DFRduino GPS Shield-LEA-6H (SKU:TEL0044)

Introduction

The LEA-6H is a high performance stand-alone GPS and GALILEO receiver module designed to allow easy, straightforward migration from its LEA-4 predecessors. It features u-blox' KickStart weak signal acquisition technology, as well as flexible connectivity options. The LEA-6H comes with built-in Flash memory that enables firmware updates and the storage of specific configuration settings in a non-volatile RAM. The built-in antenna supervisor supports external and active antennas, such as u-blox' ANN high performance GPS antenna.

u-blox KickStart provides accelerated startup at weak signals, and our featured SuperSense® Indoor GPS is capable of tracking and acquiring even extremely weak signals. This makes the LEA-6H suitable for solutions using small or covert antennas.

Specification

- Easy migration from LEA-4H and LEA-4P modules
- Accelerated startup at weak signals thanks to KickStart Technology
- Operating voltage: 2.7 - 3.6 V
- LEA-6H Reference design documentation available with ceramic or GeoHelix antenna, UART and USB
- 2 Hz position update rate
- Built-In Flash memory for fimware upgrades and storage of specific configuration settings

- Antenna supervisor and supply
- Antenna short and open circuit detection and protection for external antennas
- 1 UART, 1 USB and 1 DDC (I2C compliant) interface
- GALILEO-ready
- 50-channel u-blox 5 engine with over 1 million effective correlators
- Under 1 second Time-To-First-Fix for Hot and Aided Starts
- SuperSense® Indoor GPS: -160 dBm tracking sensitivity
- Supports AssistNow Online and AssistNow Offline A-GPS services; OMA SUPL compliant
- Supports SBAS (WAAS, EGNOS, MSAS, GAGAN)
- Operating temperature range: -40 to 85°C
- RoHS compliant (lead-free)

Sample Code

```
// # Editor     : Lauren from DFRobot
// # Date       : 22.02.2012


// # Product name: DFRduino GPS Shield-LEA-6H
// # Product SKU : TEL0044
// # Version     : 1.0


// # Update the library and sketch to compatible with IDE V1.0 and earlier


// # Description:
// # The sketch for using the DFRduino GPS Shield-LEA-6H


// # Connection:
// #        Directly connect the shield to the Arduino controller
// #        If you'd like to drive the shield via UART interface, you may nee
d to connect the jumpers on the board.
// #



#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#define WireSend(args) Wire.write(args)
#define WireRead(args) Wire.read(args)
#define printByte(args) Serial.write(args)
#define printlnByte(args)  Serial.write(args),Serial.println()
```

```
#else
#include "WProgram.h"
#define WireSend(args) Wire.send(args)
#define WireRead(args) Wire.receive(args)
#define printByte(args) Serial.print(args,BYTE)
#define printlnByte(args)  Serial.println(args,BYTE)
#endif


#include <Wire.h>


#define BUFFER_LENGTH 10//Define the buffer length


int GPSAddress = 0x42;//GPS I2C Address


double Datatransfer(char *data_buf,char num)//Data type converter：convert cha
r type to float
{                                           //*data_buf:char data array ;num:
float length
  double temp=0.0;
  unsigned char i,j;


  if(data_buf[0]=='-')//The condition of the negative
  {
    i=1;
    //The date in the array is converted to an integer and accumulative
    while(data_buf[i]!='.')
      temp=temp*10+(data_buf[i++]-0x30);
    for(j=0;j<num;j++)
      temp=temp*10+(data_buf[++i]-0x30);
    //The date will converted integer transform into a floating point number
    for(j=0;j<num;j++)
      temp=temp/10;
    //Converted to a negative number
    temp=0-temp;
```

```
  }
  else//Positive case
  {
    i=0;
    while(data_buf[i]!='.')
      temp=temp*10+(data_buf[i++]-0x30);
    for(j=0;j<num;j++)
      temp=temp*10+(data_buf[++i]-0x30);
    for(j=0;j<num;j++)
      temp=temp/10 ;
  }
  return temp;
}
void rec_init()//initial GPS
{
  Wire.beginTransmission(GPSAddress);
  WireSend(0xff);//To send data address
  Wire.endTransmission();

  Wire.beginTransmission(GPSAddress);
  Wire.requestFrom(GPSAddress,10);//Require 10 bytes read from the GPS device
}
char ID()//Receive the statement ID
{
  char i = 0;
  char value[7]={
    '$','G','P','G','G','A',','      };//To receive the ID content of GPS sta
tements
  char buff[7]={
    '0','0','0','0','0','0','0'      };

  while(1)
  {
    rec_init();//Receive data initialization
```

```
    while(Wire.available())
    {
      buff[i] = WireRead();//Receive serial data
      if(buff[i]==value[i])//Contrast the correct ID
      {
        i++;
        if(i==7)
        {
          Wire.endTransmission();//End of receiving
          return 1;//Receiving returns 1

        }
      }
      else
        i=0;
    }
    Wire.endTransmission();//End receiving
  }
}
void UTC()//Time information
{
  char i = 0,flag=0;
  char value[7]={
    '$','G','P','G','G','A',','   };
  char buff[7]={
    '0','0','0','0','0','0','0'       };
  char time[9]={
    '0','0','0','0','0','0','0','0','0'    };//Storage time data

  double t=0.0;

  while(1)
  {
    rec_init();
```

```
    while(Wire.available())
    {
      if(!flag)
      {
        buff[i] = WireRead();
        if(buff[i]==value[i])
        {
          i++;
          if(i==7)
          {
            i=0;
            flag=1;
          }
        }
        else
          i=0;
      }
      else
      {
        time[i] = WireRead();
        i++;
        if(i==9)
        {
          t=Datatransfer(time,2);//Converts floating-point data
          t=t+80000.00;//To convert time to Beijing time
          Serial.println(t);//The time data output
          Wire.endTransmission();
          return;
        }
      }
    }
    Wire.endTransmission();
  }
}
```

```
void rec_data(char *buff,char num1,char num2)//Receive data function
{                          //*buff：Receive data array；num1：Number of commas ；
num2：The   length of the array
  char i=0,count=0;


  if(ID())
  {
    while(1)
    {
      rec_init();
      while(Wire.available())
      {
        buff[i] = WireRead();
        if(count!=num1)
        {
          if(buff[i]==',')
            count++;
        }
        else
        {
          i++;
          if(i==num2)
          {
            Wire.endTransmission();
            return;
          }
        }
      }
      Wire.endTransmission();
    }
  }
}
void latitude()//Latitude information
{
```

```
  char lat[10]={
    '0','0','0','0','0','0','0','0','0','0' };//Store the latitude data
  rec_data(lat,1 ,10);//Receive the latitude data
  Serial.println(Datatransfer(lat,5),5);//output
}
void lat_dir()//Latitude direction information
{
  char dir[1]={'0'};//Store latitude direction data

  rec_data(dir,2,1);//Receive latitude direction data
  printlnByte(dir[0]);//output latitude direction information
}
void  longitude()//Longitude information
{
  char lon[11]={
    '0','0','0','0','0','0','0','0','0','0','0' };//Store longitude data
  rec_data(lon,3,11);//Receive the longitude data
  Serial.println(Datatransfer(lon,5),5);//out put date
}
void lon_dir()//Longitude direction information

{
  char dir[1]={'0'};
  rec_data(dir,4,1);
  printlnByte(dir[0]);//output latitude direction information
}
void altitude()//Altitude information
{
  char i=0,count=0;
  char alt[8]={
    '0','0','0','0','0','0','0','0' };

  if(ID())
  {
```

```
    while(1)
    {
      rec_init();
      while(Wire.available())
      {
        alt[i] = WireRead();
        if(count!=8)
        {
          if(alt[i]==',')
            count++;
        }
        else
        {
          if(alt[i]==',')
          {
            Serial.println(Datatransfer(alt,1),1);
            Wire.endTransmission();
            return;
          }
          else
            i++;
        }
      }
      Wire.endTransmission();
    }
  }
}
void setup()
{
  Wire.begin();//IIC Initialize
  Serial.begin(9600);//set baud rate
  Serial.println("DFRobot DFRduino GPS Shield v1.0");
  Serial.println("$GPGGA statement information: ");
}
```

```
void loop()
{
  while(1)
  {
    Serial.print("UTC:");
    UTC();
    Serial.print("Lat:");
    latitude();
    Serial.print("Dir:");
    lat_dir();
    Serial.print("Lon:");
    longitude();
    Serial.print("Dir:");
    lon_dir();
    Serial.print("Alt:");
    altitude();
    Serial.println(' ');
    Serial.println(' ');
  }
}
```

Notice:When you use above code.Please unplug the jumper caps before you upload the code to Arduino.And when it has been finished, don't forget to plug it back.