

## 모바일 센서 차량 Hardware의 설계

박재영<sup>o</sup>, 이현승, 송하윤, 박준

홍익대학교 컴퓨터공학과

pjyoungs@hotmail.com, lhseung@oodb.cs.hongik.ac.kr, hayoon@wow.hongik.ac.kr,  
jpark@hongik.ac.kr

## The Design of Mobile Sensor Vehicle

Jae Yeong Park<sup>o</sup>, Hyun Seung Lee, Ha Yoon Song, Joon Park

Department of Computer Engineering, Hongik University

### 요 약

본 논문에서는 각종 센서를 탑재한 차량을 센서 네트워크의 노드로써 움직이는 하드웨어를 구현하였다. 센서 네트워크에서 이동할 수 있는 노드들은 주위의 환경을 인지하여 대처하며 그에 대한 정보를 컴퓨터에 전달하여 차량이 있는 곳의 지도를 만들어 낸다. 이러한 센서 차량의 하드웨어를 만들기 위해 정면에 존재하는 장애물을 인지하는 라인레이저와 그것을 인지할 수 있는 카메라, 초음파 센서를 설치하였다. 절대적인 지표 없이 정확한 좌표를 알아내기 힘들고 차량 이동 시 미끄러짐 등과 같은 오차가 생길 수 있으므로 정확한 방위각을 알아낼 수 있는 디지털 나침반을 차량 설계에 포함 시켰다.

### 1. 서론

모바일 센서 차량들은 네트워크상의 움직이는 노드를 표방한다. 각각의 여러 센서 차량들은 개별적으로 움직이며 주위를 탐색하고 서로 정보를 교환하여 주변 환경에 대한 지도를 만든다. 그러기 위해서는 오차 없이 움직일 수 있어야 하며, 주위의 장애물을 식별할 수 있어야 하고, 다른 차량에게 자신의 존재를 확인 시켜 줄 수 있어야 한다.

본 논문에서는 모바일 센서 차량의 하드웨어 부분을 설계하여 센서 네트워크의 노드로 동작할 수 있도록 구현 하였다.

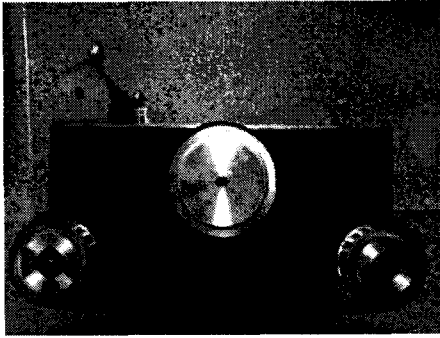
본 논문은 2장에서 모바일 센서 차량의 구성을 살펴보고 3장에서 센서 간의 통신 규약을 알아본다. 4장에서 모터와 나침반, 서보 모터, 초음파 센서, P3P의 제어 및 적용 방법에 대해 알아보고, 5장에서 결론을

맺는다.

### 2. 모바일 센서 차량의 구성

차량의 몸체는 알루미늄 재질의 합판을 사용하여 내구성을 높이며, 무게를 가볍게 하였다. 바퀴는 각각 3개의 바퀴와 L형 고무벨트, 기어를 사용한 [그림 1]과 같이 캐터필러 방식을 채택하여 지형의 영향을 덜 받으며 미끄러짐을 통한 이동거리 오차를 최소로 줄였다.

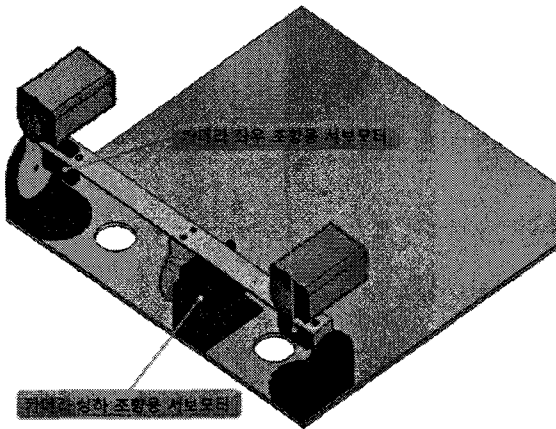
차량의 정확한 위치를 추정하기 위하여 디지털 나침반을 차량의 상단에 배치하여 차량이동 시 오차를 보정하며, 차량 상호간 자기 위치의 정확한 판단을 내릴 수 있게 하였다. 또한 모터에 의한 전자기장의 발생으로 나침반에 오차가 발생 할 수 있다는 것을 감안하여



[ 그림 1. 모바일 센서 차량의 측면 ]

바디의 윗부분을 0.3mm 철판을 보강하였다.

차량 정면의 적외선 라인레이저를 설치하고, 적외선 필터를 설치한 카메라 1대와 일반 카메라 1대를 상단에 설치하여 장애물을 인식하게끔 만들었다. 카메라를 고정하는 몸체에 [그림 2]와 같이 서보 모터 3개를 사용하여 상하, 좌우의 움직임을 줄 수 있게 만들어 피아식별 및 장애물 파악을 보다 원활히 할 수 있게 만들었다. (이것을 스테레오 아이라 정의하기로 한다.)



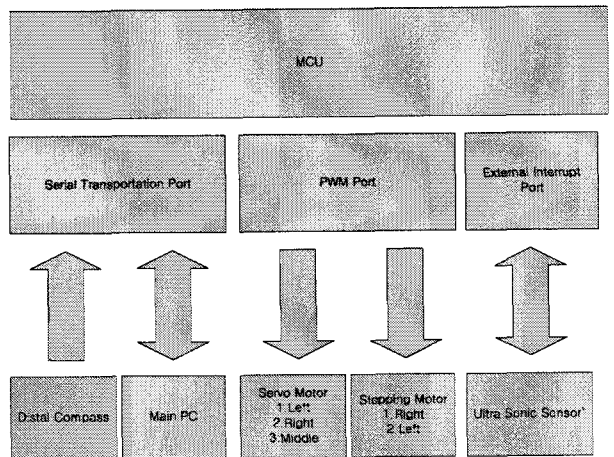
[ 그림 2. 스테레오 아이의 프레임 ]

또한 적외선 특성상 검은 물체의 식별이 힘든 점과 레이저의 직진성으로 인한 낭떠러지 및 공중에 있는 물체의 식별이 어려운 점을 보완하기 위하여 바닥을 지향하는 초음파 센서 1개, 라인레이저의 상단에 위치한 장애물을 인식하는 초음파 센서 2개를 배치하였다.

다른 차량과의 정확한 상호간의 위치정보가 필요 할 경우 원거리에서 직접 차량을 카메라로 확인하여 정보를 알아낼 수 있게 하기 위하여 피아식별을 할 수 있는 P3P를 도입하여 3개의 적외선 LED를 삼각형으로 설치하였다.

[그림 3]은 MCU와 상호 작용하는 모든 하드웨어 들의 데이터 흐름도를 표현 하였다.

관련된 모든 하드웨어는 Freescale 사의 MC9S12DG256B MCU를 탑재한 작은 Embedded Board로 제어한다. 이에 대한 소프트웨어는 Metrowerks 사의 CodeWarrior for HCS12 및 Editplus 이며, 디버깅 작업은 CodeWarrior 와 BDM Multilink 를 통하여 이루어졌다.



[ 그림 3. 데이터 흐름도 ]

### 3. 통신 규약

MCU는 현재 이동 정보 및 장애물 판단, PC는 MCU가 보내주는 데이터를 가지고 고차원적인 상황판단을 함으로써 효율성을 극대화 시켰다. MCU와 PC상호간의 통신으로 수단으로써 시리얼 통신을 사용하였다. 이는 정확하고 간단하지만 8bit단위의 정보만을 전송할 수 있어 약속된 통신 규약이 필요로 한다. 먼저 최상위 비트 7과 6을 무엇을 제어하겠는가를 판단하는 bit로 정의한다.[표1]

Bit 7	Bit 6	Mode
0	0	Motor
0	1	Servo Motor
1	0	Digital Compass
1	1	Distance

[ 표 1. Bit 7, Bit 6의 정의 ]

나머지 bit에 관한 정보는 각 부분에 대한 세부사항을 언급할 때 명시하겠다. PC 와 MCU간의 시리얼 통신을 하기 위한 기본설정은 9600bps, 8bit 데이터 length, stop bit = 1 과 같다.

#### 4. 구성 요소의 제어 및 적용

##### 4.1. 모터의 제어

모터는 2상 스테핑 모터를 사용했으며, 1:3 비율의 감속기어를 장착하였다. MCU를 통하여 8개의 PWM 포트를 사용할 수 있는데 PWM 펄스의 주기가 짧아질수록 모터는 빨리 돌아가게 된다. MCU의 PWMPER[0:7]까지의 8개 레지스터로 펄스의 주기를 결정할 수 있고, 이 주기의 기본 시간 단위를 결정해주는 Prescaler 레지스터인 PWMSCLA, PWMSCLB 의 값은 3으로 설정하였다. 이 Prescaler 의 값을 작게 하면 더 빠른 속력을 낼 수 있지만 너무 줄여주게 되면 느린 속력에서의 정밀한 제어가 어렵게 되며, Prescaler 의 값을 너무 크게 하면 빠른 속력을 낼 수 없게 된다. 시리얼 통신으로 주고 받는 데이터의 정의는 [표 2]와 같다.

Bit 7	MODE = MOTOR (0 0)
Bit 6	
Bit 5	Right Direction
Bit 4	Right Speed
Bit 3	
Bit 2	Left Direction
Bit 1	Left Speed
Bit 0	

[ 표 2. Motor제어의 Bit정의 ]

Bit 5, Bit 2는 왼쪽과 오른쪽 바퀴의 진행 방향을 나타내는데 0이면 앞으로 1이면 뒤로 가도록 정의하였다. Bit 4, Bit 3, Bit 1, Bit 0은 각각 0~3 단계로 속도를 제어 할 수 있다. 이동 거리를 저장하기 위하여 각각의 바퀴의 PWM을 MCU의 pulse의 개수를 세어주는 Pulse Accumulator에 연결시켜 한 개의 Pulse 발생시 이동하는 거리를 측정하여 그에 따른 총 이동 정보를 MCU의 RAM상에 저장하고 있다가 PC에서 거리에 대한 정보를 [표 3]과 같이 요청하면 이동거리를 전달한다.

Bit 7	Bit 6	Bit 5~0
1	1	Don't care

[ 표 3. 이동거리 요청 패킷 ]

##### 4.2. 나침반의 제어

차량에 장착된 나침반은 OCDMC-0601M[2]으로써 MCU와 시리얼 통신을 한다. OCDMC-0601M은 모듈 자체에서 연산된 방위각을 출력한다. 이때, 출력되는 값은 TTL LEVEL이며, RS-232C 규격에 의하여 컴퓨터 또는 MCU에 연동하여 방위각을 숫자로 표현할 수 있다. 또한 2축 방식에 의한 자북을 표현하며, 정북과는 일반적으로 11.5도 벗어난 값을 표현한다. 0.1도의 분해능을 갖고 있으며, 정확도는 CALIBRATION후 ±1.5 이내이며, 3.5단위 즉 000.0도에서 359.9도로 방위각이 표시된다. Broadcasting 방식으로 ASCII CODE의 숫자와 마지막 NULL 문자를 포함한 문자열 5 byte를 계속적으로 보내 준다. PC와 MCU의 통신 규약에 따라 [표 4]에 대한 정보가 PC로부터 들어오면 그때 나침반이 보내고 있는 정보를 전달한다.

Bit 7	Bit 6	Bit 5~0
Compass mode(10)		Don't care

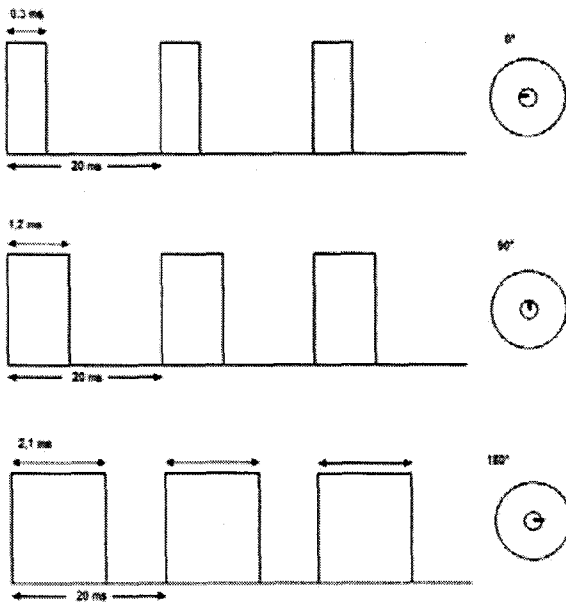
[ 표 4. 나침반 데이터 요청 패킷 ]

##### 4.3. 서보 모터의 제어

Stereo Eye의 움직임을 위하여 장착된 서보 모터는 상하를 움직이게 하는 토크가 상대적으로 큰 것을 사용

하였고 각각의 카메라에 대한 좌우 회전은 크기가 작은 서보 모터를 사용하였다.

서보 모터는 PWM의 duty ratio가 1.5 ms일 때 정 중 앙을 가리키며 0.7ms~2.3ms사이에서 움직임을 나타낼 수 있다. 현재 사용하고 있는 MCU의 PWM의 duty ratio에 대한 resister는 PWMDTYx 로써 x는 해당 핀에 대한 번호를 나타내며 8bit 단위로 제어할 수 있다. 하지만 8bit로 제한한다면 180도의 각도를 전부 제어할 수 없으므로 16bit로 제어할 수 있는 PWMDTYxx를 사용하였다. [표 5]는 MCU의 Resister사용 현황을 표시했다.



[그림 4. 서보 모터의 제어 타이밍 ]

PWMDTY23	Right Servo
PWMDTY45	Middle Servo
PWMDTY67	Left Servo

[표 5. PWM의 duty ratio에 대한 Resister ]

오른쪽 서보 모터와 왼쪽 서보 모터는 좌우 회전을 담당하는 서보 모터로, 중간 서보 모터는 상하 움직임을 담당하는 서보 모터로 정의한다. Main PC가 서보 모터를 제어하기 위하여 보내는 명령은 2byte를 보내며 내용은 <표 6>과 같다.

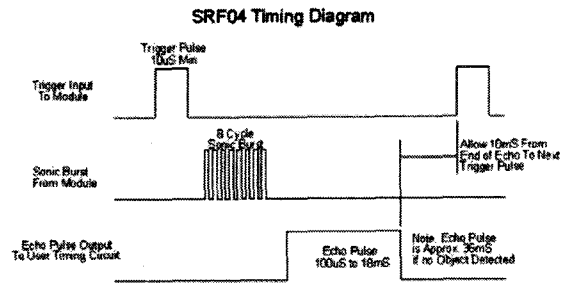
Bit 7	Servo Mode(01)
-------	----------------

Bit 6	Which Servo?
Bit 5	
Bit 4	
Bit 3	If Bit5, Bit4 = 3
Bit 2	
Bit 1	Otherwise
Bit 0	High 4 Bits

[ 표 6. Servo 제어 요청 패킷 ]

Bit5, Bit4에서 어떤 서보 모터를 선택하는 것인지 선택하고(0:Right, 1:Middle, 2:Left) 나머지 bit는 하위 4bit를 담당한다. 그 다음 1byte에 대한 정보는 Bit5, Bit4가 3일 때 2번째 bit라고 선언하는 flag이며 나머지 bit가 하위 bit로 정의된다. 2번째 byte가 도착함과 동시에 상위 bit와 하위 bit를 통합하여 8bit의 정보로 70~230까지를 표현하는데 이것은 70\*10~230\*10[US] 라는 정보를 나타낸다.

#### 4.4. 초음파 센서의 제어



<그림 5> 초음파 센서의 제어 타이밍

장애물 회피를 보완하기 위하여 장착된 초음파센서 [2] (SRF04) 는 1개의 입력 핀과 1개의 출력 핀 그리고 전원을 공급하는 2개의 핀으로 구성되어있다. <그림 5>와 같이 초음파 센서의 입력핀에 10µs의 pulse를 주면 sensor가 초음파를 발생 시킨다. 초음파발생을 멈추는 순간 output pin에 rising edge가 생기는데 이는 초음파가 장애물에 의해 반사되어 나오는 것이 감지되는 순간 falling edge가 발생한다. MCU는 이 pulse를 external interrupt에 의해 Rising edge가 발생하는 시간과 falling edge가 일어나는 시간의

차이를 계산하여 장애물과 차량간의 거리를 구한다.

$$\text{Pulse Width [uS]} / 58 = \text{Distance [cm]}$$

$$\text{Pulse Width [uS]} / 148 = \text{Distance [inches]}$$

만약 차량의 앞에 적외선 라인 레이저로 발견하지 못한 장애물을 인지했을 경우, PC에게 <표 7>과 같이 현재 상황을 전달한다.

왼쪽	'L'
오른쪽	'R'
아래(바닥이 없는 경우)	'D'

[ 표 7. 장애물 데이터 전달 ]

다른 데이터는 요청에 의해서만 전달 되지만 초음파 센서에 의한 데이터는 MCU의 판단에 의해 PC로 전달 된다. 그래서 정보의 모호함을 막기 위하여 위의 Character를 보내기 전에 255를 먼저 보내어준다.

#### 4.5. P3P의 적용

피아 식별을 위한 장치로써 3각형의 적외선 LED가 차량마다 다른 의미 있는 정보로 깜빡 꺼린다. 이는 카메라의 프레임 수에 의존적인데 초당 30프레임을 소화할 수 있는 것에 대해서는 0.1초 정도의 주기를 주어 켜짐(1)과 꺼짐(0)의 반복으로 시리얼 통신과 비슷하게 정보를 줄 수 있다. 또한 3각형의 실제 거리를 알면 원거리에서 다른 차량을 발견 하였을 때 그 차량과의 거리를 알 수 있다.

#### 5. 결론

본 논문에서는 각종 센서들이 MCU를 통하여 서로를 보완해 가며 현재의 상황을 PC와 시리얼 통신하는 방법을 구현하였다. 센서에 의한 정보를 MCU가 분석하고 통합하여 PC의 요청에 의해서 혹은 MCU의 판단에 의해서 데이터를 보낸다. 이에 따라 역할분담이 확실하게 되어 좀더 고차원적인 일을 PC가 할 수 있게 되어 효율성이 커진다. 하지만 디지털 나침반의 경우 기울어짐에

의해 데이터의 오차가 심하게 나타나는 것을 확인할 수 있었다. 따라서 향후 연구과제로 위치를 파악하는데 있어서 나침반에 대한 의존성을 낮추거나 기울기에 따른 정확한 계산을 구하는 연구를 계획 하고 있다.

#### 6. Acknowledgement

본 연구는 『서울시 산학연 협력사업』 (Seoul R&BD Program)의 일환으로 수행하였다.

#### 7. 참고문헌

- [1] Parallax, Inc. SRF04 Ultrasonic Range Finder Datasheet.
- [2] Opsel, Inc. OCDMC-0601M Digital Magnetic Compass Datasheet.