```
/* Date 16/08/2014
        Ignacio Betancurt,  Alejandro Brusco,  Maite Ibarburu
*****************************************************************************/
/** I N C L U D E S **********************************************************/
#include <p18cxxx.h>
#include <usart.h>
#include "system/typedefs.h"
#include "system/usb/usb.h"
#include "user/usr_temp.h"
#include "io_cfg.h"             // I/O pin mapping
#include "user/handlerManager.h"
#include "user/usb4butia.h"
/** V A R I A B L E S ********************************************************/
#pragma udata
byte* sendBufferUsrTemp; /* buffer to send data*/

/** P R I V A T E  P R O T O T Y P E S ***************************************/
void UserTempInit(byte handler);
void UserTempReceived(byte* recBuffPtr, byte len, byte handler);
void UserTempRelease(byte handler);

// Tabla para mantener en un lugar fijo una referencia a los modulos que el usuario introduce en USB4all
/** USER MODULE REFERENCE****************************************************/
#pragma romdata user
const uTab userTempModuleTable = {&UserTempInit, &UserTempRelease,"temp"};
#pragma code

/** D E C L A R A T I O N S **************************************************/
#pragma code module
void UserTempInit(byte handler){
  setHandlerReceiveFunction(handler, &UserTempReceived);
  sendBufferUsrTemp = getSharedBuffer(handler);
        getPortDescriptor(handler)->change_port_direction(IN);
}

void UserTempRelease(byte handler){
  unsetHandlerReceiveBuffer(handler);
  unsetHandlerReceiveFunction(handler);
}

void UserTempReceived(byte* recBuffPtr, byte len, byte handler){
  WORD data;
  byte userTempCounter = 0;

  switch(((DATA_PACKET*)recBuffPtr)->CMD)
  {
    case READ_VERSION:
      ((DATA_PACKET*)sendBufferUsrTemp)->_byte[0] = ((DATA_PACKET*)recBuffPtr)->_byte[0];
      ((DATA_PACKET*)sendBufferUsrTemp)->_byte[1] = TEMP_MINOR_VERSION;
      ((DATA_PACKET*)sendBufferUsrTemp)->_byte[2] = TEMP_MAJOR_VERSION;
```

```
            userTempCounter=0x03;
            break;

        case GET_VALUE:
            ((DATA_PACKET*)sendBufferUsrTemp)->_byte[0] = ((DATA_PACKET*)recBuffPtr)->_byte[0];
            data = getPortDescriptor(handler)->get_data_analog();
            ((DATA_PACKET*)sendBufferUsrTemp)->_byte[1] = LSB(data);
            ((DATA_PACKET*)sendBufferUsrTemp)->_byte[2] = MSB(data);
            userTempCounter=0x03;
            break;

        case RESET:
            Reset();
            break;

        default:
            break;

    }//end switch()

    USBGenWrite2(handler, userTempCounter);

}//end userTempReceived

/** EOF usr_temp.c ************************************************************/
```