

دانشگاهصنعتیاصفهان

دانشکدهبرقوکامپیوتر

**دستورکار آزمایشگاه طراحی مدارهای واسط**

**آزمایش سیزدهم**

تهیه کننده: مهران صفایانی

پاییز95

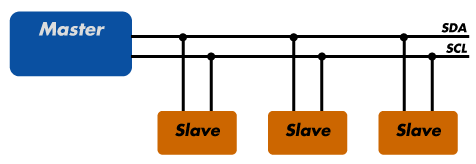
**I2Cآشنایی با پروتکل**

* **اهداف:**
* **آشنایی با پروتکل ارتباطی I2C و امکانات آن**
* **راه اندازی بلوک I2C میکروکنترلر LPC1768**
* **مقدمه:**

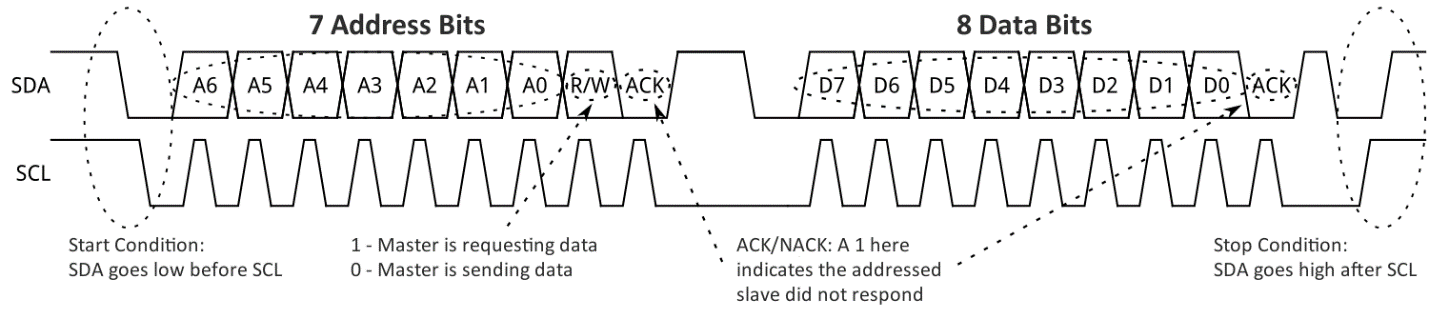
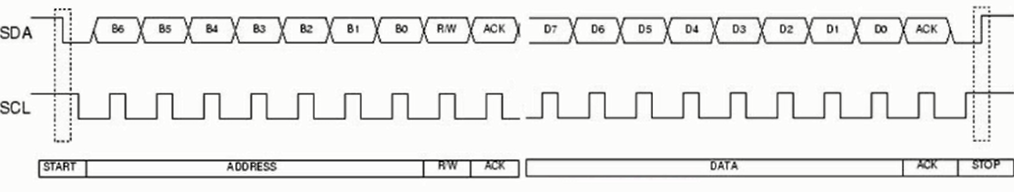
پروتکل ارتباطی I2C که مخفف Inter Integrated Circuit است و نام دیگر آن TWI می باشد ،یک پروتکل ارتباطی جهت ارتباط با وسایل جانبی ، حافظه ها ، سنسورها و... می باشد .این پروتکل توسط شرکت فیلپس معرفی شده است. از قابلیت های این پروتکل برقراری ارتباط از طریق تنها دو سیم می باشد.

* **واحدI2C میکروکنترلر LPC1768 :**

همانطور که در مقدمه ذکر شد از طریق این پروتکل تنها از طریق دو سیم می توان ارتباط سنکرون برقرار نمود که این سیمها به نامهای SDA (Serial Data) وSCL (Serial clock) می باشد.



با توجه به شکل به وسیله این پروتکل یک MASTER می تواند با چند SLAVE ارتباط برقرار نماید.ارسال اطلاعات در پروتکل I2C در ۵ مرحله صورت می پذیرد که ابتدا ۱-شرایط اولیه شروع ۲-ارسال آدرس ۳- تصدیق صحت آدرس ۴-ارسال دیتا ۵-توقف که در زیر شکل مربوط به این مراحل قرار داده شده است.

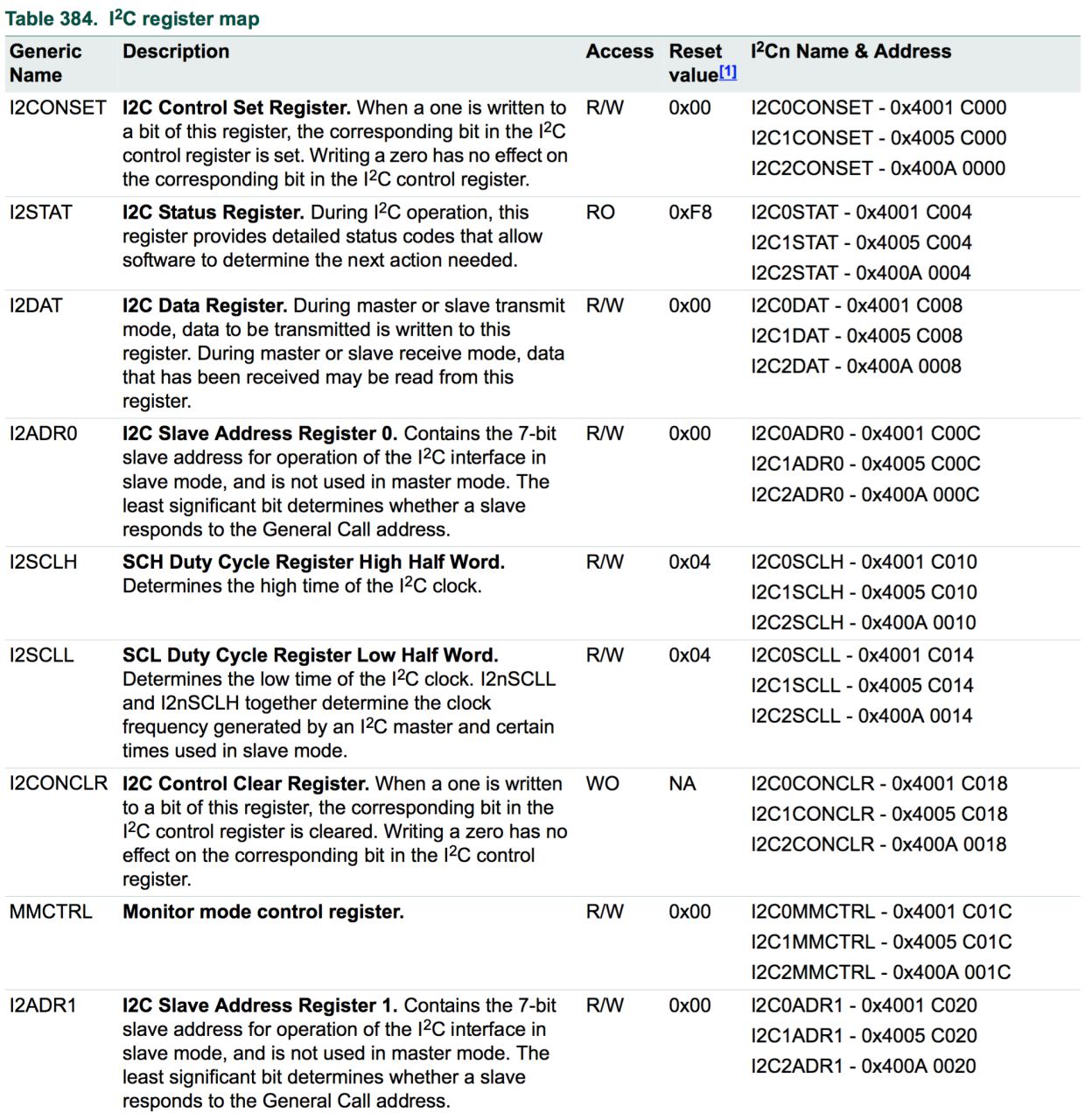


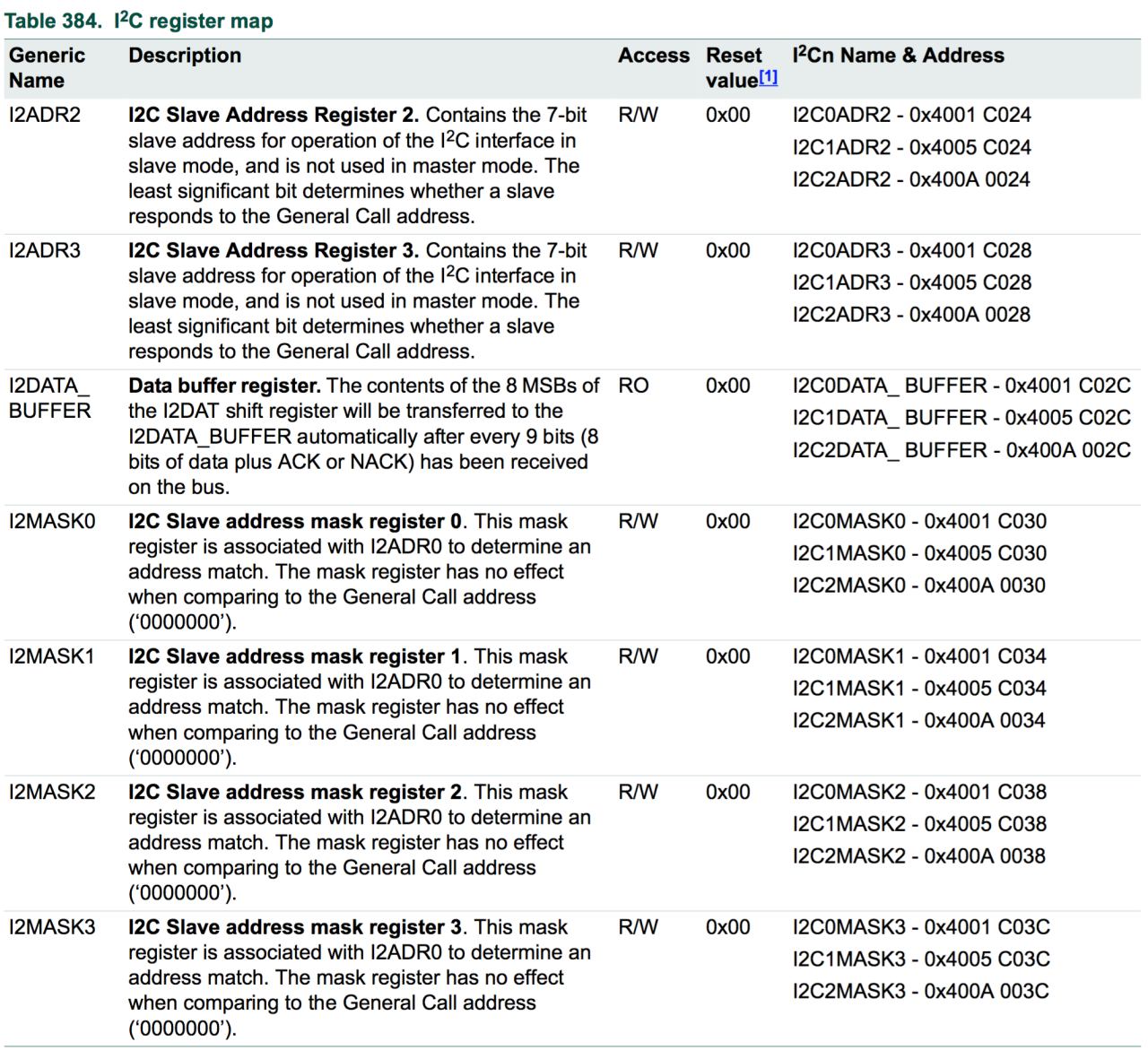
نکته مهم در رابطه با این پروتکل آدرس می باشد که توسط بیت مورد استفاده قرار می گیرد. ۴ بیت پرارزش برای مشخص کردن وسیله جانبی و ۳ بیت کم ارزش جهت آدرس دستگاه است . پس به وسیله این پروتکل تا ۱۲۸ دستگاه جانبی مختلف و ۸ وسیله یکسان را به باس متصل نمود.

میکروکنترلر LPC1768 دارای ۳ کانال I2C است که باید توسط رجیستر PINSEL در مد SDA وSCL قرار گیرند.

* **رجیسترهای I2C:**

رجیسترهای I2C در ۵ گروه اصلی کنترلی ، وضعیت ، کلاک ،آدرس و داده تقسیم می شوند که در زیر شمای کلی آن ها را مشاهده می کنید. به دلیل فراوانی و پیچیدگی های رجیستر های I2C از کتابخانه آماده این واسط برای برقراری ارتباط استفاده می نماییم. در ادامه به بررسی توابع مهم این کتابخانه می پردازیم:

****



* **بررسی توابع مهم کتابخانه I2C**

**بررسی تابع I2C\_Init :** این تابع برای مقدار دهی اولیه رجیسترهای واسط I2C استفاده می شود وMaster یا Slave بودن را به عنوان پارامتر دریافت می کند.در زیرکدهای موجود در این تابع را مشاهده می نمایید:

**بررسی تابع I2C\_WriteNByte :** با استفاده از این تابع می توان مقادیر موجود در یک بافر را بر روی واسط I2C انتقال داد. در زیر امضای تابع ذکر شده را مشاهده می نمایید:

uint32\_t I2CInit( uint32\_t I2cMode )

{

LPC\_SC->PCONP |= (1 << 19);

/\* set PIO0.27 and PIO0.28 to I2C0 SDA and SCK \*/

/\* function to 01 on both SDA and SCK. \*/

LPC\_PINCON->PINSEL1 &= ~~0x03C00000;

LPC\_PINCON->PINSEL1 |= 0x01400000;

/\*--- Reset registers ---\*/

LPC\_I2C0->I2SCLL = I2SCLL\_SCLL;

LPC\_I2C0->I2SCLH = I2SCLH\_SCLH;

if ( I2cMode == I2CSLAVE )

{

LPC\_I2C0->I2ADR0 = 0xA0; //Set Slave Address

}

/\* Install interrupt handler \*/

NVIC\_EnableIRQ(I2C0\_IRQn);

LPC\_I2C0->I2CONSET = I2CONSET\_I2EN; //Enable I2C

return( 1 );

}

uint8\_t I2C\_WriteNByte(uint8\_t sla, uint8\_t suba\_type, uint32\_t suba, uint8\_t \*s, uint32\_t num);

در پارامتر های دریافتی slaهمان slave address ، suba\_type میزان بایت sub address ، suba همان subaddress ، s اشاره گر به بافر و num میزان داده برای نوشتن را مشخص نماید.

**بررسی تابع I2C\_ReadNByte :** با استفاده از این تابع می توان مقادیر موجود در یک بافر را از روی واسط I2C خواند . در زیر امضای تابع ذکر شده را مشاهده می نمایید:

uint8\_t I2C\_ReadNByte (uint8\_t sla, uint8\_t suba\_type, uint32\_t suba, uint8\_t \*s, uint32\_t num);

توضیح پارامتر های دریافتی این تابع مانند تابع قبلی می باشد.

**برنامه نمونه:** در برنامه ی زیر نحوه نوشتن خواندن برروی حافظه EEPROMبا استفاده از پروتکلI2C آمده است.

#include "lpc17xx.h"

#include "i2c.h"

extern uint8\_t buf[32];

void Delay(uint32\_t delaydata){

uint32\_t i,j,k;

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

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

for(k=0;k<100;k++);

}

int main (void){

uint32\_t i;

SystemInit();

LPC\_GPIO2->FIODIR = 0x000000ff; //LEDs PORT2 are Output

LPC\_GPIO0->FIODIR |= 0x03f80000;

LPC\_GPIO0->FIOSET = 0x03f80000;

if ( I2CInit( (uint32\_t)I2CMASTER ) == 0 ){ //initialize I2c

while ( 1 ); //Fatal error

}

for ( i = 0; i < 8; i++ ){ //clear buffer

buf[i]=i+1;

}

I2C\_WriteNByte(0xa0, 1, 0x00, buf, 8);//write buf array ito

Delay(50); EEPROM

for ( i = 0; i < 8; i++ ){ //clear buffer

buf[i] =0;

}

I2C\_ReadNByte (0xa0, 1, 0x00, buf, 8); //read from EEPROM & save

Delay(50); in buff array

while(1){

for(i=0;i<8;i++){

LPC\_GPIO2->FIOPIN = buf[i];

Delay(300);

}

}

}

* **دستور کار:**

1. برنامه بالا یه صورت عملی بروی میکرو ببندید و تست کنید.
2. به وسیله ارتباط I2Cدو میکرو را به هم متصل نمایید و میکروی اول پیامی را به میکرو دوم می فرستد و میکرو دوم پیام دریافتی را بروی LCD چاپ و پیام دریافت شده را برای میکرو اول ارسال می کند.