**Ошибки:**

**Неверно сохраняется целевая температура в EEPROM.** Float **преобразуем в** uint8\_t/ **и сохраняем в одну ячейку EEPROM.**

**Закон регулирования протока шаровым краном:**

**https://www.youtube.com/watch?v=G2H5UQRad-w**

**Характеристики Arduino Mega 2560**  
Микроконтроллер ATmega2560  
Рабочее напряжение 5 В  
Входное напряжение (рекомендовано) 7-12 В  
Входное напряжение (предельное) 6-20 В  
Цифровые входы/выходы 54 (из которых 15 могут работать как выходы ШИМ)  
Аналоговые входы 16  
Макс.ток на входе/выходе 40 мА  
Макс.ток для вывода 3.3 В 50 мА  
Флеш-память (Flash memory) 256 Кб из которых 8Кб используется загрузчиком  
ОЗУ (SRAM) 8 Кб  
Энергонезависимая память (EEPROM) 4 Кб  
Тактовая частота 16 МГц

**DS18B20**

Младший разряд имеет вес 0,0625 °C.

Потоки в системе

|  |  |  |  |
| --- | --- | --- | --- |
| Операция | Время цикла | Блокирующая функция | Примечание |
| Самодиагностика при включении  -работа моторизированных кранов  -проверка датчиков температуры | 1 раз при включении | Не параллельный процесс |  |
| Считывание времени с м/с часов и вывод на индикатор | 1 с |  | Внешнее прерывание от RTC |
| Считывание информации с датчиков температуры DS18B20 (время интеграции температуры макс750мс и время считывания) | 5 с |  |  |
| Вывод информации о температуре на индикаторы на TM1637 | 5 с |  |  |
| Вывод информации о системном времени на индикаторы на TM1637 | 0,5 с |  |  |
| 1. Регулировка и поддержание оптимальной температуры ТТК (защита от холодной обратки)  2. Получение температуры дымовых газов на выходе ТТК котла | 30с | \*  (на время изменения положения крана) |  |
| Регулировка и поддержание температуры в системе отопления дома | 60с | \*  (на время изменения положения крана) | Возможно, увеличить время до нескольких минут.  Возможно, могут появиться колебательные процессы. |
| Обмен с wi-fi модулем ESP. Передача данных в систему IoT | 1c |  |  |
| Сохранение данных о температуре в ПЗУ | 300 с (при времени кратном 5 мин.) |  |  |
| Сканирование клавиатуры | 20 мс |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Проверка аварийного перегрева ТТК | 1с |  |  |
|  |  |  |  |
| Управление, общая логика работы системы | 10 мс |  |  |

Алгоритм управления моторизированным краном ТТК котла

Тестирование крана и определение времени полного переключения -switchtime

Температура в системе хранится в переменной целого типа.

Если при считывании получаем ошибку, то значению температуры соответствующего датчика присваиваем значение “-99”, что свидетельствует об ошибке на любом этапе получения температуры. Это значение должно анализироваться в управляющих функциях.

Идентификация датчиков в системе:

// 0- t1, подача cистемы

// 1- t2, обратка системы

// 2- t3, отвод трехходового крана ТТ

// 3- t4, подача ТТК

// 4- t5, обратка ТТК

// 5- t6, верх ТА подача от ТТК

// 6- t7, низ ТА обратка ТТК

// 7- t8, верх ТА подача в систему

// 8- t9, отвод трехходового крана системы

// 9- t10, низ ТА, обратка системы

// 10-t11, верх ТА

// 11-t12, низ ТА

// 12-t13, температура в помещении

// 13-t14, температура на улице

// 14-t15, подача ЭК

// 15-t16, обратка ЭК

// 16 - g\_t\_flueGases)

// 17 - g\_RoomSetpointCurrent

**Время. RTC**

Желательно использовать точный модуль часов на DS3231.

Вывод SQW часов настроен на 1Hz и генерирует прерывание int0 на Arduino. Соответственно, при прерывании считываю с часов время.

**Настройка при развертывании системы:**

1. Функция temperatureControlTTK – задать температуру дымовых газов ТТК котла g\_t\_flueGas. По умолчанию 120оС.
2. #define MAX\_TIME\_TESTING\_VALVE 10000 –время тестирования моторизироанных кранов 10с.
3. #define PIN\_TM1637\_DIO2 50//44 поставить 50

**Минимальная темпаратура теплоносителя в системе:**

//float mTminSysPodacha = 33; //минимальная температура подачи системы

if (temperature[13] < 0) { mTminSysPodacha = 33; } //вариант для зимы)

if (temperature[13] > 0) { mTminSysPodacha = 30; }

if (temperature[13] > 5) { mTminSysPodacha = 27; }

if (temperature[13] > 10) { mTminSysPodacha = 23; }

if (temperature[13] > 15) { mTminSysPodacha = 20; }

**MQTT**

**MQTT ThingSpeak**

<https://ww2.mathworks.cn/help/thingspeak/mqtt-api.html>

ThingSpeak™ MQTT supports only QoS 0.

## Syntax

[channels/<channelID>/subscribe/fields/field<fieldnumber>](https://ww2.mathworks.cn/help/thingspeak/subscribetoachannelfieldfeed.html#d123e20421)

[channels/<channelID>/subscribe/fields/+](https://ww2.mathworks.cn/help/thingspeak/subscribetoachannelfieldfeed.html#d123e20436)

Канал управления: **Tonshaevo\_HCS\_MQTT**

Channel ID: 1627034

Параметры брокера MQTT для канала:

Регистрационные данные для устройства:

USERNAME "KjMLOiQ0FyYiGDwDHwIkACA"

CLIENT\_ID "KjMLOiQ0FyYiGDwDHwIkACA"

PASSWORD "5IcKiXd+RpWddVPe7FPJ/tHJ"

Topic:

channels/<channelID>/publish

channels/<channelID>/publish/fields/field<fieldnumber>

<https://ww2.mathworks.cn/help/thingspeak/mqtt-api.html?s_tid=CRUX_lftnav>

Используются следующие поля:

|  |  |  |
| --- | --- | --- |
| Поле MQTT брокера | Наименование поля | Назначение |
| Field1 | cmdFromClient | Запрос, команда от клиентского приложения к устройству |
| Field2 | cmdFromDevice | Сообщение от устройства клиентам. |
| Field3 | GTargetTemp | Переданное от Mega значение глобальной целевой температуры в системе, без учета расписания. |
| Field4 |  |  |
| Field5 |  |  |
| Field6 |  |  |
| Field7 |  |  |
| Field8 |  |  |
|  | **err** | уведомления об ошибке клиенту |
|  | **sta** | уведомление о старте прошивки (перезапуск микроконтроллера) |
|  | **err** | уведомления об ошибке клиенту |
|  | **tme** | сообщения о синхронизации времени |
|  | **pwr** | сообщения о включении/выключении устройства (с флагом retain) |
|  |  |  |

Сообщения поступают в следующем формате, передаются в виде строки:

?параметр = значение

?команда

|  |  |  |  |
| --- | --- | --- | --- |
| Сообщения от клиентов | | | |
| **Команда** | Описание | **Пояснение** | Примечание |
| **?mega=1** | - | Модуль mega посылает данные о своей работоспособности. |  |
| **?sendtemp=[floatX]** | Поступление температуры датчика номер X | Модуль mega посылает данные о температуре датчика Х (X =A, то всех датчиков по порядку) | Ответ модуля mega на команду **?reqesttemp=X** |
| **?sendtemp=[float1];[float2];….;** | Поступление температуры всех датчиков |
| **?sendtempХ=[floatX]** | Поступление температуры датчика номер X |  |  |
|  |  |  |  |
| **?sendTargetTemp=[float]** | Поступление текущей целевой температуры с учетом расписания |  | Ответ модуля mega на команду **?reqestTargetTemp** |

### Топики сообщений

Используемые устройством топики и их назначение приведены в таблице

| **Топик** | **Назначение** |
| --- | --- |
| **cmd** | получение команды управления от клиента |
| **dta** | отправка запрошенных данных клиенту, в ответ на полученную команду, например '$6 7 |
| **err** | уведомления об ошибке клиенту |
| **sta** | уведомление о старте прошивки (перезапуск микроконтроллера) |
| **alm** | сообщения о событиях будильника |
| **amd** | сообщения о наступлении событий авторежимов по времени |
| **wtr** | сообщения о получении актуальной погоды |
| **tme** | сообщения о синхронизации времени |
| **pwr** | сообщения о включении/выключении устройства |
| **sdc** | сообщения о событиях SD-карты (загрузка файла эффекта) |
| **txt** | сообщения о событиях бегущей строки - запуск, окончание |
| **stt/xx** | сообщения о текущем статусе параметров устройства - индивидуальный режим, где XX - имя (ключ) параметра |