

《物联网技术及应用实验》

实验报告本

|  |  |
| --- | --- |
| 班级： |  |
| 学号： |  |
| 姓名： |  |
| 指导教师： |  |

信息科学与工程学院

年 月

**实验一** 光照传感器采集实验

1. **实验目的**

（1）掌握光照传感器的操作方法；

（2）掌握光照传感器采集程序的编程方法。

**二、实验装置**

硬件：计算机一台（操作系统为Windows XP或Windows 7）；CVT-IOT-VSL实验箱一台；CC DEBUGGER仿真器；USB数据线一根。

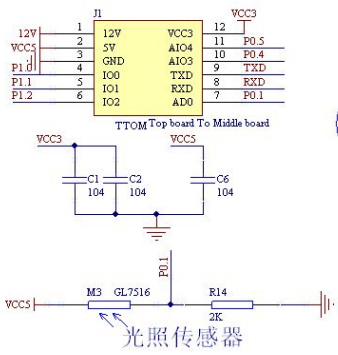
软件：IAR Embedded Workbench for MCS-51开发环境。

**三、基础知识**

①光敏电阻

本实验箱采用的光照传感器为光敏电阻，光敏电阻属半导体光敏器件，在半导体光敏材料两端装上电极引线，将其封装在带有透明窗的管壳里就构成光敏电阻，为了增加灵敏度，两电极常做成梳状。用于制造光敏电阻的材料主要是金属的硫化物、硒化物和碲化物等半导体。通常采用涂敷、喷涂、烧结等方法在绝缘衬底上制作很薄的光敏电阻体及梳状欧姆电极，接出引线，封装在具有透光镜的密封壳体内，以免受潮影响其灵敏度。在光敏电阻两端的金属电极加上电压，其中便有电流通过，受到一定波长的光线照射时，电流就会随光强的增大而变大，从而实现光电转换。入射光消失后，由光子激发产生的电子—空穴对将复合，光敏电阻的阻值也就恢复原值。光敏电阻没有极性，纯粹是一个电阻器件，使用时既可加直流电压，也加交流电压。半导体的导电能力取决于半导体导带内载流子数目的多少。因此，可以通过简单的电阻分压，通过计算光敏电阻两端的电压值来转换成相应的光照值。

②光照传感器接口电路



1-1 光照传感器接口电路

③实验说明

本实验直接采集光照传感器两端的电压值，并通过串口，在串口调试助手中显示出来，改变光照强度可观察到相应的光照值发生明显变化。

光照传感器实验不需要组网，只需要一个子节点和光照传感器板，按照步骤操作就可完成实验。

**四、实验内容**

（1）使用mini USB将子节点板与PC相连。

（2）启动IAR Embedded Workbench开发环境，创建一个新文件，在文件中添加如下代码

#ifndef HAL\_LED\_H

#define HAL\_LED\_H

#ifdef \_\_cplusplus

extern "C"

{

#endif

#define uint8 unsigned char

#define uint16 unsigned int

/\* LEDS - The LED number is the same as the bit position \*/

#define HAL\_LED\_1 0x01

/\* Modes \*/

#define HAL\_LED\_MODE\_OFF 0x00

#define HAL\_LED\_MODE\_ON 0x01

#define HAL\_LED\_MODE\_TOGGLE 0x08

extern void HalLedInit( void );

/\*

\* Set the LED ON/OFF/TOGGLE.

\*/

extern void HalLedSet( uint8 led, uint8 mode );

#ifdef \_\_cplusplus

}

#endif

#endif

将该文件保存为Led.h。同种方法创建Led.c：

#include <ioCC2540.h>

#include "Led.h"

#define uint8 unsigned char

#define uint16 unsigned int

#define BV(n) (1<<(n))

#define LED1\_DDR P0DIR

#define LED1\_SEL P0SEL

#define LED1\_BV BV(0)

#define LED1\_SBIT P0\_0

void HalLedInit (void)

{

// Set LED GPIOs to general io port.

LED1\_DDR &= ~LED1\_BV;

// Set LED GPIOs to outputs.

LED1\_DDR |= LED1\_BV;

//LED1 OFF

LED1\_SBIT = 1;

}

void HalLedSet (uint8 leds, uint8 mode)

{

switch (mode)

{

case HAL\_LED\_MODE\_ON:

LED1\_SBIT = 0;

break;

case HAL\_LED\_MODE\_OFF:

LED1\_SBIT = 1;

break;

case HAL\_LED\_MODE\_TOGGLE:

if (LED1\_SBIT) { LED1\_SBIT = 0; } else { LED1\_SBIT = 1;}

break;

default:

break;

}

}

同种方法创建 Uart.h：

#ifndef HAL\_UART\_H

#define HAL\_UART\_H

#ifdef \_\_cplusplus

extern "C"

{

#endif

#define uint8 unsigned char

#define uint16 unsigned int

extern void HalUartInit( void );

extern void HalUARTWrite(uint8 \*buf, uint16 len);

#ifdef \_\_cplusplus

}

#endif

#endif

同种方法创建 Uart.c：

#include <ioCC2540.h>

#include "Uart.h"

#define uint8 unsigned char

#define uint16 unsigned int

void HalUartInit(void)

{

CLKCONCMD &= ~(1<<6); //选择 32M 晶振

while(!(SLEEPSTA & (1<<6))); //等待 XSOC 稳定

CLKCONCMD &=~ ((1<<6) | (7<<0)) ;//TICHSPD 128分频，CLKSPD不分频

PERCFG = 0x00;

P0SEL |= (0x0f<<2) ; //P0 用作串口

U0CSR |= (1<<7); //UART 方式

U0GCR |= 11;

U0BAUD |= 216; //波特率设为 115200

UTX0IF = 0;

U0CSR |= (1<<6);

IEN0 |= 0x84; //开总中断，接收中断

URX0IE = 1;

EA = 1;

}

void HalUARTWrite(uint8 \*buf, uint16 len)

{

uint16 i;

for(i=0;i<len;i++)

{

U0DBUF = \*buf++;

while(UTX0IF == 0);

UTX0IF = 0;

}

}

创建main.c：

#include <ioCC2540.h>

#include "Uart.h"

#include "Led.h"

#include "hal\_adc.h"

#include "stdio.h"

#define uint8 unsigned char

uint8 light\_value = 0x00;

uint8 buf[5];

void delay(void)

{

unsigned int i;

unsigned int j;

for(i=0;i<1500;i++)

{

for(j=0;j<1000;j++)

{

asm("NOP");

asm("NOP");

asm("NOP");

}

}

}

void main()

{

HalLedInit();

HalUartInit();

HalAdcInit();

while(1)

{

HalUARTWrite("light value:",13);

light\_value =(uint8) HalAdcRead(1,HAL\_ADC\_RESOLUTION\_8);

sprintf( (char\*)buf, "%d" ,light\_value);

HalUARTWrite(buf,5);

HalUARTWrite("\r\n",2);

delay();

}

}

（3）新建一个工程LIGHT SENSOR，将上述5个文件添加进工程中，并修改LIGHT SENSOR的工程设置，如图1-2所示。

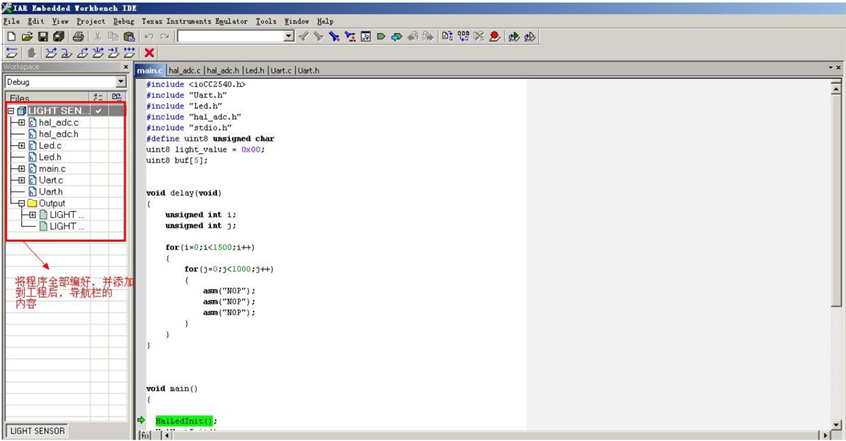


图1-2 LIGHT SENSOR工程创建

（4）编译LIGHT SENSOR，成功后，下载并运行，选择miniUSB的串口号，配置波特率，从串口调试工具观察结果。

五**、实验结果**

将光照传感器模块插好，实验箱通电后，改变照射到传感器上的光的亮度，记录显示屏上的结果。

文本, 表格

描述已自动生成

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 节点名称 | 节点类型 | 源地址 | 父地址 | 光照度 |
| 第一次 | 光照传感器 | 终端节点 | 0xE74A | 0x8723 | 7F |
| 第二次 | 光照传感器 | 终端节点 | 0xE74A | 0x8723 | 39 |
| 第三次 | 光照传感器 | 终端节点 | 0xE74A | 0x8723 | 34 |
| 第四次 | 光照传感器 | 终端节点 | 0xE74A | 0x8723 | 3A |
| 第五次 | 光照传感器 | 终端节点 | 0xE74A | 0x8723 | 73 |

**实验一成绩：\_\_\_\_\_\_\_\_\_\_\_\_ 教师签名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**