ESP32-S3 Pin Map

ESP32-S3 SOIC -> WROOM-1 -> ESP32-DEVKITC-1 -> T-embed -> MarauderESP32

ESP32-S3 WROOM-1 Module		ESP32-S3-DEVKITC-1 board	T-embed			ESP32Marauder	
Pin	GPIO/NET	Description *Pin = GPIO	RD	Component	Description	current state	file
27	0	BOOT	ENC_BTN	n/a	Rotary UI		
39	1	TOUCH1, **ADC1_CH0**	ENC_B	n/a	Rotary UI		
38	2	TOUCH2, **ADC1_CH1**	ENC_A	n/a	Rotary UI		
15	3	**TOUCH3, ADC1_CH2**	JTAG_PD				
4	4	TOUCH4, ADC1_CH3					
5	5	TOUCH5, **ADC1_CH4**	WCLK	1511	Speaker		
6	6	TOUCH6, **ADC1_CH5**	DOUT	1511	Speaker		
7	7	TOUCH7, **ADC1_CH6**	BCLK	1511	Speaker		
12	8	TOUCH8, **ADC1_CH7**, SUBSPICS1	SDA	GPIO	P2		
17	9	TOUCH9, ADC1_CH8, SUBSPIHD, FSPIHD	LCD_RST	ST7789	TFT	-DTFT_RST=9	platformio.ini
18	10	TOUCH10, ADC1_CH9, FSPIIO4, SUBSPICS0, **FSPICS0**	LCD_CS	ST7789	TFT	-DTFT_CS=10	platformio.ini
19	11	TOUCH11, ADC2_CH0, FSPIIO5, SUBSPID, **FSPID**	LCD_MOSI	ST7789	TFT	-DTFT_MOSI=11	platformio.ini
20	12	TOUCH12, ADC2_CH1, FSPIIO6, SUBSPICLK, **FSPICLK**	LCD_CLK	ST7789	TFT	-DTFT_SCLK=12	platformio.ini
21	13	TOUCH13, ADC2_CH2, FSPIIO7, SUBSPIQ, FSPIQ	NC_MISO			-DTFT_MISO=13	platformio.ini
22	14	TOUCH14, ADC2_CH3, FSPIDQS, SUBSPIWP, **FSPIWP**	ES_DIN	ES7210	Mic		
8	15	U0RTS, ADC2_CH4, XTAL_32K_P					
9	16	U0CTS, ADC2_CH5, XTAL_32K_N	LCD_BL	GPIO, ST7789	P4	-DTFT_BL=16	platformio.ini
10	17	U1TXD, ADC2_CH6	LCD_DC	GPIO, ST7789	P3	-DTFT_DC=17	platformio.ini
11	18	U1RXD, ADC2_CH7, **CLK_OUT3**	SCL	GPIO	P1		
13	19	U1RTS, ADC2_CH8, CLK_OUT2, **USB_D-**	USB-				
14	20	U1CTS, ADC2_CH9, CLK_OUT1, **USB_D+**	USB+				
23	21	RTC	ES_LRCK	ES7210	Mic		
#N/A	22						
#N/A	23						
#N/A	24						
#N/A	25						
#N/A	26	SPICS1					
#N/A	27	SPIHD					
#N/A	28	SPIWP					
#N/A	29	SPICS0					

#N/A	**30**	**SPICLK**					
#N/A	31	SPIQ					
#N/A	32	SPID					
#N/A	33	SPIIO4, FSPIHD, SUBSPIHD					
#N/A	34	SPIIO5, FSPICS0, SUBSPICS0					
28	35	SPIIO6, FSPID, SUBSPID					
29	36	SPIIO7, FSPICLK, SUBSPICLK					
30	37	SPIDQS, FSPIQ, SUBSPIQ					
31	**38**	FSPIWP, **SUBSPIWP**	**SD_MISO**	**SD**	**SD slot**		
32	**39**	MTCK, CLK_OUT3, **SUBSPICS1**	**SD_CS**	**SD**	**SD slot**		
33	**40**	MTDO, **CLK_OUT2**	**SD_SCLK**	**SD**	**SD slot**		
34	**41**	**MTDI**, CLK_OUT1	**SD_MOSI**	**SD**	**SD slot**		
35	**42**	**MTMS**	**SW**	**APA_102**	**RGB-LED**		
37	**43**	**U0TXD**	**TX**	**GROOVE**	**GP3**		
36	**44**	**U0RXD**	**RX**	**GROOVE**	**GP2**		
26	**45**	**VSPI_PD**	**DI**	**APA_102**	**RGB-LED**		
38	**46**	**BOOT MODE**	**3V3**	**LED**	**PWR**		
24	**47**	**SPICLK_P**	**ES_BCLK**	**ES7210**	**Mic**		
25	**48**	**SPICLK_N**	**ES_MCLK**	**ES7210**	**Mic**		
2	**3V3**	**VDDA**	**3V3**	**GPIO**	**P7**		
2	**3V3**	**VDDA**	**3V3**	**GROOVE**	**GP4**		
1,40,41	**GND**		**GND**	**GPIO**	**P6**		
1,40,41	**GND**		**GND**	**GPIO**	**P5**		
1,40,41	**GND**		**GND**	**GPIO**	**P8**		
1,40,41	**GND**		**GND**	**GROOVE**	**GP1**		
nc	**LNA_IN**		**ANT**	**Module**			
	VDD_IN			**SOIC**			
	VDD_RTC			**SOIC**			
	VSPI			**SOIC**			
	VDD_CPU			**SOIC**			
	VDDA			**SOIC**			

buttoncycler.ino

```
// Digital IO pin connected to the button. This will be driven with a
// pull-up resistor so the switch pulls the pin to ground momentarily.
// On a high -> low transition the button press logic will execute.
#define BUTTON_PIN   2
#define PIXEL_PIN    6  // Digital IO pin connected to the NeoPixels.
#define PIXEL_COUNT 16  // Number of NeoPixels
// Declare our NeoPixel strip object:
```

configs.h

```
  //// NEOPIXEL STUFF
  #if defined(ESP32_LDDB)
    #define PIN 17
  #elif defined(MARAUDER_DEV_BOARD_PRO)
    #define PIN 16
  #else
    #define PIN 25
  #endif
```

jsonConfigfile.ino

```
// To run this program, you need an SD card connected to the SPI bus as follows:
// * MOSI <-> pin 11
// * MISO <-> pin 12
// * CLK  <-> pin 13
// * CS   <-> pin 4
```

jsonserver.ino

jsonupdbeacon.ino

```
// Create the "analog" array
  JsonArray analogValues = doc.createNestedArray("analog");
  for (int pin = 0; pin < 6; pin++) {
    // Read the analog input
    int value = analogRead(pin);
    // Add the value at the end of the array
    analogValues.add(value);
  }
  // Create the "digital" array
  JsonArray digitalValues = doc.createNestedArray("digital");
  for (int pin = 0; pin < 14; pin++) {
    // Read the digital input
    int value = digitalRead(pin);
    // Add the value at the end of the array
    digitalValues.add(value);
```

longpresshandler.ino

```
#define BUTTON_A_PIN  2
```

multiplebuttons.ino

singlebutton.ino

singlebuttonsimple.ino

```
#define BUTTON_A_PIN  2
#define BUTTON_B_PIN  0
```

button2.cpp

```
pin = attachTo;
```

Kconfig

```
menu "Display Data pins"
        depends on TFT_PARALLEL_8_BIT
        config TFT_D0
            int "Data 0 pin"
            default -1
            range -1 31
        config TFT_D1
            int "Data 1 pin"
            default -1
            range -1 31
        config TFT_D2
            int "Data 2 pin"
            default -1
            range -1 31

        config TFT_D3
            int "Data 3 pin"
            default -1
            range -1 31
```

TFT_config.h

```
**********************************************************************************
**                          Section 3: Data bus Pin configuration
**********************************************************************************
// 8 BIT PARALLEL BUS
#ifdef CONFIG_TFT_PARALLEL_8_BIT
    #if CONFIG_TFT_D0 == -1
        #error "Invalid Data 0 pin. Check TFT_eSPI configuration"
    #else
        #define TFT_D0      CONFIG_TFT_D0
    #endif
```

TFT_eSPI.cpp

```
/*********************************************************************************
** Function name:           initBus
** Description:             initialise the SPI or parallel bus
**********************************************************************************
void TFT_eSPI::initBus(void) {
#ifdef TFT_CS
  pinMode(TFT_CS, OUTPUT);
  digitalWrite(TFT_CS, HIGH); // Chip select high (inactive)
#endif
// Configure chip select for touchscreen controller if present
#ifdef TOUCH_CS
  pinMode(TOUCH_CS, OUTPUT);
  digitalWrite(TOUCH_CS, HIGH); // Chip select high (inactive)
#endif
```

TFT_eSPI.h

```
int8_t pin_tft_mosi; // SPI pins
int8_t pin_tft_miso;
int8_t pin_tft_clk;
int8_t pin_tft_cs;
int8_t pin_tft_dc;   // Control pins
```

user_setup.h *see all places incl libdeps*

ReadIDBithash.ino

```
/ UNO etc
//#define TFT_MOSI  11
//#define TFT_SCK 13
//#define TFT_CS 9
//#define TFT_DC  8
//#define TFT_RESET 7
//NodeMCU
#define TFT_MOSI  D7
#define TFT_SCK D5
#define TFT_CS D8
#define TFT_DC  D3
#define TFT_RESET D4
```

config.h

```
// JOYSTICK_X_PIN and JOYSTICK_Y_PIN specify analog input pins for manually
// controlling the eye with an analog joystick.  If set to -1 or if not
// defined, the eye will move on its own.
// IRIS_PIN speficies an analog input pin for a photocell to make pupils
// react to light (or potentiometer for manual control).  If set to -1 or
// if not defined, the pupils will change on their own.
// BLINK_PIN specifies an input pin for a button (to ground) that will
// make any/all eyes blink.  If set to -1 or if not defined, the eyes will
// only blink if AUTOBLINK is defined, or if the eyeInfo[] table above
// includes wink button settings for each eye.
//#define JOYSTICK_X_PIN A0 // Analog pin for eye horiz pos (else auto)
//#define JOYSTICK_Y_PIN A1 // Analog pin for eye vert position (")
//#define JOYSTICK_X_FLIP   // If defined, reverse stick X axis
//#define JOYSTICK_Y_FLIP   // If defined, reverse stick Y axis
#define TRACKING            // If defined, eyelid tracks pupil
#define AUTOBLINK           // If defined, eyes also blink autonomously
```

servo_pin

```
#define CAPTOUCH_PIN A5 // Capacitive touch pin - attach conductive thread here
#define SERVO_PIN     4 // Servo plugged in here
```

user_xmas.cpp

```
// Pin 8 is the built-in NeoPixels on Circuit Playground Express & Bluetooth.
// With a TFT Gizmo attached, you can use A1 or A2 to easily connect a strand.
#define LED_PIN          8
#define LED_COUNT        10
#define LED_BRIGHTNESS   50 // about 1/5 brightness (max = 255)
#define TWINKLE_INTERVAL 333 // Every 333 ms (1/3 second), change a pixel
#define LIT_PIXELS       (LED_COUNT / 3) //
```

wiring.ino

```
Function     ESP32 pin       TFT 1       TFT 2
 MOSI          23      ->    SDA   ->    SDA      // The TFT pin may be named DIN
 MISO          19                                 // Not connected
 SCLK          18      ->    CLK   ->    CLK      // The TFT pin may be named SCK
 TFT_DC         2      ->    DC    ->    DC       // The TFT pin may be named AO
 TFT_RST        4      ->    RST   ->    RST
 CS 1          22      ->    CS                   // Connected to TFT 1 only
 CS 2          21                  ->    CS       // Connected to TFT 2 only
 +5V/VIN               ->    VCC   ->    VCC
 0V                    ->    GND   ->    GND
 +5V/VIN               ->    LED   ->    LED      // Some displays do not have a back
The displays used for testing were 128x128 ST7735 displays, the TFT_eSPI library se
to be changed as these displays come in many configuration variants.
```

readusersetup.ino

pio8bitparallel.h

pio8bitparallel18bpp.pio.h

pio16bitparallel.h

pio_SPI_18bit_pio.h

pio_spi.pio.h