**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. |
| 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

**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