Applying Passive RFID System to Wireless Headphones for Extreme Low Power Consumption

Joon Goo Lee Dongha Jung Jiho Chu Seok Joong Hwang Jong Kook Kim Janam Ku Seon Wook Kim

Department of Electronics and Computer Engineering, Korea University, Seoul, Korea 136-701
+ Embedded Systems Solution Lab, Samsung Advanced Institute of Technology, Gyeonggi-Do, Korea

{nextia9,goodbye,malboro,nzthing,jongkook,seon}@korea.ac.kr + kujanam@samsung.com

ABSTRACT

One of major design concerns about wireless headphone is power consumption. In this paper, we propose a novel design for extreme low power headphone implementation by extending the EPC Class-1 Generation2 RFID protocol for delivering stream data. We prototyped a reader as a stream generator, and a passive tag as an audio receiver to consume less power than any other protocols for wireless headphones.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communication; C.3 [Special Purpose and Application-based Systems]: Real-time and embedded systems

General Terms

Design

Keywords

RFID, passive RFID, EGen2, low-power, wireless headphone

1. INTRODUCTION

In general, wireless communications can be categorized into two areas depending on the existence of central control units: an infrastructure-based and an infrastructure-less wireless communications. The protocols in the former category provide telecommunication services and/or IP-based Internet services for substituting currently existing wired services for conveniences. A cellular network and wireless LAN belong to this category. The infrastructure-based wireless communication provides good mobility, high data rate and bandwidth, long communication range, strong reliability and QoS, security and so on [1]. The latter approach, an infrastructure-less wireless communication, has been developed in order that a device sends and receives small data periodically, communicates with other devices at low data rate, and makes a network by connecting neighboring devices on low power architecture with a relatively simple communication protocol. The transferred data could be sensor information in WSN (Wireless Sensor Network), audio information for a wireless headset, an identification code for an RFID system, and so on. Figure 1 compares several well-known wireless communications in terms of protocols, applications, data rates and power consumption.

When designing a wireless communication device, the power consumption is the most critical design issue to developers because wireless devices have been limited by the battery power. Intrinsically, RF transceivers use huge amounts of power for transmitting and receiving data. Even a commercially available low power transceiver consumes more than 15 mW for reception and 20 mW for transmission [2]. Consequently, they often consume more than half of the total power dissipation for communication devices [3]. Also, there are several other sources of energy wastes which result from idle listening, over hearing, channel collision, over emitting.
and control packet overheads [4]. In terms of power consumption, an interesting approach has been developed, and it is a passive RFID technology. The passive RFID technology has been accepted widely in logistics and supply chains as an alternative of bar-code systems due to many advantages such as longer reading distance, faster recognition speed, larger memory capacity, non-line-of-sight, and reusability [5]. In passive RFID systems, a tag/transponder keeps an identification code and a small amount of information about the corresponding product inside its nonvolatile memory. A reader/interrogator reads the code or necessary information from tags using backscattering. By reading ID codes and information, the logistics system can track, manage, and manipulate versatile objects. Because the operational power for tags is delivered by a reader, tags do not include any battery.

Currently this RFID system has been only applied for automatic data acquisition methods, however it may be used for many other applications. For example, we can apply this RFID technology to a sound transmission system. Before applying our idea to implement a wireless headphone with the passive RFID system, we analyzed the feasibility of our idea. Because a tag is only powered by a reader, power constraint is a critical factor for our feasibility study. We used the equations from the Friis transmission equation [6] for the calculation of path loss, and the equation is given as the following:

\[
PL(dB) = \begin{cases} 
32 + 25 \log(D) & 0 \leq d < 8m \\
23 + 35 \log(D) & d \geq 8m 
\end{cases}
\] (1)

This equation is commonly used for the indoor modeling of 900MHz RFID systems.

Figure 2 shows the power budget for a tag, and it is computed with the following assumption: 1) a reader transmits 10 dBm power to tags, 2) the antenna gain of the reader is 6 dBi, 3) the antenna gain of a tag is zero, 4) the efficiency of a tag rectenna (rectifying antenna) is 80% [7], 5) The half of converted power from the rectenna is available to the digital parts of a tag. From the figure, we can see that under 3.37µW is available to the digital parts of a tag for a 1.8 meter working range. In other words, for a passive tag to operate at a distance of 1.8 meter, the power consumption of the digital parts should not be over 3.37µW.

We marked important application points on the figure, and these points represent available power and operating distance when we increase the operating clock frequency for higher forward data rate. Currently, the conventional Gen2 supports 128kbps forward and under 320kbps backward data rates at 1.28MHz operating frequency. Therefore, the Gen2 specification is suitable for voice communication. If we increase the operating frequency of the tag, it is possible to process also high quality audio data and even video stream.

In this paper, we propose a novel idea which guides a new way to use passive RFID systems and overcomes the power consumption problem in a specific wireless application, i.e. an extreme low power wireless headphone. We extended EPC Class-1 Generation2, which is one of the major passive RFID protocols, by adding new reader commands and a tag state. We call this extended protocol as EGen2 (Extended Gen2). The EGen2 protocol can deliver streaming data, such as audio/video samples, from a reader to a device which is connected to an EGen2 tag. The EGen2 reader acts as a stream data generator which makes packets with audio samples and the tag system (e.g., wireless headphone receiver) receives the packets to play the sound. In order to show feasibility, we designed and implemented a prototype system. We carefully modified the tag’s digital blocks considering constraints of the tag’s digital part, such as area and power consumption. The reconfigured digital logics of the tag needed 11.34K gates, and power consumption was 3.33µW with 25% toggle rate. Our analysis shows that the power consumption is acceptable for operating passive tags, and therefore the proposed EGen2 protocol can be used for wireless headphones.

The paper is organized as follows. In the next section, we present the extended Gen2 protocol in detail and Section 3 introduces our prototype system. We evaluate the performance in terms of area and power consumption in Section 4. The last section summarizes the research.

2. EXTENDING EPC CLASS-1 GEN2 PROTOCOL

We can classify an RFID system by several criteria, such as the operating frequency of a reader, the physical coupling method, the range of the system, and the memory capacity and usage. Among several passive RFID standards [8, 9, 10], we chose the EPC Class-1 Generation2 protocol (ISO18000-6 Type C) for our prototype system. The chosen protocol uses a UHF (860MHz~960MHz) frequency band, DSB-ASK, SSB-ASK, or PR-ASK for forward modulation, and ASK and/or PSK for backward modulation. The protocols in this band support several meters of operating range and use sufficiently small size passive tags. Amongst the protocols, the chosen protocol has the highest data link frequency for forward (26.7kHz~128kHz) and backward (40kHz~64kHz).

In order to apply a passive RFID system to wireless headphones for extreme low power consumption, we extended the EPC Class-1 Generation2 protocol and it is called the extended Gen2 (EGen2). In order to deal with streaming data, we added two new reader commands, Burst and BurstAckVS, and one tag state, Data to the original Gen2 protocol. The added command formats and detailed parameter descriptions are shown in Table 1 and Table 2.

The command Burst is used for the transmission of data streams, and BurstAckVS is for verifying the transmission. The Burst command can embed sample data from 32Byte to
Table 1: Burst command format and description.

<table>
<thead>
<tr>
<th>Field</th>
<th># of bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmd</td>
<td>4</td>
<td>1011b</td>
</tr>
<tr>
<td>Important</td>
<td>1</td>
<td>If 0, there will be no response from tags.</td>
</tr>
<tr>
<td>ShortID</td>
<td>4</td>
<td>Lower nibble of Short ID</td>
</tr>
<tr>
<td>SeqNum</td>
<td>15</td>
<td>Short ID must be smaller than 2^16-1. All tags process this burst process.</td>
</tr>
<tr>
<td>EndSig</td>
<td>1</td>
<td>If 1, this packet is the last one.</td>
</tr>
<tr>
<td>Len</td>
<td>4</td>
<td>Length of Bitstream</td>
</tr>
<tr>
<td>Bitstream</td>
<td>256–8M</td>
<td>Bitstream length is 32 × 2&lt;sup&gt;len&lt;/sup&gt;Bytes</td>
</tr>
<tr>
<td>CRC16</td>
<td>16</td>
<td>CRC16 for data integrity</td>
</tr>
</tbody>
</table>

Table 2: BurstAckVS command format and description and its response format.

<table>
<thead>
<tr>
<th>Command</th>
<th>Field</th>
<th># of bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BurstAckVS</td>
<td>Cmd</td>
<td>4</td>
<td>1011b</td>
</tr>
<tr>
<td>VeryShortID</td>
<td>Len</td>
<td>4</td>
<td>Lower nibble of Short ID</td>
</tr>
<tr>
<td>Result of process</td>
<td>EndSig</td>
<td>1</td>
<td>If 1, the processed had CRC</td>
</tr>
<tr>
<td>ShortID</td>
<td>7</td>
<td>The tag’s Short ID</td>
<td></td>
</tr>
</tbody>
</table>

1MBYTE by adjusting a value of Len. However, up to 4KByte data are acceptable according to the mathematical analysis of CRC16 [11]. If Burst contains more than 4KByte, we may need to consider other methods to guarantee the data integrity. A reliable communication can be supported by the Important flag. If this flag was set to 1, the reader which sent the previous Burst can check the result of the tag’s stream process by sending BurstAckVS. The tag which had the same VeryShortID sends an acknowledgment with the result of the process. The reader can retransmit the previous Burst command to the tag with the same SeqNum. The SeqNum can be used to support out-of-order acknowledgment and retransmission in the future. The EndSig flag notifies the end of a transmission sequence. If the flag is set to 0, the next Burst command will be transmitted to tags.

The modified state transition diagram of the proposed EGen2 is shown in Figure 3.1. The dark shaded parts are the extended portions from the original state diagram and details of the original are omitted. The process that a reader delivers stream data to the specific tag(s) takes the following steps:

1. The RFID reader chooses one or more tags by sending the Select command with inventoried IDs. If a tag is chosen, the tag asserts its SL flag which means it is selected.
2. The reader tests the tag by sending Query with a proper Q and Sel=11b. The tag should respond RN16.
3. The reader writes a unique ShortID which is not equal to zero into the tag’s specific memory region.
4. After choosing all tags for the reader by repeating the steps from 1 to 3, or after setting ShortID to 00000000b, the reader sends the Burst command. If the EndSig flag is 1, the selected tag changes its state to Ready. Otherwise the tag’s state is changed to Data.
5. After the tag state becomes Data, the reader can send Burst continuously.

6. Optionally, the reader can send the BurstAckVS command to the tag to support a reliable communication. In this case, the Important flag of Burst should be set to 1.

3. DESIGN OF EGEN2 PROTOTYPE SYSTEM

Our prototype system is shown in Figure 4. There are two separate systems; one is a sender which consists of an RFID reader and ADC to sample audio signals, and the other is a receiver which consists of a tag and its peripheral system to play the music from the received data. In order to test our new protocol, we used the following scenario:

1. The RFID reader selects tags through the original Gen2 protocol by asserting the SL flag of tags that are ready to receive audio streams. The reader can also use a broadcasting communication method by setting the ShortID of Burst command to zero.
2. After choosing the tags, a sender plays an audio file and outputs the analog signal of the music to the ADC of the sender. In this experiment, we used the audio file which was encoded in 44kHz with 14-bit resolution.
3. The sender system converts the output signal into digital samples using the ADC and stores the sample values into FIFO. If we were to preserve the quality of the audio file perfectly, the sampling frequency must exceed the double of audio signal frequency. However, due to the limited forward data rate of the current protocol (128kbps) [10], the ADC works at 14kHz. The resolution of a quantized audio sample was 8bits.
4. The EGen2 reader dequeues the data from FIFO and loads them on the burst packet. When a demanded packet is ready, the reader sends out the packet on Burst command.
5. The EGen2 tag, regardless of its state, extracts the audio sample data from the packet and passes them to the tag controller. At each arrival of packets, the tag notifies it to the controller. When there’s a problem with the received packet, for example, CRC error or out of sequence number, the tag controller is notified and the controller may discard problematic data.
6. The tag controller checks the availability of the system, and if it is ready, stores the audio data into FIFO between the tag controller and the DAC controller.
7. The DAC controller dequeues the data from FIFO at the same rate of audio data sampling (14kHz). And finally, the DAC recovers the analog signal and sends it to a speaker to produce the music.

Although the prototyped system seems to be very complex, only small parts (ADC/DAC and small logics to process the two new commands and the state) are added to the original RFID reader and tag system. The reader was implemented on an Altera’s Excalibur FPGA [12], and the receiving node was implemented on the same platform without ARM but with a small general purpose 8-bit microprocessor (AVR128). A Xilinx’s SPARTAN-3 FPGA [13] was used to implement the digital part of the EGen2 tag. We used the different FPGA platform in order to use a gated-clock for low power tag implementation.
3.1 EGen2 Sender System

In the prototype system, a sender acts as an audio stream data generator, and it is shown in Figure 5. For data generation, an audio sampler which comprises ADC, an ADC controller and a FIFO module was added to the Gen2 reader, and a packet generator was modified to process new commands. For fast prototyping, the analog front-end was discarded and replaced by the wired cable.

The sender takes 8-bit quantized digital samples with 14kHz sampling frequency (sufficient for voice and low quality audio) from analog audio signals. In case of the Burst command, a packet generator checks the number of used words in FIFO, and if the sample data are enough to make a packet, it starts encoding NRZ bits and transmitting the encoded bits. In order to reduce erroneous sample data, the reader can change the packet length of Burst in accordance with PER (Packet Error Rate) by adjusting Len. The reader calculates the PER from the most recent packets by checking the responses of BurstAckVS and then decides whether to double, cut in half, or keep the packet length.

3.2 EGen2 Receiver System

The receiver system is composed of two main parts. One is an EGen2 tag to receive data from the EGen2 reader of a sender, and the other is a tag peripheral system to produce the music from the received data.

3.2.1 EGen2 Tag

The EGen2 tag was extended from the Gen2 tag of our own, and its architecture is shown in Figure 6. The original tag supports all mandatory commands and an optional Access command in the EPC Class-1 Generation-2 UHF (860~960MHz) RFID Protocol 1.1.0 version. The tag satisfies all specification (55 items) in the protocol including functionalities, error handling, forward/return link frequencies, frequency tolerance, and response time.

In order to add new commands and a new state to the original Gen2 tag, we modified the command decoder module. For low power design, we used multi-frequency/multi-phase clocking. A power management unit generates 7 local clocks by using a clock-gating scheme, and differentiates each clock phases to minimize the peak current. For operating frequency, we assumed that an analog front-end provides 1.28MHz or 1.92MHz in accordance with return link frequency of a tag.

3.2.2 Tag Peripheral System

The tag peripheral system consists of three heterogeneous devices as shown in Figure 7; a microprocessor for an operating system and applications, DAC, and a communication unit. The microprocessor runs a small operating system (TinyOS 2.0 [14]) for IP control, and an application for playing the audio data from the tag. DAC converts the received digital data into analog. The communication unit connects the microprocessor and the other IPs (an EGen2 tag and DAC).

In order to achieve the low power consumption, we intro
duce three design techniques: a hardware interface control, an inter-IP communication unit, and a hardware IP scheduler. The proposed interface protocol is not only for the easy reusability of IPs, but also for an efficient control of IPs. We call the interface protocol as the hardware control interface (HCI). The protocol is composed of five basic commands (INIT, START, STOP, WRITE, and READ) and IP-specific commands which can be defined by a user. These five basic commands are quite efficient at controlling hardware modules. To support the interface protocol, all IPs should be attached with protocol wrappers. Each wrapper stores the state of the corresponding IP that can be easily accessed by the operating system.

Most of current embedded systems use a shared bus architecture to connect each hardware module. However, a shared bus is inefficient in the aspect of power consumption because it requires high fan-out and complex decoder logics. Therefore, instead of a shared bus, a network-packet-router-like communication unit, called an inter-IP communication unit, is used in our system. All IPs in the system are directly connected to the communication unit. Every message sent and received among the IPs passes through the unit. The messages follow the hardware control interface (HCI) have fields for a source and a destination. The communication unit is free from the disadvantages of shared buses, moreover it features better scalability and allows more opportunities of enhancing power efficiency.

Turning off unused IPs is a basic technique for low power operation. Most of existing embedded systems are general-processor-centered, and an operating system on the general processor periodically checks whether there's a task to be processed or not. Even when there's nothing to do, the general processor is turned on and power is wasted, and it is the most power consuming part of the system. To relieve this problem, a hardware IP scheduler separated from the general processor, is added. The hardware IP scheduler monitors all the IPs in the system, and turns on only the necessary ones. Because the scheduler needs only a reference table to keep the status of IPs, it nearly uses no power.

4. PERFORMANCE EVALUATION

We measured the performance of the receiver in terms of power consumption and area with 1KByte audio data per packet. Since there is no power limit in a sender and there was a little modification of the Gen2 reader, we did not analyze the performance of the sender. For gate level simulation of the receiver, especially for the EGen2 tag, we used Synopsys Design Compiler and Synopsys VCS under Anam CMOS 0.18µm technology, and for the power estimation, we used Prime Compiler with 25% toggle rate. We also measured the power consumption of the tag peripheral system. The evaluation included all parts of the tag except for EEPROM. It is acceptable because when the tag processes stream data, they are not stored into EEPROM and bypassed to DAC directly. The power consumption and the area of the original Gen2 and the proposed EGen2 tags are compared in Table 3.

EGen2 still perfectly supports all the commands of the original Gen2 protocol, and only Burst and BurstAckVS commands are added to the command decoder. As a result, the EGen2 tag increases the power by 3.8%, and the number of gates by 3.5% from the original. By applying the power result (3.33µW) to the previous analysis of Figure 2, we can get the available distance of 1.8 meter, which is adequate for a body area network. We can further reduce the power and increase the distance by removing unnecessary command such as Kill and Access which are not necessary for stream data transmission.

The power consumption of a tag peripheral system is the sum of power consumptions of the general processor, DAC,
Table 3: The number of gates and the power consumption of tag components except for EEPROM.

<table>
<thead>
<tr>
<th>Components</th>
<th>Area (gates)</th>
<th>Power (µW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen2</td>
<td>EGen2</td>
<td>Gen2</td>
</tr>
<tr>
<td>Response Generator</td>
<td>600</td>
<td>297</td>
</tr>
<tr>
<td>Miller Encoder</td>
<td>392</td>
<td>949</td>
</tr>
<tr>
<td>EEPROM Controller</td>
<td>952</td>
<td>933</td>
</tr>
<tr>
<td>Command Decoder</td>
<td>4588</td>
<td>4982</td>
</tr>
<tr>
<td>PIE Decoder</td>
<td>392</td>
<td>1.14</td>
</tr>
<tr>
<td>Matched Filter</td>
<td>297</td>
<td>289</td>
</tr>
<tr>
<td>Power Management Unit</td>
<td>297</td>
<td>255</td>
</tr>
<tr>
<td>Net</td>
<td>3848</td>
<td>3914</td>
</tr>
<tr>
<td>Total</td>
<td>11711</td>
<td>11013</td>
</tr>
</tbody>
</table>

and the hardware system which includes a DAC FIFO, 2 wrappers and a communication unit. The DAC FIFO consists of 8-bit 256 entries and consumes 12K gates. And the rest of the hardware system consumes 5K gates. The power consumption of the hardware system is 44µW.

To see the effect of our system configuration, we compared our hardware aid-system with a general-processor-only-system. Our system rarely wakes up the general processor. The H/W scheduler and the wrapper handle the required operations when the processor is asleep. The conventional general-processor-only-system rarely lets the processor sleep and consumes large power (24 mW at 3V power supply when active mode) [15]. In our experiment, the wake-up duty of the processor is set to 1%. So the total power consumed in our system is the sum of the power of the general processor (1% active, 99% power save) and the power of our hardware system. The result is 614µW, which is only 2.6% of the general-processor-only-system. The experiment implies that the new hardware components we added can greatly enhance the power efficiency, and as a result, it will prolong the life time of a battery-supported embedded system.

5. CONCLUSION

In this paper, we introduced the extended version of the EPC Class-1 Generation2 RFID protocol, called EGen2, and showed the feasibility of new use of passive RFID systems with this protocol. The EGen2 tag can extract stream data, such as audio samples, sent by a reader and pass the data to the tag peripheral device to reproduce the audio signal. The existing RFID system has been applied only for automation data acquisition methods, and our proposal is the first attempt to apply this RFID technology to a sound transmission system for extreme low power headphones.

We prototyped the system by adding new hardware modules to the Gen2 reader and tag for our proposal. The power consumption of the digital part for the EGen2 tag is sufficiently small for the operation as a passive tag. The area was increased by only 3.5%, and the total power consumption was 3.33µW which satisfies the protocol performance requirement for voice transmission at 1.8 meter working distance. Also, in order to achieve the low power consumption, we proposed a hardware control interface, an inter-IP communication unit, and a hardware IP scheduler. Since the RF logics consume 17.81µW in general [16], the total power consumption of the EGen2 tag can be assumed as much as 21.14µW. The power consumption of our proposal is much less than any other protocols for wireless headphones.

6. REFERENCES