

```

#include "int16CXX.H"
#pragma chip PIC16F873
#include "math16.H"

// shortcut definition

#define Temp_ok          PORTB.7 //temperature is good
#define Temp_nok        PORTB.6 //temperature is not good
#define Pressure_ok      PORTB.5 //pressure is good
#define Pressure_nok     PORTB.4 //pressure is not good
#define short_circuit_GND PORTB.3 //short circuit GND
#define short_circuit_supply PORTB.2 //short circuit supply
#define ill              PORTB.1 //sensor is ill
#define open_circuit     PORTB.0 //open circuit
#define battery_output   PORTC.7 //battery
#define sensor_pressure_input PORTA.0 //pressure signal sensor
#define sensor_temp_input PORTA.1 //temperature signal sensor
#define battery_input    PORTA.5 //signal battery

//constant and variable declaration

#define s1 20
#define s2 180
#define s3 220
#define s4 315
#define s5 355
#define s6 480

void main (void)
{
    uns16 Result;
    int SensorValue;
    enum etattype {
        SCG,
        ILL,
        GOOD,
        OC,
        SCP
    } etat;

    // initialization (we initialize all variable)
    PORTB.1 = 0;
    PORTB.2 = 0;
    PORTB.3 = 0;
    PORTB.4 = 0;
    PORTB.5 = 0;
    PORTB.6 = 0;
    PORTB.7 = 0;

```

```

PORTB.0 = 0;
PORTC.7 = 0;

while (1) {
// Analog Digital Conversion configuration
//
    ADCON0.0 = 1; //switch on ADC

    ADCON0.1 = 0;
    ADCON0.3 = 0; //CHS0=0 //bit which enable to select
analog input to convert
    ADCON0.4 = 0; //CHS1=0 //bit which enable to select
analog input to convert
    ADCON0.5 = 0; //CHS2=0 //bit which enable to select
analog input to convert
    ADCON0.6 = 1; //ADCS1=0 // bit which enable to choose
frenquency for ADC
    ADCON0.7 = 0; //ADCSO=1 // bit which enable to
choose frequency for ADC

    ADCON1.0 = 0; //PDCFG0=0; // enable to config pin like analog or
digital input with this config AN0=AN1=AN4=analog input
    ADCON1.1 = 0; //PDCF1=0; // enable to config pin like analog
or digital input
    ADCON1.2 = 0; //PDCF2=0; // enable to config pin like analog
or digital input
    ADCON1.3 = 1; //PDCF3=1; // enable to config pin like analog
or digital input
    ADCON1.4 = 0;
    ADCON1.5 = 0;
    ADCON1.6 = 0;
    ADCON1.7 = 1; //ADFM=1 //left justified

    while (ADCON0.2 == 1); //(ADCON0.2) bit which launch conversion
    {
        nop();
    }
    Result.low8 = ADRESH; // to retrieve only 16-Bits word instead 2 8-bits word
    Result.high8 = ADRESL; // to retrieve only 16-Bits word instead 2 8-bits word

    SensorValue = Result>>6; //right justified
    SensorValue *= 500;
    SensorValue /= 1024;

// on sort avec une valeur V

// level definition
//
//
//          0<value<s1 // short circuit GND
//          s1<value<s2 // 1

```

```
//          s2<value<s3      // s3= good sensor
//          s3<value<s4      // s4=ill
//          s4<value<s5      // s5=open circuit
//          s5<value6<s6     // s6=ill
//          s6<value         // s7=short circuit supply
//
//
//
//
//level pressure comparaison
    if (SensorValue < s1)
    {
        etat = SCG;
    }

    else if (SensorValue < s2)
    {
        etat = ILL;
    }

    else if (SensorValue < s3)
    {
        etat = GOOD;
    }

    else if (SensorValue < s4)
    {
        etat = ILL;
    }

    else if (SensorValue < s5)
    {
        etat = OC;
    }

    else if (SensorValue < s6)
    {
        etat = ILL;
    }
    else if (SensorValue > s6)
    {
        etat = SCP;
    }

//      Applique la valeur sur les sortie.

switch (etat) {
case SCG:
    PORTB.7=0;           //temperature is good
    PORTB.6=1;           //temperature is not good
    PORTB.3=1;           //short circuit GND
    PORTB.2=0;           //short circuit supply
```

```
PORTB.1=0;    //sensor is ill
PORTB.0=0;    //open circuit
```

```
break;
```

```
case ILL:
```

```
PORTB.7=0;    //temperature is good
PORTB.6=1;    //temperature is not good
PORTB.3=0;    //short circuit GND
PORTB.2=0;    //short circuit supply
PORTB.1=1;    //sensor is ill
PORTB.0=0;    //open circuit
break;
```

```
case GOOD:
```

```
PORTB.6=0;    //temperature is not good
PORTB.3=0;    //short circuit GND
PORTB.2=0;    //short circuit supply
PORTB.1=0;    //sensor is ill
PORTB.0=0;    //open circuit
break;
```

```
case OC:
```

```
PORTB.7=0;    //temperature is good
PORTB.6=1;    //temperature is not good
PORTB.3=0;    //short circuit GND
PORTB.2=0;    //short circuit supply
PORTB.1=0;    //sensor is ill
PORTB.0=1;    //open circuit
break;
```

```
case SCP:
```

```
PORTB.7=0;    //temperature is good
PORTB.6=1;    //temperature is not good
PORTB.3=0;    //short circuit GND
PORTB.2=0;    //short circuit supply
PORTB.1=1;    //sensor is ill
PORTB.0=0;    //open circuit
break;
```

```
default:
```

```
break;
```

```
}
```

```
}
```

```
}
```