DOI:10.4459/12.002.TRP.EN
NEUNET Technical Report No. 12.002

VirtualSense alpha: Open-Hardware Ultra-Low-Power Wireless Sensor Node

An IEEE 802.15.4-compliant wireless sensor module running a Java-compatible 16 bit VM

Emanuele Lattanzi (DiSBeF Università di Urbino e NeuNet), **Andrea Seraghiti** (DiSBeF Università di Urbino), **Massimo Zandri** (DiSBeF Università di Urbino e NeuNet), **Alessandro Bogliolo** (DiSBeF Università di Urbino e NeuNet)

The availability of off-the-shelf micro controller units based on energy efficient 16-bit RISC processors which provide a wide range of low-power inactive modes with average current in the range of micro Watts and wake-up times in the range of micro seconds makes it possible to develop ultra-low-power sensor nodes able to run a virtual machine to speedup the development and the deployment of sensing/monitoring applications. VirtualSense is an open-hardware/open-source project which aims at the development of IEEE 802.15.4-compliant low-cost ultra-low-power wireless sensor nodes providing a Java-compatible runtime environment which grants to the programmer full control of the low-power states of the hardware. This white paper presents the hardware architecture of the alpha release of VirtualSense, based on a Texas Instruments MSP430F54xxa microcontroller unit.

Indice

General Description and Key Features

VirtualSense is an ultra-low power wireless node for use in wireless sensor networks (WSNs) subject to tight power constraints. Thanks to the on board Java compatible virtual machine (VM) it allows programmers to rapidly develop monitoring applications and communication protocols.

VirtualSense makes use of IEEE 802.15.4 wireless transceivers [1] in order to standardize communication and to inter-operate with other existing devices. The set of on board sensors (including humidity, temperature, and light), together with the possibility to easily connect any external sensor/actuator, allows VirtualSense to be used in a wide range of application fields.

Thanks to the Java-compatible run-time environment Virtualsense allows programmers to rapidly develop and test monitoring applications and communication protocols

In order to promote research and development VirtualSense adopts an open-hardware/open-source model. In particular, it mounts widely available off-the-shelf components and it makes publicly available

all PCB schematics. The open-source software stack is based on a modified version of Darjeeling java-compatible VM [2] running on top of Contiki operating system [3].

VirtualSense adopts an open-hardware/open-source model, making publicly available both the source code and the PCB schematics

Key components and features

The key components of VirtualSense alpha are listed below:

- 250kbps 2.4GHz IEEE 802.15.4 Texas Instruments cc2520 Wireless Transceiver [4]
- 25MHz Texas Instruments MSP430f54xxa microcontroller unit (MCU) with 16k RAM and 128k Flash [5]
- Integrated Humidity, Temperature, and Light sensors
- 512K I²C™ Serial EEPROM [6]
- On-board 48-bit I²C™ Extended Unique Identifier (EUI-48™) [7]
- On-board programmable ultra-low-power RTC [8]

The distinguishing features include:

- Ultra low power consumption (≈10µW in hibernation, ≈100µW in sleep mode, 50/60mW in send/receive modes, respectively)
- Compatibility with state-of-the-art energy harvesting modules
- Fast wakeup from sleep mode (<5µs)
- Programmable timed wake-up from any low-power mode
- Sensitivity to asynchronous external events
- Integrated 12-bit ADC/DAC
- Integrated Supply Voltage Supervisor (SVS)
- Integrated DMA Controller
- USB 2.0 RS232/UART communication with a PC
- Interoperability with other IEEE 802.15.4 devices
- Open-source software stack
- Contiki MAC-layer compatibility [9]

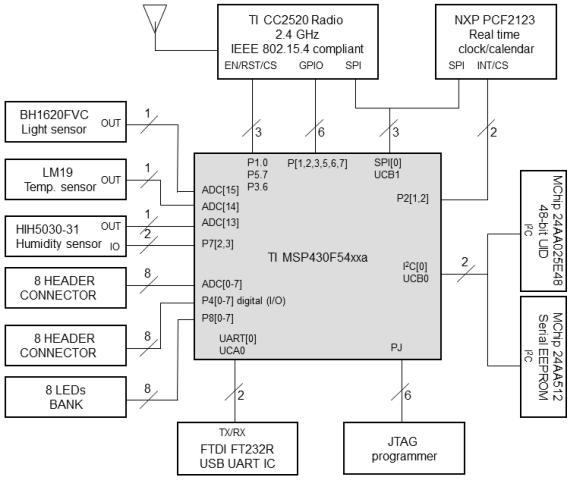

Figure 1. Virtualsense functional block diagram

- Java-compatible run-time environment [1]
- Easy Over the Air (OTA) programming

The full list of components can be downloaded from NeuNet website.

Architecture

VirtualSense is made of ultra-low-power components in order to keep the average consumption compatible with state-of-the-art energy harvesters. Figure 1 shows the functional block diagram representing the node architecture. The core is a MCU belonging to the Texas Instrument MSP430F54xxa family [5]. It communicates through I²C™ bus with a Microchip 24AA025E48 Extended Unique Identifier and with a Microchip 24AA512 serial 512K EEPROM [6][7]. Using the SPI bus, the MCU manages the Texas Instruments CC2520 2.4GHz IEEE 802.15.4 RF transceiver and communicates with the NXP PCF2123 ultra low-power real time clock/calendar [4][8].

Virtualsense module is made of off-the-shelf ultra low power components

A FTDI FT232R chip provides USB 2.0 communication capabilities and power supply to the sensor node, while a JTAG interface enables on-lab node programming [10]. Finally, 3 ADC channels are used to sample data from a BH1620FVC light sensor, from a LM19 temperature sensor and from a HIH5030 humidity sensor[11][12][13].

The Microcontroller Unit

Virtualsense node is based upon Texas Instruments MSP430F54xxa ultra-low-power microcontrollers. The microcontroller unit (MCU) has an optimized architecture and exhibits a large set of low-power inactive modes conceived to achieve extended battery life in portable measurement applications and to enable the implementation of autonomous WSNs harvesting supply power from the environment. Nevertheless, the MCU is a powerful 16-bit RISC CPU running at up to 25MHz with a built-in digitally controlled oscillator (DCO) which allows wake-up from low-power to active modes in less than 5 µs.

Moreover the MCU is equipped with three 16-bit timers, a high performance 12-bit ADC, a hardware multiplier, a DMA controller, and four universal serial communication interfaces (USCI) which can be used as SPI/UART/I^2C buses.

Low power modes

The MCU has one active mode and 6 software selectable low-power modes called LPM0, LPM1, LPM2, LPM3, LPM4, and LPM4.5. Transitions from low power modes to active mode are triggered by internal or external interrupt events. Upon an interrupt event the MCU wakes up, handles the interrupt, and goes back to the original low-power mode. From LPM0 to LPM3 the MCU progressively shuts down the CPU and the internal clock modules while maintaining the internal DCO up and running in order to make it possible for the programmer to schedule self-events to reactivate the MCU (in less than 5µs) in order to perform periodic monitoring tasks while taking advantage of the inactive states to save power in between.

On the contrary, the internal DCO and the crystal oscillator are turned off in LPM4 and LPM4.5, thus avoiding the generation of self-events. In order to wake up the MCU from these deep low-power states, an external interrupt needs to be raised on digital I/O ports P1 and P2. In our architecture, external timer interrupts can be generated by the RTC.

In low power modes LPM0-LPM4 the MCU retains both the content of main memory and the configuration/state of its I/O ports, so that wakeup events can be processed by the corresponding interrupt service routines without any reset.

On the other hand, in the lowest inactive mode (namely, LPM4.5) the voltage regulator of the MCU power management module (PMM) is completely disabled, so that no power supply is provided to the CPU, to main memory, and to all digital modules. As a consequence, the device is completely disabled and all the volatile data and configurations are lost. The only piece of information which is retained is the condition of the pins.

Low power mode	Current consumption	unit
Active (@ 25Mhz)	8.9	mA
LPM0	73	µA
LPM1	11.7	µA
LPM2	2.3	µA
LPM3	1.4	µA
LPM4	1.3	µA
LPM4.5	0.1	µA

Table 1. MCU power modes current consumption

Exit from LPM4.5 causes a complete reset of the system, entailing the reboot of the MCU and of all firmware components. Wakeup time from LPM4.5 is significantly longer than from other inactive modes for two main reasons: first, the supply voltage of the core needs to be regenerated, second, the entire software stack has to be restarted before resuming operation. Moreover, data persistency has to be explicitly handled by the application developer exploiting external non-volatile memory devices.

It is worth noticing that, whenever the MCU enters in LPM4.5, the LOCKIO bit of register PM5CTL0 belonging to the PMM module is automatically set causing all I/O pins to be locked in their current status (please refer to Section "Low-Power Design Tips" for the details). This makes it necessary, at wakeup, to unlock the PMM module and clear the PM5CTL0 register in order to properly reconfigure the I/O pins. Figure 2 shows the code fragment needed to enable I/O pin configuration, while Table 1 shows typical power consumption of the MCU in its different power states.

```
/* remove PM5 lock to enable I/O pin
 * configuration needed when wake-up
 * from LPM4.5
 */
 PMMCTL0_H = PMMPW_H; // open PMM with password
 PM5CTL0 &= ~LOCKIO;  // Clear LOCKIO
 PMMCTL0_H = 0x00;    // close PMM
```

Figure 2. Enable I/O pins configuration after exiting from LPM4.5

Power Management Module (PMM) and Supply Voltage Supervisor (SVS)

The PMM has the main function to generate the supply voltage for the core (VCORE) starting from the voltage applied to the device (DVCC), while the SVS provides several mechanisms for the supervision and monitoring of the values of both DVCC and VCORE.

In general, VCORE supplies the CPU, the memories (flash/RAM), and the digital modules, while DVCC supplies I/Os and all analog modules (including oscillators). The PMM can provide four different levels of VCORE and it can be programmed in order to select the lowest supply voltage compatible with the speed of the CPU. In fact, the minimum voltage needed by the core depends on the selected CPU frequency.

The PMM and the SVS make available four software-programmable voltage thresholds to supervise and monitor the levels of DVCC and VCORE. In particular DVCC is supervised and monitored by the high-side supervisor (SVSH) and high-side monitor (SVMH), respectively while VCORE is supervised and monitored by the low-side supervisor (SVSL) and low-side monitor (SVML), respectively. Whenever a monitoring threshold is crossed, the PMM generates an interrupt that can be handled by a specific software routine. On the contrary, whenever a supervisor threshold is crossed, the SVS automatically resets the MCU by means of a power-on reset (POR) event.

Digital I/O Module

The MCU has up to twelve 8-bit digital I/O ports (the real number depends on the pinout of the chip) called P1 to P11 plus a unnumbered port called PJ, which can be used to implement the JTAG communication protocol in order to easily reprogram the MCU. Each port (but PJ) provides eight I/O lines which can be independently configured in input or output (by means of PxDIR registers) and independently read or written. Moreover, it is possible to enable or disable a pullup or pulldown resistor in each port using the PxREN register. Finally, each port, while configured in output direction (PxDIR=1), can be programmed in full or reduced drive strength by means of PxDS registers. In reduced drive strength mode each I/O line can drive up to 15mA while in full drive strength the maximum drive current reaches 50mA.

Only ports P1 and P2 provide interrupt capabilities and can wake up the MCU from any low power mode. The 16 lines of the two ports can be individually enabled and configured to be sensitive to rising or falling edges of the corresponding input signals. Each I/O port is characterized by a low leakage current in the 50nA range.

Port pins are often multiplexed with other peripheral module functions, such as SPI, I^2C, UART, which can be activated by means of the corresponding PxSEL bits. It is worth mentioning that the binding between port pins and interface bits is hardwired.

Universal Serial Communication Interface (USCI)

The MSP430f54xxa MCU is equipped with four different USCI modules, supporting multiple serial communication modes. In particular, the MCU has two USCI_A modules (USCI_A0 and USCI_A1) and two USCI_B modules (USCI_B0 and USCI_B1). While USCI_Ax modules support UART mode, pulse shaping for IrDA communications, and SPI mode, USCI_Bx modules support only I^2C and SPI modes.

In order to use an I/O line in USCI mode the corresponding PxSEL bit needs to be set to disconnect the pin from the general purpose I/O module and to establish the connection with the corresponding bit of the USCI module.

12 bit ADC Module

The ADC module supports fast 12-bit analog-to-digital conversions. This module implements a 12-bit sample select control, a reference generator, and a 16-word conversion-and-control buffer. Thanks to the built-in buffer, the ADC can sample and convert up to 16 independent signals without involving the CPU. Depending on the conversion mode, subsequent samples taken from each channel are overwritten in the buffer. So that the CPU can decide when to access the buffer to read the last sampled values.

The ADC module allows the programmer to select: the sampling period, the on-chip reference voltage, and the internal conversion clock source.

Moreover, it provides four different conversion modes: i) single-channel, ii) repeat-single-channel, which overwrite the samples, iii) autoscan, which takes and convert a sample from all enabled channels, iv) and repeated autoscan, which repeatedly scans all the enabled channels.

Low-Power Design Tips

Achieving the nominal low-power levels specified in the datasheet of the MCU is not a trivial task, since the actual consumption of each low-power mode

strongly depends on the configuration of the MSP430f54xxa MCU.

This section outlines a few tricks which can help the developer to approach the lower bound.

First of all, the programmer has to take care of the status of the I/O pins in order to avoid uncontrolled current leakage. To this purpose, we need to distinguish connected/used pins and unconnected/unused I/O pins. Each unconnected pin has to be configured as general purpose I/O with output direction during code initialization and left unconnected (rather than grounded) on the PCB. Connected pins, on the contrary, have to be properly

Typical output power (dBm)	Typical current consumption (mA)
5	33.6
3	31.3
2	28.7
1	27.9
0	25.8
-2	24.9
-4	23.1
-7	19.9
-18	16.2

Table 2. CC2520 transmission power and current consumption

configured when not used (i.e. when the MCU enters in a low-power state or when the connected peripheral is shut down or not used). For this reason, for example, before entering in LPM4.5 all the pins have to be set to general purpose I/O and configured in "input high impedance". In fact, it is critical that no inputs are left floating or otherwise an uncontrolled extra current can be drawn by the MCU. In the same way digital I/O pins controlling an external peripheral should be configured (if possible) to "input high impedance" in order to prevent significant current drain between the MCU and the peripheral, when it is not used.

Finally, each internal MCU module, such as ADC converter, USCI interfaces, DMA controller, SVS, PMM, and internal RTCs should be shut down when unused in order to minimize power consumption.

Radio transceiver

Virtualsense communicates through a Texas Instruments CC2520 RF transceiver, a second-generation ZigBee®/IEEE 802.15.4 RF transceiver working in the 2.4 GHz band. CC2520 RF provides several integrated features, including: hardware support for frame handling, data buffering, burst transmissions, data encryption, data authentication, clear channel assessment, link quality indication, frame filtering, standard RX mode, and low-current RX mode. All these built-in features significantly reduce the MCU computational load and power consumption.

Radio-MCU Communication

The CC2520 is controlled by the MCU through the SPI port and six general purpose I/O pins (GPIOs), as shown in Figure 1. GPIOs are fully programmable and can be configured to route different state signals and exceptions. Notice that, in the proposed architecture, CC2520 GPIOs are fully mapped on the P1 port of the MCU which provides interrupt capabilities. In this way it is possible to raise interrupts associated with specific RF events.

Power Modes

CC2520 provides three power modes (Active, LPM1, and LPM2) and an extra low-current RX mode. In all these modes the power supply is always provided to the device.

LPM2 is the lowest power consumption mode where the digital voltage regulator is turned off, no clocks are running, no data is retained, and all analog modules are in power down state. In this state the power consumption is about 4.5µA, in spite of the complete inactivity and of the need to reboot the embedded controller at wakeup.

In LPM1 the digital voltage regulator is on but no clocks are running. In this state all data and configurations are retained and the analog modules can be controlled by the MCU. The power consumption is about 1mA.

In Active mode (AM) the digital voltage regulator is on and the crystal oscillator clock is running. The power consumption is of 2.6mA.

The analog RX/TX module can be activated and de-activated when needed by means of a byte instruction provided by the MCU through the SPI bus. In TX mode the power consumption of the entire CC2520

ranges from 16.2mA @ -18dBm to 33.6mA @ +5dBm, while in standard RX mode the power consumption is 23.3mA. Using the low-current RX mode reduces the power consumption in the receiving phase down to 18.5mA at the expense of a decrease in sensitivity from -98dBm to -50dBm.

Transmission output power can be tuned among nine different power levels affecting the power consumption consequently. Table 2 shows the tunable transmission power levels and the corresponding power consumption.

Frame Filtering

The frame filtering function rejects non-intended frames and it can be configured to reject frames not matching the local PAN id, the local short address, or the local extended address previously stored in CC2520 RAM. When frame filtering is enabled and the filtering algorithm accepts a received frame, an RX_FRM_ACCEPTED exception is generated to notify the reception of a new frame to the MCU.

Although the frame filtering mechanism is mainly targeted to support firewall functionality, it is possible to use it to save power. In fact, the RX_FRM_ACCEPTED exception is generated immediately after receiving the fields required to determine the frame matching without waiting for the entire frame to be received. In this way it is possible to shut down immediately the receiver in case of a non-intended frame and to avoid processing the rest of the frame and waking up the MCU. This trick is particularly effective in case of large packets.

On board Ultra-low power RTC

VirtualSense uses the NPX PCF2123 RTC to implement an "hibernation" mode that exploits LPM4.5 while granting to the programmer the capability of scheduling period wakeup call in spite of the complete shut down of the internal clock. For self-wakeup purposes, the timer interrupt of the RTC needs to be programmed before entering in LPM4. The use of an external RTC does make sense tanks to the extremely low power consumption of the PCF2123, which can generate timed interrupts even from a low-power mode where it consumes only 100nA.

Notice that without an external RTC the MCU could schedule self wakeup events only from LPM3, which consumes approximately 1.4µA.

The only drawback of hibernation is the lack of data retention, which imposes, at wakeup, to reboot the MCU, to restart the VM, and to restore the software execution state from the on-board EEPROM. Needless to say, is the execution state needs to be resumed, it has to be explicitly stored in the EEPROM before entering hibernation.

Conclusions

VirtualSense alpha is the first prototype of a low-cost open-hardware wireless sensor module which enables the deployment of energy-harvesting wireless sensor networks programmable in Java. Both the PCB schematics and the list of hardware components can be downloaded from NeuNet web site (http://www.neunet.it/virtualsense/).

The energy-efficiency of all the components, together with the power management opportunities and with the careful design of the architecture keep the overall power consumption of VirtualSense between 1mW and 60mW in active mode (depending on the operating frequency of the MCU and on the activity of the transceiver) and provide low-power states with power consumption of about 10µW to be exploited while waiting for external events.

References

[1] IEEE std. 802.15.4 - 2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

[2] N. Brouwers, P. Corke, and K. Langendoen. "Darjeeling, a Java compatible virtual machine for microcontrollers", in Proc. Of the ACM/IFIP/USENIX Middleware Conference Companion, pp. 18-23, 2008

[3] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors", in Proc. of the IEEE Conf. on Local Computer Networks, pp. 455-462, 2004.

[4] Texas Instruments CC2520 datasheet. http://www.ti.com/lit/ds/symlink/cc2520.pdf

[5] Texas Instruments. MSP430F54xxA Mixed Signal Microcontroller datasheet. http://www.ti.com/lit/ds/symlink/msp430f5418a.pdf

[6] Microchip 24AA512 serial 512K EEPROM datasheet. http://ww1.microchip.com/downloads/en/devicedoc/21754e.pdf

[7] Microchip 24AA025E48 Extended Unique Identifier datasheet. http://ww1.microchip.com/downloads/en/DeviceDoc/22124D.pdf

[8] NXP PCF2123 ultra low-power real time clock/calendar datasheet. http://www.nxp.com/documents/data_sheet/PCF2123.pdf

[9] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol" SICS Technical Report T2011:13 - ISSN 1100-3154 - December 2011

[10] FTDI FT232R datasheet. http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

[11] BH1620FVC Analog current output ambient light sensor datasheet. http://www.rohm.com/products/databook/sensor/pdf/bh1620fvc-e.pdf

[12] Texas Instruments LM19 temperature sensor datasheet. http://www.ti.com/lit/ds/symlink/lm19.pdf

[13] HIH-50301 Low Voltage Humidity Sensors datasheet. http://sensing.honeywell.com/index.php/ci_id/49692/la_id/1/document/1/re_id/0

NeuNet is a non-for-profit cultural association founded in 2010 for promoting interdisciplinary studies, pilot projects, and technology transfer in the fields of access networks, multimedia services, and sensor networks. NeuNet involves computer scientists, electrical engineers, economists, sociologists, and jurists.

For further information please visit http://www.neunet.it/, join the **Net Neutrality & Future Internet Forum** group on LinkedIn, and follow **@neutralaccess** on Twitter.