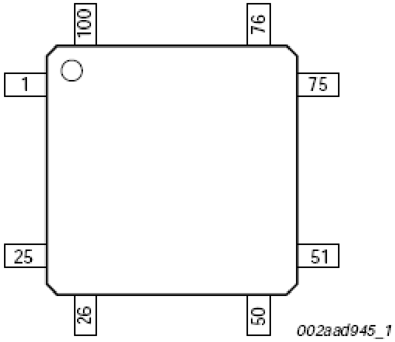
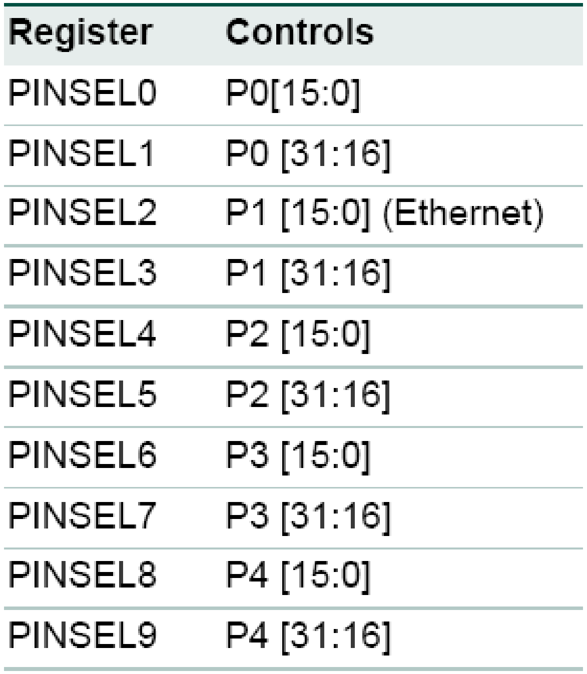
LPC 1768 PIN CONFIGURATION



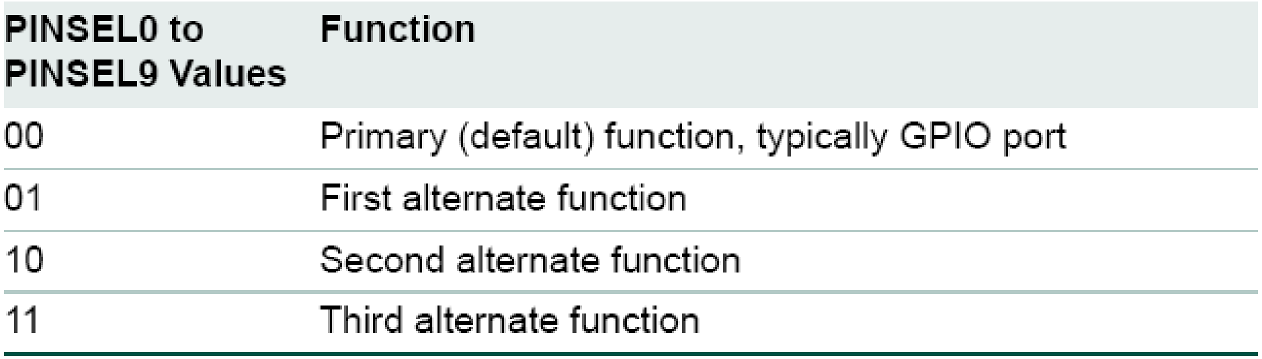
PX.Y – Port X Pin Y

P0[0] to P0[31]	I/O	**Port 0:** Port 0 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 0 pins depends upon the pin function selected via the pin connect block. Pins 12, 13, 14, and 31 of this port are not available.
P1[0] to P1[31]	I/O	**Port 1:** Port 1 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Pins 2, 3, 5, 6, 7, 11, 12, and 13 of this port are not available.
P2[0] to P2[31]	I/O	**Port 2:** Port 2 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 2 pins depends upon the pin function selected via the pin connect block. Pins 14 through 31 of this port are not available.
P3[0] to P3[31]	I/O	**Port 3:** Port 3 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 3 pins depends upon the pin function selected via the pin connect block. Pins 0 through 24, and 27 through 31 of this port are not available.
P4[0] to P4[31]	I/O	**Port 4:** Port 4 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 4 pins depends upon the pin function selected via the pin connect block. Pins 0 through 27, 30, and 31 of this port are not available.

PIN CONNECT BLOCK

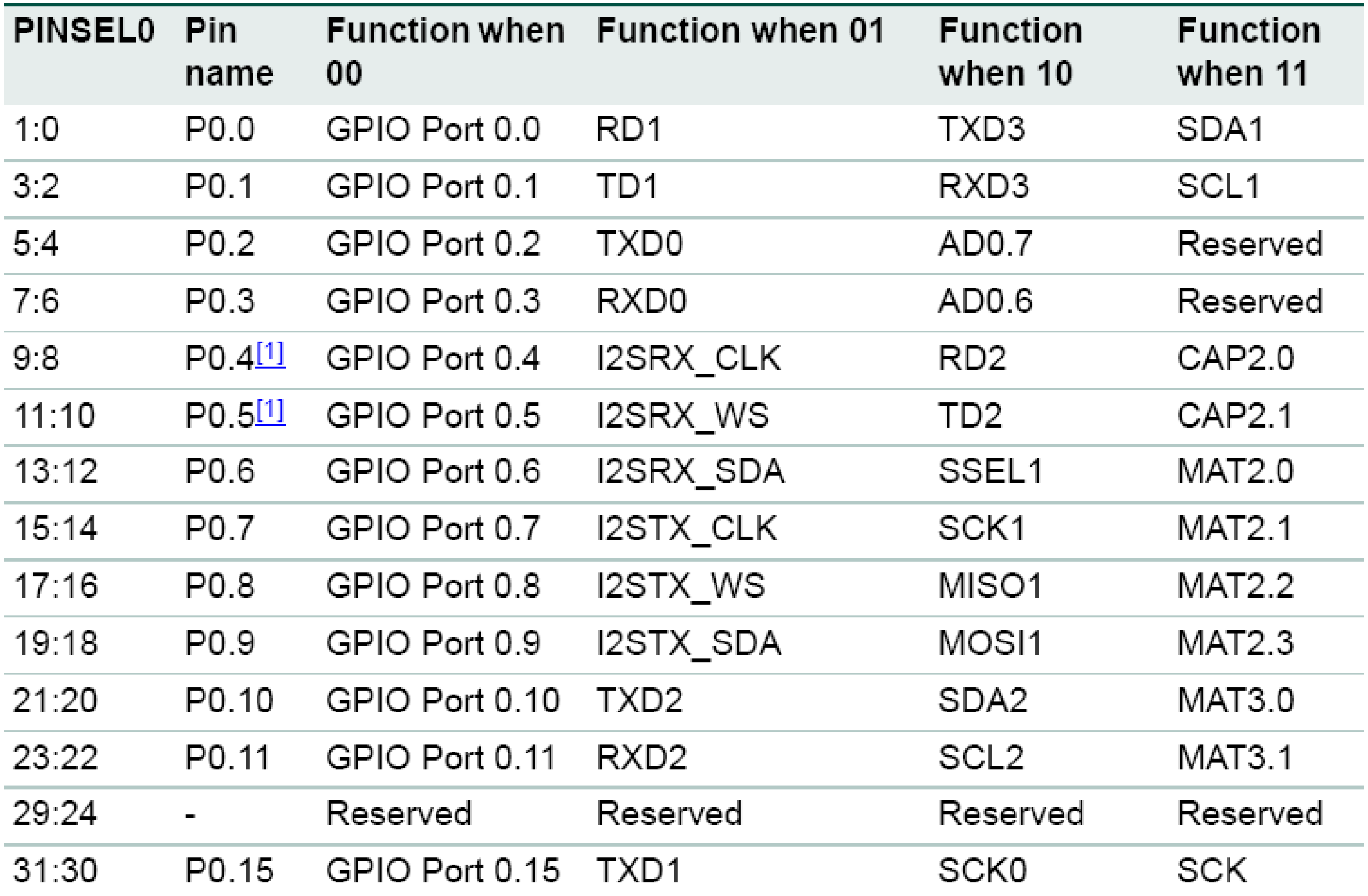


Register	Controls
PINSEL0	P0[15:0]
PINSEL1	P0 [31:16]
PINSEL2	P1 [15:0] (Ethernet)
PINSEL3	P1 [31:16]
PINSEL4	P2 [15:0]
PINSEL5	P2 [31:16]
PINSEL6	P3 [15:0]
PINSEL7	P3 [31:16]
PINSEL8	P4 [15:0]
PINSEL9	P4 [31:16]



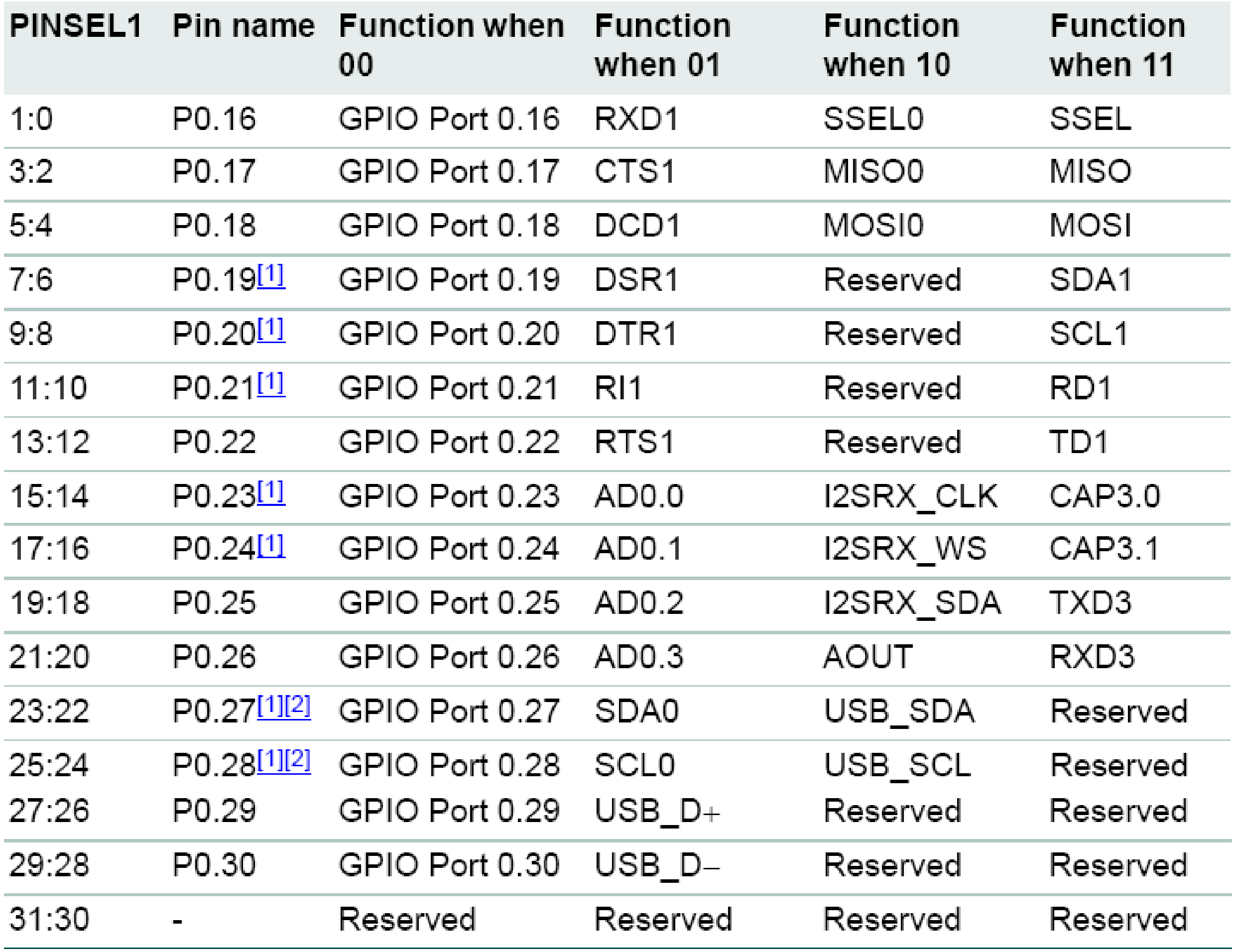
PINSEL0 to PINSEL9 Values	Function
00	Primary (default) function, typically GPIO port
01	First alternate function
10	Second alternate function
11	Third alternate function

PIN CONNECT BLOCK



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

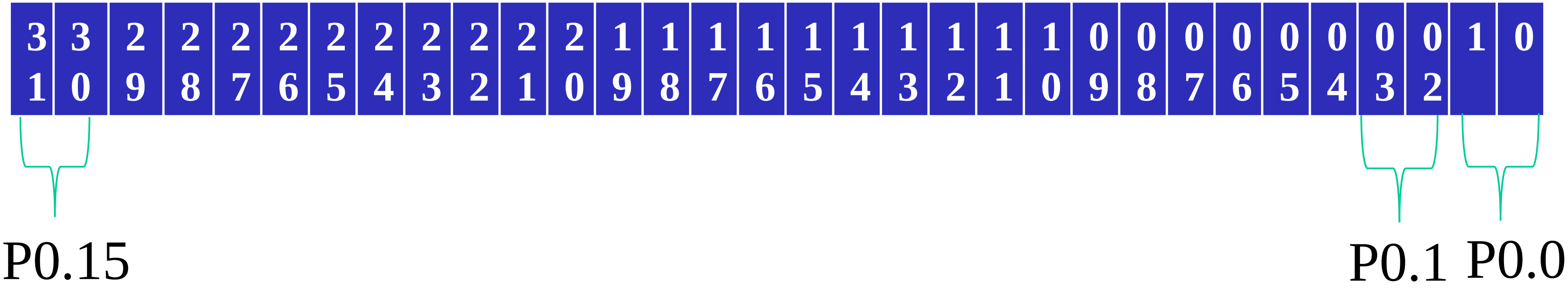
PIN CONNECT BLOCK



PINSEL1	Pin name	Function when 00	Function when 01	Function when 10	Function when 11
1:0	P0.16	GPIO Port 0.16	RXD1	SSEL0	SSEL
3:2	P0.17	GPIO Port 0.17	CTS1	MISO0	MISO
5:4	P0.18	GPIO Port 0.18	DCD1	MOSI0	MOSI
7:6	P0.19[1]	GPIO Port 0.19	DSR1	Reserved	SDA1
9:8	P0.20[1]	GPIO Port 0.20	DTR1	Reserved	SCL1
11:10	P0.21[1]	GPIO Port 0.21	RI1	Reserved	RD1
13:12	P0.22	GPIO Port 0.22	RTS1	Reserved	TD1
15:14	P0.23[1]	GPIO Port 0.23	AD0.0	I2SRX_CLK	CAP3.0
17:16	P0.24[1]	GPIO Port 0.24	AD0.1	I2SRX_WS	CAP3.1
19:18	P0.25	GPIO Port 0.25	AD0.2	I2SRX_SDA	TXD3
21:20	P0.26	GPIO Port 0.26	AD0.3	AOUT	RXD3
23:22	P0.27[1][2]	GPIO Port 0.27	SDA0	USB_SDA	Reserved
25:24	P0.28[1][2]	GPIO Port 0.28	SCL0	USB_SCL	Reserved
27:26	P0.29	GPIO Port 0.29	USB_D+	Reserved	Reserved
29:28	P0.30	GPIO Port 0.30	USB_D−	Reserved	Reserved
31:30	-	Reserved	Reserved	Reserved	Reserved

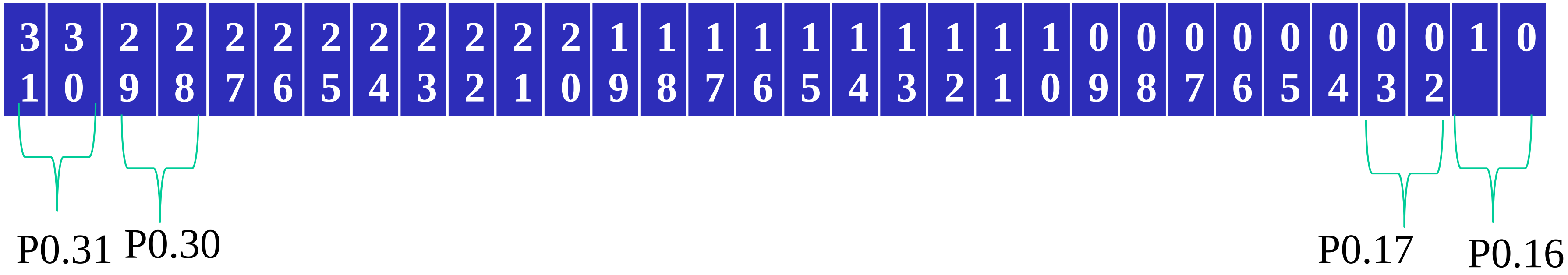
PIN CONNECT BLOCK

PINSEL0



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	1	0

PINSEL1



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	1	0

Configure P0.1 with Function-01 and P0.15 with Function-02, Value to be loaded to PINSEL0:

10000000000000000000000000000100 = 0x80000004 = (2<<30) | (1<<2)= (2<<(15-0)*2 | (1<<(1-0)*2)

Configure P0.17 with Function-03 and P0.30 with Function-01, Value to be loaded to PINSEL0:

00010000000000000000000000001100 = 0x1000000C = (1<<28) | (3<<2)
= (1<< (30-16)*2) | (3<<(17-16)*2)

GPIO (General Purpose Input/Output)

FIODIR Fast GPIO Port Direction control register. This register individually controls the direction of each port pin.

0	Controlled pin is input.
1	Controlled pin is output.

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 an FIOPIN register is read, its bit(s) masked with 1 in the FIOMASK register will be read as 0 regardless of the physical pin state.

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.

0	Controlled pin output is unchanged.
1	Controlled pin output is set to HIGH.

FIOCLR Fast Port Output Clear register using FIOMASK. 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 FIOMASK can be altered.

0	Controlled pin output is unchanged.
1	Controlled pin output is set to LOW.

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.

0	Controlled pin is affected by writes to the port's FIOxSET, FIOxCLR, and FIOxPIN register(s). Current state of the pin can be read from the FIOxPIN register.
1	Controlled pin is not affected by writes into the port's FIOxSET, FIOxCLR and FIOxPIN register(s). When the FIOxPIN register is read, this bit will not be updated with the state of the physical pin.

GPIO (General Purpose Input/Output)

FIOxDIR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	1	0

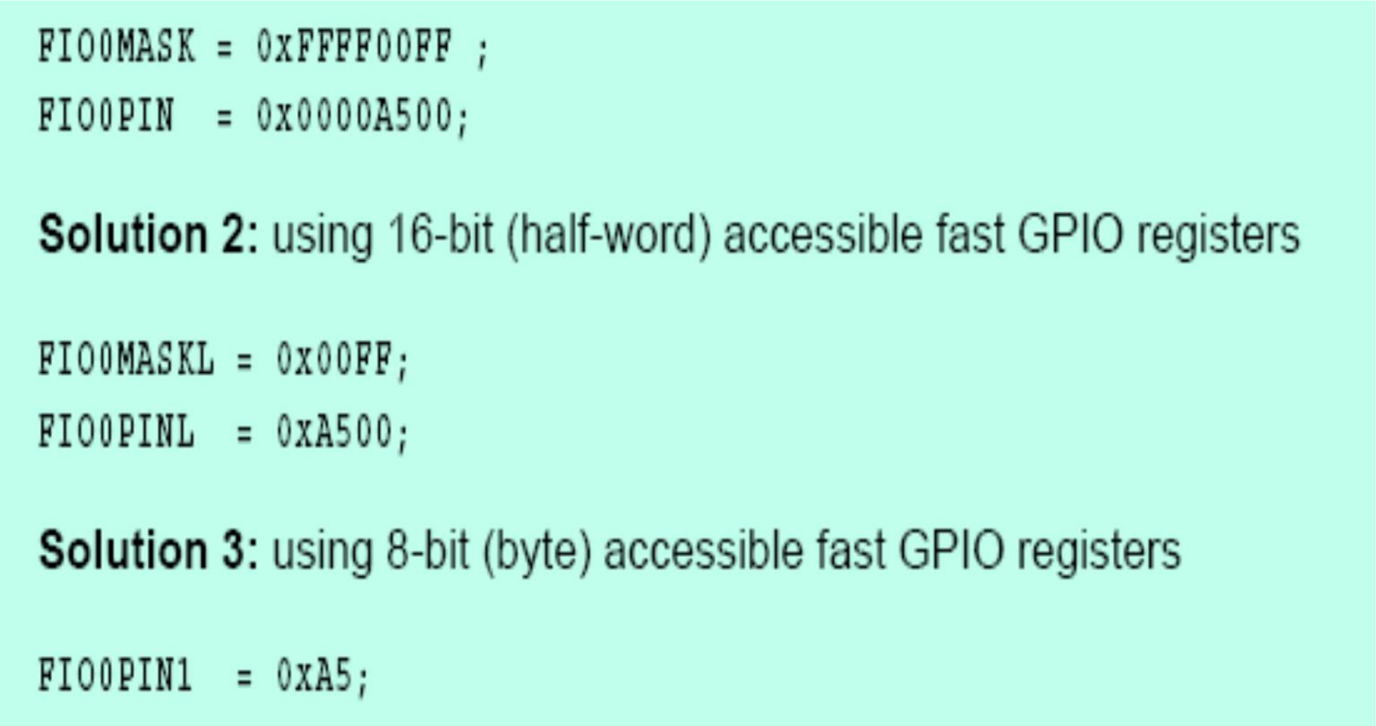
FIOxDIRH	FIOxDIRL

FIOxDIR3	FIOxDIR2	FIOxDIR1	FIOxDIR0

Same concept applicable to FIOxSET, FIOxCLR, FIOxPIN, FIOxMASK (Ex: For Port-1 , we can have FIO1SET, FIO1SETH, FIO1SETL, FIO1SET3, FIO1SET2, FIO1SET1, FIO1SET0……

GPIO (General Purpose Input/Output)

Ex: Send 0xA5 to P0.15-P0.8 without affecting values on the remaining pins.
This can be accomplished in several ways



```
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

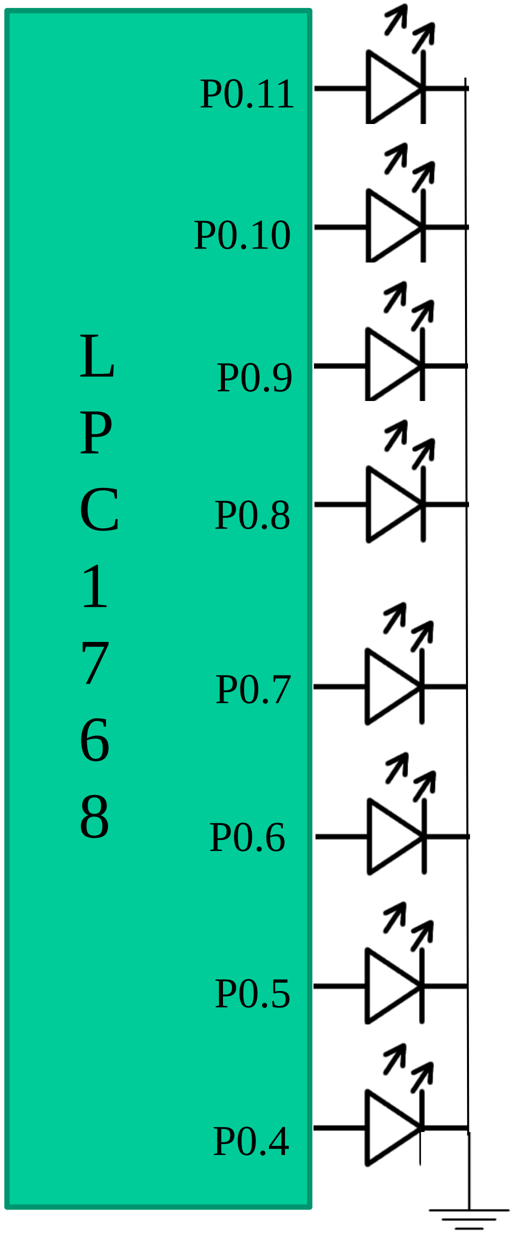
```
FIO0PIN1  = 0xA5;
```

GPIO (General Purpose Input/Output)

Write an embedded C program to turn ON and OFF LEDs connected to P0.11 – P0.4

```
#include <LPC17xx.h>
 unsigned int j;
 unsigned long LED = 0x00000FF0;

 int main(void)
 {
        SystemInit();
        SystemCoreClockUpdate();

        LPC_PINCON->PINSEL0 =  0x00000000; // P0.15-P0.0 GPIO
        LPC_GPIO0->FIODIR = 0x00000FF0; // P0.11-P0.4 as output

        while(1)
        {
                LPC_GPIO0->FIOSET = LED; // SET P0.11-P0.4
                for(j=0;j<10000;j++); // Delay

                LPC_GPIO0->FIOCLR = LED; // CLEAR P0.11-P0.4
                for(j=0;j<10000;j++); //Delay

        }
 }
```

```
LPC_GPIO0->FIOPIN= ~(LPC_GPIO0->FIOPIN & 0x00000FF0);
for(j=0;j<10000;j++); //Delay
```

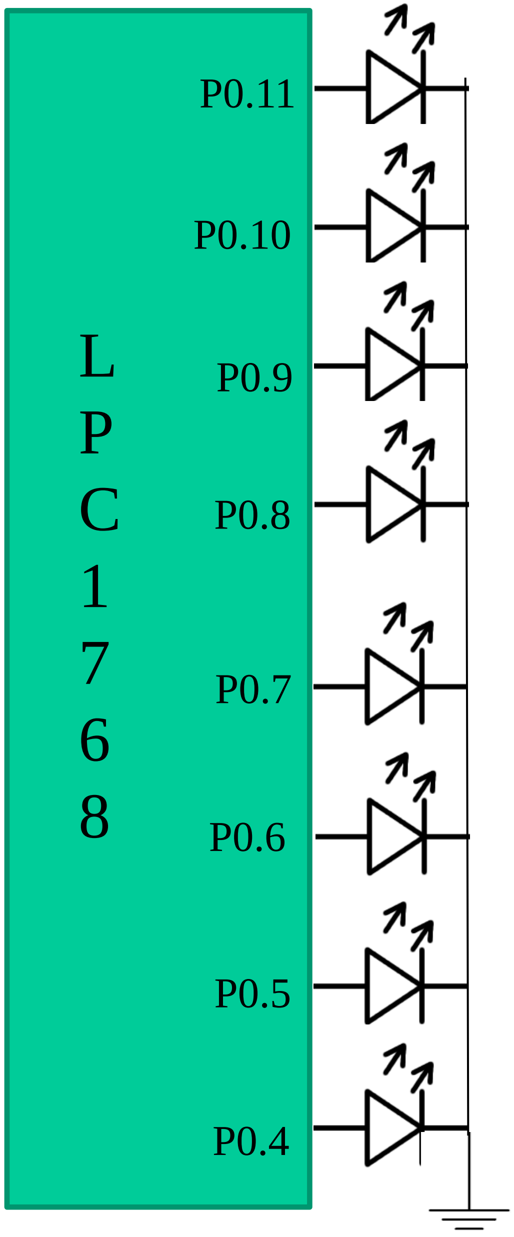


GPIO (General Purpose Input/Output)

Write an embedded C program to turn ON and OFF LEDs connected to P0.11 – P0.4

```
#include <LPC17xx.h>
 unsigned int j;
 unsigned int LED = 0x0FF0;

 int main(void)
 {
        SystemInit();
        SystemCoreClockUpdate();

        LPC_PINCON->PINSEL0 =  0x00000000; // P0.15-P0.0 GPIO
        LPC_GPIO0->FIODIRL = 0x0FF0; // P0.11-P0.4 as output

        while(1)
        {
                LPC_GPIO0->FIOSETL = LED; // SET P0.11-P0.4
                for(j=0;j<10000;j++); // Delay

                LPC_GPIO0->FIOCLRL = LED; // CLEAR P0.11-P0.4
                for(j=0;j<10000;j++); //Delay

        }
 }
```

```
LPC_GPIO0->FIOPINL= ~(LPC_GPIO0->FIOPINL & 0x0FF0);
for(j=0;j<10000;j++); //Delay
```



GPIO (General Purpose Input/Output)

8 bit Johnson Counter on LEDs

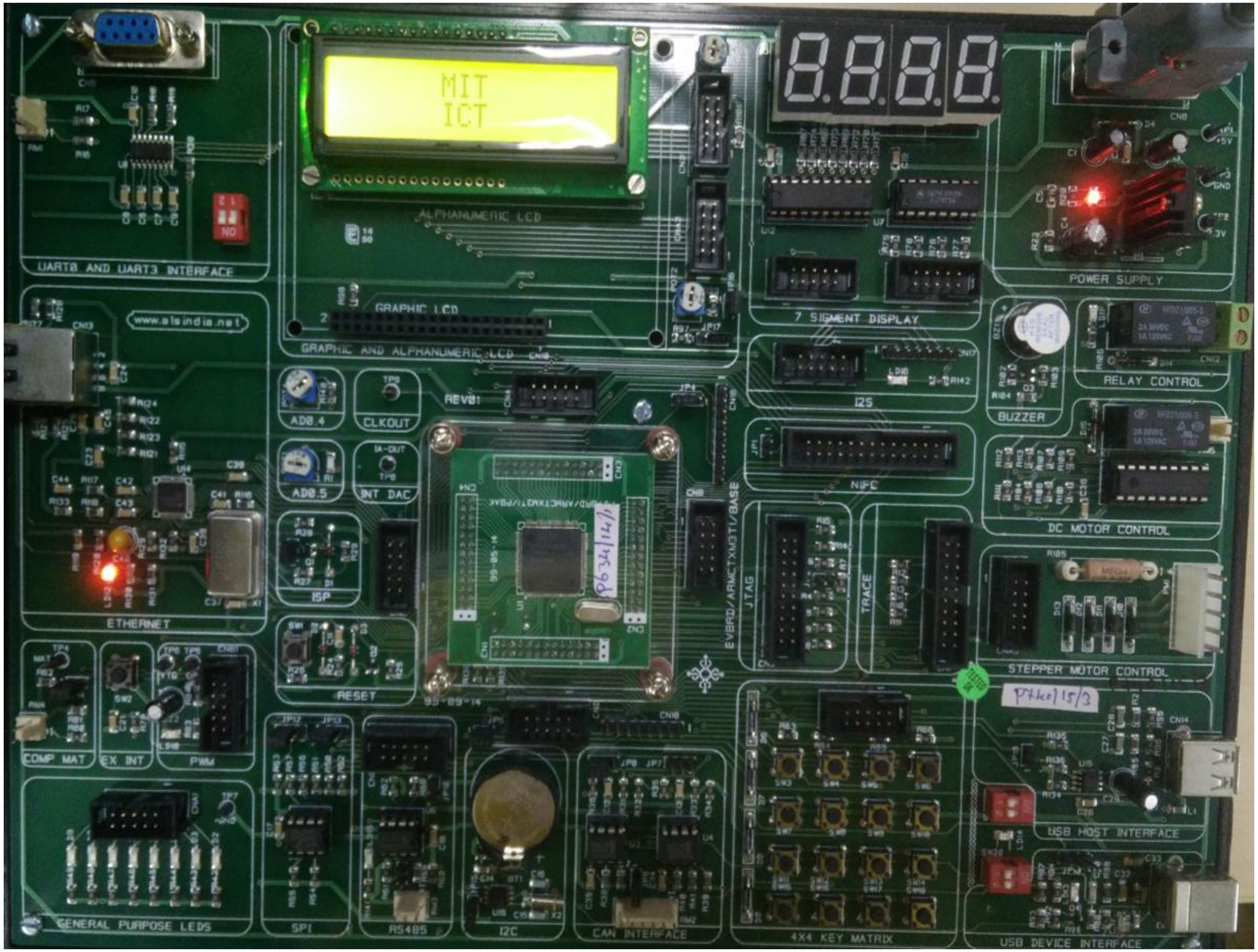
```
#include <LPC17xx.h>

 unsigned int i,j;
 unsigned long LED = 0x00000010;

 int main(void)
 {
   SystemInit()

   SystemCoreClockUpdate();


   LPC_PINCON->PINSEL0 = 0
                                  ;Configure  Port0 pins  P0.4-P0.11 ;as GPIO

   LPC_GPIO0->FIODIR = 0x00000FF0;
                                                           ;Configure P0.4-P0.11 as output
```

```
00000000
00000001
00000011
00000111
00001111
00011111
00111111
01111111
11111111
11111110
11111100
11111000
11110000
11100000
11000000
10000000
00000000
```

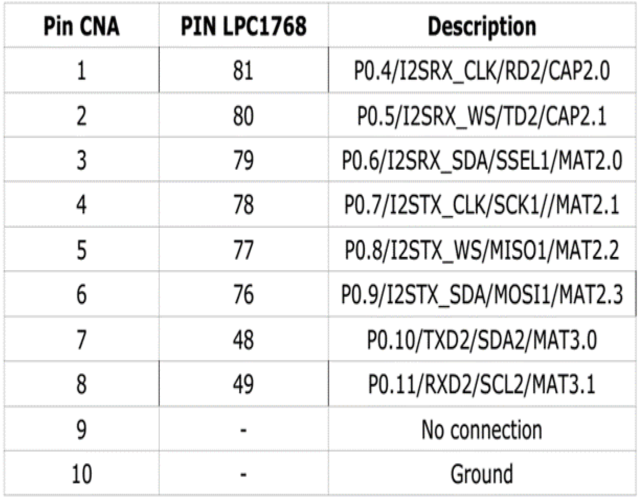
GPIO (General Purpose Input/Output)

```
while(1)
                {
                                LED = 0x00000010; Initial value on LED
                                for(i=1;i<9;i++)            //ON the LED's serially
                                {
                                  LPC_GPIO0->FIOSET = LED;

                                  for(j=0;j<10000;j++);
                                   LED <<= 1;
                                }


                                LED = 0x00000010;

                                    for(i=1;i<9;i++)          //OFF the LED's serially
                                                   {
                                    LPC_GPIO0->FIOCLR = LED
                                     for(j=0;j<10000;j++);
                                    LED <<= 1;
                                                   }

                    }
                     }
```

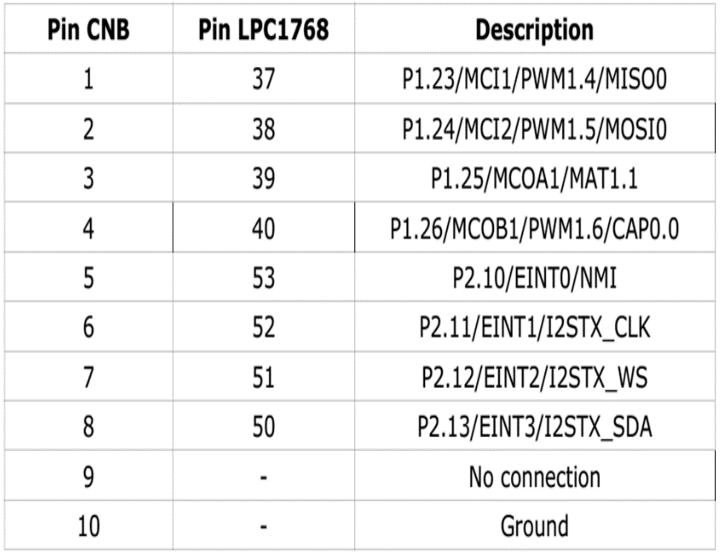
GPIO (General Purpose Input/Output)



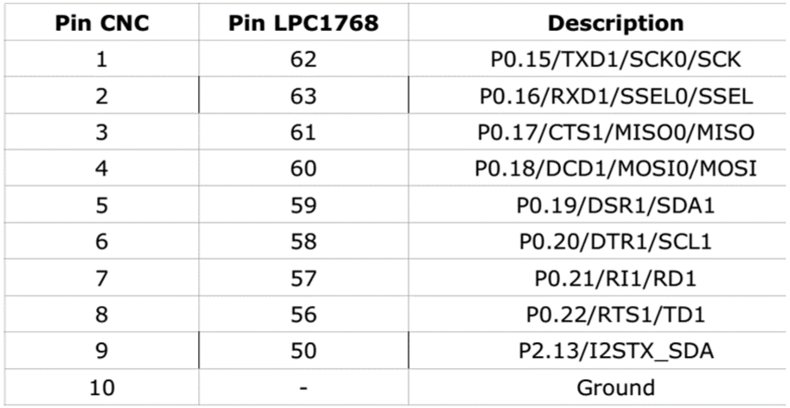
GPIO (General Purpose Input/Output)



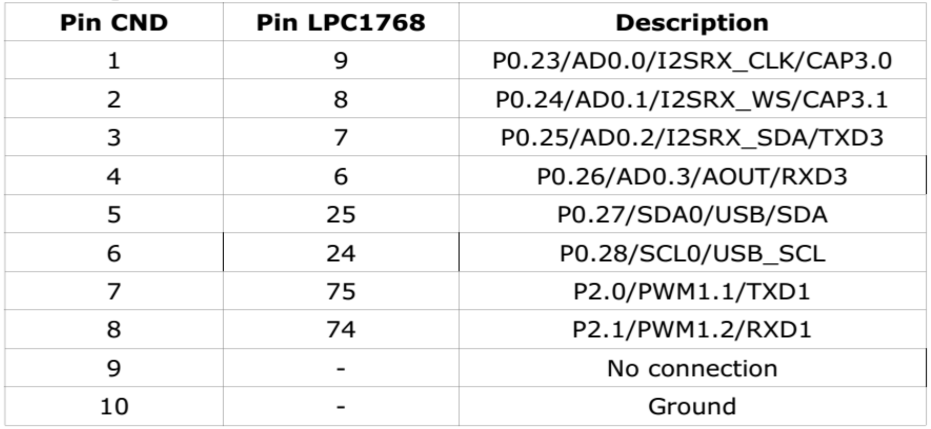
Pin CNA	PIN LPC1768	Description
1	81	P0.4/I2SRX_CLK/RD2/CAP2.0
2	80	P0.5/I2SRX_WS/TD2/CAP2.1
3	79	P0.6/I2SRX_SDA/SSEL1/MAT2.0
4	78	P0.7/I2STX_CLK/SCK1//MAT2.1
5	77	P0.8/I2STX_WS/MISO1/MAT2.2
6	76	P0.9/I2STX_SDA/MOSI1/MAT2.3
7	48	P0.10/TXD2/SDA2/MAT3.0
8	49	P0.11/RXD2/SCL2/MAT3.1
9	-	No connection
10	-	Ground



Pin CNB	Pin LPC1768	Description
1	37	P1.23/MCI1/PWM1.4/MISO0
2	38	P1.24/MCI2/PWM1.5/MOSI0
3	39	P1.25/MCOA1/MAT1.1
4	40	P1.26/MCOB1/PWM1.6/CAP0.0
5	53	P2.10/EINT0/NMI
6	52	P2.11/EINT1/I2STX_CLK
7	51	P2.12/EINT2/I2STX_WS
8	50	P2.13/EINT3/I2STX_SDA
9	-	No connection
10	-	Ground



Pin CNC	Pin LPC1768	Description
1	62	P0.15/TXD1/SCK0/SCK
2	63	P0.16/RXD1/SSEL0/SSEL
3	61	P0.17/CTS1/MISO0/MISO
4	60	P0.18/DCD1/MOSI0/MOSI
5	59	P0.19/DSR1/SDA1
6	58	P0.20/DTR1/SCL1
7	57	P0.21/RI1/RD1
8	56	P0.22/RTS1/TD1
9	50	P2.13/I2STX_SDA
10	-	Ground

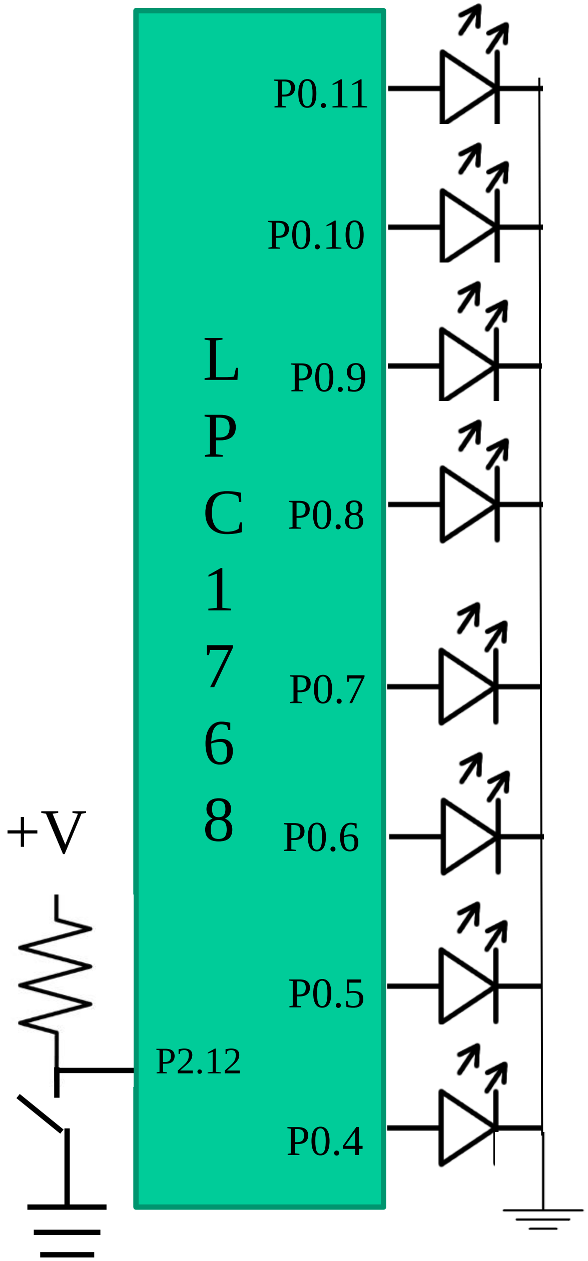


Pin CND	Pin LPC1768	Description
1	9	P0.23/AD0.0/I2SRX_CLK/CAP3.0
2	8	P0.24/AD0.1/I2SRX_WS/CAP3.1
3	7	P0.25/AD0.2/I2SRX_SDA/TXD3
4	6	P0.26/AD0.3/AOUT/RXD3
5	25	P0.27/SDA0/USB/SDA
6	24	P0.28/SCL0/USB_SCL
7	75	P2.0/PWM1.1/TXD1
8	74	P2.1/PWM1.2/RXD1
9	-	No connection
10	-	Ground

GPIO (General Purpose Input/Output)

Write an embedded C program to turn ON LEDs connected to P0.11 – P0.4 when key connected to P2.12 pressed, else turn OFF.

```
#include <LPC17xx.h>
 unsigned int j;
 unsigned long LED = 0x0000FF00;

 int main(void)
 {
                 SystemInit();
                 SystemCoreClockUpdate();

                 LPC_PINCON->PINSEL0 =  0x00000000; // P0.15-P0.0 GPIO
                 LPC_GPIO0->FIODIR = 0x00000FF0; // P0.11-P0.4 as output

                 while(1)
                 {
                     if ( !(LPC_GPIO2->FIOPIN & 1<<12))

                                 LPC_GPIO0->FIOSET = LED; // SET P0.11-P0.4
                     else

                                 LPC_GPIO0->FIOCLR = LED; // CLEAR P0.11-P0.4

                 }
 }
```


P0.11
P0.10
P0.9
P0.8
P0.7
P0.6
P0.5
P0.4
LPC1768
+V
P2.12