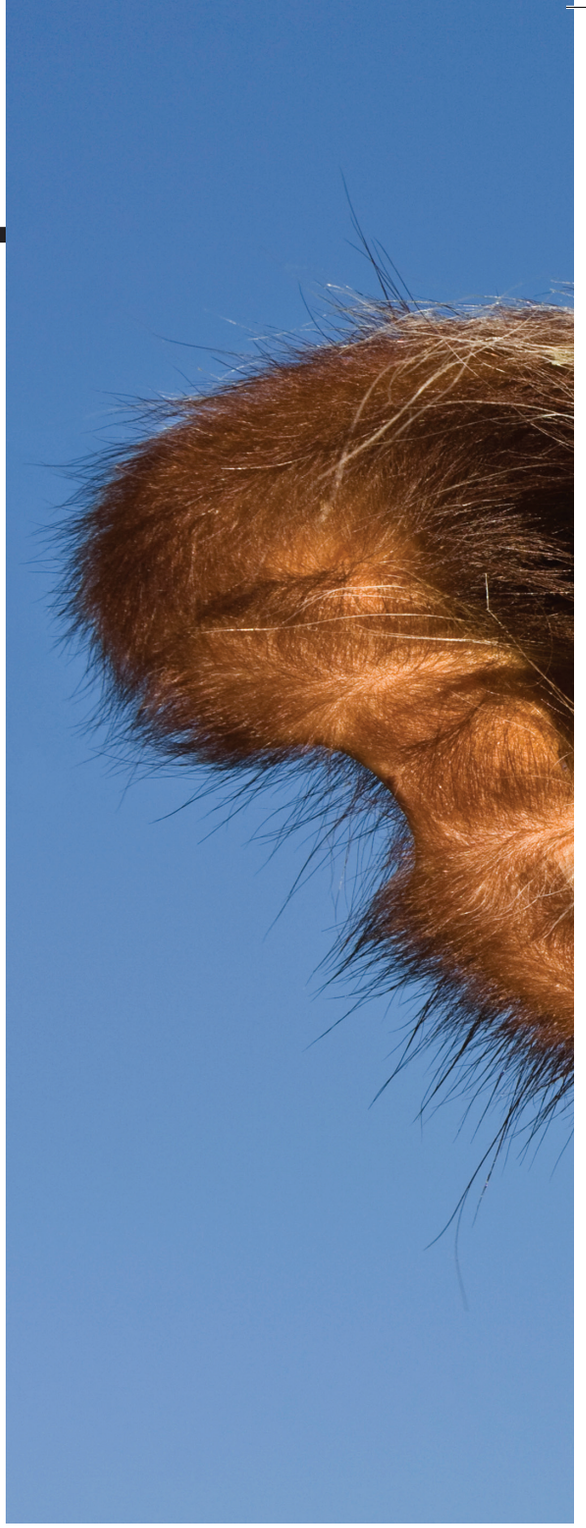
Chenren Xu *Peking University* **Pengyu Zhang** *Alibaba Group*

Editor: Ardalan Amiri Sani

OPEN-SOURCE SOFTWARE AND HARDWARE PLATFORMS FOR BUILDING BACKSCATTER SYSTEMS



In this article, we will provide a brief introduction of available open-source backscatter platforms that can be used by researchers and engineers to build interesting applications or investigate novel backscatter communication technologies. Instead of diving into the details of each platform, we will focus our discussion on the principles used in designing each platform and the unique feature provided by each platform. At the end of this article, we will discuss several key hardware and software modules that can be used by researchers or engineers to build their own backscatter platforms.

The wireless subsystems, such as Wi-Fi and Bluetooth, have long been considered as the bottleneck of reducing the power consumption of an IoT device [9]. The wireless subsystem consumes lots of power because it has to generate wireless signals for communicating its data. Generating a wireless radio wave is very expensive in terms of energy because it involves power-hungry RF analog modules as well as complicated digital signal processing.

Knowing the disadvantages of traditional wireless radios, backscatter radios take an alternative approach to communicating data. Instead of generating a wireless radio wave, they communicate the data over the air by reflecting radio waves produced by another device, named the *excitation signal generator*. The most successful backscatter communication system deployed today is RFID, shown in Figure 1, which is widely used in supply chain monitoring and tracking. In an RFID system, an RFID reader provides a continuous carrier wave, which is reflected by low-power RFID tags to embed their own information. In this article, we will first discuss several open-source platforms that can be used for understanding how commercial RFID systems work and how to improve the performance of the state-of-the-art RFID systems in "EPC Gen2 RFID Reader and Tags" and "NFC-WISP: 13.5 MHz NFC Tags."

Photo, istockphoto.com

RFID is primarily used for identification and tracking. Therefore, only a small amount of static data is transferred from RFID tags to an RFID reader. Since the power consumption of backscatter radio is so low, researchers and engineers started exploring how to use backscatter for communicating a large amount of data. In this article, we will also discuss several open-source platforms that target the high data rate and low-power wireless data communication in "Programmable Wi-Fi/Bluetooth Tags." At the end of our discussion, we will briefly outline several key hardware and software modules needed for building a new backscatter platform.

EPC GEN2 RFID READER AND TAGS

EPC Gen2 [1] is the protocol used for the communication between commercial RFID readers and tags. Millions of EPC Gen2 RFID readers and tags are fabricated each year and widely deployed to support various real-world applications. We provide a brief summary of software and hardware platforms available for experimenting with the EPC Gen2 protocol.

Commercial RFID Reader: Impinj Speedway R420: As shown in Figure 2, Impinj Speedway R420 RAIN RFID reader [2] is a commercial RFID reader that can communicate with up to four antennas.

Impinj has heavily optimized the RF analog front end of this reader. As a result, this reader can read tags at a speed of 200 tags/second and at a distance of more than 10 meters. However, Impinj does not optimize the control of this RFID reader because the control is closely tied with applications.

A user can control the R420 RFID reader via the Low-Level Reader Protocol (LLRP). We can leverage LLRP to dynamically change the transmission power, channel hopping patterns, bit rates and other parameters of the RFID reader. One example of this is the Blink system, [3] which adapts the bitrates used by the RFID reader according to the environment.

Special information offered by this RFID reader is the phase of the backscattered signal. Such phase information is not widely available on other commercial RFID readers. Phase is especially useful for identifying the location of tags or tracking the movement of RFID tags.

When a tag moves a bit, even by one centimeter, the RFID reader can observe a significant change on the phase of the backscattered signal. Therefore, the phase is an important metric that indicates the movement of tags. Tagoram [4] and other systems are built based on the phase information and achieve accurate tag localization and movement tracking.

Software Defined Radio Based RFID Reader: As shown in Figure 3, USRP-based EPC Gen2 RFID reader [5] is a programmable RFID reader implemented on USRP SDR for rapid prototyping. Different from commercial RFID readers, which only provide limited interfaces to users, the PHY, link and MAC layers of the USRP-based reader can be reprogrammed and tailored to meet the requirement of users. Using the default software, we can use this reader to read commercial EPC Gen 2 tags. However, because the MAC and PHY layers are implemented in the software on the host computer, the signal processing latency is much higher than commercial RFID readers. Therefore, when modifying the software on the host computer, we have to be careful about violating the timing specifications of the EPC Gen protocol.

The initial prototype was built using the USRP N210 motherboard and two RFX900 daughterboards. Two daughterboards are used for isolating the transmission chain and the reception chain and avoiding self-interference. Unfortunately, RFX900 daughterboards are not available on the market now. We can use Ettus SBX daughterboards to build the hardware platform because the SBX daughterboard works in the 915 MHz.

Using the programmable reader and the programmable tags, researchers can build a quick prototype and test new protocols. The Zero-One Estimator (ZOE) protocol [5] is built with such a programmable reader and tags and is designed for fast RFID cardinality estimation in large-scale RFID systems. It only requires a one-bit response from the RFID tags per estimation round, while prior works require several time slots during RFID cardinality estimation.

FIGURE 1. RFID operates in low power and even battery-free mode.

FIGURE 2. Impinj Speedway R420 RAIN RFID reader.

FIGURE 3. USRP-based EPC Gen2 RFID reader.

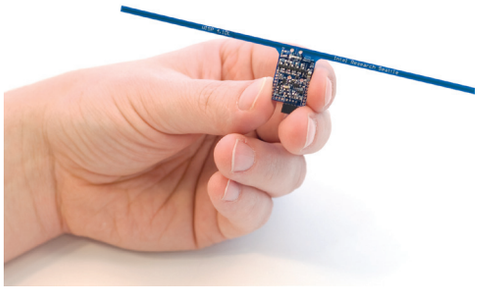


FIGURE 4. Wireless Identification and Sensing Platform (WISP).

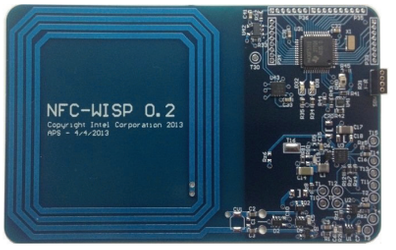


FIGURE 5. NFC-WISP platform.

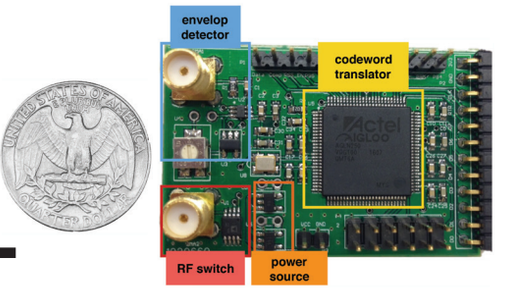


FIGURE 6. HitchHike backscatter platform.

WISP: Programmable EPC Gen2 RFID Tags: As shown in Figure 4, the wireless identification and sensing platform (WISP) project [6] is the first open source far field RF-powered sensing and computing platform. A typical WISP tag mainly consists of an RF analog front end and an ultra-low power 16-bit MSP430 microcontroller, which runs the EPC Gen 2 protocol. The RF analog front end is used to harvest power from the carrier wave produced by the reader and backscatter data on the tag-to-reader link. Current WISP firmware (WISP 5) is designed to be compatible with commercial RFID readers, such as Impinj Speedway readers.

Unlike the WISP, commercial RFID tags are black boxes that cannot execute arbitrary computer programs and do not support sensors. With the programmable microcontroller, the EPC Gen2 protocol can be implemented and extended on WISP, which has enabled many applications in the areas of sensing, cryptography, and security. For example, the ZOE [5] tags are implemented based on the WISP 4.1 hardware and firmware to support ZOE protocol for fast RFID cardinality estimation in large-scale RFID systems.

The WISP tags have some limitations compared to commercial RFID tags. Constrained by the RF analog front end, the WISP tag can only operate in the 915 MHz ISM band while different commercial RFID tags can work on LF, HF and UHF bands. In addition, due to a small amount of production, the WISP tags are more expensive than traditional commercial RFID tags. They also have shorter operational ranges compared to commercial tags.

NFC-WISP: 13.5 MHZ NFC TAGS

In this section, we will discuss an open source NFC backscatter platform [7] shown in Figure 5 that operates on 13.56 MHz. With the increasing availability of NFC enabled smartphones, we have the opportunities to develop new tag hardware with enhanced capabilities to explore new NFC sensors and human interface applications. NFC-WISP is a fully programmable sensing and computing platform that allows researchers to rapidly explore new NFC related applications. The NFC-WISP tag can be fully powered and read by commercial NFC readers (including NFC-enabled smartphones) and can support up to 106 kbps data communication rate of the ISO 14443 protocol.

Limited by the inherent characteristics of the NFC technology, the read range of NFC-WISP tags is typically on the order of 1 cm to 10 cm. However, thanks to the short read range, NFC-WISP tags are able to harvest relatively a large amount of power through inductive coupling compared to UHF-WISP tags.

The NFC-WISP tag has an ultra-low power display, large data storage, and computing and sensing capability. It has integrated temperature and acceleration sensors, 2 MB of FRAM, and an optional 2.7" active bistable matrix E-ink display. Moreover, a user can create customized applications by accessing the microcontroller through extension headers.

Photo (this page), istockphoto.com

However, the performance of the NFC-WISP tag is not only range- and alignment-sensitive but is also a function of its run-time load impedance. To balance the trade-off between power delivery and communication performance, a new version (NFC-WISP 2.0) has been released recently. This platform has a configurable multi-coil receiver antenna with two different configurations (2-coil/3-coil), which can dynamically adapt to the requirements of varied range, alignment and load impedance in real-time. Moreover, it is able to tolerate large load variation and antenna alignment. As a result, the operation distance is improved from 0.5 cm to 1.5 cm, and two times more power is delivered to the tag compared to the previous version.

PROGRAMMABLE WI-FI/ BLUETOOTH TAGS

In this section, we provide a brief overview of backscatter platforms that operate on 2.4 GHz and 5 GHz ISM bands. As far as we know, HitchHike [8] shown in Figure 6 is the only open source platform that operates on the two bands. Although WISP is programmable, we cannot use it to build 2.4 GHz and 5 GHz backscatter systems because the WISP RF analog front end is designed for operating around 915 MHz. As a result, WISP's performance is poor when operating at 2.4 GHz and 5 GHz.

Having seen the limitations of the WISP platform, HitchHike is designed to operate across various frequencies. Such flexibility is achieved by decoupling the RF analog front end and the digital logic. In other words, the RF analog front end can be tuned to operate at different frequencies by replacing the tag antenna, while the digital logic remains the same. In order to support flexible RF analog front end without changing the hardware components on the tag PCB board, HitchHike cuts the energy harvesting engine, which heavily depends on the RF frequency. Therefore, we need to use a small battery to drive the HitchHike platform. Compared to WISP, HitchHike obtains the flexibility of operating across frequencies but sacrifices the onboard energy harvesting engine.

A key feature of the HitchHike platform is the low-power FPGA that drives the tag's digital logic. HitchHike uses a tiny FPGA as its core processor because it has to meet

THIS DISCUSSION PROVIDES A SNAPSHOT OF THE CURRENT STATUS OF BACKSCATTER RESEARCH/SYSTEMS

the tight timing needed for backscattering with Wi-Fi. MCUs are usually too slow and suffer from timing jitters. The Igloo Nano FPGA used by the HitchHike platform is low-power and strong enough for implementing most of the functions needed for backscattering with Wi-Fi.

HitchHike is designed for backscattering with commercial Wi-Fi devices without changing the hardware of Wi-Fi. Therefore, software tools are available for receiving the backscattered Wi-Fi packets and extracting the tag information. Despite that HitchHike is designed for backscattering with Wi-Fi, researchers have used this platform for running backscatter with other radios, such as Bluetooth [10].

KEY MODULES NEEDED TO BUILD A BACKSCATTER PLATFORM

Instead of introducing platforms, in this section, we will summarize several key modules needed for building your own backscatter communication platform.

Hardware: The most important part of a backscatter hardware is the analog front end, which reflects the excitation wireless signals and harvests the energy from the excitation signals. The analog front end should be tuned in a way such that the reflected signal strength is strong and the harvested energy is sufficient. One trick for designing the analog front end is that before laying out the design on PCBs, we can use circuit simulator tools to fine-tune the parameters and understand the performance of the circuits.

Another important part is the low-power processor, which needs to execute backscatter functions with precise timing and run in a low-power fashion. We prefer using a small and low-power FPGA compared to an MCU because of its lower power consumption. However, FPGA is hard to program and prevent its use by other researchers.

Software: When using a software defined radio, such as USRP, to build an RFID reader, it is important to understand the timing constraints of Linux kernels because the RFID reader logic is usually hosted on a Linux computer. The scheduler and other factors in Linux cannot guarantee the timing for executing a task of the backscatter receiver. As a result, the timing of the communication protocol might be violated. To guarantee precise timing in backscatter data reception, we have to understand the timing constraints of Linux kernels and try to avoid timing violations as much as possible.

CONCLUSION

In summary, we briefly introduce several open source platforms that can be used by researchers and engineers to understand the performance of backscatter systems and build their own applications. We hope this discussion provides a snapshot of the current status of backscatter research/systems and encourages more researchers and engineers to explore interesting solutions in this area. ■

Chenren Xu is an assistant professor at the School of Electronics Engineering and Computer Science at Peking University. He received his BE in Automation from Shanghai University, his MS and PhD in Computer Engineering from WINLAB, Rutgers University. His research interests span wireless, networking and systems, with a focus on visible light backscatter communication for IoT and V2X applications, and future mobile Internet architecture for high mobility data networking.

Pengyu Zhang is a staff engineer at the Alibaba Group Seattle. He received his PhD from University of Massachusetts Amherst and his bachelor's degree from Tsinghua University. Before joining Alibaba, he was a postdoc at Stanford University. He designs and builds embedded systems and wireless systems for IoT applications.

REFERENCES

[1] https://www.gs1.org/standards/epc-rfid/uhf-air-interface-protocol
[2] https://www.impinj.com/platform/connectivity/speedway-r420/
[3] https://github.com/pengyuzhang/Blink
[4] https://github.com/tagsys/tagsee
[5] https://github.com/yqzheng/usrp2reader
[6] https://github.com/wisp/wisp5
[7] https://github.com/wisp/nfc-wisp-hw
[8] https://github.com/pengyuzhang/HitchHike
[9] "Energy-efficiency in wireless sensor networks," Tifenn Rault, 2015
[10] FreeRider: Backscatter Communication Using Commodity Radios, CoNEXT 2017