

Author:		Sierra Wireless					Date:		January 10, 2012			
APN Content Level		BASIC		INTERMEDIATE	✓	ADVANCED		Confidentiality		Public	✓	Private
Hardware Compatibility		Product Line		AirPrime		Series		Q26xx		SL6087		
								WMPxx				
Software Compatibility		Series				OASIS 2.11 and above for Q26xx and WMPxx OASIS 2.34 and above for SL6087						



1 Version

Application Notes may be updated over their lifetime. To ensure you design with the correct version, please check the application notes page in www.sierrawireless.com for latest versions.

2 Introduction

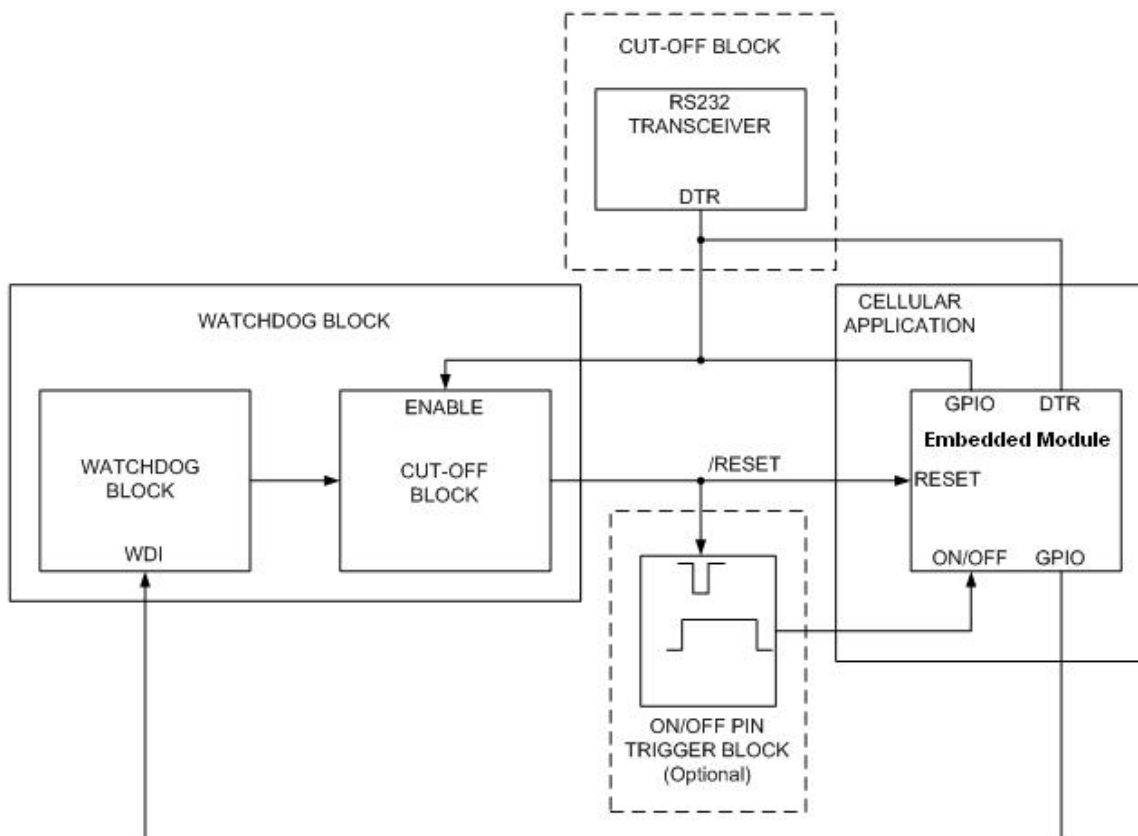
This APN (Application Note) is provided to Sierra Wireless distributors and clients to aid more rapid development of embedded applications using the Sierra Wireless portfolio of cellular solutions. To request a new application note, contact your regional Sierra Wireless Product Marketing Manager.

3 Application Description

Watchdog Timer Reset is designed for Remote products (when the product is located far away from the user and requires automatic operation) using the Application. It generates a hardware reset to the Intelligent Embedded Module in the event when the Open AT framework hangs or in a wrong state (blocked), allowing the SW to restart.

4 Hardware Description

This application "Watchdog Timer Reset" is composed of three blocks: Watchdog, Cut-off block and ON/~OFF pin trigger block (optional).



4.1 Watchdog Block

4.1.1 Description

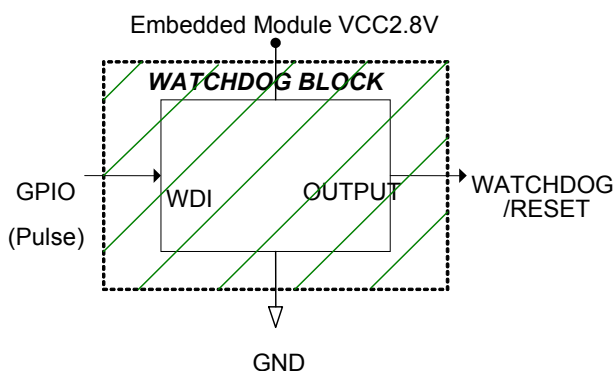
The watchdog block is a single chip which must receive a pulse at its input (generated by the embedded module controlled by the application).

When the pulse is received (meaning the application is running well), an internal counter is restarted, preventing to generate a reset to the Intelligent Embedded Module. This circuit must be supplied by VCC2.8V coming from the embedded module, without using external power supply and allowing watchdog to turn off at the same time as the embedded module does.

Parameters of the watchdog block:

- Watchdog reaction time (Timer): Minimum delay between two pulses to enable the reset.
- Pulse period: The pulse period has to stay always below the watchdog reaction time (Timer) to prevent unexpected reset of the embedded module. Depending on the embedded module operation and software priority task management, the pulse period can be affected by a jitter. It is recommended to set the pulse period at a maximum of one half of the timer delay.
- Power supply: The current drained from VCC2.8V line of the embedded module has to stay under the limit of 15mA maximum (from the embedded module specification).

4.1.2 Block Diagram



4.1.3 Recommendation

Depending on the watchdog reaction time (timer), different circuits can be proposed:

1. STM6321 from STMicroelectronics with a timer at 1.6s.
2. TPS3813L30 from TI with a timer adjustable to 20s.

In case of timer below 8s, the Watchdog reset must be disconnected from the embedded module reset pin during power on, and until the application software is ready to generate the pulses (prevent constant reset loop due to missing pulses during software initialisation).

For this reason, we highly recommend using the second solution (TI) with a 20s timer.

4.2 Cut-Off Block

4.2.1 Description

The Cut-off block allows disconnecting the embedded module reset from the watchdog reset output. This is necessary in case of downloading (firmware or Open AT Framework) or when the embedded module is in Sleep Mode.

Note: This disconnection is necessary for application using IDS (there could be a chance for no application running for more than 30s).

This disconnection is not necessary for DOTA Type I.

The Embedded Module reset must be driven by an open collector transistor.

Different signals can be used to disable the watchdog operation:

- DTR

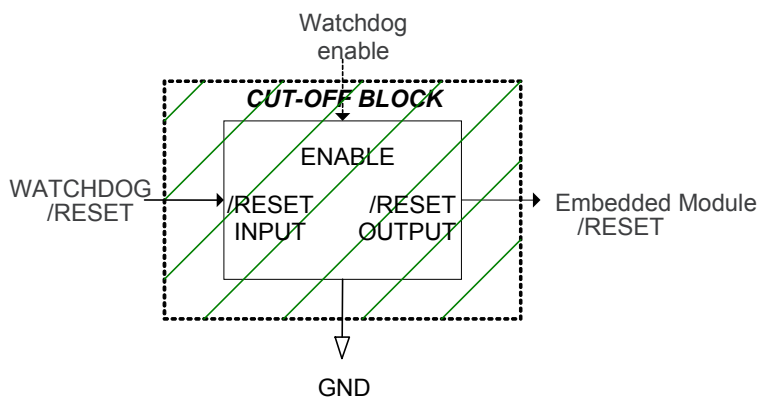
This signal is in low level when the RS232/UART is active (controlled by host) and especially during application download through the RS232/UART interface. Low level at DTR will disable the watchdog. A pull-up resistor will maintain the watchdog enabled when the RS232/UART is disconnected.

Note: If DTR is not used, the DTR Block (see section 11.2 Schematics) should be removed.

- GPIO from the embedded module
- This signal is used to control an open collector transistor. GPIO with reset state equal to “1” has to be selected. To disable the watchdog, the GPIO has to be of high level. During the firmware update by IDS or when the embedded module is in Sleep Mode or Boot Mode, the GPIO will resume to its reset state (i.e. high level). As a result, the watchdog is disabled.
- Other External signal

For illustrations, approaches using DTR and GPIO are shown in section 11.2 Schematics.

4.2.2 Block Diagram



4.2.3 Recommendation for Cut-Off Block

Different circuits are recommended depending on the security level and price.

1. A simple mechanical switch.
2. MUN5314DW1T1 (2 built-in resistor transistors: NPN & PNP)
3. NL27WZ126 (Buffer) & DTC144EE transistor.
4. NAND gate & DTC144EE transistor.

The MUN option (solution 2) was selected in this application note.

4.3 ON/OFF Trigger Block (Optional)

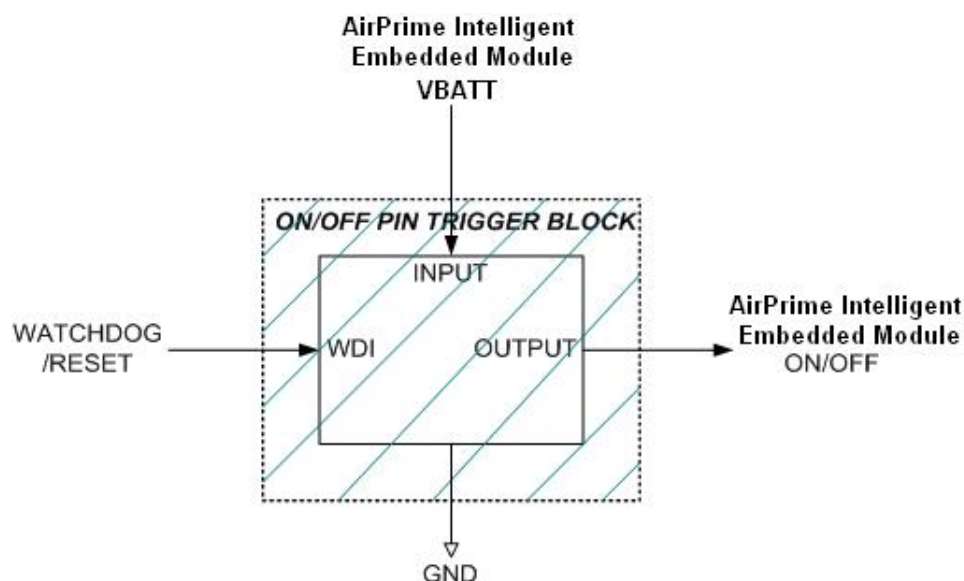
4.3.1 Description

For applications, particularly when alarm mode is needed, the ON/OFF pin would be in low state (OFF) most of the time after the initialization of the Embedded Module. If the watchdog reset occurs, the Embedded Module will be switched OFF.

This block is mandatory for applications with ON/OFF pin being set low most of the time. It triggers the ON/OFF pin to high level when watchdog reset occurs. Several conditions could trigger the ON/OFF pin when

- the main power jack is first plugged (For the Embedded Module VBATT)
- watchdog reset occurs.

4.3.2 Block Diagram



4.3.3 Recommendation for ON/OFF Pin Trigger Block

Below are the reference components for building up the block.

1. MUN5314DW1T1 transistor
2. ICM7555ID general purpose timer
3. Value of C3 is critical to handle the trigger when the power is plugged. It should be of 10 μ F min.
4. Value of R2 and C4 are critical in maintaining the timing of high level at ON/OFF pin. Recommended values for R2 and C4 are 1M Ω and 10 μ F respectively.

5 Software Description

The application contains functions which can be used within a customer application to provide watchdog functionality in collaboration with external hardware components.

6 Resource Usage

Flash Memory	RAM	MIPS
~80 kBytes for complete application	negligible (ignoring use by ADL library)	negligible

7 Custom AT Commands

(Custom AT commands given here are for testing only – in the actual application, users of the code should directly call the functions in the file “PeriodicPulse.c”, described in section 15 Package Deliverables).

7.1 AT+PPSTART – Start Periodic Pulse

This command starts a periodic pulse with the given properties. Values entered with this AT command will be stored in the flash until the **AT+PPSTOP** command is entered (or the flash is erased).

AT Command Syntax
AT+PPSTART=<period>,<width>,<pulsePin>,<activeLo>,<wdPin>
Description
Starts a periodic pulse with the specified properties on the given pin.

Parameters	Values	Description
<period>	1-9999	Period of the pulse; the value specifies the duration in multiples of 100 ms. For example, <period> = 20 means the period is 20 * 100 ms = 2 sec
<width>	1-9999	Width of the active portion of the pulse; the value specifies the duration in multiples of 100 ms. For example, <period> = 5 means the period is 5 * 100 ms = 500 ms Limitation: The value of the argument <width> must not exceed the value of the argument <period>
<pulsePin>	(depends on HW platform)	The number of the pin where the pulse is to be generated. Consult the PTS/CDG for the valid pin number to use. The pin must not be used by any other application.
<level>	0-1	1 for active-lo pulse (the pulse is default high, and goes to low for the duration specified by the <width> argument every period); 0 for active-hi pulse
<wdPin>	(depends on HW platform)	The number of the pin used to gate the watchdog signal. Consult the PTS/CDG for the valid pin numbers to use. The pin must not be used by any other application.

PossibleResponses

- +PPSTART: OK
- +PPSTART: <ErrNum>

AT Command Syntax

AT+PPSTART?

Description

Displays current IP properties of the periodic pulse

Parameters	Values	Description
(none)		

Possible Responses

- +PPSTART:
OK
- +PPSTART: <period>,<width>,<pulsePin>,<activeLo>,<wdPin>
OK

Note: The 1st response is shown if no periodic pulse is active. The 2nd response is shown if a periodic pulse is active.

AT Command Syntax

AT+PPSTART=?

Description

Shows range of input of the command

Parameters	Values	Description
(none)		

Possible Responses

+PPSTART: (1-9999),(1-9999),[pulsePin],[0-1],[watchdogPin]

Parameters in responses

Parameters	Values	Description
<ErrNum>	1	Redundant subscription (AT+PPSTART issued more than once in a row)
	2	GPIO subscription failure
	3	Watchdog-disable GPIO subscription failure
	4	Timer subscription failure

7.2 AT+PPSTOP – Stop Periodic Pulse

This command stops the periodic pulse started by the previous AT+PPSTART command.

AT Command Syntax

AT+PPSTOP

Description

Stops the periodic pulse

Parameters	Values	Description
(none)		

Possible Responses

- +PPSTOP: OK
- +PPSTOP: <ErrNum>

AT Command Syntax

AT+PPSTOP=?

Description

Shows if the current command is supported

Parameters	Values	Description
(none)		

Possible Responses

OK

Parameters in responses

Parameters	Values	Description
<ErrNum>	1	Redundant unsubscription (AT+PPSTART issued more than once in a row)
	2	GPIO unsubscription failure
	3	Watchdog-disable GPIO unsubscription failure
	4	Timer unsubscription failure

7.3 AT+PPMODE – Sets (Special) Mode for the Periodic Pulse

This command allows controlling of the periodic pulse that was specified in the AT+PPSTART command

AT Command Syntax		
AT+PPMODE=<mode>		
Description		
Sets the periodic pulse into the given mode		
Parameters	Values	Description
<mode>	0-2	0: generates the periodic pulse if one has been started using AT+PPSTART 1: Holds the GPIO pin(s) at the active level 2: Holds the GPIO pin(s) at the inactive level
Possible Responses		
<ul style="list-style-type: none"> +PPMODE: OK +PPMODE: <ErrNum> 		
AT Command Syntax		
AT+PPMODE?		
Description		
Shows the current PPMODE setting		
Parameters	Values	Description
(none)		
Possible Responses		
<ul style="list-style-type: none"> +PPMODE: <mode> OK 		
AT Command Syntax		
AT+PPMODE=?		
Description		
Shows if the current command is supported		

Parameters	Values	Description
(none)		
Possible Responses		
<ul style="list-style-type: none"> +PPMODE: (0-2) OK 		

Parameters in responses

Parameters	Values	Description
<ErrNum>	1	Pulse not subscribed yet

7.4 AT+WDDISABLE – Controls the Gate for the Periodic Pulse

This command allows controlling the GPIO pin that is used to gate the periodic pulse. This function can be used, for example, during DOTA update, when the firmware is unable to generate any periodic pulse for 30 seconds.

If the periodic pulse needs to be gated during reset (as in the DOTA case mentioned above, for example), the GPIO pin chosen for this purpose should have a reset value of “1”.

AT Command Syntax		
AT+WDDISABLE=<mode>		
Description		
Change the gate setting of the periodic pulse		
Parameters	Values	Description
<mode>	0-1	0: do not gate the periodic pulse 1: gate the periodic pulse
Possible Responses		
OK		
AT Command Syntax		
AT+ WDDISABLE?		
Description		
Shows the current WDDISABLE setting		
Parameters	Values	Description
(none)		
Possible Responses		
<ul style="list-style-type: none"> + WDDISABLE: <mode> OK 		
AT Command Syntax		
AT+ WDDISABLE =?		

Description		
Shows if the current command is supported		
Parameters	Values	Description
(none)		
Possible Responses		
<ul style="list-style-type: none"> +WDDISABLE: (0-1) OK 		

7.5 +PPINIT – Unsolicited Response at Startup

This unsolicited response appears at startup time to indicate the status of the periodic pulse application

AT Command Syntax		
+PPINIT: <status>		
Description		
Change the gate setting of the periodic pulse		
Parameters	Values	Description
<status>		0 if pulse is started successfully 1 for flash subscription failure 2 for flash read problem (not necessarily an error – could be due to empty flash) 9x for initialization failure; see the mapping for “x” in the function “PeriodicPulseSubscribe” (the following are indications – not failures) 10 for empty flash indication 11 for pulse not started indication (because the previous setting stops the pulse)

8 Examples for AT Commands

Action	Response	Action
(On AirPrime WMP100) AT+PPSTART= 50,10,19,1,5	OK	Starts a periodic pulse with period 5 sec, rest state high and going low for 1 sec every period, on GPIO-19; and uses GPIO-5 to gate the pulse
(On AirPrime WMP100) AT+PPSTART= 100,100,19,1,5	+PPSTART: 1	(Redundant subscription)
AT+PPMODE=2	OK	Holds the pin at high level
AT+PPSTOP	OK	Stops the previous pulse

9 Functional Interface

The following functions can be called from the program which uses the AT library:

9.1 PeriodicPulseInit() Function – Initialization of Periodic Pulse

This function can be called during the startup of an Open AT program. It checks from the flash if a periodic pulse needs to be started, and if so, the values for the parameters, and then tries to start the periodic pulse.

Function Syntax

PeriodicPulseInit(u8 PP_traceLevel)

Description

(see above)

Arguments	Values	Description
-----------	--------	-------------

PP_traceLevel	1-32	Trace level in TMT for the functions in the library
---------------	------	---

Possible Return Values

Same as the status for "+PPINIT" unsolited response

9.2 PeriodicPulseSubscribe – Start Periodic Pulse

This function starts a periodic pulse with the given properties. The function parameters will be stored in the flash until the PeriodicPulseUnsubscribe() function is issued (or the flash is erased)

Function Syntax

PeriodicPulseSubscribe(u16 period, u16 width, u32 pulsePin, bool activeLo, u32 watchdogPin, u8 traceLevel)

Description

(See above)

Arguments	Values	Description
-----------	--------	-------------

period	1-9999	See the description of the <period> parameter in AT+PPSTART
width	1-9999	See the description of the <width> parameter in AT+PPSTART
pulsePin	u32	See the description of the <pulsePin> parameter in AT+PPSTART
activeLo	bool	See the description of the <level> parameter in AT+PPSTART
watchdogPin	u32	See the description of the <watchdogPin> parameter in AT+PPSTART
traceLevel	u8	Trace level in TMT for the functions in the library

Possible Return Values

- OK for success
- 1 for redundant subscription (function called more than once in a row)
- 2 for pulse GPIO subscription failure
- 3 for watchdog GPIO subscription failure
- 4 for timer subscription failure

9.3 PeriodicPulseUnsubscribe() – Stop Periodic Pulse

This function stops the periodic pulse started by the previous call of the PeriodicPulseSubscribe() function.

Function Syntax

PeriodicPulseUnsubscribe(void)

Description

(see above)

Arguments	Values	Description
(none)		
Possible Return Values		
<ul style="list-style-type: none"> OK for success 1 for redundant unsubscription (function called more than once in a row) 2 for pulse GPIO unsubscription failure 3 for watchdog GPIO unsubscription failure 4 for timer unsubscription failure 		

9.4 PeriodicPulseForce() - Sets (Special) Mode for the Periodic Pulse

This function allows controlling the GPIO pin(s) that is used in the PeriodicPulseSubscribe() function.

Function Syntax		
PeriodicPulseForce(u8 mode)		
Description		
(see above)		
Arguments	Values	Description
<mode>	0-2	See the description of the <mode> parameter in AT+PPMODE
Possible Return Values		
(none)		

9.5 WatchdogDisable() - Sets Gating for the Periodic Pulse

This function allows controlling of the gating the GPIO pin for the periodic pulse.

Function Syntax		
WatchdogDisable(bool state)		
Description		
(see above)		
Arguments	Values	Description
<state>	bool	FALSE: do not gate the periodic pulse TRUE: gate the periodic pulse
Possible Return Values		
(none)		

9.6 AT_PP_Init() Function

This command subscribes the custom AT commands described above. Furthermore, it calls the PeriodicPulseInit() function to do its work. Therefore, this function should also be used at the startup of the Open AT program.

The different between this function and the PeriodicPulseInit() function lies in the subscription of custom AT commands.

This function will also report the result of the PeriodicPulseInit() function through unsolicited response

Function Syntax		
AT_PP_Init(u8 traceLevel)		
Description		
(See above)		
Arguments	Values	Description
traceLevel	1-32	Trace level in TMT for the functions in the library
Possible Return Values		
(See the description of the return values for the PeriodicPulseInit() function)		

10 Examples for Functions

Action	Return Value	Action
(On AirPrime WMP100) PeriodicPulseSubscribe (50, 10, 19, 1, 5, 2);	OK	Starts a periodic pulse with period 5 sec, rest state high and going low for 1 sec every period, on GPIO-19. GPIO-5 is used to gate the watchdog pulse. TMT trace level-2 is used for debug information
(On AirPrime WMP100) PeriodicPulseSubscribe (100, 50, 19, 1, 5, 2);	1	Redundant subscription
PeriodicPulseForce(2);	(n/a)	Holds the pin at high level
PeriodicPulseUnsubscribe()	OK	Stops the previous pulse

11 Application Example

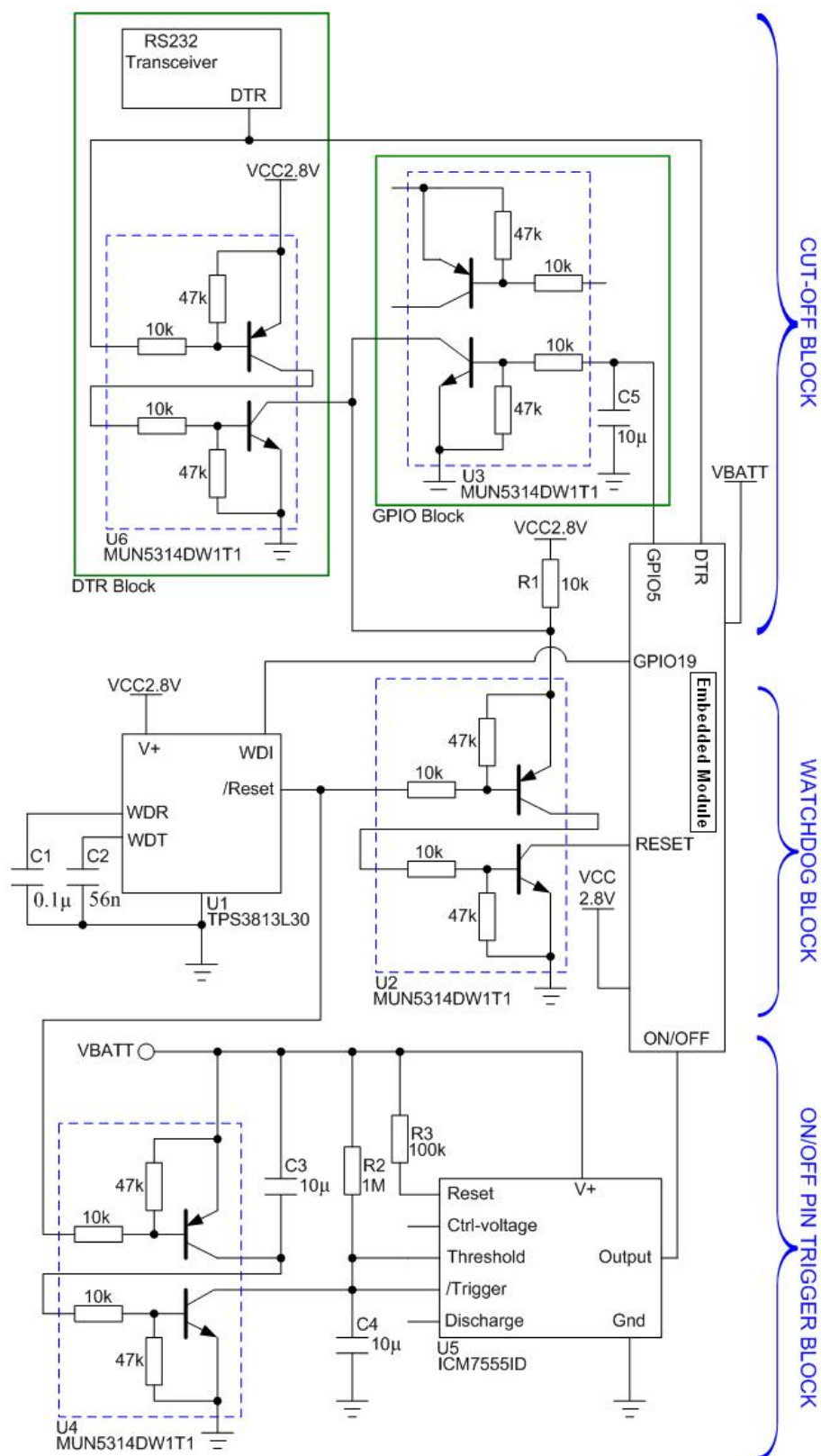
11.1 Description

Watchdog and cut-off circuits chosen: TPS3813L30 from TI (Watchdog) and MUN5314DW1T1 from ONsemi (Cut-off).

Watchdog Circuit (TPS3813L30) recommendations:

- The choice of the capacitors (C1 and C2) are important, to reach timer expected accuracy in the operating temperature range (-30°C to +85°C for the design proposed in this example).
- The reset output is an open collector, a pull-up resistor is necessary (provided by R1 in this example).

11.2 Schematics



12 Bill of Materials

Reference	Description	Supplier
U1	TPS3813L30DBV	TI
U2, U3, U4, U6	MUN5314DW1T1	ONSEMI
U5	ICM7555ID	NXP
R1	Resistor 10kΩ 5% 1/16W 0402	ROHM
R2	Resistor 1M Ω 5% 1/16W 0402	ROHM
R3	Resistor 100kΩ 5% 1/16W 0402	ROHM
C1	Ceramic Capacitor 0.1μF 16V Y5V +80/-20%	AVX
C2	Ceramic Capacitor 56nF (low ESR, low Tolerance)	AVX
C3, C4, C5	Ceramic Capacitor 10 μF 6.3V X5R +/- 10%	AVX

13 Software Settings

Action	Response	Action
AT+PPSTART= 50,10,19,1,5	+PPSTART :OK	Sets the Periodic Pulse.
AT+PPMODE=1	+PPMODE:OK	Hold the GPIO active high.

These two AT command allow you to set and test the watchdog reset:

- After the first command, a positive pulse is generated on GPIO19 every 5 sec for 1 sec pulse duration. This command must be entered at first start only (the embedded module memory will save this setting).
- The second command will test if the watchdog reset is functional. The GPIO19 will be hold (high state) until a reset appears (for testing the feature). The hold on GPIO19 will disappear automatically after the reset (the software come back to normal operation).

14 Testing

Specific information can be provided upon request.

15 Package Deliverables

This application note is delivered as a single compressed zip archive as follows:

Filename	Description
Application Note for Watchdog Timer Reset - Rev4.0.zip	Application note, source files and header files for the library

15.1 Software

Filename	Description
src\appli.c src\PeriodicPulse.c src\AT_PeriodicPulse.c	OAT Source files
inc\PeriodicPulse.h inc\Test_WDReset.h inc\AT_PeriodicPulse.h	Header files

15.2 Documentation

Filename	Description
Application Note for Watchdog Timer Reset – Rev4.0.pdf	Watchdog Timer Reset Application Note

16 Support

For direct clients: contact your Sierra Wireless FAE

For distributor clients: contact your distributor FAE

For distributors: contact your Sierra Wireless FAE

17 Document History

Level	Date	History
001	April 20, 2009	Creation
002	August 20, 2009	Updated
003	March 05, 2010	Reformatted to the rebranded SWI template.
4.0	January 10, 2012	Added SL6087. Reformatted to new document template.

18 Legal Notice

Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

Safety and Hazards

Do not operate the Sierra Wireless modem in areas where blasting is in progress, where explosive atmospheres may be present, near medical equipment, near life support equipment, or any equipment which may be susceptible to any form of radio interference. In such areas, the Sierra Wireless modem **MUST BE POWERED OFF**. The Sierra Wireless modem can transmit signals that could interfere with this equipment. Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

Limitations of Liability

This manual is provided "as is". Sierra Wireless makes no warranties of any kind, either expressed or implied, including any implied warranties of merchantability, fitness for a particular purpose, or noninfringement. The recipient of the manual shall endorse all risks arising from its use.

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Patents

This product may contain technology developed by or for Sierra Wireless Inc.

This product includes technology licensed from QUALCOMM®.

This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from InterDigital Group.

Copyright

© 2012 Sierra Wireless. All rights reserved.

Trademarks

AirCard® is a registered trademark of Sierra Wireless. Sierra Wireless™, AirPrime™, AirLink™, AirVantage™, Watcher™ and the Sierra Wireless logo are trademarks of Sierra Wireless.

       are filed or registered trademarks of Sierra Wireless S.A. in France and/or in other countries.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh and Mac OS are registered trademarks of Apple Inc., registered in the U.S. and other countries.

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of the respective owners.



SIERRA
WIRELESS