

//C programming of LPC1768 I2C to send data to slave

```
#include<stdio.h>
#include <lpc17xx.h>

void i2c_init(void);
int i2c_start();
int i2c_write(unsigned char buff);
void i2c_stop(void);
void wait(unsigned int delay);

void i2c_init(void)
{
//Select alternate function 2 (I2C) for pins P0.10 and P0.11
    LPC_PINCON->PINSEL0 |= 0x02<<20;    //Pin P0.10 allocated to alternate function 2
    LPC_PINCON->PINSEL0 |= 0x02<<22;    //Pin P0.11 allocated to alternate function 2

//Enable power to peripheral I2C2
    LPC_SC->PCONP |= 1 << 26;    //re-enable POWER to I2C_2 if required

//Set the peripheral clock for I2C2 to the system clock frequency (CCLK).
    LPC_SC->PCLKSEL1 |= 1<<20;    //pclk = cclk

//I2C2 pins as output, open drain and no pull up or pull down resistors.
    LPC_GPIO0->FIODIR |= 1<<10;    //Bit P0.10 an output
    LPC_GPIO0->FIODIR |= 1<<11;    //Bit P0.11 an output
    LPC_PINCON->PINMODE0 &= ~(3<<20); //P0.10 has pull up/down resistor
    LPC_PINCON->PINMODE0 |= (2<<20);    //omit to use internal pull up
    LPC_PINCON->PINMODE0 &= ~(3<<22); //P0.11 has pull up/down resistor
    LPC_PINCON->PINMODE0 |= (2<<22);    //omit to use internal pull up
    LPC_PINCON->PINMODE_OD0 |= 1<<10;    //Bit P0.10 is open drain
    LPC_PINCON->PINMODE_OD0 |= 1<<11;    //Bit P0.11 is open drain

//Determine possible values of I2CSCLH and I2CSCLL to give standard (100kHz) I2C rate given
    LPC_I2C2->I2SCLH = 60; //100kHz from 12MHz
    LPC_I2C2->I2SCLL = 60; //100kHz from 12MHz

//Enable I2C2 interface
    LPC_I2C2->I2CONSET =(1<<6); //I2EN bit
}

int i2c_start()                //I2C communication start
{
    LPC_I2C2->I2CONCLR=(1<<3); //SIC bit
    LPC_I2C2->I2CONSET=(1<<5); //STA bit
    while(!(LPC_I2C2->I2STAT==0x08));
    return 0;
}
```

```

int i2c_write(unsigned char buff) //Data writing through I2C
{
    LPC_I2C2->I2CONSET|=(1<<3); //SI bit
    LPC_I2C2->I2DAT=buff; //write data
    wait(5000);
    LPC_I2C2->I2CONCLR=(1<<3); //SIC bit
    wait(5000);
    return 0;
}

void i2c_stop(void)          //Stop the I2C communication
{
    LPC_I2C2->I2CONSET=(1<<4); //STO bit
    LPC_I2C2->I2CONCLR=(1<<6); //I2ENC bit
}

void wait(unsigned int delay)
{
    while(delay--);
}

int main(void)
{
    while(1)
    {
        i2c_init();
        i2c_start();
        i2c_write('0');
        wait(500000);
        i2c_write('1');
        wait(500000);
        i2c_write('2');
        wait(500000);
        i2c_write('4');
        wait(500000);
        i2c_stop();
    }
}

```

//C programming of LPC1768 I2C to send data to slave and port1

```
#include<stdio.h>
#include <lpc17xx.h>
void i2c_init(void);
int i2c_start();
int i2c_write(unsigned char buff);
void i2c_stop(void);
void wait(unsigned int delay);

void i2c_init(void)
{
//Select alternate function 2 (I2C) for pins P0.10 and P0.11
    LPC_PINCON->PINSEL0 |= 0x02<<20;    //Pin P0.10 allocated to alternate function 2
    LPC_PINCON->PINSEL0 |= 0x02<<22;    //Pin P0.11 allocated to alternate function 2

//Enable power to peripheral I2C2
    LPC_SC->PCONP |= 1 << 26;    //re-enable POWER to I2C_2 if required

//Set the peripheral clock for I2C2 to the system clock frequency (CCLK).
    LPC_SC->PCLKSEL1 |= 1<<20;    //pclk = cclk

//I2C2 pins as output, open drain and no pull up or pull down resistors.
    LPC_GPIO0->FIODIR |= 1<<10;    //Bit P0.10 an output
    LPC_GPIO0->FIODIR |= 1<<11;    //Bit P0.11 an output
    LPC_PINCON->PINMODE0 &= ~(3<<20); //P0.10 has pull up/down resistor
    LPC_PINCON->PINMODE0 |= (2<<20);    //omit to use internal pull up
    LPC_PINCON->PINMODE0 &= ~(3<<22); //P0.11 has pull up/down resistor
    LPC_PINCON->PINMODE0 |= (2<<22);    //omit to use internal pull up
    LPC_PINCON->PINMODE_OD0 |= 1<<10;    //Bit P0.10 is open drain
    LPC_PINCON->PINMODE_OD0 |= 1<<11;    //Bit P0.11 is open drain

//Determine possible values of I2CSCLH and I2CSCLL to give standard (100kHz)
//I2C rate given
    LPC_I2C2->I2SCLH = 60; //100kHz from 12MHz
    LPC_I2C2->I2SCLL = 60; //100kHz from 12MHz

//Enable I2C2 interface
    LPC_I2C2->I2CONSET =(1<<6); //I2EN bit
}

int i2c_start()                //I2C communication start
{
    LPC_I2C2->I2CONCLR=(1<<3);//SIC bit
    LPC_I2C2->I2CONSET=(1<<5);//STA bit
    while(!(LPC_I2C2->I2STAT==0x08));
    return 0;
}
```

```

int i2c_write(unsigned char buff) //Data writing through I2C
{
    LPC_GPIO1->FIODIR = 0xff;
    LPC_I2C2->I2CONSET|=(1<<3);//SI bit
    LPC_I2C2->I2DAT=buff;//write data
    LPC_GPIO1->FIOPIN=LPC_I2C2->I2DAT;//write data to port1
    wait(5000);
    LPC_I2C2->I2CONCLR=(1<<3);//SIC bit
    wait(5000);
    return 0;
}

void i2c_stop(void)          //Stop the I2C communication
{
    LPC_I2C2->I2CONSET=(1<<4);//STO bit
    LPC_I2C2->I2CONCLR=(1<<6);//I2ENC bit
}

void wait(unsigned int delay)
{
    while(delay--);
}

int main(void)
{
    while(1)
    {
        i2c_init();
        i2c_start();
        i2c_write('0');
        wait(500000);
        i2c_write('1');
        wait(500000);
        i2c_write('2');
        wait(500000);
        i2c_write('4');
        wait(500000);
        i2c_stop();
    }
}

```