AN11608

Heart-rate Monitor using the Low Power LPC5410x

Rev. 1 — 24 February 2015 Application note

Document information

Info	Content
Keywords	LPC5410x, Heart-Rate Monitor, Cortex-M4F, Cortex-M0+, Low Power Modes, 12-bit ADC, Power Consumption, Code Example
Abstract	This application note introduces the methodology to implement a low power heart-rate monitor with the LPC54100 series. This includes information on the low power ADC, low power modes, and efficient use of the cores on the LPC54100 series. This application note also provides a software example of a heart rate monitor to illustrate the power savings possible by efficiently using the various low power features on the LPC45100 series.

Revision history

Rev	Date	Description
1	20150224	Initial version.

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

The LPC54100 series MCU are dual-core microcontrollers for embedded applications featuring multiple communication interfaces with very low power consumption in active and low power modes. Such applications include sensor hubs. The LPC54102 parts have a Cortex-M4F and a Cortex M0+ core included. The LPC54101 parts only have one Cortex-M4F core. Both cores can operate at frequencies of up to 100 MHz.

The LPC5410x includes up to 512 kB of flash memory and up to 104 kB of SRAM. The peripheral complement includes two SPI interfaces, four USARTs, three Fast-mode Plus I^2C Bus interfaces, a 32-bit, general-purpose State Configurable Timer plus an assortment of other timers, including a Real-Time Clock module with a dedicated oscillator, a 12-channel/12-bit, 4.8 MS/s ADC. A DMA controller can service most of the peripherals.

1.1 Low Power Methodology

A heart-rate monitor can be implemented in several ways due to the simplicity of the application. From a high level perspective, making a heart-rate monitor with an MCU like the LPC5410x would involve hardware that converts a human heart beat into a signal the LPC5410x can understand. From there, software would need to periodically retrieve data and compute the average beats per minute.

One attribute about heart-rate monitors is that because they are so simple, they can be implemented with very little impact on power consumption. With the simplicity of the software and relatively minimal hardware resources used, there are numerous approaches to optimize a heart-rate monitor for low power.

This application note will discuss the low power techniques used for applications involving periodic sampling, such as the duty-cycle approach. This application note will also focus on illustrating the low power features on the LPC5410x involved with the creation of a heart-rate monitor, such as the low power ADC and low power modes. The software included in this application note demonstrates the low power methodology discussed with a heart-rate monitor software example on the LPC54102 LPCXpresso board.

1.2 Duty-cycle Approach

When periodically sampling, it is possible to save power by implementing a duty-cycle approach of operation. This kind of operation puts the MCU into one of three states: listening state, sampling state, and the processing state. It is called the duty-cycle approach because the MCU's operation becomes split into periods. The MCU will spend a certain percentage of a given period performing work in the sampling or processing state and spend the rest of the period in the listening state, as shown in Figure 1.

To maintain flexibility in the frequency of these periods, it is also important to consider the overhead of waking up. The wake-up time of a particular low power mode dictates the maximum frequency of each period.

Fig 1. Duty-cycle approach diagram

2. LPC5410x Low Power ADC

The LPC54100 series feature a low power 12-bit successive approximation analog to digital converter. The ADC can sample at a variety of conversion speeds to comply with different application power requirements. The sampling rate can also be configured to synchronous mode which minimizes trigger latency by clocking the ADC with the MCU clock or asynchronous mode which maximizes the flexibility of the ADC's clock; that is, the core can run at a different frequency. Table 1 shows the average ADC power consumption on the LPC5410x at various samplings rates when the ADC is configured for operation but idle and when the ADC is converting. In both situations, the ADC is configured to be in synchronous mode.

Table 1. Average ADC Power Consumption While Converting (VDD = 3.3V, room temp, synchronous mode)

MCU Core Clock and ADC Sampling Rate	ADC Power Consumption While Configured and Idle	ADC Power Consumption While Converting
15 MHz core clock, 1 MS/s ADC sample rate	0.05 mA	0.23 mA
30 MHz core clock, 2 MS/s ADC sample rate	0.1 mA	0.42 mA
45 MHz core clock, 3 MS/s ADC sample rate	0.16 mA	0.62 mA
60 MHz core clock, 4 MS/s ADC sample rate	0.23 mA	0.76 mA
72 MHz core clock, 4.8 MS/s ADC sample rate	0.28 mA	1.0 mA

Figure 2 gives an illustrative view of how the power consumption scales with the sampling rate. As shown below, the power consumption of the ADC while converting scales nearly linearly with the ADC sample rate, permitting the use of higher sampling rates without the penalty of exponential growth in power consumption.

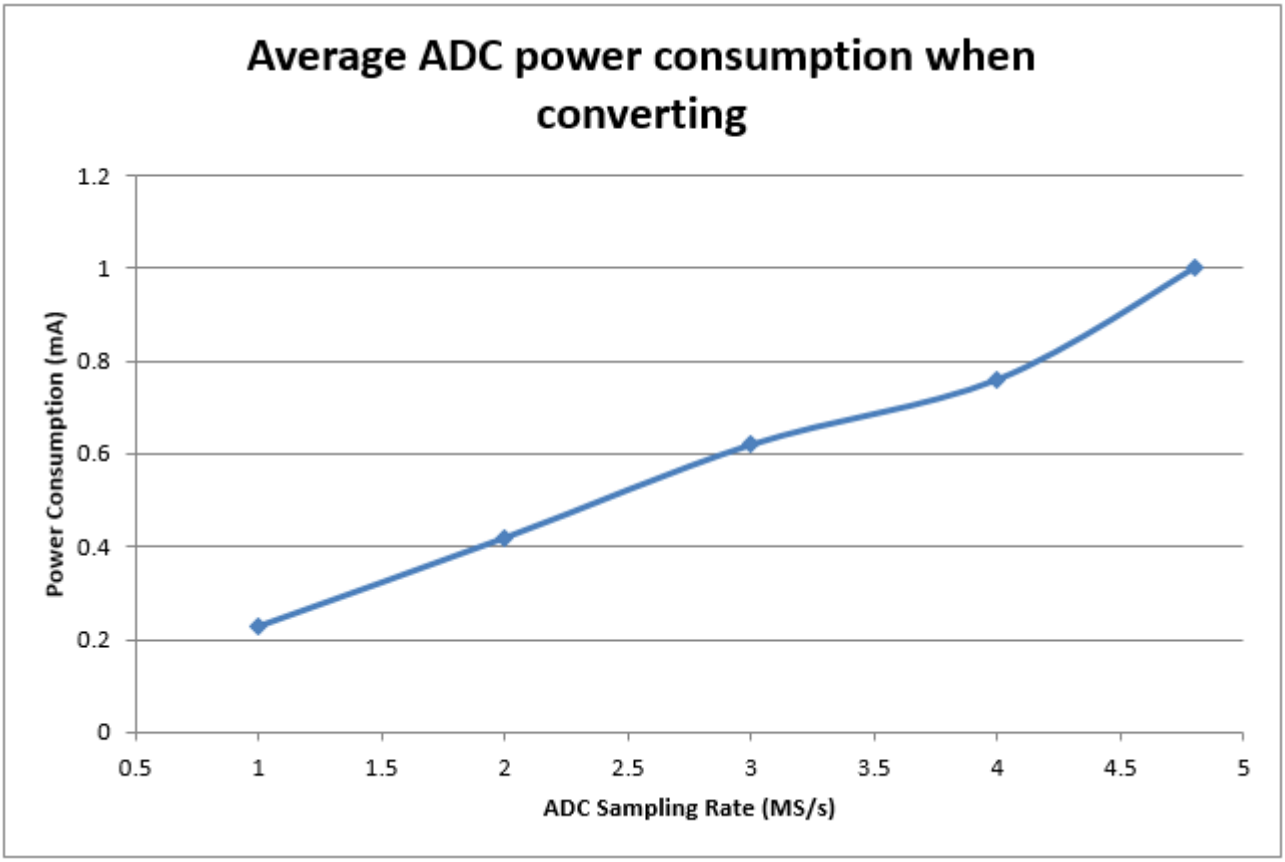

Fig 2. ADC sampling rate versus power consumption when converting graph

3. LPC5410x Power Modes Introduction

On the LPC5410x series, there are four reduced power modes: Sleep, Deep-Sleep, Power-down, and Deep Power-down modes. For more information on the low power modes, see AN11607:

http://www.nxp.com/documents/application_note/AN11611.zip

3.1 Typical Low Power Mode Power Consumption and Wake-up Times

Table 2 contains the current consumption and wake-up times measured on an LPCXpresso LPC54102 development board. From these measurements, Deep Power-down has the lowest power consumption, but also behaves as a reset once the MCU has woken up. This means that the program context will not be saved, making it unusable in this particular application. With that said, Power-down is the best low power mode for an application optimizing for power consumption. Sleep mode has a very fast wake-up time that makes it more flexible for applications that require a fast frequency for the duty-cycle appoach.

Table 2. Low Power Mode Power Consumption and Wake-up Time (VDD = 3.3V, room temp, peripherals off)

Low Power Mode	Power Consumption	Wake-up Time
Sleep	1.21 mA	1.6 µs
Deep-sleep	333 µA	17.9 µs
Power-down from flash	2.8 µA	69.5 µs
Power-down from SRAM	2.8 µA	18 µs
Deep Power-down	400 nA	N/A

3.1.1 Wakeup to SRAM from Power-down Mode

When a low power mode such as Power-down mode turns off the flash, the wake up routine will include the latency of turning the flash back on. It is possible to avoid this latency penalty by entering Power-down mode from SRAM and later wake-up to SRAM. This gives Power-down mode a comparable wake-up time to Deep-sleep while saving more power by turning off the flash.

4. Low Power Techniques

The LPC5410x series is capable of being tailored to reach low power levels depending on application requirements. For the purpose of this application note, the discussion of low power tweaks will be tailored toward a duty-cycle approach when using the ADC, which may be applicable for other applications.

4.1 The Cortex-M4F and Cortex-M0+

The LPC54102 is a dual-core MCU with Cortex-M4F and Cortex-M0+ cores. The Cortex-M4F is a fully featured core capable of executing DSP and floating point instructions at 100µA/MHz. The Cortex-M0+ core is a low power, efficient core suited for doing typical calculations capable of running at 55µA/MHz. To save as much power as possible, it is preferred to put the M4F core in a low power state and utilize the M0+ core for the application unless complex calculations need to be executed. Due to the inherent design of the MCU, Sleep mode must be the low power mode used for the unused core; the other three low power modes shut down the chip, causing both cores to enter the low power mode selected.

4.1.1.1 CPU Control Register (CPUCTRL)

The CPUCTRL register defines the roles of the Cortex-M4F and Cortex-M0+ cores, such as which core is the master and who controls the lower power modes.

When implementing an application with the Cortex-M0+ core and a low power mode other than Sleep mode, the POWERCPU bit of the CPUCTRL register will need to be cleared. Otherwise, any attempt in entering a low power that shuts down the chip to save power will instead put the Cortex-M0+ into Sleep mode.

The bit assignments are shown in Figure 3.

Table 81. CPU Control register (CPUCTRL, address 0x4000 0300) bit description

Bit	Symbol	Value	Description	Reset value
0	MASTERCPU		Indicates which CPU is considered the master. This is factory set assign the Cortex-M4 as the master. The master CPU cannot have its clock turned off via the related CMnCLKEN bit or be reset via the related CMxRSTEN in this register. The slave CPU wakes up briefly following device reset, then goes back to sleep until activated by the master CPU.	1
		0	M0+. Cortex-M0+ is the master CPU.	
		1	M4. Cortex-M4 is the master CPU.	
1	-	-	Reserved. Read value is undefined, only zero should be written.	-
2	CM4CLKEN		Cortex-M4 clock enable.	1
		0	Disabled. The Cortex-M4 clock is not enabled.	
		1	Enabled. The Cortex-M4 clock is enabled.	
3	CM0CLKEN		Cortex-M0+ clock enable.	1
		0	Disabled. The Cortex-M0+ clock is not enabled.	
		1	Enabled. The Cortex-M0+ clock is enabled.	
4	CM4RSTEN		Cortex-M4 reset.	0
		0	Disabled. The Cortex-M4 is not being reset.	
		1	Enabled. The Cortex-M4 is being reset.	
5	CM0RSTEN		Cortex-M0+ reset.	0
		0	Disabled. The Cortex-M0+ is not being reset.	
		1	Enabled. The Cortex-M0+ is being reset.	
6	POWERCPU		Identifies the owner of reduced power mode control: which CPU can cause the device to enter Deep Sleep, Power-down, and Deep Power-down modes.	1
		0	M0+. Cortex-M0+ is the owner of reduced power mode control.	
		1	M4. Cortex-M4 is the owner of reduced power mode control.	
14:7	-	-	Reserved. Read value is undefined, only zero should be written.	-
15	-	-	Must be written as a 1.	-
31:16	-	-	Must be written as 0xC0C4 for the write to have an effect.	-

Fig 3. LPC5410x CPU control register

4.2 Clock Frequency

For a low power application such as collecting ADC or sensor data, it may not be necessary to run the MCU at its fastest clock rate. For this reason, NXP recommends using the internal 12MHz IRC and dividers to clock the MCU down to minimize power consumption while still meeting application requirements. By clocking down the MCU, this can also clock the ADC lower while maintaining the benefits of synchronous mode, which can be desirable if the ADC does not require faster sampling than the MCU clock speed. For applications that require the ADC to run at full speed, asynchronous mode can be used to clock the ADC irrespective of the MCU clock. It is necessary to clock the MCU back to 12 MHz prior to entering Deep-sleep and Power-down mode to ensure correct operation.

4.3 Individually Clocked SRAM Banks

The LPC5410x has a total of 104KB of SRAM, which is split up into three SRAM blocks. Each of these three blocks of SRAM have a clock gating feature that gives better granularity of control over power consumption. Table 3 shows the three sections of SRAM that reside in the LPC5410x MCU.

Table 3. SRAM Blocks and Power Switches in LPC5410x

	SRAM0	SRAM1	SRAM2
Size	Up to 64 kB	Up to 32 kB	8 kB
Address range	Begins at 0x0200 0000	Begins at end of SRAM0	Begins at 0x0340 0000
Power Control (via Power API)	First 8kB has a separate switch	Entire SRAM1 has a single power switch	Entire SRAM2 has a single power switch

Depending on the program size of the application, you can turn on just the right amount of SRAM blocks to minimize wake-up time and power consumption. Table 4 shows typical power consumption measurements for different SRAM configurations in Power-down mode.

Table 4. Power-down Power Consumption With Different SRAM Configurations

Low Power Mode	RAM retention	Power Consumption
Power-down	First 8kB of SRAM0 (SRAM0A)	2.8 µA
	64kB (SRAM0A+SRAM0B)	5.2 µA
	96kB (SRAM0+SRAM1)	6.5 µA
	104kB (SRAM0+SRAM1+SRAM2)	6.9 µA

4.4 Analog block Shutdown

Certain analog blocks can remain powered while the MCU is in a low power state, such as the RTC. To save as much power as possible, disable any unused analog blocks when entering a low power mode and repower them upon wake-up. The code snippet in Figure 4 shows an example of setting the MCU to Power-down mode without de-powering SRAM1 and the 32 KHz oscillator for the RTC.

Note: When entering Power-down mode and depowering the ADC, a one-time calibration must be done via the LPCOpen API upon wake-up before the ADC can successfully sample again.

```
/* Use Chip_POWER_EnterPowerMode() to enter a low power mode. The
first argument is the desired low power mode, and the second argument
is the peripherals to not turn off. */

Chip_POWER_EnterPowerMode(POWER_POWER_DOWN, (SYSCON_PDRUNCFG_PD_SRAM1
| SYSCON_PDRUNCFG_PD_32K_OSC);
```

Fig 4. Code example (entering Power-down mode with select peripherals on)

5. Duty-cycle of Approach Implementation

The duty-cycle approach is best used for minimizing power consumption by putting the MCU in a listening state as often as possible. Section 3.1 shows that Power-down mode is the optimal low power mode because of its SRAM retention, low power draw, and flexibility to minimize wake-up time by waking up to SRAM. The MCU periodically exits this listening state to retrieve and process the data sampled from the ADC. It is important to minimize power while performing these tasks as sampling and processing data can be a light workload. The signal that periodically wakes up the MCU is asserted by the RTC.

Figure 5 gives an illustrative view on the program flow.

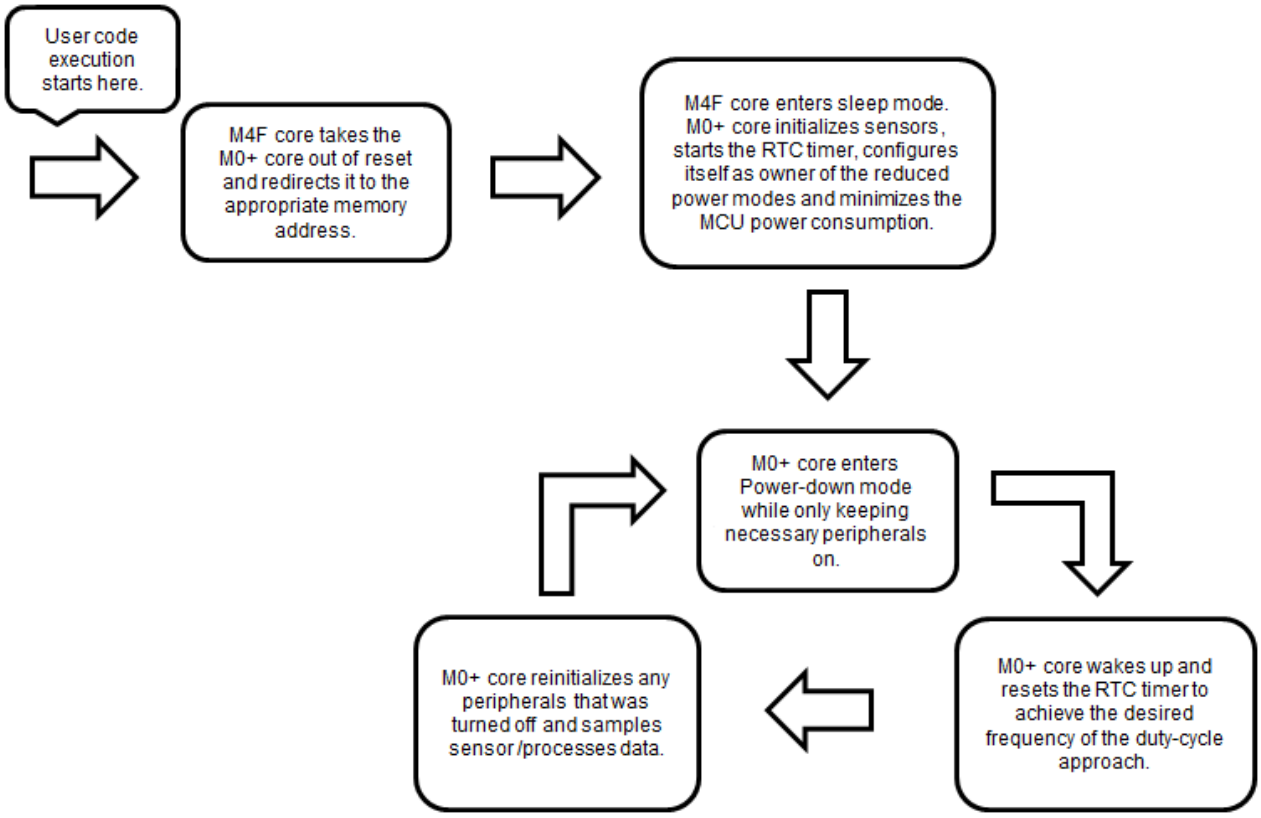

Fig 5. Duty-cycle approach program flow

6. Heart-Rate Monitor Demo

6.1 Objective

This application note provides a software example of a heart-rate monitor implemented with the duty-cycle approach. The software example uses a Pulse Sensor Amped! heart-rate sensor to convert a person's heart beat into an analog signal. This analog signal is then converted to a digital value through the LPC5410x low power ADC. The software is based on NXP's LPCOpen software platform and supports Keil MDK, IAR EWARM, and LPCXpresso.

To re-program the device, the user must put the device in ISP mode by power cycling or resetting the board while pushing down the ISP button. Thereafter, the device can be erased and programmed again. This is necessary because the MCU does not communicate with a debugger when it is in a low power state other than Sleep mode.

6.2 Requirements

1. Keil MDK, IAR EWARM or LPCXpresso.
2. Pulse Sensor Amped! heart-rate sensor (see Figure 6).
3. NXP LPCXpresso LPC54102 Board (see Figure 7).

6.3 Software and Hardware Setup

6.3.1 Pulse Sensor Amped! Hardware

The Pulse Sensor Amped! is a small device that converts heart rate into an analog signal. The pulse sensor has three color coded pins: analog output (purple), power (red), and ground (black). See Figure 6.

For more information on the Pulse Sensor Amped!, go to:

http://pulsesensor.myshopify.com/

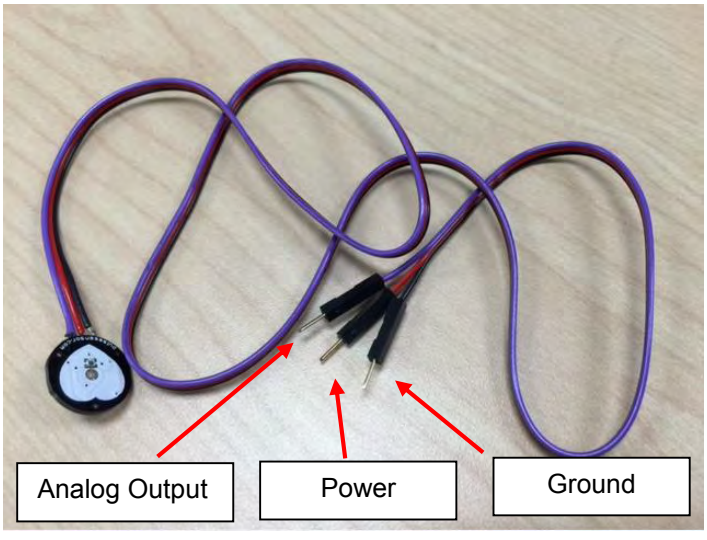

Fig 6. Pulse Sensor Amped! Heart-Rate Sensor

6.3.2 LPCXpresso LPC54102 Hardware

The OM13077 LPCXpresso LPC5102 development board was used in this software example. Figure 7 has various points of interest on the board, including the pin headers that the pulse sensor should be connected to.

For more information on the OM13077 LPCXpresso LPC5102 development board, visit the product page here:

http://www.nxp.com/demoboard/OM13077.html

Note: While the pulse sensor can be powered from a 5V source, use 3.3V as the power source because the heart rate algorithm is expecting voltage levels from a 3.3V power source.

There are two measurement approaches to measure the current consumed by the LPC54102. Remove JS6 and measurement LPC54102 core current directly at JP4 or measure the voltage across P2 and divide by 8.2 ohm.

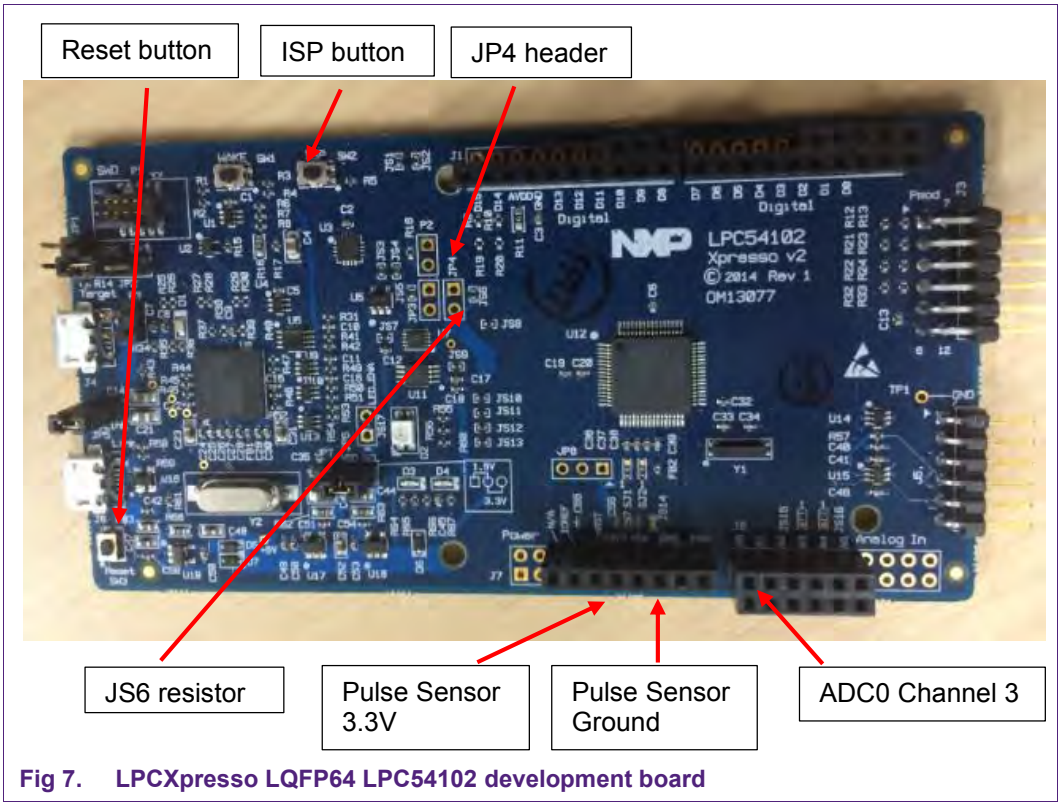

Fig 7. LPCXpresso LQFP64 LPC54102 development board

6.4 Heart-rate Monitor Software Example

The heart-rate sensor software example follows the program flow as shown in Figure 5. The MCU is clocked at 6 MHz with the ADC configured in synchronous mode at a sample rate of 1.5 KS/s. The Cortex-M0+ core is the core driving the application, leaving the Cortex-M4F core inactive in Sleep mode.

The average resting heart-rate can range from 60 – 100 beats per minute depending on the person, meaning one or two heart beats in any given one second interval. Ergo, the heart-rate sensing algorithm will be executed every half second to account for the intervals where there are two heartbeats a second. In order to save power, the ADC will be sampled 20 times a second with results to 10 data points each time the heart-rate sensing algorithm is executed.

1. Connect the pulse sensor to the pins shown in Figure 7.
2. Connect the USB cable to the J6 micro USB port to power-up the LPCXpresso V2 LPC54102 evaluation board. The green LED on the pulse sensor should light up.
3. Open a terminal emulator to the corresponding COM port with a baud rate of 115200-8-n-1-none.
4. Open the included Keil project file. Compile the chip and board libraries for both cores.
5. Compile and download the M4F and M0+ projects. Power cycle the board.
6. When attaching the pulse sensor to your finger or your ear, the MCU should print the detected beats per minute to the terminal emulator. See Figure 8.

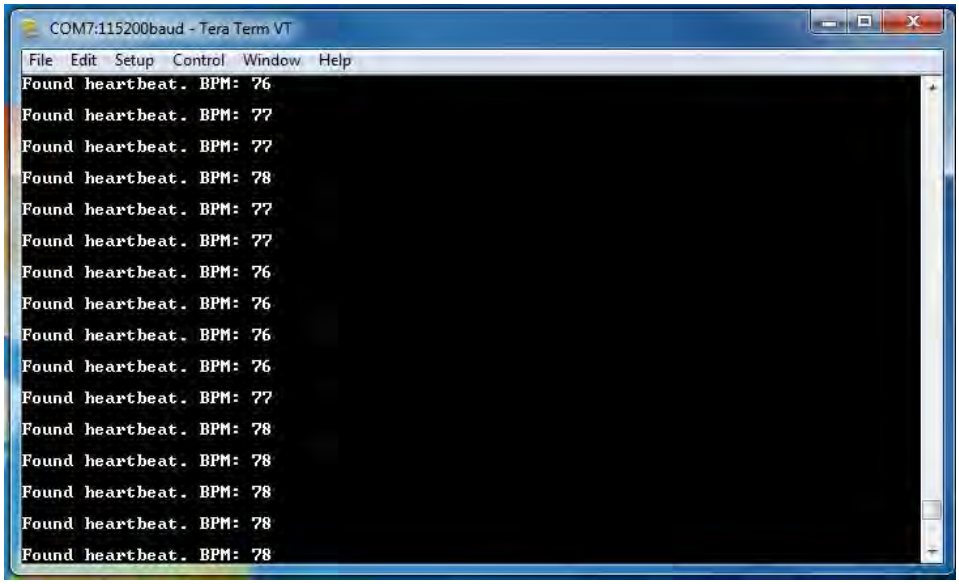

Fig 8. Pulse sensor software example output

The pulse sensor is a proximity sensor; therefore, slight movements between the pulse sensor and your body can alter the output of the pulse sensor and produce unrealistic BPM. NXP recommends using the ear clip provided with the pulse sensor to capture your heart-rate. Figure 9 contains a snapshot from an oscilloscope that is capturing the analog signal from the pulse sensor.

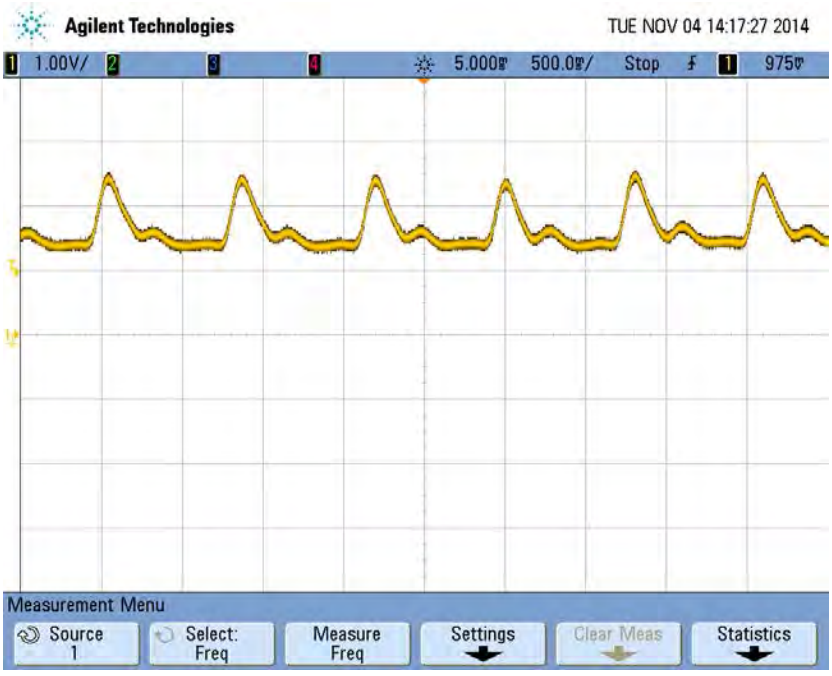

Fig 9. Analog output from the pulse sensor

A Monsoon Power Monitor was used to measure the current drawn on by the MCU. Table 5 shows the approximate current in the different states of the software example.

Table 5. Heart-rate sensor software example power consumption

	Listening State	Sampling State	Processing State	Application Current Consumption
Average Power Consumption	10 µA	1.6 mA	2.1 mA	70 µA

Figure 10 is a snapshot taken from a Monsoon generating the average current being drawn when running the software example.

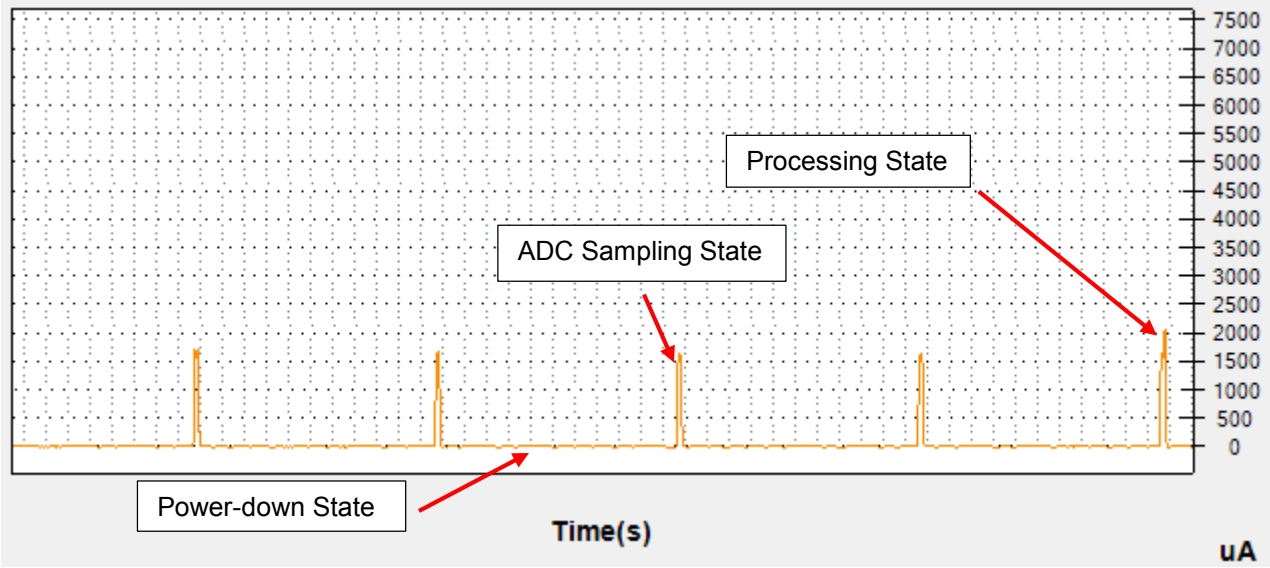

Fig 10. Average current consumed by the LPC54102 when executing the software example

7. Conclusion

The LPC5420x series MCU provides great flexibility and various options for users to customize the MCU to fit their low power needs. As shown in Table 2 and Table 3, the user has various low power mode options to choose from to achieve the desired power consumption by trading up or down the wake-up time of the device. In the duty-cycle approach, Power-down mode is the ideal low power mode because it consumes the least power while retaining SRAM contents. This flexibility coupled with other low power customization options, such as clocking down the MCU with clock dividers, allows for more granularity in controlling the trade-off of excess speed for power savings.

Table 6 compares various implementations of the heart-rate monitor software example to provide tangible effects on power consumption. The first implementation will be an unrefined implementation with no optimization to the core, frequency, or low power mode. Each following implementation will optimize one of these three parameters, showing how impactful each optimization is in terms of average current consumed and expected battery life under the assumption that the heart-rate monitor was powered by a 200 mAh battery.

Table 6. Current Consumption Comparison

Core Used	Frequency	Low Power Mode Used	Average Current Consumed	Expected Battery Life (200 mAh battery)
Cortex-M4F	100 MHz	None	11.6 mA	17.18 hours
Cortex-M0+	100 MHz	None	8.16 mA	24.52 hours
Cortex-M0+	6 MHz	None	1.71 mA	116.57 hours
Cortex-M0+	6 MHz	Sleep Mode	1.32 mA	151.83 hours
Cortex-M0+	6 MHz	Power-down Mode	70 µA	2954.6 hours

The first implementation is a simple active mode polling implementation similar to what was mentioned in the beginning of this application note. When utilizing the optimizations available on the LPC5410x, the expected battery life increased by a factor of 100. The most effective optimization was the correct use of the duty-cycle approach. Using Sleep mode as the listening state for an application using the Cortex-M0+ core at 6 MHz decreased the current consumed by a respectable 22.8%. However, by using Power-down for the listening state, this decrement in current consumption grows to 95.9%.

Given that the Cortex-M0+ core is more power efficient for something like sampling the ADC, the power saved increases as the duty-cycle increases. This is one of the several low power implementations available on the LPC5410x, which when used all together, can add up to a significant amount of power saved.

8. Legal information

8.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

8.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

9. List of figures

10. List of tables

11. Contents

For more information, visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 24 February 2015
Document identifier: AN11608