



## 目 录

1 概述.....	2
2 硬件设计.....	3
2.1 IC 引脚定义.....	3
2.2 封装尺寸 .....	6
2.3 基本性能参数 .....	7
2.4 硬件设计注意事项 .....	7
3 OS CPU 功能模块编程说明.....	8
3.1 初始化配置 .....	8
3.2 存储器 .....	8
3.2.1 代码存储器（64KBytes） .....	8
3.2.2 DGUS 变量存储器（256KBytes） .....	9
3.2.3 数据存储器（32KBytes） .....	10
3.2.4 扩展 SFR 寄存器 .....	10
3.3 数学运算单元（MDU） .....	11
3.4 定时器 .....	12
3.5 看门狗定时器（WDT） .....	12
3.6 IO 口.....	13
3.7 UART 通信接口.....	13
3.7.1 UART2 接口 .....	13
3.7.2 UART3 接口 .....	14
3.7.3 UART4 接口 .....	14
3.7.4 UART5 接口 .....	14
3.8 CAN 通信接口.....	15
3.9 中断系统 .....	16
3.9.1 中断控制 SFR .....	16
3.9.2 中断优先级.....	16
3.10 T5L ASIC 的 8051 指令集.....	18
4 仿真调试.....	19
5 EK043 评估板.....	20
附录 1 修订记录 .....	21

## 1 概述



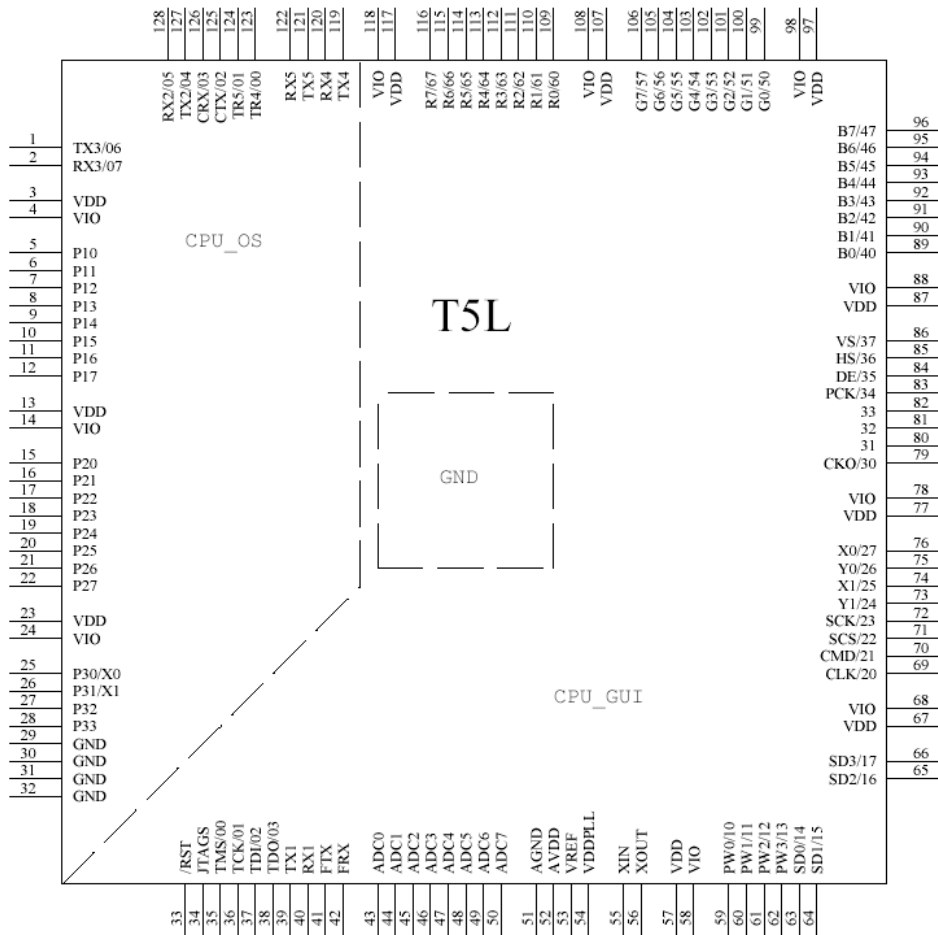
T5L 系列 ASIC 是迪文科技针对 AIoT 应用设计的低功耗、高性价比、GUI 和应用高度整合的单芯片双核 ASIC IC，包括 T5L0、T5L1、T5L2 三个版本，其主要特点包括：

- (1) 采用应用最广泛、成熟和稳定的 8051 核，1T（单指令周期）高速工作，最高主频 400MHz。
- (2) 单独 CPU 核（GUI CPU）运行 DGUS II 系统：
  - 内置高速显存，2.4GB/S 带宽，18bit（T5L0）或 24bit（T5L1、T5L2）彩色显示分辨率支持到 480\*854（T5L0）、800\*600（T5L1）、1366\*768（T5L2）。
  - 2D 硬件加速，JPEG 解压缩速度高达 200fps@1280\*800，动画和图标为主的 UI 极其炫酷、流畅。
  - JPEG 压缩模式存储图片、图标，大幅度缩小外置存储器到低成本的 16Mbytes SPI Flash。
  - 支持电阻或电容触摸屏，灵敏度可以调节，最快 400Hz 触控打点速度。
  - 高品质语音压缩存储和播放。
  - 128Kbytes 变量存储器空间，存储器接口和 OS CPU 核交换数据，应用及其简单。
  - 2 路 10bit 800KHz DC/DC 控制器，简化 LED 背光、模拟电源设计并节约成本和空间。
  - 1 路 15bit 32Ksps PWM 数字功放驱动扬声器，节约功放成本并获得高信噪比和音质还原。
  - 支持 PC 端组态开发和仿真，支持后台远程升级。
- (3) 单独 CPU 核（OS CPU）运行用户 8051 代码或迪文 DWIN OS 系统，应用中省掉用户 CPU：
  - 标准 8051 架构和指令集，64Kbytes 代码空间，32Kbytes 片内 RAM。
  - 64bit 整数型数学运算单元（MDU），包括 64bit MAC 和 64bit 除法器。
  - 内置软件 WDT，3 个 16bit Timers，12 路中断信号支持最高四级中断嵌套。
  - 22 个 IO，4 路 UARTs，1 路 CAN 接口，最多 8 路 12bit A/D，1 路 16bit 分辨率可调的 PWM。
  - 支持 IAP 在线仿真和调试，断点数量无限制；可以透过 DGUS 系统在线升级代码。
- (4) 1Mbytes 片内 Flash，迪文专利加密技术，确保代码和数据安全，杜绝山寨和克隆。
- (5) 针对各种廉价的宽范围调谐阻抗晶体设计的振荡器和 PLL，降低晶体要求和 PCB 设计难度。
- (6) 3.3V IO 电压，可以适应 1.8/2.5/3.3 各种电平。
- (7) 支持 SD 接口或 UART1 下载和配置，支持 SD 卡文件的读取和改写。
- (8) 支持迪文 WiFi 模块直接接入迪文云，轻松开发各种云端应用。
- (9) -40℃ - +85℃ 工作温度范围（可定制 -55℃ - 105℃ 工作温度范围 IC）。
- (10) 功耗低，抗干扰能力强，可以稳定工作在双面 PCB 设计上，轻松通过 EMC/EMI 测试。
- (11) 采用 0.4mm ELQFP128 封装，制造加工难度低，成本低。
- (12) 针对行业客户提供 **T5L IC+液晶屏+触摸屏+设计支持** 的高性价比配套方案销售和全方位技术服务支持。

## 2 硬件设计

### 2.1 IC引脚定义

T5L ASIC 采用 ELQFP128 封装（16\*16\*1.5 mm），管脚排列如图 2.1-1。



OS	11	P1.6	I/O 口。				
OS	12	P1.7	I/O 口。				
OS	13	VDD	T5L0、1=1.25V T5L2=1.2V				
OS	14	VIO	3.3V				
OS	15	P2.0	I/O 口。				
OS	16	P2.1	I/O 口。				
OS	17	P2.2	I/O 口。				
OS	18	P2.3	I/O 口。				
OS	19	P2.4	I/O 口。				
OS	20	P2.5	I/O 口。				
OS	21	P2.6	I/O 口。				
OS	22	P2.7	I/O 口。				
OS	23	VDD	T5L0、1=1.25V T5L2=1.2V				
OS	24	VIO	3.3V				
OS	25	P3.0	I/O 口。	EX0	外部中断 0 输入。		
OS	26	P3.1	I/O 口。	EX1	外部中断 1 输入。		
OS	27	P3.2	I/O 口。				
OS	28	P3.3	I/O 口。				
	29	GND					
	30	GND					
	31	GND					
	32	OS/GUI	0:GUI JTAG, 1=OS JTAG。				
	33	/RST	系统复位输入。				
	34	JTAGS	PIN#35-PIN#38 选择: 0=JTAG, 1=I/O 口。				
GUI	35	P0.0	I/O 口。	TMS	JTAG 接口。	TCON_CS	液晶屏 TCON 接口。
GUI	36	P0.1	I/O 口。	TCK	JTAG 接口。	TCON_CLK	液晶屏 TCON 接口。
GUI	37	P0.2	I/O 口。	TDI	JTAG 接口。	TCON_DATA	液晶屏 TCON 接口。
GUI	38	P0.3	I/O 口。	TDO	JTAG 接口。	TCON_RST	液晶屏 TCON 接口。
GUI	39	P0.4	I/O 口。	TX1	UART1 数据发送。		
GUI	40	P0.5	I/O 口。	RX1	UART1 数据接收。		
GUI	41	P0.6	I/O 口。	FSK_TX	FSK 收发器数据发送。		
GUI	42	P0.7	I/O 口。	FSK_RX	FSK 收发器数据接收。		
GUI	43	ADC0	AD 输入。	RTP_X0	4 线电阻触摸屏接口。		
GUI	44	ADC1	AD 输入。	RTP_Y0	4 线电阻触摸屏接口。		
GUI	45	ADC2	AD 输入。	RTP_X1	4 线电阻触摸屏接口。		
GUI	46	ADC3	AD 输入。	RTP_Y1	4 线电阻触摸屏接口。		
GUI	47	ADC4	AD 输入。	IF_0.4	DC/DC 0.4V 电压反馈。		
GUI	48	ADC5	AD 输入。	VF_1.25	DC/DC 1.25V 电压反馈。		
GUI	49	ADC6	AD 输入。				
GUI	50	ADC7	AD 输入。				
GUI	51	AGND	AD GND。				
GUI	52	AVDD	AD 供电, 3.3V, 就近 470pJ (C0G 材料) 并联 105 滤波。				
GUI	53	VREF	AD 参考电源, 最高 AVDD+0.3V, 就近 470pJ (C0G 材料) 并联 105 滤波。				
GUI	54	VDDPLL	T5L0、1=1.25V T5L2=1.2V 必须就近 470pJ (C0G 材料) 并联 105 滤波。				
GUI	55	XIN	晶体, 11.0592MHz。	CLK_IN	3.3V 时钟输入。		
GUI	56	XOUT	晶体				
GUI	57	VDD	T5L0、1=1.25V T5L2=1.2V				
GUI	58	VIO	3.3V				
GUI	59	P1.0	I/O 口。	PWM0	16bit PWM 输出。		
GUI	60	P1.1	I/O 口。	PWM1	16bit PWM 输出。	PWM_V	液晶屏 AVDD DC/DC。

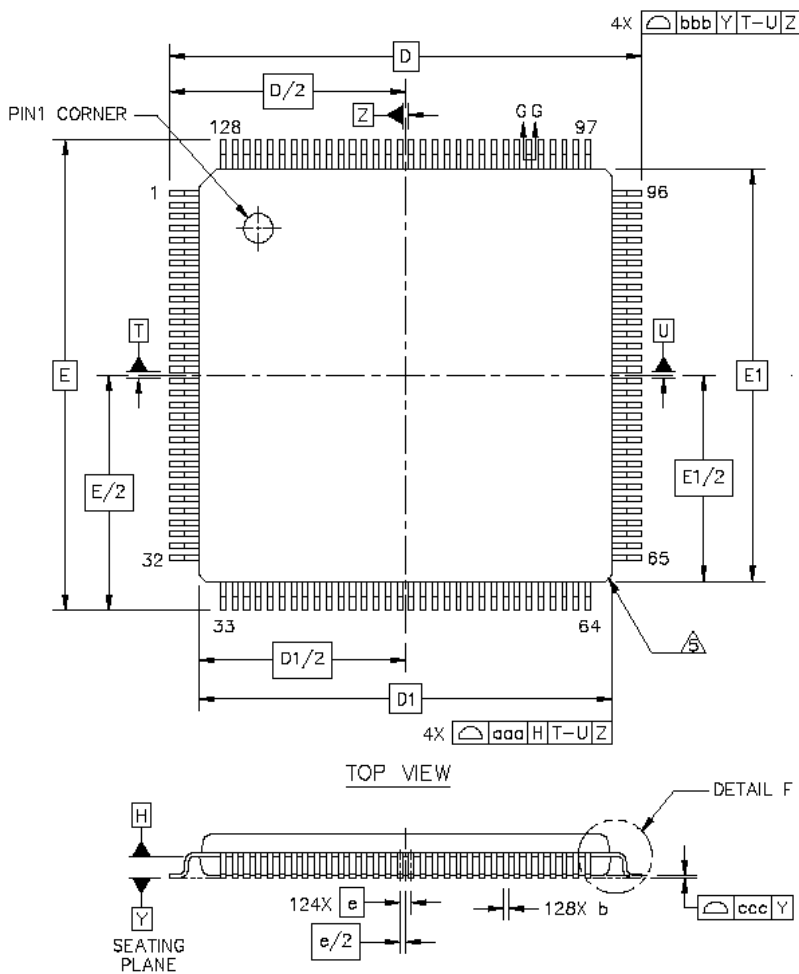


GUI	61	P1.2	I/O 口。	PWM2	16bit PWM 输出。	PWM_I	液晶屏背光 DC/DC。
GUI	62	P1.3	I/O 口。	PWM3	16bit PWM 输出。	BUZZ&SPK	蜂鸣器/扬声器驱动。
GUI	63	P1.4	I/O 口。	SDD0	SD 接口：数据。	SPI_D0	SPI Flash 接口：数据。
GUI	64	P1.5	I/O 口。	SDD1	SD 接口：数据。	SPI_D1	SPI Flash 接口：数据。
GUI	65	P1.6	I/O 口。	SDD2	SD 接口：数据。	SPI_D2	SPI Flash 接口：数据。
GUI	66	P1.7	I/O 口。	SDD3	SD 接口：数据。	SPI_D3	SPI Flash 接口：数据。
GUI	67	VDD	T5L0、1=1.25V T5L2=1.2V				
GUI	68	VIO	3.3V				
GUI	69	P2.0	I/O 口。	SDCK	SD 接口：时钟。	PA_EN	语音播放的功放开关。
GUI	70	P2.1	I/O 口。	SDCD	SD 接口：指令。		
GUI	71	P2.2	I/O 口。	SPI_CS	SPI Flash 接口：片选。		
GUI	72	P2.3	I/O 口。	SPI_CLK	SPI Flash 接口：时钟。		
GUI	73	P2.4	I/O 口。	RTP_Y1	4 线电阻触摸屏接口。	CTP_SDA	电容触摸屏接口。
GUI	74	P2.5	I/O 口。	RTP_X1	4 线电阻触摸屏接口。	CTP_INT	电容触摸屏接口。
GUI	75	P2.6	I/O 口。	RTP_Y0	4 线电阻触摸屏接口。	CTP_SCL	电容触摸屏接口。
GUI	76	P2.7	I/O 口。	RTP_X0	4 线电阻触摸屏接口。	CTP_RST	电容触摸屏接口。
GUI	77	VDD	T5L0、1=1.25V T5L2=1.2V				
GUI	78	VIO	3.3V				
GUI	79	P3.0	I/O 口。	CLK_OUT	系统时钟分频输出。		
GUI	80	P3.1	I/O 口。	FSK_TR	FSK 收发器半双工使用的 T/R 切换信号。		
GUI	81	P3.2	I/O 口。				
GUI	82	P3.3	I/O 口。				
GUI	83	P3.4	I/O 口。	LCD_PCLK	LCD 接口。		
GUI	84	P3.5	I/O 口。	LCD_DE	LCD 接口。		
GUI	85	P3.6	I/O 口。	LCD_HS	LCD 接口。		
GUI	86	P3.7	I/O 口。	LCD_VS	LCD 接口。		
GUI	87	VDD	T5L0、1=1.25V T5L2=1.2V				
GUI	88	VIO	3.3V				
GUI	89	P4.0	I/O 口。	LCD_B0	LCD 接口。		
GUI	90	P4.1	I/O 口。	LCD_B1	LCD 接口。		
GUI	91	P4.2	I/O 口。	LCD_B2	LCD 接口。		
GUI	92	P4.3	I/O 口。	LCD_B3	LCD 接口。		
GUI	93	P4.4	I/O 口。	LCD_B4	LCD 接口。		
GUI	94	P4.5	I/O 口。	LCD_B5	LCD 接口。		
GUI	95	P4.6	I/O 口。	LCD_B6	LCD 接口。		
GUI	96	P4.7	I/O 口。	LCD_B7	LCD 接口。		
GUI	97	VDD	T5L0、1=1.25V T5L2=1.2V				
GUI	98	VIO	3.3V				
GUI	99	P5.0	I/O 口。	LCD_G0	LCD 接口。		
GUI	100	P5.1	I/O 口。	LCD_G1	LCD 接口。		
GUI	101	P5.2	I/O 口。	LCD_G2	LCD 接口。		
GUI	102	P5.3	I/O 口。	LCD_G3	LCD 接口。		
GUI	103	P5.4	I/O 口。	LCD_G4	LCD 接口。		
GUI	104	P5.5	I/O 口。	LCD_G5	LCD 接口。		
GUI	105	P5.6	I/O 口。	LCD_G6	LCD 接口。		
GUI	106	P5.7	I/O 口。	LCD_G7	LCD 接口。		
GUI	107	VDD	T5L0、1=1.25V T5L2=1.2V				

GUI	108	VIO	3.3V				
GUI	109	P6.0	I/O 口。	LCD_R0	LCD 接口。		
GUI	110	P6.1	I/O 口。	LCD_R1	LCD 接口。		
GUI	111	P6.2	I/O 口。	LCD_R2	LCD 接口。		
GUI	112	P6.3	I/O 口。	LCD_R3	LCD 接口。		
GUI	113	P6.4	I/O 口。	LCD_R4	LCD 接口。		
GUI	114	P6.5	I/O 口。	LCD_R5	LCD 接口。		
GUI	115	P6.6	I/O 口。	LCD_R6	LCD 接口。		
GUI	116	P6.7	I/O 口。	LCD_R7	LCD 接口。		
GUI	117	VDD	T5L0、1=1.25V T5L2=1.2V				
GUI	118	VIO	3.3V				

注意，IC 底部的焊盘必须要可靠接地，否则会影响 IC 的性能。

## 2.2 封装尺寸



	SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS	A	---	---	1.6
STAND OFF	A1	0.05	---	0.15
MOLD THICKNESS	A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)	b	0.13	0.18	0.23
LEAD WIDTH	b1	0.13	---	0.19
L/F THICKNESS(PLATING)	c	0.09	---	0.2
L/F THICKNESS	c1	0.09	---	0.16
	X D	16 BSC		
	Y E	16 BSC		
BODY SIZE	X D1	14 BSC		
	Y E1	14 BSC		
EP SIZE	X M	5.0	5.1	5.2
	Y N	5.0	5.1	5.2
LEAD PITCH	e	0.4 BSC		
	L	0.45	0.6	0.75
FOOTPRINT	L1	1 REF		
	8	0'	3.5'	7'
	81	0'	---	---
	82	11'	12'	13'
	83	11'	12'	13'
	R1	0.08	---	---
	R2	0.08	---	0.2
	S	0.2	---	---
PACKAGE EDGE TOLERANCE	aaa	0.2		
LEAD EDGE TOLERANCE	bbb	0.2		
COPLANARITY	ccc	0.08		
LEAD OFFSET	ddd	0.07		
MOLD FLATNESS	eee	0.05		

PCB 设计请使用迪文科技提供的器件封装和参考设计。

## 2.3 基本性能参数

参 数	单 位	最小值	典型值	最大值	说 明
CPU 核电压	V	1.10	1.20	1.30	T5L2, 推荐 1.2V、纹波不超过 50mV 供电。
	V	1.15	1.25	1.40	T5L0、T5L1, 推荐 1.25V、纹波不超过 50mV 供电。
CPU 核电流	mA		100		双核 200MHz 全速工作。
I/O 电压 (VIO)	V	1.8	3.3	3.6	5V TTL/CMOS 电平输入需要做分压或钳位保护。
AD 工作电压 (VADD)	V	1.8	3.3	3.6	
AD 参考电压 (Vref)	V			VADD+0.3	
AD 输入电压	V			Vref+0.3	
I/O 高电平输出幅度 (VOH)	V	3.0			VIO=3.3V, I/O 负载电流 8mA。
I/O 低电平输出幅度 (VOL)	V			0.3	VIO=3.3V, I/O 负载电流 8mA。
I/O 高电平输出电流	mA	-10			VIO=3.3V, VOH=3V。
I/O 低电平输入电流	mA	10			VIO=3.3V, VOL=0.3V。
I/O 口翻转速度	MHz		100		
I/O 高电平识别电压 (VIH)	V	1.6			
I/O 低电平识别电压 (VIL)	V			0.6	
外接晶体频率	MHz	10.0	11.0592	12.0	
CPU 主频 (CPU_CLK)	MHz		206.4384		对应 11.0592MHz 晶体, CPU 主频=晶体频率*56/3。 可以定制 350MHz 版本的代码。
工作温度	°C	-40		+85	
存储温度	°C	-55		+105	
ESD 防护能力	KV		2		HBM

## 2.4 硬件设计注意事项

(1) 核电压务必稳定, 不然会导致 CPU 运行异常 (低于最小值会导致死机, 超过最大值可能会损坏 IC)。

(2) 复位推荐使用 SGM809S 之类的低电平复位 IC 处理, 不要使用简单的 RC 复位电路。

T5L 每个 CPU 核都内置有看门狗 (WDT), 无需外接 WDT IC。

(3) 推荐使用四层板应用设计, 可以获得极好的电磁兼容特性。

采用双面板应用设计时, 请尽可能靠近 IC 供电引脚接 470pF 并联 104 (或 105) 的电容滤波并降低噪声辐射。

(4) 超过 VIO 电压 0.3V 以上的 I/O 信号输入, 必须使用分压或钳位保护, 不然可能会导致信号异常或损坏 IC。

(5) 所有 I/O 口配置为输入使用时, 均为浮空输入, 没有内部上拉或下拉;

所有 I/O 口在复位过程中为输入状态, 如作为输出时可以外部下拉或上拉以确保复位过程有确定的电平。

(6) T5L 和外部 SPI Flash 的 4bit 总线速度为 100MHz, 布线应尽可能靠近, 并必须在 SPI Flash 的电源引脚布置 470pF 并联 105 的电容滤波。



### 3 OS CPU 功能模块编程说明

T5L 的 OS CPU 采用工业应用最广泛，量产时间最久，久经考验的 8051 内核。在保留 8051 实时性好、I/O 速度快，稳定可靠的基础上，迪文通过优化代码处理、扩展 SFR 总线、增加硬件数学处理器，大幅度提升了 8051 存储器访问和计算能力。

对于以计算和数据处理为主的应用，用户也可以让 OS CPU 运行 DWIN OS 平台，在 DWIN OS 平台进行二次应用开发，具体开发方式请参考《基于 T5L CPU 的 DWIN OS 程序开发指南》。

#### 3.1 初始化配置

OS 8051 内核上电后，必须对下表中的特殊功能寄存器（SFR）进行正确的初始化。

SFR 名称	地 址	初始化配置值	说 明
CKCON	0x8E	0x00	CPU 运行在 1T 模式。
T2CON	0xC8	0x70	配置扩展中断系统，配置定时器 T2 运行在 Autoload 模式。
DPC	0x93	0x00 或 0x01	MOVX 指令操作后，DPTR 的变化模式，如果使用 C51 开发，必须配置为 0x00。 0x00：不变。0x01：DPTR=DPTR+1。0x03：DPTR=DPTR-1。
PAGESEL	0x94	0x01	64KB 代码空间。
D_PAGESEL	0x95	0x02	32KB MOVX 访问的 RAM 空间，地址为 0x8000-0xFFFF。
MUX_SEL	0xC9	0x60 或根据应用需要配置。	外设复用选择。 . 7 1=CAN 接口引出到 P0.2、P0.3， 0=CAN 接口不引出，为 I/O 口。 . 6 1=UART2 接口引出到 P0.4、P0.5， 0=UART2 接口不引出，为 I/O 口。 . 5 1=UART3 接口引出到 P0.6、P0.7， 0=UART3 接口不引出，为 I/O 口。 . 4-2 保留 . 1 WDT 控制，1=开启 0=关闭。 . 0 WDT 喂狗，1=喂狗一次（WDT 计数值归零，看门狗的溢出时间为 1 秒）。
PORTDRV	0xF9	0x01 或根据应用需要配置。	I/O 口输出模式的驱动能力配置，0x00=4mA 0x01=8mA 0x02=16mA 0x03=32mA。
RAMMODE	0xF8	0x00	DGUS 变量存储器访问接口控制：

#### 3.2 存储器

OS 的 8051 核可以访问 7 种不同的存储器，如下表所示。

存储器类型	大小	地址空间	访问方式
代码存储器	64KBytes	0x0000-0xFFFF	只能用 MOVX 指令读，同标准 8051。
数据寄存器	256Bytes	0x00-0xFF	同标准 8051。
SFR 寄存器	128Bytes	0x80-0xFF	同标准 8051，迪文可提供用户 SFR 定义.INC 或.H 头文件。
扩展 SFR 寄存器	64Bytes	0x00-0x3F	使用 EXADR、EXDATA 定义的 SFR 寄存器接口来访问。
数据存储器	32KBytes	0x8000-0xFFFF	使用 MOVX 指令访问，DPC 配置为 0x00 时同标准 8051。
DGUS 变量存储器	256KBytes	0x00:0000-0x00:FFFF	使用 DGUS 变量存储器接口来访问。
CAN 通信接口	48Bytes	0xFF:0060-0xFF:006B	使用 DGUS 变量存储器接口来访问。

##### 3.2.1 代码存储器（64KBytes）

代码存储器空间的功能分割和定义如下表所示。

地 址	定 义	说 明
0x0000	Reset_PC	复位后程序开始运行地址。
0x0003	EX0_ISR_PC	外部中断 0 程序接口。
0x000B	T0_ISR_PC	Timer0 中断程序接口。
0x0013	EX1_ISR_PC	外部中断 1 程序接口。
0x001B	T1_ISR_PC	Timer1 中断程序接口。
0x0023	UART2_ISR_PC	UART2 TX/RX 中断程序接口。
0x002B	T2_ISR_PC	Timer 中断程序接口。
0x004B	CAN_ISR_PC	CAN 接口中断程序接口。
0x0053	UART4_TX_ISR_PC	UART4 TX 中断程序接口。
0x005B	UART4_RX_ISR_PC	UART4 RX 中断程序接口。
0x0063	UART5_TX_ISR_PC	UART5 TX 中断程序接口。
0x006B	UART5_RX_ISR_PC	UART5 RX 中断程序接口。
0x0083	UART3_ISR_PC	UART3 TX/RX 中断程序接口。
0x00F8	JTAG 接口使能	0xFFFF 将允许连接 JTAG 接口进行仿真调试；其它值将被禁止。
0x00FA	“DWINT5”	代码识别，非法值将导致 OS 8051 停止运行。
0x0100	应用代码开始	最大 63.75KB。

OS 8051 代码保存在 1Mbytes 片内 Flash 的 0x01:0000-0x01:FFFF 位置，上电复位后由系统加载到 RAM 中运行。代码只能通过 SD 接口或 UART1 接口（或 WiFi 等网络接口）写入到片内 Flash。



### 3.2.2 DGUS 变量存储器（256KBytes）

DGUS 变量存储器位宽为 32bit，每个地址对应 4Bytes 数据，其中：

（1）0x00:0000-0x00:7FFF 地址，对应 DGUS II 系统目前使用的 128Kbytes 变量存储器空间。

比如 0x1000 和 0x1001 两个 DGUS II 变量存储器，对应 OS 8051 DGUS 变量存储器的 0x0800 地址，其中 D3 对应 0x1000 变量高字节，D2 对应 0x1000 变量低字节，D1 对应 0x1001 变量高字节，D0 对应 0x1001 变量低字节。

（2）0x00:8000-0x00:FFFF 地址，DGUS II 系统目前没有使用，用户可以自行定义作为变量存储器使用。

（3）0xFF:0060-0xFF:006B 地址，CAN 通信接口的配置和收发 Buffer。

访问 DGUS 变量存储器使用下表的 SFR 寄存器接口。

SFR 名称	地址	说明
RAMMODE	0xF8	DGUS 变量存储器访问接口控制， <b>可以位寻址</b> ： .7 写 1 请求占用 DGUS 变量存储器读写，不占用时必须清零。 .6 APP_EN 写 1 启动 1 次读写，硬件执行后清零。 .5 APP_RW 1=读变量存储器 0=写变量存储器 .4 APP_ACK 硬件对 8051 占用变量存储器请求的应答，1=OK，0=BUSY，需要继续等待。 .3-.0 对应 DATA3:DATA0 写使能 1=对应 Byte 写入，0=对应 Byte 不写入。
ADR_H	0xF1	DGUS 变量存储器高 8 位地址，A23:A16。
ADR_M	0xF2	DGUS 变量存储器中 8 位地址，A15:A8。
ADR_L	0xF3	DGUS 变量存储器低 8 位地址，A7:A0。
ADR_INC	0xF4	DGUS 变量存储器读写后地址自动增加的增量，即 读写后 ADR_H:M:L=读写前 ADR_H:M:L+ADR_INC。
DATA3	0xFA	DGUS 变量数据接口，写入选择对应 RAMMODE. 3。
DATA2	0xFB	DGUS 变量数据接口，写入选择对应 RAMMODE. 2。
DATA1	0xFC	DGUS 变量数据接口，写入选择对应 RAMMODE. 1。
DATA0	0xFD	DGUS 变量数据接口，写入选择对应 RAMMODE. 0。

DGUS 变量存储器必须按照下面的流程来读写（**如果要在中断应用，主程序读写时必须关中断，不能嵌套**）：

（1）配置好地址和地址增量；

（2）设置 RAMMODE=0x8F（写）或 0xAF（读），检查 RAMMODE. 4=1 确认获得读写控制权；

（3）读写数据，读写结束后设置 RAMMODE=0x00。

**应用举例：对 0x0800 地址（对应 DGUS II 变量存储器地址 0x1000）读写 2 个双字。**

```

MOV    ADR_H, #00H           ;配置 DGUS 变量存储器地址
MOV    ADR_M, #08H
MOV    ADR_L, #00H
MOV    ADR_INC, #01H        ;配置地址增量
MOV    RAMMODE, #0AFH       ;启动读模式
JNB    APP_ACK, $           ;等待确认
MOV    R0, #TEST_BUF        ;演示读
MOV    R1, #2
RDVP:  SETB  APP_EN          ;启动 1 次读数据
        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               ;写模式，演示写
MOV    ADR_L, #00H          ;调整地址到 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               ;启动 1 次写数据。
        JB   APP_EN, $
INC    R0

```

DJNZ R1, WRVP  
MOV RAMMODE, #00H ;变量存储器读写结束

### 3.2.3 数据存储器（32KBytes）

T5L 的 OS 8051 CPU 有 32KBytes RAM 做为数据存储器，对应地址为 0x8000-0xFFFF，相关的接口 SFR 如下表。

SFR 名称	地 址	说 明
DPC	0x93	使用 DPTR 的 MOVX 指令操作后，DPTR 的变化模式： DPC=0x00: MOVX 指令操作后 DPTR 不变， <b>如果使用 C51 开发，必须配置为 0x00。</b> DPC=0x01: MOVX 指令操作后，DPTR=DPTR+1。 DPC=0x03: MOVX 指令操作后，DPTR=DPTR-1。
DPH	0x83	DPTR 数据指针。
DPL	0x82	

**0x0000-0x7FFF 地址空间禁止使用 MOVX 指令读写，可能会导致代码运行异常。**

T5L 的 MOVX 指令是 3 个指令周期(3T, 11.0592MHz 晶体下为 14.5nS)，加上 DPC 可以配置 DPTR 自动增量（或减量）模式，使得 T5L 读写数据存储器的速度比标准 8051 快很多，尤其是逆序存储读写的应用。

**应用举例：**

```
MOV DPC, #01H ;DPTR++
MOV DPTR, #8000H
MOVB A, @DPTR ;A=@8000
MOVB A, @DPTR ;A=@8001, 读取后 DPTR=8002
```

### 3.2.4 扩展 SFR 寄存器

扩展 SFR 寄存器使用 EXADR、EXDATA 寄存器接口来读写。

SFR 名称	地 址	说 明
EXADR	0xFE	扩展 SFR 地址，每次读写后地址自动加 1 指向下一个 SFR。
EXDATA	0xFF	扩展 SFR 的数据接口。

扩展 SFR 寄存器定义了数学运算单元（MDU）寄存器接口，以及额外提供用户 48 个数据存储器，定义如下表：

EXADR	定 义	说 明
0x00	MDU_A7	MDU A 寄存器（64bit）的最高位。
.....		
0x07	MDU_A0	MDU A 寄存器（64bit）的最低位。
0x08	MDU_B7	MDU B 寄存器（64bit）的最高位。
.....		
0x0F	MDU_B0	MDU B 寄存器（64bit）的最低位。
0x10	MDU_C7	MDU C 寄存器（64bit）的最高位。
.....		
0x17	MDU_C0	MDU C 寄存器（64bit）的最低位。
0x18	EXR0	第 1 个扩展数据寄存器。
0x19	EXR1	第 2 个扩展数据寄存器。
0x3F	EXR39	第 40 个扩展数据寄存器。

**中断应用中需要读写扩展 SFR 的话，主程序读写扩展 SFR 时必须关中断，不能嵌套。**

### 3.3 数学运算单元（MDU）

由于 8051 的计算能力有限，T5L 中扩展了硬件数学运算单元（MDU）来提升计算能力，OS 8051 应用软件可以访问的包括硬件 MAC 和硬件除法器，相关 SFR 定义如下表：

SFR 名称	地 址	说 明
MAC_CN	0xE5	MAC 硬件乘加器控制寄存器，定义如下： .7 MAC 使能，写 1 执行一次计算，硬件执行完清零。 .6 MAC 模式，1 为乘加器模式：C=A*B+C， 0 为乘法器模式：C=A*B。 .5 写 0。 .4 1=64bit MAC 0=32bit MAC (A3:0/B3:0/C7:0，注意 C 还是 64bit)。 .3 1=带符号数，0=无符号数。 .2-.0 写 0。 A、B、C 寄存器组是扩展 SFR 寄存器的 MDU_A、MDU_B、MDU_C 寄存器组。
DIV_CN	0xE6	DIV 硬件除法器控制寄存器（除法 C/A，商是 A，余数是 B），定义如下： .7 DIV 使能写 1 执行一次计算，硬件执行完清零。 .6 DIV 模式 1:四舍五入 0:不四舍五入。 .5-.4 未定义，写 0。 .3 1=带符号数，0=无符号数。 .2-.0 写 0。 A、B、C 寄存器组是扩展 SFR 寄存器的 MDU_A、MDU_B、MDU_C 寄存器组。

应用举例：32bit MAC 计算 0x1234\*0x5678-0x2000。

```

MOV     EXADR, #04H      ;写 A3:A0=0x 00 00 12 34
MOV     EXDATA, #00H
MOV     EXDATA, #00H
MOV     EXDATA, #12H
MOV     EXDATA, #34H
MOV     EXADR, #0CH      ;写 B3:B0=0x 00 00 56 78
MOV     EXDATA, #00H
MOV     EXDATA, #00H
MOV     EXDATA, #56H
MOV     EXDATA, #78H
MOV     EXADR, #10H      ;写 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, #0FFH
MOV     EXDATA, #0E0H
MOV     EXDATA, #00H
MOV     MAC_CN, #0C8H    ;32bit 整数 MAC
WTMAC:  MOV     A, MAC_CN
        JB      ACC. 7, WTMAC
MOV     EXADR, #10H      ;读取结果 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 定时器

T5L 的 OS 8051 有 3 个定时器 T0/T1/T2，其中 T0、T1 和标准 8051 一致，T2 只能工作在 16bit Autoload 模式。T0、T1 的时钟均为 CPU 主频的 1/12，T2 的时钟可以配置为 CPU 主频的 1/12 或 1/24。

相关的 SFR 如下表所示。

SFR 名称	地 址	说 明
TCON	0x88	T0、T1 控制寄存器，同标准 8051， <b>可以位寻址</b> 。 .7=TF1 .6=TR1 .5=TF0 .4=TR0 .3=IE1 .2=IT1 .1=IE0 .0=IT0 IT1、IT0 是外部中断触发模式选择：0=低电平触发 1=下跳沿触发。
TMOD	0x89	T0、T1 模式选择，同标准 8051。
TH0	0x8C	
TL0	0x8A	
TH1	0x8D	
TL1	0x8B	
T2CON	0xC8	T2 控制寄存器， <b>可以位寻址</b> 。 .7: 时钟分频选择，0=CPU 主频/12，1=CPU 主频/24。 .6-.4: 必须写 1。 .3-.1: 必须写 0。 .0: TR2，1=T2 运行 0=T2 关闭。
TH2	0xCD	T2 运行值，每次加计数溢出时自动装载 TH2=CRCH TL2=CRCL。
TL2	0xCC	
TRL2H	0xCB	T2 的重装载值=65536-T2 定时器间隔 (uS) *T2 时钟频率 (MHz)。 CPU 主频=晶体频率*56/3，相应的 CPU 主频/12=晶体频率*14/9、CPU 主频/24=晶体频率*7/9。 比如，CPU 主频 206.4384MHz，T2 选择 1/12 分频，1ms 定时器间隔的设定值为 48333 (0xBCCD)。
TRL2L	0xCA	

定时器中断的相关设置如下：

中断类型	程序入口地址	触发标记	中断使能控制	备注
T0 中断	0x000B	TF0 (TCON.5)	IEN0.1	中断响应时自动清 TF0。
T1 中断	0x001B	TF1 (TCON.7)	IEN0.3	中断响应时自动清 TF0。
T2 中断	0x002B	TF2 (IRCON.6)	IEN0.5	中断响应后需要软件清除 TF2，不然会再次触发中断。

**应用举例：**11.0592MHz 晶体下（对应 CPU 主频 206.4384MHz）设置 T2 1ms 中断在 P1.0 输出 500Hz 方波。

```

ORG      002BH          ;T2 中断程序入口
LJMP     T2INT

T2INT:   CLR      TF2          ;T2 中断程序
        CPL      P1.0
        RETI

;初始化 T2 相关 SFR
MOV      CRCH, #HIGH(48333)   ;1ms 定时器
MOV      CRCL, #LOW(48333)
MOV      T2CON, #71H          ;TR2=1
ORL      IEN0, #20H          ;ET2=1 开启 T2 中断

```

### 3.5 看门狗定时器 (WDT)

WDT 计数溢出时间保存在 0xFF001B 变量地址，WDT 复位时间=设置值/CPU 主频 (Hz)，单位是秒。

WDT 操作的相关参考代码如下：

```

ANL      MUX_SEL, #0FDH          ;设置溢出时间前必须关闭 WDT
MOV      ADR_H, #OFFH          ;配置 WDT 复位时间 1 秒=206438400 (0x0C 4E 00 00)
MOV      ADR_M, #00H
MOV      ADR_L, #1BH
MOV      RAMMODE, #8FH
JNB      APP_ACK, $
MOV      DATA3, #0CH
MOV      DATA2, #4EH
MOV      DATA1, #00H
MOV      DATA0, #00H
SETB     APP_EN
JB       APP_EN, $
MOV      RAMMODE, #00H
ORL      MUX_SEL, #02H          ;开启 WDT；一旦开启，就必须在 WDT 溢出前喂狗避免系统复位产生。

ORL      MUX_SEL, #01H          ;WDT 复位（喂狗）

```

### 3.6 I/O 口

T5L 的 OS 8051 有 3 个 8bit 并口 (P0-P2) 和一个 4bit 并口 (P3.0-P3.3)，一共 28 个 I/O 口。

P0 口的 P0.2-P0.7 和 UART 及 CAN 接口复用，可以通过 MUX\_SEL 寄存器选择复用功能或 I/O 功能。

所有 I/O 口的输入始终是开启的，输入浮空，无内部上拉或下拉。

I/O 口做为输出时，需要打开输出控制，并且输出驱动能力也可以配置。由于上电复位过程 I/O 口为输入模式，做为输出使用时，必须外部上拉或下拉确保上电过程有确定可靠的输出，不会因为 I/O 浮空误动作。

P3.0 是外部中断 0，P3.1 是外部中断 1 输入，可以通过 IT0、IT1 配置为低电平触发 (0) 或下跳沿触发 (1)。

除输出需要控制开关，输出强度及外设复用上电初始化配置后，I/O 的后续使用和标准 8051 一致。

I/O 相关的 SFR 如下表所示。

SFR 名称	地 址	说 明
P0	0x80	可以位寻址，同标准 8051。
P1	0x90	可以位寻址，同标准 8051。
P2	0xA0	可以位寻址，同标准 8051。
P3	0xB0	可以位寻址，同标准 8051，只有 P3.0-P3.3 有效。
P0MDOUT	0xB7	P0 口输出配置。 .7 0=P0.7 输出关闭 1=P0.7 输出开启 (推挽输出)。 .6 0=P0.6 输出关闭 1=P0.6 输出开启 (推挽输出)。 .5 0=P0.5 输出关闭 1=P0.5 输出开启 (推挽输出)。 .4 0=P0.4 输出关闭 1=P0.4 输出开启 (推挽输出)。 .3 0=P0.3 输出关闭 1=P0.3 输出开启 (推挽输出)。 .2 0=P0.2 输出关闭 1=P0.2 输出开启 (推挽输出)。 .1 0=P0.1 输出关闭 1=P0.1 输出开启 (推挽输出)。 .0 0=P0.0 输出关闭 1=P0.0 输出开启 (推挽输出)。
P1MDOUT	0xBC	P1 口输出配置。
P2MDOUT	0xBD	P2 口输出配置。
P3MDOUT	0xBE	P3 口输出配置。
MUX_SEL	0xC9	外设复用选择。 .7 1=CAN 接口引出到 P0.2、P0.3，0=CAN 接口不引出，为 I/O 口。 .6 1=UART2 接口引出到 P0.4、P0.5，0=UART2 接口不引出，为 I/O 口。 .5 1=UART3 接口引出到 P0.6、P0.7，0=UART3 接口不引出，为 I/O 口。 .4-.2 保留 .1 WDT 控制，1=开启 0=关闭。 .0 WDT 喂狗，1=喂狗一次 (WDT 计数值归零，看门狗的溢出时间为 1 秒)。
PORTDRV	0xF9	I/O 口输出模式的驱动能力配置，0x00=4mA 0x01=8mA (推荐值) 0x02=16mA 0x03=32mA。

I/O 口外部中断的相关设置如下：

中断类型	程序入口地址	触发标记	中断使能控制	备注
EX0 中断	0x0003	IE0 (TCON.1)	IEN0.0	中断响应时自动清 IE0，对应 P3.0。 IT0 (TCON.0)=0 低电平触发中断，IT0=1 下跳沿触发中断。
EX1 中断	0x0013	IE1 (TCON.3)	IEN0.2	中断响应时自动清 IE1，对应 P3.1。 IT1 (TCON.2)=低电平触发中断，IT1=1 下跳沿触发中断。

### 3.7 UART 通信接口

#### 3.7.1 UART2 接口

UART2 相关 SFR 控制接口如下表所示。

SFR 名称	地 址	说 明
MUX_SEL	0xC9	.6 1=UART2 接口引出到 P0.4、P0.5，0=UART2 接口不引出，为 I/O 口。
SCON0	0x98	UART2 控制接口，同标准 8051，可以位寻址。 .7=SM0 .6=SM1 .5=SM2 (多机通信位) .4=REN0 .3=TB80 .2=RB80 .1=TI0 .0=RI0。
SBUF0	0x99	UART2 收发数据接口。
ADCON	0xD8	波特率发生器选择，0x00=T1 定时器 (标准 8051)，0x80=使用 SRELOH:L。
PCON	0x87	.7=SMOD 波特率倍频选择，0=不倍频，1=倍频。
SRELOH	0xBA	当 ADCON=0x80 时，使用 SRELOH:L 来设置波特率，不占用 T1。 SMOD=0 SREOH:L=1024-CPU 主频 / (64*波特率)；
SRELOL	0xAA	SMOD=1 SREOH:L=1024-CPU 主频 / (32*波特率)。 CPU 主频=晶体频率*56/3，11.0592MHz 晶体对应 206.4384MHz 主频。

UART2 中断的相关设置如下：

中断类型	程序入口地址	触发标记	中断使能控制	备注
UART2 中断	0x0023	RI0 (SCON0.0) TI0 (SCON0.1)	IEN0.4	中断处理后需要软件清除相应的中断触发标记。

### 3.7.2 UART3 接口

UART3 相关 SFR 控制接口如下表所示。

SFR 名称	地 址	说 明
MUX_SEL	0xC9	.5 1=UART3 接口引出到 P0.6、P0.7， 0=UART3 接口不引出，为 IO 口。
SCON1	0x9B	UART3 控制接口，不能位寻址。 .7 0=9bit UART 1=8bit UART; .6 未定义; .5=SM2（多机通信位） .4=REN .3=TB8 .2=RB8 .1=TI .0=RI 。 清除 SCON1 位标记时需要连续写两次，比如 ANL SCON1, #0FEH ANL SCON1, #0FEH
SBUF1	0x9C	UART3 收发数据接口。
SREL1H	0xBB	UART3 波特率设置（CPU 主频=晶体频率*56/3，11.0592MHz 晶体对应 206.4384MHz 主频。）： SREL1H:L=1024-CPU 主频/（32*波特率）。
SREL1L	0x9D	

UART3 中断的相关设置如下：

中断类型	程序入口地址	触发标记	中断使能控制	备注
UART3 中断	0x0083	SCON1.0, SCON1.1	IEN2.0	中断处理后需要软件清除相应的中断触发标记。

### 3.7.3 UART4 接口

UART4 相关 SFR 控制接口如下表所示。

SFR 名称	地 址	说 明
SCON2T	0x96	UART4 发送控制： .7 UART4 发送使能，0=闭，1=开启。 .6 0=8bit Mode, 1=9bit Mode。 .5 TB8, 9bit 模式下发送的第 9bit。 .4-.1 写 0。 .0 TI, 发送标记，开始发送停止位时置位。
SCON2R	0x97	UART4 接收控制： .7 UART4 接收使能，0=闭，1=开启。 .6 写 0。 .5 RB8, 9bit 模式下，接收的第 9bit。 .4-.1 写 0。 .0 RI, 接收标记，接收到有效停止位，停止位结束时置位。
SBUF2_TX	0x9E	UART4 发送数据接口。
SBUF2_RX	0x9F	UART4 接收数据接口。
BODE2_DIV_H	0xD9	UART4 波特率设置： BODE2_DIV_H:L=CPU 主频/（8*波特率）。
BODE2_DIV_L	0xE7	

UART4 中断的相关设置如下：

中断类型	程序入口地址	触发标记	中断使能控制	备注
UART4 发送中断	0x0053	SCON2T.0	IEN1.2	中断处理后需要软件清除相应的中断触发标记。
UART4 接收中断	0x005B	SCON2R.0	IEN1.3	中断处理后需要软件清除相应的中断触发标记。

### 3.7.4 UART5 接口

UART5 相关 SFR 控制接口如下表所示。

SFR 名称	地 址	说 明
SCON3T	0xA7	UART5 发送控制： .7 UART5 发送使能，0=闭，1=开启。 .6 0=8bit Mode, 1=9bit Mode。 .5 TB8, 9bit 模式下发送的第 9bit。 .4-.1 写 0。 .0 TI, 发送标记，开始发送停止位时置位。
SCON3R	0xAB	UART5 接收控制： .7 UART5 接收使能，0=闭，1=开启。 .6 写 0。 .5 RB8, 9bit 模式下，接收的第 9bit。 .4-.1 写 0。 .0 RI, 接收标记，接收到有效停止位，停止位结束时置位。
SBUF3_TX	0xAC	UART5 发送数据接口。
SBUF3_RX	0xAD	UART5 接收数据接口。
BODE3_DIV_H	0xAE	UART5 波特率设置： BODE3_DIV_H:L=CPU 主频/（8*波特率）。
BODE3_DIV_L	0xAF	



UART5 中断的相关设置如下:

中断类型	程序入口地址	触发标记	中断使能控制	备注
UART5 发送中断	0x0063	SCON3T. 0	IEN1. 4	中断处理后需要软件清除相应的中断触发标记。
UART5 接收中断	0x006B	SCON3R. 0	IEN1. 5	中断处理后需要软件清除相应的中断触发标记。

### 3.8 CAN 通信接口

CAN 接口相关的 SFR 如下表:

SFR 名称	地 址	说 明
MUX_SEL	0xC9	. 7 1=CAN 接口引出到 P0. 2、P0. 3, 0=CAN 接口不引出, 为 I/O 口。
CAN_CR	0x8F	CAN 接口控制寄存器。 . 7 CAN 接口使能, 1=开启 0=关闭。 . 6 CAN 接口模式, 1=软件复位 0=正常工作。 . 5 写 1 配置一次 CAN 接口配置数据 (0xFF:0060-0xFF:0062), 硬件处理后清零。 . 4 设置速度模式, 1=1 采样 0=3 采样。 . 3 设置滤波器模式, 1=double 0=single。 . 2 写 1 启动一次发送, 硬件处理 (发送成功、仲裁失败、出现 EI (CAN_IR. 3) 错误、软件复位) 后清零。 . 1-0 未定义, 写 0。
CAN_IR	0x91	CAN 接口中断状态寄存器。 . 7 RF_IF, 远程帧接收中断标记, 硬件置位, 需要软件清零。 . 6 CAN_RX_IF, CAN 接收完成中断标记, 硬件置位, 需要软件清零。置位期间, 硬件不能再更新数据。 . 5 CAN_TX_IF, CAN 发送成功中断标记, 硬件置位, 需要软件清零。 . 4 OI, 接收溢出标记, 硬件置位, 需要软件清零。 . 3 EI, 错误标记, CAN_ET[4:0] 有一种错误产生时, 该位拉高, 需要软件清零。 . 2 JI, 发送仲裁标记, 1=发送仲裁失败, 0=发送仲裁成功。 . 1-0 未定义。
CAN_ET	0xE8	CAN 接口错误类型寄存器, 硬件置位, 需要软件清零。 <b>可以位寻址。</b> . 7 节点挂起标识。 . 6 主动错误标识。 . 5 被动错误标识。 . 4 CRC 校验错误标识。 . 3 应答错误标识。 . 2 格式错误标识。 . 1 位填充错误标识。 . 0 位错误标识。

CAN 通信接口按照 DGUS 变量空间来范围, 相关变量定义如下表:

地 址	位置	长 度	定 义	说 明
0xFF:0060	D3	1	BRP	BRP: 波特率分频寄存器。
	D2	1	BTR0	BTR0: [7:5] 为同步跳转宽度 sjw, [3:0] prop 传播时间段 T0。
	D1	1	BTR1	BTR1: [7:4] 为相位缓冲段 T1, [3:0] 相位缓冲段 2, T2。
	D0	1	未定义, 写 0。	<b>T0+T1+T2 = CPU 主频 / (波特率 * (BRP*2+1)) - 4。</b>
0xFF:0061	D3:D0	4	ACR3:0	验收代码寄存器。
0xFF:0062	D3:D0	4	AMR3:0	验收代码寄存器。
0xFF:0063	D3	1	RXERR	接收错误计数寄存器。
	D2	1	TXERR	发送错误计数寄存器。
	D1:D0	2	未定义	写 0。
0xFF:0064	D3	1	CAN_TX_BUFFER	[7] IDE, [6] RTR, [3:0]—DLC, 帧数据长度。
	D2:D0	3	未定义	
0xFF:0065			ID	ID, 扩展帧时 29bit 有效, 标准帧时 11bit 有效。
	D3	1		ID 第一个字节, 标准帧与扩展帧。
	D2	1		ID 第二个字节, [7:5] 为标准帧的高 3bit, 扩展帧第 2 字节。
	D1	1		ID 第三个字节, 标准帧无效, 扩展帧第 3 字节。
	D0	1		ID 第四个字节, 标准帧无效, [7:3]—扩展帧的高 5bit。
0xFF:0066	D3:D0	4	数据	发送数据, DATA1—DATA4。
0xFF:0067	D3:D0	4	数据	发送数据, DATA5—DATA8。
0xFF:0068	D3	1	CAN_RX_BUFFER	[7] IDE, [6] RTR, [3:0]—DLC, 帧数据长度。
	D2:D0	3	未定义	
0xFF:0069			ID	ID, 扩展帧时 29bit 有效, 标准帧时 11bit 有效。
	D3	1		ID 第一个字节, 标准帧与扩展帧。
	D2	1		ID 第二个字节, [7:5] 为标准帧的高 3bit, 扩展帧第 2 字节。
	D1	1		ID 第三个字节, 标准帧无效, 扩展帧第 3 字节。
	D0	1		ID 第四个字节, 标准帧无效, [7:3]—扩展帧的高 5bit。

0xFF:006A	D3:D0	4	数据	接收数据，DATA1-DATA4。
0xFF:006B	D3:D0	4	数据	接收数据，DATA5-DATA8。

CAN 接口中断的相关设置如下：

中断类型	程序入口地址	触发标记	中断使能控制	备注
CAN 接口中断	0x004B	CAN_IR	IEN1.1	中断处理后需要软件清除相应的中断触发标记。

## 3.9 中断系统

### 3.9.1 中断控制 SFR

T5L OS CPU 一共有 12 个中断，相关控制 SFR 列表如下：

SFR 名称	地 址	说 明
IEN0	0xA8	中断使能控制器 0， <b>可以位寻址</b> 。 .7 EA 中断总控制位；0=所有中断关闭 1=中断是否打开由每个中断的控制位控制。 .6 必须写 0。 .5 ET2 T2 定时器中断使能控制位。 .4 ES0 USRT2 中断使能控制位。 .3 ET1 T1 定时器中断使能控制位。 .2 EX1 外部中断 1 (P3.1 引脚) 中断使能控制位。 .1 ET0 T0 定时器中断使能控制位。 .0 EX0 外部中断 0 (P3.0 引脚) 中断使能控制位。
IEN1	0xB8	中断使能控制器 1， <b>可以位寻址</b> 。 .7 -.6 必须写 0。 .5 ES3R UART5 接收中断使能控制位。 .4 ES3T UART5 发送中断使能控制位。 .3 ES2R UART4 接收中断使能控制位。 .2 ES2T UART4 发送中断使能控制位。 .1 ECAN CAN 通信中断使能控制位。 .0 必须写 0。
IEN2	0x9A	中断使能控制器 2。 .7 -.1 必须写 0。 .0 ES1 USRT3 中断使能控制位。。
IEN3	0xD1	中断使能控制器 3，必须写 0x00。
IP0	0xA9	中断优先级控制器 0。
IP1	0xB9	中断优先级控制器 1。

### 3.9.2 中断优先级

T5L OS CPU 的中断优先级按照如下规则来处理：

(1) 12 个中断分成 6 组，每组 2 个中断，组内的优先级固定，比如同组的外部中断 0 优先级比 UART3 中断优先级高，如下表所示。

分组	IP0 对应	IP1 对应	中断对应	
			高优先级	低优先级
G0	.0	.0	外部中断 0	UART3 中断
G1	.1	.1	T0 定时器中断	CAN 通信中断
G2	.2	.2	外部中断 1	UART4 发送中断
G3	.3	.3	T1 定时器中断	UART4 接收中断
G4	.4	.4	UART2 中断	UART5 发送中断
G5	.5	.5	T2 定时器中断	UART5 接收中断

(2) 6 组之间的优先级有 4 级，可以通过 IP0 和 IP1 的对应位按照下表来配置。

组间优先级	IP1 对应位	IP0 对应位
3 (最高)	1	1
2	1	0
1	0	1
0 (最低)	0	0

比如要把 G5 组的 T2 定时器中断、UART5 接收中断优先级设置为最高，可以设置 IP1=0x20，IP0=0x20。

(3) 如果配置的优先级都相同时 (IP1=0x00 IP1=0x00)，那么按照 G0 组优先级最高，G5 组优先级最低来处理。配置相同时的中断优先级权值表如下：

权 值	11	10	9	8	7	6	5	4	3	2	1	0
优先级	最高											最低
中 断	EX0	UART3	ET0	CAN	EX1	UART4-TX	ET1	UART4-RX	UART2	UART5-TX	ET2	UART5-RX

(4) 低优先级中断中可嵌套高优先级中断 (即权值小的中断可以被权值大的中断打断)，最多嵌套四级。



## 特别提示:

T5L OS CPU 速度快 (1uS 平均可执行 130~150 条指令), 中断执行时间短, 实时性已经非常高, 不推荐用户使用中断嵌套导致程序架构复杂, 在每个中断服务程序执行时直接关闭中断 (EA=0), 退出时再开启中断 (EA=1)。

北京迪文科技有限公司技术文档

### 3.10 T5L ASIC 的 8051 指令集

指令格式	指令长度	指令周期	指令格式	指令长度	指令周期
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

CPU 主频=晶体频率\*56/3, 11.0592MHz 晶体对应 206.4384MHz 主频, 相应的 1 个指令周期 (1T) 为 4.844nS。  
比如,

11.0592MHz 晶体下, 下面代码将在 P3.3 IO 口上输出 206.4384/((2+2)\*2)=25.8048MHz 的方波。

```

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

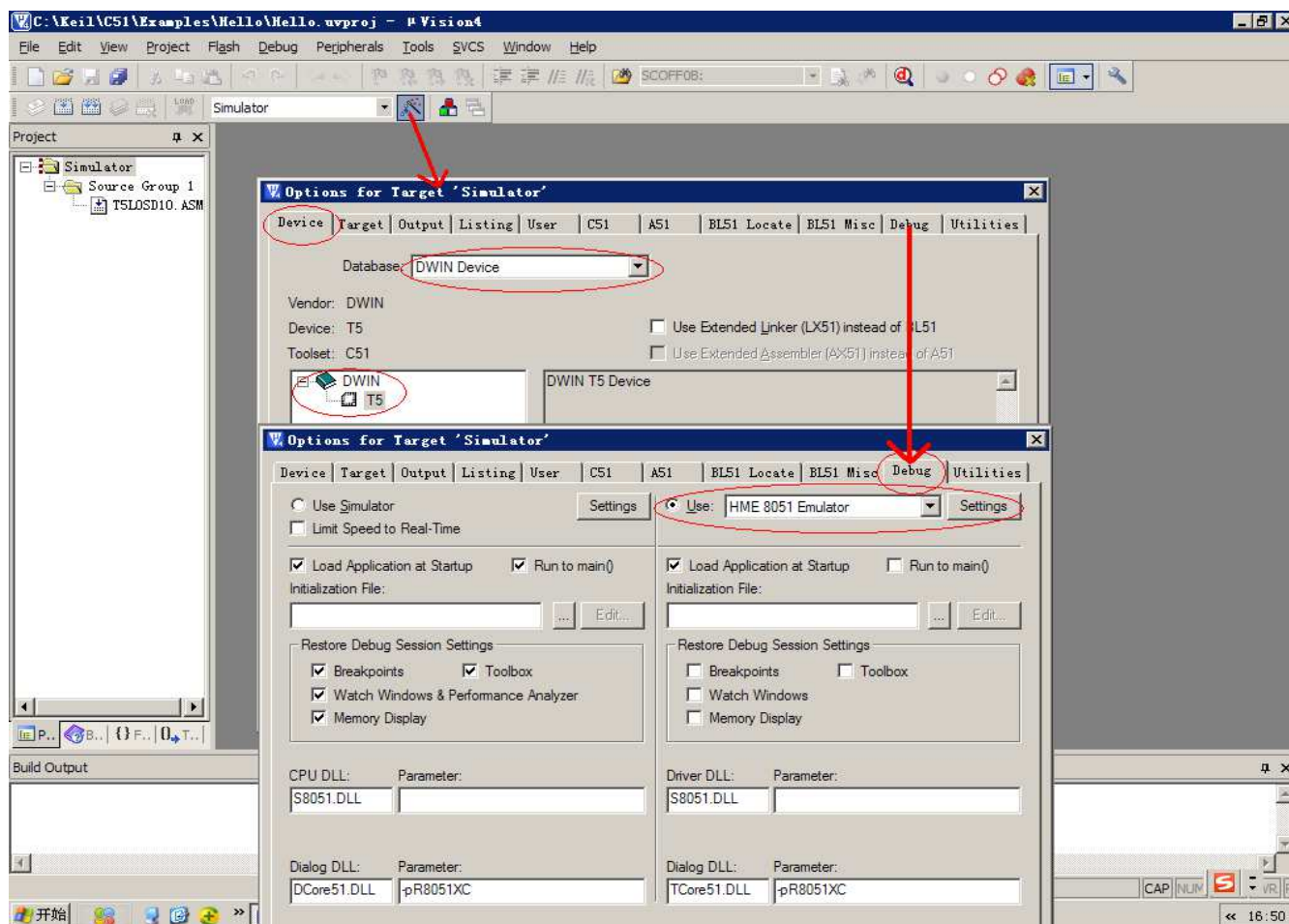
```

## 4 仿真调试

借助 HME05 仿真器（需要安装相应的 USB 驱动），用户可以连接 T5L 的 JTAG 接口，在 Keil 开发环境下进行代码 IAP 调试和仿真运行。

仿真调试中要注意以下事项：

- (1) JTAG 接口必须选择到 OS CPU，并且必须是 JTAG 模式，即 OS/GUI (PIN#32)=1 JTAGS (PIN#34)=0。使用 4.3 寸评估板仿真时，JTAG 接口边上的跳线焊盘为断开状态。
- (2) 安装好 AGDI 驱动，使得 Keil 支持 T5 和 HME05 仿真器，安装好之后按照下图选择和配置。安装好之后，把 T5L OS CPU 的头文件 (\*.INC 或 \*.h) Copy 到 KEIL/C51/INC/DWIN 目录下。



- (3) 设置断点读取数据存储器 (XDATA) 内容前，必须确保 DPC=0x00，不然数据会错位。
- (4) 仿真前，必须确保烧录在 T5L Flash 中的 OS CPU 代码 0x00F8 位置开始是 0xFF FF 44 57 49 4E 54 35，不然 OS CPU 的 JTAG 接口被禁止，连接不上 HME05 仿真器。
- (5) 用户在标准 C 或其他 8051 平台代码移植时，注意编译时 SFR 头文件加载要选择 T5L 相应的 .INC 或 .H 文件，如果客户代码中的 SFR 定义和 T5L 定义不同时，可以修改代码或 T5L 的 SFR 头文件保持一致。
- (6) HME05 是通过把代码下载到 T5L OS CPU 的 Code RAM 来实现硬件仿真的，代码并没有烧录到片内的 Flash 上。把代码烧录到芯片内部，需要通过 SD 卡接口或者 UART1 调试接口，SD 接口烧录时 T5L 底层软件会自动把 OS 代码的 0x00F8 位置修改为 0x0000（禁止 JTAG 接口） 44 57 49 4E 54 35。

T5L OS CPU 采用标准 8051 架构，除 SFR 和扩展外设访问略有差异外，指令集完全相同，移植用户原有 8051 代码时，注意好以下几个方面就可以快速完成了：

- (1) 按照硬件设计，在复位后用迪文提供的 startup.A51 (C51 启动代码) 或 initcpu() 汇编程序简单修改、配置好 T5L 特有的 SFR 和参数设置。

典型的差异在 IO、定时器主频、波特率、中断、WDT 之类。

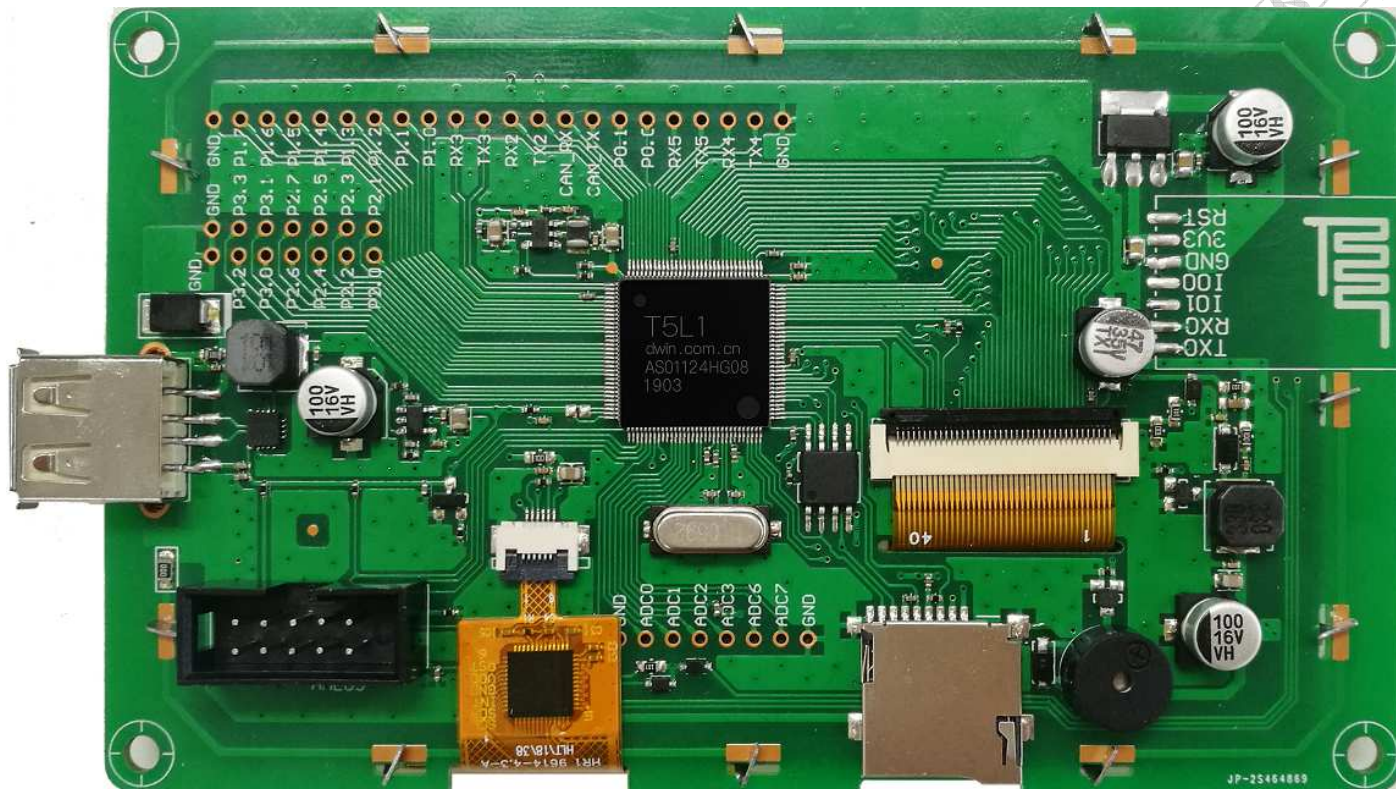
- (2) T5L 的 IO 输出模式是可控的，在输入、输出模式之间切换时，要相应配置 PxMDOUT 寄存器，不然会出错。
- (3) 把中断嵌套关闭，每个中断服务程序进入时 EA=0，退出时 EA=1。
- (4) 代码中用到片外 RAM (XRAM) 做数据存储时，注意 T5L 的 32KB 数据 RAM 起始地址是从 0x8000 开始。
- (5) 代码 0x00F8 位置加上 0xFFFF（或 0x0000 禁止 JTAG 接口） 44 57 49 4E 54 35。
- (6) 用 MDU 硬件运算加速优化原来代码的算法；把 UI 和互联网（比如通过迪文 WiFi 模块接入迪文云）功能挪到 DGUS II 平台上实现，用户代码通过简单的读写 DGUS 变量空间来处理，大幅度提升产品性能，提升研发效率。



## 5 EK043 评估板

EK043 评估板采用 T5L1 驱动 4.3 寸 480\*272 TFT 屏，并配套电容触摸屏，其主要特点包括：

- (1) 11.0592MHz 晶体，206.4384MHz 运行主频，支持 HME05 仿真器。
- (2) 外接 16MB SPI Flash，运行标准 DGUS II UI；
- (3) 使用 USB 接口供电和调试；
- (4) 可以通过 USB 接口下载 OS CPU 8051 程序并自动硬件复位运行；
- (5) 可以通过 USB 接口读写 DGUS 变量存储器和下载图片、字库，方便 DGUS 调试；
- (6) 保留迪文 WiFi 模块装配位置，方便接入迪文云（WiFi 模块和 USB 都占用 UART1 接口，波特率 921600bps）；
- (7) 用 2.54mm 间距通孔焊盘引出 T5L OS CPU 的所有接口，并进行了标识；
- (8) 用 2.54mm 间距通孔焊盘引出用户可以使用的 6 路 AD，ADC0-ADC3，ADC6-ADC7，并进行标识。







## 附录 1 修订记录

日期	修订内容	文档版本
2019.03.15	首次发布。	V1.0
2020.06.22	修正了文档的一些谬误。 增加了对 OS 核 WDT 复位时间设置的说明。	V1.1

使用本文档或迪文产品过程中如存在任何疑问，或欲了解更多迪文产品最新信息，请及时与我们联系：

400 免费电话：400 018 9008

企业 QQ 和微信：400 018 9008

企业 mail：dwinhmi@dwin.com.cn

感谢大家一直以来对迪文的支持，您的支持是我们进步的动力！

谢谢大家！