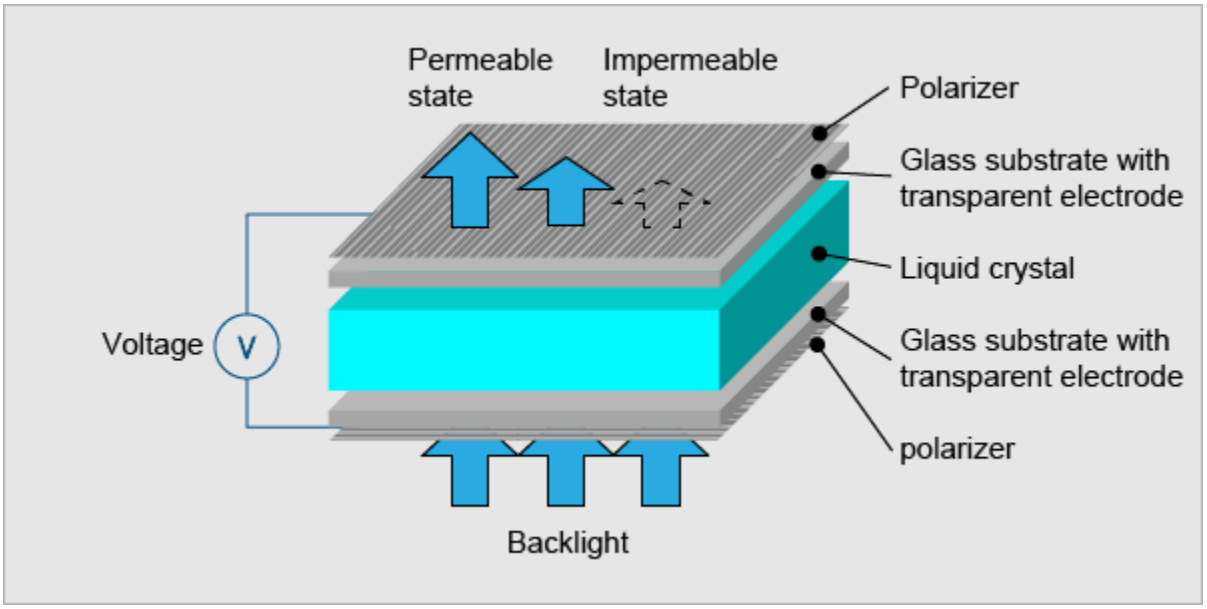
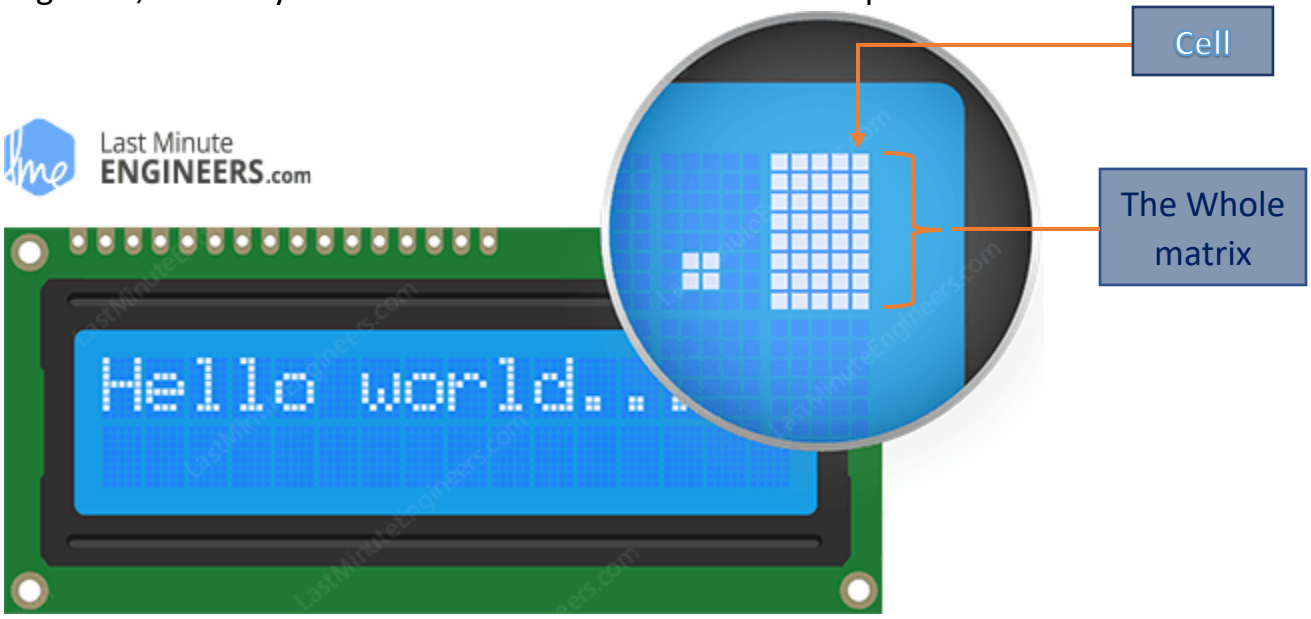
Session-6 (LCD)

- The work principle of LCD:
  - Is that there are two electrodes and between them the liquid crystal, if we apply some voltage to those two electrodes the liquid crystal between them will turn its color to black, remove the voltage, the liquid crystal will return to its default color (grey).



- So, based on that principle of working, we can combine more than one cell of liquid crystal together, which by that we can make a matrix like in the photo below.

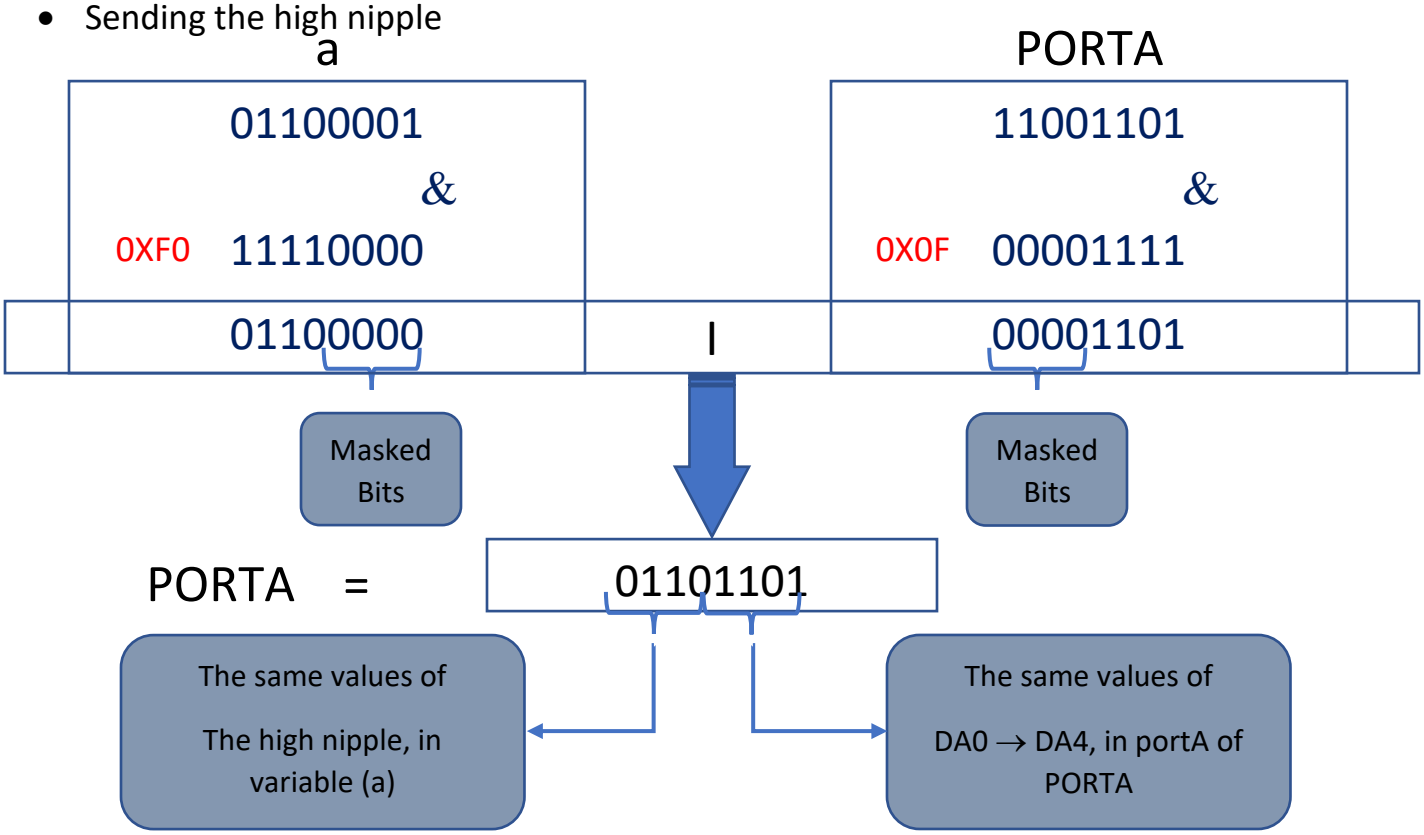


- So, at that single matrix we can display any character of our choice!
- In the LCD, there is a potentiometer its job is to control the intensity of each cell of the whole LCD, which means that if we increased the value of the pot, the output voltage will decrease, and vice versa, which means if we increase the value of the pot, the intensity decreases which means that the displayed characters on the LCD will look dimed and dimed until the shown character cannot be visible anymore, which mean that the value of pot is at the maximum value and no voltage is supplied to the LCD's cells.

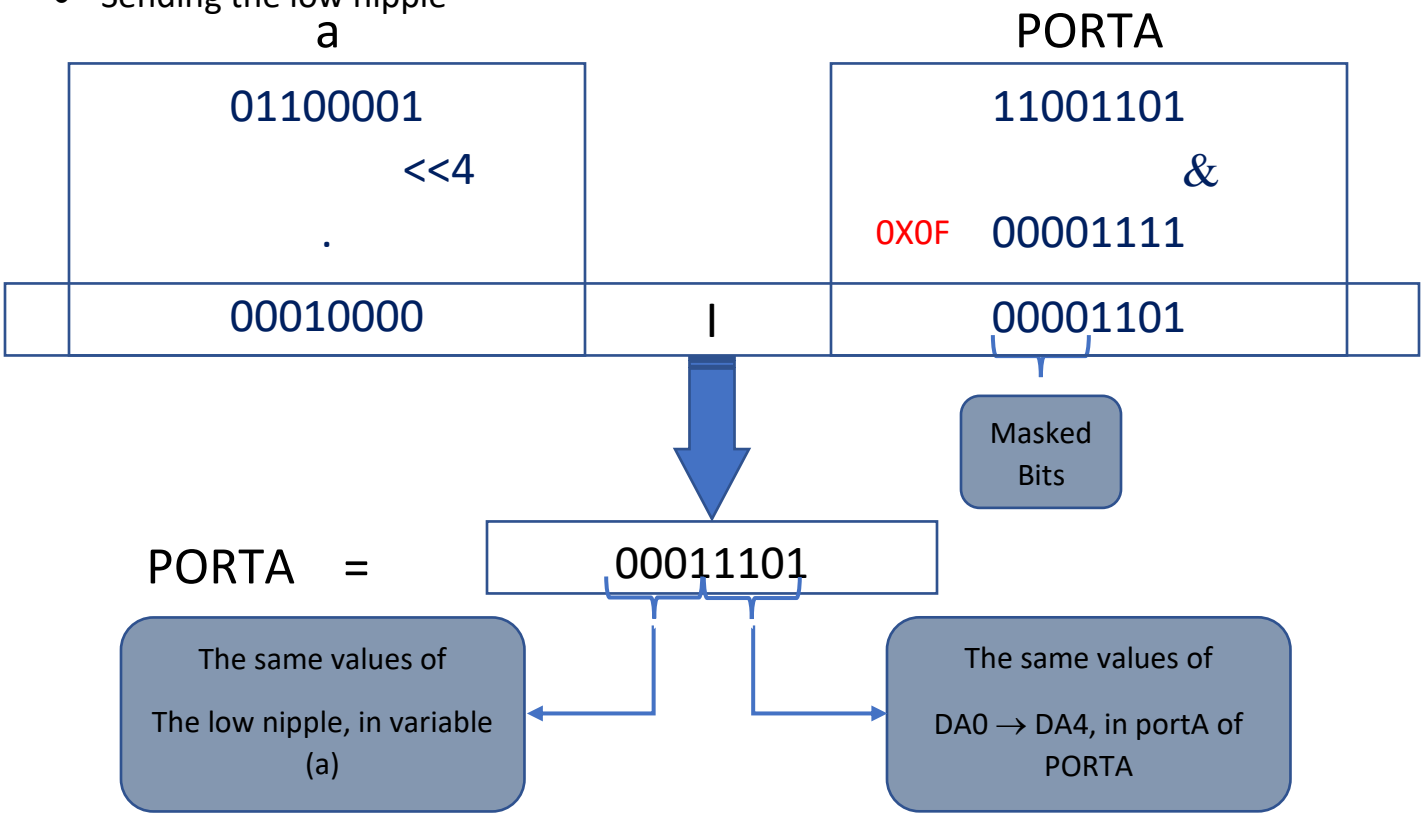
- The technology of character LCD depends that to display the cells of LCD clearly, there must be a backlight with a white color that hits the LCD from behind. If the backlight is off, we will not be able to see the displayed characters clearly.

- With 4-bit mode of communicating with the LCD, we must send the binary to the last four pins in the LCD, and any ASCII or command in binary we must send the biggest four bits which is called the high nipple, then to the low nipple.
- The advantage of using 4-bit mode is that instead of using 11 pin to connect the LCD we here use only 7 pins.
- The disadvantage of using 4-bit mode is that it is slower than 8-bit mode by half, as every 8 bits sent in one clock cycle by 8-bit mode, will be sent in two clock cycles in 4-bit mode.
- So, now how to send ASCII of the character (a) which is consisting of 8 bits, through 4 pins?!
- Simply it is done using what we call Masking, so let's say that we connect the LCD to the MC with the pins DA4 $\rightarrow$ DA7 in MC to the pins D4 $\rightarrow$ D7 in LCD, and we want to send the character (a) and its ASCII binary is 01100001, and at the same time we connect other devices to the pins DA0 $\rightarrow$ DA3, so right now the value of PORTA register of portA is 11001101, so what we do to send the high nipple of the character (a) without affecting the value of each pin of DA0 $\rightarrow$ DA3, we mask the low nipple of the character (a), and also mask the high nipple of PORTA register of portA, then ORing the two binaries and store the result value in PORTA register of portA.

- Sending the high nipple



- Sending the low nipple



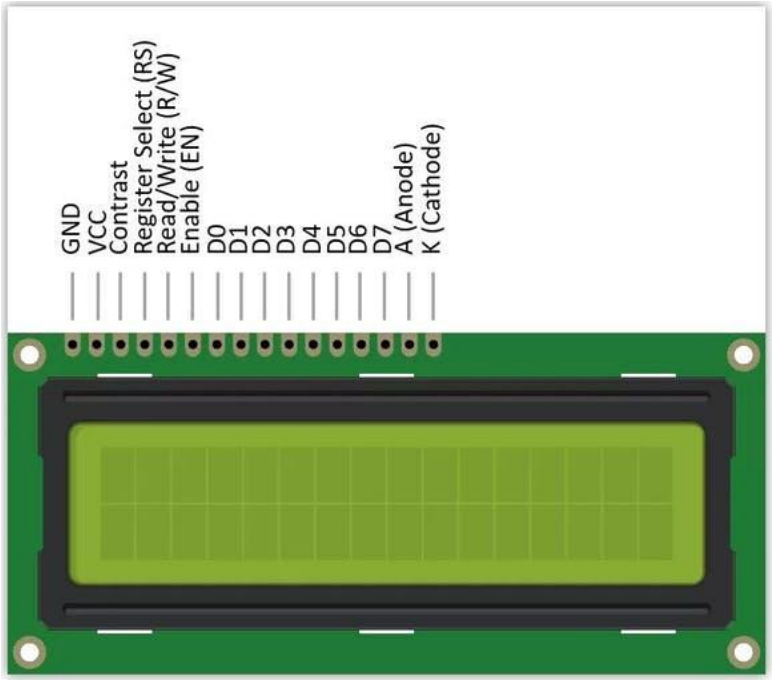
– In the build of HAL devices, we need header file dedicated to configurations, with the usual two files (driver.c), (driver.h), which is called (driver_CFG.h), so when we change the connection of that device with the MC, we don't change any file except the header file (driver_CFG.h).

Character LCD

Overview

Character LCD 2 X 16 consists of two rows, and 16 columns, with total 32 cell each cell is a matrix of dots consists of 5 columns X 8 rows with total 5 x 8 dots, so for the single cell there are 40 dots to display any character of our choice.

LCD Interface



So, to deal with this type of LCDs, we have to use 11 pins from the LCD, as three pins for control and they are RS, R/W, EN, and 8 pins to write/ read data, or command to/ from the LCD and these pins are from D0 to D7.

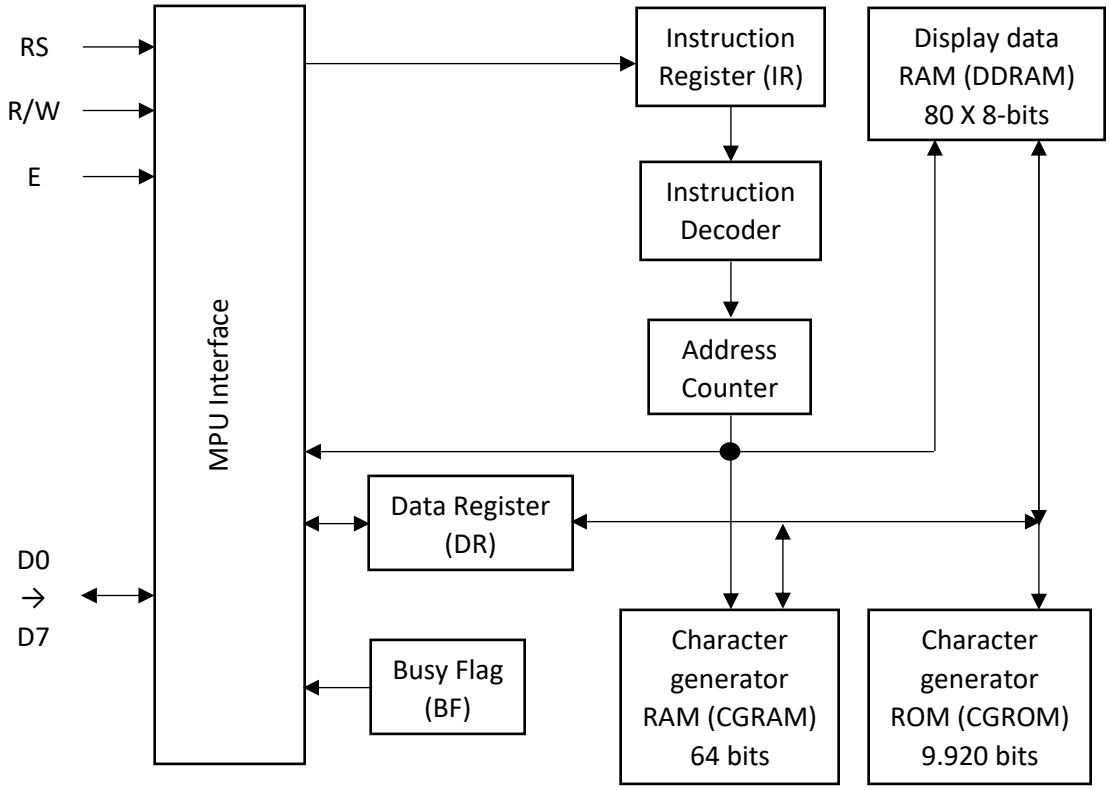
Pin Functions

Table 1. LCD Pin Functions

Name	Pin Number	I/O	Interface With	Function
GND	1	-	Power Supply	the ground pin of the chip
VCC	2	-	Power Supply	the power source of the chip, 2.7V to 5.5V
Contrast	3	-		it is the pin by which we control how much voltage will be supplied into each cell of the LCD. By using POT, and the output of the POT to Contrast pin of LCD.
RS	4	I	MPU	Register Select (RS) it is the pin by which we control whether the binary number the LCD received using 8 pins (D0 → D7), is data to be displayed on the LCD, or command we want the LCD to execute. So, if we send 0 to RS pin this mean that we send a command to LCD, if we send 1 to RS pin this mean that we send ASCII code to be displayed on LCD.
R/W	5	I	MPU	Read/Write (RW) it is the pin by which we control whether we want to read from LCD, or write into it. So, we send 0 to RW pin if we want to write on LCD, and 1 if we want to read from LCD RAM as we can use the RAM of LCD in any project, as a normal RAM that we can store data on, and read data from it.
E	6	I	MPU	Enable (EN) it is the pin by which we control that all binary data send using 8 pins, sent at the same time as in parallel communication the most critical problem is the timing skew, so the default of EN pin is low and when we send any data whether is ASCII, or command we write 1 to EN pin of dedicated amount of time usually 1ms then we turn it low again.
D0 → D7	7 - 14	I/O	MPU	These are the pins used to send/ receive either data or instruction to/ from the LCD.

LCD Block Diagram

Figure 2. Character LCD Block Schematic



Function Description

System Interface

This chip has all two kinds of interface type with Microelectronic Processing Unit (MPU): 4-bit bus and 8-bit bus. 4-bit bus or 8-bit bus is selected by DL bit in the instruction register;

1. 8-bit mode: where the binary data/command send or received through 8 buses in parallel.
2. 4-bit mode: where the binary data/command send or received through 4 buses in parallel.

During read or write operation, two 8-bit registers are used. One is data register (DR) to store the data sent to LCD at one of two memories of the LCD either DDRAM or CGRAM, the other 8-bits register is instruction register (IR) to temporary store the command to be executed by LCD.

The data register (DR) is used as temporary data storage place for being written into or read from DDRAM/CGRAM, target RAM is selected by RAM address setting instruction. Each internal operation, reading from or writing into RAM, is done automatically. So, to speak, after MPU reads DR data, the data in the next DDRAM/CGRAM address is transferred into DR automatically. Also, after MPU writes data to DR, the data in DR is transferred into DDRAM/CGRAM automatically.

The Instruction register (IR) is used only to store instruction code transferred from MPU. MPU cannot use it to read instruction data.

To select register, use RS input pin in 4-bit/8-bit bus mode.

Table 1. Various kinds of operations according to RS and R/W bits.

RS	RW	Operation
L	L	Instruction Write operation (MPU writes Instruction code into IR)
L	H	Read Busy Flag (DB7) and address counter (DB0 ~ DB6)
H	L	Data Write operation (MPU writes data into DR)
H	H	Data Read operation (MPU reads data from DR)

Busy Flag (BF)

When BF = "High", it indicates that the internal operation is being processed. So, during this time the next instruction cannot be accepted. BF can be read, when RS = Low and R/W = High (Read Instruction Operation), through DB7 port. Before executing the next instruction, be sure that BF is not High.

Address Counter (AC)

Address Counter (AC) stores DDRAM/CGRAM address, transferred from IR. After writing into (reading from) DDRAM/CGRAM, AC is automatically increased (decreased) by 1. When RS = "Low" and R/W = "High", AC can be read through DB0 ~ DB6 ports.

Display Data RAM (DDRAM)

DDRAM stored display data and its maximum capacity is 80 x 8 bits, or 80 characters.

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 x 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.

Each section (Address) of this RAM has a direct relationship with each cell of the character LCD, any binary number we store on any address of DDRAM will be translated into some character based on ASCII table which means that if we wrote on any address of DDRAM memory, the binary number corresponding to the decimal number 48 it will display the character '0' on the LCD, and further more we will know how exactly that happened!

So, there are two modes of display, 1-line display mode, and 2-line display mode. In 1-line display mode we can display data on the first line only of the two lines of the LCD, but in 2-line display mode we can write on both lines of the LCD.

- 1-line Display (N = 0)

  In 1-line display mode, there will be 80 different address to store the display data, the addresses of DDRAM will be from 0X00 to 0X4F where the address 0X00 represent the first cell of the first line of the LCD, and address 0X01 will represent the second cell, address 0X02 will represent the third cell, and so forth until the address 0X0F which will represent the last cell with number 16 of the LCD and the rest of the addresses we can write data on them but the value that will be stored at these addresses won't be shown unless we make shift to the whole display to the right.

  The following figure represent each DDRAM address corresponding to its position on the LCD and the two cases if we shift the entire display left and right.

Figure 2. 1-Line Display

Display Position	1	2	3	4	……	12	13	14	15	16	……..	78	79	80
DDRAM Address (Hexadecimal)	00	01	02	03	……	0B	0C	0D	0E	0F	……..	4D	4E	4F
	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	2 X 16 Character LCD													

Figure 6. 1-Line by 16-Character Display Example

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	2 X 16 Character LCD															
For Shift Left	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	2 X 16 Character LCD															
For right Left	4F	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	2 X 16 Character LCD															

- 2-line Display (N = 1)

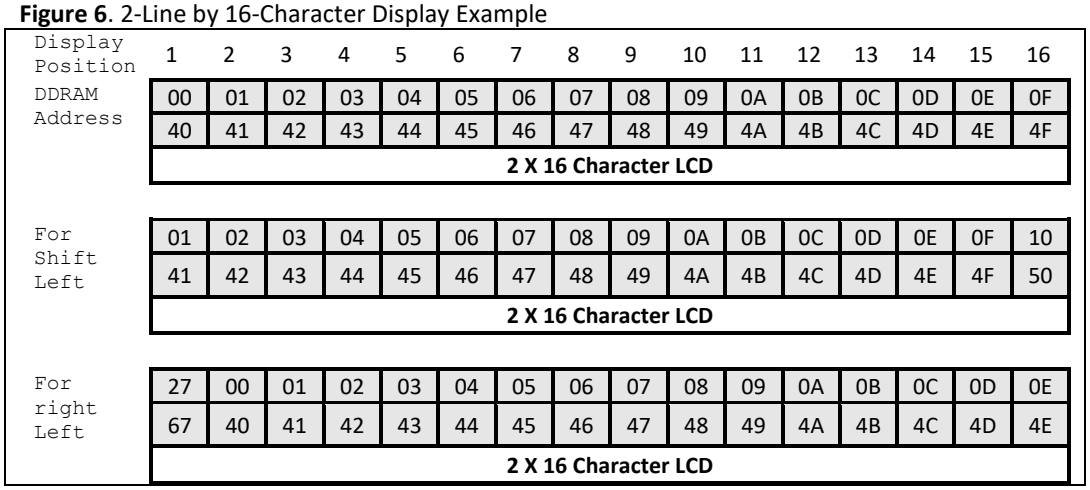
  In 2-line Display, there will be also 80 different address to store the display data but separated into two groups, each group consists of 40 consecutive addresses, and each group belongs to one line of the two lines of the LCD.

  So, the 40 address of the first line will start from the address 0X00 to the address 0X27 with total 40 address, and the 40 address of the second line of the LCD will start from the address 0X40 to the address 0X67 with total 40 address, and please Note that the first line end address and the second line start address are not consecutive, As the end address of the first line is 0X27 while the start address of the second line is 0X40.

Figure 2. 2-Line Display

Display Position	1	2	3	4	……	12	13	14	15	16	……..	38	39	40
DDRAM Address (Hexadecimal)	00	01	02	03	……	0B	0C	0D	0E	0F	……..	25	26	27
	40	41	42	43	……	4B	4C	4D	4E	4F	……..	65	66	67
	2 X 16 Character LCD													

Figure 6. 2-Line by 16-Character Display Example



Character Generator ROM (CGROM)

The character generator ROM contains the pixel mapping of each English letters plus special characters as well, as it can generate 5 x 8 dot or 5 x 11 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 11 dot character patterns.

Each 5 X 8 character pattern consists of 40 dots, each dot represent to one bit of CGROM memory, So, each 5 X 8 character pattern consists of 8 bytes to represent its pattern, so by using the character code which we store in the places of DDRAM memory we can display the character corresponding to that character code based on the ASCII table, and that happen as the character code is actually converted to represent the address of the first byte of that character pattern by multiplying the character code by 8, and by typing the address of the first byte of the 8 bytes of the character pattern, we actually tell the LCD driver that we want to display the pattern of all 8 bytes that their address starts with the address of the first byte which is the output of the character code we store in the places of DDRAM memory multiplied by 8. And when we increase the character code by 1, it actually increases the address by 8 addresses to go to the next character pattern and display it byte by byte.

Character Generator RAM (CGRAM)

In the character generator RAM, the user can rewrite character patterns by program as it can store the pixel mapping of any user-defined character we want, or the pixel mapping of any other language (e.g., Arabic). For 5 x 8 dots, eight-character patterns can be written, and for 5 x 11 dots, four-character patterns can be written.

Table 5. Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character patterns (CGRAM Data)

Character Code (DDRAM Data)								CGRAM Address						Character Patterns (CGRAM Data)							
B7	B6	B5	B4	B3	B2	B1	B0	B5	B4	B3	B2	B1	B0	B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	-	0	0	0	0	0	0	0	0	0	-	-	-	1	1	1	1	1
					0	0	0				0	0	1				0	0	1	0	0
					0	0	0				0	1	0				0	0	1	0	0
					0	0	0				0	1	1				0	0	1	0	0
					0	0	0				1	0	0				0	0	1	0	0
					0	0	0				1	0	1				0	0	1	0	0
					0	0	0				1	1	0				0	0	1	0	0
					0	0	0				1	1	1				0	0	0	0	0
0	0	0	0	-	0	0	1	0	0	1	0	0	0	-	-	-	1	1	1	1	0
					0	0	1				0	0	1				1	0	0	0	1
					0	0	1				0	1	0				1	0	0	0	1
					0	0	1				0	1	1				1	1	1	1	0
					0	0	1				1	0	0				1	0	1	0	0
					0	0	1				1	0	1				1	0	0	1	0
					0	0	1				1	1	0				1	0	0	0	1
					0	0	1				1	1	1				0	0	0	0	0

Notes:

1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor. Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display. If the 8th line data is 1, 1 bit will light up the 8th line regardless of the cursor presence.
3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

"-": Indicates no effect.

Table 4. Correspondence between Character Codes and Character Patterns

Upper 4bit / Lower 4bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)															
LLLH	(2)															
LLHL	(3)															
LLHH	(4)															
LHLL	(5)															
LHLH	(6)															
LHHL	(7)															
LHHH	(8)															
HLLL	(1)															
HLLH	(2)															
HLHL	(3)															
HLHH	(4)															
HHLL	(5)															
HHLH	(6)															
HHHL	(7)															
HHHH	(8)															

ST7066
Instructions

Instruction Table

Table 7. Instruction Table

Ins	Instruction Code										Description	Ins Time (270 Khz)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.52 ms
Return Home	0	0	0	0	0	0	0	0	1	x	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.52 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 us
Display ON/ OFF	0	0	0	0	0	0	1	D	C	B	D=1: entire display on C=1: cursor on B=1: cursor position on	37 us
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	x	x	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	37 us
Function Set	0	0	0	0	1	DL	N	F	x	x	DL: interface data is 8/4 bits NL: number of lines is 2/1 F: font size is 5x11/5x8	37 us
Set CGRAM address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter	37 us
Set DDRAM address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	37 us
Read Busy flag and address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 us
Write data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM)	43 us
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM)	43 us

Note:

Be sure the ST7066 is not in the busy state (BF = 0) before sending an instruction from the MPU to the ST7066.

If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Instruction Table for the list of each instruction execution time.

Instruction Description

Clear Display

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	0	0	0	0	0	0	0	1

Clear all the display data by writing "20H" (space code) to all DDRAM address, and set DDRAM address to "00H" into AC (address counter). Return cursor to the original status, namely, bring the cursor to the left edge on first line of the display. Make entry mode increment (I/D = "1").

And here are all the possible instruction codes of Clear Display Instruction in Hexadecimal

Table 32. Clear Display Instruction Code

Instruction	Instruction Macro	Instruction Bits								Instruction Hexadecimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Clear Display	`Clear_Display_Screen`	0	0	0	0	0	0	0	1	0X01

Return Home

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	0	0	0	0	0	0	1	x

Return Home is cursor return home instruction. Set DDRAM address to "00H" into the address counter. Return cursor to its original site and return display to its original status, if shifted. Contents of DDRAM does not change.

Table 32. Return Home Instruction Code

Instruction	Instruction Macro	Instruction Bits								Instruction Hexadecimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Return Home	`Return_Home`	0	0	0	0	0	0	1	x	0X02

Entry Mode Set

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	0	0	0	0	0	1	I/D	S

Set the moving direction of cursor and display.

I/D: Increment / decrement of DDRAM address (cursor or blink)
When I/D = "High", cursor/blink moves to right and DDRAM address is increased by 1.
When I/D = "Low", cursor/blink moves to left and DDRAM address is decreased by 1.
* CGRAM operates the same as DDRAM, when read from or write to CGRAM.

S: Shift of entire display
When DDRAM read (CGRAM read/write) operation or S = "Low", shift of entire display is not performed. If S = "High" and DDRAM write operation, shift of entire display is performed according to I/D value (I/D "1": shift left, I/D = "0": shift right).

Table 12. Entry Mode Set

S	I/D	Description
L	L	cursor moves to left and DDRAM address is decreased by 1.
L	H	cursor moves to right and DDRAM address is increased by 1.
H	L	Shift the display to the right per Write or read on DDRAM
H	H	Shift the display to the left per Write or read on DDRAM

Table 32. Entry Mode Set Instruction Codes

Instruction	Instruction Macro	Instruction Bits								Instruction Hexadecimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
		0	0	0	0	0	1	I/D	S	
Entry Mode Set	Shift_Cursor_Left_Per_Read_Write	0	0	0	0	0	1	0	0	0X04
	Shift_Cursor_Right_Per_Read_Write	0	0	0	0	0	1	1	0	0X06
	Shift_Display_Right_Per_Read_Write	0	0	0	0	0	1	0	1	0X05
	Shift_Display_Left_Per_Read_Write	0	0	0	0	0	1	1	1	0X07

Display ON/OFF

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	0	0	0	0	1	D	C	B

Control display/cursor/blink ON/OFF 1 bit register.

D: Display ON/OFF control bit
When D = "High", entire display is turned on.
When D = "Low", display is turned off, but display data is remained in DDRAM.

C: Cursor ON/OFF control bit
When C = "High", cursor is turned on.
When C = "Low", cursor is disappeared in current display, but I/D register remains its data.

B: Cursor Blink ON/OFF control bit
When B = "High", cursor blink is on, that performs alternate between all the high data and display character at the cursor position.
When B = "Low", blink is off.

Table 32. Display ON/OFF Instruction Codes

Instruction	Instruction Macro	Instruction Bits								Instruction Hexadecimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
		0	0	0	0	1	D	C	B	
Display ON/OFF	Display_Cursor_Blink_ALL_OFF	0	0	0	0	1	0	0	0	0X08
	Display_Cursor_OFF_Blink_ON	0	0	0	0	1	0	0	1	0X09
	Display_Blink_OFF_Cursor_ON	0	0	0	0	1	0	1	0	0X0A
	Display_OFF_Cursor_Blink_ON	0	0	0	0	1	0	1	1	0X0B
	Display_ON_Cursor_Blink_OFF	0	0	0	0	1	1	0	0	0X0C
	Display_Blink_ON_Cursor_OFF	0	0	0	0	1	1	0	1	0X0D
	Display_Cursor_ON_Blink_OFF	0	0	0	0	1	1	1	0	0X0E
	Display_Cursor_Blink_ALL_ON	0	0	0	0	1	1	1	1	0X0F

Cursor or Display Shift

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	0	0	0	1	S/C	R/L	x	x

Without writing or reading of display data, shift right/left cursor position or display. This instruction is used to correct or search display data. During 2-line mode display, cursor moves to the 2nd line after 40th digit of 1st line. Note that display shift is performed simultaneously in all the line. When displayed data is shifted repeatedly, each line shifted individually. When display shift is performed, the contents of address counter are not changed.

Table 12. Cursor or Display Shift

S/C	R/L	Description	AC Value
L	L	Shift cursor to the left	AC = AC – 1
L	H	Shift cursor to the right	AC = AC + 1
H	L	Shift display to the left. Cursor follows the display shift	AC = AC
H	H	Shift display to the right. Cursor follows the display shift	AC = AC

Table 32. Cursor or Display shift Instruction Codes

Instruction	Instruction Macro	Instruction Bits								Instruction Hexadecimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
		0	0	0	1	S/C	R/L	x	x	
Cursor or Display Shift	Shift_Cursor_Position_Left	0	0	0	1	0	0	0	0	0X10
	Shift_Cursor_Position_Right	0	0	0	1	0	1	0	0	0X14
	Shift_Entire_Display_Left	0	0	0	1	1	0	0	0	0X18
	Shift_Entire_Display_Right	0	0	0	1	1	1	0	0	0X1C

Function Set

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	0	0	1	DL	N	F	x	x

Control display/cursor/blink ON/OFF 1 bit register.

DL: Interface data length control bit
When DL = "High", it means 8-bit bus mode with MPU.
When DL = "Low", it means 4-bit bus mode with MPU. So, to speak, DL is a signal to select 8-bit or 4-bit bus mode.
When 4-bit bus mode, it needs to transfer 4-bit data by two times.

N: Display line number control bit
When N = "Low", it means 1-line display mode.
When N = "High", 2-line display mode is set.

F: Display font type control bit
When F = "Low", it means 5 x 8 dots format display mode
When F = "High", 5 x11 dots format display mode.

Table 12. Function Set

N	F	No. of display lines	Character Font
L	L	1	5 X 8
L	H	1	5 X 11
H	x	2	5 X 8

Table 32. Function Set Instruction Codes

Instruction	Instruction Macro	Instruction Bits								Instruction Hexadecimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
		0	0	1	DL	N	F	x	x	
Function Set	LCD_4_Bit_Mode_1_Line_5_8_Dots	0	0	1	0	0	0	0	0	0X20
	LCD_4_Bit_Mode_1_Line_5_11_Dots	0	0	1	0	0	1	0	0	0X24
	LCD_4_Bit_Mode_2_Line_5_8_Dots	0	0	1	0	1	0	0	0	0X28
	LCD_4_Bit_Mode_2_Line_5_11_Dots	0	0	1	0	1	1	0	0	0X2C
	LCD_8_Bit_Mode_1_Line_5_8_Dots	0	0	1	1	0	0	0	0	0X30
	LCD_8_Bit_Mode_1_Line_5_11_Dots	0	0	1	1	0	1	0	0	0X34
	LCD_8_Bit_Mode_2_Line_5_8_Dots	0	0	1	1	1	0	0	0	0X38
	LCD_8_Bit_Mode_2_Line_5_11_Dots	0	0	1	1	1	1	0	0	0X3C

Set CGRAM Address

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Set CGRAM address to AC.
This instruction makes CGRAM data available from MPU.

Table 32. Set CGRAM Address Instruction Codes

Instruction	Instruction Macro	Instruction Bits								Instruction Hex decimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
		0	1	AC5	AC4	AC3	AC2	AC1	AC0	
Set CGRAM Address	CGRAM_First_Address_First_Character	0	1	0	0	0	0	0	0	0X40
	CGRAM_Second_Address_First_Character	0	1	0	0	0	0	0	1	0X41
	CGRAM_Third_Address_First_Character	0	1	0	0	0	0	1	0	0X42
	CGRAM_First_Address_Second_Character	0	1	0	0	1	0	0	0	0X48
	CGRAM_First_Address_Third_Character	0	1	0	1	0	0	0	0	0X50
	CGRAM_First_Address_Fourth_Character	0	1	0	1	1	0	0	0	0X58
	CGRAM_First_Address_Fifth_Character	0	1	1	0	0	0	0	0	0X60
	CGRAM_First_Address_Sixth_Character	0	1	1	0	1	0	0	0	0X68
	CGRAM_First_Address_Seventh_Character	0	1	1	1	0	0	0	0	0X70
	CGRAM_First_Address_Eighth_Character	0	1	1	1	1	0	0	0	0X78

Set DDRAM Address

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Set DDRAM address to AC.
This instruction makes DDRAM data available from MPU.
When 1-line display mode (N = 0), DDRAM address is from "00H" to "4FH".
In 2-line display mode (N = 1), DDRAM address in the 1st line is from "00H" to "27H", and DDRAM address in the 2nd line is from "40H" to "67H".

Table 32. Set DDRAM Address Instruction Codes

Instruction	Instruction Macro	Instruction Bits								Instruction Hexadecimal
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
		1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	
Set DDRAM Address	DDRAM_Cursor_Begin_1st_Line	1	0	0	0	0	0	0	0	0X80
	DDRAM_Cursor_Begin_2st_Line	1	1	0	0	0	0	0	0	0XC0

Read Busy Flag and Address

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

When BF = “High”, indicates that the internal operation is being processed. So, during this time the next instruction cannot be accepted.

The address Counter (AC) stores DDRAM/CGRAM addresses, transferred from IR.

After writing into (reading from) DDRAM/CGRAM, AC is automatically increased (decreased) by 1.

Write Data to CGRAM or DDRAM

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write binary 8-bit data to DDRAM/CGRAM.

The selection of RAM from DDRAM, CGRAM, is set by the previous address set instruction: DDRAM address set, CGRAM address set. RAM set instruction can also determine the AC direction to RAM.

After write operation, the address is automatically increased/decreased by 1, according to the entry mode.

Read Data from CGRAM or DDRAM

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Code	1	1	D7	D6	D5	D4	D3	D2	D1	D0

Read binary 8-bit data from DDRAM/CGRAM.

The selection of RAM is set by the previous address set instruction. If address set instruction of RAM is not performed before this instruction, the data that read first is invalid, because the direction of AC is not determined.

If you read RAM data several times without RAM address set instruction before read operation, you can get correct RAM data from the second, but the first data would be incorrect, because there is no time margin to transfer RAM data.

In case of DDRAM read operation, cursor shift instruction plays the same role as DDRAM address set instruction: it also transfer RAM data to output data register. After read operation address counter is automatically increased/decreased by 1 according to the entry mode. After CGRAM read operation, display shift may not be executed correctly.

* In case of RAM write operation, after this AC is increased/decreased by 1 like read operation. In this time, AC indicates the next address position, but you can read only the previous data by read instruction.