SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

19ECB211 – MICROCONTROLLER PROGRAMMING & INTERFACING

II YEAR IV SEM

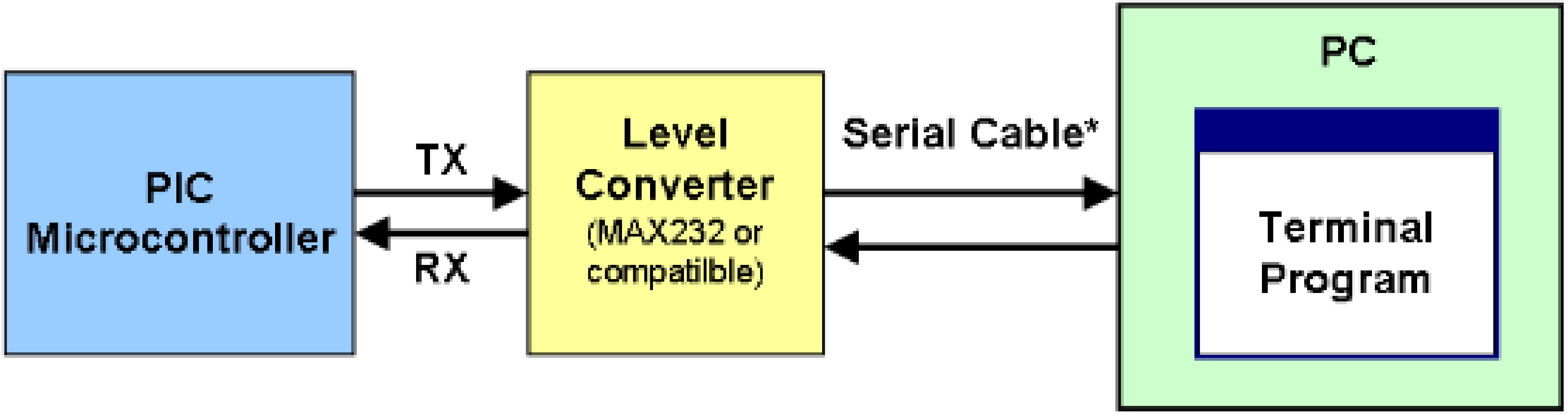
UNIT II – PIC TIMER, SERIAL PORT AND INTERRUPT

TOPIC 4 – PIC connection to RS232

RS232

- Built in Universal Synchronous Asynchronous Receiver Transmitter (USART)
- Allows to communicate using RS232, RS422 and RS485 protocols
- 5V logic level receive and transmit signals of the PIC are converted to RS232 levels by a MAX232 device
- Baud rates are generated by a dividing down the system clock
- USART receive and transmit pins are c7 and c6

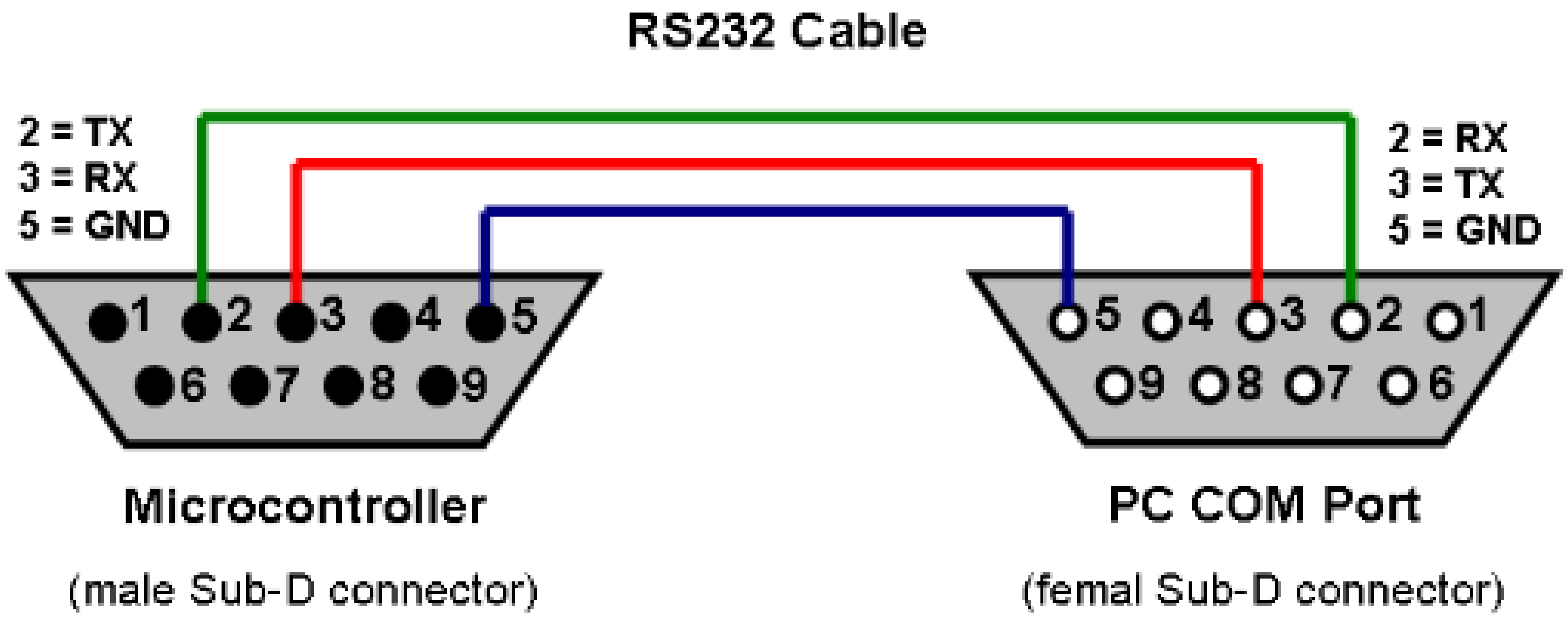
Block Diagram



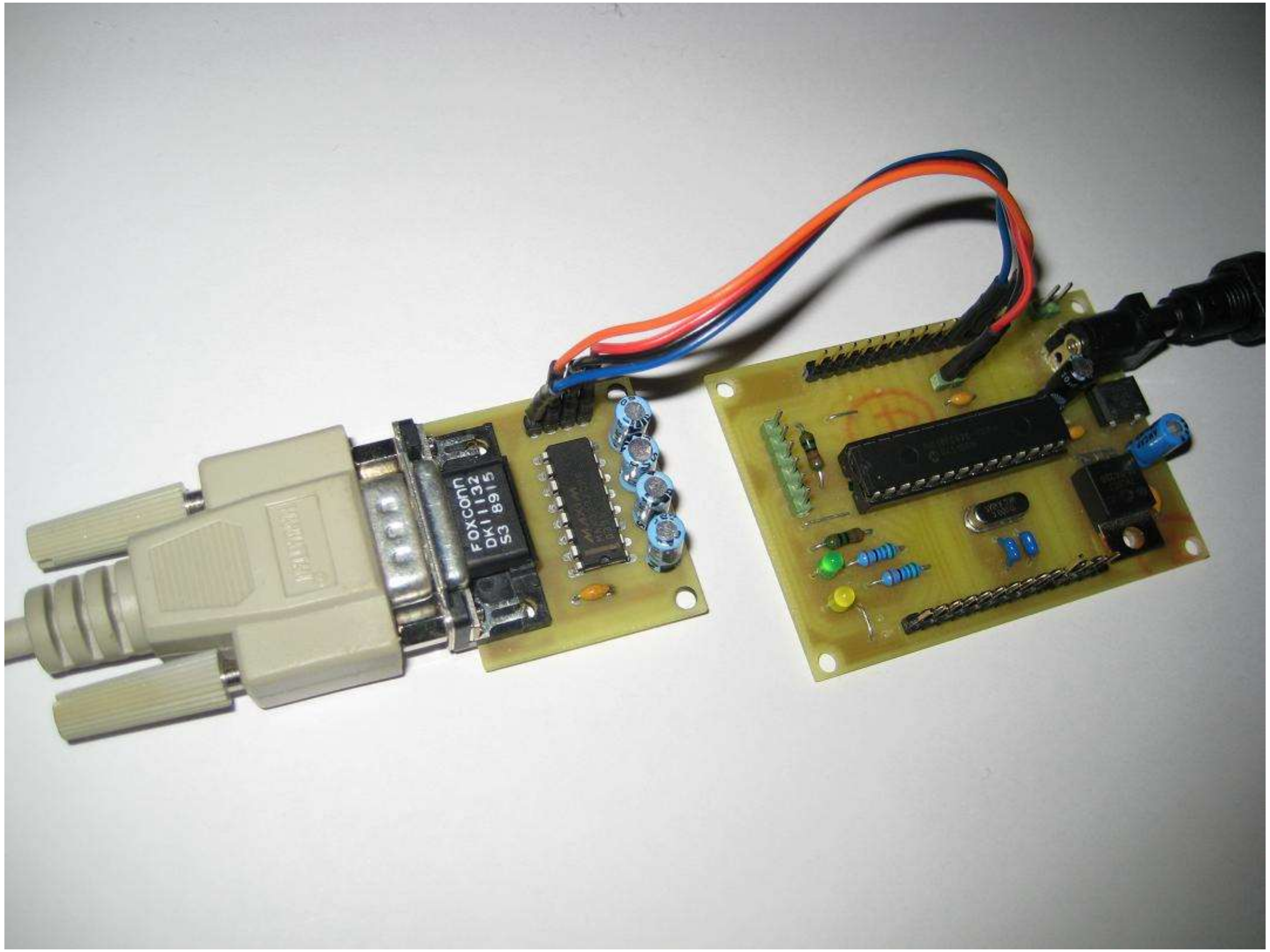
- For serial communication the line used to transmit data is called TX and the line used to receive data is called RX.
- The level converter is required to translate the voltage level of the microntroller to RS232 voltage level.
- The microntroller operates at TTL level (0V = logic 0, +5V logic 1) whereas RS232 uses around +/-12V. A very famous RS232 level converter is the MAX232 chip.

Hardware

- PIC16F877A
- Crystal Oscillator
- MAX232 chip (for Level Convertor)
- Capacitors



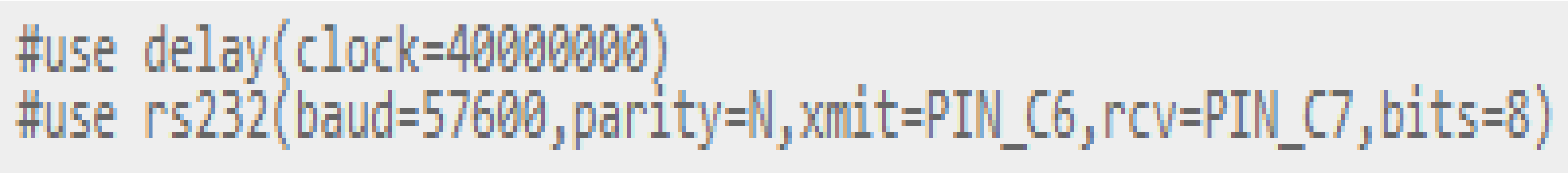
Hardware Picture



Software

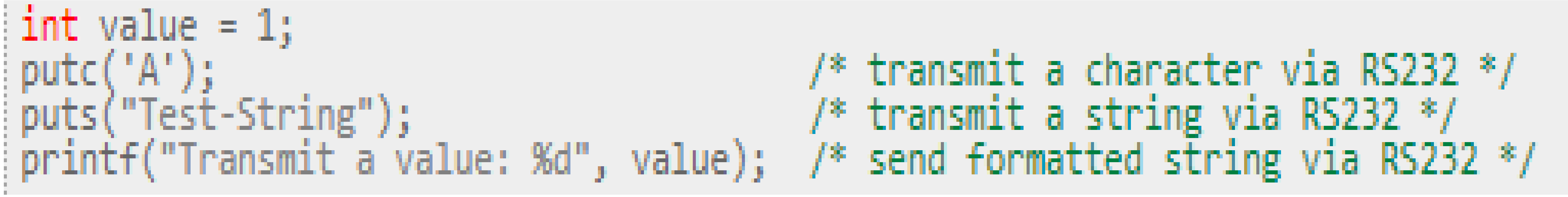
RS232 communication with CCS C compiler

Code lines which are required to setup the UART for RS232 communication.



```
#use delay(clock=40000000)
#use rs232(baud=57600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
```

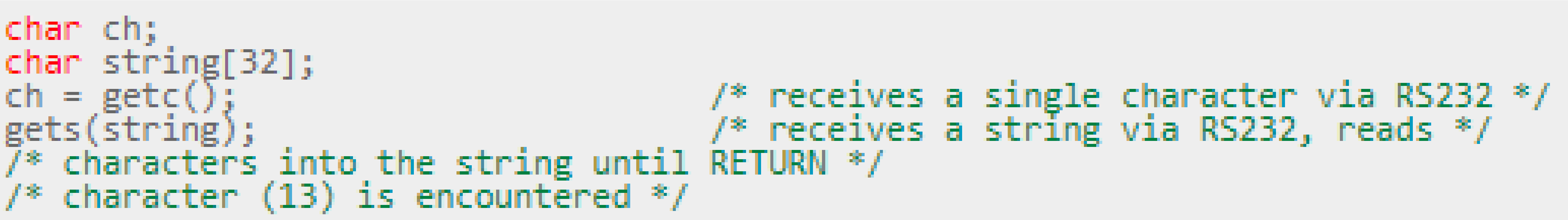
To transmit data



```
int value = 1;
putc('A');                             /* transmit a character via RS232 */
puts("Test-String");                   /* transmit a string via RS232 */
printf("Transmit a value: %d", value); /* send formatted string via RS232 */
```

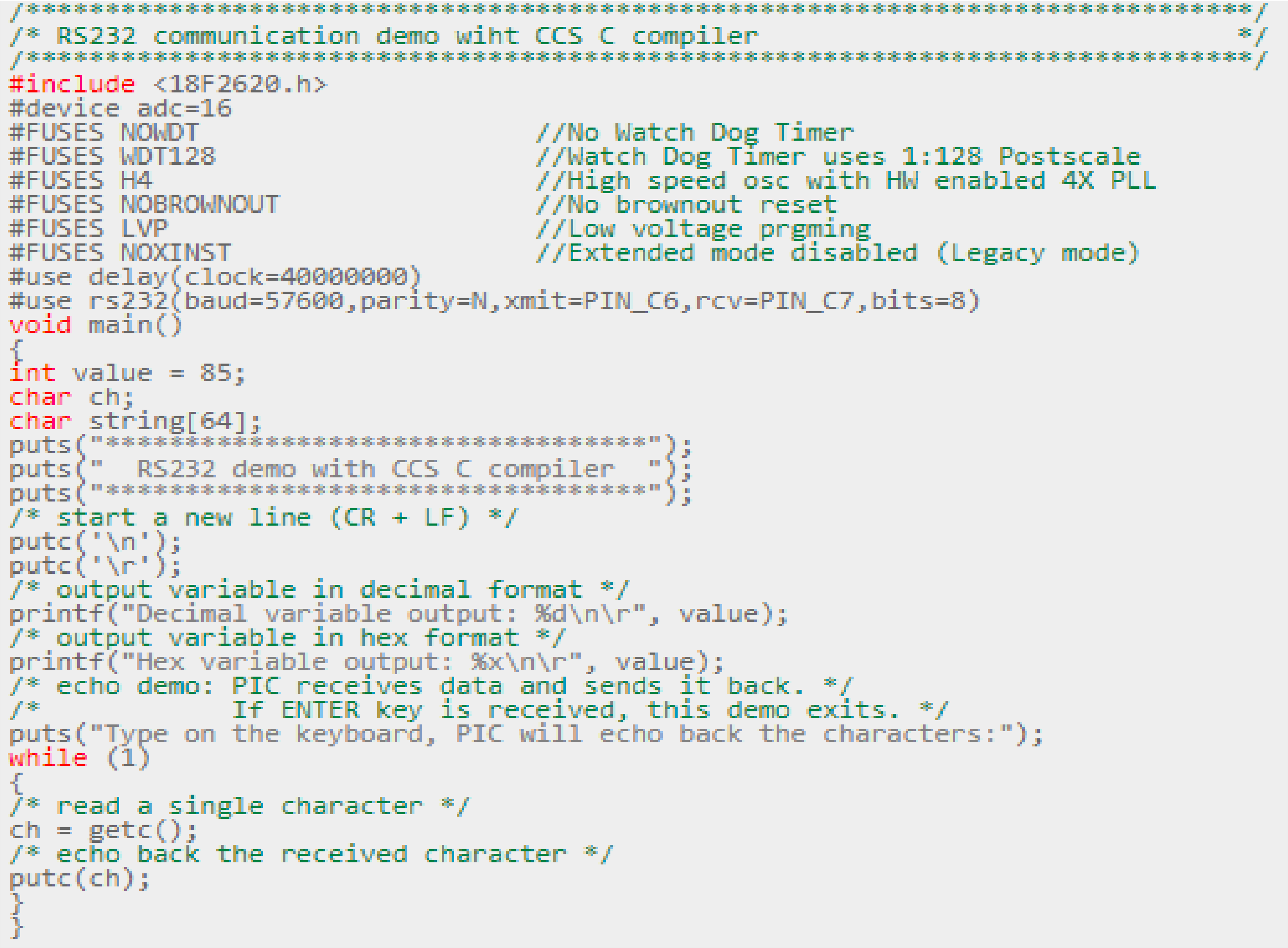
Software

Receive Data



```
char ch;
char string[32];
ch = getc();                        /* receives a single character via RS232 */
gets(string);                       /* receives a string via RS232, reads */
/* characters into the string until RETURN */
/* character (13) is encountered */
```

Program



```
/*****************************************************************************/
/* RS232 communication demo wiht CCS C compiler                              */
/*****************************************************************************/
#include <18F2620.h>
#device adc=16
#FUSES NOWDT                     //No Watch Dog Timer
#FUSES WDT128                    //Watch Dog Timer uses 1:128 Postscale
#FUSES H4                        //High speed osc with HW enabled 4X PLL
#FUSES NOBROWNOUT                //No brownout reset
#FUSES LVP                       //Low voltage prgming
#FUSES NOXINST                   //Extended mode disabled (Legacy mode)
#use delay(clock=40000000)
#use rs232(baud=57600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
void main()
{
int value = 85;
char ch;
char string[64];
puts("**********************************");
puts("  RS232 demo with CCS C compiler  ");
puts("**********************************");
/* start a new line (CR + LF) */
putc('\n');
putc('\r');
/* output variable in decimal format */
printf("Decimal variable output: %d\n\r", value);
/* output variable in hex format */
printf("Hex variable output: %x\n\r", value);
/* echo demo: PIC receives data and sends it back. */
/*            If ENTER key is received, this demo exits. */
puts("Type on the keyboard, PIC will echo back the characters:");
while (1)
{
/* read a single character */
ch = getc();
/* echo back the received character */
putc(ch);
}
}
```

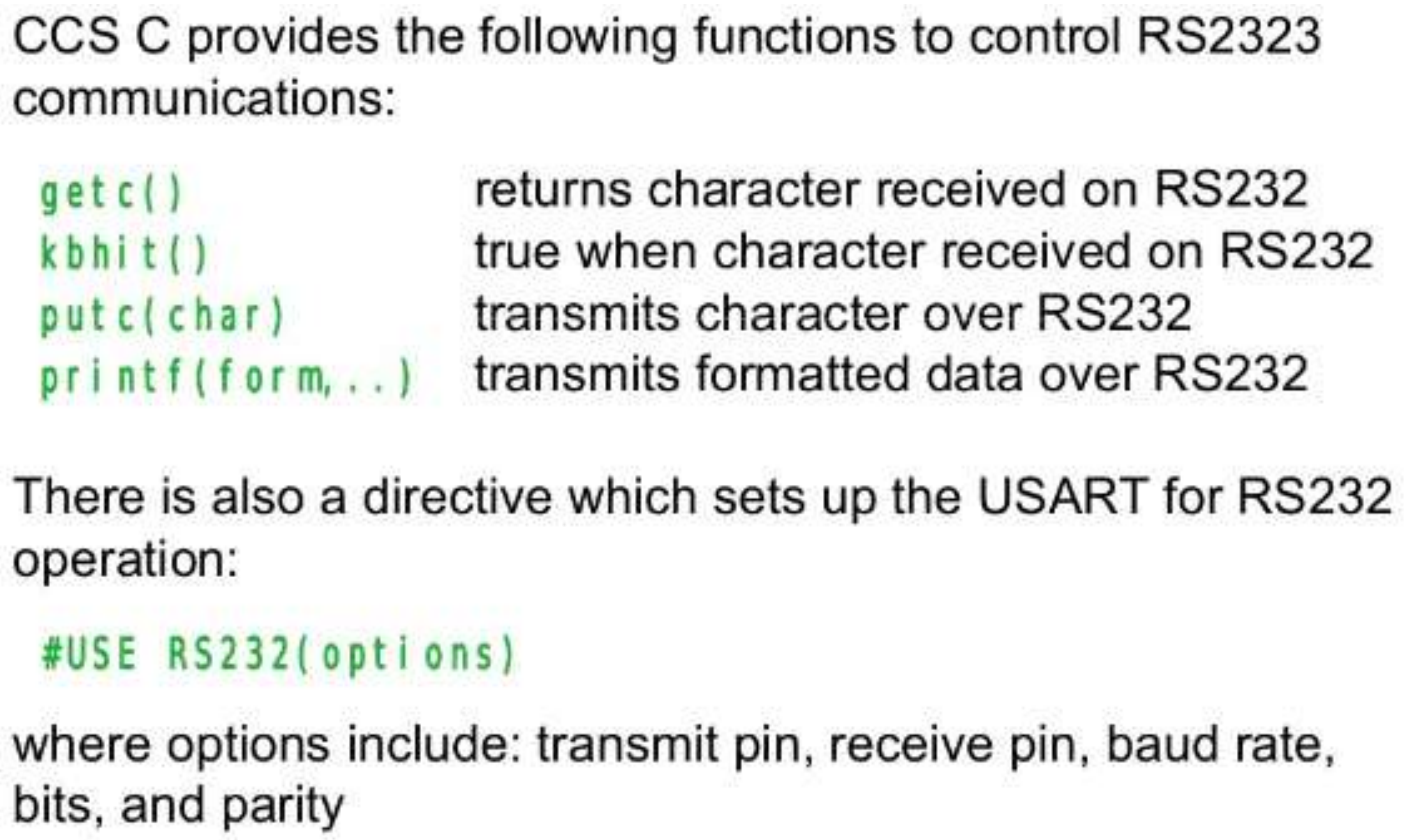
RS232

- Built in Universal Synchronous Asynchronous Receiver Transmitter (USART)
- Allows to communicate using RS232, RS422 and RS485 protocols
- 5V logic level receive and transmit signals of the PIC are converted to RS232 levels by a MAX232 device
- Baud rates are generated by a dividing down the system clock
- USART receive and transmit pins are c7 and c6

RS232

- RS232 is a standard for a serial communication interface which allows to send and receive data via at least three wires.
- It is possible to setup a connection between a microcontroller and a PC (via PC's COM port) or between two microcontrollers.
- RS232 interface can be used for many purposes like sending commands from a PC to a microcontroller, send debug information from a micontroller to a terminal

RS232



CCS C provides the following functions to control RS2323 communications:

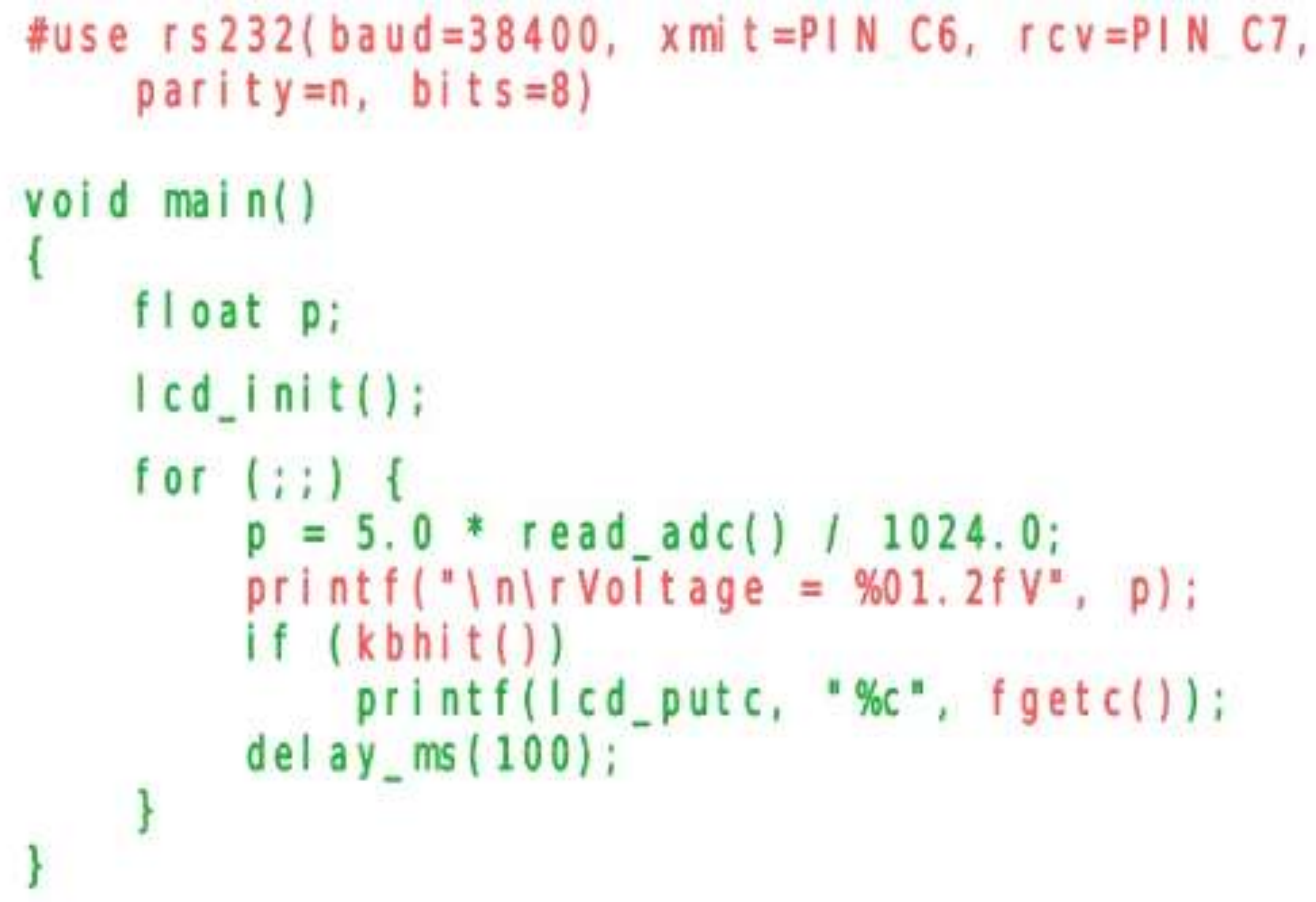
`getc()`	returns character received on RS232
`kbhit()`	true when character received on RS232
`putc(char)`	transmits character over RS232
`printf(form,..)`	transmits formatted data over RS232

There is also a directive which sets up the USART for RS232 operation:

```
#USE RS232(options)
```

where options include: transmit pin, receive pin, baud rate, bits, and parity

RS232



```
#use rs232(baud=38400, xmit=PIN_C6, rcv=PIN_C7,
    parity=n, bits=8)

void main()
{
    float p;

    lcd_init();

    for (;;) {
        p = 5.0 * read_adc() / 1024.0;
        printf("\n\rVoltage = %01.2fV", p);
        if (kbhit())
            printf(lcd_putc, "%c", fgetc());
        delay_ms(100);
    }
}
```

References

https://www.embedded.com/the-evolution-of-embedded-devices-addressing-complex-design-challenges/

http://iamtechnical.com/org-origin-end-list-include-config-radix-directives

https://www.electronicspecifier.com/products/design-automation/embedded-systems-the-evolution-of-embedded-system-design

Mazidi M. A., McKinlay R. D., Causey D. “PIC Microcontroller And Embedded Systems” Pearson Education International, 2008(Unit I,II,III, IV & V)

Thank You