DS580 Miniature Directional Sensor

1 INTRODUCTION

The DS580 contains both a 3-axis fluxgate magnetometer and a 3-axis accelerometer，it is designed to enable high accuracy measurement of the roll, inclination and azimuth orientation angles of a system to which it is mounted. The accuracy of roll and inclination measurement is ±0.1°.The accuracy of azimuth measurements is ±0.5°. The DS580 communicates with the outside world over a serial bidirectional TTL or a RS232 interface.

2 SYSTEM SPECIFICATIONS

SN	Items		Value
1	Inclination Range		0°～180°
2	Inclination (Drift)		±0.1°
3	Azimuth Range		0°～360°
4	Azimuth	Inclination≥5°	±2°
		Inclination≥10°	±1°
		Inclination=90°	±0.5°
5	Roll (Toolface) Range		0°～360°
6	Roll (Toolface)		±0.2°
7	Temperature Range(Operating)		0℃～125℃
8	Temperature Range(Storage)		-40℃～130℃
9	Input Voltage Range		+4.9～+12V
10	Input Current		≤50mA（Vin=+5V）
11	Baud rate		9600~115200
12	Communications		TTL/RS232
13	Shock		5000g，0.1ms
14	Vibration Peak Sine		30g
15	Leads		20”×4

Table 1. DS580 System Specifications

3 Electrical Interface

Pin Name	Wire Color	Function
VIN	Red	+5~+12VDC
GND	Black	Ground
RX	Orange	Serial In
TX	Yellow	Serial Out

Table 2. Electrical Interface

4 Computer Interface

The computer interface of the DS580 System is a TTL level serial interface. In the standard configuration, data is transmitted to the DS580 on pin RX and transmitted out of the DS580 on pin TX. The default baud rate is 9600 with one stop bit and no parity.

In order to communicate with an external personal computer (PC), the TTL levels of the DS580 must be shifted to the RS232 levels used by the serial ports on PCs.

Internal Constants

The operating characteristics of the DS580 are controlled by the value of internal binary constants. The most important constants are:

Binary Function	Data	Action
01	5A	enables autosend upon power up
02	02	sensor output (in Gauss and Gees)
	04	angle output (roll, inclination, azimuth)
08	10	enables ASCII autosend upon power up
09	Baud rate lock (must be =5A if any baud rate other than 9600 is used)	
10	Sets baud rate	

Table 3. Operating characteristics

In order to change the internal system binary constants, a write enable command must first be issued. This is:

0L##

When this command is sent to the DS580, it will respond with the reply:

enabled

To write binary constant 02=04, the command

0WC02B04##

After receiving this and acting upon it, the 750/850 will respond with the reply

done

The reading of internal constants can be accomplished by issuing the command

0SC02B##

When this command is sent, the DS580 will respond by sending the value of constant 02. Wildcards are also recognized.

4.1 Sets baud rate

The communications baud rate can be changed by using the following sequence:

1. Set binary constant 10 according to Table 4.
2. Set binary constant 09 to 0x5a.

3. Cycle power off and on.

The following commands illustrate setting the baud rate to 115200

0L##

0WC10B07##

0L##

0WC09B5A##

When the binary constant“09”is set to any value other than 0x5a,the baud rate is 9600

Baud rate	Binary 10 Value
9600	0x30
19200	0x05
38400	0x06
115200	0x07

Table 4. Baud Rate Settings

4.2 Binary Communication Modes

4.2.1 Component data upload modes

<<Sent First <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Sent Last<<<<

<80> <06> <01> <02> <Mx> <Gx> <My><Gy> <Mz> <Gz> <Temp > <Voltage> <CRC> <06> <04>

<80>return command

<06> <01>is the Token

<02> is the command indicating sensor data is being sent.

<Mx> is a 2 byte signed integer (16 bits) giving the value of the x magnetometer output in Gauss multiplied by 10000. For example a sensor output of 0.3424 Gauss would first be converted to 3424 decimal and then sent as the hex equivalent 0x0D60

<Gx> is a 2 byte signed integer (16 bits) giving the value of the x accelerometer output in Gees multiplied by 10000. For example a sensor output of 0.1345 Gees would first be converted to 1345 decimal and then sent as the hex equivalent 0x23A3 (=0010001110100011 Binary).

<My><Gy> <Mz> <Gz> are similarly defined as discussed above.

<Temp> is the system temperature encoded as a 2 byte signed integer (16 bits) with value equal to the temperature multiplied by 100. For example a temperature of 98 deg. C would be sent as 0x2648.

<Voltage> are 2 byte signed integers representing the system voltage times 100. A system voltage of 5V would be sent as 0x1F4.

<CRC> is the 16-bit cyclic redundancy check. This a two byte number that is sent MSB first followed by LSB.

<06> is the token

CRC:

80 06 01 02 dataByte [0] ~ dataByte [15] CrcByte[0] CrcByte [1] 06 04

Example:

```
unsigned short crc=0; ( unsigned short is 16bits data format )
unsigned short crc16tab[256]={
    0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0, 0x0280, 0xc241,
    0xc601, 0x06c0, 0x0780, 0xc741, 0x0500, 0xc5c1, 0xc481, 0x0440,
    0xcc01, 0x0cc0, 0x0d80, 0xcd41, 0x0f00, 0xcfc1, 0xce81, 0x0e40,
    0x0a00, 0xcac1, 0xcb81, 0x0b40, 0xc901, 0x09c0, 0x0880, 0xc841,
    0xd801, 0x18c0, 0x1980, 0xd941, 0x1b00, 0xdbc1, 0xda81, 0x1a40,
    0x1e00, 0xdec1, 0xdf81, 0x1f40, 0xdd01, 0x1dc0, 0x1c80, 0xdc41,
    0x1400, 0xd4c1, 0xd581, 0x1540, 0xd701, 0x17c0, 0x1680, 0xd641,
    0xd201, 0x12c0, 0x1380, 0xd341, 0x1100, 0xd1c1, 0xd081, 0x1040,
    0xf001, 0x30c0, 0x3180, 0xf141, 0x3300, 0xf3c1, 0xf281, 0x3240,
    0x3600, 0xf6c1, 0xf781, 0x3740, 0xf501, 0x35c0, 0x3480, 0xf441,
    0x3c00, 0xfcc1, 0xfd81, 0x3d40, 0xff01, 0x3fc0, 0x3e80, 0xfe41,
    0xfa01, 0x3ac0, 0x3b80, 0xfb41, 0x3900, 0xf9c1, 0xf881, 0x3840,
    0x2800, 0xe8c1, 0xe981, 0x2940, 0xeb01, 0x2bc0, 0x2a80, 0xea41,
    0xee01, 0x2ec0, 0x2f80, 0xef41, 0x2d00, 0xedc1, 0xec81, 0x2c40,
    0xe401, 0x24c0, 0x2580, 0xe541, 0x2700, 0xe7c1, 0xe681, 0x2640,
    0x2200, 0xe2c1, 0xe381, 0x2340, 0xe101, 0x21c0, 0x2080, 0xe041,
    0xa001, 0x60c0, 0x6180, 0xa141, 0x6300, 0xa3c1, 0xa281, 0x6240,
    0x6600, 0xa6c1, 0xa781, 0x6740, 0xa501, 0x65c0, 0x6480, 0xa441,
    0x6c00, 0xacc1, 0xad81, 0x6d40, 0xaf01, 0x6fc0, 0x6e80, 0xae41,
    0xaa01, 0x6ac0, 0x6b80, 0xab41, 0x6900, 0xa9c1, 0xa881, 0x6840,
    0x7800, 0xb8c1, 0xb981, 0x7940, 0xbb01, 0x7bc0, 0x7a80, 0xba41,
    0xbe01, 0x7ec0, 0x7f80, 0xbf41, 0x7d00, 0xbdc1, 0xbc81, 0x7c40,
    0xb401, 0x74c0, 0x7580, 0xb541, 0x7700, 0xb7c1, 0xb681, 0x7640,
    0x7200, 0xb2c1, 0xb381, 0x7340, 0xb101, 0x71c0, 0x7080, 0xb041,
    0x5000, 0x90c1, 0x9181, 0x5140, 0x9301, 0x53c0, 0x5280, 0x9241,
    0x9601, 0x56c0, 0x5780, 0x9741, 0x5500, 0x95c1, 0x9481, 0x5440,
    0x9c01, 0x5cc0, 0x5d80, 0x9d41, 0x5f00, 0x9fc1, 0x9e81, 0x5e40,
```

```
        0x5a00, 0x9ac1, 0x9b81, 0x5b40, 0x9901, 0x59c0, 0x5880, 0x9841,
        0x8801, 0x48c0, 0x4980, 0x8941, 0x4b00, 0x8bc1, 0x8a81, 0x4a40,
        0x4e00, 0x8ec1, 0x8f81, 0x4f40, 0x8d01, 0x4dc0, 0x4c80, 0x8c41,
        0x4400, 0x84c1, 0x8581, 0x4540, 0x8701, 0x47c0, 0x4680, 0x8641,
        0x8201, 0x42c0, 0x4380, 0x8341, 0x4100, 0x81c1, 0x8081, 0x4040

    };
    //CRC generate
    void CalcCRC16(unsigned char    c)
    {
        crc = ((unsigned short)((crc >> 8) ^ crc16tab[ (crc ^ c) & 0xFF ]));
     }

   void    CheckCrc( void )
   {
    crc = 0;
        for (int i=0;i<16;i++)// all data number
        {
                CalcCRC16(dataByte [i]);
        }
        if (CrcByte [0] ==    (unsigned char)(crc>>8) && CrcByte [1] == (unsigned char)(crc))
        {
          CrcCheckPass;
        }
        else
        {
          CrcCheckFail;
        }
   }
```

4.4.2 Angle data upload modes

<<Sent First <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Sent Last<<<<

<83><06> <02> <03> <Roll> <MRoll> <Inc><TMag> <Azimuth> <TGrav> <Temp > <Voltage> <CRC> <06> <04>

<83>return command

<06> <02> is the Token

<03> is the command indicating sensor angle data is being sent.

<Roll> is a 2 byte sequence giving the value of Roll (i.e.Toolface) output in Degrees multiplied by 10. For example a sensor roll angle of 30 deg would be encoded as 0x12C since 12C hex is equal to 300 decimal. All other angles in the packet are encoded in the same manner. eg, MRoll: Magnetic roll, Inc: inclination, Azimuth: Azimuth. All total field values e.g.Tgrav=Total Gravity are encoded by multiplying by 10000. TMag: total magnetic field, TGrav: total gravity field.

<Voltage> and **<Temp>** are 2 byte signed integers representing the system voltage and temperature times 100. A system voltage of 5V would be sent as 0x1F4.

<CRC> is the 16-bit cyclic redundancy check. This a two byte number that is sent MSB first followed by LSB.

<06> is the token

<04> is EOT

Note : All 2 byte data is sent as MSB first and LSB second;

CRC:

80 06 01 02 dataByte [0] ~ dataByte [15] CrcByte[0] CrcByte [1] 06 04

Example:

```
unsigned short crc=0; ( unsigned short is 16bits data format )
unsigned short crc16tab[256]={
    0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0, 0x0280, 0xc241,
    0xc601, 0x06c0, 0x0780, 0xc741, 0x0500, 0xc5c1, 0xc481, 0x0440,
    0xcc01, 0x0cc0, 0x0d80, 0xcd41, 0x0f00, 0xcfc1, 0xce81, 0x0e40,
    0x0a00, 0xcac1, 0xcb81, 0x0b40, 0xc901, 0x09c0, 0x0880, 0xc841,
    0xd801, 0x18c0, 0x1980, 0xd941, 0x1b00, 0xdbc1, 0xda81, 0x1a40,
    0x1e00, 0xdec1, 0xdf81, 0x1f40, 0xdd01, 0x1dc0, 0x1c80, 0xdc41,
    0x1400, 0xd4c1, 0xd581, 0x1540, 0xd701, 0x17c0, 0x1680, 0xd641,
    0xd201, 0x12c0, 0x1380, 0xd341, 0x1100, 0xd1c1, 0xd081, 0x1040,
    0xf001, 0x30c0, 0x3180, 0xf141, 0x3300, 0xf3c1, 0xf281, 0x3240,
    0x3600, 0xf6c1, 0xf781, 0x3740, 0xf501, 0x35c0, 0x3480, 0xf441,
    0x3c00, 0xfcc1, 0xfd81, 0x3d40, 0xff01, 0x3fc0, 0x3e80, 0xfe41,
    0xfa01, 0x3ac0, 0x3b80, 0xfb41, 0x3900, 0xf9c1, 0xf881, 0x3840,
    0x2800, 0xe8c1, 0xe981, 0x2940, 0xeb01, 0x2bc0, 0x2a80, 0xea41,
    0xee01, 0x2ec0, 0x2f80, 0xef41, 0x2d00, 0xedc1, 0xec81, 0x2c40,
    0xe401, 0x24c0, 0x2580, 0xe541, 0x2700, 0xe7c1, 0xe681, 0x2640,
    0x2200, 0xe2c1, 0xe381, 0x2340, 0xe101, 0x21c0, 0x2080, 0xe041,
    0xa001, 0x60c0, 0x6180, 0xa141, 0x6300, 0xa3c1, 0xa281, 0x6240,
```

```
            0x6600, 0xa6c1, 0xa781, 0x6740, 0xa501, 0x65c0, 0x6480, 0xa441,
            0x6c00, 0xacc1, 0xad81, 0x6d40, 0xaf01, 0x6fc0, 0x6e80, 0xae41,
            0xaa01, 0x6ac0, 0x6b80, 0xab41, 0x6900, 0xa9c1, 0xa881, 0x6840,
            0x7800, 0xb8c1, 0xb981, 0x7940, 0xbb01, 0x7bc0, 0x7a80, 0xba41,
            0xbe01, 0x7ec0, 0x7f80, 0xbf41, 0x7d00, 0xbdc1, 0xbc81, 0x7c40,
            0xb401, 0x74c0, 0x7580, 0xb541, 0x7700, 0xb7c1, 0xb681, 0x7640,
            0x7200, 0xb2c1, 0xb381, 0x7340, 0xb101, 0x71c0, 0x7080, 0xb041,

0x5000, 0x90c1, 0x9181, 0x5140, 0x9301, 0x53c0, 0x5280, 0x9241,
            0x9601, 0x56c0, 0x5780, 0x9741, 0x5500, 0x95c1, 0x9481, 0x5440,
            0x9c01, 0x5cc0, 0x5d80, 0x9d41, 0x5f00, 0x9fc1, 0x9e81, 0x5e40,
            0x5a00, 0x9ac1, 0x9b81, 0x5b40, 0x9901, 0x59c0, 0x5880, 0x9841,
            0x8801, 0x48c0, 0x4980, 0x8941, 0x4b00, 0x8bc1, 0x8a81, 0x4a40,
            0x4e00, 0x8ec1, 0x8f81, 0x4f40, 0x8d01, 0x4dc0, 0x4c80, 0x8c41,
            0x4400, 0x84c1, 0x8581, 0x4540, 0x8701, 0x47c0, 0x4680, 0x8641,
            0x8201, 0x42c0, 0x4380, 0x8341, 0x4100, 0x81c1, 0x8081, 0x4040
      };
      //CRC generate
      void CalcCRC16(unsigned char    c)
      {
            crc = ((unsigned short)((crc >> 8) ^ crc16tab[ (crc ^ c) & 0xFF ]));
       }

       void    CheckCrc( void )
       {
      crc = 0;
          for (int i=0;i<16;i++)// all data number
          {
                    CalcCRC16(dataByte [i]);
          }
          if (CrcByte [0] ==    (unsigned char)(crc>>8) && CrcByte [1] == (unsigned char)(crc))
          {
            CrcCheckPass;
          }
          else
          {
```

```
            CrcCheckFail;
        }
    }
```

5 Definitions

The following sections describe equations for determining the DS580 orientation angles. These equations make use of the following definitions:

Gx accelerometer x axis output

Gy accelerometer y axis output

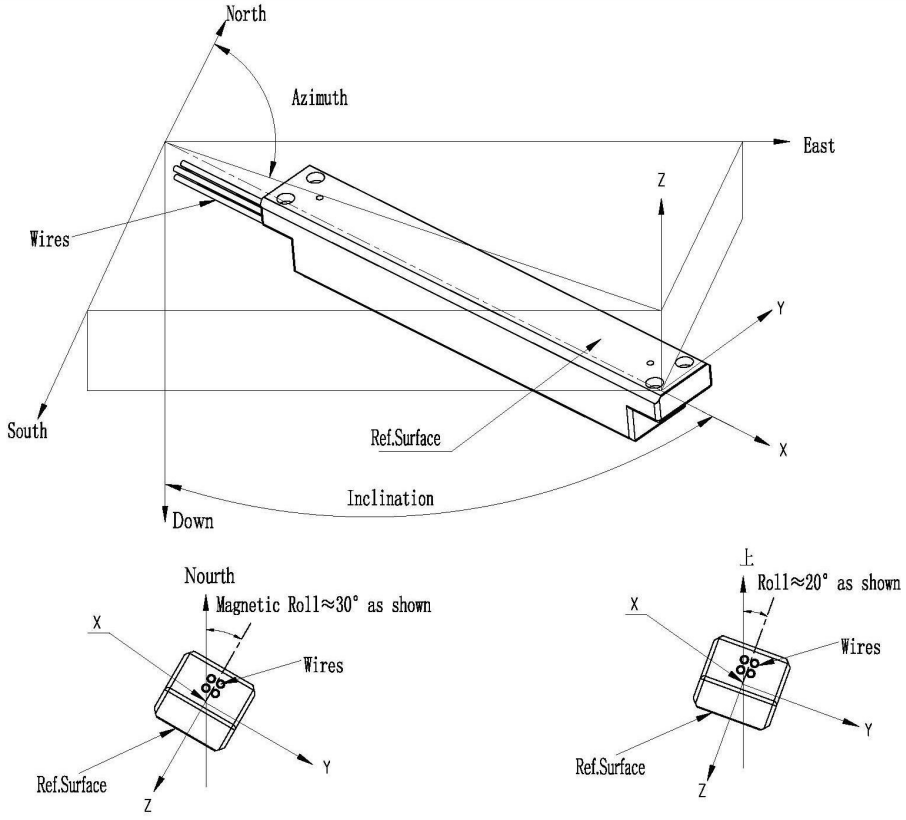
Gz accelerometer y axis output

Mx magnetometer x axis output

My magnetometer y axis output

Mz magnetometer z axis output

Fig. 1. DS580 Coordinate System and Orientation Angles.



5.1 Calculation of Gravity Toolface and Magnetic Toolface

The roll angle, θ, is determined by using the following equations ($0 < \theta < 2\pi$)

$$\sin\theta = \frac{Gy}{\sqrt{Gy^2 + Gz^2}}$$

$$\cos\theta = \frac{Gz}{\sqrt{Gy^2 + Gz^2}}$$

$$\tan\theta = \frac{Gy}{Gz}$$

Roll is 0° when Gy = 0 and Gz > 0. Roll is 2π radians when Gy = 0 and Gz < 0.

When the x axis is near vertical (inclination < 5°), the quantities Gy and Gz become very small and the above expressions yield a less accurate value of θ.

In this situation, magnetic roll is often used to determine the angular orientation of the DS580 about the longitudinal (X) axis. Magnetic roll, θ_m, is given by the following ($0 < \theta < 2\pi$)

$$\sin\theta_m = \frac{My}{\sqrt{My^2 + Mz^2}}$$

$$\cos\theta_m = \frac{Mz}{\sqrt{My^2 + Mz^2}}$$

$$\tan\theta_m = \frac{My}{Mz}$$

5.2 CALCULATION OF INCLINATION

Inclination, ε, is determined from the following equations $0 < \varepsilon < 2\pi$)

$$\sin\varepsilon = \frac{\sqrt{Gy^2 + Gz^2}}{g}$$

$$\cos\varepsilon = \frac{Gx}{g}$$

$$\tan\varepsilon = \frac{\sqrt{Gy^2 + Gz^2}}{Gx}$$

Where g=**Error! Reference source not found.**

5.3 Magnetic Heading (Azimuth)

Magnetic heading, φ, is then given by (0 <ε<2π)

$$\tan\varphi = \frac{(MzGy - MyGz)g}{Mx(Gy^2 + Gz^2) - MyGxGy - MzGxGz}$$

Where g=**Error! Reference source not found.**

Fig. 2. DS580 Mechanical Diagram