ECE 372A Spring 2015 - Lecture 6

Garrett Vanhoy

February 3, 2015

Outline

Timers

Reference Material

Section 14 in the PIC24F Family Reference Manual
Section 11 and 12 in the PIC24F Data Sheet

Types of Timers

Applications of Each Timer Type

- Type A timers can be operated from external oscillators and asynchronously from the system clock. It can also be used with the 32 kHz clock for low-power applications.

Types of Timers

Applications of Each Timer Type

- Type A timers can be operated from external oscillators and asynchronously from the system clock. It can also be used with the 32 kHz clock for low-power applications.
- Type B timers can be used with Type C timers to form 32-bit timers. It is can be used with relatively high clock rates.

Types of Timers

Applications of Each Timer Type

- Type A timers can be operated from external oscillators and asynchronously from the system clock. It can also be used with the 32 kHz clock for low-power applications.
- Type B timers can be used with Type C timers to form 32-bit timers. It is can be used with relatively high clock rates.
- Type C can be used to throttle Analog-to-Digital Conversion (talked about later.)

Types of Timers

Applications of Each Timer Type

- Type A timers can be operated from external oscillators and asynchronously from the system clock. It can also be used with the 32 kHz clock for low-power applications.
- Type B timers can be used with Type C timers to form 32-bit timers. It is can be used with relatively high clock rates.
- Type C can be used to throttle Analog-to-Digital Conversion (talked about later.)
- Each of them have slightly different control register structures.

Making a 32-bit timer

Considerations

- Timer 2 and Timer 3 can be combined. Timer 4 and Timer 5 can be combined. No other combination works.

Making a 32-bit timer

Considerations

- Timer 2 and Timer 3 can be combined. Timer 4 and Timer 5 can be combined. No other combination works.
- TMR2/TMR4 holds the least significant digits. TMR3/TMR5 holds the most significant digits.

Making a 32-bit timer

Considerations

- Timer 2 and Timer 3 can be combined. Timer 4 and Timer 5 can be combined. No other combination works.
- TMR2/TMR4 holds the least significant digits. TMR3/TMR5 holds the most significant digits.
- 32-bit mode is enabled in Timer 2 and Timer 4.

Making a 32-bit timer

Considerations

- The interrupt for the 32-bit timer is in Timer 3/Timer 5. Interrupts have to be enabled here.
- Enabling the interrupt for Timer 2 and Timer 4 *do nothing*. You cannot use the 16-bit part of a 32-bit timer.

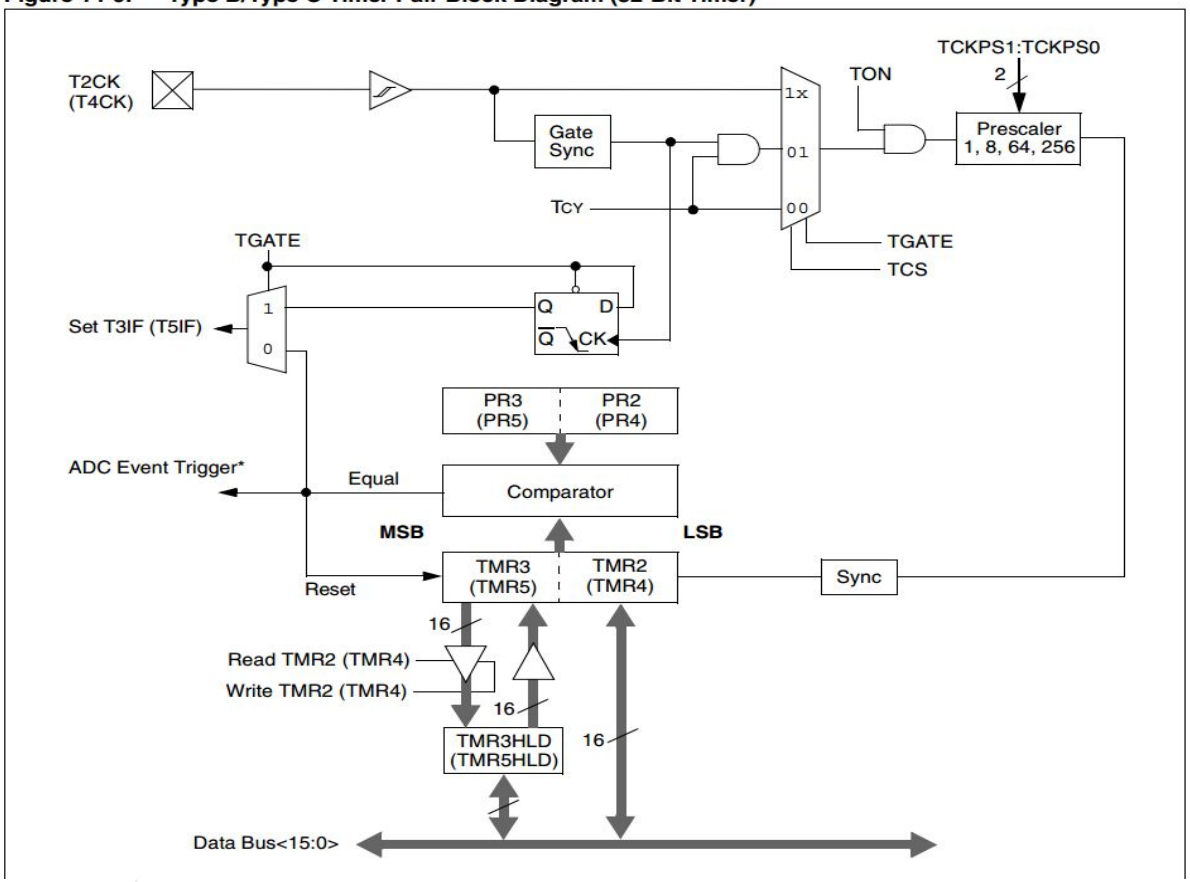
Making a 32-bit timer

Considerations

- The interrupt for the 32-bit timer is in Timer 3/Timer 5. Interrupts have to be enabled here.
- Enabling the interrupt for Timer 2 and Timer 4 *do nothing*. You cannot use the 16-bit part of a 32-bit timer.
- All configuration bits are derived from the Timer 2 and Timer 4 control registers (T2CON, T4CON).
- T3CON and T5CON are "don't care" bits.

Making a 32-bit timer

Figure 14-6: Type B/Type C Timer Pair Block Diagram (32-Bit Timer)



Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.

Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.
- $PR = 5 * (14745600) - 1 = 73727999$

Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.
- $PR = 5 * (14745600) - 1 = 73727999$
- In *binary*: 0b00000100011001001111111111111111

Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.
- $PR = 5 * (14745600) - 1 = 73727999$
- In *binary*: 0b00000100011001001111111111111111
- Now, PR3/PR5 are the most significant bits and PR2/PR4 are the least significant bits

Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.
- $PR = 5 * (14745600) - 1 = 73727999$
- In *binary*: 0b00000100011001001111111111111111

Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.
- $PR = 5 * (14745600) - 1 = 73727999$
- In *binary*: 0b00000100011001001111111111111111
- $PR2/PR45 = 0b1111111111111111 = 65535$

Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.
- $PR = 5 * (14745600) - 1 = 73727999$
- In *binary*: 0b**00000100011001001**11111111111111

Making a 32-bit timer

Example

- Make a timer that is 5 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 1.
- $PR = 5 * (14745600) - 1 = 73727999$
- In *binary*: 0b**00000100011001001**11111111111111
- $PR3/PR5 = 0b00000100011001001 = 1124$

Making a 32-bit timer

Example 2

- Make a timer that is 20 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 64.

Making a 32-bit timer

Example 2

- Make a timer that is 20 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 64.
- $PR = \frac{20*(14745600)}{64} - 1 = 4607999$

Making a 32-bit timer

Example 2

- Make a timer that is 20 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 64.
- $PR = \frac{20*(14745600)}{64} - 1 = 4607999$
- In *binary*: 0b10001100100111111111111

Making a 32-bit timer

Example 2

- Make a timer that is 20 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 64.
- $PR = \frac{20*(14745600)}{64} - 1 = 4607999$
- In *binary*: 0b10001100100111111111111
- $PR2/PR4 = 0b01001111111111111 = 20479$

Making a 32-bit timer

Example 2

- Make a timer that is 20 seconds long using a 32-bit timer using the development board oscillator and a prescalar of 64.
- $PR = \frac{20*(14745600)}{64} - 1 = 4607999$
- In *binary*: 0b10001100100111111111111
- $PR2/PR4 = 0b0100111111111111 = 20479$
- $PR3/PR5 = 0b1000110 = 70$

Making a 32-bit timer

Implementation

- There is an easier way to calculate PR for 32 bit timers using bit-shifting. This will be shown later in this lecture.

Outline

LCD Reference

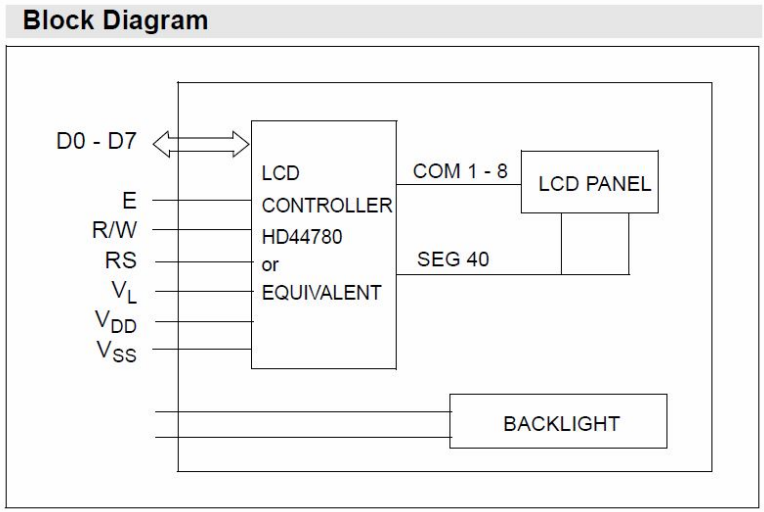
Reference Material

LCD Data Sheet by Mouser
Also online on D2L

LCD Hardware Interface

The Interface

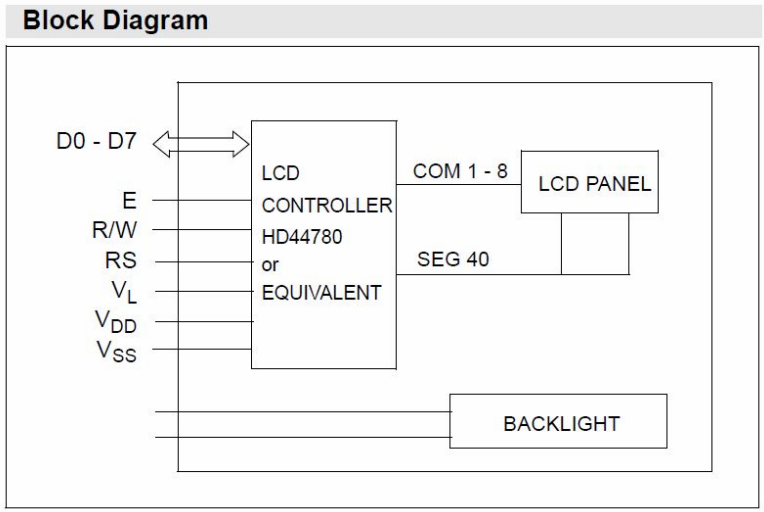
- 16 pins, 14 are usable. 8 data pins, the rest are control related.



LCD Hardware Interface

The Interface

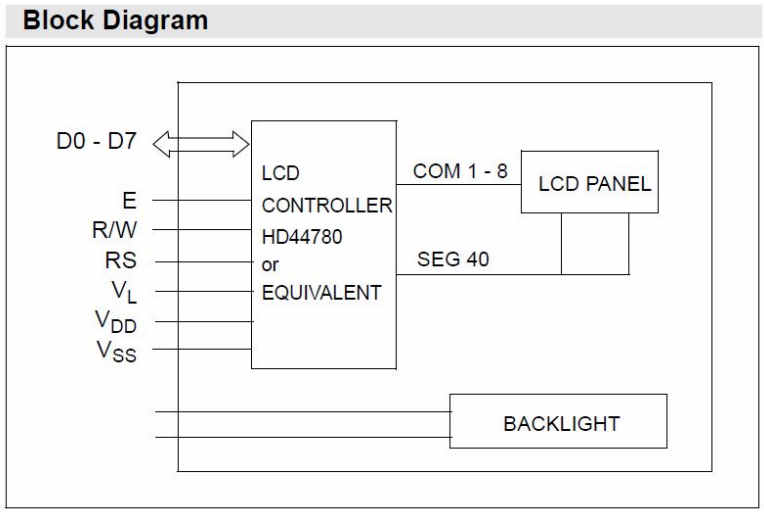
- 16 pins, 14 are usable. 8 data pins, the rest are control related.
- The LCD given to you has no header attached. You will have to solder on your own header to the LCD and make a connector for it.



LCD Hardware Interface

The Interface

- 16 pins, 14 are usable. 8 data pins, the rest are control related.
- The LCD given to you has no header attached. You will have to solder on your own header to the LCD and make a connector for it.
- Test the LCD once you have gotten the headers on. Only three pins are necessary to test this (V_{DD}, V_{SS}, V_L).



LCD Software Interface

Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	RE	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0	0	0	1	A_{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs
Set DD RAM Address	0	0	1	A_{DD}							Sets the DD RAM address. Data may be written or read after making this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line RE = 1: Ext. Reg. Ena. F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*										DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

LCD Software Interface

Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents	40μs~1.64ms
Entry Mode Set	0	0									…on …play.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	RE	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0	0	0	1	A_{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs
Set DD RAM Address	0	0	1	A_{DD}							Sets the DD RAM address. Data may be written or read after making this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line RE = 1: Ext. Reg. Ena. F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*										DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

Every command is specified in terms of data/control bits.

LCD Software Interface

Command	Code										Description	Execution Time
	RS	R/										
Clear Display	0	0									urns sition (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	RE	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0										Sets the CG RAM address. CG altered	40μs
Set DD RAM Address	0	0									. Data r mak-	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment S = 1: Accompanies displ shift S/C= 1: Display shift R/L= 1: Shift to the right. DL = 1: 8 bits N = 1: 2 lines RE = 1: Ext. Reg. Ena. BF = 1: Busy *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*						I/D = 0: Decrement S/C = 0: cursor move R/L= 0: Shift to the left. DL = 0: 4 bits N = 0: 1 line F = 0: 5 x 7 dots BF = 0: Can accept data				DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

(*) Means ``don't care''

S/C is defined below. Others are as well.

LCD Software Interface

Command	Code										Description	Execution Time
	RS	R/										
Clear Display	0	0									urns sition (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	RE	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0										Sets the CG RAM address. CG altered	40μs
Set DD RAM Address	0	0									. Data r mak-	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment S = 1: Accompanies displ shift S/C= 1: Display shift R/L= 1: Shift to the right. DL = 1: 8 bits N = 1: 2 lines RE = 1: Ext. Reg. Ena. BF = 1: Busy *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*						I/D = 0: Decrement S/C = 0: cursor move R/L= 0: Shift to the left. DL = 0: 4 bits N = 0: 1 line F = 0: 5 x 7 dots BF = 0: Can accept data				DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

(*) Means ``don't care''

S/C is defined below. Others are as well.

LCD Software Interface

Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
											ves the cursor and shifts the play without changing the DD M contents.	40μs
											s the data width (DL), the nber of lines in the display (L), the character font (F).	40μs
											s the CG RAM address. CG M data can be read or altered r making this setting.	40μs
											s the DD RAM address. Data be written or read after mak- this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							eads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line RE = 1: Ext. Reg. Ena. F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*											DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

Every command has a *minimum* execution time. You have to wait at least this long. Waiting ***too*** long will also get you into trouble. However, this length is not specified.

LCD Software Interface

Moving the Cursor Example

- The cursor location depends on an address held in a register.

2) 8 x 2: HDM08216H-1, HDM08216H-3, HDM08216L-3

	1	2	3	4	5	6	7	8	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	

LCD Software Interface

Moving the Cursor Example

- The cursor location depends on an address held in a register.
- This can be changed by writing to the DD RAM.

2) 8 x 2: HDM08216H-1, HDM08216H-3, HDM08216L-3

	1	2	3	4	5	6	7	8	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	

LCD Software Interface

Moving the Cursor Example

- The cursor location depends on an address held in a register.
- This can be changed by writing to the DD RAM.
- Each location is expressed in **hexadecimal**.

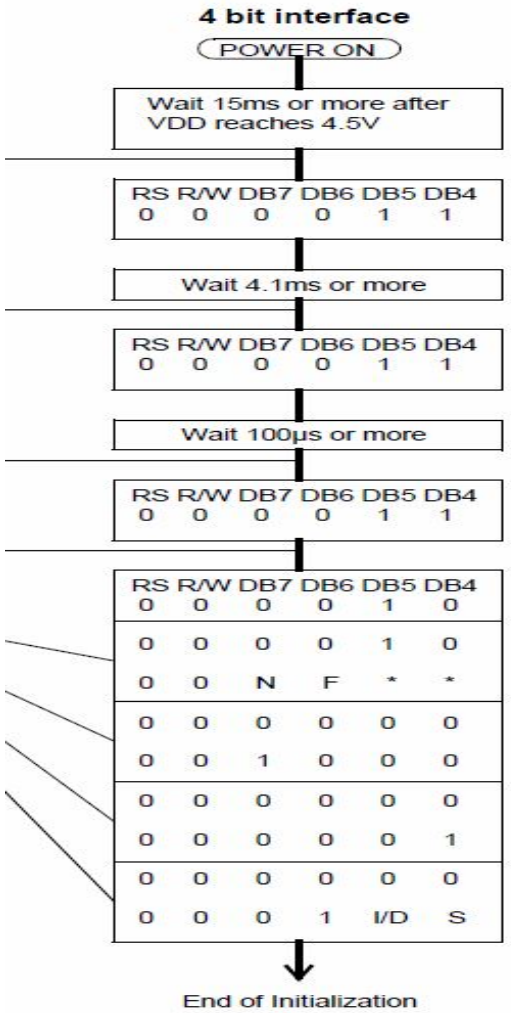
2) 8 x 2: HDM08216H-1, HDM08216H-3, HDM08216L-3

	1	2	3	4	5	6	7	8	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	

LCD Initialization

Notes:

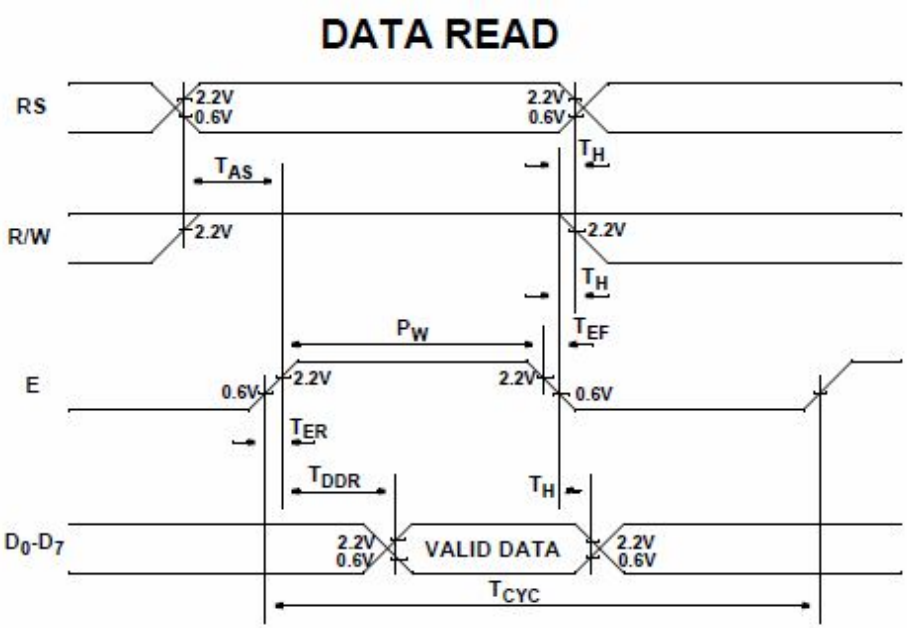
- This sequence has to be followed *precisely* to begin using the LCD.



LCD Timing Diagram

TIMING CHARACTERISTICS

ITEM	SYMBOL	MAX.	MIN.	UNIT
ENABLE CYCLE TIME	T_{CYC}		500	nS
ENABLE PULSE WIDTH	P_W		230	nS
ENABLE RISE/FALL TIME	T_{ER}, T_{EF}	20		nS
RS, R/W SET UP TIME	T_{AS}		40	nS
DATA DELAY TIME	T_{DDR}	360		nS
DATA SETUP TIME	T_{DSW}		60	nS
HOLD TIME	T_H		10	nS



LCD Software Interface

Timeline:

1. Set data bits first. When enable is set, these will be accepted. Do not wait until the enable is set to change the bits.

LCD Software Interface

Timeline:

1. Set data bits first. When enable is set, these will be accepted. Do not wait until the enable is set to change the bits.
2. Set RS and R/W. Wait 40 ns.

LCD Software Interface

Timeline:

1. Set data bits first. When enable is set, these will be accepted. Do not wait until the enable is set to change the bits.
2. Set RS and R/W. Wait 40 ns.
3. Set Enable bit to 1. Wait at least 230 ns. 300 ns is safe.

LCD Software Interface

Timeline:

1. Set data bits first. When enable is set, these will be accepted. Do not wait until the enable is set to change the bits.
2. Set RS and R/W. Wait 40 ns.
3. Set Enable bit to 1. Wait at least 230 ns. 300 ns is safe.
4. Set Enable bit to 0.

LCD Software Interface

Timeline:

1. Set data bits first. When enable is set, these will be accepted. Do not wait until the enable is set to change the bits.
2. Set RS and R/W. Wait 40 ns.
3. Set Enable bit to 1. Wait at least 230 ns. 300 ns is safe.
4. Set Enable bit to 0.
5. Repeat...

Demonstration

Pulse Formation

1. Writing a "read-data" and "write-data" function for the LCD
2. Creating a delay function
3. Writing a character to the LCD screen