

FINAL CODE

DOCUMENT _ 01

```
char dataRead;

void main() {
    TRISB = 0x00;
    portb = 0x00;

    UART1_Init(9600);
    Delay_ms(100);

    while(1)
    {
        UART1_Write_Text("Press 1 for Red LED, 2 for Yellow LED and 3 for Green LED \r\n");while (1) { //
        Endless loop

        if (UART1_Data_Ready()) { // If data is received,
        dataRead = UART1_Read(); // read the received data

        switch(dataRead){

        case '1':
            portb.f0=1;
            break;

        case '2':
            portb.f1=1;

            break;

        case '3':
            portb.f2=1;
```

```
break;
default:
    portb=0x00;
break;

}
}
}

}}
```

DOCUMENT _ 02

```
_main:

;Test.c,2 ::          void main() {
;Test.c,4 ::          TRISB = 0X00;
                    CLRF      TRISB+0
;Test.c,5 ::          PORTB = 0X00;
                    CLRF      PORTB+0
;Test.c,7 ::          UART1_Init(9600);
                    MOVLW     129
                    MOVWF     SPBRG+0
                    BSF       TXSTA+0, 2
                    CALL      _UART1_Init+0
;Test.c,8 ::          Delay_ms(100);
                    MOVLW     3
                    MOVWF     R11+0
                    MOVLW     138
                    MOVWF     R12+0
                    MOVLW     85
                    MOVWF     R13+0
L_main0:
                    DECFSZ    R13+0, 1
                    GOTO      L_main0
                    DECFSZ    R12+0, 1
                    GOTO      L_main0
                    DECFSZ    R11+0, 1
                    GOTO      L_main0
                    NOP
                    NOP
;Test.c,9 ::          UART1_Write_Text("Press 1 for Red, 2 for Yellow
and 3 for Green");
                    MOVLW     ?lstr1_Test+0
                    MOVWF     FARG_UART1_Write_Text_uart_text+0
                    CALL      _UART1_Write_Text+0
;Test.c,10 ::         UART1_Write(10);
                    MOVLW     10
                    MOVWF     FARG_UART1_Write_data_+0
                    CALL      _UART1_Write+0
;Test.c,11 ::         UART1_Write(13);
                    MOVLW     13
                    MOVWF     FARG_UART1_Write_data_+0
                    CALL      _UART1_Write+0
;Test.c,13 ::         while(1)
L_main1:
;Test.c,15 ::         if(UART1_Data_Ready()==1)
                    CALL      _UART1_Data_Ready+0
                    MOVF      R0+0, 0
                    XORLW     1
                    BTFSS     STATUS+0, 2
                    GOTO      L_main3
;Test.c,17 ::         dataRead = UART1_Read();
                    CALL      _UART1_Read+0
                    MOVF      R0+0, 0
                    MOVWF     _dataRead+0
```

```

;Test.c,18 ::      switch(dataRead)
      GOTO      L_main4
;Test.c,20 ::      case '1':
L_main6:
;Test.c,21 ::      portb.f0=1;
      BSF      PORTB+0, 0
;Test.c,22 ::      portb.f1=0;
      BCF      PORTB+0, 1
;Test.c,23 ::      portb.f2=0;
      BCF      PORTB+0, 2
;Test.c,24 ::      break;
      GOTO      L_main5
;Test.c,26 ::      case '2':
L_main7:
;Test.c,27 ::      portb.f0=0;
      BCF      PORTB+0, 0
;Test.c,28 ::      portb.f1=1;
      BSF      PORTB+0, 1
;Test.c,29 ::      portb.f2=0;
      BCF      PORTB+0, 2
;Test.c,30 ::      break;
      GOTO      L_main5
;Test.c,32 ::      case '3':
L_main8:
;Test.c,33 ::      portb.f0=0;
      BCF      PORTB+0, 0
;Test.c,34 ::      portb.f1=0;
      BCF      PORTB+0, 1
;Test.c,35 ::      portb.f2=1;
      BSF      PORTB+0, 2
;Test.c,36 ::      break;
      GOTO      L_main5
;Test.c,38 ::      default:
L_main9:
;Test.c,39 ::      portb=0x00;
      CLRF     PORTB+0
;Test.c,40 ::      break;
      GOTO      L_main5
;Test.c,42 ::      }
L_main4:
      MOVF     _dataRead+0, 0
      XORLW   49
      BTFSC   STATUS+0, 2
      GOTO    L_main6
      MOVF     _dataRead+0, 0
      XORLW   50
      BTFSC   STATUS+0, 2
      GOTO    L_main7
      MOVF     _dataRead+0, 0
      XORLW   51
      BTFSC   STATUS+0, 2
      GOTO    L_main8
      GOTO    L_main9
L_main5:
;Test.c,43 ::      }
L_main3:

```

```
;Test.c,44 ::      }
      GOTO      L_main1
;Test.c,46 ::      }
L_end_main:
      GOTO      $+0
; end of _main
```

DOCUMENT _ 03

```
char dataRead;
void main() {

TRISB = 0X00;
PORTB = 0X00;

UART1_Init(9600);
Delay_ms(100);
UART1_Write_Text("Press 1 for Red, 2 for Yellow and 3 for Green");
UART1_Write(10);
UART1_Write(13);

while(1)
{
if(UART1_Data_Ready()==1)
{
dataRead = UART1_Read();
switch(dataRead)
{
case '1':
portb.f0=1;
portb.f1=0;
portb.f2=0;
break;

case '2':
portb.f0=0;
portb.f1=1;
portb.f2=0;
break;

case '3':
portb.f0=0;
portb.f1=0;
portb.f2=1;
break;

default:
portb=0x00;
break;

}
}
}
}
```

DOCUMENT _ 04

```
<?xml version="1.0"?>
<MCU_DEVICE_FLAGS>
  <DEVICE>
    <DEVICE_NAME>P16F877A</DEVICE_NAME>
    <VALUE>
      <COUNT>1</COUNT>
      <VALUE0>
        <VAL>$002007:$2F4A</VAL>
      </VALUE0>
    </VALUE>
  </DEVICE>
</MCU_DEVICE_FLAGS>
```

DOCUMENT _ 05

```
#line 1 "D:/OneDrive/Online/Educational Engineering Team/PIC Microcontroller Courses/PIC  
Microcontroller Meets Python/Circuit and Code/Final Code/Test.c"
```

```
char dataRead;
```

```
void main() {
```

```
    TRISB = 0X00;
```

```
    PORTB = 0X00;
```

```
    UART1_Init(9600);
```

```
    Delay_ms(100);
```

```
    UART1_Write_Text("Press 1 for Red, 2 for Yellow and 3 for Green");
```

```
    UART1_Write(10);
```

```
    UART1_Write(13);
```

```
    while(1)
```

```
    {
```

```
        if(UART1_Data_Ready()==1)
```

```
        {
```

```
            dataRead = UART1_Read();
```

```
            switch(dataRead)
```

```
            {
```

```
                case '1':
```

```
                    portb.f0=1;
```

```
                    portb.f1=0;
```

```
                    portb.f2=0;
```

```
                    break;
```

```
                case '2':
```

```
                    portb.f0=0;
```



```
portb.f1=1;  
portb.f2=0;  
break;
```

```
case '3':  
portb.f0=0;  
portb.f1=0;  
portb.f2=1;  
break;
```

```
default:  
portb=0x00;  
break;
```

```
}
```

```
}
```

```
}
```

```
}
```

FINAL CODE

DOCUMENT _ 06

___doicp
___doifc
__ca2aw
__cc2dw
__cca2aw
__ccs2s
__cs2s
__doicpaddr
__flash_size
__fzins
__math_tempb
__math_tempbd
__saveplath
__savestatus
ackdt
ackdt_bit
acken
acken_bit
ackstat
ackstat_bit
adcon0
adcon1
adcs0
adcs0_bit
adcs1
adcs1_bit
adcs2
adcs2_bit

add_32x32_fp
adden
adden_bit
adfm
adfm_bit
adie
adie_bit
adif
adif_bit
adon
adon_bit
adresh
adresl
b0
b1
b2
b3
b4
b5
b6
b7
bclie
bclie_bit
bclif
bclif_bit
bf
bf_bit
boot16_delay_5ms
brgh
brgh_bit
byte2double

c
c_bit
c1inv
c1inv_bit
c1out
c1out_bit
c2inv
c2inv_bit
c2out
c2out_bit
ccp1con
ccp1ie
ccp1ie_bit
ccp1if
ccp1if_bit
ccp1m0
ccp1m0_bit
ccp1m1
ccp1m1_bit
ccp1m2
ccp1m2_bit
ccp1m3
ccp1m3_bit
ccp1x
ccp1x_bit
ccp1y
ccp1y_bit
ccp2con
ccp2ie
ccp2ie_bit
ccp2if

ccp2if_bit
ccp2m0
ccp2m0_bit
ccp2m1
ccp2m1_bit
ccp2m2
ccp2m2_bit
ccp2m3
ccp2m3_bit
ccp2x
ccp2x_bit
ccp2y
ccp2y_bit
ccpr1
ccpr1h
ccpr1l
ccpr2
ccpr2h
ccpr2l
chs0
chs0_bit
chs1
chs1_bit
chs2
chs2_bit
cis
cis_bit
cke
cke_bit
ckp
ckp_bit

cm0
cm0_bit
cm1
cm1_bit
cm2
cm2_bit
cmcon
cmie
cmie_bit
cmif
cmif_bit
comfrem
compare_double
cren
cren_bit
csrc
csrc_bit
cvr0
cvr0_bit
cvr1
cvr1_bit
cvr2
cvr2_bit
cvr3
cvr3_bit
cvrcon
cvren
cvren_bit
cvroe
cvroe_bit
cvrr

cvrr_bit
d
d_a
d_a_bit
d_bit
d_not_a
d_not_a_bit
data_address
data_address_bit
dataread
dc
dc_bit
delay_100ms
delay_10ms
delay_10us
delay_1ms
delay_1sec
delay_1us
delay_22us
delay_500us
delay_50us
delay_550us
delay_5ms
delay_80us
delay_8ms
delay_cyc
div_16x16_s
div_16x16_s_l
div_16x16_u
div_32x32_fp
div_32x32_s

div_32x32_u
div_8x8_s
div_8x8_u
double2byte
double2int
double2longint
double2longword
double2word
eadr
eadrh
eecon1
eecon2
eedata
eedath
eeie
eeie_bit
eeif
eeif_bit
eepgd
eepgd_bit
equals_double
f
ferr
ferr_bit
fixsign32
fsr
fsrptr
gcn
gcn_bit
get_fosc_khz
gie

gie_bit
go
go_bit
go_done
go_done_bit
go_not_done
go_not_done_bit
ibf
ibf_bit
ibf_trise_bit
ibov
ibov_bit
ibov_trise_bit
ics_auto
ics_off
indf
int2double
intcon
inte
inte_bit
intedg
intedg_bit
intf
intf_bit
irp
irp_bit
longint2double
longword2double
main
mul_16x16_s
mul_16x16_u

mul_32x32_fp
mul_32x32_s
mul_32x32_u
mul_8x8_s
mul_8x8_u
not_a
not_a_bit
not_address
not_address_bit
not_bo
not_bo_bit
not_bor
not_bor_bit
not_done
not_done_bit
not_pd
not_pd_bit
not_por
not_por_bit
not_rbp
not_rbp_bit
not_rc8
not_rc8_bit
not_t1sync
not_t1sync_bit
not_to
not_to_bit
not_tx8
not_tx8_bit
not_w
not_w_bit

not_write
not_write_bit
nrm3232
nrm4032
obf
obf_bit
obf_trise_bit
oerr
oerr_bit
option_reg
p
p_bit
pcfg0
pcfg0_bit
pcfg1
pcfg1_bit
pcfg2
pcfg2_bit
pcfg3
pcfg3_bit
pcl
pclath
pcon
peie
peie_bit
pen
pen_bit
pie1
pie2
pir1
pir2

port_to_tris_offset

porta

portb

portc

portd

porte

pr2

ps0

ps0_bit

ps1

ps1_bit

ps2

ps2_bit

psa

psa_bit

pspie

pspie_bit

pspif

pspif_bit

pspmode

pspmode_bit

pspmode_trise_bit

r

r_bit

r_not_w

r_not_w_bit

r_w

r_w_bit

r0

r1

r10

r11

r12

r13

r14

r15

r2

r3

r4

r5

r6

r7

r8

r9

ra0

ra0_bit

ra1

ra1_bit

ra2

ra2_bit

ra3

ra3_bit

ra4

ra4_bit

ra5

ra5_bit

rb0

rb0_bit

rb1

rb1_bit

rb2

rb2_bit

rb3
rb3_bit
rb4
rb4_bit
rb5
rb5_bit
rb6
rb6_bit
rb7
rb7_bit
rbie
rbie_bit
rbif
rbif_bit
rc0
rc0_bit
rc1
rc1_bit
rc2
rc2_bit
rc3
rc3_bit
rc4
rc4_bit
rc5
rc5_bit
rc6
rc6_bit
rc7
rc7_bit
rc8_9

rc8_9_bit

rc9

rc9_bit

rcd8

rcd8_bit

rcen

rcen_bit

rcie

rcie_bit

rcif

rcif_bit

rcreg

rcsta

rd

rd_bit

rd0

rd0_bit

rd1

rd1_bit

rd2

rd2_bit

rd3

rd3_bit

rd4

rd4_bit

rd5

rd5_bit

rd6

rd6_bit

rd7

rd7_bit

re0
re0_bit
re1
re1_bit
re2
re2_bit
read_write
read_write_bit
res032
rp0
rp0_bit
rp1
rp1_bit
rsen
rsen_bit
rx9
rx9_bit
rx9d
rx9d_bit
s
s_bit
sdiv3232l
sen
sen_bit
setfdz32
setfov32
setfun32
setiov3224
smp
smp_bit
spbrg

spen
spen_bit
sren
sren_bit
sspadd
sspbuif
sspcon
sspcon2
sspen
sspen_bit
sspie
sspie_bit
sspif
sspif_bit
sspm0
sspm0_bit
sspm1
sspm1_bit
sspm2
sspm2_bit
sspm3
sspm3_bit
sspov
sspov_bit
sspstat
status
sub_32x32_fp
swap
sync
sync_bit
t0cs

t0cs_bit
t0ie
t0ie_bit
t0if
t0if_bit
t0se
t0se_bit
t1ckps0
t1ckps0_bit
t1ckps1
t1ckps1_bit
t1con
t1insync
t1insync_bit
t1oscen
t1oscen_bit
t1sync
t1sync_bit
t2ckps0
t2ckps0_bit
t2ckps1
t2ckps1_bit
t2con
tmr0
tmr0ie
tmr0ie_bit
tmr0if
tmr0if_bit
tmr1cs
tmr1cs_bit
tmr1h

tmr1ie
tmr1ie_bit
tmr1if
tmr1if_bit
tmr1l
tmr1on
tmr1on_bit
tmr2
tmr2ie
tmr2ie_bit
tmr2if
tmr2if_bit
tmr2on
tmr2on_bit
toutps0
toutps0_bit
toutps1
toutps1_bit
toutps2
toutps2_bit
toutps3
toutps3_bit
trisa
trisa0
trisa0_bit
trisa1
trisa1_bit
trisa2
trisa2_bit
trisa3
trisa3_bit

trisa4
trisa4_bit
trisa5
trisa5_bit
trisb
trisb0
trisb0_bit
trisb1
trisb1_bit
trisb2
trisb2_bit
trisb3
trisb3_bit
trisb4
trisb4_bit
trisb5
trisb5_bit
trisb6
trisb6_bit
trisb7
trisb7_bit
trisc
trisc0
trisc0_bit
trisc1
trisc1_bit
trisc2
trisc2_bit
trisc3
trisc3_bit
trisc4

trisc4_bit

trisc5

trisc5_bit

trisc6

trisc6_bit

trisc7

trisc7_bit

trisd

trisd0

trisd0_bit

trisd1

trisd1_bit

trisd2

trisd2_bit

trisd3

trisd3_bit

trisd4

trisd4_bit

trisd5

trisd5_bit

trisd6

trisd6_bit

trisd7

trisd7_bit

trise

trise0

trise0_bit

trise1

trise1_bit

trise2

trise2_bit

trmt
trmt_bit
tx8_9
tx8_9_bit
tx9
tx9_bit
tx9d
tx9d_bit
txd8
txd8_bit
txen
txen_bit
txie
txie_bit
txif
txif_bit
txreg
txsta
ua
ua_bit
uart1_data_ready
uart1_init
uart1_read
uart1_read_text
uart1_tx_idle
uart1_write
uart1_write_text
vdelay_advanced_ms
vdelay_ms
w
wcol

wcol_bit
word2double
wr
wr_bit
wren
wren_bit
wrerr
wrerr_bit
z
z_bit

DOCUMENT _ 07

```
0      1      mikroCPIC1618.exe -MSF -DBG -pP16F877A -DL -O111111114 -fo20 -
N"D:\OneDrive\Online\Educational Engineering Team\PIC Microcontroller Courses\PIC
Microcontroller Meets Python\Circuit and Code\Final Code\Test.mcppi" -
SP"C:\Users\Public\Documents\Mikroelektronika\mikroC PRO for PIC\Defs\" -
SP"C:\Users\Public\Documents\Mikroelektronika\mikroC PRO for PIC\Uses\P16\" -
SP"D:\OneDrive\Online\Educational Engineering Team\PIC Microcontroller Courses\PIC
Microcontroller Meets Python\Circuit and Code\Final Code\" -
IP"C:\Users\Public\Documents\Mikroelektronika\mikroC PRO for PIC\Uses\P16\" "Test.c"
"__Lib_Math.mcl" "__Lib_MathDouble.mcl" "__Lib_System.mcl" "__Lib_Delays.mcl"
"__Lib_UART_c67.mcl"
```

```
hint: 0 1139 Available RAM: 352 [bytes], Available ROM: 8192 [bytes]
```

```
diagnostics: 0 122 Compilation Started Test.c
```

```
warning: 7 1509 Generated baud rate is 9615 bps (error = 0.16 percent) Test.c
```

```
diagnostics: 46 123 Compiled Successfully Test.c
```

```
diagnostics: 0 127 All files Compiled in 62 ms
```

```
hint: 0 1144 Used RAM (bytes): 53 (15%) Free RAM (bytes): 299 (85%) Used RAM (bytes): 53
(15%) Free RAM (bytes): 299 (85%)
```

```
hint: 0 1144 Used ROM (program words): 216 (3%) Free ROM (program words): 7976 (97%) Used
ROM (program words): 216 (3%) Free ROM (program words): 7976 (97%)
```

```
diagnostics: 0 125 Project Linked Successfully Test.mcppi
```

```
diagnostics: 0 128 Linked in 31 ms
```

```
diagnostics: 0 129 Project 'Test.mcppi' completed: 109 ms
```

```
diagnostics: 0 103 Finished successfully: 15 Aug 2020, 10:13:04 Test.mcppi
```


DOCUMENT _ 08

; LST file generated by mikroListExporter - v.2.0

; Date/Time: 8/15/2020 10:13:04 AM

;-----

```
;Address Opcode      ASM
0x0000 0x2858      GOTO    88
____DoICP:
;__Lib_System.c,6 ::
;__Lib_System.c,9 ::
0x0003 0x1283      BCF     STATUS, 5
0x0004 0x1303      BCF     STATUS, 6
0x0005 0x084F      MOVF    ____DoICPAddr+1, 0
0x0006 0x008A      MOVWF   PCLATH
;__Lib_System.c,10 ::
0x0007 0x084E      MOVF    ____DoICPAddr, 0
0x0008 0x0082      MOVWF   PCL
;__Lib_System.c,12 ::
L_end____DoICP:
0x0009 0x0008      RETURN
; end of ____DoICP
_UART1_Write:
;__Lib_UART_c67.c,39 ::
;__Lib_UART_c67.c,40 ::
L_UART1_Write3:
0x000A 0x1683      BSF     STATUS, 5
0x000B 0x1303      BCF     STATUS, 6
0x000C 0x1898      BTFSC  TXSTA, 1
0x000D 0x2810      GOTO    L_UART1_Write4
;__Lib_UART_c67.c,41 ::
```

```

0x000E 0x0000      NOP
0x000F 0x280A      GOTO   L_UART1_Write3
L_UART1_Write4:
;__Lib_UART_c67.c,42 ::
0x0010 0x1283      BCF    STATUS, 5
0x0011 0x0854      MOVF   FARG_UART1_Write_data_, 0
0x0012 0x0099      MOVWF  TXREG
;__Lib_UART_c67.c,43 ::
L_end_UART1_Write:
0x0013 0x0008      RETURN
; end of _UART1_Write
_UART1_Init:
;__Lib_UART_c67.c,2 ::
;__Lib_UART_c67.c,5 ::
0x0014 0x1683      BSF    STATUS, 5
0x0015 0x1303      BCF    STATUS, 6
0x0016 0x1698      BSF    TXSTA, 5
;__Lib_UART_c67.c,6 ::
0x0017 0x3090      MOVLW  144
0x0018 0x1283      BCF    STATUS, 5
0x0019 0x0098      MOVWF  RCSTA
;__Lib_UART_c67.c,7 ::
0x001A 0x1683      BSF    STATUS, 5
0x001B 0x1787      BSF    TRISC7_bit, BitPos(TRISC7_bit+0)
;__Lib_UART_c67.c,8 ::
0x001C 0x1307      BCF    TRISC6_bit, BitPos(TRISC6_bit+0)
;__Lib_UART_c67.c,10 ::
L_UART1_Init0:
0x001D 0x1283      BCF    STATUS, 5
0x001E 0x1E8C      BTFSS  PIR1, 5
0x001F 0x2823      GOTO   L_UART1_Init1

```

```

;__Lib_UART_c67.c,11 ::
0x0020 0x081A      MOVF   RCREG, 0
0x0021 0x00F0      MOVWF  R0
0x0022 0x281D      GOTO   L_UART1_Init0
L_UART1_Init1:
;__Lib_UART_c67.c,12 ::
L_end_UART1_Init:
0x0023 0x0008      RETURN
; end of _UART1_Init
_UART1_Read:
;__Lib_UART_c67.c,21 ::
;__Lib_UART_c67.c,24 ::
0x0024 0x1283      BCF    STATUS, 5
0x0025 0x1303      BCF    STATUS, 6
0x0026 0x081A      MOVF   RCREG, 0
0x0027 0x00F1      MOVWF  R1
;__Lib_UART_c67.c,25 ::
0x0028 0x1C98      BTFSS  RCSTA, 1
0x0029 0x282C      GOTO   L_UART1_Read2
;__Lib_UART_c67.c,26 ::
0x002A 0x1218      BCF    RCSTA, 4
;__Lib_UART_c67.c,27 ::
0x002B 0x1618      BSF    RCSTA, 4
;__Lib_UART_c67.c,28 ::
L_UART1_Read2:
;__Lib_UART_c67.c,29 ::
0x002C 0x0871      MOVF   R1, 0
0x002D 0x00F0      MOVWF  R0
;__Lib_UART_c67.c,30 ::
L_end_UART1_Read:
0x002E 0x0008      RETURN

```

```

; end of _UART1_Read
__CC2DW:
;__Lib_System.c,134 ::
;__Lib_System.c,137 ::
_CC2D_Loop1:
;__Lib_System.c,139 ::
0x002F 0x2003      CALL   ____DoICP
0x0030 0x118A      BCF   PCLATH, 3
0x0031 0x120A      BCF   PCLATH, 4
;__Lib_System.c,141 ::
0x0032 0x0080      MOVWF  INDF
;__Lib_System.c,142 ::
0x0033 0x0A84      INCF  FSR, 1
;__Lib_System.c,143 ::
0x0034 0x0ACE      INCF  ____DoICPAddr, 1
;__Lib_System.c,145 ::
0x0035 0x1903      BTFSC STATUS, 2
;__Lib_System.c,146 ::
0x0036 0x0ACF      INCF  ____DoICPAddr+1, 1
;__Lib_System.c,147 ::
0x0037 0x03F0      DECF  R0, 1
;__Lib_System.c,149 ::
0x0038 0x1D03      BTFSS STATUS, 2
;__Lib_System.c,150 ::
0x0039 0x282F      GOTO  _CC2D_Loop1
;__Lib_System.c,152 ::
L_end__CC2DW:
0x003A 0x0008      RETURN
; end of __CC2DW
_UART1_Write_Text:
;__Lib_UART_c67.c,47 ::

```

```

;__Lib_UART_c67.c,48 ::
0x003B 0x1283      BCF    STATUS, 5
0x003C 0x1303      BCF    STATUS, 6
0x003D 0x01D3      CLRF   UART1_Write_Text_counter_L0
;__Lib_UART_c67.c,50 ::
0x003E 0x0851      MOVF   FARG_UART1_Write_Text_uart_text, 0
0x003F 0x0084      MOVWF  FSR
0x0040 0x0800      MOVF   INDF, 0
0x0041 0x00D2      MOVWF  UART1_Write_Text_data__L0
;__Lib_UART_c67.c,51 ::
L_UART1_Write_Text5:
0x0042 0x0852      MOVF   UART1_Write_Text_data__L0, 0
0x0043 0x3A00      XORLW  0
0x0044 0x1903      BTFSC  STATUS, 2
0x0045 0x2850      GOTO   L_UART1_Write_Text6
;__Lib_UART_c67.c,52 ::
0x0046 0x0852      MOVF   UART1_Write_Text_data__L0, 0
0x0047 0x00D4      MOVWF  FARG_UART1_Write_data_
0x0048 0x200A      CALL   _UART1_Write
;__Lib_UART_c67.c,53 ::
0x0049 0x0AD3      INCF   UART1_Write_Text_counter_L0, 1
;__Lib_UART_c67.c,54 ::
0x004A 0x0853      MOVF   UART1_Write_Text_counter_L0, 0
0x004B 0x0751      ADDWF  FARG_UART1_Write_Text_uart_text, 0
0x004C 0x0084      MOVWF  FSR
0x004D 0x0800      MOVF   INDF, 0
0x004E 0x00D2      MOVWF  UART1_Write_Text_data__L0
;__Lib_UART_c67.c,55 ::
0x004F 0x2842      GOTO   L_UART1_Write_Text5
L_UART1_Write_Text6:
;__Lib_UART_c67.c,56 ::

```

```

L_end_UART1_Write_Text:
0x0050 0x0008      RETURN
; end of _UART1_Write_Text
_UART1_Data_Ready:
;__Lib_UART_c67.c,16 ::
;__Lib_UART_c67.c,17 ::
0x0051 0x3000      MOVLW   0
0x0052 0x1283      BCF     STATUS, 5
0x0053 0x1303      BCF     STATUS, 6
0x0054 0x1A8C      BTFSC   PIR1, 5
0x0055 0x3001      MOVLW   1
0x0056 0x00F0      MOVWF   R0
;__Lib_UART_c67.c,18 ::
L_end_UART1_Data_Ready:
0x0057 0x0008      RETURN
; end of _UART1_Data_Ready
_main:
0x0058 0x20D4      CALL    212
;Test.c,2 ::      void main() {
;Test.c,4 ::      TRISB = 0X00;
0x0059 0x1683      BSF     STATUS, 5
0x005A 0x1303      BCF     STATUS, 6
0x005B 0x0186      CLRF    TRISB
;Test.c,5 ::      PORTB = 0X00;
0x005C 0x1283      BCF     STATUS, 5
0x005D 0x0186      CLRF    PORTB
;Test.c,7 ::      UART1_Init(9600);
0x005E 0x3081      MOVLW   129
0x005F 0x1683      BSF     STATUS, 5
0x0060 0x0099      MOVWF   SPBRG
0x0061 0x1518      BSF     TXSTA, 2

```

```

0x0062 0x2014      CALL    _UART1_Init
;Test.c,8 ::      Delay_ms(100);
0x0063 0x3003      MOVLW   3
0x0064 0x00FB      MOVWF   R11
0x0065 0x308A      MOVLW  138
0x0066 0x00FC      MOVWF   R12
0x0067 0x3055      MOVLW   85
0x0068 0x00FD      MOVWF   R13

L_main0:
0x0069 0x0BFD      DECFSZ  R13, 1
0x006A 0x2869      GOTO    L_main0
0x006B 0x0BFC      DECFSZ  R12, 1
0x006C 0x2869      GOTO    L_main0
0x006D 0x0BFB      DECFSZ  R11, 1
0x006E 0x2869      GOTO    L_main0
0x006F 0x0000      NOP
0x0070 0x0000      NOP

;Test.c,9 ::      UART1_Write_Text("Press 1 for Red, 2 for Yellow and 3 for Green");
0x0071 0x3020      MOVLW   ?lstr1_Test
0x0072 0x00D1      MOVWF   FARG_UART1_Write_Text_uart_text
0x0073 0x203B      CALL    _UART1_Write_Text
;Test.c,10 ::     UART1_Write(10);
0x0074 0x300A      MOVLW   10
0x0075 0x00D4      MOVWF   FARG_UART1_Write_data_
0x0076 0x200A      CALL    _UART1_Write
;Test.c,11 ::     UART1_Write(13);
0x0077 0x300D      MOVLW   13
0x0078 0x00D4      MOVWF   FARG_UART1_Write_data_
0x0079 0x200A      CALL    _UART1_Write
;Test.c,13 ::     while(1)

L_main1:

```

```

;Test.c,15 ::      if(UART1_Data_Ready()==1)
0x007A 0x2051     CALL   _UART1_Data_Ready
0x007B 0x0870     MOVF   R0, 0
0x007C 0x3A01     XORLW  1
0x007D 0x1D03     BTFSS STATUS, 2
0x007E 0x289E     GOTO  L_main3
;Test.c,17 ::      dataRead = UART1_Read();
0x007F 0x2024     CALL   _UART1_Read
0x0080 0x0870     MOVF   R0, 0
0x0081 0x00D0     MOVWF  _dataRead
;Test.c,18 ::      switch(dataRead)
0x0082 0x2891     GOTO  L_main4
;Test.c,20 ::      case '1':
L_main6:
;Test.c,21 ::      portb.f0=1;
0x0083 0x1406     BSF   PORTB, 0
;Test.c,22 ::      portb.f1=0;
0x0084 0x1086     BCF   PORTB, 1
;Test.c,23 ::      portb.f2=0;
0x0085 0x1106     BCF   PORTB, 2
;Test.c,24 ::      break;
0x0086 0x289E     GOTO  L_main5
;Test.c,26 ::      case '2':
L_main7:
;Test.c,27 ::      portb.f0=0;
0x0087 0x1006     BCF   PORTB, 0
;Test.c,28 ::      portb.f1=1;
0x0088 0x1486     BSF   PORTB, 1
;Test.c,29 ::      portb.f2=0;
0x0089 0x1106     BCF   PORTB, 2
;Test.c,30 ::      break;

```



```

0x008A 0x289E      GOTO   L_main5
;Test.c,32 ::      case '3':
L_main8:
;Test.c,33 ::      portb.f0=0;
0x008B 0x1006      BCF    PORTB, 0
;Test.c,34 ::      portb.f1=0;
0x008C 0x1086      BCF    PORTB, 1
;Test.c,35 ::      portb.f2=1;
0x008D 0x1506      BSF    PORTB, 2
;Test.c,36 ::      break;
0x008E 0x289E      GOTO   L_main5
;Test.c,38 ::      default:
L_main9:
;Test.c,39 ::      portb=0x00;
0x008F 0x0186      CLR    PORTB
;Test.c,40 ::      break;
0x0090 0x289E      GOTO   L_main5
;Test.c,42 ::      }
L_main4:
0x0091 0x0850      MOVF   _dataRead, 0
0x0092 0x3A31      XORLW  49
0x0093 0x1903      BTFSC  STATUS, 2
0x0094 0x2883      GOTO   L_main6
0x0095 0x0850      MOVF   _dataRead, 0
0x0096 0x3A32      XORLW  50
0x0097 0x1903      BTFSC  STATUS, 2
0x0098 0x2887      GOTO   L_main7
0x0099 0x0850      MOVF   _dataRead, 0
0x009A 0x3A33      XORLW  51
0x009B 0x1903      BTFSC  STATUS, 2
0x009C 0x288B      GOTO   L_main8

```

```

0x009D 0x288F      GOTO   L_main9
L_main5:
;Test.c,43 ::      }
L_main3:
;Test.c,44 ::      }
0x009E 0x287A      GOTO   L_main1
;Test.c,46 ::      }
L_end_main:
0x009F 0x289F      GOTO   $+0
; end of _main
0x00D4 0x3020      MOVLW  32
0x00D5 0x0084      MOVWF  FSR
0x00D6 0x302E      MOVLW  46
0x00D7 0x1283      BCF    STATUS, 5
0x00D8 0x1303      BCF    STATUS, 6
0x00D9 0x00F0      MOVWF  R0
0x00DA 0x30A5      MOVLW  165
0x00DB 0x00CE      MOVWF  ___DoICPAddr
0x00DC 0x3000      MOVLW  0
0x00DD 0x00CF      MOVWF  79
0x00DE 0x202F      CALL   47
0x00DF 0x0008      RETURN
;Test.c,0 :: ?ICS?lstr1_Test
0x00A5 0x3450 ;?ICS?lstr1_Test+0
0x00A6 0x3472 ;?ICS?lstr1_Test+1
0x00A7 0x3465 ;?ICS?lstr1_Test+2
0x00A8 0x3473 ;?ICS?lstr1_Test+3
0x00A9 0x3473 ;?ICS?lstr1_Test+4
0x00AA 0x3420 ;?ICS?lstr1_Test+5
0x00AB 0x3431 ;?ICS?lstr1_Test+6
0x00AC 0x3420 ;?ICS?lstr1_Test+7

```

0x00AD 0x3466 ;?ICS?lstr1_Test+8
0x00AE 0x346F ;?ICS?lstr1_Test+9
0x00AF 0x3472 ;?ICS?lstr1_Test+10
0x00B0 0x3420 ;?ICS?lstr1_Test+11
0x00B1 0x3452 ;?ICS?lstr1_Test+12
0x00B2 0x3465 ;?ICS?lstr1_Test+13
0x00B3 0x3464 ;?ICS?lstr1_Test+14
0x00B4 0x342C ;?ICS?lstr1_Test+15
0x00B5 0x3420 ;?ICS?lstr1_Test+16
0x00B6 0x3432 ;?ICS?lstr1_Test+17
0x00B7 0x3420 ;?ICS?lstr1_Test+18
0x00B8 0x3466 ;?ICS?lstr1_Test+19
0x00B9 0x346F ;?ICS?lstr1_Test+20
0x00BA 0x3472 ;?ICS?lstr1_Test+21
0x00BB 0x3420 ;?ICS?lstr1_Test+22
0x00BC 0x3459 ;?ICS?lstr1_Test+23
0x00BD 0x3465 ;?ICS?lstr1_Test+24
0x00BE 0x346C ;?ICS?lstr1_Test+25
0x00BF 0x346C ;?ICS?lstr1_Test+26
0x00C0 0x346F ;?ICS?lstr1_Test+27
0x00C1 0x3477 ;?ICS?lstr1_Test+28
0x00C2 0x3420 ;?ICS?lstr1_Test+29
0x00C3 0x3461 ;?ICS?lstr1_Test+30
0x00C4 0x346E ;?ICS?lstr1_Test+31
0x00C5 0x3464 ;?ICS?lstr1_Test+32
0x00C6 0x3420 ;?ICS?lstr1_Test+33
0x00C7 0x3433 ;?ICS?lstr1_Test+34
0x00C8 0x3420 ;?ICS?lstr1_Test+35
0x00C9 0x3466 ;?ICS?lstr1_Test+36
0x00CA 0x346F ;?ICS?lstr1_Test+37
0x00CB 0x3472 ;?ICS?lstr1_Test+38

```

0x00CC 0x3420 ;?ICS?lstr1_Test+39
0x00CD 0x3447 ;?ICS?lstr1_Test+40
0x00CE 0x3472 ;?ICS?lstr1_Test+41
0x00CF 0x3465 ;?ICS?lstr1_Test+42
0x00D0 0x3465 ;?ICS?lstr1_Test+43
0x00D1 0x346E ;?ICS?lstr1_Test+44
0x00D2 0x3400 ;?ICS?lstr1_Test+45
; end of ?ICS?lstr1_Test

```

Symbol List:

```

/** Routines locations **
//ADDRESS  SIZE  PROCEDURE
//-----
0x0003   [7]  ____DoICP
0x000A   [10] _UART1_Write
0x0014   [16] _UART1_Init
0x0024   [11] _UART1_Read
0x002F   [12] __CC2DW
0x003B   [22] _UART1_Write_Text
0x0051   [7]  _UART1_Data_Ready
0x0058   [71] _main
/** Variables locations **
//ADDRESS  SIZE  VARIABLE
//-----
0x0000   [1]  INDF
0x0002   [1]  PCL
0x0003   [1]  STATUS
0x0004   [1]  FSR
0x0006   [1]  PORTB
0x000A   [1]  PCLATH
0x000C   [1]  PIR1
0x0018   [1]  RCSTA

```

```
0x0019 [1] TXREG
0x001A [1] RCREG
0x0020 [46] ?lstr1_Test
0x004E [2] ___DoICPAddr
0x0050 [1] _dataRead
0x0051 [1] FARG_UART1_Write_Text_uart_text
0x0052 [1] UART1_Write_Text_data__L0
0x0053 [1] UART1_Write_Text_counter_L0
0x0054 [1] FARG_UART1_Write_data_
0x0070 [1] R0
0x0071 [1] R1
0x0072 [1] R2
0x0073 [1] R3
0x0074 [1] R4
0x0075 [1] R5
0x0076 [1] R6
0x0077 [1] R7
0x0078 [1] R8
0x0079 [1] R9
0x007A [1] R10
0x007B [1] R11
0x007C [1] R12
0x007D [1] R13
0x007E [1] R14
0x007F [1] R15
0x0086 [1] TRISB
0x0087 [0] TRISC6_bit
0x0087 [0] TRISC7_bit
0x0098 [1] TXSTA
0x0099 [1] SPBRG
```

```
/** Constants locations **
```

```
//ADDRESS SIZE CONSTANT
```

```
//-----
```

```
0x00A5 [46] ?ICS?lstr1_Test
```

DOCUMENT _ 09

[DEVICE]

Name=P16F877A

Clock=20000000

[FILES]

File0=Test.c

Count=1

[BINARIES]

Count=0

[IMAGES]

Count=0

ActiveImageIndex=-1

[OPENED_FILES]

File0=Test.c

Count=1

[EEPROM]

Count=0

[ACTIVE_COMMENTS_FILES]

Count=0

[OTHER_FILES]

Count=0

[SEARCH_PATH]

Count=0

[HEADER_PATH]

Count=0

[HEADERS]

Count=0

[PLDS]

Count=0

[Useses]

File0=UART
Count=1
[MEMORY_MODEL]
Value=0
[BUILD_TYPE]
Value=0
[ACTIVE_TAB]
Value=Test.c
[USE_EEPROM]
Value=0
[USE_HEAP]
Value=0
[HEAP_SIZE]
Value=0
[EEPROM_DEFINITION]
Value=
[EXPANDED_NODES]
Count=0
[LIB_EXPANDED_NODES]
0=mikroE Libraries
1=System Libraries
Count=2
[PROGRAMMER_TYPE]
Value=mikroE mikroProg
[CODEGRIP_OPTIONS]
CODEGRIP_SPEED=4
CODEGRIP_VERIFY_AFTER=1
CODEGRIP_SWBP_ENABLED=0
CODEGRIP_PROGRAMMING_TYPE=

DOCUMENT _ 10

main

__CC2DW

____DoICP

UART1_Init

UART1_Write_Text

UART1_Write

UART1_Write

UART1_Data_Ready

UART1_Read