DS360A Rotary Measure Directional Sensor

1.INTRODUCTION

The DS360A is fabricated by 3 axis fluxgate magnetometer and 3 axis accelerometer，the output signal of the DS360A is digitized by a 24-bit high resolution ADC,the temperature drift and orthogonal compensation were carried out.It is designed to enable high accuracy measurement of the roll, inclination and azimuth orientation angles of a system to which it is mounted while operating between 0~150 ℃.
The built-in digital filter and compensation algorithm can remove the vibration and rotation interference signals,can accurately measure the different angles under vibration and ratation.
The DS360A can be customized the probe size,installation and digital output interface (UART,RS232).

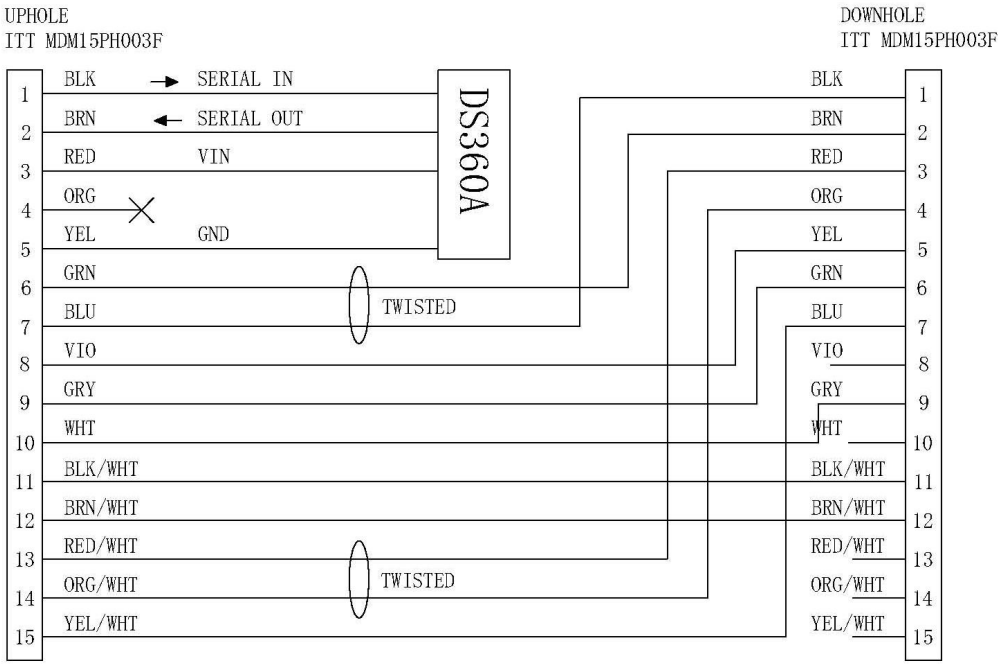
2.FEATURES

- High temperature resistance
- Vibration and shock resistance
- Rotary measure
- Low power consumption

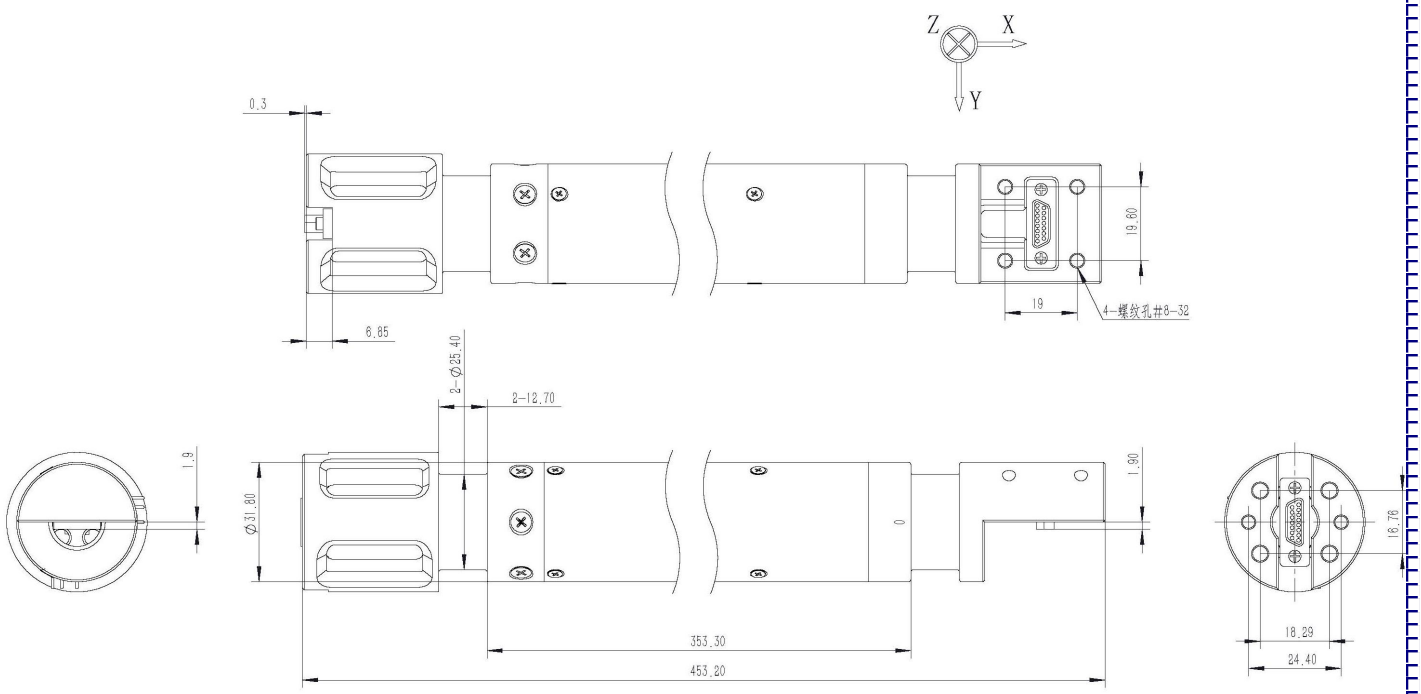
3.PERFORMANCE

Parameter	Test condition	Standard	Unit
Azimuth Accuracy(static)	Inclination≥10°	±1	°
Azimuth Accuracy(rotary)	Inclination≥45°，300RPM	±1.5	
Deviation Angles（static）	0~180°	±0.1	
Inclination（rotary）	300RPM	±0.2	
	Sine vibration：10~300Hz ， 6G	±0.2	
Tool surface（static）	Inclination≥10°	±1	
Operating tempeature	-	0~150	℃
storage	-	-40~160	
Input voltage range	-	10~36	V
Input power current	TA=25℃,Vs=15V	≤80	mA
Communication interface	-	TTL/RS-232	-
Baud rate	-	9600~115200	Hz
Outlook size	-	384×31	mm

4. ELECTRICAL INTERFACES AND DEFINITION



5.OUTLOOK



6.COMMUNICATION PROTOCOL

DS360A adopts TTL communication interface, baud rate 9600, data transmission adopts binary communication mode, protocol content (dynamic measurement function only supports 6.2 Angle data upload mode) :

6.1 Component data upload mode

After receiving 0x80, DS360A responds by sending binary sensor component packets (double-byte data, high-byte before) :

<80> <10> <MX> <AX> <MY> <AY> <MZ> <AZ> <T><V><80>< Checksum> <7FFF>

<80> instruction return (that is, the return value of host sending instruction 0x80);

<10> component data upload synchronization header;

<MX> is a double byte (x axis magnetic sensor data times 10000);

<AX> is a double byte (x axis acceleration sensor data times 10000);

<MY><AY><MZ><AZ>;

<T> is a double byte (temperature sensor data multiplied by 100);

<V> is a double byte (input voltage data multiplied by 100);

<80> is a status byte (fixed 0x80);

< Checksum > data Checksum (low 8 bits);

<7FFF> synchronization tail 0x7FFF.

Among them: < Checksum > = < MX > + < AX > + < MY > + < AY > + < MZ > + < AZ > + < T >, < V >.

6.2 Angle data upload mode

On receipt of 0x83, the DS360A responds by sending a binary sensor Angle packet (double byte data, high byte in front).

<83> <10> <TF> <MS> <INC> <BT> <AZ> <GT> <T> <V> <80> < Checksum > <7FFF>

<83> instruction return (that is, the return value of host sending instruction 0x83);

<10> Angle data upload synchronization header;

<TF> is a double byte (gravity tool surface Angle calculation data multiplied by 10);

<MS> is a double byte (magnetic tool surface Angle calculation data multiplied by 10);

<INC> is a double byte (tilt calculation times 10);

<BT> is double byte (calculated data of total magnetic field strength multiplied by 10000);

<AZ> is a double byte (azimuth computed data times 10);

<GT> is a double byte (the total speed calculation is multiplied by 10000);

<T> is a double byte (temperature sensor data multiplied by 100);

<V> is a double byte (input voltage data multiplied by 100);

<80> is a status byte (fixed 0x80);

< Checksum > data Checksum (low 8 bits);

<7FFF> synchronization tail 0x7FFF.

Among them: < Checksum > = < MX > + < AX > + < MY > + < AY > + < MZ > + < AZ > + < T >, < V >.

6.3 Speed data upload mode

On receipt of 0x88, the DS360A responds by sending the binary sensor speed packet (double byte data, high byte in front).

<06> <51> <RPM> <MS> <BT> < CRC > <0604>

<06> upload sync header;

<51> speed data upload synchronous head;

<RPM> is a double byte (speed data multiplied by 10);

<MS> is a double byte (magnetic tool surface Angle calculation data multiplied by 182);

<BT> is a double byte (calculated data of total magnetic field strength multiplied by 32768);

<CRC> is a double-byte CRC check, with the high byte before;

<0604> is a double-byte synchronization tail;

CRC:

The packet structure is as follows：

06 51 dataByte [0] ~ dataByte [5] CrcByte[0] CrcByte [1] 06 04

Program 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<6;i++)// all data number
        {
                CalcCRC16(dataByte [i]);
        }
        if (CrcByte [0] ==    (unsigned char)(crc>>8) && CrcByte [1] == (unsigned char)(crc))
        {
          CrcCheckPass;
        }
        else
        {
          CrcCheckFail;
        }
   }
```

7.INSTALLATION REQUIRMENTS

- Ensure that the DS360A sensor and instrument are coaxial when installed.

8.Notice

- DS360A is an electrostatic sensitive device, anti-static bracelet should be worn in the process of installation, welding and debugging. In particular, special pay attention to electrostatic protection when the instrument framework is directly connected with the power source.
- DS360A is a wearing part, please handle and put gently, do not drop or touch.Shock and vibration shall not exceed the technical requirements of the product;
- DS360A shall be packaged reliably in transportation, with buffer foam no less than 20mm around, and shall be fixed with foam.Multiple sensors are packed to ensure that each part cannot collide with the other;
- Check the wiring of each pin before energizing, and do not connect wrongly to avoid damaging the sensor;