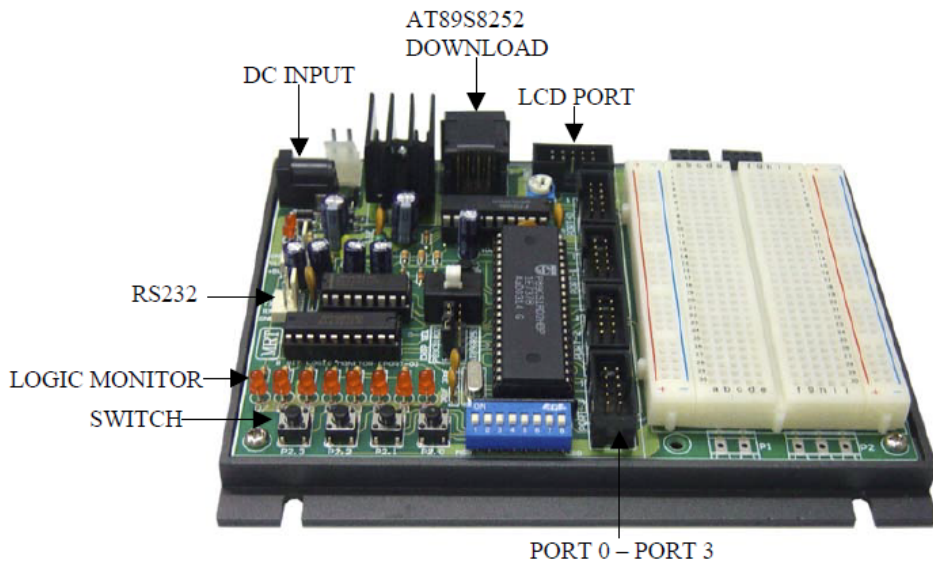


# 14-pin LCD

Vcc Gnd Vee RS RW EN DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7



PIN	NEME	Function connected with CPU
1	GND	GND
2	VCC	+VCC
3	VEE	VR Adjust the LCD light LCD
4	RS	P1.0 Instruction/Data Select
5	RW	P1.1 Read/Write Data
6	E	P1.2 Enable
7	D4	P1.4 Data Bit 4
8	D5	P1.5 Data Bit 5
9	D6	P1.6 Data Bit 6
10	D7	P1.7 Data Bit 7

# Control lines

Line	Function
EN	Enable line  Tells the LCD when it is being sent data or a command. Set EN=1, set other lines and input bus, then set EN=0, 1-0 transition causes LCD to read.
RS	Register select  RS = 0, data is a command RS = 1, data is text for display
RW	Read/write  RW = 0, information on bus is for writing to LCD RW = 1, for querying the LCD, e.g. checking if LCD busy

# Data bus

- LCD can be set to either 4-bit or 8-bit mode
  - 8-bit easier, but requires more wires
    - Hook up to DB0-7
    - Total of 11 data/control lines, +3 power/ground
  - 4-bit, must send command/text a nibble at a time
    - Hook up to DB4-7
    - Total of 7 data/control lines, +3 power/ground
- Command or output text
  - Sent by placing 8-bit char value on data bus

# Checking busy status

- Instructions take LCD time to process
  - LCD signals it is done by lowering level on DB7
  - Make a function that will be used by other LCD functions:
    - Specify a command, RS = 0
    - Specify we want to query LCD, RW = 1
    - Mark start of command, EN = 1
    - Set all pins on data bus to 1
    - Repeat process until DB7 is 0
    - Finish the command, EN = 0
    - Specify future commands will write to LCD, RW = 0

# Issuing a command

- LCD accepts a variety of commands
  - Create a function that issues a command
  - Command is a byte on the input bus
  - Procedure (8-bit mode):
    - Set RS = 0 to indicate a command
    - Set RW = 0 to indicate a write
    - Move command onto data bus
    - Set EN = 1 to signal start of command
    - Wait 4 cycles
    - Set EN = 0 to mark end of command
    - Wait while LCD is busy

# Commands

- Initializing the LCD, issue three commands:
  - 0x38 = 8-bit bus, 5x8 character font
  - 0x0C = turn on, with no cursor
  - 0x06 = turn on cursor auto-advance
- Clearing screen
  - Issue command 0x01
- Changing cursor position
  - Issue command, 0x80 + desired location
  - Only the blue spots are visible

Display	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16						
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	...
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	...

# Writing to LCD

- Writing a single character
  - Will be placed at current cursor position
  - Procedure (8-bit mode):
    - Set RS = 1 to indicate text data instead of command
    - Set RW = 0 to indicate write operation
    - Move 8-bit char value to data bus
    - Set EN = 1 to mark start of command
    - Wait 4 cycles
    - Set EN = 0 to mark end of command
    - Wait while LCD busy