Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Data Sheet: JN516x

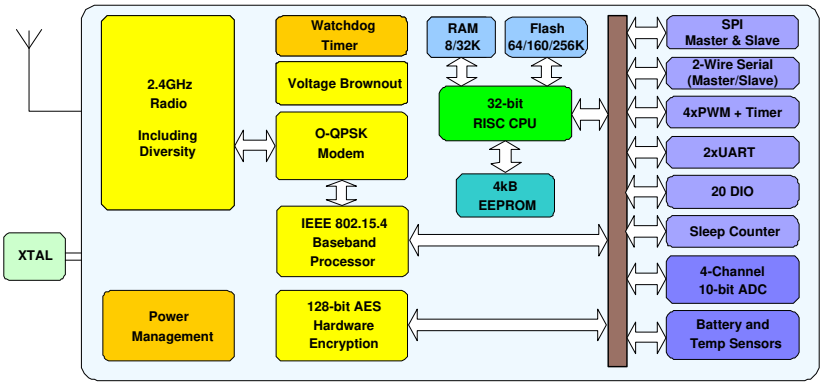
IEEE802.15.4 Wireless Microcontroller

Overview

The JN516x series is a range of ultra low power, high performance wireless microcontrollers supporting JenNet-IP, ZigBee PRO or RF4CE networking stacks to facilitate the development of Home Automation, Smart Energy, Light Link and Remote control applications. They feature an enhanced 32-bit RISC processor with embedded Flash and EEPROM memory, offering high coding efficiency through variable width instructions, a multi-stage instruction pipeline and low power operation with programmable clock speeds. They also include a 2.4GHz IEEE802.15.4 compliant transceiver and a comprehensive mix of analogue and digital peripherals. Three memory configurations are available to suit different applications. The best in class operating current of 15mA, with a 0.6uA sleep timer mode, gives excellent battery life allowing operation direct from a coin cell.

The peripherals support a wide range of applications. They include a 2-wire I^2C, and SPI ports which can operate as either master or slave, a four channel ADC with battery and a temperature sensor. It can support a large switch matrix of up to 100 elements, or alternatively a 20 key capacitive touch pad.

Block Diagram



Benefits

- Single chip device to run stack and application
- Very low current solution for long battery life – over 10 yrs
- Supports multiple network stacks
- Highly featured 32-bit RISC CPU for high performance and low power
- System BOM is low in component count and cost
- Flexible sensor interfacing options

Applications

- Robust and secure low power wireless applications
- RF4CE Remote Controls
- JenNet-IP networks
- ZigBee SE networks
- ZigBee Light Link networks
- Lighting & Home automation
- Toys and gaming peripherals
- Smart Energy
- Energy harvesting, for example self powered light switch

Features: Radio

- 2.4GHz IEEE802.15.4 compliant
- 128-bit AES security processor
- MAC accelerator with packet formatting, CRCs, address check, auto-acks, timers
- Integrated ultra low power sleep oscillator – 0.6µA
- 2.0V to 3.6V battery operation
- Deep sleep current 0.12µA (Wake-up from IO)
- <$0.15 external component cost
- RX current 17mA , TX 15mA
- Receiver sensitivity -95dBm
- Transmit power 2.5dBm
- Time of Flight engine for ranging
- Antenna Diversity (Auto RX)

Features: Microcontroller

- 32-bit RISC CPU, 1 to 32MHz clock speed
- Variable instruction width for high coding efficiency
- Multi-stage instruction pipeline
- JN5161: 64kB/8kB/4kB
- JN5164: 160kB/32kB/4kB
- JN5168: 256kB/32kB/4kB (Flash/RAM/EEPROM)
- Data EEPROM with guaranteed 100k write operations.
- RF4CE, JenNet-IP, ZigBee SE and ZigBee Light Link stacks
- 2-wire I2C compatible serial interface. Can operate as either master or slave
- 5xPWM (4x timer & 1 timer/counter)
- 2 low power sleep counters
- 2x UART
- SPI Master & Slave port, 3 selects
- Supply voltage monitor with 8 programmable thresholds
- 4-input 10-bit ADC, comparator
- Battery and temperature sensors
- Watchdog & Brown Out Reset
- Up to 20 Digital IO Pins (DIO)
- Infra-red remote control transmitter

Temp range (-40°C to +125°C)
6x6mm 40-lead
Lead-free and RoHS compliant

Contents

1 Introduction

The JN516x is an IEEE802.15.4 wireless microcontroller that provides a fully integrated solution for applications using the IEEE802.15.4 standard in the 2.4 - 2.5GHz ISM frequency band [1], including Zigbee PRO, ZigBee Smart Energy, ZigBee LightLink, RF4CE and JenNet-IP. There are 3 versions in the range, differing only by memory configuration

JN5161-001: 64kB Flash, 8kB RAM, 4 kB EEPROM, suitable for IEEE802.15.4 and RF4CE applications

JN5164-001: 160kB Flash, 32kB RAM, 4 kB EEPROM suitable for Jennet-IP, IEEE802.15.4 and RF4CE applications

JN5168-001: 256kB Flash, 32kB RAM, 4 kB EEPROM suitable for all applications

Applications that transfer data wirelessly tend to be more complex than wired ones. Wireless protocols make stringent demands on frequencies, data formats, timing of data transfers, security and other issues. Application development must consider the requirements of the wireless network in addition to the product functionality and user interfaces. To minimise this complexity, NXP provides a series of software libraries and interfaces that control the transceiver and peripherals of the JN516x. These libraries and interfaces remove the need for the developer to understand wireless protocols and greatly simplifies the programming complexities of power modes, interrupts and hardware functionality.

In view of the above, it is not necessary to provide the register details of the JN516x in the datasheet.

The device includes a Wireless Transceiver, RISC CPU, on chip memory and an extensive range of peripherals.

1.1 Wireless Transceiver

The Wireless Transceiver comprises a 2.45GHz radio, a modem, a baseband controller and a security coprocessor. In addition, the radio also provides an output to control transmit-receive switching of external devices such as power amplifiers allowing applications that require increased transmit power to be realised very easily. Appendix B.4, describes a complete reference design including Printed Circuit Board (PCB) design and Bill Of Materials (BOM).

The security coprocessor provides hardware-based 128-bit AES-CCM* modes as specified by the IEEE802.15.4 2006 standard. Specifically this includes encryption and authentication covered by the MIC –32/-64/-128, ENC and ENC-MIC –32/-64/-128 modes of operation.

The transceiver elements (radio, modem and baseband) work together to provide IEEE802.15.4 (2006) MAC and PHY functionality under the control of a protocol stack. Applications incorporating IEEE802.15.4 functionality can be developed rapidly by combining user-developed application software with a protocol stack library.

1.2 RISC CPU and Memory

A 32-bit RISC CPU allows software to be run on-chip, its processing power being shared between the IEEE802.15.4 MAC protocol, other higher layer protocols and the user application. The JN516x has a unified memory architecture, code memory, data memory, peripheral devices and I/O ports are organised within the same linear address space. The device contains up to 256kbytes of Flash, up to 32kbytes of RAM and 4kbytes EEPROM .

1.3 Peripherals

The following peripherals are available on chip:

- Master SPI port with three select outputs
- Slave SPI port
- Two UART's, one capable of hardware flow control (4-wire, includes RTS/CTS), and the other just 2-wire (RX/TX)
- One programmable Timer/Counter which supports Pulse Width Modulation (PWM) and capture/compare, plus four PWM timers which support PWM and Timer modes only.
- Two programmable Sleep Timers and a Tick Timer
- Two-wire serial interface (compatible with SMbus and I^2C) supporting master and slave operation
- Twenty digital I/O lines (multiplexed with peripherals such as timers, SPI and UARTs)
- Two digital outputs (multiplexed with SPI port)
- 10-bit, Analogue to Digital converter with up to four input channels. Autonomous multi-channel sampling
- Programmable analogue comparator
- Internal temperature sensor and battery monitor
- Two low power pulse counters
- Random number generator
- Watchdog Timer and Supply Voltage Monitor
- JTAG hardware debug port
- Infra-red remote control transmitter, supported by one of the PWM timers
- Transmit and receive antenna diversity with automatic receive switching based on received energy detection
- Time of Flight engine for ranging

User applications access the peripherals using the Integrated Peripherals API. This allows applications to use a tested and easily understood view of the peripherals allowing rapid system development.

1.4 Block Diagram – JN516x

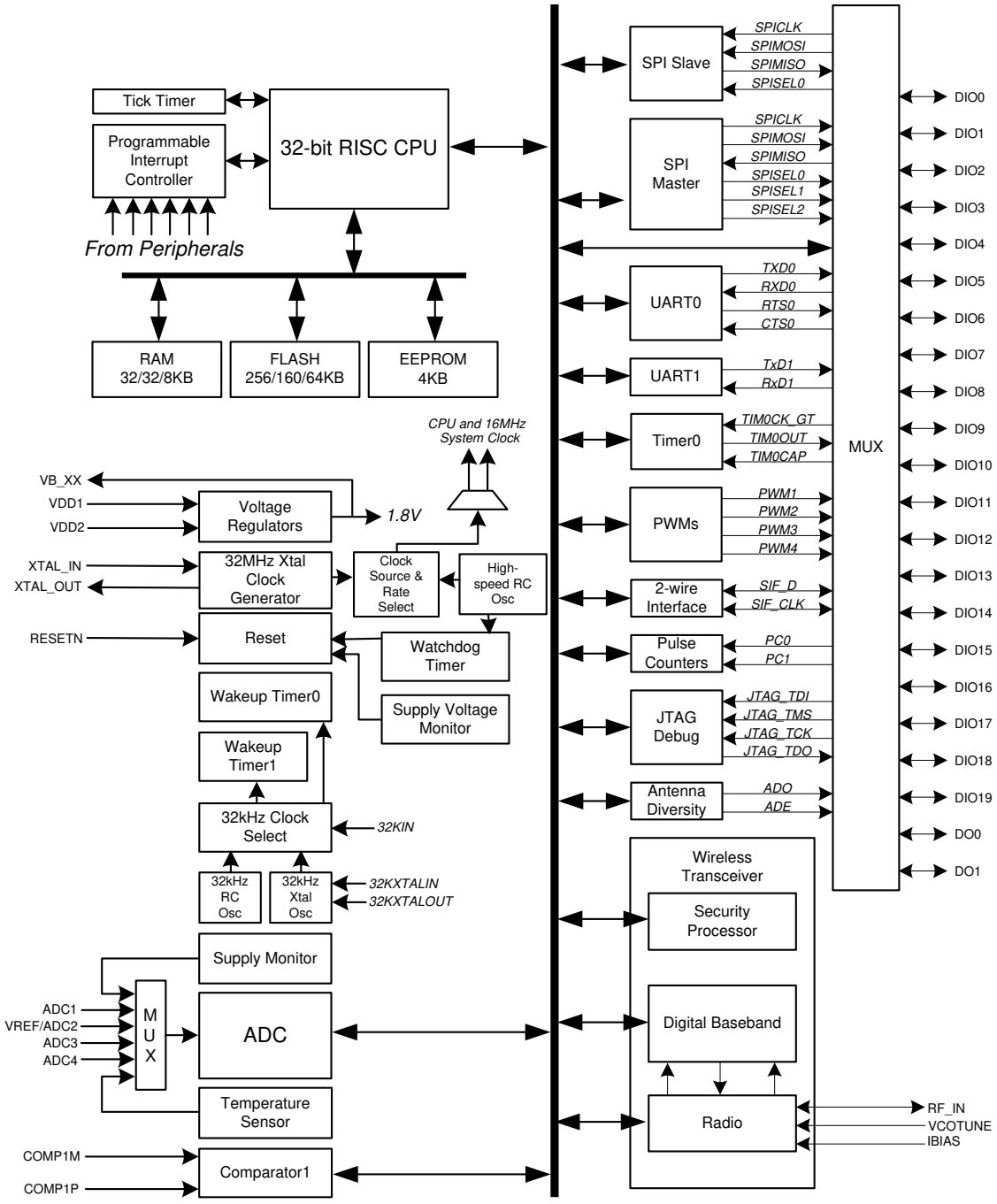


Figure 1: JN516x Block Diagram

2 Pin Configurations

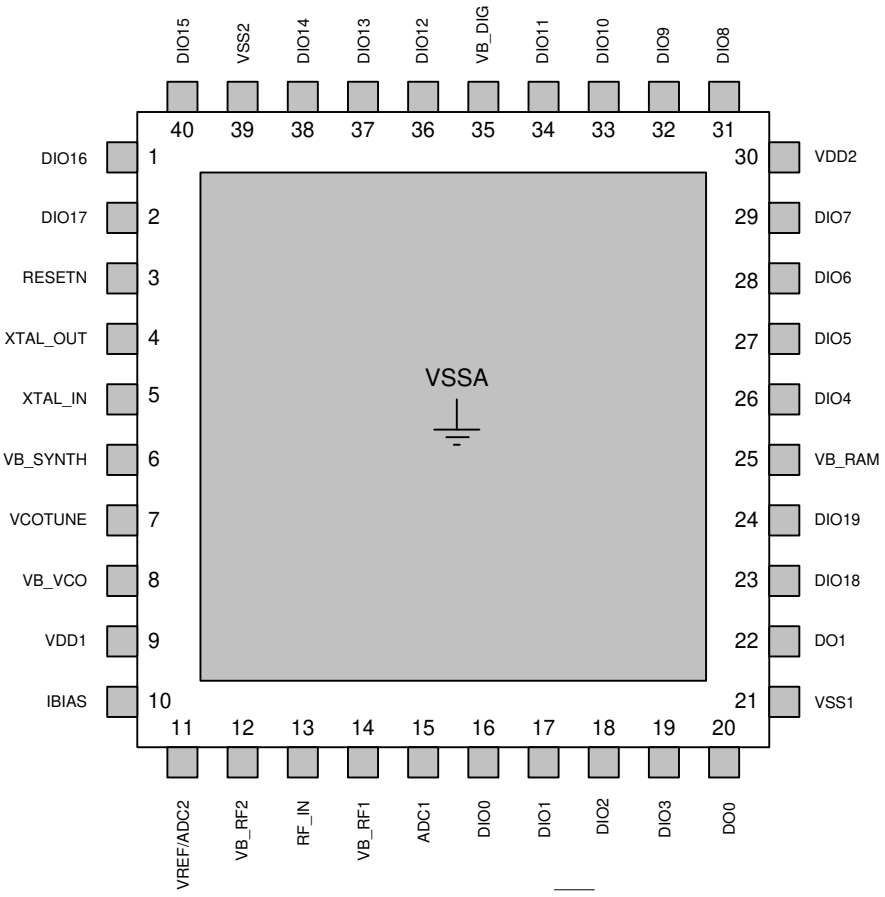


Figure 2: 40-pin QFN Configuration (top view)

Note: Please refer to Appendix B.4 JN516x Module Reference Design for important applications information regarding the connection of the PADDLE to the PCB.

2.1 Pin Assignment

Pin No	Power supplies						Signal Type	Description
6, 8, 12, 14, 25, 35	VB_SYNTH, VB_VCO, VB_RF2, VB_RF1, VB_RAM, VB_DIG						1.8V	Regulated supply voltage
9, 30	VDD1, VDD2						3.3V	Supplies: VDD1 for analogue, VDD2 for digital
21, 39, Paddle	VSS1, VSS2, VSSA						0V	Grounds (see appendix A.2 for paddle details)
	General							
3	RESETN						CMOS	Reset input
4,5	XTAL_OUT, XTAL_IN						1.8V	System crystal oscillator
	Radio							
7	VCOTUNE						1.8V	VCO tuning RC network
10	IBIAS						1.8V	Bias current control
13	RF_IN						1.8V	RF antenna
	Analogue Peripheral I/O							
15, 16, 17	ADC1, DIO0 (ADC3), DIO1 (ADC4)						3.3V	ADC inputs
11	VREF/ADC2						1.8V	Analogue peripheral reference voltage or ADC input 2
1, 2	DIO16 (COMP1P), DIO17 (COMP1M)						3.3V	Comparator inputs
	Digital Peripheral I/O							
	Primary	Alternate Functions						
16	DIO0	SPISEL1	ADC3				CMOS	DIO0, SPI Master Select Output 1 or ADC input 3
17	DIO1	SPISEL2	ADC4	PC0			CMOS	DIO1, SPI Master Select Output 2, ADC input 4 or Pulse Counter 0 Input
18	DIO2		RFRX	TIM0CK_GT			CMOS	DIO2, Radio Receive Control Output or Timer0 Clock/Gate Input
19	DIO3		RFTX	TIM0CAP			CMOS	DIO3, Radio Transmit Control Output or Timer0 Capture Input
26	DIO4	CTS0	JTAG_TCK	TIM0OUT	PC0		CMOS	DIO4, UART 0 Clear To Send Input, JTAG CLK Input, Timer0 PWM Output, or Pulse Counter 0 input
27	DIO5	RTS0	JTAG_TMS	PWM1	PC1		CMOS	DIO5, UART 0 Request To Send Output, JTAG Mode Select Input, PWM1 Output or Pulse Counter 1 Input
28	DIO6	TXD0	JTAG_TDO	PWM2			CMOS	DIO6, UART 0 Transmit Data Output, JTAG Data Output or PWM2 Output
29	DIO7	RXD0	JTAG_TDI	PWM3			CMOS	DIO7, UART 0 Receive Data Input, JTAG Data Input or PWM 3 Output
31	DIO8	TIM0CK_GT	PC1	PWM4			CMOS	DIO8, Timer0 Clock/Gate Input, Pulse Counter1 Input or PWM 4 Output

32	DIO9	TIM0CAP	32KXTALIN	RXD1	32KIN		CMOS	DIO9, Timer0 Capture Input, 32K External Crystal Input, UART 1 Receive Data Input or 32K external clock Input
33	DIO10	TIM0OUT	32KXTALOUT				CMOS	DIO10, Timer0 PWM Output or 32K External Crystal Output
34	DIO11	PWM1		TXD1			CMOS	DIO11, PWM1 Output or UART 1 Transmit Data Output
36	DIO12	PWM2	CTS0	JTAG_TCK	ADO	SPISMOSI	CMOS	DIO12, PWM2 Output, UART 0 Clear To Send Input, JTAG CLK Input, Antenna Diversity Odd Output or SPI Slave Master Out Slave In Input
37	DIO13	PWM3	RTS0	JTAG_TMS	ADE	SPISMISO	CMOS	DIO13, PWM3 Output, UART 0 Request To Send Output, JTAG Mode Select Input, Antenna Diversity Even output or SPI Slave Master In Slave Out Output
38	DIO14	SIF_CLK	TXD0 TXD1	JTAG_TDO	SPISEL1	SPISSEL	CMOS	DIO14, Serial Interface Clock, UART 0 Transmit Data Output, UART 1 Transmit Data Output, JTAG Data Output, SPI Master Select Output 1 or SPI Slave Select Input
40	DIO15	SIF_D	RXD0 RXD1	JTAG_TDI	SPISEL2	SPISCLK	CMOS	DIO15, Serial Interface Data, UART 0 Receive Data Input, UART 1 Receive Data Input, JTAG Data Input, SPI Master Select Output 2 or SPI Slave Clock Input
1	DIO16	COMP1P	SIF_CLK	SPISMOSI			CMOS	DIO16, Comparator Positive Input, Serial Interface clock or SPI Slave Master Out Slave In Input
2	DIO17	COMP1M	SIF_D	SPISMISO			CMOS	DIO17, Comparator Negative Input, Serial Interface Data or SPI Slave Master In Slave Out Output
23	DIO18	SPIMOSI					CMOS	SPI Master Out Slave In Output
24	DIO19	SPISEL0					CMOS	SPI Master Select Output 0
20	DO0	SPICLK			PWM2		CMOS	SPI Master Clock Output or PWM2 Output
22	DO1	SPIMISO			PWM3		CMOS	SPI Master In Slave Out Input or PWM3 Output

The PCB schematic and layout rules detailed in Appendix B.4 must be followed. Failure to do so will likely result in the JN516x failing to meet the performance specification detailed herein and worst case may result in device not functioning in the end application.

2.2 Pin Descriptions

2.2.1 Power Supplies

The device is powered from the VDD1 and VDD2 pins, each being decoupled with a 100nF ceramic capacitor. VDD1 is the power supply to the analogue circuitry; it should be decoupled to ground. VDD2 is the power supply for the digital circuitry; and should also be decoupled to ground. In addition, a common 10µF tantalum capacitor is required for low frequencies. Decoupling pins for the internal 1.8V regulators are provided which each require a100nF capacitor located as close to the device as practical. VB_SYNTH, VB_RAM and VB_DIG require only a 100nF capacitor. VB_RF and VB_RF2 should be connected together as close to the device as practical, and require one 100nF capacitor and one 47pF capacitor. The pin VB_VCO requires a 10nF capacitor. Refer to B.4.1 for schematic diagram.

VSSA (paddle), VSS1, VSS2 are the ground pins.

Users are strongly discouraged from connecting their own circuits to the 1.8v regulated supply pins, as the regulators have been optimised to supply only enough current for the internal circuits.

2.2.2 Reset

RESETN is an active low reset input pin that is connected to a 500kΩ internal pull-up resistor. It may be pulled low by an external circuit. Refer to Section 6.2 for more details.

2.2.3 32MHz Oscillator

A crystal is connected between XTAL_IN and XTAL_OUT to form the reference oscillator, which drives the system clock. A capacitor to analogue ground is required on each of these pins. Refer to Section 5.1 for more details. The 32MHz reference frequency is divided down to 16MHz and this is used as the system clock throughout the device.

2.2.4 Radio

The radio is a single ended design, requiring a capacitor and just two inductors to match to 50Ω microstrip line to the RF_IN pin.

An external resistor (43kΩ) is required between IBIAS and analogue ground (paddle) to set various bias currents and references within the radio.

2.2.5 Analogue Peripherals

The ADC requires a reference voltage to use as part of its operation. It can use either an internal reference voltage or an external reference connected to VREF. This voltage is referenced to analogue ground and the performance of the analogue peripherals is dependent on the quality of this reference.

There are four ADC inputs and a pair of comparator inputs. ADC1 has a designated input pin but ADC2 uses the same pin as VREF, invalidating its use as an ADC pin when an external reference voltage is required. The remaining 2 ADC channels are shared with the digital I/Os DIO0 and DIO1 and connect to pins 16 and 17. When these two ADC channels are selected, the corresponding DIOs must be configured as Inputs with their pull-ups disabled. Similarly, the comparator shares pins 1 and 2 with DIO16 and DIO17, so when the comparator is selected these pins must be configured as Inputs with their pull-ups disabled. The analogue I/O pins on the JN516x can have signals applied up to 0.3v higher than VDD1. A schematic view of the analogue I/O cell is shown in Figure 3. Figure 4 demonstrates a special case, where a digital I/O pin doubles as an input to analogue devices. This applies to ADC3, ADC4, COMP1P and COMP1M.

In reset, sleep and deep sleep, the analogue peripherals are all off. In sleep, the comparator may optionally be used as a wakeup source.

Unused ADC and comparator inputs should not be left unconnected, for example connected to analogue ground.

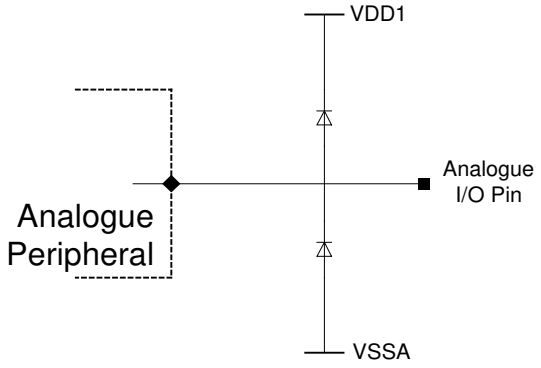


Figure 3: Analogue I/O Cell

2.2.6 Digital Input/Output

For the DC properties of these pins see Section 19.2.3.

When used in their primary function all Digital Input/Output pins are bi-directional and are connected to weak internal pull up resistors (50kΩ nominal) that can be disabled. When used in their secondary function (selected when the appropriate peripheral block is enabled through software library calls), their direction is fixed by the function. The pull up resistor is enabled or disabled independently of the function and direction; the default state from reset is enabled.

A schematic view of the digital I/O cell is in Figure 4. The dotted lines through resistor R_{ESD} represent a path that exists only on DIO0, DIO1, DIO16 and DIO17 which are also inputs to the ADC (ADC3, ADC4) and Comparator (COMP1P, COMP1M) respectively. To use these DIO pins for their analogue functions, the DIO must be set as an Input with its pull-up resistor, R_{PU}, disabled.

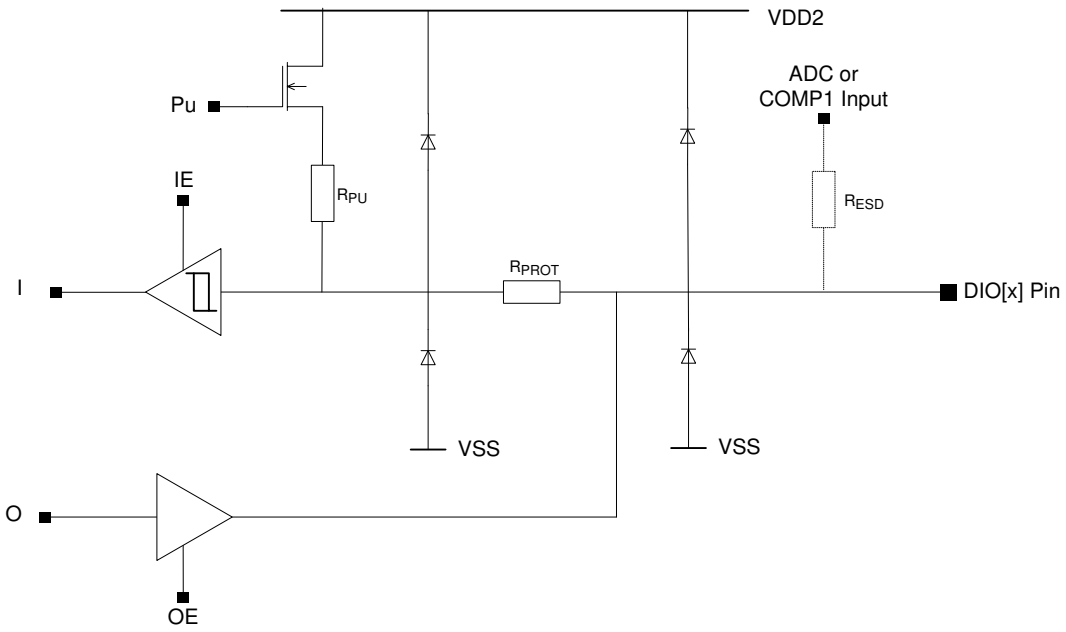


Figure 4: DIO Pin Equivalent Schematic

In reset, the digital peripherals are all off and the DIO pins are set as high-impedance inputs. During sleep and deep sleep, the DIO pins retain both their input/output state and output level that was set as sleep commences. If the DIO pins were enabled as inputs and the interrupts were enabled then these pins may be used to wake up the JN516x from sleep.

3 CPU

The CPU of the JN516x is a 32-bit load and store RISC processor. It has been architected for three key requirements:

- Low power consumption for battery powered applications
- High performance to implement a wireless protocol at the same time as complex applications
- Efficient coding of high-level languages such as C provided with the Software Developers Kit

It features a linear 32-bit logical address space with unified memory architecture, accessing both code and data in the same address space. Registers for peripheral units, such as the timers, UART and the baseband processor are also mapped into this space.

The CPU has access to a block of 15 32-bit General-Purpose (GP) registers together with a small number of special purpose registers which are used to store processor state and control interrupt handling. The contents of any GP register can be loaded from or stored to memory, while arithmetic and logical operations, shift and rotate operations, and signed and unsigned comparisons can be performed either between two registers and stored in a third, or between registers and a constant carried in the instruction. Operations between general or special-purpose registers execute in one cycle while those that access memory require a further cycle to allow the memory to respond.

The instruction set manipulates 8, 16 and 32-bit data; this means that programs can use objects of these sizes very efficiently. Manipulation of 32-bit quantities is particularly useful for protocols and high-end applications allowing algorithms to be implemented in fewer instructions than on smaller word-size processors, and to execute in fewer clock cycles. In addition, the CPU supports hardware Multiply that can be used to efficiently implement algorithms needed by Digital Signal Processing applications.

The instruction set is designed for the efficient implementation of high-level languages such as C. Access to fields in complex data structures is very efficient due to the provision of several addressing modes, together with the ability to be able to use any of the GP registers to contain the address of objects. Subroutine parameter passing is also made more efficient by using GP registers rather than pushing objects onto the stack. The recommended programming method for the JN516x is by using C, which is supported by a software developer kit comprising a C compiler, linker and debugger.

The CPU architecture also contains features that make the processor suitable for embedded, real-time applications. In some applications, it may be necessary to use a real-time operating system to allow multiple tasks to run on the processor. To provide protection for device-wide resources being altered by one task and affecting another, the processor can run in either supervisor or user mode, the former allowing access to all processor registers, while the latter only allows the GP registers to be manipulated. Supervisor mode is entered on reset or interrupt; tasks starting up would normally run in user mode in a RTOS environment.

Embedded applications require efficient handling of external hardware events. Exception processing (including reset and interrupt handling) is enhanced by the inclusion of a number of special-purpose registers into which the PC and status register contents are copied as part of the operation of the exception hardware. This means that the essential registers for exception handling are stored in one cycle, rather than the slower method of pushing them onto the processor stack. The PC is also loaded with the vector address for the exception that occurred, allowing the handler to start executing in the next cycle.

To improve power consumption a number of power-saving modes are implemented in the JN516x, described more fully in Section 18. One of these modes is the CPU doze mode; under software control, the processor can be shut down and on an interrupt it will wake up to service the request. Additionally, it is possible under software control, to set the speed of the CPU to 1, 2, 4, 8, 16 or 32MHz. This feature can be used to trade-off processing power against current consumption.

4 Memory Organisation

This section describes the different memories found within the JN516x. The device contains Flash, RAM, and EEPROM memory, the wireless transceiver and peripherals all within the same linear address space.

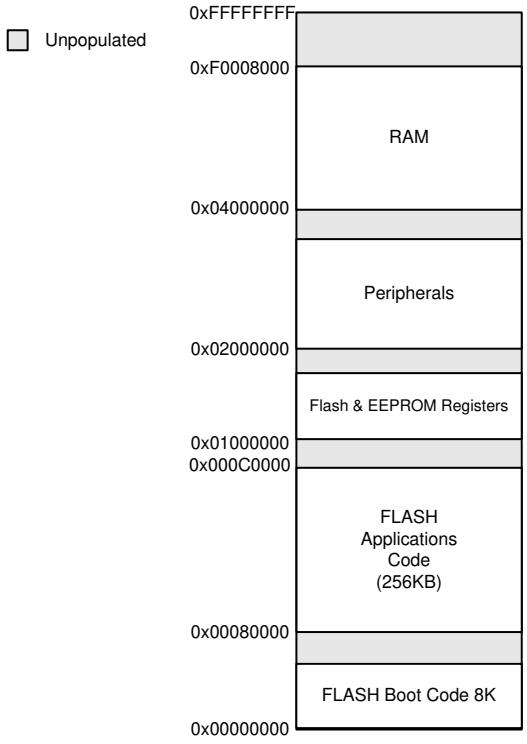


Figure 5: JN5168 Memory Map

4.1 FLASH

The embedded Flash consists of 2 parts: an 8K region used for holding boot code, and a 256K region (JN5168) used for application code. The sector size of the application code is always 32K, for any size of Flash memory. The maximum number of write cycles or endurance is, 10k guaranteed and typically 100k, while the data retention is guaranteed for at least 10 years. The boot code region is pre-programmed by NXP on supplied parts, and contains code to handle reset, interrupts and other events (see section 7). It also contains a Flash Programming Interface to allow interaction with the PC-based Flash Programming Utility which allows user code compiled using the supplied SDK to be programmed into the Application space. For further information, refer to the Flash Programmer User Guide.[9]. The memory can be erased by a single or multiple sectors and written to in units of 256 bytes, known as pagewords.

4.2 RAM

The JN516x devices contain up to 32Kbytes of high speed RAM, which can be accessed by the CPU in a single clock cycle. It is primarily used to hold the CPU Stack together with program variables and data. If necessary, the CPU can execute code contained within the RAM (although it would normally just execute code directly from the embedded Flash). Software can control the power supply to the RAM allowing the contents to be maintained during a sleep period when other parts of the device are un-powered, allowing a quicker resumption of processing once woken.

4.3 OTP Configuration Memory

The JN516x devices contain a quantity of One Time Programmable (OTP) memory as part of the embedded Flash (Index Sector). This can be used to securely hold such things as a user 64-bit MAC address and a 128-bit AES security key. By default the 64-bit MAC address is pre-programmed by NXP on supplied parts; however customers

can use their own MAC address and override the default one. The user MAC address and other data can be written to the OTP memory using the Flash programmer [9]. Details on how to obtain and install MAC addresses can be found in the Flash Programmer User Guide. In addition 384bits are available, organised as three 128bit words, for customer use for storage of configuration or other information.

4.4 EEPROM

The JN516x devices contain 4Kbytes of EEPROM. The maximum number of write cycles or endurance is, 100k guaranteed and 1M typically while the data retention is guaranteed for at least 20 years. (The Persistent Data Manager, includes a wear-levelling algorithm which can help to extend the endurance.) This non-volatile memory is primarily used to hold persistent data generated from such things as the Network Stack software component (e.g. network topology, routing tables). As the EEPROM holds its contents through sleep and reset events, this means more stable operation and faster recovery is possible after outages. Access to the EEPROM is via registers mapped into the Flash and EEPROM Registers region of the address map. The memory can be erased by a single or multiple pages of 64 bytes. It can be written to in single or multiple bytes up to 64 bytes. The customer may use part of the EEPROM to store its own data if desired by interfacing with the Persistent Data Manager. Optionally the PDM can also store data in an external memory. For further information, please read - JenOS User Guide [12].

4.5 External Memory

An optional external serial non-volatile memory (eg Flash or EEPROM) with a SPI interface may be used to provide additional storage for program code, such as a new code image or further data for the device when external power is removed. The memory can be connected to the SPI Master interface using select line SPISEL0 (see fig 6 for details)

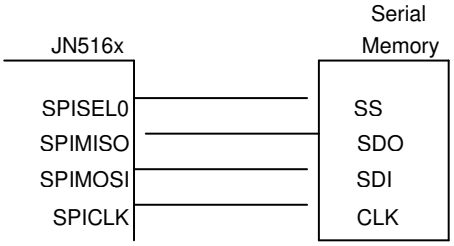


Figure 6: Connecting External Serial Memory

The contents of the external serial memory may be encrypted. The AES security processor combined with a user programmable 128-bit encryption key is used to encrypt the contents of the external memory. The encryption key is stored in the flash memory index section. When bootloading program code from external serial memory, the JN516x automatically accesses the encryption key to execute the decryption process, user program code does not need to handle any of the decryption process; it is transparent. For more details, including the how the program code encrypts data for the external memory, see the application note Boot Loader Operation. [8]

4.6 Peripherals

All peripherals have their registers mapped into the memory space. Access to these registers requires 3 peripheral clock cycles. Applications have access to the peripherals through the software libraries that present a high-level view of the peripheral's functions through a series of dedicated software routines. These routines provide both a tested method for using the peripherals and allow bug-free application code to be developed more rapidly. For details, see Peripherals API User Guide [4].

4.7 Unused Memory Addresses

Any attempt to access an unpopulated memory area will result in a bus error exception (interrupt) being generated.

5 System Clocks

Two system clocks are used to drive the on-chip subsystems of the JN516x. The wake-up timers are driven from a low frequency clock (notionally 32kHz). All other subsystems (transceiver, processor, memory and digital and analogue peripherals) are driven by a high-speed clock (notionally 32MHz), or a divided-down version of it.

The high-speed clock is either generated by the accurate crystal-controlled oscillator (32MHz) or the less accurate high-speed RC oscillator (27-32MHz calibrated). The low-speed clock is either generated by the accurate crystal-controlled oscillator (32.768kHz), the less accurate RC oscillator (centered on 32kHz) or can be supplied externally

5.1 High-Speed (32MHz) System Clock

The selected high-speed system clock is used directly by the radio subsystem, whereas a divided-by-two version is used by the remainder of the transceiver and the digital and analogue peripherals. The direct or divided down version of the clock is used to drive the processor and memories (32, 16, 8, 4, 2 or 1MHz).

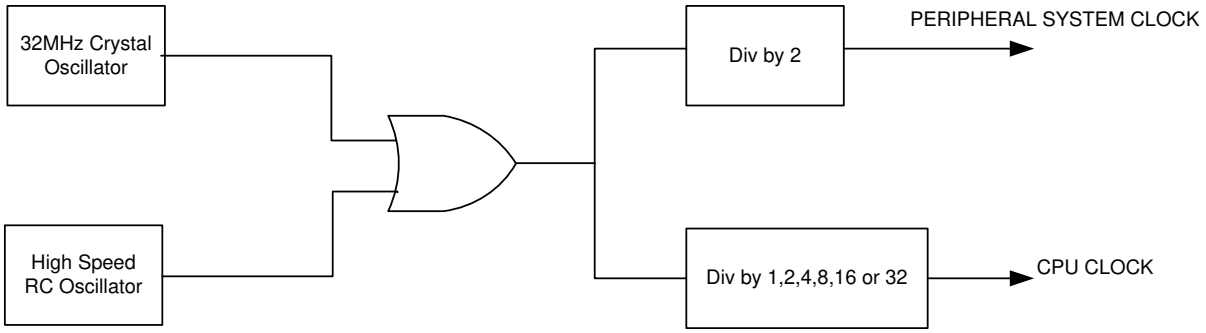


Figure 7 System and CPU Clocks

Crystal oscillators are generally slow to start. Hence to provide a fast start-up following a sleep cycle or reset, the fast RC oscillator is always used as the initial source for the high-speed system clock. The oscillator starts very quickly and will run at 25-32MHz (uncalibrated) or 32MHz +/-5% (calibrated). Although this means that the system clock will be running at an undefined frequency (slightly slower or faster than nominal), this does not prevent the CPU and Memory subsystems operating normally, so the program code can execute. However, it is not possible to use the radio or UARTs, as even after calibration (initiated by the user software calling an API function) there is still a +/-5% tolerance in the clock rate over voltage and temperature. Other digital peripherals can be used (eg SPI Master/Slave), but care must be taken if using Timers due to the clock frequency inaccuracy.

Further details of the High-Speed RC Oscillator can be found in section 19.3.11.

On wake-up from sleep, the JN516x uses the Fast RC oscillator. It can then either:

- Automatically switch over to use the 32MHz clock source when it has started up.
- Continue to use the fast RC oscillator until software triggers the switch-over to the 32MHz clock source, for example when the radio is required.
- Continue to use the RC oscillator until the device goes back into one of the sleep modes.

The use of the fast RC Oscillator at wake-up means there is no need to wait for the 32MHz crystal oscillator to stabilise Consequently, the application code will start executing quickly using the clock from the high-speed RC oscillator.

5.1.1 32MHz Crystal Oscillator

The JN516x contains the necessary on chip components to build a 32MHz reference oscillator with the addition of an external crystal resonator and two tuning capacitors. The schematic of these components are shown in Figure 8. The two capacitors, C1 and C2, should typically be 15pF and use a COG dielectric. Due to the small size of these capacitors, it is important to keep the traces to the external components as short as possible. The on chip transconductance amplifier is compensated for temperature variation, and is self-biasing by means of the internal resistor R1. This oscillator provides the frequency reference for the radio and therefore it is essential that the reference PCB layout and BOM are carefully followed. The electrical specification of the oscillator can be found in

Section 19.3.11. Please refer to Appendix B for development support with the crystal oscillator circuit. The oscillator includes a function which flags when the amplitude of oscillation has reached a satisfactory level for full operation, and this is checked before the source of the high-speed system clock is changed to the 32MHz crystal oscillator

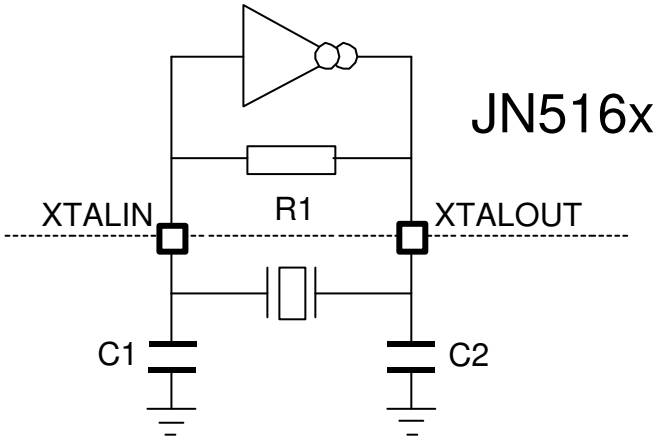


Figure 8: 32MHz Crystal Oscillator Connections

For operation over the extended temperature range, 85 to 125 deg C, special care is required; this is because the temperature characteristics of crystal resonators are generally in excess of +/-40ppm frequency tolerance defined by the IEEE802.15.4 standard. The oscillator cell contains additional circuitry to compensate for the poor performance of the crystal resonators above 100 deg C. Full details, including the software API function, can be found in the application note JN516x Temperature-dependent Operating Guidelines [2]

5.1.2 High-Speed RC Oscillator

An on-chip High-Speed RC oscillator is provided in addition to the 32MHz crystal oscillator for two purposes, to allow a fast start-up from reset or sleep and to provide a lower current alternative to the crystal oscillator for non-timing critical applications. By default the oscillator will run at 27MHz typically with a wide tolerance. It can be calibrated, using a software API function, which will result in a nominal frequency of 32MHz with a +/-1.6% tolerance at 3v and 25 deg C. However, it should be noted that over the full operating range of voltage and temperature this will increase to +/-5%. The calibration information is retained through speed cycles and when the oscillator is disabled, so typically the calibration function only needs to be called once. No external components are required for this oscillator. The electrical specification of the oscillator can be found in Section 19.3.12.

5.2 Low-speed (32kHz) System Clock

The 32kHz system clock is used for timing the length of a sleep period (see Section 18). The clock can be selected from one of three sources through the application software:

- 32kHz RC Oscillator
- 32kHz Crystal Oscillator
- 32kHz External Clock

Upon a chip reset or power-up the JN516x defaults to using the internal 32kHz RC Oscillator. If another clock source is selected then it will remain in use for all 32kHz timing until a chip reset is performed.

5.2.1 32kHz RC Oscillator

The internal 32kHz RC oscillator requires no external components. The internal timing components of the oscillator have a wide tolerance due to manufacturing process variation and so the oscillator runs nominally at 32kHz -10% /+40%. To make this useful as a timing source for accurate wakeup from sleep, a frequency calibration factor derived from the more accurate 16MHz clock may be applied. The calibration factor is derived through software, details can be found in Section 11.3.1. Software must check that the 32kHz RC oscillator is running before using it. The oscillator has a default current consumption of around 0.5uA, optionally this can be reduced to 0.375uA, however, the

calibrated accuracy and temperature coefficient will be worse as a consequence. For detailed electrical specifications, see Section 19.3.9.

5.2.2 32kHz Crystal Oscillator

In order to obtain more accurate sleep periods, the JN516x contains the necessary on-chip components to build a 32kHz oscillator with the addition of an external 32.768kHz crystal and two tuning capacitors. The crystal should be connected between 32KXTALIN and 32KXTALOUT (DIO9 and DIO10), with two equal capacitors to ground, one on each pin. Due to the small size of the capacitors, it is important to keep the traces to the external components as short as possible.

The electrical specification of the oscillator can be found in Section 19.3.10. The oscillator cell is flexible and can operate with a range of commonly available 32.768kHz crystals with load capacitances from 6 to 12.5pF. However, the maximum ESR of the crystal and the supply current are both functions of the actual crystal used, see Appendix B.1 for more details.

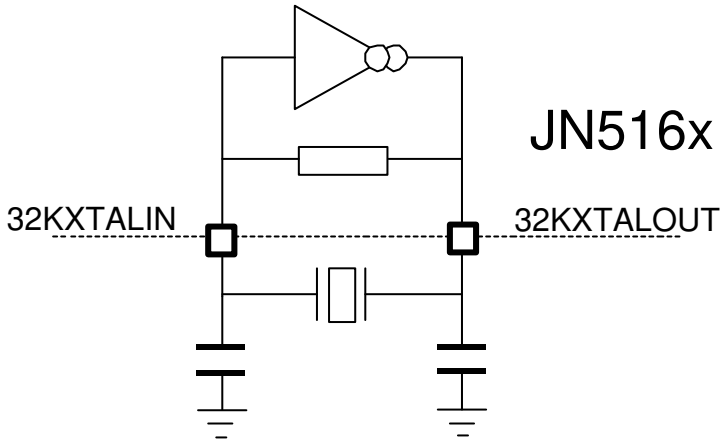


Figure 9: 32kHz Crystal Oscillator Connections

5.2.3 32kHz External Clock

An externally supplied 32kHz reference clock on the 32KXTALIN input (DIO9) may be provided to the JN516x. This would allow the 32kHz system clock to be sourced from a very stable external oscillator module, allowing more accurate sleep cycle timings compared to the internal RC oscillator. (See Section 19.2.3)

6 Reset

A system reset initialises the device to a pre-defined state and forces the CPU to start program execution from the reset vector. The reset process that the JN516x goes through is as follows.

When power is first applied or when the external reset is released, the High-Speed RC oscillator and 32MHz crystal oscillator are activated. After a short wait period (13μsec approx) while the High-Speed RC starts up, and so long as the supply voltage satisfies the default Supply Voltage Monitor (SVM) threshold (2.0V+0.045V hysteresis), the internal 1.8V regulators are turned on to power the processor and peripheral logic. The regulators are allowed to stabilise (about 15us) followed by a further wait (150usec approx) to allow the Flash and EEPROM bandgaps to stabilise and allow their initialisation, including reading the user SVM threshold from the Flash. This is applied to the SVM and, after a brief pause (approx 2.5usec), the SVM is checked again. If the supply is above the new SVM threshold, the CPU and peripheral logic is released from reset and the CPU starts to run code beginning at the reset vector. This runs the bootloader code contained within the flash, which looks for a valid application to run, first from the internal flash and then from any connected external serial memory over the SPI Master interface. Once found, required variables are initialised in RAM before the application is called at its AppColdStart entry point. More details on the bootloader can be found in the application note - Boot Loader Operation. [8]

The JN516x has five sources of reset:

- Internal Power-on / Brown-out Reset (BOR)
- External Reset
- Software Reset
- Watchdog timer
- Supply Voltage detect

Note: When the device exits a reset condition, device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, then the device must be held in reset until the operating conditions are met. (See Section 19.3)

6.1 Internal Power-On / Brown-out Reset (BOR)

For the majority of applications the internal power-on reset is capable of generating the required reset signal. When power is applied to the device, the power-on reset circuit monitors the rise of the VDD supply. When the VDD reaches the specified threshold, the reset signal is generated. This signal is held internally until the power supply and oscillator stabilisation time has elapsed, when the internal reset signal is then removed and the CPU is allowed to run.

The BOR circuit has the ability to reject spikes on the VDD rail to avoid false triggering of the reset module. Typically for a negative going square pulse of duration 1uS, the voltage must fall to 1.2v before a reset is generated. Similarly for a triangular wave pulse of 10us width, the voltage must fall to 1.3v before causing a reset. The exact characteristics are complex and these are only examples.

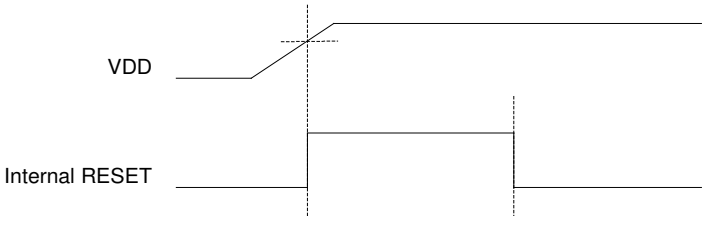


Figure 10: Internal Power-on Reset

When the supply drops below the power on reset 'falling' threshold, it will re-trigger the reset. If necessary, use of the external reset circuit show in Figure 11 is suggested.

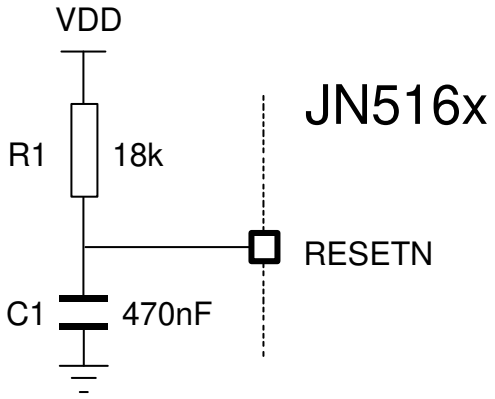


Figure 11: External Reset Generation

The external resistor and capacitor provide a simple reset operation when connected to the RESETN pin but are not neccessary.

6.2 External Reset

An external reset is generated by a low level on the RESETN pin. Reset pulses longer than the minimum pulse width will generate a reset during active or sleep modes. Shorter pulses are not guaranteed to generate a reset. The JN516x is held in reset while the RESETN pin is low. When the applied signal reaches the Reset Threshold Voltage (V_{RST}) on its positive edge, the internal reset process starts.

The JN516x has an internal 500kΩ pull-up resistor connect to the RESETN pin. The pin is an input for an external reset only. By holding the RESETN pin low, the JN516x is held in reset, resulting in a typical current of 6uA.

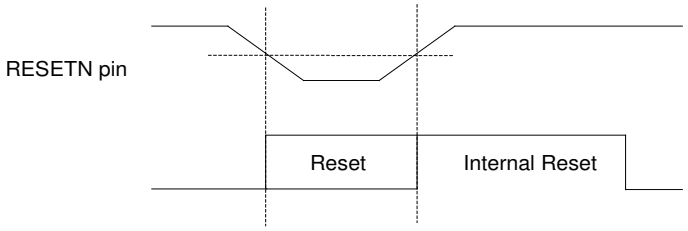


Figure 12: External Reset

6.3 Software Reset

A system reset can be triggered at any time through software control, causing a full chip reset and invalidating the RAM contents. For example this can be executed within a user's application upon detection of a system failure.

6.4 Supply Voltage Monitor (SVM)

An internal Supply Voltage Monitor (SVM) is used to monitor the supply voltage to the JN516x; this can be used whilst the device is awake or is in CPU doze mode. Dips in the supply voltage below a variable threshold can be detected and can be used to cause the JN516x to perform a chip reset. Equally, dips in the supply voltage can be

detected and used to cause an interrupt to the processor, when the voltage either drops below the threshold or rises above it.

The supply voltage detect is enabled by default from power-up and can extend the reset during power-up. This will keep the CPU in reset until the voltage exceeds the SVM threshold voltage. The threshold voltage is configurable to 1.95V, 2.0V, 2.1V, 2.2V, 2.3V, 2.4V, 2.7V and 3.0V and is controllable by software. From power-up the threshold is set by a setting within the flash and the default chip configuration is for the 2.0V threshold. It is expected that the threshold is set to the minimum needed by the system..

6.5 Watchdog Timer

A watchdog timer is provided to guard against software lockups. It operates by counting cycles of the high-speed RC system clock. A pre-scaler is provided to allow the expiry period to be set between typically 8ms and 16.4 seconds (dependent on high-speed RC accuracy: +30%, -15%). Failure to restart the watchdog timer within the pre-configured timer period will cause a chip reset to be performed. A status bit is set if the watchdog was triggered so that the software can differentiate watchdog initiated resets from other resets, and can perform any required recovery once it restarts. Optionally, the watchdog can cause an exception rather than a reset, this preserves the state of the memory and is useful for debugging.

After power up, reset, start from deep sleep or start from sleep, the watchdog is always enabled with the largest timeout period and will commence counting as if it had just been restarted. Under software control the watchdog can be disabled. If it is enabled, the user must regularly restart the watchdog timer to stop it from expiring and causing a reset. The watchdog runs continuously, even during doze, however the watchdog does not operate during sleep or deep sleep, or when the hardware debugger has taken control of the CPU. It will recommence automatically if enabled once the debugger un-stalls the CPU.

7 Interrupt System

The interrupt system on the JN516x is a hardware-vectored interrupt system. The JN516x provides several interrupt sources, some associated with CPU operations (CPU exceptions) and others which are used by hardware in the device. When an interrupt occurs, the CPU stops executing the current program and loads its program counter with a fixed hardware address specific to that interrupt. The interrupt handler or interrupt service routine is stored at this location and is run on the next CPU cycle. Execution of interrupt service routines is always performed in supervisor mode. Interrupt sources and their vector locations are listed in Table 1 below:

Interrupt Source	Vector Location	Interrupt Definition
Bus error	0x08	Typically cause by an attempt to access an invalid address or a disabled peripheral
Tick timer	0x0e	Tick timer interrupt asserted
Alignment error	0x14	Load/store address to non-naturally-aligned location
Illegal instruction	0x1a	Attempt to execute an unrecognised instruction
Hardware interrupt	0x20	interrupt asserted
System call	0x26	System call initiated by b.sys instruction
Trap	0x2c	caused by the b.trap instruction or the debug unit
Reset	0x38	Caused by software or hardware reset.
Stack Overflow	0x3e	Stack overflow

Table 1: Interrupt Vectors

7.1 System Calls

The b.trap and b.sys instructions allow processor exceptions to be generated by software.

A system call exception will be generated when the b.sys instruction is executed. This exception can, for example, be used to enable a task to switch the processor into supervisor mode when a real time operating system is in use. (See Section 3 for further details.)

The b.trap instruction is commonly used for trapping errors and for debugging.

7.2 Processor Exceptions

7.2.1 Bus Error

A bus error exception is generated when software attempts to access a memory address that does not exist, or is not populated with memory or peripheral registers.

7.2.2 Alignment

Alignment exceptions are generated when software attempts to access objects that are not aligned to natural word boundaries. 16-bit objects must be stored on even byte boundaries, while 32-bit objects must be stored on quad byte boundaries. For instance, attempting to read a 16-bit object from address 0xFFF1 will trigger an alignment exception as will a read of a 32-bit object from 0xFFF1, 0xFFF2 or 0xFFF3. Examples of legal 32-bit object addresses are 0xFFF0, 0xFFF4, 0xFFF8 etc.

7.2.3 Illegal Instruction

If the CPU reads an unrecognised instruction from memory as part of its instruction fetch, it will cause an illegal instruction exception.

7.2.4 Stack Overflow

When enabled, a stack overflow exception occurs if the stack pointer reaches a programmable location.

7.3 Hardware Interrupts

Hardware interrupts generated from the transceiver, analogue or digital peripherals and DIO pins are individually masked using the Programmable Interrupt Controller (PIC). Management of interrupts is provided in the Peripherals API User Guide [4]. For details of the interrupts generated from each peripheral see the respective section in this datasheet.

Interrupts can be used to wake the JN516x from sleep. The peripherals, baseband controller, security coprocessor and PIC are powered down during sleep but the DIO interrupts and optionally the pulse counters, wake-up timers and analogue comparator interrupts remain powered to bring the JN516x out of sleep.

Prioritised external interrupt handling (i.e., interrupts from hardware peripherals) is provided to enable an application to control an events priority to provide for deterministic program execution.

The priority Interrupt controller provides 15 levels of prioritised interrupts. The priority level of all interrupts can be set, with value 0 being used to indicate that the source can never produce an external interrupt, 1 for the lowest priority source(s) and 15 for the highest priority source(s). Note that multiple interrupt sources can be assigned the same priority level if desired.

If while processing an interrupt, a new event occurs at the same or lower priority level, a new external interrupt will not be triggered. However, if a new higher priority event occurs, the external interrupt will again be asserted, interrupting the current interrupt service routine.

Once the interrupt service routine is complete, lower priority events can be serviced.