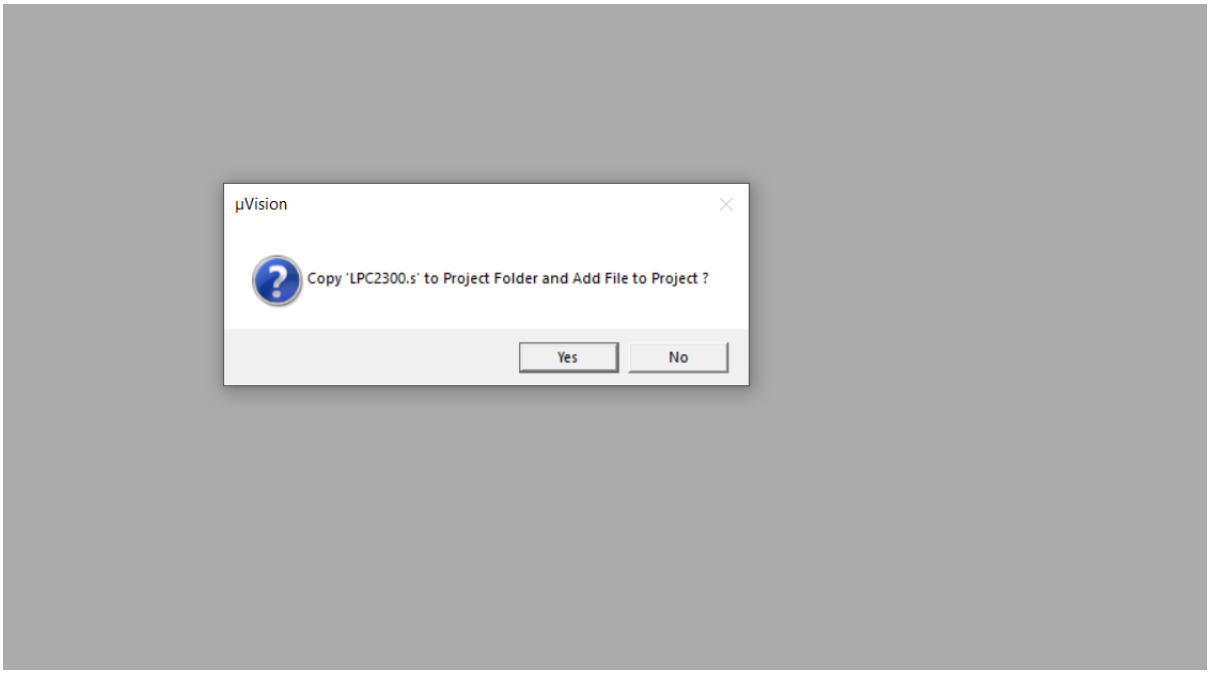
EXTRA SUPPLEMENT MATERIAL FOR LAB 4

The purpose of this Lab Extra Material is to make your lab experience smoother.

After going through the lab-4 handout please do watch the following video;

https://drive.google.com/file/d/1vBmvLss6gpJ-J4nzH6ooPNBlVi0d1Grs/view?ts=617fb412



While creating project in Keil, Do not forget to click Yes. It is needed and mandatory when we are writing in C

The following content is from LPC23xx Manual that is shared with you. As it is difficult for you to look into the entire manual, we are sharing the important tables here. Please look into the manual for additional information.

Pin Function Select register 0 (PINSEL0 - 0xE002 C000):

The PINSEL0 register controls the functions of the pins. The direction control bit in the IO0DIR register (or the FIO0DIR register if the enhanced GPIO function is selected for port 0) is effective only when the GPIO function is selected for a pin. For other functions, the direction is controlled automatically

9.5.1.2 144-pin packages

Table 106. Pin function select register 0 (PINSEL0 - address 0xE002 C000) bit description (LPC2377/78 and LPC2388)

PINSEL0	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P0.0	GPIO Port 0.0	RD1[1]	TXD3	SDA1	00
3:2	P0.1	GPIO Port 0.1	TD1[1]	RXD3	SCL1	00
5:4	P0.2	GPIO Port 0.2	TXD0	Reserved	Reserved	00
7:6	P0.3	GPIO Port 0.3	RXD0	Reserved	Reserved	00
9:8	P0.4	GPIO Port 0.4	I2SRX_CLK	RD2[1]	CAP2.0	00
11:10	P0.5	GPIO Port 0.5	I2SRX_WS	TD2[1]	CAP2.1	00
13:12	P0.6	GPIO Port 0.6	I2SRX_SDA	SSEL1	MAT2.0	00
15:14	P0.7	GPIO Port 0.7	I2STX_CLK	SCK1	MAT2.1	00
17:16	P0.8	GPIO Port 0.8	I2STX_WS	MISO1	MAT2.2	00
19:18	P0.9	GPIO Port 0.9	I2STX_SDA	MOSI1	MAT2.3	00
21:20	P0.10	GPIO Port 0.10	TXD2	SDA2	MAT3.0	00
23:22	P0.11	GPIO Port 0.11	RXD2	SCL2	MAT3.1	00
25:24	P0.12	GPIO Port 0.12	$\overline{\text{USB\_PPWR2}}$[2]	MISO1	AD0.6	00
27:26	P0.13	GPIO Port 0.13	USB_UP_LED2[1]	MOSI1	AD0.7	00
29:28	P0.14	GPIO Port 0.14	$\overline{\text{USB\_HSTEN2}}$[2]	USB_CONNECT2[1]	SSEL1	00
31:30	P0.15	GPIO Port 0.15	TXD1	SCK0	SCK	00

Chapter 10 of LPC23XX Manual (especially from Page 170-180)

10.4 Pin description

Table 131. GPIO pin description

Pin Name	Type	Description
P0.[31:0] P1.[31:0] P2.[31:0] P3.[31:0] P4.[31:0]	Input/ Output	General purpose input/output. These are typically shared with other peripherals functions and will therefore not all be available in an application. Packaging options may affect the number of GPIOs available in a particular device (see Table 2 and Table 101). Some pins may be limited by requirements of the alternate functions of the pin. For example, the pins containing the I^2C0 function are open-drain for any function of that pin. Details may be found in the LPC2300 pin description.

10.5 Register description

LPC2300 has up to five 32-bit General Purpose I/O ports. PORT0 and PORT1 are controlled via two groups of registers as shown in Table 132 and Table 133. Apart from them, LPC2300 can have three additional 32-bit ports, PORT2, PORT3 and PORT4. Details on a specific GPIO port usage can be found in Section 8.1 and Section 9.5.

Legacy registers shown in Table 132 allow backward compatibility with earlier family devices, using existing code. The functions and relative timing of older GPIO implementations is preserved. Only PORT0 and PORT1 can be controlled via the legacy port registers.

The registers in Table 133 represent the enhanced GPIO features available on all of the LPC2300's GPIO ports. These registers are located directly on the local bus of the CPU for the fastest possible read and write timing. They can be accessed as byte or half-word long data, too. A mask register allows access to a group of bits in a single GPIO port independently from other bits in the same port.

When PORT0 and PORT1 are used, user must select whether these ports will be accessed via registers that provide enhanced features or a legacy set of registers (see Section 3.7 "Other system controls and status flags" on page 38). While both of a port's fast and legacy GPIO registers are controlling the same physical pins, these two port control branches are mutually exclusive and operate independently. For example, changing a pin's output via a fast register will not be observable via the corresponding legacy register.

The following text will refer to the legacy GPIO as "the slow" GPIO, while GPIO equipped with the enhanced features will be referred as "the fast" GPIO.

Table 132. GPIO register map (legacy APB accessible registers)

Generic Name	Description	Access	Reset value[1]	PORTn Register Address & Name
IOPIN	GPIO Port Pin value register. The current state of the GPIO configured port pins can always be read from this register, regardless of pin direction. By writing to this register port's pins will be set to the desired level instantaneously.	R/W	NA	IO0PIN - 0xE002 8000 IO1PIN - 0xE002 8010
IOSET	GPIO Port Output Set register. This register controls the state of output pins in conjunction with the IOCLR register. Writing ones produces highs at the corresponding port pins. Writing zeroes has no effect.	R/W	0x0	IO0SET - 0xE002 8004 IO1SET - 0xE002 8014
IODIR	GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0x0	IO0DIR - 0xE002 8008 IO1DIR - 0xE002 8018
IOCLR	GPIO Port Output Clear register. This register controls the state of output pins. Writing ones produces lows at the corresponding port pins and clears the corresponding bits in the IOSET register. Writing zeroes has no effect.	WO	0x0	IO0CLR - 0xE002 800C IO1CLR - 0xE002 801C

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Table 133. GPIO register map (local bus accessible registers - enhanced GPIO features)

Generic Name	Description	Access	Reset value[1]	PORTn Register Address & Name
FIODIR	Fast GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0x0	FIO0DIR - 0x3FFF C000 FIO1DIR - 0x3FFF C020 FIO2DIR - 0x3FFF C040 FIO3DIR - 0x3FFF C060 FIO4DIR - 0x3FFF C080
FIOMASK	Fast Mask register for port. Writes, sets, clears, and reads to port (done via writes to FIOPIN, FIOSET, and FIOCLR, and reads of FIOPIN) alter or return only the bits enabled by zeros in this register.	R/W	0x0	FIO0MASK - 0x3FFF C010 FIO1MASK - 0x3FFF C030 FIO2MASK - 0x3FFF C050 FIO3MASK - 0x3FFF C070 FIO4MASK - 0x3FFF C090
FIOPIN	Fast Port Pin value register using FIOMASK. The current state of digital port pins can be read from this register, regardless of pin direction or alternate function selection (as long as pins are not configured as an input to ADC). The value read is masked by ANDing with inverted FIOMASK. Writing to this register places corresponding values in all bits enabled by zeros in FIOMASK. **Important:** if a FIOPIN register is read, its bit(s) masked with 1 in the FIOMASK register will be set to 0 regardless of the physical pin state.	R/W	0x0	FIO0PIN - 0x3FFF C014 FIO1PIN - 0x3FFF C034 FIO2PIN - 0x3FFF C054 FIO3PIN - 0x3FFF C074 FIO4PIN - 0x3FFF C094
FIOSET	Fast Port Output Set register using FIOMASK. This register controls the state of output pins. Writing 1s produces highs at the corresponding port pins. Writing 0s has no effect. Reading this register returns the current contents of the port output register. Only bits enabled by 0 in FIOMASK can be altered.	R/W	0x0	FIO0SET - 0x3FFF C018 FIO1SET - 0x3FFF C038 FIO2SET - 0x3FFF C058 FIO3SET - 0x3FFF C078 FIO4SET - 0x3FFF C098
FIOCLR	Fast Port Output Clear register using FIOMASK0. This register controls the state of output pins. Writing 1s produces lows at the corresponding port pins. Writing 0s has no effect. Only bits enabled by 0 in FIOMASK0 can be altered.	WO	0x0	FIO0CLR - 0x3FFF C01C FIO1CLR - 0x3FFF C03C FIO2CLR - 0x3FFF C05C FIO3CLR - 0x3FFF C07C FIO4CLR - 0x3FFF C09C

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

FIOxDIR INSTRUCTION Instructions:

10.5.1 GPIO port Direction register IODIR and FIODIR(IO[0/1]DIR - 0xE002 80[0/1]8 and FIO[0/1/2/3/4]DIR - 0x3FFF C0[0/2/4/6/8]0)

This word accessible register is used to control the direction of the pins when they are configured as GPIO port pins. Direction bit for any pin must be set according to the pin functionality.

Remark: GPIO pins P0.29 and P0.30 are shared with the USB D+/– pins and must have the same direction. If either P0DIR bits 29 **or** 30 are configured LOW in the IO0DIR or FIO0DIR registers, both, P0.29 and P0.30, are inputs. If both, P0DIR bit 29 **and** bit 30 are HIGH, both, P0.29 and P0.30, are outputs.

Legacy registers are the IO0DIR and IO1DIR while the enhanced GPIO functions are supported via the FIO0DIR, FIO1DIR, FIO2DIR, FIO3DIR and FIO4DIR registers.

Table 135. GPIO port Direction register (IO0DIR - address 0xE002 8008 and IO1DIR - address 0xE002 8018) bit description

Bit	Symbol	Value	Description	Reset value
31:0	P0xDIR or P1xDIR		Slow GPIO Direction PORTx control bits. Bit 0 in IOxDIR controls pin Px.0, bit 31 IOxDIR controls pin Px.31.	0x0
		0	Controlled pin is an input pin.	
		1	Controlled pin is an output pin.	

Table 136. Fast GPIO port Direction register (FIO[0/1/2/3/4]DIR - address 0x3FFF C0[0/2/4/6/8]0) bit description

Bit	Symbol	Value	Description	Reset value
31:0	FP0xDIR FP1xDIR FP2xDIR FP3xDIR FP4xDIR		Fast GPIO Direction PORTx control bits. Bit 0 in FIOxDIR controls pin Px.0, bit 31 in FIOxDIR controls pin Px.31.	0x0
		0	Controlled pin is input.	
		1	Controlled pin is output.	

Aside from the 32-bit long and word only accessible FIODIR register, every fast GPIO port can also be controlled via several byte and half-word accessible registers listed in Table 137, too. Next to providing the same functions as the FIODIR register, these additional registers allow easier and faster access to the physical port pins.

Table 137. Fast GPIO port Direction control byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxDIR0	Fast GPIO Port x Direction control register 0. Bit 0 in FIOxDIR0 register corresponds to pin Px.0 ... bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0DIR0 - 0x3FFF C000 FIO1DIR0 - 0x3FFF C020 FIO2DIR0 - 0x3FFF C040 FIO3DIR0 - 0x3FFF C060 FIO4DIR0 - 0x3FFF C080
FIOxDIR1	Fast GPIO Port x Direction control register 1. Bit 0 in FIOxDIR1 register corresponds to pin Px.8 ... bit 7 to pin Px.15.	8 (byte) R/W	0x00	FIO0DIR1 - 0x3FFF C001 FIO1DIR1 - 0x3FFF C021 FIO2DIR1 - 0x3FFF C041 FIO3DIR1 - 0x3FFF C061 FIO4DIR1 - 0x3FFF C081
FIO0DIR2	Fast GPIO Port x Direction control register 2. Bit 0 in FIOxDIR2 register corresponds to pin Px.16 ... bit 7 to pin Px.23.	8 (byte) R/W	0x00	FIO0DIR2 - 0x3FFF C002 FIO1DIR2 - 0x3FFF C022 FIO2DIR2 - 0x3FFF C042 FIO3DIR2 - 0x3FFF C062 FIO4DIR2 - 0x3FFF C082
FIOxDIR3	Fast GPIO Port x Direction control register 3. Bit 0 in FIOxDIR3 register corresponds to pin Px.24 ... bit 7 to pin Px.31.	8 (byte) R/W	0x00	FIO0DIR3 - 0x3FFF C003 FIO1DIR3 - 0x3FFF C023 FIO2DIR3 - 0x3FFF C043 FIO3DIR3 - 0x3FFF C063 FIO4DIR3 - 0x3FFF C083
FIOxDIRL	Fast GPIO Port x Direction control Lower half-word register. Bit 0 in FIOxDIRL register corresponds to pin Px.0 ... bit 15 to pin Px.15.	16 (half-word) R/W	0x0000	FIO0DIRL - 0x3FFF C000 FIO1DIRL - 0x3FFF C020 FIO2DIRL - 0x3FFF C040 FIO3DIRL - 0x3FFF C060 FIO4DIRL - 0x3FFF C080
FIOxDIRU	Fast GPIO Port x Direction control Upper half-word register. Bit 0 in FIOxDIRU register corresponds to Px.16 ... bit 15 to Px.31.	16 (half-word) R/W	0x0000	FIO0DIRU - 0x3FFF C002 FIO1DIRU - 0x3FFF C022 FIO2DIRU - 0x3FFF C042 FIO3DIRU - 0x3FFF C062 FIO4DIRU - 0x3FFF C082

Table 146. Fast GPIO port Pin value byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxPIN0	Fast GPIO Port x Pin value register 0. Bit 0 in FIOxPIN0 register corresponds to pin Px.0 ... bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0PIN0 - 0x3FFF C014 FIO1PIN0 - 0x3FFF C034 FIO2PIN0 - 0x3FFF C054 FIO3PIN0 - 0x3FFF C074 FIO4PIN0 - 0x3FFF C094
FIOxPIN1	Fast GPIO Port x Pin value register 1. Bit 0 in FIOxPIN1 register corresponds to pin Px.8 ... bit 7 to pin Px.15.	8 (byte) R/W	0x00	FIO0PIN1 - 0x3FFF C015 FIO1PIN1 - 0x3FFF C035 FIO2PIN1 - 0x3FFF C055 FIO3PIN1 - 0x3FFF C075 FIO4PIN1 - 0x3FFF C095
FIOxPIN2	Fast GPIO Port x Pin value register 2. Bit 0 in FIOxPIN2 register corresponds to pin Px.16 ... bit 7 to pin Px.23.	8 (byte) R/W	0x00	FIO0PIN2 - 0x3FFF C016 FIO1PIN2 - 0x3FFF C036 FIO2PIN2 - 0x3FFF C056 FIO3PIN2 - 0x3FFF C076 FIO4PIN2 - 0x3FFF C096
FIOxPIN3	Fast GPIO Port x Pin value register 3. Bit 0 in FIOxPIN3 register corresponds to pin Px.24 ... bit 7 to pin Px.31.	8 (byte) R/W	0x00	FIO0PIN3 - 0x3FFF C017 FIO1PIN3 - 0x3FFF C037 FIO2PIN3 - 0x3FFF C057 FIO3PIN3 - 0x3FFF C077 FIO4PIN3 - 0x3FFF C097
FIOxPINL	Fast GPIO Port x Pin value Lower half-word register. Bit 0 in FIOxPINL register corresponds to pin Px.0 ... bit 15 to pin Px.15.	16 (half-word) R/W	0x0000	FIO0PINL - 0x3FFF C014 FIO1PINL - 0x3FFF C034 FIO2PINL - 0x3FFF C054 FIO3PINL - 0x3FFF C074 FIO4PINL - 0x3FFF C094
FIOxPINU	Fast GPIO Port x Pin value Upper half-word register. Bit 0 in FIOxPINU register corresponds to pin Px.16 ... bit 15 to Px.31.	16 (half-word) R/W	0x0000	FIO0PINU - 0x3FFF C016 FIO1PINU - 0x3FFF C036 FIO2PINU - 0x3FFF C056 FIO3PINU - 0x3FFF C076 FIO4PINU - 0x3FFF C096

10.6 GPIO usage notes

10.6.1 Example 1: sequential accesses to IOSET and IOCLR affecting the same GPIO pin/bit

State of the output configured GPIO pin is determined by writes into the pin's port IOSET and IOCLR registers. Last of these accesses to the IOSET/IOCLR register will determine the final output of a pin.

In the example code:

```
IO0DIR = 0x0000 0080 ;pin P0.7 configured as output
IO0CLR = 0x0000 0080 ;P0.7 goes LOW
IO0SET = 0x0000 0080 ;P0.7 goes HIGH
IO0CLR = 0x0000 0080 ;P0.7 goes LOW
```

pin P0.7 is configured as an output (write to IO0DIR register). After this, P0.7 output is set to low (first write to IO0CLR register). Short high pulse follows on P0.7 (write access to IO0SET), and the final write to IO0CLR register sets pin P0.7 back to low level.

10.6.2 Example 2: an instantaneous output of 0s and 1s on a GPIO port

Write access to port's IOSET followed by write to the IOCLR register results with pins outputting 0s being slightly later then pins outputting 1s. There are systems that can tolerate this delay of a valid output, but for some applications simultaneous output of a binary content (mixed 0s and 1s) within a group of pins on a single GPIO port is required. This can be accomplished by writing to the port's IOPIN register.

Following code will preserve existing output on PORT0 pins P0.[31:16] and P0.[7:0] and at the same time set P0.[15:8] to 0xA5, regardless of the previous value of pins P0.[15:8]:

```
IO0PIN = (IO0PIN && 0xFFFF00FF) || 0x0000A500
```

The same outcome can be obtained using the fast port access.

Solution 1: using 32-bit (word) accessible fast GPIO registers

```
FIO0MASK = 0xFFFF00FF;
FIO0PIN  = 0x0000A500;
```

Solution 2: using 16-bit (half-word) accessible fast GPIO registers

```
FIO0MASKL = 0x00FF;
FIO0PINL  = 0xA500;
```

Solution 3: using 8-bit (byte) accessible fast GPIO registers

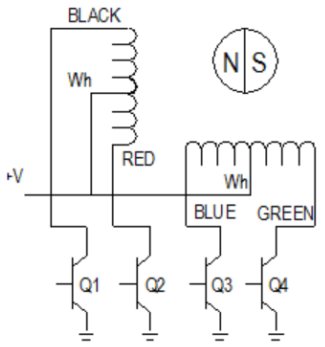
```
FIO0PIN1  = 0xA5;
```

10.6.3 Writing to IOSET/IOCLR vs. IOPIN

Write to the IOSET/IOCLR register allows easy change of the port's selected output pin(s) to high/low level at a time. Only pin/bit(s) in the IOSET/IOCLR written with 1 will be set to high/low level, while those written as 0 will remain unaffected. However, by just writing to either IOSET or IOCLR register it is not possible to instantaneously output arbitrary binary data containing a mixture of 0s and 1s on a GPIO port.

Write to the IOPIN register enables instantaneous output of a desired content on the parallel GPIO. Binary data written into the IOPIN register will affect all output configured pins of that parallel port: 0s in the IOPIN will produce low level pin outputs and 1s in IOPIN will produce high level pin outputs. In order to change output of only a group of port's pins, application must logically AND readout from the IOPIN with mask containing 0s in bits corresponding to pins that will be changed, and 1s for all others. Finally, this result has to be logically ORred with the desired content and stored back into the IOPIN register. Example 2 from above illustrates output of 0xA5 on PORT0 pins 15 to 8 while preserving all other PORT0 output pins as they were before.

STEPPER MOTOR:



CW Rotation ⟶

Unipolar Step	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF

⟵ CCW Rotation

Figure 5: Wiring diagram and step sequence for unipolar motor

Example program for first question:

Example – 2 Program For 8-Way DIP Switch Interface

```
#include <iolpc2378.h>
#include "irq.h"
#include "config.h"

unsigned int k;
unsigned int dat[] = {0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
void delay()
{
  unsigned int i,j;
  for(i=0;i<0x1Fff;i++)
    for(j=0;j<0xff;j++);
}
int main (void)                          /*Main function*/
{
  TargetResetInit();
  FIO4PIN = 0X0;
  FIO4DIR = 0XFFFF00FF;
    while(1)
  {
     FIO4PIN = FIO4PIN >> 0X08;
   }

}
```