

C Sample Code

THIS CODE WAS WRITEN FOR A PIC 18f14k50 using the Mikro C pro Compiler.
Change the sample code to fit your processor
this sample code will work for a pH/OPR/Dissolved Oxygen Stamp
you must enable the following libraries:
C_String|UART|Software UART

```

char input[20];           // this char array is used to store the RX input data
volatile bit rx_event;   //this is a bit var to signal an rx receive event

void interrupt()         // all interrupts are dealt with here

{
    if (PIR1.RCIF) {     // if we get an interrupt from the RX pin
        UART1_Read_Text(input, "\r", 20); // we read the incoming chars until we receive
        rx_event=1;      // a <CR> ( "\r") or 20 chars
    }                    // set rx_event=1 so we know data is holding

void main() {

short Start_up=0;       //used to control the start-up sequence
short i;                //counter

    //set up system clock to run at 8 MHz
    OSCCON.b6=1;
    OSCCON.b5=1;
    OSCCON.b4=0;
    OSCCON.b1=1;

    //set up hardware uart system
    UART1_Init(38400);
    ANSELH.ANS11 =0;    //turn off analog functions on the RX pin
    IOCB.IOCB7=0;       //interrupt on change pin, disabled
    TRISB.TRISB7=0;     //config TX as output
    TRISB.TRISB5=1;     //config RX as input

```

```

Soft_UART_Init(&PORTB, 4, 6, 38400, 0);    // Initialize Soft UART at 38400 bps
    ^.....port to be used
    ^.....RX pin (we are to going to use the RX function)
    ^.....TX pin (pin 11)
    ^.....baud rate 38400
    ^.....rs-232 data is NOT inverted

```

```
//CONFIG UART INTERUPTERS
```

```

INTCON.PEIE = 1; //peripheral interrupt enable
PIE1.RCIE = 1;  //Receive char Interrupt Enable bit
PIR1.RCIF = 0;  //Receive char Interrupt flag- reset to 0
INTCON.GIE = 1; //global interrupt enable
rx_event=0;     //initialize rx_evnet to = 0

```

```

delay_ms(1000);           //wait one sec for the stamp to stabilize

if(Start_up==0){         //when the program firsts starts it will
    for(l = 1; l<=3;i++){ //flash on / off the stamps led
        uart1_write_text("L0"); // "L0" = led's off
        uart1_write(13);        //<CR>
        delay_ms(1000);         //wait one sec

        uart1_write_text("L1"); // "L1" = led's on
        uart1_write(13);        //<CR>
        delay_ms(1000);         //wait one sec
    }
    Start_up = 1;             //by setting Start_up to 1, we stop the leds from flashing on/off
}

```

```
delay_ms(1000);  
uart1_write_text("c");  
uart1_write(13);  
delay_ms(500);  
  
while(1){  
  short len=0;  
  
  if(rx_event){  
    rx_event=0;  
    len = strlen(input);  
    for(i=0;i<len;i++){  
      Soft_UART_Write(input[i]);  
    }  
    Soft_UART_Write(13);  
  }  
}  
}
```

*//wait one sec for the stamp to stabilize after the led flashing
//the command "c" will tell the stamp to take continues readings
//<CR>*

*//reset rx_event to 0
//we need to know the length of the string we just received
from the stamp is
//we now loop through each char of the char array "input"
//now we output each char through the soft serial port
// when we finish outputting the data, we end with a <CR>*