Модуль измерения углов наклона и поворота

WAVE-03

Руководство пользователя

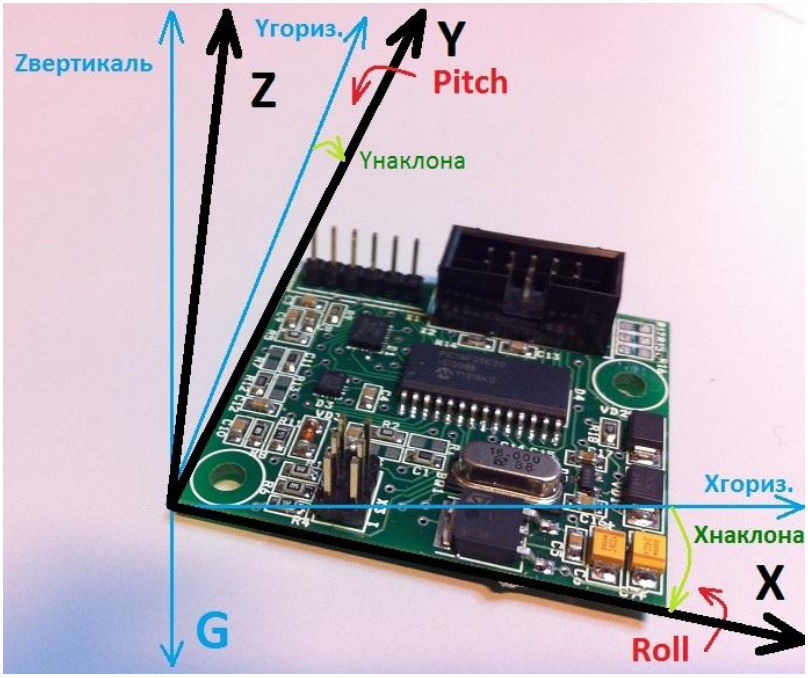


Рис.1

Данное устройство предназначено для измерения своего положения относительно горизонтальной плоскости по двум осям в диапазоне ±90º. Результатом его измерений являются либо углы наклона, либо углы Эйлера (тангаж (pitch) и крен (roll)), по выбору пользователя. Может использоваться как на стационарных, так и на движущихся объектах, например, в балансирующих роботах. Точность измерения ±1º.

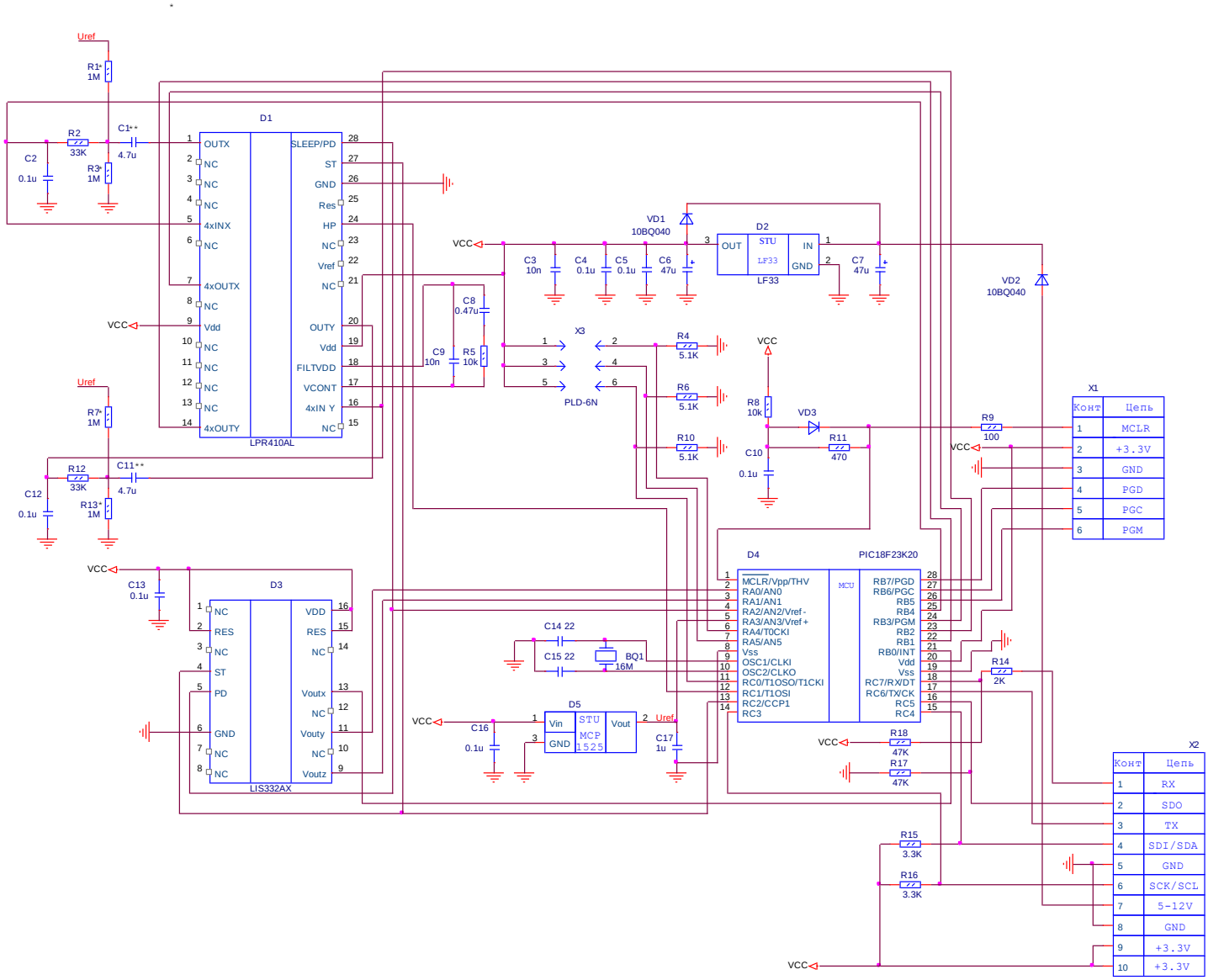
Функционально состоит из трехосевого акселерометра LIS332AX и двухосевого гироскопа LPR410AL, результаты измерений с которых поступают на процессор PIC18F25K20. В процессоре реализован цифровой фильтр, объединяющий данные с акселерометра и гироскопа с последующей передачей суммарного результата пользователю по каналу I2C или каналу UART. К модулю придается компьютерная программа, позволяющая проводить анализ поведения объекта в реальном времени, а так же задавать режимы работы модуля. Программа получает данные из модуля через COM-порт, при этом пользовательский внешний процессор может одновременно работать через I2C.

СОДЕРЖАНИЕ

1. Технические характеристики модуля.

1. Измеряет свое положение при наклоне в пределах ±90º относительно горизонтальной плоскости. Диапазон измерения углов поворота: pitch - ±90º, roll - ±180º.
2. Точность измерения: не хуже ±0.5º, типовая ±0.3º.
3. Диапазон рабочей угловой скорости: 0 – 400 º/сек.
4. Рабочее напряжение: 3.3 В.
5. Питание: 3.3 – 12 В, максимальное напряжение 16В. На плате установлен регулятор напряжения LF33, дающий возможность пользователю питать свои схемы от 3.3В.
6. Выходные порты: I2C (ведомый), UART.
7. Данные, выдаваемые пользователю (далее каналы): рассчитанные углы наклона относительно горизонтальной плоскости по осям X и Y, прямые откалиброванные результаты измерений с акселерометра по осям X, Y, Z, и с гироскопа по осям X и Y. У гироскопа на каждую ось предусмотрено два выхода, один обычный, а другой усиленный в четыре раза. Оба этих значения могут быть переданы пользователю (усиленные значения далее по тексту обозначаются как 4X и 4Y). Возможна выдача не всех данных, а только заданных пользователем. Вместо углов наклона пользователь может получать углы Эйлера (тангаж (pitch) и крен (roll)).
8. При работе по I2C данные выдаются пользователю по запросу. Возможно подключение нескольких модулей на одну шину. При этом пользователь должен задать I2C-адрес каждого модуля.
9. Частота запросов внешнего процессора по шине I2C: при запросе менее 5-ти каналов - 200Гц, при запросе 5-ти и более - 180Гц
10. При работе через UART данные могут выдаваться как по запросу, так и с заданной частотой в диапазоне от 16 до 200Гц.
11. Температурный диапазон -40... +85. (согласно техописаниям на микросхемы, реальные термоиспытания не проводились).

12. Размеры, мм: 41x47.



*R1,R3,R7,R13,R15,R16 на плате не установлены

**Вместо C1 и C11 установлены перемычки

Рис.2

2. Подключение.

Подключение модуля осуществляется через разъем X2. Разъем X1 – технологический (ICSP), для работы не используется. На разъем X2 выведены порты UART, I2C, SPI (в этой версии модуля не используется).

Назначение контактов X2:

Номер контакта	Название	Назначение
1	Rx	Вход UART
2	SDO	Выход данных SPI (не используется, запрограммирован как вход)
3	Tx	Выход UART
4	SDI/SDA	I2C данные
5	GND	Общий
6	SCK/SCL	I2C тактовые импульсы
7	5-12V	Вход питания
8	GND	Общий
9	+3,3В	Выход 3,3В или вход питания 3,3 (при отсутствии питания 5-12В)

10	+3,3В	Выход 3,3В или вход питания 3,3 (при отсутствии питания 5-12В)

Так же для жесткой конфигурации модуля используется разъем X3, контакты которого можно замыкать съемной перемычкой (джампером):

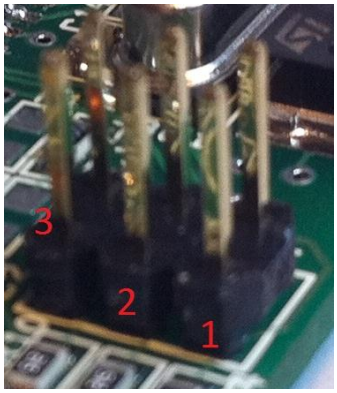


Рис.3

В положении перемычки «1», по включению модуля будет произведен сброс скорости порта UART и адреса I2C модуля в значения, установленные по умолчанию. Так же будет выключена трансляция по UART, если она была включена. При этом запись в EEPROM не производится.

В положении «2» не будет производится инициализация UART, а линия 3 разъема X2 будет запрограммирована как вход. Это необходимо для подключения нескольких модулей к шине I2C через один шлейфовый кабель.

В положение «3» перемычку ставить нельзя. Контакт, обозначенный цифрой «3» (на схеме 6-й контакт разъема X3) является контрольным выходом. При обычном режиме работы на него выводятся импульсы с частотой оцифровки (которая задается пользователем), деленной на 2. Если перемычка была установлена в положение «1» (т.е. был произведен сброс), на этом выводе будет постоянное напряжение 3,3В.

Подключение к модулю предусмотрено либо разъемами типа BLS (такие, как на сервоприводах), либо, как уже упоминалось выше, шлейфовым кабелем. Подключение через шлейфовый кабель имеет смысл при использовании нескольких модулей на одной линии I2C. При этом выхода UART разных модулей необходимо отключить, установив перемычку в положение «2». Но один из нескольких модулей, подключенных к I2C, можно оставить с активным UART с тем, что бы можно было пересылать данные компьютерной программе.

Питание модуля можно осуществлять либо от источника напряжения 5-12В, либо от 3,3В. В первом случае на выводы 9 и 10 разъема X2 будет выдаваться напряжение 3,3В , которое можно использовать для питания внешних схем. При этом необходимо помнить, что хоть для стабилизатора напряжения LF33 заявлен максимальный ток 1А, нагрузить его можно существенно меньше, т.к. он установлен без теплоотвода. При этом ток нагрузки зависит от питающего напряжения – чем выше напряжение, тем меньше ток. На практике, при напряжении питания 5В можно нагрузить примерно на 500ма, а при 12В до 200ма. При этом рекомендуется контролировать степень нагрева стабилизатора.

При питании от 3,3В , питающее напряжение подается непосредственно на выводы 9 или 10. При этом на вывод 7 подавать напряжение нельзя.

3. Принцип работы

Все измерения процессор модуля производит с заданной пользователем частотой оцифровки (16… 200 Гц). Если включен режим трансляции по UART, каждый такт частоты оцифровки производится передача па-

кета данных по UART. Также внешний процессор в любой момент может запросить данные по I2C, что позволяет пользователю одновременно работать с модулем в системе, и наблюдать за поведением объекта через компьютер. Если режим трансляции выключен, то по UART тоже можно запрашивать данные.

Модуль измеряет углы наклона – т.е. углы между текущим положением осей X и Y, и их проекциями на горизонтальную плоскость (см. Рис.1). Вместо углов наклона модуль может выдавать углы Эйлера. Формат выдачи углов – двухбайтный int с точностью до десятичного знака. (Значение угла перед выдачей умножается на 10.) Т.е. если измеренный угол, допустим, равен 43,7, то будет выдано значение 437. Данные с акселерометра и гироскопа также выдаются формате двухбайтный int. При этом, из измеренным АЦП значения будет вычитаться значение нулевого смещения. Например, если по оси X акселерометра было измерено значение 600, а значение смещения равно 612, то будет выдано 600-612 = -12.

Расчет результатов производится по показаниям акселерометра и гироскопа, которые объединяются в цифровом фильтре процессора. Объединением данных с акселерометра и гироскопа можно управлять, меняя весовой коэффициент. Этот коэффициент определяет, какие данные имеют большее значение в результате. Коэффициент может меняться в пределах 0… 20. 0 – результат считается только по показаниям акселерометра, 20 – только гироскопа. При изготовлении задается коэффициент = 16.

Более подробно про использование устройства и про углы Эйлера можно прочитать на сайте производителя www.robowell.ru

4. Калибровка

При изготовлении модуля выполняются начальные калибровки. Однако, в изделие модуль будет установлен с некоторой погрешностью, т.е. его оси не смогут точно совпасть с осями изделия. Поэтому, после установки его необходимо откалибровать, привязав оси модуля к осям устройства, в которое он установлен. Калибровки можно производить как подачей команд с внешнего процессора по I2C или по UART, так и с помощью компьютерной управляющей программы.

В модуле предусмотрены следующие виды калибровок:

1. Калибровка акселерометра в горизонтальной плоскости. Для этого устройство с модулем устанавливается в горизонтальной плоскости, и устройству подается команда CALIBRATE_ACC (см. описания команд). Далее процессор модуля производит измерения показаний акселерометра, усредняет их, и определяет напряжение смещения для каждой оси. Для осей X и Y определяется смещение, соответствующее нулю, а для оси Z -g. При этом после калибровки диапазон измерения сузится на величину реальной ошибки установки модуля. Например, если модуль был установлен с ошибкой по оси X в 1°, то вместо диапазона ±90° будет 90… -89 (или 89… -90, в зависимости от направления ошибки).
2. Калибровка гироскопа. При неподвижном модуле производится определение нулевого уровня гироскопа. Эту калибровку после установки производить необходимости нет, т.к. для привязки к месту установки она не имеет значения. Выполняется командой CALIBRATE_GYRO.
3. Калибровка соосности акселерометра и устройства. Это калибровка расхождения их осей в горизонтальной плоскости. Она проявляется в том, что при наклоне только, допустим по оси X, ось Y будет не совсем горизонтальна, что приведет к появлению сигнала по оси Y. Для выполнения этой калибровки по оси X, модуль необходимо сначала наклонить по оси Y в положение 90°, и запомнить показания угла по оси X. Затем его нужно наклонить в положение -90° и сравнить текущий угол оси X с предыдущим. Если текущий угол больше, то необходимо подать команду CALIBRATE_ACCX. Если меньше, то модуль нужно опять наклонить на 90°, и затем подать команду CALIBRATE_ACCX. Аналогичную операцию нужно выполнить и для оси X, а затем подать команду CALIBRATE_ACCY. Если модуль планируется использовать при небольших углах отклонения, эту калибровку можно не производить.

В подобных устройствах также желательно производить калибровку соосности гироскопа и устройства. При этом устройство должно выполнить поворот модуля по одной из осей с максимальной скоростью (400°/сек) и дать команду на калибровку другой оси. Вследствие сложности осуществления такой калибровки, она не предусмотрена. Описанных выше калибровок достаточно для достижения заявленной точности. Однако, модуль все равно рекомендуется устанавливать настолько точно, насколько это возможно.

В модуле также предусмотрена принудительная загрузка нулевых калибровочных значений как для акселерометра, так и для гироскопа. Это может потребоваться, например, при использовании устройства в широком температурном диапазоне для компенсации ухода нуля при изменении температуры.

5. Протокол обмена

Принцип обмена данными по портам UART и I2C одинаковый. Внешний процессор работает в режиме «ведущий» и может либо запросить данные, либо просто передать их. В первом случае внешний процессор посылает пакет с командой и ждет ответ. Во втором посылает пакет с командой и с данными. Исключение составляет работа процессора устройства в режиме трансляции по UART. Однако, даже в этом случае, благодаря тому, что модуль UART процессора является дуплексным, процессор может принять внешнюю команду в любой момент. Каждый пакет завершается байтом контрольной суммы, который принимающая сторона должна проверить.

Все заданные пользователем настройки модуля могут быть записаны в EEPROM. Сначала должны быть заданы все настройки, а затем подана команда записи в EEPROM. Необходимо избегать частых (особенно циклических) выполнений команды записи во избежание выхода EEPROM из строя. Так же настройки могут быть заданы с помощью управляющей программы.

Ниже приводится список команд модуля. Он общий для UART и для I2C. Существует только одна команда, которая не выполняется по UART: GET_STATUS_I2C.

Команды, передаваемые в модуль:

Название	Код	Описание
SET_ACTIVE_CHANNELS	1	Задание каналов, передаваемых внешнему процессору
CALIBRATE_ACC	2	Калибровка нулевого смещения акселерометра
CALIBRATE_GYRO	3	Калибровка нулевого смещения гироскопа
CALIBRATE_ACCX	4	Калибровка несовпадения осей акселерометра, ось X
CALIBRATE_ACCY	5	Калибровка несовпадения осей акселерометра, ось Y
SET_ACCX_BIAS	6	Непосредственное задание значения смещения нуля акселерометра, ось X
SET_ACCY_BIAS	7	Непосредственное задание значения смещения нуля акселерометра, ось Y
SET_ACCZ_BIAS	8	Непосредственное задание значения смещения нуля акселерометра, ось Z
SET_GYROX_BIAS	9	Непосредственное задание значения смещения нуля гироскопа, ось X
SET_GYROY_BIAS	0xa	Непосредственное задание значения смещения нуля гироскопа, ось Y
SET_GYRO4X_BIAS	0xb	Непосредственное задание значе-

		ния смещения нуля гироскопа, ось 4X
SET_GYRO4Y_BIAS	0xc	Непосредственное задание значения смещения нуля гироскопа, ось 4Y
WRITE_TO_EEPROM	0xd	Запись изменений в EEPROM
SET_FREQUENCY	0xe	Задание частоты оцифровки
GET_OPTIONS	0xf	Запрос настроек модуля
BROADCAST_ON	0x10	Включить трансляцию по UART
BROADCAST_OFF	0x11	Выключить трансляцию по UART
GET_DATA_PACKET	0x12	Запрос пакета с результатами измерения
SET_EUSART_BAUD_RATE	0x13	Изменение скорости UART
ST_ON	0x14	Включение самотестирования
ST_OFF	0x15	Выключение самотестирования
SET_I2C_ADDR	0x16	Задание адреса I2C
GET_STATUS_I2C	0x17	Запрос слова состояния
RESET	0x18	Перезапуск модуля (на всякий случай)
SET_GYRO_ACCX_RATIO	0x19	Задание весового коэффициента акселерометр-гироскоп
SET_TILT_ANGLE	0x1a	Задание типа получаемых данных (угол наклона или углы Эйлера)
GET_VERSION	0x1b	Запрос версии программы процессора

Команды, принимаемые из модуля:

Название	**Код**	**Описание**
DATA_PACKET	0x81	Пакет с результатами измерений
CRC_ERROR	0x82	Ошибка контрольной суммы
GOOD_RESULT	0x83	Успешное выполнение команды
BAD_RESULT	0x84	Команда не выполнена
EEPROM_ERROR	0x85	Ошибка записи в EEPROM
UNRECOGNIZED_COMMAND	0x86	Неизвестная команда
OPTIONS	0x87	Пакет с настройками модуля
VERSION	0x88	Пакет с номером версии программы процессора

5.1 Обмен данными по UART

Настройки порта при изготовлении: 115200, 8 бит, без проверки четности.

Формат пакета, передаваемого или принимаемого по UART следующий:

Номер байта:	0	1	2	3	4	5		N+3	N+4
Название, или значение:	0xFF	0xFF	N+1	CMD	DATA1	DATA 2	…	DATA N	CHK

Где:

CMD – код команды, DATA 1… DATA N – N передаваемых байт данных, CHK – контрольная сумма. Во втором байте N+1 – это число передаваемых байт данных + 1.

Контрольная сумма вычисляется со 2-го байта пакета по N+3 включительно. Функция вычисления контрольной суммы:

```
unsigned char CalcCheckSumm(unsigned char N, unsigned char *Mass)

{ unsigned int Summ=0,j,n=N;

  for (j=0;j<n;j++)

  Summ=Summ+Mass[j];

  Summ=~Summ;

  return (unsigned char) Summ;}
```

5.1.1 Примеры передачи пакета в модуль.

Пусть требуется включить трансляцию данных по UART. Тогда в модуль необходимо передать следующий пакет:

0xff,0xff,1,0x10 (Команда BROADCAST_ON),0xee – контрольная сумма

Если пакет записывать, например, в массив TrmMass, то на Си это будет выглядеть так:

```
TrmMass[0]=0xff;
TrmMass[1]=0xff;
TrmMass[2]=1;
TrmMass[3]=BROADCAST_ON;
TrmMass[4]=CalcCheckSumm(TrmMass[2]+1, &TrmMass[2]);
Trm(TrmMass);//Функция передачи
```

Пусть требуется задать частоту оцифровки 100Гц. Тогда в модуль необходимо передать следующий пакет:

0xff,0xff,2,0xe (Команда SET_FREQUENCY),100,0x8b – контрольная сумма

Аналогично:

```
TrmMass[0]=0xff;
TrmMass[1]=0xff;
TrmMass[2]=2;//N
TrmMass[3]=SET_FREQUENCY;
TrmMass[4]=100;
TrmMass[5]=CalcCheckSumm(TrmMass[2]+1, &TrmMass[2]);
Trm(TrmMass);//Функция передачи
```

5.1.2 Пример передачи пакета из модуля.

Подробно пакет данных будет рассмотрен при описании команды DATA_PACKET. В данном примере передаются только данные об углах наклона. Перед ними в пакет включены два байта слова состояния процессора модуля.

```
TrmMass[0]=0xff;
TrmMass[1]=0xff;
TrmMass[2]=7;
TrmMass[3]=DATA_PACKET;
TrmMass[4]=3;//Байт состояния, младший байт
```

```
TrmMass[5]=0x9e; //Байт состояния, старший байт
TrmMass[6]=0xde;//Угол наклона X, младший байт
TrmMass[7]=1; //Угол наклона X, старший байт
TrmMass[8]=7; //Угол наклона Y, младший байт
TrmMass[9]=0; //Угол наклона Y, старший байт
TrmMass[10]=CalcCheckSumm(TrmMass[2]+1, &TrmMass[2]);//0xf0
Trm(TrmMass);//Функция передачи
```

5.1.3 Пакеты, передаваемые модулю.

5.1.3.1 SET_ACTIVE_CHANNELS.

Действие: Задание каналов, которые будут передаваться в пакете данных.

Код: 1

Число байт данных (далее N): 2

Если соответствующий бит установлен, канал передается:

Байт 1								Байт 2							
Бит7	Бит6	Бит5	Бит4	Бит3	Бит2	Бит1	Бит0	Бит 7	Бит6	Бит5	Бит4	Бит3	Бит2	Бит1	Бит0
Гиро 4X	Гиро Y	Гиро X	Акс. Z	Акс. Y	Акс. X	Y или roll	X или pitch	-	-	-	-	-	-	-	Гиро 4Y

Биты байта1:

0 - Угол наклона по оси X, или угол Эйлера – pitch (задается командой SET_TILT_ANGLE).

1 - Угол наклона по оси Y, или угол Эйлера – roll (задается командой SET_TILT_ANGLE).

2 – Данные акселерометра с учетом калибровочного нулевого смещения, ось X. Т.е. если значение акселерометра обозначить, как AccX, значение акселерометра при нулевом ускорении как AccX0, то будет передаваться AccX - AccX0. Нулевые смещения можно получить в пакете OPTIONS.

3 – Данные акселерометра с учетом калибровочного нулевого смещения, ось Y.

4 – Данные акселерометра с учетом калибровочного нулевого смещения, ось Z.

5 – Данные гироскопа с учетом калибровочного нулевого смещения, ось X.

6 – Данные гироскопа с учетом калибровочного нулевого смещения, ось Y.

7 – Данные гироскопа с учетом калибровочного нулевого смещения, ось 4X.

Бит байта2:

0 – Данные гироскопа с учетом калибровочного нулевого смещения, ось 4Y.

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.2 CALIBRATE_ACC

Действие: Запуск калибровки акселерометра по 3-м осям. Подробно см. п.4

Код: 2

N: 0

Команда выполняется около двух секунд. После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.3 CALIBRATE_GYRO

Действие: Запуск калибровки гироскопа. Подробно см. п.4

Код: 3
N: 0
Команда выполняется около трех секунд. После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.4 CALIBRATE_ACCX

Действие: Запуск калибровки несовпадения осей акселерометра, ось X. Подробно см. п.4
Код: 4
N: 0

Команда выполняется около двух секунд. После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.5 CALIBRATE_ACCY

Действие: Запуск калибровки несовпадения осей акселерометра, ось Y. Подробно см. п.4
Код: 5
N: 0

Команда выполняется около двух секунд. После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.6 SET_ACCX_BIAS

Действие: Непосредственное задание значения смещения нуля акселерометра, ось X.
Код: 6
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.7 SET_ACCY_BIAS

Действие: Непосредственное задание значения смещения нуля акселерометра, ось Y.
Код: 7
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.8 SET_ACCZ_BIAS

Действие: Непосредственное задание значения смещения нуля акселерометра, ось Z.
Код: 8
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.9 SET_GYROX_BIAS

Действие: Непосредственное задание значения смещения нуля гироскопа, ось X.
Код: 9
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.10 SET_GYROY_BIAS

Действие: Непосредственное задание значения смещения нуля гироскопа, ось Y.
Код: 0xa
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.11 SET_GYRO4X_BIAS

Действие: Непосредственное задание значения смещения нуля гироскопа, ось 4X.
Код: 0xb
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.12 SET_GYRO4Y_BIAS

Действие: Непосредственное задание значения смещения нуля гироскопа, ось 4Y.
Код: 0xc
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.13 WRITE_TO_EEPROM

Действие: Производится запись всех изменений в EEPROM.
Код: 0xd
N: 0

После записи процессор модуля считывает записанные данные и сравнивает с заданными. В случае совпадения посылает пакет GOOD_RESULT, в противном случае - EEPROM_ERROR.

5.1.3.14 SET_FREQUENCY

Действие: Задание частоты работы модуля. Вместе с командой передается байт со значением частоты в пределах 16… 200.
Код: 0xe
N: 1

Если значение частоты выходит за заданные пределы, модуль посылает пакет BAD_RESULT, и частота не меняется. В противном случае – GOOD_RESULT.

5.1.3.15 GET_OPTIONS

Действие: Запрос настроек модуля.
Код: 0xf
N: 0

В ответ модуль высылает пакет OPTIONS со всеми байтами настроек. См. п. 5.1.4.8

5.1.3.16 BROADCAST_ON

Действие: Включение трансляции заданных пакетом SET_ACTIVE_CHANNELS данных по UART.
Код: 0x10
N: 0

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.17 BROADCAST_OFF

Действие: Выключение трансляции данных по UART.
Код: 0x11
N: 0
После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.18 GET_DATA_PACKET

Действие: Запрос на пакет данных, заданных пакетом SET_ACTIVE_CHANNELS.
Код: 0x12
N: 0
В ответ модуль посылает пакет DATA_PACKET с данными измерений.

5.1.3.19 SET_EUSART_BAUD_RATE

Действие: Этой командой производится непосредственная запись значений в регистры процессора, отвечающие за скорость работы UART.
Код: 0x13
N: 4
Подробно про работу последовательного порта см. pdf на процессор. Ниже описан только механизм задания скорости.
К заданию скорости порта имеют отношение следующие регистры и биты:
SPBRG – делитель внутренней частоты
SPBRGH – старший байт делителя
Бит BRG16 – если установлен, то используются оба байта делителя, если сброшен, только – SPBRG.
Бит BRGH – определяет формулу, по которой рассчитывается скорость.
Формулы для расчета скорости приведены в таблице:

Биты		Формула	
BRG16	BRGH	Скорость	Значение пары SPBRGH/SPBRG
0	0	Fosc/[64(n+1)]	Fosc/(64*Baud)-1
0	1	Fosc/[16(n+1)]	Fosc/(16*Baud)-1

1	0	Fosc/[16(n+1)]	Fosc/(16*Baud)-1
1	1	Fosc/[4(n+1)]	Fosc/(4*Baud)-1

Где Fosc=64000000, n – значение регистров SPBRGH/SPBRG, Baud – требуемое значение скорости
При изготовлении задается скорость 115200, BRG16=1, BRGH=1.
Таким образом, по приведенным формулам получаем:

SPBRGH/SPBRG = 64000000/(4*115200)-1 ≈ 138
Или:
SPBRG=138
SPBRGH=0

И обратно, подсчет ошибки:
Baud = 64000000/[4*(138+1)] = 115107,91

Ошибка = (115107,91 - 115200)/ 115200 = -0,08%

Значения этих регистров и битов можно посмотреть и изменить в управляющей программе.

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.20 ST_ON

Действие: Включение режима самотестирования акселерометра и гироскопа.
Код: 0x14
N: 0
После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.21 ST_OFF

Действие: Выключение режима самотестирования акселерометра и гироскопа.
Код: 0x15
N: 0
После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.22 SET_I2C_ADDR

Действие: Задание I2C slave-адреса модуля. Вместе с командой передается байт адреса
Код: 0x16
N: 1
После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.23 RESET

Действие: Перезапуск процессора модуля.
Код: 0x18
N: 0
После выполнения команды модуль ничего не посылает.

5.1.3.24 SET_GYRO_ACCX_RATIO

Действие: Задается весовой коэффициент акселерометр-гироскоп. Его значение определяет, какие данные больше участвуют в расчете углов – данные с акселерометра или с гироскопа. Диапазон значений 0… 20. При значении 0 расчет будет производиться только по данным с акселерометра, при 20 – только с гироскопа. При изготовлении задается значение 16.

Код: 0x19

N: 1

Если значение байта данных больше 20, то модуль передает BAD_RESULT, в противном случае GOOD_RESULT.

5.1.3.25 SET_TILT_ANGLE

Действие: Задание типа передаваемых данных – углы наклона или тангаж и крен. Для задания углов в байте данных необходимо передать 1, в противном случае 0.

Код: 0x1a

N: 1

После выполнения команды модуль посылает пакет GOOD_RESULT.

5.1.3.26 GET_VERSION

Действие: Запрос версии программы процессора.

Код: 0x1b

N: 0

В ответ модуль посылает пакет VERSION с байтом версии.

5.1.4 Пакеты, принимаемые из модуля.

5.1.4.1 DATA_PACKET

Действие: Передача данных из модуля.

Код: 0x81

N: 2… 20

Байты, передаваемые модулем (следом за командой):

1. Младший байт слова состояния.
2. Старший байт слова состояния.
3. Значение угла наклона оси X (или угол крена, в зависимости от бита Tilt), младший байт
4. Значение угла наклона оси X (или угол крена, в зависимости от бита Tilt), старший байт
5. Значение угла наклона оси Y (или угол тангажа, в зависимости от бита Tilt), младший байт
6. Значение угла наклона оси Y (или угол тангажа, в зависимости от бита Tilt), старший байт
7. Значение акселерометра X, младший байт
8. Значение акселерометра X, старший байт
9. Значение акселерометра Y, младший байт
10. Значение акселерометра Y, старший байт
11. Значение акселерометра Z, младший байт
12. Значение акселерометра Z, старший байт
13. Значение гироскопа X, младший байт
14. Значение гироскопа X, старший байт

15. Значение гироскопа Y, младший байт
16. Значение гироскопа Y, старший байт
17. Значение гироскопа 4X, младший байт
18. Значение гироскопа 4X, старший байт
19. Значение гироскопа 4Y, младший байт
20. Значение гироскопа 4Y, старший байт

Если какие-то каналы не активны, то соответствующие значения не передаются, а передаются следующие из этого списка

Биты слова состояния:

Байт 1								Байт 2							
Бит7	Бит6	Бит5	Бит4	Бит3	Бит2	Бит1	Бит0	Бит 7	Бит6	Бит5	Бит4	Бит3	Бит2	Бит1	Бит0
Гиро 4X	Гиро Y	Гиро X	Акс. Z	Акс. Y	Акс. X	Y или roll	X или pitch	Tilt	I2C_Last Cmd_Err	I2C_CHK_Err	I2C_BUSY	BRG 16	BRGH	Broad cast	Гиро 4Y

Первые девять бит показывают активный канал. Если бит установлен, канал активен. Остальные – это биты состояния процессора модуля.

Биты байта1:

0 - Угол наклона по оси X, или угол Эйлера – pitch.

1 - Угол наклона по оси Y, или угол Эйлера – roll.

2 – Данные акселерометра с учетом калибровочного нулевого смещения, ось X. Т.е. если значение акселерометра обозначить, как AccX, значение акселерометра при нулевом ускорении как AccX0, то будет передаваться AccX - AccX0. Нулевые смещения можно получить в пакете OPTIONS.

3 – Данные акселерометра с учетом калибровочного нулевого смещения, ось Y.

4 – Данные акселерометра с учетом калибровочного нулевого смещения, ось Z.

5 – Данные гироскопа с учетом калибровочного нулевого смещения, ось X.

6 – Данные гироскопа с учетом калибровочного нулевого смещения, ось Y.

7 – Данные гироскопа с учетом калибровочного нулевого смещения, ось 4X.

Биты байта2:

0 – Данные гироскопа с учетом калибровочного нулевого смещения, ось 4Y.

1 – Broadcast Если установлен, по порту UART происходит постоянная передача пакетов DATA_PACKET с частотой равной частоте работы модуля (задается пакетом SET_FREQUENCY).

2, 3 – биты настройки скорости модуля UART: BRGH и BRG16. Подробно см. п. 5.1.3.19 .

4 – I2C_BUSY – установлен, когда выполняется команда, поданная по I2C (например, во время калибровки). По завершении выполнения – сбрасывается.

5 – I2C_CHK_Err – установлен, если при приеме последнего пакета была обнаружена ошибка контрольной суммы. Если ошибки не было, этот бит равен 0.

6 – I2C_LastCmd_Err – установлен, если в процессе исполнения команды по I2C была ошибка, в противном случае равен 0.

7 – Tilt. Если установлен, то в пакете данных передаются углы наклона. Если сброшен, то углы Эйлера. Задается командой SET_TILT_ANGLE.

5.1.4.2 CRC_ERROR

Действие: Передается, если принят пакет с ошибкой.

Код: 0x82

N: 0

5.1.4.3 GOOD_RESULT

Действие: Передается в случае успешного завершения команды.

Код: 0x83
N: 0

5.1.4.4 BAD_RESULT

Действие: Передается в случае ошибки при исполнении команды.
Код: 0x84
N: 0

5.1.4.5 EEPROM_ERROR

Действие: Передается в случае ошибки записи в EEPROM.
Код: 0x85
N: 0

5.1.4.6 UNRECOGNIZED_COMMAND

Действие: Передается в случае принятия неизвестной команды.
Код: 0x86
N: 0

5.1.4.7 OPTIONS

Действие: Передача настроек модуля.
Код: 0x87
N: 21

После команды передаются следующие байты:
1. Частота оцифровки
2. Калибровочное значение смещения нуля акселерометра X. Младший байт.
3. Калибровочное значение смещения нуля акселерометра X. Старший байт.
4. Калибровочное значение смещения нуля акселерометра Y. Младший байт.
5. Калибровочное значение смещения нуля акселерометра Y. Старший байт.
6. Калибровочное значение смещения нуля акселерометра Z. Младший байт.
7. Калибровочное значение смещения нуля акселерометра Z. Старший байт.
8. Калибровочное значение смещения нуля гироскопа X. Младший байт.
9. Калибровочное значение смещения нуля гироскопа X. Старший байт.
10. Калибровочное значение смещения нуля гироскопа Y. Младший байт.
11. Калибровочное значение смещения нуля гироскопа Y. Старший байт.
12. Калибровочное значение смещения нуля гироскопа 4X. Младший байт.
13. Калибровочное значение смещения нуля гироскопа 4X. Старший байт.
14. Калибровочное значение смещения нуля гироскопа 4Y. Младший байт.
15. Калибровочное значение смещения нуля гироскопа 4Y. Старший байт.
16. Слово состояния процессора, байт 1, см. п. 5.1.4.1
17. Слово состояния процессора, байт 2
18. SPBRG
19. SPBRGH
20. Адрес I2C
21. Весовой коэффициент акселерометр/гироскоп

5.1.4.8 VERSION

Действие: Передается версия программы процессора модуля. Передается одним байтом
Код: 0x88
N: 1

В старшей половине байта передается старшее число версии, в младшей – младшее.
Например, байт 0x12 означает версию 1.2

5.1.5 Пример функции обработки прерывания принимаемых данных по UART.

```
//Глобальные переменные

unsigned char GettingPacket_FL=0;//Флаг приема пакета
unsigned char BtRcPrev=0;//Регистр для хранения последнего принятого по UART байта
unsigned char CntRec;//Счетчик принимаемых байт
unsigned char PacketRec[30];//Массив для записи принимаемого пакета
unsigned char CHK_Err=0;   //Флаг ошибки контрольной суммы
unsigned char UARTPacket=0;//Флаг принятого пакета

//Функция формирует принятый пакет в массиве PacketRec и устанавливает флаг UARTPacket.
//Основная программа должна этот флаг обработать и сбросить
//При ошибке контрольной суммы устанавливается флаг CHK_Err

void __interrupt _UARTRXInterrupt()
{
unsigned char BtRc;

       BtRc=ReadPORT();//Чтение байта из регистра порта


       if (!GettingPacket_FL)//Этот флаг установлен, если ведется прием пакета
              {
              if ((BtRc==BtRcPrev)&&(BtRc==0xff))
                     {
                     BtRcPrev=0;                        //Принято два байта 0xff подряд, начать прием
пакета
                     GettingPacket_FL=1;
                     CntRec=0;
                     }
              else
                     {
                     BtRcPrev=BtRc;                     //Запомнить принятый байт
                     }
              }
       else
              {
        if (CntRec>30)
            GettingPacket_FL=0;          //Если счетчик принимаемых байт превысил размер массива, зна-
чит ошибка - прервать прием
        else
            {
                     PacketRec[CntRec]=BtRc; //Запись байта в массив пакета
                     CntRec++;                          //Инкремент счетчика принятых байт
                     if (CntRec>(PacketRec[0]+1))//В PacketRec[0] будет находиться число N+1
                            {
                            GettingPacket_FL=0;  //Приняты все байты
                  if (CalcCheckSumm(PacketRec[0]+1, PacketRec)!=PacketRec[PacketRec[0]+1])//проверка кон-
трольной суммы
```

```
                        {
                        CHK_Err=1;          //Ошибка контрольной суммы
                        }
                    else
                        {
                        UARTPacket=1; //Флаг принятого пакета
                        }
                    }
                }
            }
}
```

Пример передачи по UART см. в проекте примеров I2C. В нем данные с модуля считываются по I2C, а затем отправляются по UART в компьютер.

5.2 Обмен данными по I2C

Настройки порта I2C следующие:
Ведомый, адрес 7 бит, Slew Rate Control включен, адрес = 1
Скорость типовая - 100 кбит/с, максимальная 200 кбит/с.
Подробнее о работе порта I2C можно узнать из pdf на процессор. Здесь же описан только алгоритм, имеющий отношение к модулю пространственной ориентации. Для лучшего понимания работы алгоритма обмена по I2C рекомендуется изучить примеры функций с сайта производителя www.robowell.ru

5.2.1 Передача пакета в модуль.

Формат передаваемого пакета:

Номер байта:	0	1	2	3	4		N+2	N+3
Название, или значение:	I2C_addr <<1	N+1	CMD	DATA1	DATA2	…	DATAN	CHK

CMD – код команды, DATA 1… DATA N – N передаваемых байт данных, CHK – контрольная сумма. В нулевом байте N+1 – это число передаваемых байт данных + 1, I2C_addr – адрес модуля, сдвинутый на 1 бит влево (младший бит должен быть равен 0, что означает передачу).

Алгоритм передачи пакета следующий:

1. Передача стартового бита
2. Передача адреса I2C_addr<<1, младший бит байта адреса = 0
3. Проверка подтверждения. Если подтверждения нет – Стоп
4. Передача байтов данных с проверкой подтверждения после каждого байта.
5. Передача стопового бита.

5.2.2 Прием пакета из модуля.

Для инициации приема необходимо передать следующий пакет:

Номер байта:	0	1	2

Название, или значение:	I2C_addr <<1	1	CMD

CMD – код команды, I2C_addr – адрес модуля, сдвинутый на 1 бит влево (младший бит должен быть равен 0, что означает передачу). Например, для инициации приема данных байт 2 должен быть равен GET_DATA_PACKET.

После передачи пакета стоповый бит не выдается. Затем внешний процессор выдает повторный старт, и передает (I2C_addr<<1)+1 , т.е. адрес сдвинутый на один бит влево с установленным младшим битом(что означает прием). По приему этого байта процессор модуля притягивает линию SCL к нулю до момента окончания цикла измерения. Как только цикл измерения закончился, и данные готовы для передачи, линия SCL освобождается, и ведущий процессор может принять данные. Такой алгоритм работы позволяет синхронизировать работу внешнего процессора по отсчетам модуля, без использования дополнительных таймеров.

Далее производится прием следующего пакета:

Номер байта:	0	1	2	…	N
Название, или значение:	N	DATA1	DATA2	…	DATAN

Где N – число передаваемых байт данных, DATA1… DATAN – байты данных

После каждого принятого байта выдается подтверждение, за исключение последнего. После приема последнего байта выдается НеПодтверждение (т.е. просто сразу выдать стоповый бит нельзя, необходимо пропустить один такт SCL).

Таким образом, при приеме внешний процессор осуществляет следующие действия:

1. Выдает стартовый бит.
2. Передает I2C_addr<<1 (с нулевым младшим битом).
3. Проверяет подтверждение.
4. Передает 1.
5. Проверяет подтверждение.
6. Передает команду.
7. Проверяет подтверждение.
8. Выдает повторный старт.
9. Передает (I2C_addr<<1) +1 (с установленным младшим битом)
10. Проверяет подтверждение.
11. Ждет готовности данных.
12. Принимает число байт данных.
13. Выдает подтверждение.
14. Принимает все байты данных, кроме последнего. После каждого выдает подтверждение.
15. Принимает последний байт, выдает НеПодтверждение.
16. Выдает стоповый бит.

5.2.3. Алгоритм исполнения команд.

В слове состояния процессора модуля находятся три бита, имеющие отношение к модулю I2C:

1. I2C_BUSY – Устанавливается в начале исполнения команды, сбрасывается по завершению исполнения. Проверкой этого флага можно определить момент завершения команды.
2. I2C_CHK_Err – ошибка контрольной суммы последнего пакета.
3. I2C_LastCmd_Err – ошибка исполнения последней команды.

При исполнении тех команд, которые требуют длительного времени (например калибровки), можно циклически считывать слово состояния и проверять I2C_BUSY.

Прием слова состояния производится командой GET_STATUS_I2C (код 0x17). По этой команде процессор в ответ передаст байты, описанные в п. 5.1.4.1

Остальные два бита можно проверять по мере необходимости (см. примеры).

Команды по I2C в точности повторяют команды UART (п.5.1.3)