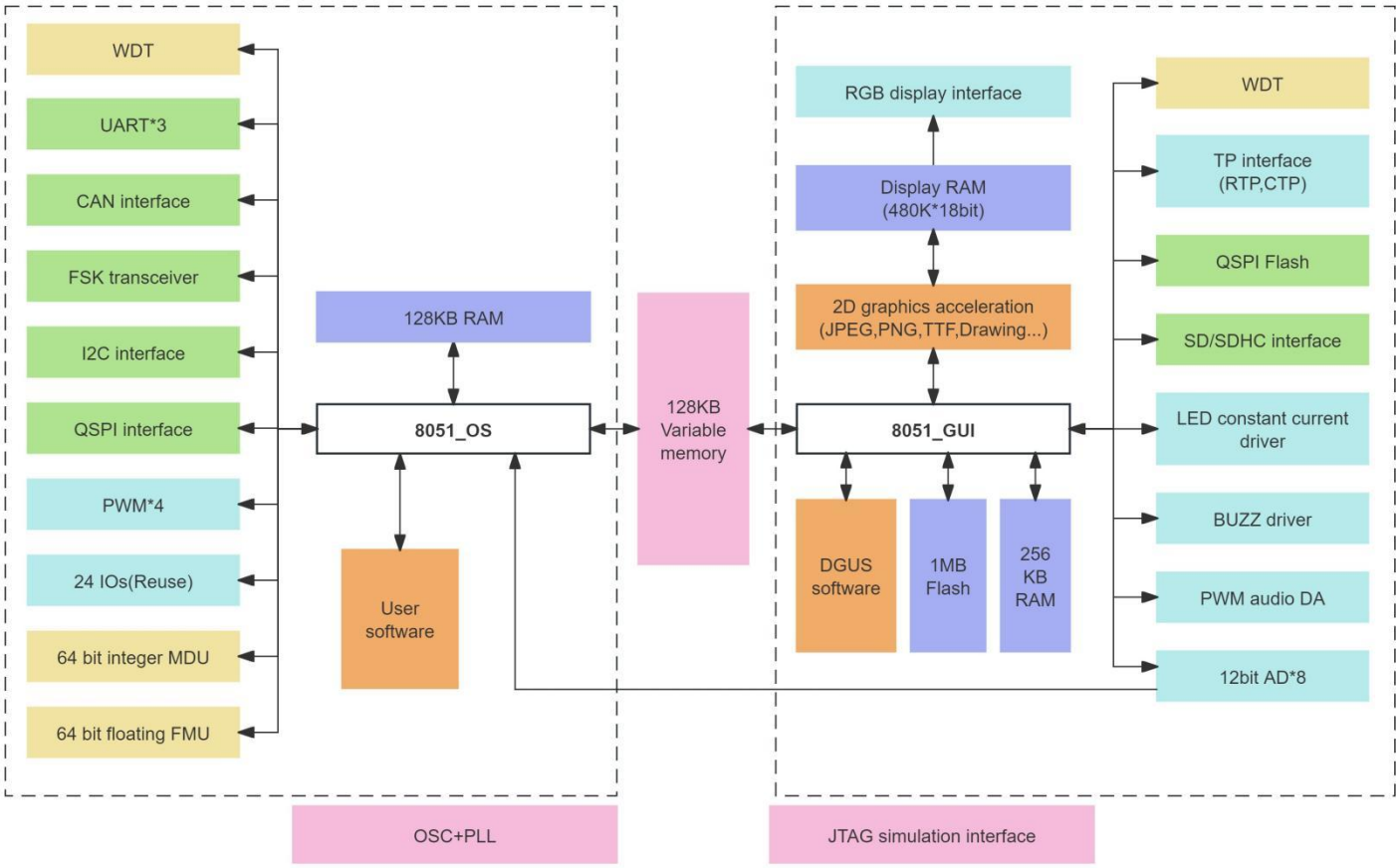
T5F0 ASIC Application Development Guide

Contents

1. Overview

T5F0 is a cost-effective dual-core ASIC designed for small size LCD screens.

1.1 IC Function Module Schematic



1.2 Performance Introduction

(1) Adopting the most mature and stable 8051 core,with a 1T single instruction cycle design,it can operate at 75MHz (low power) or 300MHz (high performance).

(2) A single CPU core (GUI CPU) runs the GUI system,with an extremely cool and smooth UI:

- Built in graphics memory,18bit (262K) colors display,resolution support up to 480*480 pixels (DGUS configuration) or 800 * 600 pixels(TA instruction set).
- 2D hardware acceleration,including drawing,JPEG,PNG icon decompression,vector font library text display.
- Supports resistive or capacitive touch screens,with a maximum touch tapping speed of 400Hz.
- 128K bytes of variable memory space,memory interface and OS CPU core exchange data,and UI control is extremely simple.
- 1 channel low-cost 250KHz PWM boost voltage type LED constant current drive.
- 1 channel 16 bit 32Ksps PWM digital DA output,achieving high signal-to-noise ratio sound quality restoration.
- High speed QSPI memory interface: supports a single 128Mbit SPI NOR or 1Gbit SPI NAND.
- SD card interface download and configuration engineering.

(3) Running user code on a separate CPU core (OS CPU),no user CPU required during use:

- 128K bytes of code and memory space (typical configuration of 64KB code,64KB XRAM memory).
- 64bit integer arithmetic unit (MDU),including 64bit MAC and 64bit divider.
- 64bit single precision,double precision floating-point arithmetic unit (FMU),including floating-point to integer conversion,floating-point addition,subtraction,multiplication,and division,and MAC.
- Built in software WDT,3-channel 16 bit Timers,4 channels of 16 bit resolution carrier frequency adjustable PWM.
- 24-channel IOs,3-channel UARTs,1-channel CAN,up to 6-channel 12bit A/D are available to users.
- 1-channel hardware QSPI interface,supporting DMA operation.
- 1-channel FSK transceiver interface enables easy high-speed,long-distance,and power carrier transmission.
- 17-channel interrupt signals support up to four levels of interrupt nesting,including:

  1-channel ADC interruption,1-channel FSK communication interruption,and 1-channel CAN communication interruption;

  2-channel external interrupts (EX0,EX1) can be configured for low-level or down edge triggering;

  3-channel timer interrupt: T0,T1,T2;

  4-channel PWM interrupts: PWM0,PWM1,PWM2,PWM3;

  5-channel UART interrupts: UART0 (Tx and Rx),UART2Rx,UART2Rx,UART3Tx,UART3Rx.
- Support IAP online simulation and debugging,with unlimited number of breakpoints; Code can be upgraded online through the DGUS system.

(4) 1Mbytes of on-chip Flash,patented encryption technology from DWIN,ensures code and data security,and eliminates counterfeiting and cloning.

(5) Working temperature range: -40 ℃ ~+85 ℃ (We can customize IC with the working temperature range of -55 ℃ -105 ℃).

(6) Low power consumption,strong anti-interference ability,can work stably on double-side designed PCB,and easily pass EMC/EMI testing.

(7) Using a 0.35mm spacing of 9×9mm QFN88 small package,the design and production are simple.

2. Hardware Design

2.1 IC Pin Definition

T5F0 ASIC adopts QFN88 packaging (9×9×0.9 mm,PIN spacing 0.35mm),and the pin arrangement is shown in Figure 2.1-1.

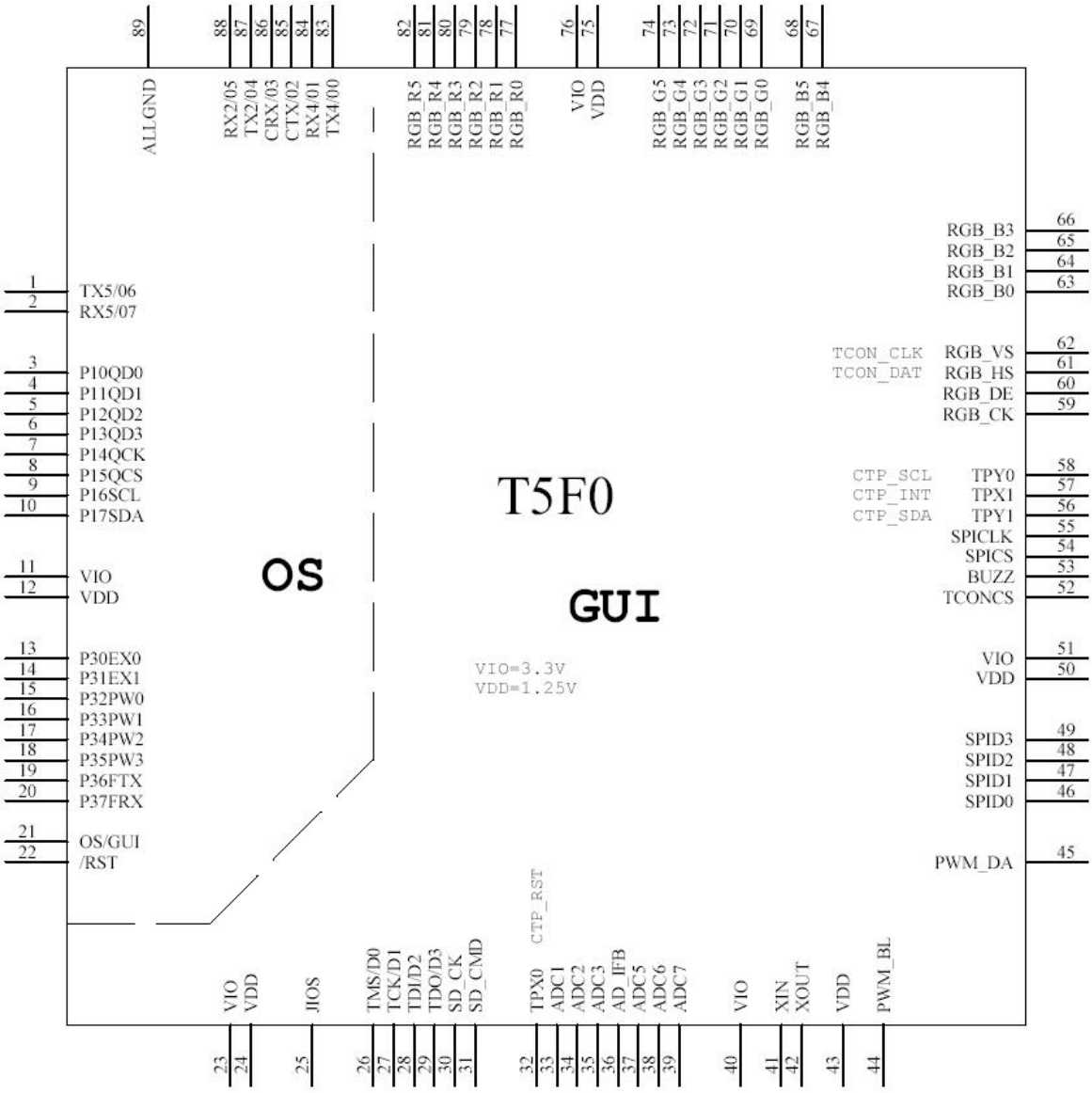


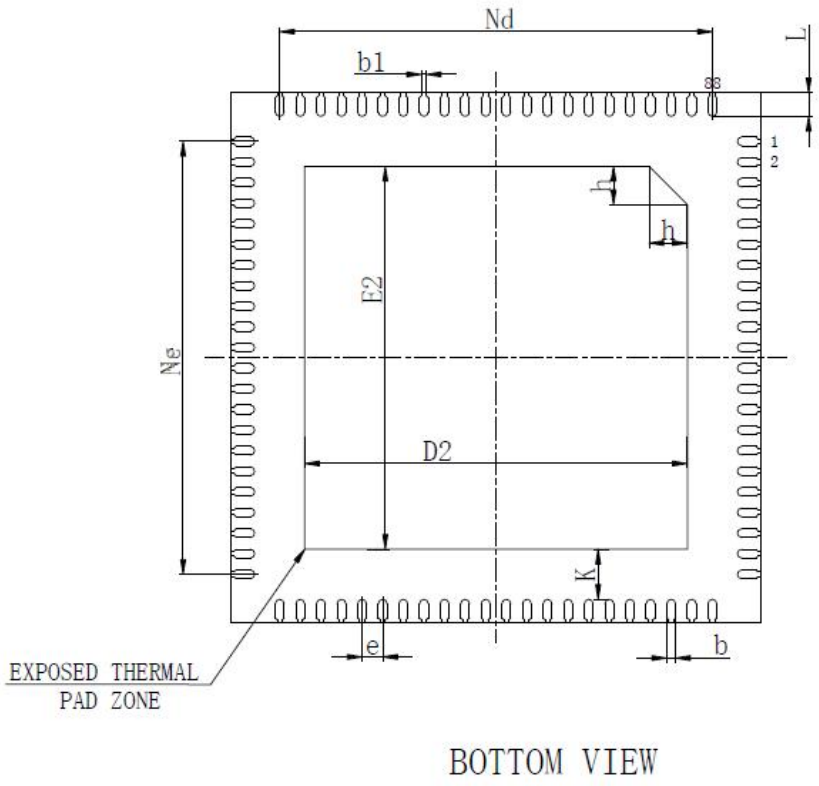
Figure 2.1-1 T5F0 pin arrangement

CPU	PIN	Function 1	Description	Function 2	Description
OS	83	TX4	UART4 output,connects an external 10K resistor to pull the level up to 3.3V	P0.0	I/O
OS	84	RX4	UART4 input,connects an external 10K resistor to pull the level up to 3.3V	P0.1	I/O
OS	85	CTX	CAN interface output	P0.2	I/O
OS	86	CRX	CAN interface input	P0.3	I/O
OS	87	TX2	UART2 output,connects an external 10K resistor to pull the level up to 3.3V	P0.4	I/O
OS	88	RX2	UART2 input,connects an external 10K resistor to pull the level up to 3.3V	P0.5	I/O
OS	1	TX5	UART5 output,connects an external 10K resistor to pull the level up to 3.3V	P0.6	I/O
OS	2	RX5	UART5 input,connects an external 10K resistor to pull the level up to 3.3V	P0.7	I/O
OS	3	QSPI_D0	4-wire QSPI data interface,supporting read and write of variable memory hardware DMA,and the main frequency can be configured. When using the QSPI interface, QSPI-CS requires an external 10K resistor to pull the level up to 3.3V	P1.0	I/O
OS	4	QSPI_D1		P1.1	I/O
OS	5	QSPI_D2		P1.2	I/O
OS	6	QSPI_D3		P1.3	I/O
OS	7	QSPI_CLK		P1.4	I/O
OS	8	QSPI_CS		P1.5	I/O
OS	9	RTC_SCL	I2C interface clock for external RTC	P1.6	I/O
OS	10	RTC_SDA	I2C interface data for external RTC	P1.7	I/O
OS	11	VIO	3.3V IO voltage		
OS	12	VDD	1.25V CPU core voltage		
OS	13	EX0	External interrupt 0 input	P3.0	I/O
OS	14	EX1	External interrupt 1 input	P3.1	I/O
OS	15	PWM0	PWM0 output	P3.2	I/O
OS	16	PWM1	PWM1 output	P3.3	I/O
OS	17	PWM2	PWM2 output	P3.4	I/O
OS	18	PWM3	PWM3 output	P3.5	I/O
OS	19	FTX	FSK bus data output	P3.6	I/O
OS	20	FRX	FSK bus data input	P3.7	I/O
System	21	OS/GUI	0=GUI JTAG,1=OS JTAG; Detect during reset	FTR	The FSK bus sends and receives control signals,and when using it,an external 10K pull-down resistor is required
System	22	RST	System reset input,and low level reset. T5F0 is required to connect a 470pF capacitor and a 10K resistor to GND		

GUI	23	VIO	3.3V IO voltage		
GUI	24	VDD	1.25V CPU core voltage		
GUI	25	JIOS	0: PIN26~PIN29 are JTAG interfaces; 1: PIN26~PIN29 are the data buses of the SD card interface		
GUI	26	TMS	JTAG interface,selecting whether to connect to GUI CPU or OS CPU based on the level of the OS/GUI pin during reset	SD_D0	SD/SDHC interface data bus with a maximum speed of 4MHz
GUI	27	TCK		SD_D1	
GUI	28	TDI		SD_D2	
GUI	29	TDO		SD_D3	
GUI	30	SD_CK	Connect a 22pF capacitor to ground near the SD card interface	PA_EN	Enabling signal of external amplifier for music playback
GUI	31	SD_CMD	Connect an external 10K resistor to pull the voltage level to 3.3V		
GUI	32	TPX0	4-wire RTP interface X0,externally connected to a 220K pull-down resistor	CTP_RST	CTP interface reset signal output
GUI	33	TPY0	4-wire RTP interface Y0,externally connected to a 220K pull-down resistor	ADC1	ADC1 input
GUI	34	TPX1	4-wire RTP interface X1,externally connected to a 220K pull-down resistor	ADC2	ADC2 input
GUI	35	TPY1	4-wire RTP interface Y1,externally connected to a 220K pull-down resistor	ADC3	ADC3 input
GUI	36	AD_IFB	The backlight current feedback voltage by LED backlight driver. 0-400mV corresponds to 0% -100% brightness of the backlight. Near T5F0,connect a 102 capacitor to GND		
GUI	37	ADC5	ADC5 input		
GUI	38	ADC6	ADC6 input		
GUI	39	ADC7	ADC7 input		
GUI	40	VIO	3.3V IO voltage,also serving as AD conversion reference voltage (Vref),ripple voltage should not exceed 50mV		
GUI	41	XIN	Externally connected 11.0592MHz crystal oscillator		
GUI	42	XOUT			
GUI	43	VDD	1.25V CPU core voltage		
GUI	44	PWM_BL	PWM control signal by LED backlight driver,with the frequency of 250KHz. It needs to be connected to an external 10K pull-down resistor		
GUI	45	PWM_DA	PWM DA signal for music playback,with the frequency of 32KHz,requires an external RC filtering network to convert the DA signal into analog signal and then connect it to the amplifier		
GUI	46	SPI_D0	QSPI bus for external connection to SPI Flash,with the bus frequency of 50MHz. It is recommended to connect magnetic beads in series to reduce EMI		
GUI	47	SPI_D1			
GUI	48	SPI_D2			
GUI	49	SPI_D3			
GUI	50	VDD	3.3V IO voltage		
GUI	51	VIO	1.25V CPU core voltage		

GUI	52	TCON_CS			
GUI	53	BUZZ	buzzer drive,needs to be connected to an external 10K pull-down resistor		
GUI	54	SPI_CS	QSPI bus chip selection signal,externally connected to SPI Flash,and it needs to be connected to a 10K external resistor to pull the voltage level to 3.3V		
GUI	55	SPI_CLK	QSPI bus clock signal,externally connected to SPI Flash.Recommended to connect magnetic beads in series to reduce EMI		
GUI	56	TPY1	4-wire RTP interface Y1	TCON_SDA	CTP interface data,when using it,connect an external 4.7K resistor to pull the level up to 3.3V
GUI	57	TPX1	4-wire RTP interface X1	TCON_INT	CTP interface interrupt signal
GUI	58	TPY0	4-wire RTP interface Y0	TCON_SCL	CTP interface clock
GUI	59	RGB_PCLK	RGB display interface,it is recommended to connect magnetic beads in series to reduce EMI. Taking the blue component signal as an example (red and green component signals can be analogized),explain the data cable connection: (1) For 18bit interface TFT screen TFT_B5 -- RGB_B5 TFT_B4 -- RGB_B4 TFT_B3 -- RGB_B3 TFT_B2 -- RGB_B2 TFT_B1 -- RGB_B1 TFT_B0 -- RGB_B0 (2) For 24bit interface TFT screen TFT_B7 -- RGB_B5 TFT_B6 -- RGB_B4 TFT_B5 -- RGB_B3 TFT_B4 -- RGB_B2 TFT_B3 -- RGB_B1 TFT_B2 -- RGB_B0 TFT_B1 -- RGB_B5 TFT_B0 -- RGB_B5		
GUI	60	RGB_DE			
GUI	61	RGB_HS		TCON_DAT	Data signal of TCON initialization interface
GUI	62	RGB_VS		TCON_CLK	Clock signal of TCON initialization interface
GUI	63	RGB_B0			
GUI	64	RGB_B1			
GUI	65	RGB_B2			
GUI	66	RGB_B3			
GUI	67	RGB_B4			
GUI	68	RGB_B5			
GUI	69	RGB_G0			
GUI	70	RGB_G1			
GUI	71	RGB_G2			
GUI	72	RGB_G3			
GUI	73	RGB_G4			
GUI	74	RGB_G5			
GUI	75	VDD	1.25V CPU core voltage		
GUI	76	VIO	3.3V IO voltage		
GUI	77	RGB_R0	RGB display interface,it is recommended to connect magnetic beads in series to reduce EMI		
GUI	78	RGB_R1			
GUI	79	RGB_R2			
GUI	80	RGB_R3			
GUI	81	RGB_R4			
GUI	82	RGB_R5			
System	-	GND	The grounding pad on the back must ensure reliable welding and good grounding		

2.2 Package Size



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.85	0.90	0.95
A1	0	0.02	0.05
b	0.10	0.15	0.20
b1	0.08REF		
c	0.203REF		
D	8.90	9.00	9.10
D2	6.40	6.50	6.60
e	0.35BSC		
Ne	7.35BSC		
Nd	7.35BSC		
E	8.90	9.00	9.10
E2	6.40	6.50	6.60
L	0.35	0.40	0.45
h	0.60	0.65	0.70
K	0.85REF		

For PCB design,please use the component packaging and reference design provided by DWIN Technology.

2.3 Basic Performance Parameters

Parameter	Unit	Min.	Typ.	Max.	Description
CPU core voltage	V	1.20	1.25	1.40	The total filtering capacitance should not be lower than 70uF,and the ripple voltage should not exceed 50mV
CPU core current	mA		150		High performance operating mode,dual core full speed operation
	mA		50		Low performance operating mode,dual core full speed operation
IO voltage (VIO)	V	1.8	3.3	3.6	5V TTL/CMOS level input,requires voltage divider or clamp protection
AD input voltage	V	-0.1		VIO+0.3	
AD input impedance	KΩ		1000		
AD sampling frequency	KSPS		16		When a higher sampling frequency is required,multiple channels can be uniformly spaced in parallel
IO high-level output amplitude (VOH)	V	3.0			VIO=3.3V,IO load current 8mA
IO low-level output amplitude (VOL)	V			0.3	VIO=3.3V,IO load current 8mA
IO high-level output current	mA	-10			VIO=3.3V,VOH=3V
IO low-level output current	mA	10			VIO=3.3V,VOL=0.3V
IO port flipping speed	MHz		100	150	
IO high-level identification voltage (VIH)	V	1.6			
IO low-level identification voltage (VIH)	V			0.6	
External crystal oscillator frequency	MHz	10.0	11.0592	12.0	The corresponding PLL frequency and CPU operating frequency can be converted proportionally
CPU operating frequency	MHz		74.6496		Low power operating mode
	MHZ		298.5984		High performance operating mode
PLL frequency	MHz		1194.394		PWM,FSK,UART4,and UART5 all operate with PLL clock frequency division
Operating temperature	℃	-40		+85	
Storage temperature	℃	-55		+105	
ESD protection capability	KV		2		HBM mode

2.4 Notices for Hardware Design

(1) When powered on,the CPU core voltage (1.25V) cannot be powered on later than the CPU IO voltage (3.3V). The total filtering capacitance of the CPU core power supply should not be less than 70uF (a minimum of 3* 22uF stacked capacitors with a voltage resistance of not less than 5V are required),and ensure that the CPU core voltage is not less than 1.2V before the CPU IO voltage (3.3V) fails (below 1.6V) during power off.

(2) The CPU core voltage must be stable,otherwise it may cause the CPU abnormal operation (below the minimum value will cause a crash,and exceeding the maximum value may damage the IC). It is recommended to connect one LC filtering circuit in series (recommended value: L=10uH,C=66uF,three 22uF capacitors in parallel) when powered by the CPU core to improve anti-interference performance.

(3) It is recommended to use low-level reset ICs such as SGM809S for reset processing,rather than simple RC reset circuits. Connecting a 10K resistor and a 470pF capacitor to the ground outside the reset pin can enhance its anti-interference ability.Each CPU core of T5F0 is equipped with a separate watchdog (WDT),which does not require an external WDT IC.

(4) It is recommended to use a four layer board application design to achieve excellent electromagnetic compatibility characteristics.When using a double-sided board application design,please connect a 470pF capacitor in parallel with a 104 capacitor filtering circuit at the IC power supply pin to reduce noise radiation.

(5) IO signal inputs exceeding VIO voltage of 0.3V or above must use voltage divider or clamp protection,otherwise it may cause signal abnormalities or damage to the IC.

(6) When all IO ports are configured for input,they are all floating inputs without internal pull-up or pull-down; All IO ports are in the input state during the reset process. When used as outputs,they can be externally pulled down or up to ensure a certain level during the reset process.

(7) The 4-bit bus speed of the external SPI Flash of T5F0 is 75MHz,and the wiring should be as close as possible. It is necessary to connect a 470pF capacitor in parallel with a 105 capacitor filtering circuit at the power pin of the SPI Flash.

3. Programming Instructions for OS CPU Function Module

The OS CPU of T5F0 adopts the 8051 core,which is widely used in industry,has the longest production time,and has been tested for a long time. On the basis of retaining the real-time performance,fast IO speed,and stable reliability of the 8051,DWIN has greatly improved the memory access and computing power of the 8051 by optimizing code processing,expanding the SFR bus,and adding hardware mathematical processors.

3.1 Initial Configuration

After powering on the OS CPU core,the Special Function Register (SFR) in the table below must be properly initialized.

SFR name	Address	Initialize configuration value	Description
CKCON	0x8E	0x00	CPU running in 1T command rate mode.
T2CON	0xC8	0x70	Configure the extended interrupt system and set timer T2 to run in Autoload mode.
DPC	0x93	0x00or 0x01	The change mode of DPTR after MOVX command operation. If using C51 development,it must be configured as 0x00. 0x00: unchanged. 0x01: DPTR=DPTR+1. 0x03: DPTR=DPTR-1.
PAGESEL	0x94	0x01-0x03, or configure according to application needs	The OS CPU has 128KB of RAM for running code and accessing stored data through MOVX. The 128KB RAM is divided into four 32KB pages,corresponding to page addresses 0x00-0x03. Page 0 can only be used for code storage,and its corresponding DPTR addresses are 0x0000-0x7FFF; Pages 1-3 can be switched between PAGESEL (code space) or D0PAGESEL (data space),corresponding to addresses 0x8000-0xFFFF for reading and writing.
D_PAGESEL	0x95	0x01-0x03, or configure according to application needs	
MUX_SEL	0xC9	0x40,or configure according to application needs	.7 1=CAN interface leads to P0.2 and P0.3; 0=CAN interface does not lead out,it is an IO port. .6 1=UART2 interface leads to P0.4 and P0.5; 0=UART2 interface does not lead out,it is an IO port. .5 Reserved,write 0. .4 1=FSK interface leads out to P3.6 and P3.7; 0=FSK interface does not lead out,it is an IO port. .3 Reserved,write 0. .2 Reserved,write 0. .1 WDT control,1=on; 0=off. .0 WDT service the dog,1=service the dog once (WDT count value reset to zero).
MUX_SEL1	0xE9	0x40,or configure according to application needs	.7 The interrupt flag of the completed ADC conversion,and it will be reset to zero by the CPU after the conversion is completed (ANL MUX_SEL1,# 7FH). .6 Undefined. .5 1=UART5 interface leads out to P0.6 and P0.7; 0=UART5 interface does not lead out,it is an IO port. .4 1=UART4 interface leads out to P0.0 and P0.1; 0=UART4 interface does not lead out,it is an IO port. .3 1=PWM3 interface leads out to P3.5; 0=PWM3 interface does not lead out,it is an IO port. .2 1=PWM2 interface leads out to P3.4; 0=PWM2 interface does not lead out,it is an IO port. .1 1=PWM1 interface leads out to P3.3; 0=PWM1 interface does not lead out,it is an IO port. .0 1=PWM0 interface leads out to P3.2; 0=PWM0 interface does not lead out,it is an IO port.

SFR name	Address	Initialize configuration value	Description
P0MDOUT	0xB7	Configure according to application needs	P0 port output configuration (IO input reading is always valid,if it is configured as output,the IO status can still be read). .7 0=P0.7 output off; 1=P0.7 output on (push-pull output). .6 0=P0.6 output off; 1=P0.6 output on (push-pull output). .5 0=P0.5 output off; 1=P0.5 output on (push-pull output). .4 0=P0.4 output off; 1=P0.4 output on (push-pull output). .3 0=P0.3 output off; 1=P0.3 output on (push-pull output). .2 0=P0.2 output off; 1=P0.2 output on (push-pull output). .1 0=P0.1 output off; 1=P0.1 output on (push-pull output). .0 0=P0.0 output off; 1=P0.0 output on (push-pull output).
P1MDOUT	0xBC	Configure according to application needs	P1 port output configuration (IO input reading is always valid,if it is configured as output,the IO status can still be read).
P3MDOUT	0xBE	Configure according to application needs	P2 port output configuration (IO input reading is always valid,if it is configured as output,the IO status can still be read).
PORTDRV	0xF9	0x01,or configure according to application needs	Driver capability configuration for IO port output mode,effective from a minimum of 3 bits: 0x00=4mA 0x01=8mA 0x02=16mA 0x03=32mA 0x04=48mA 0x05=64mA 0x06=80mA 0x07=100mA.
RAMMODE	0xF8	0x00	Variable memory access interface control.

3.2 Memory

The 8051 core of the OS CPU can access six different types of memory,as shown in the table below.

Memory type	Size	Address	Access method
Code memory	128KBytes	0x0000-0xFFFF	PAGESEL: Switch to code page space,and it can only be read with MOVC command.
Data register	256Bytes	0x00-0xFF	Same as the standard 8051.
SFR register	128Bytes	0x80-0xFF	Same as the standard 8051,DWIN can provide users with SFR definition files(.INC or .H header files).
Extended SFR Register	384Bytes	0x000-0x17F	Use the EXADR_H,EXADR_L,and EXDATA register interfaces to access
Data storage	96KBytes	0x8000-0xFFFF	D_PAGESEL: Switch to data page space,and use MOVX command to access.
Variable memory	128KBytes	0x00:0000-0x00:7FFF	Use variable memory interface to access

3.2.1 Code Memory (64KBytes)

The functional divisions and definitions of code memory space are shown in the table below.

Address	Definition	Description
0x0000	Reset_PC	The entry address of the program after resetting
0x0003	EX0_ISR_PC	External interrupt 0 program interface
0x000B	T0_ISR_PC	Timer0 interrupt program interface
0x0013	EX1_ISR_PC	External interrupt 1 program interface
0x001B	T1_ISR_PC	Timer1 interrupt program interface
0x0023	UART2_ISR_PC	UART2 TX/RX interrupt program interface
0x002B	T2_ISR_PC	Timer2 interrupt program interface
0x0043	FSK_ISR_PC	Interrupt program interface for FSK data frame input and output
0x004B	CAN_ISR_PC	CAN interrupt program interface
0x0053	UART4_TX_ISR_PC	UART4 TX interrupt program interface
0x005B	UART4_RX_ISR_PC	UART4 RX interrupt program interface
0x0063	UART5_TX_ISR_PC	UART5 TX interrupt program interface
0x006B	UART5_RX_ISR_PC	UART5 RX interrupt program interface
0x008B	PWM0_ISR_PC	PWM0 interrupt program interface
0x0093	PWM1_ISR_PC	PWM1 interrupt program interface
0x009B	PWM2_ISR_PC	PWM2 interrupt program interface
0x00A3	PWM3_ISR_PC	PWM3 interrupt program interface
0x00AB	AD_ISR_PC	AD data conversion interrupt program interface
0x00F8	JTAG interface enabled	0xFFFF enables connection to the JTAG interface for simulation debugging; Other values will be disabled.
0x00FA	“DWINT5”	Code recognition,illegal values will cause the OS CPU to stop running.
0x0100	Application code start	

- OS CPU codes are saved in 1Mbytes of on-chip Flash,and after power on reset,they are loaded into RAM by the system hardware for operation.
- OS CPU codes can only be written to on-chip Flash through the GUI CPU (SD card interface or variable memory interface).

3.2.2 Variable Memory (128KBytes)

Variable memory is a dual port RAM used for high-speed data exchange between two CPU cores(OS core and GUI core),with the addresses range of 0x000000-0x007FFF,and each address corresponds to 4 bytes of data.

The SFR register interface in the following table can be used to access variable memory:

SFR name	Address	Description
RAMMODE	0xF8	Variable memory access interface control,capable of bit addressing: .7 Write 1 to request occupation of variable memory for reading and writing,and it must be reset to zero when not occupying variable memory. .6 APP-EN Write 1 to start reading and writing once,and it will be reset to zero after hardware execution. .5 APP-RW 1=Read variable memory; 0=Write variable memory .4 APP-ACK Hardware response to request for 8051 to occupy variable memory: 1=OK; 0=BUSY,need to continue waiting. .3-.0 corresponds to DATA3: DATA0. 1=Write the corresponding byte of DATA,0=Do not write the corresponding byte of DATA.
ADR_H	0xF1	The high 8-bit address of the variable memory,A23: A16,is always 0x00.
ADR_M	0xF2	The middle 8-bit address of the variable memory,A15: A8
ADR_L	0xF3	The low 8-bit address of the variable memory,A7: A0
ADR_INC	0xF4	The automatic increment of address after reading and writing variable memory, Before reading/writing ADR_H:M:L=After reading/writing ADR_H:M:L+ADR_INC
DATA3	0xFA	Variable data interface,select corresponding RAMMODE.3 when writing data
DATA2	0xFB	Variable data interface,select corresponding RAMMODE.2 when writing data
DATA1	0xFC	Variable data interface,select corresponding RAMMODE.1 when writing data
DATA0	0xFD	Variable data interface,select corresponding RAMMODE.0 when writing data

The variable memory must be read and written according to the following process (if the variable memory is applied to an interrupt,the interrupt must be closed when the main program reads and writes,and it cannot be nested):

(1) Configure the address and address increment;

(2) Set RAMCODE=0x8F (write) or 0xAF (read),check RAMCODE.4=1 to confirm obtaining read and write control;

(3) Read and write data,set RAMMODE=0x00 after reading and writing.

Application example:

Read and write 2*double words to address 0x0800 (corresponding to DGUS variable memory address 0x1000).

```
        MOV     ADR_H,#00H              ;Configure the variable memory address
        MOV     ADR_M,#08H
        MOV     ADR_L,#00H
        MOV     ADR_INC,#01H            ;Configure the address increment
        MOV     RAMMODE,#0AFH           ;Initiate reading mode
        JNB     APP_ACK,$               ;Waiting for confirmation
        MOV     R0,#TEST_BUF            ;Reading
        MOV     R1,#2
RDVP:   SETB    APP_EN                  ;Initiate reading data once
        JB      APP_EN,$
        MOV     @R0,DATA3
        INC     R0
        MOV     @R0,DATA2
        INC     R0
        MOV     @R0,DATA1
        INC     R0
        MOV     @R0,DATA0
        INC     R0
        DJNZ    R1,RDVP
        CLR     APP_RW                  ;Writing mode
        MOV     ADR_L,#00H              ;Adjust the address to 0x08:0000
        MOV     R0,#TEST_BUF
        MOV     R1,#2
WRVP:   MOV     DATA3,@R0
        INC     R0
        MOV     DATA2,@R0
        INC     R0
        MOV     DATA1,@R0
        INC     R0
        MOV     DATA0,@R0
        SETB    APP_EN                  ;Initiate reading data once
        JB      APP_EN,$
        INC     R0
        DJNZ    R1,WRVP
        MOV     RAMMODE,#00H            ;Variable memory reading and writing completed
```

3.2.3 Data Storage (Up to 96KBytes)

The OS CPU of T5F0 has a maximum of three 32KBytes of RAM page blocks as the data storage,which are switched using the D_PAGESEL register (0x01-0x03). After switching,the corresponding address is 0x8000-0xFFFF,and the relevant interface SFR are shown in the table below.

SFR name	Address	Description
D_PAGESEL	0x95	Page block selection register,0x01-0x03. If only 32K bytes of data storage is used,it is recommended to configure it to 0x03.
DPC	0x93	After using the MOVX instruction of DPTR,the change modes of DPTR are as follows: DPC=0x00: DPTR remains unchanged after MOVX instruction operation. If using C51 development,it must be configured as 0x00. DPC=0x01: After the MOVX instruction operation,DPTR=DPTR+1. DPC=0x03: After the MOVX instruction operation,DPTR=DPTR-1.
DPH	0x83	DPTR data pointer.
DPL	0x82	

The address of 0x0000-0x7FFF prohibits the use of MOVX instructions for reading and writing,which may cause abnormal code execution.

The MOVX instruction of T5F0 has 3 instruction cycles,and DPC can configure DPTR automatic increment (or decrement) mode,which makes the speed of T5F0 reading and writing data storage much faster than the standard 8051,especially for reverse order storage applications.

Application example:

```
MOV     DPC, #01H           ;DPTR++
MOV     DPTR, #8000H
MOVX    A, @DPTR            ;A=@8000
MOVX    A, @DPTR            ;A=@8001,After reading,DPTR=8002
```

3.2.4 Extended SFR Register

Extended SFR register using EXADR_H,EXADR_L,and EXDATA register interfaces for reading and writing.

SFR name	Address	Description	
EXADR_H	0xF5	High byte of extended SFR address	After each reading and writing,the address will automatically add 1 to point to the next SFR
EXADR_L	0xFE	Low byte of extended SFR address	
EXDATA	0xFF	Data interface of extended SFR address	

If you use the extended SFR register in an interrupt,the main program must close the interrupt when reading or writing the extended SFR register and cannot nest it.

The definitions of the extended SFR register are as follows:

Category	Address	length	Definition	Description
MDU FMU	0x000	1	A7	Register set for MDU and FMU operation acceleration.7 is the highest byte and 0 is the lowest byte
	0x001-0x007	6	A6-A0	
	0x008	8	B7:B0	
	0x010	8	C7:C0	
	0x07F	1	FMU_CN	Floating-point operation control register: .7 1=Initiate a floating-point operation.After completing the operation,reset to zero by hardware. .6 Rounding off control when converting floating-point to integer. 0=not rounding off; 1=rounding off. .5 Floating-point type. 0=single precision; 1=double precision. .4 Floating point calculation error feedback, 0=normal; 1=error .3-.0 Floating-point operation modes: 0=Single precision floating A (using A0-A3),output as integer B (8 bytes),controlled by.6 for rounding off; 1=Integer A (8 bytes),output as single precision floating B (using A0-A3); 2=Floating-point addition,C=A+B,all using A0-A3,B0-B3,C0-C3; 3=Floating-point subtraction,C=A-B; 4=Floating point multiplication,C=A * B; 5=Floating-point division,A=C/A; 6=Floating point MAC,C=C+A * B; 7-15 is not defined.
ADC	0x018	16	AD0-AD7	8-channel AD value,2 bytes per channel.
QSPI	0x028	1	QSPI_CLK_DIV	QSPI bus clock configuration,QSPI clock frequency=CPU operating frequency/QSPI_CLK_DIV
	0x029	1	QSPI_CLK_PHS	.7 Read phase configuration: 1=Read data after rising edge; 0=Read data after falling edge. .6 Write phase configuration: 1=high-level rewrite data; 0=low-level rewrite data. .5-.0 Write 0.
	0x02A	3	QSPI_DMA_LEN	The length of read-write data bytes,with a maximum of 128K bytes
	0x02D	3	QSPI_DMA_APP_ADR	Variable memory address pointer for reading and writing data

Category	Address	length	Definition	Description
WDT	0x030	4	WDT_Time_Set	WDT count setting,which based on the CPU clock. Reset T5F0 when WDT count exceeds the set value
PWM	0x034	1	PWM0_CLK_DIV	PWM0 operating frequency =PLL frequency (1194.394MHz) / PWM0_CLK_DIV
	0x035	2	PWM0_TH:L	PWM0 count upper limit setting,PWM0 carrier frequency=PWM0 operating frequency/(PWM0-TH: L+1)
	0x037	2	PWM0_H:L	High level width value of PWM0 output
	0x039	5	PWM1 configuration value	
	0x03E	5	PWM2 configuration value	
	0x043	5	PWM3 configuration value	
FSK	0x048	1	FSK_DIV	FSK transceiver operating frequency=PLL_CLK/FSK_DIV
	0x049	1	N_start	When sending FSK,the number of starting pulses B0, 0x01-0xFF
	0x04A	1	N_sync	Number of synchronization pulses B2,0x01-0x08
	0x04B	2	TB0	B0 pulse cycle number (corresponding to FSK transceiver operating frequency),>=6
	0x04D	2	E0	B0 pulse decoding tolerance,which means that all pulses within the TB0+/- E0 cycle range are B0
	0x04F	2	TB1	Number of B1 pulse cycles,>=6; TB1>TB0
	0x051	2	E1	B1 pulse decoding tolerance,which means that all pulses within the TB1+/- E1 cycle range are B1
	0x053	2	TB2	Number of B2 pulse cycles,>=6; TB2>TB1
	0x055	2	E2	B2 pulse decoding tolerance,which means that all pulses within the TB2+/- E2 cycle range are B2
	0x080	128	FSK_TX_BUFFER	The first byte is the data length,and a maximum of 127 bytes of data can be sent per frame
	0x100	128	FSK_RX_BUFFER	The first byte is the data length
CAN	0x057	1	BRP	CAN Baud rate divider register
	0x058	1	BTR0	Bit timing register 0,.7-.4 is the synchronization jump width,and .3-.0 is the time period 2 (0-7)
	0x059	1	BTR1	Bit timing register 1,.4-.0 is time period 1 (0x00-0x0F)
	0x05A	4	ACR3:0	Acceptance code register
	0x05E	4	AMR3:0	Acceptance code register
	0x062	1	RXERR	Receive error count register
	0x063	1	TXERR	Send error count register
	0x064	1	TX_BUFFER_SET	.7=IDE .6=RTR .3-.0=DLC (frame data length)
	0x065	4	TX_ID	Extended frame ID.31: ID.3 29bit valid Standard frame ID.31: ID.21 11bit valid
	0x069	8	TX_DATA	Sending data frame,D1-D8
	0x064	1	RX_BUFFER_SET	.7=IDE .6=RTR .3-.0=DLC (frame data length)
	0x065	4	RX_ID	Extended frame ID.31: ID.3 29bit valid Standard frame ID.31: ID.21 11bit valid
	0x069	8	RX_DATA	Sending data frame,D1-D8

3.3 Integer and Floating-point Operating Unit (MDU,FMU)

Due to the limited computing power of 8051,T5F0 has extended the hardware integer operation unit (MDU) and floating-point operation unit (FMU) to enhance computing power. The longest execution time for division instructions in mathematical operation units is 64T (usually 8T),and the rest of the instructions do not exceed 2T.The relevant control SFR is defined in the table below:

SFR name	Address	Description
MAC_CN	0xE5	MAC hardware integer multiply add accumulator control register,defined as follows: .7 MAC enable,write 1 to perform a calculation,and after hardware execution,it will be reset to zero. .6 MAC mode,1 is multiply add accumulator mode: C=A * B+C, 0 is multiplier mode: C=A * B. .5 Write 0. .4 1=64bit MAC,0=32bit MAC (A3:0/B3:0/C7:0,it's worth noting that C is still 64bit). .3 1=signed number,0=unsigned number. .2-.0 Write 0. A,B,C are the register groups for extended SFR registers.
DIV_CN	0xE6	DIV hardware integer divider control register (division C/A,quotient A,remainder B),defined as follows: .7 DIV enable,write 1 to perform a calculation,and after the hardware execution is completed,it will be reset to zero. .6 DIV mode,1: rounding 0: not rounding. .5-.4 undefined,write 0. .3 1=signed number,0=unsigned number. .2-.0 Write 0. A,B,C are the register groups for extended SFR registers.
FMU_CN	0x07F (Extended SFR)	Hardware floating-point operation control register: .7 1=Initiate a floating-point operation and after hardware execution,it will be reset to 0. .6 When converting floating-point to integer,perform rounding off, 0=not rounding off;1=rounding off. .5 Floating-point type,0=single precision,1=double precision. .4 Floating point calculation error feedback,0=normal,1=error .3-.0 Floating-point operation modes: 0=Floating-point to integer conversion,input single precision A (using A0-A3),output integer B (8 bytes),.6 control rounding off; 1=Integer convert to floating-point,input integer A (8 bytes),output single precision B (using A0-A3); 2=Floating-point addition,C=A+B,all using A0-A3,B0-B3,C0-C3; 3=Floating-point subtraction,C=A-B; 4=Floating point multiplication,C=A * B; 5=Floating-point division,C/A,quotient A; 6=Floating point MAC,C=C+A * B; 7-15 Undefined.

Application example: 32-bit integer MAC calculation,0x1234*0x5678-0x2000.

```
        MOV     EXADR_H,#00H
        MOV     EXADR_L,#04H    ;Write A3:A0=0x 00 00 12 34
        MOV     EXDATA,#00H
        MOV     EXDATA,#00H
        MOV     EXDATA,#12H
        MOV     EXDATA,#34H
        MOV     EXADR_L,#0CH    ;Write B3:B0=0x 00 00 56 78
        MOV     EXDATA,#00H
        MOV     EXDATA,#00H
        MOV     EXDATA,#56H
        MOV     EXDATA,#78H
        MOV     EXADR_L,#10H    ;Write C7:C0=0xFF FF FF FF FF FF E0 00 (-0x2000)
        MOV     EXDATA,#0FFH
        MOV     EXDATA,#0FFH
        MOV     EXDATA,#0FFH
        MOV     EXDATA,#0FFH
        MOV     EXDATA,#0FFH
        MOV     EXDATA,#0FFH
        MOV     EXDATA,#0E0H
        MOV     EXDATA,#00H
        MOV     MAC_CN,#0C8H    ;32bit integer MAC
WTMAC:  MOV     A,MAC_CN
        JB      ACC.7,WTMAC
        MOV     EXADR_L,#10H    ;Read result 00 00 00 00 06 25 E0 60
        MOV     R7,EXDATA
        MOV     R6,EXDATA
        MOV     R5,EXDATA
        MOV     R4,EXDATA
        MOV     R3,EXDATA
        MOV     R2,EXDATA
        MOV     R1,EXDATA
        MOV     R0,EXDATA
```

3.4 Timer

The OS CPU of T5F0 has three timers T0/T1/T2,where T0 and T1 are consistent with the standard 8051,and T2 can only operate in 16bit Autoload mode. The clocks of T0 and T1 are both 1/12 of the CPU frequency,while the clock of T2 can be configured as 1/12 or 1/24 of the CPU frequency.

The SFR related to timers is shown in the table below:

SFR name	Address	Description
TCON	0x88	T0 and T1 control registers,same as standard 8051,can be bit addressed. .7=TF1; .6=TR; .5=TF0; .4=TR0; .3=IE1; .2=IT1; .1=IE0; .0=IT0 IT1 and IT0 are external interrupt triggering mode options: 0=low-level triggering,1=low-edge triggering.
TMOD	0x89	T0,T1 mode selection,same as standard 8051
TH0	0x8C	
TL0	0x8A	
TH1	0x8D	
TL1	0x8B	
T2CON	0xC8	T2 control register,can be bit addressed. .7: Clock division selection,0=CPU main frequency/12,1=CPU main frequency/24. .6-.4: Must write 1. .3-.1: Must write 0. .0: TR2,1=T2 running,0=T2 closing.
TH2	0xCD	T2 running value,automatically loaded when the add count overflows each time. TH2=CRCH,TL2=CRCL.
TL2	0xCC	
TRL2H	0xCB	T2 reinstallation load value=65536-T2 timer interval (uS) * T2 clock frequency (MHz).
TRL2L	0xCA	

The relevant settings for timer interrupts are as follows:

Interrupt type	Program entry address	Trigger marker	Interrupt enable control	Remark
T0 interrupt	0x000B	TF0（TCON.5)	IEN0.1	Automatically clear TF0 during interrupt response
T1 interrupt	0x001B	TF1（TCON.7)	IEN0.3	Automatically clear TF0 during interrupt response
T2 interrupt	0x002B	TF2（IRCON.6)	IEN0.5	After the interrupt response,it is necessary to clear the TF2 using software,otherwise the interrupt will be triggered again

Application example: CPU main frequency 74.6496MHz,set T2 1mS interrupt,output 500Hz square wave at P1.0.

```
        ORG     002BH           ;T2 interrupt program entry
        LJMP    T2INT

T2INT:  CLR     TF2             ;T2 interrupt program
        CPL     P1.0
        RETI

;Initialize the relevant SFR for T2
MOV  CRCH,#HIGH(59315)          ;1mS timer
MOV  CRCL,#LOW(59315)
MOV  T2CON,#71H                 ;TR2=1
ORL  IEN0,#20H                  ;ET2=1,enable T2 interrupt
```

3.5 Watchdog Timer (WDT)

The relevant SFR for watchdog timer control is shown in the table below:

SFR name	Address	Description
MUX_SEL	0xC9	.1 WDT control,1=on,0=off. .0 WDT feeding the dog,1=feeding the dog once (WDT count value reset to zero).

The WDT count overflow time is saved at the 0x030 extended SFR address,and the WDT reset time=set value/CPU main frequency (Hz),which is measured in seconds.

The relevant reference codes for WDT operation are as follows:

```
ANL   MUX_SEL,#0FDH        ;WDT must be turned off before setting the overflow time
MOV   EXADR_H,#00H         ;WDT reset time with the main frequency configuration of 74.6496MHz :
                           1 second=74649600 （0x04 73 10 00)
MOV   EXADR_L,#30H
MOV   EXDATA,#04H
MOV   EXDATA,#73H
MOV   EXDATA,#10H
MOV   EXDATA,#00H
ORL   MUX_SEL,#02H         ;Enable WDT; Once enabled,it is necessary to feed the dog before WDT overflow to avoid
                           system reset.

ORL   MUX_SEL,#01H         ;WDT reset (feeding dog)
```

3.6 Pulse Width Modulation Module (PWM)

SFR registers related to PWM control:

SFR name	Address	Description
MUX_SEL1	0xE9	.3 1=PWM3 interface leads out to P3.5, 0=PWM3 interface does not lead out,it is an IO port. .2 1=PWM2 interface leads out to P3.4, 0=PWM2 interface does not lead out,it is an IO port. .1 1=PWM1 interface leads to P3.3,0=PWM1 interface does not lead out,it is an IO port. .0 1=PWM0 interface leads out to P3.2, 0=PWM0 interface does not lead out,it is an IO port.
PWM0_CN	0xEB	.7 W,PWM0_EN,1=PWM0 on,0=PWM0 off. .6 R/W,PWM0_IF,PWM0 interrupt flag, after the hardware is set,it will be reset to zero by the CPU. .5 R/W,PWM0 configuration (0x034-0x036 extended SFR) update once ,and reset it to zero after hardware is set. .4-.0 Undefined
PWM1_CN	0xEC	The definition is the same as PWM0_CN
PWM2_CN	0xED	The definition is the same as PWM0_CN
PWM3_CN	0xEE	The definition is the same as PWM0_CN
P3MDOUT	0xBE	P3 port output configuration (IO input reading is always valid.When it is configured as output,IO status can still be read). .5 0=P3.5 output off,1=P3.5 output on (push-pull output). .4 0=P3.4 output off,1=P3.4 output on (push-pull output). .3 0=P3.3 output off,1=P3.3 output on (push-pull output). .2 0=P3.2 output off,1=P3.2 output on (push-pull output).

Extended SFR registers related to PWM control:

Address	Size	Definition	Description
0x034	1	PWM0_CLK_DIV	PWM0 operating frequency= PLL frequency (1194.394MHz)/PWM0_CLK_DIV
0x035	2	PWM0_TH:L	PWM0 count upper limit setting,PWM0 carrier frequency= PWM0 operating frequency/(PWM0_TH: L+1)
0x037	2	PWM0_H:L	High level width value of PWM0 output
0x039	5	PWM1 configuration	
0x03E	5	PWM2 configuration	
0x043	5	PWM3 configuration	

Related settings for PWM interrupts:

interrupt type	Program entry address	Trigger marker	Interrupt enable control	Note
PWM0 interrupt	0x008B	PWM0_CN.6	IEN2.1	After the interrupt response,software needs to be used to reset PWM0_CN. 6
PWM1 interrupt	0x0093	PWM1_CN.6	IEN2.2	After the interrupt response,software needs to be used to reset PWM1_CN. 6
PWM2 interrupt	0x009B	PWM2_CN.6	IEN2.3	After the interrupt response,software needs to be used to reset PWM2_CN. 6
PWM3 interrupt	0x00A3	PWM3_CN.6	IEN2.4	After the interrupt response,software needs to be used to reset PWM3_CN. 6

Application example: Configure PWM0 as the frequency of about 8bit 300KHz,and adjust the PWM output duty cycle during PWM interruption.

```
; PWM0 initialization configuration
MOV     EXADR_H, #00H
MOV     EXADR_L, #34H
MOV     EXDATA, #15         ;PWM0 frequency=1194.394MHz/(15*256)=311KHz
MOV     EXDATA, #00H        ;Count upper limit
MOV     EXDATA, #255
MOV     EXDATA, #00H        ;The initialization configuration for outputting high levels is set to 0%
MOV     EXDATA, #00H
MOV     PWM0_CN, #0A0H      ;Configure PWM0 once and turn it on
ORL     P3MDOUT, #04H       ;P3.2 output
ORL     MUX_SEL1, #01H      ;Configure P3.2 as PWM0 output
ORL     IEN2, #02H          ;Enable PWM0 interrupt

;PWM0 Interrupt program (if real-time adjustment of output frequency is required,use the yellow background code above to adjust in the interrupt program)
MOV     EXADR_H, #00H       ;New value for PWM high-level width configuration
MOV     EXADR_L, #37H       ;When the interrupt program uses the extended SFR,the interrupt must be turned off when the main program uses the extended SFR.
MOV     EXDATA, #00H
MOV     EXDATA, PWM0_DUTY
ANL     PWM0_CN, #0BFH      ;Clear PWM0 interrupt flag
```

3.7 IO Port

The OS CPU of T5F0 has three 8-bit parallel ports (P0,P1,P3),totaling 24 IO ports.

All inputs to the IO ports are always on,with floating inputs and no internal pull-up or pull-down.

When using the IO port as an output,it is necessary to turn on the output control,and the output drive capability can also be configured. Due to the power on reset process,the IO port is in input mode. When used as an output,it must be externally pulled up or down to ensure a reliable output during the power on process,and will not malfunction due to IO floating.

P3.0 is the external interrupt 0,P3.1 is the input of external interrupt 1,which can be configured as low-level trigger (0) or down edge trigger (1) through IT0 and IT1. The subsequent use of IO is consistent with the standard 8051,except for the need to control the switch for output,output strength,and external device reuse after power initialization configuration.

The SFR related to IO is shown in the table below.

SFR name	Address	Description
P0	0x80	Can be bit addressed,same as standard 8051.
P1	0x90	Can be bit addressed,same as standard 8051.
P3	0xB0	Can be bit addressed,same as standard 8051.
P0MDOUT	0xB7	P0 port output configuration. .7 0=P0.7 output off 1=P0.7 output on (push-pull output). .6 0=P0.6 output off 1=P0.6 output on (push-pull output). .5 0=P0.5 output off 1=P0.5 output on (push-pull output). .4 0=P0.4 output off 1=P0.4 output on (push-pull output). .3 0=P0.3 output off 1=P0.3 output on (push-pull output). .2 0=P0.2 output off 1=P0.2 output on (push-pull output). .1 0=P0.1 output off 1=P0.1 output on (push-pull output). .0 0=P0.0 output off 1=P0.0 output on (push-pull output).
P1MDOUT	0xBC	P1 port output configuration
P3MDOUT	0xBE	P3 port output configuration
MUX_SEL	0xC9	.7 1=CAN interface leads to P0.2 and P0.3,0=CAN interface does not lead out,it is an IO port. .6 1=UART2 interface leads to P0.4 and P0.5,0=UART2 interface does not lead out,it is an IO port. .5 Reserved,write 0. .4 1=FSK interface leads out to P3.6 and P3.7,0=FSK interface does not lead out,it is an IO port. .3 Reserved,write 0. .2 Reserved,write 0. .1 WDT control,1=on 0=off. .0 WDT feeding the dog,1=feeding the dog once (WDT count value reset to zero).
MUX_SEL1	0xE9	.7 ADC conversion completed interrupt flag. After the hardware is set,this will be reset to zero by the CPU (ANL MUX_SEL1,# 7FH). .6 Undefined .5 1=UART5 interface leads to P0.6 and P0.7,0=UART5 interface does not lead,it is an IO port. .4 1=UART4 interface leads out to P0.0 and P0.1,0=UART4 interface does not lead out,it is an IO port. .3 1=PWM3 interface leads out to P3.5,0=PWM3 interface does not lead out,it is an IO port. .2 1=PWM2 interface leads out to P3.4,0=PWM2 interface does not lead out,it is an IO port. .1 1=PWM1 interface leads to P3.3,0=PWM1 interface does not lead out,it is an IO port. .0 1=PWM0 interface leads out to P3.2,0=PWM0 interface does not lead out,it is an IO port.
PORTDRV	0xF9	The driver capability configuration for IO port output mode,only valid for a minimum of 3 bits: 0x00=4mA 0x01=8mA 0x02=16mA 0x03=32mA 0x04=48mA 0x05=64mA 0x06=80mA 0x07=100mA.

The relevant settings for external interrupts on the IO port are as follows:

Interrupt type	Program entry address	Trigger marker	Interrupt enable control	Note
EX0 interrupt	0x0003	IE0（TCON.1)	IEN0.0	Automatically clear IE0 when interrupt response,input corresponding P3.0 IO. IT0 (TCON. 0)=0,low-level trigger interrupt; IT0=1,lower edge trigger interrupt
EX1 interrupt	0x0013	IE1（TCON.3)	IEN0.2	Automatically clear IE1 when interrupt response,input corresponding P3.1 IO. IT1 (TCON. 2)=0,low level trigger interrupt; IT1=1,lower edge trigger interrupt.

3.8 AD Data Reading

The AD of T5F0 is controlled by the GUI core,with a fixed sampling rate of 16KSPS,and the OS CPU can read in one direction. When using a resistive touch screen,users can use 3-channel ADs (AD5-AD7); If not use resistive touch screen,an additional 3 AD channels (AD1-AD3) can be provided to the user.

AD data is updated in the extended SFR,and its definition is shown in the table below:

Address	Size	Definition	Description
0x018	16	AD0-AD7	8-channel AD value,2 bytes per channel,high byte first,12 bit resolution,VIO as reference voltage

The relevant settings for external interrupts on the IO port are as follows:

Interrupt type	Program entry address	Trigger mark	Interrupt enable control	Note
AD interrupt	0x00AB	MUX_SEL1.7	IEN2.5	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag

3.9 UART Communication Interface

3.9.1 UART2 Interface

The SFR control interfaces related to UART2 are shown in the table below.

SFR name	Address	Description
MUX_SEL	0xC9	.6 1=UART2 interface leads out to P0.4 and P0.5,0=UART02 interface does not lead out,it is an IO port
SCON0	0x98	UART2 control interface,same as standard 8051,can be bit addressed. .7=SM0 .6=SM1 .5=SM2(multi machine communication bit) .4=REN0 .3=TB80 .2=RB80 .1=TI0 .0=RI0
SBUF0	0x99	UART2 receive/ transmit data interface
ADCON	0xD8	Baud rate generator selection,0x00=T1 timer (standard 8051),0x80=use SREL0H: L
PCON	0x87	.7=SMOD baud rate frequency doubling selection,0=no doubling,1=doubling
SREL0H	0xBA	When ADCON=0x80,using SREL0H: L to set the baud rate without occupying T1. SMOD=0 SRE0H: L=1024 CPU main frequency/(64 * baud rate); SMOD=1 SRE0H: L=1024 CPU main frequency/(32 * baud rate). The main frequency of the CPU is 74.6496MHz (low-power mode) or 298.5984MHz (high-performance mode)
SREL0L	0xAA	

The relevant settings for UART2 interrupts are as follows:

Interrupt type	Program entry address	Trigger mark	Interrupt enable control	Note
UART2 interrupt	0x0023	RI0（SCON0.0) TI0（SCON0.1)	IEN0.4	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag

3.9.2 UART4 Interface

The SFR control interfaces related to UART4 are shown in the table below.

SFR name	Address	Description
MUX_SEL1	0xE9	.4 1=UART4 interface leads out to P0.0 and P0.1, 0=UART4 interface does not lead out,it is an IO port.
SCON2T	0x96	.7 UART4 transmission enable,0=closed,1=open. .6 UART4 operating mode,0=8-bit mode,1=9-bit mode. .5 TB8,the 9th bit sent in 9bit mode. .4-.1 Write 0. .0 TI,send flag,set when starting to send stop bit,interrupt will trigger the flag.
SCON2R	0x97	.7 UART4 receive enable,0=closed,1=open. .6 Write 0. .5 RB8,the 9th bit received in 9bit. .4-.1 Write 0. .0 RI,receive flag,receive valid stop bit,it will be set when the stop bit ends; Interrupt will trigger the flag.
SBUF2_TX	0x9E	UART4 transmission data interface
SBUF2_RX	0x9F	UART4 receive data interface
BODE2_DIV_H	0xD9	UART4 baud rate setting: BODE2_DIV_H: L=37324800/baud rate (bps). The corresponding setting value for 115200bps is 324 (0x0144)
BODE2_DIV_L	0xE7	

The relevant settings for UART4 interrupts are as follows:

Interrupt type	Program entry address	Trigger mark	Interrupt enable control	Note
UART4 transmission data interrupt	0x0053	SCON2T.0	IEN1.2	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag
UART4 receive data interrupt	0x005B	SCON2R.0	IEN1.3	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag

3.9.3 UART5 Interface

The SFR control interfaces related to UART5 are shown in the table below.

SFR name	Address	Description
MUX_SEL1	0xE9	. 5 1=UART5 interface leads out to P0.6 and P0.7,0=UART5 interface does not lead out,it is an IO port
SCON3T	0xA7	.7 UART5 transmission enable,0=closed,1=open. .6 UART5 operating mode,0=8-bit mode,1=9-bit mode. .5 TB8,the 9th bit sent in 9bit mode. .4-.1 Write 0. .0 TI,send flag,set when starting to send stop bit,interrupt will trigger the flag
SCON3R	0xAB	.7 UART5 receive enable,0=closed,1=open. .6 Write 0. .5 RB8,the 9th bit received in 9bit. .4-.1 Write 0. .0 RI,receive flag,receive valid stop bit,it will be set when the stop bit ends; Interrupt will trigger the flag.
SBUF3_TX	0xAC	UART5 transmission data interface
SBUF3_RX	0xAD	UART5 receive data interface
BODE3_DIV_H	0xAE	UART5 baud rate setting: BODE3-DIV_H: L=37324800/baud rate (bps). The corresponding setting value for 115200bps is 324 (0x0144).
BODE3_DIV_L	0xAF	

The relevant settings for UART5 interrupts are as follows:

Interrupt type	Program entry address	Trigger mark	Interrupt enable control	Note
UART5 transmission data interrupt	0x0063	SCON3T.0	IEN1.4	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag
UART5 receive data interrupt	0x006B	SCON3R.0	IEN1.5	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag

3.10 CAN Communication Interface

The SFR related to CAN interface control are shown in the table below:

SFR name	Address	Description
MUX_SEL	0xC9	.7 1=CAN interface leads out to P0.2 and P0.3, 0=CAN interface does not lead out,it is an IO port
CAN_CR	0x8F	.7 1=CAN interface on 0=CAN interface off. Initially in a closed state .6 1=software reset 0=normal operation,initially in reset state .5 1=configure CAN interface configuration data once (0xFF: 0060-0xFF: 0062),and after hardware processing,it will be reset to zero. . 4 Set speed mode,1=1 sampling; 0=3 sampling . 3 Set filter mode,1=double; 0=single . 2 1=Start sending once,it will be reset to zero after hardware processing (successful send,arbitration failure,EI (CAN_IR.3) error,software reset). .1-.0 Undefined.
CAN_IR	0x91	.7 RF-IF remote frame reception interrupt flag. After the hardware is set,it will be reset to zero by the CPU. .6 CAN_RX_IF CAN. Receive completion interrupt flag. After the hardware is set,it will be reset to zero by the CPU. During the setup period,the hardware cannot update data anymore. .5 CAN_TX_IF CAN. After sending a successful interrupt flag,it will be reset by the CPU after the hardware is set. .4 OI receive overflow flag. After the hardware is set,it will be reset to zero by the CPU. .3 EI error flag. After the hardware is set,it will be reset to zero by the CPU. When an error occurs in CAN_ET [4:0],this bit will be pulled up. .2 Sending arbitration. 1=Sending arbitration failed; 0=Sending arbitration successful. .1-.0 Undefined.
CAN_ET	0xE8	Wrong type register. After the hardware is set,it will be reset to zero by the CPU. .7 Node Hanging mark .6 Proactive error mark .5 Passive error mark .4 CRC verification error mark .3 Response error mark .2 Format error mark .1 Bit padding error mark .0 Bit error mark

CAN communication interface is defined in the extended SFR,using EXADR_H,EXADR_L,and EXDATA for reading and writing,as shown in the table below:

Address	Size	Definition	Description
0x057	1	BRP	CAN baud rate divider register
0x058	1	BTR0	Bit timing register 0,.7-.4 is the synchronization jump width, and .3-.0 is the time period 2 (0-7)
0x059	1	BTR1	Bit timing register 1,.4-.0 is time period 1 (0x00-0x0F)
0x05A	4	ACR3:0	Acceptance code register
0x05E	4	AMR3:0	Acceptance code register
0x062	1	RXERR	Receiving error count register
0x063	1	TXERR	Transmitting error count register
0x064	1	TX_BUFFER_SET	.7=IDE .6=RTR .3-.0=DLC (frame data length)
0x065	4	TX_ID	Extended frame ID.31: ID.3,29bit valid Standard frame ID.31: ID.21,11bit valid
0x069	8	TX_DATA	Transmitting data frame,D1-D8
0x064	1	RX_BUFFER_SET	.7=IDE .6=RTR .3-.0=DLC (frame data length)
0x065	4	RX_ID	Extended frame ID.31: ID.3,29bit valid Standard frame ID.31: ID.21,11bit valid
0x069	8	RX_DATA	Receiving data frame,D1-D8

The relevant settings for CAN interface interrupt is as follows:

Interrupt type	Program entry address	Trigger mark	Interrupt enable control	Note
CAN interface interrupt	0x004B	CAN_IR	IEN1.1	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag

3.11 FSK Transceiver

FSK transceiver is a low-cost,long-distance,high data rate,and high reliability communication method that uses frequency encoding,transmits data frame by frame,and performs FEC (Forward Error Correction) in communication. The physical layer of FSK transceiver can use RS485,RF,plastic fiber,etc.,and is also suitable for power line carrier transmission.

The SFR related to FSK transceiver control are shown in the table below:

SFR name	Address	Description
MUX_SEL	0xC9	.4 1=FSK interface leads out to P3.6 and P3.7, 0=FSK interface does not lead out,it is an IO port. When configuring the FSK interface as a lead out,the OS/GUI (PIN # 21) will automatically convert to the FSK interface's transceiver control signal after resetting: 1=TX,0=RX
FSK_CN	0xEA	.7 FSK control. 1=on,0=off .6 1=Execute an update of FSK control register data (extended SFR 0x048-0x056),and it will be cleared to zero after hardware processing. .5 1=Initiate FSK transmission once,and after the transmission is completed,it will be cleared to zero. .4 1=FSK receiver is in reset state,0=FSK receiver working normally .3 FSK_RX_IF FSK receive completion interrupt flag. It will be cleared to zero by the CPU after hardware processing. During the setup period,the hardware no longer receives and updates data. .2 FSK_TX_IF FSK send completion interrupt flag,It will be cleared to zero by the CPU after hardware processing. .1-.0 Undefined

The communication interface of FSK transceiver is defined in the extended SFR,using EXADR_H,EXADR_L,and EXDATA for reading and writing,as shown in the table below:

Address	Size	Definition	Description
0x048	1	FSK_DIV	FSK transceiver operating frequency=1194.394MHz/FSK_DIV
0x049	1	N_start	When transmitting,the number of starting pulses B0,0x01-0xFF; Generally taken from 16 to 32
0x04A	1	N_sync	The number of synchronous pulses B2,0x01-0x08,Generally taken from 0x04-0x06
0x04B	2	TB0	B0 pulse cycle number (corresponding to FSK transceiver operating frequency),>=6; Frequency when FSK interface sends bit 0 =(1194.394/(FSK_DIV*TB0)) MHz
0x04D	2	E0	B0 pulse decoding tolerance,which means that pulses with a period in the TB0+/- E0 range are all B0
0x04F	2	TB1	B1 pulse cycle number,>=6; TB1>TB0. Frequency when FSK interface sends bit 1=(1194.394/(FSK_DIV*TB1)) MHz
0x051	2	E1	B1 pulse decoding tolerance,which means that pulses with a period within the TB1+/- E1 range are all B1. Generally the same as E0
0x053	2	TB2	B2 pulse cycle number,>=6; TB2>TB1.Frequency when FSK interface sends synchronous bits=(1194.394/(FSK_DIV*TB2)) MHz
0x055	2	E2	B2 pulse decoding tolerance,which means that pulses with a period in the TB2+/- E2 range are all B2
0x080	128	FSK_TX_BUFFER	The first byte is the data length,and a maximum of 127 bytes of data can be sent per frame
0x100	128	FSK_RX_BUFFER	The first byte is the data length
Reference design: FSK_DIV=1 N_START=24 N_SYNC=6 TB1=1.2TB0 TB2=1.3TB1 E0=E1=（TB1-TB0)/2 E2=(TB2-TB1)/2.			

The relevant settings for FSK transceiver interruption are as follows:

Interrupt type	Program entry address	Trigger mark	Interrupt enable control	Note
FSK transceiver interrupt	0x0043	FSK_RX_IF FSK_TX_IF	IEN1.0	After interrupt processing,it is necessary to use software to clear the corresponding interrupt trigger flag

Application example: Configure the FSK transceiver communication rate to be around 1Mbytes/S and send one frame of data "AB".

```
        MOV     EXADR_H,#00H        ;Configure FSK transceiver
        MOV     EXADR_L,#48H
        MOV     EXDATA,#01H         ;FSK clock equals to PLL clock,1194.394MHz
        MOV     EXDATA,#24          ;Number of startup pulses in front
        MOV     EXDATA,#6           ;Number of synchronous pulses
        MOV     EXDATA,#HIGH(90)    ;TB0  F0=13.271MHz
        MOV     EXDATA,#LOW(90)
        MOV     EXDATA,#9           ;E0
        MOV     EXDATA,#HIGH(108)   ;TB1  F1=11.059MHz
        MOV     EXDATA,#LOW(108)
        MOV     EXDATA,#9
        MOV     EXDATA,#HIGH(140)   ;TB2  F2=8.531MHz
        MOV     EXDATA,#LOW(140)
        MOV     EXDATA,#16
        ORL     MUX_SEL,#10H        ;FSK transceiver leads out to IO
        MOV     FSK_CN,#0C0H        ;Open the FSK transceiver and configure it once
        MOV     EXADR_H,#00H        ;Send data “AB”
        MOV     EXADR_L,#80H
        MOV     EXDATA,#02H         ;Send the byte length of data
        MOV     EXDATA,#41H         ;Data ”A”
        MOV     EXDATA,#42H         ;Data ”B”
        MOV     FSK_CN,#0A0H        ;Start sending
WAITFTX:MOV     A,FSK_CN
        JB      ACC.5,WAITFTX       ;Waiting for sending to end
```

3.12 QSPI Bus DMA Operation

The QSPI (SPI interface with 4 data lines) interface on the T5F0 OS CPU is an IO port simulation,and the instructions for QSPI bus operations are processed by user software. T5F0 provides DMA hardware acceleration between the QSPI bus and variable memory during the data read and write phase.

The SFR related to QSPI bus DMA control are shown in the table below:

SFR name	Address	Description
SYSREG	0xF6	.7 1=Initiate the QSPI interface DMA reading once (QSPI bus reads to variable memory),and after hardware execution,it will be reset to zero. .6 1=Initiate the QSPI interface DMA writing once (variable memory written to the QSPI bus),and after the hardware execution is completed,it will be reset to zero.

The timing of QSPI Bus DMA defined in extended SFR,using EXADR_H,EXADR_L,and EXDATA for reading and writing,as shown in the table below:

Address	Size	Definition	Description
0x028	1	QSPI_CLK_DIV	QSPI bus clock frequency configuration,QSPI clock frequency=CPU operating frequency/QSPI_CLK_DIV. For general QSPI interface Flash or PSRAM,When the operating frequency of the CPU is 74.6496MHz (low power mode),it is recommended to configure ASPI_CLK_DIV as 0x02; When the operating frequency of the CPU is 298.5984MHz (high-performance mode),it is recommended to configure ASPI_CLK_DIV as 0x04.
0x029	1	QSPI_CLK_PHS	.7 Read phase configuration: 1=Read data after rising edge, 0=Read data after falling edge. Usually write 1. .6 Write phase configuration: 1=high-level rewrite data, 0=low-level rewrite data. Usually write 0. .5-.0: Write 0.
0x02A	3	QSPI_DMA_LEN	DMA read and write data byte length,0x000001-0x020000,maximum 128K bytes.
0x02D	3	QSPI_DMA_APP_ADR	The variable memory address pointer for reading and writing data. When data is read from the QSPI bus to the variable memory,it is written in the order of DATA0-DATA3. When data is written from the variable memory to the QSPI bus,it is read in the order of DATA0-DATA3.

3.13 Interrupt System

3.13.1 Interrupt Control SFR

The T5F0 OS CPU has a total of 17 interrupts,and the list of related control SFRs are as follows:

SFR name	Address	Description
IEN0	0xA8	Interrupt enables controller 0,which can be bit addressed. .7 EA interrupt total control bit; 0=all interrupts closed,1=whether interrupts are opened is controlled by the control bit of each interrupt. .6 Must be written as 0. .5 ET2,T2 timer interrupt enable control bit. .4 ES0,UART2 interrupt enable control bit. .3 ET1,T1 timer interrupt enable control bit. .2 EX1,external interrupt 1 (P3.1 pin) interrupt enable control bit. .1 ET0,T0 timer interrupt enable control bit. .0 EX0,external interrupt 0 (P3.0 pin) interrupt enable control bit.
IEN1	0xB8	Interrupt enable controller 1,which can be bit addressed. .7-.6 Must be written as 0. .5 ES3R,UART5 receive interrupt enable control bit. .4 ES3T,UART5 transmit interrupt enable control bit. .3 ES2R,UART4 receive interrupt enable control bit. .2 ES2T,UART4 transmit interrupt enable control bit. .1 ECAN,CAN communication interrupt enable control bit. .0 EFSK,FSK transceiver interrupt enable control bit.
IEN2	0x9A	Interrupt enable controller 2. .7-.6 Must be written as 0. .5 EADC,ADC data conversion end interrupt enable control bit. .4 EPWM3,PWM3 interrupt enable control bit. .3 EPWM2,PWM2 interrupt enable control bit. .2 EPWM1,PWM1 interrupt enable control bit. .1 EPWM0,PWM0 interrupt enable control bit. .0 Must be written as 0.
IEN3	0xD1	Interrupt enable controller 3,must be written as 0x00.
IP0	0xA9	Interrupt priority controller 0
IP1	0xB9	Interrupt priority controller 1

3.13.2 Interrupt Priority

The interrupt priority of T5F0 OS CPU is processed according to the following rules:

(1) 17 interrupts are divided into 6 groups,with 2-3 interrupts in each group,and the priority within each group is fixed. For example,external interrupt 1 in the same group has a higher priority than PWM1 interrupt,and PWM1 interrupt has a higher priority than UART4 transmit interrupt,as shown in the table below.

Group	IP0 counterpart	IP1 counterpart	Interrupt counterpart		
			High priority	Medium priority	Low priority
G0	.0	.0	External interrupt 0	FSK transceiver interrupt	
G1	.1	.1	T0 timer interrupt	PWM0 interrupt	CAN communication interrupt
G2	.2	.2	External interrupt 1	PWM1 interrupt	UART4 transmit interrupt
G3	.3	.3	T1 timer interrupt	PWM2 interrupt	UART4 receive interrupt
G4	.4	.4	UART2 interrupt	PWM3 interrupt	UART5 transmit interrupt
G5	.5	.5	T2 timer interrupt	ADC data conversion end interrupt	UART5 receive interrupt

(2) There are 4 levels of priority between 6 groups,which can be configured by the corresponding bits of IP0 and IP1. See the table below.

Priority between groups	IP1 corresponding bit	IP0 corresponding bit
3(highest)	1	1
2	1	0
1	0	1
0(lowest)	0	0

For example,to set the priority of T2 timer interrupt,ADC data conversion end interrupt,and UART5 receive interrupt in G5 group to the highest,it can be set: IP1=0x20 (IP1.5=1),IP0=0x20 (IP0.5=1).

(3) If the configured priorities are all the same (IP1=0x00 IP1=0x00),then the priority of the G0 group is the highest,and the priority of the G5 group is the lowest. The table of interrupt priority weights for the same configuration are as follows:

Weight	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Priority	highest																lowest
Interrupt	EX0	FSK	T0	PWM0	CAN	EX1	PWM1	TX4	T1	PWM2	RX4	UART2	PWM3	TX5	T2	ADC	RX5

(4) High priority interrupts can be nested within low priority interrupts (smaller weight interrupt can be interrupted by larger weight interrupt),with a maximum of four levels of nesting.

T5F0 OS has a fast CPU speed (in high-performance mode,an average of 200-250 instructions can be executed in 1uS),and due to the short running time of interrupt programs,its real-time performance is already very high,it is not recommended for users to use interrupt nesting,which leads to complex program architecture. It is recommended to directly close the interrupt (EA=0) during the execution of each interrupt service program,and then open the interrupt (EA=1) when exiting.

3.14 8051 Instruction Set for OS CPU Core

Instruction Format	Instruction length	Instruction cycle	Instruction Format	Instruction length	Instruction cycle
ADD/ADDC A,Rn	1	1	MOV @Ri,direct	2	2
ADD/ADDC A,direct	2	2	MOV @Ri,#data	2	2
ADD/ADDC A,@Ri	1	2	MOV DPTR,#data16	3	3
ADD/ADDC A,#data	2	2	MOVC A,@A+DPTR	1	3
SUBB A,Rn	1	1	MOVC A,@A+PC	1	3
SUBB A,direct	2	2	MOVX A,@Ri	1	3
SUBB A,@Ri	1	2	MOVX A,@DPTR	1	3
SUBB A,#data	2	2	MOVX @Ri,A	1	3
INC/DEC A	1	1	MOVX @DPTR,A	1	3
INC/DEC Rn	1	1	PUSH/POP	1	2
INC/DEC direct	2	2	XCH A,Rn	1	1
INC/DEC @Ri	1	2	XCH A,direct	2	2
INC DPTR	1	1	XCH A,@Ri	1	2
MUL AB	1	4	XCHD A,@Ri	1	2
DIV AB	1	4	ACALL addr11	2	2
DA A	1	1	LCALL addr16	3	3
ANL/ORL/XRL A,Rn	1	1	RET/RETI	1	4
ANL/ORL/XRL A,direct	2	2	AJMP addr11	2	2
ANL/ORL/XRL A,@Ri	1	2	LJMP addr16	3	3
ANL/ORL/XRL A,#data	2	2	SJMP rel	2	3
ANL/ORL/XRL direct,A	2	2	JMP @A+DPTR	1	3
ANL/ORL/XRL direct,#data	3	3	JZ/JNZ/JC/JNC rel	2	3
CLR A	1	1	JB/JNB/JBC	3	4
CPL A	1	1	CJNE A,direct,rel	3	4
RL/RR A	1	1	CJNE A,#data,rel	3	4
RLC/RRC A	1	1	CJNE Rn,#data,rel	3	4
SWAP A	1	1	CJNE @Ri,#data,rel	3	5
MOV A,Rn	1	1	DJNZ Rn,rel	2	3
MOV A,direct	2	2	DJNZ direct,rel	3	4
MOV A,@Ri	1	2	NOP	1	1
MOV A,#data	2	2	CLR/SETB C	1	1
MOV Rn,A	1	1	CLR/SETB bit	2	2
MOV Rn,direct	2	2	CPL C	1	2
MOV Rn,#data	2	2	CPL bit	2	2
MOV direct,A	2	2	ANL C,bit	2	2
MOV direct,Rn	2	2	ANL C,/bit	2	2
MOV direct1,direct2	3	3	ORL C,bit	2	2
MOV direct,@Ri	2	2	ORL C,/bit	2	2
MOV direct,#data	3	3	MOV C,bit	2	2
MOV @Ri,A	1	1	MOV bit,C	2	2

When the operating frequency of T5F0 is 74.6496MHz (low-power mode),one instruction cycle (1T) takes 13.396nS.

When the operating frequency of T5F0 is 298.5984MHz (high-performance mode),one instruction cycle (1T) takes 3.349nS.

For example,when T5F0 operates in high-performance mode,the following code will output a 37.3248MHz square wave on the P3.3 IO port.

```
OUTTEST: CPL     P3.3          ;2T
         AJMP    OUTTEST       ;2T
```

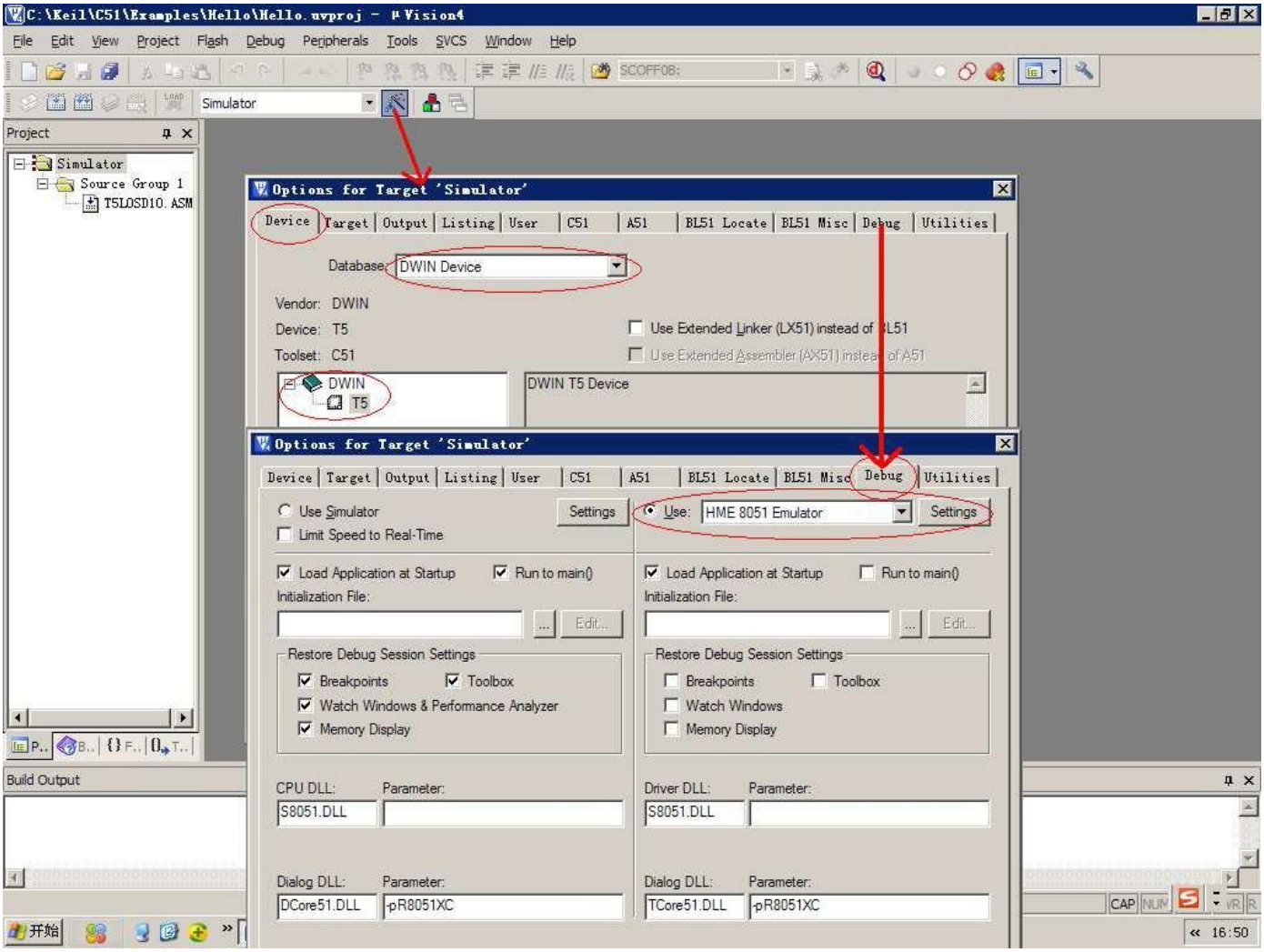
4. Simulation Debugging

With the help of the HME05 simulator (which requires the installation of the corresponding USB driver),users can connect to the JTAG interface of T5F0 and perform code IAP debugging and simulation running in the Keil development environment.

During simulation debugging,pay attention to the following:

(1) The JTAG interface must be selected to the OS CPU and must be in JTAG mode,where OS/GUI (PIN # 21)=1 JTAGS (PIN # 25)=0. There are corresponding jumper selection and instructions on the DWIN evaluation board.

(2) Install the AGDI driver to enable Keil to support T5F0 and HME05 simulators. After installation,select and configure according to the following diagram. And copy the header files (*. INC or *. h) of the T5F0 OS CPU to the directory:KEIL/C51/INC/DWIN.



(3) Before setting a breakpoint to read the contents of the data storage (XDATA),it is necessary to ensure that DPC=0x00,otherwise the data will be misaligned.

(4) Before simulation,it is necessary to ensure that the OS CPU code 0x00F8 burned in T5F0 Flash starts at 0xFF FF FF 44 57 49 4E 54 35,otherwise the JTAG interface of the OS CPU will be disabled and the HME05 simulator cannot be connected.

(5) When porting code to standard C or other 8051 platforms,users should pay attention to selecting the corresponding T5F0 file for SFR header loading during compilation INC or H file. If the SFR definition in the customer code is different from the T5F0 definition,the SFR header file in the code or T5F0 can be modified to be consistent.

(6) HME05 achieves hardware simulation by downloading code to the Code RAM of the T5F0 OS CPU,and the code is not burned into the on-chip Flash. To burn the code into the chip,it is necessary to use the SD card interface. When the SD interface is burned,the T5F0 underlying software will automatically change the 0x00F8 position of the OS code to 0x0000 (JTAG interface prohibited) 44 57 49 4E 54 35.

T5F0 OS CPU adopts the standard 8051 architecture,which is identical to the instruction set except for slight differences in SFR and extended peripheral access. When porting the user's original 8051 code,pay attention to the following aspects to quickly complete it:

(1) According to the hardware design,after resetting,use the startup.A51(C51 startup code) or initcpu() assembly program provided by DWIN to simply modify and configure the unique SFR and parameter settings of T5F0.

(2) The IO output mode of T5F0 is controllable. When switching between input and output modes,it is necessary to configure the PxMDOUT register accordingly,otherwise errors will occur.

(3) Close nested interrupts,when each interrupt service program enters,EA=0; when exits,EA=1

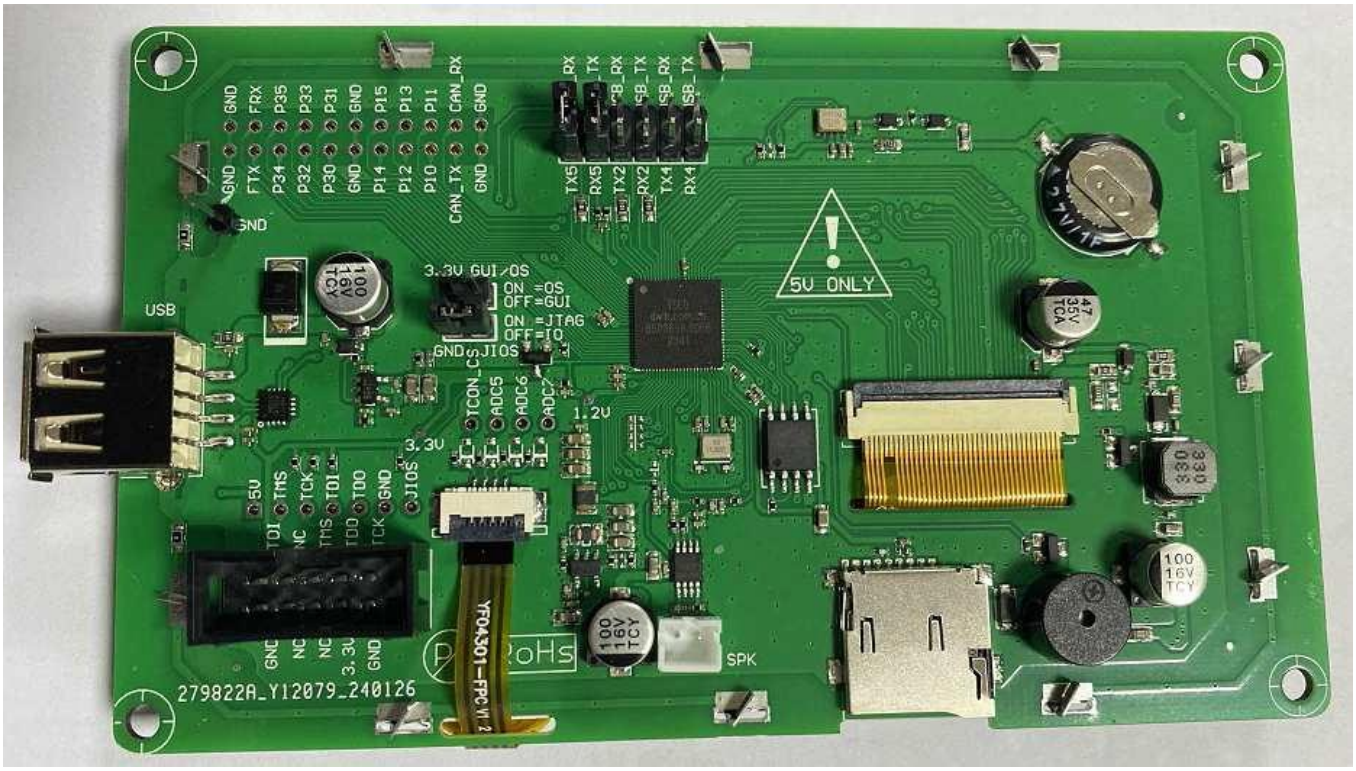
(4) When using off chip RAM (XRAM) for data storage in the code,please note that the 32KB data RAM starting address of T5F0 starts from 0x8000.

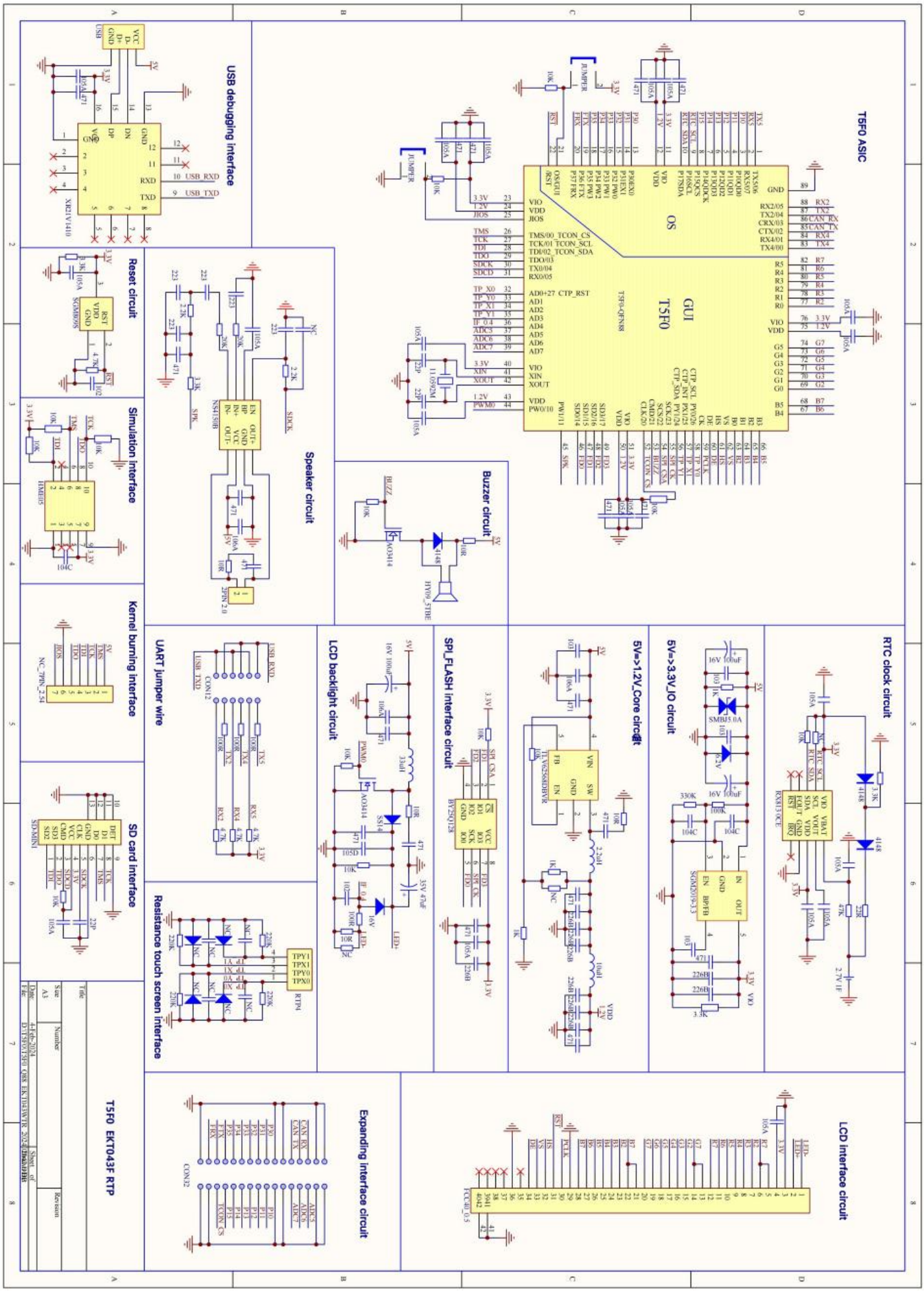
(5) Write 0xFFFF (or 0x0000 to disable JTAG interface)44 57 49 4E 54 35 at position 0x00F8.

(6) Using MDU/FMU hardware operation acceleration to optimize the algorithm of the original code.The UI function can be implemented on the DGUS configuration development running on the GUI core or TA instruction set development platform. User code can be processed through simple read and write variable memory,greatly improving product performance and enhancing research and development efficiency.

5. EK043F Evaluation Board

EK043F evaluation board adopts a 4.3-inch 480 * 272 TFT screen driven by T5F0,and is equipped with a four wire resistive touch screen.




T5F0 ASIC
USB debugging interface
Reset circuit
Simulation interface
Kernel burning interface
SD card interface
Speaker circuit
Buzzer circuit
UART jumper wire
LCD backlight circuit
Resistance touch screen interface
SPI_FLASH interface circuit
5V>1.2V_Core circuit
5V>3.3V_IO circuit
RTC clock circuit
LCD interface circuit
Expanding interface circuit
T5F0 EKT043F RTP

Revision Records

Version	Revise Date	Content	Editor
1.0	2024-03-26	First Edition	Xu Ying

Please contact us if you have any questions about the use of this document or our products,or if you would like to know the latest information about our products:

- Customer service Tel: +86 400 018 9008
- Customer service email: dwinhmi@dwin.com.cn
- DWIN Developer Forum: https://forums.dwin-global.com/

Thank you all for continuous support of DWIN,and your approval is the driving force of our progress!