

Protokol o semestrálním projektu z předmětu Elektronika a komunikace

Zpracování dat z IMU

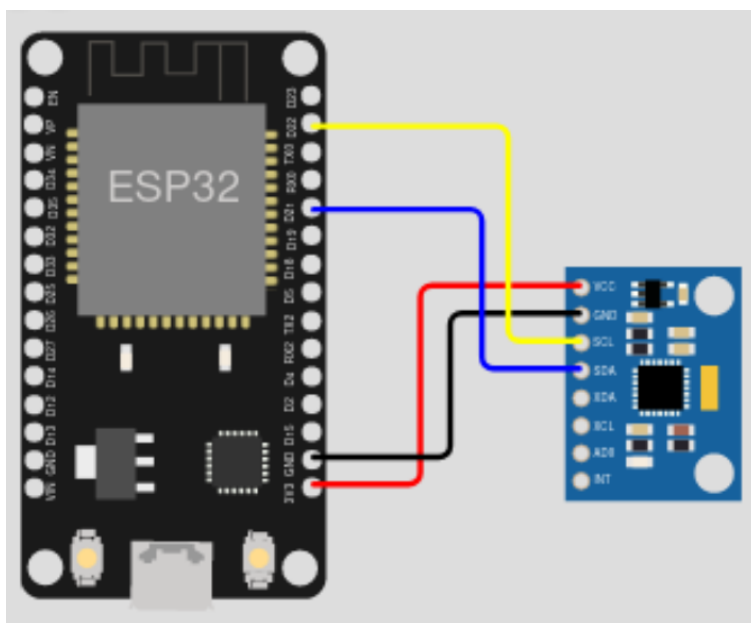
Macháček Tomáš

January 6, 2023

1 Popis zapojení

Senzor inerciálních veličin MPU9250 (dále IMU) je připojen čtyřmi vodiči k vývojovému kitu ESP32. Dva slouží k napájení IMU (poskytováno kitem) a ostatní dva pro přenos dat po sběrnici I2C. Deska je napájena USB kabelem a zároveň je přes něj řešen přenos dat mezi ESP a počítačem.

2 Schéma zapojení



3 Popis kódu - ESP32

```
1 //=====
2 //
3 // File: board.ino
4 // Description: Code for the board.
5 // Author: Tomas Machacek
6 // Date: 2 Jan 2023
7 // Notes: Heavily based on
8 //       https://github.com/wollewald/MPU9250\_WE/
9 //       blob/main/examples/MPU6500_all_data/MPU6500_all_data.ino
10 //
11 //=====
12
13 // system headers
14 #include <Wire.h>
15 #include <MPU6500_WE.h>
16
17 // Adress of the sensor.
18 #define MPU9250_ADDRESS 0x68
19
20 // Terminal symbols.
21 #define CR '\r'
22 #define FF '\f'
23
24 // Creating new instance of MPU6500_WE class.
25 MPU6500_WE imu = MPU6500_WE(MPU9250_ADDRESS);
26
27 // Init message, use with arduino boards.
28 // #define __INIT__ 'i'
29
30 unsigned long prevTime_ms = 0;
31
32 // Logging frequency set to 400Hz.
33 const unsigned int period_us = 1000000 / 400;
34
35 // Variables for storing accelerometer and gyroscope data.
36 xyzFloat acc;
37 xyzFloat gyr;
38
39 void setup() {
40
41     // Initialize Wire.
42     Wire.begin();
43
44     // Start serial communication.
45     Serial.begin(460800, SERIAL_8N1);
46
47     // Use for arduino boards.
48     // if(Serial) {
49     //     Serial.print(__INIT__);
50     //     delay(100);
51     // }
52
53     // Initialize imu.
54     imu.init();
55
56     // Store actual offset of the accelerometer and gyroscope.
57     imu.autoOffsets();
58
59     // Enable low pass filter for gyroscope.
60     imu.enableGyrDLPF();
61
62     // Set the filter to max.
63     imu.setGyrDLPF(MPU6500_DLPF_6);
64
65     // Sample rate divider to 0.
66     imu.setSampleRateDivider(0);
```

```

67
68 // Set the gyroscope range to 250 deg/s.
69 imu.setGyrRange(MPU6500_GYRO_RANGE_250);
70
71 // Enable low pass filter for accelerometer.
72 imu.enableAccDLPF(true);
73
74 // Set the filter to max.
75 imu.setAccDLPF(MPU6500_DLPF_6);
76
77 // Set the accelerometer range to 4g.
78 imu.setAccRange(MPU6500_ACC_RANGE_4G);
79 delay(200);
80 }
81
82 void loop() {
83
84 // Every period_ms send imu data.
85 if (micros() - prevTime_ms >= period_us) {
86     prevTime_ms = micros();
87
88     // Get data from sensor.
89     acc = imu.getGValues();
90     gyr = imu.getGyrValues();
91
92     // Send the data.
93     SendData();
94 }
95 }
96
97 // Function for sending formatted dato to serial.
98 // It sends timestamp along with accelerometer and gyroscope data.
99 void SendData() {
100     Serial.print(FF);
101     Serial.print(micros());
102     Serial.print(CR);
103     Serial.print(acc.x);
104     Serial.print(CR);
105     Serial.print(acc.y);
106     Serial.print(CR);
107     Serial.print(acc.z);
108     Serial.print(CR);
109     Serial.print(gyr.x);
110     Serial.print(CR);
111     Serial.print(gyr.y);
112     Serial.print(CR);
113     Serial.print(gyr.z);
114 }

```

Program slouží pouze k inicializaci senzoru MPU9250, periodickému záznamu dat z IMU a jejich následnému odeslání po sériové lince do počítače (sběrnice UART).

4 Popis kódu na počítači

Na počítači probíhá záznam (uložení) a zpracování dat. V souboru *board.conf* se definují parametry sériové komunikace (musí být shodné s parametry na ESP). Po spuštění programu se inicializuje sériový port podle konfiguračního souboru (pakliže se nastavení v souboru neuvede, nastaví se na základní hodnotu), poté se spustí záchyt dat. Po uplynutí časové konstanty (opět se dá nastavit v konfiguračním souboru), se mohou data buď uložit či zahodit. Data se ukládají do složky *data* do souboru *output.csv*. Dále se mohou zpracovat prostřednictvím skriptu v pythonu, který vygeneruje grafy (nezpracované hodnoty, Eulerovy úhly, zrychlení, rychlost, pozici ve složkách a ve 3D). Výstupem je pdf soubor obsahující všechny grafy.

4.1 Obsluha sériového portu

Prostřednictvím rozhraní knihovny *termios2* jsou nakonfigurovány parametry sériového portu. Jedná se o počet bitů v jednom rámci, paritní bit, počet stop bitů, baud rate, timeout, echo.

4.2 Skript ke zpracování dat

Pomocí knihovny *matplotlib* jsou vytvářeny všechny grafy, tím prvním je reprezentace dat vyčtených z IMU. Prostřednictvím Madgwickova filtru jsou hodnoty zpracovány do Eulerových úhlů (další graf). Dále je vytvořen graf akcelerace. Kvůli odstínění šumu ze senzoru bylo nutné zakomponovat do skriptu detektor pohybu, který rozhoduje zda je senzor v pohybu, či ne. Takto ošetřená data se mohou dále převést na rychlost jednoduchým vzorcem.

$$v = v_0 + a \cdot \Delta t$$

Rychlost je zobrazena složkově v grafu. Dále je převedena na polohu po složkách.

$$s = s_0 + v \cdot \Delta t$$

Poloha je zanesena do grafu a poté je vizualizovaná 3D grafem.

5 Možnosti, návrhy na zlepšení

IMU bylo testováno ve všech třech směrech, přičemž chyba byla v řádech centimetrů. Délka záznamu byla pokaždé pět vteřin, při delším záznamu by se výrazně projevil drift oproti skutečné poloze. Řešením by byl kvalitnější senzor vybavený magnetometrem.

6 Závěr

Cílem projektu bylo zjistit možnosti levného IMU.

7 Zdroj

Všechny zdrojové jsou dostupné zde: https://github.com/piroot01/ekp_2022.