www.JetPackAcademy.com

TM1638 quick reference sheet

This quick reference sheet was provided as part of the "Robotics: Learn by building" series of online courses at www.JetPackAcademy.com You are granted permission to reproduce this paper as long as it is reproduced in its entirety and not for profit.

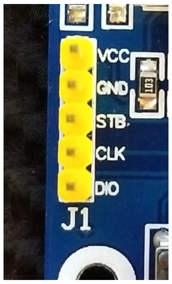
These instructions should also be applicable to the TM1637 and TM1639 LED driver chips

Board Pinouts:

Vcc: +5V or +3.3V. 3.3V should work on boards with red or green LED's

GND: Ground, 0 volts

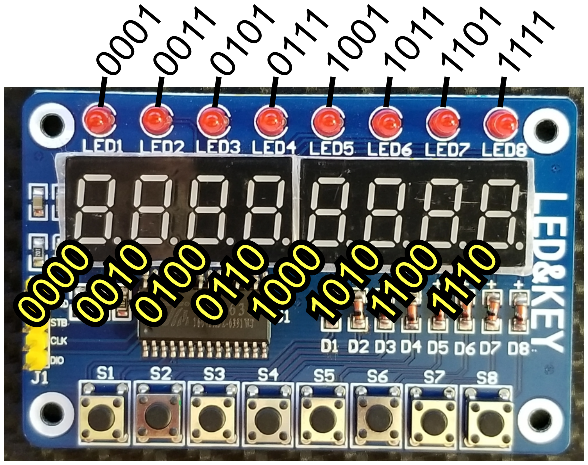
DIO: Digital Input/Output. Bi-directional data line, receiving data for the displays & LED's or sending data read from the keys. Data direction is set by instructions. **NOTE:** when reading data from the DIO pin, the datasheet recommends using a pullup resistor of 10k on the DIO line. If using a microcontroller such as PIC or Arduino, just simply use its internal pullup resistors (i.e. the INPUT_PULLUP instruction with Arduino).



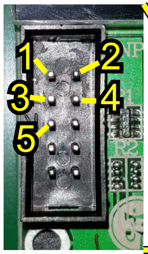
CLK: Clock. As the next bit of data is held on the DIO line, the CLK line is pulsed to commit that data bit to the internal shift registers. Or if data is being output on DIO, a CLK pulse causes the data line to be loaded with the next data bit from the keypad shift registers.

STB or STRB: Strobe. When strobe goes low, the controller chip starts listening to the DIO line, expecting an instruction or data. Because an instruction can be followed by multiple bytes of data, strobe is set high upon completion of all instruction and data transmissions. **NOTE:** Some boards (like the bicolour LED board shown below) have multiple strobe lines. Simply use STRB0 for your strobe line.

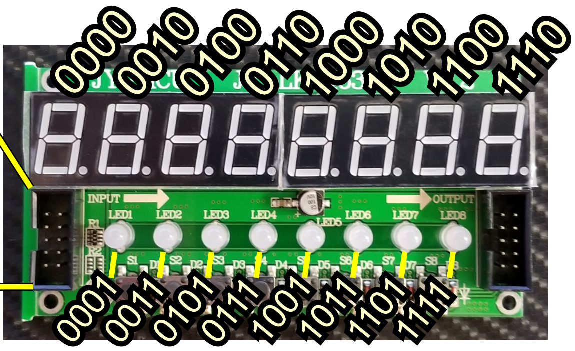
7 segment display & LED addressing:



Addresses shown are the lower nibble required for display instructions. The upper nibble sets the addressing mode, the lower nibble dictates where the data byte should go.

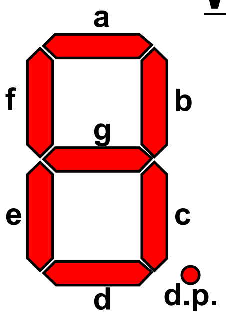


Pins:
1: Vcc
2: GND
3: CLK
4: DIO
5: STRB0



On boards with bi-coloured LED's, the addressing is exactly the same, though the board layout may be different.

Writing to the 7 segment displays

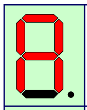
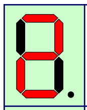
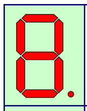
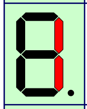
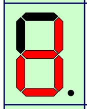
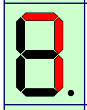
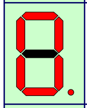
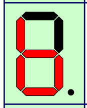
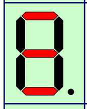
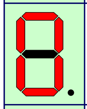
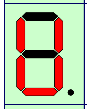
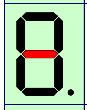
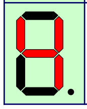
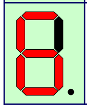
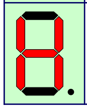


7 segment display control byte

d.p.	g	f	e	d	c	b	a

To illuminate a particular segment in the display, simply send a 1 to the segment's corresponding bit in the display's control byte.

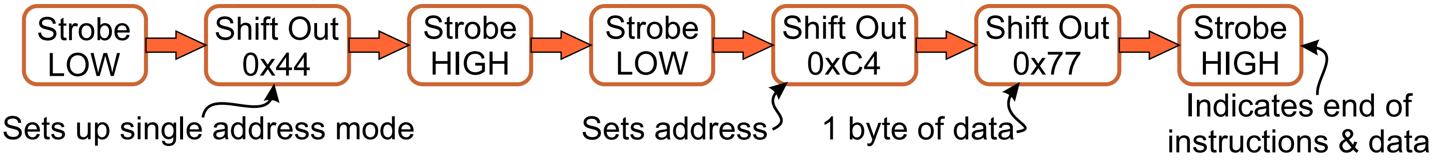
Example 7 segment characters:

	0111 0111 0x77		0101 1011 0x5B		1111 1111 0xFF
	0000 0110 0x06		0101 1110 0x5E		0000 0111 0x07
	1011 1111 0xBF		0111 1100 0x7C		0100 1001 0x49
	0011 1111 0x3F		0011 1110 0x3E		0100 0000 0x40
	0110 0110 0x66		0111 1101 0x7D		0111 0110 0x76

Sending instructions and data

Instruction	Details on its use
0x8? 1000 obbb	Turn display on or off (bit o) and set brightness (bbb bits) **Example:** To turn on display and set full brightness would be 1000 1111 or
0x44 0100 0100	Sets up single address mode to write to a single address.
0x40 0100 0000	Sets up sequential address mode.
0xC? 1100 aaaa	This instruction follows one of the addressing commands and sets the address (lower four bits, aaaa) that you wish to write to. **Example:** Sending 0x57 to address 0x02 in single address mode, you send the instruction 0x44, then 0xC2 (0xC combined with the address of 2 as the lower nibble), and then followed up by the data byte 0x57. See examples below.
0x42 0100 0010	Read the buttons. Once the instruction is received, the controller will set the DIO line as an output and will send 4 bytes of data. See page 3 for details on using this instruction and retrieving the data.

Example 1: To put a capital A (0x77) on the 3rd 7 segment display (address 0100, or 4), in single address mode (0x44), perform the following steps:



Strobe LOW → Shift Out 0x44 → Strobe HIGH → Strobe LOW → Shift Out 0xC4 → Shift Out 0x77 → Strobe HIGH

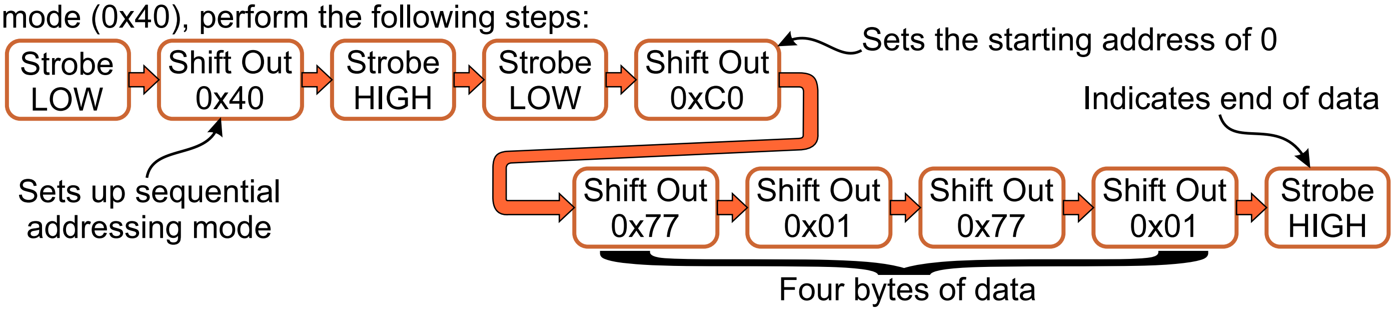
Sets up single address mode

Sets address

1 byte of data

Indicates end of instructions & data

Example 2: To put a capital A (0x77) on the 1st and 2nd 7 segment display (hex address 0 and 2), *and* turn on discrete LED's 1 & 2 (setting bit 0 at addresses 1 and 3) in sequential address mode (0x40), perform the following steps:



Strobe LOW → Shift Out 0x40 → Strobe HIGH → Strobe LOW → Shift Out 0xC0 → Shift Out 0x77 → Shift Out 0x01 → Shift Out 0x77 → Shift Out 0x01 → Strobe HIGH

Sets the starting address of 0

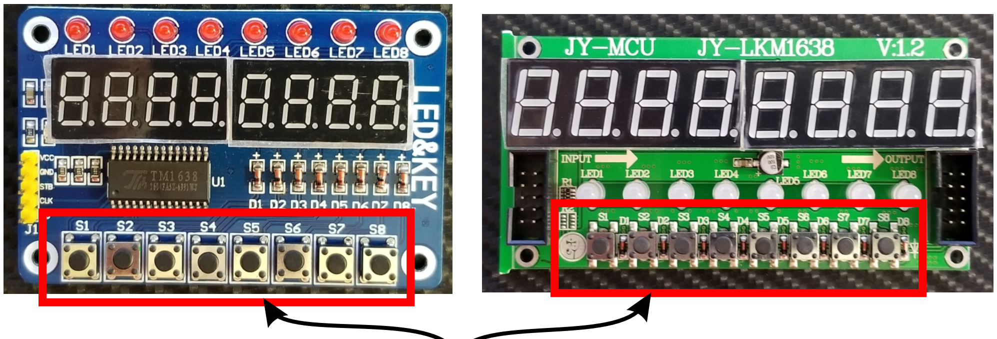
Indicates end of data

Sets up sequential addressing mode

Four bytes of data

Reading the pushbuttons

The TM1638 controller chip can multiplex up to 16 pushbuttons in four rows. How many pushbuttons there are on a particular circuit board, and how the TM1638 is wired to those pushbuttons is decided by the board manufacturer. The controller chip reads four rows of pushbuttons and thus returns four bytes of data over the DIO line. In the case of the two boards shown, the pushbuttons (marked S1 through S8 on both boards) return values on the individual bytes as show in the chart below:



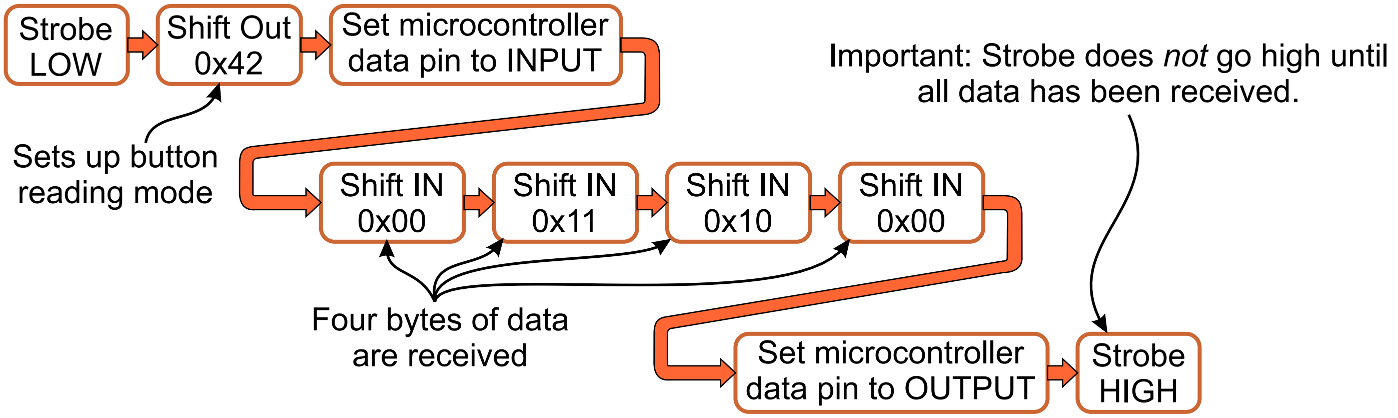
Pushbuttons

bit	7	6	5	4	3	2	1	0
Byte 0	X	X	X	S5	X	X	X	S1
Byte 1	X	X	X	S6	X	X	X	S2
Byte 2	X	X	X	S7	X	X	X	S3
Byte 3	X	X	X	S8	X	X	X	S4

X= null return. These bits will always return a 0 because they're not hooked up to anything

Examples: If the user is pressing button S7 only, byte 2 will return decimal 16 (0x10), all other bytes will return 0. If user is pressing buttons S2 and S6, byte 2 will return decimal 17 (0x11), all other bytes will return 0. If user is pressing buttons S8 and S3, byte3 will return decimal 16 (0x10) and byte 2 will return decimal 1 (0x01), all other bytes will return 0. Pressing S1 will return decimal 1 (0x01) on byte 0, all other bytes will read 0.

Reading the pushbutton registers: After sending the instruction to read the pushbutton registers, the controller will switch the DIO line to output and transmit 4 bytes of data in the usual shift out fashion. The microcontroller needs to switch data direction on the pin it has connected to the board's DIO line after sending the instruction. Perform the following steps:



Strobe LOW

Shift Out 0x42

Set microcontroller data pin to INPUT

Sets up button reading mode

Shift IN 0x00

Shift IN 0x11

Shift IN 0x10

Shift IN 0x00

Four bytes of data are received

Set microcontroller data pin to OUTPUT

Strobe HIGH

Important: Strobe does *not* go high until all data has been received.