**Dr. Bob Davidov**

# Показания навигационного датчика MPU-6050

***Цель работы:*** изучение свойств цифрового навигационного датчика с I2C интерфейсом.

***Задача работы:*** построение стенда для наглядного отображения характеристик датчика угловых и линейных перемещений в пространстве.

***Приборы и принадлежности:*** Модуль 3-х осевого гироскопа и 3-х осевого акселерометра GY-521 на базе микросхемы MPU-6050, контроллер Arduino UNO, персональный компьютер с MATLAB.

## ВВЕДЕНИЕ

Для управления перемещением объектов в пространстве необходимы соответствующие датчики. Существует множество методов определения текущих координат местоположения объекта с использованием внешних источников сигналов с заранее известным расположением в пространстве и датчиков автономной навигации, закрепляемых непосредственно на движущемся объекте. В этой работе рассматриваются характеристики компактного модуля GY-521 применяемого, в основном, в системах навигации коптеров и авиамоделей. Этот модуль на базе микросхемы MPU-6050 подключается через I2C интерфейс и контроллер Arduino UNO к среде MATLAB для наглядного отображения сигналов датчика и работы алгоритмов коррекции в реальном времени.

## ОБЩИЕ СВЕДЕНИЯ

### Характеристики модуля GY-521

Закрепляемый на объекте модуль GY-521 (см. Рисунок 1) содержит микросхему MPU-6050 с 3-х осевым гироскопом, 3-х осевым акселерометром и датчиком температуры. Гироскоп показывает угловую скорость по трем координатам в градусах в секунду. Чтобы определить относительное перемещение модуля показание гироскопа необходимо проинтегрировать. Акселерометр показывает линейное ускорение объекта, которое суммируется с проекцией вектора ускорения свободного падения. Например, при показаниях акселерометра по оси Z, X и Y равным 9,81; 0,0 и 0,0 м/с2 соответственно, плоскости микросхемы и модуля параллельны горизонтальной плоскости, ускорение модуля равно нулю, модуль находится в состоянии покоя или перемещается с постоянной скоростью.

.

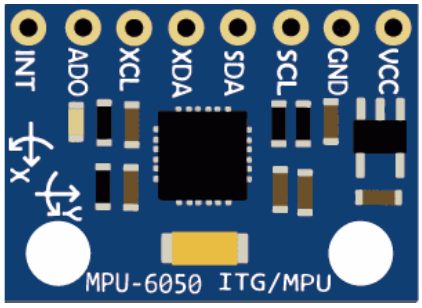



**Рисунок 1**. Внешний вид модуля GY-521 на базе микросхемы MPU-6050 3-х осевого гироскопа и акселерометра. Оси X и Y лежат в плоскости модуля, как показано на рисунке. Ось Z перпендикулярна плоскости рисунка.

Модуль GY-521 имеет следующие характеристики [1].

- Модуль GY-521 включает микросхему датчиков MPU-6050
- Напряжение питания модуля: от 3,5V до 6V (DC)
- Модуль содержит стабилизатор напряжения 3,3В для питания MPU-6050
- Потребляемый ток модуля: до 4 мА
- Гироскоп с 16 битным АЦП (показывает угловую скорость)
    - Диапазон: ± 250, ± 500, ± 1000, ± 2000 °/с
    - Чувствительность в диапазонах ± 250, ± 500, ± 1000, ± 2000 °/с: 131, 65.5, 32.8, 16.4 бит/(°/с)
    - Чувствительность к изменению диапазона: ± 2%
    - Чувствительность к показаниям по другим осям: ± 2%
    - Нелинейность: ± 0.2%
    - Показания при нулевой скорости: ± 20 °/с
    - Чувствительность к линейным ускорениям: ± 0,1 (°/с)/g
    - Частота программируемого сэмплирования АЦП: от 4 до 8000 преобразований в секунду.
    - Диапазон программируемой полосы пропускания встроенного низкочастотного фильтра: 5 .. 260 Гц (устанавливается одновременно для гироскопа и акселерометра)
    - Время установки показаний гироскопа после его включения: 30 мс
- Акселерометр с 16 битным АЦП
    - Диапазон: ± 2 ± 4 ± 8 ± 16g
    - Чувствительность в диапазонах ± 2, ± 4, ± 8, ± 16g: 16384, 8192, 4096, 2048 бит/g
    - Точность начальной калибровки: ± 3%
    - Нелинейность: 0.5%
    - Чувствительность к температуре: ± 0.02%/°C
    - Чувствительность к показаниям по другим осям: ± 2%
    - Частота программируемого сэмплирования АЦП: от 4 до 1000 преобразований в секунду.
    - Диапазон программируемой полосы пропускания встроенного низкочастотного фильтра: 5 .. 260 Гц (устанавливается одновременно для гироскопа и акселерометра)
    - Показания акселерометра микросхемы MPU-6050 установленной в горизонтальной плоскости: 0g по X и Y осям и +1g по оси Z
- Датчик температуры
    - Диапазон измерения: -40°C to +85°C
    - Чувствительность: 340 бит/°C

- Смещение при 35°C: -521 бит
- Нелинейность: ± 1°C
- Формула вычисления температуры, °C: (показание датчика)/340+35-(-521/340) или *(показание датчика)/340+36,53*

- Цифровой процессор микросхемы MPU-6050 выполняет обработку движения с высокой скоростью - около 200 Гц, чтобы обеспечить точность измерения при малой задержке.
- Интерфейс связи: I2C (максимальная скорость I2C - 400 кГц)
- Размер: 15x20 мм.
- Вес: 5 г

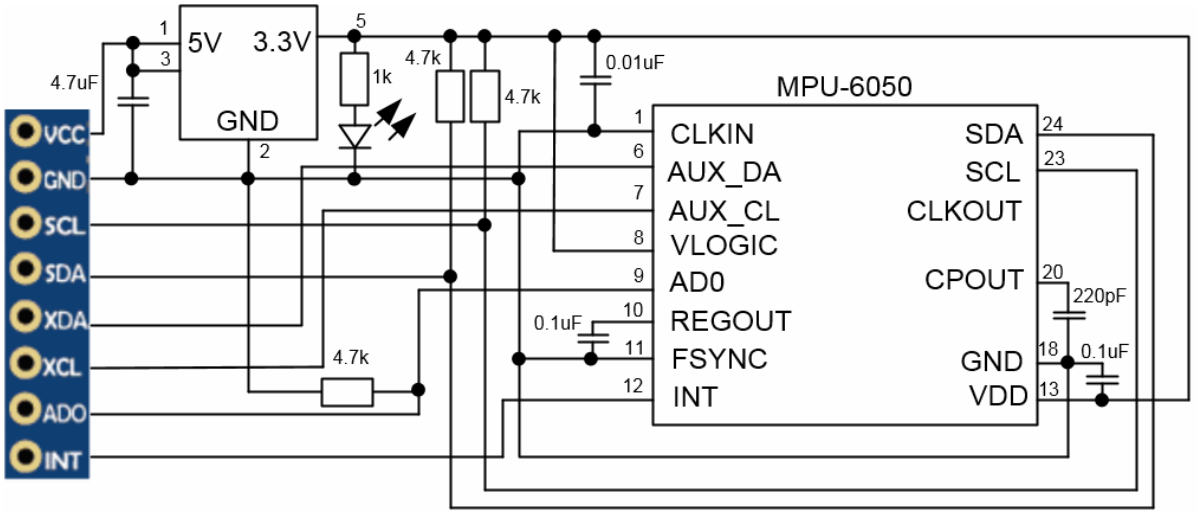



**Рисунок 2.** Электрическая принципиальная схема модуля GY-521.

**Минимизация погрешности гироскопа**

Гироскоп показывает угловую скорость. Ориентация гироскопа вычисляется интегрированием скорости. Даже при малой погрешности скорости интегральная ошибка ориентации со временем увеличивается. Ориентацию гироскопа можно уточнить по показаниям акселерометра. При фиксации микросхемы MPU6050 в горизонтальной плоскости ее ускорение по оси z равно 1g - ускорению свободного падения (9.81 м/с2). Этот вектор ускорения и его проекции на оси X и Y можно использовать для расчета ориентации.

Отклонение микросхемы от оси X и Y можно рассчитать по показаниям акселерометра:

$$\theta_x = arctg\frac{accel\_X}{\sqrt{accel\_Y^2 + accel\_Z^2}}; \qquad \theta_y = arctg\frac{accel\_Y}{\sqrt{accel\_X^2 + accel\_Z^2}}$$

Для вычисления *arctg* чаще используется функция atan2, диапазон которой ± 180° - в два раза шире, чем диапазон ± 90° функции atan (см. Рисунок 3). Добавление π к функции atan2 смещает диапазон функции (atan2(x,y)+ π)*180/π из зоны ± 180° в зону 0 .. 360 градусов со значением 180° при x=1, y=0.

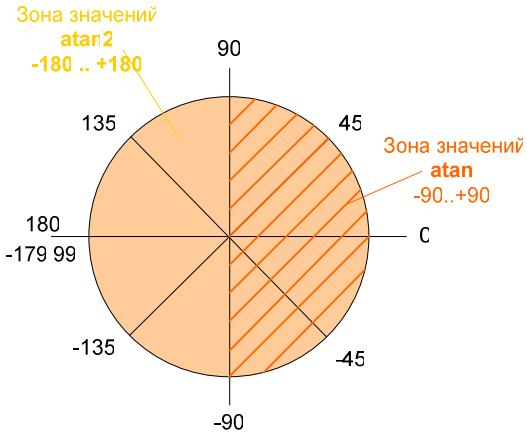



**Рисунок 3.** Диапазоны изменения функций atan и atan2 в угловых градусах.

Ускорение в м/с$^2$ вычисляется путем деления показаний акселерометра на его чувствительность. В соответствии со спецификацией, чувствительность акселерометра в диапазонах ± 2, ± 4, ± 8, ± 16g соответственно составляет 16384, 8192, 4096, 2048 бит/g.

При перемещении и вибрации датчика выходные сигналы акселерометра сильно зашумлены. Фильтрация шумов позволяет повысить точность показаний акселерометра и улучшить корректировку гироскопа.

Корректировку разворотов гироскопа в горизонтальной плоскости нельзя выполнить по показаниям датчика ускорения поскольку проекция вектора ускорения свободного падения в этом случае не изменяется. Угловое положение модуля в горизонтальной плоскости можно измерить при помощи внешнего магнитометра (компаса), подключаемого к портам AUX_DA и AUX_CL интерфейса I2C микросхемы MPU-6050.

***Примечание.*** Микросхема MPU-9150 содержит в своей структуре гироскоп MPU-6050 и встроенный магнитометр. На рынке предлагаются модули MPU-9150 на базе микросхемы MPU-9150.

**Алгоритмы коррекции показаний гироскопа**

Коррекцию показаний гироскопа осуществляют с применением фильтра Кальмана или “альфа-бета” композитного фильтра.

Вычисление углового положения гироскопа $\alpha$ включает низкочастотную фильтрацию интегрирования (в первом слагаемом) и высокочастотную фильтрацию показаний акселерометра (во втором слагаемом:

$$\alpha_{i+1} = (1-\delta)\times(\alpha_i + \omega_i \Delta t) + \delta \times a\tan 2(g_i^x, -g_i^y),$$

где $\alpha$ - угол поворота гироскопа; $\omega$ - измеряемая скорость поворота гироскопа; $g$ - проекция ускорения свободного падения на оси микросхемы; $\Delta t$ - интервал времени между считыванием показаний датчика; $\delta$ - малый коэффициент, который выбирается с учетом требуемого времени сходимости процесса вычисления угла поворота к правильному значению:

$$\tau = \frac{(1-\delta)}{\delta}\Delta t$$

Например, для $\delta = 0{,}02$ и $\Delta t = 0{,}01$ сек сходимость $\tau$ равна 0,49 сек.

**Определение расстояния по акселерометру**

При вычитании ускорения свободного падения g из модуля результирующего ускорения датчика получается собственное ускорение датчика, дважды интегрируя его по времени ($\Delta t$) можно вычислить изменение расстояния. Однако ошибка измерения расстояния по показаниям акселерометра растет со временем и особенно быстро при резких изменениях ускорения.

**Подключение модуля к микроконтроллеру Arduino UNO**

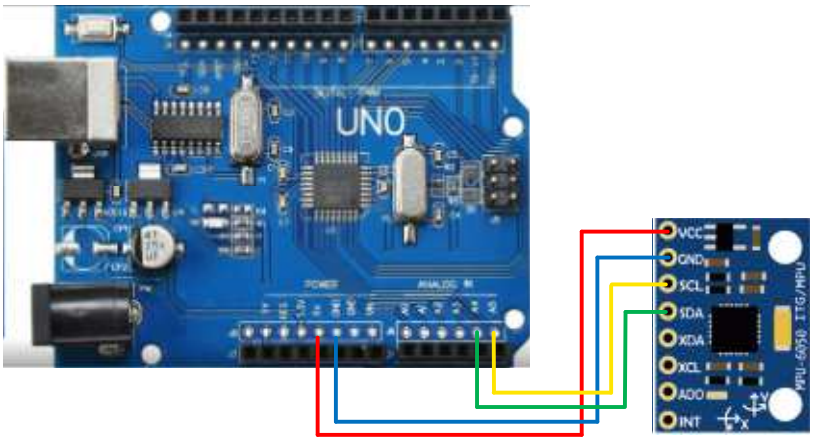



**Рисунок 4.** Схема подключения модуля GY-521 к контроллеру Arduino UNO.

Установка режимов работы (полоса пропускания, задержки, диапазоны измерения, и др.) и считывание показаний датчиков выполняется через регистры микросхемы. Спецификации основных регистров [2] показаны ниже.

РЕГИСТР 0x1A, CONFIG

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 1A | 26 | - | - | EXT_SYNC_SET[2:0] | | | DLPF_CFG[2:0] | | |

DEFAULT = 0 ==> SYNC input disabled, Max bandwidth.

| DLPF_CFG | Accelerometer ($F_s$ = 1kHz) | | Gyroscope | | |
|---|---|---|---|---|---|
| | Bandwidth (Hz) | Delay (ms) | Bandwidth (Hz) | Delay (ms) | Fs (kHz) |
| 0 | 260 | 0 | 256 | 0.98 | 8 |
| 1 | 184 | 2.0 | 188 | 1.9 | 1 |
| 2 | 94 | 3.0 | 98 | 2.8 | 1 |
| 3 | 44 | 4.9 | 42 | 4.8 | 1 |
| 4 | 21 | 8.5 | 20 | 8.3 | 1 |
| 5 | 10 | 13.8 | 10 | 13.4 | 1 |
| 6 | 5 | 19.0 | 5 | 18.6 | 1 |
| 7 | RESERVED | | RESERVED | | 8 |

РЕГИСТР 0x1B, GYRO_CONFIG

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 1B | 27 | XG_ST | YG_ST | ZG_ST | FS_SEL[1:0] | | - | - | - |

DEFAULT = 0 ==> +/- 250°/s;

| FS_SEL | Full Scale Range |
|---|---|
| 0 | ± 250 °/s |
| 1 | ± 500 °/s |
| 2 | ± 1000 °/s |
| 3 | ± 2000 °/s |

РЕГИСТР 0x1C, ACCEL_CONFIG

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 1C | 28 | XA_ST | YA_ST | ZA_ST | AFS_SEL[1:0] | | - | | |

DEFAULT = 0 ==> +/- 2g;

| AFS_SEL | Full Scale Range |
|---|---|
| 0 | $\pm 2g$ |
| 1 | $\pm 4g$ |
| 2 | $\pm 8g$ |
| 3 | $\pm 16g$ |

РЕГИСТРЫ 0x3B и 0x3C – Старший и младший байты датчика ускорения по оси X
РЕГИСТРЫ 0x3D и 0x3E – Старший и младший байты датчика ускорения по оси Y
РЕГИСТРЫ 0x3F и 0x40 – Старший и младший байты датчика ускорения по оси Z
РЕГИСТРЫ 0x41 и 0x42 – Старший и младший байты датчика температуры
РЕГИСТРЫ 0x43 и 0x44 – Старший и младший байты скорости гироскопа вокруг оси X
РЕГИСТРЫ 0x45 и 0x46 – Старший и младший байты скорости гироскопа вокруг оси Y
РЕГИСТРЫ 0x47 и 0x48 – Старший и младший байты скорости гироскопа вокруг оси Z

РЕГИСТР 0x6B, PWR_MGMT_1

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 6B | 107 | DEVICE _RESET | SLEEP | CYCLE | - | TEMP_DIS | CLKSEL[2:0] | | |

При DEVICE_RESET=1 все внутренние регистры принимают исходные (default) значения
SLEEP = 1 переводит датчик в спящий режим
При CYCLE = 1 спящий режим не установить
При TEMP_DIS = 1 датчик температуры не работает
Три бита CLKSEL устанавливают частоту работы устройства

**Запись в регистр I2C модуля (команды библиотеки <Wire.h> контроллера Arduino)**

Запись данных в регистры гироскопа выполняется следующими командами контроллера.

```
Wire.beginTransmission(MPU); // MPU - адрес микросхемы MPU-6050: 0x68
Wire.write(Addr_byte);       // Addr_byte - адрес регистра
Wire.write(Data_byte);       // Data_byte - байт записываемых данных в регистр Addr_byte
Wire.endTransmission(true);  // Окончание записи
```

**Чтение регистра (ов) I2C модуля**

Чтения нескольких регистров (N) относительно базового регистра (Addr_base) гироскопа выполняется в следующей последовательности.

```
Wire.beginTransmission(MPU); // MPU - адрес микросхемы MPU-6050: 0x68
Wire.write(Addr_base);       // Addr_base – базовый адрес
Wire.endTransmission(false);

Wire.requestFrom(MPU,N,true);  // N – количество последовательно считываемых регистров
AcX=Wire.read();               // чтение регистра Addr_base при N ≥ 1
AcX=Wire.read();               // чтение следующего регистра Addr_base+1 при N ≥ 2
…
```

**Чтение и передача выходных данных датчика в COM порт по запросу MATLAB**

Ниже показана программа контроллера Arduino, которая опрашивает COM порт и после получения ASCII кода 1 через канал I2C посылает запрос гироскопу (адрес 0x68) на прием данных 14 регистров (адреса с 0x3B по 0x48), считывает содержимое регистров и передает их в COM порт. Каждая пара регистров содержит старший и младший байты датчиков. Сначала считываются значения ускорения по X, Y и Z координатам, затем датчика температуры и, наконец, угловой скорости гироскопа вокруг X, Y и Z осей.

```
#include<Wire.h>
  const int MPU=0x68;  // I2C address of the MPU-6050
  int mode;            // request for reading

 void setup(){
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);  // PWR_MGMT_1 register
  Wire.write(0);     //  wakes up MPU-6050
  Wire.endTransmission(true);

  Serial.begin(115200);
 }
 void loop(){
  if (Serial.available() >0) {
    if (Serial.read()==1) {
       Wire.beginTransmission(MPU);
       Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
       Wire.endTransmission(false);
```

```
    Wire.requestFrom(MPU,14,true);  // request a total of 14 registers
                                    // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
                                    // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
                                    // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
                                    // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
                                    // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
                                    // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
                                    // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
    for (int i = 0; i < 14; i++) Serial.write(Wire.read());
  }
 }
}
```

Пример MATLAB программы приема и обработки данных гироскопа приведен ниже. Программа открывает COM порт для связи с контроллером Arduino, ожидает 3 секунды для гарантированного завершения процедуры открытия COM порта, затем k раз повторяет следующую последовательность:

- посылает в COM порт единицу – запрос контроллеру на данные гироскопа,
- получает 14 байт гироскопа от контроллера через COM порт,
- переставляет байты пары байт местами,
- переводит пары в слова,
- вычисляет значение ускорения по трем координатам, температуру и угловые скорости по трем координатам,
- выводит показания датчика в командное окно MATLAB

по окончанию циклов приема данных программа закрывает COM порт:

```
close all; clear all;
% create port
COM_Port_Number = 'COM5';

s=serial(COM_Port_Number,'Baudrate',115200); % 9600, 115200
fopen(s);
pause(3);              % waits COM port opening

DC_mode = 1;
for k = 1:10
   fwrite(s,typecast(uint8(DC_mode), 'uint8'));
   %pause(0.005);
   while (s.BytesAvailable)
      B = fread(s, 14 , 'uint8');
      B  =  double(typecast(uint8(B([2,1,4,3,6,5,8,7,10,9,12,11,14,13])),'int16'));  % reformat

      ACCEL_X = B(1)/16384;
      ACCEL_Y = B(2)/16384;
      ACCEL_Z = B(3)/16384;
      TEMP = B(4)/340+36.53;
      GYRO_X = B(5)/131;
      GYRO_Y = B(6)/131;
      GYRO_Z = B(7)/131;

      disp(sprintf('ACCEL X/Y/Z=%4.3f/%4.3f/%4.3f g; TEMP=%4.2f deg; GYRO X/Y/Z = %4.2f/%4.2f/%4.2f
deg/sec',ACCEL_X,ACCEL_Y,ACCEL_Z,TEMP,GYRO_X,GYRO_Y,GYRO_Z));
```

```
    end
%    pause(0.1);
end

% close port
fclose(s)
delete(s)
clear s
```

Используемые в программе коэффициенты для вычисления ускорения (16384), температуры (340 и 36.53) и скорости гироскопа (131) взяты из спецификации на гироскоп MPU-6050 для диапазона ускорения ± 2g и скорости ± 250 °/с.

Выводимые программой MATLAB на дисплей компьютера данные, находящегося в покое датчика, имеют следующий формат.

ACCEL X/Y/Z=0.032/-0.029/0.973 g; TEMP=27.54 deg; GYRO X/Y/Z = -1.60/0.91/0.08 deg/sec
ACCEL X/Y/Z=0.032/-0.028/0.986 g; TEMP=27.64 deg; GYRO X/Y/Z = -1.82/0.86/0.07 deg/sec
ACCEL X/Y/Z=0.039/-0.028/0.978 g; TEMP=27.68 deg; GYRO X/Y/Z = -1.76/0.90/0.02 deg/sec
ACCEL X/Y/Z=0.033/-0.031/0.986 g; TEMP=27.73 deg; GYRO X/Y/Z = -1.52/0.62/0.08 deg/sec
ACCEL X/Y/Z=0.030/-0.025/0.971 g; TEMP=27.64 deg; GYRO X/Y/Z = -1.43/0.70/0.20 deg/sec
ACCEL X/Y/Z=0.037/-0.035/0.986 g; TEMP=27.73 deg; GYRO X/Y/Z = -1.57/0.63/0.01 deg/sec
ACCEL X/Y/Z=0.032/-0.033/0.974 g; TEMP=27.64 deg; GYRO X/Y/Z = -1.60/0.86/0.14 deg/sec
ACCEL X/Y/Z=0.036/-0.031/0.985 g; TEMP=27.68 deg; GYRO X/Y/Z = -1.48/0.76/0.28 deg/sec

Без вывода на экран, скорость опроса датчика в среде MATLAB равна 3 мс.

На Рисунок 5 показаны начальные смещения сигналов покоящегося гироскопа по трем координатам, которые необходимо вычитать из последующих показаний гироскопа. В этом примере амплитуда шумов смещения достигает 40 единиц. При ошибке смещения 40 единиц или 40/131 °/сек ошибка углового положения, вычисляемого интегрированием скорости, растет примерно на 40 градусов за 130 секунд. Ошибку вычисления смещения можно минимизировать усреднением накопленных значений.

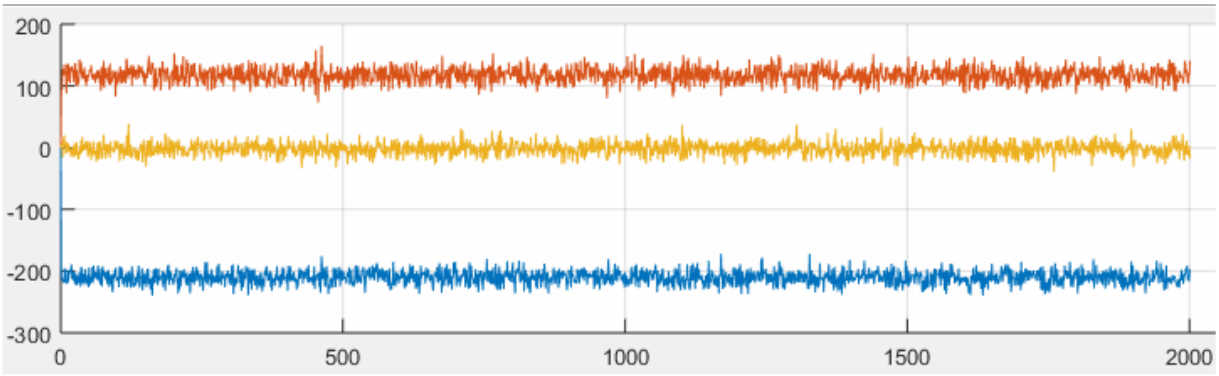



**Рисунок 5.** Зависимость неотмасштабированных показаний гироскопа от порядкового номера считывания показаний. Гироскоп лежит на столе при комнатных условиях. Среднее значение и максимальное отклонение на интервале: X/Roll/Blue = -210.2199 (-240..-172); Y/Pitch/Brown = 117.5257 (73..165); Z/Yaw/Yellow = -1.7391 (-39..+39).

**Отображение разворотов гироскопа в 2d и 3d форматах**

Программа MATLAB для вычисления углового положение гироскопа и отображения положения в 2d и 3d форматах приведена под **Рисунок 6**. На самом рисунке показан пример угловых положений гироскопа. Смещение выходного сигнала гироскопа считывается в точке (без усреднения) перед циклическим отображением данных.

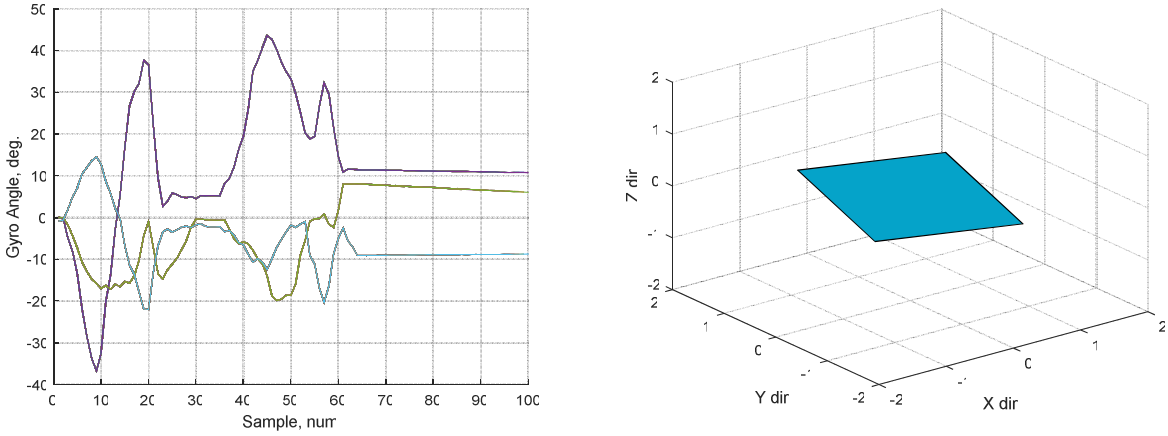



**Рисунок 6.** 2d (слева) и 3d (справа) отображение углового положения гироскопа в MATLAB.

```
close all; clear all;

% create port
COM_Port_Number = 'COM3';
s=serial(COM_Port_Number,'Baudrate',115200); % 9600, 115200
fopen(s);

% open 2D figure
figure(1);
grid;
xlabel('Sample, num');
ylabel('Gyro Angle, deg.');
hold on;

% open 3D figure
figure (2)
[xx,yy]=ndgrid([-1 1],[-1 1]);
z = zeros(2,2);
hSurface = surf(xx, yy, z); % plot the surface
axis ([-2 2 -2 2 -2 2]);    % axis size

pause(2); % waiting (in sec) for opening of COM port and figures

% initialization
DC_mode = 1;
fwrite(s,typecast(uint8(DC_mode), 'uint8'));

while ~s.BytesAvailable
end
B = fread(s, 14 , 'uint8')';
```

```
B = double(typecast(uint8(B([2,1,4,3,6,5,8,7,10,9,12,11,14,13])),'int16'));  % reformat

GYRO_ini = B(5:7);

ANG = zeros(1,3); % reset angle position
ANG_buffer(1:100,1:3) = NaN;  % reserved array

min_ANG = 0;
max_ANG = 0;

tic % Start timer
N = 100; % Loop numbers
for x = 1:N
   fwrite(s,typecast(uint8(DC_mode), 'uint8'));

   while ~s.BytesAvailable
   end
   B = fread(s, 14 , 'uint8')';
   B = double(typecast(uint8(B([2,1,4,3,6,5,8,7,10,9,12,11,14,13])),'int16'));  % reformat    ACCEL = B(1:3);

   dt = toc;   % read timer
   tic;        % start timer

   GYRO = B(5:7);

   dANG = (GYRO - GYRO_ini).*(dt/131);
   ANG = ANG + dANG;

   % plotting 2D
   figure(1);
   if ~(x>100)
      ANG_buffer(x,:) = ANG;
      plot(1:x,ANG_buffer(1:x,:)); %
   else
      ANG_buffer(end+1,:) = ANG;
      ANG_buffer(1,:) = [];
      plot((x-99:x),ANG_buffer); %
      axis([x-99 x min_ANG-1 max_ANG+1]);
   end
   min_ANG = min([min_ANG,ANG]);
   max_ANG = max([max_ANG,ANG]);

   % plotting 3D
   figure (2)
   hSurface = surf(xx, yy, z); % Plot the surface
   rotate(hSurface,[1 0 0],ANG(1)); % Roll, (around X)
   rotate(hSurface,[0 1 0],ANG(2)); % Pitch, (around Y)
   rotate(hSurface,[0 0 1],ANG(3)); % Yaw, (around Z)
   axis ([-2 2 -2 2 -2 2]);    % axis size
   xlabel('X dir'); ylabel('Y dir'); zlabel('Z dir');

   pause(0.1)
end
```

```
% close port
fclose(s)
delete(s)
clear s
```

**ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ**

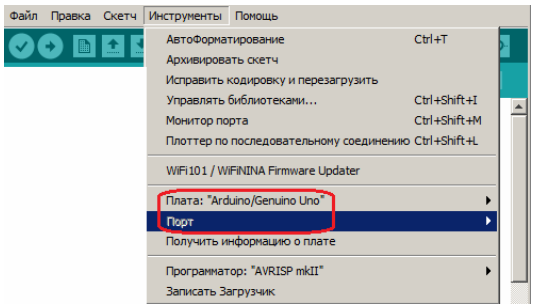
**Задание 1.** Построение макета визуализации показаний датчика пространственного положения (модуль GY-521) на базе микросхемы MPU-6050.

1. Подключите модуль GY-521 к контроллеру Arduino UNO по схеме Рисунок 2. Электрическая принципиальная схема модуля GY-521..

2. Подсоедините контроллер Arduino UNO к USB порту персонального компьютера (ПК).

3. Загрузите программу контроллера Arduino UNO из раздела этого документа: “Чтение и передача выходных данных датчика в COM порт по запросу MATLAB” в среду IDE программирования контроллеров Arduino.

4. Настройте (выберите из списка) тип Платы и номер Порта, к которому подключен контроллер:




5. В коде программы

   - замените скорость передачи последовательного порта с Serial.begin(115200); на Serial.begin(9600); это необходимо для связи с контроллером через монитором порта консоли IDE;
   - исправьте строку if (Serial.read()==1) { на if (Serial.read()==49) {, где 49 – ASCII код символа 1. Примечание, консоль IDE принимает коды в ASCII символах.

6. Загрузите программу в память контроллера, нажатием на значок:  .

7. Откройте 

8. Введите ASCII единицу в монитор порта:


9. Убедитесь, что контроллер снимает показания датчика, посылает их в канал последовательной передачи данных. Данные канала можно увидеть в ASCII кодах на мониторе порта консоли IDE.

10. Уберите исправления кода программы, выполненные в п. 5.

11. Загрузите среду MATLAB с программой из раздела “Чтение и передача выходных данных датчика в COM порт по запросу MATLAB” этого документа.

12. Настройте номер COM порта контроллера в строке COM_Port_Number = 'COM3';

13. Запустите MATLAB программу. Убедитесь, что данные датчика выводятся в окно Command Window MATLAB.

14. Разработайте программу MATLAB отображения показаний датчика на графопостроителе MATLAB, например, как показано на Рисунок 5.

15. Используя пример, приведенный под Рисунок 6, разработайте MATLAB программу отображения положения датчика в 2d и 3d формате (см. Рисунок 6).

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие данные физические величины можно измерить модулем GY-521 на базе микросхемы MPU-6050?
2. Почему ошибка измерения расстояния по показаниям датчика скорости увеличивается во времени?
3. Почему ошибка измерения расстояния по показаниям акселерометра увеличивается во времени?

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. InvenSense: MPU-6000 and MPU-6050 Product Specification Revision 3.4
2. InvenSense: MPU-6000and MPU-6050 Register Map and Descriptions Revision 4.0.
2. Dr. Bob Davidov. Компьютерные технологии управления в технических системах http://portalnp.ru/author/bobdavidov.