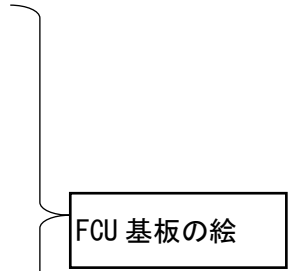


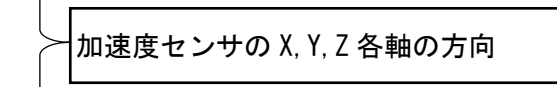
```

1 /*
2 * The purpose of this file is to convert the sensor data to proper coordinate
3 * X-axis and Y-axis orientation is in line with Accelerometer sensor mounted on board
4 *
5 * STEVAL-FCU001V1
6 *
7 *
8 *      +-----+
9 *      | I2C  MOTOR  BAT |
10 *      |          ^     |
11 *      |          |     |
12 *      |          |     |
13 *      | RC      |     |
14 *      |          |     |
15 *      | BLE     |     |
16 *      |          |     |
17 *      +-----+
18 *
19 *          ^ (ACC X)
20 *          |
21 *      (ACC Y) <-- 0 (z)
22 *
23 * (1) Drone FWD direction V (ACC -X)
24 * (2) Drone FWD direction > (ACC -Y)
25 * (3) Drone FWD direction ^ (ACC X)
26 * (4) Drone FWD direction < (ACC Y)
27 *
28 * Translation of coordinates for AHRS
29 * (1) Drone FWD direction > (ACC -X)
30 *
31 *      (x) <-- 0 (z)
32 *      |
33 *      V (y)
34 *
35 * (2) Drone FWD direction > (ACC -Y)
36 *
37 *          ^ (x)
38 *          |
39 *      (y) <-- 0 (z)
40 *
41 * (3) Drone FWD direction ^ (ACC X)
42 *
43 *          ^ (y)
44 *          |
45 *      (z) 0--> (x)
46 *
47 * (4) Drone FWD direction < (ACC Y)
48 *
49 *      (z) 0--> (y)
50 *          |
51 *          V (x)

```



FCU 基板の絵



加速度センサの X, Y, Z 各軸の方向

機体軸（機体固定座標系）の取り方は (1) ~ (4) の 4 つが選べる。Config_drone.h 内の定数 COORDINATE_SYSTEM で設定する。デフォルトは (3)。この場合、FCU 基板中心から見て、モータコネクタ側が Y 軸正方向、USB, JTAG コネクタ側が X 軸正方向、基板の垂直上方（部品実装面側）が Z 軸正方向となる。

```

51 */
52
53
54 #include "sensor_data.h"
55
56
57 /*
58 * This function read sensor data and prepare data for proper coordinate system
59 * according to definition of COORDINATE_SYSTEM
60 * The unit of each data are:
61 *     Acc - mg
62 *     Gyro - mdps
63 *     Mag - mguass
64 */
65 void ReadSensorRawData(void *ACC_handle, void *GYR_handle, void *MAG_handle, void
66 *PRE_handle, AxesRaw_TypeDef *acc, AxesRaw_TypeDef *gyro, AxesRaw_TypeDef *mag, float *pre)
67 {
68     int32_t t1;
69     SensorAxes_t acc_temp_int16, gyro_temp_int16, mag_temp_int16;          /* Data Type
70     int16_t */
71     AxesRaw_TypeDef acc_temp, gyro_temp;                                  /* Data
72     Type int32_t */
73     // Read data is in mg unit
74     BSP_ACCELERO_Get_Axes(ACC_handle, &acc_temp_int16);
75     acc_temp.AXIS_X = (int32_t) acc_temp_int16.AXIS_X;                    /* Casting data to
76     int32_t */
77     acc_temp.AXIS_Y = (int32_t) acc_temp_int16.AXIS_Y;
78     acc_temp.AXIS_Z = (int32_t) acc_temp_int16.AXIS_Z;
79     // Read data is in mdps unit
80     BSP_GYRO_Get_Axes(GYR_handle, &gyro_temp_int16);
81     gyro_temp.AXIS_X = (int32_t) gyro_temp_int16.AXIS_X;                 /* Casting data to
82     int32_t */
83     gyro_temp.AXIS_Y = (int32_t) gyro_temp_int16.AXIS_Y;
84     gyro_temp.AXIS_Z = (int32_t) gyro_temp_int16.AXIS_Z;
85     // Read data is in mg unit
86     if (USE_MAG_SENSOR) {
87         BSP_MAGNETO_Get_Axes(MAG_handle, &mag_temp_int16);
88         mag->AXIS_X = (int32_t) mag_temp_int16.AXIS_X;
89         mag->AXIS_Y = (int32_t) mag_temp_int16.AXIS_Y;
90         mag->AXIS_Z = (int32_t) mag_temp_int16.AXIS_Z;
91     }
92     else
93     {
94         mag->AXIS_X = 0;
95         mag->AXIS_Y = 0;
96         mag->AXIS_Z = 0;
97     }
98
99     if (USE_PRESSURE_SENSOR)
100         BSP_PRESSURE_Get_Press(PRE_handle, &pre);

```

センサデータの読み込み

```

96     else
97         *pre = 0;
98
99     if (COORDINATE_SYSTEM == 1)
100    {
101        // convert acc
102        t1 = acc->AXIS_X;
103        acc->AXIS_X = acc->AXIS_Y;
104        acc->AXIS_Y = -t1;
105        // convert gyro
106        t1 = gyro->AXIS_X;
107        gyro->AXIS_X = gyro->AXIS_Y;
108        gyro->AXIS_Y = -t1;
109        // convert mag
110        t1 = mag->AXIS_X;
111        mag->AXIS_X = mag->AXIS_Y;
112        mag->AXIS_Y = -t1;
113    }
114    else if (COORDINATE_SYSTEM == 2)
115    {
116        // No need to convert in this case
117    }
118    else if (COORDINATE_SYSTEM == 3)
119    {
120
121        acc->AXIS_X = -acc_temp.AXIS_Y;
122        acc->AXIS_Y = acc_temp.AXIS_X;
123        acc->AXIS_Z = acc_temp.AXIS_Z;
124
125        gyro->AXIS_X = -gyro_temp.AXIS_Y;
126        gyro->AXIS_Y = gyro_temp.AXIS_X;
127        gyro->AXIS_Z = gyro_temp.AXIS_Z;
128
129        // convert mag
130        t1 = mag->AXIS_X;
131        mag->AXIS_X = - mag->AXIS_Y;
132        mag->AXIS_Y = t1;
133    }
134    else if (COORDINATE_SYSTEM == 4)
135    {
136        // convert acc
137        acc->AXIS_X = - acc->AXIS_X;
138        acc->AXIS_Y = - acc->AXIS_Y;
139        // convert gyro
140        gyro->AXIS_X = - gyro->AXIS_X;
141        gyro->AXIS_Y = - gyro->AXIS_Y;
142        // convert mag
143        mag->AXIS_X = - mag->AXIS_X;
144        mag->AXIS_Y = - mag->AXIS_Y;
145    }

```

センサ軸を機体軸へ座標変換する。
 機体軸として(3)を選択した場合、
 機体軸+X軸は加速度センサ-Y軸、
 機体軸+Y軸は加速度センサ+X軸、
 機体軸+Z軸は加速度センサ+Z軸、
 に等しくなる。
 (ジャイロセンサも同様)

146}

147

148

149