;

    store_wuf = WUF;
    if (store_wuf == 0x00){
        LFCDM0 = 0x1C;
        LFRX0 = CBYTE[0x5810];

        LFRX1 = 0x10;
        LFCDFLT = 0x00;

        WUM &= ~(0x24);

        ITPR=0x00;
        LFOOT = 0x22;
        LFPFCFG = 0x02;
        LFP0H = 0x12; LFP0L = 0x34;
        StartXtalOsc(40);
        LFBaudrateCalibration(3906);
        IntervalTimerCalibration(2);
        StopXtalOsc();
    }
    if (store_wuf & 0x20){
        LFRXC = 0x14;
        TORUN = 0;
        TMOD = 0x51;
        TH0 = 0x92; TL0 = 0x7C;
        TORUN = 1;
        do {
            store_wuf = WUF;
            if (store_wuf & 0x04){
                pattern_match=1;
            }
        }while ((!TOFULL)&&!(store_wuf & 0x04)); //Exit loop if time out or pattern match
        RS232_Init(PP2,PP1); //Use predefined RS232 functions for demonstration
        if (pattern_match==1){
            // *** Place your user code here *** //Place code to be executed on pattern match here
            printf("\r\nP0 Match"); //Transmit string for demonstration purpose
        }
        else printf("\r\nCarrier Detected"); //Print this string if no pattern match detected
        RS232_UnInit(PP2,PP1);
    }
    LFRXC &= 0xFB; //LF Receiver OFF
    Wait100usMultiples(1); //Delay for synchronization purposes (approx. 128µs)
    LFRXC = 0x0C; //LF Receiver ON, Enable interval timer, Baseband OFF
    pattern_match=0; //Reset pattern match indicator
    Powerdown(); //This ROM-library function switches device into
                //power-down mode.
}
```

Figure 12 Example 5 code for On-Off-Timer with Mixed Mode

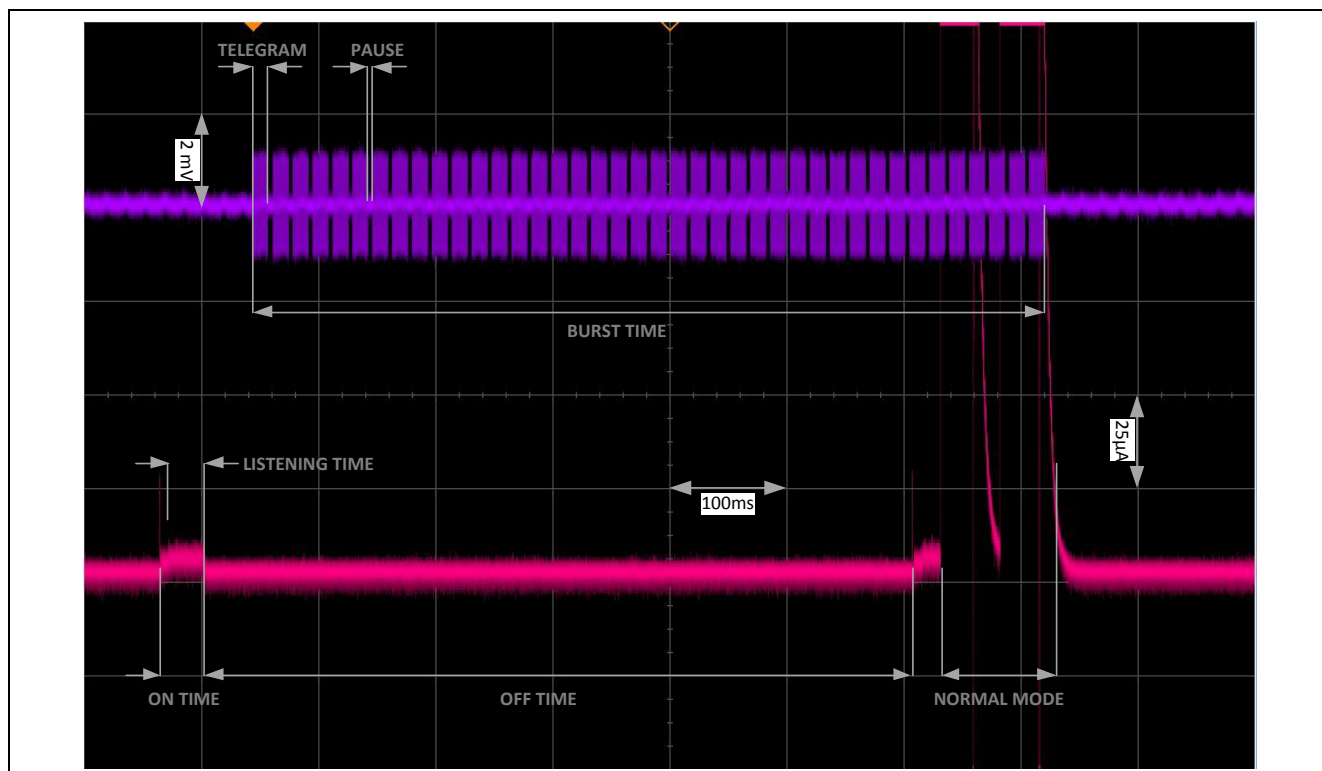


Figure 13 On-Off-Timer combined with Telegram Detection Mode

5.6 Example 6 - LF Data Reception

The SP37 baseband includes an LF Receiver Data Interface which automatically receives and decodes Manchester encoded data bits and bytes following the sync and optional wakeup ID matching pattern. As each bit is received it is latched in bit LFDATA (SFR LFRXS<0>) and the data bit pending indicator flag LFBP (SFR LFRXS<1>) is set. If a new data bit arrives before the previous LF data bit is read from LFDATA, the Serial Decoded Data Overwritten flag, LFOV (SFR LFRXS<2>) will be set. Reading LFRXS will clear LFBP and LFOV. As each group of eight data bits (i.e. a full data byte) is received, the entire byte is latched in the SFR LFRXD and the data byte pending indicator flag LFDP (SFR LFRXS<3>) is set. If a new data byte arrives before the previous LF data byte is read from LFRXD, the Data Byte Overwritten flag, LFDOV (SFR LFRXS<4>) will be set. During user code execution, LFDP may be polled to see if a new data byte is available in LFRXD.

In idle state the μ C core is stopped but the LF-receiver and other circuits like the timer module are still working. Hence current consumption in idle mode is lower than in normal mode but still higher than in power-down mode. Because LF data byte reception is one of the events that can cause the SP37 to resume from idle state, there is an opportunity to reduce power consumption during LF reception.

After wakeup due to LF pattern match the SP37 can be switched to idle state. As soon as a data byte has been received the device resumes from idle state and the data can be read from LFRXD. In this fashion, the overall power consumption is reduced during LF data reception. Figure 14 shows how this method may be implemented. Timer0 is used for generation of a timeout event for resuming the SP37 from idle state if no more data is received. For all resume events a corresponding bit exists in the SFR REF. Hence REF is analyzed in the code in order to distinguish between the two resume events. In case of timeout the code leaves the reception loop and jumps back into power-down mode.

```

void main (void)
{
    unsigned char store_wuf;

    store_wuf = WUF;                                //Load store_wuf with WUF. This action clears WUF.
    if (store_wuf == 0x00){                          //Reset value of WUF is 0x00
        LFPCFG = 0x02;                              //Use 16Bit pattern P0 for wakeup
        LFP0H = 0x12; LFP0L = 0x34;                //Definition of P0 high byte and low byte
        LFRX1 = 0x20;                              //Choose auto-calibration time for telegram detection
        WUM &= ~(0x04);                            //Enable Pattern Match Wakeup
        LFRXC = 0x14;                              //LF-Receiver including LF Baseband enabled
        ITPR=0x00;                                  //Set Interval Timer to approx. 2 min
        StartXtalOsc(40);                          //Start RF quartz oscillator and wait 40x42.67µs
        LFBaudrateCalibration(3906);               //Calibrate LF Baud Rate to 3906
        StopXtalOsc();                             //Stop RF quartz oscillator
    }
    if (store_wuf & 0x04) Data_Receive();           //Call data receive function after Pattern Match
                                                    // Wakeup
    Powerdown();                                   //This ROM-library function switches device into
                                                    // power-down mode.
}

void Data_Receive()
{
    unsigned char LFData[MAX_LF_DATA];
    unsigned char i, index=0, store, RECEIVING=TRUE;

    LFData[0]= LFRXD;                             //Reset data receiver modul by reading LFRXD
    while (RECEIVING)                             //Enter receiving loop
    {
        TORUN = 0;                                //Stop Timer0
        TMOD = 0x51;                              //Set timer mode 1 and timer clock = 1.5MHz
        TH0 = 0x11; TL0 = 0x94;                  //Configure Timer0 for 3ms timeout
        TORUN = 1;                                //Start Timer0
        IDLE = 1;                                 //Enter IDLE Mode
        store = REF;                              //Load store with REF. This action clears REF
        if (store & 0x10)                         //Resume from idle due to data reception event?
            LFData[index++] = LFRXD;              // if yes, then read LF data
        if (store & 0x01)                         //Resume from idle due to timer0 timeout?
            RECEIVING = FALSE;                   // if yes, then exit receiving loop
    }
    RS232_Init(P2,P1);                            //Use predefined RS232 functions for demonstration
    printf("\r\nPATTERN Match! LF-Data: ");      // purposes.
    for(i=0;i<index;i++)                          // Send out received LF-data via RS232 interface
        RS232_Send_Hex(LFData[i]);              //
    RS232_UnInit(P2,P1);                          //
}

```

Figure 14 Example 6 code for Telegram Mode with data reception

6 Measurement setup considerations

6.1 Current consumption monitoring

The evaluation board has a 10Ω resistor in series with the SP37 power supply pin. This resistor is intended for current consumption measurement. If only average or power-down current consumption is of interest, a micro voltmeter can be connected directly to the jumper pins X12 which represent the two resistor terminals, provided that no terminal of the µV-meter is grounded. For time resolved current monitoring a differential amplifier is needed because a normal oscilloscope cannot be directly connected to the resistor terminals. For measuring the supply current in Figure 10 and Figure 13 the circuit shown in Figure 15 has been used. It is important to notice that an offset is generated by the current through the 100kΩ resistor which is connected to the non-inverting input of the operational amplifier. For compensation of this offset as well as for other offsets the 1kΩ trimmer needs to be adjusted properly. It is recommended to use operational amplifiers with low offset voltage and low input bias current.

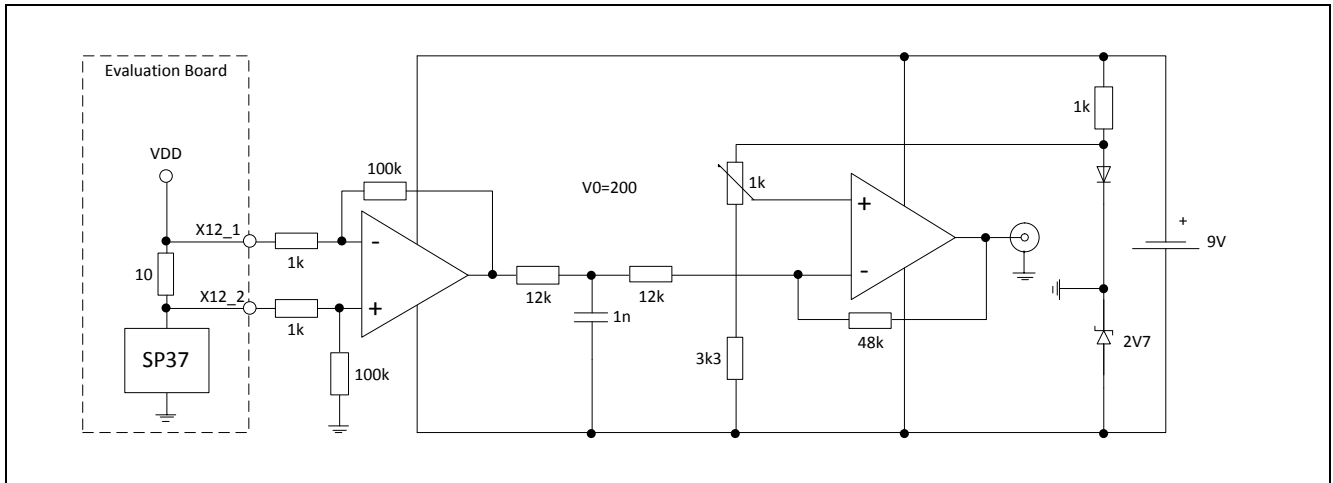


Figure 15 Circuit for current monitoring. The DC gain is 200

6.2 LF sensitivity measurement

The SP37 evaluation board comes with a SMA connector for LF-Input (50Ω input impedance) which is intended for LF sensitivity measurements. Figure 16 shows the recommended measurement setup.

The function generator should allow a minimum amplitude resolution of 0.1 mV. For the tests described here an Agilent 33250A function generator was used. The -20dB attenuator improves the amplitude resolution of the test setup. Alternatively, an RF signal generator capable of operating at 125 kHz may be used; suitable types include the Rhode & Schwarz SMT and SME series. An RF signal generator will generally not require the 20dB attenuator in order to achieve the required resolution.

The BALUN L1 transforms the single ended LF-input signal into a differential signal and doubles the voltage. The relation between the generator voltage V_{gen} and the actual voltage V_{in} between the terminals LF-XLF with an optional attenuator in-line is:

$$V_{in} = \left(10^{\frac{-Attenu_{dB}}{20}}\right) \cdot 2V_{gen} \quad (3)$$

Because the generator voltage is often calibrated in volts RMS and the SP37 LF sensitivity specification is expressed in peak-to-peak voltage, the following equation may be used:

$$V_{in_{mVpp}} = 4\sqrt{2}V_{gen_{RMS}} \left(10^{\frac{-Attenu_{dB}}{20}}\right) \quad (4)$$

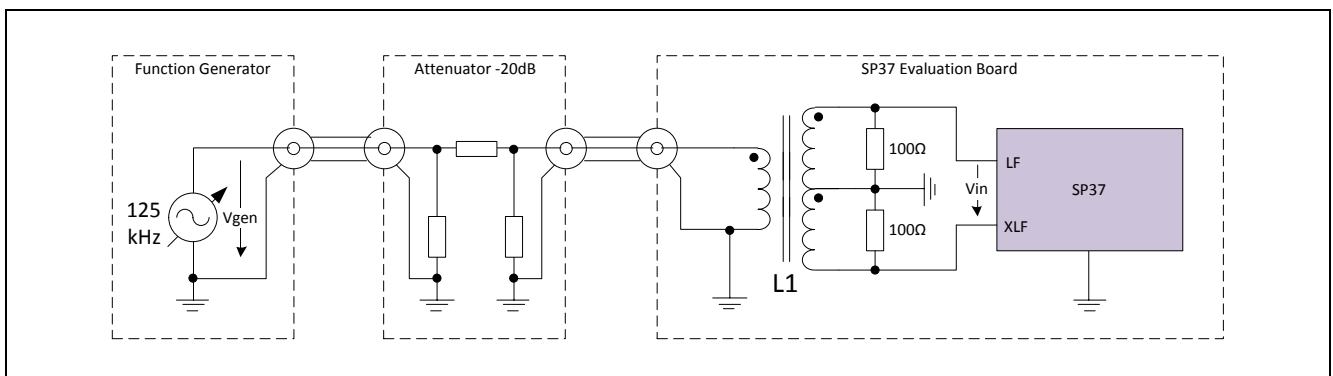


Figure 16 Measurement setup for LF sensitivity measurement

A complete measurement setup, however, requires that the function generator (or RF signal generator) is capable of some form of amplitude modulation (AM), and a modulating source. For simple LF Carrier Detection mode testing, the modulation source can be as simple as a pulse generator. For LF Telegram mode testing, however, an Arbitrary Waveform Generator (AWG) is a very good modulation source. One suitable AWG is an Agilent 33220A.

When applying the modulating signal to the generator's AM input, be sure that its coupling mode and amplitude are correct. For example, the Rhode & Schwarz SME and SMT signal generators must be configured for DC coupled, 100% AM modulation, the amplitude of the modulation source must be 1Vpk with no (0V) DC offset present. Note that when AM modulation is used, there is an additional factor of 2 now present in the amplitude, and so the voltage between the LF-xLF SP37 terminals becomes:

$$V_{in_{mVpp}} = 8\sqrt{2}V_{gen_{RMS}} \left(10^{\frac{-Attenu_{dB}}{20}} \right) \quad (5)$$

Some signal generators have a "Pulse Modulation" capability, which may be used instead of linear AM modulation. In the case of Pulse Modulation, however, care must be taken to ensure that the modulation source amplitude and offset are compatible with the pulse modulation input of the signal generator. Furthermore, even if a signal generator is capable of Pulse Modulation, it may not necessarily accurate over the entire operating frequency range of the signal generator. For example, the Rhode & Schwarz SME generator Pulse Modulation option is only specified for accuracy above 100 MHz. Using Pulse Modulation below 100 MHz with this particular signal generator is still possible, but the amplitude is no longer accurate and an additional loss factor must be empirically determined. For this reason, it is not recommended to use Pulse Modulation for 125 kHz LF testing unless it is absolutely clear that the equipment in question supports it with specified amplitude accuracy.

7 LF antenna design

In a typical application a ferrite coil is used for LF reception. The SP37 evaluation board can be configured for use with a ferrite coil antenna (see board documentation).

The requirements for the LF antenna are:

1. Should be selective to carrier frequency (125kHz)
2. Must be highly sensitive to carrier frequency
3. Must allow a certain data rate (3906 Baud)

Requirement 1 can be best met if the LF coil is part of a LC resonant circuit. Requirements 2 and 3 mean a tradeoff for the quality factor of the resonant circuit, since the higher the quality factor the higher the carrier sensitivity and the lower the possible baud rate. Hence, for adjusting the quality factor a resistor should be added to the resonant circuit. Figure 17 shows the resulting LF antenna circuit with and without parasitic elements.

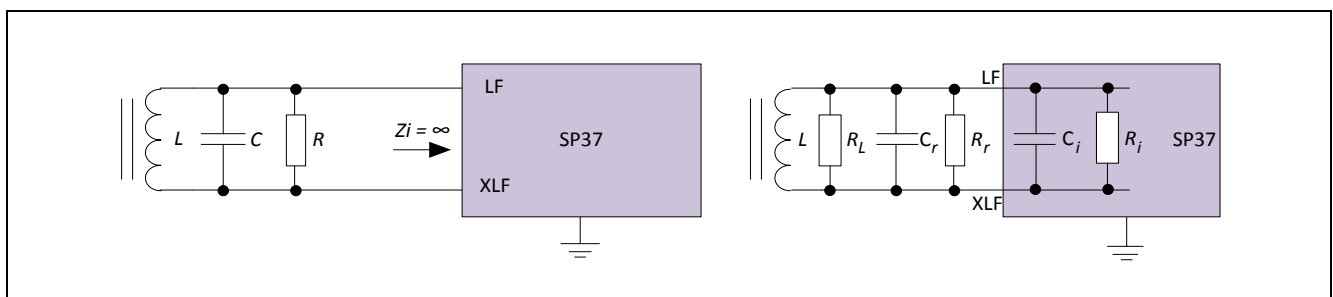


Figure 17 LF antenna circuit. Left: ideal circuit. Right: real circuit with parasitic elements

The resonant frequency f_0 of the resonant circuit is calculated as follows:

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \quad (6)$$

The maximum allowed quality factor Q_{max} in order to meet requirement 3 is given by

$$Q_{max} = \frac{f_c}{\Delta f_c + \Delta f_0 + 2BW_{signal}} \quad (7)$$

Where f_c is the carrier frequency, Δf_c the carrier frequency tolerance and Δf_0 the resonant frequency tolerance due to component tolerances. The contribution $2BW_{signal}$ is due to the amplitude modulation scheme used for LF data transmission. If only LF Carrier Detect is used, with relatively long duration LF carrier bursts, then the $2BW_{signal}$ term can be ignored.

From Q_{max} the parallel resistor R is calculated as:

$$R = Q_{max} 2\pi f_0 L \quad (8)$$

Example: For a ferrite coil like the one provided with the SP37 evaluation board the calculation is as follows:

- Coil = Coilcraft type 4513TC-715XGL, $L = 7.1\text{mH}$, $Q_{Coil}=51$
- Resonant frequency $f_0 = 125\text{ kHz}$

From equation (6) the capacitor C is calculated as $C = 228\text{ pF}$. Considering the input capacitance C_i of the SP37 of about 10pF the external capacitance C_r is **218 pF**.

Also from equation (6) the resonant frequency tolerance can be estimated. The application of the error propagation law to equation (6) yields

$$\Delta f_0 = \frac{1}{2} (\Delta L + \Delta C) \quad (9)$$

If a tolerance of $\pm 5\%$ is assumed for both, L and C , Δf_0 also becomes $\pm 5\%$ or $\Delta f_0 = 0.1 f_0$.

The signal bandwidth for a Manchester coded signal is given by the bit Baud rate, i.e. 3.906 kHz . Assuming zero tolerance of the carrier frequency ($\Delta f_c = 0$) the maximum allowed quality factor Q_{max} becomes:

$$Q_{max} = \frac{125\text{kHz}}{0 + 0.1 \times 125\text{kHz} + 2 \times 3.906\text{ kHz}} = 6.15 \quad (10)$$

From equation (8) the ideal parallel resistor R is calculated as:

$$R = 6.15 \times 2\pi \times 125\text{kHz} \times 7.1\text{mH} = 34.3\text{k}\Omega \quad (11)$$

The SP37 input impedance R_i at 125 kHz is typically $300\text{k}\Omega$. The coil equivalent parallel resistor R_L is calculated by using equation (8) as:

$$R_L = 51 \times 2\pi \times 125\text{kHz} \times 7.1\text{mH} = 284\text{k}\Omega \quad (12)$$

Hence the external resistor becomes:

$$R_r = \left(\frac{1}{34.3\text{k}\Omega} - \frac{1}{300\text{k}\Omega} - \frac{1}{284\text{k}\Omega} \right)^{-1} = 44.8\text{k}\Omega \quad (13)$$

8 Appendix: Complementary C definitions

The code examples 1 to 6 are not complete. In order to generate complete code that can be compiled the functions `main()` and `Data_Receive()` defined in the examples must be inserted in the code listing of Figure 18. The include-libraries are provided in the software package of the SP37 evaluation kit. It is recommended to start the Keil environment by opening one of the sample code projects of the software package and subsequently replace the code by the listings given in this application note.

```

/* ***** */
/* Example code for SP37 LF Application Note */
/* Autor MKA */
/* Environment: Keil C51 V9.03, µVision V4.14 */
/* Date of C code creation: September 2011 */
/* */
/* (C)copyright Infineon Technologies AG. All rights reserved. */
/* */
/* This SOFTWARE is Provided "AS IS" Without ANY WARRANTIES OF ANY KIND, WHETHER */
/* EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF */
/* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE or warranties of */
/* non-infringement of THIRD PARTY intellectual property rights. Infineon */
/* disclaims all liability regardless of the cause in law, in particular, but */
/* not limited to, the liability for indirect or consequential damages arising */
/* from interrupted operation, loss of profits, loss of information and data, */
/* unless in cases of gross negligence, intent, lack of assured characteristics */
/* or in any cases where liability is mandatory at law. */
/* ***** */

#include <stdio.h>
#include <ctype.h>
#include <absacc.h>
#include "Reg_SP37.h"
#include "SP37_ROMLibrary.h"
#include "SP37_DevLib.h"

#define TRUE 1
#define FALSE 0
#define MAX_LF_DATA 20 //Max LF Data that can be received

void main (void) //Replace this section by main function from
{ // listing 1, 2, 3, 4, 5 or 6
}

void Data_Receive() //Replace this section with Data_receive function
{ // from listing 6
}

```

Figure 18 Complementary C definitions

www.infineon.com