

A Wirelessly-Powered Platform for Sensing and Computation

Joshua R. Smith¹, Alanson P. Sample², Pauline S. Powledge¹,
Sumit Roy², and Alexander Mamishev²

¹ Intel Research Seattle
1100 NE 45th Street
Seattle, WA 98105
USA

² Department of Electrical Engineering,
Box 352500, University of Washington,
Seattle, WA 98195
USA

Joshua.r.smith@intel.com, alanson@u.washington.edu,
pauline.s.powledge@intel.com, {roy, mamishev}@ee.washington.edu

Abstract. We present WISP, a wireless, battery-free platform for sensing and computation that is powered and read by a standards compliant Ultra-High Frequency (UHF) RFID reader. To the reader, the WISP appears to be an ordinary RFID tag. The WISP platform includes a general-purpose programmable flash microcontroller and implements the bi-directional communication primitives required by the Electronic Product Code (EPC) RFID standard, which allows it to communicate arbitrary sensor data via an EPC RFID reader by dynamically changing the ID it presents to the reader. For each 64 bit “packet,” the WISP’s microcontroller dynamically computes the 16-bit CRC that the EPC standard requires of valid packets. Because the WISP device can control all bits of the presented ID, 64 bits of sensor data can be communicated with a single RFID read event. As an example of the system in operation, we present 13 hours of continuous-valued light-level data measured by the device. All the measurements were made using power harvested from the RFID reader. No battery, and no wired connections (for either power or data) were used. As far as we are aware, this paper reports the first fully programmable computing platform that can operate using power transmitted from a long-range (UHF) RFID reader and communicate arbitrary, multi-bit data in response to a single RFID reader poll event.

1 Introduction and Prior Work

This paper describes WISP, a wireless, battery-free platform for sensing and computation that is powered and read by a standards-compliant Ultra-High Frequency (UHF) RFID reader. The notable features of the device are a wireless power supply, low bit-rate UHF backscatter communication, and a fully programmable ultra-low-power 16-bit flash microcontroller. This particular point in the sensor platform design space

offers some attractive features for ubiquitous computing, but has not yet been explored very thoroughly.

One approach to ubiquitous sensing is to use wired sensors. This approach is well-suited to creating purpose-built instrumented environments supporting long-term observation.[1] This approach has the advantage that there is no battery lifetime or battery size constraint. The drawback is the need for wires. In [1], the authors describe such a purpose-built living space, which includes custom cabinetry to house sensors as well as cables for both power and data. A second approach, favored by the Wireless Sensor Networks community, is to use battery-powered devices that communicate by ordinary radio communication, often in a peer-to-peer fashion.[2] One disadvantage of this approach is the size and lifetime constraints imposed by batteries. Because of the lifetime constraint, it would not be possible to permanently embed a battery-powered device in a building or civil structure such as a bridge. A third approach includes generating power from environmental sources[3], such as vibration, light, or human motion,[4] and then communicating the sensor data by ordinary RF transmission. The final class of approaches, within which this work fits, is to deliberately transmit power from a large source device to the sensor platforms, which then harvest this “planted” power, rather than relying on generation of electrical energy from naturally occurring or “wild” sources as in the third approach.

The space of wireless power/data transmission can be subdivided further. Chip-less approaches such as Theremin’s cavity resonator microphone[5], as well as more recent examples such as [6] and [7], are based on an analog technique in which a quantity to be sensed modifies the frequency or quality factor of a resonant structure. Changes in the resonance can be detected by a “reader” device that is effectively supplying power and collecting analog sensor data. This analog approach is generally limited to a small number of sensors, since the devices cannot be given arbitrarily long unique IDs. Furthermore, these analog sensor devices are not capable of on-board computation, which means the system cannot benefit from channel sharing, error detection and correction, embedded compression and filtering, and other capabilities enabled by digital computation and communication.

RFID tags are wirelessly powered digital devices that include a conventional Integrated Circuit (IC).[8] Conventional RFID tags are fixed function devices that typically use a minimal, non-programmable state machine to report a hard-coded ID when energized by a reader. The Electronic Product Code (EPC) standard operates in the Ultra-High Frequency (UHF) band (915MHz in the U.S.), which has substantially improved the range and field-of-view for RFID reading over previous generations of RFID technology. The standard’s broad adoption is enabling a new generation of applications and interoperable products. The “EPC Class 1 Generation 1” specification [9] (“Gen1”) is the most widely deployed so far. Conventional RFID tags with a worn RFID reader have been used for activity monitoring in eldercare scenarios[15]; in this case, the RFID read event is in effect a reader-tag proximity sensor.

Integrating RFID tags with secondary sensors has been proposed [10] or implemented [11] in various contexts, and a small number of commercially available RFID sensors exist. In almost all cases, these devices are fixed-function, and simply report a unique ID and sensor data. Most of the commercially available products assume that

the sensor platform is battery-powered, and use the RFID channel for communication but not power. Also, existing RFID-sensor devices are generally not programmable platforms supporting arbitrary computation. What may be the only commercially available fully programmable microcontroller with an RFID interface is described in [12]. However, this device can only transmit one bit of sensor data per read event, and, like [11], operates at 125kHz, which limits its range to inches, and thus substantially curtails its applicability for ubiquitous computing applications. The Near Field Communication (NFC) standard is also a relevant point of comparison[16]. NFC uses short-range RFID-style backscatter communication to link powered devices. It does not provide any power harvesting capability. In prior work [13], the WISP team described a microcontroller powered by a long-range EPC-compatible RFID reader, but this prior system used a very rudimentary communication mechanism: an RF semiconductor switch multiplexed two commercially manufactured RFID chips to one RFID antenna. This system, therefore, was capable of transmitting just one bit of sensor data per read event, which substantially limited its usefulness for ubiquitous sensing and computing applications. Further disadvantages of the earlier system were the requirement of two separate antennae (one for harvesting and one for communication), and the highly non-standard manual fabrication process required to connect the RF switch, the two RFID chips, and the RFID tag antenna. The device presented in the present paper uses a single antenna for power harvesting and communication, and (with the possible exception of the antenna) its printed circuit board can be assembled via standard automated equipment using conventional electronic components (i.e. it does not require any commercial RFID tags, which are generally proprietary, undocumented black boxes).

As far as we are aware, this paper reports the first fully programmable computing platform that can operate using power transmitted from a long-range (UHF) RFID reader and communicate arbitrary, multi-bit data in response to a single RFID reader poll event.

2 System Design

By encoding the sensor data in the field normally used for the EPC ID, we are able (in principle) to read our devices with any EPC-standard commercial RFID reader. This is attractive because it allows us to make use of the large installed base of EPC-standard readers, and leverage the RFID industry's significant past and future investments in reader technology development.

The EPC standard provides a mechanism that allows a single reader to communicate with multiple tags in its field of view [9]. The "singulation" mechanism that allows the tags to share the reader channel requires bi-directional communication between reader and tags. To sketch the protocol briefly, the reader polls the population of tags asking "which tag present has high bit 1?" If more than one tag replies by broadcasting their full IDs, the reader detects the collision and further sub-divides the tag population, by refining the query to ask for tags with high bits "11," and then (if necessary, depending on the results), "111," "110," "10," and so on. Once a tag's ID has been read without a collision, the reader causes it to go "quiet" until further notice, and not participate in

the protocol until it is “awakened.” This singulation procedure is sometimes referred to informally as “tree walking,” because the reader explores the “tree” of IDs. Note that the depth of the tree is logarithmic in the number of IDs (and linear in the number of ID bits), so this procedure is efficient in a computational complexity sense.

The EPC standard provides multiple low-level query commands, and particular readers provide multiple modes that compose these commands in different fashions, to allow application developers to make different trade-offs among performance attributes such as read latency, read reliability, number of tags that can be acquired per read event, and so forth.

Because of the bi-directional communication requirements of the EPC standard, our device has to implement three distinct analog functions: power harvesting, reader-to-WISP communication, and WISP-to-reader communication.

We have taken a “software radio” approach to the timing and protocol logic: the analog hardware provides minimal physical layer (PHY) functionality, demodulating the 915MHz carrier and thresholding to extract raw amplitude levels (low or high) from the reader-to-WISP signal, and generating raw backscatter signals (low or high) for detection by the reader. All the timing and protocol logic is generated via software running on the microcontroller.

2.1 Hardware

Figure 1 shows a block diagram of the WISP platform. Figure 3 contains photos of a functional WISP unit. The WISP’s computational capability is provided by the TI MSP-430F-1232 [14], a 16-bit, ultra-low-power flash microcontroller. This device provides 8 KBytes + 256 Bytes of Flash memory, 256 bytes of RAM, and a 10 bit, 200kilo-samples-per-second Analog to Digital Converter (ADC). The low power consumption features of this relatively new device family are a major part of the reason it has become possible to power a general purpose microcontroller using a long-range RFID reader.

The analog circuitry for power harvesting, reader-to-WISP communication, and WISP-to-reader communication is provided by a custom analog printed circuit board of our own design. The main WISP board has header connectors for a daughterboard to which sensors can be added, and which has a connector for programming and debugging via a JTAG interface.

2.1.1 Power Harvesting

The power harvester rectifies the 915MHz carrier generated by the RFID reader into a DC voltage that then powers the rest of the system. Our harvester is a multi-stage voltage multiplier, shown schematically in Figure 2. The power harvester functions in “half wave” rectifier mode: current is only passed to the next stage during the positive phase of the RF signal. Low threshold Schottky diodes are used to reduce losses and increase the maximum operating distance. Care must be taken to properly match the impedance of the antenna to that of the rest of the circuit. To design the impedance matching network for optimal power transfer (or alternatively maximum voltage output), we connected a network analyzer to the antenna ports of the WISP and adjusted a variable capacitor in the LC impedance-matching network. We adjusted

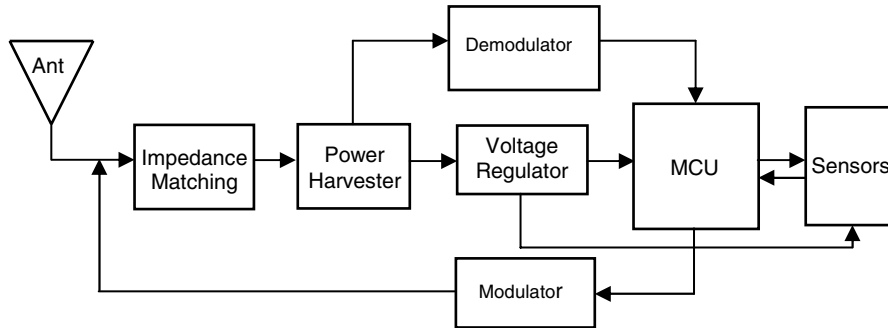


Fig. 1. WISP block diagram. Many of the same elements are found in conventional Wireless Sensor Network nodes. One key difference is that the WISP contains an RF power harvesting module instead of a battery. A second key difference is the box labeled “Modulator.” In the WISP presented in this paper, the Modulator box contains just a single transistor, all that is required for WISP-to-reader communication. In a conventional WSN node, a radio frequency oscillator and other radio components are required to transmit data.

the value of this variable component to ensure that the impedance of the power harvesting circuit viewed from the antenna is purely real. Once an appropriate value has been discovered in this fashion, the design can be fixed, and non-variable components used.

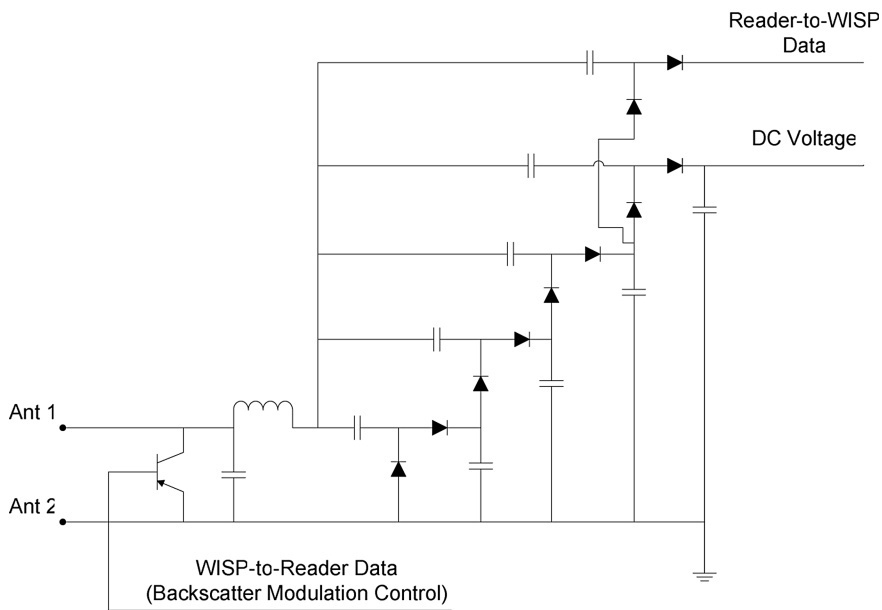


Fig. 2. WISP Analog Front End (AFE) schematic. The WISP AFE has connections for the two branches of a dipole antenna, an input for controlling the uplink (WISP-to-reader) modulation, an output for DC power, and an output for downlink (reader-to-WISP) data.

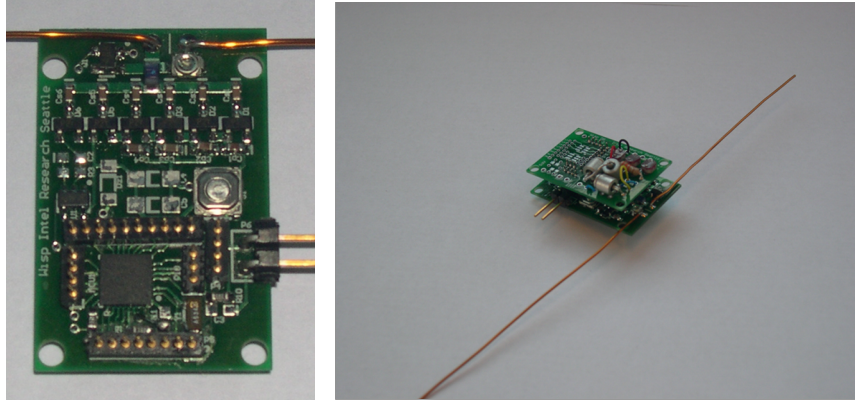


Fig. 3. Photographs of WISP. **LEFT IMAGE:** the microcontroller is the square black QFN package in the lower portion of the image. The two branches of the dipole antenna are visible at the top of the image. The two pins exiting the right side of the image are power output and ground. The microcontroller's pins are broken out to the small header pins surrounding the microcontroller. The large silver square is a reset switch. **RIGHT IMAGE:** WISP with sensor daughterboard attached. A light sensor and three tilt switches are mounted on the daughterboard. This image shows the size of the dipole antenna.

When implementing the software-radio functions our current design requires, the MSP430 microcontroller consumes more power than the harvester is typically able to supply. Thus the WISP microcontroller is duty-cycled so that the harvester has enough time to charge up storage capacitors which are then discharged when the MCU is active.

2.1.2 Reader to WISP Communication

To encode reader-to-tag data, the reader amplitude-modulates the 915MHz RF carrier wave it emits. Normally the carrier wave form remains at a constant amplitude; when bits are transmitted, the amplitude of the carrier drops to approximately ten percent of its normal value. The duration of the low indicates a logical "one" or a "zero." A short break indicates a "zero," and a long break indicates a "one." Before the signal can be decoded by the microcontroller it passes through an additional multiplier stage in parallel with the final voltage output stage. Unlike the ordinary multiplier stages, the decoder stage has no final buffer capacitor. The decoder produces a signal that matches the envelope of the carrier emitted by the reader. The MCU's software then decodes the duration of the lows into reader-to-tag bits.

2.1.3 WISP to Reader Communication

Unlike more conventional sensor nodes, RFID tags do not actively transmit radio signals. Instead they modulate the impedance of their antenna which causes a change in the amount of energy reflected back to the reader. This modulated reflection is typically referred to as backscatter radiation. In order to change the impedance of the antenna, a transistor is placed between the two branches of the dipole antenna. When

the transistor is in conducting mode, it short circuits the two branches of the antenna together, changing the antenna impedance; in the non-conducting state, the transistor has no effect on the antenna, and the power harvesting and data downlink functions occur as if it were not present. The WISP-to-reader communication is currently implemented with a bipolar junction transistor rated for RF operation.

2.1.4 Communication Implementation; Verify ID

For our initial proof of concept, we have implemented just the simplest EPC read mode, Verify ID. This read command is ordinarily used to verify the ID of a tag whose ID has just been re-programmed. A timing diagram for Verify ID is shown in figure 4. This mode does not do any “tree walking” singulation, and is really suited for reading just one tag at a time. We believe that singulation protocols that provide for both sensing and ID functionality are a promising area for future research.

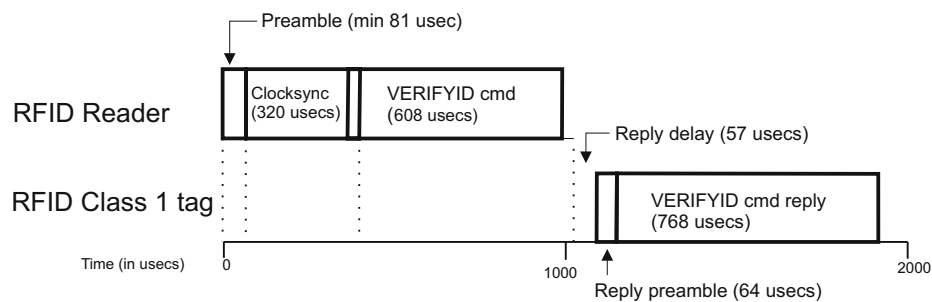


Fig. 4. Timing of EPC Verify ID transaction. The values shown are not the nominal timings provided in the EPC specification, but the (substantially different) actual values measured for an Alien Nanoscanner 915MHz RFID reader.

4 Experimental Results

As an end-to-end test of our system in operation, we built a sensor daughterboard with a light sensor and 5 one bit tilt sensors (mercury switches). The microcontroller has an integrated temperature sensor. The microcontroller was programmed to include, in each response packet, 10 bits of light-level data, 10 bits of temperature data, 5 bits of tilt-switch data, and 8 bits of configuration data. The configuration field can be thought of as a pointer to a configuration record that is intended to allow the application to learn what sensors are being reported.

Crucially, the WISP must also compute the CRC for the “live” sensor data to form a valid packet. (Older models of Alien RFID readers ignore the CRC in Verify mode, but newer models seem to filter out most Verify ID reads containing invalid CRCs.) Note that ordinary fixed-ID RFID tags have the CRC hard-coded; they do not need to dynamically compute it. We made use of the WISP’s general-purpose computing capability to implement this dynamic CRC functionality.

Figure 5 shows the output voltage from the power harvester as a function of distance, under several different loads: no load, 100K, and 10K. The different load

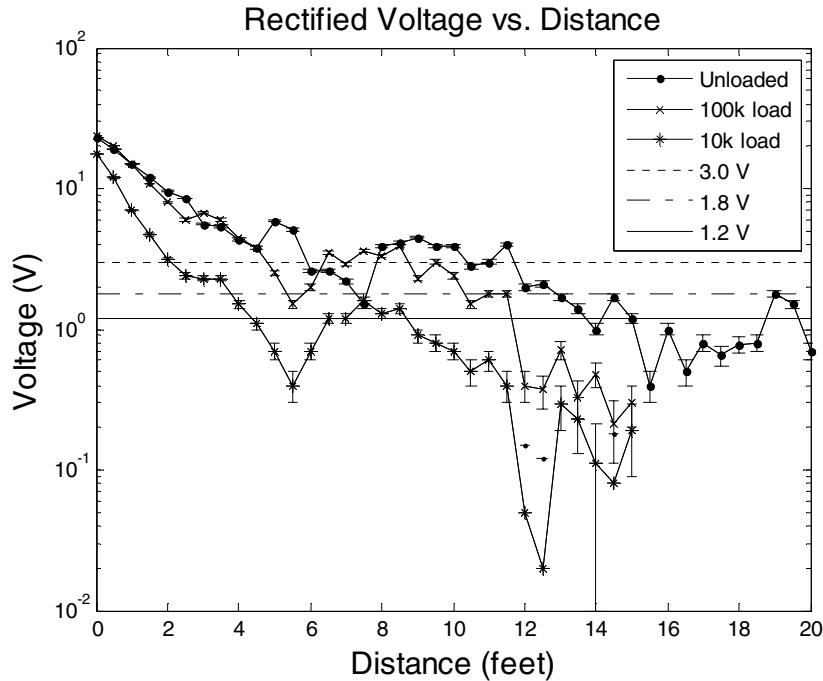


Fig. 5. Power harvesting performance of WISP under several loads. The WISP's power requirements, modeled by the load resistance in the figure, can be changed by adjusting the duty cycle at which it operates. The "no load" line can be thought of as the low duty-cycle limit. In addition to the power constraint, there is a voltage constraint: if the microcontroller is not supplied with sufficient voltage, it either will not be able to operate at all, or will not operate at the desired frequency. The figure shows three voltage thresholds: 3V, 1.8V, and 1.2V. Below the 3V threshold, the current WISP implementation cannot operate at the speed required to implement the EPC communication protocol. The microcontroller can also be run at 1.8V, but at slower clock speeds. We believe that incremental improvements to the system design may enable the device to run at these lower voltages. The thresholds help illustrate the range improvements that can be expected as operating voltage requirements are dropped.

resistors model different power consumption "workloads." As we will explain later, the 10K value corresponds most closely to our microcontroller running at full speed, though in fact the microcontroller consume more power than a 10K resistor.

When the WISP is running at full speed (i.e. running the EPC communication code), it consumes 900 μ A at 3V, which can be modeled by a load resistance of 3.3K. In sleep mode it consumes 5 μ A at 3V. The power harvesting performance plot of figure 5 shows that at 3 feet, the harvester outputs less than 3V with a 10K load; with a 100K load, the harvester outputs more than 3V.

Thus the harvester does not produce enough power to run the WISP continuously. By duty-cycling the device, we can reduce the net power consumption so that it can be run steady state. We programmed the WISP to collect a packet of sensor measurements, compute the CRC, and then transmit this data repeatedly when polled for

0.25s, and then sleep for 2s before resuming the cycle. This is a duty cycle of 1/9, which can be mapped into a load resistance of $9 \times 3.3K = 30K$. When the WISP is asleep, the harvester accumulates power.

To demonstrate steady-state operation for a long period of time, we mounted the WISP to an exterior window using a suction cup, and placed an Alien RFID reader about 3 ft away from the WISP. We collected data using the Verify ID command as fast as the system would allow (about 15 polls per second).

When the WISP is asleep, the reader returns “No tag detected.” When the WISP is awake, the reader usually receives a burst of ID read packets. When a non-trivial packet is received, the time and the packet contents are logged to disk.

Later, for analysis, we loaded the read log data into Matlab and recomputed the CRCs. According to the CRC, the packet error rate is about 0.26: about one quarter of the packets are bad. So, just as in the ordinary EPC ID reading case, the CRC is a very convenient, practical, and reliable error detection mechanism.

The raw communication rate of the WISP device was about 15 packets per second, each packet containing 64 bits, resulting in 960 bits per second throughput (uncorrected). Future versions of the EPC specification (especially the Class 1 Gen 2 protocols) promise much higher data rates.

4.1 Light Level Measurement Experiment

The WISP was mounted by suction cup to the inside surface of an exterior window in an office environment, pictured in figure 6. The light sensor was oriented inward. Figure 7 shows the light level measured by the WISP over a 13 hour period. The measurement was begun at 5:20 pm, and the initial part of the measurement shows the light level dropping due to the sunset. After sunset, the light level remains constant for about three hours. We believe that all the interior lights were on during this period. Then, we believe that the interior lights nearest the WISP were extinguished, with lights further away remaining on, for about one hour. Then, the light level drops to a minimum when all the remaining interior lights were extinguished. This low light level persists until sunrise begins the next morning.



Fig. 6. WISP mounted by suction cup to interior window for light level measurement experiment

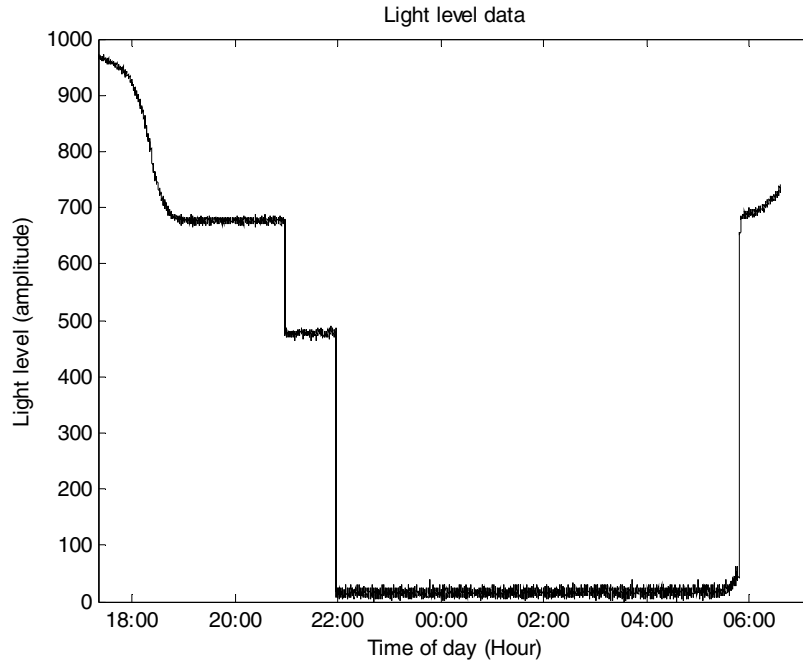


Fig. 7. Light level measured by WISP in a 13 hour period. The experiment began at about 5:40pm (17:40 hours). The first curve down is sunset. At around 9pm (21:00 hours), the measured light level drops substantially, probably because some lights in the laboratory were extinguished. At about 10pm (22:00 hours), the light level drops very low, probably because the remainder of the lights in the laboratory were extinguished. Just before 6am (06:00 hours), the beginning of sunrise is visible, and the laboratory lights turn on. The total number of measurement packets received was 115,413. Of these, 85,135 had valid CRCs. Only messages with valid CRCs are plotted above. This corresponds to a packet error rate of 0.26.

5 Discussion and Future Work

One of the primary challenges we are addressing next is improving the range of the WISP. There are two constraints that can prevent WISP operation: power and voltage. If either resource is insufficient, the WISP cannot function. As we have demonstrated with the duty cycling technique, the power constraint is relatively easy to work around: by dialing down the system's duty cycle, we can dial down the net power requirements as much as desired. It turns out that voltage is a harder constraint. If sufficient voltage is not available, the system cannot function at all, regardless of the duty cycle.

There are two basic ways to address the voltage constraint: lowering the voltage required by the microcontroller, and increasing the voltage delivered by the power harvester. We plan to make use of both approaches to improve read range.

To implement the EPC protocol in the “full software radio” fashion reported here, we had to run the microcontroller at 5MHz, which requires a 3V power supply. The microcontroller can also be run at 1.8V or 1.2V, at lower clock rates. By offloading into dedicated hardware key pieces of protocol logic that are now implemented in software (or possibly with improved firmware alone), we believe it will be possible to run the microcontroller at slower speeds. This should allow it to operate at 1.8V or 1.2V, which would increase our range substantially. Because the plot of voltage vs distance (figure 5) is relatively flat between 5 feet and 15 feet, we believe that small reductions in platform voltage requirements will deliver relatively large increases in range.

Improving the WISP antenna one obvious route for increasing the available voltage. In future work, we will investigate this and other approaches for increasing the voltage delivered by the power harvester, in order to improve the system’s range.

Once the WISP hardware is more mature (for example offering better range), we believe that a range of interesting systems questions can be addressed. Using the WISP’s on-board processing, data compression and streaming query functions can be implemented. Also, we believe that reader-WISP protocols that address identification and sensing jointly will be a rich area for future research. Current RFID protocols are designed to solve the problem of identification (i.e. singulation) only. The identification and sensing functions can be traded off in many different ways, and the desired trade off will be heavily application dependent. Thus the design of ID-sensing protocols, informed by application requirements, appears to be a promising area for future research.

6 Conclusion

We believe that the WISP platform represents an under-explored point in the sensor node design space with properties that differentiate it from other approaches. It requires no batteries, and no wires for power or data. The device appears to offer the promise of perpetual embedded wireless sensing, in appropriate usage model settings. A key constraint of the WISP system is the requirement of a reader. Nevertheless, we believe that many applications for this ubiquitous, battery-free sensing and computing platform exist. Since the constraints overlap substantially with those of conventional RFID, applications of conventional RFID are a natural place to begin seeking sensing applications of WISP.

Another category of applications we believe may be promising is hybridizing WISP with more conventional sensor nodes. WISP could be used in a battery-powered mode as a “backscatter modem” for a more conventional device. The power requirements of its backscatter radio are quite different than ordinary radios. Or, the power harvesting capability of WISP could be used to wake up more conventional and power hungry sensing and computing platforms.

As the WISP platform matures and its range increases, we are looking forward to exploring both “pure” perpetual sensing as well as “hybrid” applications of the WISP platform in future work.

References

1. S. S. Intille, K. Larson, E. Munguia Tapia, J.S. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research" in *Proceedings of PERVASIVE 2006*. Berlin Heidelberg: Springer-Verlag, 2006, to appear.
2. D.E. Culler and H. Mulder, "Smart Sensors to Network the World," *Scientific American*, June 2004, pp. 85–91.
3. S. Roundy et al., *Energy Scavenging for Wireless Sensor Networks*, Kluwer Academic Publishers, 2003.
4. J. Paradiso and M. Feldmeier, "A Compact, Wireless, Self-Powered Pushbutton Controller," *Proc. 3rd Int'l Conf. Ubiquitous Computing (UbiComp 2001)*, Springer-Verlag, 2001, pp. 299-304.
5. Webpages on the cavity resonator microphone can be found at <http://www.nsa.gov/museum/museu00029.cfm> and http://www.spybusters.com/Great_Seal_Bug.html.
6. R. Fletcher, *Low-Cost Electromagnetic Tagging: Design and Implementation*, PhD dissertation, MIT, 2001.
7. J. Paradiso, K. Hsiao, and A. Benbasat, "Tangible Music Interfaces Using Passive Magnetic Tags," *Proc. ACM Conf. Human Factors in Computing Systems: Special Workshop on New Interfaces for Musical Expression (CHI 2001)*, ACM Press, 2001.
8. K. Finkenzeller, *RFID Handbook*, 2nd ed., John Wiley & Sons, 2003.
9. http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class1.pdf
10. R. Want, "Enabling Ubiquitous Sensing with RFID." *Computer*, Vol. 37 No. 4, April 2004, pp. 84-86.
11. K. Opasjumruskit, T. Thanthipwan, O. Sathusen, P. Sirinamarattana, P. Gadmanee, E. Pootarapan, N. Wongkomet, A. Thanachayanont, M. Thamsirianunt, "Self-Powered Wireless Temperature Sensors Exploit RFID Technology," *IEEE Pervasive Computing Magazine*, January-March 2006 (Vol. 5, No. 1) pp. 54-61.
12. http://www.datasheetcatalog.com/datasheets_pdf/M/C/R/F/MCRF202.shtml
13. J.R. Smith, K.P. Fishkin, B. Jiang, A. Mamishev, M. Philipose, A.D. Rea, S. Roy, K. Sundara-Rajan, "RFID-based Techniques for Human-Activity Detection," *Communications of the ACM*, September 2005/Vol. 48, No. 9, pp. 39-44.
14. <http://focus.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=msp430f1232>
15. M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, and D. Hähnel, "Inferring Activities from Interactions with Objects," *IEEE Pervasive Computing*, vol. 3, no. 4, 2004, pp. 50–57.
16. <http://www.nfc-forum.org/home>