Moo: A Batteryless Computational RFID and Sensing Platform

Hong Zhang
zhangh@cs.umass.edu

Jeremy Gummeson
gummeson@cs.umass.edu

Benjamin Ransford
ransford@cs.umass.edu

Kevin Fu
kevinfu@cs.umass.edu

June 27, 2011

Abstract

The *UMass Moo* is a passively powered computational RFID that harvests RFID reader energy from the UHF band, communicates with an RFID reader, and processes data from its onboard sensors. Its function can be extended via its general-purpose I/Os, serial buses, and 12-bit ADC/DAC ports. Based on the Intel DL WISP (revision 4.1), the Moo provides an RFID-scale, reprogrammable, batteryless sensing platform. This report compares the Moo to its ancestor, documents our design decisions, and details the Moo's compatibility with other devices. It is meant to be a companion document for the open-source release of code and specifications for the Moo (revision 1.x). We made an initial batch of Moo 1.1 hardware available to other researchers in June 2011.

1 Introduction

Conventional RFID tags perform simple, hard-coded computations on harvested radio frequency (RF) power gathered from RFID readers. A class of devices called *computational RFIDs* [2] (CRFIDs) challenges the notion that only simple computations are possible under harvested RF power. Computational RFIDs feature reprogrammable microcontrollers, electronic sensors and actuators, nonvolatile memory, and small energy buffers that temporarily store harvested energy for computation.

The *UMass Moo* (hereafter: "Moo"), an open-source computational RFID under development at the University of Massachusetts Amherst, builds on the prototype Intel DL WISP (revision 4.1) [3] with additional memory, more pins for controlling sensors and actuators, and redesigned voltage regulation circuitry. Like a traditional RFID tag and like its predecessor the WISP, the Moo operates in the UHF band with a center frequency of 915 MHz. The communication protocol between the Moo and an RFID reader is EPC Class 1 Generation 2 protocol. A prototype Moo is pictured in Figure 1.

2 Why Build Moo?

The Moo's design is derived from the open-source specifications of the Intel DL WISP (revision 4.1) [3]. Our decision to design and build the Moo was inspired by a need for hardware that extended the capabilities of the DL WISP.

1. The TI MSP430F2132 microcontroller (MCU) on the DL WISP offers only 8 KB of flash memory, a considerable amount of which is program storage dedicated to the software implementation of the Class 1 Generation 2 protocol.

2. The external memory on the DL WISP is a low-speed EEPROM, namely a 128 KB Microchip 24AA1025 with a maximal clock frequency of 100 KHz (versus up to 16 MHz on the MCU). The EEPROM's speed becomes a bottleneck when the WISP is trying to save real-time sensor data.

3. Although there are a total of 23 ports for external devices on the DL WISP board, only 14 of them are GPIOs, including 9 ports shared with onboard devices. The remaining 5 GPIO ports can also be configured as 1 ADC, 1 3-wire serial (SPI, I2C, or UART) bus, and 1 SMCLK. There is therefore little space to extend the board to access more external memory, sensors, or other digital devices.

4. The DL WISP's MCU lacks an integrated DAC, and there is no onboard digital-to-analog converter (DAC) chip, which mean that the DL WISP lacks the ability to output an analog signal.

5. The DL WISP's Seiko S1000C20-N4T1G voltage detector is difficult to obtain. This chip is used as part of a supervisor that detects the output voltage of the DL WISP's energy harvester (transducer and rectifiers) and outputs a level indicating whether the voltage is above or below 2.0 V.

6. It is difficult to study the performance of energy harvesting or power consumption on the DL WISP, which makes tuning difficult.

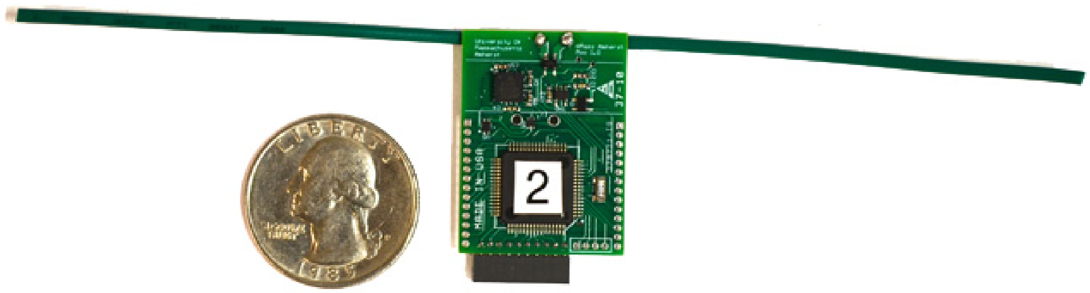

Figure 1: A fabricated UMass Moo, actual size. Pictured is a Moo version 1.0; the Moo 1.1 differs only slightly in appearance.

7. On the software side, a variety of firmware bugs limit the DL WISP's compatibility with readers other than the Impinj Speedway, such as the ThingMagic Mercury 5 (which cannot read the DL WISP's EPC).

3 Moo Improvements

Inspired by the limiting parameters of the DL WISP described in Section 2, the Moo aims to build on the experimental capabilities of the DL WISP. Revision 1.x[1] of the Moo makes the following improvements:

1. Upgraded microcontroller. The Moo's onboard MCU is a TI MSP430F2618. Compared to the DL WISP's TI MSP430F2132, the Moo's MCU offers: more RAM and on-chip flash memory; both ADC and DAC with higher precision; more I/O ports to connect external devices; and additional, faster clock speeds. Although they share an instruction set, the F2618 also reduces the number of cycles required for certain operations (e.g., 3 cycles for a register-to-memory move versus the F2132's 4 cycles). Table 1 compares Moo and DL WISP hardware at a high level.

2. The Moo has an onboard, off-chip flash memory that is faster and more capacious than the DL WISP's EEPROM. This external memory makes storing sensor data off-chip easier, freeing up more on-chip flash for program storage. Table 2 compares the off-chip memory capabilities of the Moo and the DL WISP.

3. The Moo's design considered the availability of parts, resulting in several design choices different from the DL WISP's. For example, the Moo uses a S1000C20-I4T1G voltage detector, which is in the same family as the corresponding part (S1000C20-N4T1G) on the DL WISP, but the former is much less expensive and easier to obtain.

4. To facilitate measurements of the electric current from energy harvesting, the Moo includes a 10 Ω current-sensing resistor at the harvester's outputs. The Moo optionally attaches (via two ports) to an external amplifier that magnifies the voltage drop over this sensing resistor.

5. The Moo's software implementation of the Class 1 Gen 2 protocol is compatible with more RFID readers in addition to the Impinj Speedway the DL WISP supports; for example, the Moo supports ThingMagic Mercury 5/5e readers.

Feature	**Moo**	**WISP**
Microcontroller	MSP430F2618	MSP430F2132
RAM	8 KB	512 B
Integrated flash	116 KB	8 KB
ADC	12-bit, 8-ch.	10-bit, 8-ch.
DAC	12-bit, 2-ch.	—
No. of ports	42	21
No. of GPIOs	31	14

Table 1: Side-by-side feature comparison between the Moo (revision 1.0) and the Intel DL WISP (revision 4.1).

Features	**Moo**	**WISP**
Part No.	SST25WF040	24AA1025
Type	Flash	EEPROM
Memory size	4 Mb	1Mb
Bus	SPI	I2C
Clock freq.	40 MHz	100 KHz
Standby current	2 μA	100 nA

Table 2: Comparison of the off-chip memories available on the Moo (revision 1.0) and the DL WISP (revision 4.1).

[1] As of this writing, $x \leq 1$

6. The Moo offers more physical board space: its board size is twice that of the DL WISP. Besides hosting the new MCU, the expanded board leaves space for more parts, such as a switching-regulator circuit for greater energy-conversion efficiency than the linear regulator the Moo and DL WISP both use.

Despite these improvements, the Moo's hardware modifications also import some drawbacks. The MCU, with its increased volatile memory capacity, draws more current than the DL WISP's (365 μA vs. 250 μA). The increased power consumption in some cases results in shorter read ranges or decreased read rates compared to the DL WISP.

4 Compatible RFID Readers

Initial tests have shown that the following readers are able to elicit query responses from the Moo[1]:

- ThingMagic Mercury 5e
- ThingMagic Mercury 5
- Impinj Speedway UHF Gen 2

5 Common Design Parameters

This section serves to document the design parameters of the Moo that also apply to the DL WISP. Note that 1 mil equals 1/1000 inch.

PCB layout rules and guidelines for low-frequency signals or digital I/O.

- 5x5 Routing
- 5 mil clearance
- 5 mil trace width (8 mil preferred)
- Via 8×12 (hole 8 mil, annular ring 12 mil)
- Prototron rules hole size plus 2 mil of annular on either side (total size: hole + 4 mil)
- More annular is needed for vias and probe points that will be repeatedly soldered and re-soldered
- The harvesting front end needs larger vias and careful routing

Pin map rules governing the use of pins on the Moo.

- Transmit: must be connected to TAO (timer A output), usually P1.1
- Receive: Code must be checked before this port can be changed; must be on a low value port number of port 1 (e.g., port 1.0 or 1.2); this has to do with the constant generator in the MSP430 and its ability to quickly reference this port
- Supervisor interrupt: Must be placed on port 2, so it does not collide with the read interrupt
- RTC interrupt: Must be on port 2 by the same logic

PCB fabrication rules governing the manufacture of the Moo.

- Material Type: FR4
- Finish thickness: 31 mil
- Copper Weight (inner): 1 oz
- Copper Weight (outer): 2 oz

6 Engineering Notes

6.1 Firmware Changes

The Moo 1.x firmware forked from the DL WISP 4.1 firmware in February 2011, though they remain largely the same. The key differences are as follows. These changes have already been implemented in the publicly available Moo firmware as of this writing.

- The Moo's use of a 12-bit ADC instead of the WISP's 10-bit ADC necessitates superficial changes to the sensing code (e.g., that which reads the accelerometer). Any corresponding reader-side demos that interpret ADC results (e.g., the "Saturn" demo originally developed for the WISP) require corresponding changes.
- The firmware of implementing partial C1G2 protocol should be modified to suit to new clock frequency and instruction cycle numbers
- The C1G2 protocol implementation in the Moo's firmware, which contains many carefully timed sequences of MSP430 assembly instructions, must be tuned to account for cycle-time differences mentioned in Section 3.

The firmware lacks fully integrated drivers for the Moo's expanded hardware (e.g., onboard flash memory and DAC).

6.2 Header Boards

The DL WISP uses a 4-pin header board for programming and debugging. However, the Moo's MSP430F2618 microcontroller lacks support for the Spy-Bi-Wire programming interface that enables the DL WISP's low pin count. The F2618 requires 5 pins for programming and two pins for power and ground (respectively).

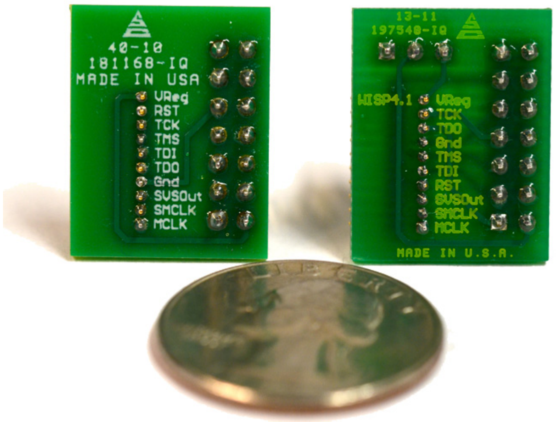

Figure 2: Header boards for the Moo 1.0 (left) and Moo 1.1 (right).

The Moo's header board, version 1.1, is compatible with the Moo 1.1 (or above) and DL WISP 4.1. This version of the header board is not compatible with Moo version 1.0 hardware because of layout changes. Version 1.1 of the Moo header board uses 10 pins. The first 4 pins can be used to program DL WISP 4.1; the first 7 pins can be used to program a Moo (version 1.1 or above), and the last 3 pins are left open as GPIO ports. Figure 2 shows detail.

The Moo 1.0 header board, also pictured in Figure 2, is not compatible with the DL WISP; it is compatible only with version 1.0 of the Moo.

A subtle fix to header pin layout is worth noting. The male pins of the DL WISP's header board extend from the PCB border, which makes them vulnerable to bending or breakage. The Moo's header board hides the male pins inside the PCB border to solve this problem.

6.3 Voltage Regulation

The DL WISP uses a 1.8 V linear voltage regulator that holds the microcontroller's voltage at or below 1.8 V. The linear voltage regulator performs poorly when the energy harvesting front end is performing well: while the regulator's efficiency is nearly 90% at 2.0 V, at 5.6 V its efficiency falls to 33%. Substituting a switching regulator for the linear regulator would result in higher conversion efficiency at some voltages; a typical switching regulator offers uniform efficiency (85% is common) over its entire input range.

In future work, we will present experimental designs to compare linear and switching voltage regulation in practical applications.

7 Availability

Moo firmware can be found via the Moo webpage, http://spqr.cs.umass.edu/moo. A compatible reader-side application, derived from code developed for the WISP, is available from the same place.

References

[1] A. Molina-Markham. Personal communication, 2011.

[2] B. Ransford, S. Clark, M. Salajegheh, and K. Fu. Getting things done on computational RFIDs with energy-aware checkpointing and voltage-aware scheduling. In *USENIX Workshop on Power Aware Computing and Systems (HotPower '08)*, Dec. 2008.

[3] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev, and J. R. Smith. Design of an RFID-based battery-free programmable sensing platform. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2608–2615, Nov. 2008.