8. ATEC302 Programming Code Example

*Please refer to the protocol section in the manual for more detail information.

If you would like to write your own software, we listed 3 examples here. Basically, when a PC sends a command to the controller, either is Read or Write command. The controller always returns with response char array. These response char array needs to be read and cleared from the buffer.

8.1 Code Example in pseudo C

These three are global define used in the following examples:

```
unsigned char       CRCbuf[2];
unsigned short int  ByRdCnt
unsigned short int  ComTalking;
```

Review of Protocol:

Combuf[8]={IDadr, R/W, Addr0,Addr1,Val1,Val2,CRC1,CRC2}
IDadr: controller ID number
R/W: Perform Read(x03) from the controller or Write(x06) to the controller
Addr0,Addr1: For SV value address: 0x0000; For SV+SVoffset: 0x1001; For PV+PVoffset: 0x1000
Val1,Val2: For Read process, this is the address that value will be read and put into.
For Write process, this is the address that value will be put into controller
CRC1,CRC2: perform CRC check to make ensure the reliability of the communication

(a) Set SV value:

Write(x06) controller ID#1(x01) SV(x0000) with 75.5°C (75.5x10.=755,
Hex number for 755 is x02F3)

```
SetSVval(void)
{
    unsigned char Combuf[8]={0x01,0x06,0x00,0x00,0x02,0xF3};
    unsigned char Inbyte[8];

    while (ComTaking) {Delay(.01);} // check if others routine is talking, if yes, wait 10mSec
    ComTaking=1; //if no one is talking, now, I'm talking to controller
    FlushOutQ (ComPort); // Now clean up ComPort I/O buffer
    FlushInQ (ComPort);
    Delay (0.03); //wait 30mSec

    do{
        ModbusCalcCRC(Combuf,6); //Sub-Routine to calc CRC value
        Combuf[6]=CRCbuf[0]; //put CRC byte0 value to the com buffer
        Combuf[7]=CRCbuf[1]; //put CRC byte1 value to the com buffer
        ComWrt(ComPort,Combuf,8); //Write to the controller
        Delay(.05); //Wait for 50mSec
        ByRdCnt = ComRd (ComPort, Inbyte, 8); //Read from the controller; should be in ComPort Buffer
        ModbusCalcCRC(sAbyte,6); //Sub-Routine to calc CRC value
        //if CRC not match, clean up buffer redo Set SV
        if(Inbyte[6]!=CRCbuf[0] || Inbyte[7]!=CRCbuf[1] || ByRdCnt!=8){
            FlushOutQ (ComPort);
            FlushInQ (ComPort);
            Delay(0.03);
            ByRdCnt = -1;
            }
    }while (ByRdCnt<0);
}
```

(b) Read PV+PV offset value:

*PV: Process Value - the temperature reading from the sensor, Rd:x03,PVPVOF:x1000

```
float ReadPVPVOF (unsigned short int IDadr)
{
    unsigned char Combuf[8]={0x01,0x03,0x10,0x00,0x00,0x01};
    unsigned char Inbyte[8];
    short int RdVal;
    float fPVval;

    Combuf[0] = IDadr; // set the controller ID number
    While (ComTaking) {Delay(.01); }  // check if others routine is talking, if yes, wait 10mSec
    ComTaking=1; //if no one is talking, now, I'm talking to controller

    // Now clean up ComPort I/O buffer
    FlushOutQ (ComPort);
    FlushInQ (ComPort);
    Delay(0.03); //wait 30 mSec

do{
        ModbusCalcCRC(Combuf,6); //Routine to calc CRC value
        Combuf[6]=CRCbuf[0]; //put CRC byte0 value to the com buffer
        Combuf[7]=CRCbuf[1]; //put CRC byte1 value to the com buffer
        ComWrt(ComPort,Combuf,8 ); //Write to the controller
        Delay(0.05); //wait for 50 mSec
        ByRdCnt = ComRd(ComPort, Inbyte, 8); //Read from the controller
        ModbusCalcCRC(Inbyte,6); //calc CRC value

        //if CRC not match, clean up buffer redo reading PV+PVOF
        if(Inbyte[6]!=CRCbuf[0] || Inbyte[7]!=CRCbuf[1] || ByRdCnt!=8)
        {
            FlushOutQ (ComPort);
            FlushInQ (ComPort);
            Delay(0.03);
            ByRdCnt = -1;
        }
    }while (ByRdCnt<0);
    ComTaking=0; //end talking to controller

    RdVal=(short int)Inbyte[4]*256+(short int)Inbyte[5]; //convert to int
    fPVval=(float)RdVal/10; //adjust to one decimal points //convert to one decimal point float
    return fPVval;
}
```

(c) Read SV+SVOF value:

SV: Set Value – the expecting temperature you want to control at; SVOF: SetPoint Offset, Rd:x03, SVSVOF:x1001

```
float ReadSVSVOF (unsigned short int IDadr)
{
    unsigned char Inbyte[8];
    short int RdVal;
    // address for reading SV + SVOF is 0x1001
    unsigned char Combuf[8]={0x01,0x03,0x10,0x01,0x00,0x01};

    Combuf[0]=IDadr; //controller ID number
    while (ComTaking) {Delay(.01);}
    ComTaking=1;
    FlushOutQ (ComPort);
    FlushInQ (ComPort);
    Delay (0.03);

    do{
        ModbusCalcCRC(Combuf,6);
        databuf[6]=CRCbuf[0];
        databuf[7]=CRCbuf[1];
        ComWrt(ComPort,databuf,8);
        Delay(.05);
        ByRdCnt = ComRd (ComPort, Inbyte, 8);
        ModbusCalcCRC(sAbyte,6);
        if(Inbyte[6]!=CRCbuf[0] || Inbyte[7]!=CRCbuf[1] || ByRdCnt!=8){
            FlushOutQ (ComPort);
            FlushInQ (ComPort);
            Delay(0.03);
            ByRdCnt = -1;
            }
        }while (ByRdCnt<0);

        ComTaking=0;
        RdVal=(short int)Inbyte[4]*256+(short int)Inbyte[5];
        fSV=(float)RdVal/10;
    return fSV;

}
```

(d) Modbus CRC calculate sub-routine

*Please google Modbus CRC if you need more detail information
//Example of calling sub-routine: ModbusCalcCRC(Combuf, 6);

```
void ModbusCalcCRC(unsigned char* Frame, unsigned char LenFrame)
{
    unsigned char CntByte;
    unsigned char j;
    unsigned char bitVal;

    CRC=0xFFFF;
    for(CntByte=0; CntByte<LenFrame;CntByte++)
        {
            CRC^=Frame[CntByte];
            for (j=0;j<8;j++)
            {
                bitVal=CRC & 0x0001;
                CRC=CRC>>1;
                if(bitVal==1) CRC^=0xA001; //MODBUS_GENERATOR;
            }
        }
    CRCbuf[1]=(CRC>>8);
    CRCbuf[0]=CRC & 0x00FF;

}
```