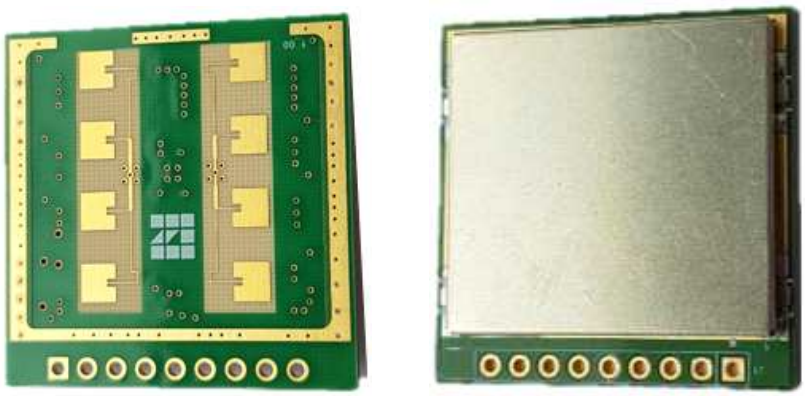
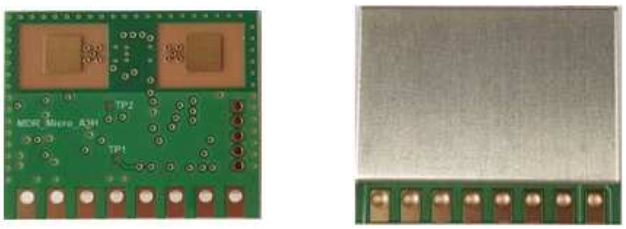
Documentary

MDR Series(Mini_A, Micro_A)

Arduino Example Manual



[MDR_Mini_A]



[MDR_Micro_A]

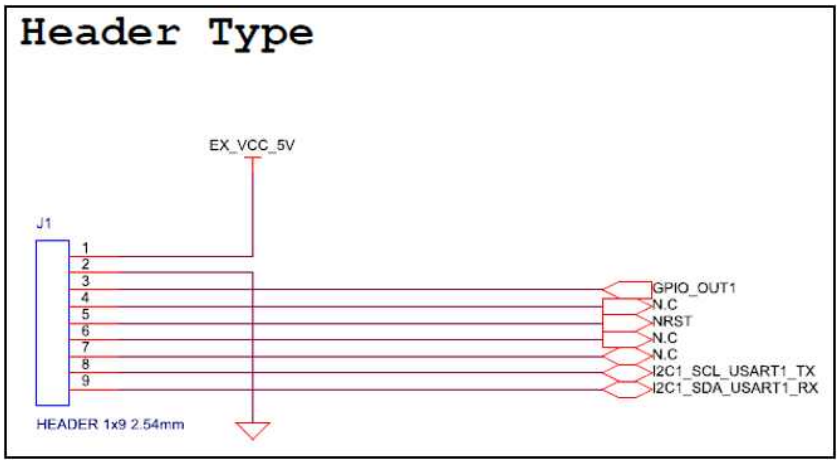
[목 차]

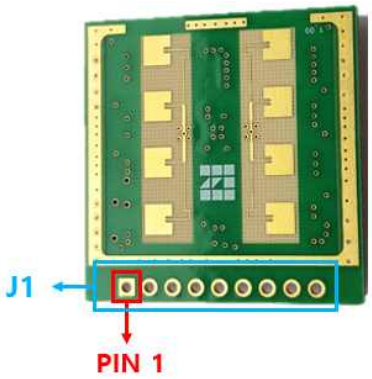
1. MDR Series 동작 원리

- MDR은 Motion Detect Radar을 의미함
- MDR은 전방의 움직임을 감지하여 기본적으로 움직임이 있을 시 High, 없을 시 Low 신호가 3번 Pin에서 출력됨
- Detect Value는 MDR에서 측정된 전방의 움직임에 대한 수치를 나타냄
  => 수치가 클수록 큰 움직임이거나 움직이는 물체가 가까이 있는 것을 의미함
- Detect Level은 High 신호를 주는 기준인 Detect Value의 Threshold 값을 나타냄
- MotionData는 Detect Value, 속도(Ver 2.0만 해당), GPIO_OUT1 상태를 통합한 용어임
- Detect Value가 설정된 Detect Level보다 높아지면 3번 Pin 출력이 High로 동작함
- Hold Time은 전방에 움직임이 있을 경우, High 신호의 지속시간(ms)을 나타냄
- Uart 통신을 통해 실시간으로 현재 측정되는 MotionData를 알 수 있음
- Uart 통신을 통해 Detect Level과 Hold Time은 가변 가능함

2. Arduino와 Radar 모듈의 Pin 연결

2.1 MDR_Mini_A Pin



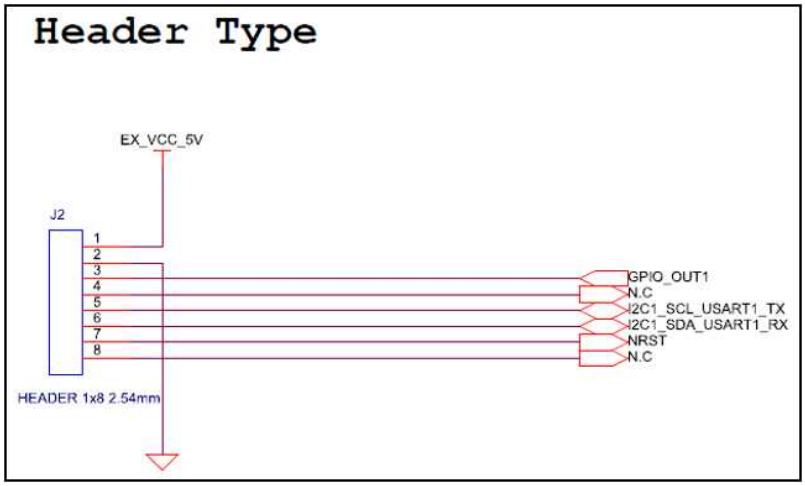


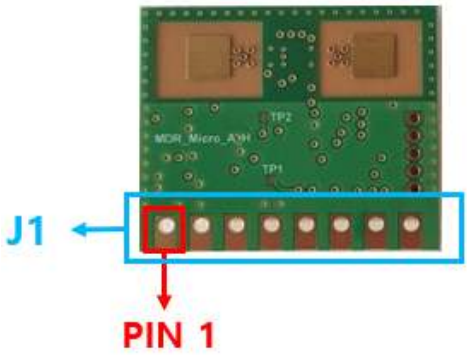
PIN	이름	설명
1	VCC_5V	전원 +
2	GND	전원 -
3	GPIO_OUT1	움직임 감지 상태 출력
4	N.C.	사용 안함
5	NRST	Reset(Active Low)
6	N.C.	사용 안함
7	N.C.	사용 안함
8	I2C1_SCL_UART1_TX	UART 데이터 송신(TTL 레벨 : 3.3 V)
9	I2C1_SDA_UART1_RX	UART 데이터 수신(TTL 레벨 : 3.3 V)

2.2 Arduino Uno와 MDR_Mini_A의 Pin 연결

PIN 정보	Arduino Uno	MDR_Mini_A
VCC_5V	5V	1번
GND	GND	2번
GPIO_OUT1	7번	3번
I2C1_SCL_UART1_TX	8번(Rx)	8번(Tx)
I2C1_SDA_UART1_RX	9번(Tx)	9번(Rx)

2.3 MDR_Micro_A Pin



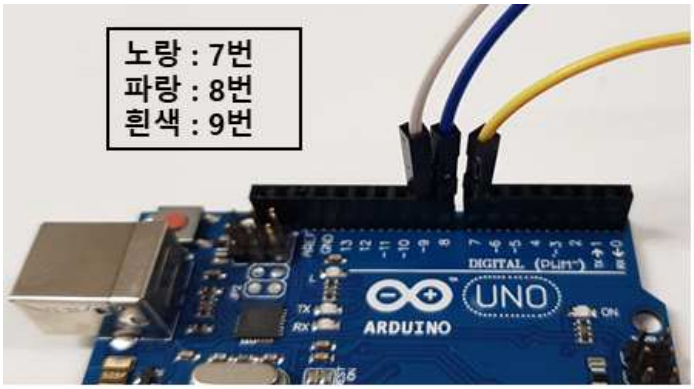


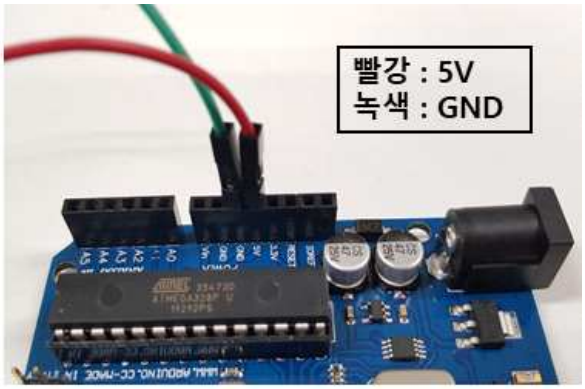
PIN	이름	설명
1	VCC_5V	전원 +
2	GND	전원 -
3	GPIO_OUT1	움직임 감지 상태 출력
4	N.C.	사용 안함
5	I2C1_SCL_UART1_TX	UART 데이터 송신(TTL 레벨 : 3.3 V)
6	I2C1_SDA_UART1_RX	UART 데이터 수신(TTL 레벨 : 3.3 V)
7	NRST	Reset(Active Low)
8	N.C.	사용 안함

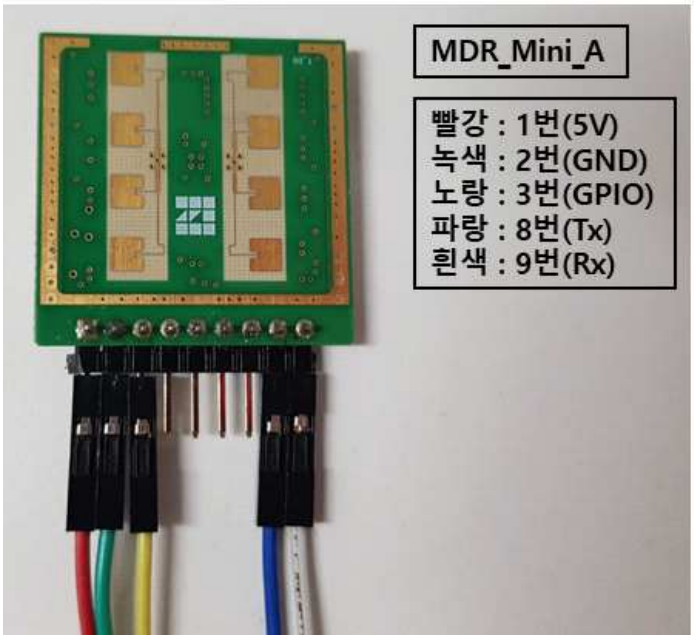
2.4 Arduino Uno와 MDR_Micro_A의 Pin 연결

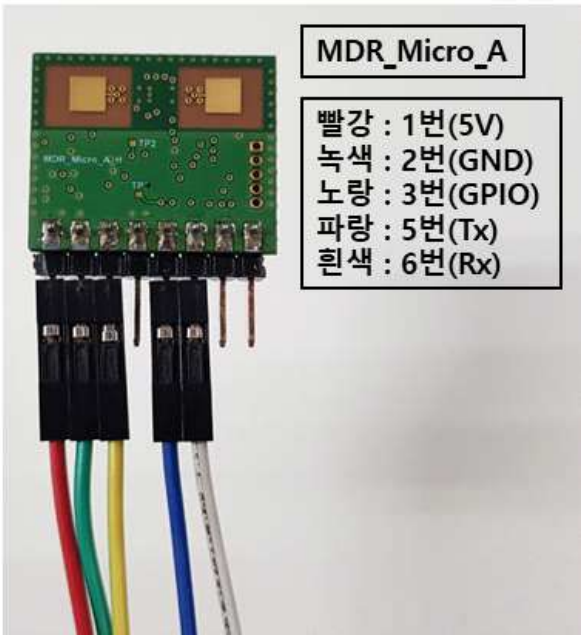
PIN 정보	Arduino Uno	MDR_Micro_A
VCC_5V	5V	1번
GND	GND	2번
GPIO_OUT1	7번	3번
I2C1_SCL_UART1_TX	8번(Rx)	5번(Tx)
I2C1_SDA_UART1_RX	9번(Tx)	6번(Rx)

2.5 MDR, Arduino Uno 연결 참조









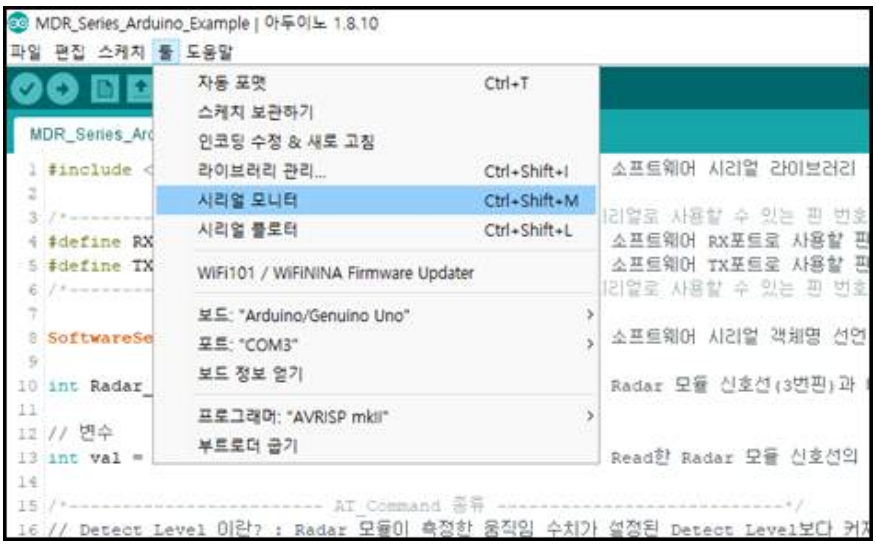
3. Arduino Example 수행 방법

3.1 소스코드 업로드

- Arduino와 MDR을 Pin 번호에 맞게 연결 후, Arduino Example 소스코드를 Arduino Uno에 업로드

3.2 시리얼 모니터 활성화

- 아래의 그림을 참고하여 시리얼 모니터 켜기

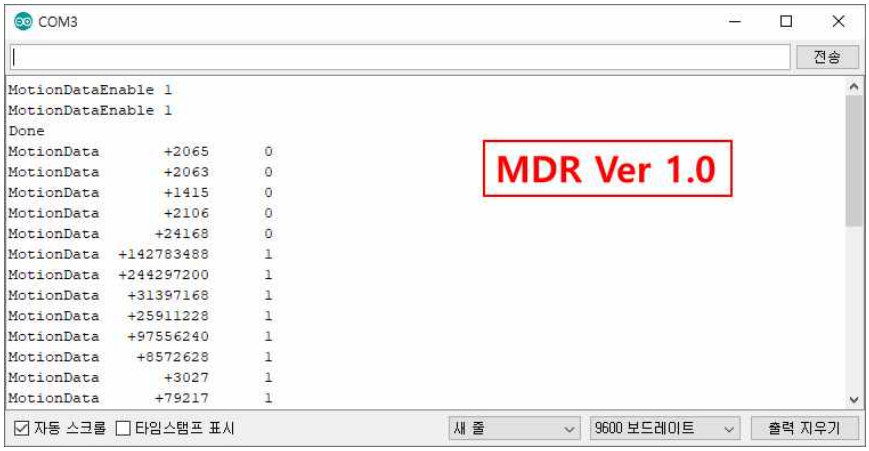


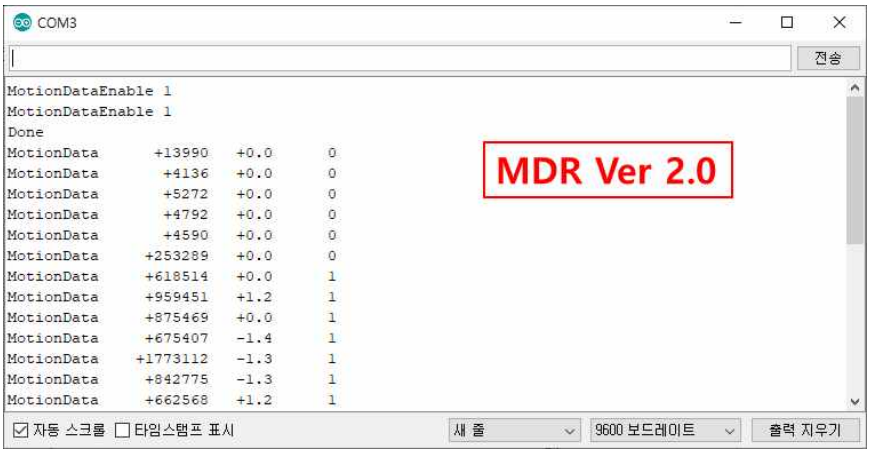
3.3 CLI Commnand 입력

- 시리얼 모니터에 아래의 그림을 참고하여 CLI Command를 입력하고 전송 버튼을 누름

3.4 CLI Command 입력에 대한 수신 데이터 확인

- CLI Command를 송신하면 그에 따른 수신 데이터를 확인함
- "MotionDataEnable 1"을 전송하면 MDR이 측정하고 있는 MotionData를 주기적으로 출력
- MDR Ver 2.0은 속도와 방향성 정보까지 출력됨
- 주요 CLI Command는 3.5를 참조
- 자세한 MotionData의 표현식은 3.6을 참조





3.5 주요 CLI Command

CLI Command	설명
MotionDataEnable 0	MDR이 측정하고 있는 MotionData를 주기적으로 출력
MotionDataEnable 1	MDR이 측정하고 있는 MotionData를 주기적으로 출력
DetectLevel	현재 설정되어 있는 DetectLevel 확인
DetectLevel 11000	DetectLevel [11000] 설정
HoldTime	현재 설정되어 있는 HoldTime 확인
HoldTime 2000	HoldTime [2000ms] 설정

3.6 MotionData 표현식

- "MotionDataEnable 1" 커맨드로 실시간 MotionData를 UART로 확인 가능하며, 움직임이 없을 시, 속도는 0으로 출력됨
- "MotionDataEnable 2" 커맨드는 움직임이 있을 시에만 MotionData를 UART로 출력함
- 아래의 글에서 'Space'(0x20)는 '␣'로 표현
- MotionData 표현식 : "MotionData␣+DetectValue␣±속도␣GPIO_State(0 or 1)"
- MotionData의 printf 포맷은 아래와 같음(자리수를 채우기 위한 공백은 '@'으로 표현)

"MotionData␣%+11d␣%+6.1f␣%6u₩n", (int32_t)DetectValue, (float)Speed, (uint32_t)GPIO_State

예시 1) MotionData␣+1234567890␣@@-5.5␣@@@@@1<LF>

예시 2) MotionData␣@@@@@+36516␣@@+5.5␣@@@@@1<LF>

예시 3) MotionData␣@@@@@@+8848␣@@+0.0␣@@@@@0<LF>

- DetectValue는 항상 양수로 출력되며, 부호를 포함하여 11자리수를 지킴
- 속도는 실수형이고, 앞에 '+' 또는 '-' 문자가 포함됨(속도 방향 : 가까워짐 - / 멀어짐 +)
- 속도는 부호와 '.'을 포함하여 6자리수를 지키며, 단위는 km/h 임
- GPIO_State는 0 또는 1로 표현되며 6자리수를 지킴