

UPD-M010UA PM1.0/PM2.5/PM10 Find Particulate Sensor

소프트웨어 프로그래밍 가이드 문서

(OPERATION MODE 4)

v1.01



Prerequisites

- **Hardware Prerequisites**

- UPD-M010UA Sensor
- Personal Computer (PC) or MCU as Master
- Used Interface between PC and UPD- M010UA : UART⁽¹⁾ Communication (RS232 Com)
[(1) UART : Universal Asynchronous Receiver, Transmitter]

- **Software Prerequisites**

- Compiler : ANSI C Compiler or Equivalent
- Programming Language : C language

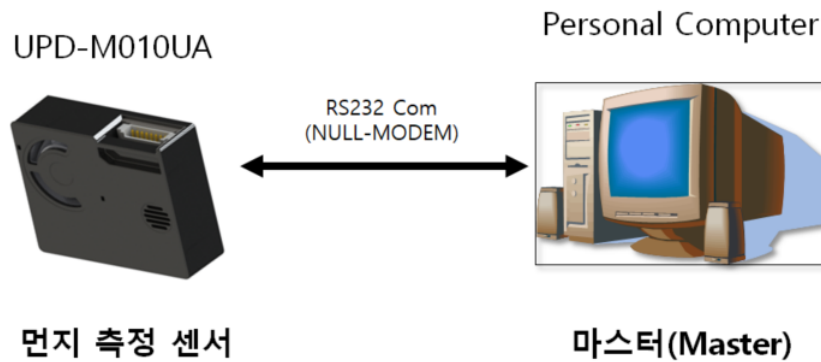
▶ **Revision History**

문서 버전	수정 사항	작성일	기타
v1.00	최초 작성 (MS KIM)	2019.12.11	
v1.01	PM10 읽기 예제 코드 추가	2020.05.18	

목 차

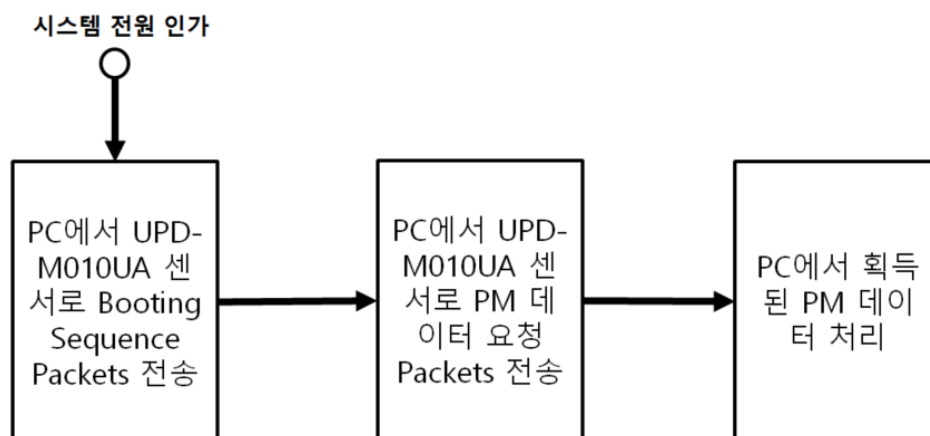
1. UPD-M010UA 제어 시스템 개요
2. UPD-M010UA 센서 동작 모드4 요청, 응답 메시지 구조
3. UPD-M010UA 센서 동작 모드4 초기화 및 PM 데이터 획득
 - 3.1 마스터(ex. PC)와 UPD-M010UA 통신 메시지 흐름도
 - 3.2 UPD-M010UA 센서 초기화
 - 3.3 UPD-M010UA 센서 PM1.0 / PM10 / PM2.5 데이터 획득
 - 3.4 UPD-M010UA 센서 PM10 읽기 예제 코드

1. UPD-M010UA 제어 시스템 개요



[그림 1] PC와 UPD-M010UA 센서 인터페이스 개념도

- PC(마스터)와 UPD- M010UA를 RS232 Null modem 방식(RxD, TxD, GND)으로 연결
- RS232 통신 설정 : 9600bps, Data 8 bit, Stop 1 bit, None Parity
- 기본 동작 방법



2. UPD-M010UA 센서 요청, 응답 메시지 구조

UPD-M010UA 센서 동작 모드4는 마스터(Master)의 요청에 응답하는 슬레이브(Slave) 장치로써 동작된다.

2.1 동작 모드4의 요청 메시지 구조

- UPD-M010UA 센서로부터 데이터를 획득하고자 하는 주체(i.e. PC 또는 Atemga128 등)가 UPD-M010UA로 보내는 패킷(Packets) 구조를 말한다.
- 요청 메시지는 마스터가 UPD-M010UA 센서로부터 데이터를 획득하는 모드(읽기 모드)와 센서에 데이터를 쓰는 모드(쓰기 모드)로 구분된다.
- 요청 메시지는 읽기 모드일 때는 1 바이트 크기를 가지며, 쓰기 모드일 경우는 2 바이트 크기를 갖는다.
- 읽기 모드와 쓰기 모드의 구분은 센서로 보내지는 처음 1바이트 요청 메시지의 최상위 비트 설정에 따라 구분된다. [그림 2]는 그 형태를 나타낸다.

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
동작모드							
UPD-M010UA 레지스터 어드레스							
• BIT7 : 동작 모드, 0(쓰기 모드), 1(읽기 모드) • BIT6 ~ BIT0 : UPD-M010UA 레지스터 주소							

[그림 2] UPD-M010UA 요청 메시지 구조

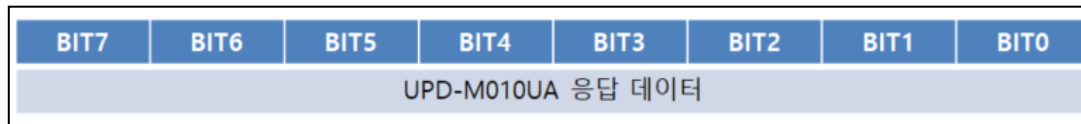
- 읽기 모드일 경우는 [그림 2]와 같이 1 바이트만 센서로 전송.
- 만약 요청 메시지가 쓰기 모드일 경우는 [그림 2]의 1 바이트의 패킷을 마스터가 센서로 전송한 후에 그 다음 패킷으로 [그림 3]과 같이 1 바이트 데이터를 전송한다.

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
UPD-M010UA 레지스터에 쓰고자 하는 데이터							

[그림 3] UPD- M010UA 요청 메시지 쓰기 모드 데이터

2.2 응답 메시지 구조

- UPD-M010UA 센서가 마스터에 데이터를 전송할 때 패킷을 말한다.
- 데이터는 1 바이트 크기로 마스터에 전송된다. [그림 4]는 그 형태를 나타낸다.

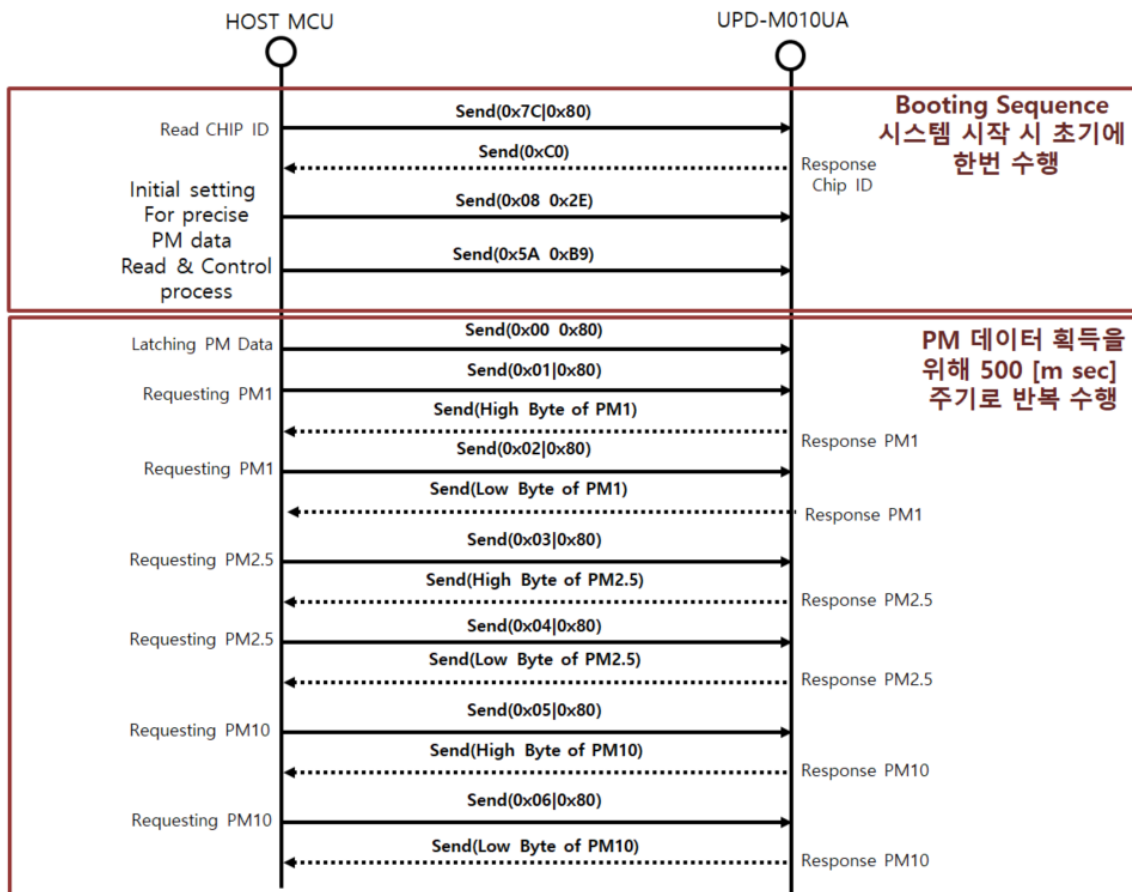


[그림 4] UPD-M010UA 응답 메시지 구조

3. UPD-M010UA 초기화 및 PM 데이터 획득

3.1 마스터와 UPD-M010UA 센서간 데이터 통신 전체 흐름도

다음은 센서가 동작하기 위한 마스터와 센서 사이의 전체 통신 데이터 흐름을 나타낸다.



① Booting Sequence

시스템 전원 인가 후 Master에 의해 한번만 수행되는 절차

② PM 데이터 획득

먼저 농도를 실시간으로 얻기 위해서 500[msec] 마다 수행되어야 하는 절차

3.2 UPD-M010UA 센서 초기화 (Booting Sequence)

해당 센서로부터 올바른 데이터를 얻기 위해서는 Booting Sequence 라는 절차를 전체 시스템 부팅 초기에 반드시 한번 수행해야 한다. 아래 [표 1]은 해당 부팅 시퀀스를 나타낸다.

[표 1] UPD-M010UA 부팅 시퀀스

Seq.	RD/WR	Address	Data	Account	Example
①	RD	0x7C	0xC0	Reading Chip ID if different data is read, please do seq.① again.	0xC0(RD)
②	WR	0x08	0x2E	Initial setting for precise PM data read process	
③	WR	0x5A	0xB9		

TYPE	RO							
ADDR[7:0]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
0x7c	CHIP_ID[7:0]							

CHIP_ID CHIP_ID 값은 0xC0

TYPE	RW							
ADDR[7:0]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
0x08	USER_KEY[7:0]							

USER_KEY USER 컨트롤 영역 접근제한을 위한 key 값
key값이 일치하지 않는 경우 read/write 불가
<key value : 0x2e>

TYPE	RW							
ADDR[7:0]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
0x5a	HIDDEN_KEY[7:0]							

HIDDEN_KEY HIDDEN KEY 의 값은 < 0xB9 >

3.3 UPD-M010UA 센서 PM1.0 / PM2.5 / PM10 데이터 획득

해당 센서는 500 [msec] 마다 새로운 PM1 / PM2.5 / PM10 데이터를 센서 내부 레지스터에 업데이트한다.

마스터는 실시간 데이터 수집을 위해 500[msec] 주기로 데이터를 획득하면 된다. 단, 마스터에서 데이터 수집 전에 반드시 데이터 Latching 명령을 센서로 먼저 전송한 후, PM1.0 / PM2.5 / PM10 데이터를 수집해야 올바른 결과를 얻을 수 있다.

아래는 관계된 레지스터의 설명이다.

TYPE	RW							
ADDR[7:0]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
0x00	SAMP_N_HOLD							

SAMP_N_HOLD data capture command, dust data 또는 adc 데이터 등 8-bit 넘는 register를 읽어낼 때 read 동작 이전에 capture 수행 후 read 수행
해당 register : adc_rdata_dust/ext/bgr/vref, pm10/25_out, lpf10/25_in
<1=active high>

TYPE	RO							
ADDR[7:0]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
0x01							PM1_OUT[9:8]	
0x02	PM1_OUT[7:0]							

PM1_OUT PM1 먼지 농도
<0~1000>

TYPE	RO							
ADDR[7:0]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
0x03							PM2_5_OUT[9:8]	
0x04	PM2_5_OUT[7:0]							

PM2_5_OUT PM2.5 먼지 농도
<0~1000>

TYPE	RO							
ADDR[7:0]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
0x05							PM10_OUT[9:8]	
0x06	PM10_OUT[7:0]							

PM10_OUT PM10 먼지 농도
<0~1000>

3.4 UPD-M010UA 센서 PM10 읽기 예제 코드

아래는 (3.1) 내용으로 기반으로 작성된 BootingSequence() 함수와 ReadPM10() 예제 코드이다.

```
int BootingSeq(unsigned char IsOpenSerial)
{
    int ret = false;
    unsigned char txBuff[2];
    unsigned char chipId;
    unsigned char retrycount = 0;
    unsigned char retrymax = 3;

    if (IsOpenSerial == true)
    {
        //1.Check Chip ID (만약 Chip ID Read 실패시 3번정도 반복 체크)
CHIPID:
        txBuff[0] = (0x80 | 0x7C);
        Write(txBuff, 1);          //Write Data to Serial Port

        chipId = Read();           //Read Data From Serial Port

        if (chipId != 0xC0)
        {
            if (++retrycount < retrymax)
                goto CHIPID;
            else
                return ret;
        }

        //2.Write Keys
        //2.1 Key1
        txBuff[0] = 0x08;
        txBuff[1] = 0x2e;
        Write(txBuff, 2);

        //2.1 Key1
        txBuff[0] = 0x5a;
        txBuff[1] = 0xb9;
        Write(txBuff, 2);

        ret = true;
    }

    return ret;
}

unsigned short ReadPM10()
{
    unsigned short pm10 = 0;
    unsigned char rxVal = 0;
    unsigned char txBuff[2];
}
```

```
//1.Latch Register Update
txBuff[0] = 0x00;
txBuff[1] = 0x80;
Write(txBuff, 2);

//2.Read PM10
txBuff[0] = (0x80 | 0x05); //Read PM10_H
Write(txBuff, 1);
rxVal = Read();
pm10 = (rxVal & 0x03) * 256;

txBuff[0] = (0x80 | 0x06); //Read PM10_L
Write(txBuff, 1);
rxVal = Read();
pm10 += rxVal;

return pm10;
}

int main(int argc, char **argv)
{
    int ret = false;
    unsigned short pm10;

    OpenSerialPort();

    ret = BootingSeq(true);

    while (1)
    {
        if (ret == true)
        {
            Delayms(500);    //500 [msec]

            pm10 = ReadPM10();
        }
    }

    return true;
}
```