

```

/*
Cockpit Sensor evaluation code
David Wallis Edinburgh 28/12/16
This version has reorganised variable names
V16 makes Data channel selectable
*****

This code uses include files and parts of the example
provided by Andrew Davies of Hobby Components LTD.
I acknowledge the assistance of HCMP180_Example from
their website.
*****/
char VerSion[14] = "CkptAltV1.17";
/*****
This is clean up version dated 20/10/2017
*****

MPX Data Types
0 Special Data
1 Voltage
2 Current
3 Variometer
4 Speed
5 Rev Counter
6 Temperature
7 Direction
8 Height / Short Distance
*/

//Analog Address
#include <SoftwareSerial.h>
#include <HCBMP180.h> // This is Hobby Components BMP180.h file

HCBMP180 HCBMP180(I2CBMP180ADD); //Address of BMP180 Sensor (x77)
#define PRESSURE_AT_SEA_LEVEL 102100// Pascals, arbitrary value

const int Hgt1 = A0; //Hgt input used to control version output
int HgtInput = 0; //Initial BMP180 Altitude
int i = 0;
int Hgt = 0; // Interim altitude variable
int Dum = 0; // Dum variable #1
int Datach0 = 2; //Data channel select bit0
int Datach1 = 3; //Data channel select bit1
//char VerSion[14] = "CkptAltV1.16";
char Address = 2; // Default MPX data channel address. Can be changed
char DataIn = 0;
char Descript = 8; // MPX data code for Altitude
char FirstByte = 0;
char SecondByte = 0;
char thirdByte = 0;
double Reference_Altitude; // BMP180 Reference Altitude

SoftwareSerial mySerial(10, 11); // RX, TX set up for Arduino

void setup()
{
// The next 5 lines setup the data channel:2, 3, 4 or 5
pinMode(Datach0, INPUT_PULLUP); //Sets I/O pin #2 to Input-Pullup
pinMode(Datach1, INPUT_PULLUP); //Sets I/O pin #3 to Input-Pullup

```

```

    if((digitalRead(Datach0) == 1) && (digitalRead(Datach1) == 0)) Address
= 3;
    if((digitalRead(Datach0) == 0) && (digitalRead(Datach1) == 1)) Address
= 4;
    if((digitalRead(Datach0) == 0) && (digitalRead(Datach1) == 0)) Address
= 5;
// Default data channel is 2
FirstByte = Address * 16;          // Part 1 of FirstByte
FirstByte = FirstByte + Descript; // Part 2 of FirstByte
Serial.begin(38400); // Setup serial comms at 38400Baud

    Serial.println(VerSion); //Output Version in ASCII at 38400

// Initialise the HCBMP180 library and the serial interface
HCBMP180.Init();
Reference_Altitude = HCBMP180.Altitude(HCBMP180.GetPressure(),
PRESSURE_AT_SEA_LEVEL); //Sets zero altitude at at launch site
}

void loop() // run over and over
{
    if (Serial.available())
    {
        DataIn = Serial.read(); // Input data from MPX Receiver
        if(DataIn == Address) // Check for MPX data address
        {
            delay(1); // Wait till MPX Rx ready for sensor data
// Now write previous altitude input - delay of ~ 1/10 Seconds
Serial.write(FirstByte); // This has data address and type
Serial.write(SecondByte); //First 7 bits of Altitude + Alarm bit
Serial.write(thirdByte); //Second 8 bits of Altitude
HgtInput = GetData(); //Get Altitude for next transmission
Hgt = HgtInput; // not sure on this. see note 3
if (Hgt >= 0) // Look for +Ve altitude. (Slope use only)
        {
            Dum = Hgt & 127; // Trim Hgt to 7 bits
            SecondByte = Dum * 2; // and shift Left into SecondByte
            Dum = Hgt / 128; // Trim Altitude for next 8 bits
            thirdByte = Dum; // set thirdByte
        }
        else // Model gone below launch point with -Ve altitude
        {
            Hgt = 32768 + Hgt; // 65536 sets Bit16 so no -1m value
            Dum = Hgt + 32768; // Hence adding 32768 twice
            SecondByte = (Dum & 127) * 2; //Trim Hgt to 7 bits & Left
shift
            thirdByte = Hgt / 128; //Trim Altitude for next 8 bits
        }
        }
    }
}

int GetData(void) //Truncated version of Hobby Components example
{
    long Pressure, SeaLevel;
    double Altitude;
// Get the current pressure in Pascals
    Pressure = HCBMP180.GetPressure();

```

```
// Calculate the absolute height above sea level using the current
pressure at sea level
Altitude = HCBMP180.Altitude(Pressure, PRESSURE_AT_SEA_LEVEL);
Dum = (Altitude - Reference_Altitude);
return Dum;
}
```