

**İ.T.Ü.**  
**Faculty of Computer and Informatics**  
**Computer Engineering**



# **MICROCOMPUTER LAB REPORT**

**Lab No** : 08  
**Lab Date** : 21.11.2013  
**Group** : B09  
**Group Members** : 040100014 Teoman Turan  
040100018 Mustafa Durmuş  
040100117 Tuğrul Yatağan  
040100124 Emre Gökrem

**Research Assistant** : Mustafa Ersen

## 1. THE AIM/CONTENT of THE EXPERIMENT

This experiment contains another application of serial communication mechanism of CSM12C32. We have implemented the bubble sort algorithm.

## 2. EQUIPMENT

To write and compile the code on CodeWarrior, then to use PUTTY, one of the laboratory's desktop computers has been used. CSM12CS32 Kit has been connected to that computer. After the compilation of the code and closing CodeWarrior, the kit has been reset, then Putty has been run.

## 3. EXPERIMENT

The following code implements the bubble sort algorithm on CSM12C32:

```
#include <hidef.h> /* common defines and macros */
#include <mc9s12c32.h> /* derivative information */
#pragma LINK_INFO DERIVATIVE "mc9s12c32"

int numbers[10]; // array for keeping numbers to be sorted
int count_digit; // digit number of one number for UART process
int count = 0; // number of numbers to be sorted
int tempc = 0; // actual digit number of one number

char message[10] = {'S','o','r','t','i','n','g','.','.','.'};
unsigned char temparray[3]; //temporary array for keeping as charset
unsigned char tempchar; //temporary array for digit keeping as char
int temp; // keeping number as temporary integer
int j; // keeping digit as temporary integer
```

```

void sci_init(void)
{
    SCIBDL = 0x34; /*Configure baud rate 9600 bps*/
    SCIBDH = 0x00; /*an SCI clock modulo of 8MHz*/
    SCICR1 = 0x00; /*8 data bits, no parity*/
    SCICR2 = 0x0C; /*Enable Tx, Rx Bits*/
}

void send_char(unsigned char value)
{
    while(!(SCISR1 & 0x40)); // While serial is not busy
    SCIDRL = value; // sent character to data register
}

void print_string(unsigned char * ptr)
{
    int i;
    for(i=count_digit; i>0; i--) // Counting strings characters
        send_char(ptr[i-1]); // Characters are sent one by one
    count_digit = 0;
}

```

```
void BubbleSort() // This function implements the bubble sort
{
    int i;
    int swapped = 1;
    int tempswap; // Temporary key value for swapping operation

    while(swapped == 1) // While swapping
    {
        swapped = 0;
        for(i=0; i<9 ; i++) // 10 numbers will be investigated
        {
            // If a number is smaller than its preceding one
            if(numbers[i+1] < numbers[i])
            {
                // Swapping operation
                tempswap = numbers[i+1];
                numbers[i+1] = numbers[i];
                numbers[i] = tempswap;
                swapped = 1;
            }
        }
    }
}
```

```

void PrintAll() // This function prints numbers
{
    count_digit = 0;
    for(j=0 ; j<10 ; j++) // There are 10 numbers to be sorted.
    {
        temp = numbers[j];
        while(temp > 0)
        {
            temparray[count_digit] = (temp%10) + 0x30;
            // convert first integer digit to ascii char
            count_digit++;
            temp = temp/10; // go on left digit by dividing 10
        }

        print_string(temparray); // sent back to user
        send_char('\n');
        send_char('\r');
    }
}

```

```

void main(void)
{
    sci_init(); // Setup registers
    while(count<10) // while 10 numbers are coming
    {
        if(SCISR1 & 0x20) // If reception flag is set
        {
            if(SCIDRL == 0x0D) // if enter is pressed
            {
                // Enter character
                send_char('\n');
                send_char('\r');

                // Creating numbers from concatenating their digits
                if(tempc == 3) // Consisting of 3 digits
                // build number from character one digit by one digit
                numbers[count] = (temparray[0] - 0x30) *100 +
                (temparray[1] -0x30) *10 + temparray[2] - 0x30;
                // (ABC) = (100*A) + (10*B) + C

                else if(tempc == 2) // Consisting of 2 digits
                // build number from character one digit by one digit
                numbers[count] = (temparray[0] - 0x30)
                *10 + temparray[1] -0x30;
                // (AB) = 10*A + C

                else if (tempc == 1) // Single digit
                numbers[count] = temparray[0] - 0x30;
                // (A) = A

                count++;
                tempc=0;
            }
        }
    }
}

```

```

        if(count == 10) // if 10th number is entered
        {
            int i = 0;
            for (i; i < 10; i++ )
            {
                // send "sorting" message
                send_char(message[i]);
            }

            send_char('\n');
            send_char('\r');

            BubbleSort(); // Numbers are sorted.
            PrintAll(); // Numbers are printed.
            for(;;); // infinite loop for halting
        }
    }

else
{
    // send chars to data register
    send_char(SCIDRL);
    // read chars from data register
    temparray[tempc++] = SCIDRL;
    // Number of digits is increased.
}

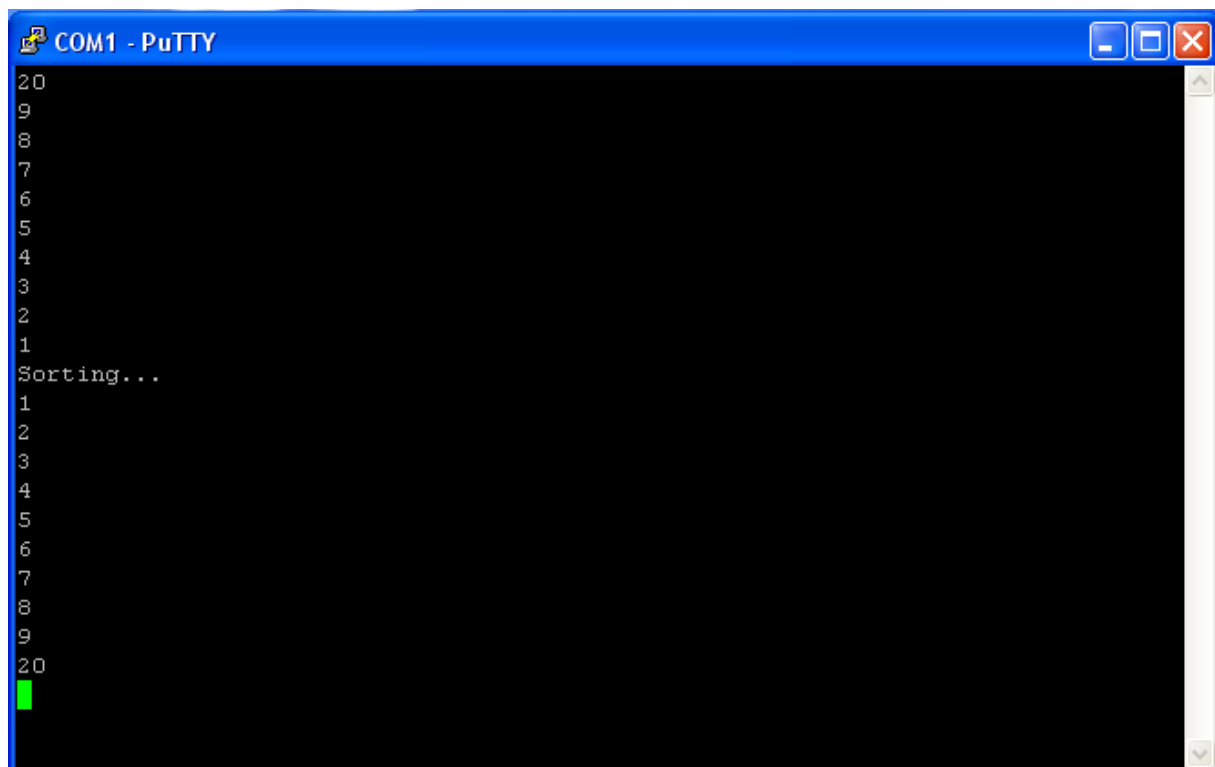
}

}

```

This program receives numbers via PUTTY. Each number can be between 0 and 255. Each number is typed on PUTTY. When *enter* is pressed after a number is typed, that number is saved into an array on the microcontroller. After 10 numbers have been entered, they are sorted according to the bubble sort algorithm in ascending order. These sorted numbers are then printed on PUTTY.

Here is a screenshot demonstrating our test result for this algorithm:



```
COM1 - PuTTY
20
9
8
7
6
5
4
3
2
1
Sorting...
1
2
3
4
5
6
7
8
9
20
█
```