

```

#include <LiquidCrystal_I2C.h>
// Initialize resource for LCD Display
LiquidCrystal_I2C lcd(0x27, 16, 2); // address: 0x27, cols:16, rows:2
unsigned long lastSwitchDataTime = 0;
unsigned long lastUpdateTime = 0;
bool showPM10 = false;
// Initialize resource for PM Sensor
#define DATA_LENGTH 31
#define PM1D0_POS 1
#define PM2D5_POS 2
#define PM10_POS 3
#define CHECKSUM_POS 14
int pm1d0Value = 0;
int pm2d5Value = 0;
int pm10Value = 0;
char pm2d5Str[16] = {0};
char pm10Str[16] = {0};
char pm1d0Str[16] = {0};
unsigned char pmBuffer[DATA_LENGTH];

int parseDataAtPosition(unsigned char* buf, int pos) {
    return (buf[1 + 2 * pos] << 8) + buf[2 + 2 * pos];
}

bool validate_checksum(unsigned char* buf, char len) {
    // calculate checksum
    int checksum = 0x42;
    for (int i = 0; i < (len - 2); i++) {
        checksum += buf[i];
    }

    // validate checksum
    if (checksum == parseDataAtPosition(buf, CHECKSUM_POS)) {
        return true;
    }
    return false;
}

void setup() {
    // put your setup code here, to run once:
    // Initialize LCD Display
    lcd.init();
    lcd.backlight();
    lcd.clear();

    // Initial serial for PM Sensor
    Serial.begin(9600);
    Serial.setTimeout(1500);
}

void loop() {
    // put your main code here, to run repeatedly:
    // Parse data sent from the PM sensor
    if (Serial.find(0x42)) {
        Serial.readBytes(pmBuffer, DATA_LENGTH);
        if (pmBuffer[0] == 0x4d && validate_checksum(pmBuffer, DATA_LENGTH)) {
            pm1d0Value = parseDataAtPosition(pmBuffer, PM1D0_POS);
            pm2d5Value = parseDataAtPosition(pmBuffer, PM2D5_POS);
            pm10Value = parseDataAtPosition(pmBuffer, PM10_POS);
        }
    }
    //Switch display type
    if (millis() - lastSwitchDataTime > 5000) {
        lastSwitchDataTime = millis();
        showPM10 = !showPM10;
    }
    // display data
    if (millis() - lastUpdateTime > 1000) {
        lastUpdateTime = millis();
        memset(pm1d0Str, 0, 16);
        memset(pm2d5Str, 0, 16);
        memset(pm10Str, 0, 16);
        sprintf(pm1d0Str, "PM1.0:%dug/m3", pm1d0Value);
        sprintf(pm2d5Str, "PM2.5:%dug/m3", pm2d5Value);
        sprintf(pm10Str, "PM10:%dug/m3", pm10Value);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(pm2d5Str);
        lcd.setCursor(0, 1);
        lcd.print(showPM10 ? pm10Str : pm1d0Str);
    }
}

```