

Stepwise 동기화 지원을 위한 CMOS 이미지 센서 Firmware 설계 및 개발

박현문² · 박수현^{1*} · 이명수² · 서해문² · 박우철² · 장윤정²

Firmware Design and system of stepwise synchronization for CMOS image sensor

Hyun Moon Park · Soo-Huyn Park · Myung Soo Lee · Hae-Moon Seo ·
Woo Chool Park · Yun Jung Jang

ABSTRACT

Lately, since Complementary Metal Oxide Semiconductor(CMOS) image sensor system has low power, low cost and been miniaturized, hardware and applied software studies using these strengths are being carrying on actively. However, the products equipped with CMOS image sensor based polling method yet has several problems in degree of completeness of applied software and firmware, compared with hardware's. CMOS image sensor system has an ineffective synchronous problem due to superfluous message exchange. Also when a sending of data is delayed continually, overhead of re-sending is large. So because of these, it has a problem in structural stability according to Polling Method. In this study, polling cycle was subdivided in high-speed synchronization method of firmware-based through MCU and synchronization method of Stepwise was proposed. Also, re-connection and data sending were advanced more efficiently by using interrupt way. In conclusion, the proposed method showed more than 20 times better performance in synchronization time and error connection. Also, a board was created by using C328R board of CMOS image sensor-based and ATmega128L which has low power, MCU and camera modules of proposed firmware were compared with provided software and analyzed in synchronization time and error connection.

Key words : CMOS image sensor, Fast synchronization, Serial camera, Interrupt firmware

요약

최근 CMOS 이미지 센서가 저전력, 저가격, 소형화를 이루면서 이를 이용한 하드웨어 및 응용 소프트웨어 연구가 활발하게 이루어지고 있다. 하지만 CMOS 이미지 센서 제품들은 하드웨어에 비해 아직 응용 소프트웨어 및 펌웨어의 완성도에서 여러 가지 문제를 가진다. CMOS 이미지 센서 기반 폴링 기법은 불필요한 메시지 교환으로 인해 비효율적인 동기화 문제 및 전송 지연이 일정 수준으로 높아지면 데이터 재전송에 대한 오버헤드가 크다. 이러한 이유로 폴링 방식의 구조적 안정성(structural stability)에 문제점을 가진다. 본 논문에서는 MCU를 통한 펌웨어 기반의 고속 동기화 기법으로 폴링 주기를 세분화하여 Stepwise 동기화 기법을 제안하고, 인터럽트 방식을 적용하여 재접속 및 데이터 전송을 개선하였다. 결과적으로 제안한 기법이 동기화 시간 및 에러 커넥션에서 20% 이상 뛰어난 성능을 보여주는 것으로 나타났다. 또한 CMOS 이미지 센서 기반의 C328R 보드와 저전력 MCU인 ATmega128L을 이용한 보드를 개발하고, 제공 소프트웨어와 제안된 펌웨어의 카메라 모듈과 동기화 시간 및 에러 커넥션(Error Connection) 등을 비교, 분석하였다.

주요어 : CMOS 이미지 센서, 빠른 동기화, 시리얼 카메라, 인터럽트 펌웨어

2008년 11월 11일 접수, 2008년 11월 30일 채택

¹⁾ 국민대학교 비즈니스IT학부

²⁾ 한국전자부품연구원

주 저 자 : 박현문

교신저자 : 박수현

E-mail; shpark21@kookmin.ac.kr

1. 서 론

초소형, 저전력의 영상 센서 모듈에 대하여 최근 활발하게 연구되고, 이를 이용한 시스템 온 칩(System on Chip; SoC) 개발이 활발하게 이루어지고 있다. 예를 들어, 영상 센서로는 CMOS를 들 수 있다. CMOS 이미지 센서 모듈은 초소형, 저전력으로 인하여 센서 네트워크, 무선 네트워크, 로봇 등에 응용되고 있다. 따라서 이러한 영상 센서에 의해 저전력에서 작동하며, 비교적 큰 크기의 이미지 데이터를 연속적으로 일정한 품질로 전송하는 것이 필요하다. 그렇기 때문에 저전력, 저가격 기반에서 신뢰성 있게 작동하면서 효율적으로 데이터를 전송하기 위해 영상 센서로부터 데이터를 수신하기 전에 가능한 짧은 시간 내에 동기화를 맞출 필요가 있다.

대표적으로는 CMU에서 개발한 CMUcam와 UCLA의 사이클롭스(cyclops)를 들 수 있다. CMUcam은 cc6620 이미지 센서와 카메라 기반의 실시간 영상 처리 프로젝트로 ARM7, ATmega128, MSP430, i8051 등의 다양한 MCU와 연동되며, 버전 3까지 공개되었다. 또한 확장 인터페이스에도 많은 연구가 진행되고 있다^[1]. 사이클롭스는 크로스보우(Crossbow)사의 Mica 제품군에 부착하여 센서 네트워크 연구에 사용하고 있다^[2]. 하지만, 두 가지 모두 이미지 비압축형(uncompress)이기 때문에 이미지를 압축하기 위한 시스템 프로세서의 오버헤드와 네트워크 대역폭의 관리가 필요하다. 또한, 이미지 센서를 처리하는 칩 OV76xx과 촬영된 이미지를 JPEG로 실시간 압축할 수 있는 OV528 칩을 장착한 COMedia 사의 C328R 제품이 있다^[3,4]. 하드웨어 칩에서 JPEG로 압축하기 때문에 전달 데이터 크기가 작고 이미지 프로세싱에 시스템 부하가 적다는 장점을 가지고 있다. 이미지 센서에서 UART로 바로 전송하는 Preview 모드와 SDRAM에 이미지를 저장하고 이를 UART에 전송하는 Snapshot 모드를 지원한다. 하지만 이와 같은 기존 시리얼 카메라는 MCU와의 재동기 및 폴링 기반의 구조로 인한 수신의 견고성 문제, 데이터 전송의 비효율성 등을 지닌다. 인터럽트 방식의 CMOS들도 MCU와의 동기화 이후 이미지 데이터 전송에서 적용되고 있다^[2-6]. 전송의 안정성을 위해 외부 메모리 저장 방식을 사용하기 때문에 외부 메모리를 장착하거나 64kb이상의 내부메모리를 가진 MCU가 필요하다. 본 연구에서는 싱크의 프레임 및 접속 기법의 개선으로 동기시간을 개선하였으며, I/O 인터럽트 방식을 통해 접속 유지 및 외부 최소 크기의 메모리에 적용방안을 제안

하였다.

본 논문 II장에서는 연구가 활발하게 진행되고 있는 CMOS 이미지 센서 현황과 폴링 기반 방식의 문제점과 비교하였으며, 구현에 적용된 C328x에 전송 모드의 문제점을 분석하였다. 그리고 III장에서는 C328x에서 싱크 프레임에 보오레이트(Baud-Rate)를 포함하는 Stepwise 동기화 기법은 개선된 싱크방안과 인터럽트 기반의 펌웨어를 통한 카메라와 MCU의 재접속 개선과 Preview, Snapshot 모드에 상관없이 4kbyte 이내의 내부메모리를 가진 MCU에서도 영상 데이터를 전송할 수 있는 방안을 제안하였다. IV장에서는 제조사에 제공 소프트웨어와 제안된 펌웨어의 카메라 모듈과 동기화 시간 및 에러 커넥션(Error Connection)을 비교 분석하여 성능 측정을 하였다. 결론에서는 향후 구현 내용 및 적용 분야에 대해서 간략하게 언급한다.

2. 관련 연구

2.1 CMOS 이미지 센서 모듈

표 1은 시리얼 기반의 CMOS 카메라 모듈이다. 동영상, 광학줌, 하드웨어 압축 등을 지원하는 다양한 모듈이 존재한다. 사이클롭스 모듈이 가장 적은 전원 소비를 하지만 앞에서 설명한 것 같이 이미지를 압축 전송 할 수 없다. 주로 낮은 대역폭(Bandwidth)을 지원하는 저전력 센서 네트워크나 초소형 항공기 등에 사용된다. 그렇기 때문에 상대적으로 수 바이트 정도의 작은 크기가 요구되며, 이미지 데이터 전송의 연속성 및 이미지 품질을 일정하게

표 1. 시리얼 카메라의 종류와 기능

제조사	소모 전력 (mW)	지 원 기 능 (해상도/프레임/방식)	가격(\$)
Cyclops*	46	352×288 / 10fps	54
CMUCam*	200	352×288 / 60fps	50
Web-Cam**	200	640×480 / 30fps	50
High-end PTZ Camera ***	1000	1024×768 / 30fps	1000
COMedia**	60	640×480 / 10fps (low resolution) JPEG compressed support	39

* 수동 포커스 지원

** 오토 포커스 지원

*** 줌 렌즈 지원

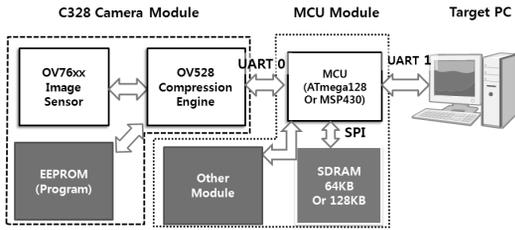


그림 1. 카메라 모듈과 MCU 연결 구조

표 2. 압축과 비압축모드일 때 이미지 데이터 크기

Resolution	Data Size		
	AVRAGE	MAX	MIN
160×120 (uncompress)	3800	7930	1669
160×120	2440	5108	895
320×240	5635	8764	2816
640×480	7830	9328	5350

* 수신데이터 샘플 이미지 ≤ 1000

보장해야 한다. 이를 위해서는 이미지 데이터를 하드웨어적으로 압축 할 수 있는 칩이 매우 중요하며, 특히 무선 네트워크 및 저 사양 MCU와의 데이터 전송에서 높은 효율성을 지닐 수 있다.

COMedia사의 C328R은 그림 1과 같이 두 개의 칩(OV76xx, OV528)으로 구성되어 있으며, 저전력, 저가격 기반에 이미지 압축을 지원한다. 외부에는 64~128k의 SRAM를 확장할 수 있다. 64-512byte의 데이터 프레임과 이미지 ID, Data Size 정보 및 프레임의 꼬리(Tail Frame)를 포함하여 6byte의 길이를 지니며, 총 이미지를 정해진 프레임 크기에 나누어 MCU에 전달한다. 수신된 MCU는 외부 메모리영역에 이미지 ID 넘버에 맞춰 저장하거나 프레임의 Data Frame 이외의 영역을 제거한 후 목적지에 전송하게 된다^[3]. Preview 지원에는 압축 모드와 비압축모드를 지원한다. 비압축모드에서는 160x120, 압축방식인 JPEG Preview Mode는 640x480까지 가능하다. 이미지 압축 방식은 표 2와 같이 다양한 크기를 지원한다. 하지만, 비압축 모드의 160x120 해상도에 데이터가 압축 모드 320x240보다 더 클 수 있다.

Preview모드는 C328x CMOS 모듈에서 목적지로 직접 전송할 수 있다. 하지만, C328x은 폴링 기반 구조로 전송로에서 데이터 손실이 발생할 경우 전송된 프레임에 대한 재전송 요구를 할 수 없다. 또한 발신 데이터 크기가 표 2와 같이 동일한 해상도라도 큰 차이를 가지고 있다. 그렇기 때문에 수신단에서 수백 ms이내의 메시지의 주기

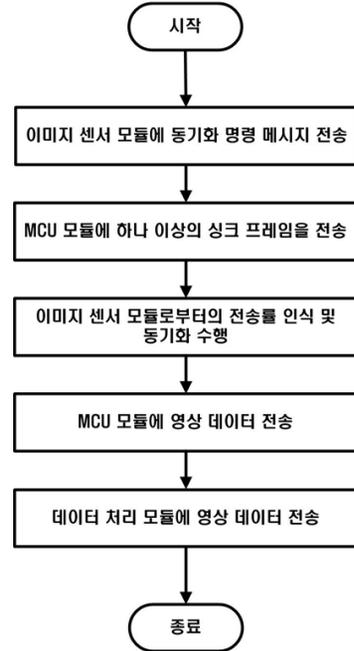


그림 2. CMOS 카메라에서의 데이터 전송과정

적 처리 요청이 명령충돌(command collision)로 인한 데이터 유실 및 수신지연이 발생된다. 이를 해결하는 기타 방안으로 TinyOS 기반 연구가 되고 있다. 그 밖에 다른 연구영역에서는 외부메모리를 사용하는 Snapshot 모드로 사용되고 있다^[6-8].

MCU의 내부 SDRAM이 작을 경우 하나의 프레임 데이터가 전부 전송되기 전에 가득 차게 되어 이후에 들어오는 프레임이 삭제되거나 수신 지연이 발생된다. 그렇기 때문에 외부 SDRAM의 확장을 통해 안전한 데이터의 송수신을 보장한다^[2,4,6].

2.2 소형 카메라 기반의 펌웨어 시스템

그림 2는 일반적인 CMOS 카메라와 MCU에 데이터 전송 과정을 간략하게 나타내었다. 이런 구조의 CMOS 카메라는 폴링기반의 제품이 많을 뿐만 아니라, 본 연구에 사용되는 C328R 또한 폴링 기반으로 제작되었다. 이를 적용한 대부분의 디바이스들의 MCU와 동기 비정확성으로 인해, 데이터 송수신을 위한 초기 동작 수행도 느릴 뿐만 아니라 재동기에 수초의 시간이 발생된다^[6].

그림 3은 카메라 모듈과 MCU, 타깃보드의 메시지 전송 과정이다.

A) 목적지에서 MCU로 동기화를 요청하면 MCU는

CMOS 이미지 카메라나 이미지센서에 동기화 메시지를 보낸다. 일정시간 동안 동기화 메시지를 MCU에 송신하면서 처음에 MCU로부터 수신된 메시지의 시간과 보오레이트를 확인하는 과정을 거치게 된다. 이때 한 번에 송신 메시지로 보오레이트를 확인할 수 없기 때문에 수십번의 동일한 메시지를 송신하면 보오레이트의 정확한 속도와 오실레이터 시간을 동기화동안 맞춘다. 기존 폴링 방식은 일정시간 동안 지속적인 MCU의 응답과 상관없이 계속 메시지를 전송하고 이후 시간에 수신 시간을 갖는다. CMOS 카메라에서 100ms씩 60번을 동기 메시지를 통해 동기화와 양방향의 보오레이트 설정을 확인한다. 만약, 수신 시간동안 MCU로부터 메시지가 오지 않는다면, 이와 같은 과정을 재반복 하게 되고, 4회 반복을 해도 응답이 없다면 연결 시도가 해제 되어 사용자가 재연결을 해야 한다. 테스트 결과 최소 1s부터 최대 10s의 동기화 시간이 필요했다.

그림 3과 같이 하나에 동기화 사이클에 대한 평균 동기화 시간은 다음과 같이 나타낼 수 있다. 그림 3, 4는 초당 k 개의 동기화 신호가 카메라로부터 MCU로 전달되고, 이 메시지의 개당 도달시간을 s_i 로 가정했을 때, 메시지 하나가 목적지까지 도달하는 평균 시간을 \bar{s}_i 라 정의할 수 있다. 생성되는 패킷은 Poisson 분포를 따른다고 가정한다. 전체 동기화 도달 시간의 집단을 U_i 일 때, U_i 를 $Unif(0, T)$ 에 따르는 $U_i = \bar{s}_1 + \bar{s}_2 + \dots + \bar{s}_n$ 를 iid(independent and identically distributed) 확률변수라 하고 $U_{(i)}$ 를 순서통계량(order statistic)이라 가정한다.

이들은 확률적으로 볼 때 수식 (1)은 동치이다.

$$U_i = \bar{s}_1 + \bar{s}_2 + \dots + \bar{s}_n = \sum_{i=1}^n E[U_i] = \sum_{i=1}^n E[U_{(i)}] \quad (1)$$

$S_m(T) = (0, T]$ 동안 수신된 메시지의 수이고, $Y(T)$ 는 동기화를 위한 누적된 수신 시간이라고 가정하면, 수식 (2)과 같이 나타낼 수 있다.

$$Y(T) = \sum_{i=1}^{S_m(T)} (T - \bar{s}_i) \quad (2)$$

$S_m(T) = n$ 면, 싱크 메시지의 수신 시간에 대한 누적 평균은 수식 (3)과 같이 정의 할 수 있다. 기존의 폴링 방식 동기화에서 동기시간(T)동안 일정하게 동기신호(λ)로 도착한다.

$$\begin{aligned} E[S_m(T)] &= E\left[\sum_{i=1}^{S(T)} (T - \bar{s}_i)\right] \\ E[S_m(T)] &= \sum_{n=0}^{\infty} E\left[\sum_{i=1}^{S(T)} (T - \bar{s}_i) | S_m(T) = n\right] \frac{e^{-\lambda T} (\lambda T)^n}{n!} \\ E[S_m(T)] &= \sum_{n=0}^{\infty} [nT - n \frac{T}{2}] \frac{e^{-\lambda T} (\lambda T)^n}{n!} \\ E[S_m(T)] &= \frac{\lambda T^2}{2} \end{aligned} \quad (3)$$

모든 동기 메시지 수신 후에 기존 폴링방식은 동기가 이루어진다. 이때, $S_m(T) = (0, T]$ 동안 동기를 못할 경우 재동기(k)의 동기화 시간 $S_m(T)$ 는 수식 (4)과 같이 표현된다.

$$S_m(T) = \frac{\lambda T^2}{2} k \quad (4)$$

단위 시간당 평균(T)은 동기화 시간당 평균을 노드의 동기 시간이라 가정할 수 있다. 또한 λ 에 \bar{s}_i 은 폴링 방식 임으로 주기적이다.

따라서 $U_i = \lambda = \sum_{i=1}^n E[U_i] = \sum_{i=1}^n E[U_{(i)}]$ 로 정의할 수 있으며, 기존 방식은 $\lambda = \bar{s}_1 + \bar{s}_2 + \dots + \bar{s}_{60}$ 임으로, 그러므로 수식 (5) 같이 나타낼 수 있다.

$$ES_m(T) = \frac{\lambda T}{2} k \quad (5)$$

그림 3의 B)는 타깃보드로부터 수신될 필요한 이미지 크기와 해상도 그리고 목적지 ID를 알려주는 역할을 한다. MCU에서는 필요한 이미지 크기에 맞게 프레임 크기를 정의 하고 이에 따른 보오레이트가 적합한지를 판단을 한다. 만약 보오레이트가 낮다면, 지연시간을 크게 주어 이미지 전송에 문제가 없도록 제공한다. 또한 타깃 보드가 요구하는 이미지 크기에 맞게 와내부의 SDRAM을 할당하는 역할을 한다.

C) 이미지 데이터를 순차적으로 수신하고, 수신된 이미지 ID에 맞게 SDRAM에 저장하고 이미지의 이상 유무를 체크한후 이것을 차레로 노드 ID를 통해 타깃보드로 전송하는 역할을 하게 된다. 데이터 전송을 위해서 보오레이트 속도에 따라 설정된 512k로 구성된 데이터 프레임의 전송 크기는 이미지 크기에 따라 6-19번으로 나누어 우선순위에 맞춰 전송하게 된다. 이 경우 NAK 같은 메시지를 수신하지 않기 때문에 잘못된 데이터 전송지연 및 무결성에 재전송 요청, 복원이 불가능하다.

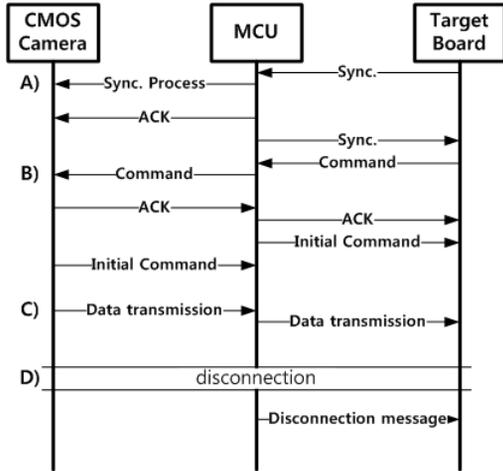


그림 3. 데이터 및 메시지 전송 과정

D) 데이터 전송중에 연결 해제(Disconnection)가 된다 면, MCU는 타깃보드에 연결 해제 메시지를 보내고, 사용자의 연결 요구를 기다린다. 모듈간의 동기화가 된 상태에서도 폴링 방식으로 인해, 동기화가 단락되어 재동기화가 되지 않는 등 목적지와 카메라의 동기화가 견고하게 이루어지지 않는다.

이렇듯이 순차적 폴링 방식은 중간에 명령을 제어하거나 ACK가 느릴 경우 C328R에서 명령을 수신하고 처리하는데 카메라 모듈의 작업이 끝나칠 때까지 MCU는 다른 부분을 수행할 수 없다. 또한 폴링시간간격(Polling Interval Time)에 종속되어 잘못된 폴링간격으로 인한 동기화 시간이나 실질적 데이터 송수신에 악영향을 줄 수 있다. 폴링 메시지는 MCU 상태가 변할 때까지 상태 레지스터를 통해 계속 확인해야 하는 비효율성을 가지고 있다. 폴링 방식의 문제점을 해결하기 위해 제공 소프트웨어에서는 내부 타이머 및 연결된 오실레이터의 시스템 타이머를 통해 MCU 내부나 C328R의 명령이 수행되는지를 확인한다. 하지만 내부 타이머를 사용한 접근 제어 문제에서도 근본적인 폴링방식의 구조적 안정성(structural stability)을 해결한 것이 아니기 때문에 동기화 이후의 강건성(robustness)을 보장하지 못한다. 그런 이유로 실 제품에서는 데이터 수신도 일정시간 이상이 지나면 동기화가 해지되는 문제도 발생하게 된다.

3. Stepwise 동기화 방식 및 하드웨어 구조

본 문에서는 기존의 폴링 주기를 세분화하여 불필요한 메시지 교환을 최소화함으로써 제안된 Stepwise 동기화에

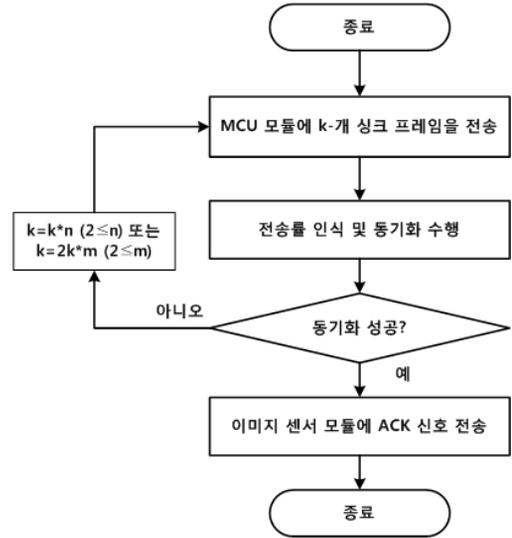


그림 4. 제안 Stepwise 동기화 기법

대한 설명을 하고, 데이터 전송에 의한 재전송 오버헤드를 줄이기 위해 인터럽트 방식을 효율적인 데이터 전송을 주요하게 제안하고 이를 구현하였다.

제안된 것과 같이 그림 4, 6은 기존에 비해서 세분화된 동기화를 Stepwise 기법으로 정의하고 이를 통해, 동기화에 소요되는 싱크 프레임의 개수를 적응적으로 조절하여 동기화에 소요되는 시간을 줄일 수 있다. 카메라의 보오레이트 확인은 연속적인 데이터를 안정적으로 보내기 위한 방법으로 사용된다. 하지만 MCU의 프레임 변화를 통해 보오레이트를 안정화 시킨다면 충분히 빠르게 동기화를 시킬 수 있고, 메시지의 손실을 줄일 수 있다. 그렇다고, 카메라와 MCU의 보오레이트를 위한 동기 메시지를 현저하게 줄일 수 없기 때문에 실제적으로 가장 빠른 동기화 메시지 수를 Stepwise 동기화 기법에서는 10번으로 제시하였으며, 이진 수백 번 실험을 통해 최적의 값을 찾은 것이다. 보편성을 위해 제안된 방식을 구현 후 여러 대의 PC 및 임베디드 보드에 동일하게 장착하여 테스트를 하였다. 구현된 기법에서도 보듯이 10번이 최적화이지만, 이 경우는 가장 좋은 싱크시간이며, 그림 6의 B)과정까지 반복되는 경우가 일반적이다.

그림 4는 제안된 Stepwise 동기화 기법을 나타내었다. 처음 기존에 비해 1/6수준의 k개의 싱크 프레임을 전송을 하고 MCU로부터 ACK를 짧은 시간동안 수신하고 만약 ACK가 없다면, k개의 싱크 프레임을 재전송을 하고 ACK를 기다린다. 이 과정을 $n \leq 2$ 이 이루어진 후 k개의 싱크 프레임을 2배로 증가해서 다시 $m \leq 2$ 번을 반복하여 전송

하게 된다.

기존 방식과 비교한다면, 그림 6과 같이 동기화 구간 A), B), C) 3가지로 구분하고, $S_m(T) = (0, T]$ 는 수식 (7)로 표현될 수 있다.

$$ES_m(T) = (S_A(T) + S_B(T) + S_C(T) + 3I_{ACK})k \quad (7)$$

여기서 $3I_{ACK}$ 는 카메라로부터 대기하는 ACK이며 발생하는 대기시간은 전체 동기화 시간에서 무시 될 정도로 작다.

$$ES_m(T) = \left(\sum_{i=1}^{10} \overline{s_i} + \sum_{i=1}^{10} \overline{s_i} + \sum_{j=1}^{20} \overline{s_j} + 3I_{ACK} \right) k \quad (8)$$

$S_A(T)$, $S_B(T)$, $S_C(T)$ 에 대한 동기시간은 $S_A(T)$, $S_B(T)$, $\frac{1}{2}S_C(T)$ 이다. 이를 λ_1 , λ_2 , λ_3 라 할 때, $S_A(T) = \overline{s_1} + \overline{s_2} \dots + \overline{s_{10}}$ 고, $S_A(T) = \lambda_1$ 는 $\lambda_1 = \overline{s_1} + \overline{s_2} \dots + \overline{s_{10}}$ 이다. $S_A(T) = S_B(T)$ 동치이고, $S_A(T) = \frac{\lambda_1 T}{12}$ 면 각각에 대해 $\frac{\lambda_1 T}{12}$, $\frac{\lambda_2 T}{12}$, $\frac{\lambda_3 T}{6}$ 이다. $\overline{s_i}$ 와 λ 는 폴링에서 동일한 메시지 도착 시간 간격을 지니기 때문에, 수식 (9)로 나타낼 수 있다.

$$ES_m(T) = \left(\frac{\lambda T}{3} \right) k \quad (9)$$

그림 5는 기존 폴링방식인 수식 (5)와 제안된 Stepwise 방식인 수식 (9)에 대해 성능 비교를 하였다. 기존 방식보다 제안된 방식에서 동기화시간이 길어질수록 상대적으로 짧은 시간에 동기화가 이루어진다는 것을 알 수 있었다.

그림 6 A)는 100ms 이내 시간동안 k번의 동기 메시지를 보낸 후 10ms 동안 C328R로부터 ACK 메시지를 기다린다. 수신 메시지가 없다면, 이 과정을 B)와 같이 n번 반복한다. C)는 2k번의 동기 메시지를 보낸 후 10ms 동안 C328R로부터 ACK 메시지를 기다린다. 이후 재반복을 1회 하고 이후에도 ACK가 없고, 2100ms 동안 동기화가 되지 않는다면, 처음으로 동기화 과정을 돌아가 반복한다. nst sync frame 부터는 동기화 중에 CMOS 이미지 카메라에서 ACK가 수신될 경우 반복하는 과정에서도 모든 과정의 동기화가 이루어졌다고 가정하고 타깃보드에 영상정보를 보낼 수 있는 메시지를 준비한다. 바로 타깃보드에 ACK 메시지로 전송 가능 상태를 알려준다. 이렇

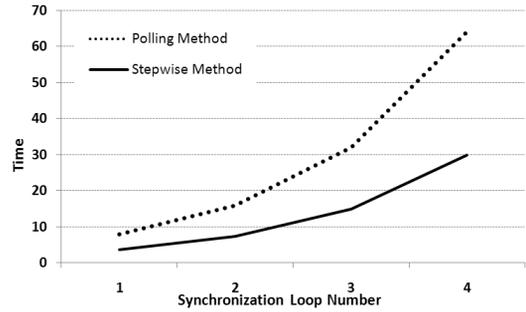


그림 5. 폴링방식과 제안된 Stepwise 성능 비교

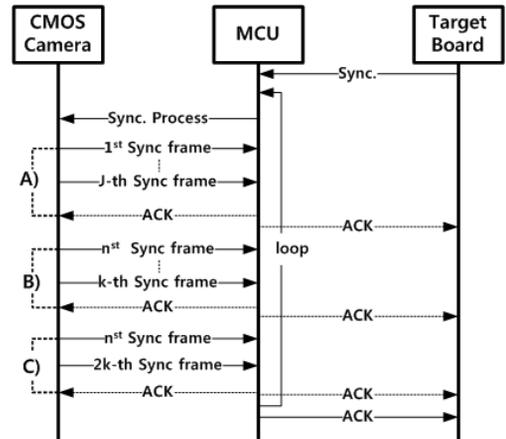


그림 6. 제안된 세분화된 동기화과정

게 세분화된 Stepwise 동기화 기법을 통하여 좀 더 빠른 동기화를 수행 할 수 있다.

제안된 인터럽트 방식에서는 그림 6,7과 같이 C328R에서 데이터를 송수신할 때 ATmega128의 입출력 인터럽트 벡터를 I/O에서 시리얼 통신의 동기화 메시지를 기존보다 짧은 시간동안 반복하면서 접속 상태를 확인한다. 그림 7 A)는 타깃보드에서 명령 없이 동기가 해제(disconnection)되면 ATmega128에서 수신되는 프레임은 확인 후 카메라에 재동기(re-connection)를 요청한다. MCU내에서의 데이터 정보를 다른 UART를 통하여 전송하며, 데이터 프레임의 전송 완료나 수신 프레임에 문제가 발생할 때 처음 명령을 인터럽트를 통해 확인한 후 재설정 할 수 있다. 또한 재접속 기간동안 타깃 보드에서 메시지에 대한 요청이 발생하기 때문에 MCU는 NAK(Negative Acknowledge)를 통해 목적지로부터 불필요한 메시지를 수신하지 않도록 하였다. 그 밖에도 그림 7과 같이 수신 데이터에 의한 시간(Time Slice)동안 목적지 보드로부터

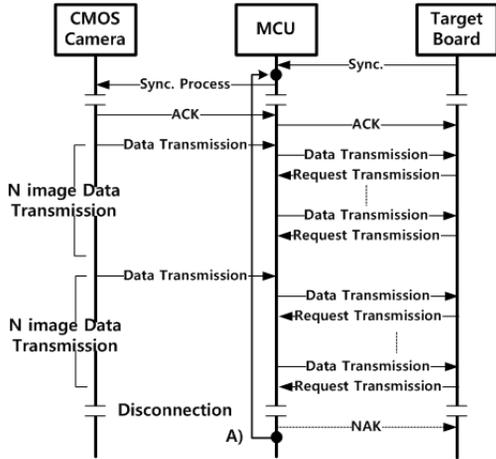


그림 7. 제안한 재동기화 및 전송 기법

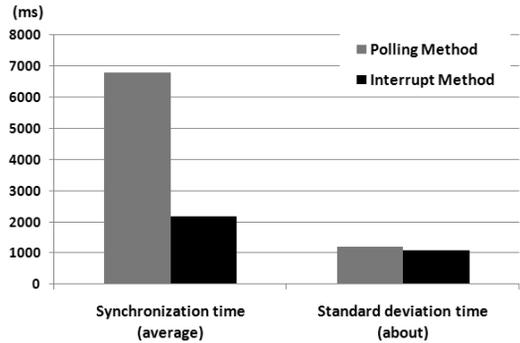


그림 9. 제안된 방식의 동기화 시간 비교

는 전후로 송수신된 프레임에 Frame Counter를 추가하고 I/O 인터럽트를 통해 지속적으로 확인하였다.

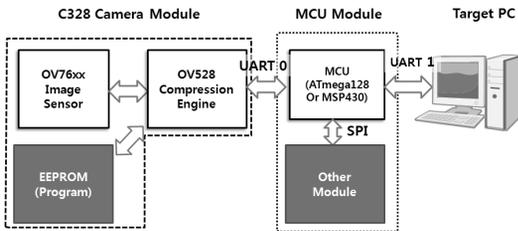


그림 8. 제안된 카메라 모듈과 MCU 구조

MCU에 추가 요청이 수신되면, NAK 명령을 통해 MCU의 Busy 상태를 응답하여 메시지 수신을 방지하였다.

그림 8과 같이 구현된 디바이스는 외부의 SDRAM이 필요 없다. C328R과 ATmega128만 통해서 저전력 구동(88mW/h 이하)이 가능하다. 뿐만 아니라 기존에 제공되던 폴링 방식을 개선하기 위해 인터럽트 기반의 펌웨어를 개발하였다. 앞서 말한 것과 같이 폴링 방식의 이미지 크기가 2-8k 정도로 저장영역이 부족한 MCU의 경우 직접 데이터 전송으로 외부메모리 없이 전달 할 수 있지만, 내부 메모리가 작을 경우 전체 이미지 데이터가 전송되기 전 버퍼에 가득 차게 되어 이후에 들어오는 프레임이 삭제되거나 수신 지연이 발생된다. 구현된 것은 내부의 작은 메모리만으로도 외부 신호 인터럽트 및 인터럽트 I/O를 이용하여 동기화 유지 및 C328R에서의 이미지 데이터를 MCU로 수신하는 동시에 타깃보드로 송신을 이룰 수 있었다. C328R로 수신된 프레임 데이터는 지연 없이 직접 타깃보드로 전송되고 중간에 연결된 MCU의 SDRAM은 전후의 프레임 버퍼에 포함된 1536k의 메모리영역을 사용하게 된다. 프레임의 정확한 전송을 위해 MCU에서

4. 성능평가

구현된 디바이스의 성능 평가에는 COMedia사에서 제품의 제공 소프트웨어와 제안한 기법에서의 수신 프레임 크기와 카메라 모듈과 동기화 시간 및 에러 커넥션(Error Connection)을 비교하였다. 압축 모델에 320x240 픽셀을 사용한다. 320x240은 QVGA이면서 모바일 단말기나 서비스 측면에서 매우 유용하게 사용되는 해상도이다. 이미지는 2k~8k의 크기를 지닌다. 벤치마크는 동일 조건에서 100번 이상의 테스트를 하였다.

그림 9는 그림 1과 같이 구성된 시스템에서 MCU와 카메라 모듈과의 평균 동기화 시간을 비교한 것이다. 기존 제공된 소스보다 2배 이상 빠른 결과를 보여준다. 제안된 Stepwise 동기화 기법의 접속시간은 표준편차까지 고려하여도 제공 소프트웨어의 최적의 커넥션 시간보다 빠른 동기화를 보인다. 제공 소프트웨어의 동기화 시간은 가장 빠를 때 5초 내외로 이루어졌지만, 제안된 방식에서는 4초 이전(Worst)에 이루어졌다.

그림 10¹⁾에서는 수신되는 프레임 당 전송시간과 목적지에 도착되는 시간을 비교하였다. 인터럽트 펌웨어 기반이 조금 빠른 것으로 나타났으며, 전체 프레임 수신에서도 제안한 방법이 좀 더 빠른 것으로 나타났다. 카메라 모듈과 동기화 이후의 에러 커넥션 및 동작시간을 나타내었다. 즉, 폴링 방식의 모델보다 데이터 수신에 견고성에서도 제안된 인터럽트 방식의 기법이 오랜 시간 동안 동기화 에러

1) 이미지는 동일한 화면을 두고 수신을 하였으나 10%까지 데이터 크기가 차이 날수 있음.

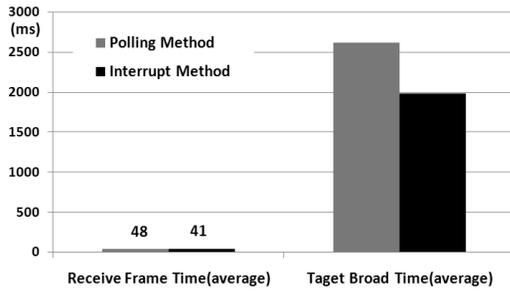


그림 10. 프레임당 전송 시간 및 보드 전송시간

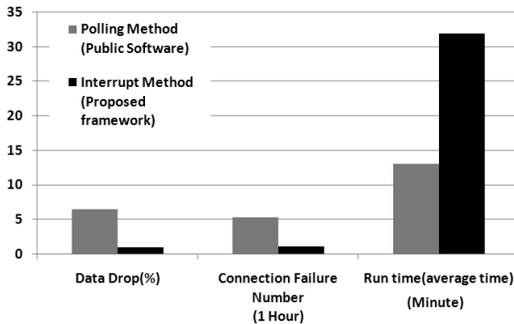


그림 11. 연결 에러 및 이미지 전송 연속 시간

없이 이미지 데이터를 연속적으로 제공할 수 있다.

그림 11은 이미지 데이터 손실 및 접속해제 없이 계속 유지되는 횟수와 시간을 나타내었다. 접속 유지시간도 제안된 재접속 기법으로 인하여, 에러 없이 오랜 시간동안 유지되었다. 또한 데이터 수신 손실(Data Drop)도 시간당 1% 정도로 매우 낮은 것을 알 수 있다.

5. 결 론

본 연구는 이미지 센서를 통해 싱크 프레임 전송 부분은 전송 타이밍과 보오레이트에 맞게 설정하였다. 그리고 동기화에 소요되는 싱크 프레임의 개수를 적응적으로 조절하여 동기화에 소요되는 시간을 개선하고 MCU를 통한 펌웨어 기반에 동기화 기법과 인터럽트 방식으로 재접속 유지 및 데이터 전송의 효율성을 제안하였다. 이를 통해 20~130%의 빠른 동기화의 향상과 타겟보드까지의 전체 데이터 프레임 수신에서도 10%의 향상을 가지고 왔다. 또한 접속유지의 향상으로 강건성 보장과 데이터 수신에 대한 손실율도 감소하였다. 부가적인 연구 결과로 MCU의 작은 내부 메모리만으로도 이미지 데이터를

전송할 수 있었다. 마지막으로 국내에 점차적으로 연구 및 판매가 증가되고 있는 압축가능 CMOS 이미지 센서 카메라 모듈과 ATmega128의 구현을 실질적으로 보여줌으로써 제조사에서 제공된 폴링 방식에 비해, 인터럽트 방식이 동기화 및 접속유지 측면에서 뛰어난 결과를 보여 주었으며, 이는 실증적 평가에 의의를 둘 수 있다.

참 고 문 헌

1. *The CMUcam Vision Sensors*. <http://www.cs.cmu.edu/cmucam/>, 2008.
2. M. Rahimi, R. Baer, Obimdinachi I. Iroezzi, Juan C. Garcia, Jay Warrior, Deborah Estrin, and Mani Srivastava, "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, November. 2005.
3. *COMedia Ltd., C328R User Manual*, June, 2007.
4. Ikdong Kim, Jaechang Shim, Jason Schlessman and Wayne Wolf, "Remote wireless face recognition employing zigbee," *DSC 2006, in conjunction with ACM SenSys 2006*, October 2006.
5. Chulsung Park, Pai H. Chou, "eCAM: ultra compact, high data-rate wireless sensor node with a miniature camera," *ACM 2006*, pp. 359-360, 2006.
6. Changsu Suh, Zeeshan Hameed Mir, and Young-Bae Ko, "Design and Implementation of Enhanced IEEE 802.15.4 for Supporting Multimedia Service in Wireless Sensor Networks," *Special Issue on Wireless Multimedia Sensor Networks, Elsevier Computer Network Journal*, Vol. 52, No. 13, pp. 2568-2581, September, 2008.
7. *TinyOS Korea Forum*, <http://www.tinyos.re.kr/>
8. *Berkeley, CA*. <http://www.tinyos.net/>
9. Purushottam Kulkarni, Prashant Shenoy and Deepak Ganesan, "Approximate Initialization of Camera Sensor Networks," *EWSN 2007, LNCS 4373*, pp. 67-82, 2007.
10. Stephan Hengstler and Hamid Aghajan, "WiSNAP: A Wireless Image Sensor Network Application Platform," *TridentCom*, pp. 6-12, March 2006.
11. Thiago Teixeira, Dimitrios Lymberopoulos, Eugenio Culurciello, Yiannis Aloimonos, and Andreas Savvides, "A Lightweight Camera Sensor Network Operating on Symbolic Information," *DSC06*, pp. 87-92, October 2006.
12. Purushottam Kulkarni, Deepak Ganesan and Prashant Shenoy, "The Case for Multi-tier Camera Sensor Networks," *Proceeding of the 13th annual ACM international conference on Multimedia*, pp. 141-146, 2005.

13. Cintia B. Margi, Vladislav Petkov, Katia Obraczka and Roberto Manduchi, "Characterizing energy consumption in a visual sensor network testbed," *TRIDENTCOM 2006*, pp. 8-16, March 2006.

14. Chulsung Park and Pai H. Chou, "eCAM: Ultra Compact, High Data-Rate Wireless Sensor Node with a Miniature Camera," *ACM SenSys*, pp. 359-360, November 2006.



박 현 문 (kimagu@naver.com)

2004 한세대학교 정보통신학부 공학사
2006 국민대학교 전자공학과 정보통신학 공학석사
2006~2008 국민대학교 BIT 비즈니스 정보통신 박사수료
2008~현재 전자부품연구원 연구원

관심분야 : 위치인지, USN, WLAN, 해양 통신



박 수 현 (shpark21@kookmin.ac.kr)

1988 고려대학교 컴퓨터학과 이학사
1990 고려대학교 대학원 전산학 이학석사
1998 고려대학교 대학원 컴퓨터학 이학박사
1990 (주) LG 전자 중앙연구소 선임연구원
1999~2001 동의대학교 공과대학 컴퓨터소프트웨어공학과 조교수
2002~현재 국민대학교 비즈니스 IT학부 부교수

관심분야 : USN, Underwater Sensor Network



이 명 수 (leems@keti.re.kr)

1989 아주대학교 전자공학과 공학사
2007~현재 광운대학교 정보디스플레이학과 석사과정
1989 현대 전자 R&D 센터 연구원
1996~현재 전자부품연구원 책임연구원

관심분야 : USN, ITS, 임베디드 시스템



서 해 문 (bmms@keti.re.kr)

2000 경북 대학교 전자공학 공학석사
2002~2004년 삼성전자 통신연구소 R&D 센터
2004~현재 전자부품연구원 선임연구원

관심분야 : RFIC, RF 시스템, 통신시스템



박 우 출 (wcpark@keti.re.kr)

1997 한양대학교 전자공학 공학석사
2002 한양대학교 전자공학과 공학박사
2002~현재 전자부품연구원 선임연구원

관심분야 : ZigBee, 보안 시스템, USN, Mesh Networking



장 윤 정 (aldidncjsrn@naver.com)

2008 대구대학교 전자시스템공학 공학사
2008~현재 전자부품연구원 연구원

관심분야 : 위치인지, USN, 임베디드 시스템