

# SmarShow User Manual

Yunhui Fu

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Hardware</b>	<b>1</b>
2.1	Other Solutions . . . . .	1
<b>3</b>	<b>Software</b>	<b>1</b>
3.1	AVR . . . . .	3
3.2	ESP . . . . .	4

## 1 Introduction

SmartShow implemented a digital interface to the vintage Fluke 8050a DMM, fetches the display data via UART port, and optional connected by WiFi via an ESP8266 module.

## 2 Hardware

The Arduino UNO or Micro are used as the hardware interface to the MCU of the Fluke 8050a DMM, an ESP8266 module is used as WiFi UART bridge.

The Arduino UNO is ATMega328p based module and have no enough pin resource to handle both the TTL signal and an SPI LCD module, so it's only used as headless UART port for outside. While it still have the ability to drive an I2C OLED display, and an I2C slot is reserved.

The Arduino Micro is ATMega32U4 based module, it can drive a TFT SPI display module. So it's used to be a replacement of original Fluke 8050a LCD display.

ESP8266 would not start if the TX port is connected directly or via bi-direction level shifting to the RX of Arduino UNO, the solution is to use an LM393 level shifting.

### 2.1 Other Solutions

## 3 Software

The software are for both the AVR and ESP. The ESP can also upgrade the AVR's firmware remotely.

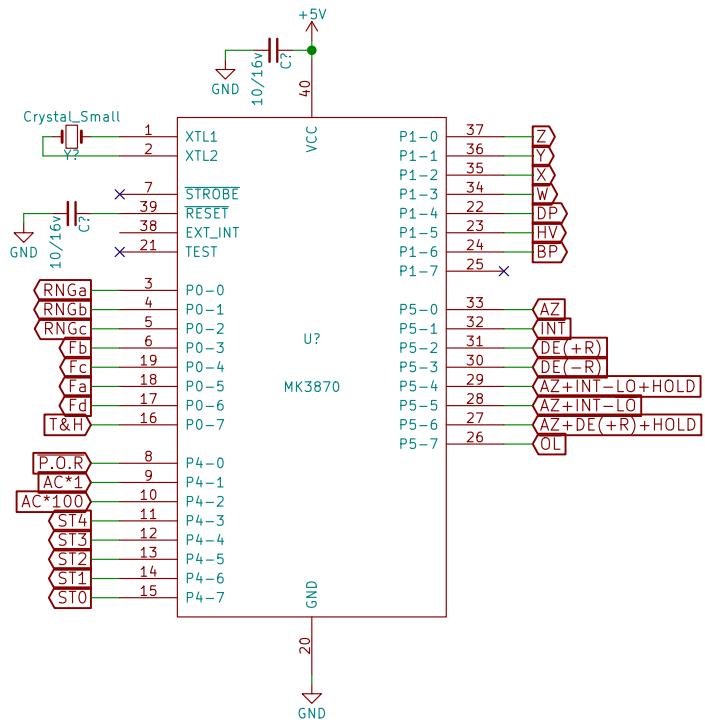


Figure 1: Fluke 8050a DMM U17.

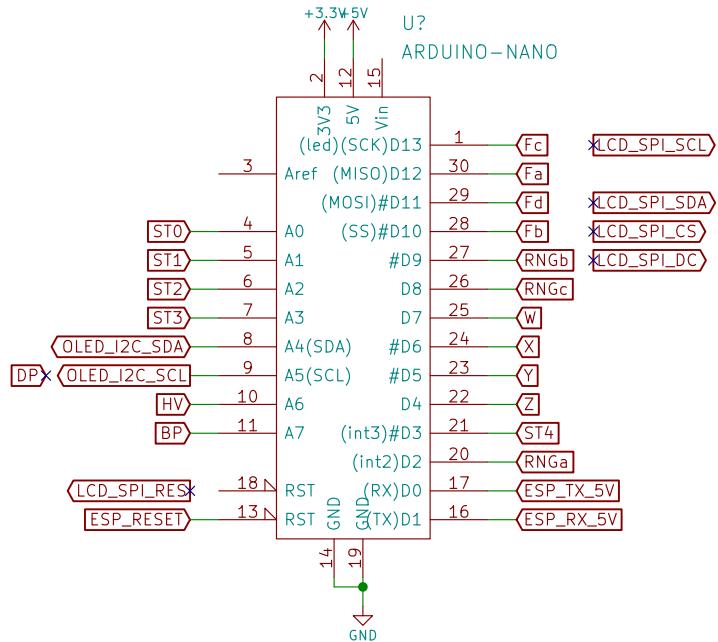


Figure 2: ATMega328p as headless interface.

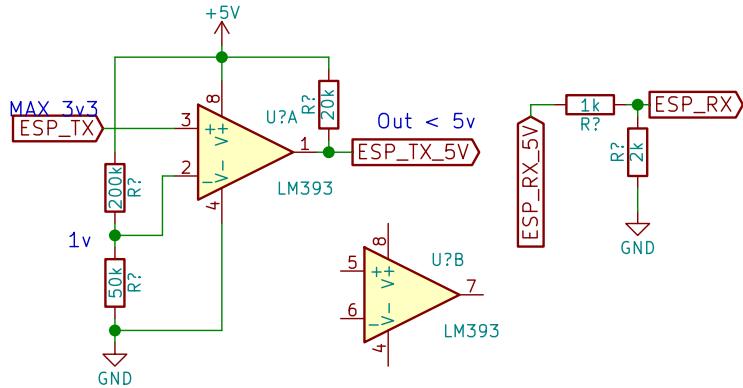


Figure 3: Level shifting works between Arduino UNO and ESP8266.

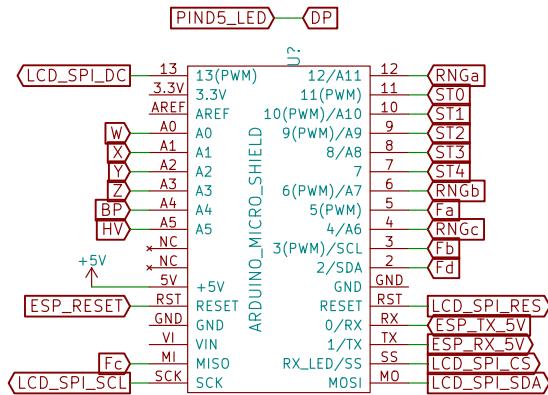


Figure 4: ATMega32U4 as display interface.

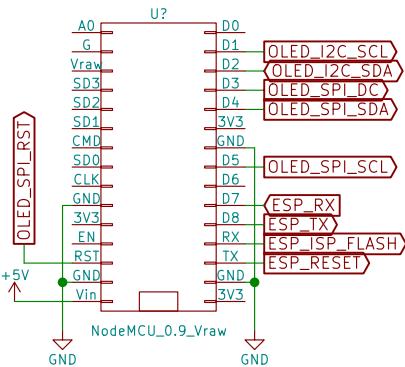


Figure 5: ESP8266 module as WiFi UART bridge. (UART Swap mode)

### 3.1 Configuration

To config the hardware used in your project, you may find the file conf-func.h in the directory software/smartsheet-fluke8050a/.

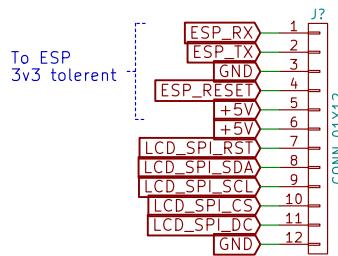


Figure 6: The connector for ESP8266.

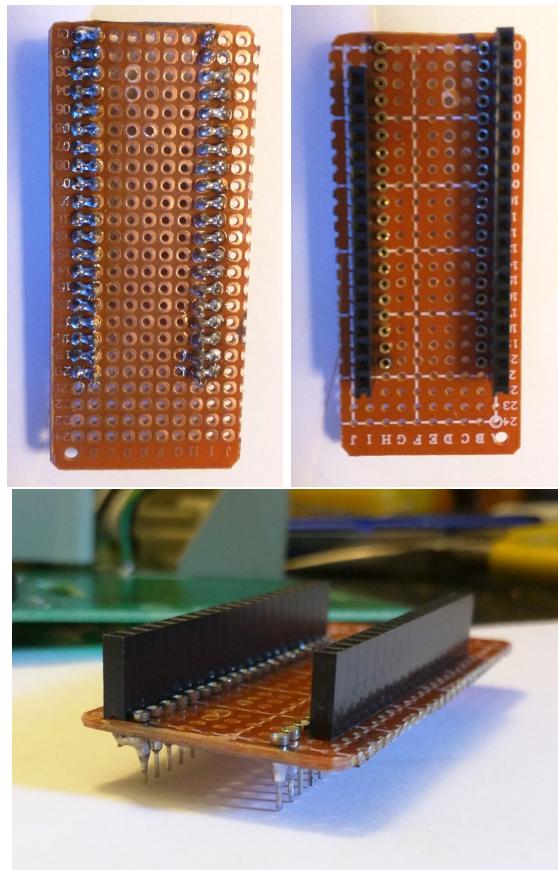


Figure 7: The Fluke8050a U17 piggybacked board.

### 3.2 AVR

The AVR firmware monitor the pins ST0, ST1, ST2, ST3, ST4, ST5 etc, and decrypt the strobe signal to value displayed in the LCD.

The strobe signal decrypt part of the source code of firmware are from the project [Fluke 8050a graphic display](#).

The full source project can be gotten from the repository <https://github.com/yhfudev/fluke8050a-screen.git>

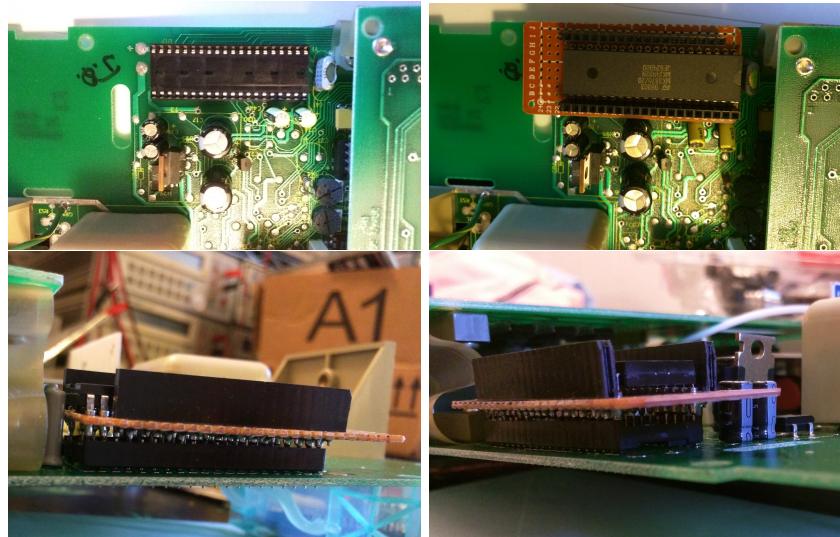


Figure 8: The Fluke8050a U17 piggybacked board installed.

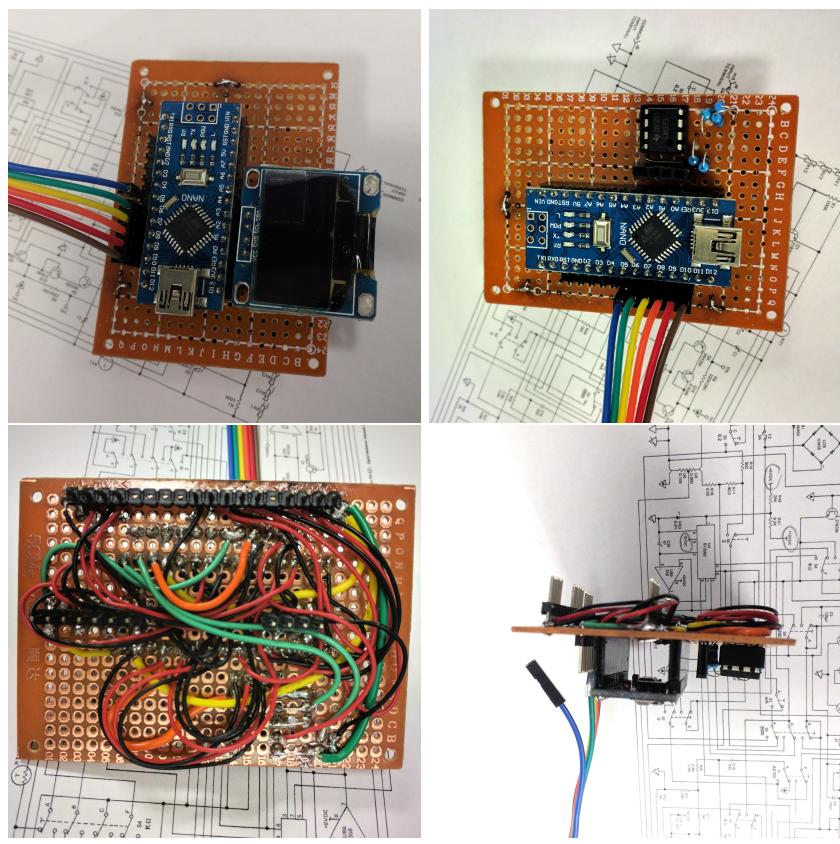


Figure 9: The SmartShow main board.

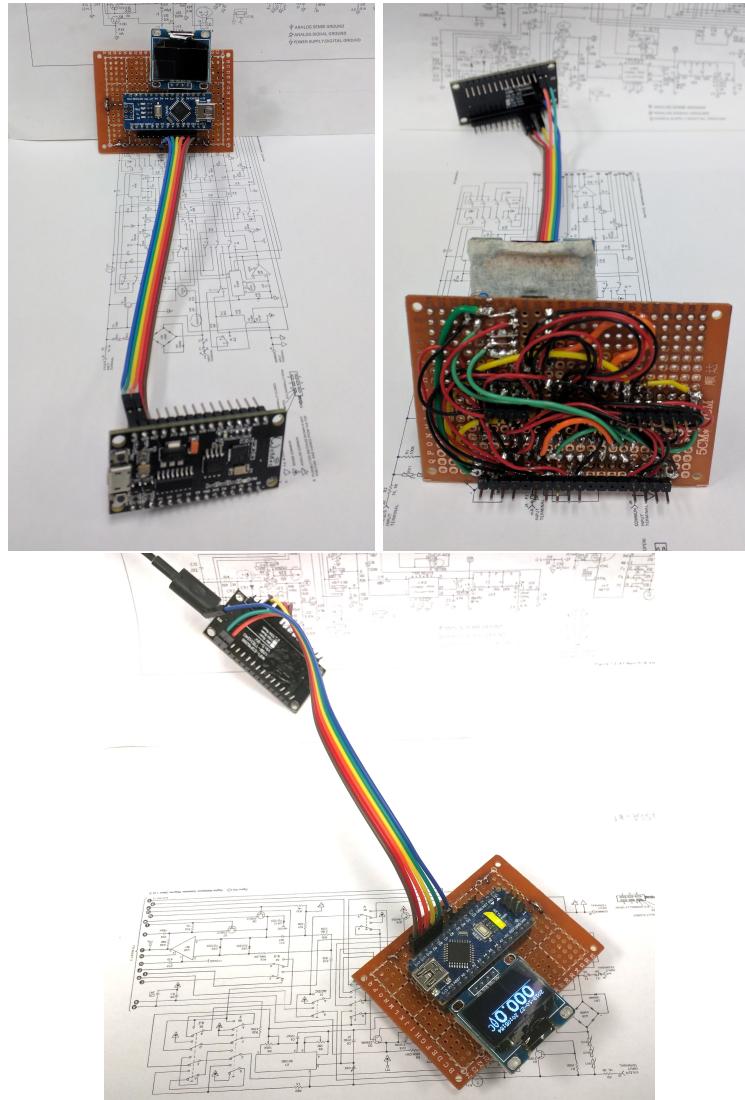


Figure 10: The SmartShow main board connected to ESP module.

### 3.3 ESP

For ESP, we use the project [esp-link](#). The pins are configured showed in the Figure 13.

The data port can be accessed by either from the web UI(Figure 14) or TCP port 23.

```
IP_ARDUINO=192.168.1.123
telnet ${IP_ARDUINO} 23
```

To upgrade the AVR's firmware, for example, flash ATMega328p over Wifi:

```
DN_ARDUINO="/opt/applications/arduino-1.8.5"
IP_ARDUINO=192.168.1.123
```

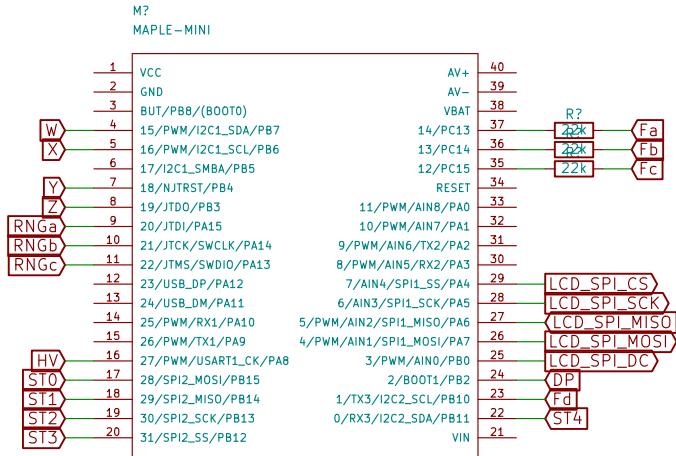


Figure 11: STM32(Maple-Mini).

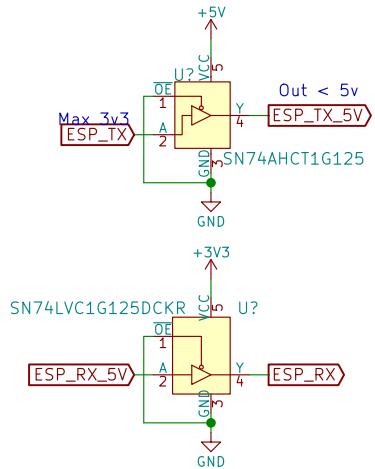


Figure 12: Alternative level shifting.

```
DN_HEX="/tmp/arduino_build_623246/fluke8050a-display.ino.hex"  
${DN_ARDUINO}/hardware/tools/avr/bin/avrdude \  
-C${DN_ARDUINO}/hardware/tools/avr/etc/avrdude.conf \  
-v -patmega328p -carduino \  
-P net:${IP_ARDUINO}:23 -b57600 -D -Uflash:w:${DN_HEX}:i
```

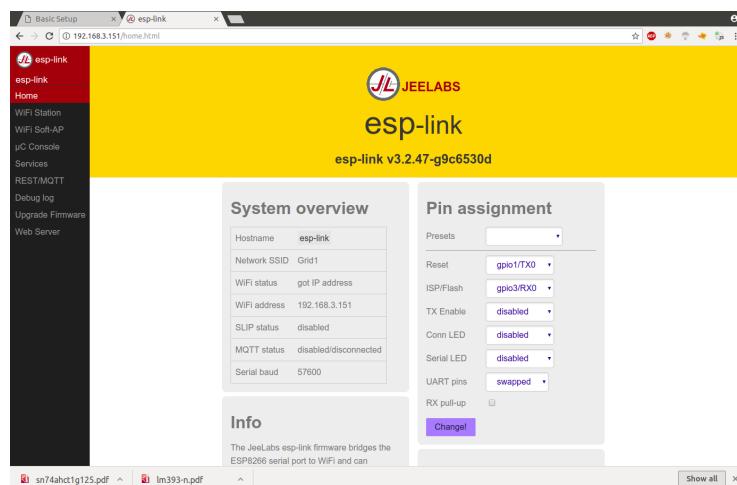


Figure 13: Configure Pins.

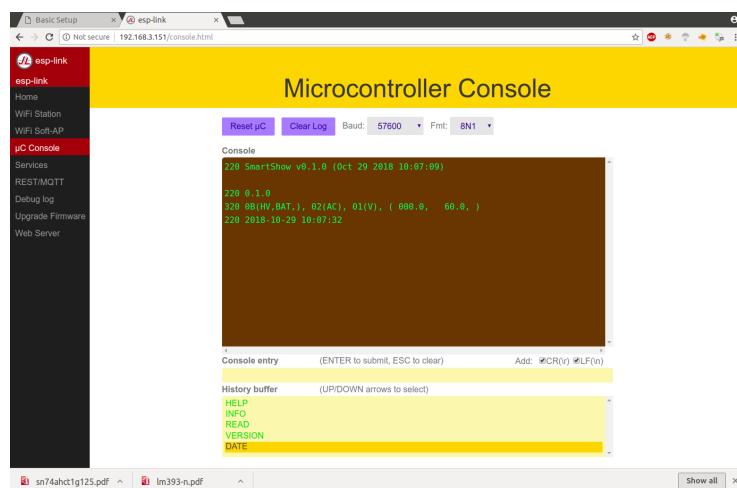


Figure 14: UART via WiFi.