

ESP32 LCD Controller Manual

V 0.2 – 29 Jan 2021

This manual describes the features of the LCD Controller module as a stand-alone entity. All information about lab instruments that use this controller is contained in their individual manuals.

This manual is intended as a reference, construction and commissioning details are available in the relevant Silicon Chip article.

Contents

ESP32 LCD Controller Manual	1
Features	1
Overview	2
ESP32 Pinout and Expansion Connector	3
Expansion Header	3
REM_ON_OFF Header	3
REM_ENC Header	4
ESP32 Pin Assignments.....	4
Software and Libraries	5
Communications.....	5
Display	6
Schematic	7

Features

Processor:	ESP32. 32-bit, 240MHz, dual core.
Memory:	4MB Flash, 520k RAM, full-size and micro SD CARD sockets
Display:	2.8" or 3.5" colour LCD touch screen. Jumper-selectable touch screen interrupt.
User interface:	LCD touch panel Rotary encoder and up/down buttons Separate start/stop buttons with LED indicator
Flexible configuration:	Switches and encoder are detachable, allowing flexibility of controls and panel layout. Several functions
IO and Expansion:	20 pin connector carrying power, Serial RXD/TXD, I2C with an optional second channel SPI – shared with the LCD display and SD-CARD 2 x 8-bit DAC (200kHz) 2 x 12-bit ADC (27kHz under non-DMA operation) 3 uncommitted GPIO pins. 17 total GPIO pins. All specialist pins other than SPI, can be repurposed as GPIO and PWM. Additional GPIO pins are available if onboard switches and encoder are not required, via the REM_ON_OFF and REM_SCRUB headers.
Communication:	WiFi (802.11 b/g/n 2.4GHz). 150 Mbps maximum throughput. QoS (802.11 e) for multimedia applications. Station or soft-AP modes, TCP, UDP, mDNS over IPV4 and IPV6.

Webserver and web client functions.
Bluetooth and BLE
TTL and USB serial.
Extensive examples provided for all communication modes.

Powering: USB
5-12V via barrel jack and regulator
5V via expansion header
3.3V up to 50mA available, via the ESP32's onboard regulator.

Updating Arduino (and other IDE) native programming
Off-the-air (OTA) binary uploads.

SCPI over WiFi (802.11 b/g/n/e/i) for PC control or master control of multiple modules.

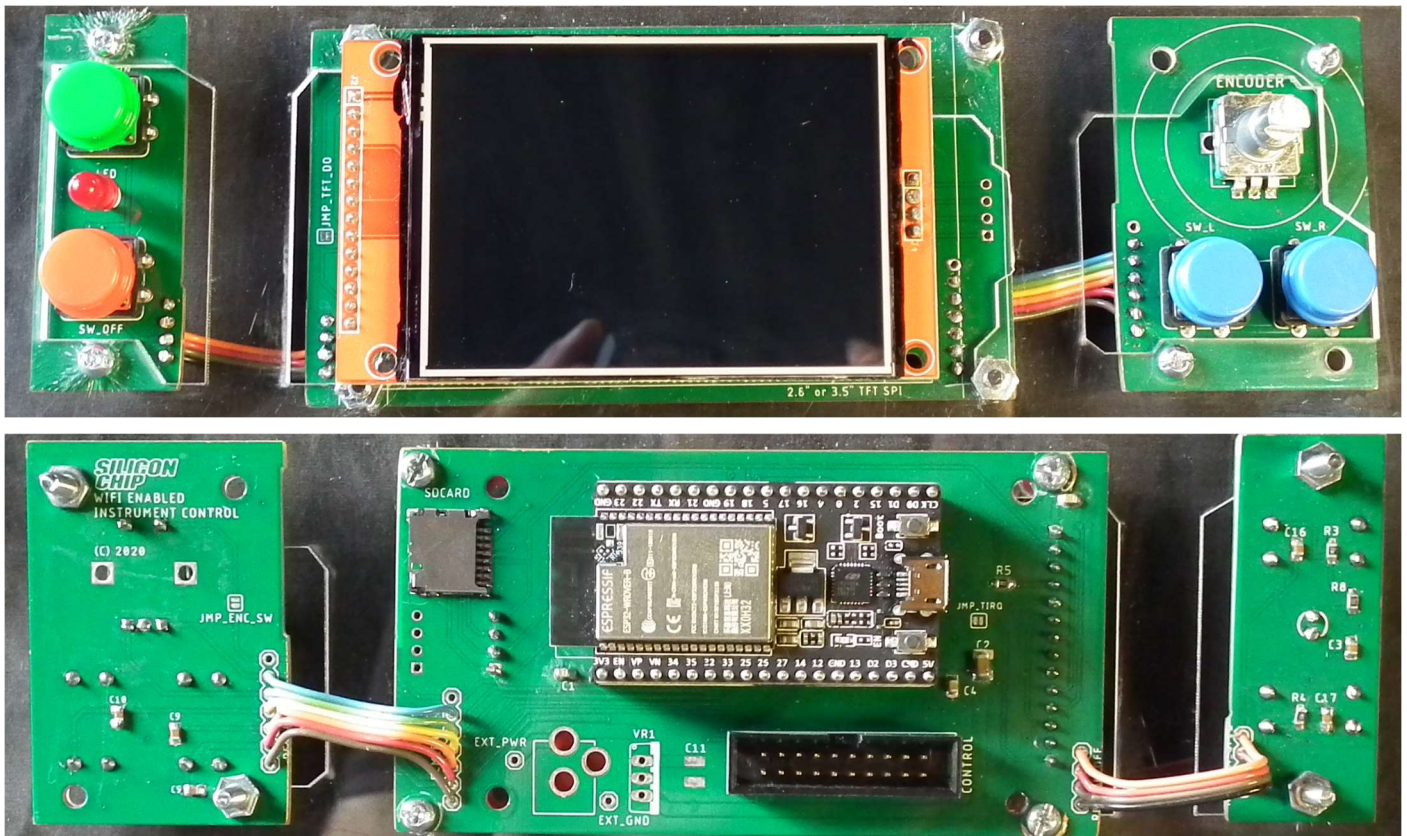
Overview

The controller module consists of a central section with the display on the front and the ESP32 DevKitC module plugged into headers on the rear. Standard 2.8" or 3.5" SPI touchscreen LCD modules are supported.

To one side of the LCD is a detachable panel containing a LED and two tactile switches, intended as on/off buttons.

On the other side two momentary switches and a rotary encoder are mounted on a second detachable panel, intended to complement the on-screen touch menu, in setting the instrument's configuration and control values.

The controller's expansion capabilities are provided on a 20 pin header and include I2C, SPI, serial, GPIO, ADC, DAC and power (3.3V and 5V) pins. It can be powered via a USB cable, an external 7-12V plugpack, or from the 'controlled' board.



In the pictures above, the side panels have been separated from the central section, and the components for external powering (barrel jack and regulator) have been omitted.

While 3.3V power is available from the ESP32 module, a maximum of 50mA should be drawn from this source.

ESP32 IO pins are not 5V tolerant.

ESP32 Pinout and Expansion Connector

Any ESP32 pin, other than SPI, may be repurposed as a GPIO or PWM pin.

Several ESP32 pins (GPIO 0, 2, 12, 14, 15, TXD, RXD) have special functions during the boot cycle and should be used carefully for other purposes.

Expansion Header

Exp. Pin	Function	GPIO	ESP32 Pin
1	GND		14, 32, 38
2			
3	SPI: MISO	19	31
4	SPI: SCK	18	30
5	I2C: SDA	21	33
6	SPI: MOSI	23	37
7	I2C: SCL	22	36
8	GPIO or I2C: SDA-1	0	25
9	GPIO 2	2	24
10	TTL Serial: RXD-2	16	27
11	TTL Serial: TXD-2	17	28
12	SW_ON	4	26
13	DAC1 or GPIO	25	9
14	DAC2, GPIO or SCL-1	26	10
15	ADC1-7 or GPIO (input)	35	6
16	SW_OFF	12	13
17	ADC1-6 or GPIO (input)	34	5
18	+5V		19
19	+3.3V		1
20	+5V		19

REM_ON_OFF Header

Exp. Pin	Function	GPIO pin
1	GND	
2	+3.3V	
3	SW_OFF	12
4	SW_ON	4

REM_ENC Header

Exp. Pin	Function	GPIO pin
1	GND	
2	+3.3V	
3	CRTL_L	14
4	CTRL_R	27
5	ENC_A	33
6	ENC_B	32
7	Encoder switch (jumper LK3)	26

ESP32 Pin Assignments

ESP32 Pin	Function	GPIO pin		ESP32 Pin	Function	GPIO pin
1	3V3 power (output)			20	DO NOT USE	
2	DO NOT USE			21	SD0, SD1, CLK	
3	EN, SVP, SVN			22		
4				23	Touch CS	15
5	ADC1-6 or GPIO	34		24	SD CS	2
6	ADC1-7 or GPIO	35		25	GPIO, SDA-1	0
7	Encoder	32		26	On switch	4
8	Encoder	33		27	GPIO, RXD2	16
9	DAC1 or GPIO	25		28	GPIO, TXD2	17
10	DAC2, GPIO or SCL-1	26		29	TFT CS	5
11	Switch R (rotary encoder side)	27		30	SCK	18
12	Switch L (rotary encoder side)	14		31	MISO	19
13	Off switch	12		32	GND	
14	GND			33	SDA	21
15	D_C (TFT)			34	RXD	RXD
16	DO NOT USE			35	TXD	TXD
17	SD2, SD3, CMD			36	SCL	22
18				37	MOSI	23
19	5v power (input)			38	GND	

Software and Libraries

The controller can be programmed using the Arduino IDE by loading the ESP32 files into Board Manager.

- Add the entry https://dl.espressif.com/dl/package_esp32_index.json to the *Additional Boards Manager URLs* list in the File > Preferences menu.
- Open the Boards Manager (*Tools > Board > Board Manager*) search for ESP32 and click ‘Install.’
- The board settings used for the Expressif DevKitC (WROOM 32) boards are:

Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Disabled"

The controller has been tested with the following general Arduino libraries:

- Adafruit GFX graphics.
- Adafruit ILI9488 and ILI9341.
- XPT2046_Touchscreen (Paul Stoffregen) in polled mode.
- ESProtary and Button2 (Lennart Hennigs) – ESP8266 libraries work on ESP32.
- ESP32 WiFi and EEPROM libraries.
- Adafruit ADS1015.

The ESP32 DevKitC example programs *File > Examples > Examples for ESP32 Dev Module* should function correctly.

The board may be programmed via USB serial or over WiFi, as explained in the *ESP32 Update > OTA example*.

Other programming environments, compatible with the ESP32, may also be used.

Communications

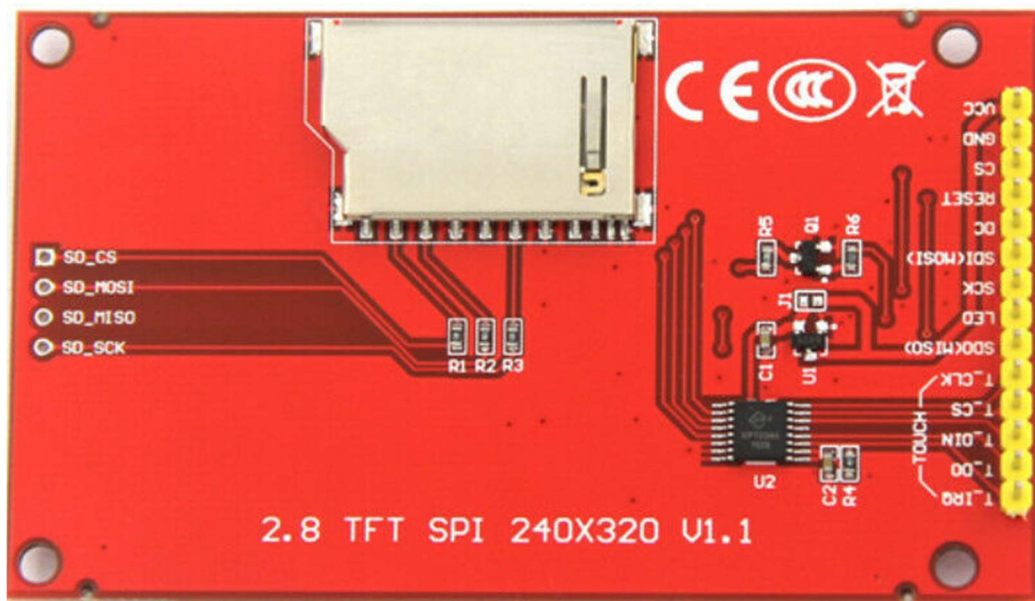
Multiple comms channels are supported, and can be used as described in the example programs for the ESP32.

- USB serial – which should be *isolated* if an alternate ground path is possible.
- WiFi - TCP/IP, UDP and Telnet are all supported.
Note: WPS connection does not function reliably with the ESP32 modules (as at January 2021).
- Serial (logic-level) up to 1MHz or more.

ESP32 pins have only modest protection against static electricity and wrong voltages (they are not 5V tolerant). Therefore, they should not be connected directly to remote equipment. Use opto-isolators or a RS232 or RS485 converter for serial communications.

Display

The controller can support any SPI LCD module with the same pinouts as the popular ILI9341 (2.8" 240 x 320) or ILI9488 (3.5" 320 x 480) Arduino modules. The SD card pins on the controller are aligned specifically for these two displays.



MOSI and SCK are shared between the LCD screen, touch panel and the onboard SD card.

Do not connect MISO if using an ILI9488 controller as it does not release the MISO line when CS is inactive. MISO is available to the TFT via a jumper LK4 for other controllers.

The interrupt pin for the touch panel controller is shared with SD_CS and is enabled by bridging LK1.

Signal	Pin (ESP 32 DevKitC)	ESP Logical	Use
MOSI	37	IO 23	TFT, Touch, SD
MISO	31	IO 19	Touch, SD (Jumper for TFT)
SCK	30	IO 18	TFT, Touch, SD
RST	N/C – pullup	–	TFT
DC	15	IO 13	TFT
CS	29	IO 5	TFT
T_CS	23	IO 15	Touch
SD_CS	24	IO 2	SD
T_IRQ	24	IO 2	Touch
LED & VCC	5V	–	All

Schematic

<copy from SC>