REAL LIFE CASE STUDIES FOR ENCRYPTION AND DECRYPTION

1. Simple Exercises

Best way of learning is starting with simple exercises.

Serial communication is required in number of case studies. Section 11.3 discussed the case study of network data transmission. Example 4.1 gave an example of functions for decrypting received data and encrypting again for retransmission. A secret key is used to decipher and encrypt the data between two ends.

Case study shows how to decrypt and encrypt serial received characters in an 8051 based system.

2. Case Study─ Decryption and Encryption during UART mode Serial Communication

Objective of the case study is to learn how a system does decryption and encryption serial communication in UART mode (Section 3.2.3) with a Computer COM port (Section 3.2.2) using an 8051 based system serial port (Section 2.1.6).

2.1 Requirements

Assume that there is COM port in a PC with RS232C signals (Section 3.2.2) and the system has 8051 microcontroller with serial interface (SI) port (Section 3.2.5). Requirements of the serial communication from the system to computer can be understood through a requirement table given in Table 1.

Table 1

Requirements of the serial communication from the system to PC

Requirement	Description
Purpose	Serial communication from the system to PC RS232C COM port
User Inputs	From keyboard the secret characters (password) for encryption and then decryption
Events	(i) When 8 is pressed the program start for serial communication and encryption for sending the programmed set of secret characters (password) (ii) When 9 is pressed the decrypted characters are displayed
System outputs	System serial output of characters after encryption from the system for display on the PC and display encrypted data if E is pressed and decrypted data if D is pressed
Functions of the system	Whenever the program starts, serial interface port of 8051 based system is sent outputs from COM port when the secret characters keyed in at the keyboard of PC. The system transmits back the decrypted characters.
Design metrics	(i) Use 8051 microcontroller based system (ii) System has interfacing circuit for connection SI port pins (pins 0 and 3 of port P3, P3^0 and P3^1) to the computer COM port RxD and TxD RS232C voltage level pins) (iii) 9600 baud rate UART mode 1 communcation (iv) encryption using a given algorithm defined by encryption ( ) function (v) decryption using a given algorithm defined by decryption ( ) function
Test and validation conditions	Each programmed set of characters tested for serial data encryption and decryption and results displayed at the PC hyper terminal.

2.2 Hardware Architecture

Figure 1 shows the system architecture. A PC is used as development host. (Chapter 13 Section 13.2) The PC connects to the system using RS232C COM1 port, operated at 9600 baud. The system has P89C51RD2 microcontroller of 8051 family. Prefix 89 means, flash memory is internal ROM. Total flash memory (internal plus external) for the program memory is 64KB. The system includes the followings: (i) A reset switch to restart the system after erasing the previous program and downloading a program into flash from PC (ii) An LED for power ON state (iii) interfacing circuit for connection SI port pins (pins 0 and 3 of port P3, P3^0 and P3^1) to the computer COM port RxD and TxD RS232C voltage level pins).

2.3 IDE Software

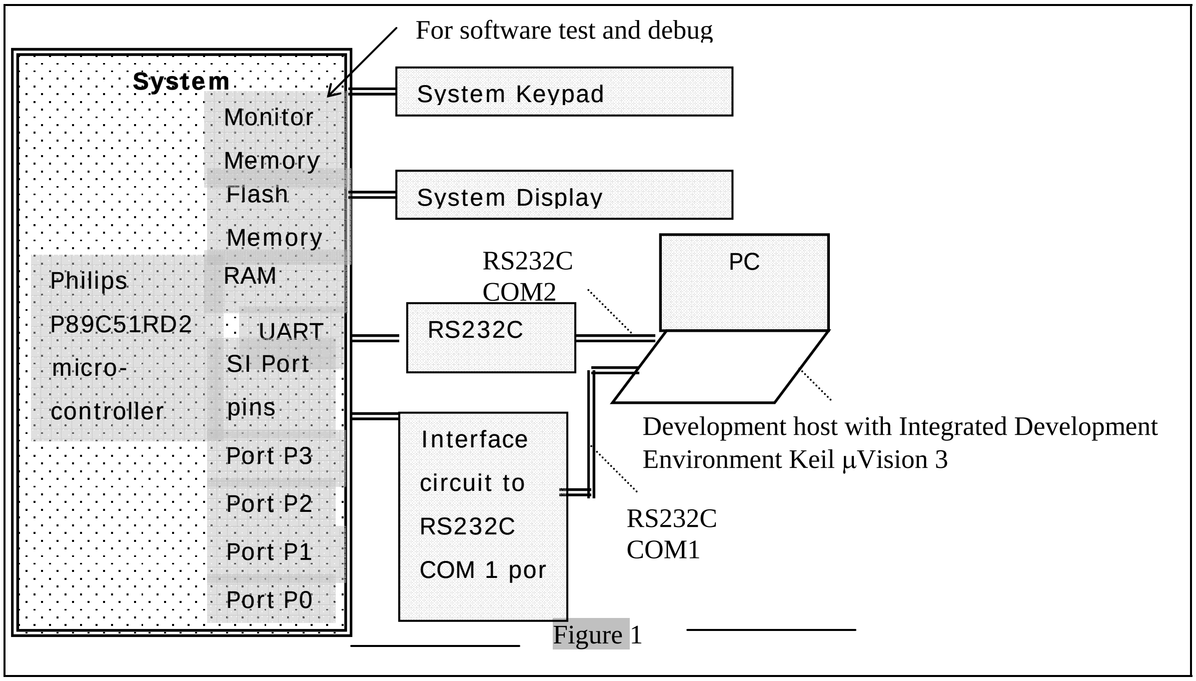

For software test and debug
System
Monitor Memory
Flash Memory
RAM
Philips P89C51RD2 micro-controller
UART
SI Port pins
Port P3
Port P2
Port P1
Port P0
System Keypad
System Display
RS232C COM2
PC
RS232C
Interface circuit to RS232C COM 1 por
Development host with Integrated Development Environment Keil μVision 3
RS232C COM1

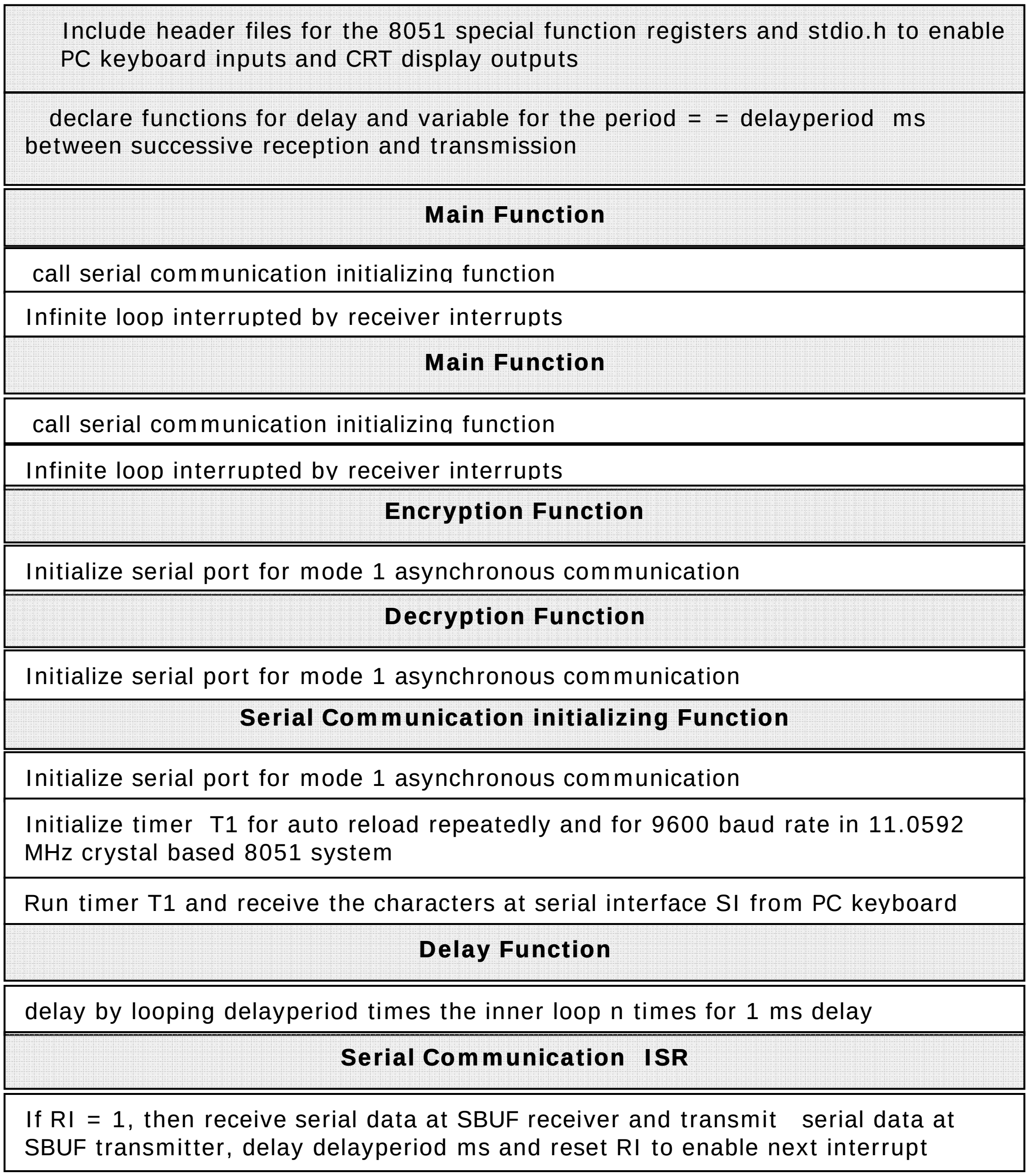
Figure 1

Software μVision3 integrated development environment (IDE) of Keil Software. μVision3 runs on the PC. μVision3 has provision for system microcontroller device selection. μVision3 has provision for enabling user to write a program in C and then check for errors and compile for the 8051 based system. μVision3 also has simulator and debugger.

When COM port receives the characters at PC, then and display the received characters on a hyper terminal (a window set to display characters received at the COM port) at the serial communication simulator in software μVision3. The IDE also enables preparation of hex file for the specified system. It enables user to download the hex file having final ROM mage into the system flash memory.

2.4 Programs

Figure 2 shows the steps in the program. Following program enables serial



Include header files for the 8051 special function registers and stdio.h to enable PC keyboard inputs and CRT display outputs

declare functions for delay and variable for the period = = delayperiod ms between successive reception and transmission

Main Function

call serial communication initializing function

Infinite loop interrupted by receiver interrupts

Main Function

call serial communication initializing function

Infinite loop interrupted by receiver interrupts

Encryption Function

Initialize serial port for mode 1 asynchronous communication

Decryption Function

Initialize serial port for mode 1 asynchronous communication

Serial Communication initializing Function

Initialize serial port for mode 1 asynchronous communication

Initialize timer T1 for auto reload repeatedly and for 9600 baud rate in 11.0592 MHz crystal based 8051 system

Run timer T1 and receive the characters at serial interface SI from PC keyboard

Delay Function

delay by looping delayperiod times the inner loop n times for 1 ms delay

Serial Communication ISR

If RI = 1, then receive serial data at SBUF receiver and transmit serial data at SBUF transmitter, delay delayperiod ms and reset RI to enable next interrupt

Figure 2

Steps in the program Example 1

communication of characters (data) from the system SI port pins RxD and TxD to PC RS232C COM port RxD and TxD pins and display the result on PC hyper terminal.

Example 1

1.

```
 /* Encryption and decryption of serial communication of characters (data) from the system to PC
RS232C COM port */

/*------------------------------------------------------------------------------------------------------------------------
-*/
/* include files to define 8051 special function registers  and stdio*/
# include <stdio.h>
# include <reg51.h>
```

2.

```
/*------------------------------------------------------------------------------------------------------------------------
-*/
# define int n 112
/* n is number of times a loop with no operation runs so that the system spends 1 ms. The value of n =
112 is found by an actual system tests for various values of n starting from 100 */
unsigned int numChars, leftChars;
 /* numChars, a variable for number of characters  received and leftChars, a variable for number of
characters waiting to be received for  the character string charsString */
unsigned char stringSize, serialcode; /* Variables for the ASCII code and  serial character received or
transmitted  at SBUF.  stringSize,is for the size of character string and serialcode is code for encryption
or decryption actionw, which is specified by user */
/* define variables for strings for serial input and output */
unsigned char charsString [ ];      /* Input Character String from keyboard from user*/
unsigned char encryString [ ];       /* Encrypted Character String at SI_ISR by calling encryption ( )
function */
unsigned char decryString [ ];    /* Decrypted Character String by SI_ISR */
unsigned int delayperiod;      /* Declare delay function for delayperiod *
/* declare functions */
void delaydelayperiodms (void); /* declare delay routine for delayperiod  ms delay */
void SI_IntializeMode1 (void); /* serial interface initialization mode 1 and T1 proogrammed for 9600
baud */
void SI_serialdataRecv (void); /* serial data Receiving function */
void encryption ( ); /* Encryption function */
void decryption ( );      /* decryption function */
/*------------------------------------------------------------------------------------------------------------------------
-*/
```

3.

```
/* Hyper terminal Display on PC for user interaction */
```

```
unsigned char enterString [ ] = “\n Please enter the number of secret characters:      ”  /* Input
Character String from keyboard from user*/
unsigned char displChars [ ] = “\n
 “Encryption and decryption of serial communication of characters (data) from the system to PC
RS232C COM port” “
“\n ……………………”
“Please enter ‘8’: Enter secret characters at keyboard for encryption. Maximum 36 characters”
“Please enter ‘9’: Display secret characters after decryption”
```

4.

```
/*-------------------------------------------MAIN FUNCTION   ------------------------------------*/

void main (void)
{
/* call function to initialize ISR for serial port SI of 8051 to mode 1 and baud rate to 9600 using timer
T1*/
SI_IntializeMode1 ();
delayperiod = 1; /* delayperiod = 1 ms */
delaydelayperiodms ( ); /* delay 1 ms */
SI_serialdataRecv (); /* Receive serial data till keyed input character is not null the end of string or all
characters received */
/*------------------------------------------------------------------------------------------------------------------------
-*/
while (1) {

} /* wait for interrupt while looping in infinite loop */
} /* end of the program */
/*------------------------------------------------------------------------------------------------------------------------
-*/
```

5.

```
void SI_IntializeMode1 (void ) /* initialize serial interface port and timer for baud rate */

{
/* Function to initialize serial port SI of 8051 to mode 1 and baud rate to 9600 using timer T1*/
/* 0x50 = 01010000 SCON^7, …, SCO^0 are SM0, SM1, SM2, REN, TB8, RB8, TI and RI,
respectively
SM0= 0 and SM1 = 1 for mode 1, start, 8 data bits and stop bit total 10 UART bits
SM2 = 0
REN = 1, receiver enable
TB8 = 0 as not used  in mode 1
RB8 = 0 as not used  in mode 1
TI = when read initial value of transmitter complete interrupt flag = 0
RI = when read initial value of receiver complete interrupt flag = 0
TI = 1 when write to initialize serial transmitter
RI = 0 when write not to initialize serial receiver
*/
SCON = 0x50;
```

```
/*-------------------------------------------------------------------------------------------------------------------------
-*/
/* Initialize timer T1 to 8-bit mode 2 timer (auto-reload from TH1 into TL0 after each timer overflow
interrupt */
/* 0x20 = 00100000 TMOD^7, …, TMOD^1 are Gate1, C/T1, M1, M0, Gate0, C/T0, M1, M0,
respectively
Gate1= 0 as external pin gate control of timer 1 start not used
C/T = 0 as external pin input (counter) not used
M1, M0 = 10 for mode 2, eight bit timer TL1 with auto-reload from TH1 after each overflow of T1
Gate0, C/T0, M1, M0 = 0, 0, 0 and 0 as T0 not used
*/
TMOD = 0x20;

/* For overflow after each (0x100 – 0xFD) = 3 timer ticks. Mode 1 has clock pre-scaling factor =32.
Time for 3 timer ticks = 3 × 32 × 12/11.0592 µs for 11.0592 MHz crystal in 8051 system. Baud rate =
1000000× 11.0592/ (3 × 32  × 12) = 9600 */
TH1 = 0xFD;

/* Set TCON^6 TR1 to run timer T1 */
TR1 = 1;

/* Enable Serial Interrupts */
ES =1; EA= 1;

/* initialize serial transmitter */
TI = 1;
}
/*-------------------------------------------------------------------------------------------------------------------------
-*/
```

6.

```
/* Serial data receive at SI interface till not null or till all characters received */
void SI_serialdataRecv ()
{
numChars = 0;
while (displChars [ numChars] != ‘\0’) /* Till end of string null character received \0 */
  {SBUF = displChars [ numChars]; /* Receive character at SBUF receiver */
  delayperiod = 1; / *delay 1 ms */
  delay
numChars ++; /* Increment by 1 the variable numChars for number of characters */
  leftChars - - ; /* Decrement by 1 the variable leftChars for number of characters left to be received*/
  if (leftChars = = 0) {
             leftChars = 36; /* respecify the number of left characters for next string */
             SBUF = 0x0d; /* assign ASCII code for the carriage return to SBUF */
            delayperiod = 1;
            delaydelayperiodms ( ); /* delay 1 ms */
          }
  } /* end of while loop */
} /* end of function */
```

6.

```
/* Statements for  encryption function */

void encryption (void)
{
  unsigned int i;
    numChars = 0;
   SBUF = 0x0d; /* Receive character = ASCII code for carriage return */
   delayperiod = 1; /* delay 1 ms*/
   delaydelayperiodms ( ); /*delay delayperiod ms*/
   while (enterString [ ] != ‘\0’ ) /* Till enter string is not null */
        {SBUF = enter [i];
       delayperiod = 1; /* delay 1 ms*/
       delaydelayperiodms ( ); /*delay delayperiod ms*/
       numChars++;      /* Increment i by one as one more character is  received */
         };
    while (RI = = 0) { stringSize = SBUF; /* Receive stringSize from SBUF*/
        };
    RI =0;
    SBUF = stringSize; /* Transmit stringSize from SBUF*/
    delayperiod = 1; /* delay 1 ms*/
    delaydelayperiodms ( ); /*delay delayperiod ms*/
    SBUF= ‘\n’; /* Transmit ASCII code for new line */
    delayperiod = 1; /* delay 1 ms*/
    delaydelayperiodms ( ); /*delay delayperiod ms*/
    SBUF= ‘\n’; /* Transmit ASCII code for new line */
    delayperiod = 1; /* delay 1 ms*/
    delaydelayperiodms ( ); /*delay delayperiod ms*/
    SBUF= ‘\n’; /* Transmit ASCII code for new line */
    delayperiod = 1; /* delay 1 ms*/
    delaydelayperiodms ( ); /*delay delayperiod ms*/

for (numChars =0; i<= stringSize; numChars ++) {
             while  (RI = =0) {charsString [i ]=  SBUF; /* Receive character string for secret characters at
serial SBUF receiver */
              }; /* end of while-loop on serial receiver interrupt at SI */
           delayperiod = 1; /* delay 1 ms*/
           delaydelayperiodms ( ); /*delay delayperiod ms*/
           printf (“\n”);
           leftChars = (stringSize – numChars) + 1;
          if (leftChars > 0) {printf (“ Please enter %d number of characters: \n”, leftChars);};
           printf (“\n”);
           RI = 0;
       } /* end of for-loop */
EA = 1; ES = 1; /*Enable next serial interrupt */
printf (“ Text after Encryption:       ”);
for (i =0; charsString [i ] != ‘\0’;  numChars) {
 encryString [ ] = charsString [i]  + i – 5 } /* During encryption the ASCII code of the received
character was subtracted by 5 and added by the character number i. */

} /* end of for- loop */
delayperiod = 1; /* delay 1 ms*/
delaydelayperiodms ( ); /*delay delayperiod ms*/
printf (“\n”);
}/* end of function  */
/*-------------------------------------------------------------------------------------------------------------------------
-*/
```

7.

```
/* Statements for  decryption function */

void decryption (void)
{
unsigned int i;
printf (“\n ------------------- Text after Decryption ----------------------- ”
    /* Loop numChars times */
    for (i=0; i< numChars; i++) {
          decryString [i ]= encryString [i ] - i + 5} /* During encryption the ASCII code of the received
character was subtracted by 5 and added by the character number i. Therefore, for deciphering the i is
subtracted and 5 is added */
           delayperiod = 1; /* delay 1 ms*/
           delaydelayperiodms ( ); /*delay delayperiod ms*/
           printf (“%c, decryString [i ]); /* Print on hyper terminal deciphered text */
 } /* end of loop */
printf(“\n ------------------- ----------------------------------------------------- ”

}/* end of function  */
/*-------------------------------------------------------------------------------------------------------------------------
-*/
```

8.

```
/* Statements for  delay function for delay of delayperiod ms*/
void delaydelayperiodms (void)
{
  unsigned int i, j;
   /* Loop n  delayperiod times with no operation to spend delayperiod  ms. */
    for (i=0; i< delayperiod; i++) {
           for (j = 0; j < = n; j++) {
            } /* end of inner delay loop */
   } /* end of outer delay loop */
}/* end of function  */
/*-------------------------------------------------------------------------------------------------------------------------
-*/
```

9.

```
/*--------  ISR FUNCTION FOR SERIAL INTERFACE INITIALIZING FUNCTION ----- -------------
*/
```

```
void SI_ISR () interrupt 4  /* declare serial interrupt routine */

{
/* ISR Function , which runs on serial receiver interrupt of serial port SI of 8051 */
/* Check RI flag */
if (RI == 1) {
serialcode = SBUF;
RI =0; /* Reset RI to enable next interrupt */

switch (serialcode) {
case 0x38 encryption ( ); break; /* call encryption function if received Character is ASCII code for 8 */
case 0x39 decryption ( ); break;  /* call decryption function if received Character is ASCII code for 9 */
  }/* end of swicth statement */
delayperiod = 1; /* delay 1 ms*/
delaydelayperiodms ( ); /*delay delayperiod ms*/
         } /* end of if  statement */
} /* end of ISR */
/*---------------------------------------------------------------------------------------------------------------------------
-*/
```

2.5 Method

Install μVision3 integrated development environment (IDE) of Keil Software. Now run μVision3. First create and edit a new project file *EnDecrSerialInOut.C* with C program of Example 1.

Encryption algorithm is that the ASCII code of the character in the secret string of characters is ASCII code of the received character was subtracted by 5 and added by the character number i.

Decryption algorithm is that the ASCII code of the character is recovered from the secret encrypted string of characters by adding by 5 and subtracting the character number i.

Now compile the program. If no errors then a hex file *EnDecrSerialInOut.C.hex* will be created. Now, start the program for flash programming. After erasing the previous program, the new hex file is down loaded. The results are obtained from the program as described in Table 3 Section 2.6.

Now create and edit a new project file *EnDecrSerialInOut.C* with C program of Example 2. Now compile the program. If no errors then a hex file *EnDecrSerialInOut.hex* will be created. Now, start the program for flash programming. After erasing the previous program, the new hex file is down loaded. The results are obtained from the program as described in Table 3 Section 2.6.

2.6 Results

Table 2 gives the results obtained after (i) editing program serialOut.*C* in Example 1, (ii) compiling, (iii) creating and (iv) downloading hex file in the system. The program is started after pressing reset key.

Enter the number of characters in secret string of characters and secret string. Enter the secret string of characters. Check the encryption algorithm that every character is encrypted as per the algorithm. Check the decryption algorithm that every character is same as entered in the beginning.

Repeat the steps in above paragraphs by another string or by modifying pair of statements: encryString [ ] = charsString [i] + i – 5 and decryString [ ] = encryString [i] – i + 5.

Results of Program in Example 1 on PC hyper terminal window

S.No.	Display at Hyper terminal
1	“ Please enter number of characters: 7 s: Please enter 6 number of characters: p: Please enter 5 number of characters: q: Please enter 4 number of characters: r: Please enter 3 number of characters: t: Please enter 2 number of characters:

```
            p: Please enter 1 number of characters

            q:

            Text after Encryption:    nlnpspr

            ------------------- Text after Decryption -----------------------
            spqrtpq
.       .
.       .
.       .
.       .
.       .
.       .
.       .
.       .
.       .
.       .
.       .
.       .
.       .
```