

EPC Class-1 Generation-2 규격에 적합한 RFID 리더 시스템 설계

An Implementation of a RFID Reader System for EPC Class-1 Generation-2 Specification

양정규 · 배성우 · 송의석 · 안시영 · 오하령 ·
성영락 · 박준석 · 정명섭 · 곽호길

Jung-Kyu Yang · Sung-Woo Bae · Eui-Seok Song · Si-Young Ahn · Ha-Ryoung Oh ·
Yeong-Rak Seong · Jun-Seok Park · Myoung-Sub Joung · Ho-Kil Kwak

요 약

RFID 기술이 다양한 분야에 적용됨에 따라 각 분야의 특성에 맞게 RFID 리더에 요구되는 기능들도 다양해지고 있다. 그러나 현재 구현된 8051 프로세서를 기반으로 한 시스템은 프로세서의 처리 용량의 한계 및 메모리 크기의 제한 등으로 인하여 다양한 요구 조건들을 충족시키기에 부족하다. 본 논문에서는 다양한 요구 조건들을 충족시키기 위해서 임베디드 시스템에서 많이 사용되고 있는 ARM920T 기반의 프로세서를 이용하여 EPC Class-1 Generation-2 규격을 만족하는 UHF 대역의 RFID 시스템을 설계하였다.

Abstract

As RFID systems are applied to various fields and applications such as supply chain management, asset management, location based applications etc. the requirements becomes diverse. For example, Much higher performance, TCP/IP protocol stack are required in some applications. However, low end processors based systems such as 8051 processor can not meet such requirements due to their low processing capacity and limited size of memory. In this paper a UHF band RFID system which meets the EPC Class-1 Generation-2 specification with ARM920T-based processor is implemented and tested.

Key words : RFID, EPC Class-1 Generation, Type C

I. 서 론

현재 RFID(Radio Frequency Identification)는 유통 분야의 재고 조사와 제조 관리 중심으로 발전해 오던 바코드(Barcode)를 대체하여 물류 유통의 혁명을 초래할 기술로 주목받고 있다. RFID는 비접촉 무선 인식 기술로 바코드의 느린 인식 속도, 인식률, 다중 인식, 저장 능력의 한계를 극복하고, 산업 전반에 걸

쳐 응용이 가능한 분야이다. 대표적인 응용 산업은 고전적인 물류 관리, 재고 관리, 출입 통제 등을 포함하여, 위치 기반 서비스, 진품 확인, 여권 위조 방지, 가상 URL 등 다양한 분야로 적용 범위가 넓어지고 있다.

적용 분야가 다양해짐에 따라 각 분야의 특성에 맞게 RFID 리더에 요구되는 기능들도 다양해지고 있다. 단일 태그의 인식률과 인식 거리 향상을 요구

「이 연구는 2006년도 정보통신부의 대학 ITRC 지원사업의 연구 결과로 수행되었음(IITA-2006-(C1090-0602-0011)).」

국민대학교 전자공학과(Department of Electronic Engineering, Kookmin University)

· 논문 번호 : 20070518-155

· 수정완료일자 : 2007년 8월 8일

하는 분야가 있고, 복수 태그의 인식률을 중점으로 요구하는 분야도 있다. 또한, RFID 리더는 인식 속도, 저장할 수 있는 정보량, 미들 웨어와의 연동을 위한 다양한 통신 환경 등 다양한 조건들을 기본적으로 제공해야 한다.

그러나 현재 구현된 8051 등의 저 사양 프로세서를 기반으로 한 시스템은 가격적 우위가 있는 반면, 처리 용량의 한계 및 제한된 메모리 크기 등으로 인하여 다양한 요구 조건들을 충족시키기에 어려움이 많다. 본 논문에서는 다양한 요구 조건들을 충족시키기 위해서 현재 내장형 시스템에서 많이 사용되고 있는 ARM920T 기반의 프로세서를 이용하여 EPC Class-1 Generation-2(이하 Gen2) 규격을 만족하는 UHF 대역의 RFID 시스템을 구현하였다. ARM 프로세서는 다양한 내장형 운영 체제 및 실시간 운영 체제를 지원한다. 특히, ARM-9 기반의 프로세서는 32 비트 RISC 프로세서로서 little/big endian, fast interrupt request(FIQ), 단순하고 강력한 명령어 등을 지원하며 기존의 저 사양 프로세서와 비교하여 상당히 고속 처리가 가능하다.

본 논문에서는 UART, 이더넷 인터페이스를 제공하고 있으며 TCP/IP 스택을 이식하여 다양한 미들 웨어 업체들의 요구를 만족할 수 있다. 그리고 RT-OS를 이식하여 다양한 이벤트에 대해 실시간 처리가 가능하다. 특히, FIQ를 이용하여 기존의 8051 프로세서를 이용한 시스템보다 태그의 인식 실패율을 약 1/20 정도로 감소시켰다.

II. 관련 연구

2-1 RFID 기술 개요

RFID 기술은 사물의 고유 정보를 저장할 수 있는 태그와 태그로부터 정보를 읽고 쓸 수 있는 리더, 태그와 리더간의 정의된 주파수와 프로토콜을 사용하여 사물의 정보를 교환할 수 있는 기술이다. RFID 시스템은 그림 1과 같이 태그, 리더, 미들 웨어 및 응용 서비스 플랫폼으로 구성된다.

리더는 특정 주파수 대역의 RF 신호를 태그로 전송한다. RF 신호는 고주파 캐리어 신호와 태그에 대한 질문을 포함하는 명령 신호로 구성된다. 이 RF 신호에 의해 형성되는 리더의 전자기장 내에 태그가

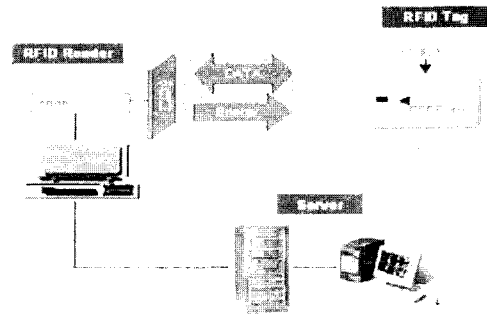


그림 1. RFID 구성
Fig. 1. Organization of RFID.

있을 경우 그 태그는 리더의 RF 신호로부터 전원을 공급 받고 수신된 RF 신호를 근거로 태그에 저장된 ID 등의 데이터를 백스캐터(backscatter)하여 리더에 전송한다.

현재 RFID 시스템은 적용 분야에 따라 13.56 MHz, 433 MHz, 860~960 MHz, 2.45 GHz 등 여러 주파수 대역에 대한 표준이 정해져 있거나 표준화가 진행 중이다. 이중 860~960 MHz 대의 RFID 시스템의 인식 거리가 타 주파수 대역보다 길다는 장점을 가지고 있어 주목을 받고 있다.

2-2 EPC Class-1 Generation-2 프로토콜

Gen2 프로토콜의 기본 개념은 '리더가 먼저 통신(ITF: Interrogator Talks First)한다'는 것이다. 즉, 리더가 먼저 태그에 명령을 전송한 후에 태그의 응답을 분석하고 그에 따른 다음 명령을 리더가 전송하는 것이다.

리더는 명령어를 전송하고 태그의 응답 구간에선 안정적인 전원을 공급해야 하는데, 이때 리더로부터 방사되는 신호를 CW(Continuous Wave) 신호라고 한다. 리더는 통신을 시작하고 끝내게 될 때까지 항상 태그에게 명령어나 CW 신호를 방사함으로써 태그가 초기화 되지 않도록 해야 한다. 태그는 리더로부터의 명령을 받아, 응답 변조 방식과 전송 속도 등이 결정되고, 백스캐터 방식으로 응답하게 된다. 또한, 기존 프로토콜과 달리 리더가 태그로부터의 응답을 받은 후, 정해진 시간 안에 다음 명령어를 전송해야만 태그가 현재 상태를 유지하게 된다. Gen2 프로토콜에 대한 특성을 표 1에서 기술하였다.

표 1. Gen2 프로토콜

Table 1. Gen2 protocol.

표준 제안		EPC Class-1 Generation-2
리더	전송 속도	128~26.7 Kbps
	변조 방식	DSB-ASK, SSB-ASK, or PR-ASK 90~100 %
	인코딩 방식	PIE(Pulse Interval Encoding)
태그	전송 속도	40~640 kHz
	변조 방식	ASK and/or PSK
	인코딩 방식	FM0 and/or Miller-modulated subcarrier
충돌 중재 방식		Enhanced ALOHA
리더 적합 요구 사항		모든 Mandatory와 Optional 명령을 구현
리더-태그 응답 시간		MAX(RTcal, 10 T _{pri})
태그-리더 응답 시간		20 T _{pri}

III. 구현 및 실험

3-1 시스템 개발 환경

3-1-1 하드웨어 구성

본 논문에서 구현한 900 MHz RFID 시스템은 크게 디지털 파트와 RF 파트로 구분되며, 디지털 파트에는 CPU가 포함된 시스템 제어 및 신호 처리부가 있다.

또한, RF 파트는 수신부, 송신부 그리고 국부 발진부로 구성되어 있다. 그림 2은 구현된 리더 시스템의 모습이며, 각 부분의 명칭을 설명하고 있다. 제어 및 신호 처리부의 사양은 다음과 같다.

3-1-2 펌웨어 개발 환경

본 논문에 구현한 RFID 리더를 구현하기 위한 환경을 표 2에서 정리하였다.

3-2 RFID 리더 시스템 구현

3-2-1 시스템 블록도

그림 3의 블록도를 살펴보면 CPU는 UART, eth-

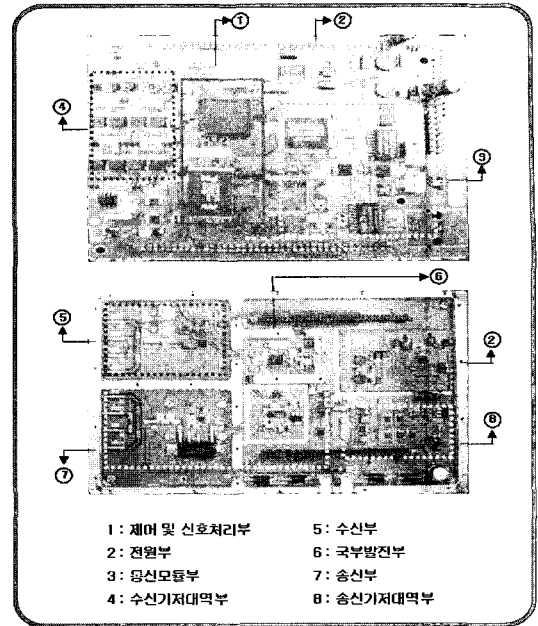


그림 2. 리더 시스템의 구성

Fig. 2. Organization of RFID system.

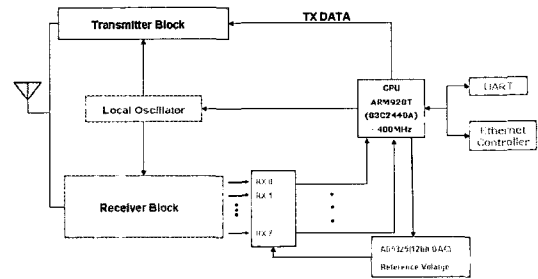


그림 3. 시스템 블록도

Fig. 3. System block diagram.

ernet 인터페이스를 모두 제어한다. 그리고 AD5325 (12 bit DAC), local oscillator를 제어한다.

그림 3의 블록도를 살펴보면 CPU는 UART, ethernet 인터페이스를 모두 제어한다. 그리고 AD5325 (12 bit DAC), local oscillator를 제어한다.

3-2-2 메모리 구성

본 논문에서 구현한 시스템에는 NAND Flash(32 MB), NOR Flash(1 MB), SRAM(1 MB), SDRAM(64 MB) 메모리를 사용하고 있다. 보통의 시스템의 경우, NOR 플래시 메모리를 이용하여 부팅(booting)을 하지만 본 시스템에서는 CPU(S3C2240) 내부에 있는

표 2. 펌웨어 개발 환경

Table 2. Firmware development environment.

개발 환경		세부 사항
프로세서	S3C2440A(400 MHz)	
메모리	RAM	K4M513233C-D(64 MB)
	ROM	AM29LV800BBC(1 MB: Nor Flash) K9F1208U0B(32 MB: Nand Flash)
부트로더	U-Boot	
운영 체제	uCOS-II	
통신 환경	UART, 10BaseT LAN	
플랫폼	Linux Enterprise 3.0	
툴체인	gcc 3.3.2 binutil 2.14 gdbc 2.2.5	
개발 도구	ARM Development Suite 1.2	

SRAM(4 KB)를 이용하여 NAND 플래시 메모리로부터의 부팅을 구현하였다. 이는 NAND 플래시 메모리가 NOR 플래시 메모리에 비하여 가격이 싸고 용량이 크기 때문이며, 따라서 NAND 플래시에 파일 시스템 등을 구현하고 NOR 플래시를 제거하여 크기와 가격을 낮출 수 있다. 그림 4는 시스템의 메모리 구성을 보여주고 있다.

3-2-3 TCP/IP 스택 이식

본 논문에서 구현한 시스템에 “LWIP(Light-Weight Implementation of the TCP/IP Protocol)”라는 TCP/IP 스택을 이식하였다. 기존에는 임베디드 시스템을 위한 TCP/IP 스택으로 “uCIP”도 널리 사용되었으나, 현재 새로운 버전이 발표되지 않고 있다. 그러나

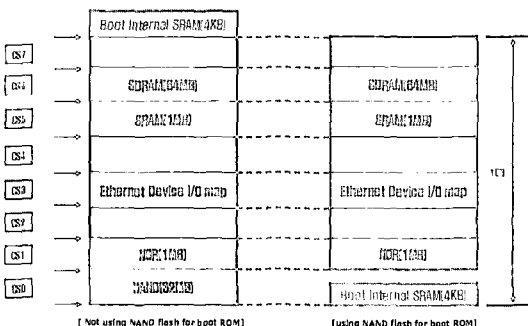


그림 4. 메모리 구성
Fig. 4. Memory map.

“LWIP”는 현재까지도 계속해서 새로운 버전이 발표되고 있기 때문에 안정적으로 지원을 받을 수 있다. 뿐만 아니라 “LWIP”는 TCP/IP 프로토콜의 모든 기능과 Berkeley socket API를 제공하므로 UNIX 계열의 TCP/IP 프로그램을 접해 본 사람이라면 쉽게 구현할 수 있는 장점이 있다.

3-2-4 uC/OS-II 이식

본 논문에서 구현한 시스템에 “uCOS-II”를 이식함으로써 다양한 실시간 이벤트 처리가 가능하다. 그림 5에서 보듯 이식 시에 프로세서와 의존적인 코드 3개가 있다^[4]. 즉 “OS_CPU.H”, “OS_CPU_A.ASM”, “OS_CPU_C.C” 등의 코드들을 본 시스템의 메모리 구성이나 사용하는 프로세서에 맞게 수정하였다.

그림 6은 본 논문에서 구현한 태스크 구성을 보여주고 있다. 이더넷 통신과 관련된 태스크, UART와 관련된 태스크와 호스트 명령에 따른 처리를 하기 위한 RFID TRX 태스크 등 3개의 태스크로 구성되어 있다. RFID TRX 태스크는 기본적으로 RFID와 관련된 모든 기능들을 포함하고 있다. 호스트에서 이더넷이나 UART로 통해 명령을 내리면 이더넷, UART 태스크에서 명령을 분석한 후에 RFID TRX 태스크로 명령을 보낸다. 그러면 이 태스크에서는 명령에 따라 환경 설정 정보를 세팅하거나 태그 정보를 읽거나 쓴다.

3-3 Gen2 프로토콜 구현

본 논문에 개발한 펌웨어(Firmware)는 Gen2 명령

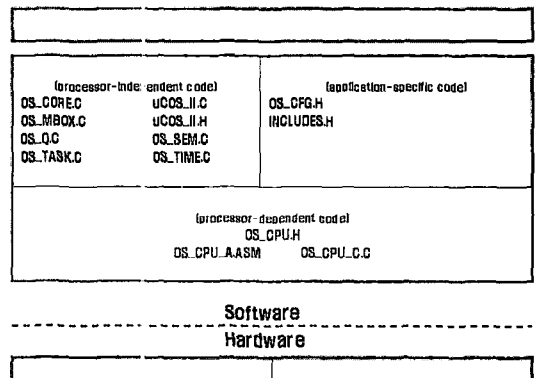


그림 5. uC/OS-II 구조^[2]
Fig. 5. uC/OS-II structure.

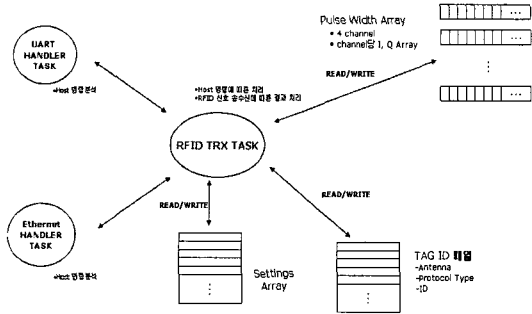


그림 6. 태스크 설계
Fig. 6. The design of task.

을 모두 구현하고 국내 규격에도 적합하도록 시스템을 제어하도록 만들었다. 또한, 호스트와의 통신을 위하여 프로토콜을 정의하고 구현하였다.

펌웨어는 호스트로부터 받은 명령을 분석한 후 적절한 Gen2 명령을 태그로 전송하고 태그로부터 받은 응답을 분석하여 다시 호스트로 처리 결과를 전송하는 과정을 반복한다.

그림 7은 호스트로부터 받은 명령을 처리하는 과정 중 일부를 나타낸 것이다. Select 명령을 통해 태그의 세션과 인벤토리를 구분하여 응답하도록 설정한다. Query 명령은 응답할 태그의 변조 방식과 응답

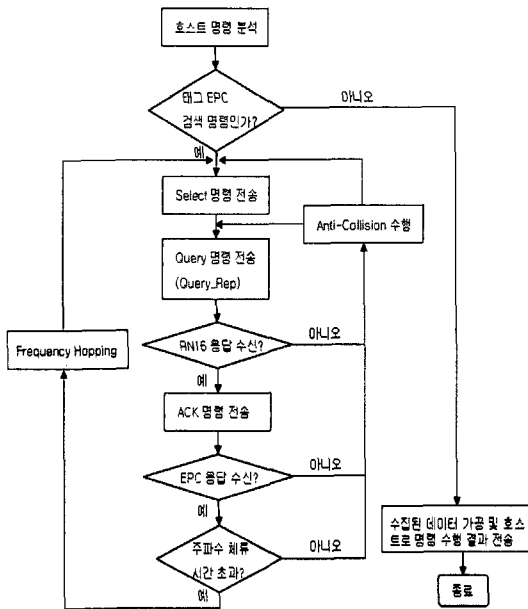


그림 7. 펌웨어 동작
Fig. 7. Firmware action.

속도를 결정한다. 두 명령을 받은 태그는 16비트 난수(RN16)를 발생시켜 응답하게 된다. 이 RN16은 태그가 생성하기 때문에 신뢰성이 보장될 뿐만 아니라 동일한 태그라 하더라도 리더 명령을 받게 될 때마다 매번 다르게 발생되기 때문에 리더-태그 통신의 보안성을 한층 더 보장 받을 수 있게 된다. 리더는 ACK 명령어에 태그의 RN16을 다시 전송하여 해당 태그의 EPC를 얻게 된다. 이때 태그는 자신이 응답한 RN16값과 리더로부터 다시 받은 RN16값이 일치하고 시간 타이밍이 맞을 경우에만 EPC값을 응답하게 된다. 이 정보를 기초로 태그의 메모리에 읽기, 쓰기, 잠금(LOCK) 등의 명령이 수행된다.

그리고 태그로 검색 명령을 전송한 후 규격에서 정한 제한 시간 동안 태그로부터 응답이 있는지 검사한 후 아무런 응답 신호가 없는 경우에는 바로 검색 과정이 종료되지만 태그 ID가 들어오거나 알 수 없는 신호가 들어오면 충돌 여부를 판단한 후 필요에 따라 충돌 중재(collision arbitration) 과정을 거쳐 다시 태그 검색을 시도한다. 태그 검색 도중 발견된 ID는 메모리에 누적하고 검색 과정이 종료된 후에 호스트로 태그 정보를 전송하게 된다.

또한, 현재 국내의 RFID 시스템 표준에서는 동일한 주파수 내에서의 체류 시간을 400 ms 이내로 제한하고 있기 때문에 타이머를 이용하여 펌웨어 동작 중 400 ms마다 동작을 멈추고 캐리어 변경(carrier change)을 하여 이 규정에 만족하도록 하였다.

3-4 실험 및 측정 결과

3-4-1 태그 응답

리더에서 태그로 보내는 신호는 preamble이나 Frame-Sync, 각 명령에 따른 파라미터, CRC로 구성된다. Query 명령만 preamble로 시작하고 나머지 명령들은 Frame-Sync로 시작한다. Preamble은 리더와 태그 사이의 통신하는 속도와 관련된 정보를 담고 있다. Query 명령의 파라미터 중 Q값은 태그 내부의 슬롯 개수를 지정한다. 이 Q값은 4 bit로 0~15까지 설정할 수 있으며, $2^0 \sim 2^{15}$ (1~32,768)개까지의 태그의 충돌을 중재할 수 있게 된다. 단일 태그 시에 Q값을 0으로 하게 될 경우, 태그 내부의 슬롯 개수가 1개이기 때문에 태그는 항상 반응하게 된다. 반면 다

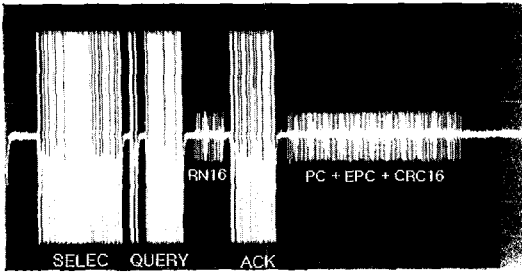


그림 8. 실제 리더의 명령 및 태그의 응답 신호
Fig. 8. Reader command and tag response.

중 태그일 경우에는 상황에 따라 Q 값을 조정하여 충돌 중재를 하는데, 본 논문에서는 기본 Q 값을 4로 하여 테스트하였다.

리더는 select 명령을 태그로 전송하여 태그의 session, membank 등을 결정한다. 다음에 Query 명령을 전송하여 DR, Q 값 등을 결정한다. Select, Query 명령을 차례로 전송 받은 태그는 내부의 난수 발생기를 만들어진 16비트 난수를 응답하게 된다. 이 난수를 RN16이라고 부른다. 리더는 태그의 RN16을 전송받은 후 ACK 명령에 RN16을 실어서 태그로 다시 전송하게 되면 태그는 자신이 내부적으로 가지고 있는 RN16과 리더로부터 전송받은 RN16 값이 같을 경우에 자신의 EPC를 리더로 전송하게 된다. 그림 8은 태그의 EPC를 얻어오기까지의 리더-태그간의 실제 신호를 보여주고 있다.

Gen2에서는 ISO 18000-6 Type A 프로토콜에서 사용됐던 “ALOHA” 방식을 더 개선시킨 Slotted ALOHA 방식을 사용한다. Slotted ALOHA 방식은 난수를 이용하여 응답 순서를 정해서 태그를 인식한다는 것이 주된 내용이다. 최초 Query 명령의 Q 값을 받은 태그들은 $0 \sim 2^Q - 1$ 중의 하나의 난수를 발생시켜 갖게 된다. 첫 Query 명령에 대해 태그가 생성한 난수가 0이었던 태그가 응답을 하게 된다. 리더는 Q 값에 따른 round 크기만큼 QueryRep 명령을 반복적으로 태그로 전송하고 태그는 자신이 생성한 난수를 1씩 감소시킨다. 이때 난수가 0인 태그는 응답을 하게 된다. 라운드가 끝나면 리더는 라운드 동안에 “몇 번의 충돌이 일어났는지?”, “응답이 전혀 없었는지?” 등의 정보를 가지고 QueryAdjust 명령을 이용해 Q 값을 재조정한다. 이런 식으로 반복적으로 수행하여

복수 태그의 정보를 읽어 온다.

이러한 Slotted ALOHA 방식의 단점은 주변에 적은 수의 태그가 있을 경우, Q 값을 너무 큰 값으로 설정하게 되면 모든 태그의 응답을 받아내는데 걸리는 시간이 매우 늦어진다는 것이고, 반대로 주변에 매우 많은 태그가 있을 경우엔 적은 Q 값으로 설정하게 되면, Q 값을 적절한 크기의 값으로 재설정하기 전까지의 작은 Q 값의 경우는 충돌을 중재할 수 없는 버려지는 시간이 된다는 것이다. 그러나 Type B의 경우, 매번 태그 ID(Return Preamble+64 bit Data+16 bit CRC16)의 응답을 분석하는 반면에, Gen2의 충돌 중재는 RN16(Return Preamble+16bit RN16) 응답만을 가지고 분석하기 때문에 시간적으로 유리한 면도 있다.

따라서 Gen2 프로토콜에서의 충돌 중재는 적절한 Q 값을 찾아내는 방법과, 상황에 따라 Q 값을 재설정하는 것이 필요하다.

3-4-2 변조 방식 및 응답 속도

Gen2 태그는 FM0, Miller 방식을 모두 지원한다. 본 논문에서는 FM0 방식으로 태그와 통신하였다. 태그의 변조 방식은 Query 명령의 M이라는 파라미터를 1로 선택할 경우에만 FM0 방식으로 응답하도록 되어 있다.

태그의 응답 속도는 preamble 신호 중 TRcal, Query 명령의 DR값에 따라 정해진다. 수식은 아래의 식 (1), (2)와 같다.

$$LF = \frac{DR}{TRcal} \quad (1)$$

$$1.1 \times RTcc1 \leq TRcal \leq 3 \times RTcc1$$

$$(RTcc1 := Data0_{length} + Data1_{length}) \quad (2)$$

본 시스템에서는 리더의 Tari를 25 usec로 설정하였기 때문에 Data 0의 길이는 25 us, Data 1의 길이는 50 us이다. Data 1의 길이는 37.5~50 us 범위로 설정 가능하지만 본 시스템에서는 50 us가 되도록 구현하였다. 즉, RTcal은 75 us이다. 위의 수식에 대입해 보면 TRcal은 “32.5 us ≤ TRcal ≤ 225 us” 범위를 가지게 되고 “DR”값은 8로 설정하였다. 그러므로 LF(태그의 응답 속도)는 TRcal을 200 us으로 했을 때 “8/200

us=40 Kbps”가 된다. 표 5에서 TRcal, DR 값에 따른 태그의 응답 속도를 정리하였다.

3-4-3 링크 타이밍

Gen2 프로토콜은 ISO/IEC 18000-6 Type B 프로토콜에 비해 리더-태그 간의 응답 시간을 엄격하게 규정하고 있다. 그림 9와 표 3은 리더-태그 간의 응답 시간을 자세하게 설명하고 있다.

태그의 응답 속도가 40 Kbps일 때 표 3의 파라미터들을 계산해 하면 “ $T_{pri}=1/LF$ ”이기 때문에 25 us가 된다. 그러므로 T1은 대략 238 us 이상 262 us 이하가 되고, T2는 75 us 이상 500 us 이하가 된다. 또한, T3 시간은 제한이 없고 T4 시간은 150 us 이상이 된다. T1 시간은 리더의 명령 후에 태그의 응답 시간이다. 실험에 적용된 태그의 경우, 평균적으로 약 240 us 정도 되는 것을 확인하였다. T4 시간은 select 명령과 Query 명령 사이의 시간이다. Select 명령에 대한 태그의 응답이 없기 때문에 T4 시간을 규정하고 있다. 본 시스템에서는 200 us로 설정하였다. T2 시간은 태그가 응답한 후 리더의 다음 명령이 전송되기까지의 시간이다. 시스템은 이 시간 안에 태그의 응답 신호를 분석하여 다음 명령을 준비해야 된다. 만약 이 시간 안에 다음 리더의 명령이 태그로 전송이 되지 않을 경우에는 그 명령에 대한 태그의 응답을 보장할 수가 없기 때문에 반드시 지켜야 하는 규

표 3. 링크 타이밍 파라미터^[1]

Table 3. Link timing parameter.

	최소	평균	최대
T ₁	MAX(RTcal, 10 T _{pri})×(1-FT) - 2 us	MAX(RTcal, 10 T _{pri})	MAX(RTcal, 10T _{pri})×(1+FT) + 2 us
T ₂	3.0 T _{pri}		20.0 T _{pri}
T ₃	0.0		
T ₄	2.0 RTcal		

정이다. 이 시간은 시스템에서 사용한 프로세서 성능 및 응답 신호 분석 알고리즘에 따라 상당한 차이가 나타날 것이다. 본 시스템에서는 S3C2240(400 MHz)를 사용하였으며, 실시간으로 8개의 RX 신호를 폴링(polling) 방식으로 동시에 처리하였다. 실험상 약 10 us 이내에 모든 처리가 끝나기 때문에 태그 응답 속도를 40 Kbps로 하였을 경우 T2가 최소 75 us이므로 100 us 정도의 지연을 두어 이 규정을 만족하도록 하였다. 태그의 응답 속도를 높일수록 T2 시간이 짧아지므로 이에 대한 검증이 필요하다.

3-4-4 인식률 비교 측정

기존의 8051 시스템에서는 입력 신호의 에지에서 인터럽트가 발생되므로 1개의 채널 내의 2개의 신호 경로를 동시 처리를 하지 못하였다. 또한, 기준 전압

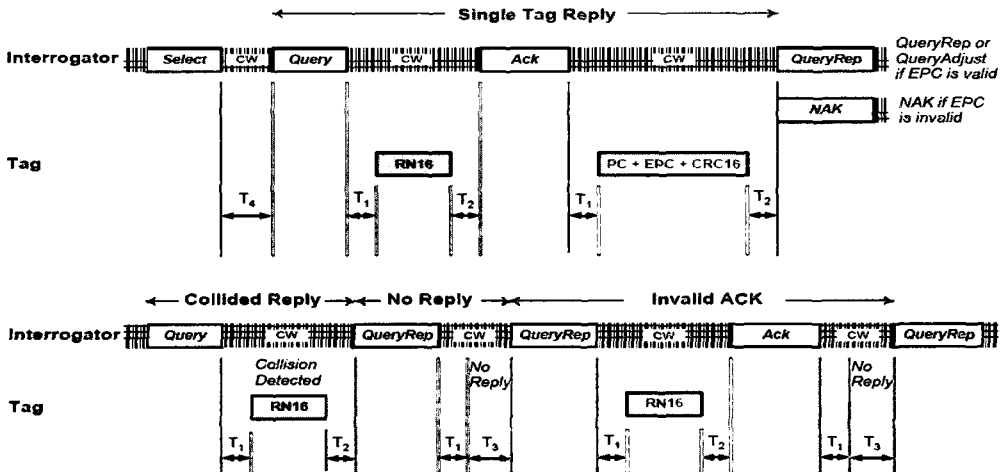


그림 9. 리더-태그 간 링크 타이밍^[1]

Fig. 9. Reader-tag link timing.

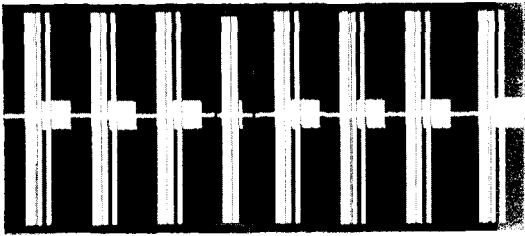


그림 10. 태그 인식 실패
Fig. 10. Tag detecting fail.

을 단계별로 적용해야 되는 문제점이 있었다. 이런 문제점은 1개의 채널 내의 2개의 신호 경로를 번갈아 분석해야 하며, 태그 응답의 수신 감도에 따라 기준 전압이 맞지 않을 경우에는 태그가 정상적으로 반응함에도 불구하고 응답 신호를 처리하지 못하는 경우가 많았다. 이런 상황은 충돌 중재 시에 알고리즘 구현 및 성능에 치명적인 영향을 미치므로 반드시 개선되어야 할 문제점이었다.

그러나 본 논문에서는 이러한 문제점을 해결하고자 ARM을 이용하여 프로세서의 처리 속도를 향상시켰다. 그리고 기존의 인터럽트 방식의 알고리즘을 주기적으로 RX 포트를 폴링(polling)하는 알고리즘으로 수정하여 총 4개 채널(총 8 path)을 동시에 처리하였다. 또한, 기준 전압이 각 채널별로 할당이 되어서 기존의 8051 시스템에서 문제되었던 부분들을 해결하였다.

8051 기반 시스템과 ARM 기반 시스템의 비교실험을 통해서 인식률을 측정해 보았다. 기본적으로 전체적인 하드웨어 구조가 다르므로 정확한 비교 실험이 어렵지만 태그가 잘 인식되는 곳을 선택하여 TX 10번 당 태그 인식의 성공률을 측정하여 비교하였다.

그림 10과 같이 태그가 정상적으로 Query 명령에 대한 응답을 하였지만 리더에서 분석을 못하여 리더의 ACK 명령이 태그로 전송이 안 된 경우를 실패한 경우로 보아서 두 시스템 간의 인식률을 비교 측정하였다. 표 4에서 나타난 인식률을 분석해 보면 8051 프로세서를 이용한 시스템의 77 %의 인식 실패율에 비해 ARM 프로세서를 이용한 시스템은 4 %의 인

표 4. 인식률 비교

Table 4. Recognition comparison.

	1	2	3	4	5	6	7	8	9	10	
ARM	9	10	10	9	10	9	10	10	9	10	96 %
8051	2	3	4	3	2	2	3	1	2	1	23 %

식 실패율을 보였다. 결국 인식 실패율이 약 1/20 정도로 감소되었음을 확인할 수 있었다.

그리고 ARM 프로세서를 이용한 시스템은 태그 응답 중간에 노이즈가 나타날 경우에는 인식률에 문제가 있음을 확인하였다. 이를 위하여 RF noise를 안정화 시키거나 FM0 인코딩 규칙을 이용한 노이즈 제거 알고리즘을 개발하여 인식률을 개선할 것이다.

IV. 결론 및 향후 연구 방향

본 논문에서는 Gen2 규격에 적합한 리더 시스템을 설계하고 구현하였다. 그리고 uCOS-II를 이식하여 다양한 이벤트 처리에 유연하게 대처하도록 하였으며, LWIP라는 TCP/IP 스택을 이식하고 시리얼 인터페이스뿐만 아니라 이더넷 인터페이스도 지원하였다. 호스트 또는 미들 웨어와의 연동을 위해 호스트 프로그램과 통신 프로토콜도 구현하였다. 또한 ARM 920T 기반의 프로세서를 이용하여 기존의 8051 기반의 시스템에서 문제되었던 부분들을 해결함으로써 약 1/20 정도 인식 실패율이 감소되었다.

현재 다양한 응용 분야에서 충돌 중재 성능이 중요하게 대두되고 있다. 다양한 환경에서 충돌 중재 알고리즘을 검증 및 개선을 하여 리더의 성능을 개선할 것이다. 뿐만 아니라 다중 리더 환경의 리더들 간의 충돌 문제를 해결할 수 있는 알고리즘을 연구하여 해결 방안을 모색할 것이다.

참 고 문 헌

- [1] ISO/IEC FDIS 18000-6, "Parameters for air interface communication at 860 MHz to 96-MHz".
- [2] "MicroC/OS-II The Real-Time Kernel 2/E", Jean JLabrosse.

양 정 규



tem

2005년 2월: 국민대학교 전자공학과 (공학사)
 2007년 2월: 국민대학교 전자공학과 (공학석사)
 2007년 3월~현재: 국민대학교 전자공학과 박사과정
 [주 관심분야] RFID, Embedded System

오 하 령



학부 교수

[주 관심분야] RFID/USN, Embedded System, 실시간 처리, ASIC 설계

1983년: 서울대학교 전기공학과 (공학사)
 1988년: 한국과학기술원 전기 및 전자공학과 (공학석사)
 1992년: 한국과학기술원 전기 및 전자공학과 (공학박사)
 현재: 국민대학교 전자정보통신공

배 성 우



tem

2002년 2월: 국민대학교 전자공학과 (공학사)
 2004년 2월: 국민대학교 전자공학과 (공학석사)
 2004년 3월~현재: 국민대학교 전자공학과 박사과정
 [주 관심분야] RFID, Embedded System

성 영 락



학부 교수

[주 관심분야] 온라인 게임, RFID 실시간 처리, 고장감내 시스템, 이산사건 시스템 모델링 및 시뮬레이션

1989년 2월: 한양대학교 전자공학과 (공학사)
 1991년 2월: 한국과학기술원 전기 및 전자공학과 (공학석사)
 1995년 2월: 한국과학기술원 전기 및 전자공학과 (공학박사)
 현재: 국민대학교 전자정보통신공

송 의 석



장형 시스템, 영상처리

1993년: 국민대학교 전자공학과 (공학사)
 1995년: 국민대학교 전자공학과 (공학석사)
 2005년: 국민대학교 전자공학과 (공학박사)
 [주 관심분야] RFID, 병렬처리, 내

박 준 석



LA (PostDoctoralFellow)

1987년: 국민대학교 전자공학과 (공학사)
 1993년: 국민대학교 전자공학과 (공학사)
 1996년: 국민대학교 전자공학과 (공학박사)
 1997년~1998년: Dept. of EE, UCLA (PostDoctoralFellow)
 2000년~2003년: 순천향대학교 정보기술공학부 조교수
 2000년~현재: (주)아모텍 기술고문
 2001년~현재: (주)Wavics 기술고문
 2003년~현재: 국민대학교 전자정보통신대학 부교수
 2004년 9월~현재: 정보통신부 RFID/USN 기획위원
 2005년 3월~현재: mRF 단말분과 위원
 2005년 4월~현재: TTA RFID/USN 프로젝트 그룹 특별위원
 [주 관심분야] Mobile RFIC, RFID Active Tag, Wireless LAN

1993년: 국민대학교 전자공학과 (공학사)
 1996년: 국민대학교 전자공학과 (공학박사)
 1997년~1998년: Dept. of EE, UCLA (PostDoctoralFellow)

안 시 영



2003년: 국민대학교 전자공학과 (공학사)
 2007년: 국민대학교 전자공학과 (공학석사)
 [주 관심분야] RFID, Embedded System

정 명 섭



1996년 2월: 홍익대학교 전자공학과 (공학사)

1999년 2월: 국민대학교 전자공학과 (공학석사)

2004년 2월: 국민대학교 전자전파공학과 (공학박사)

[주 관심분야] UHF RFID 시스템

설계, RF 통신, 증계기

곽 호 길



2000년 2월: 순천향대학교 전자공학과 (공학사)

2006년 9월~현재: 국민대학교 전자공학과 석사과정

[주 관심분야] 증계기, RFID, RF 통신