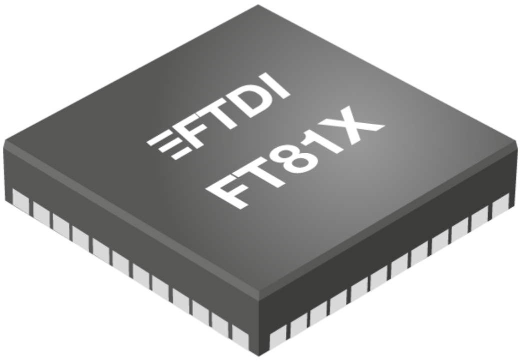
Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

Future Technology Devices International Ltd.

FT81x

(Advanced Embedded Video Engine)



The FT81x is a series of easy to use graphic controllers targeted at embedded applications to generate high-quality Human Machine Interfaces (HMIs). It has the following features:

- Advanced Embedded Video Engine(EVE) with high resolution graphics and video playback
- FT81x functionality includes graphic control, audio control, and touch control interface.
- Pinout backward compatible with FT800 (FT810) and FT801 (FT811).
- Support multiple widgets for simplified design implementation
- Built-in graphics operations allow users with little expertise to create high-quality displays
- Support 4-wire resistive touch screen (FT810/FT812)
- Support capacitive touch screen with up to 5 touches detection (FT811/FT813)
- Hardware engine can recognize touch tags and track touch movement. Provides notification for up to 255 touch tags.
- Enhanced sketch processing
- Programmable interrupt controller provides interrupts to host MCU
- Built-in 12MHz crystal oscillator with PLL providing programmable system clock up to 60MHz
- Clock switch command for internal or external clock source. External 12MHz crystal or clock input can be used for higher accuracy.
- Video RGB parallel output; configurable to support PCLK up to 60MHz and R/G/B output of 1 to 8 bits
- Programmable timing to adjust HSYNC and VSYNC timing, enabling interface to numerous displays
- Support for LCD display with resolution up to SVGA (800x600) and formats with data enable (DE) mode or VSYNC/HSYNC mode
- Support landscape and portrait orientations
- Display enable control output to LCD panel
- Integrated 1MByte graphics RAM, no frame buffer RAM required
- Support playback of motion-JPEG encoded AVI videos
- Mono audio channel output with PWM output
- Built-in sound synthesizer
- Audio wave playback for mono 8-bit linear PCM, 4-bit ADPCM and µ-Law coding format at sampling frequencies from 8kHz to 48kHz. Built-in digital filter reduces the system design complexity of external filtering
- PWM output for display backlight dimming control
- Advanced object oriented architecture enables low cost MPU/MCU as system host using SPI interfaces
- Support SPI data lines in single, dual or quad mode; SPI clock up to 30MHz
- Power mode control allows the chip to be put in power down, sleep and standby states
- Supports I/O voltage from 1.8V to 3.3V
- Internal voltage regulator supplies 1.2V to the digital core
- Build-in Power-on-reset circuit
- -40°C to 85°C extended operating temperature range
- Available in a compact Pb-free, VQFN-48 and VQFN-56 package, RoHS compliant

Disclaimer:

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied. Future Technology Devices International Ltd will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected. This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury. This document provides preliminary information that may be subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document.

Future Technology Devices International Ltd
Unit 1, 2 Seaward Place
Centurion Business Park
Glasgow G41 1HH
United Kingdom

Scotland Registered Company Number: SC136640

1 Typical Applications

- Point of Sales Machines
- Multi-function Printers
- Instrumentation
- Home Security Systems
- Graphic touch pad – remote, dial pad
- Tele / Video Conference Systems
- Phones and Switchboards
- Medical Appliances
- Blood Pressure displays
- Heart monitors
- Glucose level displays
- Breathalyzers
- Gas chromatographs
- Power meter
- Home appliance devices
- Set-top box
- Thermostats
- Sprinkler system displays
- Medical Appliances
- GPS / Satnav
- Vending Machine Control Panels
- Elevator Controls
- ……and many more

1.1 Part Numbers

Part Number	Description	Package
FT810Q-x	EVE with 18 bit RGB, resistive touch	48 Pin VQFN, body 7 x 7 mm, pitch 0.5mm
FT811Q-x	EVE with 18 bit RGB, capacitive touch	48 Pin VQFN, body 7 x 7 mm, pitch 0.5mm
FT812Q-x	EVE with 24 bit RGB, resistive touch	56 Pin VQFN, body 8 x 8 mm, pitch 0.5mm
FT813Q-x	EVE with 24 bit RGB, capacitive touch	56 Pin VQFN, body 8 x 8 mm, pitch 0.5mm

Table 1- FT81x Embedded Video Engine Part Numbers

Note: Packaging codes for x is:

-R: Taped and Reel (3000pcs per reel)

-T: Tray packing (260 pcs per tray for VQFN-48, 348 pcs per tray for VQFN-56)

For example: FT810Q-R is 3000 VQFN pieces in taped and reel packaging

2 Block Diagram

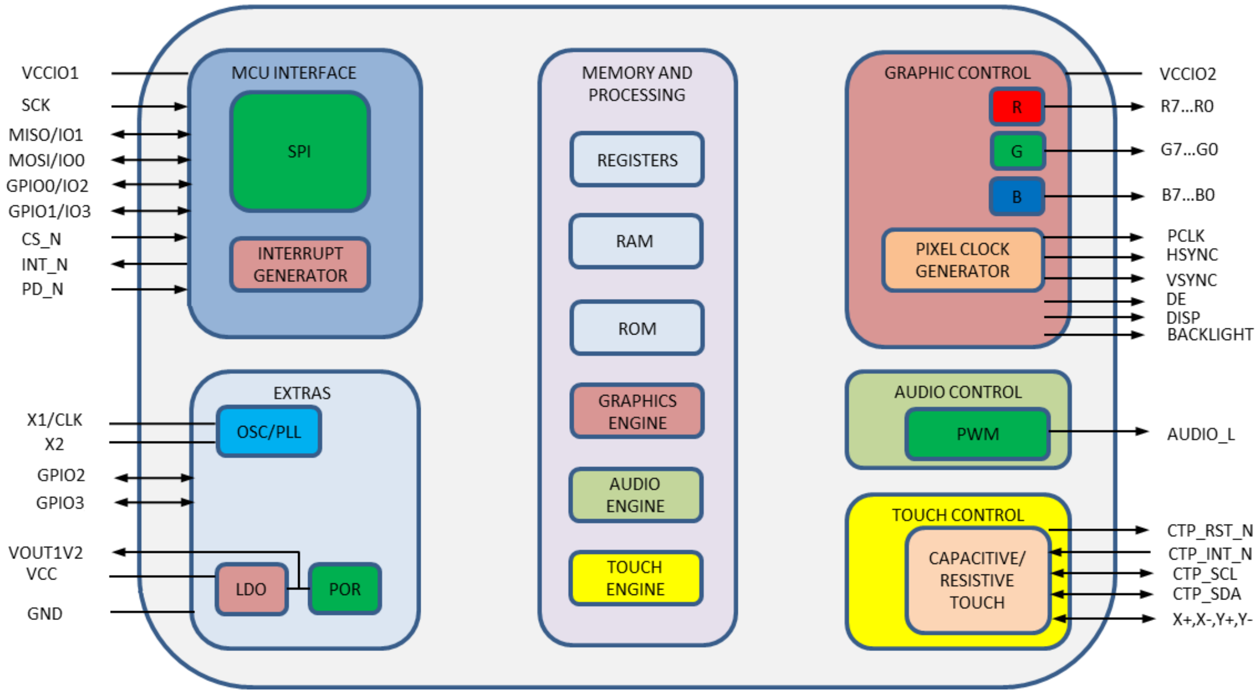


Figure 2-1 FT81x Block Diagram

For a description of each function please refer to Section 4.

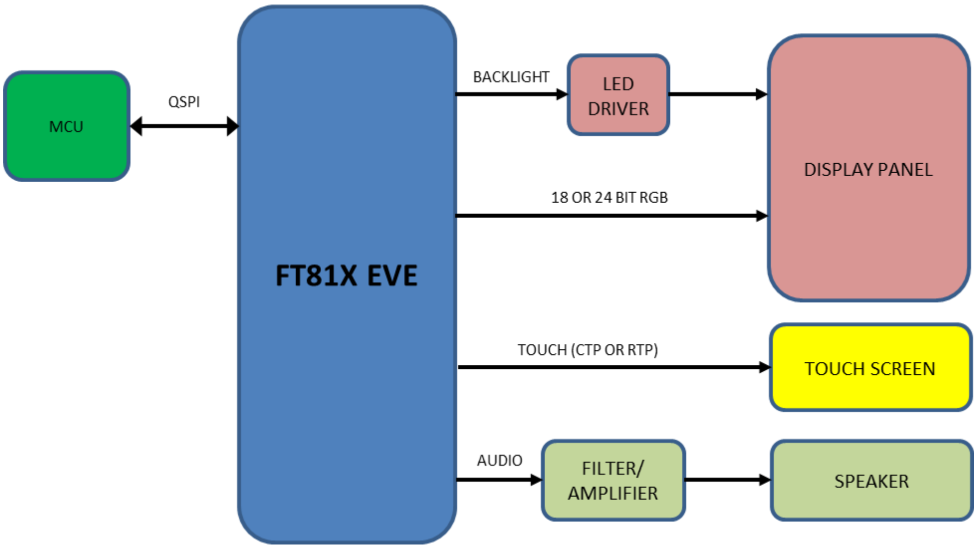


Figure 2-2 FT81x System Design Diagram

FT81x with EVE (Embedded Video Engine) technology simplifies the system architecture for advanced human machine interfaces (HMIs) by providing support for display, audio, and touch as well as an object oriented architecture approach that extends from display creation to the rendering of the graphics.

Contents

1 Typical Applications 3
1.1 Part Numbers 3
2 Block Diagram 4
3 Device Pin Out and Signal Description 7
3.1 FT810 VQFN-48 Package Pin Out 7
3.2 FT811 VQFN-48 Package Pin Out 7
3.3 FT812 VQFN-56 Package Pin Out 8
3.4 FT813 VQFN-56 Package Pin Out 8
3.5 Pin Description 9
4 Function Description 13
4.1 Quad SPI Host Interface 13
4.1.1 QSPI Interface 13
4.1.2 Serial Data Protocol 15
4.1.3 Host Memory Read 15
4.1.4 Host Memory Write 16
4.1.5 Host Command 16
4.1.6 Interrupts 20
4.2 System Clock 20
4.2.1 Clock Source 20
4.2.2 Phase Locked Loop 21
4.2.3 Clock Enable 21
4.2.4 Clock Frequency 22
4.3 Graphics Engine 22
4.3.1 Introduction 22
4.3.2 ROM and RAM Fonts 22
4.4 Parallel RGB Interface 26
4.5 Miscellaneous Control 28
4.5.1 Backlight Control Pin 29
4.5.2 DISP Control Pin 29
4.5.3 General Purpose IO pins 29
4.5.4 Pins Drive Current Control 29
4.6 Audio Engine 30
4.6.1 Sound Synthesizer 30
4.6.2 Audio Playback 32
4.7 Touch-Screen Engine 32
4.7.1 Resistive Touch Control 32
4.7.2 Capacitive Touch Control 33
4.7.3 Compatibility mode 34

4.7.4 Extended mode ... 34
4.7.5 Short-circuit protection ... 35
4.7.6 Capacitive touch configuration ... 35
4.7.7 Touch detection in none-ACTIVE state ... 35
4.8 Power Management ... 35
4.8.1 Power supply ... 35
4.8.2 Internal Regulator and POR ... 36
4.8.3 Power Modes ... 37
4.8.4 Reset and boot-up sequence ... 38
4.8.5 Pin Status at Different Power States ... 38
5 Memory Map ... 40
5.1 Registers ... 40
5.2 Chip ID ... 45
6 Devices Characteristics and Ratings ... 46
6.1 Absolute Maximum Ratings ... 46
6.2 ESD and Latch-up Specifications ... 46
6.3 DC Characteristics ... 46
6.4 AC Characteristics ... 49
6.4.1 System clock and reset ... 49
6.4.2 SPI interface timing ... 49
6.4.3 RGB Interface Timing ... 50
7 Application Examples ... 52
8 Package Parameters ... 54
8.1 VQFN-48 Package Dimensions ... 54
8.2 VQFN-56 Package Dimensions ... 54
8.3 Solder Reflow Profile ... 55
9 Contact Information ... 56
Appendix A – References ... 57
Appendix B - List of Figures and Tables ... 58
Appendix C - Revision History ... 60

3 Device Pin Out and Signal Description

3.1 FT810 VQFN-48 Package Pin Out

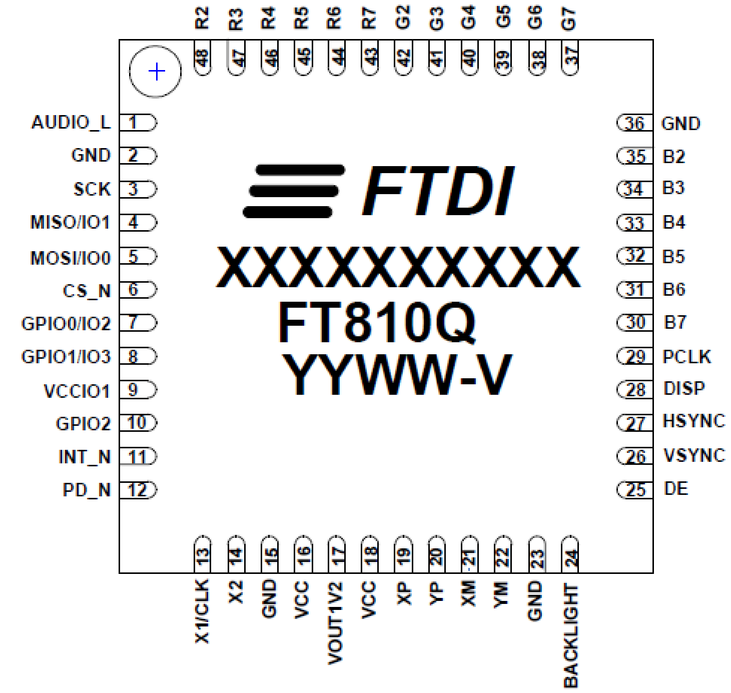


Figure 3-1 Pin Configuration FT810 VQFN-48 (top view)

3.2 FT811 VQFN-48 Package Pin Out

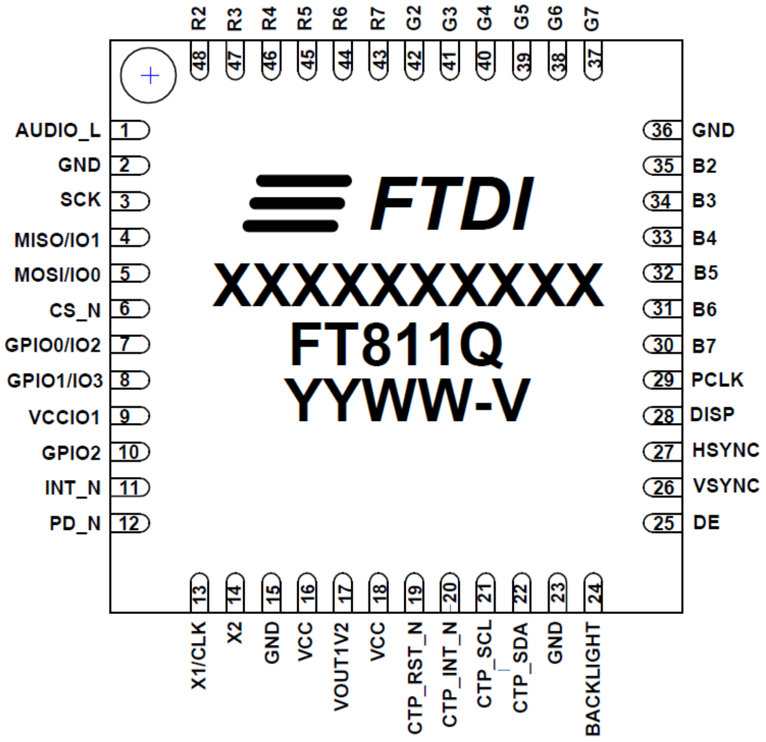


Figure 3-1 Pin Configuration FT811 VQFN-48 (top view)

3.3 FT812 VQFN-56 Package Pin Out

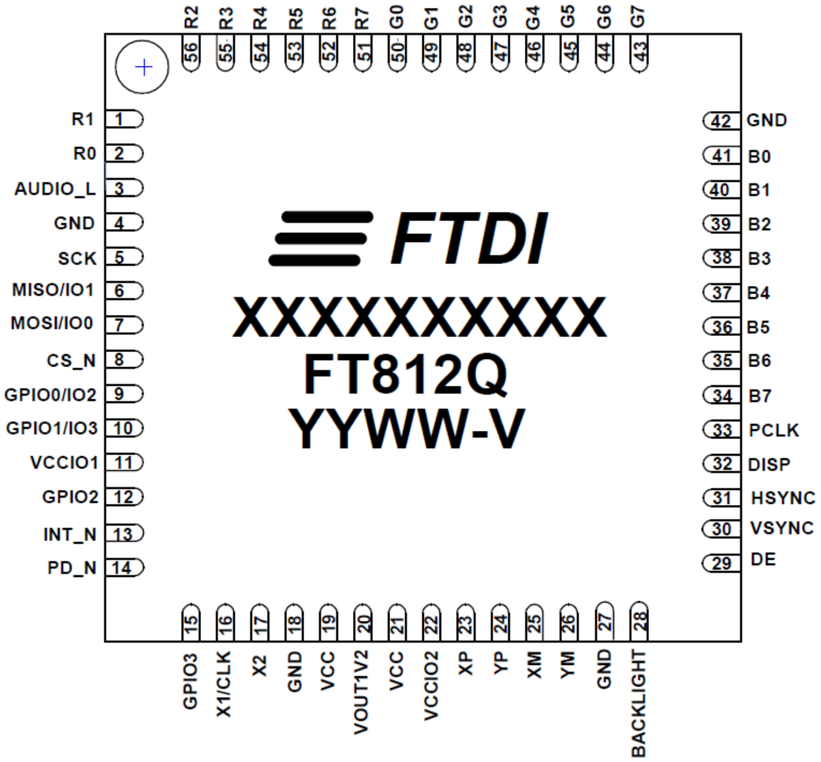


Figure 3-1 Pin Configuration FT812 VQFN-56 (top view)

3.4 FT813 VQFN-56 Package Pin Out

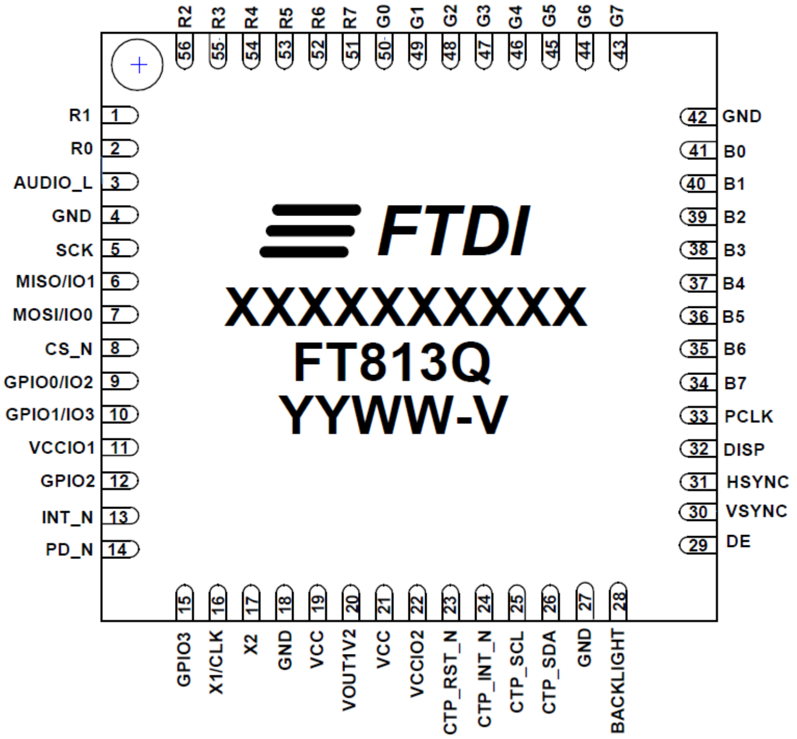


Figure 3-1 Pin Configuration FT813 VQFN-56 (top view)

3.5 Pin Description

Table 3-1 FT81x pin description

Pin Number				Pin Name	Type	Description
FT810	FT811	FT812	FT813			
-	-	1	1	R1	O	Bit 1 of Red RGB signals Powered from pin VCCIO2
-	-	2	2	R0	O	Bit 0 of Red RGB signals Powered from pin VCCIO2
1	1	3	3	AUDIO_L	O	Audio PWM out Powered from pin VCC
2	2	4	4	GND	P	Ground
3	3	5	5	SCK	I	SPI clock input Powered from pin VCCIO1
4	4	6	6	MISO/IO1	I/O	SPI Single mode: SPI MISO output SPI Dual/Quad mode: SPI data line 1 Powered from pin VCCIO1
5	5	7	7	MOSI/IO0	I/O	SPI Single mode: SPI MOSI input SPI Dual/Quad mode: SPI data line 0 Powered from pin VCCIO1
6	6	8	8	CS_N	I	SPI slave select input Powered from pin VCCIO1
7	7	9	9	GPIO0/IO2	I/O	SPI Single/Dual mode: General purpose IO 0 SPI Quad mode: SPI data line 2 Powered from pin VCCIO1
8	8	10	10	GPIO1/IO3	I/O	SPI Single/Dual mode: General purpose IO 1 SPI Quad mode: SPI data line 3 Powered from pin VCCIO1
9	9	11	11	VCCIO1	P	I/O power supply for host interface pins. Support 1.8V, 2.5V or 3.3V.
10	10	12	12	GPIO2	I/O	General purpose IO 2 Powered from pin VCCIO1
11	11	13	13	INT_N	OD/ O	Interrupt to host, open drain output(default) or push-pull output, active low
12	12	14	14	PD_N	I	Chip power down mode control input, active low. Connect to MCU GPIO for power management or hardware reset function, or pulled up to VCCIO1 through 47kΩ resistor and 100nF to ground. Powered from pin VCCIO1
-	-	15	15	GPIO3	I/O	General purpose IO 3 Powered from pin VCCIO1

Pin Number				Pin Name	Type	Description
FT810	FT811	FT812	FT813			
13	13	16	16	X1/CLK	I	Crystal oscillator or clock input; Connect to GND if not used. 3.3V peak input allowed. Powered from pin VCC.
14	14	17	17	X2	O	Crystal oscillator output; leave open if not used. Powered from pin VCC.
15	15	18	18	GND	P	Ground
16	16	19	19	VCC	P	3.3V power supply input.
17	17	20	20	VOUT1V2	O	1.2V regulator output pin. Connect a 4.7uF decoupling capacitor to GND.
18	18	21	21	VCC	P	3.3V power supply input.
		22	22	VCCIO2	P	I/O power supply for RGB and touch pins. For QFN-48 package, VCCIO2 is bonded together with VCC pin; For QFN-56 package, VCCIO2 is separate from VCC pin. VCCIO2 supports 1.8V, 2.5V or 3.3V. VCCIO2 can be connected to different voltage with VCCIO1.
19		23		XP	AI/O	Connect to X right electrode of 4-wire resistive touch-screen panel. Powered from pin VCCIO2.
20		24		YP	AI/O	Connect to Y top electrode of 4-wire resistive touch-screen panel. Powered from pin VCCIO2.
21		25		XM	AI/O	Connect to X left electrode of 4-wire resistive touch-screen panel. Powered from pin VCCIO2.
22		26		YM	AI/O	Connect to Y bottom electrode of 4-wire resistive touch-screen panel. Powered from pin VCCIO2.
-	19	-	23	CTP_RST_N	O	Connect to reset pin of the CTPM. Powered from pin VCCIO2.
-	20	-	24	CTP_INT_N	I	Connect to interrupt pin of the CTPM. Powered from pin VCCIO2.
-	21	-	25	CTP_SCL	I/OD	Connect to I2C SCL pin of the CTPM. Powered from pin VCCIO2.
-	22	-	26	CTP_SDA	I/OD	Connect to I2C SDA pin of the CTPM. Powered from pin VCCIO2.
23	23	27	27	GND	P	Ground
24	24	28	28	BACKLIGHT	O	LED Backlight brightness PWM control signal.

Pin Number				Pin Name	Type	Description
FT810	FT811	FT812	FT813			
						Powered from pin VCCIO2.
25	25	29	29	DE	O	LCD Data Enable. Powered from pin VCCIO2.
26	26	30	30	VSYNC	O	LCD Vertical Sync. Powered from pin VCCIO2.
27	27	31	31	HSYNC	O	LCD Horizontal Sync. Powered from pin VCCIO2.
28	28	32	32	DISP	O	LCD Display Enable. Powered from pin VCCIO2.
29	29	33	33	PCLK	O	LCD Pixel Clock. Powered from pin VCCIO2.
30	30	34	34	B7	O	Bit 7 of Blue RGB signals. Powered from pin VCCIO2.
31	31	35	35	B6	O	Bit 6 of Blue RGB signals. Powered from pin VCCIO2.
32	32	36	36	B5	O	Bit 5 of Blue RGB signals. Powered from pin VCCIO2.
33	33	37	37	B4	O	Bit 4 of Blue RGB signals. Powered from pin VCCIO2.
34	34	38	38	B3	O	Bit 3 of Blue RGB signals. Powered from pin VCCIO2.
35	35	39	39	B2	O	Bit 2 of Blue RGB signals. Powered from pin VCCIO2.
-	-	40	40	B1	O	Bit 1 of Blue RGB signals. Powered from pin VCCIO2.
-	-	41	41	B0	O	Bit 0 of Blue RGB signals. Powered from pin VCCIO2.
36	36	42	42	GND	P	Ground
37	37	43	43	G7	O	Bit 7 of Green RGB signals. Powered from pin VCCIO2.
38	38	44	44	G6	O	Bit 6 of Green RGB signals. Powered from pin VCCIO2.
39	39	45	45	G5	O	Bit 5 of Green RGB signals. Powered from pin VCCIO2.
40	40	46	46	G4	O	Bit 4 of Green RGB signals. Powered from pin VCCIO2.
41	41	47	47	G3	O	Bit 3 of Green RGB signals. Powered from pin VCCIO2.

Pin Number				Pin Name	Type	Description
FT810	FT811	FT812	FT813			
42	42	48	48	G2	O	Bit 2 of Green RGB signals. Powered from pin VCCIO2.
-	-	49	49	G1	O	Bit 1 of Green RGB signals. Powered from pin VCCIO2.
-	-	50	50	G0	O	Bit 0 of Green RGB signals. Powered from pin VCCIO2.
43	43	51	51	R7	O	Bit 7 of Red RGB signals. Powered from pin VCCIO2.
44	44	52	52	R6	O	Bit 6 of Red RGB signals. Powered from pin VCCIO2.
45	45	53	53	R5	O	Bit 5 of Red RGB signals. Powered from pin VCCIO2.
46	46	54	54	R4	O	Bit 4 of Red RGB signals. Powered from pin VCCIO2.
47	47	55	55	R3	O	Bit 3 of Red RGB signals. Powered from pin VCCIO2.
48	48	56	56	R2	O	Bit 2 of Red RGB signals. Powered from pin VCCIO2.
EP	EP	EP	EP	GND	P	Ground. Exposed thermal pad.

Note:

P	: Power or ground
I	: Input
O	: Output
OD	: Open drain output
I/O	: Bi-direction Input and Output
AI/O	: Analog Input and Output

4 Function Description

The FT81x is a single chip, embedded video controller with the following function blocks:

- Quad SPI Host Interface
- System Clock
- Graphics Engine
- Parallel RGB video interface
- Audio Engine
- Touch-screen support and interface
- Power Management

The functions for each block are briefly described in the following subsections.

4.1 Quad SPI Host Interface

The FT81x uses a quad serial parallel interface (QSPI) to communicate with host microcontrollers and microprocessors.

4.1.1 QSPI Interface

The QSPI slave interface operates up to 30MHz. Only SPI mode 0 is supported. Refer to section 6.4.2 for detailed timing specification. The QSPI can be configured as a SPI slave in SINGLE, DUAL or QUAD channel modes.

By default the SPI slave operates in the SINGLE channel mode with MOSI as input from the master and MISO as output to the master. DUAL and QUAD channel modes can be configured through the SPI slave itself. To change the channel modes, write to register REG_SPI_WIDTH. The table below depicts the setting.

Table 4-1 QSPI channel selection

REG_SPI_WIDTH[1:0]	Channel Mode	Data pins	Max bus speed
00	SINGLE – default mode	MISO, MOSI	30 MHz
01	DUAL	IO0, IO1	30 MHz
10	QUAD	IO0, IO1, IO2, IO3	25 MHz
11	Reserved	-	-

With DUAL/QUAD channel modes, the SPI data ports are now unidirectional. In these modes, each SPI transaction (signified by CS_N going active low) will begin with the data ports set as inputs.

Hence, for writing to the FT81x, the protocol will operate as in FT800, with "WR-Command/Addr2, Addr1, Addr0, DataX, DataY, DataZ …" The write operation is considered complete when CS_N goes inactive high.

For reading from the FT81x, the protocol will still operate as in FT800, with "RD-Command/Addr2, Addr1, Addr0, Dummy-Byte, DataX, DataY, DataZ". However as the data ports are now unidirectional, a change of port direction will occur before DataX is clocked out of the FT81x. Therefore it is important that the firmware controlling the SPI master changes the SPI master data port direction to "input" after transmitting Addr0. The FT81x will not change the port direction till it starts to clock out DataX. Hence, the Dummy-Byte cycles will be used as a change-over period when neither the SPI master nor slave will be driving the bus; the data paths thus must have pull-ups/pull-downs. The SPI slave from the FT81x will reset all its data ports' direction to input once CS_N goes inactive high (i.e. at the end of the current SPI master transaction).

The diagram depicts the behaviour of both the SPI master and slave in the master read case.

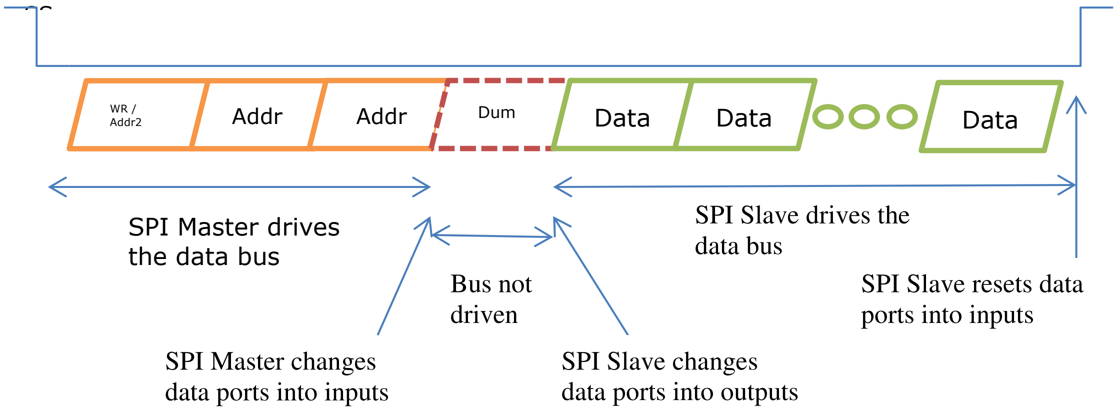

WR / Addr2
Addr
Addr
Dum
Data
Data
Data
SPI Master drives the data bus
Bus not driven
SPI Slave drives the data bus
SPI Slave resets data ports into inputs
SPI Master changes data ports into inputs
SPI Slave changes data ports into outputs

Figure 4-1 SPI master and slave in the master read case

In the DUAL channel mode, MISO (MSB) and MOSI are used while in the QUAD channel mode. IO3 (MSB), IO2, MISO and MOSI are used.

Figure 4-2 illustrates a direct connection to a 1.8-3.3V IO MPU/MCU with single or dual SPI interface.

Figure 4-3 illustrates a direct connection to a 1.8-3.3V IO MPU/MCU with Quad SPI interface.

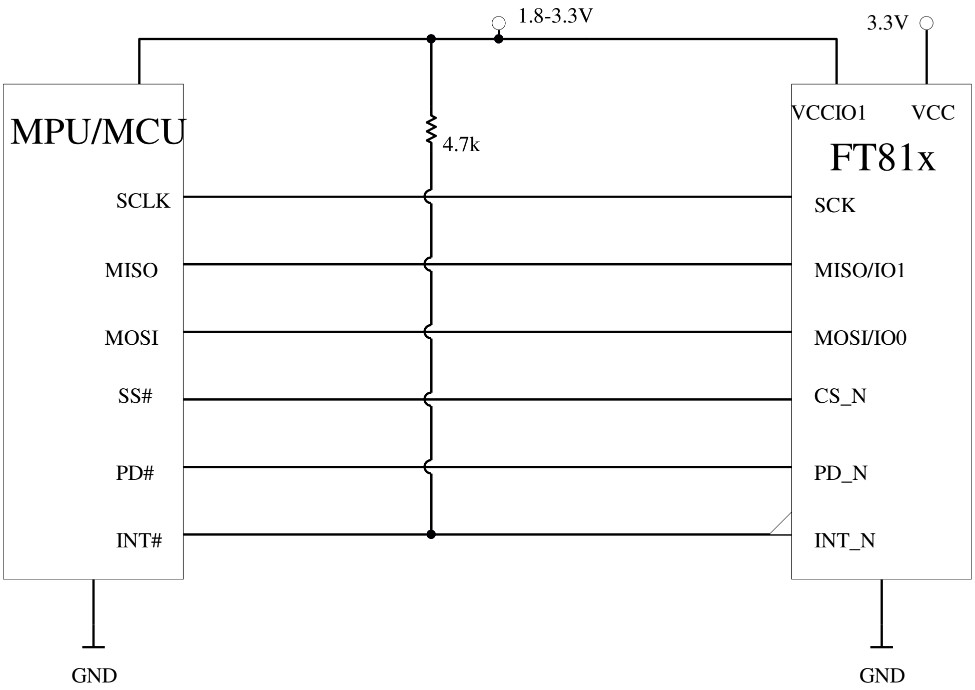


Figure 4-2 Single/Dual SPI Interface connection

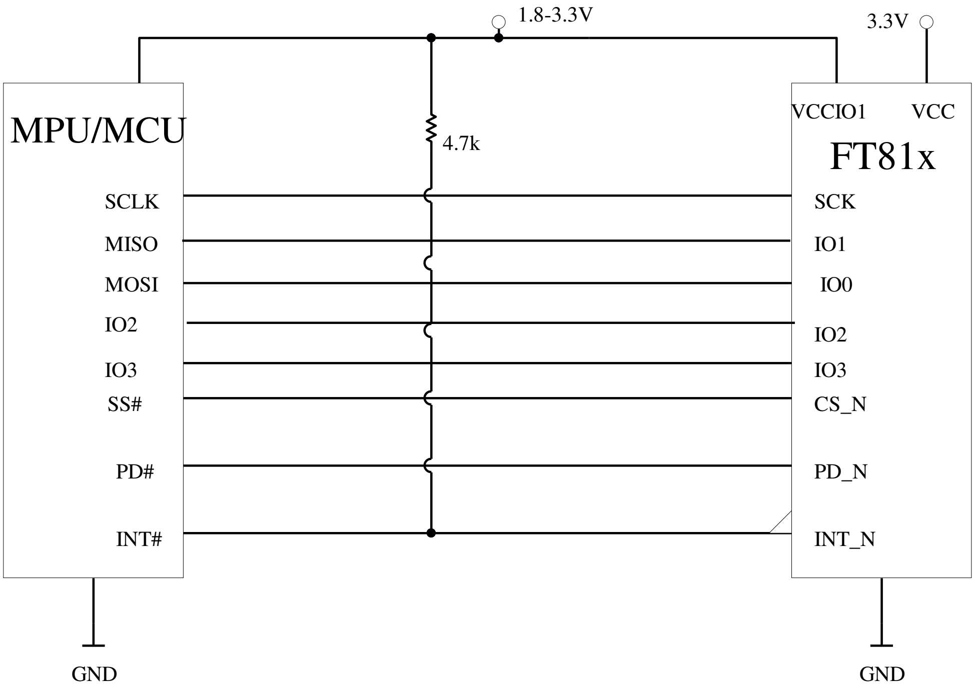


Figure 4-3 Quad SPI Interface connection

4.1.2 Serial Data Protocol

The FT81x appears to the host MPU/MCU as a memory-mapped SPI device. The host communicates with the FT81x using reads and writes to a large (4 megabyte) address space. Within this address space are dedicated areas for graphics, audio and touch control. Refer to section 5 for the detailed memory map.

The host reads and writes the FT81x address space using SPI transactions. These transactions are memory read, memory write and command write. Serial data is sent by the most significant bit first.

Each transaction starts with CS_N goes low, and ends when CS_N goes high. There's no limit on data length within one transaction, as long as the memory address is continuous.

4.1.3 Host Memory Read

For SPI memory read transactions, the host sends two zero bits, followed by the 22-bit address. This is followed by a dummy byte. After the dummy byte, the FT81x responds to each host byte with read data bytes.

Table 4-2 Host memory read transaction

7	6	5	4	3	2	1	0	
0	0	Address [21:16]						Write Address
Address [15:8]								
Address [7:0]								
Dummy byte								
Byte 0								Read Data
Byte *n*								

4.1.4 Host Memory Write

For SPI memory write transactions, the host sends a '1' bit and '0' bit, followed by the 22-bit address. This is followed by the write data.

Table 4-3 Host memory write transaction

7	6	5	4	3	2	1	0	
1	0	Address [21:16]						Write Address
Address [15:8]								
Address [7:0]								
Byte 0								Write Data
Byte *n*								

4.1.5 Host Command

When sending a command, the host transmits a 3 byte command. Table 4-5 Host command lists all the host command functions.

For SPI command transactions, the host sends a '0' bit and '1' bit, followed by the 6-bit command code. The 2nd byte can be either 00h, or the parameter of that command. The 3rd byte is fixed at 00h.

All SPI commands except the system reset can only be executed when the SPI is in the Single channel mode. They will be ignored when the SPI is in either Dual or Quad channel mode.

Some commands are used to configure the device and these configurations will be reset upon receiving the SPI PWRDOWN command, except those that configure the pin state during power down. These commands will be sticky unless reconfigured or power-on-reset (POR) occurs.

Table 4-4 Host command transaction

7	6	5	4	3	2	1	0	
0	1	Command [5:0]						1st Byte
Parameter for the command								2nd Byte
0	0	0	0	0	0	0	0	3rd Byte

Table 4-5 Host command list

1st Byte	2nd byte	3rd byte	Command	Description
Power Modes				
00000000b	00000000b	00000000b	00h ACTIVE	Switch from Standby/Sleep/PWRDOWN modes to active mode. Dummy memory read from address 0(read twice) generates ACTIVE command.

<table>
<tr><th>1st Byte</th><th>2nd byte</th><th>3rd byte</th><th>Command</th><th>Description</th></tr>
<tr><td>01000001b</td><td>00000000b</td><td>00000000b</td><td>41h
STANDBY</td><td>Put FT81x core to standby mode. Clock gate off, PLL and Oscillator remain on (default). ACTIVE command to wake up.</td></tr>
<tr><td>01000010b</td><td>00000000b</td><td>00000000b</td><td>42h
SLEEP</td><td>Put FT81x core to sleep mode. Clock gate off, PLL and Oscillator off. ACTIVE command to wake up.</td></tr>
<tr><td>01000011b
01010000b</td><td>00000000b</td><td>00000000b</td><td>43h/50h
PWRDOWN</td><td>Switch off 1.2V core voltage to the digital core circuits. Clock, PLL and Oscillator off. SPI is alive. ACTIVE command to wake up.</td></tr>
<tr><td>01000100b</td><td>xx</td><td>00000000b</td><td>49h
PD_ROMS</td><td>Select power down individual ROMs; Byte2 determines which ROM to power down or up. A 1 on a bit powers down the corresponding block; a 0 on a bit powers up the corresponding block. As these are not readable, the host must remember the setting on its own.
<table>
<tr><td>Byte2[7]</td><td>ROM_MAIN</td></tr>
<tr><td>Byte2[6]</td><td>ROM_RCOSATAN</td></tr>
<tr><td>Byte2[5]</td><td>ROM_SAMPLE</td></tr>
<tr><td>Byte2[4]</td><td>ROM_JABOOT</td></tr>
<tr><td>Byte2[3]</td><td>ROM_J1BOOT</td></tr>
<tr><td>Byte2[2-0]</td><td>reserved</td></tr>
</table></td></tr>
<tr><td colspan="5"><b>Clock and Reset</b></td></tr>
<tr><td>01000100b</td><td>00000000b</td><td>00000000b</td><td>44h
CLKEXT</td><td>Select PLL input from external crystal oscillator or external input clock. No effect if external clock is already selected, otherwise a system reset will be generated</td></tr>
<tr><td>01001000b</td><td>00000000b</td><td>00000000b</td><td>48h
CLKINT</td><td>Select PLL input from internal relaxation oscillator (default). No effect if internal clock is already selected, otherwise a system reset will be generated</td></tr>
<tr><td>01100001b
01100010b</td><td>xx</td><td>00000000b</td><td>61h/62h
CLKSEL</td><td>This command will only be effective when the PLL is stopped (SLEEP mode).
For compatibility to FT800/FT801, set Byte2 to 0x00. This will set the PLL clock back to default (60 MHz).
<table>
<tr><td>Byte2 [5:0]</td><td>sets the clock frequency</td></tr>
<tr><td>0</td><td>Set to default clock speed</td></tr>
<tr><td>1</td><td>Reserved</td></tr>
<tr><td>2 to 5</td><td>2 to 5 times the osc frequency (i.e. 24 to 60MHz with 12MHz oscillator)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>1st Byte</th><th>2nd byte</th><th>3rd byte</th><th>Command</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td><table><tr><td>Byte2 [7:6]</td><td>sets the PLL range</td></tr><tr><td>0</td><td>When Byte2[5:0] = 0, 2, 3</td></tr><tr><td>1</td><td>When Byte2[5:0] = 4, 5</td></tr></table></td></tr>
<tr><td>01101000b</td><td>00000000b</td><td>00000000b</td><td>68h
RST_PULSE</td><td>Send reset pulse to FT81x core. The behaviour is the same as POR except that settings done through SPI commands will not be affected</td></tr>
<tr><td colspan="5"><b>Configuration</b></td></tr>
<tr><td>01110000b</td><td>xx</td><td>00000000b</td><td>70h
PINDRIVE</td><td>This will set the drive strength for various pins. For FT800/FT801 compatibility, by default those settings are from the GPIO registers. FT81x supports setting the drive strength via SPI command instead.
When PINDRIVE for a pin from the SPI command is not updated, the drive strength will be determined by its corresponding GPIO register bits, if they exist. If they don't exist, a hard coded setting is used. Please refer to Table 4-20 for default values.
When PINDRIVE for a pin from the SPI command is updated, it will override the corresponding setting in the GPIO register bits.
Byte2 determines which pin and the setting are to be updated.
Byte2[1:0] determine the drive strength:<table><tr><td>Byte2 [1:0]</td><td colspan="2">Drive Strength</td></tr><tr><td>0h</td><td>5mA</td><td></td></tr><tr><td>1h</td><td>10.0mA</td><td></td></tr><tr><td>2h</td><td>15.0mA</td><td></td></tr><tr><td>3h</td><td>20.0mA</td><td></td></tr></table>Byte[7:2] determine which pin/pin group to set:<table><tr><td>Byte2 [7:2]</td><td>Pin / Pin Group</td></tr><tr><td>00h</td><td>GPIO 0</td></tr><tr><td>01h</td><td>GPIO 1</td></tr><tr><td>02h</td><td>GPIO 2</td></tr><tr><td>03h</td><td>GPIO 3</td></tr></table></td></tr>
</table>

<table>
<tr><th>1st Byte</th><th>2nd byte</th><th>3rd byte</th><th>Command</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><td>04-07h</td><td>Reserved</td></tr>
<tr><td>08h</td><td>DISP</td></tr>
<tr><td>09h</td><td>DE</td></tr>
<tr><td>0Ah</td><td>VSYNC / HSYNC</td></tr>
<tr><td>0Bh</td><td>PCLK</td></tr>
<tr><td>0Ch</td><td>BACKLIGHT</td></tr>
<tr><td>0Dh</td><td>R[7:0], G[7:0], B[7:0]</td></tr>
<tr><td>0Eh</td><td>AUDIO_L</td></tr>
<tr><td>0Fh</td><td>INT_N</td></tr>
<tr><td>10h</td><td>CTP_RST_N</td></tr>
<tr><td>11h</td><td>CTP_SCL</td></tr>
<tr><td>12h</td><td>CTP_SDA</td></tr>
<tr><td>13h</td><td>SPI MISO/MOSI/IO2/IO3</td></tr>
<tr><td>Others</td><td>Reserved</td></tr>
</table>
Note: GPIO0 shares the same pin as SPI IO2 and GPIO1 with SPI IO3. When SPI is set in Quad mode, IO2 and IO3 will inherit the drive strength set in GROUP 13h; otherwise GPIO0 and GPIO1 will inherit the drive strength from GROUP 00h and 01h respectively.</td></tr>
<tr><td>01110001b</td><td>xx</td><td>00000000b</td><td>71h
PIN_PD_STATE</td><td>During power down, all output and in/out pins will not be driven. Please refer to Table 4-20 for their default power down state.
These settings will only be effective during power down and will not affect normal operations. Also note that these configuration bits are sticky and, unlike other configuration bits, will not reset to default values upon exiting power down. Only POR will reset them.
Byte2 determines which pin and the setting are to be updated.
Byte2[1:0] determine the pin state.
<table>
<tr><td>Byte2 [1:0]</td><td>Pin Setting</td></tr>
<tr><td>0h</td><td>Float</td></tr>
<tr><td>1h</td><td>Pull-Down</td></tr>
</table></td></tr>
</table>

1st Byte	2nd byte	3rd byte	Command	Description
				2h Pull-Up 3h Reserved Byte2[7:2] determine which pin/pin group to set. Please refer to the table in command PINDRIVE entry.

NOTE: Any command code not specified is reserved and should not be used by the software

4.1.6 Interrupts

The interrupt output pin is enabled by REG_INT_EN. When REG_INT_EN is 0, INT_N is tri-state (pulled to high by external pull-up resistor). When REG_INT_EN is 1, INT_N is driven low when any of the interrupt flags in REG_INT_FLAGS are high, after masking with REG_INT_MASK. Writing a '1' in any bit of REG_INT_MASK will enable the corresponding interrupt. Each bit in REG_INT_FLAGS is set by a corresponding interrupt source. REG_INT_FLAGS is readable by the host at any time, and clears when read.

The INT_N pin is open-drain (OD) output by default. It can be configured to push-pull output by register REG_GPIOX.

Table 4-6 Interrupt Flags bit assignment

Bit	7	6	5	4
Interrupt Sources	CONVCOMPLETE	CMDFLAG	CMDEMPTY	PLAYBACK
Conditions	Touch-screen conversions completed	Command FIFO flag	Command FIFO empty	Audio playback ended
Bit	3	2	1	0
Interrupt Sources	SOUND	TAG	TOUCH	SWAP
Conditions	Sound effect ended	Touch-screen tag value change	touch detected	Display list swap occurred

4.2 System Clock

4.2.1 Clock Source

The FT81x can be configured to use any of the three clock sources for system clock:

- Internal relaxation oscillator clock (default)
- External 12MHz crystal
- External 12MHz square wave clock

Figure 4-4, Figure 4-5 and Figure 4-6 show the pin connections for these clock options.

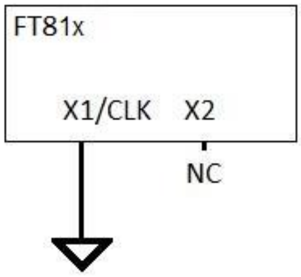


Figure 4-4 Internal relaxation oscillator connection

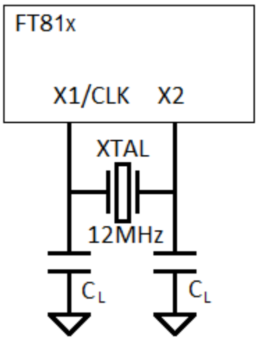


Figure 4-5 Crystal oscillator connection

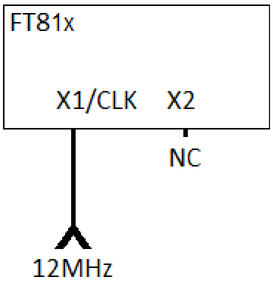


Figure 4-6 External clock input

4.2.2 Phase Locked Loop

The internal PLL takes an input clock from the oscillator, and generates clocks to all internal circuits, including the graphics engine, audio engine and touch engine.

4.2.3 Clock Enable

At power-on the FT81x enters sleep mode. The internal relaxation oscillator is selected for the PLL clock source. The system clock will be enabled when the following step is executed:

- Host sends an "ACTIVE" command

If the application chooses to use the external clock source (12MHz crystal or clock), the following steps shall be executed:

- Host sends a "CLKEXT" command
- Host sends an "ACTIVE" command

4.2.4 Clock Frequency

By default the system clock is 60MHz when the input clock is 12MHz. The host is allowed to switch the system clock to other frequencies (48MHz, 36MHz, 24MHz) by the host command "CLKSEL". The clock switching command shall be sent in SLEEP mode only.

When using the internal relaxation oscillator, its clock frequency is trimmed to be 12MHz at factory. Software is allowed to change the frequency to a lower value by programming the register REG_TRIM. Note that software shall not change the internal oscillator frequency to be higher than 12MHz.

4.3 Graphics Engine

4.3.1 Introduction

The graphics engine executes the display list once for every horizontal line. It executes the primitive objects in the display list and constructs the display line buffer. The horizontal pixel content in the line buffer is updated if the object is visible at the horizontal line.

Main features of the graphics engine are:

- The primitive objects supported by the graphics processor are: lines, points, rectangles, bitmaps (comprehensive set of formats), text display, plotting bar graph, edge strips, and line strips, etc.
- Operations such as stencil test, alpha blending and masking are useful for creating a rich set of effects such as shadows, transitions, reveals, fades and wipes.
- Anti-aliasing of the primitive objects (except bitmaps) gives a smoothing effect to the viewer.
- Bitmap transformations enable operations such as translate, scale and rotate.
- Display pixels are plotted with 1/16th pixel precision.
- Four levels of graphics states
- Tag buffer detection

The graphics engine also supports customized built-in widgets and functionalities such as jpeg decode, screen saver, calibration etc. The graphics engine interprets commands from the MPU host via a 4 Kbyte FIFO in the FT81x memory at RAM_CMD. The MPU/MCU writes commands into the FIFO, and the graphics engine reads and executes the commands. The MPU/MCU updates the register REG_CMD_WRITE to indicate that there are new commands in the FIFO, and the graphics engine updates REG_CMD_READ after commands have been executed.

Main features supported are:

- Drawing of widgets such as buttons, clock, keys, gauges, text displays, progress bars, sliders, toggle switches, dials, gradients, etc.
- JPEG and motion-JPEG decode
- Inflate functionality (zlib inflate is supported)
- Timed interrupt (generate an interrupt to the host processor after a specified number of milliseconds)
- In-built animated functionalities such as displaying logo, calibration, spinner, screen saver and sketch
- Snapshot feature to capture the current graphics display

For a complete list of graphics engine display commands and widgets refer to FT81x Series Programmer Guide, Chapter 4.

4.3.2 ROM and RAM Fonts

The FT81x has built in ROM character bitmaps as font metrics. The graphics engine can use these metrics when drawing text fonts. There are a total of 19 ROM fonts, numbered with font handle 16-34. The user can define and load customized font metrics into RAM_G, which can be used by display command with handle 0-15.

Each font metric block has a 148 byte font table which defines the parameters of the font and the pointer of font image. The font table format is shown in Table 4-7.

Table 4-7 Font table format

Address Offset	Size(byte)	Parameter Description
0	128	width of each font character, in pixels
128	4	font bitmap format, for example L1, L4 or L8
132	4	font line stride, in bytes
136	4	font width, in pixels
140	4	font height, in pixels
144	4	pointer to font image data in memory

The ROM fonts are stored in the memory space ROM_FONT. The ROM font table is also stored in the ROM. The starting address of the ROM font table for font index 16 is stored at ROM_FONT_ADDR, with other font tables following. The ROM font table and individual character width (in pixel) are listed in Table 4-8 through Table 4-10. Font index 16, 18 and 20-31 are for basic ASCII characters (code 0-127), while font index 17 and 19 are for Extended ASCII characters (code 128-255). The character width for font index 16 through 19 is fixed at 8 pixels for any of the ASCII characters.

Table 4-8 ROM font table

Font Index	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Font format	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L4	L4	L4	L4	L4	L4	L4	L4	L4
Line stride	1	1	1	1	2	2	2	3	3	4	7	8	9	11	14	18	23	30	39
Font width (max)	8	8	8	8	11	13	17	18	25	34	13	15	19	21	28	37	49	63	82
Font height	8	8	16	16	13	17	20	22	29	38	16	20	25	28	36	49	63	83	108
Image pointer start address (hex)	2FF7FC	2FFBFC	2FE7FC	2FEFFC	2FDAFC	2FCD3C	2FBD7C	2FA17C	2F7E3C	2F3D1C	2F181C	2ED61C	2E799C	2DFBBC	2D263C	2BAC3C	2945FC	251E1C	1E1B5C

Table 4-9 ROM font ASCII character width in pixels

Font Index =>			16/18	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
ASCII Character width in pixels	0	NULL	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	SOH	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	STX	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	ETX	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	EOT	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	ENQ	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	ACK	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	BEL	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	BS	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	HT	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	LF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	11	VT	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	FF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	13	CR	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	SO	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	SI	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	DLE	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	17	DC1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	18	DC2	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	19	DC3	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	20	DC4	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Font Index =>		16/18	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
21	NAK	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	SYN	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	ETB	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	CAN	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	EM	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	SUB	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	ESC	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	FS	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	GS	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	RS	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	US	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	space	8	3	4	5	5	6	9	3	4	5	6	8	10	13	18	23
33	!	8	3	4	5	6	6	9	3	4	6	6	9	11	15	19	25
34	"	8	4	5	6	5	8	12	5	6	7	8	12	15	19	25	33
35	#	8	6	8	9	10	14	19	10	11	14	15	19	26	33	44	57
36	$	8	6	8	9	10	13	18	8	10	11	15	18	25	31	41	54
37	%	8	9	12	14	16	22	29	11	13	16	17	23	31	40	52	68
38	&	8	8	10	11	13	17	22	9	11	14	15	19	26	34	44	57
39	'	8	2	3	3	3	6	6	3	4	4	5	7	10	11	15	20
40	(	8	4	5	6	6	8	11	5	6	7	9	11	15	18	24	31
41	)	8	4	5	6	6	8	11	5	6	8	8	10	14	18	24	31
42	*	8	4	7	6	7	10	13	7	8	10	11	14	18	24	31	40
43	+	8	6	9	10	10	14	19	9	10	12	14	17	24	30	41	52
44	,	8	3	3	4	5	6	9	3	4	4	5	7	9	12	16	20
45	-	8	4	4	5	6	8	11	6	7	10	11	15	18	24	32	41
46	.	8	3	3	4	5	6	9	3	4	6	7	8	11	14	19	24
47	/	8	3	4	5	5	7	9	6	7	9	10	13	17	22	29	38
48	0	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
49	1	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
50	2	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
51	3	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
52	4	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
53	5	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
54	6	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
55	7	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
56	8	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
57	9	8	6	8	9	10	13	18	8	10	12	14	17	24	30	40	52
58	:	8	3	3	4	5	6	9	3	4	6	6	7	10	13	18	23
59	;	8	3	4	4	5	6	9	3	4	6	6	8	10	14	18	23
60	<	8	6	8	10	10	15	19	8	9	11	12	16	21	28	36	46
61	=	8	5	9	10	11	15	19	8	9	13	14	18	23	30	40	52
62	>	8	6	8	10	10	15	19	8	9	11	13	16	22	29	37	48
63	?	8	6	8	9	10	12	18	7	9	10	12	15	20	26	34	44
64	@	8	11	13	17	18	25	34	13	15	19	21	28	37	49	63	82
65	A	8	7	9	11	13	17	22	9	11	13	15	20	27	34	45	58
66	B	8	7	9	11	13	17	22	9	10	14	15	19	27	34	45	58
67	C	8	8	10	12	14	18	24	9	11	13	15	20	26	34	45	58
68	D	8	8	10	12	14	18	24	9	11	14	17	22	28	36	48	63
69	E	8	7	9	11	13	16	22	7	9	12	13	16	23	29	39	50
70	F	8	6	8	10	12	14	20	7	9	12	13	17	22	29	39	50
71	G	8	8	11	13	15	19	25	9	11	14	16	22	28	37	48	62
72	H	8	8	10	12	14	18	24	9	11	15	17	23	29	37	50	65
73	I	8	3	4	4	6	8	9	4	5	6	7	9	12	15	20	26
74	J	8	5	7	8	10	13	16	8	9	12	13	17	23	30	40	50
75	K	8	7	9	11	13	18	22	9	11	14	16	19	26	34	45	58

Font Index =>			16/18	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
	76	L	8	6	8	9	11	14	18	7	9	12	13	17	22	29	39	51
	77	M	8	9	12	13	16	21	27	11	14	19	21	26	35	46	62	79
	78	N	8	8	10	12	14	18	24	9	11	15	17	23	29	37	50	65
	79	O	8	8	11	13	15	18	25	10	12	14	16	22	28	37	49	63
	80	P	8	7	9	11	13	16	22	9	10	14	15	19	26	34	45	58
	81	Q	8	8	11	13	15	18	26	10	12	14	17	22	29	38	50	64
	82	R	8	7	10	12	14	17	24	9	11	13	15	19	27	33	45	58
	83	S	8	7	9	11	13	16	22	9	11	12	14	20	26	33	43	56
	84	T	8	5	9	10	12	16	20	10	12	14	15	19	26	32	42	56
	85	U	8	8	10	12	14	18	24	9	11	13	17	21	28	37	48	62
	86	V	8	7	9	11	13	17	22	9	11	14	15	20	27	34	45	58
	87	W	8	9	13	15	18	22	31	12	15	18	21	27	36	46	61	79
	88	X	8	7	9	11	13	17	22	9	11	13	15	20	27	34	45	58
	89	Y	8	7	9	11	13	16	22	9	10	14	15	19	26	34	45	58
	90	Z	8	7	9	10	12	15	20	9	11	13	14	18	25	32	42	55
	91	[	8	3	4	5	5	7	9	4	5	6	7	9	12	15	19	25
	92	\	8	3	4	5	5	7	9	6	7	9	10	13	18	22	29	38
	93	]	8	3	4	5	5	7	9	4	5	7	7	9	12	15	19	25
	94	^	8	6	7	8	9	12	16	6	7	9	10	13	18	23	30	38
	95	_	8	6	8	9	11	14	18	8	10	11	13	16	21	26	34	43
	96	`	8	3	5	6	4	7	11	4	5	7	8	10	13	17	22	29
	97	a	8	5	8	9	11	13	18	8	9	11	13	17	23	30	39	50
	98	b	8	6	7	9	11	14	18	8	9	11	14	17	24	31	40	52
	99	c	8	5	7	8	10	12	16	8	9	11	12	16	22	28	37	48
	100	d	8	6	8	9	11	14	18	8	10	12	14	17	24	31	40	52
	101	e	8	5	8	9	10	13	18	8	9	11	12	16	22	29	37	48
	102	f	8	4	4	5	6	8	9	6	7	8	10	12	15	19	25	31
	103	g	8	6	8	9	11	14	18	8	10	11	14	18	24	31	41	52
	104	h	8	6	8	9	10	13	18	8	9	11	14	17	24	31	41	52
	105	i	8	2	3	3	4	6	7	3	4	6	6	7	10	13	18	23
	106	j	8	2	3	4	4	6	7	3	4	6	6	8	11	14	18	23
	107	k	8	5	7	8	9	12	16	7	9	11	13	16	22	28	36	47
	108	l	8	2	3	3	4	6	7	3	4	6	6	7	10	13	18	23
	109	m	8	8	11	14	16	20	27	11	15	18	21	27	36	47	63	80
	110	n	8	6	8	9	10	14	18	8	9	11	14	17	24	31	41	52
	111	o	8	6	8	9	11	13	18	8	10	12	13	17	24	31	40	52
	112	p	8	6	8	9	11	14	18	8	9	11	14	17	24	31	40	51
	113	q	8	6	8	9	11	14	18	8	10	12	13	17	24	31	40	52
	114	r	8	4	5	5	6	9	11	5	6	7	9	11	15	19	25	32
	115	s	8	5	7	8	9	12	16	7	9	11	12	17	22	29	38	48
	116	t	8	4	4	5	6	8	9	6	7	8	9	11	14	17	23	29
	117	u	8	5	7	9	10	14	18	8	9	12	14	17	24	31	41	52
	118	v	8	6	7	8	10	13	16	7	9	11	12	16	21	27	36	46
	119	w	8	8	10	12	14	18	23	11	13	16	18	23	32	41	54	70
	120	x	8	6	7	8	10	12	16	7	9	11	12	16	21	27	36	46
	121	y	8	5	7	8	10	13	16	7	9	11	12	16	21	27	36	46
	122	z	8	5	7	8	9	12	16	8	9	11	12	15	22	27	36	46
	123	{	8	3	5	6	6	8	11	5	6	8	8	11	15	18	24	31
	124	|	8	3	3	4	5	6	9	3	4	5	6	7	10	14	18	23
	125	}	8	3	5	6	6	8	11	5	6	7	9	10	15	18	24	31
	126	~	8	7	8	10	10	14	19	10	11	14	15	21	29	36	47	63
	127	DEL	8	0	0	0	0	0	0	3	4	5	6	5	10	13	18	23

Table 4-10 ROM font Extended ASCII characters