

Протокол общения по UART между STM32 и ESP8266(MQTT)

Автор: Волков Олег

Библиотека: uart_mqtt_lib.h

Краткое описание:

Данный протокол нужен лишь для того, чтоб появилась возможность подключить Stm32 по Wifi к MQTT серверу.

Ссылки:

[GitHub](#)

[YouTube](#)

Параметры порта:

- Бит в секунду: 115200
- Биты данных: 8
- Четность: нет
- Стоповые биты: 1
- Управление потоком: нет

Таблица 1. Описание протокола обмена данными, между ESP8266 и STM32

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Стартовый байт	Адрес переменной	Тип переменной	Младшие 2 байта данных		Старшие 2 байта данных		CRC	
0xFF	0X01	0x08	0xC2	0x0A	0x2E	0x14	0xFD	0xEA

где:

Byte0 - Стартовый байт. Всегда будет один и тот же: 0xFF

Byte1 - Всего может быть 254 переменных. Адрес 0xFF занят для служебных команд.

Пример: переменная, выделенная для обработки показаний температурного датчика имеет адрес - 0x01

Byte2 - Тип переменной:

Bool	0x01
int8_t	0x02
uint8_t	0x03
int16_t	0x04
uint16_t	0x05
int32_t	0x06
uint32_t	0x07
float	0x08

Byte3-4 - Младшие 2 байта данных.

Byte5-6 - Старшие 2 байта данных.

Примечание: допустим число (-34.545)float в hex будет как 0xC20A2E14, тогда Byte3 = 0xC2, Byte4 = 0x0A, Byte5 = 0x2E, Byte6 = 0x14

Byte7 - Контрольная сумма

Формула для расчета контрольной суммы: $(\sim(\text{Byte1}+\text{Byte2}+\text{Byte3}+\text{Byte4}+\text{Byte5}+\text{Byte6}))+1;$

Пример передачи данных типа float:	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
	0xFF	0x01	0x08	0xC2	0x0A	0x2E	0x14	0xFD	0xEA

Список служебных команд, передаваемых от ESP8266:

Служебная команда	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Wifi = OK	0xFF	0xFF	0xFF	0xAA	0xAA	0xAA	0xAA	0xFB	0x5A

Служебная команда	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Wifi != OK	0xFF	0xFF	0xFF	0xBB	0xBB	0xBB	0xBB	0xFB	0x16

Служебная команда	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
MQTT = OK	0xFF	0xFF	0xFF	0xCC	0xCC	0xCC	0xCC	0xFA	0xD2

Служебная команда	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
MQTT != OK	0xFF	0xFF	0xFF	0xDD	0xDD	0xDD	0xDD	0xFA	0x8E