

Руководство по эксплуатации магнитного датчика углового положения МРТ

Редакция: 0.3



Магнитный датчик углового положения МРТ



Магнитный датчик углового положения МРТ — миниатюрный датчик, принцип работы которого 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.

Предложенная технология не накладывает жестких ограничений на монтаж устройства на движущейся или неподвижной частях, упрощая тем самым процедуру монтажа для пользователя, но сохраняя при этом точность бесконтактных измерений.

Такая комбинация с отдельной установкой датчика Холла и магнита представляет собой эффективное решение и обеспечивает стойкость к воздействию факторов окружающей среды, таких как вибрация, а также наличие пыли и масляных загрязнений. 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 выходных сигнала, 24 бита, при абсолютных измерениях
- Точность позиционирования при абсолютных измерениях $\pm 0.01^\circ$
- Миниатюрное исполнение (зазор в радиальном направлении с одной стороны равен 5 мм)
- Имеются модели с шагом 5 мм по внутреннему диаметру
- Технология не предусматривает жестких монтажных допусков при установке устройства на движущейся и неподвижной частях
- Технология экранирования магнитных помех
- Полое конструктивное исполнение позволяет не накладывать пространственных ограничений при монтаже
- Максимальная скорость 8000 об/мин
- Уникальный набор данных, получаемый в процессе калибровки
- Батарея позволяет сохранять данные о полных оборотах вала
- Различные выходные интерфейсы
- Устойчивость к воздействиям окружающей среды

Обозначение модели

MPT-20-30-A-24-S-N-A-M

Внутренний диаметр ротора
Конкретный **размер**

Наружный диаметр статора
Конкретный **размер**

Выходные данные
А – абсолютное значение

Параметры выходного сигнала
24 – 24 бита на один оборот

Опции
М – без дополнительных
опций
Скорость передачи данных –
интерфейс R или A

Рабочая температура
А – -40 ~ 85 °C
В – -40 ~ 105 °C
С – -40 ~ 125 °C

Напряжение на входе
N – 5В

Интерфейс выходного сигнала

S – SSI ⁽¹⁾ не рекомендовано

B – BISS-C рекомендовано

R – RS485 рекомендовано

A – RS422

T – T485 (Совместим с Tamagawa 17 бит) ⁽²⁾ рекомендовано

T – T485 (Совместим с Tamagawa 23 бит) ⁽²⁾ рекомендовано

D – BUS (высокоскоростная шина)

P – PERIOD (периодическая передача данных)

После отключения питания происходит сброс
данных о нескольких оборотах.
Режим отслеживания данных по нескольким
оборотам действует только при включенном
питании.

24M – 24 бит на один оборот + стандарт по
нескольким оборотам

Питание от батареи после отключения
питания.
Функция сохранения данных о нескольких
оборотам.

24BM – 24 бит на один оборот + плюс питание
от батареи в режиме отслеживания данных о
нескольких оборотах

После отключения питания допускается
вращение только на ±90°.
Батарея не требуется.

24FM – 24 бит на один оборот + плюс сохранение
на флэш-память в режиме отслеживания данных о
нескольких оборотах

(1) В протоколе SSI не предусмотрена верификация CRC: для повышения надежности рекомендуется пользоваться протоколом BISS с тем же оборудованием.

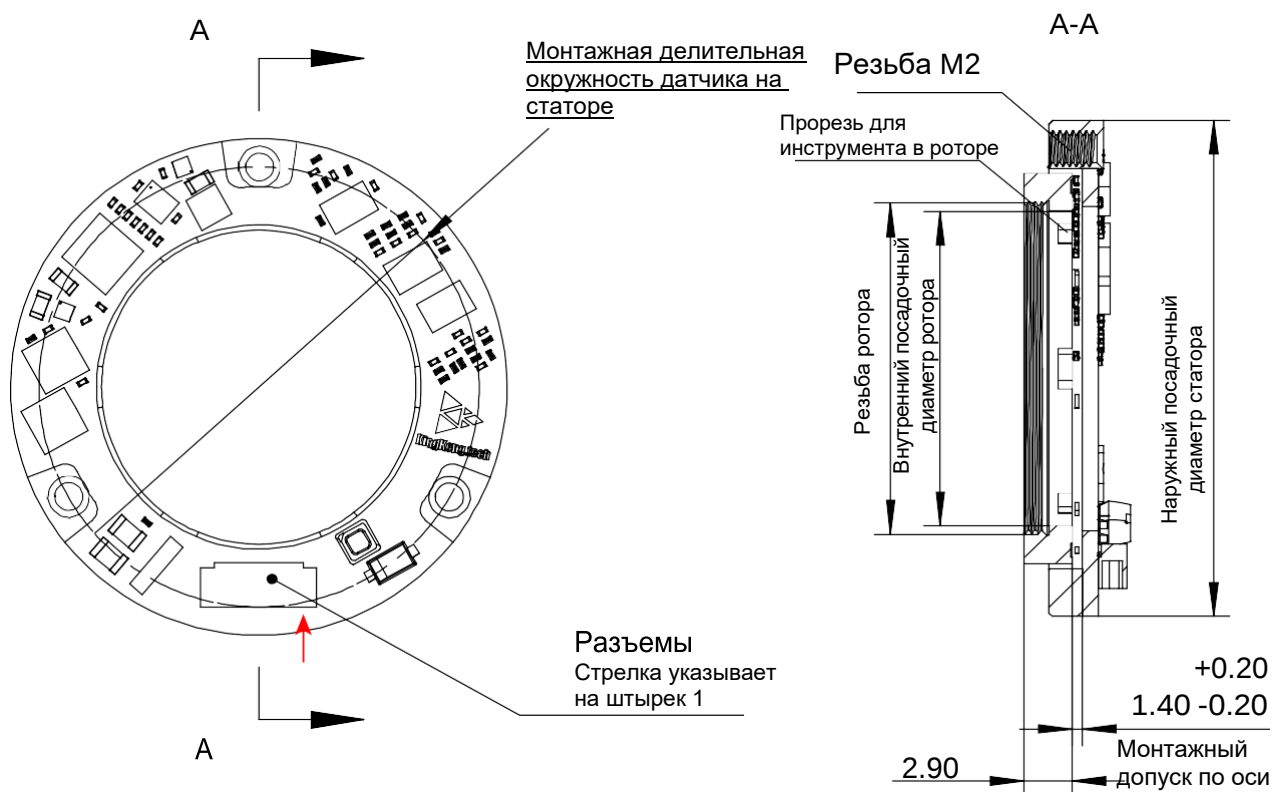
(2) При выборе протокола Tamagawa на 17 бит выходные параметры: 17M-T, 17BM-T, 17FM-T; при выборе аналогичного протокола на 23 бита: 23M-T, 23BM-T, 23FM-T.

Размеры

Серия	Размер	Внутренняя резьба ротора	Внутренний посадочный диаметр ротора (H7)	Наружный диаметр статора (h7)	Монтажная делительная окружность датчика на статоре	Тип разъема	Общая толщина
10	MPT-10-20	M10x0.4 мм	9	20	16.6	SM06B-XSRS-ETB	6.7
15	MPT-13-25	M13x0.4 мм	12	25	21.6	SM06B-XSRS-ETB	
	MPT-14-25	M14x0.4 мм	13			SM06B-XSRS-ETB	
	MPT-15-25	M15x0.4 мм	14			SM06B-XSRS-ETB	
20	MPT-20-30	M20x0.4 мм	19	30	26.6	SM08-SURS-TF	
25	MPT-25-35	M25x0.4 мм	24	35	31.6	SM08-SURS-TF	
30	MPT-30-40	M30x0.4 мм	29	40	36.6	SM08-SURS-TF	
35	MPT-35-45	M35x0.4 мм	34	45	41.6	SM08-SURS-TF	

1. Для скачивания 3D-модели перейдите по ссылке: <https://kingkong.tech/encoder/MPT>

Чертеж



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 MPT».



Подключение питания

Разъемы

	разъем типа «провод-плата»
Модель	См. размеры
Тип	провод-плата
Соединительные кабели	Тефлоновая витая пара 4PхAWG32

Контакты

Контакт	Цвет	S SSI	B BISS-C	A RS422	R RS485	T T485	D BUS	P PERIOD
1	Красный	+5В						
2	Черный	0В (ЗАЗЕМЛЕНИЕ)						
3	Синий	Clock +	MA +	RX +	A	A	A	-
4	Зеленый	Clock -	MA -	RX -	B	B	B	-
5	Желтый	Data +	SLO +	TX +	-	-	-	TX +
6	Оранжевый	Data -	SLO -	TX -	-	-	-	TX -
7	Белый	Батарея +	Батарея +	Батарея +	Батарея +	Батарея +	Батарея +	Батарея +
8	Серый	Батарея -	Батарея -	Батарея -	Батарея -	Батарея -	Батарея -	Батарея -

* Контакты батареи +/- присутствуют только у моделей с режимом сохранения данных о нескольких оборотах при питании от батареи (BM).

* Контакт батареи присоединяют к ЗАЗЕМЛЕНИЮ (GND) внутри датчика.

* Рекомендуется заземлить экран на стороне привода.



Характеристики

Параметры системы

Способ монтажа	осевая полость
----------------	----------------

Точность	$\pm 0.01^\circ$
----------	------------------

Питание

Питание	4.5 ~ 5.5 В
---------	-------------

Батарея	2.7 ~ 3.6 В
---------	-------------

Продолжительность пуска	15 мсек
-------------------------	---------

Подключение	разъем типа «провод-плата»
-------------	----------------------------

Ток	≈ 100 мА
-----	------------------

Ток при низком энергопотреблении	≈ 6 мкА (напряжение батареи 3,6 В, в режиме низкого энергопотребления в статическом режиме без движения)
----------------------------------	--

Защита от электростатического разряда	Модель человеческого тела, макс. ± 2 кВ Модель заряженного устройства, макс. ± 1 кВ
---------------------------------------	--

Механические характеристики

Материал корпуса ротора	нержавеющая сталь
-------------------------	-------------------

Материал корпуса статора	алюминиевый сплав
--------------------------	-------------------

Параметры окружающей среды

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

Пояснения к характеристикам

Максимальная частота вращения

Благодаря отсутствию контакта между ротором и статором (вращающейся и неподвижной частью) между ними отсутствует трение, что позволяет достичь высокой максимальной частоты вращения в тех условиях эксплуатации, где это необходимо.

Внешние помехи

Технология выполнения измерений с использованием магнитных датчиков углового положения гарантирует таким датчикам определенную стойкость к воздействию вибрации и позволяет не допустить поломки самого датчика. Кроме того, датчики обладают высокой стойкостью к воздействиям немагнитного характера, например, к воздействию масла и пыли, присутствие которых никак не сказывается на точности измерений.

Внешние магнитные помехи

В датчиках углового положения МРТ реализована уникальная технология защиты от электромагнитных помех. Технология экранирования помех позволяет блокировать электромагнитные помехи, создаваемые окружающей средой (например, помехи, создаваемые двигателями), а сила магнитного поля намагниченного кольца обычно значительно превышает внешние магнитные помехи. В совокупности указанные функции позволяют надежно противостоять внешним помехам.

Точность

В процессе производства в ходе намагничивания магнитного кольца могут иметь место проблемы рассогласования в работе системы. Однако каждый датчик на основе технологии KingKong проходит заводскую калибровку, в ходе которой инженеры получают уникальный набор данных для каждого намагниченного кольца.



Набор абсолютных значений

Формат выходного сигнала

Один оборот

На выходе получаем данные
об угловом положении

Несколько оборотов

На выходе получаем данные Число оборотов не сохраняется после отключения питания; при повторном
о числе оборотов и угловом включении счетчик оборотов сбрасывается на ноль.
положении

Несколько оборотов, когда возможен переход на питание от батареи

На выходе получаем данные После отключения питания датчик запитывается батареи и переходит в режим
о числе оборотов и угловом низкого энергопотребления, при этом его способность отслеживать вращение
положении двигателя и продолжать замеры сохраняется.
При повторном включении питания сброса имеющегося числа оборотов не
происходит.

Значения при абсолютных измерениях

**Число разрядов
на один оборот** 24 бит

**Максимальная
частота вращения** 8000 об./мин.

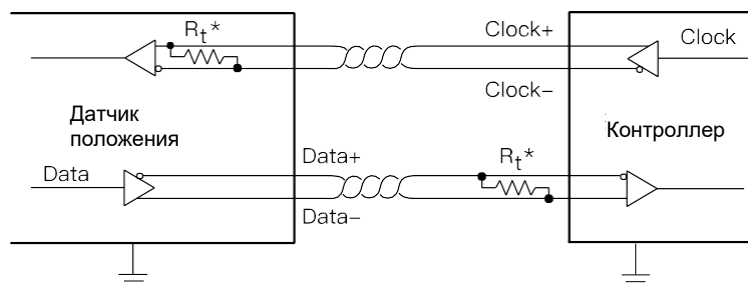
**Частота
актуализации
данных** 50 кГц

Повторяемость 0 ~ ±1 бит (показатель зависит от выбранной разрядности)

**Интерфейс
выходного
сигнала** SSI, BISS-C, RS485, RS422, T485, BUS, PERIOD

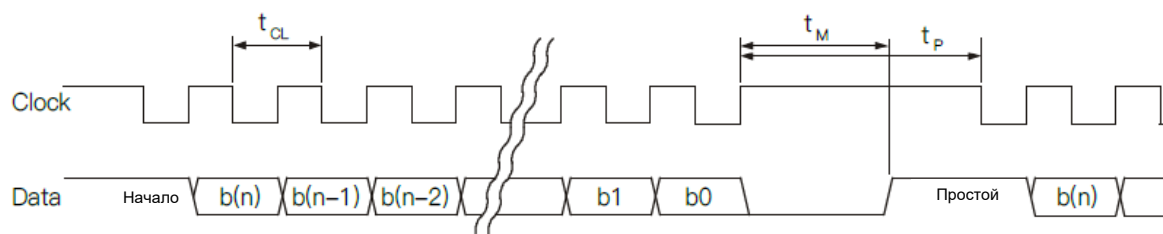
Интерфейс протокола SSI

Принципиальная схема подключения:



В интерфейсе реализована четырехпроводная система, состоящая из положительной и отрицательной фаз для тактового сигнала (Clock) и сигнала передачи данных (Data). Кроме того, для тактового сигнала (Clock) в датчик уже интегрирован оконечный резистор. Пользователю необходимо выполнить подключение оконечного резистора или же реализовать другое решение для согласования импеданса на стороне сигнала передачи данных (Data) в контроллере.

Диаграмма синхронизации:



Для синхронизации и получения последовательности тактов в протоколе используется тактовый сигнал Clock. При получении спадающего фронта сигнала система фиксирует текущие данные. Она начинает записывать данные (Data) при каждом нарастающем фронте тактового сигнала Clock, начиная со старшего значащего бита (MSB). На стороне контроллера данные сигнала передачи данных (Data) считываются на спадающем фронте тактового сигнала (Clock), и это продолжается до тех пор, пока младший значащий бит (LSB) не будет считан контроллером.

После завершения передачи данных и окончания интервала передачи данных t_M , сигнал передачи данных (Data) переходит на высокий уровень. Тактовый сигнал (Clock) должен оставаться на высоком уровне до тех пор, пока не будет разрешено следующее считывание, то есть до окончания интервала t_P . Интервал t_{CL} должен быть меньше интервала t_M ; при этом во время любой операции чтения данный интервал может превысить t_M , чтобы прервать чтение.



Параметры синхронизации:

Параметры	Обозначение	Минимальное значение	Стандартное значение	Максимальное значение
Частота тактового сигнала	tCL	400 нс		14 мкс
Частота тактового сигнала	tCL	110 кГц		1.5 МГц(1)
Время ожидания передачи	tM		10 мкс	
Длительность паузы	tP	20 мкс		

Если тактовый сигнал Clock продолжает оставаться на низком уровне в течение 500 нс (на первом низком уровне), то последующая частота тактового сигнала может достигать 10 МГц.

Формат данных:

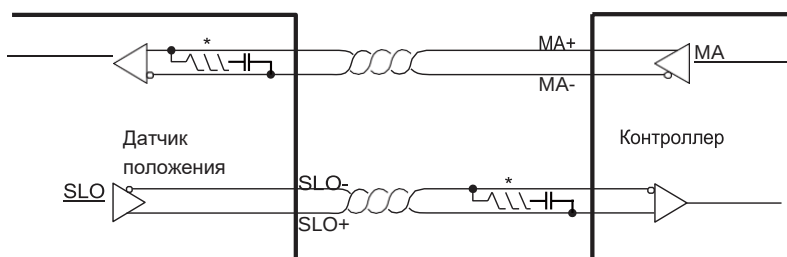
Биты	b(23 + X) : b(8 + X)	b(7 + X) : b8	b7	b6	b5 : b0
Длина	16 бит	X бит	1 бит	1 бит	6 бит
Данные	Счетчик нескольких оборотов ⁽¹⁾	Угол при одном обороте	Бит ошибки	Бит предупреждения	Бит состояния

(1) Счетчик нескольких оборотов присутствует только в моделях с функцией отслеживания данных по нескольким оборотам, в том числе в моделях с батареей.

Для получения сведений о битах состояния см. раздел «Биты состояния» далее по тексту.

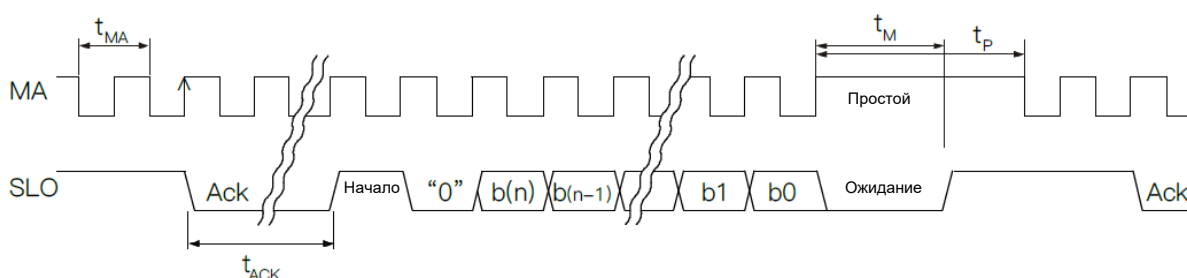
Интерфейс протокола BiSS-C

Принципиальная схема подключения интерфейсного модуля BiSS-C:



В интерфейсе реализована четырехпроводная система, состоящая из ведущего сигнала MA (прямого и инверсного), и ведомого сигнала SLO (прямого и инверсного). Кроме того, для ведущего сигнала (MA) 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:



Для синхронизации и получения последовательности тактовых импульсов в протоколе используется ведущий сигнал (MA). Пока линия MA 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 (MA). Начиная с 0, старший значащий бит (MSB) записывается на линию ведомого сигнала (SLO) на каждом нарастающем фронте ведущего сигнала (MA). На стороне контроллера данные на линии ведомого сигнала (SLO) считываются на спадающем фронте ведущего сигнала (MA), 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	
Время ожидания передачи	t_M		10 мкс	
Длительность паузы	t_P	20 мкс		

(1) При наличии у пользователя технологии фазовой компенсации для компенсации задержки между дифференциальными переходами, частота может достигать до 10 МГц.

После завершения передачи данных и окончания интервала передачи данных t_m , 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 t_p . Интервал t_{CL} должен быть меньше интервала t_m ; при этом во время любой операции чтения данный интервал может превысить t_m , чтобы прервать чтение.

Формат данных:

Биты	$b(24 + X) : b(9 + X)$	$b(8 + X) : b8$	$b7$	$b6$	$b5 : b0$
Длина	16 бит	X бит	1 бит	1 бит	6 бит
Данные	Счетчик нескольких оборотов ⁽¹⁾	Угол при одном обороте	Бит ошибки ⁽²⁾	Бит предупреждения ⁽³⁾	CRC(4)

(1) Счетчик нескольких оборотов присутствует только в моделях с функцией отслеживания данных по нескольким оборотам, в том числе в моделях с батареями.

(2) Бит ошибки активен при нахождении на низком уровне и может находиться на этом уровне только в моделях с батареями с режимом отслеживания данных по нескольким оборотам. Когда он равен 1, это указывает на то, что связанные с батареей биты состояния указывают на ее нормальный режим работы, а данные по нескольким оборотам надежны.

Когда он равен 0, это указывает на то, что биты состояния указывают на сигнал о низком напряжении батареи или об отсоединении батареи. Для решения проблемы см. раздел «Биты состояния».

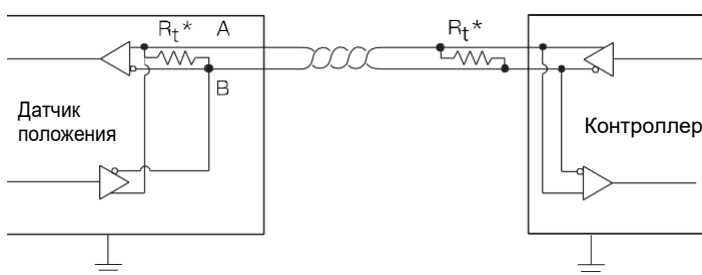
(3) Бит предупреждения активен при нахождении на низком уровне. Когда он равен 1, это указывает на отсутствие ошибок или предупреждений в системе. Когда он равен 0, это указывает на то, что система выдала, по меньшей мере, одну ошибку или предупреждение.

(4) Многочлен контроля с использованием циклического избыточного кода CRC равен $x^6 + x^1 + 1$ (т.е. 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.

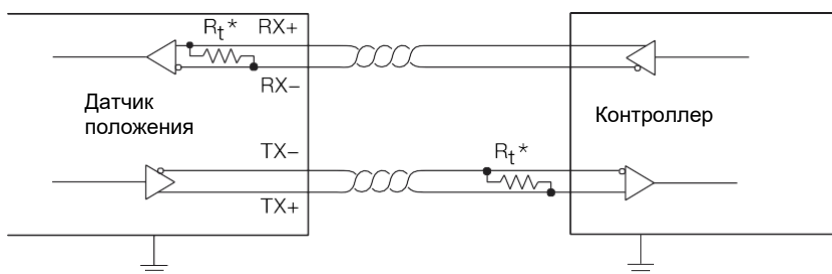
Интерфейсы протоколов RS485/RS422

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.

Принципиальная схема подключения интерфейсного модуля RS422:



В интерфейсе реализована четырехпроводная система, состоящая из дифференциального выхода TX и дифференциального входа RX. Кроме того, для дифференциального входа (RX) в датчик уже интегрирован оконечный резистор. Пользователю необходимо выполнить подключение оконечного резистора или же реализовать другое решение для согласования импеданса на стороне дифференциального входа (RX) в контроллере.

Базовый протокол интерфейсных модулей RS485 и RS422 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.

Конфигурация протокола:

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 (если не указано в приложении, то рекомендованное значение по умолчанию В):

Код	А	В
Скорость передачи данных (Мбит/с)	0.1152	2.5

Команды и данные:

Команда	Вход		Выходные данные	n	Возвращаемые данные (N байт)							
					B0	B1	B2	B3	B4	B5	B6	B7
0x30	Задание	Нулевое положение (1)	-	2	C	CRC						
0x31	Получение	Позиция	24	4	A2	A1	A0	CRC				
			24M	6	M1	M0	A2	A1	A0	CRC		
			24BM 24FM									
0x64	Получение	Позиция + состояние	24	5	A2	A1	A0	S	CRC			
			24M	7	M1	M0	A2	A1	A0	S	CRC	
			24BM 24FM									
0x74	Получение	Позиция + температура (2)	24	6	A2	A1	A0	T1	T0	CRC		
			24M	8	M1	M0	A2	A1	A0	T1	T0	CRC
			24BM 24FM									

(1) Для того, чтобы успешно задать нулевое положение необходимо с определенной периодичностью отправлять команду «0x31», а затем команду «0x30» в общей сложности 10 раз. При возвращении командой «0x30» значения счетчика 10, происходит установка нулевого положения.

(2) Информация о температуре подразумевает под собой температуру перехода на чипе.

В таблице выше буквы соответствуют данным следующим образом:

M	A	C	S	T	CRC
Данные для нескольких оборотов	Угол при одном обороте	Сброс отсчета от нуля	Состояние	Температура (2)	Проверка CRC (1)

Для получения сведений о битах состояния см. раздел «Биты состояния» далее по тексту.

(1) Байт CRC (многочлен CRC равен $x_8+x_7+x_6+x_5+x_4+x_3+x_2+x_1+1$; см. соответствующий метод расчета в приложении с алгоритмом CRC-8.), 计算 方法见附录 CRC-8 ($x_8+x_7+x_6+x_5+x_4+x_3+x_2+x_1+1$)).

(2) Метод расчета температуры: $\text{int16_t temp} = T0 \mid T1 \ll 8$; $\text{float tempFloat} = \text{temp} / 10.f$; Единица: °C

Пример (24BM):

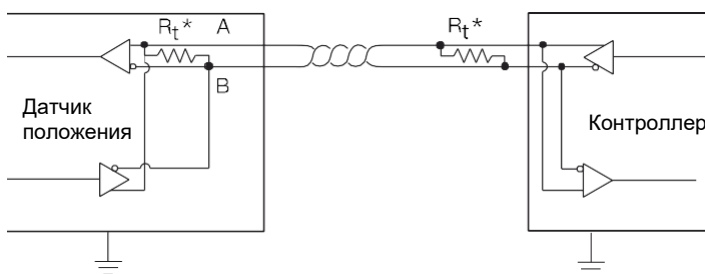
При отправке 0x32, вы получите uint8_t Buffer[7];

Затем, при использовании:

```
uint16_t multi = Buffer[0] << 8 | Buffer[1];
uint32_t angle = Buffer[2] << 16 | Buffer[3] << 8 | Buffer[4]; float
angleFloat = angle / (float)(1 << 24) * 360;
```

Интерфейс протокола T485

* интерфейс совместим с протоколом Tamagawa; 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.

Протокол T485 разработан на основе протокола RS485 и имеет определенный протокол взаимодействия. Интерфейс принимает запрос на выполнение операции размером 1 байт и возвращает соответствующие данные от датчика положения в соответствии с запросом. В конец данных добавляется контрольная сумма CRC-8 для проверки.

Конфигурация протокола:

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

Данные для запроса на выполнение операции:

Бит данных	b7 ~ b3	b2	b1	b0
Состав данных	Тип операции	0	1	0

Возвращаемые данные:

Байт	B0	B1	B(2 ~ n)	B(n + 1)
Состав данных	Запрос на выполнение операции (1)	Состояние	Возвращаемые данные	CRC (2)

(1) Возвращенный запрос на выполнение операции совпадает с отправленным запросом на выполнение операции.

(2) Байт CRC (многочлен CRC равен $x^8 + 1$, см. соответствующий метод расчета [CRC-8 в приложении](#))

B1, формат информации о состоянии:

Бит данных	b7	b6	b5	b4	b3	b2	b1	b0
Состав данных	0	Ошибка обмена информацией	Ошибка датчика положения	0	0	0	0	0

В (2 ~ n), тип операции и соответствующие возвращаемые данные (А для углового положения, М для нескольких оборотов, Е для ошибки):

Тип операции					Ввод	n	Возвращаемые данные							
b7	b6	b5	b4	b3			B2	B3	B4	B5	B6	B7	B8	B9
0	0	0	0	0	Получение углового положения	4	A0	A1	A2					
1	0	0	0	1	Получение значения по нескольким оборотам	4	M0	M1	M2					
0	0	0	1	1	Получение всех данных	9	A0	A1	A2	17	M0	M1	M2	E
1	1	0	0	0	Сброс углового положения (3)	4	A0	A1	A2					
0	1	1	0	0	Сброс значения по нескольким оборотам (4)	4	A0	A1	A2					

(1) An и Mn выровнены по левому краю, то есть если А — 17-битные данные, то старшие 7 бит A2 равны 0.

(2) Если тип операции не соответствует таблице В(2 ~ n), то выдается ошибка обмена информацией, а возвращаемые данные будут такими же, как и данные для получения углового положения.

(3) Для сброса углового положения необходимо отправить команду 10 раз подряд, чтобы она сработала.

(4) Для сброса значения по нескольким оборотам необходимо отправить команду 10 раз подряд, чтобы она сработала.

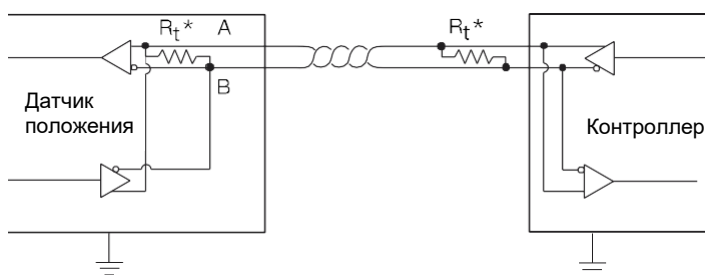
Е, бит ошибки (см. раздел «Биты состояния», в Т485 указываются только исключения ошибок):

b7	b6	b5	b4	b3	b2	b1	b0
Отсоединение батареи	Низкое напряжение батареи	0	0	0	0	0	0

См. индикацию светодиода в разделе «Биты состояния».

Интерфейс протокола BUS (высокоскоростная шина)

Принципиальная схема подключения высокоскоростной шины BUS:



Интерфейс представляет собой двухпроводную систему на основе дифференциальных сигналов А и В. Оба провода необходимо подключить к клеммам оконечных резисторов. Оконечный резистор на стороне датчика положения уже интегрирован в датчик положения. Пользователю необходимо подключить оконечный резистор на стороне контроллера.

Протокол обмена основан на протоколе RS485, передача данных осуществляется в формате UART, а рабочая частота составляет 2,5 Мбит/с. Интерфейс принимает запрос на выполнение операции размером 1 байт и возвращает соответствующие данные от датчика положения в соответствии с запросом. В конец данных добавляется контрольная сумма CRC-8 (x_8+1) для проверки.

Выбор версии:

Число разрядов на один оборот	Число разрядов для нескольких оборотов	Модель
XX	Нет	XX-D
	16	XXM-D

Конфигурация протокола:

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

Данные для запроса на выполнение операции:

Бит данных	b7	b6 ~ b5	b4 ~ b0
Состав данных	Контроль по нечетности	Тип операции	Адрес устройства

Возвращаемые данные:

Байт	B0	B(1 ~ n)	B(n + 1)
Состав данных	Запрос на выполнение операции	Возвращаемые данные	CRC

B(1 ~ n), типы операций и соответствующие возвращаемые данные (М для нескольких оборотов, А для углового положения, С для установки счетчика, S для состояния):

Тип операции Ввод		Версия модели		n	Возвращаемые данные					
b6	b5		Число разрядов на один оборот	Число разрядов для нескольких оборотов		B1	B2B3	B4	B5	B6
0	0	Получение информации	≤16	Нет	3	S	A0A1			
			≤16	16	5	S	A0A1	M0	M1	
			>16	Нет	4	S	A0A1	A2		
			>16	16	6	S	A0A1	A2	M0	M1
0	1	Задание нулевого положения			2	S	C			
1	0	Задание адреса			2	S	C			

Тип операции		Ввод	n	Возвращаемые данные		
b6	b5			B1	B2	B3
0	0	Получение информации	3	S	A0	A1
0	1	Задание нулевого положения	2	S	C	
1	0	Задание адреса	2	S	C	

Для получения сведений о битах состояния см. раздел «Биты состояния» далее по тексту.

Примечание:

1. Когда тип операции не соответствует таблице B(1 ~ n), например, 0b11, возврата данных не произойдет.
2. С — непрерывно работающий счетчик; такой возврат соответствует установке датчика без отсрочки. Возврат 10 указывает на то, что установка будет выполнена без отсрочки (процесс занимает не более 50 мс, в течение которых датчик не реагирует ни на какие команды).
3. Заводской адрес по умолчанию — 0x1F, что соответствует 0b11111.

Операция задания нулевого положения:

Для гарантии того, что установка нулевого положения не будет случайно изменена, необходимо попеременно и без перерывов отправить типы операций 00, 01, 00, 01, 00, 01, ..., 00, 01 в общей сложности десять раз (после отправки каждой команды необходимо дождаться ответа от датчика положения, прежде чем отправлять следующую команду). Число оставшихся циклов передачи данных можно определить на основе значения С, возвращаемого командой 01.

Примечание:



1. Когда команда, идущая перед 01 не является командой 00, требование начала последовательности не выполняется, и значение С равно 0.
2. Когда предпоследняя команда перед 01 не является 01, требование последовательности не выполняется, и значение С равно 1. Отсчет начинается с этого момента.

Операция настройки идентификатора (ID):

Для гарантии того, что настройка идентификатора ID не будет выполнена ошибочно, необходимо использовать определенную последовательность значений адресов для подтверждения входа в функцию настройки идентификатора (ID) перед выполнением конфигурирования идентификатора (ID). Значения адресов, отправляемые для настройки ID, соответствуют следующим (в два раза больше, чем значение С, возвращенное в прошлый раз):

Значение адреса	X	2	4	6	8	10	12	14	16	Y
С	Нет	Нет	Нет	4	5	6	7	8	9	10

Примечание:

1. Где значение X является произвольным, а значение Y — это фактическое значение адреса, который нужно сконфигурировать.
2. Первые три набора данных в последовательности не возвращают никаких данных. Это сделано для предотвращения случайного срабатывания, пока шина находится в рабочем состоянии. Поскольку данная команда располагает полномочиями, превышающими ограничения, предусмотренные для идентификатора (ID), она может приводить к произвольной передаче данных и получению от шины собирательного отклика.
3. При вставке другой команды значение С, возвращаемое следующей командой настройки, равно 1. С этого момента начинается отсчет.
4. Когда отправленное значение адреса не соответствует последовательности, возвращаемое значение С равно 1. С этого момента начинается отсчет.

Устройства на шине:

Все устройства на шине должны переходить в режим «сна» на определенное время, когда полученный идентификатор (ID) не соответствует идентификатору их собственного устройства. В это время они не должны отвечать на какие-либо команды на шине. Такой подход помогает предотвратить появление ответных пульсаций.

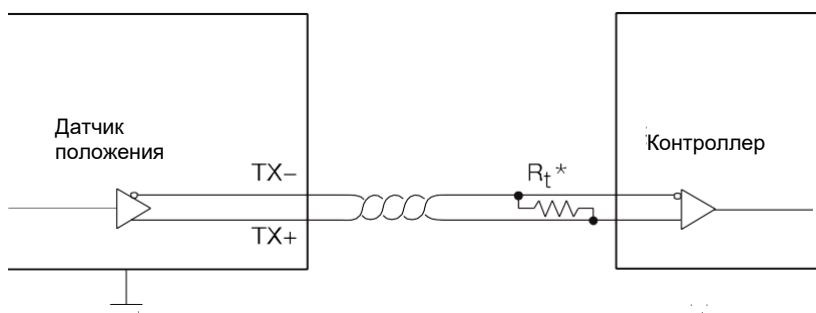
Для удобства продления интервала такой интервал в режиме «сна» определяется как $T_{SUSPEND}$. Метод расчета интервала показан в следующей таблице:

Число разрядов на один оборот	Число разрядов для нескольких оборотов	Число байт для режима «сна» шины BSUSPEND	Интервал в режиме «сна» шины TSUSPEND (мкс)
XX	YY	$\text{ceil}(XX/8) + YY/8 + 4$	$BSUSPEND * (1 + 8 + 1) / 2.5$

Пример: для модели 16M1-D $T_{SUSPEND}$ рассчитывается как $(\text{ceil}(16 / 8) + 8 / 8 + 4) * (1 + 8 + 1) / 2.5 = 28\text{мкс}$.

Интерфейс протокола циклической передачи данных PERIOD

Принципиальная схема подключения по интерфейсу PERIOD:



Интерфейс представляет собой двухпроводную систему на основе дифференциальных сигналов А и В. Оба провода необходимо подключить к клеммам оконечных резисторов. Пользователю необходимо выполнить подключение оконечного резистора на стороне контроллера.

Данный протокол сформирован на основе протокола RS422, с единственным отличием: он в активном режиме выдает данные наружу через дифференциальный выход TX с интервалом 1 мс и не реагирует на данные на линии дифференциального входа RX. Внутри датчика положения периодически будет срабатывать команда "d" (0x64) для отправки данных соответствующей команды. См. интерфейс протоколов RS485/RS422.

Биты состояния

В протоколах SSI/BISS-C/RS485/RS422 используемые биты состояния являются согласованными. При возникновении предупреждения или ошибки в выходных данных устанавливается определенный бит предупреждения или ошибки, что позволяет четко понять причину возникшей ситуации посредством битов состояния.

Позиции битов ошибки/предупреждения для каждого интерфейса:

	Бит ошибки	Бит предупреждения
SSI	b7	b6
BISS-C	b13	b12
RS485/RS422	b7	b6
BUS	b7	b6
PERIOD	b7	b6

Биты состояния:

Позиция	Присутствует только у моделей с режимом сохранения данных о нескольких оборотах при питании от батареи.		b3	b2	b1	b0
	b5	b4				
Описание	Отсоединение батареи	Низкое напряжение батареи	Слишком сильное магнитное поле	Слишком слабое магнитное поле	Температура за пределами установленного диапазона	Слишком высокая скорость
Светодиод мигает	Да	Да	Нет	Нет	Нет	Нет

Если бит предупреждения равен 1, то данные все еще продолжают оставаться корректными. В это время светодиодный индикатор состояния горит желтым, однако некоторые параметры битов состояния приближаются к своим предельным значениям. Можно проверить биты состояния, чтобы понять, что происходит. Если бит ошибки равен 1, это означает, что данные некорректны. 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.

Светодиод будет мигать с интервалом в 1 секунду, чтобы предупредить пользователя о возникновении соответствующей ошибки или предупреждения.

Биты состояния, связанные с батареей:

	b5	b4
	Отсоединение батареи	Низкое напряжение батареи
Бит предупреждения равен 1	-	Напряжение батареи ниже 2.9 В
Бит ошибки равен 1	Батарея отсоединена или напряжение слишком низкое во время отключения датчика положения от сети, что приводит к ошибке счетчика для нескольких оборотов. В результате данные по нескольким оборотам не являются корректными. При возникновении такой ошибки происходит сброс счетчика для нескольких оборотов на ноль.	Напряжение батареи ниже 2.7 В
Решение	Проверьте монтаж и напряжение батареи. Включите датчик положения заново, чтобы сбросить его.	Замените батарею датчика положения.



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];  
}
```

Расчет CRC-8 (X_8+1)

```
//poly =  $x_8+1$ 
```

```
//Искомое значение совпадает с результатом, найденным для многочлена
```

```
uint8_t calcCRC(uint8_t * buffer, uint8_t length){ uint8_t  
    temp = *buffer++;  
    while(--length){  
        temp = *buffer++ ^ temp;  
    }  
  
    return temp;  
}
```

Расчет CRC-6

```
#define DATA_TOTAL_BIT_LENGTH 47

//poly = x6+x5+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 calcBissCRC(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)
    #if 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)
    #if 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)
    #if 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)
    #if 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)
    #if DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH >= 0
        #if 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 #endif

    #undef CURRENT_CRC_BIT_LENGTH
    #define CURRENT_CRC_BIT_LENGTH (TOP_BYTE_BITLENGTH + CRC_BIT_LENGTH * 6)
    #if 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)
    #if 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)
    #if 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)
    #if DATA_WITHOUT_CRC_BIT_LENGTH - CURRENT_CRC_BIT_LENGTH >= 0
        crc = tableCRC6[crc ^ (secondWord >> (64 - CURRENT_CRC_BIT_LENGTH) & DATA_CRC_MASK)];
    #endif

    #if 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.

Например: для 17M это будет 47, а для 16 — 30.

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;                     //Первый фиксированный служебный
    байт BISS-C 0x82 uint8_t buffer[8]__attribute__((aligned(4))); //Буфер выровнен по
    границам 4 байт для быстрого расчета CRC
} receiveBuffer;

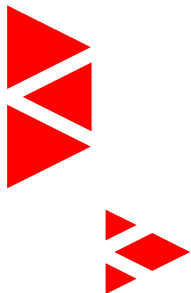
//Конфигурирование SPI (последовательного периферийного интерфейса) и DMA (прямого доступа к памяти)
//Используйте &receiveBuffer.placeholder в качестве адреса получения
//.....

//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

Период	Редакция	Внесенные изменения
28 сентября 2023 г.	Ред. 0.1	Первоначальный вариант
05 января 2024 г.	Ред. 0.2	Добавление размеров
01 мая 2023 г.	Ред. 0.3	Добавление типовых чертежей Добавление цветов проводов в скрутке



金 钢 科 技
KingKong.tech

Beijing Jingang Technology Co., Ltd.

Офис 712, улица Юньань 26, Научно-технологический парк, район Чанпин, Пекин

Веб-сайт: <https://kingkong.tech>

e-mail: contact@kingkong.tech

Тел.: 010-80111669