



APPLICATION NOTE

Duty Cycle and Power Optimization

Document	Type	Application Note
Document	Status	
Document	Version	Version 1.0
Product		Duty Cycle and Power Optimization

Table of Contents

1. Duty Cycle	4
1.1. General	4
1.2. Hardware description.....	4
1.3. Software description.....	5
Dynamic RSSI threshold level calculation.....	6
1.4. Duty cycle with AX5051	7
Scenario with 100kbit/s.....	8
Other scenarios	9
2. Conclusion	10
3. Appendix	11
3.1. References	11
AX5051 downloads.....	11
4. Contact Information	12

1. Duty Cycle

1.1. General

Consider a battery powered device (target), which should receive data from a second device (initiator) from time to time. To reduce power consumption, the target switches its receiver on for only a short while, checking if there is any RF activity, and returns to sleep if there is no data to receive.

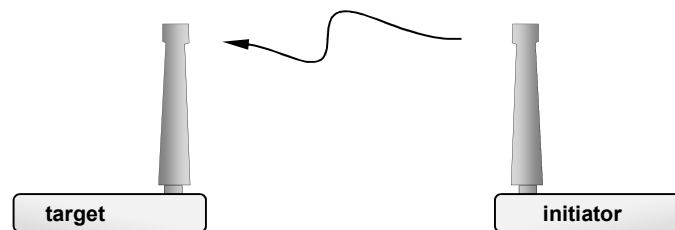


Figure 1 definition of initiator and target

The ratio of the time t_{on} during which the target is powered on, to the time t_{off} during which the target is powered off is called duty cycle. If for example the target is powered off for 1 second, powered on for 1 millisecond, powered off for 1 second and so on, its duty cycle is 1:1000.

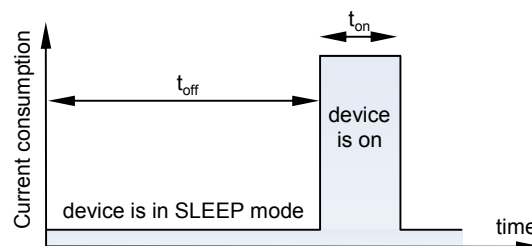


Figure 2 illustration of duty cycle

1.2. Hardware description

The hardware used in this document consists of an AX5051 evaluation board that is plugged into a suitable controller board, containing a PIC16F886 microcontroller from Microchip. The microcontroller is required to configure and control the AX5051 transmitter.

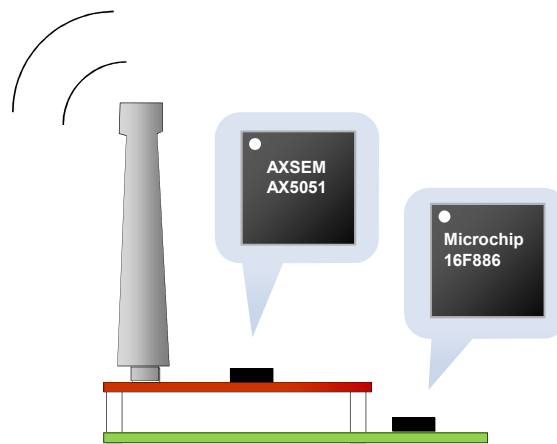


Figure 3 hardware used for duty cycle measurements

The PIC16F886 has a feature called 'Ultra low power wake up' (ULPWU). With this feature, an external capacitor is charged when the system is running, and discharged when the system is in sleep mode. When the condensator voltage reaches a defined level, the processor is woken up by an interrupt. This feature is very suitable for duty cycle applications.

1.3. Software description

The software demonstrates how duty cycle can be implemented in software. The software can be compiled for the transmitter or for the receiver. When defining `CONST_TX`, the source code will be compiled for the transmitter, if `CONST_RX` is defined, the source code will be compiled for the receiver. Make sure, only `CONST_TX` or `CONST_RX` is defined at one time. The transmitter sends a 1.2s long preamble every time a button is pressed. After the data is transmitted, it enters sleep mode and consumes only 650 nA. The receiver is in sleep mode too, but it wakes up every second. First the crystal oscillator needs to be switched on. This can take 1.7ms. After the crystal oscillator is running, the receiver is switched on and the program checks if the RSSI is above the defined threshold. If this is the case, 5 bytes of data are received, to check if there is a valid preamble present. If the transmitter is sending data at that time, the receiver will be switched on, until the data is received. This will take max. 1.2 seconds. After this the receiver is switched off, and the controller enters sleep mode again. See Figure 4 for an illustration of the flow during the wake up phase.

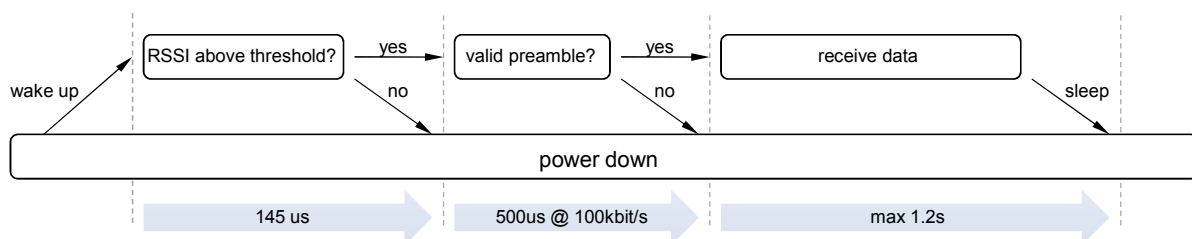


Figure 4 flow during duty cycle wake up

The software was compiled and tested with `sdcc`. To compile the code, please install the `gputils` (<http://gputils.sourceforge.net>), and modify the makefile according to your system settings. For downloading and debugging, the MPLAB IDE from Microchip was used together with the MPLAB ICD2 programmer/debugger.

Important

- Make sure, that when porting the software to another platform, calling `delay(x)` will keep the processor busy for `x` ms. The delay must be exact, otherwise some parts of the program may not work correct (e.g. starting the crystal oscillator).
- As mentioned above, `CONST_RX` and `CONST_TX` must be defined before compiling the code.
- The PIC16F886 has a hardware stack, so the number of function calls is limited to only a few levels. Because of this fact, the whole state machine is implemented in the main loop. Avoid calling functions within other functions whenever possible!

Dynamic RSSI threshold level calculation

There are several problems with a fixed RSSI threshold level. If it is set too high, valid data arriving with low levels will not wake up the target, even if it could have received the data correctly. Loss of sensitivity is the result. The other problem arises, when the RSSI level is set too low. In this case, the RSSI check will always be positive, thus the target will stay in RX mode longer than required. Higher current consumption is the result. There is a trade-off between sensitivity and current consumption that must be considered by the developer.

There is no RSSI threshold suitable for every situation. Consider a situation, where the target is placed in a very noisy environment. This will require a higher RSSI threshold than if the target is placed in an environment where the background noise is low. For targets, that change the location regularly (car key for instance) it is required, that the RSSI threshold is adapted dynamically. This can be done by averaging a certain number of RSSI values that were calculated in previous wake up phases.

1.4. Duty cycle with AX5051

All results in this document are obtained by using the AX5051 as an RF receiver, which is being switched on every second. At first the RSSI is checked against a predefined threshold to determine if there is any RF activity present. If not, the AX5051 is switched off immediately by the microcontroller. Otherwise 5 bytes have to be received to determine if a valid preamble is present. To wake up the receiver, the transmitting unit needs to transmit a preamble which is longer than the time t_{off} .

There are three stages used: Sleep, crystal startup and receive:

Sleep	Microcontroller and receiver are switched off for nearly one second. To wake up the microcontroller from sleep, a low power oscillator is required. The power consumption of the microcontroller in sleep mode was measured to be 0.65uA. The current consumption of the AX5051 in power down mode was measured to be 0.73uA.
Crystal startup	When the AX5051 comes out of power down mode, it takes some time until the crystal oscillator settles on its nominal frequency of 16MHz. The time until the crystal is fully settled, was measured to be 1.7ms. The current drawn by the controller was measured while it was running at full speed (8MHz).
Receive	After the crystal has settled, the receiver is switched on. At first the RSSI is calculated to see if it is above the defined threshold. If not, sleep mode is entered immediately. Otherwise 5 bytes are received, to check if a valid preamble is present. All diagrams in this document assume that no valid preamble is present (which is the normal case).

To provide a general impression, the currents drawn by the AX5051 and the microcontroller were measured separately. Note that the microcontroller's current consumption is device specific and must be evaluated individually.

Scenario with 100kbit/s

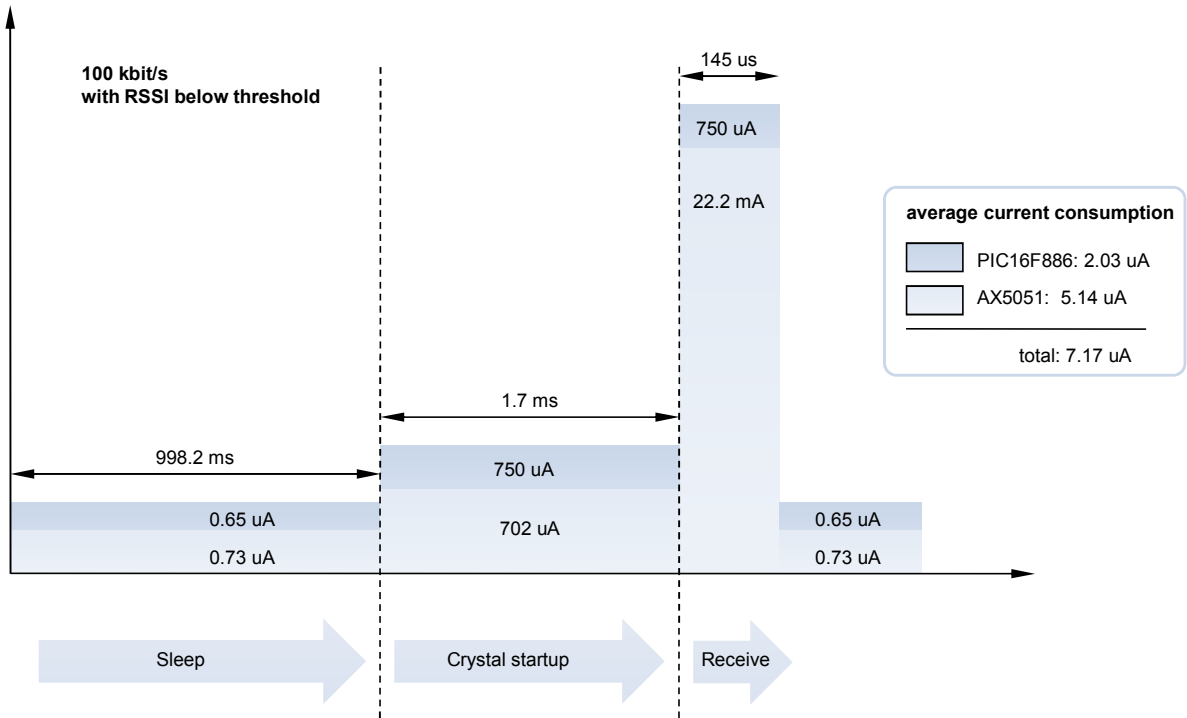


Figure 5 Current consumption with 100kbit/s, RSSI below threshold

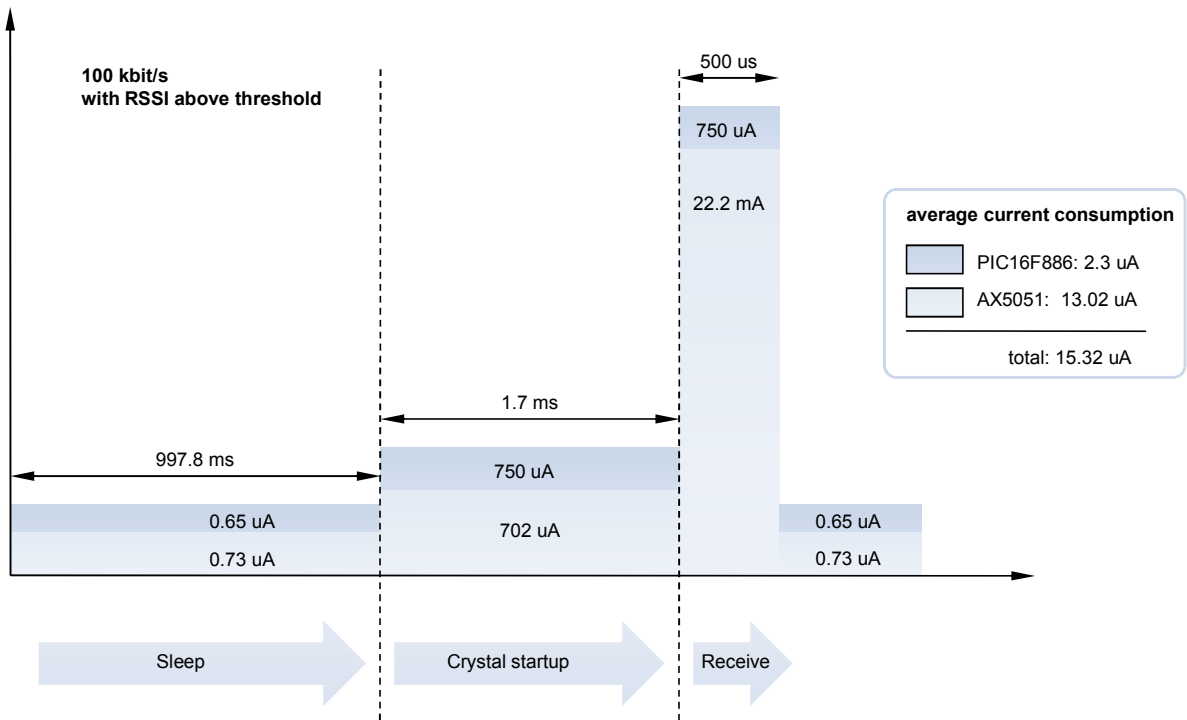


Figure 6 Current consumption with 100kbit/s, RSSI above threshold

Other scenarios

Table 1 compares the current consumption for different bitrates. Note that the current consumption for every bitrate choose is equal, if the RSSI is below the threshold. This is due to the fact that calculating the RSSI is independent of the bitrate.

bitrate [kbit/s]	AX5051 current [uA]		PIC16F886 current [uA]		total current [uA]	
	RSSI low	RSSI high	RSSI low	RSSI high	RSSI low	RSSI high
1.2 (FSK)	5.14	744.12	2.03	26.98	7.17	771.09
4.8(FSK)	5.14	189.14	2.03	8.24	7.17	197.38
9.6(FSK)	5.14	96.64	2.03	5.12	7.17	101.76
38.4(FSK)	5.14	27.27	2.03	2.78	7.17	30.05
100(FSK)	5.14	13.02	2.03	2.3	7.17	15.32
250(FSK)	5.14	7.69	2.03	2.12	7.17	9.81
400 (ASK)	5.14	6.36	2.03	2.07	7.17	8.43
600 (ASK)	5.14	5.62	2.03	2.05	7.17	7.67

Table 1 Average current consumption for different bitrates

2. Conclusion

The current used during receive mode is the main factor impacting the average current consumption (this is especially true for low bitrates). It is therefore very important to keep the time when the receiver is switched on, as short as possible. Even if the AX5051 does not have built in WOR (Wake On Radio) capabilities, the overall current consumption (including the microprocessors current) is low compared to competitors. This is achieved by using the great flexibility provided by the AX5051 that makes it possible to reduce the time, the receiver is switched on, to a minimum. With the flexibility of the AX5051, one is free to optimize for own requirements, and is not bound to a specific packet format.

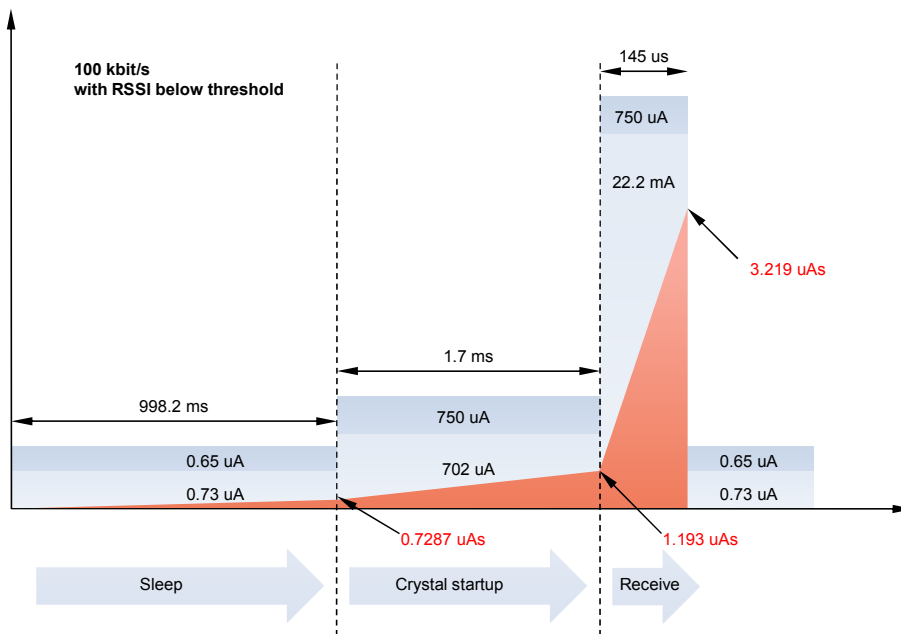


Figure 7 Distribution of current consumption during one cycle

3. Appendix

3.1. References

AX5051 downloads

Download AX5051 datasheet and programming manual from <http://www.axsem.com>

Download duty_cycle source code from <http://www.axsem.com>

AX5051 HARDWARE

Find local distributors for AX5051 transceiver at <http://www.axsem.com>

Order development kit AX5051-DVK at <http://www.axsem.com>

PIC16F886 MICROCONTROLLER

Datasheet for the microcontroller used

<http://ww1.microchip.com/downloads/en/DeviceDoc/41291E.pdf>

4. Contact Information

AXSEM AG

Oskar-Bider-Strasse 1
CH-8600 Dübendorf
SWITZERLAND

Phone +41 (44) 882 17 07

Fax +41 (44) 882 17 09

Email sales@axsem.com

www.axsem.com

For further product related or sales information please visit our website or contact your local representative.

The specifications in this document are subject to change at AXSEM's discretion. AXSEM assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of products based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights. AXSEM makes no warranties, either expressed or implied with respect to the information and specifications contained in this document. AXSEM does not support any applications in connection with life support and commercial aircraft. Performance characteristics listed in this document are estimates only and do not constitute a warranty or guarantee of product performance. The copying, distribution and utilization of this document as well as the communication of its contents to others without expressed authorization is prohibited. Offenders will be held liable for the payment of damages. All rights reserved. Copyright © 2007 AXSEM AG