

实验二 基于 ARM 处理器的 I/O 接口实验

1. W90P710 芯片简介

W90P710 是华邦电子开发的一款基于 ARM7TDMI 微处理器内核的 SoC 芯片。ARM7TDMI 微处理器内核是由 ARM 公司设计的 32 位 RISC 架构微处理器。W90P710 芯片采用 0.18um 工艺生产，典型工作频率可达 80MHz，采用 176 引脚 LQPF 封装。2008 年，华邦电子分割逻辑产品线后成立新唐科技（Nuvoton Technology Corp.）。之后，W90P710 也被称作 NUC710。虽然 W90P710 是一款已逐渐退出市场的芯片产品，但却是一个学习 ARM7 架构与编程的好平台。

W90P710 主要的特性与集成外设接口功能如下：

- 片上集成有 4KBytes I-Cache 和 4KBytes D-Cache，且该存储器可根据实际需求被配置成为片上 RAM 使用。
- 片上集成二个带有 8 分频的 24 位定时/计数器、一个看门狗定时器、最多可达 71 个可编程 I/O 接口、四组独立的 UART 串行接口、一个双通道的 GDMA 接口和一个高级中断控制器。
- 片上集成一个可配置的 LCD 控制器，可支持 TFT 型和低成本的 STN 型 LCD 模块。
- 片上集成一个 10M/100Mb 自适应的以太网 MAC 层控制器，可实现对以太网通信的支持，降低网络通信的系统成本。

- 片上集成 USB1.1 主机和设备控制器各一个，二个智能卡主控器。
- 片上集成一个 AC97/I²S 控制器，一个 SD/SDIO 卡主控器和一个 PS/2 键盘控制器。
- 片上集成一个可灵活配置的外部总线接口（External Bus Interface, EBI）接口。通过 EBI 接口可连接 SDRAM、ROM、SRAM、Flash 存储器以及其他各类 I/O 设备。

关于 W90P710 芯片更详细的资料请参考附件数据手册中的 W90P710CDG 文件。

2. W90P710 评估板使用说明

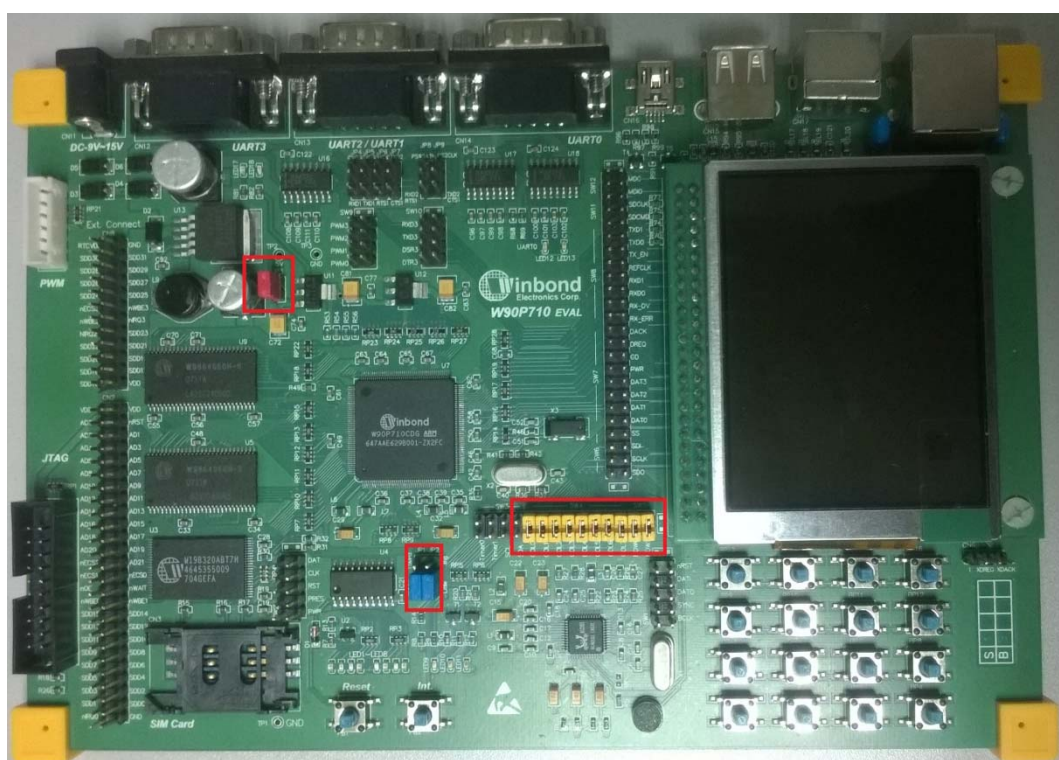


图 1 华邦 W90P710 评估板实拍图

本次实验我们选用 W90P710 评估板作为硬件实验平台，图 1 为

W90P710 评估板的实拍图。整块评估板可以划分为核心功能模块和应用功能模块两部分。其中，核心功能模块主要是围绕 W90P710 芯片扩展的最小系统，而应用功能模块是通过 W90P710 芯片的 GPIO 和 EBI 接口扩展的应用外设。

核心功能模块以 W90P710 处理器芯片为核心，外扩 16MB SDRAM、4MB Flash、1 个 JTAG 调试口、1 个 UART 调试串口、1 个 USB 主机接口和 1 个设备接口，组成 W90P710 评估板的最小系统。评估板核心功能模块的框图如下图 2 所示，主要由以下几个部分组成：

- **MCU：**基于 ARM7TDMI 内核的华邦 W90P710 芯片，最高工作频率可达到 80M
- **启动 FLASH：**评估板最大可支持 8MB (4Mx16) FLASH，板载 1 片 2MBx16 FLASH
- **SDRAM：**评估板最大可支持 32MB (8Mx16x2 banks) SDRAM，板载共有 16MB
- **USB：**一个 USB1.1 主接口和一个 USB1.1 设备接口
- **UART：**一个只有 TX/RX 信号的 UART0 端口用于调试工作台通信
- **JTAG：**14 管脚 JTAG 作为调试接口

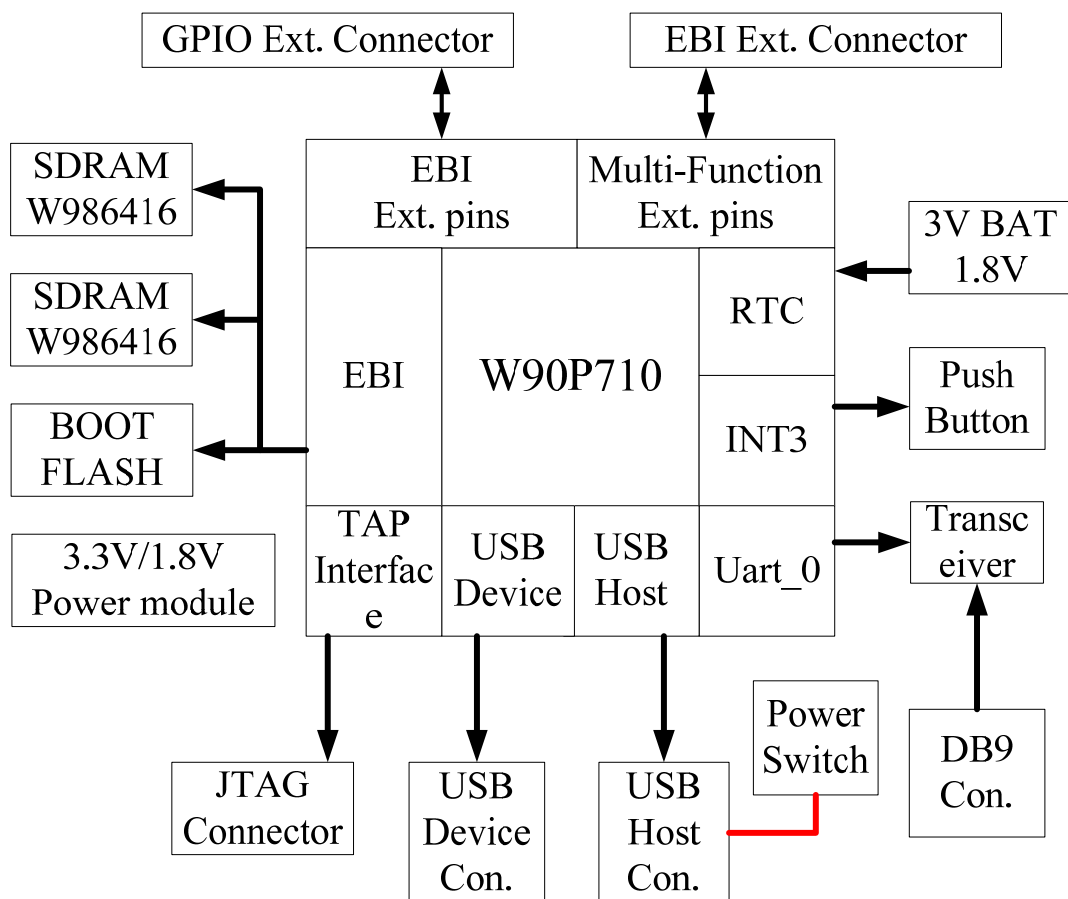


图 2 华邦 W90P710 评估板核心功能模块框图

如图 3 所示，应用功能模块通过 GPIO 和 EBI 总线扩展了丰富的应用外设接口，主要包括：

- LCD：通过 W90P710 的 LCD 控制器接口支持 AUO 960*240 TFT 液晶显示屏
- SD 卡：通过 W90P710 的 SD 接口支持 4 位模式 SD 卡
- I2C：板载一片 64KB EEPROM，连接至 W90P710 的 I2C 接口
- 智能卡：两个由 W90P710 的 SMC 控制器支持的 7816-3 端口
- 三个由 W90P710 支持的 UART 端口（公型）
- PS/2：一个由 W90P710 的 PS/2 接口支持的 PS/2 端口

- **RTC**: 由 W90P710 的实时钟控制器支持的实时时钟
- **矩阵键盘**: 由 W90P710 的 KPI 控制器支持的 2 行 * 8 列键盘
- **SPI**: 一个由 W90P710 的 SPI 总线支持的 32M -bit 串行 FLASH
- **Audio**: 由通过音频解码器 ALC-203 和 W90P710 的 AC-97 端口支持的音频输入输出功能
- **LED 调试显示**: 八个连接至 W90P710 的 EBI bank0 端口的 LED
- **PWM**: 由 W90P710 的 PWM 控制器支持的四通道 PWM 连接器
- **超时 LED 显示**: 两个连接至 W90P710 定时器输出的 LED

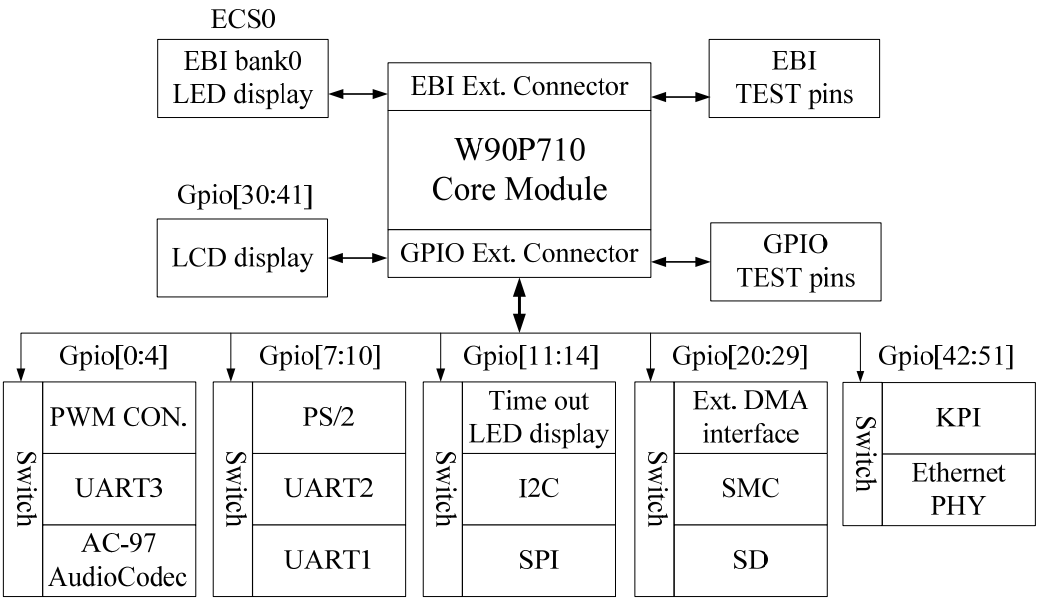


图 3 华邦 W90P710 评估板应用功能模块框图

在应用功能模块中，W90P710 除了地址总线引脚和数据总线引脚之外，各个功能模块通过 GPIO 的引脚来实现相应的功能。由于处理器芯片的引脚总数有限，所以很多部件都是共享某些特定的引脚。如 GPIO[0:4] 是由 GPIO、PWM、UART3 以及 AC-97 来共享，但是

某一个特定的时刻，只能有一种模块使用该引脚。W90P710 芯片通过相应的寄存器来设定当前时刻由那个模块来使用该 GPIO 引脚。

注意：在使用 W90P710 评估板之前，需要对其进行跳线设置来配置相应的功能。本次实验中所涉及到的跳线设置在图 1 中已经用矩形框标出，请按照图示设置。实验中，若要重启系统，请直接按下 W90P710 评估板下方的 Reset 按钮。关于 W90P710 评估板的详细介绍和使用说明请参考附件 2：W90P710CDG 芯片产品数据手册。

3. MDK-ARM 的目标调试功能

在 W90P710 评估板上开发和调试程序需要使用 MDK-ARM 的调试器。在进行实验一时，我们主要是使用了 MDK-ARM 调试器的仿真功能。本次实验，我们学习使用调试器的目标调试功能。使用 MDK-ARM 的目标调试器调试评估板，需要调试适配器的配合。实验中，我们使用 ULINK2 调试适配器。整个系统的连接如下图.4 所示，ULINK2 的 USB 端连接在 PC 机上，将另一端的连接在 W90P710 的 JTAG 端口上，等待 PC 机上的驱动自动安装完成之后，ULINK2 就可以正常工作了。

ULINK2 调试适配器可以辅助完成：（1）下载目标程序；（2）查看存储器与寄存器；（3）控制程序单步执行；（4）插入多个断点；（5）实时执行程序；（6）烧写 Flash 存储器。利用 ULINK2 调试适

配器，我们可以像使用仿真器一样实现对在评估板运行程序的控制与状态查看。关于对 ULINK2 调试的设置，将在后续的实验过程中学习。

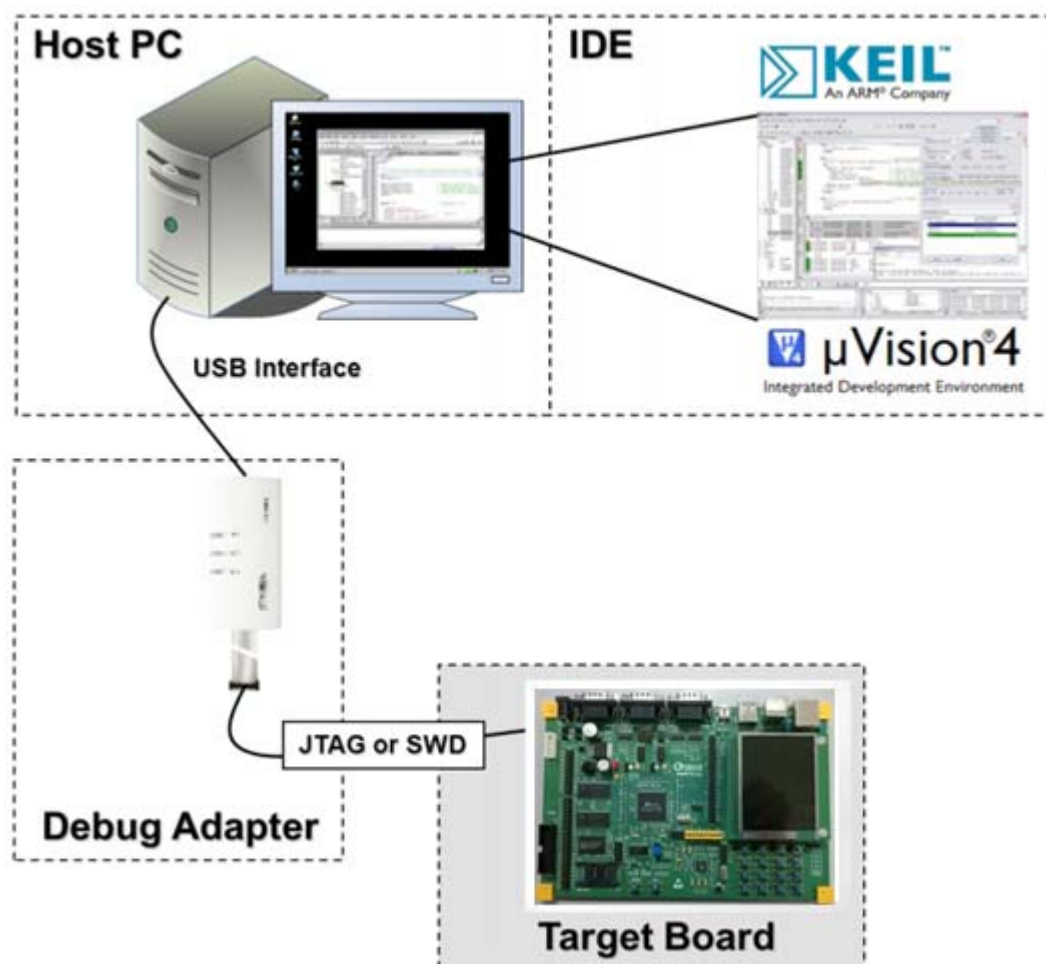


图 4 调试环境连接图

4. LED 灯闪烁实验

【实验目的】

学会使用 Keil MDK-ARM 进行 ARM 内核应用处理器的 I/O 接口编程与调试方法。

【实验设备】

1) 硬件：W90P710 实验系统、PC 机、ULINK2 仿真器

2) 软件：PC 机操作系统、Keil MDK-ARM 4.22

【实验内容及要求】

在 W90P710 评估板平台编写程序，控制板载八个发光二极管逐一循环点亮。

【实验原理】

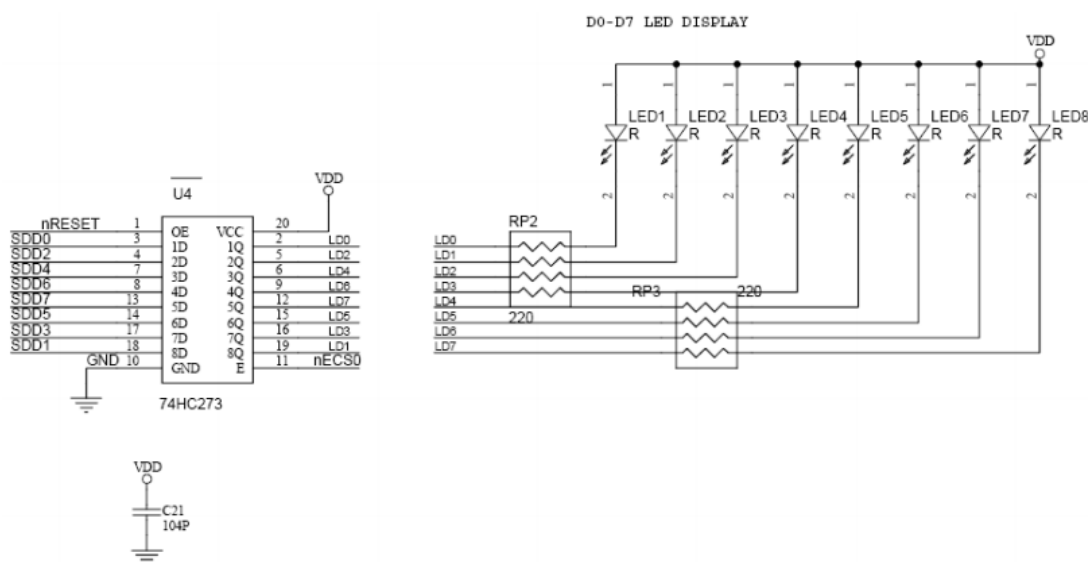


图 5 LED 控制电路原理图

W90P710 评估实验板上 LED 灯硬件电路原理图如图 5 所示。

CPU 数据总线的低 8 位连接到锁存器 74HC273 的信号输入端，CPU 外部总线接口（EBI）的 nECS0 片选信号连接到锁存器的锁存控制信号输入端。这样当 CPU 访问 BANK0 地址空间内的任一地址时，数据总线低 8 位的数据将被锁存器锁存后输出，作为控制 8 个 LED 灯的发光状态的控制信号。

本实验中，设置 CPU 外部总线 BANK0 的基地址为 0x78000000。

因此，如向该基地址写数据 0x00 时，8 个 LED 灯都将点亮；而写数据 0xFF 时，则 8 个 LED 灯都将熄灭。有关应用处理器相关寄存器的详细设置，请参考附件 2：W90P710CDG 芯片产品数据手册。有关评估板详细的电路连接与设置，请参考附件 3：W90P710 评估板硬件应用说明。

【实验步骤】

- 1) 打开 Keil，点击 Project->New uVision Project，新建名称为 led 的工程，并将工程保存在实验目录的新建 LEDBlinky 文件夹中。同时，将实验提供文件中的 Ext_SDRAM.ini 文件拷贝到 LEDBlinky 文件夹下。
- 2) 在弹出选择 ARM 芯片窗口，选择 Nuvoton 系列下的 NUC710 芯片，点击“OK”，提示“将 NUC710 启动代码添加到工程”，请选择“是”。
- 3) 点击 File->New，新建名称为 led.c 的文件，并将 led.c 文件保存在 LEDBlinky 文件夹中。将 LED 灯闪烁实验的 C 语言程序源代码通过编辑器输入到 led.c 文件中，并保存。C 语言源代码如下所示。
- 4) 右键选择 Project 窗口下的 Target1->Source Group1，在弹出的窗口中选择 Add Files to Group，将 led.c 文件添加到工程中。

```

//LED 灯闪烁程序 C 代码
#define WriteReg(Address,Value) *(unsigned int volatile *)(Address) = Value
#define U32    unsigned int
#define REG_EXT0CON      0xFFFF01018
#define EBILED_ADDRESS   0x78000000
//子函数声明
void EBILedSet(U32 Value);
void Delay(U32 t);
//主函数
int main(void)
{
    U32 i;
    WriteReg(REG_EXT0CON, 0xF0078003);
    while(1)
    {
        for(i = 0 ;i < 8; i++)
        {
            EBILedSet(0x1<<i);
            Delay(1000000);
        }
    }
    return 0;
}
//延时
void Delay(U32 t)
{
    do
    {
        t--;
    }while(t);
}
//控制 LED 灯的亮灭
void EBILedSet(U32 Value)
{
    WriteReg(EBILED_ADDRESS, ~Value);
}

```

5) 点击 Keil 工具栏的 Target Option 按钮 ，在 Options for Target 窗

口中设置 ULINK2 调试环境。

Target 标签页设置如图 6 所示。W90P710 芯片的指令地址空间从 0x00000000 开始。而评估板板载 SDRAM 存储器大小为 16MB。其中，分配 8MB 给只读存储器空间，用于存储指令；另外 8MB 分配给随机读写存储器空间，用于存储数据和栈等。因此，设置程序编译的地址映射如下。

Read/Only Memory Areas					Read/Write Memory Areas				
default	off-chip	Start	Size	Startup	default	off-chip	Start	Size	NoInit
<input checked="" type="checkbox"/>	ROM1:	0x0	0x800000	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	RAM1:	0x800000	0x800000	<input type="checkbox"/>
<input type="checkbox"/>	ROM2:			<input type="radio"/>	<input type="checkbox"/>	RAM2:			<input type="checkbox"/>
<input type="checkbox"/>	ROM3:			<input type="radio"/>	<input type="checkbox"/>	RAM3:			<input type="checkbox"/>
	on-chip					on-chip			
<input type="checkbox"/>	IROM1:			<input type="radio"/>	<input type="checkbox"/>	IRAM1:	0xFFE00000	0x2000	<input type="checkbox"/>
<input type="checkbox"/>	IROM2:			<input type="radio"/>	<input type="checkbox"/>	IRAM2:			<input type="checkbox"/>

图 6 Target 标签页设置

ASM 标签页设置与程序汇编有关的选项。如图 7 所示，在 Define 框内填写 EBI_NOINIT，避免启动代码中对 EBI 的配置和相关配置冲突。

Debug 标签页设置与程序调试器有关的设置。请按照图 8 所示，设置选择目标调试 ULink2/ME ARM Debugger，选择使用项目目录下的初始化文件 Ext_SDRAM.ini 配置评估板。

Utilities 标签页设置与程序下载到 Flash 存储器有关的选项。请按照如图 9 所示设置选择，并点击 Setting 按钮，如图 10 所示，删除在 Programming Algorithm 框中的 Flash 芯片，点击 OK 保存。



图 7 ASM 标签页设置

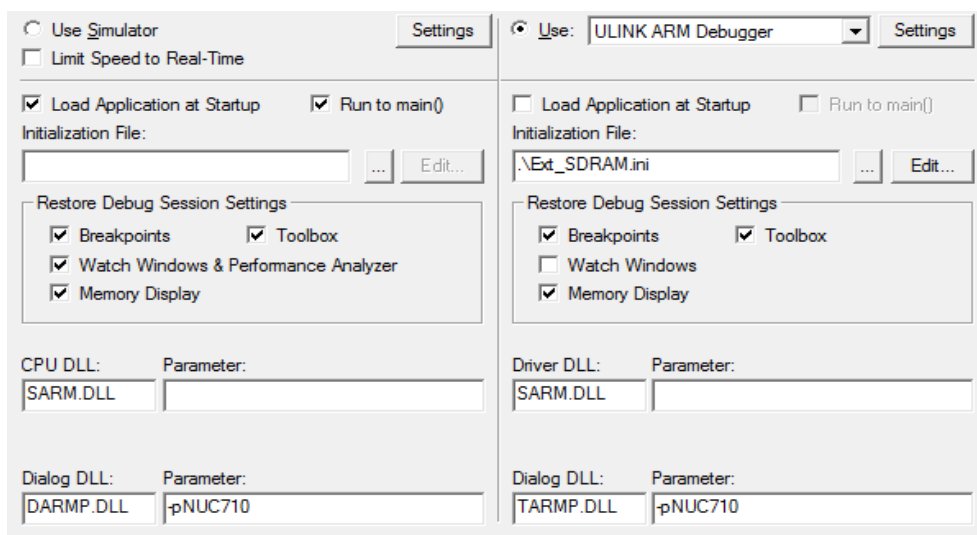


图 8 Debug 标签页设置

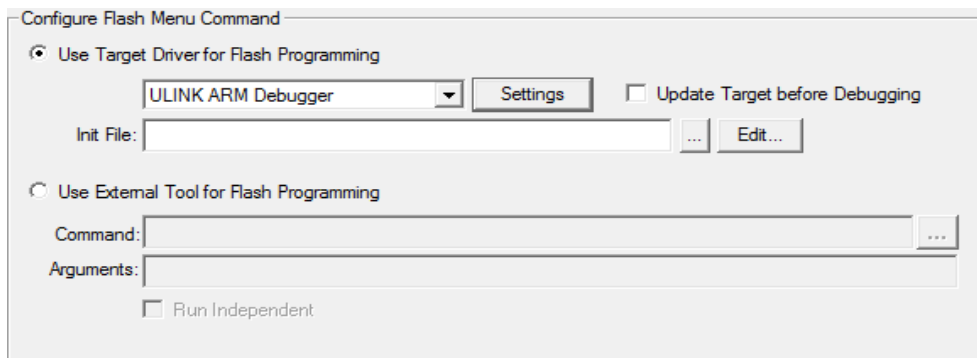


图 9 Utilities 标签页设置

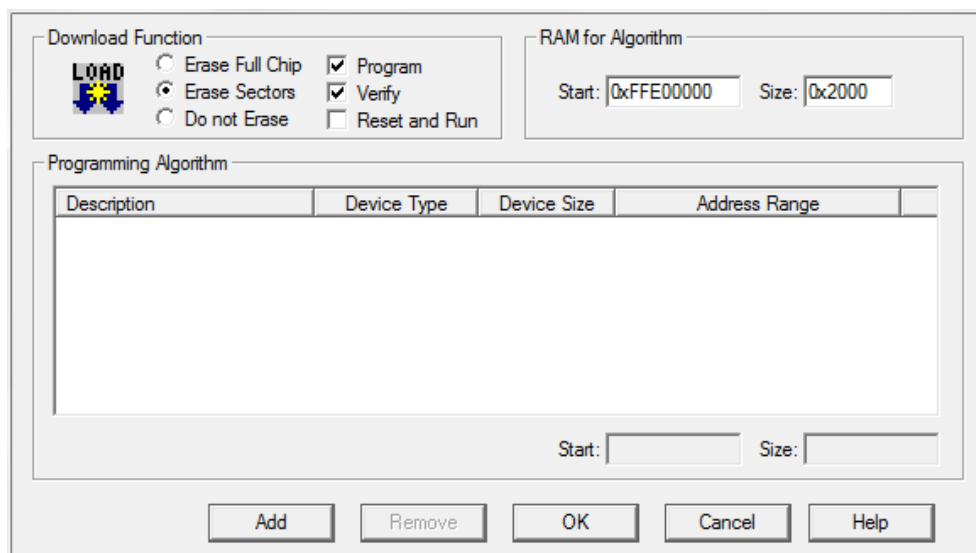


图 10 Utilities 标签页 Settings 窗口

- 6) 编译-连接工程，生成后缀名.axf 的镜像文件。
- 7) 修改并分析调试配置初始化文件 Ext_SDRAM.ini。该文件的部分内容如下所示。**最重要的是**要将装载的 axf 文件替换成自己工程产生的 axf 文件。axf 文件名称与工程名称相同，此处为 led.axf。
- 8) 将 ULINK2 仿真器的 USB 端连接到 PC 上，另一端连接到评估板的 JTAG 端口上，然后打开电源。
- 9) 点击 Keil 软件的 Debug 按钮，ULINK2 会自动地将.axf 后缀的镜像文件下载到 SDRAM 中。选择 Run 按钮或者单步调试按钮，观察程序的执行和 LED 流水灯效果。

```

// <o2.0> EBI Setup
// <o2.1> Enable Flash Programming
// <e2.2> Download Program
FUNC void Download (void) {
    if (Setup & (1 << 2)) {
// <s0.80> Command for Loading (替换自己的. axf)
        exec("LOAD led.axf INCREMENTAL");
    }
}
// </e>
// <e0.3> Setup Program Counter to Entry Point
// <o1> Program Entry Point <0x0-0xFFFFFFFF>
// </e>
// <o0.4> Execute Program untill Main Function
Setup = 0x1D;
Entry = 0x00000000;

EBI_Setup();           //设置外部总线
EnableFlashProgramming(); //Flash 编程使能
Download();             //程序代码下载至目标板 SDRAM 中
PC_Setup();             //设置 PC 指针为 0x00000000，开始执行
GoMain();               //一直执行到 main 函数入口处

```

5. 阵列键盘扫描实验

【实验目的】

- 1) 掌握 W90P710 键盘控制器的工作原理与扫描方法。
- 2) 掌握 I/O 端口扫描键盘的寄存器设定及程序设计方法。

【实验设备】

- 1) 硬件：W90P710 实验系统、PC 机、ULINK2 仿真器
- 2) 软件：PC 机操作系统、Keil MDK-ARM 4.22

【实验内容及要求】

编写基于中断的 KPI 控制程序。首先接好键盘跳线器，并设置键盘中断及相应 GPIO 接口，进入中断后能用键盘控制 LED 灯的显示。

每个按键从 0-F 依次编号，每按一个键，LED 灯按照相应的键值显示。比如：按下 BP7，相应的编号或者键值是 0x6，对应的二进制数为 0110，则 LED 灯低 4 位中的第 2 位和第 3 位亮，即 LED2 和 LED3 亮，其余 LED 全部不亮。

【实验原理】

在键盘中按键数量较多时，为了减少 I/O 口的占用，通常将按键排列成矩阵形式，在矩阵键盘中每条水平线和垂直线在交叉处不直接相连，而是通过一个按键相连接，这样在由 N 条水平线和 M 条垂直线最多可以有 $N * M$ 个按键，大大的减少了对于芯片 I/O 的占用。

矩阵键盘由行信号和列信号组成矩阵。每个按键连接到行信号和列信号的交叉点。通过矩阵键盘控制器输出行信号扫描信号（低电平）顺序逐行的对行扫描，读取列信号线的状态。

如果有按键按下，对应的行输出扫描信号会拉低列输入信号，触发键盘中断。键盘中断服务程序会将键盘状态寄存器的值读出，判断哪个按键被按下。

W90P710 评估板线路上采用 GPIO50 和 GPIO51 输出行扫描线，

键盘的列状态由 GPIO42~49 输入，实验接线图见图 11。

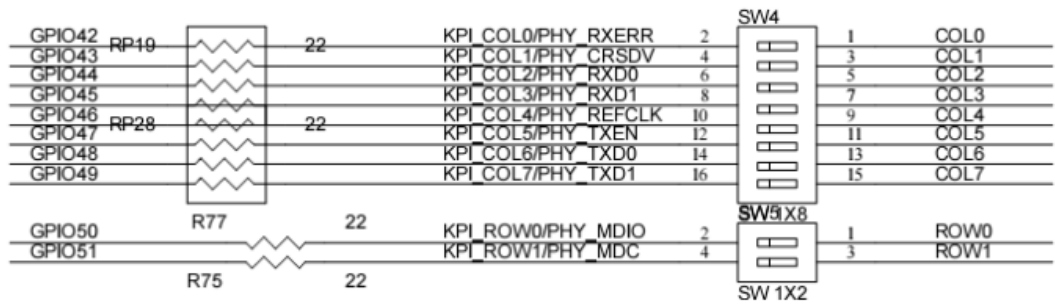


图 11(a) GPIO 口连接线路图

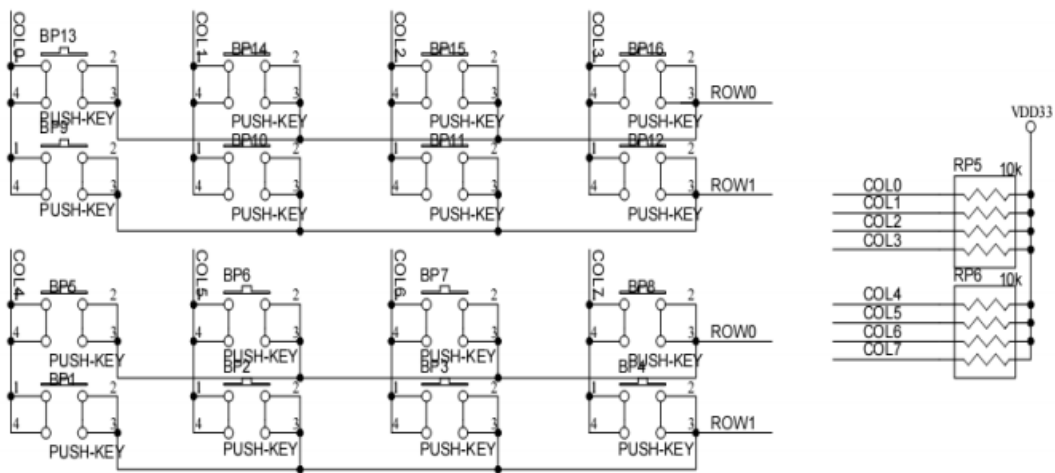


图 11(b) 2x8 键盘线路图

W90P710 评估板的键盘设计为 2 行输出扫描，8 列输入的方式。其键盘的模式可以通过相应的寄存器设定(最大支持 16x8)，例如设定为 2x8 键盘。键值的判断不需要设计者另行判断，系统提供了相应的寄存器存储按键值，只需要读键盘状态寄存器就可以知道按键值。当有 1 个或者 2 个键值按下的时候，键盘的扫描顺序是 ROW0 CL0-1-2-3-4-5-6-7，ROW1 CL0-1-2-3-4-5-6-7...ROW16 CL0-1-2-3-4-5-6-7。因此，两个键同时按下的时候，只会读到较小的键值。

实验中，键盘键值的判断采用的是中断模式，需要程序打开键盘控制器中断，并设定优先级。在中断服务程序中，读取按键键值并

点亮对应的 LED 灯。注意：W90P710 评估板上的键盘的第一行和第三行对应芯片键盘控制器的 ROW0 扫描行，第二行和第四行对应芯片键盘控制器的 ROW1 扫描行，而不是直接对应的 4x4 矩阵键盘顺序划分。因此，读入键值后需要转换以对应物理按键，并控制相应的 LED 灯。

【实验步骤】

实验步骤与“LED 灯闪烁实验”步骤一样，实验中的 C 代码如下。

```
//KPI 中断服务子程序
__irq void IRQ_Handler(void){
    u32 KeyValue, temp;
    temp = ReadReg(REG_KPISTATUS);
    temp &= 0x0000000f;
    if(temp > 0xB){
        KeyValue = temp - 0xC;
    }
    else if(temp < 0x4){
        KeyValue = temp + 0xC;
    }
    else{
        KeyValue = temp;
    }
    WriteReg(EBILED_ADDRESS, ~KeyValue);
}
```

```

//矩阵键盘扫描 C 代码
#define u32    unsigned int
#define REG_GPIO_CFG2      0xFFFF83020
#define REG_AIC_SCR29      0xFFFF82074
#define REG_AIC_MECR       0xFFFF82120
#define REG_KPICONF        0xFFFF88000
#define REG_KPISTATUS      0xFFFF8800C
#define EBILED_ADDRESS     0x78000000
#define REG_EXT0CON        0xFFFF01018
#define EBI_EXT0_VALUE     0xF0078003

#define WriteReg(Address,Value) *(unsigned int volatile*)(Address) = Value
#define ReadReg(Address)        *(unsigned int volatile*)(Address)

void KPIInit(void);

int main (void){
    WriteReg(REG_EXT0CON, EBI_EXT0_VALUE);
    KPIInit();
    while(1);
    return 0;
}

void KPIInit(void){
    WriteReg(REG_GPIO_CFG2, 0x000aaaaa);    //将 GPIO 设置为 KPI 模
式， 2X8
    WriteReg(REG_AIC_SCR29, 0x00000045);    //将 KPI 中断设置为高电平
有效，优先级为 5
    WriteReg(REG_AIC_MECR, 0x20000000);
    WriteReg(REG_KPICONF, 0x00142fff);
}

```

6. LCD 显示实验

【实验目的】

- 1) 掌握 LCD 液晶模块的基本原理和功能。
- 2) 掌握 LCD 液晶模块和微控制器的硬件接口及程序设计方法。

【实验设备】

- 1) 硬件：W90P710 实验系统、PC 机、ULINK2 仿真器
- 2) 软件：PC 机操作系统、Keil MDK-ARM 4.22

【实验内容及要求】

编写 LCD 程序，在 LCD 上显示图片。

【实验原理】

W90P710 评估板的 LCD 控制器的主要目的是将 VIDEO/OSD 的原始图像数据显示到外部液晶屏上。LCD 控制器可以支持 AUO 的 TFT 屏，LG 的 TFT 屏，CASIO 的 TFT 屏，实验中使用的是 AUO 3.0" TFT A030DL01（960x240，8 位数据总线）数字式 TFT 液晶屏。此外，LCD 控制器还能够连接到 STN LCD 和电视编码器上，并且支持 VIDEO 和 OSD 两种显示模式。这两种显示模式的设定由不同的寄存器控制，通过设定相应寄存器的值可以实现不同的模式和功能。

AUO 3.0" TFT A030DL01 液晶屏由单个像素点排列组成，向这些

像素点配置不同的值，可以显示不同的颜色。但是，由于 AUO TFT 液晶屏内部没有缓存，需要在 SDRAM 中开辟一片缓冲区存储一维数组形式的原始图像数据。然后，将存储原始图像数据的 SDRAM 缓冲区的首地址赋值给 LCD 控制器中的 FIFO1SADDR 寄存器。接着，原始图像数据在时钟脉冲的作用下依次输入到 LCD 控制器的 FIFO1 中，直至 FIFO1 被填满才将 FIFO1 中的数据输出到 LCD 颜色发生器。最后，LCD 颜色发生器将图像显示到评估板上的液晶屏中。

如果在液晶屏上定位显示图片或其他信息，需要计算出所要显示的内容在液晶屏上的位置。然后，在新建的 SDRAM 缓冲区中存入需要显示的内容即可。LCD 硬件连接图如下。

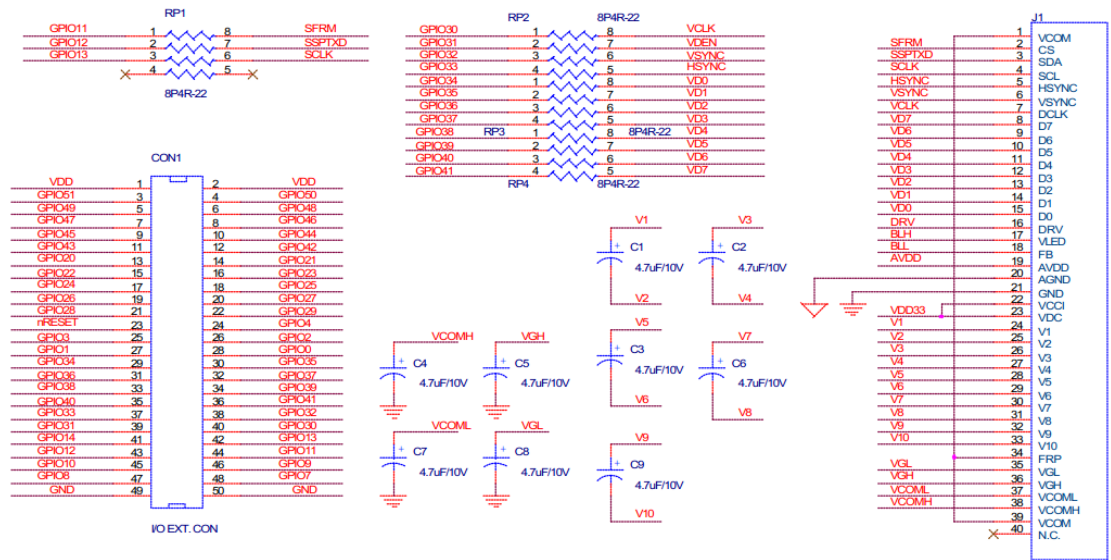


图 12 LCD 连接原理图

LCD 模块由 40 个引脚与外界电路相连，由于本身没有寄存器，需要由 W90P710 中的 LCD 控制器提供相应的寄存器，控制选择对应的液晶类型及显示形式、FIFO 使用、颜色显示模式和时序设置。

LCDCON 寄存器比较重要，需要根据实际所使用的硬件，对控制寄存器进行配置。本次实验使用 16 bpp 模式、真彩型 TFT LCD、没有使用查找表功能，分别使能相应的位实现 LCDCON 寄存器初始化操作。

【实验步骤】

实验步骤与“LED 灯闪烁实验”类似，但有两个不同点：在步骤 3 中不需要新建源文件，直接将“LCD 显示实验”提供的实验文件拷贝到实验目录的新建文件夹 LCD 中；步骤 4 中将 LCD 文件夹中的所有文件添加到工程目录中。

7. 附件

- [1] Getting Started, Creating Applications with μ Vision®4
- [2] W90P710 32bit ARM Microcontroller Product Data Sheet
- [3] W90P710 Eval. Board Hardware Application Note
- [4] W90P710 Schematics
- [5] A030DL01 3.0" TFT LCD Module Spec