ARM
LPC2148 Architecture

Dr. T.G.Panse

Assistant Professor

Department of Electronics Engineering,

Yeshwantrao Chavan College of Engineering,

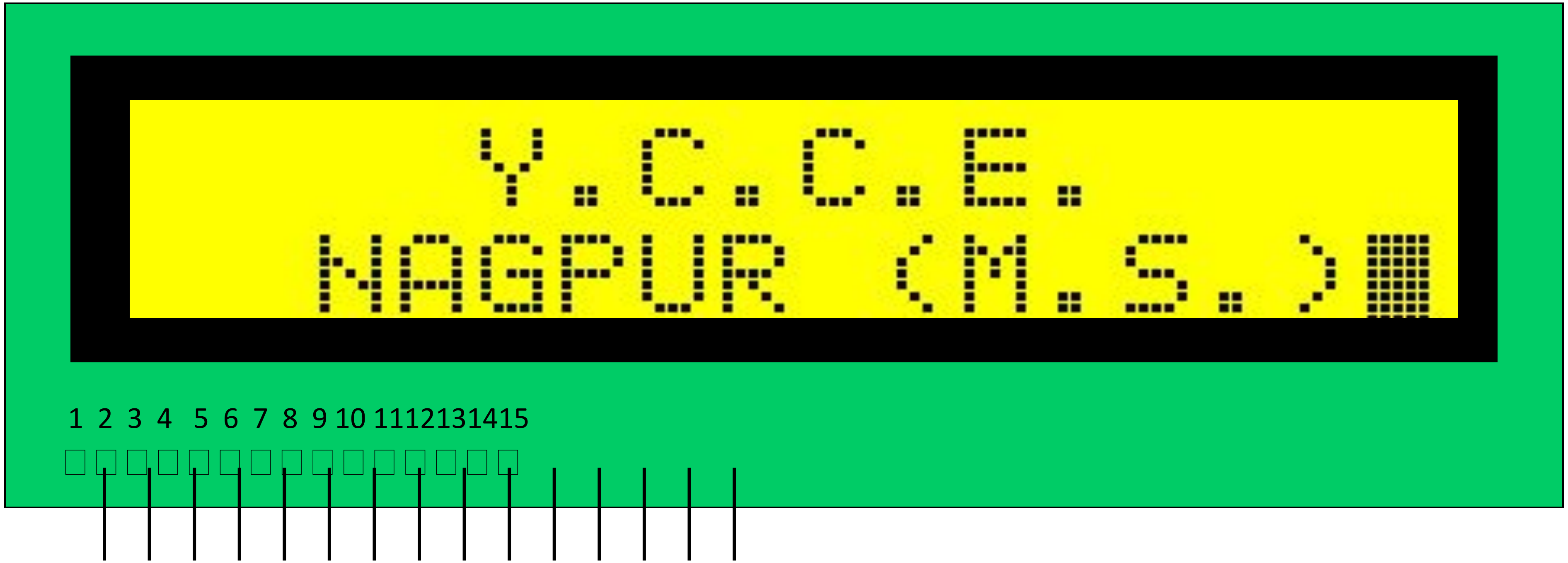
Nagpur – 441 110

Interfacing 16x2 LCD with LPC2148

LCD 16x2

- Liquid Crystal Display
- 2 Lines
- 16 Characters in each line

LCD 16x2



Pin No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Symbol	GND	VCC	CONT	RS	R/W	EN	D0	D1	D2	D3	D4	D5	D6	D7	BL

LCD 16x2 (Features)

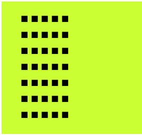
80H	
81H	
82H	
83H	
84H	Y
85H	.
86H	C
87H	.
88H	C
89H	.
8AH	E
8BH	.
8CH	
8DH	
8EH	
8FH	

C0H	
C1H	
C2H	N
C3H	A
C4H	G
C5H	P
86H	U
C7H	R
C8H	
C9H	(
CAH	M
CBH	.
CCH	S
CDH	.
CEH	)
CFH	

Y.C.C.E.
NAGPUR (M.S.)

LCD Display RAM With address

LCD 16x2 (Features)



- 5 x 7 dots plus cursor, 5 x 10 dots per character
- 4 or 8 bit interface with MPU is possible
- Display Data RAM (DR) : 80 x 8 bit, Max 80 characters
- Character Generator (CG) ROM : 160, 5 x 7 Character fonts
  32, 5 x 10 Character fonts
  Custom ROM codes available
- Character Generator RAM : Program write (64 x 8 bits)
  8, 5 x 7 Character fonts
  4, 5 x 10 Character fonts
- Both Display Data RAM & CG RAM can be read by MPU
- **Wide variety of operating instructions :**
  Display Clear, Cursor Home, Display ON / OFF,
  Display Cursor Blink, Cursor shift, display shift etc.

LCD 16x2 (Pin Description)

Signal	In/Out	Function
D7 – D0	I / O	8 bit data bus used for bi-directional data transfer between μC & LCD module
D7 – D4	I / O	These lines are used as data bus in 4 bit operation.
D7	O	Can be used as Busy Flag
RS	I	Register Select 0 = Instruction Register (IR) 1 = Data Register (DR)
R / W	I	Signal to select Read or Write 0 = Write, 1 = Read
EN	I	Enable signal to start read / write operation

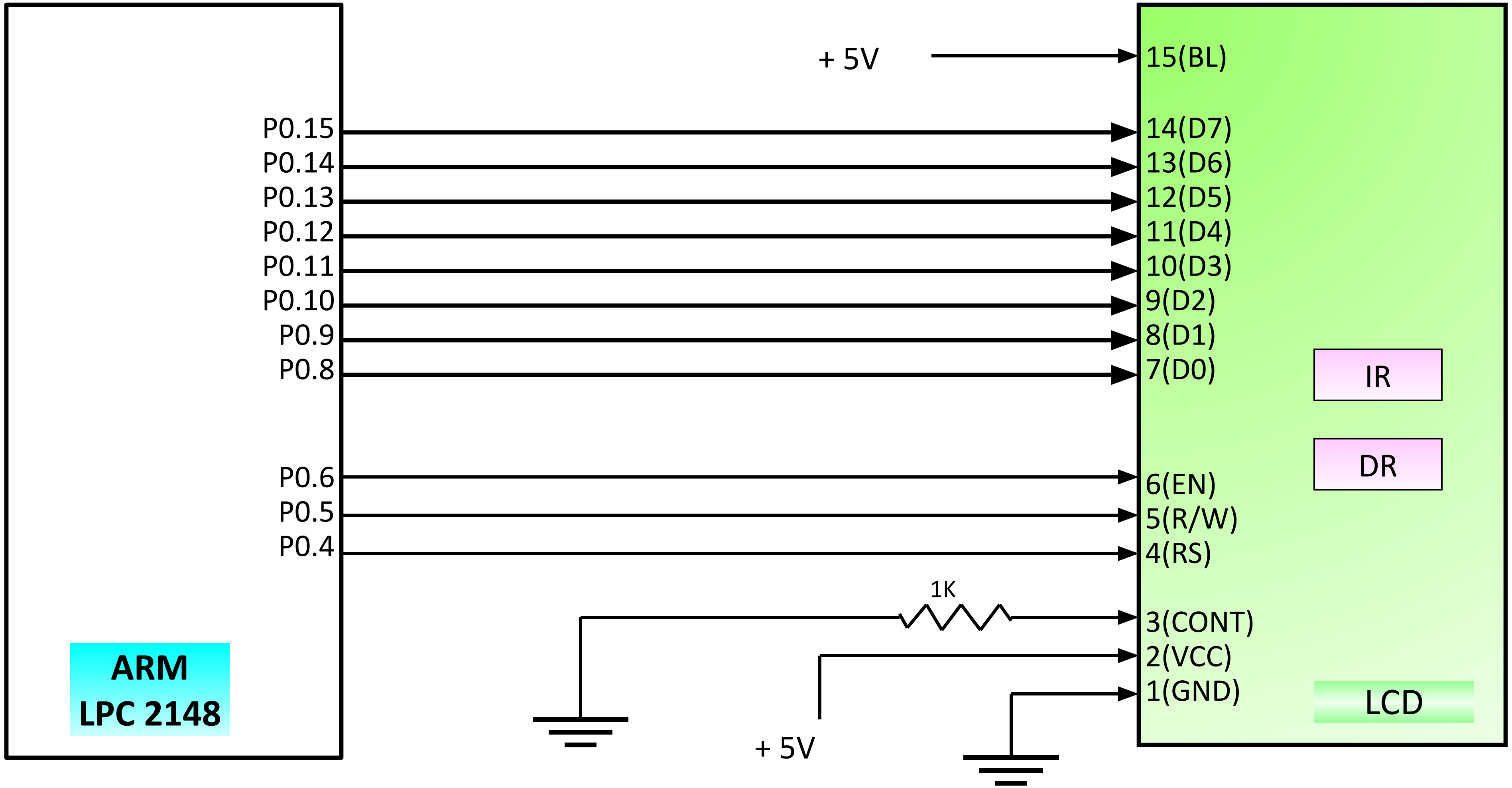
LCD 16x2 (Registers)

- Controller for LCD has two 8 bit registers,
- **instruction register (IR) and data register (DR)**
- **IR** is write only register used to store only instruction codes like display Clear, cursor shift or address of DD RAM etc.
- **DR** is read / write register used to store temporary data into DD RAM or CG RAM by an internal operation of the display controller.

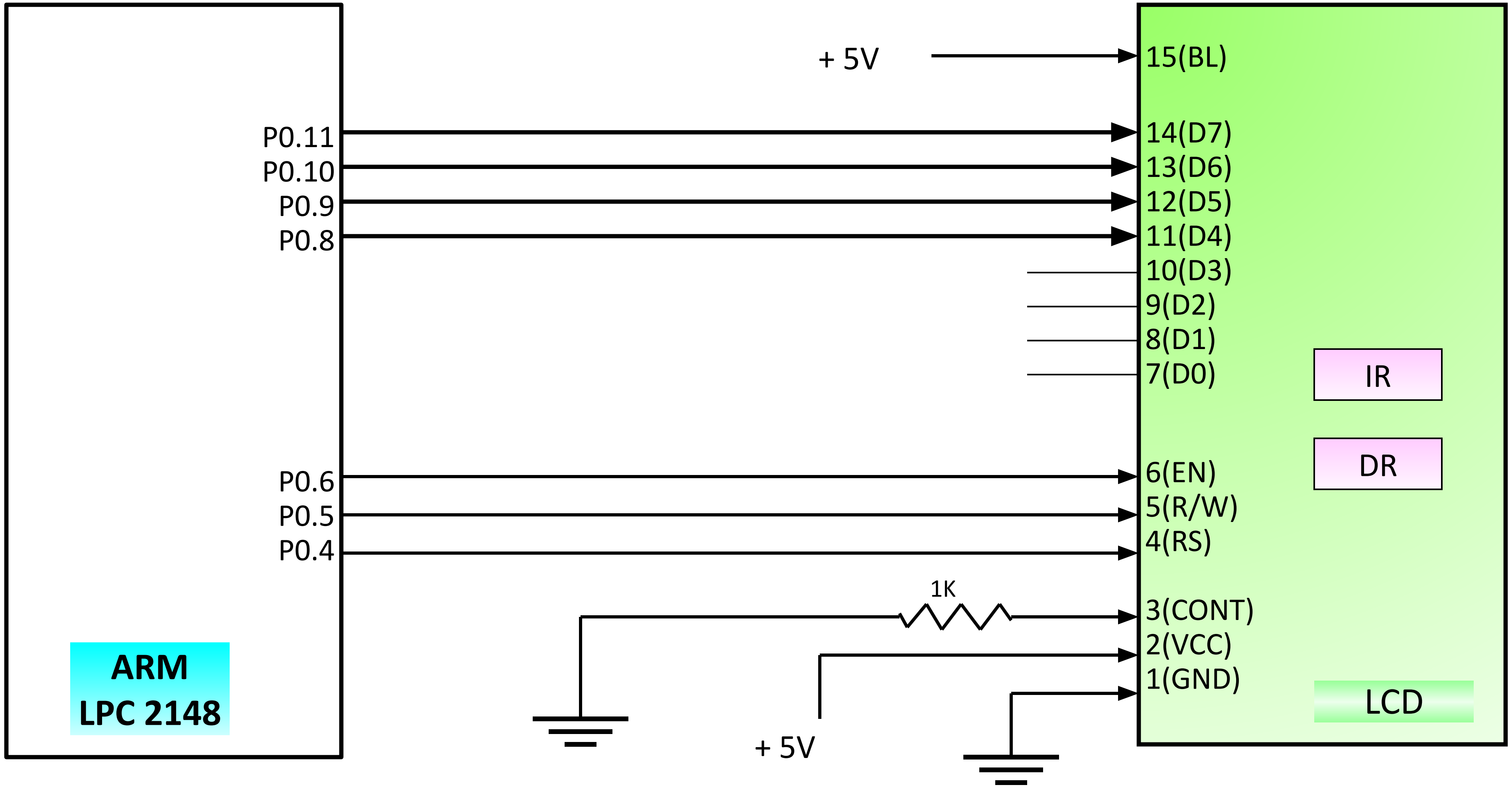
RS	R / W	Operation
0	**0**	**IR write, internal operation (Display clear, cursor position, function set etc.)**
0	1	Busy Flag (D7) & address counter (D6-D0) read
1	**0**	**DR write, internal operation (DD RAM or CG RAM)**
1	1	DR read, internal operation (DD RAM or CG RAM)

- If BF=1, then module is performing internal operation and next instruction will not be accepted

Interfacing LCD (8 bit data length)



Interfacing LCD (4 bit data length)



Command Write Operation (IR)

Command Write Function

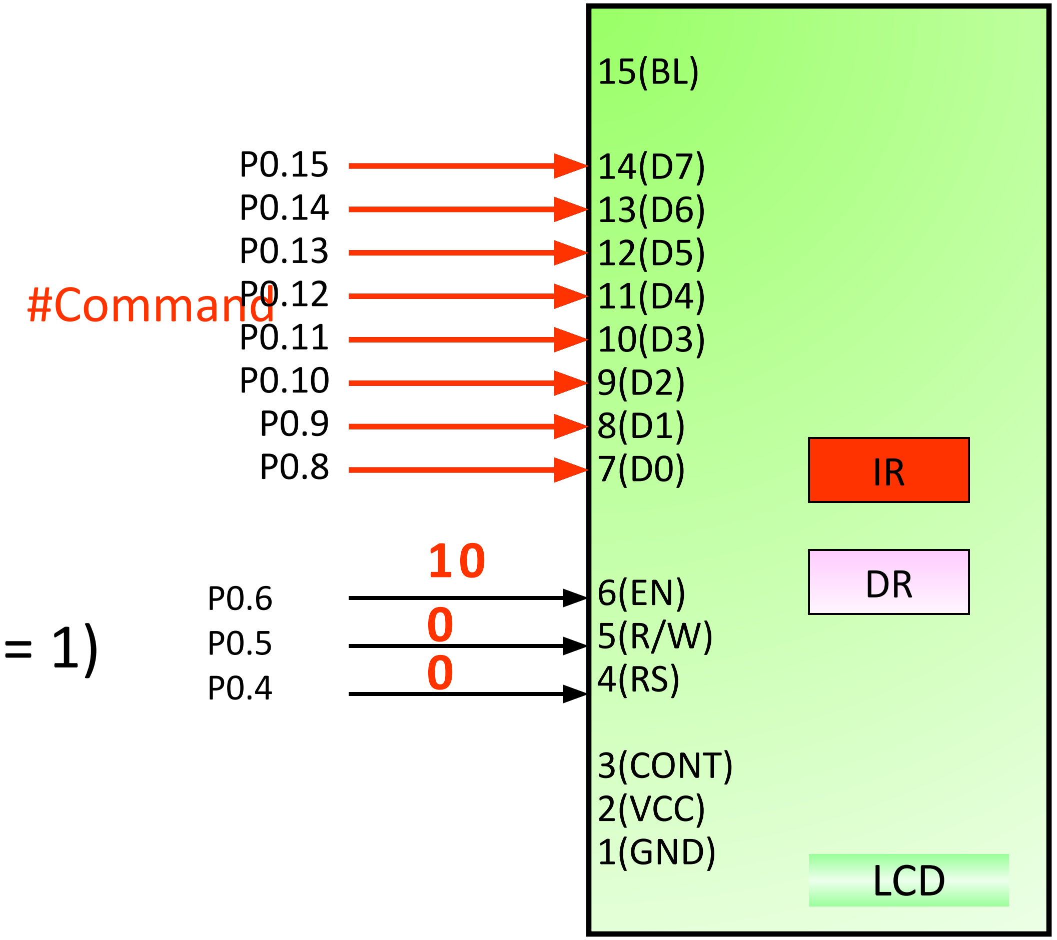
For command mode, RS = 0.

RW = 0 for write.

High to low EN pulse of minimum 450ns high pulse width.

To write command into IR

- Transfer instruction 8 bit command to Port (Via Data bus)
- Enable signal to start write operation (EN = 1)
- Select instruction Reg. (RS = 0)
- Select write (R/W = 0)
- EN = 0



#Command

P0.15 → 14(D7)
P0.14 → 13(D6)
P0.13 → 12(D5)
P0.12 → 11(D4)
P0.11 → 10(D3)
P0.10 → 9(D2)
P0.9 → 8(D1)
P0.8 → 7(D0)

P0.6 → 6(EN) 10
P0.5 → 5(R/W) 0
P0.4 → 4(RS) 0

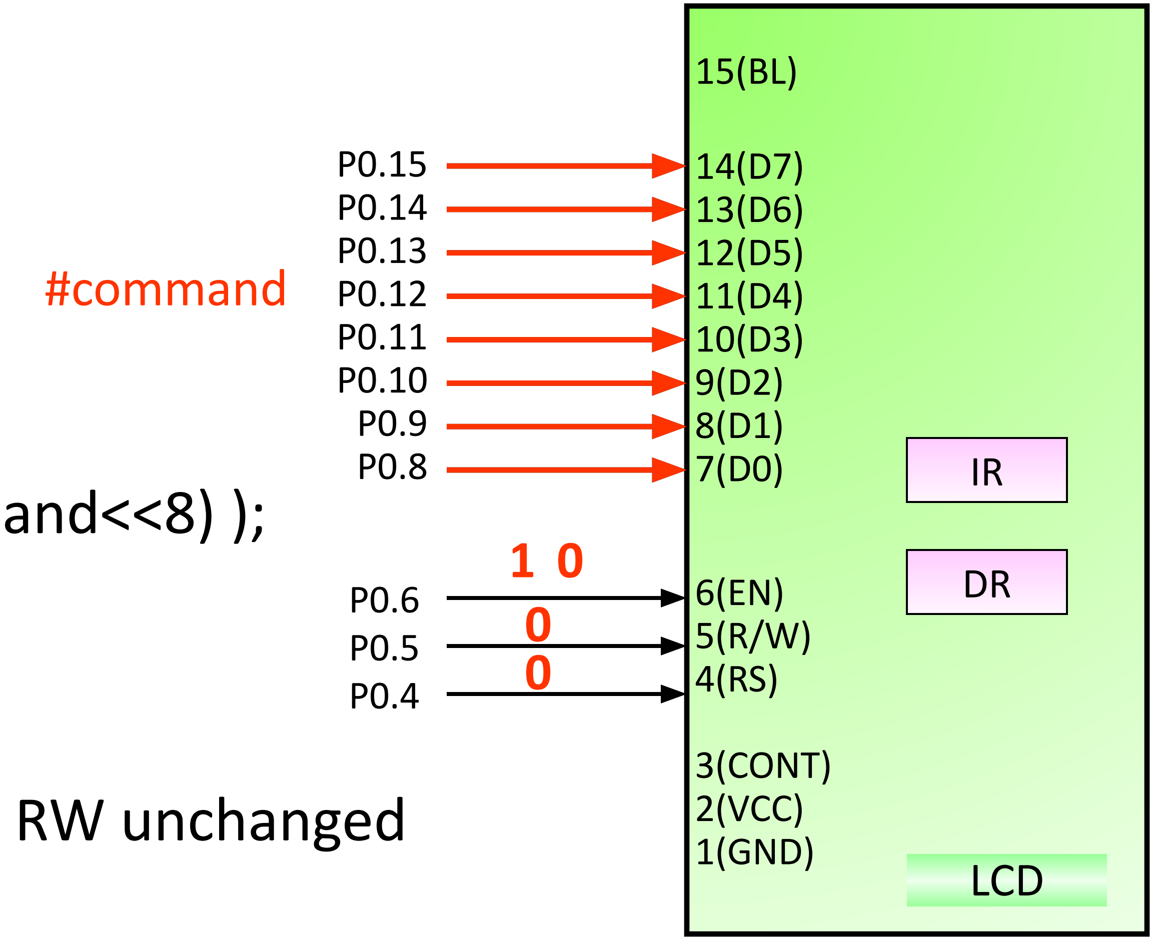
15(BL)
3(CONT)
2(VCC)
1(GND)

IR
DR
LCD

Command Write Function

Command Write Function

```
void LCD_CMD(char command)
{
IO0PIN = ( (IO0PIN & 0xFFFF00FF) | (command<<8) );
IO0SET = 0x00000040;      // EN = 1
IO0CLR = 0x00000030;      // RS = 0, RW = 0
delay_ms(2);
IO0CLR = 0x00000040;      // EN = 0, RS and RW unchanged
delay_ms(5);
}
```



Data Write Operation (DR)

Data Write Function (Single Character)

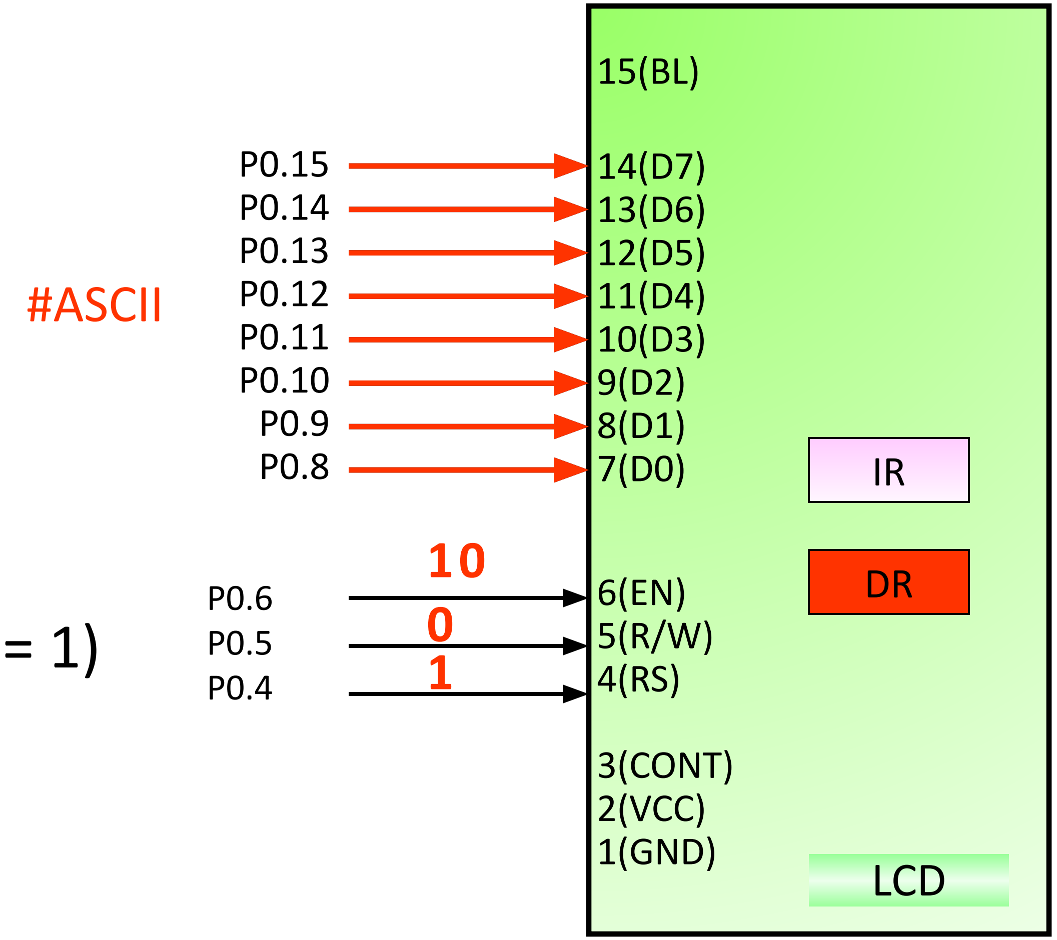
For data mode, RS = 1.

RW = 0 for write.

High to low EN pulse of minimum 450ns high pulse width.

To write ASCII data into DR

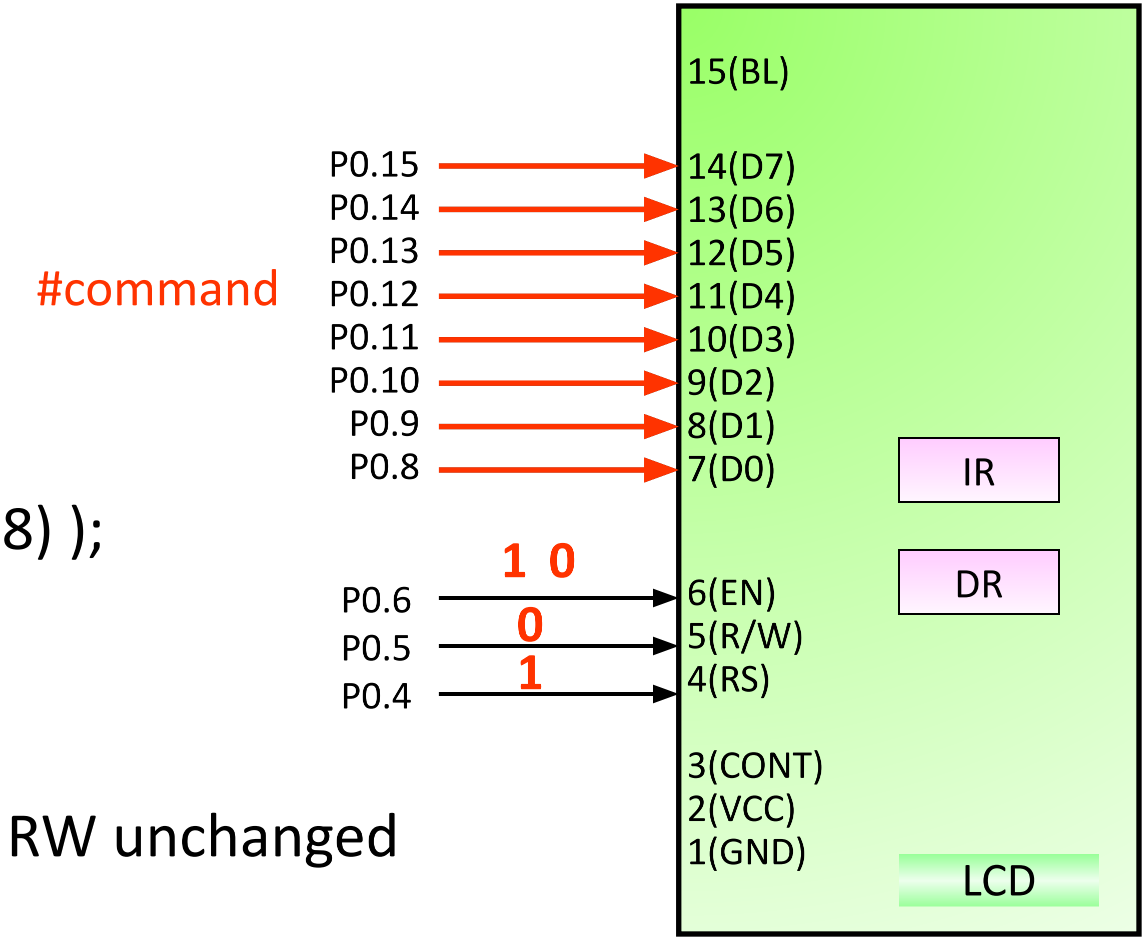
- Transfer instruction 8 bit data to Port (Via Data bus)
- Enable signal to start write operation (EN = 1)
- Select instruction Reg. (RS = 1)
- Select write (R/W = 0)
- EN = 0



Command Write Function

Data Write Function

```
void LCD_CHAR (char msg)
{
IO0PIN = ( (IO0PIN & 0xFFFF00FF) | (msg<<8) );
IO0SET = 0x00000050;      // RS = 1, , EN = 1
IO0CLR = 0x00000020;      // RW = 0
delay_ms(2);
IO0CLR = 0x00000040;      // EN = 0, RS and RW unchanged
delay_ms(5);
}
```



LCD instructions (8 bit Commands)

1. Function Set:
   - DL = Data Length (If DL=1, 8 bit & If DL=0, 4 bit)
   - N = No. of display lines (If N=0, 1 Line & If N=1, 2 Line)
   - F = Character Font (If F=0, 5x7 dot character font & F=1, 5x10
2. Display ON/OFF Control:
   - D = 0 ; Display OFF & D = 1 ; Display ON
   - C = 0 ; Cursor OFF & C = 1 ; Cursor ON
   - B = 0 ; Blink OFF & B = 1 ; Blink ON
3. Clear Display
4. Entry Mode Set:
   - I/D = 1 ; Increment Mode (Increment DD RAM address by 1)
   - S = 0 ; No display shift operation
5. Set DD RAM address:
   - Sets the address counter to the DD RAM address AAAAAAA.
     Data is then written / read to from the DD RAM.
   - For 2 line display module
     AAAAAAA = 00H to 27H for the first line &
     40H to 67H for the second line.

LCD instructions (Commands)

1. Function Set Instruction

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	DL	N	F	X	X

DL: Sets interface data length

If DL = 1, data is sent or received in 8 bit length (D7 – D0).

if DL = 0, data is sent or received in 4 bit length (D7 – D4). When 4 bit length is selected data must be sent or received twice.

N: Sets the number of lines

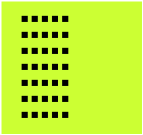
If N = 0 ; 1 line display

If N = 1 ; 2 line display

F: Sets character font

If F = 1 ; 5 x 10 dots

If F = 0 ; 5 x 7 dots



e.g. For 8 bit data length, 2 line display, 5 x 7 dots.

Function Set = 0011 1000 = 38H

LCD instructions (Commands)

2. Display ON/OFF Control

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	D	C	B

D : If D = 1, display is ON. If D = 0, display is OFF

C : If C = 1, cursor is displayed. If C = 0, cursor is not displayed.

B : If B = 1, Cursor blink is ON, If B = 0, Cursor blink is OFF.

e.g. For Display ON, Cursor display & blinking.

Display ON = 0000 1111 = 0FH

3. Clear Display

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1

= **01H**

Writes the space code “20H” into all addresses of DD RAM.

LCD instructions (Commands)

4. Entry Mode Set

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	I/D	S

Sets the increment / decrement and shift modes to the desired settings.

I/D : Increments (I/D = 1) or decrements (I/D = 0) the DD RAM address
by 1 when a character code is written or read from the DD RAM.
The cursor moves to the right when incremented by +1

S : Shift the entire display either to the right or to the left when S = 1.
If I/D = 1, shift to the right. If I/D = 0, shift to the left.

e.g.   DD RAM address should be incremented, No display shift.

Entry Mode Set = 0000 0110 = 06H

LCD instructions (Commands)

5. Set DD RAM Address

D7	D6	D5	D4	D3	D2	D1	D0
1	A	A	A	A	A	A	A

Sets the address counter to the DD RAM address AAAAAAA. Data is then written / read to from the DD RAM.

For 2 line display module AAAAAAA = 00H to 27H for the first line &
= 40H to 67H for the second line.

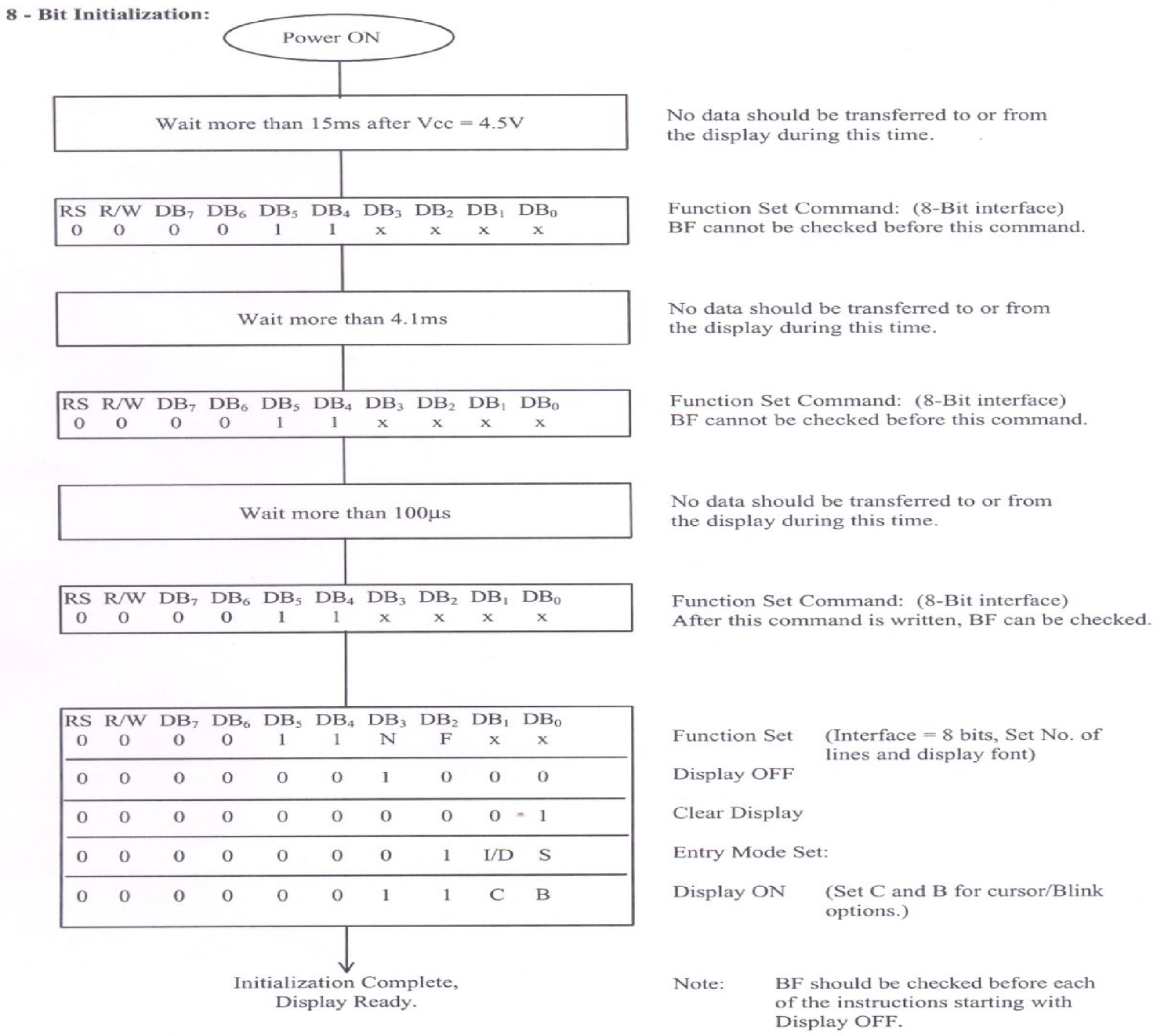
e.g. **To display characters in the first line**
Set DD RAM address = 1000 0000 = 80H

e.g. **To display characters in the second line**
Set DD RAM address = 1100 0000 = C0H

Summary of LCD instructions

Instruction	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
Function Set	0	0	0	0	1	DL	N	F	X	X
Display ON/OFF	0	0	0	0	0	0	1	D	C	B
Clear Display	0	0	0	0	0	0	0	0	0	1
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
Set DD RAM address	0	0	1	A	A	A	A	A	A	A
Write data to DD RAM	1	0	Write data							
Read data from DD RAM	1	1	Read data							

Initialization of LCD (8 bit)



8 - Bit Initialization:

Power ON

Wait more than 15ms after Vcc = 4.5V

No data should be transferred to or from the display during this time.

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	x	x	x	x

Function Set Command: (8-Bit interface)
BF cannot be checked before this command.

Wait more than 4.1ms

No data should be transferred to or from the display during this time.

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	x	x	x	x

Function Set Command: (8-Bit interface)
BF cannot be checked before this command.

Wait more than 100μs

No data should be transferred to or from the display during this time.

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	x	x	x	x

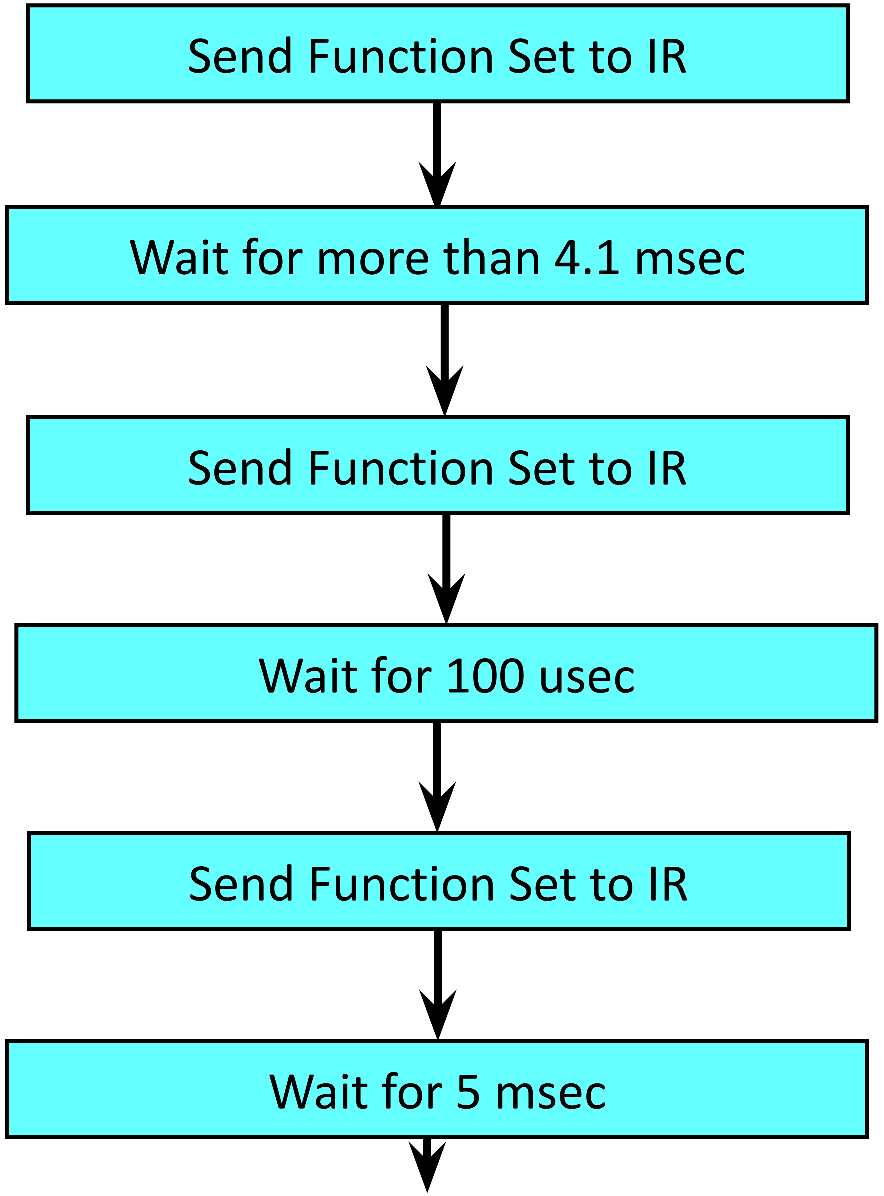
Function Set Command: (8-Bit interface)
After this command is written, BF can be checked.

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0	
0	0	0	0	1	1	N	F	x	x	Function Set (Interface = 8 bits, Set No. of lines and display font)
0	0	0	0	0	0	1	0	0	0	Display OFF
0	0	0	0	0	0	0	0	0	1	Clear Display
0	0	0	0	0	0	0	1	I/D	S	Entry Mode Set:
0	0	0	0	0	0	1	1	C	B	Display ON (Set C and B for cursor/Blink options.)

Initialization Complete,
Display Ready.

Note: BF should be checked before each of the instructions starting with Display OFF.

Initialization Flowchart



Send Function Set to IR

Wait for more than 4.1 msec

Send Function Set to IR

Wait for 100 usec

Send Function Set to IR

Wait for 5 msec

; Initialize LCD
; Send Function Set
; instruction 3 times

; wait for 5 msec

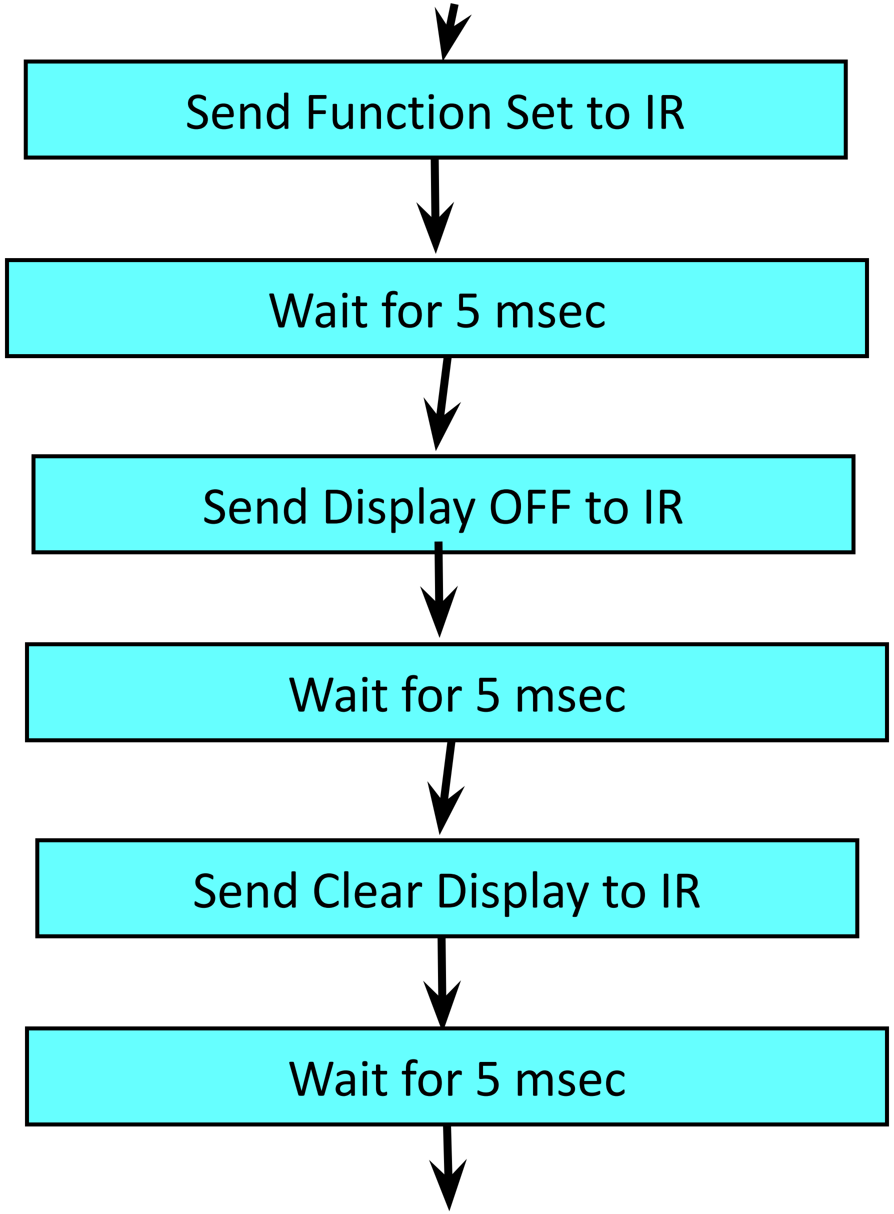
;Send Function Set instruction

; wait for 5 msec

;Send Function Set Instruction

; wait for 5 msec

Initialization Flowchart …



Send Function Set to IR

Wait for 5 msec

Send Display OFF to IR

Wait for 5 msec

Send Clear Display to IR

Wait for 5 msec

; ;Send Function Set Instruction

; wait for 5 msec

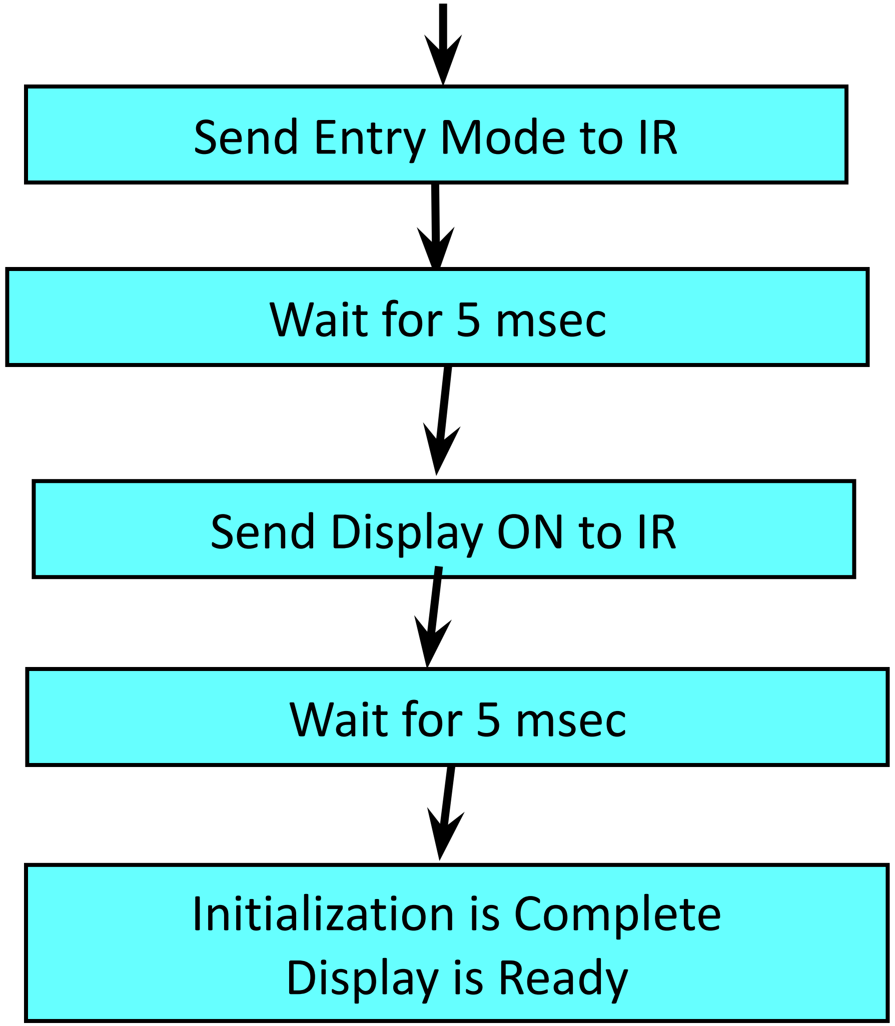
; Display OFF instruction

; wait for 5 msec

; Clear Display Instruction

; wait for 5 msec

Initialization Flowchart …



Send Entry Mode to IR

Wait for 5 msec

Send Display ON to IR

Wait for 5 msec

Initialization is Complete
Display is Ready

; ;Send Entry Mode Instruction

; wait for 5 msec

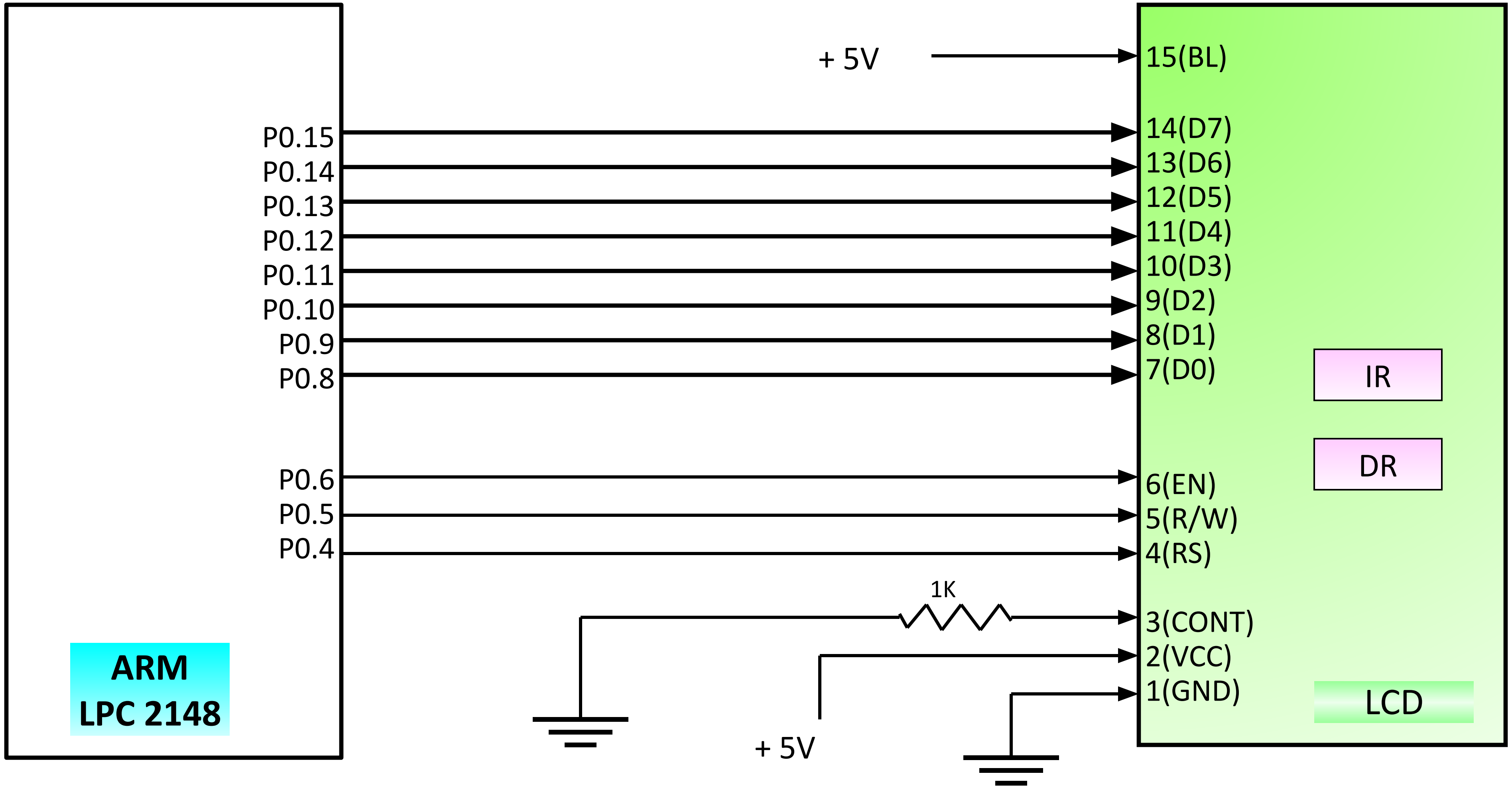
; Display ON instruction

; wait for 5 msec

; Initialization Done

LCD Program #1

- Interface 16x2 LCD with LPC2148.Use 8 bit data length and write program to isplay “Y” in first line. Use P0 for data pins and control pins



LCD instructions Setting

1. Function Set

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	DL	N	F	X	X

DL = 1 ; data is sent or received in 8 bit length (D7 – D0).

N = 1 ; 2 line display

F = 0 ; 5 x 7 dots

Function Set	0	0	1	1	1	0	0	0

LCD instructions Setting

2. Display ON/OFF Control

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	D	C	B

D = 1 ;display is ON
C = 1 ;cursor is displayed
B = 1 ;Cursor blink

Display ON Control	0	0	0	0	1	1	1	1	**= 0FH**
Display OFF Control	0	0	0	0	1	0	0	0	**= 08H**

3. Clear Display

D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	1	= **01H**

LCD instructions Setting

4. Entry Mode Set

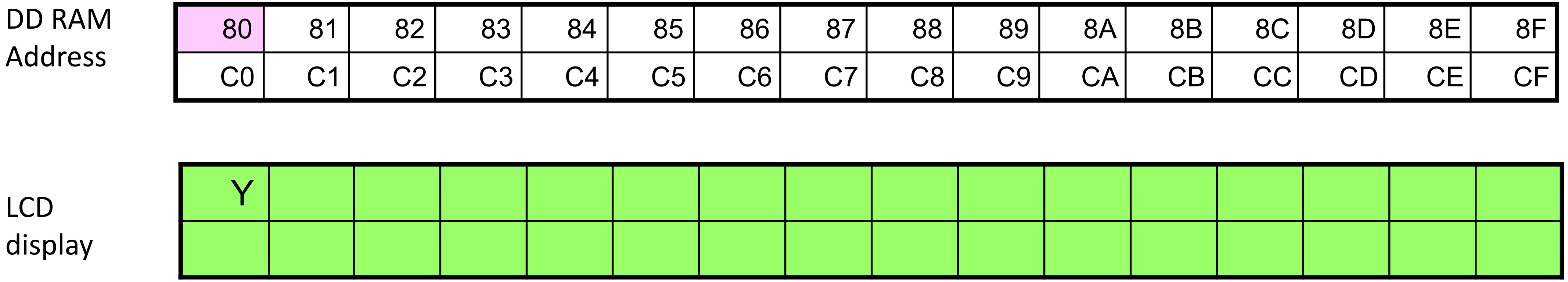
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	I/D	S

I/D = 1 ;Increments the DD RAM address by 1 when a character code is
;written or read from the DD RAM.
;The cursor moves to the right when incremented by +1
S = 0   ; No display shift

Entry Mode Set	0	0	0	0	0	1	1	0	**= 06H**

LCD instructions Setting

5. DD RAM Address



DD RAM Address															
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

LCD display															
Y															

e.g. To display characters in the first line, first position

Set DD RAM address = 1000 0000 = 80H

Program LCD Interfacing

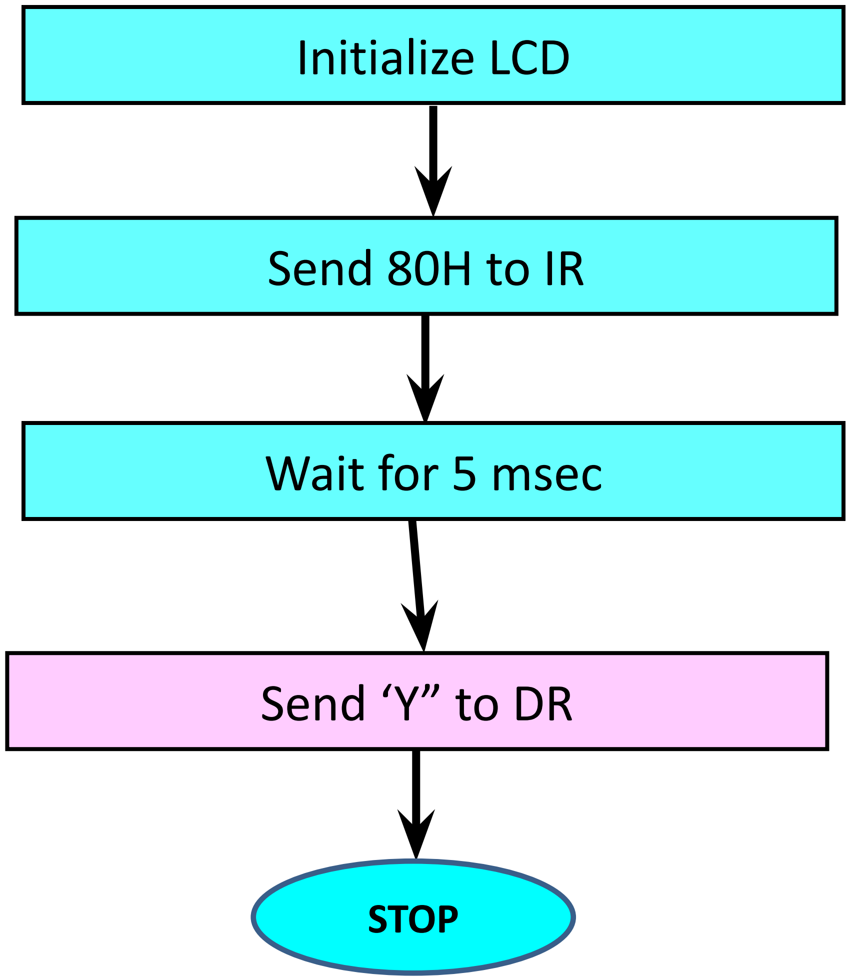
```
#include <lpc214x.h>

// Function Prototypes
void LCD_CMD(unsigned char command);
void LCD_DATA(unsigned char data);
void LCD_INIT(void);
void delay_ms(unsigned int ms);

int main(void)
{
   IO0DIR = 0x0000FFFF;  // P0.0–P0.15 configured as output (data +
control pins)
   LCD_INIT();           // Initialize LCD

   LCD_CMD(0x80);        // Move cursor to first line, first position
   LCD_DATA('Y');        // Display character 'Y'

   while(1);             // Infinite loop
}
```

```
// Function to send command to LCD
//----------------------------------------------
void LCD_CMD(unsigned char command)
{
    IO0PIN = ((IO0PIN & 0xFFFF00FF) | (command << 8));  // Send command to P0.8–P0.15

    IO0CLR = 0x00000030;  // RS = 0, RW = 0
    IO0SET = 0x00000040;  // EN = 1
    delay_ms(2);
    IO0CLR = 0x00000040;  // EN = 0
    delay_ms(5);
}
```

```
// Function to send data to LCD
//---------------------------------------------
void LCD_DATA(unsigned char data)
{
   IO0PIN = ((IO0PIN & 0xFFFF00FF) | (data << 8));   // Send
data to P0.8–P0.15

   IO0SET = 0x00000010;  // RS = 1
   IO0CLR = 0x00000020;  // RW = 0
   IO0SET = 0x00000040;  // EN = 1
   delay_ms(2);
   IO0CLR = 0x00000040;  // EN = 0
   delay_ms(5);
}
```

```
//------------------------------------------
// LCD Initialization Sequence
//------------------------------------------
void LCD_INIT(void)
{
   delay_ms(20);       // LCD power-on delay

   LCD_CMD(0x38);      // 8-bit mode, 2-line, 5x7 matrix
   LCD_CMD(0x0C);      // Display ON, Cursor OFF
   LCD_CMD(0x06);      // Auto-increment cursor
   LCD_CMD(0x01);      // Clear display
   LCD_CMD(0x80);      // Move cursor to first line
}

//------------------------------------------
// Simple software delay
//------------------------------------------
void delay_ms(unsigned int ms)
{
   unsigned int i, j;
   for(i = 0; i < ms; i++)
      for(j = 0; j < 2000; j++);   // Approx 1ms delay for 12MHz PCLK
}
```

Flowchart

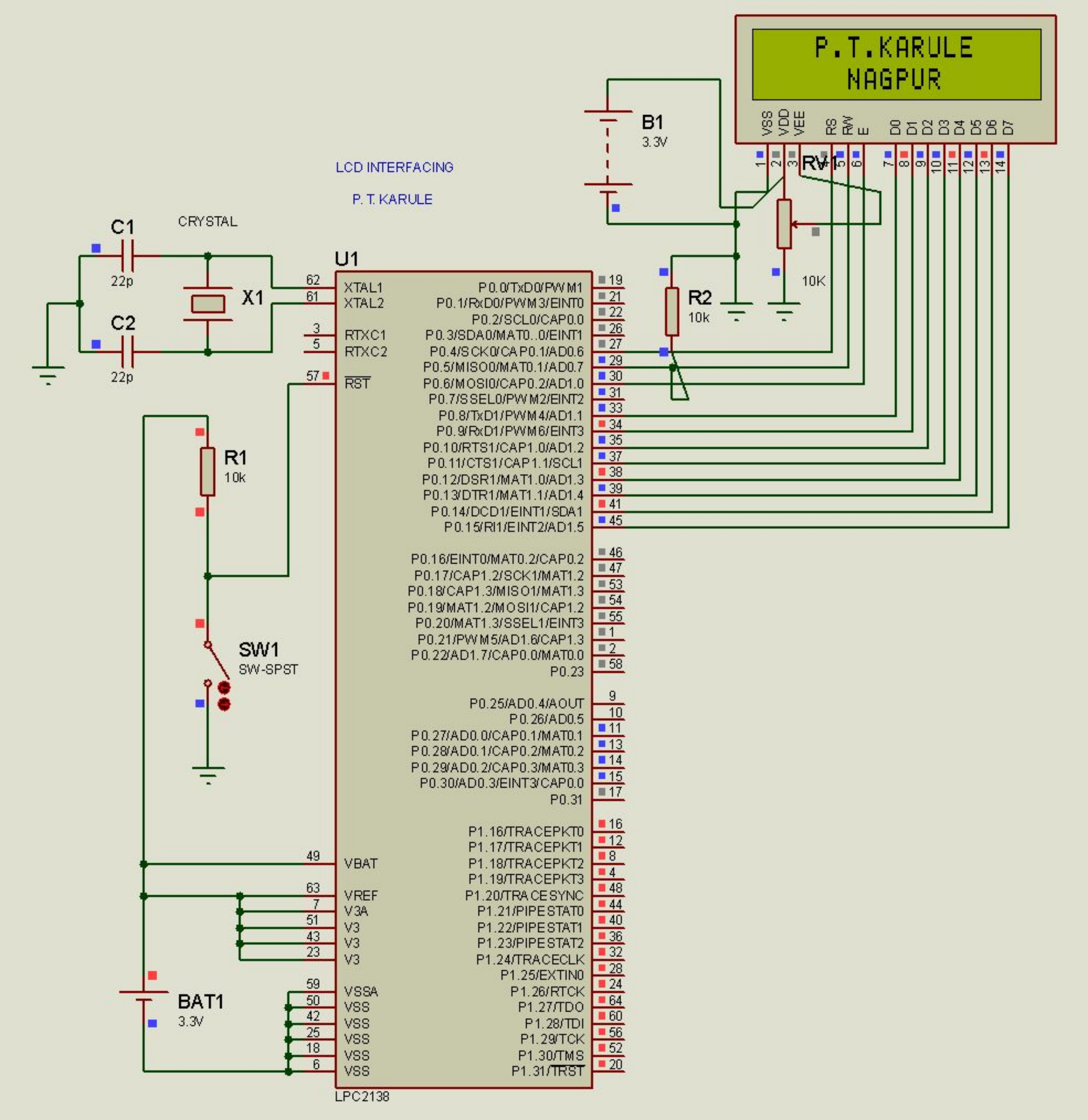


; Initialize LCD

; DD RAM Address
: First Line, first position

; wait for 5 msec

; ASCII CODE OF ‘Y’


P.T.KARULE
NAGPUR
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
1 2 3 4 5 6 7 8 9 10 11 12 13 14
B1
3.3V
RV1
10K
LCD INTERFACING
P. T. KARULE
CRYSTAL
C1
22p
C2
22p
X1
U1
R2
10k
62 XTAL1
61 XTAL2
3 RTXC1
5 RTXC2
57 RST
R1
10k
SW1
SW-SPST
49 VBAT
63 VREF
7 V3A
51 V3
43 V3
23 V3
BAT1
3.3V
59 VSSA
50 VSS
42 VSS
25 VSS
18 VSS
6 VSS
P0.0/TxD0/PWM1 19
P0.1/RxD0/PWM3/EINT0 21
P0.2/SCL0/CAP0.0 22
P0.3/SDA0/MAT0..0/EINT1 26
P0.4/SCK0/CAP0.1/AD0.6 27
P0.5/MISO0/MAT0.1/AD0.7 29
P0.6/MOSI0/CAP0.2/AD1.0 30
P0.7/SSEL0/PWM2/EINT2 31
P0.8/TxD1/PWM4/AD1.1 33
P0.9/RxD1/PWM6/EINT3 34
P0.10/RTS1/CAP1.0/AD1.2 35
P0.11/CTS1/CAP1.1/SCL1 37
P0.12/DSR1/MAT1.0/AD1.3 38
P0.13/DTR1/MAT1.1/AD1.4 39
P0.14/DCD1/EINT1/SDA1 41
P0.15/RI1/EINT2/AD1.5 45
P0.16/EINT0/MAT0.2/CAP0.2 46
P0.17/CAP1.2/SCK1/MAT1.2 47
P0.18/CAP1.3/MISO1/MAT1.3 53
P0.19/MAT1.2/MOSI1/CAP1.2 54
P0.20/MAT1.3/SSEL1/EINT3 55
P0.21/PWM5/AD1.6/CAP1.3 1
P0.22/AD1.7/CAP0.0/MAT0.0 2
P0.23 58
P0.25/AD0.4/AOUT 9
P0.26/AD0.5 10
P0.27/AD0.0/CAP0.1/MAT0.1 11
P0.28/AD0.1/CAP0.2/MAT0.2 13
P0.29/AD0.2/CAP0.3/MAT0.3 14
P0.30/AD0.3/EINT3/CAP0.0 15
P0.31 17
P1.16/TRACEPKT0 16
P1.17/TRACEPKT1 12
P1.18/TRACEPKT2 8
P1.19/TRACEPKT3 4
P1.20/TRACESYNC 48
P1.21/PIPESTAT0 44
P1.22/PIPESTAT1 40
P1.23/PIPESTAT2 36
P1.24/TRACECLK 32
P1.25/EXTIN0 28
P1.26/RTCK 24
P1.27/TDO 64
P1.28/TDI 60
P1.29/TCK 56
P1.30/TMS 52
P1.31/TRST 20
LPC2138

```
#include <lpc214x.h>
#include <stdint.h>
#include <stdlib.h>
#include <stdio.h>

void delay_ms(uint16_t j)            /* Function for delay in milliseconds  */
{
   uint16_t x,i;
     for(i=0;i<j;i++)
     {
   for(x=0; x<6000; x++);    /* loop to generate 1 millisecond delay with Cclk = 60MHz */
     }
}
```

```
void LCD_CMD(char command)
{
    IO0PIN = ( (IO0PIN & 0xFFFF00FF) | (command<<8) );
    IO0SET = 0x00000040;    /* EN = 1 */
    IO0CLR = 0x00000030;    /* RS = 0, RW = 0 */
    delay_ms(2);
    IO0CLR = 0x00000040;    /* EN = 0, RS and RW unchanged */
    delay_ms(5);
}
```

```
void LCD_INIT(void)
{
    IO0DIR = 0x0000FFF0; /* P0.8 to P0.15 LCD Data. P0.4,5,6 as RS RW and EN */
    delay_ms(20);
    LCD_CMD(0x38);  /* Initialize lcd */
    LCD_CMD(0x0C);      /* Display on cursor off */
    LCD_CMD(0x06);  /* Auto increment cursor */
    LCD_CMD(0x01);      /* Display clear */
    LCD_CMD(0x83);  /* First line fourth position */
}
```

```
void LCD_STRING (char* msg)
{
    uint8_t i=0;
    while(msg[i]!=0)
    {
    IO0PIN = ( (IO0PIN & 0xFFFF00FF) | (msg[i]<<8) );
    IO0SET = 0x00000050;     /* RS = 1, , EN = 1 */
    IO0CLR = 0x00000020;     /* RW = 0 */
    delay_ms(2);
    IO0CLR = 0x00000040;     /* EN = 0, RS and RW
    delay_ms(5);
    i++;
    }
}
```

```
void LCD_CHAR (char msg)
{
        IO0PIN = ( (IO0PIN & 0xFFFF00FF) | (msg<<8) );
        IO0SET = 0x00000050;      /* RS = 1, , EN = 1 */
        IO0CLR = 0x00000020;      /* RW = 0 */
        delay_ms(2);
        IO0CLR = 0x00000040;      /* EN = 0, RS and RW unchanged  */
        delay_ms(5);
}
```

```
int main(void)
{
    LCD_INIT();
    LCD_STRING("P.T.KARULE");
    LCD_CMD(0xC5);              /* Second line sixth position */
    LCD_STRING("NAGPUR");
    return 0;
}
```