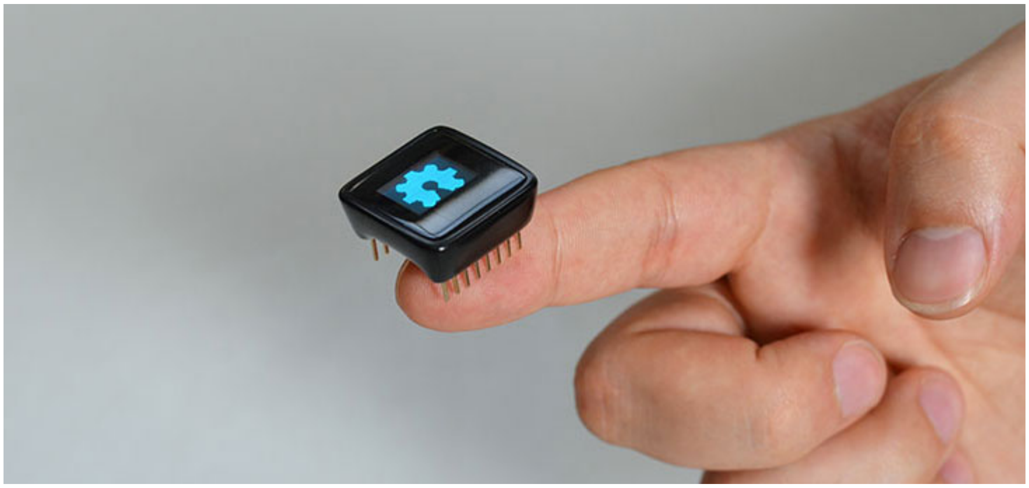
Intro

General Overview of MicroView by JP Liew



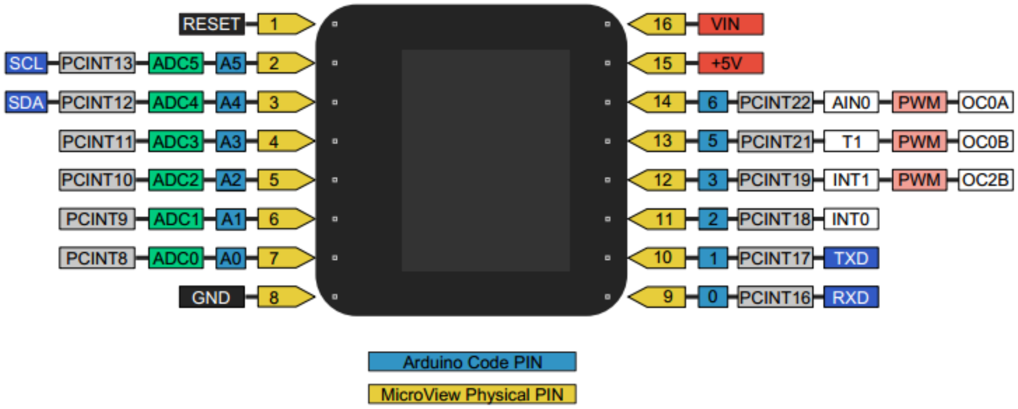
The MicroView is the first chip-sized Arduino compatible that lets you see what your Arduino is thinking by using a built-in OLED display. In the heart of MicroView there is an ATMEL's ATmega328P, 5V & 3.3V LDO and a 64x48 pixel OLED display, together with other passive components that allow the MicroView to operate without any external components other than a power supply.

> *MicroView is 100% code compatible with Arduino Uno (ATmega328P version), meaning the code that runs on an Arduino Uno will also be able to run on the MicroView if the IO pins used in the code are externally exposed on the MicroView.*

Hardware Specifications

- Display : 64x48 OLED Display
- Microcontroller : ATmega328P
- Operating Voltage : 5V
- Input Voltage : 3.3VDC - 16VDC
- Digital I/O Pins : 12 (of which 3 provide PWM output)
- Analog Input Pins : 6
- Flash Memory : 32 KB
- SRAM : 2 KB
- EEPROM : 1 Kilobyte
- Clock Speed : 16 Mhz
- No other components required

Pin Configuration



For more details on pins of MicroView, please refer to the ATmega328P Datasheet.

OLED Memory Map

The SSD1306 is the controller built into the MicroView's OLED display. It has flexible yet complex segment and common drivers. One requires vast knowledge on memory addressing in order to use the SSD1306 controller.

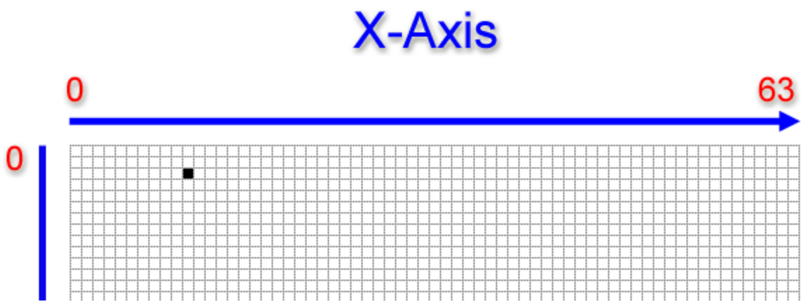
MicroView's library was written to hide away the complexities of the SSD1306 controller, so that users can issue simple commands to control the display. Although the SSD1306 has a built-in RAM (memory) for the screen, when connected using the SPI method, the ATmega328P is not able to read the RAM (memory) of the SSD1306. Therefore the software will not be able to manipulate the screen buffer to perform mathematical operations.

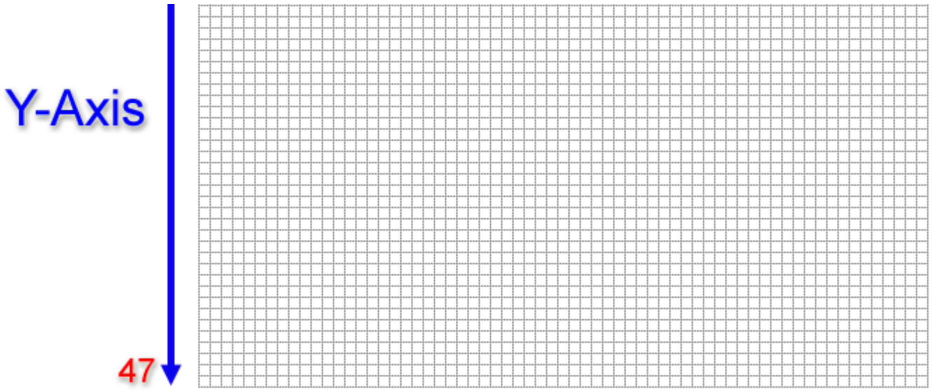
MicroView's library overcomes this by allocating 384 bytes ((64 x 48)/8 bits) of memory from ATmega328P as buffer. The library can now manipulate the screen buffer and then perform a bulk transfer from the ATmega328P's memory to the internal memory of the SSD1306 controller.

The 384 bytes of screen buffer are declared in MicroView's library as

```
static uint8_t screenmemory [] = {.total 384 bytes of data..};
```

and are arranged in a linear form representing the following 64 x 48 pixels coordinate system.





Based on the above illustration, for example, if a user wish to plot a pixel at the position of the black dot, where X=10 and Y=2, the would user issue the following command:

```
uView.pixel(10,2);
```

This command will then calculate the exact location of the screen buffer and set a BIT in the corresponding BYTE to the X,Y position.

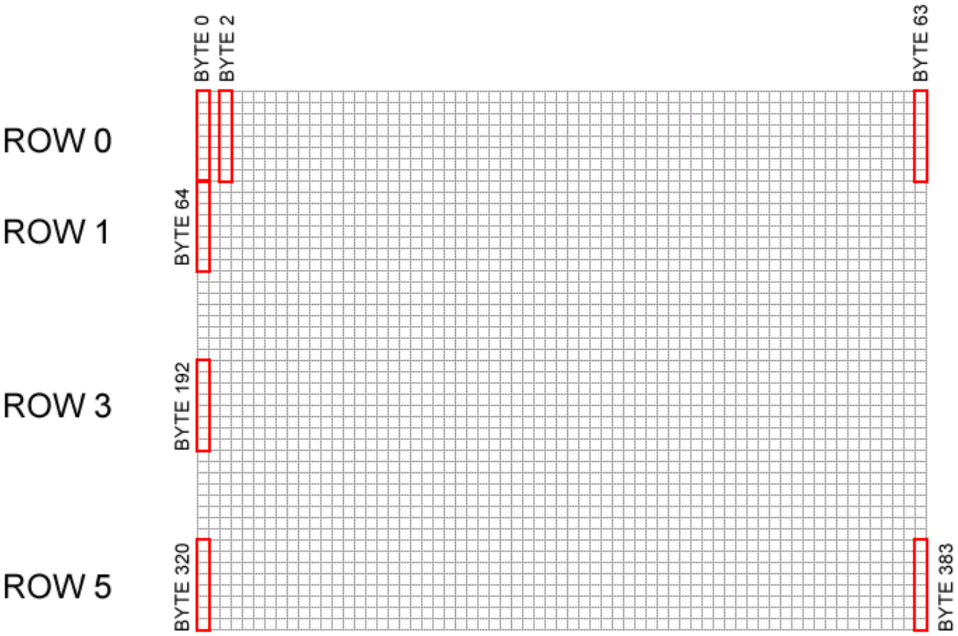
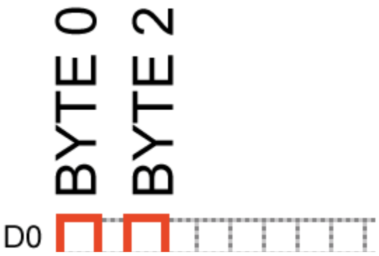


Diagram showing how a linear screen buffer in the ATmega328P aligns with the OLED pixels.



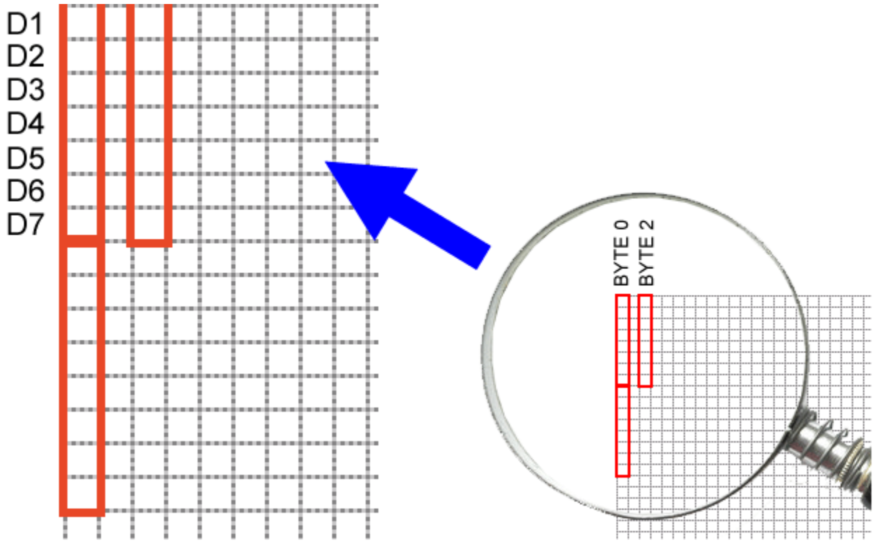


Diagram showing the BITs in a BYTE of the screen buffer corresponding to the OLED's X,Y position.

Based on the above illustration, a pixel turned on at X=2 and Y=3 means BYTE 2 of the screen buffer has data of 0x08 (hex).

Two pixels at X=2,Y=3 and X=2,Y=2 turned on means BYTE 2 of the screen buffer has data of 0x0c (hex).

To draw a straight line of 5 pixels starting from 10,2 to 10,6 , the following C code show a pixel by pixel way on how to accomplish this:

```
uView.pixel(10,2);
uView.pixel(10,3);
uView.pixel(10,4);
uView.pixel(10,5);
uView.pixel(10,6);
```

the MicroView library allows you to draw lines by specifying the the start and end coordinates. The above line could be drawn with this simple one line command:

```
uView.line(10,2,10,6);
```

In order for the library to perform extremely fast mathematical operations on the screen buffer (more than 100 frames per second), calls to the drawing functions within the MicroView library does not immediately transfer the contents of screen buffer to the SSD1306 controller. A `display()` command is required to instruct the library to perform the bulk transfer from the screen buffer to the SSD1306 controller:

```
uView.display();
```

This function takes the whole screen buffer in the ATmega328P and transfers it (via SPI bus, programmed at 8Mhz) to the internal memory of the SSD1306. As soon as the memory is being transferred, the pixels corresponding to the screen buffer will show up on the OLED display.

Published 04 June 2014

microview introduction overview memory map
ssd1306 screen buffer