|  |
| --- |
| **EMBEDDED SYSTEM LABORATORY** |
| **LAB 1** |

**ARM MICROCONTROLLER GPIO INTERFACE AND PROGRAMING**

Nguyễn Minh Đức – ITITIU21045

Tôn Long Thuận – ITITIU21322

### I. LAB OBJECTIVES

### - In this Lab students will learn about ARM-CORTEX M3 (LPC1768) Microcontroller.

### - This Lab experiments are intended to implement basic GPIO Interface of ARM-CORTEX M3 Microcotroller to pheriperal devices in MB1700 Kit and write C code programming to control these devices.

### II. PRE-LAB

### II.1 LPC1768 MICROCONTROLLER INTRODUCTION

### A close-up of a document Description automatically generated

A diagram of a computer program

Description automatically generated

A close-up of a document

Description automatically generated

A document with text on it

Description automatically generated

A page of a computer program

Description automatically generated with medium confidence

A close-up of a text

Description automatically generated

A document with text on it

Description automatically generated

A document with text on it

Description automatically generated

### II.1 LPC1768 GPIO REGISTER

### A screenshot of a computer Description automatically generated

### A table with text and numbers Description automatically generated

### 

### A screenshot of a computer program Description automatically generated

### A screenshot of a computer program Description automatically generated

### A screenshot of a computer code Description automatically generated

A screenshot of a computer code

Description automatically generated

### III. LAB PROCERUCE

### III.1 Lab Experiment 0 : Create a first blynky program using Keil Vision

### A white paper with black text Description automatically generated

### A close up of a document Description automatically generated

### 

### III.1 Lab Experiment 1 : Write the code to turn on and turn off a led which is connected to P2.2 GPIO port pin with the time delay 1 second.

### #include <lpc17xx.h>

### #define PRESCALE (25000-1) //25000 PCLK clock cycles to increment TC by 1

### void delayMS(unsigned int milliseconds);

### void initTimer0(void);

### int main(void)

### {

### //SystemInit(); //called by Startup Code before main(), hence no need to call again.

### initTimer0(); //Initialize Timer0

### LPC\_GPIO2->FIODIR |= (1<<2); //Configure P2.2 as output

### while(1)

### {

### LPC\_GPIO2->FIOSET |= (1<<2); //Turn ON LED P2.2

### delayMS(1000); //0.5 Second(s) Delay

### LPC\_GPIO2->FIOCLR |= (1<<2); //Turn OFF LED P2.2

### delayMS(1000);

### }

### }

### void initTimer0(void)

### {

### /\*Assuming that PLL0 has been setup with CCLK = 100Mhz and PCLK = 25Mhz.\*/

### LPC\_SC->PCONP |= (1<<1); //Power up TIM0. By default TIM0 and TIM1 are enabled.

### LPC\_SC->PCLKSEL0 &= ~(0x3<<3); //Set PCLK for timer = CCLK/4 = 100/4 (default)

### LPC\_TIM0->CTCR = 0x0;

### LPC\_TIM0->PR = PRESCALE; //Increment LPC\_TIM0->TC at every 24999+1 clock cycles

### //25000 clock cycles @25Mhz = 1 mS

### LPC\_TIM0->TCR = 0x02; //Reset Timer

### }

### void delayMS(unsigned int milliseconds) //Using Timer0

### {

### LPC\_TIM0->TCR = 0x02; //Reset Timer

### LPC\_TIM0->TCR = 0x01; //Enable timer

### while(LPC\_TIM0->TC < milliseconds); //wait until timer counter reaches the desired delay

### LPC\_TIM0->TCR = 0x00; //Disable timer

### }

**III.2. Lab Experiment 2 :** Write a program to interface LPC1768 GPIO port pin using Keil C to scan P1.26 pin, if this pin is 0, Toggle the led connected to P2.2.

**#include <lpc17xx.h>**

**int main(void)**

**{**

**//SystemInit(); //called by Startup Code before main(), hence no need to call again.**

**LPC\_GPIO2->FIODIR |= (1<<2); //Configure P2.2 as output pins**

**LPC\_GPIO1->FIODIR &= ~(1<<26); //Configure P1.26 as input pin**

**while(1)**

**{**

**if((LPC\_GPIO1->FIOPIN & (1<<26))==0)**

**{**

**LPC\_GPIO2->FIOPIN ^= (1<<2); //Toggle LED P2.2**

**}**

**}**

**}**

**III.3. Lab Experiment 3 :** Write a program to interface LPC1768 GPIO port pin using Keil C to scan P1.26 pin and P1.24, if pin1.26 pin is 0, Toggle the led1 (connected to P2.2),. if pin2.24 pin is 0, Toggle the led2 (connected to P2.3).

**#include <lpc17xx.h>**

**#define PRESCALE (25000-1) //25000 PCLK clock cycles to increment TC by 1**

**void delayMS(unsigned int milliseconds);**

**void initTimer0(void);**

**int main(void)**

**{**

**initTimer0(); //Initialize Timer0**

**//SystemInit(); //called by Startup Code before main(), hence no need to call again.**

**LPC\_GPIO2->FIODIR |= ((1<<2) |(1<<3)); //Configure P2.2 and p2.3 as output pins**

**LPC\_GPIO1->FIODIR &= ~((1<<26)|(1<<24)); //Configure P1.26 and P1.24 as input pin**

**while(1)**

**{**

**if((LPC\_GPIO1->FIOPIN & (1<<26))==0)**

**{**

**delayMS(20); //Debouncing**

**LPC\_GPIO2->FIOPIN ^= (1<<2); //Toggle bit P2.2**

**}**

**if((LPC\_GPIO1->FIOPIN & (1<<24))==0)**

**{**

**delayMS(20); //Debouncing**

**LPC\_GPIO2->FIOPIN ^= (1<<3); //Toggle bit P2.3**

**}**

**}**

**}**

**void initTimer0(void)**

**{**

**/\*Assuming that PLL0 has been setup with CCLK = 100Mhz and PCLK = 25Mhz.\*/**

**LPC\_SC->PCONP |= (1<<1); //Power up TIM0. By default TIM0 and TIM1 are enabled.**

**LPC\_SC->PCLKSEL0 &= ~(0x3<<3); //Set PCLK for timer = CCLK/4 = 100/4 (default)**

**LPC\_TIM0->CTCR = 0x0;**

**LPC\_TIM0->PR = PRESCALE; //Increment LPC\_TIM0->TC at every 24999+1 clock cycles**

**//25000 clock cycles @25Mhz = 1 mS**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**}**

**void delayMS(unsigned int milliseconds) //Using Timer0**

**{**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**LPC\_TIM0->TCR = 0x01; //Enable timer**

**while(LPC\_TIM0->TC < milliseconds); //wait until timer counter reaches the desired delay**

**LPC\_TIM0->TCR = 0x00; //Disable timer**

**}**

**III.4. Lab Experiment 4 :** Write a program to interface LPC1768 GPIO port pin using Keil C to scan P1.26 pin and P1.24, if pin1.26 pin is 0, control 8 leds in the KIT with Led lighting scenario 1 (Led turn on turn by turn). if pin2.24 pin is 0, control 8 leds in the KIT with Led lighting scenario 2 (Led turn on turn by turn).

**#include <lpc17xx.h>**

**#define PRESCALE (25000-1) //25000 PCLK clock cycles to increment TC by 1**

**void delayMS(unsigned int milliseconds);**

**void initTimer0(void);**

**void led\_pattern1();**

**void led\_pattern2();**

**int main(void)**

**{**

**//SystemInit(); //called by Startup Code before main(), hence no need to call again.**

**initTimer0(); //Initialize Timer0**

**LPC\_GPIO2->FIODIR = 0xFFFFFFFF; //Configure P2 as output**

**LPC\_GPIO1->FIODIR = 0xFFFFFFFF; //Configure P1 as output**

**LPC\_GPIO1->FIODIR &= ~((1<<24) | (1<<26)); //Configure P1.24 and P1.26 as input**

**while(1)**

**{**

**if((LPC\_GPIO1->FIOPIN & (1<<24))==0)**

**{**

**led\_pattern1();**

**delayMS(500);**

**}**

**if((LPC\_GPIO1->FIOPIN & (1<<26))==0)**

**{**

**led\_pattern2();**

**delayMS(500);**

**}**

**}**

**//return 0; //normally this wont execute ever**

**}**

**void led\_pattern1()**

**{**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOSET |= (1<<28); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOSET |= (1<<29); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOSET |= (1<<31); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOSET |= (1<<2); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOSET |= (1<<3); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOSET |= (1<<4); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOSET |= (1<<5); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOSET |= (1<<6); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**}**

**void led\_pattern2()**

**{**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOSET |= (1<<28); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOCLR |= (1<<28); //Turn ON LED**

**LPC\_GPIO1->FIOSET |= (1<<29); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOCLR |= (1<<29); //Turn ON LED**

**LPC\_GPIO1->FIOSET |= (1<<31); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOCLR |= (1<<31); //Turn ON LED**

**LPC\_GPIO2->FIOSET |= (1<<2); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<2); //Turn ON LED**

**LPC\_GPIO2->FIOSET |= (1<<3); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<3); //Turn ON LED**

**LPC\_GPIO2->FIOSET |= (1<<4); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<4); //Turn ON LED**

**LPC\_GPIO2->FIOSET |= (1<<5); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<5); //Turn ON LED**

**LPC\_GPIO2->FIOSET |= (1<<6); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<6); //Turn ON LED**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**}**

**void initTimer0(void)**

**{**

**/\*Assuming that PLL0 has been setup with CCLK = 100Mhz and PCLK = 25Mhz.\*/**

**LPC\_SC->PCONP |= (1<<1); //Power up TIM0. By default TIM0 and TIM1 are enabled.**

**LPC\_SC->PCLKSEL0 &= ~(0x3<<3); //Set PCLK for timer = CCLK/4 = 100/4 (default)**

**LPC\_TIM0->CTCR = 0x0;**

**LPC\_TIM0->PR = PRESCALE; //Increment LPC\_TIM0->TC at every 24999+1 clock cycles**

**//25000 clock cycles @25Mhz = 1 mS**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**}**

**void delayMS(unsigned int milliseconds) //Using Timer0**

**{**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**LPC\_TIM0->TCR = 0x01; //Enable timer**

**while(LPC\_TIM0->TC < milliseconds); //wait until timer counter reaches the desired delay**

**LPC\_TIM0->TCR = 0x00; //Disable timer**

**}**

**III.5. Lab Experiment 5 :** Using Array and for loop to optimize the led control algorithm in experiment 4.

**#include <lpc17xx.h>**

**#define PRESCALE (25000-1) //25000 PCLK clock cycles to increment TC by 1**

**void delayMS(unsigned int milliseconds);**

**void initTimer0(void);**

**void led\_pattern1();**

**void led\_pattern2();**

**int p1[] = {28,29,31};**

**int p2[] = {2,3,4,5,6};**

**int main(void)**

**{**

**//SystemInit(); //called by Startup Code before main(), hence no need to call again.**

**initTimer0(); //Initialize Timer0**

**LPC\_GPIO2->FIODIR = 0xFFFFFFFF; //Configure P2 as output**

**LPC\_GPIO1->FIODIR = 0xFFFFFFFF; //Configure P1 as output**

**LPC\_GPIO1->FIODIR &= ~((1<<24) | (1<<26)); //Configure P1.24 and P1.26 as input**

**while(1)**

**{**

**if((LPC\_GPIO1->FIOPIN & (1<<24))==0)**

**{**

**delayMS(500);**

**led\_pattern1();**

**}**

**if((LPC\_GPIO1->FIOPIN & (1<<26))==0)**

**{**

**delayMS(500);**

**led\_pattern2();**

**}**

**}**

**//return 0; //normally this wont execute ever**

**}**

**void led\_pattern1()**

**{**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**delayMS(20); //Debouncing**

**for(int a=0; a<3; a++)**

**{**

**LPC\_GPIO1->FIOSET |= (1<<p1[a]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**}**

**for(int b=0; b<5; b++)**

**{**

**LPC\_GPIO2->FIOSET |= (1<<p2[a]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**}**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**}**

**void led\_pattern2()**

**{**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**delayMS(20); //Debouncing**

**for(int a=0; a<3; a++)**

**{**

**LPC\_GPIO1->FIOSET |= (1<<p1[a]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOCLR |= (1<<p1[a]); //Turn OFF LED**

**}**

**for(int b=0; b<5; b++)**

**{**

**LPC\_GPIO2->FIOSET |= (1<<p2[a]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<p2[a]); //Turn OFF LED**

**}**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**}**

**void initTimer0(void)**

**{**

**/\*Assuming that PLL0 has been setup with CCLK = 100Mhz and PCLK = 25Mhz.\*/**

**LPC\_SC->PCONP |= (1<<1); //Power up TIM0. By default TIM0 and TIM1 are enabled.**

**LPC\_SC->PCLKSEL0 &= ~(0x3<<3); //Set PCLK for timer = CCLK/4 = 100/4 (default)**

**LPC\_TIM0->CTCR = 0x0;**

**LPC\_TIM0->PR = PRESCALE; //Increment LPC\_TIM0->TC at every 24999+1 clock cycles**

**//25000 clock cycles @25Mhz = 1 mS**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**}**

**void delayMS(unsigned int milliseconds) //Using Timer0**

**{**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**LPC\_TIM0->TCR = 0x01; //Enable timer**

**while(LPC\_TIM0->TC < milliseconds); //wait until timer counter reaches the desired delay**

**LPC\_TIM0->TCR = 0x00; //Disable timer**

**}**

**III.6. Lab Experiment 6 :** Write a program to interface LPC1768 GPIO port pin using Keil C to scan P1.23 pin, P2.4, P25 and P1.26, , if pin1.23 pin is 0, control 8 leds in the KIT with Led lighting scenario 1. if pin2.24 pin is 0, control 8 leds in the KIT with Led lighting scenario 2, if pin1.25 pin is 0, control 8 leds in the KIT with Led lighting scenario 3. if pin2.26 pin is 0, control 8 leds in the KIT with Led lighting scenario 4.

**#include <lpc17xx.h>**

**#define PRESCALE (25000-1) //25000 PCLK clock cycles to increment TC by 1**

**#define Joystick\_key\_up 23**

**#define Joystick\_key\_down 24**

**#define Joystick\_key\_left 25**

**#define Joystick\_key\_right 26**

**void delayMS(unsigned int milliseconds);**

**void initTimer0(void);**

**void led\_pattern1();**

**void led\_pattern2();**

**int p1[[]={28,29,31};**

**int p2[] = {2,3,4,5,6};**

**int main(void)**

**{**

**//SystemInit(); //called by Startup Code before main(), hence no need to call again.**

**initTimer0(); //Initialize Timer0**

**LPC\_GPIO2->FIODIR = 0xFFFFFFFF; //Configure P2 as output**

**LPC\_GPIO1->FIODIR = 0xFFFFFFFF; //Configure p1[ as output**

**LPC\_GPIO1->FIODIR &= ~((1<<Joystick\_key\_up)|(1<<Joystick\_key\_down)|(1<<Joystick\_key\_left)|(1<<Joystick\_key\_right)|); //Configure p1.23 to p1.26 as input pins**

**while(1)**

**{**

**if((LPC\_GPIO1->FIOPIN & (1<<Joystick\_key\_up))==0)**

**{**

**delayMS(100);**

**led\_pattern1();**

**}**

**if((LPC\_GPIO1->FIOPIN & (1<<Joystick\_key\_down))==0)**

**{**

**delayMS(100);**

**led\_pattern2();**

**}**

**if((LPC\_GPIO1->FIOPIN & (1<<Joystick\_key\_left))==0)**

**{**

**delayMS(100);**

**led\_pattern3();**

**}**

**if((LPC\_GPIO1->FIOPIN & (1<<Joystick\_key\_right))==0)**

**{**

**delayMS(100);**

**led\_pattern4();**

**}**

**}**

**}**

**void led\_pattern1()**

**{**

**for(int i=0; i<3;i++)**

**{**

**LPC\_GPIO1->FIOSET |= (1<<p1[[i]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**}**

**for(int j=0; j<5;j++)**

**{**

**LPC\_GPIO2->FIOSET |= (1<<p2[j]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**}**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**}**

**void led\_pattern2()**

**{**

**LPC\_GPIO2->FIOSET = 0xFFFFFFFF; //Turn all LED OFF**

**for(int i=0; i<3;i++)**

**{**

**LPC\_GPIO1->FIOCLR |= (1<<p1[[i]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**}**

**for(int j=0; j<5;j++)**

**{**

**LPC\_GPIO2->FIOCLR |= (1<<p2[j]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**}**

**}**

**void led\_pattern3()**

**{**

**for(int i=0; i<3;i++)**

**{**

**LPC\_GPIO1->FIOSET |= (1<<p1[i]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOCLR |= (1<<p1[i]); //Turn ON LED**

**}**

**for(int j=0; j<5;j++)**

**{**

**LPC\_GPIO2->FIOSET |= (1<<p2[j]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<p2[j]); //Turn ON LED**

**}**

**delayMS(500);**

**LPC\_GPIO2->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**LPC\_GPIO1->FIOCLR = 0xFFFFFFFF; //Turn all LED OFF**

**}**

**void led\_pattern4()**

**{**

**for(int j=4; j>=0;j--)**

**{**

**LPC\_GPIO2->FIOSET |= (1<<p2[i]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO2->FIOCLR |= (1<<p2[i]); //Turn ON LED**

**}**

**for(int i=2; i>=0;i--)**

**{**

**LPC\_GPIO1->FIOSET |= (1<<p1[j]); //Turn ON LED**

**delayMS(500); //0.5 Second(s) Delay**

**LPC\_GPIO1->FIOCLR |= (1<<p1[j]); //Turn ON LED**

**}**

**}**

**void initTimer0(void)**

**{**

**/\*Assuming that PLL0 has been setup with CCLK = 100Mhz and PCLK = 25Mhz.\*/**

**LPC\_SC->PCONP |= (1<<1); //Power up TIM0. By default TIM0 and TIM1 are enabled.**

**LPC\_SC->PCLKSEL0 &= ~(0x3<<3); //Set PCLK for timer = CCLK/4 = 100/4 (default)**

**LPC\_TIM0->CTCR = 0x0;**

**LPC\_TIM0->PR = PRESCALE; //Increment LPC\_TIM0->TC at every 24999+1 clock cycles**

**//25000 clock cycles @25Mhz = 1 mS**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**}**

**void delayMS(unsigned int milliseconds) //Using Timer0**

**{**

**LPC\_TIM0->TCR = 0x02; //Reset Timer**

**LPC\_TIM0->TCR = 0x01; //Enable timer**

**while(LPC\_TIM0->TC < milliseconds); //wait until timer counter reaches the desired delay**

**LPC\_TIM0->TCR = 0x00; //Disable timer**

**}**

**IV. LAB REPORT GUIDELINES**

Students write a report which includes : algorithm flowchart and C++ Code for each experiment. In each line of the code give the comments for the meaning of each register value using this code line.