Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 4 (2012) 461 – 465

Procedia Technology

C3IT-2012

Effective Utilization of a Microcontroller Port for Optimisation of Hardware

Mr. Sandeep Hanumante[a] and Prof. Kirti Sawlani[b]

[a]Student, ME (Electronics), Department of Electronics Engineering, K. J. Somaiya College of Engineering, Mumbai-400077, India.
[b]Faculty, Department of Electronics Engineering, K. J. Somaiya College of Engineering, Mumbai-400077, India.

Abstract

In embedded systems microcontrollers play a very important role because modern day's microcontrollers have in-built program as well as data memory of enough length to implement any embedded application without need of any external memory. But it has limited number of IO port pins to interface IO peripherals and better utilization of these IO port pins becomes extremely challenging.

Key words: Micro-controller; IO pins; LCD; Matrix keyboard

1. Introduction

Now a days Liquid Crystal Displays (LCDs) are becoming very popular output devices and are rapidly replacing seven segment displays. There are varieties of alphanumeric LCDs readily available in the market ranging from 16X1 up to 40X4. In fact, it is a basic output device used along with 4X4 matrix keyboard in many embedded applications.

If we interface these input and output peripherals on separate IO ports of a microcontroller, then very limited number of IO pins will remain available to interface other input and output devices like ADC, DAC, relays, I2C devices (DS1307/SHT71), etc. If we use LCD in 8-Bit mode then at least 13 IO pins will be required to interface it, but to minimize the IO pin count, generally, we use LCD in its 4-Bit mode operation. Thus at least 7 Pins will be required for the LCD. Apart from that, 4X4 matrix keyboard will require 8 more IO pins of a microcontroller. Thus both LCD and 4X4 matrix keyboard will consume 15 IO port pins of a microcontroller [1]. This is definitely poor utilization of microcontroller port pins to interface basic IO devices. To improve the utilization of port pins of a microcontroller we should interface

2212-0173 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.protcy.2012.05.073

LCD and 4X4 matrix keyboard at the same port. This technique will work without fail if both devices have Tri-state IO buffers, which otherwise will fail due to bus contention.

Both the devices are using same port of a microcontroller in time division multiplexing mode. Here, one may raise a point that if a microcontroller is communicating with LCD and at the same time user has pressed a key, then it will not be read by a microcontroller and any vital data fed from user may be lost by a microcontroller. But there are two basic assumptions, first is that, this port is available almost all the time for matrix keyboard, as LCDs are operating at very high rate and will take very less time to communicate with a microcontroller and second is that, 4X4 matrix keyboard is operated by human beings who are very slow in operation. Between any two key strokes, generally, minimum time gap of 500ms is observed and this time gap is quite sufficient for LCD communication if at all it is to be taken place. So with this basic assumption, interfacing of LCD and 4X4 matrix keyboard to the same port of a microcontroller will work without fail.

2. Interfacing of LCD & Keyboard

In this section, interfacing of a LCD and 4X4 matrix keyboard to a common port of MCS-51 family microcontroller is explained. As a case study author has chosen 89C52 microcontroller along with MM74C922 a keyboard controller. As 4X4 matrix keyboard does not have capability of tri-stating its output line, so MM74C922 keyboard controller IC is used to fulfil this task. Another advantage of using keyboard controller is that, microcontroller is fully relieved from keyboard polling, as MM74C922 generates interrupt upon a key press and generates corresponding key code in its output buffer to be read by a microcontroller. Second advantage is that de-bouncing of a pressed key is done by the keyboard controller itself. Thus vital 20ms de-bouncing delay time of a microcontroller is saved. Using this controller, 4X4 matrix keyboard is interfaced with the microcontroller at the expense of 6 IO pins instead of 8 IO pins, out of which 4 IO pins are common data lines with the LCD. That means, exclusively 2 IO pins are required for 4X4 matrix keyboard and 3 IO pins are required exclusively to control the LCD. Thus total number of IO pin count to interface basic IO devices will be limited to 9 instead of 15, which is less by 6, if both the devices would have been interfaced separately to the microcontroller.

In this case study, 8-Bits of Port 0 are used to interface both LCD and 4X4 matrix keyboard. Here LCD is used in its 4-Bit mode operation, 4 data lines of the LCD D7 to D4 are connected to P0.7 to P0.4 pins respectively [2] and to the same data lines, keyboard data lines D to A are connected respectively. Control lines EN of the LCD and OE˜ (OE BAR) of MM74C922 are connected to P0.2 and P0.3 pins respectively. Data Available (DA) is an interrupt of keyboard controller and is connected to P3.2 pin which is alternatively used as INT0 [3]. The circuit diagram is shown in fig. 1. This figure is a Proteus VSM 7 simulation run output. Microcontroller asserts only that control line of the device with whom it wants to communicate. For example, if it is going to read the output buffer of keyboard controller then EN of the LCD is de-asserted and OE˜ of MM74C922 is asserted.

3. Software Design

To illustrate proper working of LCD and 4X4 matrix keyboard both connected to a common port, author has also interfaced Real Time Clock (DS1307) to the microcontroller [4]. To initialize DS1307 at the time of power-on, user is asked to give choice to continue with old time or change to new time. This message is displayed on LCD as shown in fig.1. During this communication with the LCD, output buffer of the keyboard controller is tri-stated by de-asserting OE˜. Microcontroller communicates with the LCD via different subroutines like put_nibbles, put_string, lcd_ready, etc. During execution of any subroutine of the LCD, OE˜ signal of keyboard controller is de-asserted. The longest execution time is required for put_string subroutine and its rough estimate is 630 µs, provided that crystal frequency is 12MHz. Some of the subroutines used to communicate with LCD and keyboard are shown in the table 1.

If user presses key 'A', then he/she will be asked to input all timing parameters starting from current seconds up to current year. As user presses any key to enter tens of seconds, microcontroller is interrupted by the keyboard controller. Then microcontroller executes Interrupt Service Routine to read output buffer of the keyboard controller and the pressed key is echoed back to the user as shown fig. 2. During execution of this ISR EN signal of LCD is de-asserted so as to tri-state IO buffer of the LCD. Execution time for this ISR is 10 µs fixed.

Table 1 Different subroutines used to communicate with LCD and keyboard

Subroutine to transfer nibbles to LCD:
void put_nibbles(**void**)
{
lcd_ready();
kbd_en=1; //tri-state output buffer of 74C922
lcd_buff1=lcd_buff;
rw=0;
rs=lcd_rs;
rd_buff1=lcdport;
rd_buff1=rd_buff1&0x0f;
lcd_buff=lcd_buff&0xf0;
lcd_buff=lcd_buff|rd_buff1;
lcdport=lcd_buff;
lcd_en=1; //Enable LCD to write HI-NIB. into it
lcd_en=0; //Disable it
lcd_buff1=lcd_buff1<<4;
lcd_buff1=lcd_buff1|rd_buff1;
lcdport=lcd_buff1;
lcd_en=1; **//Enable LCD to write HI-NIB. into it**
lcd_en=0; //Disable it and return
}
	Subroutine to check LCD Ready:
	void lcd_ready(**void**)
	{
	kbd_en=1; //tri-state output buffer of 74C922
	busy=1; //set MSB as input bit
	rw=1;
	rs=0;
	lcd_busy: **lcd_en=1; //**Rd status of LCD enabling it
	rd_buff1=lcdport;
	lcd_en=0;
	rd_buff1=rd_buff1&0x80;
	lcd_en=1;
	rd_buff2=lcdport;
	lcd_en=0;
	if(rd_buff1==0x80)//if busy is still HI
	goto lcd_busy; //then do not return
	}

Interrupt Service Routine to read key code:
void isrint0(**void**) **interrupt 0** //ISR to read KBD.
{
lcd_en=0; //Disable LCD
lcdport=0xf0; //make HI-nib of P0 as input
kbd_en=0; //enable output buffer of 74C922
keycode=lcdport;
kbd_en=1; //Disable output buffer of 74C922
keycode=keycode>>4;
kbd_buff[kbd_ptr]=keycode;
}
	Subroutine to display string on LCD:
	void put_string(**unsigned char** _string[16])
	{
	lcd_rs=1;
	for(i=0;i<16;i++)
	{
	if(_string[i]==0x24) goto exit;
	lcd_buff=_string[i];
	put_nibbles();
	}
	exit:;
	}

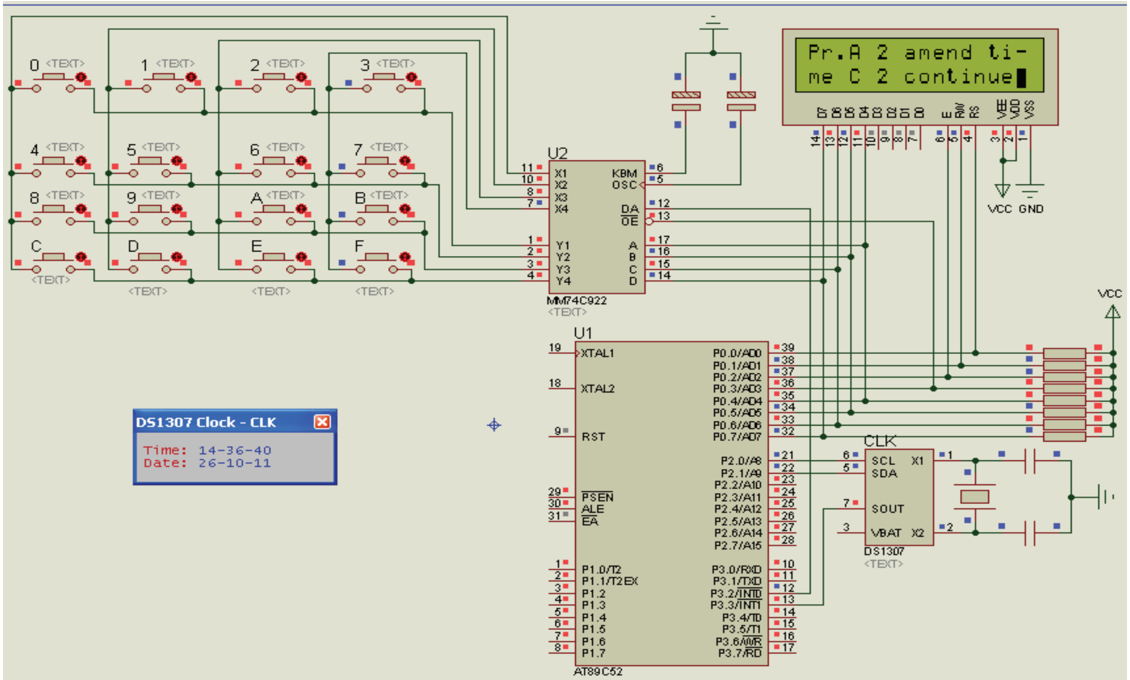

Fig. 1 Interfacing of LCD and keyboard to PORT0 of 89C52 microcontroller

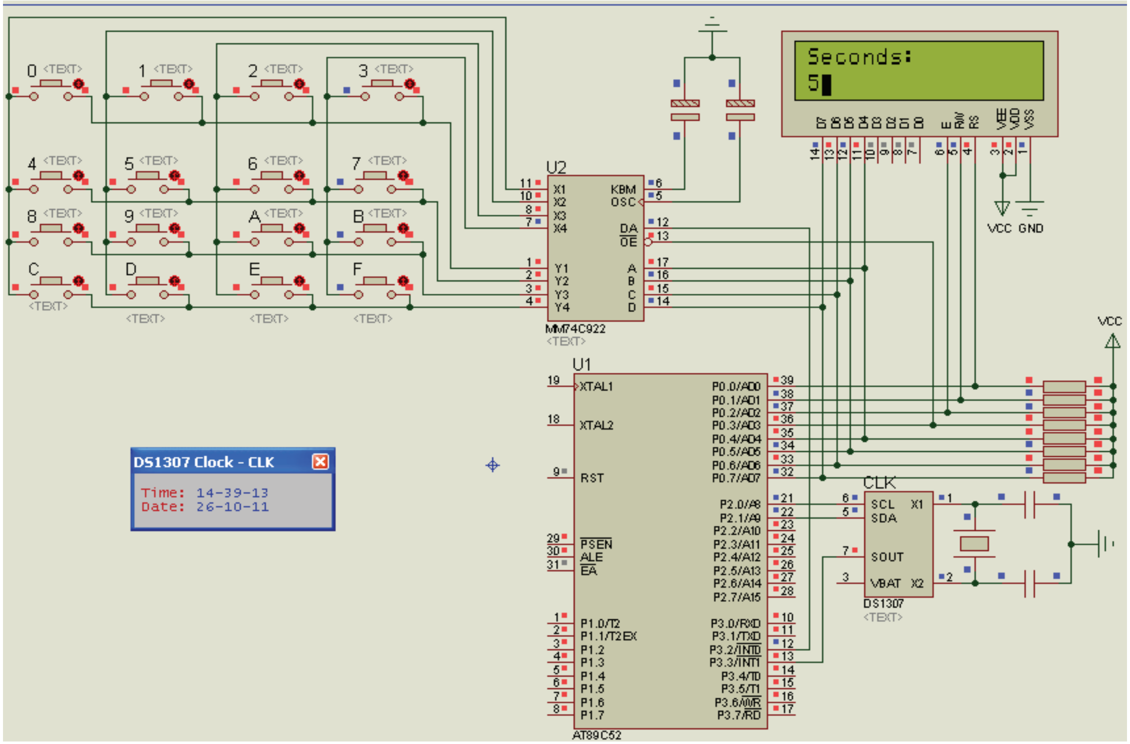

Fig. 2 Status of display when user presses key 5 to enter tens of second

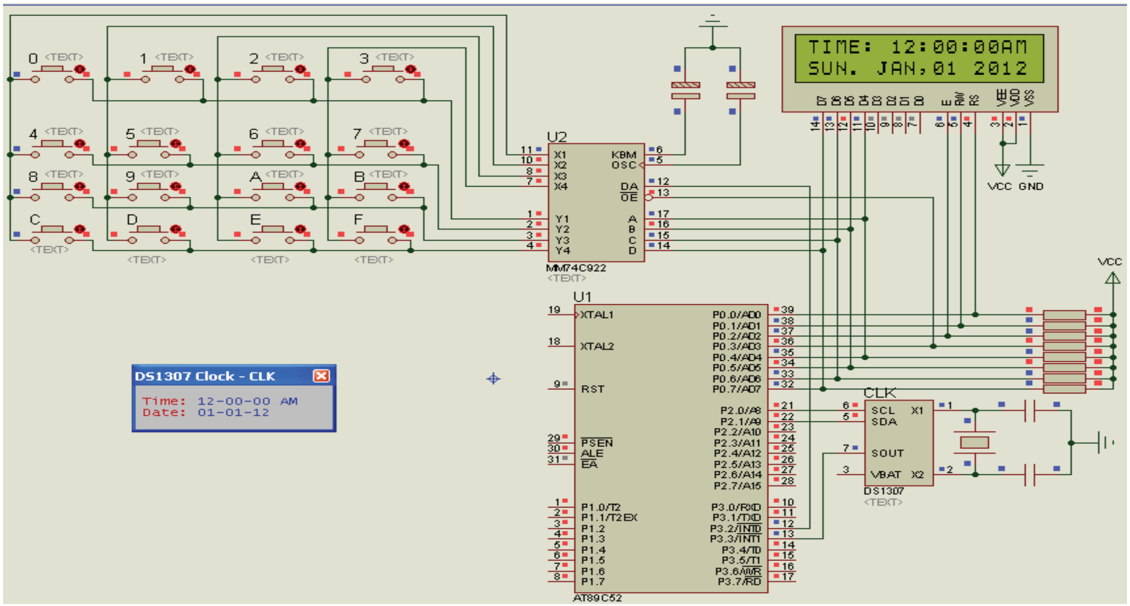
TIME: 12:00:00AM
SUN. JAN,01 2012
D7 D6 D5 D4 D3 D2 D1 D0 E RW RS VEE VDD VSS
0 1 2 3 4 5 6 7 8 9 A B C D E F
<TEXT>
U2
X1 X2 X3 X4 Y1 Y2 Y3 Y4
KBM OSC DA OE A B C D
MM74C922
U1
XTAL1 XTAL2 RST PSEN ALE EA
P1.0/T2 P1.1/T2EX P1.2 P1.3 P1.4 P1.5 P1.6 P1.7
P0.0/AD0 P0.1/AD1 P0.2/AD2 P0.3/AD3 P0.4/AD4 P0.5/AD5 P0.6/AD6 P0.7/AD7
P2.0/A8 P2.1/A9 P2.2/A10 P2.3/A11 P2.4/A12 P2.5/A13 P2.6/A14 P2.7/A15
P3.0/RXD P3.1/TXD P3.2/INT0 P3.3/INT1 P3.4/T0 P3.5/T1 P3.6/WR P3.7/RD
AT89C52
CLK
SCL SDA SOUT VBAT X1 X2
DS1307
VCC GND
VCC
DS1307 Clock - CLK
Time: 12-00-00 AM
Date: 01-01-12

Fig. 3 Output status of the system when user completes initialization of RTC DS1307

As soon as initialization process of RCT DS1307 is finished, it generates square wave of 1Hz continuously on its SOUT pin. This signal is connected to P3.3 pin of the microcontroller which has alternate function of INT1. During execution of ISR for Interrupt-1 every after 1 second, microcontroller reads current time, day and date from DS1307 and displays it on the LCD, as shown in fig. 3.

4. Conclusion

In this work, it is observed that a systematic design approach for interfacing LCD and 4X4 matrix keyboard to a single **PORT** of a micro-controller is quite possible so as to improve the hardware utilization factor of the system. 4-Bit interface mode of LCD is used to reduce number of IO pins required to interface it with a micro-controller. Tri-stating of 4X4 matrix keyboard is achieved by using MM74C922 keyboard controller. For complex embedded system applications where number of IO peripherals in the system is more, generally Hi-end microcontrollers like ARM7 are chosen where number IO port pins are in abundant, but it results in increased complexity, size and cost of the circuit. But by using the effective interfacing technique for basic IO devices like LCD and matrix keyboard, as illustrated above, one can still build complex embedded system using Low-end microcontrollers like 8051/PIC16FXXX. Applications of this technique are many, like public telephone booth, fuel (petrol/diesel) filling machine, automation of green-house, AC & DC drives, etc. Any embedded system which consists of 4X4 matrix keyboard and LCD can be built using this novel technique. The same technique can be used for those devices which have internal buffers that can be Tri-stated by enable control signal.

References

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay ‘The 8051 Microcontroller and embedded Systems Using Assembly and C’, 2/e, Pearson Education, Inc.
2. http://embeddedcraft.org/lcd4bit.html#top
3. http://www.edaboard.com/ftopic312107.html
4. www.8051projects.net/