Report For Lab 6

Connor McCullough

EEN 312

3 April 2013

**Objective:**

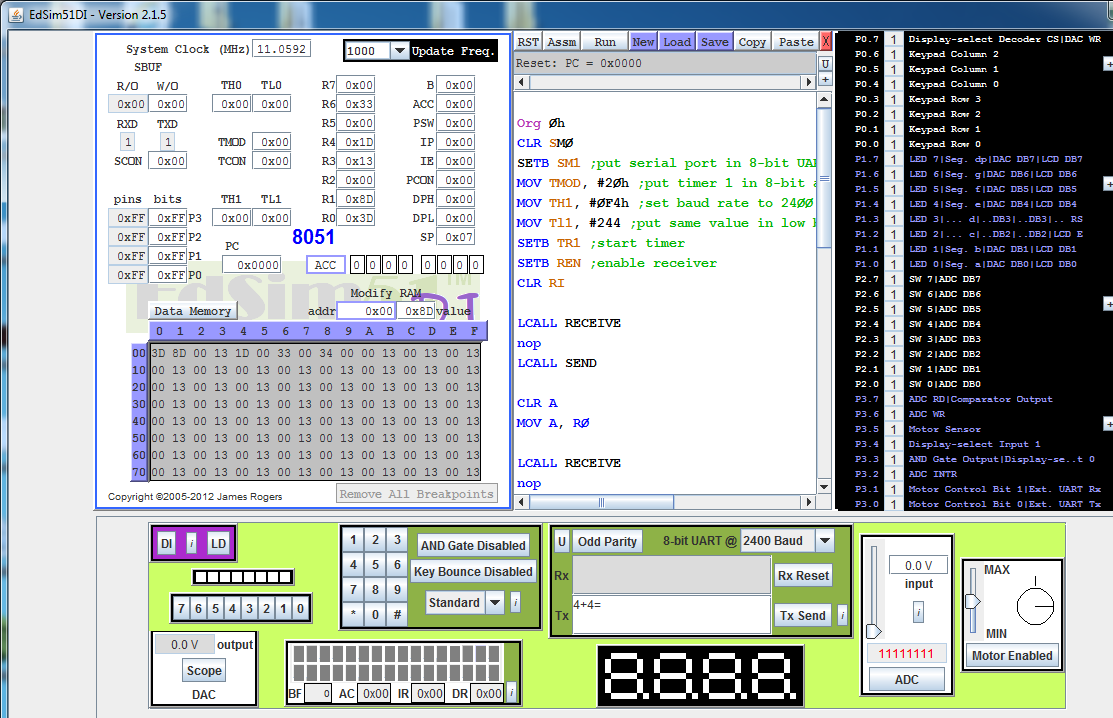
Write a program to input information serially from a keyboard panel and output it to the text monitor panel. This program will be a simple calculator that processing the input characters, adds them together, and outputs the input characters as well as the solution to the problem. This program requires knowledge of setting the baud rate using a timer, as well as configuring hardware settings.

**Equipment:**

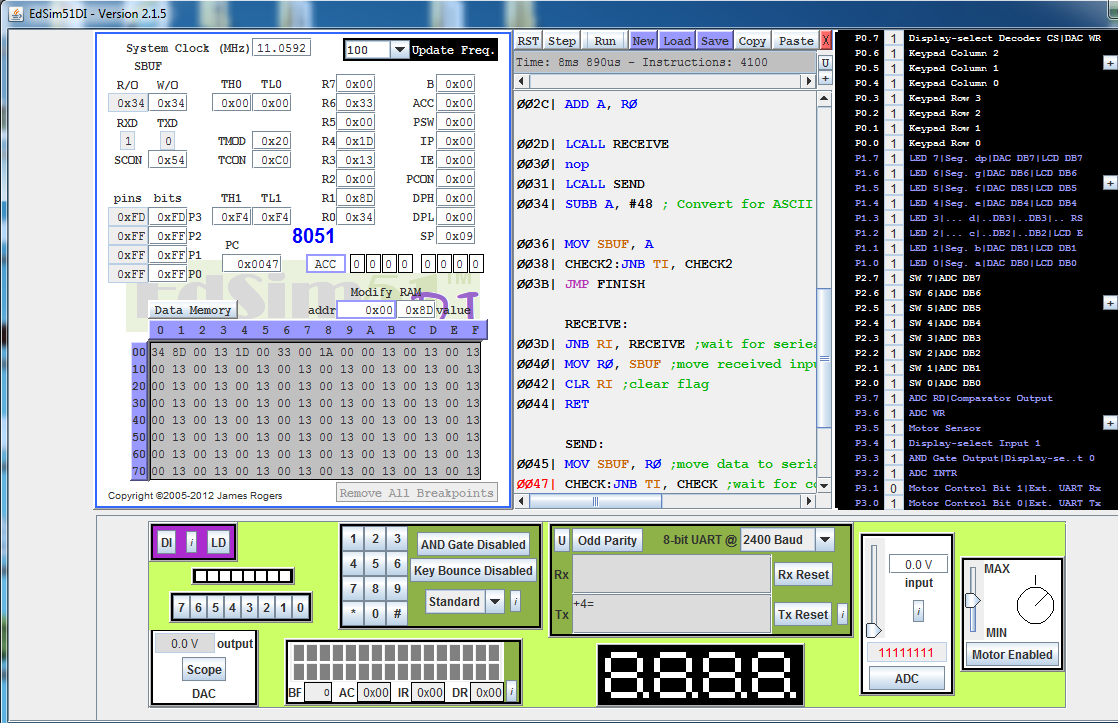
* Text editor and 8051 ASM assembler
* Step debugger to execute program one instruction at a time
* Data memory, accumulator, code memory
* Keyboard panel, Text monitor panel

**Results:**

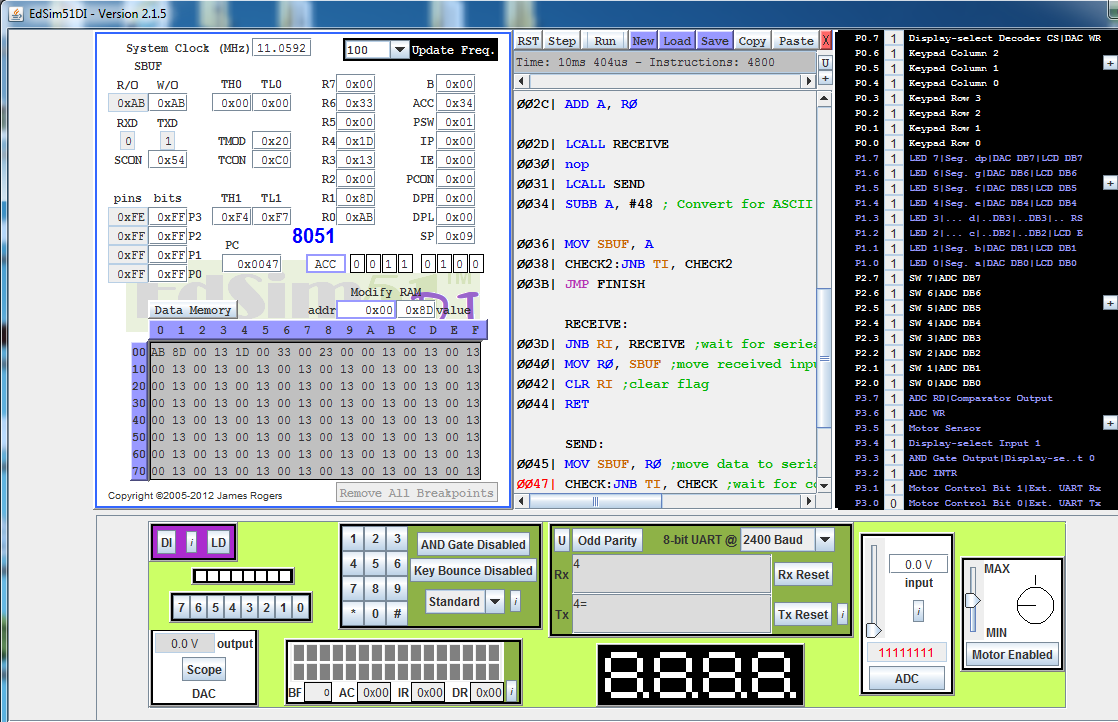
Text before sent to 8051



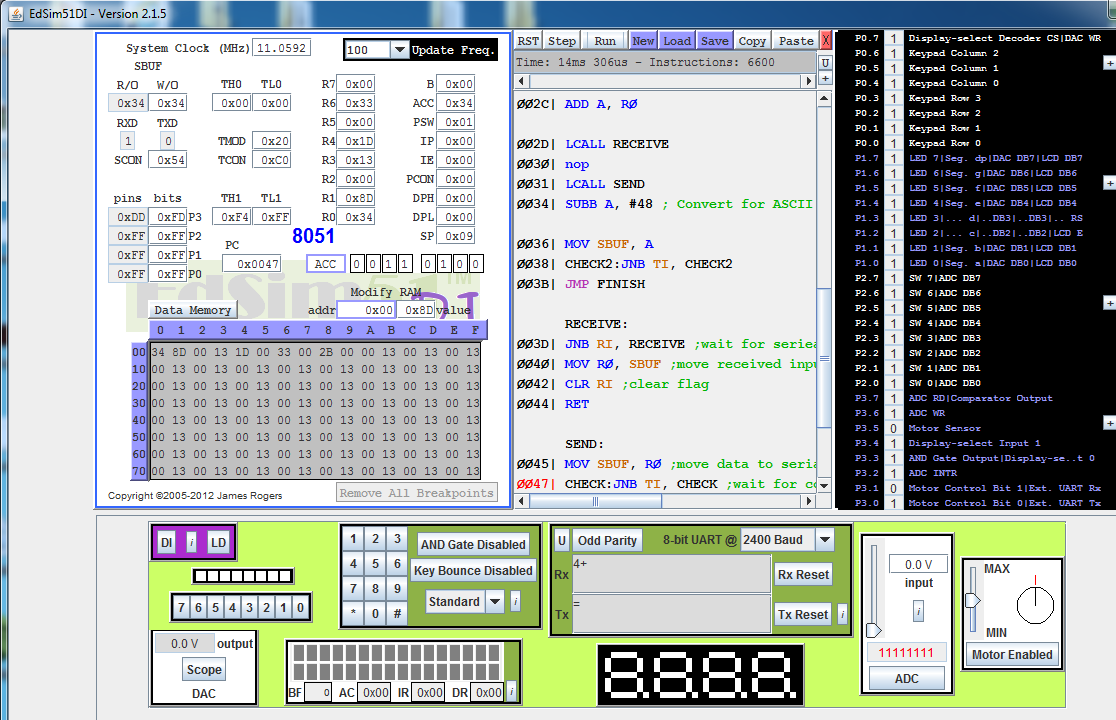
First character is read into program



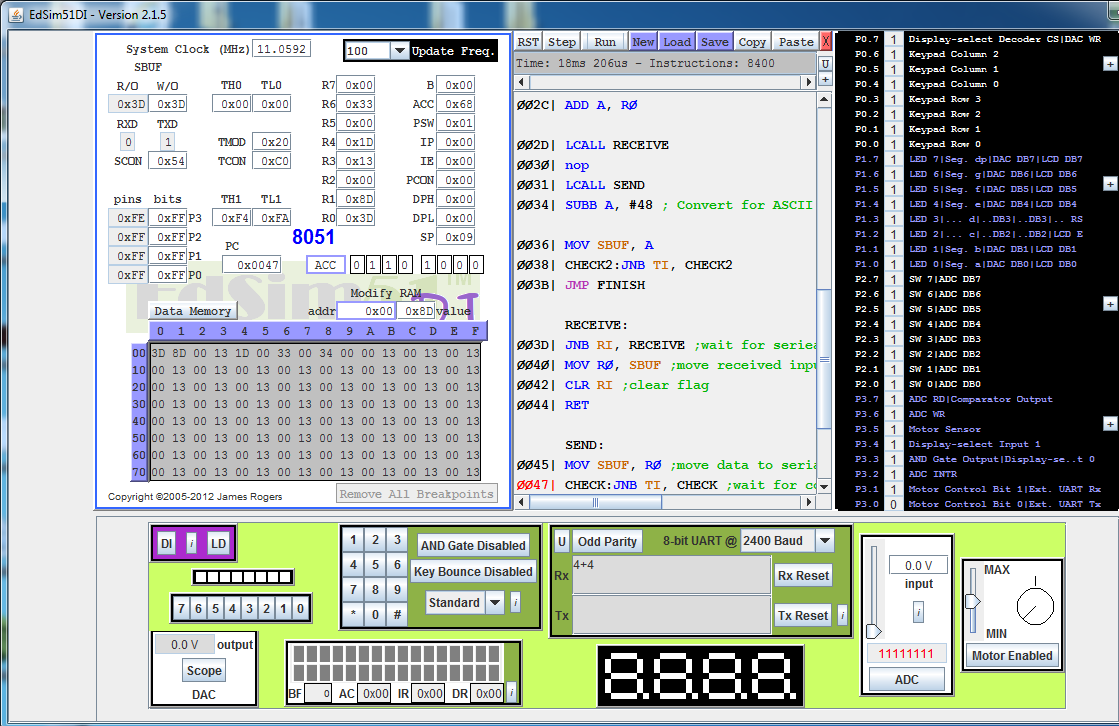
First character is printed to the text console, 2nd character read



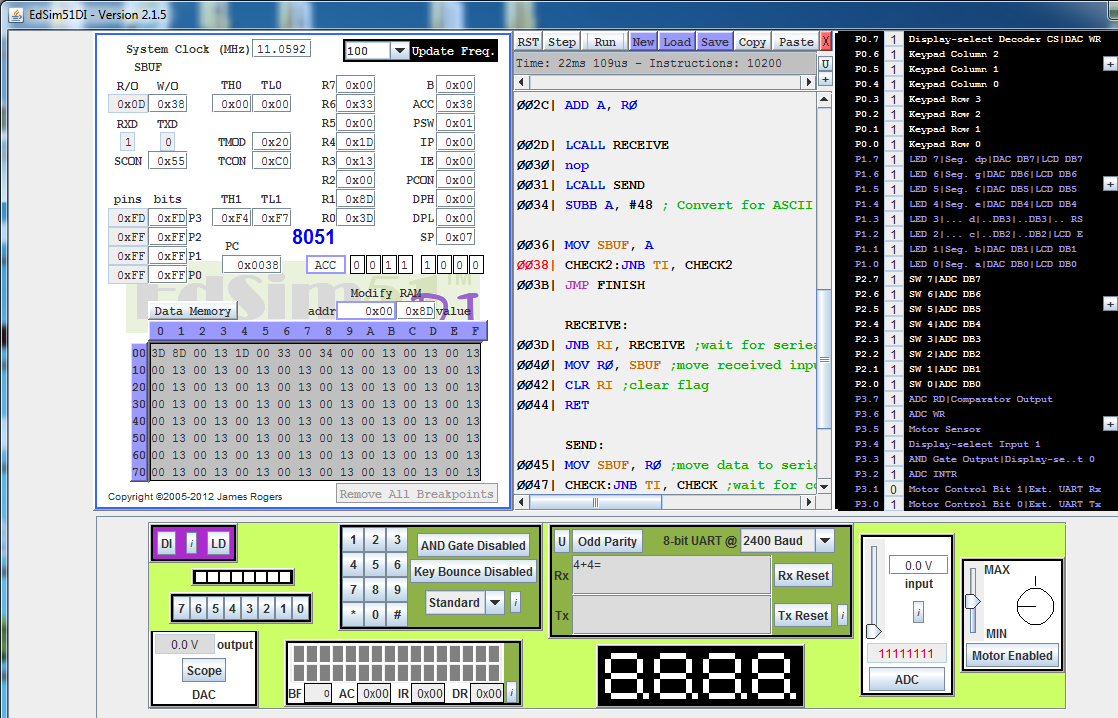
Second character is printed to the text console, third character read



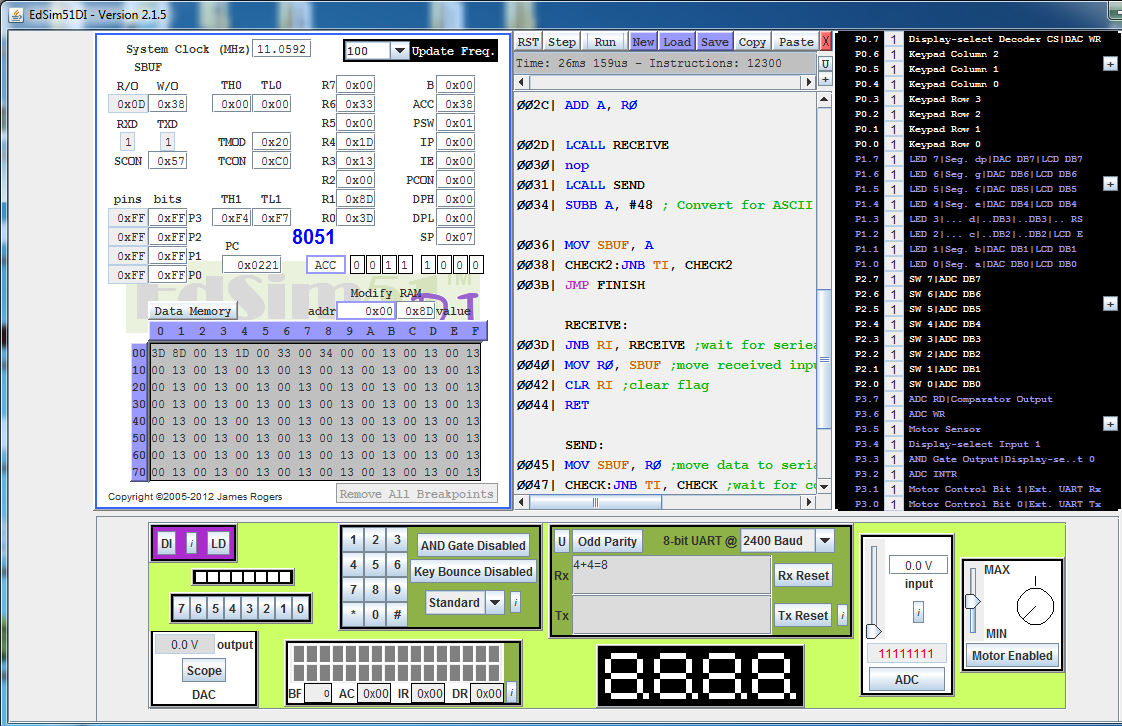
3rd character printed, last character read



Print last character to console



Print the sum to the console



**Conclusion:**

This lab taught how to manipulate the timer registers to change the baud rate. Not only is the actual code used important, but making sure hardware and software are synchronized correctly. Even with the same code, the results can change if hardware is not synched correctly.

**Code:**

Org 0h

CLR SM0

SETB SM1 ;put serial port in 8-bit UART mode

MOV TMOD, #20h ;put timer 1 in 8-bit auto-reload interval timing mode

MOV TH1, #0F4h ;set baud rate to 2400

MOV Tl1, #244 ;put same value in low byte 8-bit auto reload timer

SETB TR1 ;start timer

SETB REN ;enable receiver

CLR RI

LCALL RECEIVE

nop

LCALL SEND

CLR A

MOV A, R0 ; Move first number to the accumulator

LCALL RECEIVE

nop

LCALL SEND

nop

LCALL RECEIVE

nop

LCALL SEND

nop

ADD A, R0 ; Add second number to first

LCALL RECEIVE

nop

LCALL SEND

SUBB A, #48 ; Convert for ASCII Code

MOV SBUF, A ; Output sum to Text module

CHECK2:JNB TI, CHECK2 ; Make sure character is sent before program finishes.

JMP FINISH

RECEIVE:

JNB RI, RECEIVE ;wait for serieal receiver flag to be 1

MOV R0, SBUF ;move received input char from serial input buffer to reg

CLR RI ;clear flag

RET

SEND:

MOV SBUF, R0 ;move data to serial output buffer to be transmitted

CHECK:JNB TI, CHECK ;wait for completion of translatio

CLR TI

RET

FINISH:

nop

End