

Руководство по эксплуатации магнитного датчика углового положения DPT.
Редакция: v0.6



Двухосевой магнитный датчик углового положения DPT



Двухосевой магнитный датчик углового положения DPT специально предназначен для интегрированных шарниров роботов. Его принцип работы основан на технологии определения характеристик магнитного поля KingKong. Даже при наличии пространственных ограничений датчик точно определяет взаимное положение двух осей, демонстрируя при этом разрешение и точность, аналогичные оптическим системам. Он также отличается высокой помехоустойчивостью.

Датчик сочетает в себе две уникальные технологии: высокоточную технологию измерения величины магнитного поля и технологию экранирования помех. Сканирование пространственного распределения магнитного поля намагниченного кольца, находящегося на подвижной части, осуществляется при помощи высокоточных датчиков Холла внутри устройства. При этом точность калибровки достигается за счет использования технологии KingKong. Каждое из устройств поставляется с уникальным набором данных калибровки магнитного поля, что обеспечивает оптимальную точность измерений в ходе дальнейшей эксплуатации.

Конструкция допускает сравнительно широкие монтажные допуски между ротором и статором, упрощая тем самым процедуру монтажа для пользователя, но сохраняя при этом точность бесконтактных измерений.

Раздельная магнитно-электрическая конструкция обеспечивает высокую устойчивость к воздействию вибрации, пыли и масла. Датчик позволяет выполнять измерения при сверхвысоких скоростях, которые абсолютно не влияют на его точность и срок службы.

Интеграция в систему такой сверхтонкой и миниатюрной полый конструкции не составит труда независимо от варианта применения.

- Два 24-битных абсолютных выхода
- Абсолютная точность позиционирования по двум осям: $\pm 0,01^\circ$
- Миниатюрное исполнение (зазор в радиальном направлении с одной стороны равен 7 мм)
- Имеются модели с шагом 5 мм по внутреннему диаметру
- Технология не предусматривает жестких монтажных допусков при установке устройства на движущейся и неподвижной частях
- Технологии экранирования магнитных помех
- Полая конструкция обеспечивает гибкость при выборе места установки. Максимальная скорость 8 000 об/мин
- Индивидуальная калибровка
- Различные выходные интерфейсы
 - Устойчивость к различным внешним помехам



Обозначение модели

DPT-10-15-25-A-24-24-R-N-A-M

Внутренний диаметр внутреннего ротора
См. [Размеры](#)
XX – Модель с резьбой
XXB – Клеевое исполнение

Внутренний диаметр внешнего ротора
См. [Размеры](#)
XX – Модель с резьбой
XXB – Клеевое исполнение

Внешний диаметр статора
См. [Размеры](#)
xx – Исполнение для прямой установки
xxH – Исполнение для прямой и обратной установки

Тип выходного сигнала
A – Абсолютное значение

Разрешение внутреннего датчика
24 – 24 бит

Разрешение внешнего датчика
24 – 24 бит

Скорость передачи данных
M – По умолчанию 2.5 Мбит/с
V – 5 Мбит/с

Рабочая температура
A – -40 - 85 °C
B – -40 - 105 °C
C – -40 - 125 °C

Напряжение на входе
N – 5B

Выходной интерфейс
R – [RS485](#)
B – [BISS-C](#)

Размер

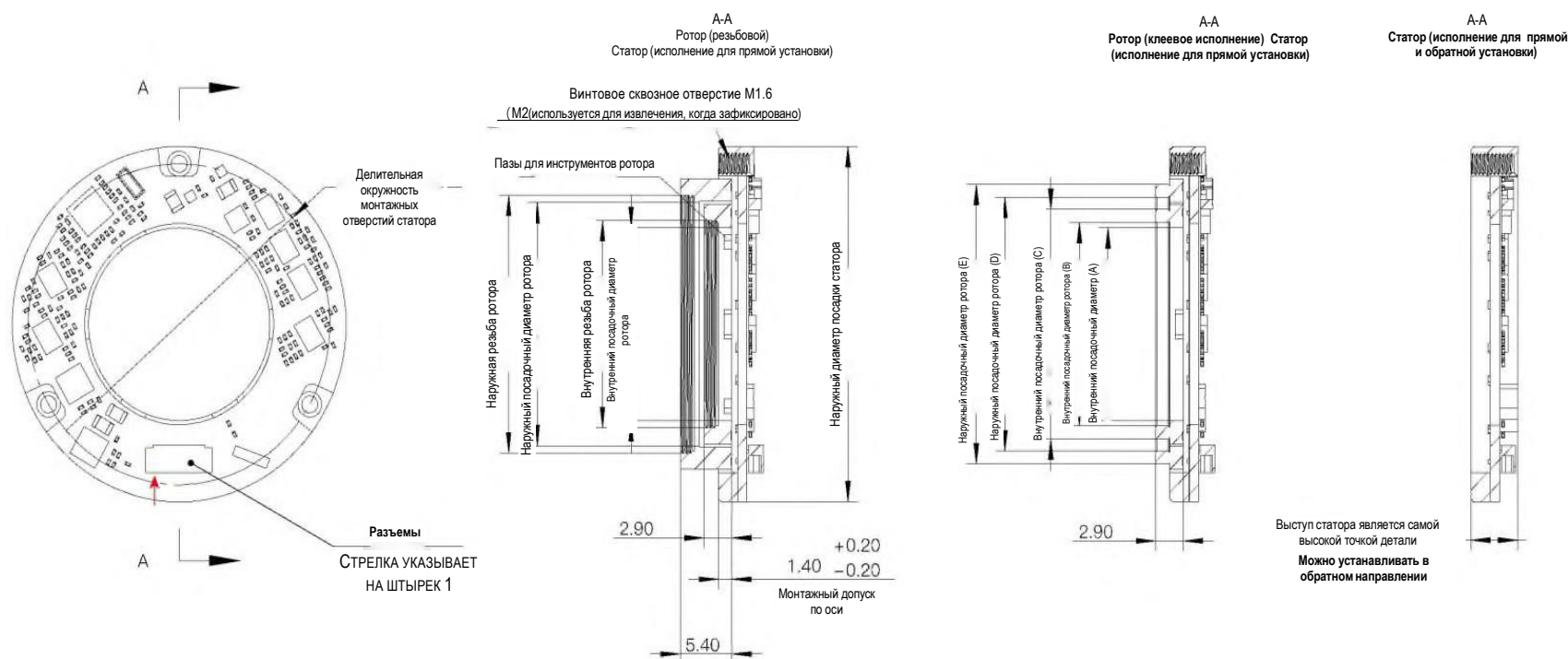
Серия	Размер	Типы роторов	Типы статора	Резьба внутреннего ротора	Высота внутреннего ротора	Внутренний ротор, посадочный диаметр (H7, h7)			Резьба внешнего ротора	Высота внешнего ротора	Внешний ротор, посадочный диаметр (H7)		Делительная окружность монтажных отверстий статора	Наружный посадочный диаметр статора (h7)	Статор можно установить в обратном направлении ⁽²⁾	Разъем, модель
						A	B	C			D	E				
10	DPT-10-15-25	Резьбовое исполнение	Исполнение для прямой установки	M10x0,4 мм	2.9	9			M15x0.4 мм	5.4	14		21.6	25	-	X0800WRS-06HF-LPV01
	DPT-10B-15B-25	Клеевое исполнение	Исполнение для прямой установки	-		9	10	12.6	-	2.9	15	17.6			-	
	DPT-10-15-25H	Резьбовое исполнение	Исполнение для обратной установки	M10x0,4 мм		9			M15x0.4 мм	5.4	14				√	
	DPT-10B-15B-25H	Клеевое исполнение	Исполнение для обратной установки	-		9	10	12.6	-	2.9	15	17.6			√	
15	DPT-15-20-30	Резьбовое исполнение	Исполнение для прямой установки	M15x0.4 мм		14			M20x0,4 мм	5.4	19		26.6	30	-	
	DPT-15B-20B-30	Клеевое исполнение	Исполнение для прямой установки	-		14	15	17.6	-	2.9	20	22.6			-	
	DPT-15-20-30H	Резьбовое исполнение	Исполнение для обратной установки	M15x0.4 мм		14			M20x0,4 мм	5.4	19				√	
	DPT-15B-20B-30H	Клеевое исполнение	Исполнение для обратной установки	-		14	15	17.6	-	2.9	20	22.6			√	
20	DPT-20-25-35	Резьбовое исполнение	Исполнение для прямой установки	M20x0,4 мм		19			M25x0,4 мм	5.4	24		31.6	35	-	
	DPT-20B-25B-35	Клеевое исполнение	Исполнение для прямой установки	-		19	20	22.6	-	2.9	25	27.6			-	
	DPT-20-25-35H	Резьбовое исполнение	Исполнение для обратной установки	M20x0,4 мм		19			M25x0,4 мм	5.4	24				√	
	DPT-20B-25B-35H	Клеевое исполнение	Исполнение для обратной установки	-		19	20	22.6	-	2.9	25	27.6			√	
25	DPT-25-30-40	Резьбовое исполнение	Исполнение для прямой установки	M25x0,4 мм		24			M30x0,4 мм	5.4	29		36.6	40	-	
	DPT-25B-30B-40	Клеевое исполнение	Исполнение для прямой установки	-		24	25	27.6	-	2.9	30	32.6			-	
	DPT-25-30-40H	Резьбовое исполнение	Исполнение для обратной установки	M25x0,4 мм		24			M30x0,4 мм	5.4	29				√	
	DPT-25B-30B-40H	Клеевое исполнение	Исполнение для обратной установки	-		24	25	27.6	-	2.9	30	32.6			√	
30	DPT-30-35-45	Резьбовое исполнение	Исполнение для обратной установки	M30x0,4 мм		29			M35x0,4 мм	5.4	34		41.6	45	-	
	DPT-30B-35B-45	Клеевое исполнение	Исполнение для прямой установки	-		29	30	32.6	-	2.9	35	37.6			-	
	DPT-30-35-45H	Резьбовое исполнение	Исполнение для обратной установки	M30x0,4 мм		29			M35x0,4 мм	5.4	34				√	
	DPT-30B-35B-45H	Клеевое исполнение	Исполнение для обратной установки	-		29	30	32.6	-	2.9	35	37.6			√	

1. Скачать 3D-модель: <https://kingkong.tech/encoder/DPT>

2. Если требуется обратная установка модели, предназначенной для прямой установки, необходимо обеспечить ее конструктивное совмещение с выступом на статоре.

3. Можно выбрать любой из вариантов диаметра для клеевого исполнения.

Чертеж



На рисунке выше «Красная стрелка» обозначает контакт 1, см. стр 5.

При монтаже статора можно воспользоваться винтом с резьбой M1.6 для прохождения через резьбовое отверстие M2, или же использовать резьбу M2 с винтом M2 для фиксации. При монтаже ротора выполните затяжку при помощи специального инструмента. Для получения более подробных сведений см. «Рекомендации по проектированию и монтажу датчика DPT». Для получения соответствующего программного обеспечения для калибровки, методики калибровки и чертежей обратитесь в компанию KingKong.

Подключение питания

Разъемы

	разъем типа «провод-плата»
Модель	X0800WRS-06HF-LPV01
Тип	Провод-плата
Соединительные кабели	3РхAWG32, витая пара с тефлоновым покрытием

Контакты

Контакты	Цвет	R	B
		RS485	BISS-C
1	Оранжевый	-	SLO —
2	Желтый	-	SLO +
3	Зеленый	B	MA -
4	Синий	A	MA +
5	Черный	0V (GND)	
6	Красный	+5B	



Технические характеристики

Параметры системы

Способы монтажа	осевая полость
Разрешение	24 бит
Точность	$\pm 0,01^\circ$ (зависит от сборки)

Питание

Источник питания	4,5 - 5,5 В
Время запуска	15 мс
Подключение	разъем типа «провод-плата»
Ток	Примерно 140 мА
Защита от электростатического разряда	Модель человеческого тела, макс. ± 2 кВ Модель заряженного устройства, макс. ± 1 кВ

Механические характеристики

Держатель ротора	нержавеющая сталь
Держатель статора	алюминиевый сплав

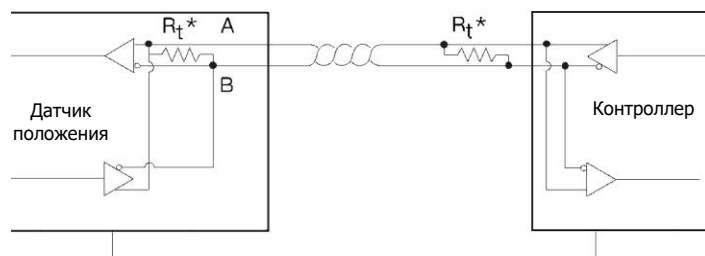
Параметры окружающей среды

Рабочая температура	-40 - 85 °C / -40 - 105 °C / -40 - 125 °C
---------------------	---

Конфигурация протокола:

Интерфейс протокола RS485

Принципиальная схема подключения интерфейсного модуля RS485



Интерфейс представляет собой двухпроводную систему на основе дифференциальных сигналов А и В. Концы двух проводов необходимо подключить к клеммам оконечных резисторов. Оконечный резистор со стороны датчика уже встроен в его корпус, поэтому пользователю необходимо подключить оконечный резистор или применить другое решение по согласованию импеданса со стороны контроллера.

Базовый протокол интерфейсного модуля RS485 является протокол UART. Поскольку в данном протоколе не предусмотрена линия передачи синхронизирующего сигнала, для организации процесса передачи данных датчик углового положения и контроллер должны работать на согласованной одинаковой частоте и использовать единый формат данных.

Длина символьной строки	Контроль четности	Стоповый бит	Управление потоком данных	Порядок следования в байте
8 бит	Нет	1	Нет	Младший (значащий) бит первым

Поддерживаемые скорости передачи данных:

Код	I	B
Скорость передачи данных (Мбит/с)	2.5	5

Диаграмма синхронизации:



(1) Датчик фиксирует текущие данные об угле в точке прохождения красного сигнала.



Команды и данные:

Команда	Ввод		N	Возвращаемые данные (N байт)							
				B0	B1	B2	B3	B4	B5	B6	B7
0x29	Настройка	Внутреннее нулевое положение ⁽¹⁾	2	C	CRC						
0x30	Настройка	Внешнее нулевое положение ⁽²⁾	2	C	CRC						
0x31	Получение	Внутренний угол	4	A0	A1	A2	CRC				
0x32	Получение	Внешний угол	4	B0	B1	B2	CRC				
0x33	Получение	Внутренний угол Внешний угол	7	A0	A1	A2	B0	B1	B2	CRC	
0x41	Получение	Внутренний угол и информация о состоянии	5	A0	A1	A2	C	CRC			
0x42	Получение	Внешний угол и информация о состоянии	5	B0	B1	B2	C	CRC			
0x43	Получение	Внутренний угол, внешний угол, информация о состоянии	8	A0	A1	A2	B0	B1	B2	C	CRC

В приведенной выше таблице буквы соответствуют следующим данным:

A	B	C	C	CRC
Внутренний угол датчика	Внешний угол датчика	Сброс отсчета от нуля	Состояние	Проверка CRC

(1) Для установки нулевого положения необходимо последовательно с интервалом отправить команду 0x31, а затем команду 0x29 в общей сложности 10 раз, чтобы установка прошла успешно; когда возвращаемое значение счетчика равняется 10, запускается установка нулевого положения. (2) Для обнуления необходимо последовательно с интервалом отправить команды 0x32 и 0x30 в общей сложности 10 раз; обнуление срабатывает, когда возвращаемое значение счетчика равно 10. (3) Чем больше число, тем выше порядок байтов.

(1) Байт CRC (многочлен CRC равен $x^8+x^7+x^4+x^2+x^1+1$; см. соответствующий метод расчета в приложении с алгоритмом расчета CRC-8 ([CRC-8 表\(\$x^8+x^7+x^4+x^2+x^1+1\$ \)](#))).

Пример:

Например, при отправке 0x33 получается uint8_t Buffer[7];

При использовании:

```
uint32_t angleInner = Buffer[2] << 16 | buffer[1] << 8 | buffer[0];
```

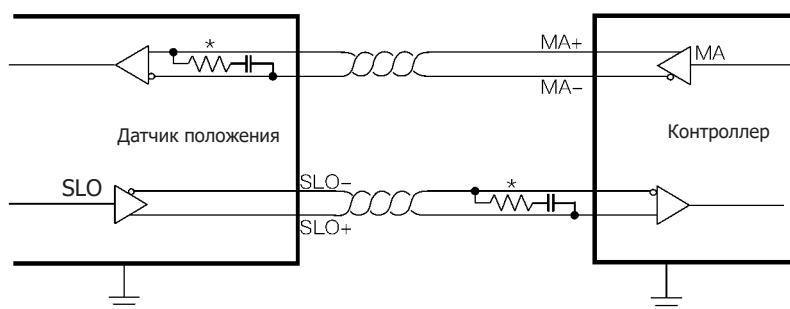
```
uint32_t anngleOutter = Buffer[5] << 16 | buffer[4] << 8 | buffer[3];
```

```
float angleInnerFloat = angleInner / (float)(1 << 24) * 360;
```

```
float angleOutterFloat = angleOutter / (float)(1 << 24) * 360;
```

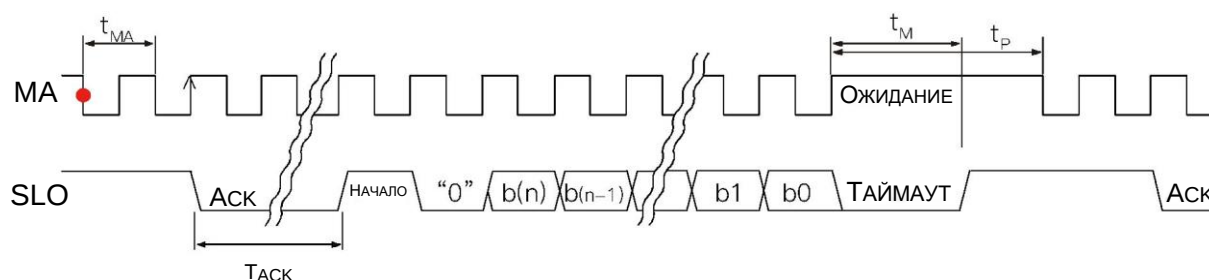
Интерфейс протокола BISS-C

Принципиальная схема подключения интерфейсного модуля BISS-C:



В интерфейсе реализована четырехпроводная система, состоящая из ведущего сигнала МА, и ведомого сигнала SLO. Кроме того, для ведущего сигнала (МА) в датчик уже интегрирован оконечный резистор. Пользователю необходимо выполнить подключение оконечного резистора или же реализовать другое решение для согласования импеданса на стороне ведомого сигнала (SLO) в контроллере.

Диаграмма синхронизации:



(1) Датчик фиксирует текущие данные об угле в точке прохождения инфракрасного сигнала

Для синхронизации и получения последовательности тактовых импульсов в протоколе используется ведущий сигнал (МА). Пока линия МА находится в режиме ожидания, она соответствует высокому уровню. При поступлении первого спадающего фронта система фиксирует текущие данные. При этом обмен информацией начинается с первого спадающего фронта. Датчик положения устанавливает ведомый сигнал (SLO) на низкий уровень на втором нарастающем фронте ведущего сигнала (МА). Начиная с 0, старший значащий бит (MSB) записывается на линию ведомого сигнала (SLO) на каждом нарастающем фронте ведущего сигнала (МА). На стороне контроллера данные на линии ведомого сигнала (SLO) считываются на спадающем фронте ведущего сигнала (МА), и этот процесс повторяется до тех пор, пока младший значащий бит (LSB) не будет считан контроллером.

Параметры синхронизации:

Параметры	Обозначение	Минимальное значение	Стандартное значение	Максимальное значение
Частота тактового сигнала	t _{MA}	400 нс		14 мкс
Частота тактового сигнала	f	120 кГц		2,5 МГц ⁽¹⁾
Длина ACK	t _{ACK}		5 бит	



Параметры	Обозначение	Минимальное значение	Стандартное значение	Максимальное значение
Время ожидания передачи	tM		10 мкс	
Длительность паузы	tP	20 мкс		

После завершения передачи данных и окончания интервала передачи данных tM, ведомый сигнал (SLO) переходит на высокий уровень. Ведущий сигнал (MA) должен оставаться на высоком уровне до тех пор, пока не будет разрешено следующее считывание, то есть до окончания интервала tP. Интервал tCL должен быть меньше интервала tM; при этом во время любой операции чтения данный интервал может превысить tM, чтобы прервать чтение.

Формат данных:

Биты	b55 : b32	b31 : b8	b7	b6	b5 : b0
Длина	24 бита	24 бита	1 бит	1 бит	6 бит
Данные	Угол внутреннего оборота	Угол наружного оборота	Бит ошибки ⁽¹⁾	Бит предупреждения ⁽²⁾	CRC ⁽³⁾

(1) Биты ошибки активны при низком уровне сигнала.

(2) Биты предупреждения активны при низком уровне. Когда он равен 1, это указывает на отсутствие ошибок или предупреждений в системе. Когда он равен 0, это указывает на то, что система выдала, по меньшей мере, одну ошибку или предупреждение.

(3) Многочлен контроля с использованием циклического избыточного кода CRC равен $x^6 + x^1 + 1$ (то есть 0x43). В соответствии с требованиями протокола BISS-C, вычисленный многочлен инвертируется перед отправкой. Для удобства в приложении приводится код для расчета CRC-6, который можно просто перенести и использовать.

Для получения сведений о битах состояния см. раздел [«Биты состояния»](#) далее по тексту.



БИТЫ СОСТОЯНИЯ

Информация, содержащаяся в битах ошибок, битах предупреждений и битах состояния, используется для характеристики текущего рабочего состояния датчика, при этом функции битов ошибок и битов предупреждений одинаковы для всех интерфейсов.

Биты ошибок, биты предупреждений, биты состояния:

Позиция	b7	b6	b5	b4	b3	b2	b1	b0
Функции	Указание на ошибку		Биты состояния					
Описание	Бит ошибки	-	Ошибка угла внешнего ротора	Ошибка угла внутреннего ротора	-	-	-	-

(1) В протоколе RS485 установка бита в «1» означает ошибку.

(2) В протоколе BISS-C установка бита в «0» означает ошибку, при этом конкретная информация о битах состояния отсутствует.

В зависимости от информации, отображаемой светодиодами ошибок и предупреждений, их цвет меняется. Если данные некорректны, то в это время светодиодный индикатор состояния горит красным, а конкретную ситуацию можно определить при помощи битов состояния. При нормальном режиме эксплуатации светодиодный индикатор горит зеленым.

Приложение

Пример расчета многочлена CRC-8($x^8+x^7+x^4+x^2+x^1+1$)

```
//poly =  $x^8+x^7+x^4+x^2+x^1+1$ 
```

```
uint8_t crcTable [256] = {  
0x00, 0x97, 0xB9, 0x2E, 0xE5, 0x72, 0x5C, 0xCB, 0x5D, 0xCA, 0xE4, 0x73, 0xB8, 0x2F, 0x01, 0x96, 0xBA, 0x2D, 0x03,  
0x94, 0x5F, 0xC8, 0xE6, 0x71, 0xE7, 0x70, 0x5E, 0xC9, 0x02, 0x95, 0xBB, 0x2C, 0xE3, 0x74, 0x5A, 0xCD, 0x06, 0x91,  
0xBF,  
0x28, 0xBE, 0x29, 0x07, 0x90, 0x5B, 0xCC, 0xE2, 0x75, 0x59, 0xCE, 0xE0, 0x77, 0xBC, 0x2B, 0x05, 0x92, 0x04, 0x93,  
0xBD,  
0x2A, 0xE1, 0x76, 0x58, 0xCF, 0x51, 0xC6, 0xE8, 0x7F, 0xB4, 0x23, 0x0D, 0x9A, 0x0C, 0x9B, 0xB5, 0x22, 0xE9, 0x7E,  
0x50,  
0xC7, 0xEB, 0x7C, 0x52, 0xC5, 0x0E, 0x99, 0xB7, 0x20, 0xB6, 0x21, 0x0F, 0x98, 0x53, 0xC4, 0xEA, 0x7D, 0xB2, 0x25,  
0x0B,  
0x9C, 0x57, 0xC0, 0xEE, 0x79, 0xEF, 0x78, 0x56, 0xC1, 0x0A, 0x9D, 0xB3, 0x24, 0x08, 0x9F, 0xB1, 0x26, 0xED, 0x7A,  
0x54,  
0xC3, 0x55, 0xC2, 0xEC, 0x7B, 0xB0, 0x27, 0x09, 0x9E, 0xA2, 0x35, 0x1B, 0x8C, 0x47, 0xD0, 0xFE, 0x69, 0xFF, 0x68,  
0x46,  
0xD1, 0x1A, 0x8D, 0xA3, 0x34, 0x18, 0x8F, 0xA1, 0x36, 0xFD, 0x6A, 0x44, 0xD3, 0x45, 0xD2, 0xFC, 0x6B, 0xA0, 0x37,  
0x19,  
0x8E, 0x41, 0xD6, 0xF8, 0x6F, 0xA4, 0x33, 0x1D, 0x8A, 0x1C, 0x8B, 0xA5, 0x32, 0xF9, 0x6E, 0x40, 0xD7, 0xFB, 0x6C,  
0x42, 0xD5, 0x1E, 0x89, 0xA7, 0x30, 0xA6, 0x31, 0x1F, 0x88, 0x43, 0xD4, 0xFA, 0x6D, 0xF3, 0x64, 0x4A, 0xDD, 0x16,  
0x81, 0xAF, 0x38,  
0xAE, 0x39, 0x17, 0x80, 0x4B, 0xDC, 0xF2, 0x65, 0x49, 0xDE, 0xF0, 0x67, 0xAC, 0x3B, 0x15, 0x82, 0x14, 0x83, 0xAD,  
0x3A,  
0xF1, 0x66, 0x48, 0xDF, 0x10, 0x87, 0xA9, 0x3E, 0xF5, 0x62, 0x4C, 0xDB, 0x4D, 0xDA, 0xF4, 0x63, 0xA8, 0x3F, 0x11,  
0x86, 0xAA, 0x3D, 0x13, 0x84, 0x4F, 0xD8, 0xF6, 0x61, 0xF7, 0x60, 0x4E, 0xD9, 0x12, 0x85, 0xAB, 0x3C };  
uint8_t calcCRC(uint8_t * buffer, uint8_t length) {  
uint8_t temp = *buffer++;  
while(--length){  
temp = *buffer++ ^ crcTable[temp];  
}  
return crcTable[temp];  
}
```

Вычисление CRC-6

```
#define DATA_TOTAL_BIT_LENGTH 47
//poly = x6 + x1 + 1
uint8_t tableCRC6[64] = {
0x00, 0x03, 0x06, 0x05, 0x0C, 0x0F, 0x0A, 0x09, 0x18, 0x1B, 0x1E, 0x1D, 0x14, 0x17, 0x12, 0x11, 0x30, 0x33, 0x36, 0x35, 0x3C, 0x3F, 0x3A, 0x39, 0x28, 0x2B, 0x2E, 0x2D,
0x24, 0x27, 0x22, 0x21, 0x23, 0x20, 0x25, 0x26, 0x2F, 0x2C, 0x29, 0x2A, 0x3B, 0x38, 0x3D, 0x3E, 0x37, 0x34, 0x31, 0x32, 0x13, 0x10, 0x15, 0x16, 0x1F, 0x1C, 0x19, 0x1A,
0x0B, 0x08, 0x0D, 0x0E, 0x07, 0x04, 0x01, 0x02
};
uint8_t calcBissCCRC(uint8_t buffer[]) {
#define CRC_BIT_LENGTH 6
#define DATA_CRC_MASK ((1 << CRC_BIT_LENGTH) - 1)
#define DATA_WITHOUT_CRC_BIT_LENGTH (DATA_TOTAL_BIT_LENGTH - CRC_BIT_LENGTH)
#define TOP_BYTE_BITLENGTH (DATA_WITHOUT_CRC_BIT_LENGTH % CRC_BIT_LENGTH)
#if TOP_BYTE_BITLENGTH == 0
#undef TOP_BYTE_BITLENGTH
#define TOP_BYTE_BITLENGTH CRC_BIT_LENGTH
#endif
uint32_t firstWord = __REV(*(uint32_t *) buffer);
#if DATA_WITHOUT_CRC_BIT_LENGTH > 32
uint32_t secondWord = __REV(*(uint32_t *) (buffer + 4));
#endif
uint8_t crc = tableCRC6[firstWord >> (32 - TOP_BYTE_BITLENGTH)];
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 1)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (firstWord >> (32 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 2)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (firstWord >> (32 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 3)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (firstWord >> (32 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 4)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (firstWord >> (32 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 5)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
#иf 32 - CURRENT_CRC_BIT_LENGTH >= 0
crc = tableCRC6[crc ^ (firstWord >> (32 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#elif 32 - CURRENT_CRC_BIT_LENGTH > -CRC_BIT_LENGTH
crc = tableCRC6[crc ^ (((firstWord << -(32 - CURRENT_CRC_BIT_LENGTH)) & DATA_CRC_MASK) | (secondWord >> (64 - CURRENT_CRC_BIT_LENGTH))));
#else
crc = tableCRC6[crc ^ (secondWord >> (64 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 6)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (secondWord >> (64 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 7)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (secondWord >> (64 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 8)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (secondWord >> (64 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#undef CURRENT_CRC_BIT_LENGTH
#define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 9)
#если DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH ≥ 0
crc = tableCRC6[crc ^ (secondWord >> (64 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
#иf 32 - DATA_TOTAL_BIT_LENGTH >= 0
crc = tableCRC6[crc ^ DATA_CRC_MASK ^ (firstWord >> (32 - DATA_TOTAL_BIT_LENGTH) & DATA_CRC_MASK)];
#elif 32 - CURRENT_CRC_BIT_LENGTH > -CRC_BIT_LENGTH
crc = tableCRC6[crc ^ DATA_CRC_MASK ^ (((firstWord << -(32 - DATA_TOTAL_BIT_LENGTH)) & DATA_CRC_MASK) | (secondWord >> (64 - DATA_TOTAL_BIT_LENGTH))));
#else
crc = tableCRC6[crc ^ DATA_CRC_MASK ^ (secondWord >> (64 - DATA_TOTAL_BIT_LENGTH) & DATA_CRC_MASK)];
#endif
return crc;
}
```

Указания по использованию программы:

Программу можно запускать на микроконтроллерах серии ARM для реализации максимально быстрого контроля с использованием циклического избыточного кода CRC-6 через компилятор.

Единственное, что необходимо сделать, — это изменить DATA_TOTAL_BIT_LENGTH на значение, соответствующее вашей модели.

Примечания по использованию программы:

При вызове используется 32-битная команда чтения для буфера, требующая, чтобы буфер был выровнен по 4 байта (при отсутствии такого выравнивания программа не будет работать при некоторых версиях ядра; даже в случае поддержки невыровненного чтения ядру потребуется дополнительное время для выполнения операции конкатенации).

В случае получения данных по протоколу BISS-C первый полученный байт будет служебным байтом с ACK, а данные о положении начинаются со второго байта. Поэтому нам нужно вычислить CRC, начиная с адреса, где начинаются данные о положении, а для быстрого чтения данных и вычисления CRC такой адрес должен иметь выравнивание по 4 байта.

Пример вызова:

```
struct{
uint8_t notUsedForAlignment[3]; //Только для выравнивания адреса
uint8_t placeholder; //BISS-Спервый фиксированный служебный байт 0x82
uint8_t buffer[8] __attribute__((aligned(4))); //4 Буфер выровнен по границам 4 байт для быстрого
расчета CRC
} receiveBuffer;
//Конфигурирование SPI (последовательного периферийного интерфейса) и DMA (прямого доступа к
памяти)
//Используйте &receiveBuffer.placeholder в качестве адреса получения
//.....
//Расчет CRC
//Для расчета используйте уже выровненный по 4 байта receiveBuffer.buffer
uint8_t crc = calcBissCCRC(receiveBuffer.buffer);
//Если результат CRC равен 0, это означает, что контроль пройден
if (crc != 0){
//Контроль CRC не пройден
}
```

Хронология редакций документа

Время Версия	Внесенные изменения
1 сентября 2023 г. Ред. 0.1	Начальная версия
18 марта 2024 г. Ред. 0.2	Обновление бита состояния Дополнено соглашение BISS-C
1 мая 2024 г. Ред. 0.3	Добавлен стандартный чертеж. Добавлен цвет жгута проводов.
8 декабря 2024 Ред. 0.4	Добавлена модель с клеевым исполнением ротора Добавлена модель со статором, установленным в обратном направлении Изменено обозначение разъема на левый 1 - штырек 1.
22 декабря 2024 г. Ред. 0.5	Добавлен индикатор положения углового фиксатора
31 марта 2026 г. Ред. 0.6	Добавлено описание частоты вращения. Добавлены типы скорости передачи данных. Доработаны описания всех протоколов.



KingKong.tech
金 钢 科 技



ИННОДРАЙВ
ТЕХНОЛОГИИ ИМЕЮТ ЗНАЧЕНИЕ

ООО «ИнноДрайв» - официальный дистрибьютор
KingKong Technology

+7 (812) 317-77-93
www.innodrive.ru
sales@innodrive.ru

