

Руководство пользователя  
OR-LINESENS-KTIRx10-I2C

Ермаков В.А.

12/2010

## 1 Описание регистров

Доступ к регистрам аналогичен работе с I<sup>2</sup>C EEPROM: при записи первый байт устанавливает адрес регистра, а дальше идет последовательная запись регистров с инкрементом указателя адреса. При чтении идет считывание регистров начиная с указателя адреса с инкрементом (см. [примеры](#)).

Регистры разделены на три области:

1. Только чтение (см. таб. [1](#));
2. Чтение-Запись (см. таб. [2](#));
3. Функции (см. таб. [3](#)).

### 1.1 Раздел только чтение

В этом разделе находятся: идентификатор устройства; версия прошивки; положение линии; результаты чтения оптодатчиков. Допускается только чтение, запись игнорируется.

Идентификатор и версия аналогична проекту OpenServo, но имеет отличный тип (device type, subtype = 0x02, 0x01).

Результаты чтения представлены в двух форматах:

1. RAW – необработанное время разрядки конденсатора, диапазон изменения [0;4000]. Т.к. такой диапазон не помещается в один байт одно значение записано в паре регистров REG\_RAW\_n\_[HI,LO]
2. CAL – диапазон RAW приведен к [0;100]. За диапазон RAW принимается [CAL\_MIN;CAL\_MAX].

Калиброванные значения вычисляются по формуле:

$$CAL_i = \frac{100(RAW_i - CAL\_MIN_i)}{CAL\_MAX_i - CAL\_MIN_i}$$

Положение линии вычисляется из калиброванных значений по формуле:

$$LINE = \frac{\sum_{i=0}^9 CAL_i \cdot i \cdot 100}{\sum_{i=0}^9 CAL_i}$$

Таким образом, когда линия находится в центре – 450. Когда линия ушла слева из-под датчика – 0. Когда линия ушла справа – 900.

В такой форме удобно использовать положение в ПИД регуляторе.

## 1.2 Раздел чтение-запись

В этом разделе находятся минимумы и максимумы для вычисления калиброванных значений. Допускается чтение и запись.

## 1.3 Раздел функции

В этом разделе находится вызов специальных функций. Допускается чтение и запись.

1. REG\_WHITE\_LINE – установка цвета линии;
2. REG\_ENABLED – включение/выключение датчиков;
3. REG\_RUN\_CALIBRATION – запуск калибровки (если записан не 0). Во время калибровки происходит 10 чтений датчиков и находится минимумы и максимумы для каждого. После чего полученные минимумы (максимумы) записываются в регистры калибровки если они меньше (больше) текущего значения. Если записать число больше 127, то калибровочные значения будут сброшены перед калибровкой;
4. REG\_SAVE\_REGS – сохранение калибровочных значений, цвета линии и адреса во встроенной энергонезависимой памяти (если записан не 0);
5. REG\_LOAD\_REGS – загрузка сохраненных настроек (если записан не 0);
6. REG\_I2C\_ADDRESS – смена I<sup>2</sup>C адреса;
7. REG\_RESET – перезагрузка если записан не 0.

## 1.4 Примеры

В примерах используется синтаксис команд ORFA.

Запись регистра REG\_ENABLED: S FE 4D 01 P

Где: FE – адрес записи; 4D – регистр; 01 – новое значение.

Запись регистровой пары REG\_CAL\_MAX\_0: S FE 24 0400 P

Где: FE – адрес записи; 24 – первый регистр; 0400 – новое значение REG\_CAL\_MAX\_0.

Чтение положения линии: S FE 04 S FF 02 P

Где: FE – адрес записи; 04 – регистр; FF – адрес чтения; 02 – количество байт для чтения.

Таблица 1: Регистры (раздел только чтение)

Имя	Адрес	Описание
REG_DEVICE_TYPE	0x00	Тип устройства (2)
REG_DEVICE_SUBTYPE	0x01	Субтип устройства (0)
REG_VERSION_MAJOR	0x02	Версия ПО
REG_VERSION_MINOR	0x03	
REG_LINE_POSITION_HI	0x04	Положение линии [HI:LO]
REG_LINE_POSITION_LO	0x05	
REG_CALIBRATED_0	0x06	Калиброванное значение сенсора. Это RAW значение диапазон которого приведен к [0;100]. За диапазон RAW принимается [CAL_MIN;CAL_MAX].
REG_CALIBRATED_1	0x07	
REG_CALIBRATED_2	0x08	
REG_CALIBRATED_3	0x09	
REG_CALIBRATED_4	0x0A	
REG_CALIBRATED_5	0x0B	
REG_CALIBRATED_6	0x0C	
REG_CALIBRATED_7	0x0D	
REG_CALIBRATED_8	0x0E	
REG_CALIBRATED_9	0x0F	
REG_RAW_0_HI	0x10	Не обработанные измерения времени разрядки конденсатора. Каждое значение в паре регистров [HI:LO]. Диапазон: [0;4000] мкс.
REG_RAW_0_LO	0x11	
REG_RAW_1_HI	0x12	
REG_RAW_1_LO	0x13	
REG_RAW_2_HI	0x14	
REG_RAW_2_LO	0x15	
REG_RAW_3_HI	0x16	
REG_RAW_3_LO	0x17	
REG_RAW_4_HI	0x18	
REG_RAW_4_LO	0x19	
REG_RAW_5_HI	0x1A	
REG_RAW_5_LO	0x1B	
REG_RAW_6_HI	0x1C	
REG_RAW_6_LO	0x1D	
REG_RAW_7_HI	0x1E	
REG_RAW_7_LO	0x1F	
REG_RAW_8_HI	0x20	
REG_RAW_8_LO	0x21	
REG_RAW_9_HI	0x22	
REG_RAW_9_LO	0x23	

Таблица 2: Регистры (раздел чтение-запись)

Имя	Адрес	Описание
REG_CAL_MAX_0_HI	0x24	Регистры калибровки максимального времени.
REG_CAL_MAX_0_LO	0x25	
REG_CAL_MAX_1_HI	0x26	
REG_CAL_MAX_1_LO	0x27	
REG_CAL_MAX_2_HI	0x28	
REG_CAL_MAX_2_LO	0x29	
REG_CAL_MAX_3_HI	0x2A	
REG_CAL_MAX_3_LO	0x2B	
REG_CAL_MAX_4_HI	0x2C	
REG_CAL_MAX_4_LO	0x2D	
REG_CAL_MAX_5_HI	0x2E	
REG_CAL_MAX_5_LO	0x2F	
REG_CAL_MAX_6_HI	0x30	
REG_CAL_MAX_6_LO	0x31	
REG_CAL_MAX_7_HI	0x32	
REG_CAL_MAX_7_LO	0x33	
REG_CAL_MAX_8_HI	0x34	
REG_CAL_MAX_8_LO	0x35	
REG_CAL_MAX_9_HI	0x36	
REG_CAL_MAX_9_LO	0x37	
REG_CAL_MIN_0_HI	0x38	Регистры калибровки минимального времени.
REG_CAL_MIN_0_LO	0x39	
REG_CAL_MIN_1_HI	0x3A	
REG_CAL_MIN_1_LO	0x3B	
REG_CAL_MIN_2_HI	0x3C	
REG_CAL_MIN_2_LO	0x3D	
REG_CAL_MIN_3_HI	0x3E	
REG_CAL_MIN_3_LO	0x3F	
REG_CAL_MIN_4_HI	0x40	
REG_CAL_MIN_4_LO	0x41	
REG_CAL_MIN_5_HI	0x42	
REG_CAL_MIN_5_LO	0x43	
REG_CAL_MIN_6_HI	0x44	
REG_CAL_MIN_6_LO	0x45	
REG_CAL_MIN_7_HI	0x46	
REG_CAL_MIN_7_LO	0x47	
REG_CAL_MIN_8_HI	0x48	
REG_CAL_MIN_8_LO	0x49	
REG_CAL_MIN_9_HI	0x4A	
REG_CAL_MIN_9_LO	0x4B	

Таблица 3: Регистры (раздел функций)

Имя	Адрес	Описание
REG_WHITE_LINE	0x4C	Если не 0 то цвет линии белый
REG_ENABLED	0x4D	Если не 0 то модуль работает
REG_RUN_CALIBRATION	0x4E	Запуск автокалибровки. Если записать 0xff то старые значения будут сброшены
REG_SAVE_REGS	0x4F	Сохранение настроек в EEPROM
REG_LOAD_REGS	0x50	Загрузка настроек из EEPROM
REG_I2C_ADDRESS	0x51	Смена адреса устройства
REG_RESET	0x52	Перезагрузка, удобно для входа в загрузчик

## 2 Bootloader

Загрузчик запускается если во время запуска замкнуты контакты MISO и MOSI.

Запись и чтение аналогичны работе с I<sup>2</sup>C EEPROM, только размер адреса 2 байта.

Размер страницы: 64 байт.

Для ATMega8: 0000–2000 – FLASH, 2000–2200 – EEPROM.

Для ATMega168: 0000–4000 – FLASH, 4000–4200 – EEPROM.

Адрес FFFF отвечает за запуск основного приложения.

Для работы с загрузчиком есть специальная программа в каталоге uploader исходных текстов прошивки. Она написана на языке python, для запуска требуется liborfa, opster и progressbar.