**Attiny13:**

Считает импльсы от водосчетчика

Контролирует КЗ и обрыв по протоколу Namur

Контролирует заряд аккумулятора

Управляет esp8266:

1 Передача подсчитанного количества воды

2 Установка подсчитанного количества воды

3 Сообщение об аварии на счетчике

4 Сообщение об аварии на аккумуляторе

**Esp8266:**

1 Принимает количество воды

2 Устанавливает количество воды

3 Принимает сообщение об аварии на счетчике

4 Сообщение об аварии на аккумуляторе

**Протокол обмена на шине I2C:**

Esp8266 содержит регистры на шине I2C:

0X – счетчик воды X, можно как писать так и читать из него. При аварии на датчике воды выдается последнее подсчитанное значение. Счетчик содержит 4 байта. При записи в esp значение счетчика передается на сервер mqtt При чтении из esp значение принимается как начальное значение и отсчет ведется уже от этого значения.

10 – Только запись в esp. Состояние датчика воды, 0 – все нормально, 1 – обрыв, 2 – короткое замыкание

20 – Только запись в esp. состояние аккумулятора. 0 – все нормально, 1 – разраяжен (тут пока не понятно, чего там еще можно контролировать)

30 – состяние Wifi, 0 – обмен с хостом завершен, можно усыплять esp, 1 – сеть WiFi найдена, идет обмен, 2 – нет сети Wifi, идет поиск и подключение, 3 – не удалось подключится к сети.

При просыпании посылаем в esp8266 команду 00 c флагом записи и содержимое счетчика(ов) Если esp не отвечает, то делаем N попыток через N миллисекунд.

Простая запись и простое чтение без повторного старта.

Формат команды передачи значения счетчика в esp:

|  |  |  |
| --- | --- | --- |
| Адрес esp на шине+W | Команда 0X | 4 байта счетчика |

Запрос значения счетчика из esp для установки

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес esp на шине+R | Команда 0X | Ответ esp – 0 – нет данных, ff – есть данные | 4 байта счетчика |