# A Multi-Protocol Baseband Modem Processor for a Mobile RFID Reader

Seok Joong Hwang[1], Joon Goo Lee[1], Seon Wook Kim[1], Sunshin Ahn[1], Si-Gyung Koo[2], Jihun Koo[2], Kyung Ho Park[2], Woo Shik Kang[2]

[1]Compiler and Embedded System Laboratory
Department of Electronics and Computer Engineering
Korea University, Seoul, Korea
{nzthing,nextia9,seon,sunshin}@korea.ac.kr
[2]Communication and Network Laboratory
Samsung Advanced Institute of Technology, Korea
{sg.koo,jihun.koo,pkh5247,wskang}@samsung.com

**Abstract.** In RFID (Radio Frequency IDentification) systems, a tag reader communicates with tags, reads their identification codes, and accesses their related database through network infrastructure. There are many research activities in RFID systems for industrial applications such as delivery, manufacturing, and so on, but there is few on mobile devices such as cellular phones and PDAs. This paper presents architecutre of a multi-protocol RFID reader on mobile devices. We have considered several design parameters, such as low power consumption, cost effectiveness and flexibility. We prototyped our system on the ARM-based Excalibur FPGA with iPAQ PDA, and also a chip with 0.18um technology for verification of our architecture.

**Keywords:** RFID, Mobile, WIPI, HAL, Multi-protocol, Baseband Modem

## 1 Introduction

Nowadays we are thinking about assigning a unique identification and its associated information to an object for a ubiquitous computing environment. To realize a ubiquitous world (u-world) which completes the ubiquitous computing and networking environment, we need to prepare for smart networking infrastructures and instinctive information capturing systems. In the u-world, huge amounts of connected computing devices provide a convenience environment to us. There are lots of activities to support network infrastructures to offer for convenience to people by using many computing devices seamlessly.

One of the potential and attractive ideas to support the u-world is to use RFID (Radio Frequency IDentification) systems. In RFID systems, we attach a tag onto an object where a unique identification code is stored in the tag. A tag reader communicates with the tag by a radio frequency signal, reads the code from the tag, and accesses its related database through network infrastructure

for getting more information. Figure 1 shows the environment especially for mobile-based RFID systems. the RFID tag is attached to a statue as an object and the tag's information is read by RFID readers on mobile devices. We can get the detailed information (name, description, and so on) about the statue by accessing Internet.

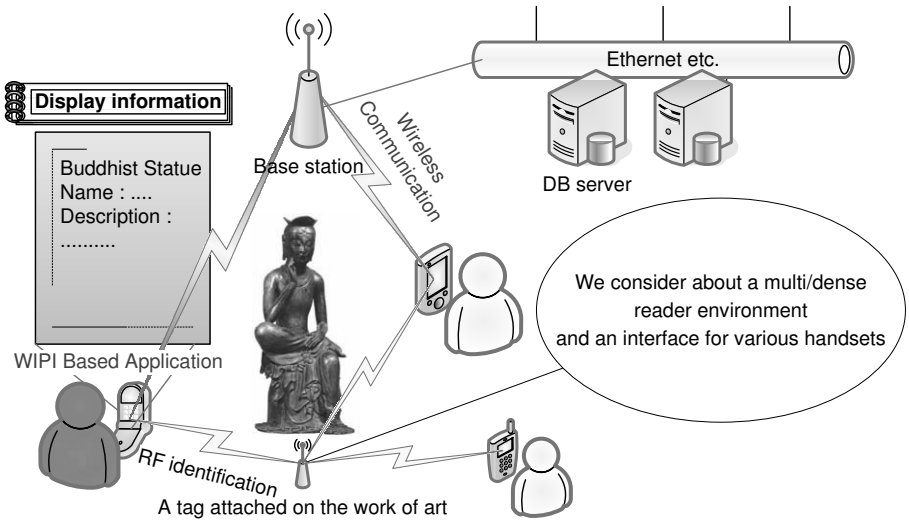


**Fig. 1.** Example of a mobile RFID environment.

In general, an industrial reader system is widely available in delivery and manufacturing services. The whole system is highly optimized for accuracy and speed performance by locating each reader at fixed and optimized location. Also, the size of the industrial reader is large, since there is no resource constraint such as power consumption and area. The RFID system in the mobile environment has some different scenarios from the industrial one. Because the reader is carried anywhere and anytime, We should consider several design parameters, such as low power consumption, cost effectiveness and flexibility. The resource optimization is very important in the mobile environment, and flexibility is required for supporting the wide scope of mobile devices and their applications.

In this paper, we present architecture of a multi-protocol RFID reader on mobile devices such as mobile phones and PDAs. We have prototyped our system on the ARM-based Excalibur FPGA [1] with iPAQ PDA, and also we have fabricated a chip with 0.18um technology for verification of our architecture.

The paper consists of the followings: In Section 2, we present an overview of our multi-protocol RFID readers architecture. We present the hardware architecture in Section 3 in detail. The experimental design is discussed in Section 4, and finally the conclusion is made in Section 5.

## 2 ARCHITECTURE OVERVIEW

Figure 2 shows an overall software and hardware organization of a multi-protocol RFID reader for our system. The RFID reader consists of tightly integrated four parts, a RF circuit, baseband hardware logics, RFID software components, and a handset.

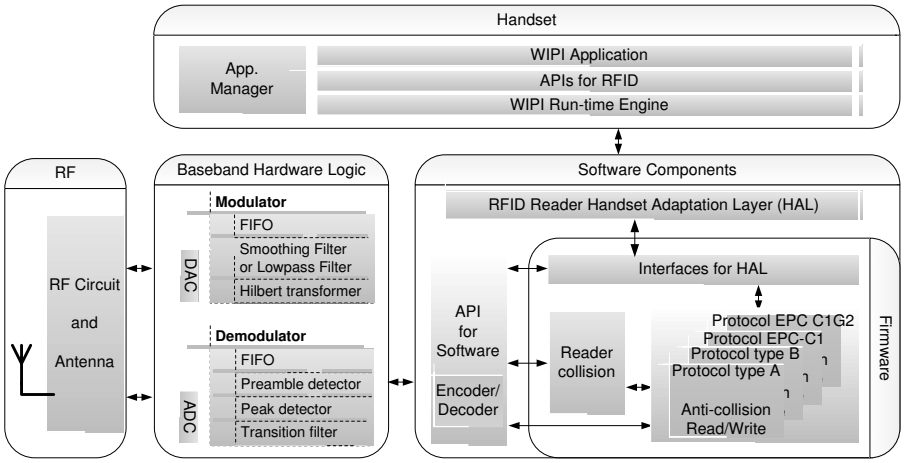


**Fig. 2.** Software and hardware organization of our mobile RFID reader.

In a physical layer of RFID systems, ASK modulation is common to all the passive RFID protocols [2–5], because it is easy to implement ASK modulation in a passive tag which has cost and power concerns. But, ASK signal is poor at performance than other modulation schemes. Thus our modulator and demodulator were carefully designed to manipulate this signal efficiently. And they were implemented in hardware logics as they requires higher performance than other workloads, moreover there is hard real-time constraint. As a result , we were able to implement most of other functionalities of a RFID reader in software to support multi-protocols and the next coming standards easily at lower operating frequency.

The RFID software components consist of baseband modem APIs, a firmware, HAL interfaces, and WIPI-based applications. And these components are running on a general purpose processor (ARM9) . The baseband modem APIs provide interfaces to access baseband modem hardware logics and RF circuits for controlling modem hardware. The firmware includes anti-collision algorithms, reader collision algorithms, interfaces with handset adaptation layer (HAL), and so on. The HAL is an abstract layer to provide hardware independence to an application. WIPI (Wireless Internet Platform for Interoperability) [6] is the common platform envisioned by the Korean Ministry of Information and Com-

munication for running mobile applications on any handset independent to vendors. WIPI serves as a backbone for content providers to develop applications that run seamlessly on any mobile platform. Any WIPI-based application can access and control a RFID reader through HAL interfaces in a uniform manner. A thorough description of the detail software architecture is provided by Lee et al [7].

In the following section, we will present the hardware architectures in detail.

# 3 HARDWARE ARCHITECTURE

To reduce performance burden for the general purpose processor, the modulator and the demodulator were designed as hardware logics. The modulator was designed to use a frequency spectrum efficiently and to give flexibility in real environments. And the demodulator was focused on low power consumption and cost effectiveness as well as maintaining acceptable performance on our environment. In this section, the architecture of the hardware is explained.

## 3.1 Modulator

Concerning multiple reader environments, each nation has its own regulation of frequency resource usage. Thus it is hard to make a universal transmission filter. And due to a strict transmission mask [5], it is difficult to achieve maximum transmission speed which is specified in each RFID standard.

In order to resolve the problem, our modulation filters were designed to support SSB (Single Side Band) transmission to use frequency spectrum efficiently in addition to DSB (Double Side Band) transmission, and all filters were design to be programmable for easy adaptation in real environments. The transmitted data on a forward link are written into FIFO. They are converted to rectangular pulses by a pulse generator, and then filtered by a 16-tap FIR pulse shaping filter. In the DSB transmission, the pulse shaped signal is transmitted to I channel and Q channel, which is tied to zero. In the SSB transmission, I and Q channel signals are generated by the Hilbert transformer, which was implemented with a 31-tap FIR filter, from the pulse shaped signal.

**Table 1.** Filter configurations for each standard.

| Standards | Forward link | | Filter configurations | | | |
|---|---|---|---|---|---|---|
| | Bit rate | Link frequency | Type | Sampling frequency | Passband frequency | Stopband frequency |
| EPC C1 | 15 Kbps | 120 kHz | SSB | 1200 kHz | 120 kHz | 180 kHz |
| ISO Type B | 40 Kbps | 40 kHz | DSB | 400 kHz | 80 kHz | 100 kHz |
| EPC C1G2 | 80 Kbps | 80 kHz | DSB | 400 kHz | 80 kHz | 100 kHz |
| | 80 Kbps | 80 kHz | SSB | 800 kHz | 80 kHz | 110 kHz |

Table 1 shows the maximum transmission speeds of each standard with our filter, and each filter configuration satisfies with the Korean RFID regulation. Resulting signal spectrum satisfies with transmission mask for the first neighbor channel where 20dB attenuation is occurred in 200 kHz channel bandwidth which is allowed in the Korean regulation.

### 3.2 Demodulator

The acquisition of a signal in the mobile-RFID system introduces several design difficulties. First, since a mobile system requires low power consumption and cost effective design, we chose a direct conversion and single antenna architecture. While the direct conversion architecture has a merit for reduction system cost compared to devices with IF-based architectures, it brings about various intermodulations and spurious signals on account of local oscillator and self-mixing interference leakage [8–11]. These are sources of DC offset fluctuation which is difficult to be eliminated, and these sources degrade overall system performance. Moreover, as both a forward link and a return link use the same frequency band with a shared single antenna, leakage power of the transmission signal is about 30dB larger than power of the receiving signal in our system. Thus the situation is deteriorated. There are several kinds of compensation techniques which are complex or expensive.

Second, the oscillator frequency of a passive tag is inaccurate. According to the protocol in [5], the reader should be working to a tolerance ±22% in a receiving link frequency. Conventional receivers estimate the signal by using a matched filter (MF). It has advantages that it results in a correlation gain by integrating the received signal over the symbol period $T$ while averaging out the zero-mean AWGN. It is known as an optimal receiver which maximizes SNR as described in Equation (1) [12]. But it is effective when there is only AWGN in noise components. With considering other noise components such as DC offset fluctuation, we can express the SNR as Equation (2).

$$SNR_{max} = \frac{2E}{N_0} \tag{1}$$

$$(\frac{S}{N})_T = \frac{|\int_{-\infty}^{\infty} H(f)S(f)e^{j2\pi fT}df|^2}{\frac{N_0}{2\int_{-\infty}^{\infty}|H(f)|^2df} + |\int_{-\infty}^{\infty} H(f)N_{Drift}(f)e^{j2\pi fT}df|^2} \tag{2}$$

If we constraint the numerator to be 1, the problem of maximizing SNR is reduced to minimizing the denominator. The second component of the denominator can be amplified if there is correlation gain between an impulse response of the MF and one of DC offset fluctuation. It means that the maximum performance of the receiver cannot be obtained by the MF if there is no additional DC offset compensation technique.

There are three kinds of solutions. One is to suppress the leakages using bandpass filtering in a carrier frequency level. It has a limitation since the frequencies are same between expected received signal and leakage components. Another

technique [8, 10, 11] is to compensate the DC offset fluctuation by tracking and estimating it, or by bandpass filtering in a baseband level. This approach requires exquisite control with additional system resources or it maybe impractical for ASK signal. The third technique is to use efficient coding rules which have closely zero correlation gain with DC offset fluctuation ("dc-free" coding rule). But it cannot be adopted in the RFID system, because most standards do not establish coding rules which provide such a property.

Thus we developed a cost effective receiver architecture with good immunity from DC offset fluctuation as well as good feasibility. This architecture is called *a transition trigger*. The transition trigger architecture focuses on providing robust performance in presence of DC offset fluctuation and supporting every coding rule in the RFID standards. While a conventional MF recovers a symbol by estimating a maximum likelihood level of a symbol, the transition trigger recovers a symbol by triggering transitions caused by the symbol. The transition trigger consists of three major components which are a transition filter, a peak detector and a bit slicer.

The transition filter is an alternative of the conventional MF. It estimates a transition component of a received signal. By using the estimation, we can determine whether transition occurred or not. While impulse and frequency responses of the conventional MF are expressed as Equation (3), (4) [12], responses of the transition filter are defined as Equation (5), (6).

$$h(t) = u(t) - u(t-T) \quad for\ 0 \le t \le T \tag{3}$$

$$H(f) = Tsinc(fT)e^{-j\pi fT} \tag{4}$$

$$h(t) = u(t) - 2u(t-T) + u(t-2T) \quad for\ 0 \le t \le 2T \tag{5}$$

$$H(f) = Tsinc(fT)(e^{-j\pi fT} - e^{-3j\pi fT}) \tag{6}$$

Figure 3 shows the frequency response of the filter which suppresses the DC offset fluctuation and maximizes the transition component of a source signal efficiently. Conceptually, we can recover a symbol using this filter if the timing of a received signal is known already. In this case, we can use a simple comparator to determine symbol values. When the output of the transition filter is greater than a certain threshold at the switching point between symbols, the output of the comparator is inverted. If not, the output is kept. To evaluate the immunity of the transition filter compared to the conventional MF, we simulated BER performance of both filters with the DC offset fluctuation in addition to AWGN. The fluctuation is modeled simply as a low frequency continuous wave. Figure 4 shows the simulation data. The 1000 bits FM0 coded tag responses were received 100 times. The link frequency of responses was 80 kHz, a sampling frequency was 640 kHz and the amplitude of the fluctuation was equal to symbol signals. In Figure 4 (a), the BER of the MF is more excellent than the transition filter in absence of the fluctuation. Because the MF is best able to average out the AWGN. But the performance of the transition filter is acceptable since we do not need extreme performance in our target environment. The required receiver

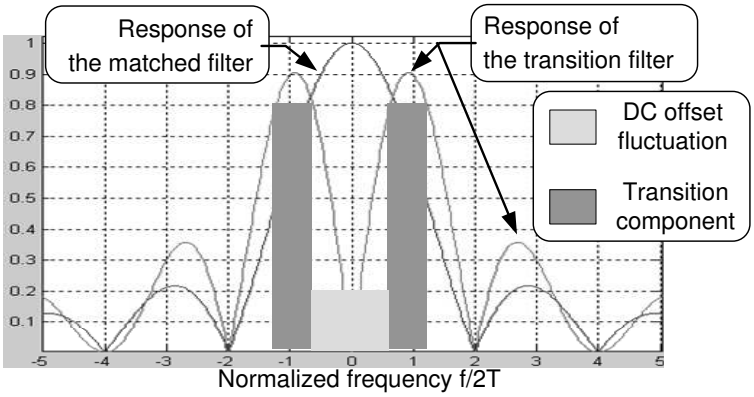


**Fig. 3.** Spectrum of the transition filter.

sensitivity is -96dBm. In this case, the output SNR of our RF circuit including ADC is 12.9dB. Thus our design targets on BER $10^{-3}$ in SNR 12.9dB. Besides the immunity of the fluctuation is more important factor. As shown in (b)∼(d) of Figure 4, while performance of the MF is degraded severely, the transition filter shows good immunity. It is important that it is done without any additional processing such as a bandpass filtering. Thus our design is cost effective as well as acceptable in a term of performance.

The peak detector finds points at which transition is occurred by tracking the output of the transition filter. Basically, it detects peaks by comparing a signal to the threshold which is adjusted dynamically and has two strategies. First, it recognizes a peak when a signal crosses zero level after it went across the threshold.

Second, even if a signal does not cross the threshold, the detector recognizes a peak when it goes across the opposite threshold level after zero-crossing. After recognizing a peak, the detector determines the peak point which has maximum correlation gain between a present point and a previous recognized point. And the detector updates the threshold to be half of the current peak value.

Using the information of peak positions, a bit slicer recovers symbols. Basically it determines symbols using peak polarities and distance between peak positions. For example, assuming a symbol has a period $T$, the slicer determines whether there is one symbol when peak-to-peak distance is closed to $T$ or not. However, it is a problem that the receiver does not know the period $T$ exactly. As previously mentioned, the oscillator frequency of a passive tag is inaccurate and the reader should be working to a tolerance $\pm 22\%$ in a receiving link frequency. It means that we cannot recover symbols using just a peak-to-peak distance. Thus we buffers the distances to obtain an actual link frequency. Our receiver has 7 bit-slicers which has different periods each other. Each slicer is connected with preamble detection buffers which detect a preamble from the received signal. If the preamble is detected from any preamble detection buffer, a bit-slicer selector

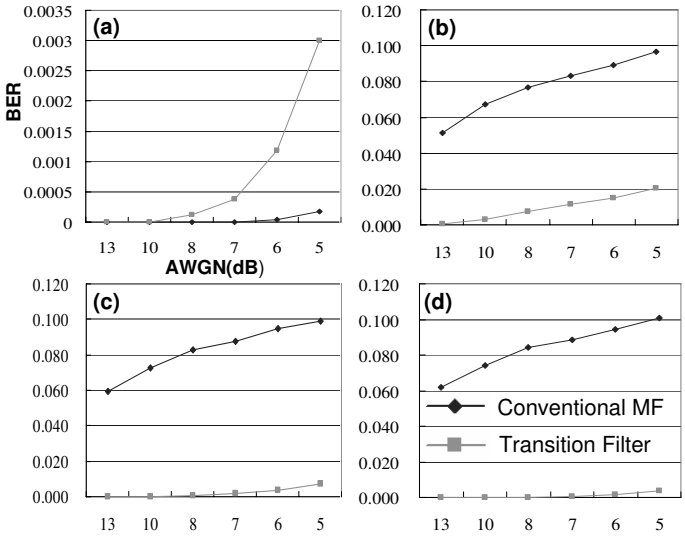


**Fig. 4.** BER performance in the DC offset fluctuation. DC offset fluctuation frequency: (a) No fluctuation (b) 16kHz (c) 8kHz (d) 4kHz.

determines one bit-slicer which is used for symbol recovery by calculating actual receiving link frequency using buffered peak distances.

Using the transition trigger architecture, we achieved the receiver performance as shown in Table 2. It was obtained from a simulation that received 160 bit tag responses of EPC class-1 Gen2 1000 times. As a result, our receiver is working to a tolerance ±22% at a receiving link frequency.

Until now, we present our hardware architecture. It has signal processing features to reduce performance burden for a general purpose processor. And it is focused on low power consumption and cost effectiveness as well as maintaining acceptable performance on our environment. In the following section, we present the experimental design for verification of our architecture.

**Table 2.** Receiver performance in AWGN.

| Timing Error | | SNR | | | | |
|---|---|---|---|---|---|---|
| | | 12.9dB | 10dB | 9dB | 8dB | 6dB |
| -22% | BER (Bit Error Rate) | 0.0 | 0.0 | 0.023 | 0.0131 | 0.1181 |
| | PER (Packet Error Rate) | 0.0 | 0.0 | 1.1% | 3.8% | 35.8% |
| 0% | BER | 0.0 | 0.0 | 0.0071 | 0.0045 | 0.0733 |
| | PER | 0.0 | 0.0 | 0.3% | 1.6% | 23.5% |
| +22% | BER | 0.0 | 0.0 | 0.001 | 0.0047 | 0.0802 |
| | PER | 0.0 | 0.0 | 0.3% | 2.6% | 32.6% |

## 4 EXPERIMENTAL DESIGN

We implemented our prototyped baseband modem logic on the ARM-based Excalibur FPGA, ultra high frequency (UHF) RF circuit with filters, direct conversion up/down mixers, a PLL, a power amplifier and a palm sized antenna. The baseband modulator and the demodulator were implemented in FPGA. And the RFID firmware is running on an ARM9 processor on the Excalibur.

The baseband modem APIs allow us to control forward/return link frequencies, transmitting/receiving bit streams, PLL, TX/RX gains, and turn on/off each RF part. Our firmware has functions of inventory using anti-collision algorithms, and also has read/write functions if a protocol supports. Also special functionalities for security such as kill or lock at EPC protocols are accomplished.

A HAL application was coded on a personal digital assistants (PDA) using a graphic user interface (GUI) based on PocketPC 2002. Using a universal asynchronous receiver/transmitter (UART), the HAL application on PDA and the RFID baseband modem were connected. When a user chooses a HAL command on GUI, the HAL application sends a message to UART with predefined format and waits for a response from the firmware. The firmware of the baseband modem receives this message from the HAL application and decodes the message. After decoding the message, the firmware calls and executes a suitable function in an algorithmic part or a baseband modem API. If the firmware executes the function successfully, it sends back a response message, also a predefined format, to the HAL application. The HAL application displays proper information according to the responded message.

Finally, we fabricated the chip version of the reader with 0.18 um technology. Figure 5 shows this chip which is implemented with about 80k gate baseband logic, ARM9 processor, ADC, DAC and SRAM. Die size of the chip is 5mm by 5.27mm. And the power consumption of the baseband logic is estimated to 21.705mW with 25MHz clock frequency, 30% Flip-flop ratio and 10% switching activity. It can be more reduced, because the necessary clock frequency is just about 4 MHz actually. But we use unified 25MHz clock frequency for the MCU and the baseband logic to eliminate an additional PLL. At last, we demonstrated our chip embedding to a PDA phone at 2005 USN/RFID Conference & Exhibition which was held in Korea.

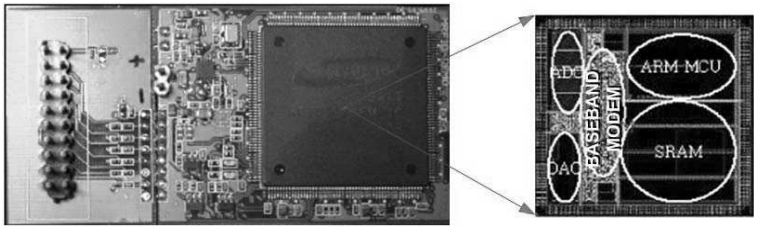


**Fig. 5.** Chip version of the multi-protocol RFID reader for mobile devices.

## 5 CONCLUSION

In this paper, we presented architecture and implementation of a multi-protocol RFID reader on mobile devices such as mobile phones and PDAs. There are many research activities in industrial RFID systems, but there is few on mobile devices. We focused on low power consumption, cost effectiveness and flexibility. Also, our firmware interacts with Handset Adaptation Layer (HAL) in order to support WIPI (Wireless Internet Platform for Interoperability) architecture on mobile devices. Any WIPI application can use our RFID reader's functionalities to query tags' information from Internet through HAL interfaces. We prototyped our system on the ARM-based Excalibur FPGA with iPAQ PDA, and also we fabricated a chip with 0.18um technology for verification of our architecture.

We have ported our implementation onto only software implementation written in Java. We have implemented our own Java core called TalusCore which can accelerate the Java applications in real time. It will support more flexibility and portability with extreme low power consumption.

## References


1. Altera Corperation: Excalibur devices. (http://www.altera.com/)
2. International Organization for Standardization: ISO/IEC FDIS 18000-6, ISO standard document. (http://www.iso.org)
3. EPCglobal Inc: EPC class 0, EPC global standard documents. (http://www.epcglobalinc.org/standards_technology/specification.html)
4. EPCglobal Inc: EPC class 1, EPC global standard documents. (http://www.epcglobalinc.org/standards_technology/specification.html)
5. EPCglobal Inc: EPC class 1 gen 2, EPC global standard documents. (http://www.epcglobalinc.org/standards_technology/specification.html)
6. The Korean Ministry of Information and Communication: Wireless Internet Platform for Interoperability (WIPI). (http://wipi.or.kr)
7. Lee, J.G., Hwang, S.J., Kim, S.W., Ahn, S., Park, K., Koo, J.H., Kang, W.S.: Software architecture for a multi-protocol rfid reader on mobile devices. In: Second International Conference on Embedded Software and Systems. (2005) 81–88
8. Abidi, A.A.: Direct-conversion radio transceivers for digital communications. IEEE journal of solid-state circuits **30** (1995) 1399–1410
9. Razavi, B.: Design considerations for direct-conversion receivers. IEEE transactions on Circuits and Systems II: Analog and Digital Signal Processing **44**(6) (1997) 428–435
10. Cavers, J.K., Liao, M.W.: Adaptive compensation for imbalance and offset losses in direct conversion transceivers. IEEE transactions on vehicular technology **42**(4) (1993) 581–588
11. Lehne, M., Stonick, J.T., Moon, U.: An adaptive offset cancellation mixer for direct conversion transceivers in 2.4ghz cmos. In: IEEE International Symposium on Circuits and Systems. (2000) I319–I322
12. Sklar, B.: Digital Communications Fundamentals and Applications, 2nd Edition. Prentice-Hall, Englewood Cliff, NJ (2001)