



## ВЕРСИЯ 1.6.\*

**GyverControl** - универсальный контроллер-таймер для теплицы и других мест, где нужна автоматизация по таймеру или показателям микроклимата/другим датчикам. Ссылки на все компоненты, схемы, инструкции и другая информация находятся на странице проекта на официальном сайте: <https://alexgyver.ru/gyvercontrol/>. Прошивка и всё относящееся к проекту (картинки схем, файлы печатных плат) находятся в репозитории на GitHub, прямая ссылка на скачивание <https://github.com/AlexGyver/GyverControl/archive/master.zip>

## Оглавление

Железо.....	2
Программные фишки .....	3
Применение как контроллер полива .....	3
Применение как контроллер инкубатора.....	3
Применение как контроллер аквариума/террариума.....	4
Другие применения контроллера .....	4
Управление системой GyverControl.....	4
Описание режимов и настроек.....	5
Режимы работы каналов.....	5
Настройки каналов реле .....	7
Настройки каналов серво.....	7
Настройки канала привода .....	8
Меню с переводом .....	8
Главное меню, уровень вложенности 0 .....	8
Настройка режима, уровень вложенности 1 .....	8
Описание интерфейса .....	9
Экран отладки (DEBUG) .....	9
Экран сервиса (SERVICE) .....	10
Экран настроек (SETTINGS).....	10
Структурная схема меню (версия 1.3) .....	10
Карта печатной платы v1.1-1.2 .....	11
Распиновка контроллера.....	11
Настройка конфигурации системы в коде программы .....	11
Подключение специальных датчиков .....	12
DHT11/DHT22 .....	12
DS18b20 .....	13
Термистор.....	14
Датчик углекислого газа MH-Z19 .....	15

ПИД и Рассвет. Схемы.....	15
Настройка ПИД регулятора .....	17
Общие понятия .....	17
Ручная настройка .....	18
Отладка ПИД .....	19
Автоматическая настройка ПИД .....	19
Расписание ПИД регулятора .....	20
Пример: инкубатор .....	21
Пример: террариум .....	22
Советы по подключению реле .....	22
Постоянный ток.....	23
Постоянный и переменный ток .....	24
Переменный ток .....	24
Проблема с нехваткой памяти .....	25
Проблема .....	25
Отключение компонентов системы.....	25
Использование другого загрузчика .....	25
Ещё раз, инструкция по загрузке прошивки .....	25
Watchdog .....	26
Прошивка загрузчика OptiBoot .....	26
Подключение .....	26
Прошивка .....	27
Загрузка GyverControl .....	28
Изменения в версиях.....	29

## Железо

- **Arduino Nano** (ATmega328p) как главный контроллер системы
- **10 независимых каналов** управления с разными функциями:
  - **7 каналов (+2)** с логическим выходом 5V, к которым можно подключать обычное/твердотельное реле, силовые ключи (транзисторы, модули на основе транзисторов)
  - **2 канала** сервоприводов, подключаются обычные модельные серво больших и маленьких размеров. Могут работать как 2 дополнительных канала реле (см. главу Настройка конфигурации системы)
  - **1 канал** двухстороннего управления линейным электроприводом с концевиками ограничения движения и с работой по тайм-ауту. Может работать по ПИД регулятору.
  - **7 каналов** могут работать в режиме ПИД регулятора
  - **4 канала** имеют поддержку режима «рассвет»
- Бортовой датчик температуры и влажности воздуха (**BME280**)
- **4 канала для датчиков**, могут работать в разных сочетаниях:
  - **4 аналоговых датчика** (влажности почвы или любых других)
  - **DHT11/DHT22** – цифровой датчик температуры и влажности
  - **Термистор** – аналоговый датчик температуры
  - **DS18b20 (Dallas)** – цифровой датчик температуры. Возможна установка нескольких (с версии 1.6)
  - **MH-Z19** – датчик углекислого газа (с версии 1.5)
- Модуль опорного (реального) времени **DS3231** с автономным питанием
- Большой **LCD дисплей** (LCD 2004, 20 столбцов, 4 строки)
- Орган управления – **энкодер**

## Программные фишки

- Хранение всех настроек в энергонезависимой памяти (**не сбрасываются при перезагрузке**)
- Датчики влажности почвы (все аналоговые датчики) не находятся под постоянным напряжением, оно подаётся **только на момент опроса**, что позволяет продлить жизнь даже самым дешёвым датчикам влажности почвы (напряжение подаётся за 50 мс до опроса и выключается через 50 мс после).
- Каждый из 10 каналов (7 реле, 2 серво и 1 привод) имеет **индивидуальные настройки** и может работать по таймеру или датчикам
- **6 режимов работы** для разных каналов: три разных таймера, работа по условию с датчиков, ПИД и рассвет
- Серво работает с библиотекой **ServoSmooth**, это обеспечивает плавное их движение: плавный разгон и торможение с ограничением максимальной скорости, а также отсутствие рывков и незапланированных движений при старте системы
- *Версия 1.4+* : добавлены режимы ПИД регулятора (**PID**) и режим Рассвет (**Dawn**) для каналов 3 и 4 (ШИМ сигнал) и обоих каналов серво
- *Версия 1.4+* : каналы серво можно использовать для подключения реле
- *Версия 1.5+* : добавлен режим ШИМ реле, низкочастотный ШИМ для режима ПИД
- *Версия 1.6+* : добавлена калибровка ПИД регулятора
- Линейный привод имеет концевики, внешние кнопки для управления и настройку скорости движения. Частота ШИМ драйвера – 31 кГц, т.е. не пищит
- **Экран отладки**, где отображается вся текущая информация о состоянии железа и датчиков
- **Графики** температуры и влажности воздуха и показаний с аналоговых датчиков, 6 графиков:
  - Графики за 15 минут, 15 часов и 15 суток. Сохраняются и доступны одновременно, отображают усреднённые значения каждый за свой период (*с версии 1.6*)
  - Почасовой и посutoчный хранятся в памяти и не сбрасываются при перезагрузке (*с версии 1.6*)
- **Сервисное меню**, позволяющее вручную управлять каждой железкой

## Применение как контроллер полива

- Периодичный полив (реле)
  - Схема с индивидуальными помпами/клапанами
  - Схема с одной помпой и несколькими клапанами
- Полив на основе показаний датчиков влажности почвы
- Полив в указанные дни недели с окончанием по таймеру ИЛИ по показанию с датчика
- Управление освещением (реле) с привязкой ко времени суток
- Проветривание (привод открывает окно/серво открывает заслонку) по датчику температуры или влажности воздуха
- Увлажнение (включение увлажнителя) по датчику влажности воздуха
- Обогрев (включение обогревателя) по датчику температуры
- Выполнение действий сервоприводом (нажатие кнопок на устройствах, поворот рукояток, поворот заслонок, перемещение предметов) по датчику или таймеру

## Применение как контроллер инкубатора

- Поддержание температуры с высокой точностью (до 0.05 градуса) при помощи ПИД регулятора
  - Можно использовать обогреватели постоянного или переменного тока
- Поддержание влажности при помощи ПИД регулятора
- Автоматическая калибровка (подбор коэффициентов) ПИД регулятора
- Расписание для ПИД регулятора, позволяющее настроить закон изменения температуры/влажности на любой период
- Линейный привод или серво могут поворачивать лотки по расписанию
- Поддержка нескольких высокоточных датчиков температуры (Dallas) для вывода среднего, максимального или минимального значений по объёму

## Применение как контроллер аквариума/террариума

- Поддержание температуры с высокой точностью (до 0.05 градуса) при помощи ПИД регулятора
  - Можно использовать обогреватели постоянного или переменного тока
- Поддержание влажности при помощи ПИД регулятора
- Автоматическая калибровка (подбор коэффициентов) ПИД регулятора
- Расписание для ПИД регулятора, позволяющее настроить закон изменения температуры/влажности в течение суток
- Сервопривод может использоваться для кормёжки по расписанию
- Режим «Рассвет» обеспечит животным приближенные к реальным условия освещения
- Поддержка нескольких высокоточных датчиков температуры (Dallas) для вывода среднего, максимального или минимального значений по объёму

## Другие применения контроллера

- Система поддерживает 4 аналоговых датчика, это не обязательно должны быть датчики влажности почвы, у китайцев полно других «датчиков-модулей», которые **точно так же** подключаются к схеме:
  - **Датчик света:** «умная» система освещения, резервное освещение
  - **Термистор** (до 80 градусов): контроль нагрева объекта
  - **Датчик звука:** закрывание окна при сильном шуме снаружи (почему нет? = )
  - **Датчик ИК излучения** (датчик пожара) – разные варианты сигнализации, или даже тушения (включаем помпу с водой, открываем кран сервой)
  - **Датчик дождя:** закрытие окон, сигнализирование, включение помп на откачу
  - **Датчик уровня воды/датчик наличия воды:** автоматическое наполнение резервуара, автоматическая откачка воды помпой из ёмкости/подвала, перекрытие водяных магистралей при протечке, сигнализация о протечке
  - **Газоанализаторы** в ассортименте: сигнализатор или даже проветривание (открываем окно) по уровню угарного газа и других промышленных газов
  - **Оптический датчик препятствия:** тут нужна фантазия
  - **Потенциометр:** как дополнительный орган контроля системы
- Сервопривод довольно универсальная штука, может открывать/закрывать заслонки, может нажимать кнопки других устройств, вращать ручки регулировки других устройств, с приделанным шатуном получает возможность линейно перемещать предметы/ползунки других устройств. Сервоприводы есть разных размеров, от микро (2 кг/см) и средних (13 кг/см) до весьма мощных (50 кг/см)
- Реле умеет замыкать контакты питания и управлять любыми устройствами, также реле может включить блок питания (например светодиодной ленты). Реле можно поставить параллельно проводам к кнопке другого устройства, и оно будет его включать или выключать.
- *Версия 1.4* и выше позволяет поддерживать температуру при помощи **ПИД** регулятора, для террариумов/инкубаторов/любого поддержания температуры:
  - Подавать **ШИМ** сигнал на полевой транзистор, управляющий нагревом
  - Поворачивать сервоприводом крутилку сетевого диммера
- *Версия 1.4* и выше имеет режим **Рассвет**, позволяющий использовать контроллер для аквариума/террариума и прочих «животных ферм»
- *Версия 1.5+* выше умеет работать с 1 и 2 каналами реле в режиме **ПИД** для управления температурой
- В *версии 1.5* появилась поддержка датчика CO2, что позволяет использовать контроллер для проветривания в жилом помещении или наоборот, для поддержания высокого уровня CO2 в теплице/боксе.

## Управление системой GyverControl

Основным органом управления является энкодер, рукоятку которого может вращать и нажимать (она является кнопкой). При запуске системы мы попадаем на настройку канала 0. Вращая рукоятку энкодера можно перемещать курсор выбора (стрелочка) по пунктам меню. Чтобы изменить значение выбранного пункта, нужно нажать рукоятку энкодера и повернуть её, удерживая нажатой. Удержаный поворот при выбранном имени канала – смена канала для настройки. Листаем направо и у нас будет по порядку 7 каналов реле, два серво и линейный привод. Чтобы перейти к настройке режима, нужно навести на него курсор и кликнуть кнопкой, не поворачивая. Откроется окно настройки режима, выйти из которого можно кликнув по надписи BACK (назад).

Удерживая и вращая рукоятку на выбранном названии режима можно сменить режим, всего их 4. В корне меню (выбор каналов) листая налево от канала 0 будет экран отладки (DEBUG), режим настроек (SETTINGS) и сервисный режим (SERVICE). На экране отладки показаны все текущие положения реле, приводов и показания с датчиков. Вращая рукоятку на экране отладки последовательно листаются суточные графики показаний с датчиков: температура воздуха, влажность и показания с аналоговых датчиков. Деления на графике имеют шаг 1.6 часа. На экране сервиса можно управлять любым каналом в ручном режиме, при активном экране сервиса автоматика не работает, система находится полностью в ручном режиме. Поворотом рукоятки можно выбрать нужный канал, положение серво или настройку текущего времени, и удержанным поворотом её изменить. Если включить систему с зажатой рукояткой энкодера, произойдёт полный сброс настроек каналов и режимов. При удержании кнопки энкодера более двух секунд (без поворота рукоятки) привод передвинется в противоположное направление, при повторном нажатии – передвинется обратно. Сделано для доступа в теплицу, у которой привод управляет дверью.

**Версия 1.2 и выше:** выбор типа управления энкодером, настройка **CONTROL\_TYPE** в прошивке. При установке в 0 навигация осуществляется как описано выше. При установке в 1 логика управления строится следующим образом: выбор пункта меню поворотом энкодера, курсор – стрелочка. При клике курсор меняется на «галочку» >, при повороте с курсором галочкой величина изменяется. Ещё раз кликаем – курсор меняется на стрелочку и можно выбирать нужный пункт меню.

**Версия 1.5 и выше:** быстрый поворот энкодера меняет величину с большим шагом

## Описание режимов и настроек

### Режимы работы каналов

1. **Timer** – простой периодический таймер: задаются периоды **Period** и время **Work** в формате ЧЧ:ММ:СС. С периодом Period совершается выбранное действие и выполняется в течение периода Work. Например, Period стоит 1 час, Work – 10 секунд. Каждый час будет совершаться действие в течение 10 секунд, то есть если выбран канал реле, то реле включится и выключится через 10 секунд, затем снова включится через час и выключится через 10 секунд и так далее. Как канал ведёт себя на участке Work задаётся в параметре **Direction**, то есть это может быть вкл/выкл и выкл/вкл (реле), направо/налево и налево/направо (серво) и открыть/закрыть и закрыть/открыть (линейный привод). Данный режим не имеет привязки к реальному времени, перезагрузка системы сбрасывает текущий таймер. Внимание! Work не должна быть больше Period! На этом же экране выводится время, оставшееся до следующего срабатывания **Left**.

- Мин. значение: 1 секунда
- Макс. значение: 999 часов
- Привязка к реальному времени: нет
- Применение: полив в гидропонных системах, проветривание без датчика

2. **Timer RTC** – периодический таймер, в отличие от предыдущего обладает привязкой к реальному времени, имеет настройку **Period** включения и продолжительности **Work** (в секундах), которая будет совершаться, и **Start hour** – начального часа, с которого начинается отсчёт периода (для периодов больше 2 часов).

Например, период 15 минут, работа 10 секунд: каждые 15 минут будет производиться действие продолжительностью 10 секунд. Привязка к реальному времени работает следующим образом: действие будет совершаться с выбранным периодом от начала часа, то есть если выбран 15 минутный, то действие будет в 0, 15, 30 и 45 **минут каждого часа**. Если выбранный Period больше часа (от двух и более) то можно выбрать час Start from, от которого пойдёт отсчёт. Все периоды кратны 24 часам, поэтому работа начинается в одни и те же часы каждого дня! Пример: Period 8 часов, начальный час 0. Действие будет выполнено в 0, 8 и 16 часов каждого дня. Если поставить начальный час (Start hour) 3 часа, то действие будет выполнено в 3, 11 и 19 часов каждого дня. При сбросе питания следующее действие будет совершено в ближайшее время «будильника». В версиях выше 1.4 добавлены периоды, кратные суткам (1, 2.. 7). В них таймер будет срабатывать в час, отмеченный как start hour. Внимание! Work не должна быть больше Period!

- Периоды на выбор: каждые 1, 5, 10, 15, 20, 30, 60 минут и 1, 2, 3, 4, 6, 8, 12, 24 часа, 1, 2.. 7 суток
- Привязка к реальному времени: да
- Применение: полив в гидропонных системах, проветривание без датчика

Период	Раз в сутки	Когда срабатывает
1 мин	1440	Каждую минуту
3 мин	480	0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57 мин. каждого часа
5 мин	288	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 мин. каждого часа
10 мин	144	0, 10, 20, 30, 40, 50 мин. каждого часа
15 мин	96	0, 15, 30, 45 мин. каждого часа
30 мин	48	0, 30 мин. каждого часа
1 час	24	Каждый час
2 часа	12	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22 часа каждого дня (+ сдвиг на стартовый час)
3 часа	8	0, 3, 6, 9, 12, 15, 18, 21 час каждого дня (+ сдвиг на стартовый час)
4 часа	6	0, 4, 8, 12, 16, 20 часов каждого дня (+ сдвиг на стартовый час)
6 часов	4	0, 6, 12, 18 часов каждого дня (+ сдвиг на стартовый час)
8 часов	3	0, 8, 16 часов каждого дня (+ сдвиг на стартовый час)
12 часов	2	0, 12 часов каждого дня (+ сдвиг на стартовый час)
24 часа	1	0 часов каждого дня (+ сдвиг на стартовый час)

3. **Week** (бывший Day) – простой таймер на одно действие с привязкой к реальному времени, имеет настройку **On** (время в формате ЧЧ:ММ:СС) – время, с которого действие активно, и **Off** (время в формате ЧЧ:ММ:СС) – время, с которого действие не активно. Также имеется 7 «ячеек» - дней недели **Days**, с понедельника по воскресенье. При перезагрузке действие вернётся в нужное положение согласно текущему времени.

Пример: таймер настроен на 6 и 20 часов (Start и Stop). Соответствующее текущему каналу и параметру Direction действие будет активно с 6 до 20 часов, и неактивно с 20 до 6 часов утра следующего дня. При внезапной перезагрузке система совершит действие так, как оно должно быть на этом отрезке времени, то есть из прошлого примера если в промежуток между 6 и 20 часами произойдёт внезапная перезагрузка, при запуске система активирует действие по каналу. Внимание! On должен быть меньше Off! В версии 1.5 и выше On может быть больше Off, таким образом можно установить время работы например с 18 до 6 утра следующего дня. С 6 до 18 канал будет выключен.

Также режим имеет настройку **Global**, которая вынуждает любой другой режим в пределах этого канала работать «по расписанию» Week. Что это даёт: например, можно настроить полив во вторник и пятницу с 17 до 18 часов вечера (из бочки), поставить галочку Global и настроить режим Sensor под полив. Как это будет работать: система будет поливать этот канал по режиму Sensor, но делать это только по расписанию (вторник и пятница 17-18). Если global активен – к названию остальных режимов прибавляется надпись +W, что означает «+ week». Таким образом например **Timer+W** будет поливать со своим периодом в выбранные дни недели в выбранное время, Sensor+W будет поливать по сенсору в выбранные дни в указанное время, и т.п.

- Выбор дня недели
- Выбор времени: 0-23 часа, кратно 1 часу
- Привязка к реальному времени: да
- Применение: идеальный режим для освещения и редкого полива
- Возможность стать глобальным для остальных режимов этого канала

4. **Sensor** – действие на основе датчика. С периодом опроса **Period** опрашивается выбранный датчик под названием **Sensor** и при превышении порогового значения **maxV** и выполняется действие согласно выбранному каналу (реле/серво/привод). Действие «отключится» при достижении величиной порога **minV**, таким образом реализован **гистерезис**. Period опроса задаётся в секундах или минутах (по мере увеличения). Датчик выбирается из списка: **Air t.** – температура воздуха, **Air h.** – влажность воздуха и 4 аналоговых датчика (влажности почвы) с **SENS\_1** по **SENS\_4**. Пороговое значение (**minV** и **maxV**) задаётся с 0 до 255 с шагом 1 до значения 50 и с шагом 10 начиная от 50 (датчики влажности почвы имеют диапазон значений 0-255). Например, выбран датчик температуры воздуха, период опроса 1 час и пороговое значение 25. Каждый час система проверяет температуру, при превышении 25 градусов будет выполнено соответствующее каналу действие (включить реле, открыть окно). Через час будет снова произведена проверка. **Важно:** пороговые значения можно указывать как максимум больше минимума, так и наоборот, это инвертирует логику работы режима Sensor независимо от направления работы канала (Direction).

- Применение: открытие/закрытие створок по температуре/влажности (привод), полив по влажности почвы, управление вентилятором/увлажнителем (реле) или заслонками (серво) по температуре/влажности.

5. **PID** – пропорционально-интегрально-дифференциальный регулятор, позволяет с высокой точностью поддерживать управляемую величину (нагреватель-температура, заслонка-температура, вентилятор-температура, вентилятор-влажность, и так далее). Режим доступен для каналов 3 и 4 (отмечены звёздочкой), а также обоих каналов серво в режиме серво. [В версии 1.5 можно включить ПИД на каналах 1 и 2 \(низкочастотный ШИМ для реле\)](#). Имеет настройки коэффициентов **P**, **I**, **D** (D вам скорее всего не пригодится в реальной работе, но он там всё равно есть). Выбираем **Sens** – источник входного сигнала – один из сенсоров, как в режиме **Sensor** (**Air t.** – температура воздуха, **Air h.** – влажность воздуха и 4 аналоговых датчика (влажности почвы) с **SENS\_1** по **SENS\_4**). Настройка **Set** указывает, к какому значению показания с выбранного датчика регулятор будет стараться приводить систему. Настройка **T** задаёт период итерации расчёта, для медленных процессов есть смысл поставить побольше (читайте в отдельной главе «Настройка ПИД регулятора»). Настройки **min** и **max** отвечают за минимальный и максимальный управляющий сигнал с данного канала, для каналов 3 и 4 это ШИМ сигнал, рабочий диапазон 0-255. Для каналов серво это угол, 0-180 градусов.

- Применение: поддержание заданной величины (температура, влажность) не релейным способом, т.е. плавно и без резких включений. ШИМ сигнал может управлять транзистором, который отвечает за нагреватель. Серво может поворачивать заслонки (проветривание) или крутилки диммеров для управления сетевыми нагревателями, вентиляторами и прочим оборудованием.

6. **Dawn** – режим «рассвета» для контроля освещения с плавным рассветом и закатом. Режим доступен для каналов 3 и 4 (отмечены звёздочкой), а также обоих каналов серво в режиме серво. Плавно включается в час **Start** на протяжении **Dur** минут, затем выключается в час **Stop** в течение **Dur** минут. Включается до максимального значения, указанного в **max**, и выключается до **min**. На каналах 3 и 4 эта величина задаёт скважность ШИМ сигнала, рабочий диапазон 0 – 255. Управлять можно полевым транзистором, например, светодиодной лентой. На каналах серво рабочий диапазон 0 – 180, градусов поворота вала серво. Может управлять крутилкой сетевого диммера, для ламп накаливания или диммируемых светодиодных.

- Применение: организация условий освещённости, приближенных к реальным, для аквариумов, терариумов, курятников и проч.

## Настройки каналов реле

1. **Direction** – как ведёт себя реле при активации по таймеру/датчику. **ВКЛ-ВЫКЛ** или **ВЫКЛ-ВКЛ**
2. **Type** – логика работы реле

- **Relay** – канал реле ведёт себя как обычное реле, может использоваться для управления любой нагрузкой постоянного или переменного тока (управлять сетевыми устройствами): полив индивидуальными помпами, полив индивидуальными клапанами от источника воды под давлением, управления увлажнителями, обогревателями, вентиляторами, приборами освещения и всем другим подобным. Не зависит от других каналов.
- **Valve** – тип канала реле для системы, где есть общая помпа/клапан от источника воды и несколько индивидуальных клапанов на полив разных участков. Канал реле, настроенный как клапан, одновременно со своей активацией (по таймеру/датчику) активирует другой канал/каналы, настроенный как **общий**.
- **Common** – тип канала реле для системы, где есть общая помпа/клапан от источника воды и несколько индивидуальных клапанов на полив разных участков. Канал реле, настроенный как общий, не имеет настроек режима. Вместо этого он **активируется сам** одновременно с любым другим каналом, настроенным как **клапан**. Автоматически сам деактивируется при отсутствии неактивных каналов клапанов.

## Настройки каналов серво

1. **Direction** – как ведёт себя серво при активации по таймеру/датчику. Поворот в направлении **МИН-МАКС** угол или наоборот, **МАКС-МИН** угол
2. **Пределы** – углы поворота серво от 0 до 180 градусов с шагом 10

3. Дополнительно: в скетче в секции настроек есть настройка максимальной скорости движения сервоприводов (SERVO1\_SPEED и SERVO2\_SPEED) и их ускорение на разгон и торможение (SERVO1\_ACC и SERVO2\_ACC). Я не стал вносить их в настройки сервисного меню и каналов, т.к. они не так часто нужны.

### Настройки канала привода

1. **Направление** – как ведёт себя привод при активации по таймеру/датчику, **ОТКРЫТЬ-ЗАКРЫТЬ** или **ЗАКРЫТЬ-ОТКРЫТЬ**
2. **Таймаут** – время, которое будет подаваться сигнал на движение привода. Концевик (если он есть) прервёт движение привода

### Меню с переводом

#### Главное меню, уровень вложенности и 0

1. **Service** (Сервис)
2. **Debug** (Экран отладки)
3. **Channel 0 – Channel 6** (Каналы реле 0-6)
  - **Mode** (Режим) - клик для перехода в настройки режима
  - **Direction** (Направление работы)
    - On-Off
    - Off-On
  - **Type** (Тип канала реле)
    - Relay (Реле)
    - Valve (Клапан)
    - Common (Общий)
4. **Servo 1 – Servo 2** (Каналы серво 1 и 2)
  - **Mode** (Режим) - клик для перехода в настройки режима
  - **Direction** (Направление работы)
    - Min-Max
    - Max-Min
  - **Limits** (Пределы поворота)
    - Min
    - Max

#### **5. Drive** (Канал привода)

- **Mode** (Режим) - клик для перехода в настройки режима
- **Direction** (Направление работы)
  - Open-Close
  - Close-Open
- **Timeout** (Время движения)

### Настройка режима, уровень вложенность и 1

1. **Timer** (Простой периодичный таймер)
  - **Period** (Время паузы, оно же период работы)
  - **Work** (Время работы)
  - **Left** (Осталось до следующего включения)
2. **Timer RTC** (Периодичный таймер с привязкой ко времени)
  - **Period** (Период работы)
  - **Work** (Время работы)
  - **Start hour** (Час, начиная с которого считается период)
3. **Week** (Суточный таймер)

- Days (дни)
- On (Час начала работы)
- Off (Час окончания работы)
- Global (Глобальный флаг)

#### 4. Sensor (Датчик)

- Period (Период опроса)
- Sensor (Выбор датчика)
- minV / maxV (Минимум и максимум значения с датчика, т.е. гистерезис)

#### 5. PID (ПИД регулятор)

- P/I/D – коэффициенты регулятора
- Sens - sensor (Сенсор)
- Set - setpoint (Установка регулятора)
- T (Время итерации)
- min/max (минимальный и максимальный порог выходного сигнала)

#### 6. Dawn (Рассвет)

- Start (Время начала рассвета)
- Stop (Время начала заката)
- Dur (Продолжительность рассвета и заката)
- min/max (минимальная и максимальная величина управляющего сигнала)

### Описание интерфейса



### Экран отладки (DEBUG)

На экране отладки в реальном времени (период обновления – 1 секунда) отображаются состояния всех компонентов системы: позиции реле, серво и привода, показания всех датчиков, системное время и аптайм (время в сутках с момента последней перезагрузки).

## Экран сервиса (SERVICE)

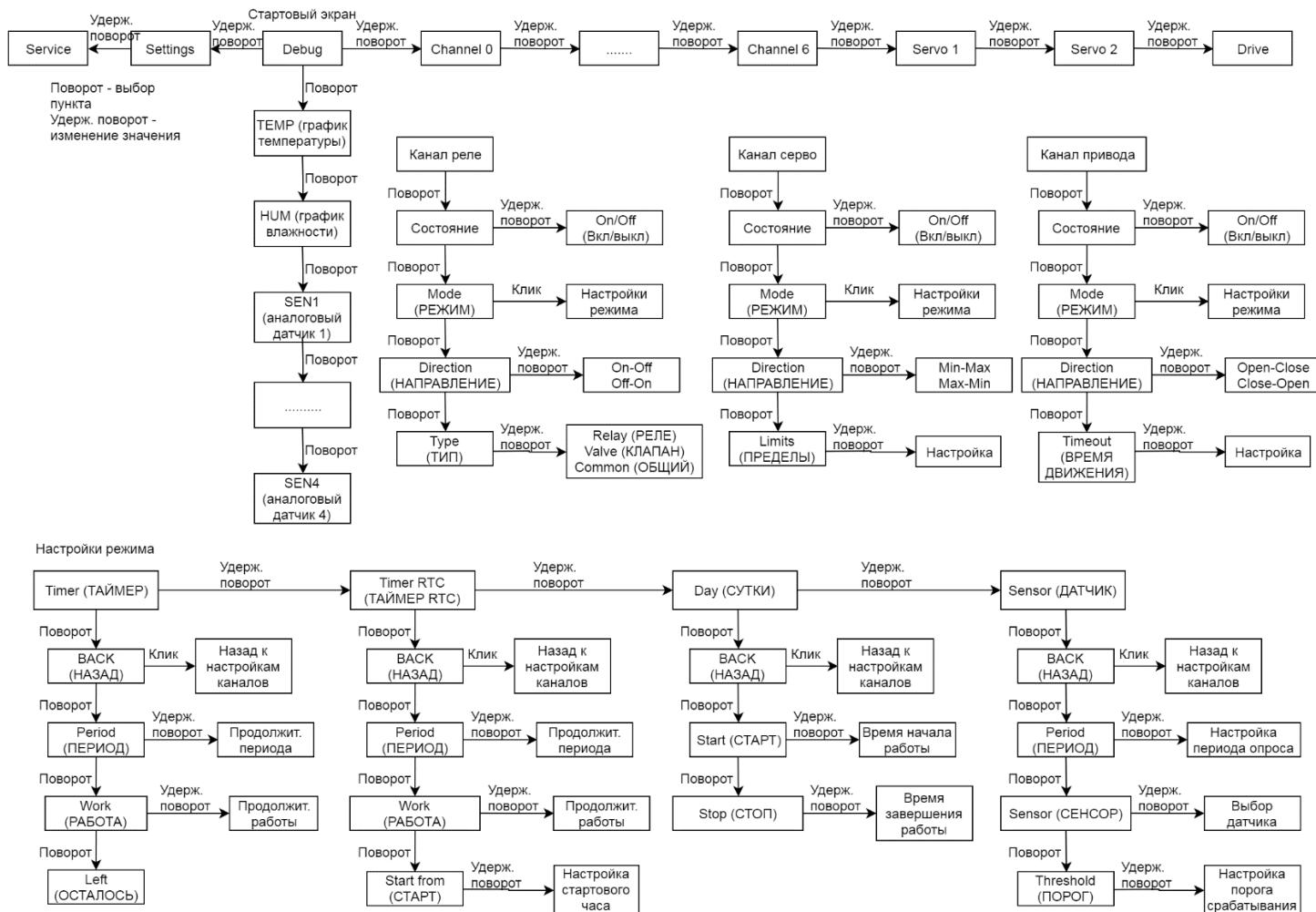
На экране сервиса можно изменить положение любой железки (реле, серво, привод). Новое положение «существует» только при активном экране сервиса, при выходе с экрана сервиса всё железо вернётся на свои места согласно режиму. Пока активен режим сервиса – таймеры остановлены, т.е. действия по таймерам при активном экране сервиса не выполняются. На экране сервиса можно также настроить время системы.

## Экран настроек (SETTINGS)

Находится между отладкой и сервисом, позволяет настроить некоторые параметры контроллера:

- **A-BKL** – вкл/выкл, автоматическое отключение подсветки дисплея
- **BKL-time** – время в секундах, через которое отключится дисплей, если не трогать энкодер
- **Drv** – 0-255, скорость линейного привода
- **Day / Month / Year** – день, месяц и год
- **Sens prd** – секунд, главный период опроса датчиков
- **Plot prd** – Min, Hour, Day – выбор отображения графиков (столбик-минута, столбик-час, столбик-сутки)
- **S1 sp** – скорость серво 1 (при активном SERVO\_SMOOTH)
- **S1 acc** – ускорение серво 1 (при активном SERVO\_SMOOTH)
- **S2 sp** – скорость серво 2 (при активном SERVO\_SMOOTH)
- **S2 acc** – ускорение серво 2 (при активном SERVO\_SMOOTH)
- **При активном тюнере ПИД появляется ещё ряд настроек, их рассмотрим в главе «Тюнер ПИД»**

## Структурная схема меню (версия 1.3)



## Выходы и компоненты платы GyverControl

На плате версии 1.2 и выше обязательные перемычки уже соединены!

Если не запаивать перемычку, то Ардуино и датчики питаем от отдельного блока на 5 Вольт вот в эти пины

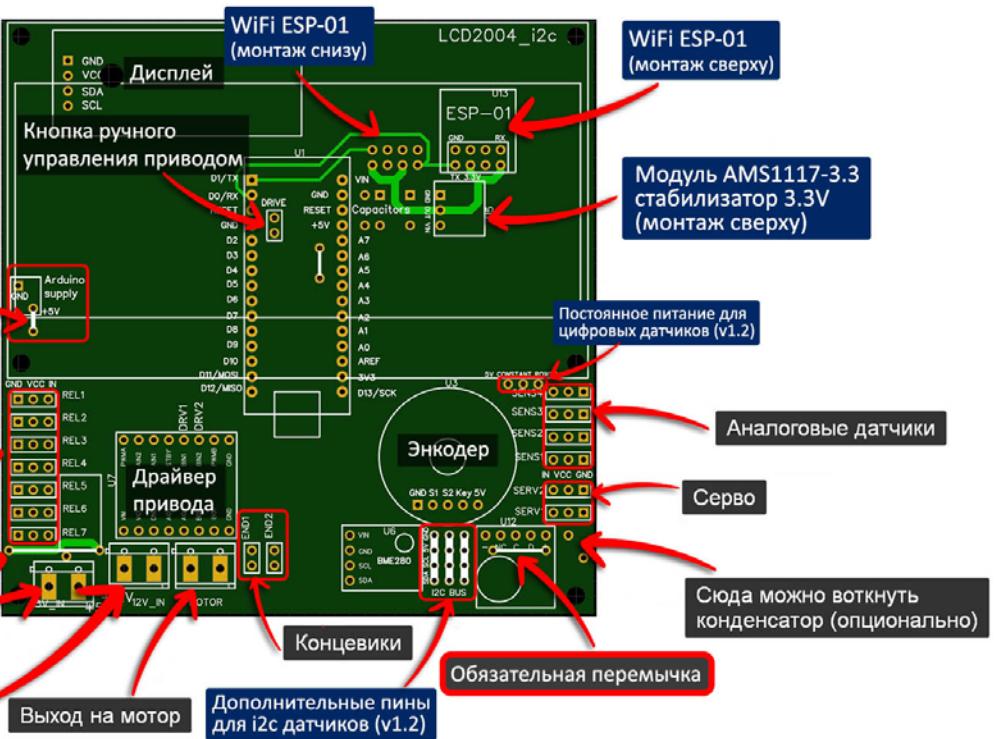
Запаять перемычку:  
Ардуино и датчики будут питаться от источника питания реле и серво (нижняя колодка) что не очень хорошо

Релейные каналы

Обязательная перемычка

Питание реле и серво

Питание мотора



## Распиновка контроллера



## Распиновка, Возможности каналов

**РЕЛЕ** - обычные каналы реле. Доступны режимы Timer, Timer RTC, Week, Sensor

**ШИМ-РЕЛЕ** - реле с низкочастотным ШИМ для режима ПИД регулятор

**ШИМ** - доступны режимы ПИД и РАССВЕТ, выход ШИМ сигнала

**АНАЛОГ** - подключается аналоговый датчик, выводится как 0-255

**ТЕРМ** - поддержка термистора с выводом в градусах (включается в коде)

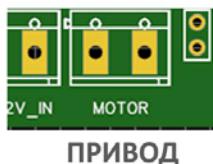
**DHT** - поддержка датчиков темп. и влажн. DHT11/DHT22. Заменяет BME280 (включается в коде)

**DALLAS** - поддержка термометра ds18b20, выводится в градусах вместо SENS1 (включается в коде)

**СЕРВО** - канал сервопривода. Доступны все режимы таймеров, ПИД и РАССВЕТ

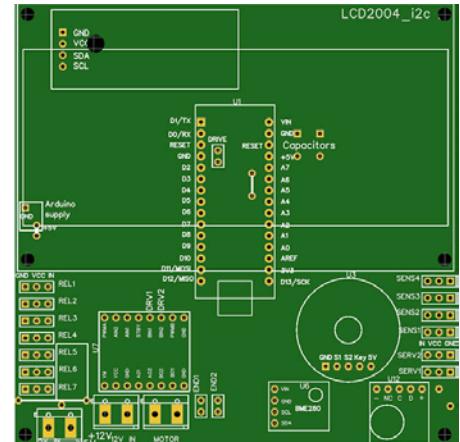
**ПРИВОД** - канал линейного привода. Доступны режимы Timer, Timer RTC, Week, Sensor, ПИД

**CO2** - поддержка датчика углекислого газа MH-Z19



ПРИВОД

ШИМ-РЕЛЕ / РЕЛЕ	SENS4	АНАЛОГ / ТЕРМ
ШИМ-РЕЛЕ / РЕЛЕ	SENS3	АНАЛОГ / ТЕРМ
ШИМ / РЕЛЕ	SENS2	АНАЛОГ / ТЕРМ / DHT / CO2
ШИМ / РЕЛЕ	SENS1	АНАЛОГ / ТЕРМ / DALLAS / CO2
РЕЛЕ	SERV2	РЕЛЕ / СЕРВО
РЕЛЕ	SERV1	РЕЛЕ / СЕРВО
РЕЛЕ	U12	



## Настройка конфигурации системы в коде программы

Перед загрузкой прошивки можно включить/выключить некоторые дополнительные возможности контроллера. Некоторые возможности заменяют друг друга, некоторые дополняют систему. Включение каждой дополнительной возможности расходует память, поэтому контроллер нужно настроить под свои цели, включив нужное и выключив ненужное.

## Подключение специальных датчиков

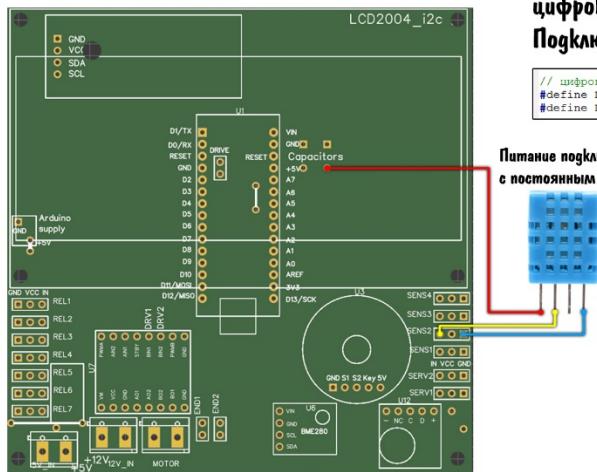
Система имеет поддержку нескольких специальных датчиков

- **BME280** – высокоточный цифровой датчик температуры/влажности
- **DHT11/DHT22** – цифровой датчик температуры/влажности
- **Термистор** – аналоговый датчик температуры
- **DS18b20 (Dallas)** – цифровой датчик температуры
- **MH-Z19** – цифровой датчик углекислого газа

### DHT11/DHT22

Цифровой датчик температуры и влажности. DHT11 имеет очень большие погрешности и не рекомендуется к использованию, DHT22 уже можно поставить. Датчик подключается ко входу SENS2 и **заменяет** датчик BME280, то есть данные по основной температуре и влажности будут браться с него, отображаться в окне Debug и в настройках режима Sensor и PID. Показания датчика на втором канале (SENS2) в окне Debug заменяются на чёрточку. Датчик необходимо питать от постоянных 5V, то есть от любой шины питания на плате, кроме шины питания датчиков (т.к. она имеет непостоянное питание). Для активации датчика нужно включить настройку **DHT\_SENS2**. Тип датчика (11 или 22) указывается в настройке **DHT\_TYPE**

### **Подключение DHT11**



В прошивке 1.4 и выше есть поддержка

цифрового датчика DHT11.

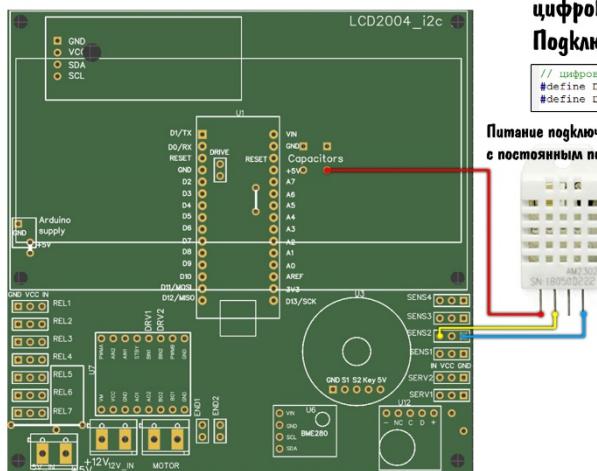
Подключается ко входу SENS2

```
// цифровой датчик температуры и влажности DHT11/DHT22 (вход SENS2)
#define DHT_SENS2 1 // 1 - ко входу SENS2 подключен DHT11/DHT22
#define DHT_TYPE DHT11 // тип DHT датчика: DHT11 или DHT22
```

Питание подключить к pinu  
с постоянным питанием 5V



### **Подключение DHT22**



В прошивке 1.4 и выше есть поддержка

цифрового датчика DHT22.

Подключается ко входу SENS2

```
// цифровой датчик температуры и влажности DHT11/DHT22 (вход SENS2)
#define DHT_SENS2 1 // 1 - ко входу SENS2 подключен DHT11/DHT22
#define DHT_TYPE DHT22 // тип DHT датчика: DHT11 или DHT22
```

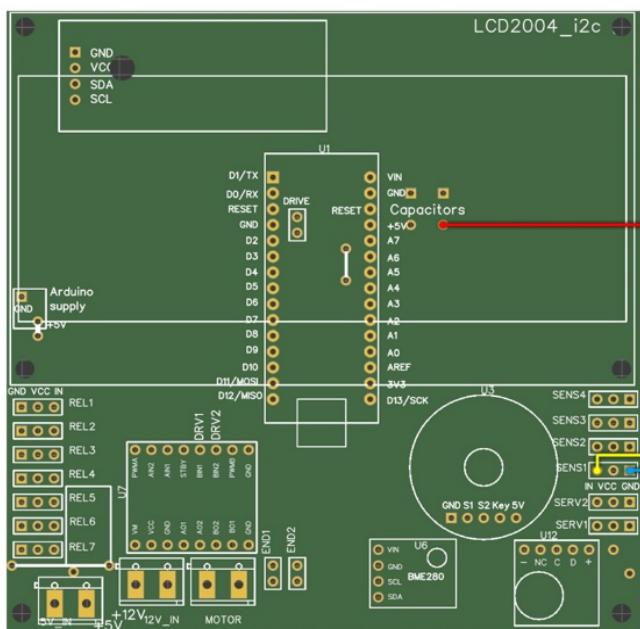
Питание подключить к pinu  
с постоянным питанием 5V



## DS18b20

Цифровой датчик температуры, удобен в своём исполнении в герметичном корпусе. Датчик подключается ко входу SENS1 температура с него выводится как показание датчика SENS1 в градусах по Цельсию. Датчик необходимо питать от постоянных 5V, то есть от любой шины питания на плате, кроме шины питания датчиков (т.к. она имеет непостоянное питание). Для активации датчика нужно включить настройку **DALLAS\_SENS1**

## Подключение ds18b20



**В прошивке 1.3 и выше есть поддержка цифрового температурного датчика ds18b20. Подключается ко входу SENS1.**

```
// цифровой датчик температуры ds18b20
#define DALLAS_SENS1 1 // 1 - ко входу SENS1 подключен ds18b20,
```

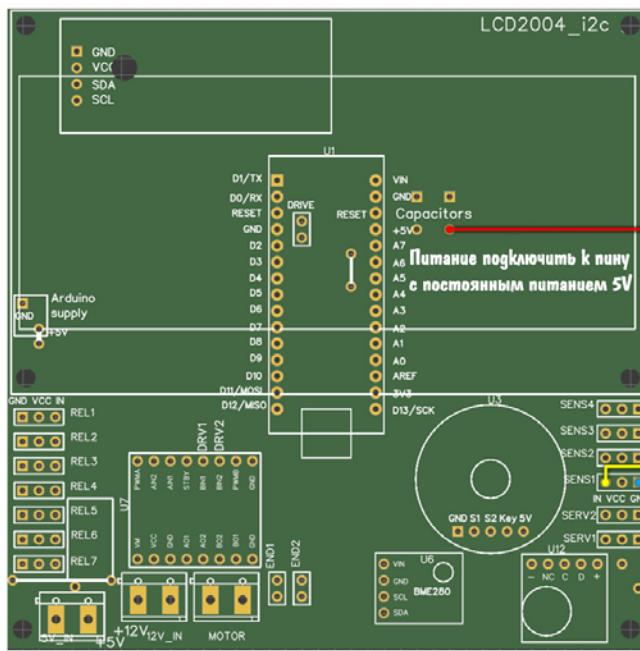
**Питание подключить к pinу с постоянным питанием 5V**

**Резистор 4.7k-10k**



В прошивке выше 1.6 имеется поддержка работы с несколькими (количество ограничено памятью, штук 10 поместится точно) датчиками ds18b20 одновременно, датчики подключаются к **одному и тому же pinу SENS1**.

## Подключение ds18b20



**В прошивке 1.6 и выше есть поддержка нескольких ds18b20. Подключаются ко входу SENS1.**

```
// вход SENS1 - цифровой датчик температуры
#define DALLAS_SENS1 1 // 1 - ко входу SENS1 подключен ds18b20, 0 - подключен обычный аналоговый
#define DALLAS_AMOUNT 3 // количество датчиков на одном pinе

// для нескольких далласов
#define DALLAS_MODE 0 // для нескольких датчиков, считать показания как:
// 0 - среднее из всех
// 1 - наибольшее из всех
// 2 - наименьшее из всех
#define DALLAS_DISP 0 // отображение показаний:
// 0 - согласно режиму (показывает один результат)
// 1 - по очереди выводят показания в одно "окно" SENS1. Номер1, значение1, номер2, значение2...
// адреса датчиков. Количество должно соответствовать DALLAS_AMOUNT
#if (DALLAS_AMOUNT > 1)
const uint8_t dsAddress[] = {
    {0x28, 0xFF, 0x42, 0x5A, 0x51, 0x17, 0x4, 0xD2}, // адрес датчика 1
    {0x28, 0xFF, 0x53, 0xE5, 0x50, 0x17, 0x4, 0xC3}, // адрес датчика 2
    {0x28, 0xFF, 0x99, 0x80, 0x50, 0x17, 0x4, 0x4D}, // адрес датчика 3
    // по аналогии можно добавить еще
};
#endif
```

**Резистор 4.7k-10k**



Каждый датчик имеет уникальный адрес, хранящийся в виде {0x28, 0xFF, 0x42, 0x5A, 0x51, 0x17, 0x4, 0xD2}

Чтобы подключить несколько датчиков ds18b20, нужно сделать следующее:

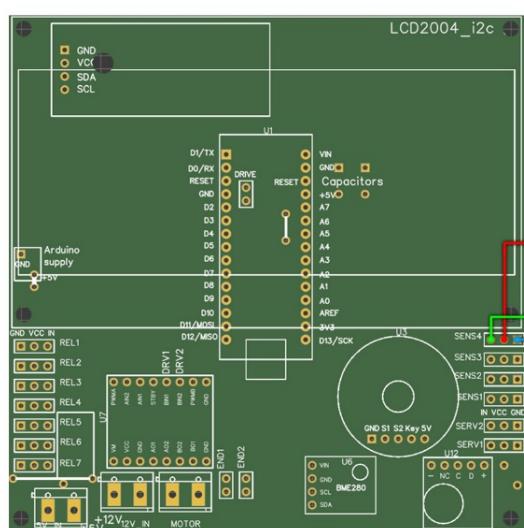
- Загрузить в контроллер прошивку **address\_read.ino**, которая находится в папке *GverControl/firmware/utility*
- Открыть **монитор порта** (иконка лупы в правом верхнем углу программы Arduino IDE)
- В мониторе выставить скорость **9600**, если стоит другая
- Каждые 3 секунды в монитор будет отправляться адрес подключенного датчика
- Подключать датчики **по одному** и копировать куда-нибудь их адреса, при желании пометить-подписать датчики по порядку проверки адреса
- Закрыть **address\_read.ino** и вернуться к прошивке теплицы
- Можно подключить все датчики на линию, как на схеме выше
- Включить настройку **DALLAS\_SENS1 1**
- В **DALLAS\_AMOUNT** поставить количество подключаемых датчиков
- Настроить **DALLAS\_MODE** и **DALLAS\_DISP** по своему усмотрению
- Пролистать чуть ниже и вставить полученные ранее адреса по порядку в массив **dsAddress** по аналогии с тем, как оно там записано. Не напутать с запятыми! Добавить новые строки или удалить лишние, чтобы количество строк с адресами соответствовало количеству подключенных датчиков.

### Термистор

В прошивках версии выше 1.3 добавлена поддержка NTC термисторов, позволяющая выводить показания с термистора в градусах Цельсия, вместо сырого аналогового сигнала. Термисторы можно подключить ко входам SENS1-SENS4, включив соответствующую настройку **THERM1-THERM4**. Температурный коэффициент термистора ( $B$ , beta, бета-коэффициент) настраивается для каждого термистора в настройке **BETA\_COEF1 - BETA\_COEF4**, данный коэффициент всегда указан в описании термистора на странице магазина или в даташите. Термисторы бывают в герметичном корпусе «пуля», что делает их применение более приятным и функциональным. **Важно:** в отличие от цифровых датчиков, на показания термистора могут оказывать влияния наводки от находящихся рядом с проводом индуктивных нагрузок, а также длина провода. Рекомендуется не подключать термисторы проводами длиннее 1 метра, а также скручивать провод в «косичку», таким образом он будет хоть как-то экранирован.

Рекомендуется использовать NTC термисторы с сопротивлением 10 кОм при температуре 25 градусов Цельсия. Температурную «базу» термистора и прочее можно настроить во вкладке sensors в районе строки 75, но при использовании рекомендуемых термисторов ничего менять не нужно.

### **Подключение термистора**



**В прошивке 1.3 и выше есть поддержка термисторов (автоматический перевод в градусы Цельсия). Установите настройки THERM1, THERM2, THERM3 или THERM4 соответственно порту датчиков, к которому подключен термистор (SENS1, SENS2...)**



**Пример: тут термистор подключен к SENS4  
В прошивке будет так:**

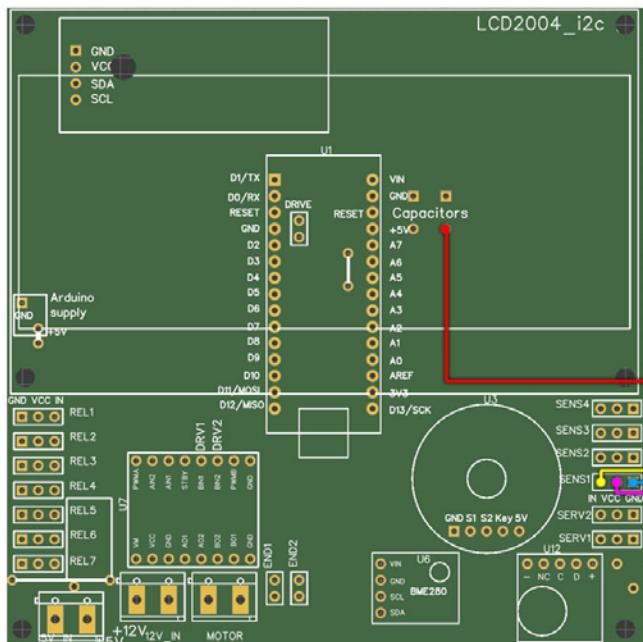
```
// ----- термисторы -----  
#define THERM1 0 // 1 - ко входу SENS1 подключен термистор,  
#define THERM2 0 // 1 - ко входу SENS2 подключен термистор,  
#define THERM3 0 // 1 - ко входу SENS3 подключен термистор,  
#define THERM4 1 // 1 - ко входу SENS4 подключен термистор,
```

## Датчик углекислого газа MH-Z19

В прошивках v1.5 и выше появилась поддержка датчика CO2 MH-Z19, датчик можно подключить к пинам SENS1 или SENS2. В настройках прошивки можно изменить пин подключения (1 или 2), максимальное измеряемое значение (в продаже есть датчики с пределом в 2000 и 5000 ppm).

В датчике по умолчанию включена автоматическая калибровка каждые 24 часа. При использовании датчика в жилом помещении для мониторинга и проветривания калибровку можно оставить, но если датчик стоит в теплице с постоянным высоким уровнем CO2 – автоматическую калибровку нужно отключить (там же, в настройках прошивки), и один раз откалибровать датчик на свежем воздухе, замкнув пины **HD** и **GND** на срок около 20 минут. Можно вывести выключатель на эти пины.

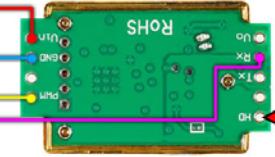
## Подключение MH-Z19



**В прошивке 1.5 и выше есть поддержка датчика CO2 (MH-Z19)**  
**Подключается ко Входу SENS1 или SENS2**

```
// датчик CO2 MH-Z19
#define USE_CO2 0 // 1 - использовать MH-Z19, 0 - не использовать
#define CO2_PIN 1 // 1 или 2, соответственно подключать к SENS_1 или SENS_2 !!!
#define CO2_MAX 5000 // диапазон датчика (обычно 2000 и 5000)
#define CO2_CALIB 0 // 1 - оставить автокалибровку датчика (для жилых комнат)
```

Питание подключить к pinu с постоянным питанием 5V



Для ручной калибровки  
нужно вынести датчик на  
свежий воздух и замкнуть  
HD и GND на 20 минут  
(при включенном питании)

\*pin питания аналоговых датчиков  
используется для отправки команды в  
MH-Z19 на отключение автокалибровки.  
Можно не подключать, если автоматическая  
калибровка Вам НУЖНА.

## ПИД и Рассвет. Схемы

Рассвет работает только на **серво** выходах (идёт управление поворотом серво) и выходах каналов **REL\_3** и **REL\_4**, данные выходы начинают генерировать ШИМ сигнал с частотой 980 Гц.

ПИД работает на каналах **REL\_1** и **REL\_2** при включении опции **USE\_PID\_RELAY**. На этих каналах будет генерироваться низкочастотный ШИМ (по умолчанию 1 Гц), что позволяет спокойно управлять инертными обогревателями переменного тока через твердотельное реле, или постоянного тока – через полевой транзистор или китайский модуль полевого транзистора. На каналах **REL\_3** и **REL\_4** ПИД выводит ШИМ сигнал с частотой 980 Гц. Также ПИД доступен на каналах **серво 1** и **серво 2** (если они активны как серво, а не как реле), и на канале **привода**.

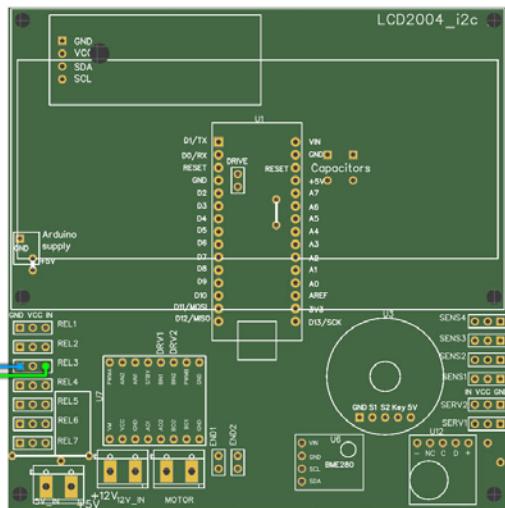
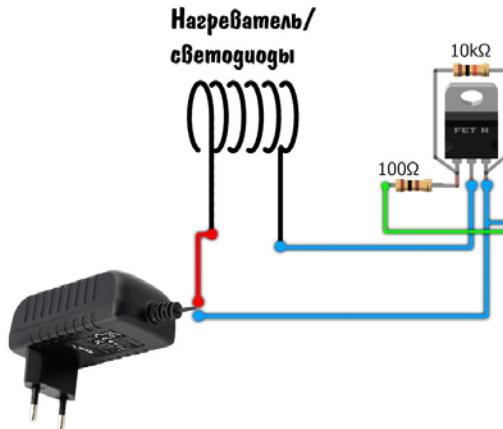
Для управления нагревателями постоянного тока или светодиодными лентами (для рассвета) нужно использовать полевой транзистор (MOSFET). Подключается следующим образом

# Подключение MOSFET транзистора

Для релейного управления - любой канал

**Для плавного (рассвет) - каналы 3 и 4**

**Для плавного (ПиД) - каналы 1-4**

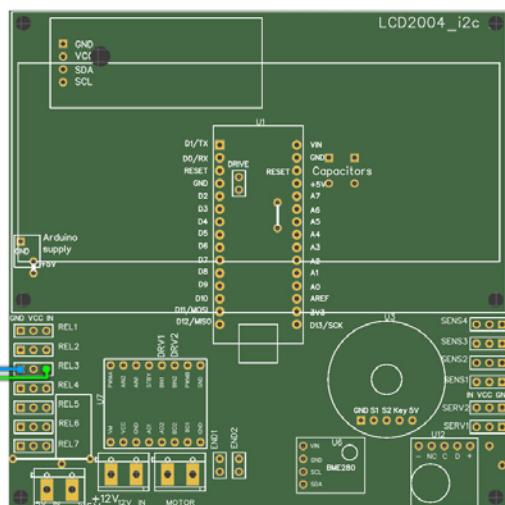
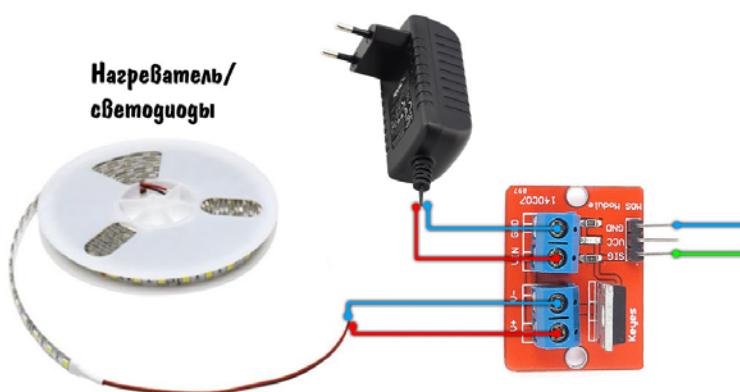


# Подключение модуля MOSFET транзистора

**Для релейного управления - любой канал**

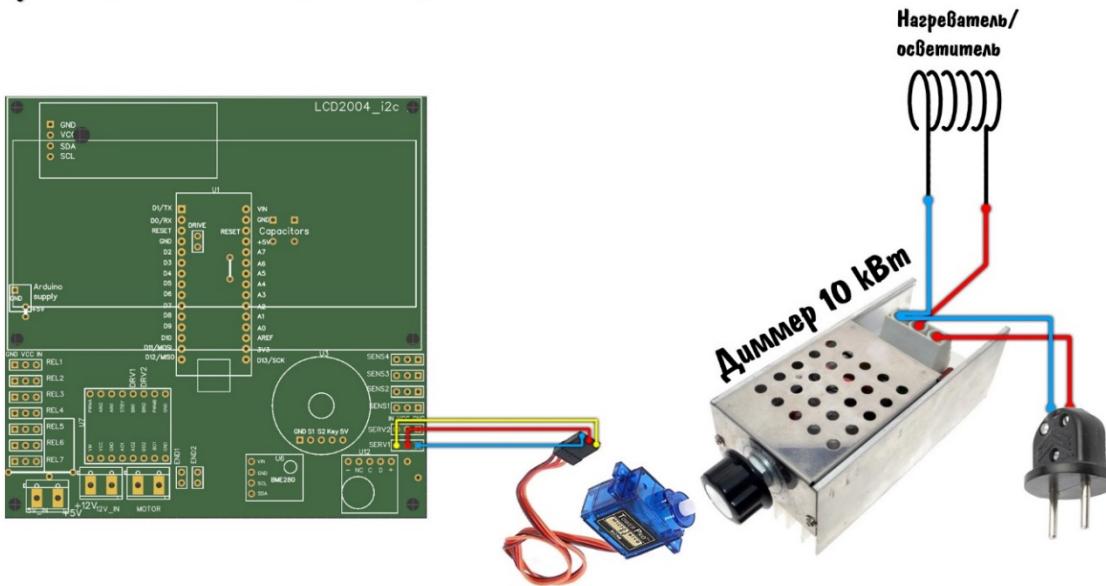
**Для плавного (рассвет) - каналы 3 и 4**

Для плавного (ПИД) - каналы 1-4



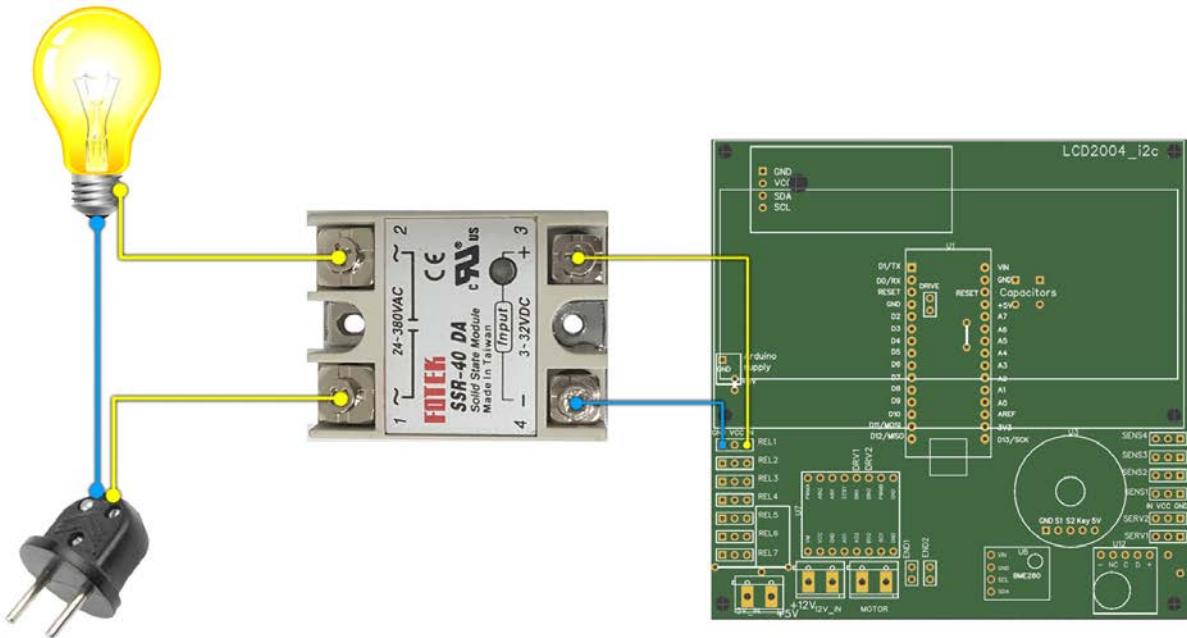
Для управления сервоприводом при помощи ПИД нужно просто подключить серво как обычно. Сервопривод может поворачивать заслонки и крутить рукоятки других приборов, как пример – управление сетевым диммером на 10 кВт (ссылка есть в ссылках проекта).

## Подключение диммера и SERVO



Таким образом можно управлять мощными сетевыми осветителями в режиме Рассвет, или нагревателями для поддержания температуры при помощи ПИД регулятора.

## Подключение твердотельного реле



### Настройка ПИД регулятора

#### Общие понятия

ПИД регулятор позволяет плавным регулированием управляющего устройства (например, нагреватель) добиться точной установки изменяемой величины (температуры). ПИД регулятор имеет для настройки три коэффициента: пропорциональный **P**, интегральный **I**, дифференциальный **D**. Дифференциальный можно не трогать, для медленно протекающих процессов он бесполезен.

Т задаёт период итерации регулирования. Её можно оставить равной 1 секунде, для наших задач это отличный период.

**Sensor** – с какого датчика регулятор будет брать значения. Доступны 6 датчиков (температура + 4 внешних).

**set** – установка, какую величину с выбранного датчика будет стараться удерживать регулятор.

**Min** и **max** – минимальная и максимальная величина сигнала, позволяет ограничить регулятор, если это нужно. Для серво нельзя превышать угол (0-180), для ШИМ каналов - заполнение (0-255). Min и max можно задавать внутри описанных выше диапазонов.

Давайте представим, как работает регулятор: допустим величина с датчика температуры равна 30 градусам, а установка set составляет 42 градуса. Разница между установкой и реальной температурой – 12 градусов. Данная величина умножается на коэффициент P и идёт в управляющий сигнал. Т.е. если P = 1, управляющий сигнал будет 12. Чем больше разница между установкой и значением с датчика – тем больше будет управляющий сигнал. Как только реальная величина станет равной установке – управляющий сигнал станет равен 0, т.е. например нагрев выключится. Интегральная составляющая постоянно суммирует ошибку, умноженную на коэффициент и время, и прибавляет в управляющий сигнал. Таким образом она позволяет удерживать величину на малых отклонениях от заданной. Как настроить регулятор? Можно почитать некоторую теорию здесь <http://lazysmart.ru/osnovy-avtomatiki/nastroika-pid-regulyatora/>

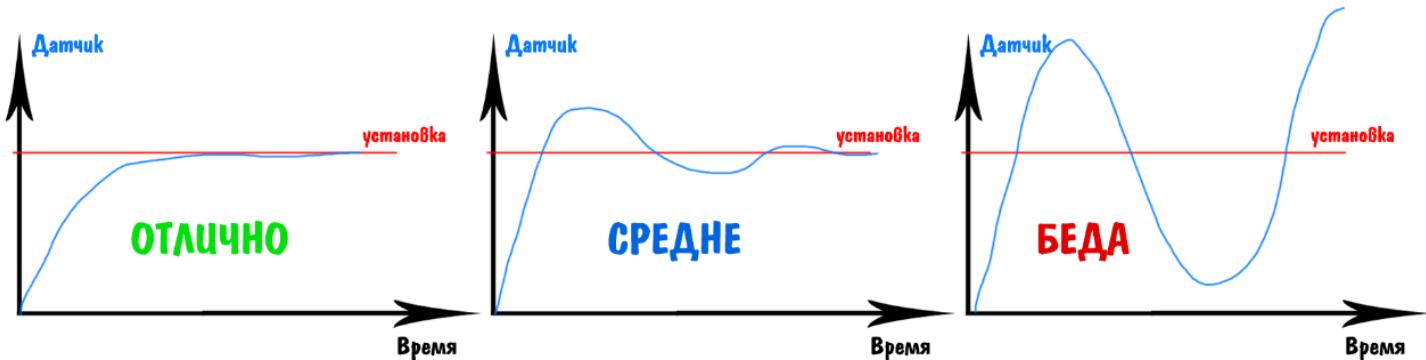
### Ручная настройка

Как ведёт себя система при разных значениях коэффициентов?

**Пропорциональная составляющая**, коэффициент P отвечает за базовое усиление сигнала, чем он больше – тем больший сигнал будет подаваться с регулятора. Чем больше разница между установкой и значением с датчика – тем больше будет управляющий сигнал. Начать ручную настройку можно с плавного увеличения P, значение с датчика будет плавно подбираться к установке, но никогда его не достигнет, т.к. при равенстве управляющий сигнал обнулится, и система снова «остынет». При слишком больших значениях P система потеряет стабильность и начнёт «раскачиваться», поэтому нужно подобрать такой P, при котором значение с датчика уверенно подберётся к установке и останется чуть «ниже», например, на пару градусов. Выведением ошибки в ноль будет заниматься интегральная составляющая. В моих экспериментах коэффициент P получался в диапазоне 12-25.

**Интегральная составляющая** суммирует ошибку в процессе работы и позволяет регулятору установить такой управляющий сигнал, при котором значение с датчика будет максимально близко к установленному. Начинаем аккуратно прибавлять I, в моих экспериментах I обычно получался в районе 0.1-0.7. Слишком большой I может опять же привести к раскачке системы. Итак, допустим значение с датчика у нас установилось равным установке, т.е. регулятор отлично делает свою работу и удерживает значение. Рекомендую отключить систему, чтобы всё «остыло», если это, например, обогреватель. И включить её снова. Таким образом мы проверим, сможет ли регулятор отработать «холодный пуск», например в инкубаторе это важно, может быть перегрев. Тут возможны три варианта:

- Регулятор плавно приведёт систему к нужному значению без перерегулирования
- Система придёт к установке с небольшим забросом вверх, совершил пару затухающих колебаний и застабилизируется на установке
- Система раскачается, всё зашкалит и сгорит



В первом случае всё отлично! Это идеальный переходный процесс. Во втором случае стоит чуть уменьшить I, возможно P тоже надо будет покрутить вниз. Также можно увеличить T (период интегрирования), если регулятор управляет медленной системой. В третьем случае явный перегруз.

Проблемой может стать интегральная составляющая: если процесс очень инерционный, то за время разогрева интегральная составляющая уйдёт настолько далеко, что вернуть её уже не получится и система раскачается (третий график). Поэтому в основных настройках программы (на главной вкладке) также есть настройки **PID\_INT\_MIN** и **PID\_INT\_MAX**, отвечающие за ограничение диапазона интегральной составляющей регулятора. Данная настройка работает сразу для всех каналов! Ограничение интегральной составляющей позволяет упростить настройку регулятора.

**Дифференциальная составляющая** позволяет системе быстро реагировать на изменение сигнала с датчика, именно на его изменение по времени. Для медленных инерционных систем (обогреватель в комнате, инкубатор) можно вообще не трогать коэффициент D и оставить его равным 0.

**Период опроса датчика**, он же период расчёта в настройках называется **T** (секунды). Позволяет замедлить рост интегральной суммы у инерционных систем, по умолчанию стоит 1. Для очень медленных процессов рекомендуется увеличить период расчёта, например для небольшого ящика с обогревателем можно поставить 1-3 секунды, для большой комнаты – 10-20 секунд.

### Отладка ПИД

Где посмотреть красивые графики? В настройках прошивки в разделе ПИД есть опция **DEBUG\_PID**, которая позволяет включить вывод графиков на компьютер. Нужно установить номер ПИД канала, который нужно изучить (1, 2, 3, 4 - обычные каналы, 5, 6 - серво, 7 – привод), прошиться, и открыть плоттер порта (*окно Arduino IDE/Инструменты/Плоттер по последовательному соединению*). И увидите графики: датчика, установки, управляющего сигнала, а также интегральной составляющей. На неё можно не обращать внимания, но иногда может быть полезно видеть, насколько потенциально качается контроллер, хватает ли лимитов **PID\_INT\_MIN** и **PID\_INT\_MAX**, может для каких-то систем придётся их расширить или сузить, и т.д.

*При активной отладке ПИД отключите всё от первого канала! Данные отправляются на компьютер через этот pin, и подключенная туда нагрузка будет активна и всё время дребезжать!*

### Автоматическая настройка ПИД

В версии 1.6 появилась автоматическая калибровка коэффициентов ПИД регулятора. Полученные в результате тюнинга коэффициенты не являются идеальными, возможно придётся чуть покрутить их вручную под свои задачи, т.к. кому-то нужна максимально резкая работа регулятора, а кому-то важнее плавность и отсутствие перерегулирования даже на 0.1 градуса.

Автокалибровка (тюнинг) коэффициентов ПИД регулятора работает следующим образом:

- Система стабилизируется на некотором значении управляющего сигнала
- Контроллер раскачивает систему вокруг положения равновесия
- Анализирует поведение системы и выдаёт +- оптимальные коэффициенты

Для включения тюнера активируем опцию **PID\_AUTOTUNE** в настройках прошивки.

- В окне **SETTINGS** появятся дополнительные настройки ниже основных
- Между **DEBUG** и первым графиком (*поворот энкодера без нажатия на окне DEBUG*) появится окно отладки тюнера
- Включение поддержки ПИД (**USE\_PID**) в настройках программы не обязательно! Тюнер работает без него.

Алгоритм настройки тюнера такой:

- Выбрать в настройках тюнера **channel** исследуемый канал ПИД (можно выбрать 1, 2, 3, 4, s1, s2, dr)
- Выбрать **sensor** датчик, на который влияет выбранный канал ПИД (один из 6 датчиков)

- Нужно узнать, на каком примерно значении управляющего сигнала достигается нужная величина с датчика. Для этого можно в ручном режиме покрутить управляющий сигнал, включив в настройках тюнера режим **manual**.
- После его активации настройкой **steady** задаётся напрямую управляющий сигнал (0-255 для ШИМ и 0-180 для серво). Крутим, смотрим графики, нужно найти такое steady, при котором будет почти достигнут искомый сигнал с датчика (например, 40 градусов). *Найти нужно примерно*, то есть хотя бы это сигнал 50 или 150, точно настраивать не нужно!
- Отключить ручной режим в **manual**
- Настроить величину заброса **step**. Величина step зависит от инертности системы и может быть подстроена уже в процессе тюнинга, можно начать с величины 10. Важный момент: **step** не должен быть больше **steady**!
- Система будет искать установившееся значение, строгость поиска задаётся настройкой **window**, например датчики bme280 и ds18b20 являются очень точными и малошумными, и можно смело ставить в районе 0.05-0.1 градус. С термисторами нужно быть помягче, меньше 0.2 градуса я бы неставил.
- Настройка **delay** (секунды) отвечает за период, с которым проверяется изменение значения с датчика. То есть если за delay секунд значение с датчика изменилось меньше, чем на window, система посчитает процесс остановившимся и стабильным.
- Настройка **kick time** отвечает за время первоначальной раскачки системы. Чем инертнее система, тем больше должно быть это время.
- Настройка **period** отвечает за период работы всего тюнера, по умолчанию стоит 1 секунда и в целом её можно не трогать.
- Настройка **result** отвечает за набор коэффициентов, которые будет считать тюнер. Это может быть **PI** или **PID**. Настройка может быть изменена в процессе работы тюнера. Что выбрать? В процессе калибровки можно выписать коэффициенты и для ПИ, и для ПИД, и потом попробовать работу системы и так и так.
- Для запуска процесса калибровки переводим настройку **tuner** в состояние **on**.
  - Для перезапуска тюнинга нужно поставить **tuner** на **off** и потом снова на **on**, калибровка начнётся сначала.

Этапы калибровки:

1. Сразу после запуска тюнер переходит в режим ожидания стабилизации, в окне отладки тюнера будет статус **wait steady**. На канал будет подан управляющий сигнал величиной **steady**
2. Когда значение с датчика начнёт меняться меньше, чем на **window** за **delay** секунд, система перейдет в режим **wait kick** и изменит управляющий сигнал на величину **step**. Начинается первоначальная раскачка системы.
3. После ожидания **kick time** секунд система "качнётся" в противоположную сторону и начнётся анализ. На графике в плottере можно будет видеть гармонические колебания синей линии значения с датчика, а также линии максимума и минимума. Статус тюнера будет меняться между **step up** и **step down** в зависимости от текущего сигнала. Как только период колебаний системы установится, минимум и максимум перестанут сильно меняться, и значения коэффициентов из нужного набора (ПИ или ПИД) можно будет выписать на бумажку и пользоваться!

## Расписание ПИД регулятора

Режим расписания ПИД позволяет настроить автоматическое изменение "установки" по расписанию, например в течение месяца поднимать температуру в инкубаторе или в течение дня держать нужную температуру в террариуме. Для активации ставим в прошивке настройку **SCHEDULE\_NUM** больше 0, цифра будет соответствовать количеству каналов расписания (не больше 2!). При "активации" расписания автоматически включится поддержка ПИД, даже если в настройках **USE\_PID** стоит 0. В конце списка каналов добавятся каналы **SCHEDULE** (англ. расписание), в которых можно настроить:

### **ПИД каналы (Channel):**

- off — расписание отключено
- каналы 1 и 2 (будет работать, если активен режим ШИМ-реле **USE\_PID\_RELAY**)
- каналы 3, 4 (работают всегда, обычный ШИМ)
- **S1** и **S2** — серво 1 и серво 2 (если активированы)
- **Dr** — привод (если активирован)

### **Начало (Start)**

- Отображается в ДЕНЬ.МЕСЯЦ. Если выкрутить в 0 — появится надпись every, и система будет работать по ежедневному расписанию

### **Конец (End)**

- Отображается в ДЕНЬ.МЕСЯЦ

### **Количество (Amount)**

- Количество «точек», на которые будет разбит выбранный период (максимум 15). Если период выбран например с 15.03 по 15.04 (месяц), и количество точек 15, каждая точка будет соответствовать двум дням. У каждой точки можно настроить свою установку.

### **Точка <номер> (Point)**

- Позволяет задать установку в каждой точке

### **Пример: инкубат ор**

Например нам нужно на полтора месяца, начиная с завтрашнего дня, сделать плавное падение температуры с 37.8 до 36.5 (или как там у яичишек). Температура поддерживается обогревателем, подключенным через твердотельное реле на 2 канале в режиме ШИМ-реле. Температуру измеряет бортовой BME280.

В настройках прошивки активируем как минимум **USE\_BME 1**, **SCHEDULE\_NUM 1**, **USE\_PID\_RELAY 1**, **USE\_PID 1**. Подразумевается, что ПИД регулятор у вас уже настроен (мы тестировали инкубатор [Кирилла Кожевникова](#), температура поддерживается с точностью до 0.05 градусов).

Переходим в настройки канала 2, режим работы выбираем ПИД, в качестве сенсора выбираем **AirT** (температура воздуха, показания с BME280). Далее возвращаемся к списку каналов и листаем дальше, за Drive. Там будет **SCHEDULE 1**. Выбираем **Channel** канал 2. **Start** 15.03 (15 марта). **End** 30.04 (30 апреля). **Amount** делаем например 10 точек, изменение будет каждые 45/10~4.5 дня. И дальше задаём наши точки как

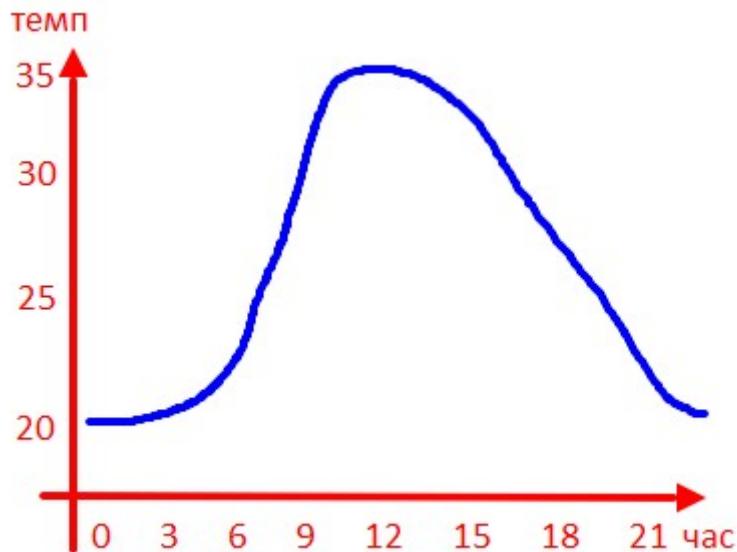
- Point 1: 37.8
- Point 2: 37.7
- Point 3: 37.6
- Point 4: 37.5
- Point 5: 37.4
- Point 6: 37.3
- Point 7: 37.2
- Point 8: 36.9
- Point 9: 36.7
- Point 10: 36.5

Всё! Теперь регулятор будет поддерживать температуру по заданной таблице. До начала периода температура будет удерживаться с Point 1, а с конца — как Point 10.

*Точно так же можно управлять влажностью и любым другим параметром, который измеряет датчик!*

## Пример: террариум

Допустим мы хотим поддерживать температуру в террариуме как в пустыне примерно по следующему графику:



Снаружи террариума у нас должно быть 20 градусов или меньше, потому что в наличии только обогреватель! Температуру измеряет опять же BME280, обогреватель — сетевой тен, который управляется при помощи диммера и сервопривода номер 1 (почему нет?).

В настройках прошивки активируем как **минимум USE\_BME 1, SCHEDULE\_NUM 1, USE\_PID\_RELAY 1, USE\_PID 1, SERVO1\_RELAY 0**. Подразумевается, что ПИД регулятор у вас уже настроен.

Переходим в настройки канала Servo 1, режим работы выбираем ПИД, в качестве сенсора выбираем **AirT** (температура воздуха, показания с BME280). Далее возвращаемся к списку каналов и листаем дальше, за Drive. Там будет **SCHEDULE 1**. Выбираем канал **S1**. **Start** крутим в минимум до появления надписи **every**. **End** автоматически заменится на прочерк. **Amount** делаем например 8 точек, изменение будет каждые  $24/8=3$  часа. И дальше задаём наши точки как

- Point 1: 20
- Point 2: 21
- Point 3: 23
- Point 4: 35
- Point 5: 33
- Point 6: 30
- Point 7: 25
- Point 8: 22

Да, точки начинают отсчёт с 0 часов и отражают сутки. 0-24 часа это Point 1 — point (последняя).

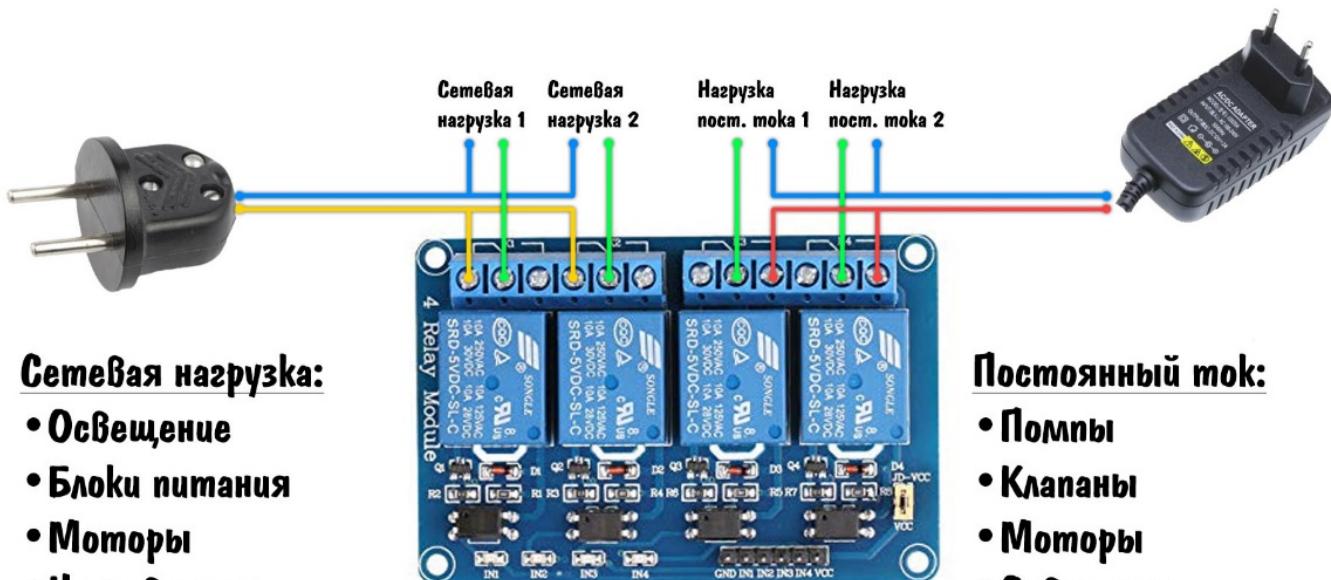
И всё, регулятор будет ежедневно поддерживать заданный температурный режим.

## Советы по подключению реле

Несмотря на простоту и очевидность подключения нагрузки через реле, можно столкнуться с практически «магическими» проблемами, проявляющимися как глюки в системе контроллера, вплоть до зависания и перезагрузки, и неадекватное поведение дисплея. Рассмотрим типичное подключение нагрузки к реле:



# Разные Варианты подключения реле



## Сетевая нагрузка:

- Освещение
- Блоки питания
- Моторы
- Нагреватели

## Постоянный ток:

- Помпы
- Клапаны
- Моторы
- Освещение

Таким образом реле может управлять практически чем угодно, но проблемы возникают именно с **индуктивной нагрузкой**, причём как постоянного, так и переменного тока. При резком включении и отключении индуктивной нагрузки создаётся выброс, напряжение которого может в несколько раз превышать напряжение питания цепи, этот выброс провоцирует электромагнитные наводки в электрических цепях, которые приводят к сбоям в работе микроконтроллера и других компонентов. Индуктивной нагрузкой являются моторы (приводы, помпы) и соленоиды (электромагниты, соленоидные клапаны и проч.). Коммутация такой нагрузки без защиты от выбросов будет приводить к сбоям в работе контроллера, поэтому давайте рассмотрим несколько способов более-менее защиты от таких проблем.

Что почитать по теме:

- <https://habr.com/ru/company/unwds/blog/390601/>
- <https://www.elec.ru/articles/mery-po-zashite-kontaktov-rele-ot-povrezhdeniya-du/>

## Пост оянный ток

Самые жизненные примеры – помпа и клапан на 12V, которые управляются от блока питания. Самый первый и **обязательный** шаг к защите от индуктивных выбросов – диод, установленный встречно-параллельно индуктивной нагрузке. Диод рекомендуется припаивать **как можно ближе к нагрузке**, а не к реле, чтобы между нагрузкой и диодом было как можно меньше проводов. Это рекомендация, совсем необязательно резать провод у помпы под корень и ставить туда диод – можно разместить диод непосредственно у выводов реле, **такой вариант тоже будет работать, но хуже**.

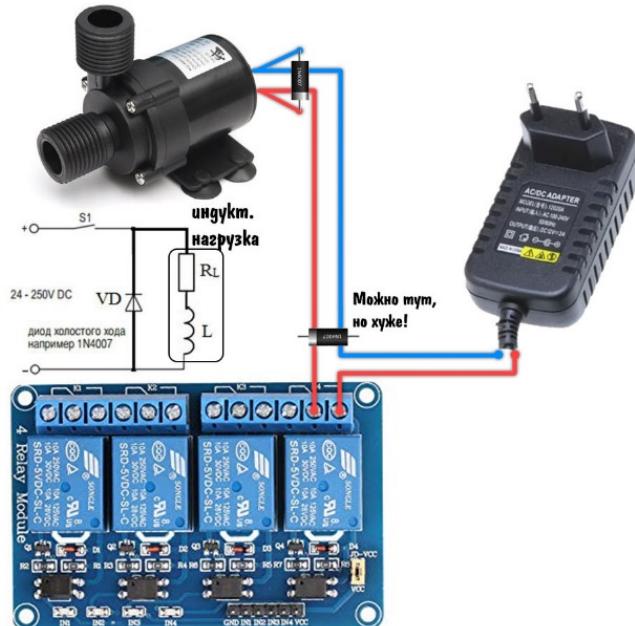


## Защита диодом (пост. ток)

- Рабочий ток и обратное напряжение диода должны быть сравнимы с номинальным напряжением и током нагрузки.

Для нагрузок с рабочим напряжением до 250 VDC и рабочим током до 5 А вполне подходит распространенный кремниевый диод 1N4007

- Выводы диода должны быть как можно короче
- Диод следует припаивать (прищипчивать) непосредственно к индуктивной нагрузке, без длинных соединительных проводов — это улучшает ЭМС при процессах коммутации



### Постоянный и переменный ток

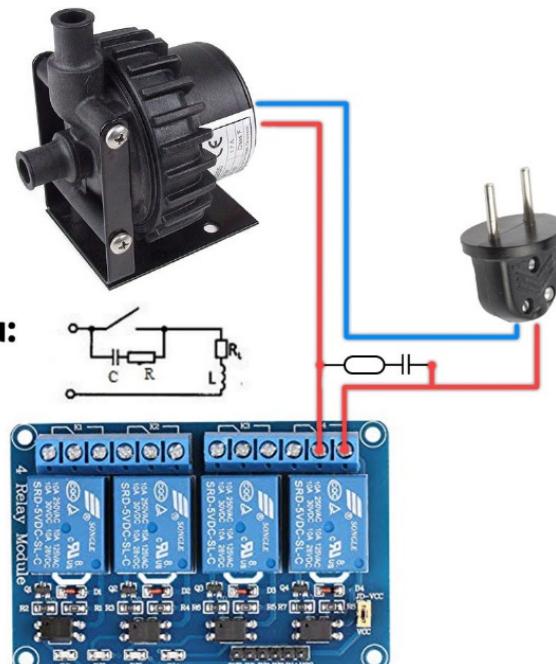
Очень распространённым способом защиты цепи является RC цепь (она же искрогасящая цепь, снаббер), представляющая собой резистор и конденсатор. RC цепь можно поставить параллельно выводам реле (т.е. последовательно с нагрузкой), что очень удобно.

## RC цепь

- Для цепей постоянного тока 12V будет достаточно просто поставить керамический конденсатор ёмкостью 0.1 мкФ параллельно выводам реле
- Для коммутации сетевого напряжения в большинстве случаев подходят номиналы: резистор: 50-100 Ом, 1 Вт  
конденсатор: 0.1 мкФ, 400 В
- Можно пользоваться формулами:

$$C = \frac{I^2}{10}; \quad R = \frac{E_0}{10 \times I \times (1 + \frac{50}{E_0})};$$

где I - ток нагрузки, E<sub>0</sub> - напряжение



### Переменный ток

Для цепей переменного тока есть ещё один совет: используйте твердотельные реле с детектором перехода через ноль (zero detection, zero-cross), они также называются «бесшумные» реле, т.к. в них коммутация происходит в момент перехода напряжения через ноль, и выброс практически равен нулю.

## Проблема с нехваткой памяти

### Проблема

Прошивка занимает очень много места в памяти микроконтроллера и с полным набором возможностей в плату не умещается. **Обновление: версию 1.5 удалось оптимизировать настолько сильно, что проблема практически полностью решилась. Особенно на OptiBoot!** Для экономии памяти можно отключить некоторые функции, сконфигурировав контроллер под себя «на низком уровне».

### Отключение компонентов системы

Версии прошивки начиная с 1.4 имеют «модульность», позволяя включить/выключить ненужные функции и возможности для экономии памяти. Делается это в настройках в самом начале главной вкладки прошивки.

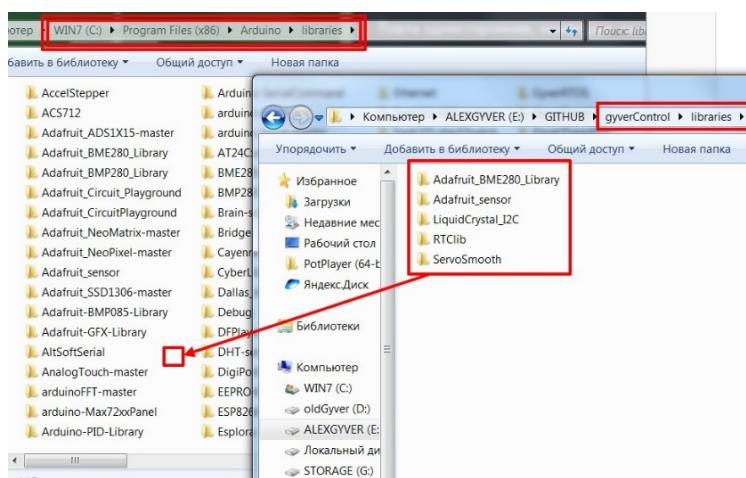
Что отключаем	Название настройки
Поддержка серво и замена их на реле	SERVO1_RELAY и SERVO2_RELAY
Плавность работы серво	SMOOTH_SERVO
Графики	USE_PLOTS
ПИД на каналах 3, 4, оба серво, привод	USE_PID
ПИД на 1 и 2 каналах реле (ШИМ-реле)	USE_PID_RELAY
Рассвет на каналах 3, 4, оба серво	USE_DAWN
Датчик BME280	USE_BME
Датчик DS18b20	DALLAS_SENS1
Датчик DHT11/DHT22	DHT_SENS2
Термисторы	THERM1/ THERM2/ THERM3/ THERM4

### Использование другого загрузчика

Использование загрузчика Optiboot позволяет не только использовать Watchdog, но и добавляет **4%** памяти к доступному лимиту. О прошивке загрузчика читайте ниже.

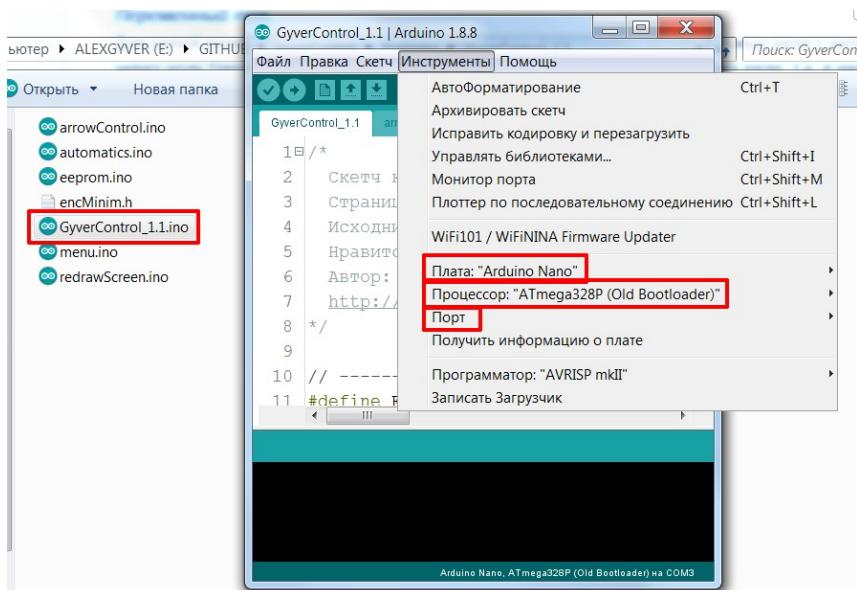
### Ещё раз, инструкция по загрузке прошивки

1. Изучить подробный гайд для новичков (<https://alexgyver.ru/arduino-first/>)
  - a. Установить свежую Java, свежую Arduino IDE и драйвер CH340
2. Извлечь архив с прошивкой, чтобы он стал обычной папкой
3. Скопировать библиотеки из папки проекта **libraries** и вставить в папку **libraries**, находящуюся по пути **C:\Program Files (x86)\Arduino\libraries**



4. Открыть актуальную версию прошивки из папки проекта **firmware**. В папке должны содержаться все остальные файлы, которые были там на момент скачивания архива!
5. Настроить Arduino IDE:
  - a. Плата – **Arduino Nano**

- b. Процессор – **Atmega328P** или **Atmega328P (Old Bootloader)**. Зависит от версии платы, одно из двух должно работать
- c. Порт – порт, к которому подключена плата



6. Нажать загрузить
7. При возникновении ошибки загрузки или ошибки компиляции читать каждую букву в гайде для новичков <https://alexgyver.ru/arduino-first/>

## Watchdog

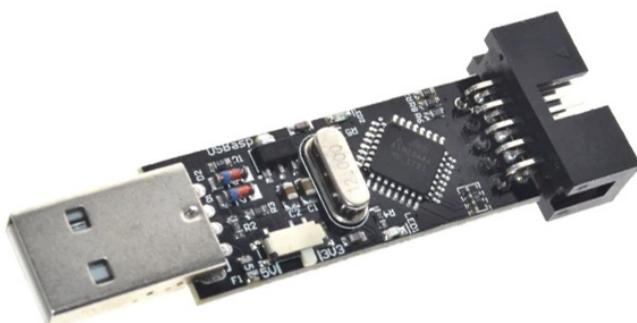
**Watchdog** (англ. Сторожевая собака) – «функция», которая позволяет осуществить автоматическую перезагрузку контроллера в случае зависания. Данная фишка работает только в том случае, если на плату Arduino прошил загрузчик **Optiboot**. Как это сделать читайте ниже.

## Прошивка загрузчика OptiBoot

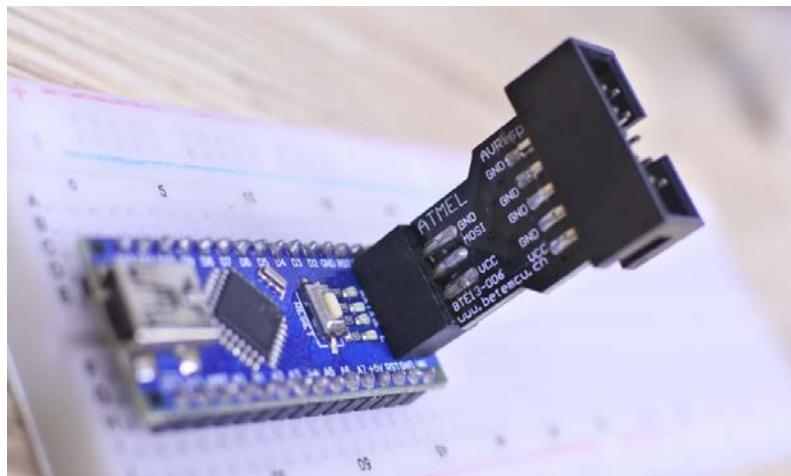
По умолчанию (после покупки) в микроконтроллере стоит старый загрузчик. Новый загрузчик под названием OptiBoot занимает меньше места (освобождает 4% для функций прошивки), а также имеет поддержку Watchdog. Прошить его можно и нужно, в прошивке загрузчика нет ничего страшного, главное точно следовать инструкциям.

### Подключение

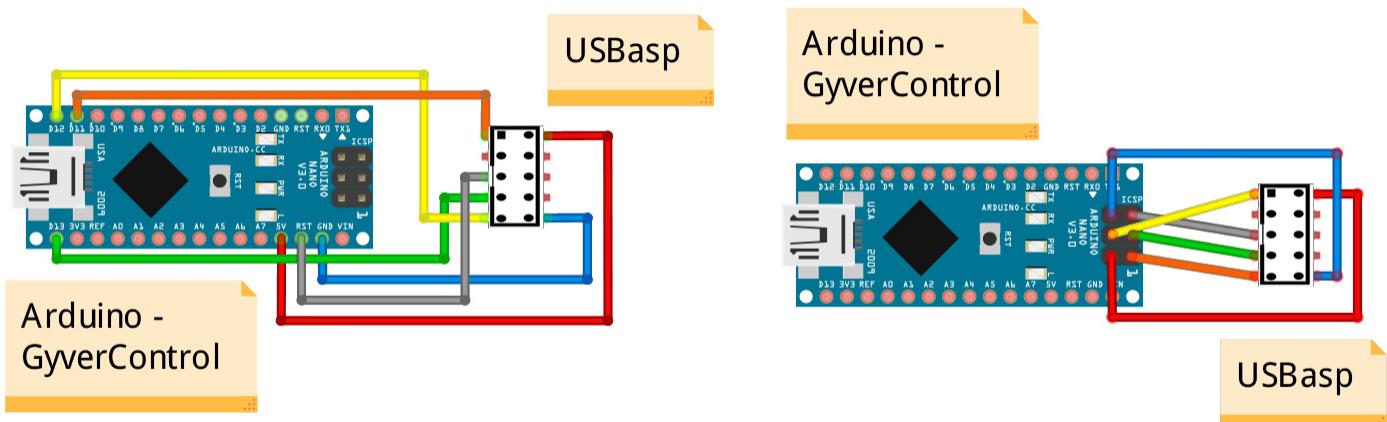
Для прошивки загрузчика вам понадобится либо программатор (USBasp – продаётся на Алике за 80р), либо плата Ардуино (Нано, Уно или другая). Вот так выглядит USBasp с 10-пиновым выходом



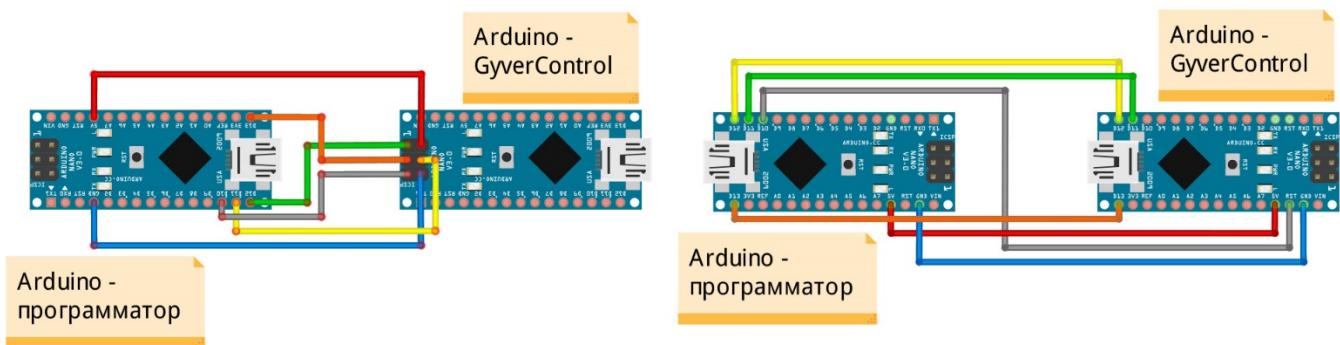
Самый удобный вариант – прошивать через USBasp с переходником на 6-пин



Но нам такой вариант не очень подходит, т.к. пины будут мешать установке дисплея на плату. Вот так подключается 10-пиновый USPasp к плате, два абсолютно равносильных варианта



Если у вас есть ещё одна Arduino, можно сделать её программатором и тоже прошить загрузчик. Подключать так:



Перед подключением нужно прошить Ардуино-программатор специальным скетчем. Читайте ниже.

### Прошивка

### Подготовка:

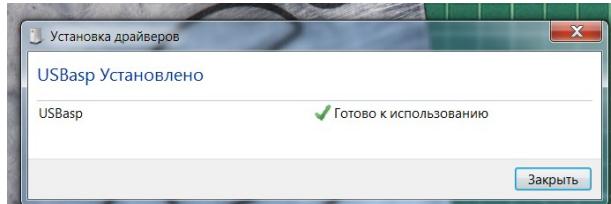
- Запустить **Arduino IDE**
- Файл/Настройки
- Найти окошко «Дополнительные ссылки для менеджера плат»
- Вставить туда [https://github.com/Optiboot/optiboot/releases/download/v8.0/package\\_optiboot\\_optiboot-additional\\_index.json](https://github.com/Optiboot/optiboot/releases/download/v8.0/package_optiboot_optiboot-additional_index.json)
- Нажать **OK**

- Инструменты/Плата/Менеджер плат...
- Поиском найти **Optiboot**, нажать установить
- Переходить к прошивке

## Прошивка при помощи USBasp

Подготовка:

- В архиве проекта найти папку **utility**, в ней архив с драйверами **USBasp\_drivers.zip**
- Распаковать, запустить **InstallDriver.exe**, следовать инструкциям установщика
- Подключить USBasp к компьютеру, дождаться подтверждения «USBasp готово к использованию»



- Отключить USBasp от компьютера
- Подключить USBasp к плате Arduino теплицы
- Подключить USBasp к компьютеру. Arduino теплицы должна запуститься и мигать лампочками
- Инструменты/Плата/Optiboot on 32-pin cpu
- Инструменты/Программатор/USBasp
- Инструменты/Записать загрузчик

После успешной загрузки на плате теплицы будет скетч, который мигает светодиодом (13 пин) по две вспышки каждую секунду. Можно переходить к загрузке прошивки GyverControl.

## Прошивка при помощи Arduino

В дальнейшем будем называть:

- Плата Arduino **ПРОГРАММАТОР** - будет прошивать загрузчик
- Плата Arduino **ПОДОПЫТНАЯ** - на неё прошивается загрузчик (это плата GyverControl)

Действия с платой **ПРОГРАММАТОР**:

- Подключить плату к компьютеру
- Файл/Примеры/11.ArduinoISP
- Инструменты/Плата/Arduino NANO
- Инструменты/Процессор/ATmega328p (Old Bootloader)
- Загрузить прошивку
- Соединяю платы по схеме выше
- Инструменты/Программатор/Arduino as ISP
- Инструменты/Плата/Optiboot on 32-pin cpu
- Инструменты/Записать загрузчик
- Если будет ругаться, перезагрузить обе платы кнопкой RESET на плате
- Отключить платы друг от друга

После успешной загрузки на **ПОДОПЫТНОЙ** плате будет скетч, который мигает светодиодом (13 пин) по две вспышки каждую секунду. Можно переходить к загрузке прошивки GyverControl.

## Загрузка GyverControl

- Подключить плату контроллера теплицы к компьютеру
- Открыть файл прошивки GyverControl

- Инструменты/Плата/Optiboot on 32-pin cpu
- Инструменты/Процессор/ATmega328p
- Так как мы всё это делали ради Watchdog – убедиться, что включена настройка WDT\_ENABLE в настройках прошивки
- Загрузить прошивку

## Изменения в версиях

- **Версия 1.1** – начальная версия
- **Версия 1.2**
  - Оптимизация Flash памяти для дальнейших доработок
    - 5% за счёт упрощения логики работы EEPROM
    - 1% за счёт оптимизации вывода на дисплей
    - 7% за счёт избавления от класса String
  - Добавлен автоматический переход в окно DEBUG по таймеру неактивности
    - В этот же момент настройки **автоматически сохраняются**
  - Исправлена критическая ошибка в построении графиков
  - Добавлена настройка **периода графика** (сутки, час, минута)
  - График меняется в реальном времени
  - К режиму "по сенсору" добавлен **гистерезис**
    - В настройках режима «Sensor» вместо настройки Threshold (как в версии 1.1) теперь две настройки – **minV** и **maxV**. Обе настройки отвечают за пороговое значение с гистерезисом. Логика такая: если величина с датчика больше **maxV** – канал включается, если меньше **minV** – выключается.
  - Добавлен режим навигации "кликнул-изменил-кликнул"
  - Настройки в начале скетча:
    - **SETT\_TIMEOUT** - таймаут неактивности (секунд), после которого автоматически открывается окно DEBUG и сохраняются настройки. Работает при всех активных окнах кроме DEBUG, SERVICE и окон графиков.
    - **CONTROL\_TYPE** - тип управления энкодером
      - 0 - удерживание и поворот для изменения значения
      - 1 - клик для входа в изменение, повторный клик для выхода (стрелочка меняется на галочку)
- **Версия 1.3-1.3.3**
  - Добавлены настройки SERVO1\_RELAY и SERVO2\_RELAY, позволяющие использовать каналы серво как реле
  - Исправлен баг в управлении CONTROL\_TYPE 1
  - Добавлена поддержка датчика температуры **ds18b20** на порту сенсора 1 (SENS1)
  - Добавлена поддержка термисторов на всех портах сенсоров (SENS1-SENS4)
    - Рассчитано на 10 кОм-ные **NTC термисторы**. Коэффициент b можно настроить
  - Поправлено несколько багов с приводом
  - Энкодер теперь работает на МЕГЕ
  - Исправлена критическая ошибка в режиме Timer RTC
- **Версия 1.4**
  - Данные с аналоговых датчиков усредняются средним арифметическим
  - Температура в градусах (с бортового термометра, термистора или далласа) выводится со значком градуса
  - Добавлен режим **ПИД** для каналов реле 3 и 4, обоих каналов серво, канала привода
  - Добавлен режим **РАССВЕТ** для каналов реле 3 и 4 и обоих каналов серво
  - Каналы **ПИД** и рассвета помечены \* (звездочкой)
  - На экране отладки каналы **ПИД** и рассвета выводятся графиком (0-255 ШИМ, 0-180 серво)

- Добавлена настройка, позволяющая убрать из кода датчик BME280
- ~~К режиму Day добавлена настройка Global, задающая время работы для всех режимов~~
- Режим **Day** заменён на режим **Week** - "Неделька"
- При активном **Global** каждый режим получает в название +W для напоминания об активном Global
- В режиме **Timer RTC** появились периоды 1, 2.. 7 суток
- Добавлена поддержка датчиков **DHT11/DHT22**. ВЫВОД ЗНАЧЕНИЙ ВМЕСТО BME280!!!
- Значения датчиков сведены к 8 битам (0 - минимум, 255 - максимум)
- В режиме **Sensor** можно установить максимум меньше минимума, это инвертирует работу режима
- Графики можно отключить, это экономит память
- На экране **Debug** после времени отображается день недели (цифра 1 - 7)
- Ещё больше оптимизации
- И ещё больше оптимизации
- Поправлены баги
- Добавлена поддержка **Watchdog** (работает только на **OptiBoot**)
- **Версия 1.5 была предварительной**
- **Версия 1.6**
  - Облегчённые библиотеки:
  - Для часов используется библиотека microDS3231
    - Для дисплея используется библиотека microLiquidCrystal\_I2C
    - Для ds18b20 используется библиотека microDS18B20
    - Для BME280 используется библиотека GyverBME280
  - Чуть оптимизации кода
  - Исправлен баг с выводом инверсных состояний реле в окне DEBUG
  - Добавлена возможность работы с отрицательными температурами в режиме Sensor
  - Исправлено незапоминание настроек SP и PP в Service
  - ServoSmooth обновлена, работа серво улучшена
  - Исправлен баг с сохранением настроек
  - В ПИД установка переделана на десятичные дроби
  - BME и Dallas выводят температуру в десятичных дробях
  - Шаги настроек изменены на более мелкие
  - Исправлена настройка времени в сервисе
  - Добавлен режим ПИД для каналов 1 и 2 (низкочастотный ШИМ). Каналы помечены \*
  - Для "обратного" режима ПИД нужно ставить отрицательные коэффициенты!
  - В недельке можно выбрать время включения меньше времени выключения
  - Ещё оптимизация памяти
  - Добавлен "быстрый поворот" энкодера: шаг изменения значения увеличивается при быстром вращении
  - Чуть оптимизирован ПИД
  - Значения с точкой в графиках
  - Автоматический масштаб графика
  - Переделана структура меню настроек
  - Добавлена настройка даты в меню настроек
  - Воскресенье теперь цифра 7 (было 0)
  - Исправлена настройка времени в Timer RTC
  - Корректное отображение и работа каналов Servo, работающих как реле
  - Добавлены названия для доп. датчиков, улучшено оформление
  - Период опроса и период графиков перенесены в настройки
  - Добавлена поддержка датчика угл. газа MH-Z19. Обновление дисплея с ним меняется на 1.5 секунды!
  - ПИД и РАССВЕТ переделаны в линейное меню
  - Оптимизированы графики
  - Добавлена возможность отключить плавность серво для облегчения памяти

- Оптимизация памяти и вывода на дисплей
- Исправлен косяк с приводом при выходе из сервиса
- Небольшие изменения в окне сервиса
- Добавлено стартовое меню с сервисом и сбросом (включается по желанию)
- Небольшая оптимизация памяти
- Исправлено отображение реле с низким уровнем в сервисе
- Добавлено расписание для ПИД. Ежедневное и на период в днях
- Исправлен косяк с подсветкой
- Исправлены критические ошибки с серво (пид и рассвет)
- ИСПРАВЛЕНЫ ОШИБКИ С СОХРАНЕНИЕМ НАСТРОЕК
- Сильная оптимизация оперативной памяти
- Графики теперь сохраняются за все периоды (15 дней, 15 часов, 15 минут)
- При выборе датчика значения сразу приравниваются к его показанию (pid, sensor)
- Добавлен режим отладки ПИД (вывод графиков на ПК)
- Небольшая оптимизация рассвета
- Исправлен ВМЕ280 для отрицательных температур (обнови библиотеку)
- Добавлена настройка направления серво для ПИД (на главной вкладке серво, Direction)
- Система теперь знает, что дверь была открыта в ручном режиме и работает корректно
- Добавлено ограничение интегральной составляющей в ПИД
- Редизайн списка настроек
- Добавлен автотюнер ПИД
- Добавлена поддержка нескольких ds18b20 с разными режимами работы
- В режиме Sensor система работает при выставлении одинаковых значений порога
- Исправлена ошибка, связывающая каналы серво и реле
- В ПИД регуляторе для инертных систем есть смысл поднять T, логика работы чуть изменилась
- Установка таймаута привода в десятых секунды
- Улучшена отзывчивость и точность энкодера