IPS Realtime Plot Instructions

This file introduces how to realtime plot the Arduino output with the python code **'IPS.py'**.

1. Files description

There are 8 files in the e-dimension IPS fold. The files description is tabulated in the following table.

python-2.7.9.zip	Python 2.7.9 for windows
Python_modules.zip	Python modules required in this project
l3g-arduino-master.zip	L3g Arduino library
lps-arduino-master.zip	LPS Arduino library
lsm303-arduino-master.zip	LSM Arduino library
minimu-9-ahrs-arduino-master.zip	Attitude and Heading Reference System
IPS.py	Python code to realtime plot the steps
IPS.pdf	IPS project description file

2. Python and modules installation

To run the plot in real-time for the IPS system, python code is attached. To run the code, firstly install python 2.7.9 on your computer. After that, unzip "Modules.zip" and follow the below

1) Copy "dateutil" folder from "python-dateutil-2.4.0" folder to C:\Python27\Lib\site-packages

2) Copy "drawnow" folder from "drawnow-0.44" folder to C:\Python27\Lib\site-packages

3) Copy "six.py" from "six-1.9.0" folder to C:\Python27\Lib\site-packages

4) Install "matplotlib"

5) Install "numpy"

6) Install "pyparsing"

7) Install "pyserial"

3. Arduino code output format

Make sure the sequence of the Arduino output is the same as the sequence in the python code.

Arduino

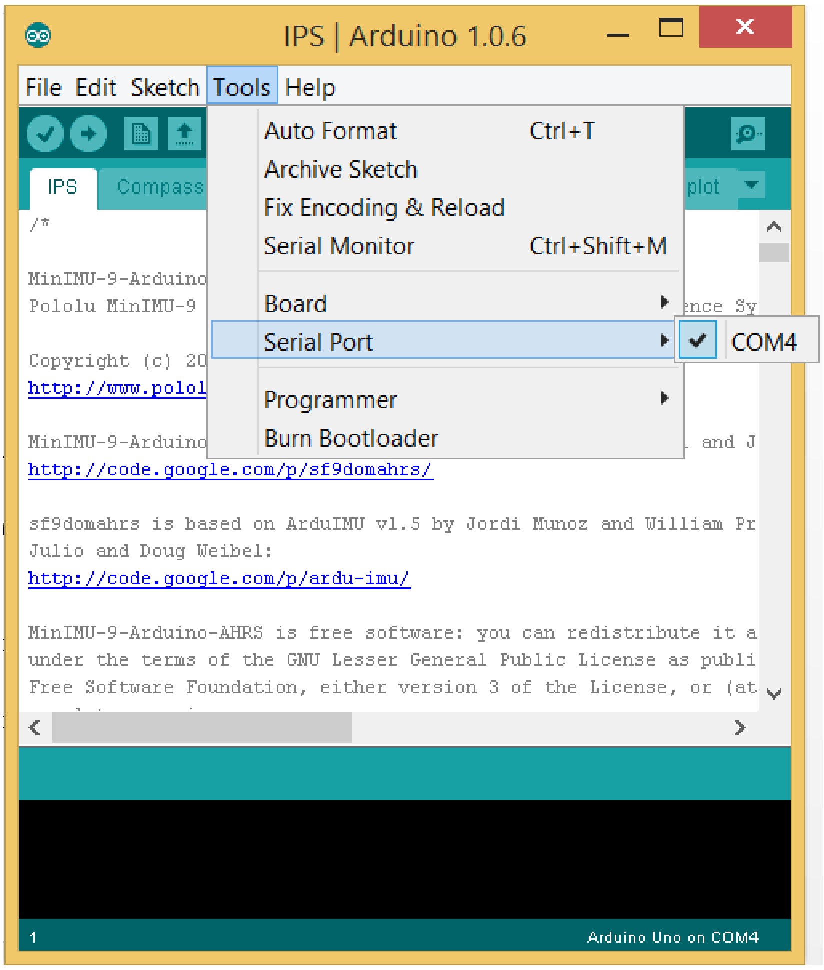
```
Serial.print(STEPs);
Serial.print(",");
Serial.print(STEP_LENGTH);
Serial.print(",");
Serial.print(Position[0]);
Serial.print(",");
Serial.print(Position[1]);
Serial.print(",");


Serial.print(Distance[0]);
Serial.print(",");
Serial.print(Distance[1]);
Serial.print(",");
Serial.print(pressure);
Serial.print(",");
Serial.print(altitude);
Serial.print(",");
Serial.println(level);
```

Python

```
STEP = float(words[0])
STEParray.append(STEP)
STEPLENGTH = float(words[1])
STEPLENGTHarray.append(STEPLENGTH)
POSITIONX = float(words[2])
XARRAY.append(POSITIONX)
POSITIONY = float(words[3])
YARRAY.append(POSITIONY)
X_offset = POSITIONX
Y_offset = POSITIONY
distance0 = float(words[4])

distance1 = float(words[5])

pressure = float(words[6])

altitude = float(words[7])

level = float(words[8])
```

4. Run IPS python code

Before running python code, make sure the COM port and baud rate are set correctly. In the following example, the COM port is COM4 and the baud rate is 115200 bps.



```
/*

MinIMU-9-Arduino
Pololu MinIMU-9                                   nce Sy

Copyright (c) 20
http://www.polol

MinIMU-9-Arduino                                  . and J
http://code.google.com/p/sf9domahrs/

sf9domahrs is based on ArduIMU v1.5 by Jordi Munoz and William Pr
Julio and Doug Weibel:
http://code.google.com/p/ardu-imu/

MinIMU-9-Arduino-AHRS is free software: you can redistribute it a
under the terms of the GNU Lesser General Public License as publi
Free Software Foundation, either version 3 of the License, or (at
```

In the Arduino code

```
Serial.begin(115200);
```

In the Python code

```
# Check your COM port and baud rate
ser = serial.Serial(port='COM4',baudrate=115200, parity=serial.PARITY_NONE,
timeout=0)
```

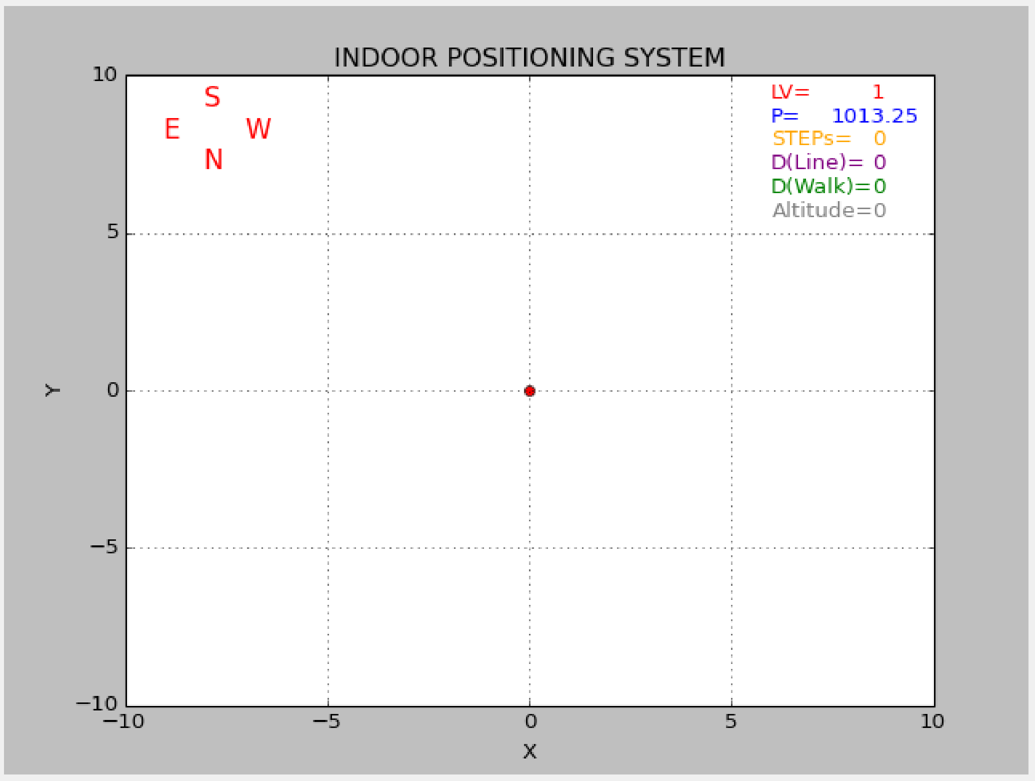
After the COM port and baud rate are set correctly, open the CMD windows and run the following command in the CMD window.

```
python <IPSfold>\IPS.py
```

If ‘python’ command is not found, add the python installation directory before the python command.

```
C:\Python27\python <IPSfold>\IPS.py
```

The following window will pop up. The window size is set to 20 m X 20 m.



Change the cnt threshold to change the number of point to be shown on the screen.

```
# Number of Points shown in the window
if(cnt>200):
        STEParray.pop(0)
        STEPLENGTHarray.pop(0)
        XARRAY.pop(0)
        YARRAY.pop(0)
```