# Concepts of 8051 Microcontroller External Interrupts: Programming and Applications

**P.Thimmaiah**

Assistant Professor, Department of Electronics, Sri Krishnadevaraya University, Anantapuramu, AP, India

***Abstract*: This paper describes the 8051 microcontroller interrupts concepts, programming and applications. The applications include an interfacing of LEDS, STEPPER MOTOR, DC MOTOR and SERIAL PORT with micro controller and these are controlled by external interrupts (INT0, INT1) signals through switches. In this experiment P89V51RD2 8-bit microcontroller is used to program the interrupts to flash LED's, rotate the stepper motor and DC motor through clockwise and anti-clock wise directions. The Hardware description and software developments are presented in this paper.**

***Keywords*: Interrupt, Stepper Motor, DC Motor, Interrupt Service Routine, Interrupt Handler**


**Published in IJIRMPS (E-ISSN: 2349-7300), Volume 1, Issue 1, September-October 2013**



## I. Introduction

The Microcontroller can serve several devices. Interrupt is one of the most important and powerful concepts and features in microcontroller/processor applications [1]. Almost all the real world and real time systems built around microcontrollers and microprocessors make use of interrupts. The interrupts refer to a notification, communicated to the controller, by a hardware device or software, on receipt of which controller momentarily stops and responds to the interrupt. Whenever an interrupt occurs the controller completes the execution of the current instruction and starts the execution of an Interrupt Service Routine (ISR) or Interrupt Handler. ISR is a piece of code that tells the processor or controller what to do when the interrupt occurs. After the execution of ISR, controller returns back to the instruction it has jumped from (before the interrupt was received). The interrupts can be either hardware interrupts or software interrupts. If the interrupts are generated by the controller's in built devices, like timer interrupts, or by the interfaced devices, they are called the hardware interrupts. If the interrupts are generated by a piece of code, they are termed as software interrupts.

In polling, the microcontroller keeps checking the status of other devices, and while doing so it does no other operation and consumes all its processing time for monitoring [2]. This problem can be addressed by using interrupts. In interrupt method, the controller responds to only when an interruption occurs. Thus in interrupt method, controller is not required to regularly monitor the status (flags, signals etc.) of interfaced and inbuilt devices. When an interrupt is invoked, the microcontroller runs the interrupt service routine. For every interrupt, there is a fixed location set aside to hold the addresses of ISRs as shown below.

| Interrupt | ROM Location |
|---|---|
| INT0 (EX0) | 0003H |
| TIMER0 (TF0) | 000BH |
| INT1 (EX1) | 0013H |
| TIMER1 (TF1) | 001BH |
| SERIAL COMMUNICATION INTERRUPTS (TI & RI) | 0023H |

The 8051 controller has six hardware interrupts of which five are available to the programmer. These are as follows:

### A. RESET Interrupts

This is also known as Power on Reset (POR). When the RESET interrupt is received, the controller restarts executing code from 0000H location. This is an interrupt which is not available to or, better to say, need not be available to the programmer.

### B. Timer Interrupts (TF0 & TF1)

Each Timer is associated with a Timer interrupt. A timer interrupt notifies the microcontroller that the corresponding Timer has finished counting.

### C. External Interrupts (INT0&INT1)

There are two external interrupts EX0 and EX1 to serve external devices. Both these interrupts are active low. In AT89C51, P3.2 (INT0) and P3.3 (INT1) pins are available for external interrupts 0 and 1 respectively. An external interrupt notifies the microcontroller that an external device needs its service.

### D. Serial Interrupts (TI & RI)

This interrupt is used for serial communication. When enabled, it notifies the controller whether a byte has been received or transmitted.

**Interrupt Enable Register (EA):**

D7 D6 D5 D4 D3 D2 D1 D0 EA - - ES ET1 EX1 ET0 EX0

## II. Programming External Interrupts

The external interrupts are the interrupts received from the (external) devices interfaced with the microcontroller. They are received at INTx pins of the controller. These can be level triggered or edge triggered. In level triggered, interrupt is enabled for a low at INTx pin; while in case of edge triggering, interrupt is enabled for a high to low transition at INTx pin. The edge or level trigger is decided by the TCON register.

**TCON (Timer/Counter) Register (Bit-addressable)** D7 D6 D5 D4 D3 D2 D1 D0

TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0

Setting the IT0 and IT1 bits make the external interrupt 0 and 1 edge triggered respectively. By default these bits are cleared and so external interrupt is level triggered.

### A. DC Motor

A Direct Current (DC) motor is another widely used device that translates electrical pulses into mechanical movement [6]. In the DC motor we have only + and – leads. Connecting them to a DC voltage source moves the motor in one direction. By reversing the polarity, the DC motor will move in the opposite direction. One can easily experiment with the DC motor.

### 1. Optical Isolation

It is better to use opto-isolator between motor circuit and microcontroller because it will protect microcontroller from EMI created by the motor brushes. If motor voltage rating is higher than the voltage used for microcontroller then opto-isolator prevents damage to the microcontroller by providing optical isolation. If there is any fault in motor circuit or power supply, microcontroller is safe.

**2. H-Bridge Configuration for Bi-directional Rotation** Direction of DC motor can be changed by changing polarity of DC voltage [9]. H-Bridge configuration is very popular for bi-directional speed control. H-bridge is available in single IC form such as L293. Four transistors are in built in this IC. This IC produces heat during the operation; hence it requires heat sink for continuous operation. Interfacing of H-bridge IC with microcontroller using Quad opto coupler IC ILQ74 is shown in the following circuit diagram. Quad opto coupler IC has four in-built opto coupler which provides optical isolation between H-bridge circuit and microcontroller 89V51RD2.Microcontroller P89V51RD2 IC is placed on bread board and crystal oscillator frequency of 12MHz is given to microcontroller pins 18, 19. Reset circuitry is connected at pin 9, switches sw1 (INT0), sw2 (INT1) are connected to the pins 12, 13 of microcontroller through 10Kohms resistors. Port2 pins are connected to optoisolator.and the output of optoisolator connected to L293 driver to rotate the DC motor in clockwise and in anti clockwise directions.

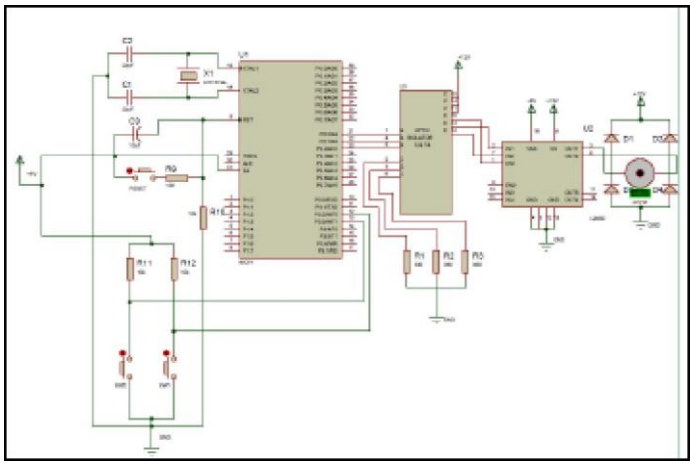

Fig. 1: Circuit Diagram of DC Motor

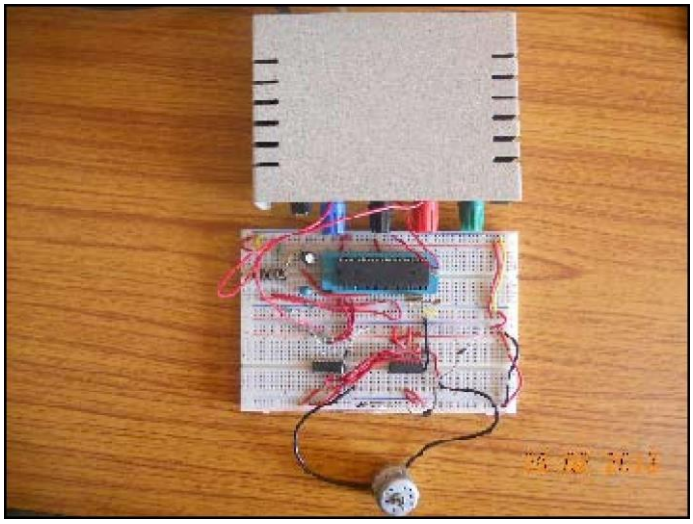

Fig.2: Interfacing of DC Motor

```
// This is a program for Dc motor in ” c”.
        #include<reg51.h>
        sbit SW=P3^2;
        sbit SW1=P3^3;
        sbit enable=P3^0;
         sbit motor1=P2^1;
        sbit motor2=P2^2;
        void main()
        {
         SW=1;
         SW1=1;
         enable=0;
         motor1=0;
         motor2=0;
         while(1)
           {
              enable=1;
              if(SW==0)
               {
                 motor1=0;
                 motor2=1;
                }
```

```
 enable=0;
if(SW1==0)
 {
  motor1=1;
  motor2=0;
          }
  }
 }
```

**B. Serial Communication**

Serial communication is a way enables different equipments to communicate with their outside world. It is called serial because the data bits will be sent in a serial way over a single line. A personal computer has a serial port known as communication port or COM Port used to connect a modem for example or any other device, there could be more than one COM Port in a PC [7]. Serial ports are controlled by a special chip called UART (Universal Asynchronous Receiver Transmitter) [8].

**1. RS232:**

RS-232 (Recommended standard-232) is a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices. In other words, RS-232 is a long established standard that describes the physical interface and protocol for relatively low-speed serial data communication between computers and related devices.

Microcontroller P89V51RD2 IC is placed on bread board and crystal oscillator frequency of 12MHz is given to microcontroller pins 18, 19. Reset circuitry is connected at pin 9, switches sw1 (INT0), sw2 (INT1) are connected to the pins 12, 13 of microcontroller through 10Kohms resistors. A line driver MAX232 chip is connected to microcontroller to convert RS232 voltage levels to TTL levels, and vice versa. The output of RS232 is interfaced to PC. When switch SW1 is pressed then external interrupt INT0 is activated then message 'SKU' is displayed on hyper terminal. When switch SW2 is pressed then external interrupt INT1 is activated then message 'ELECTRONICS' is displayed on hyper terminal.

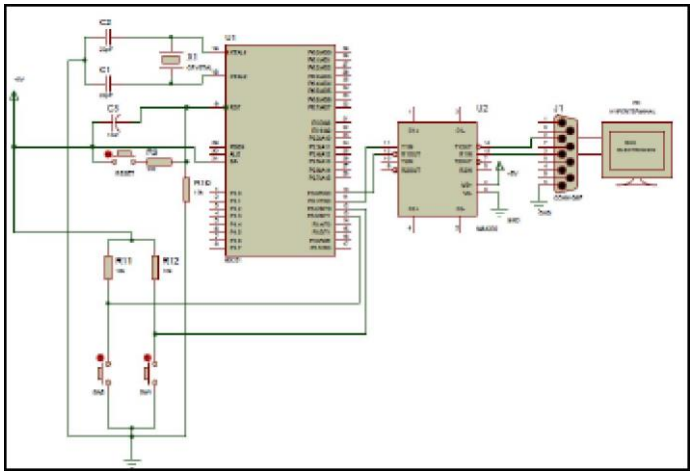

Fig. 3: Circuit Diagram of Serial Communication

**Software**

```
org 00h
ajmp start
org 03h                ; Vector location for
                                        ; External interrupt INT0
ajmp mes1
org 13h                ; Vector location for
                                        ; External interrupt INT1
ajmp mes2
start: mov IE,#85h ; Enable External
; interrupts
```

```
setb IT0                ; Negative edge trigger
; for INT0
setb IT1                ; Negative edge trigger
; for INT1
mov TMOD,#20h ; timer1,mode2 mov TH1,#-3 ; 9600 baud rate mov SCON,#50h ; 8-bit,1stop bit
; REN enable
setb TR1                        ; start timer 1
here1: sjmp here1
mes1:    mov dptr,#message1
loop3: clr a
movc a,@a+dptr
loop4: jz loop4
acall sendcom ; send to serial port inc dptr        ; move to next value
sjmp loop3
here2: sjmp here2
mes2:    mov dptr,#message2
loop5: clr a
movc a,@a+dptr
loop6: jz loop6
acall sendcom ; send to serial port inc dptr        ; move to next value
sjmp loop5
sendcom:
mov SBUF,a ; place value in buffer here:jnb TI,here ; wait until transmitted
clr TI                          ; clear
reti
message1: db ' SKU ',0
message2: db ' ELECTRONICS ',0
delay: mov r4,#02h loop9: mov r6,#0ffh loop8: mov r5,#0ffh loop7: djnz r5,loop7
djnz r6,loop8
djnz r4,loop9
reti
end
```

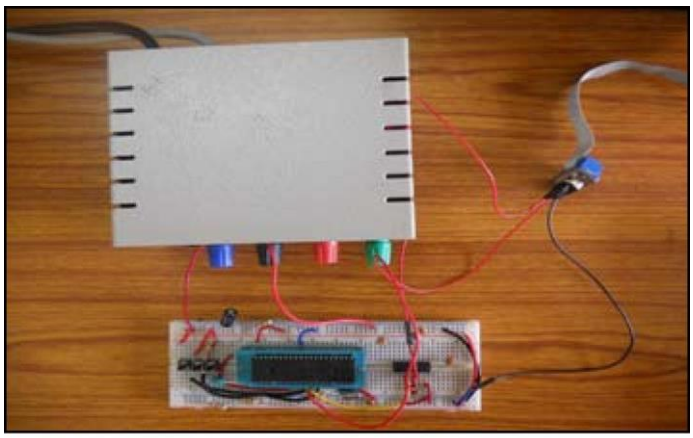

Fig.4: Interfacing of Serial Port

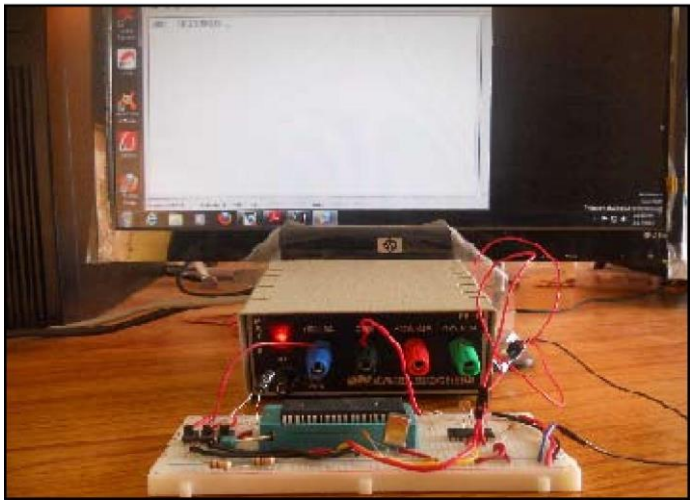

Fig. 5: Output of Serial Port

## III. Conclusion

In this paper, DC MOTOR and SERIAL PORT are interfaced with 8051micro controller and these are controlled by external interrupt (INT0, INT1) signals through switches have been presented., the rotation of DC MOTOR either in clockwise or in anti clockwise directions and messages are displayed on PC hyper terminal through SERIAL PORT can be achieved by programming the interrupts of P89V51RD2 an 8 bit microcontroller.

## References


[1] The 8051Microcontroller and Embedded systems- Muhammad Ali Mazidi, PHI Publishers.

[2] "Embedded Systems - Architecture, Programming and Design", Raj Kamal, Publs. McGraw-Hill Education

[3] P.Thimmaiah, K.R.Rao, E.R.Gopal,"An inexpensive stepper motor interface for microcontrollers", Lab Experiments, Vol. 3, pp. 185-191, Sep-2003.

[4] P.Thimmaiah, K.R.Rao, E.R.Gopal,"Interfacing Bipolar stepper motor with Microcontrollers", J.Instrum.Soc.India.34 (3), pp. 163-169, 2004.

[5] P.Thimmaiah, K.R.Rao, E.R.Gopal,"Design and development of an embedded system for web camera control", J.Instrum. Soc.India. 36 (2), pp. 125-132, 2006.

[6] Safina Al Nisa, Lini Mathew, S Chatterji, "Comparative Analysis of Speed Control of DC Motor using AI Technique", International Journal of Engineering Research and applications. Vol. 3, Issue 3, May-June 2013, pp.1137- 1146.

[7] Gade S S,Kanase A B,Shendge S B, Uplane M D.,"Serial Communication Protocol For Embedded Application", International Journal of Information Technology and Knowledge Management, July-December 2010, Vol. 2, No.
2, pp. 461-463.

[8] Ozwin Dominic Dsouza, Manjunathababu. P, Babu Naik,"Bidirectional Speed Control Of DC Motor And Stepper Motor Through Matlab Using PIC Microcontroller", International Journal of Research in Engineering and Technology.