LED PROGRAMMING

```
#include <lpc17xx.h>
void delay_ms(unsigned int ms);
int main()
{
        SystemInit();                                   //Clock and PLL configuration
        LPC_GPIO2 → FIODIR |= (1<<26);                  //Configure the PORT2.26 pins a
        while(1)
        {       LPC_GPIO2 → FIOPIN |= (1<<26);          // Make P2.26 as high --LE
                delay_ms(100);
                LPC_GPIO2 → FIOPIN &= ~(1<<26);         // Make P2.26 as low  -- L
                delay_ms(100);
        }
}
```

LED PROGRAMMING

```
void delay_ms(unsigned int ms)
{
        unsigned int i, j;
        for(i=0; i<ms; i++)
                for(j=0; j<20000; j++);
}
```

LED PROGRAMMING

LED is output device

```
LPC_GPIO2 → FIODIR |= ((1<<0)|(1<<1)|(1<<2)|(1<<3)
```

Or

```
LPC_GPIO2 → FIODIR |= 0x0000000F;
```

LED must ON for few seconds

```
LPC_GPIO2 → FIOPIN |= (1<<0)|(1<<1)|(1<<2)|(1<<3);
```

Or

```
LPC_GPIO2 → FIOPIN |= 0x0000000F;
```

LED must OFF for few seconds

```
LPC_GPIO2 → FIOPIN &= ~((1<<0)|(1<<1)|(1<<2)|(1<<
```

Or

```
LPC_GPIO2 → FIOPIN &= 0xFFFFFFF0;
```

LED PROGRAMMING

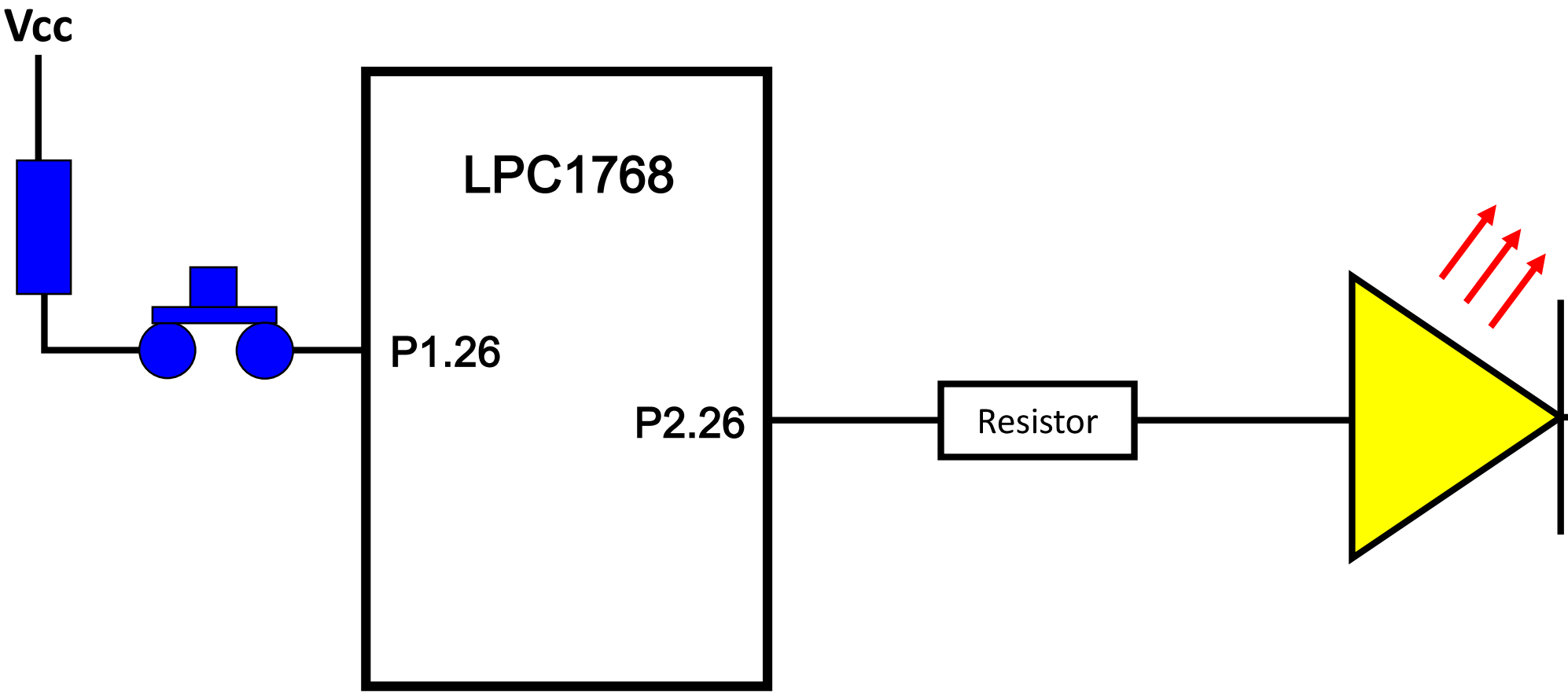
```
#include <lpc17xx.h>
void delay_ms(unsigned int ms);
int main()
{
        SystemInit();                                   //Clock and PLL configuration
        LPC_GPIO2 → FIODIR |= 0x0000000F; // Configure the PORT2.0 to P2.3 p
        while(1)
        {       LPC_GPIO2 → FIOPIN |= 0x0000000F;  // Make P2.0 to P2.3 as hig
                delay_ms(100);
                LPC_GPIO2 → FIOPIN &= 0xFFFFFFF0;  // Make P2.0 to P2.3 as lo
                delay_ms(100);
        }
}
```

LED PROGRAMMING

```
void delay_ms(unsigned int ms)
{
        unsigned int i, j;
        for(i=0; i<ms; i++)
                for(j=0; j<20000; j++);
}
```

SWITCH PROGRAMMING

Whenever Switch is pressed then LED must ON



SWITCH PROGRAMMING

Switch as input pin

LPC_GPIO1 → FIODIR &= ~(1<<26);

LED is output device

LPC_GPIO2 → FIODIR |= (1<<26);

Reading input switch

A=(LPC_GPIO1 → FIOPIN >> 26) & 0x01;

Verify whether switch is pressed or not

if(A)

SWITCH PROGRAMMING

if (Switch pressed)

LED must ON for few sec:

LPC_GPIO2 → FIOPIN |= (1<<26);

else

LED must OFF for few sec:

LPC_GPIO2 → FIOPIN &= ~(1<<26);

SWITCH PROGRAMMING

```
#include <lpc17xx.h>
void delay_ms(unsigned int ms);
int main()
{
        SystemInit();                                   //Clock and PLL configuration
        int A;
        LPC_GPIO1 → FIODIR &= ~ (1<<26);    //Configure the PORT1.26 pins a
        LPC_GPIO2 → FIODIR |= (1<<26);       //Configure the PORT2.26 pins a
        A = (LPC_GPIO1 → FIOPIN >> 26) & 0x01;
        while(1)
        {
```

SWITCH PROGRAMMING

```
            if (A)
            {
                    LPC_GPIO2 → FIOPIN |= (1<<26);     // Make P2.26 as hig
                    delay_ms(100);
            }
            else
            {
                    LPC_GPIO2 → FIOPIN &= ~(1<<26);  // Make P2.26 as lo
                    delay_ms(100);
            }
        }  // while loop termination
} // main termination
```

```
void delay_ms(unsigned int ms)
{
        unsigned int i, j;
        for(i=0; i<ms; i++)
                for(j=0; j<20000; j++);
}
```

LED PROGRAMMING : EXAMPLE-1

LED is output device

```
LPC_GPIO2 → FIODIR |= 0x0000000F;
```

State 1

```
LPC_GPIO2 → FIOPIN |= (1<<0)|(1<<2);

LPC_GPIO2 → FIOPIN &= ~((1<<1)|(1<<3));
```

State 2

```
LPC_GPIO2 → FIOPIN |= (1<<1)|(1<<3);

LPC_GPIO2 → FIOPIN &= ~((1<<0)|(1<<2));
```

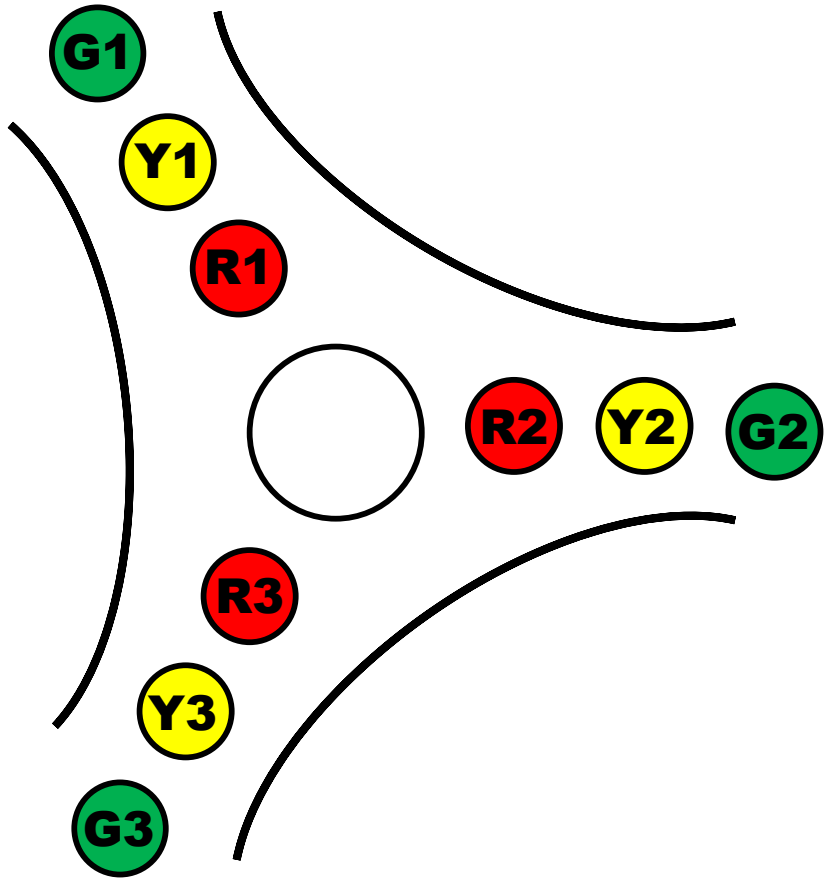
LED PROGRAMMING : EXAMPLE-1

```
#include <lpc17xx.h>
void delay_ms(unsigned int ms);
int main()
{
    SystemInit();                                   //Clock and PLL configuration
    LPC_GPIO2 → FIODIR |= 0x0000000F;               // Configure the PORT2.0 to P2.3 p
    while(1)
    {     LPC_GPIO2 → FIOPIN |= (1<<0)|(1<<2);      // Make P2.0 to P2.2 as high --
          LPC_GPIO2 → FIOPIN &= ~((1<<1)|(1<<3));   // Make P2.1 to P2.3 as low --L
          delay_ms(100);
          LPC_GPIO2 → FIOPIN |= (1<<1)|(1<<3);      // Make P2.1 to P2.3 as high --
          LPC_GPIO2 → FIOPIN &= ~((1<<0)|(1<<2));   // Make P2.0 to P2.2 as low --L
          delay_ms(100);
    }
}
```

LED PROGRAMMING : EXAMPLE-1

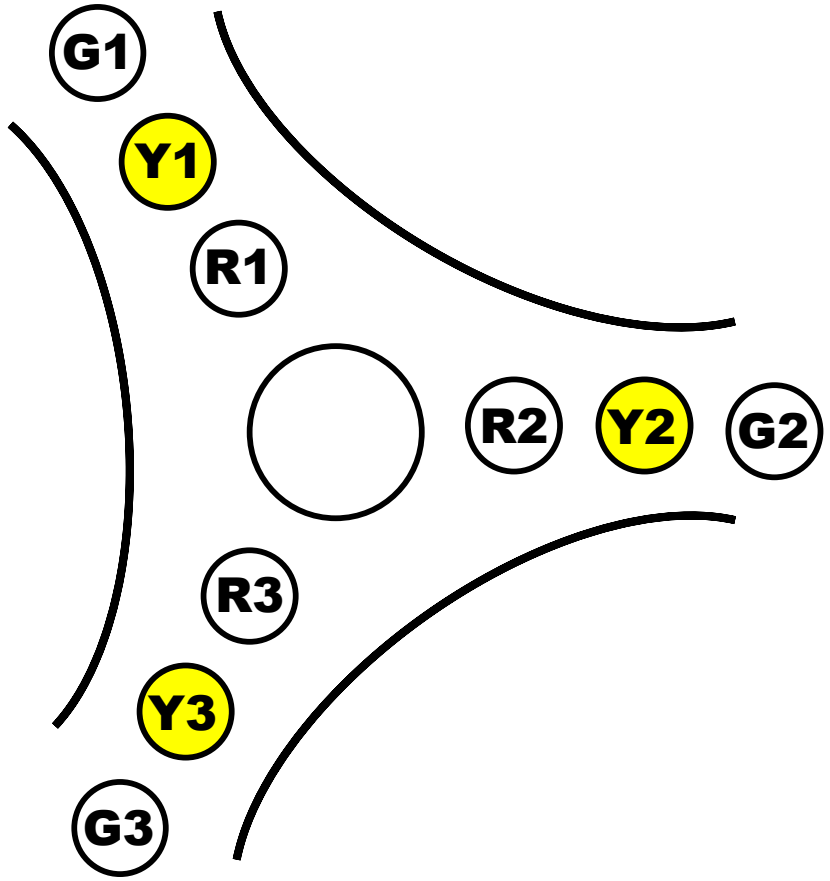
```
void delay_ms(unsigned int ms)
{
        unsigned int i, j;
        for(i=0; i<ms; i++)
                for(j=0; j<20000; j++);
}
```

TRAFFIC LIGHT CONTROL



R1	Y1	G1	R2	Y2	G2	R3	Y3	G3

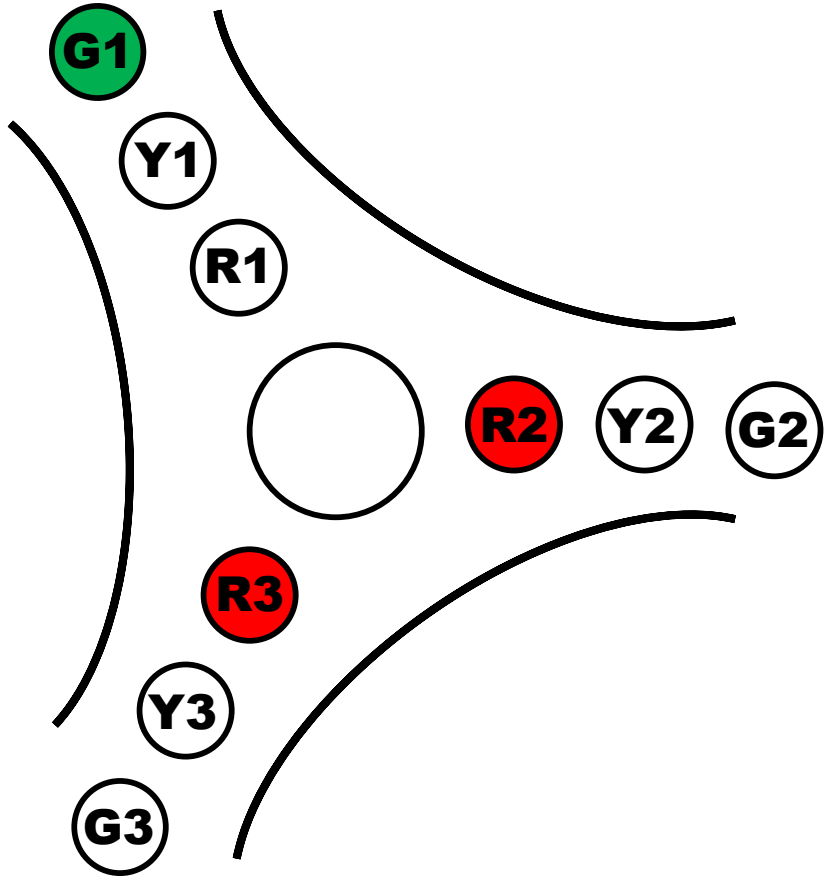
TRAFFIC LIGHT CONTROL



State 1								
R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0

0 → OFF **1 → ON**

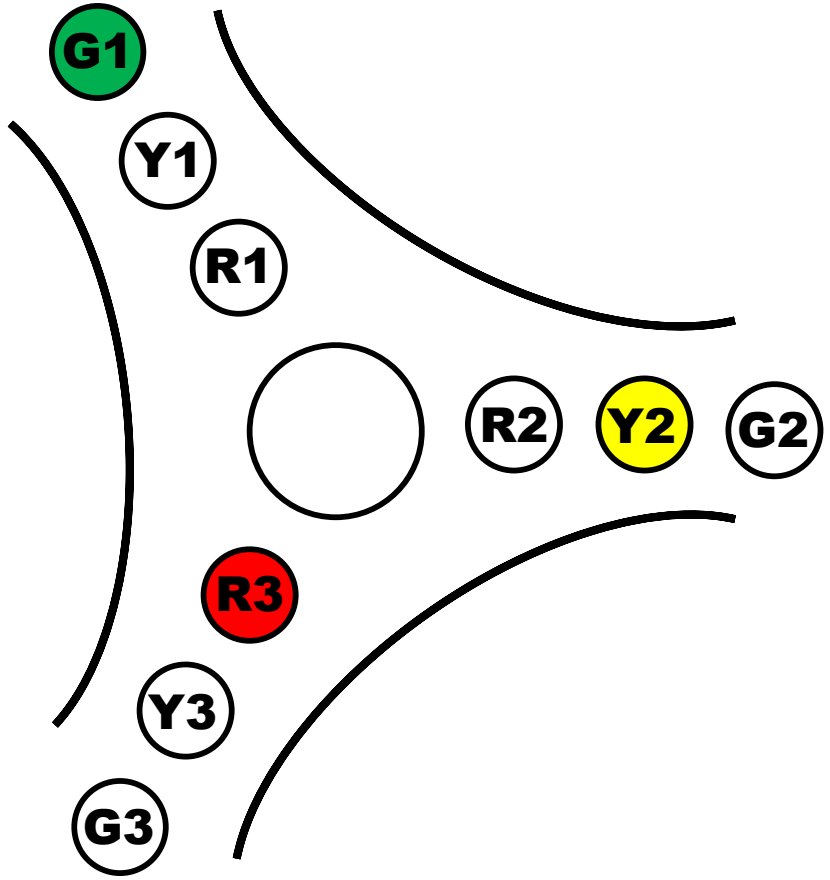
TRAFFIC LIGHT CONTROL



State 2								
R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL

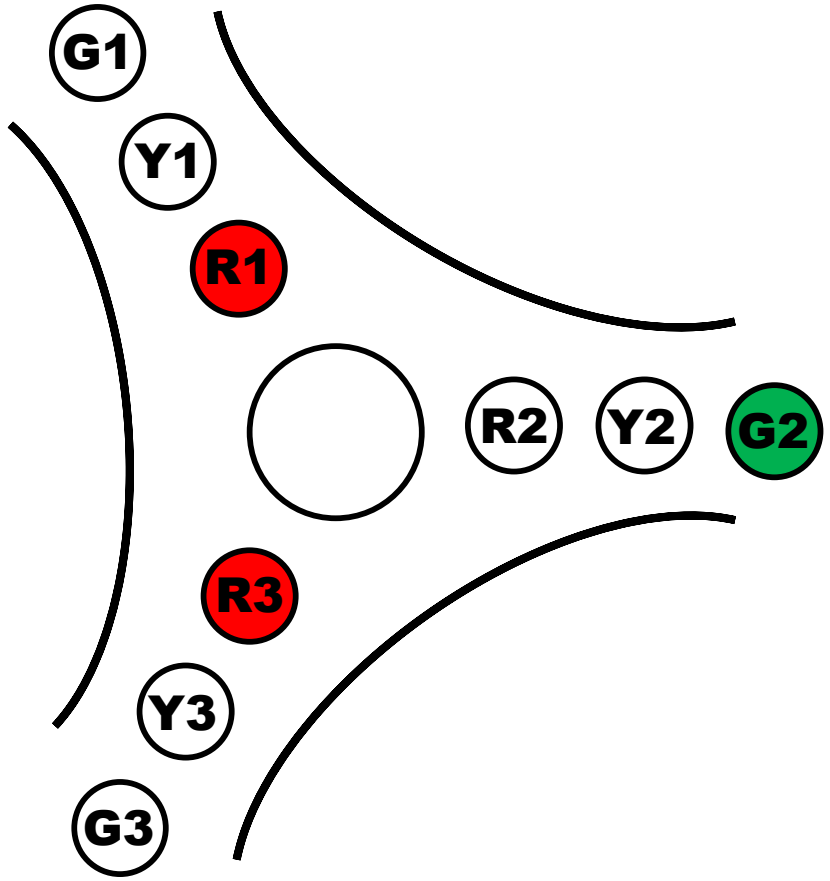


State 3

R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0

0 → OFF **1 → ON**

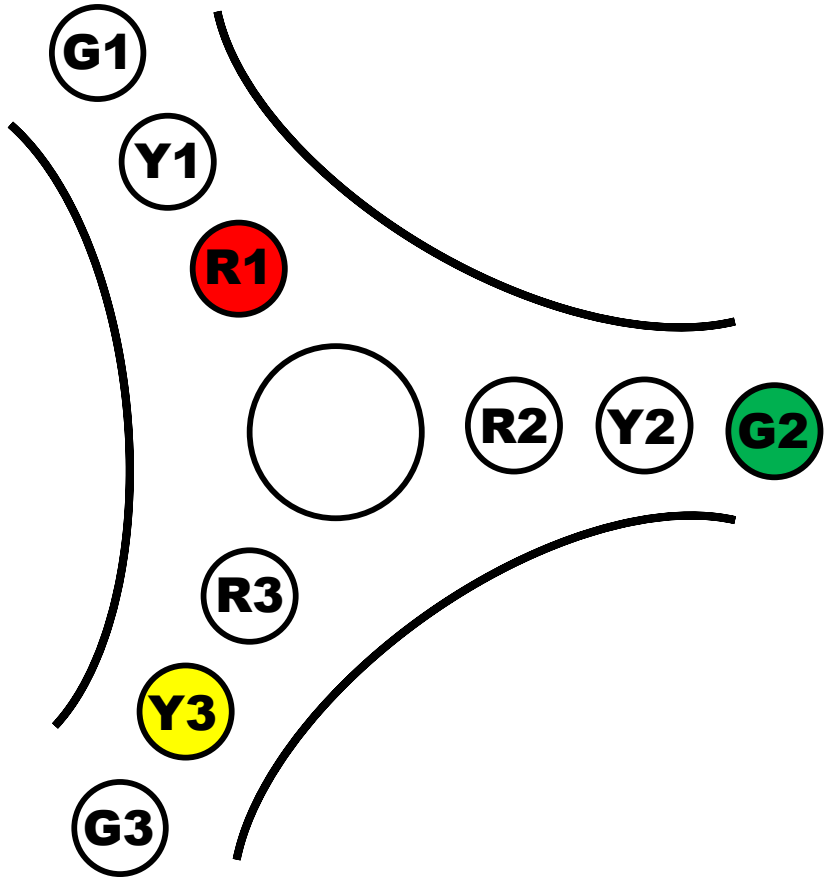
TRAFFIC LIGHT CONTROL



State 4								
R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL

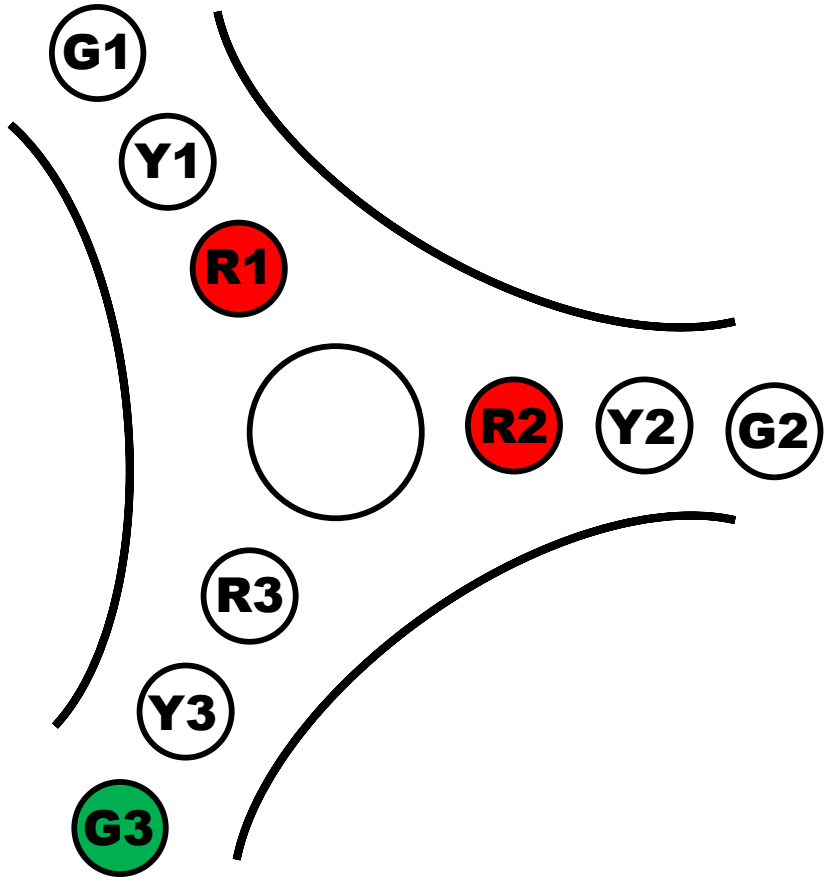


State 5

R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	1	0

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL

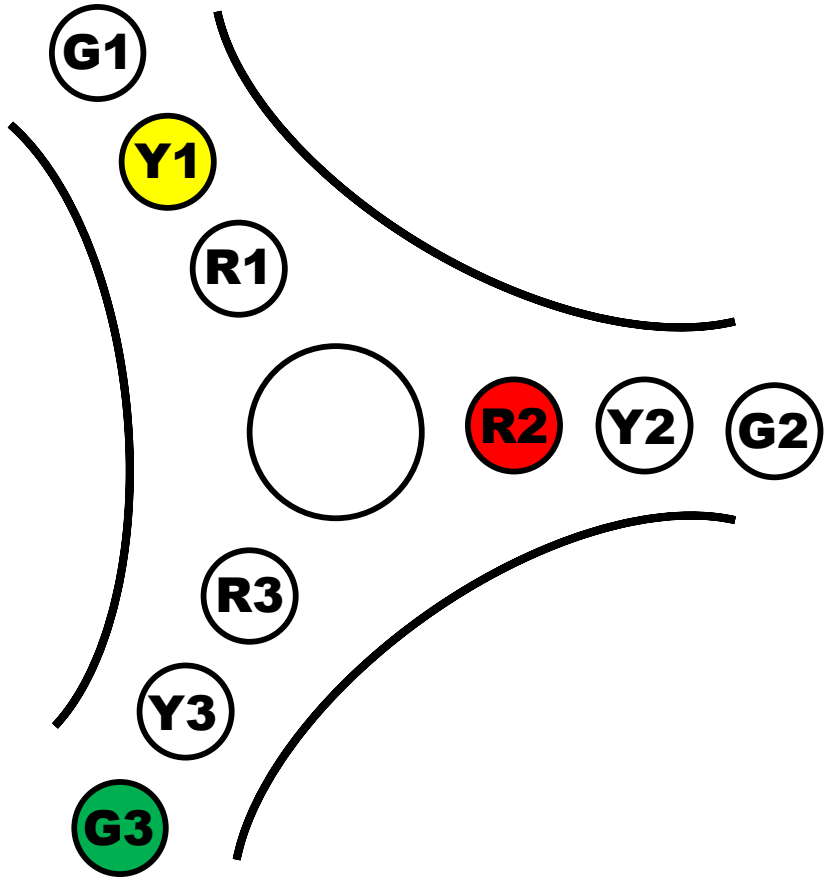


State 6

R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	1	0
1	0	0	1	0	0	0	0	1

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL

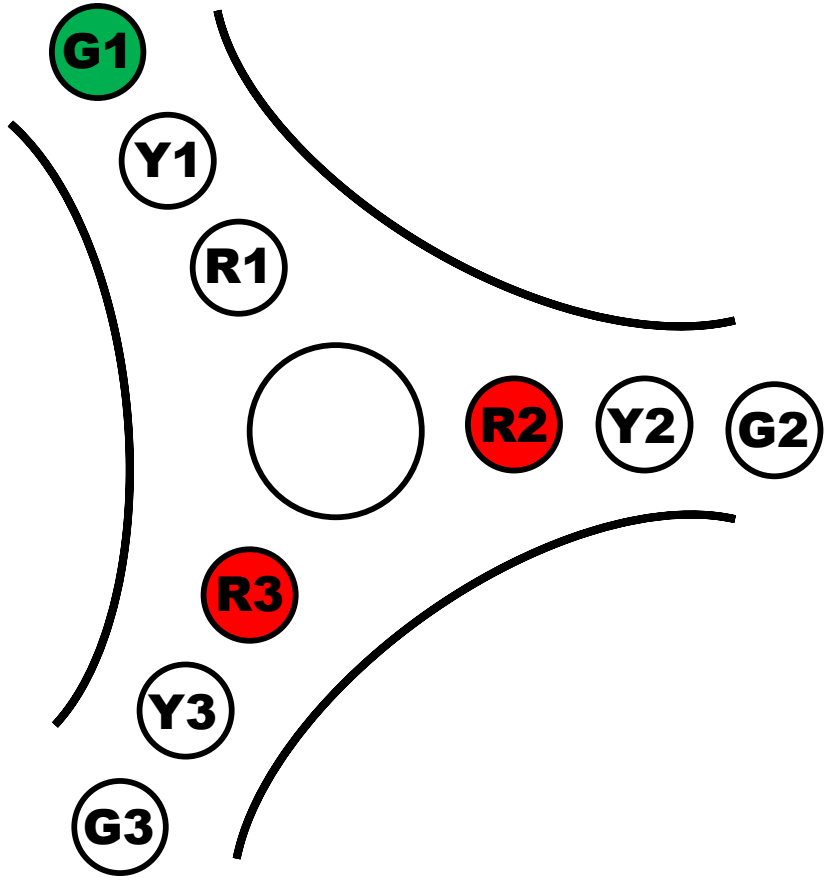


State 7

R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	1	0
1	0	0	1	0	0	0	0	1
0	1	0	1	0	0	0	0	1

0 → OFF **1 → ON**

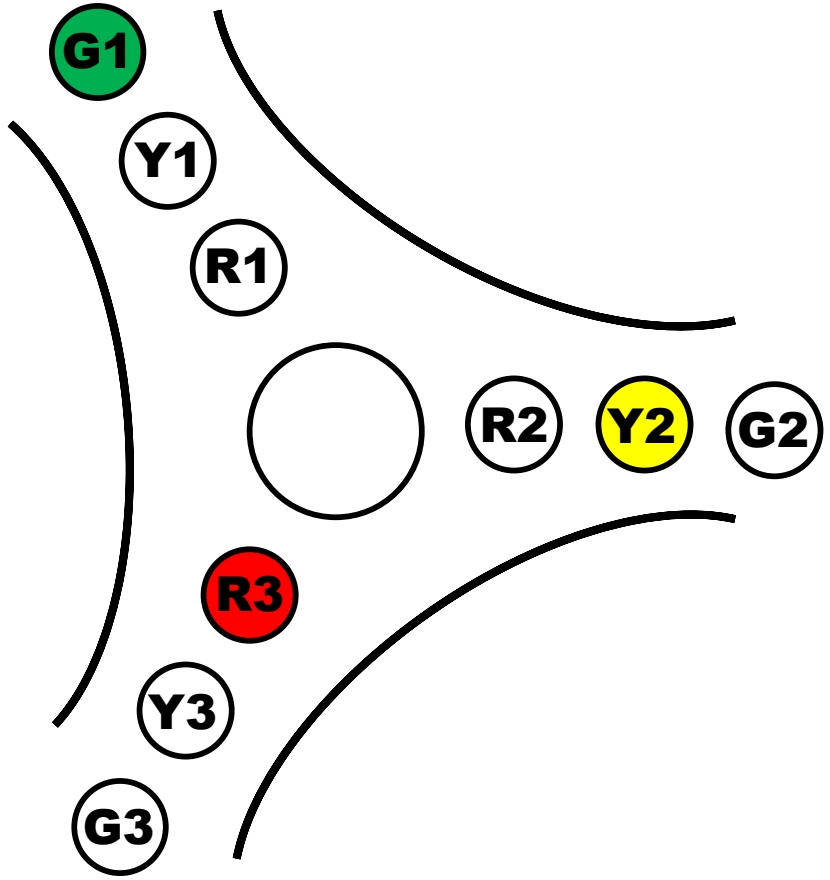
TRAFFIC LIGHT CONTROL



State 2								
R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0

0 → OFF **1 → ON**

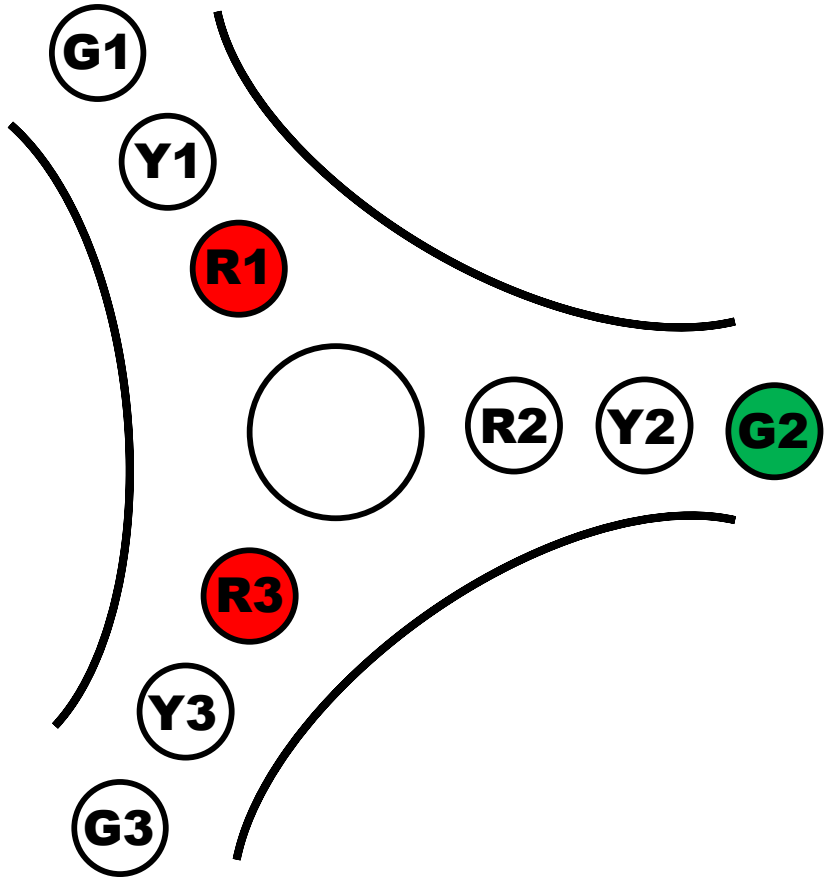
TRAFFIC LIGHT CONTROL



State 3								
R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0

0 → OFF **1 → ON**

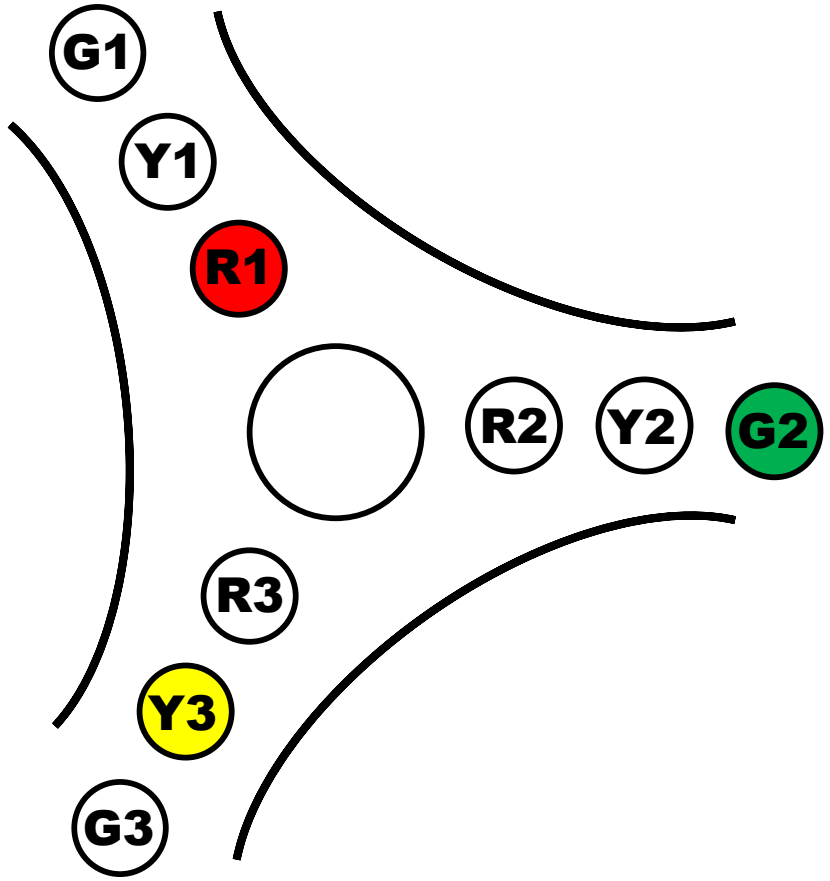
TRAFFIC LIGHT CONTROL



State 4								
R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL

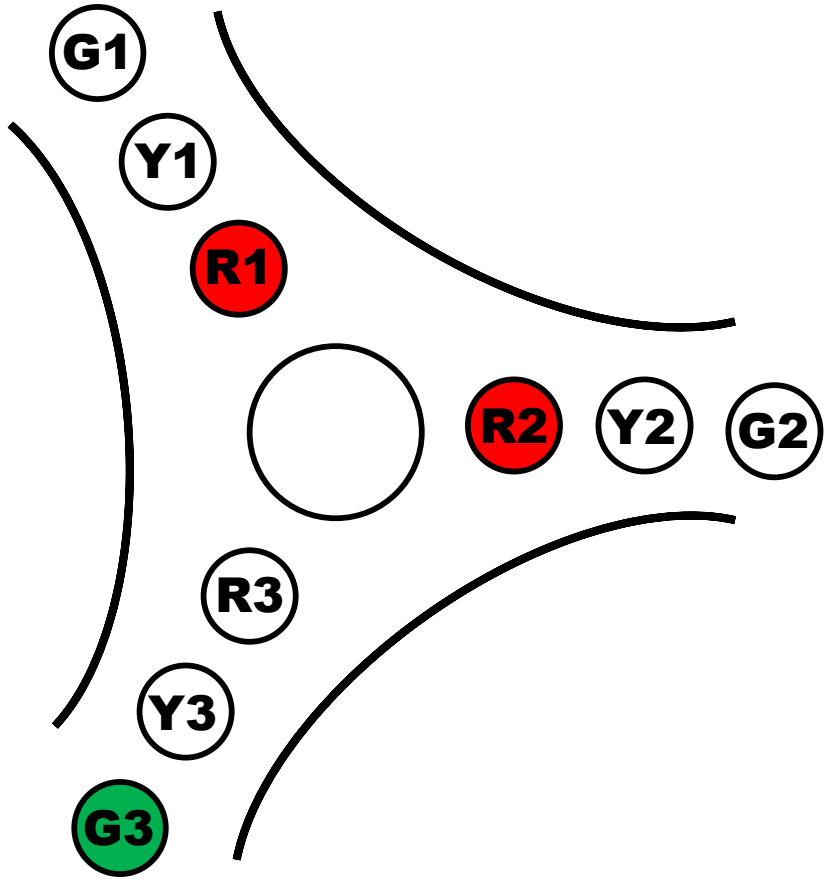


State 5

R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	1	0

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL

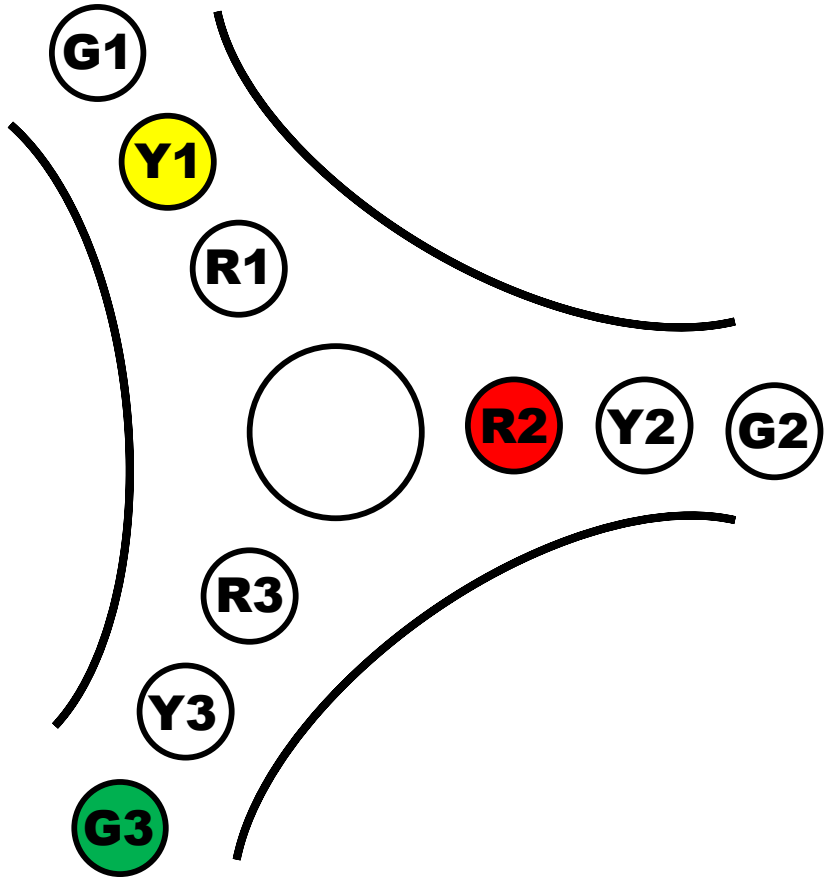


State 6

R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	1	0
1	0	0	1	0	0	0	0	1

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL



State 7

R1	Y1	G1	R2	Y2	G2	R3	Y3	G3
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	1	0
1	0	0	1	0	0	0	0	1
0	1	0	1	0	0	0	0	1

0 → OFF **1 → ON**

TRAFFIC LIGHT CONTROL

P2.8	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	Hex
R1	Y1	G1	R2	Y2	G2	R3	Y3	G3	
0	1	0	0	1	0	0	1	0	0x92
0	0	1	1	0	0	1	0	0	0x64
0	0	1	0	1	0	1	0	0	0x54
1	0	0	0	0	1	1	0	0	0x10C
1	0	0	0	0	1	0	1	0	0x10A
1	0	0	1	0	0	0	0	1	0x121
0	1	0	1	0	0	0	0	1	0xA1

TRAFFIC LIGHT CONTROL

	Hex	Embedded C Code
YYY	0x92	LPC_GPIO2 → FIOPIN |= 0x92;
GRR	0x64	LPC_GPIO2 → FIOPIN |= 0x64;
GYR	0x54	LPC_GPIO2 → FIOPIN |= 0x54;
RGR	0x10C	LPC_GPIO2 → FIOPIN |= 0x10C;
RGY	0x10A	LPC_GPIO2 → FIOPIN |= 0x10A;
RRG	0x121	LPC_GPIO2 → FIOPIN |= 0x121;
YRG	0xA1	LPC_GPIO2 → FIOPIN |= 0xA1;

In Loop

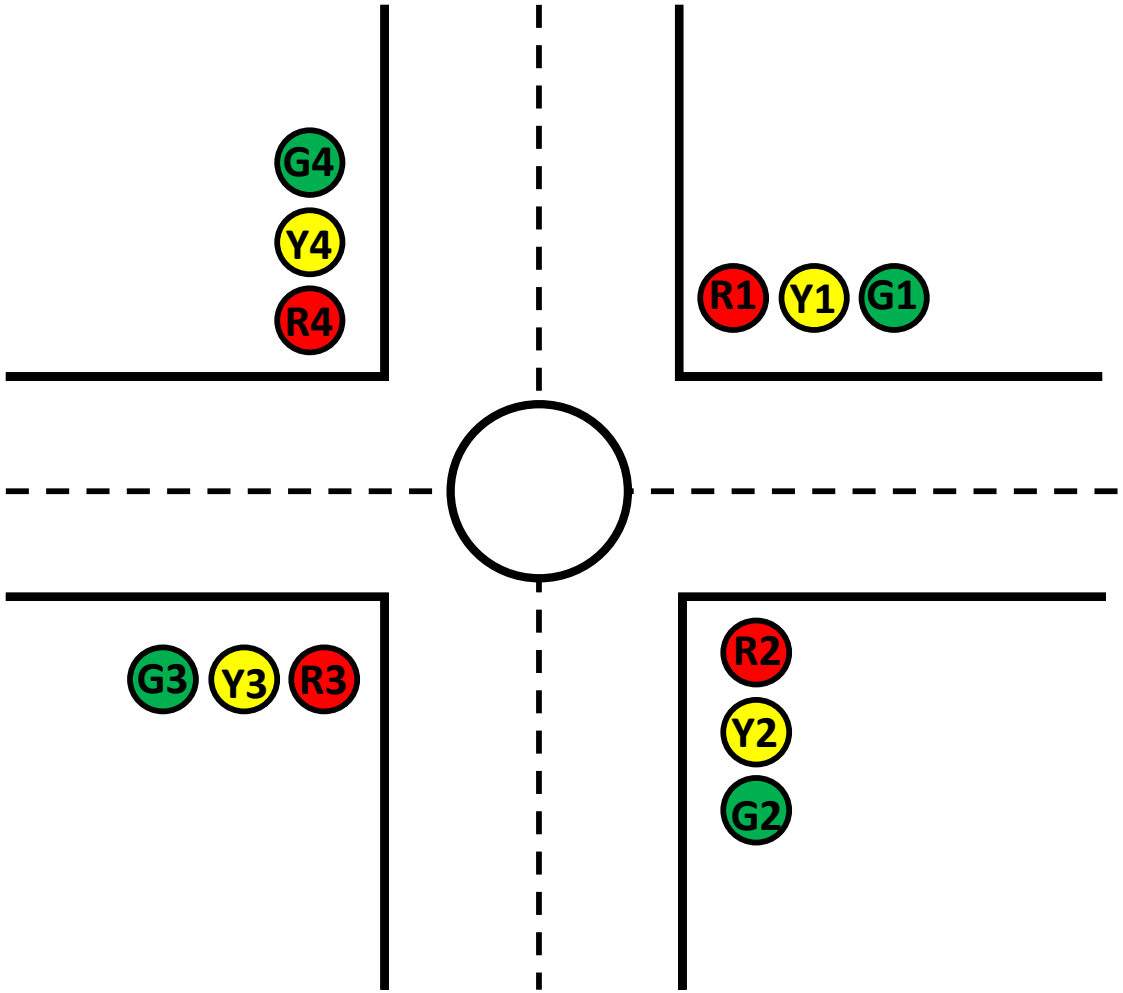
TRAFFIC LIGHT CONTROL

```
#include <lpc17xx.h>
void delay_ms(unsigned int ms);
int main()
{    SystemInit();                    //Clock and PLL configuration
     int i, a[6]= {0x64, 0x54, 0x10C, 0x10A, 0x121, 0xA1};
     LPC_GPIO2 → FIODIR |= 0x000001FF;  //Configure the P2.0 to P2.8 pins as OUTPUT
     LPC_GPIO2 → FIOPIN |= 0x92;          //YYY
     while(1)
     {   for(i=0; i<6; i++)
         { LPC_GPIO2 → FIOPIN |= a[i];
           delay(100);  }
     } }
```

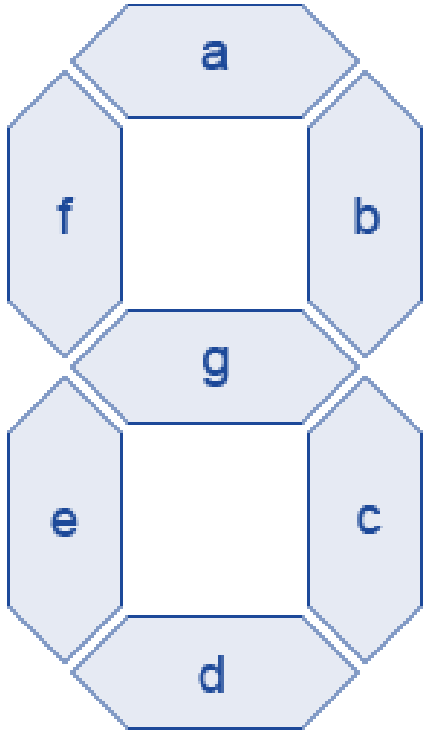
TRAFFIC LIGHT CONTROL

```
void delay_ms(unsigned int ms)
{
        unsigned int i, j;
        for(i=0; i<ms; i++)
                for(j=0; j<20000; j++);
}
```

TRAFFIC LIGHT CONTROL



7-SEGMENT DISPLAY INTERFACING



Segments (✓ = ON)							Display	Segments (✓ = ON)							Display
a	b	c	d	e	f	g		a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓		0	✓	✓	✓	✓	✓	✓	✓	8
	✓	✓					1	✓	✓	✓			✓	✓	9
✓	✓		✓	✓		✓	2	✓	✓	✓		✓	✓	✓	A
✓	✓	✓	✓			✓	3			✓	✓	✓	✓	✓	b
	✓	✓			✓	✓	4	✓			✓	✓	✓		C
✓		✓	✓		✓	✓	5		✓	✓	✓	✓		✓	d
✓		✓	✓	✓	✓	✓	6	✓			✓	✓	✓	✓	E
✓	✓	✓					7	✓				✓	✓	✓	F

7-SEGMENT DISPLAY INTERFACING

	P2.7	P2.6	P2.5	P2.4		P2.3	P2.2	P2.1	P2.0		HEX value
	dot	g	f	e		d	c	b	a		
DIGIT											
0	0	0	1	1		1	1	1	1		0x3F
1	0	0	0	0		0	1	1	0		0x06
2	0	1	0	1		1	0	1	1		0x5B
3	0	1	0	0		1	1	1	1		0x4F
4	0	1	1	0		0	1	1	0		0x66
5	0	1	1	0		1	1	0	1		0x6D
6	0	1	1	1		1	1	0	0		0x7C
7	0	0	0	0		0	1	1	1		0x07
8	0	1	1	1		1	1	1	1		0x7F
9	0	1	1	0		1	1	1	1		0x6F

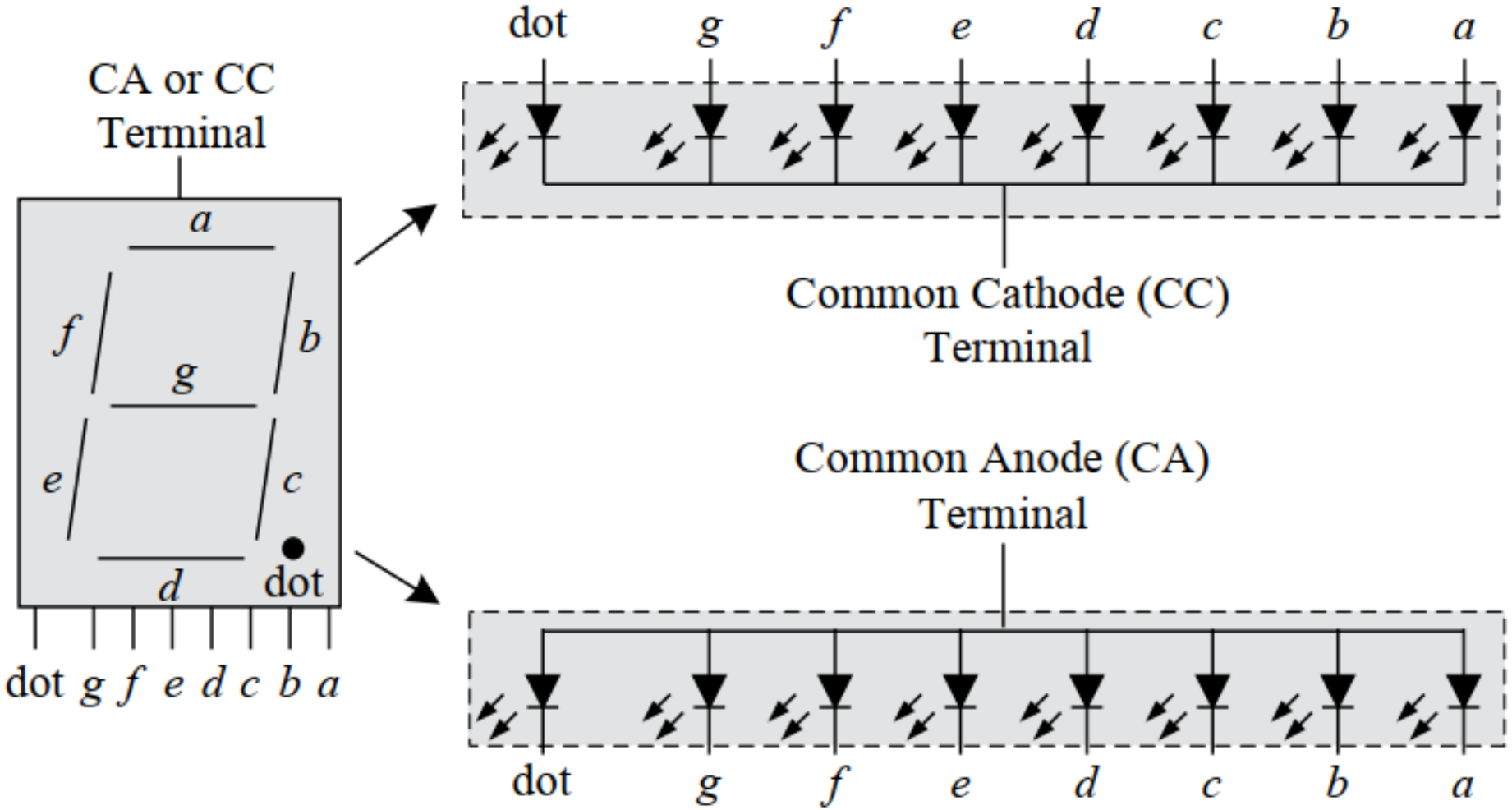
7-SEGMENT DISPLAY INTERFACING

```
#include <lpc17xx.h>
void delay_ms(unsigned int ms);
int main()
{    SystemInit();                    //Clock and PLL configuration
     int i, a[10]= {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7C, 0x07, 0x7F, 0x6F};
     LPC_GPIO2 → FIODIR |= 0x000000FF;  //Configure the P2.0 to P2.7 pins as OUTPUT
     while(1)
     {   for(i=0; i<10; i++)
         { LPC_GPIO2 → FIOPIN |= a[i];
           delay(100);  }
     }
}
```

7-SEGMENT DISPLAY INTERFACING

```
void delay_ms(unsigned int ms)
{
        unsigned int i, j;
        for(i=0; i<ms; i++)
                for(j=0; j<20000; j++);
}
```

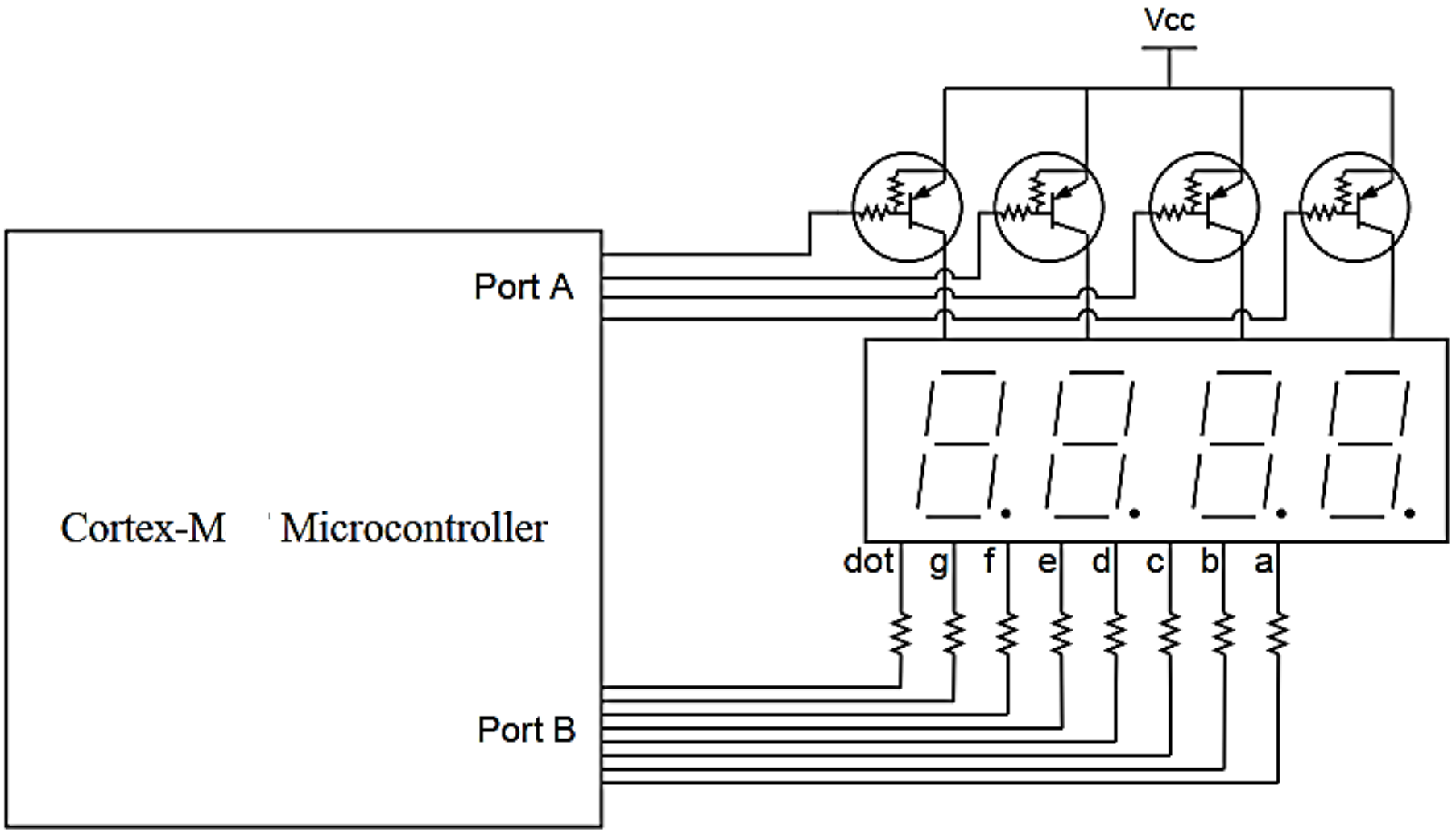
7-SEGMENT DISPLAY INTERFACING



7-SEGMENT DISPLAY INTERFACING

Character	Common Anode		Common Cathode	
	dot g f e d c b a	Hex	dot g f e d c b a	Hex
0	1 1 0 0 0 0 0 0	0xC0	0 0 1 1 1 1 1 1	0x3F
1	1 1 1 1 1 0 0 1	0xF9	0 0 0 0 0 1 1 0	0x06
2	1 0 1 0 0 1 0 0	0xA4	0 1 0 1 1 0 1 1	0x5B
3	1 0 1 1 0 0 0 0	0xB0	0 1 0 0 1 1 1 1	0x4F
4	1 0 0 1 1 0 0 1	0x99	0 1 1 0 1 1 1 1	0x66
5	1 0 0 1 0 0 1 0	0x92	0 1 1 0 1 1 0 1	0x6D
6	1 0 0 0 0 0 1 0	0x82	0 1 1 1 1 1 0 1	0x7D
7	1 1 1 1 1 0 0 0	0xF8	0 0 0 0 0 1 1 1	0x07
8	1 0 0 0 0 0 0 0	0x80	0 1 1 1 1 1 1 1	0x7F
9	1 0 0 1 0 0 0 0	0x90	0 1 1 0 1 1 1 1	0x6F
A	1 0 0 0 1 0 0 0	0x88	0 1 1 1 0 1 1 1	0x77
b	1 0 0 0 0 0 1 1	0x83	0 1 1 1 1 1 0 0	0x7C
C	1 1 0 0 0 1 1 0	0xC6	0 0 1 1 1 0 0 1	0x39
d	1 0 1 0 0 0 0 1	0xA1	0 1 0 1 1 1 1 0	0x5E
E	1 0 0 0 0 1 1 0	0x86	0 1 1 1 1 0 0 1	0x79
F	1 0 0 0 1 1 1 0	0x8E	0 1 1 1 0 0 0 1	0x71

7-SEGMENT DISPLAY INTERFACING



KEYPAD INTERFACING

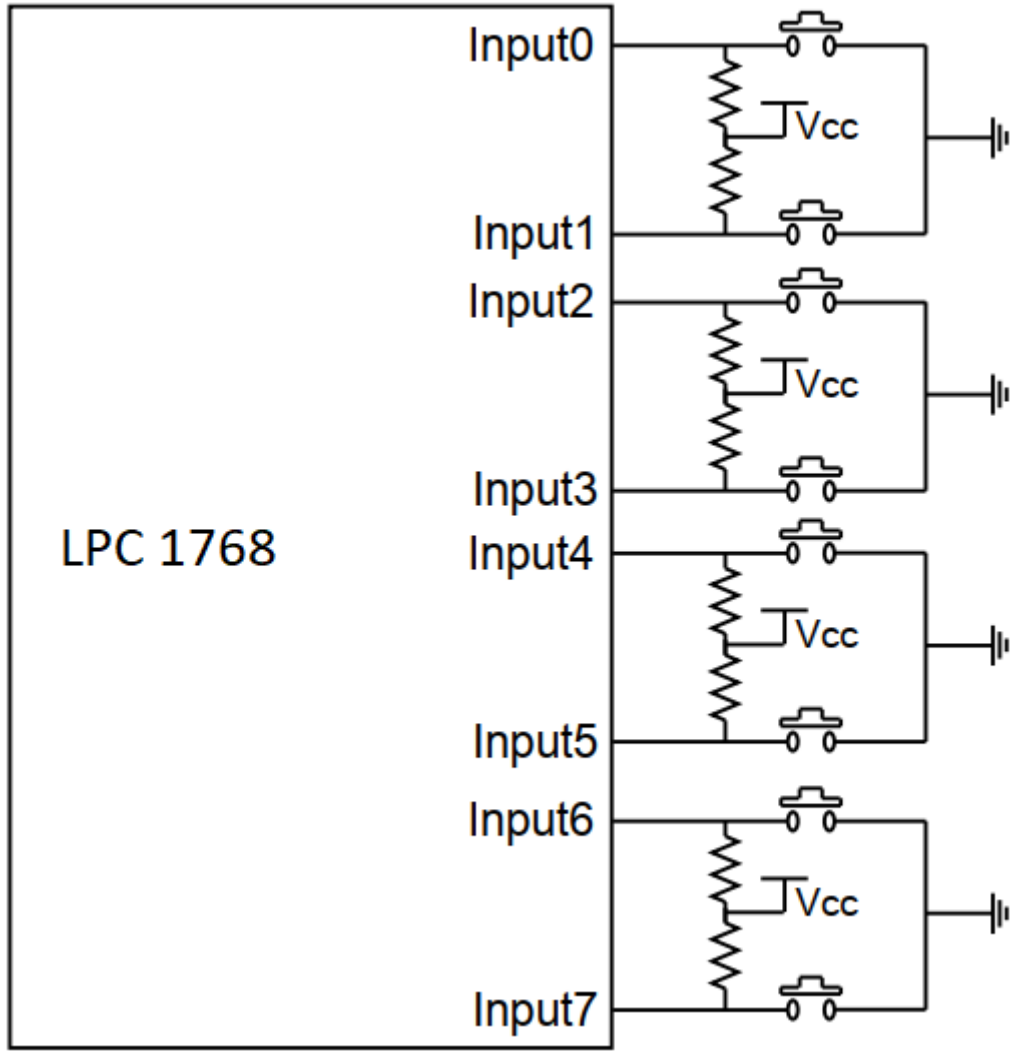
- ✓ Use of keypad is a common mechanism for user input to the system.
- ✓ From the software perspective a keypad interfacing can be either polling based or it can be interrupt driven.
- ✓ In polling based method the microcontroller spends most of its time by continuously scanning the keypad and checking for a possible key press.
- ✓ On the other hand, in interrupt driven method the microcontroller is free to execute any other tasks it is required to perform and scans the keypad only when an interrupt occurs due to a key press.

KEYPAD INTERFACING

- ✓ From the hardware perspective a keypad can be interfaced with a microcontroller using any of the three possible schemes:
  - Direct interface
  - Scanned interface
  - Multiplexed interface

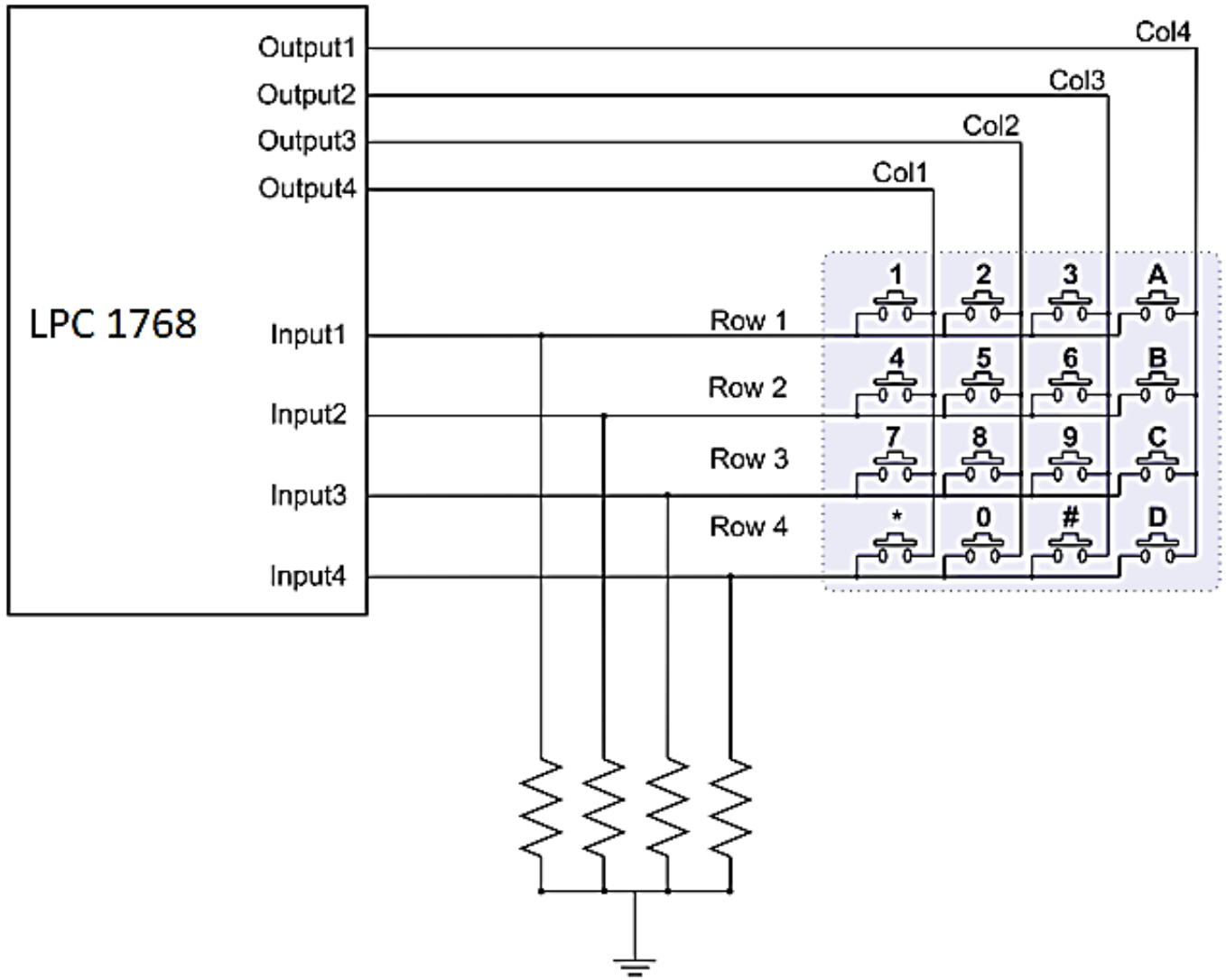
KEYPAD INTERFACING

Direct Interface



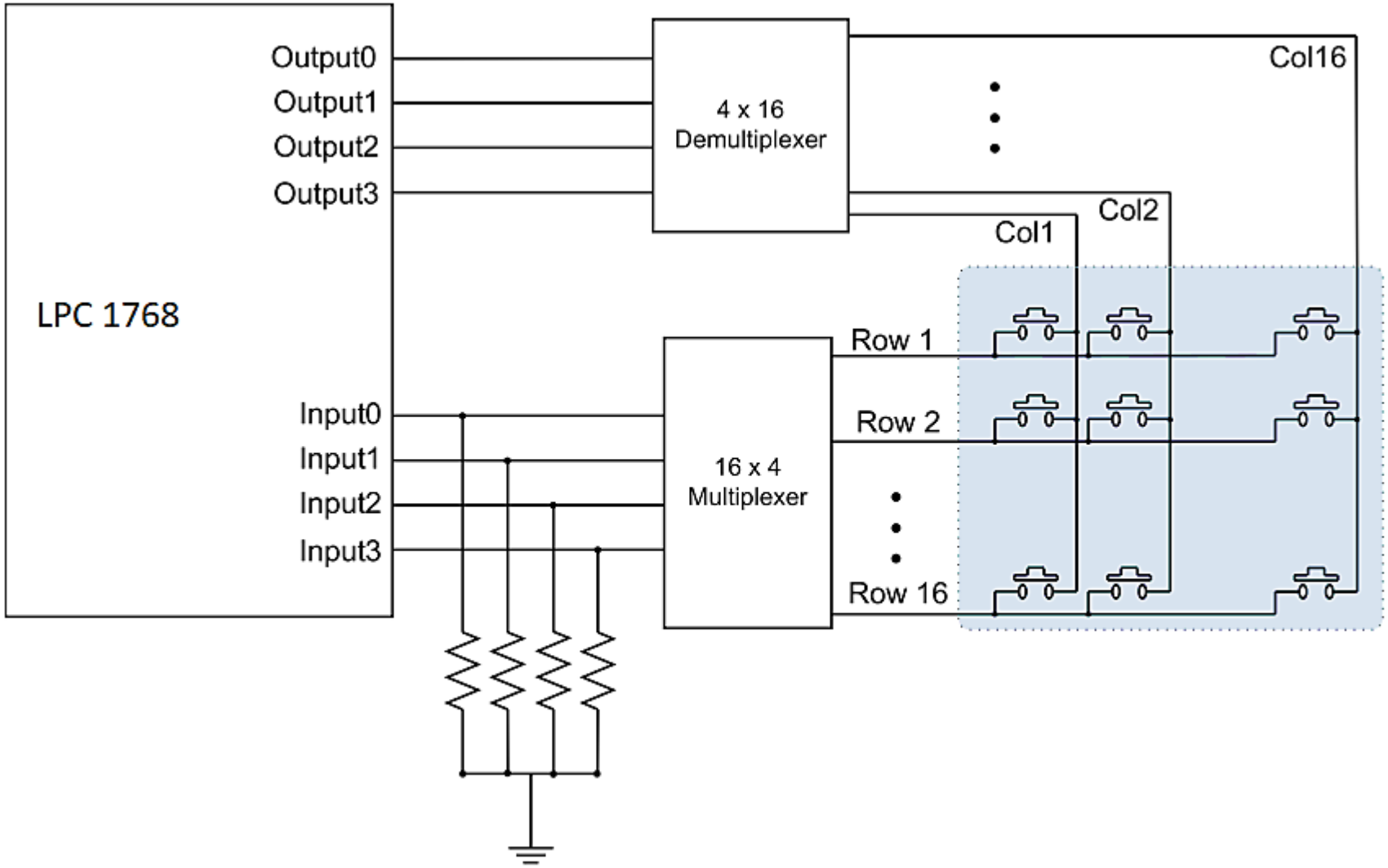
KEYPAD INTERFACING

Scanned Interface



KEYPAD INTERFACING

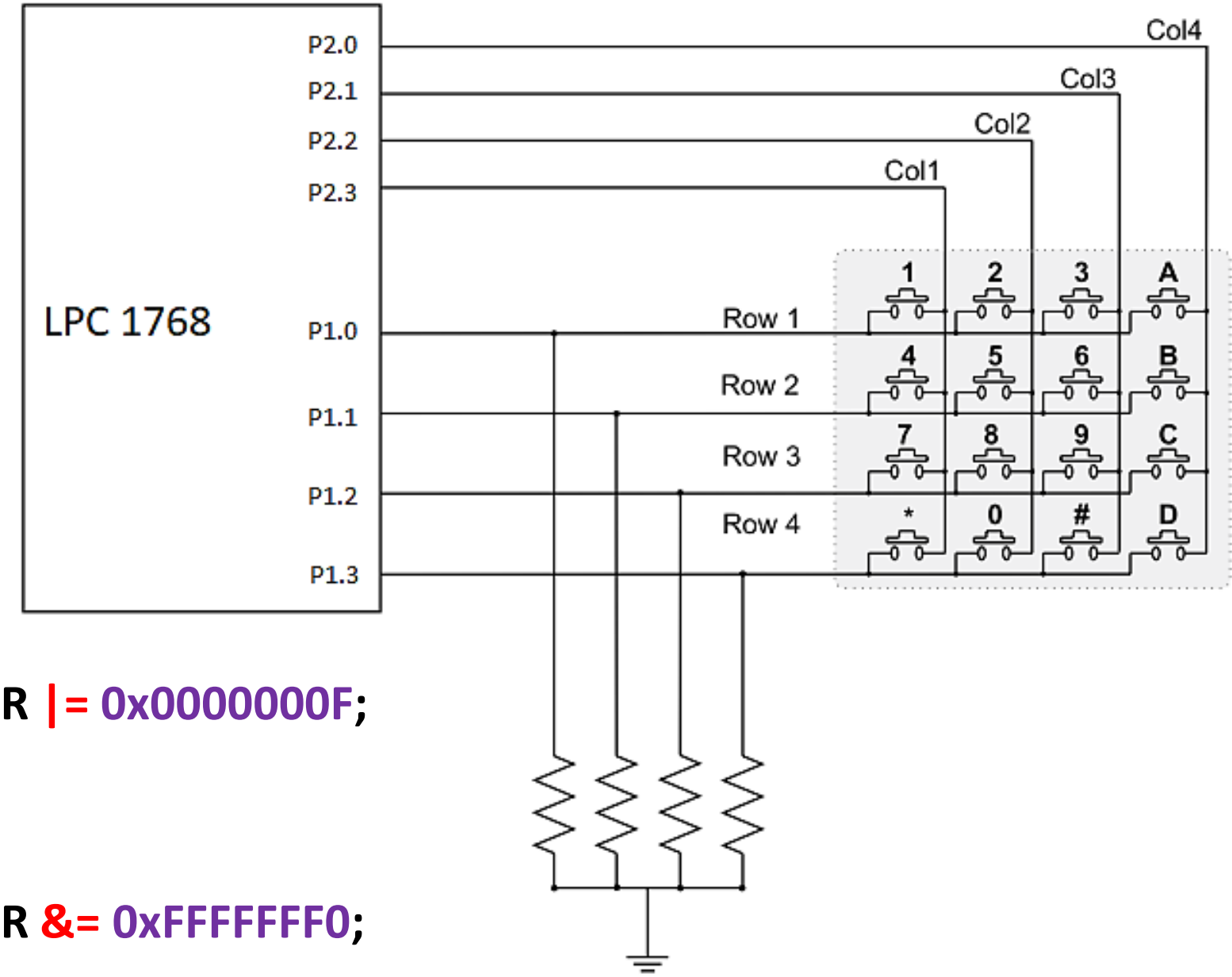
Multiplexed Interface



KEYPAD INTERFACING

Interface type	GPIO pins	Number of keypad keys
Direct interface	$m + n$	$m + n$
Scanned interface	$m + n$	$m \times n$
Multiplexed interface	$m + n$	$2^m \times 2^n$

KEYPAD INTERFACING



P2.0 – P2.3 as Output

LPC_GPIO2 → FIODIR |= 0x0000000F;

P1.0 – P1.3 as Input

LPC_GPIO1 → FIODIR &= 0xFFFFFFF0;

KEYPAD INTERFACING

```
#include <lpc17xx.h>
void delay_ms(unsigned int ms);
unsigned char key_press = 0;
# define        row1        (LPC_GPIO1 → FIOPIN >> 0) & 0x01)
# define        row2        (LPC_GPIO1 → FIOPIN >> 1) & 0x01)
# define        row3        (LPC_GPIO1 → FIOPIN >> 2) & 0x01)
# define        row4        (LPC_GPIO1 → FIOPIN >> 3) & 0x01)
unsigned char Scan_Keypad ( int * key_num )
{
        unsigned int i, key = 0;
        const unsigned char key_val [4][4] ={ // col1 , co2 , col3 , col4
                                        {'1',   '2',   '3',   'A'},  //row1
                                        {'4',   '5',   '6',   'B'},  //row2
                                        {'7',   '8',   '9',   'C'},  //row3
                                        {'*',   '0',   '#',   'D'},  //row4
                                };
```

KEYPAD INTERFACING

```
for (i = 1; i < 5; i ++)                          // loop for 4 rows
{
        // make columns high one by one
        LPC_PORT2 → FIOPIN |= (1<< i);
        if (row1 & 0x01 )                         // check row1
        {
                *key_num = (i * 4) + 1;
                while (row1 & 0x01);              // wait for release
                return key_val [key][i];          // return value of key pressed
        }
        if (row2 & 0x02)                          // check row2
        {
                key += 1;                         // next key pressed
                *key_num = (i * 4) + 2;
                while (row2 & 0x02);              // wait for release
                return key_val [ key ][i];        // return value of key pressed
        }
```

KEYPAD INTERFACING

```
                if (row3 & 0x04)                        // check row3
                {
                        key += 2;                       // next key pressed
                        *key_num = (i * 4) + 3;
                        while (row3 & 0x04);            // wait for release
                        return key_val [ key ][i];      // return value of key pressed
                }
                if (row4 & 0x08)                        // check row4
                {
                        key += 3;                       // next key pressed
                        *key_num = (i * 4) + 4;
                        while (row4 & 0x08);            // wait for release
                        return key_val [ key ][i];      // return value of key pressed
                }
                key = 0;
        }
        return 0;                                       // no key pressed
}
```

KEYPAD INTERFACING

```
int main()
{    SystemInit();                    //Clock and PLL configuration
     int val=0;
     LPC_GPIO2 → FIODIR |= 0x0000000F;     //Configure the P2.0 to P2.3 pins as OUTPUT
     LPC_GPIO1 → FIODIR &= 0xFFFFFFF0;  //Configure the P1.0 to P1.3 pins as INPUT
     while(1)
     {
          delay(100);
          key_press = Scan_Keypad (&val);
     }
}
```

KEYPAD INTERFACING

```
void delay_ms(unsigned int ms)
{
        unsigned int i, j;
        for(i=0; i<ms; i++)
                for(j=0; j<10000; j++);
}
```