International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 38453-38458

Developing a Media of Active Learning (AL) strategy of Micro controller 8051 Multiple Interrupts handling teaching by using the MCU 8051 IDE-Integrated Development Environment and C program-SDCC in supporting the implementation of Active Learning in Higher Education-ALFHE

Dahlan RP Sitompul
Lecturer,
Department of Information Technology,
ATI-Immanuel Medan-Kopertis Wil I,
Program Studi S1 Ilmu KomputerUSU
Medan
drps62@yahoo.com

Mhd.Furqan,
Lecturer ,
Department of Computer Science
Universitas Islam Negeri Sumatera Utara
S3 Ilmu Komputer USU
Medan
mhd_furqan07@yahoo.com

Abstract- On this occasion, the author discusses the development of an instructional media of active learning teaching of microcontroller 8051 Multiple Interrupts handling using a simulator called MCU 8051 IDE on Windows 7 (MCU 8051 IDE runs on various operating systems) to support the implementation of active learning for higher education. The developed Learning and teaching Media is in the form of simulation of multiple interrupts program. ; This simulator (MCU 8051 IDE) can be downloaded for free at sourceforge.net site [3]. In this paper the writer built a simulation program (in C language-SDCC) that shows of how multiple interrupts handled by the associated interrupt vector. The scenario of this multiple interrupts is as follows, five sources of interrupt namely timer 0, timer 1, External Interrupt 0, External Interrupt 1, and Serial Port (Receive Interrupt) are used, but serial port interrupt is initiated by External Interrupt 0 and has the higher Priority by setting the PS bit (bit 4 of IP SFR). Serial port Interrupt will handle or run a character receiving process from a source; it will run for ever on purpose to show the power or priority of an interrupt source that has a higher priority and no other source can interrupt it. Both, External Interrupts, External Interrupt 0 and External Interrupt 1 are initiated or triggered by two switches connected to P3.2 and P3.3 successively. External Interrupt 0, External Interrupt 1, Timer 0, and Timer 1 interrupts will complement the status of 4 LEDs connected to port 1 (p 1.0, p1.1, p1.2, and p1.3) consecutively. External Interrupt 1 will run longer than timer 0 and timer 1 interrupt; to prove that neither timer 0 nor timer 1 will be able to interrupt it when it is operating, but serial interrupt (initiated by External Interrupt 0). According to the writer's observation the developed simulation program has been stable and successful as a media of teaching and learning of Microcontroller 8051 Multiple Interrupts handling and increase the participation and engagement of the students in the learning process, allowing the teacher to apply their teaching strategies more conveniently and successfully in overcoming the lack of funds to meet the needs for equipments in supporting active learning [6]. The simulation program has also been successful in reducing errors in developing or manufacturing the real equipment (reducing costs) [7] [8].

Keywords:Micro controller, 8051,SDCC,C language,Control, Simulator, Active Learning, Teaching, Media.

Introduction

Based on the author's experience in teaching at college, the authors see the need for a changing in teaching methodologies (Instructional Strategy of teaching) in order to increase the success rate of teaching and learning process; transforming conventional learning methodologies (teacher as central or learning resources) to the active teaching methodologies or Active Learning Methodology (students as a learning center). The application of active learning strategy has demonstrated significant efficacies compared to the conventional learning [1] [2]. Active learning strategy has been shown to increase the participation and student engagement (reduction in stress level of the students and teachers) in the classroom and increasing students' learning desire and finally produce more innovative students; students are given more freedom to try new things by using innovative and interactive teaching media. They use the simulator before starting developing or manufacturing real equipment; the implication is the decrease of the production costs. One of the strategies in the application of active learning is the use of instructional

media in the process of teaching and learning. At this time there are various microcontroller simulator 8051 learning that can be downloaded for free and can then be used as a media of learning in the application of active learning [3][4]. The author used the simulator MCU 8051 IDE to develop microcontroller 8051 Multiple Interrupts handling simulator that runs on Windows 7 operating system. The programming language used in developing of this simulation program was C language 8051-SDCC and used out and input visual hardware provided by the simulator. The visual output Hardware displayed the information of the associated interrupt vector is running. The input visual hardware used are two visual switches that connected to P3.2 and P3.3 consecutively that will trigger each External Interrupt (External Interrupt 0 and External Interrupt 1) by connecting the associated pin of P3.2 or P3.3 to GND-Ground. The virtual output hardware used is an array of LED panel that display the operating Interrupt vector (LEDs connected to p1.0, p1.1, p1.2, p1.3 successively display external Interrupt 0, external interrupt 1, timer 0 interrupt, timer 1 interrupt); serial interrupt has been set to have a higher priority yet it ignited by external 0 interrupt by connecting p3.2 pin of 8051 to GND (switch on the first switch of the switch array panel). The author hopes that this simulator will contribute to the development of science, particularly in the fields of microprocessor, microcontroller, computer science, telecommunication, robotics, and computer-based control system and engineering and can also be used in higher education as a media of teaching in active learning class and can further inspire the teachers to create a variety of other media of teaching by utilizing the MCU 8051 IDE simulator [7] [8].

Related Work

Using media of active learning teaching developed both by using computer program in the form of simulator and non computer related media have been done by some researchers and scholars. Some of them have published their works in this field as follow; in this paper [5] the author reported the designing of decoder to display the result of the digital voltage level on a seven segment display in the form of character “H” and “L”. The author proposed the using of his design to be implemented in active learning class especially in electronic digital laboratory. In paper [7] the authors develop an active learning simulator using both MCU8051 IDE and SDCC. In this paper they report that the simulator will display the level voltage of a pin of an 8051 port on a virtual seven segment display and propose to use it as a media of teaching especially in the classes that have adopted active learning strategies. In this paper [8] the authors reported the development of a simulator program that simulate the process of multiple interrupts runs on a 8051 microcontroller by using MCU8051 IDE and ASEM 51.

Proposed System

In this paper the author proposed the development of an active learning media that can be used in an active learning class or a laboratory. The simulator program was developed by using free to download programs MCU8051 and SDCC from Internet and can be run on various operating systems. This simulator will enable the learning participants to run the learning process conveniently. It will simulate of how the microcontroller in handling multiple interrupts that run simultaneously. This simulator also will display the running program on a Graphical User Interface (GUI) and some virtual hardware as shown on Figure 1-Figure 3 below.

Simulation

A simulation has been done and the result of it has been captured as shown on the Figure1-Figure 3. Figure1.below shows the result of a print screen of the GUI (Graphical User Interface) of the simulator before running; the program virtual hardware consist of an array of LEDs that will display the status of the associated running interrupt vector, the LEDs connected to p1.0, p1.1, p1.2, p1.3 successively display external Interrupt 0, external interrupt 1, timer 0 interrupt, timer 1 interrupt status; each interrupt vector will complement the status of each LED (On to Off and vice versa) each time it is executed.

.

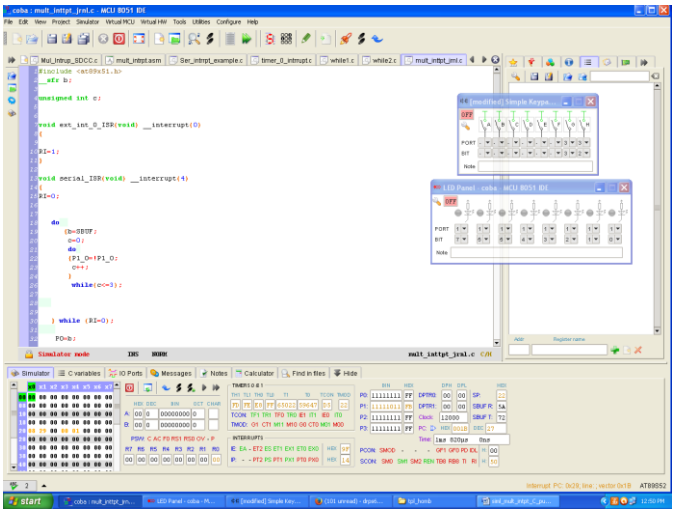

Fig.1. Print Screen of GUI Prior Executing of the Program [3] [4]

External interrupt 0 and 1

An array of simple keypad (consist of 8 pieces switches) two of which connected to P3.2 and P3.3 are used to trigger the interrupt vector of External Interrupt 0 and External Interrupt 1 successively. Interrupt vector 1 has a higher priority; by setting the bit 2 (EX1) of interrupt priority (IP) SFR with bit

addressable address of BAH (196d) as shown in the following portion of program statement,

```
IT0=0x01; external interrupt 0 is active low
IT1=0x01; external interrupt 1 is active low
EX0=0x01; enable external interrupt 0
EX1=0x01; enable external interrupt 1
EA=0x01; enable all interrupt
PX1=0x01; set interrupt priority of external interrupt 1
void ext_int_0_ISR(void) __interrupt(0)
{
RI=1;
}
void ext_int_1_ISR(void) __interrupt(2)
{
int d=0;
do
    {
        P1_1=!P1_1;
        d++;
    } while (d<=4);
}
```

It can be seen clearly above that this external interrupt 1 vector will complement the LED connected to p1.1 of 8051 status this is due to the statement P1_1=!P1_1.

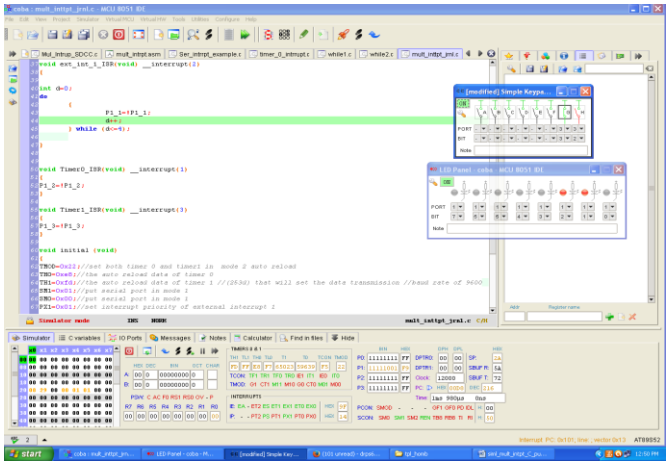

Fig.2. Print Screen of GUI shows the program executing the External Interrupt 1 program [3] [4]

Serial Interrupt (Receive Interrupt)

External interrupt 0 initiated by connecting a switch connected to p3.2 of 8051to GND. It will make the program to jump to the address of interrupt 0 vector address of 003H. This interrupt vector will complement the LED connected to p1.0 of 8051 status for several times (4 times) and will trigger the serial interrupt; this is done by inserting a statement RI=1 to the chunk of interrupt 0 vector, and it will force the program to jump to serial interrupt vector with interrupt vector address of 0023H. It will try to receive a character from serial buffer (SBUF) to P0. The following chunk of program statement will make it work,

```
void ext_int_0_ISR(void) __interrupt(0)
{
RI=1;
}
void serial_ISR(void) __interrupt(4)
{
RI=0;
  do
    {b=SBUF;
     c=0;
     do
     {P1_0=!P1_0;
      c++;
     }
      while(c<=3);

  } while (RI=0);

   P0=b;
   RI=0;
}
```

It will run forever on purpose to show that it will not be able to be interrupted until it accomplished its operation or task. Serial port interrupt priority is set by setting bit 4 (PS) of Interrupt priority IP SFR with bit addressable address of BCH (188d).

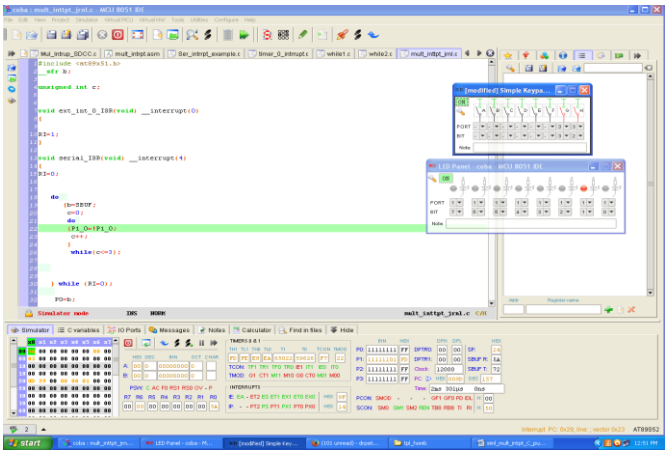

Fig.3. Print Screen of GUI shows the program executing the Serial Interrupt program initiated by external interrupt 0 [3] [4]

Timer 0 and Timer 1 interrupt

The rest of the interrupts, interrupt timer 0 and interrupt timer 1 with interrupt vector addresses of 00bh and 001bh successively will set or triggered by the overflowing of the timer 0 and timer 1 registers of TL0 and TL1 (overflow will set the appropriate flag of timer TF0 and TF1 consecutively), the initial data of each timer are put in TH0 and TH1 consecutively; these two timer set to run in auto reload mode. Timer 1 is used as the baud rate generator of serial interrupt as well; TH1 is loaded with data of 0fdh (253d) that correspond to transmission baud rate of 9600 as shown in the following chunk of program,

```
TMOD=0x22; set both timer 0 and timer1 in  mode 2 auto reload
TH0=0xE8; the auto reload data of timer 0
TH1=0xFD; the auto reload data of timer 1 (253d) that will set the data transmission baud rate of 9600
```

Both Timer 0 and timer 1 will be run by executing the following statements,

```
TR0=0x01; enable timer 0 interrupt
TR1=0x01; enable timer 1 interrupt
```

Yet beforehand both timers need to be initiated by the following statements,

```
TMOD = 0x22; set both timer 0 and timer1 in  mode 2 auto reload
TH0=0E8; the auto reload data of timer 0
TH1=0XFD; the auto reload data of timer 1 (253d) that will set the data transmission baud rate of 9600
SM1=0X01; put serial port in mode 1
SM0=0x01; put serial port in mode 1
ET0=0x01; enable timer 0
ET1=0x01; enable timer 1
EA=0x01; enable all interrupt
```

Eventually the program will do other program after executing a function called inital; in this scenario it will represented or mimicked by the following statement,

```
void main(void)
{
initial();
}
```

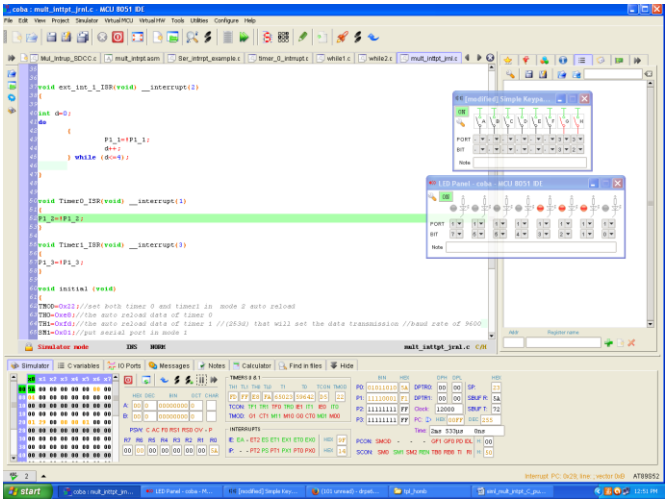

Fig.4. Print Screen of GUI shows the program executing the Timer 0 and Timer 1 Interrupts [3] [4]

The complete program of simulator

The flow chart and the full program of the simulator are shown below.

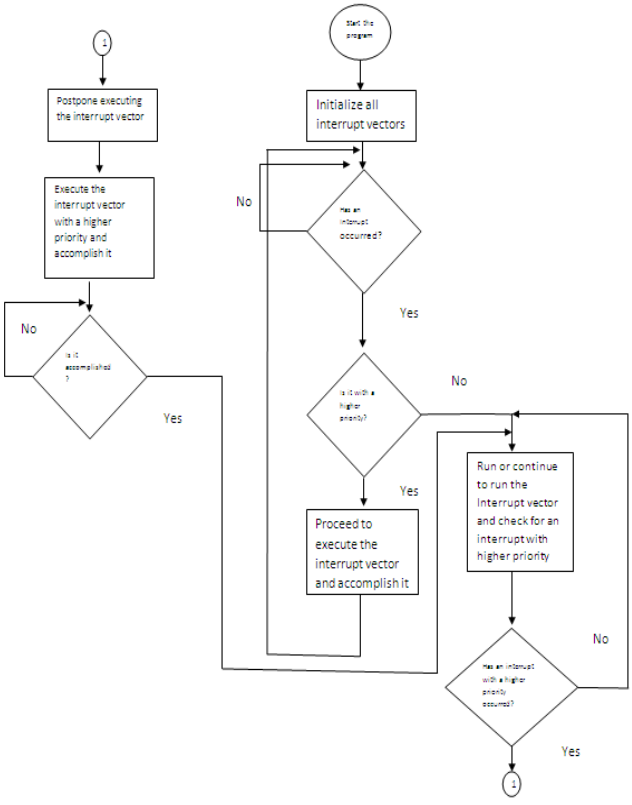

Fig.5. The flowchart program of the simulator

The complete program of the simulator is shown below [7].

```
#include <at89x51.h>
__sfr b;
```

```
unsigned int c;
void ext_int_0_ISR(void) __interrupt(0)
{
RI=1;
}
void serial_ISR(void) __interrupt(4)
{
RI=0;
  do
    {b=SBUF;
     c=0;
     do
     {P1_0=!P1_0;
      c++;
     }
      while(c<=3);

  } while (RI=0);

   P0=b;
   RI=0;
}
void ext_int_1_ISR(void) __interrupt(2)
{
int d=0;
do
    {
          P1_1=!P1_1;
          d++;
    } while (d<=4);

}
          void Timer0_ISR(void) __interrupt(1)
{
          P1_2=!P1_2;
}
          void Timer1_ISR(void) __interrupt(3)
{
          P1_3=!P1_3;
}

void initial (void)
{
TMOD=0x22;//set both timer 0 and timer1 in  mode 2 auto reload
TH0=0xe8;//the auto reload data of timer 0
TH1=0xfd;//the auto reload data of timer 1 //(253d) that will set the data transmission //baud rate of 9600
SM1=0x01;//put serial port in mode 1
SM0=0x00;//put serial port in mode 1
PX1=0x01;//set interrupt priority of external interrupt 1
PS=0x01;//set interrupt priority of serial port
ET0=0x01;//enable timer 0
ET1=0x01;//enable timer 1
IT0=0x01; //it0;external interrupt 0 is active low
IT1=0x01;//it1;external interrupt 1 is active low
EX0=0x01;//ex0;enable external interrupt 0
EX1=0x01;//enable external interrupt 1
ES=0x01;//enable serial port interrupt
REN=0x01;//enable receiver
EA=0x01;//enable all interrupt
TR0=0x01;//enbale timer 0 interrupt
TR1=0x01;//enable timer 1 interrupt
}
void main(void)
{
initial();
}
```

Conclusion

From the results of the simulation and its implementation in the class a few thing can be observed,

1. There are six sources of Interrupt of Microcontroller 8051 (RESET, External Interrupt 0, Timer 0, External Interrupt 1, Timer 1, and Serial Port (both Transmit and Receive Interrupt) with interrupt vector addresses of 0000H, 0003H, 000BH, 0013H, 001BH, 0023H successively
2. Initially all sources of Interrupts have the same Interrupt priority (the associated bit of Interrupt priority on Interrupt priority register-IP is reset)
3. The priority of an interrupt source can be upgraded by setting the associated bit of IP-Interrupt priority Register
4. An Interrupt source can be enable or disable by setting or resetting the appropriate bits of interrupt priority register-IP and bit 7 (Enable All-EA bit) in the Special Function Register-SFR
5. MCU 8051 can be used as an alternative media of teaching of Microcontroller 8051 Multiple Interrupts handling in ALFHE class
6. The C-SDCC language can be used to build up the simulation program
7. The program runs according to the plan, and is stable
8. The learning participants can immediately modify the program according to the needs or ideas that emerged in each group of study [7] [8]

Closing

This article is expected to contribute to science, particularly in the areas of microprocessor, computer, robotics, microcontroller or computer based control system engineering. And can be used as a media of learning and teaching in an active learning model in higher education that will facilitate the teaching and learning participants to master the subjects more conveniently and enable the teacher to carry out the teaching process better[7][8].

References

[1]http://ehe.pitt.edu/ojs/index.php/ehe/article/viewFile/54/29
[2] http://ehe.pitt.edu/ojs/index.php/ehe/article/view/48
[3]http://sourceforge.net/projects/mcu8051ide/files/latest/download
[4] www.edsim51.com/
[5] ilmukomputer.org/.../dahlan-PenampilHighLow20
[6]http://www.ydae.purdue.edu/lct/hbcu/documents/Active_Learning_Creating_Excitement_in_the_Classroom.pdf
[7]http://www.tjprc.org/download.php?fname=--1394618032-Designing%20of%20Microcontroller%20-FULL.Pdf
[8]http:
http://www.ripublication.com/ijaer10/ijaerv10n3_30.pdf