## Draft content

Content

[Draft content 1](#_Toc26178988)

[Functionality: 1](#_Toc26178989)

[Further Functionality/Design: 1](#_Toc26178990)

[Software level: 1](#_Toc26178991)

[BOM: 2](#_Toc26178992)

[Testing plan: 2](#_Toc26178993)

[Pin Assignment 2](#_Toc26178994)

[Module technical details: 2](#_Toc26178995)

[Bluetooth: 2](#_Toc26178996)

Documentation

1. Technical manual
2. User manual
   1. How to drop bin file
   2. Hardware setup
   3. COM port setup
   4. command

## Functionality:

1. User interface. (LCD, button/Keypad)
2. Real-Time PM 2.5 detecting
3. BLE remote control (Device and PC communication.)
   1. Show options
      1. Current PM 2.5 showing with pollution level, time, date
      2. History PM 2.5 showing with pollution level, time, date
      3. Logging current data
      4. Remove data
      5. Time manual adjust
4. Running indicator (LED blinking. Quick blink when reset)
5. Regular Data logging (TF card, File I/O)
6. RTC
7. Pollution Indicator (LED)

## Further Functionality/Design:

1. Air purifier
2. PCB design
3. Low Power Mode (Option to Sleep or Wake up)
4. Time auto adjusts
5. Touch screen instead of LCD
6. Pollution level Warning (Connect to Raspberry Pi then send email or Tweeter)

## Software level:

1. FreeRTOS
2. Separate manual log and auto log into different file

## BOM:

1. LCD
2. PM2.5 detector
3. LED
4. RTC
5. TF card adapter
6. Bluetooth module

Module to test:

1. LCD
2. PM 2.5 detector data reading
3. LED blinking (GPIO)
4. RTC reading
5. TF card reading
6. Bluetooth reading

## Testing plan:

1. Separation test.
   1. Test each module individually.

## Pin Assignment

## Module technical details:

### PM2.5

Protocol: Serial

### LCD

LCD pin mapping:

|  |  |  |
| --- | --- | --- |
| LCD pin | STM32 pin | Function |
| D4 | PB5 |  |
| D5 | PB4 |  |
| D6 | PB10 |  |
| D7 | PA8 |  |
| D8 | PA9 | RS |
| D9 | PC7 | Enable |
| D10 | PB6 | Back Light |
| A0 | PA0 | Buttion select |

LCD API:

**void** **Lcd\_string**(Lcd\_HandleTypeDef \* lcd, **char** \* string)

**void** **Lcd\_cursor**(Lcd\_HandleTypeDef \* lcd, uint8\_t row, uint8\_t col)

**void** **Lcd\_clear**(Lcd\_HandleTypeDef \* lcd)

**void** **Lcd\_int**(Lcd\_HandleTypeDef \* lcd, **int** number)

### TF card

Protocol: SPI

### RTC

Protocol: I2C

### Bluetooth:

Module: HC-05

Protocol: serial

Problem:

1. LCD take too much IO pin.

## Debugging note:

### Retargeting:

Can not use ‘’ in printf, it will cause hardfault handler. Instead, use “”.