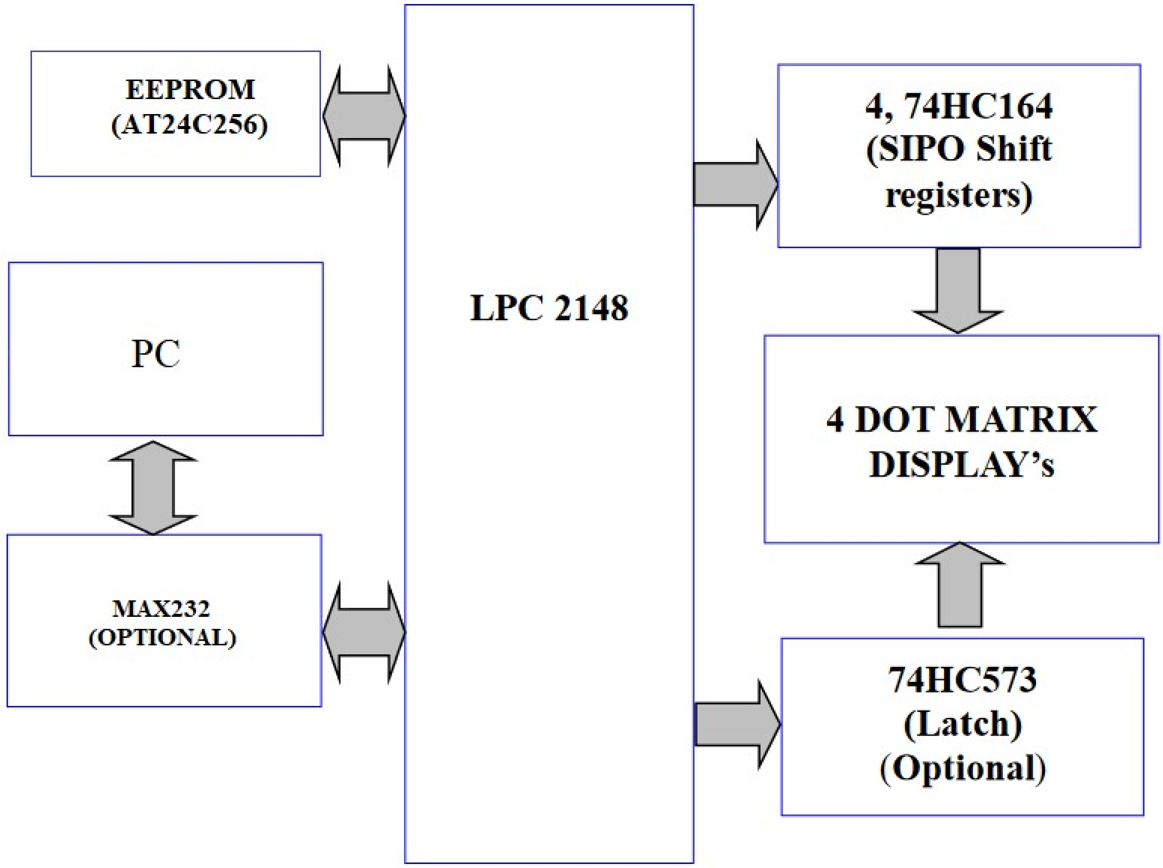
FLEXIBLE DISPLAY OVER DOT-MATRIX LED

AIM:

Customize display on 4 multiplexed, 8x8 dot matrix displays for application’s requirement.

BLOCK DIAGRAM:



CONNECTIONS TO BE USED:

DOT MATRIX DISPLAY:

Dot matrix display connections are established with the help of two different IC'S. Those are 74HC164 (8-bit serial-in, parallel-out shift register) & 74HC573 (Octal D-type transparent latch).

74HC164: (below mention connections are for 4 dot matrix displays**)**

FOR DISPLAY1:

DSA (74HC164)	-	P0.8
CP (74HC164)	-	P0.9

Q0 (74HC164)	-	COL1 (DOT MATRIX DISPLAY)
Q1 (74HC164)	-	COL2 (DOT MATRIX DISPLAY)
Q2 (74HC164)	-	COL3 (DOT MATRIX DISPLAY)
Q3 (74HC164)	-	COL4 (DOT MATRIX DISPLAY)
Q4 (74HC164)	-	COL5 (DOT MATRIX DISPLAY)
Q5 (74HC164)	-	COL6 (DOT MATRIX DISPLAY)
Q6 (74HC164)	-	COL7 (DOT MATRIX DISPLAY)
Q7 (74HC164)	-	COL8 (DOT MATRIX DISPLAY)

Note: Dot matrix display to 74HC164 connections are fixed on the board

FOR DISPLAY2:

DSA (74HC164)	-	P0.10
CP (74HC164)	-	P0.11

Q0 (74HC164)	-	COL1 (DOT MATRIX DISPLAY)
Q1 (74HC164)	-	COL2 (DOT MATRIX DISPLAY)
Q2 (74HC164)	-	COL3 (DOT MATRIX DISPLAY)
Q3 (74HC164)	-	COL4 (DOT MATRIX DISPLAY)
Q4 (74HC164)	-	COL5 (DOT MATRIX DISPLAY)
Q5 (74HC164)	-	COL6 (DOT MATRIX DISPLAY)
Q6 (74HC164)	-	COL7 (DOT MATRIX DISPLAY)
Q7 (74HC164)	-	COL8 (DOT MATRIX DISPLAY)

Note: Dot matrix display to 74HC164 connections are fixed on the board

FOR DISPLAY3:

DSA (74HC164)	-	P0.12

CP (74HC164)	-	P0.13

Q0 (74HC164)	-	COL1 (DOT MATRIX DISPLAY)
Q1 (74HC164)	-	COL2 (DOT MATRIX DISPLAY)
Q2 (74HC164)	-	COL3 (DOT MATRIX DISPLAY)
Q3 (74HC164)	-	COL4 (DOT MATRIX DISPLAY)
Q4 (74HC164)	-	COL5 (DOT MATRIX DISPLAY)
Q5 (74HC164)	-	COL6 (DOT MATRIX DISPLAY)
Q6 (74HC164)	-	COL7 (DOT MATRIX DISPLAY)
Q7 (74HC164)	-	COL8 (DOT MATRIX DISPLAY)

Note: Dot matrix display to 74HC164 connections are fixed on the board

FOR DISPLAY4:

DSA (74HC164)	-	P0.14
CP (74HC164)	-	P0.15

Q0 (74HC164)	-	COL1 (DOT MATRIX DISPLAY)
Q1 (74HC164)	-	COL2 (DOT MATRIX DISPLAY)
Q2 (74HC164)	-	COL3 (DOT MATRIX DISPLAY)
Q3 (74HC164)	-	COL4 (DOT MATRIX DISPLAY)
Q4 (74HC164)	-	COL5 (DOT MATRIX DISPLAY)
Q5 (74HC164)	-	COL6 (DOT MATRIX DISPLAY)
Q6 (74HC164)	-	COL7 (DOT MATRIX DISPLAY)
Q7 (74HC164)	-	COL8 (DOT MATRIX DISPLAY)

Note: Dot matrix display to 74HC164 connections are fixed on the board

For data input user has to use 0-7 pins on the dot matrix board. Check the dot matrix board for 0-7 pins. For better understanding refer the below mentioned details.

74HC164_1	-	0 - DSA
		1 - CP
74HC164_2	-	2 - DSA
		3 - CP
74HC164_3	-	4 - DSA
		5 - CP
74HC164_4	-	6 - DSA
		7 – CP

74HC573:

D0 (74HC573)	-	P0.0
D1 (74HC573)	-	P0.1

D2 (74HC573)	-	P0.2
D3 (74HC573)	-	P0.3
D4 (74HC573)	-	P0.4
D5 (74HC573)	-	P0.5
D6 (74HC573)	-	P0.6
D7 (74HC573)	-	P0.7

Q0 (74HC573)	-	ROW8 (DOT MATRIX DISPLAY)
Q1 (74HC573)	-	ROW7 (DOT MATRIX DISPLAY)
Q2 (74HC573)	-	ROW6 (DOT MATRIX DISPLAY)
Q3 (74HC573)	-	ROW5 (DOT MATRIX DISPLAY)
Q4 (74HC573)	-	ROW4 (DOT MATRIX DISPLAY)
Q5 (74HC573)	-	ROW3 (DOT MATRIX DISPLAY)
Q6 (74HC573)	-	ROW2 (DOT MATRIX DISPLAY)
Q7 (74HC573)	-	ROW1 (DOT MATRIX DISPLAY)

Note: Dot matrix display to 74HC573 connections are fixed on the board

Based on the project requirement, need to change the port pin connections of the DOT matrix board.

REQUIREMENTS:

HARDWRAE REQUIREMENTS:

- Lpc2148
- FOUR 8X8 DOT MATRIX DISPLAY'S
- 74HC573 (D-type transparent latches)
- 74HCT164 (8-bit serial-in/parallel-out)
- AT24C256
- MAX232
- USB to UART Converter

SOFTWARE REQUIREMENTS:

- EMBEDDED C – PROGRAMMING
- KEIL-C COMPILER
- FLASH MAGIC

BLOCK DIAGRAM EXPLANATION WITH PROJECT FLOW:

Display patterns can be selected for flexible display by using standard input via PC. Initially mode of operation (RUN mode / EDIT mode) is displayed on the hyper terminal. In this application which operation it has to run that mode status is saved in the EEPROM memory location (EEPROM basic program was already uploaded in LMS CARD. So, download use that code for your reference). Application program is reading mode status from EEPROM and based on the mode it is going to perform the particular task. Same task is performed unless until it is receiving '!' symbol form the UART. Once it is received from the PC through serial communication then below mentioned menu is displayed on the hyper terminal.

1) FIXED STRING
2) FIXED STRING WITH BLINKING
3) STRING WITH SCROLLING
4) TIME DISPLAY
5) RTC DISPLAY WITH SCROLLING
6) TEXT EDIT MODE
7) TIME EDIT MODE
8) EXIT

After menu is displayed application program is waiting for menu selection. Once menu is selected from the list then that particular task is performed. Here EEPROM is used for saving the mode status and strings which are displayed in the first three options which are available in the menu.

SEQUENCE TO BE FOLLOWED FOR IMPLEMENTATION:

- Create New Folder in your server save that folder with your project name and use same folder for required files for your project completion.
- First write one simple program to display one character on the single dot matrix led (Already sample code was uploaded in LMS card. So, download and utilise that code)
- Then write a program to display four-character string on the four-dot matrix Led's

  Example: HELP

- Then prepare code for string scrolling on the four-dot matrix Led's (which has to display scrolling more than 10 characters string)

  Example: PROJECT SUCUSSFULLY COMPLETED

- Then prepare UART program for displaying menu and menu selection using receive logic, both transmit and receive options are required to monitor by using **serial interrupts** only.
- Then display our required menu on the hyper terminal.
- Then complete the first three options by using available program.
- Then go for the fourth option i.e. display time (min & sec)
- Read the minutes and seconds from the on-chip RTC and convert those vales into string and display that string on the dot matrix Led's

  Example: MIN SEC

  45 42 45 43 45 44 ….. etc.

- Next option is displaying the total time with scrolling. For that read the all required RTC supported SFR'S from the on-chip RTC and form a string. Once the string is ready with all timing information then that string is scrolling on the dot matrix Led's which is already working in the third option.

  Syntax: TIME: HH:MM:SS DATE:DD/MM/YY DAY: SUN to SAT

  Example: TIME: 09:30:23 DATE:17/04/2015 DAY: FRIDAY

- Coming to sixth option i.e. text editing. To edit the string user input is required from the PC through hyper terminal. Here again two options are there. One is for fixed string which is having the only four characters and second one is big string which is having the more than four characters. These two strings are saved the EEPROM memory locations in different places.

  FIXED TEXT (only 4 characters)

  TEXT WITH SCROLLING (only 20 characters)

- And finally, time editing. For this user has to follow the simple syntax which is given below. After giving the information based on application program updated values are saved in the RTC equivalent SFR'S.

  Enter for editing time & date as per the format given below:

SS:MM:HH	DAY	DD/MM/YY
22:10:09	01	27/02/15

  Note: DAY: 01-SUN,02-MON,03-TUE,04-WED,05-THU,06-FRI,07-SAT

  Entry limits is as follows:

  (sec - 00-59) (min - 00 - 59) (hour - 00 - 23)

  (day - 01-07)

  (date - 01 - 31) (month - 01 - 12) (year - 00 – 99)

- If above steps are completed successfully, then your project is done.