

**Руководство
по установке и использованию
программного обеспечения «MRT V.1.2»**



MRT V.1.2

Содержание

1. Назначение и минимальные системные требования.....	3
2. Установка.....	4
3. Работа с программой.....	10
3.1. Создание и редактирование программы.....	12
3.1.1. Визуальный режим.....	13
Добавление действий.....	14
Удаление действий.....	15
Копирование, вставка, перемещение.....	16
Редактирование параметров функций.....	17
Устройства вывода.....	18
Светодиод.....	18
Двигатель постоянного тока.....	19
Серводвигатель.....	21
Зуммер (пищалка).....	23
Экран (временно не используется).....	24
Свой (включение\выключение произвольного порта).....	24
Отладка (отправка данных с платы на компьютер).....	24
Устройства ввода.....	27
Пульт дистанционного управления («ПДУ»).....	30
Кнопка.....	33
Микрофон.....	34
Инфракрасная оптопара («ИК»).....	34
Датчик освещенности («Свет»).....	35
Свой датчик.....	35
Счетчик.....	36
Функции.....	38
Стоп.....	38
Повтор.....	38
Задержка.....	39
Переход.....	40
3.1.2. Текстовый режим.....	41
3.2. Сохранение созданных программ и их открытие.....	42
3.2.1. Сохранение.....	42
3.2.2. Открытие.....	42
3.3. Компилирование и загрузка программы в плату.....	44
3.3.1. Компилирование.....	44
3.3.2. Выбор СОМ-порта.....	46
3.3.3. Загрузка.....	47
4. Удаление программы.....	50

1. Назначение и минимальные системные требования

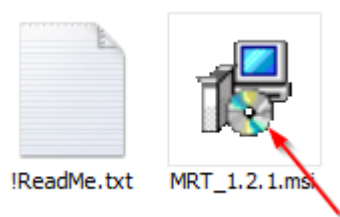
Программное обеспечение MRT 1.2 (далее – «ПО») – комплексный программный продукт, предназначенный для создания, редактирования, компилирования и загрузки исполняемых программ в программируемые платы «Hunagovo», «MRT», «T.O.P», а также для отображения данных, передаваемых с платы по последовательному порту, на экране компьютера.

Минимальные системные требования:

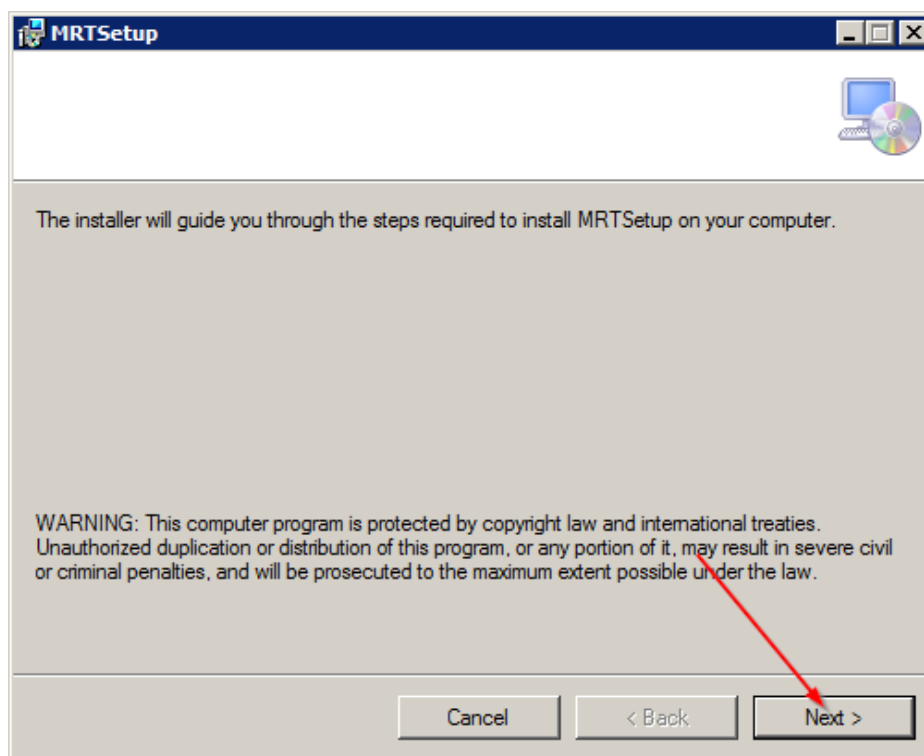
ОС Microsoft Windows XP, Vista, 7 или 8, 32-bit или 64-bit;
200 МБ свободного дискового пространства;
Монитор с разрешением не менее 1024x768;
Свободный порт USB;
Мышь, клавиатура.

2. Установка

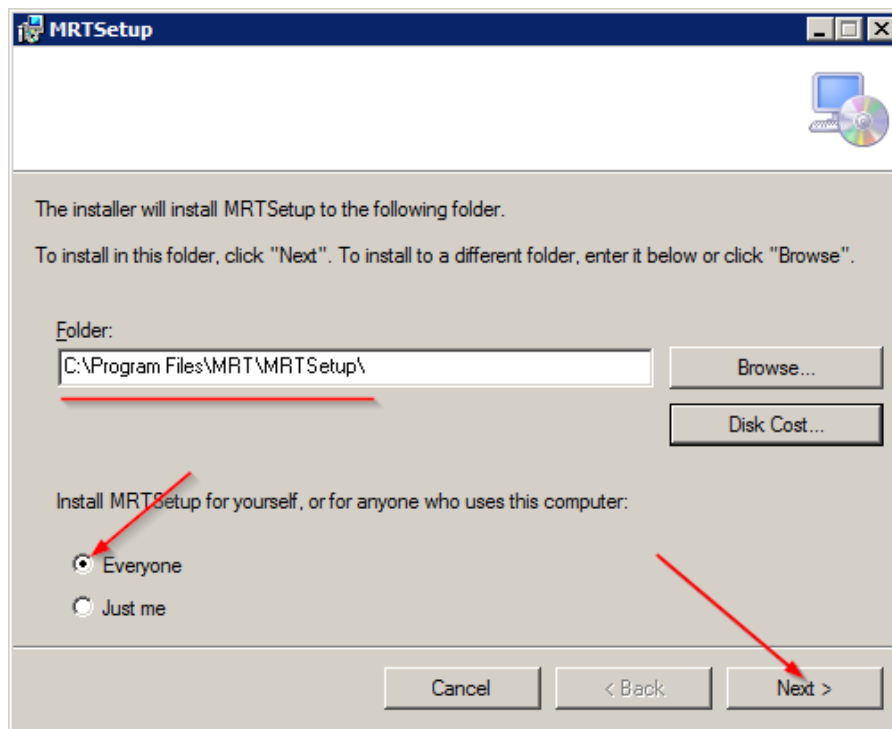
Для начала процесса установки ПО запустите файл «MRT_1.2.x», находящийся в одной папке с данной инструкцией:



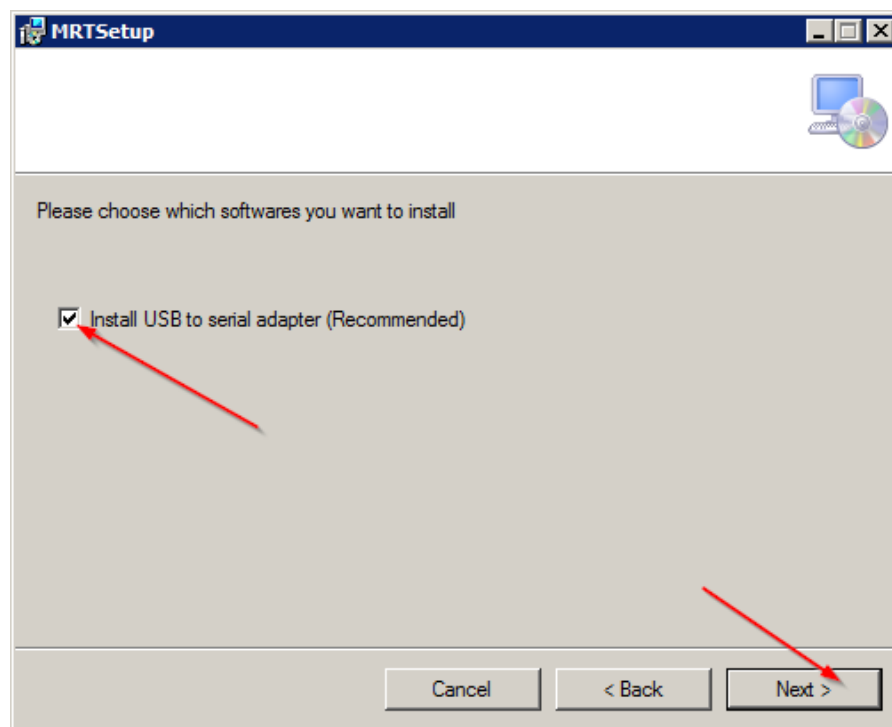
Далее следуйте указаниям программы установки:

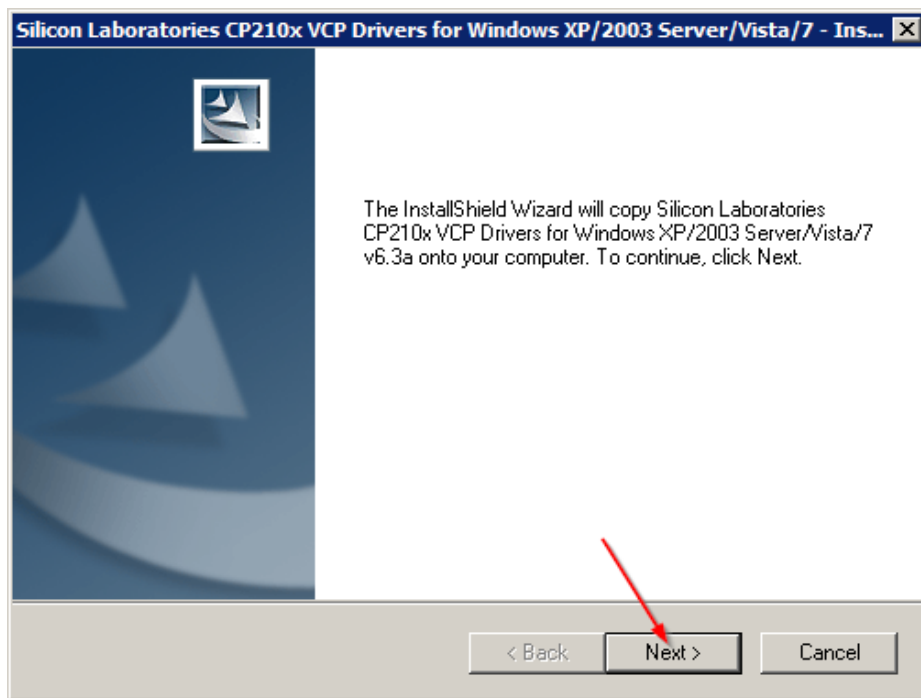
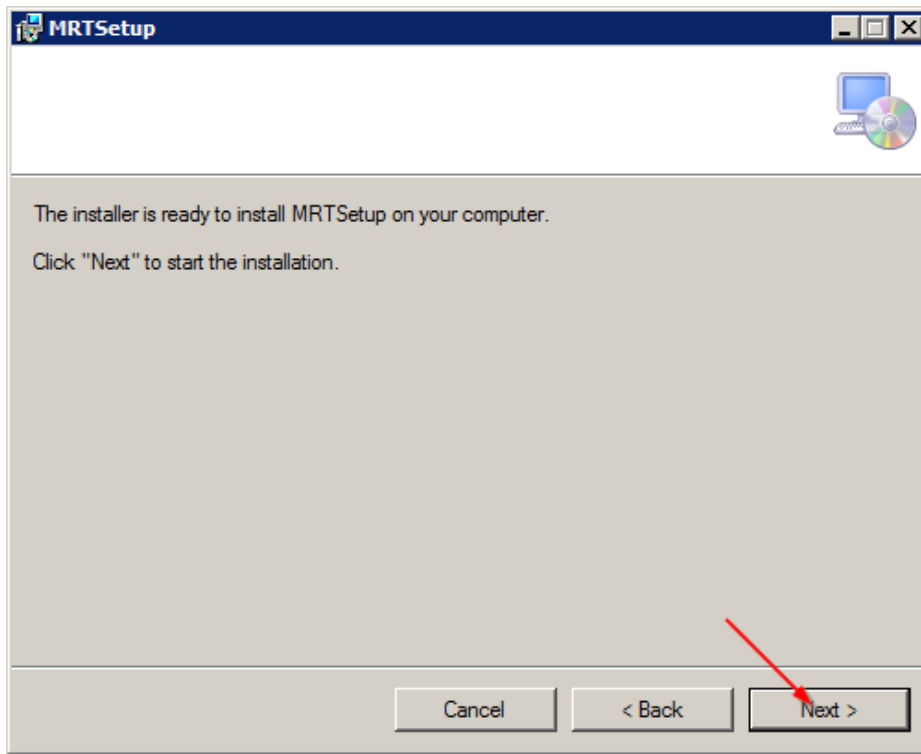


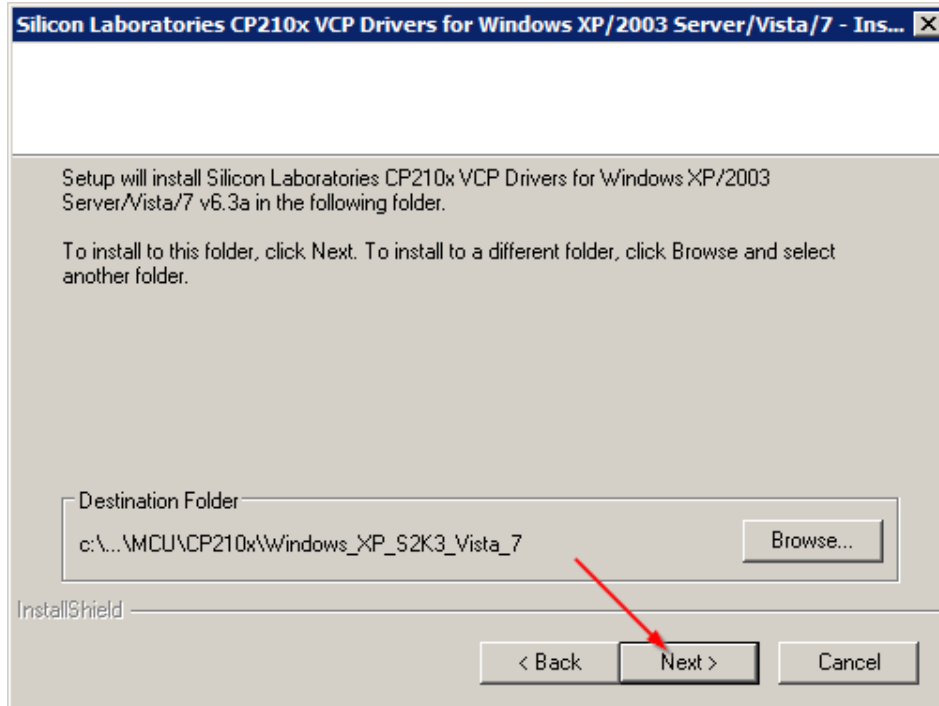
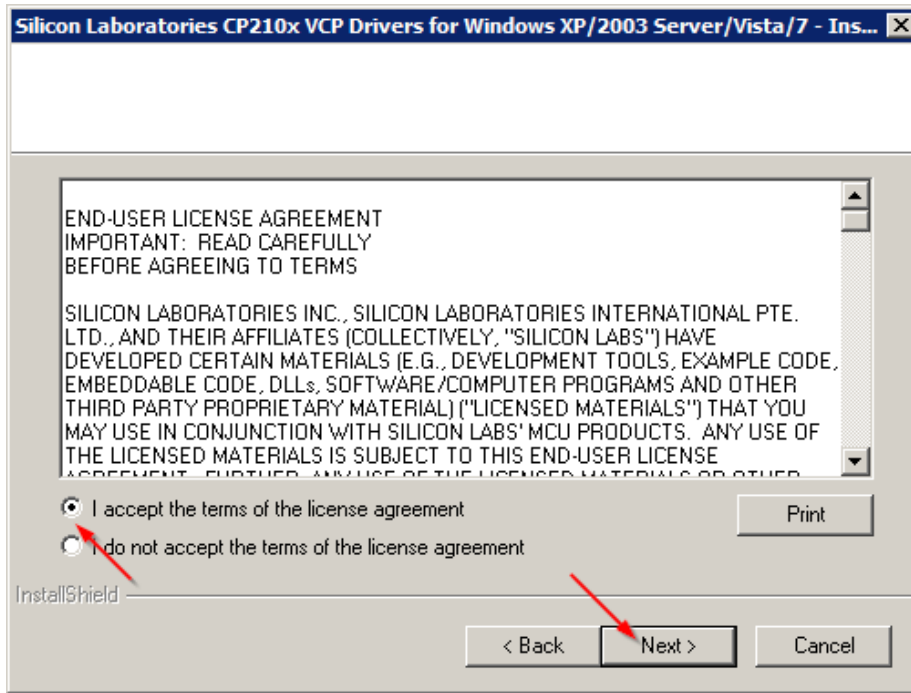
ВАЖНО: Программа должна устанавливаться на диск «С»:

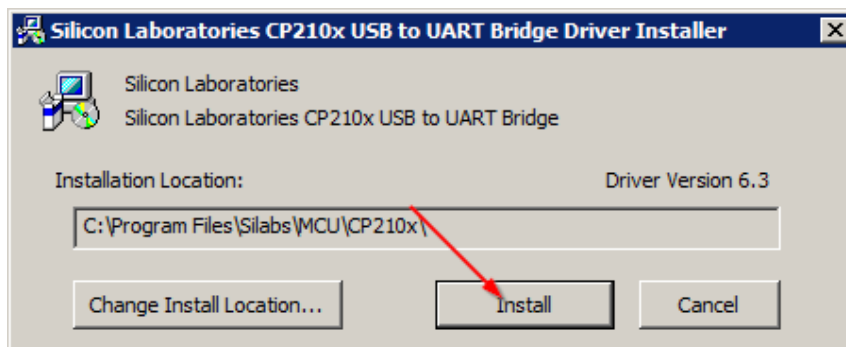
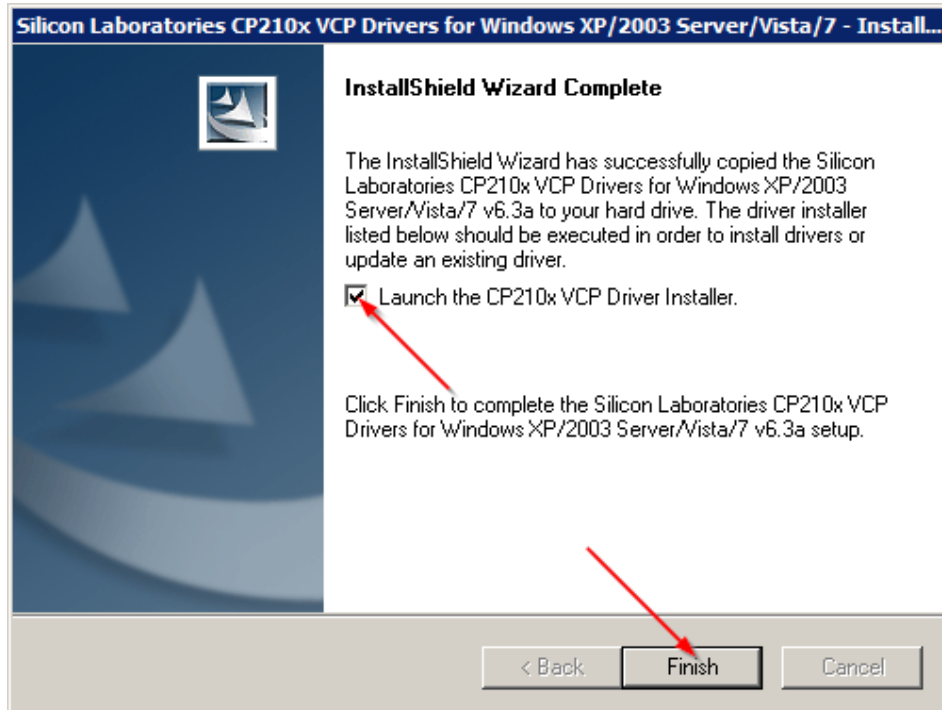
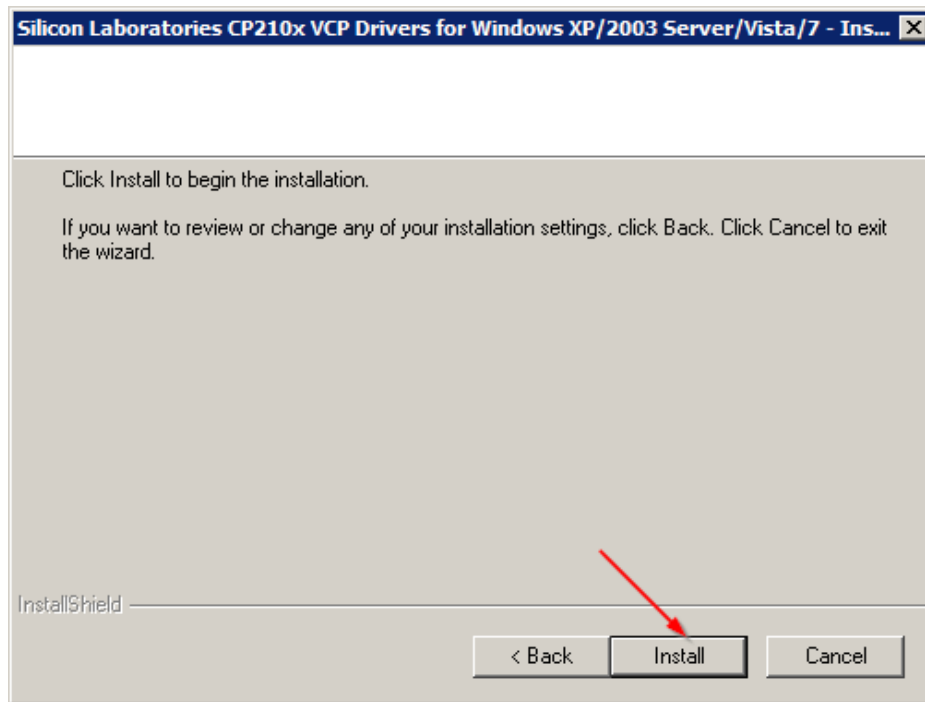


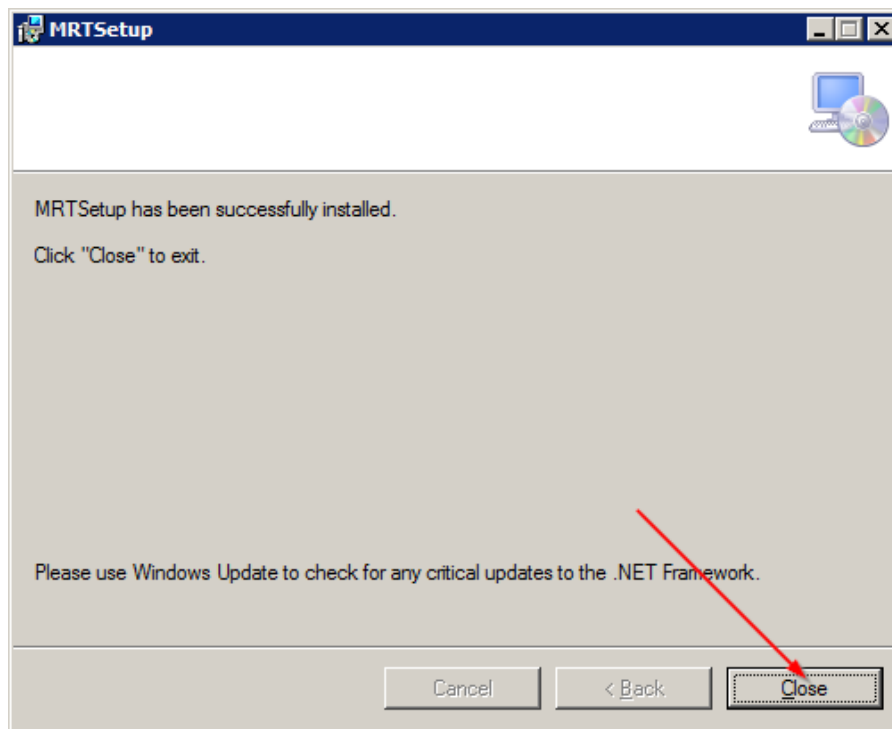
Уберите галочку, если Вы ранее уже устанавливали ПО MRT и в системе уже установлен драйвер загрузчика «Sillabs CP210x VCP drivers»:











По завершении процесса установки на Рабочем столе появится ярлык «MRT 1.2», с помощью которого и следует запускать приложение:



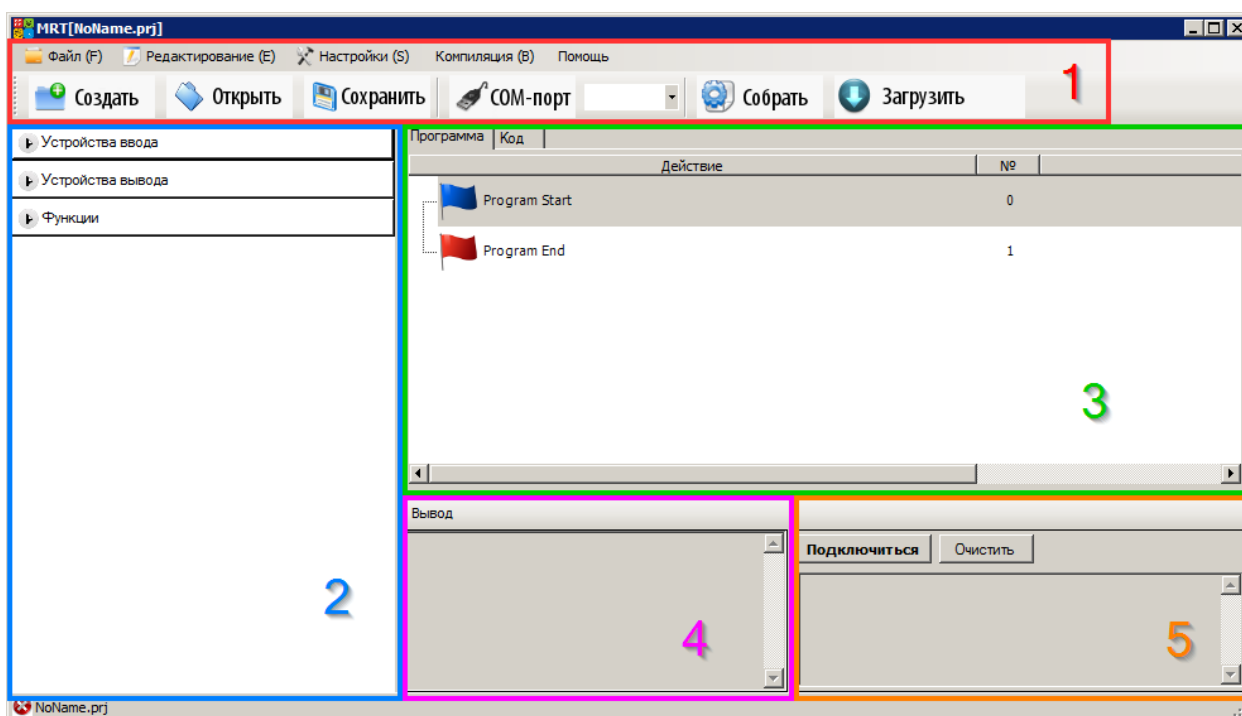
Если при запуске приложения выводится сообщение об устаревшем .NET Framework, необходимо скачать и установить последнюю версию .NET Framework с сайта Microsoft.

3. Работа с программой

Для запуска программы используйте ярлык на Рабочем столе:

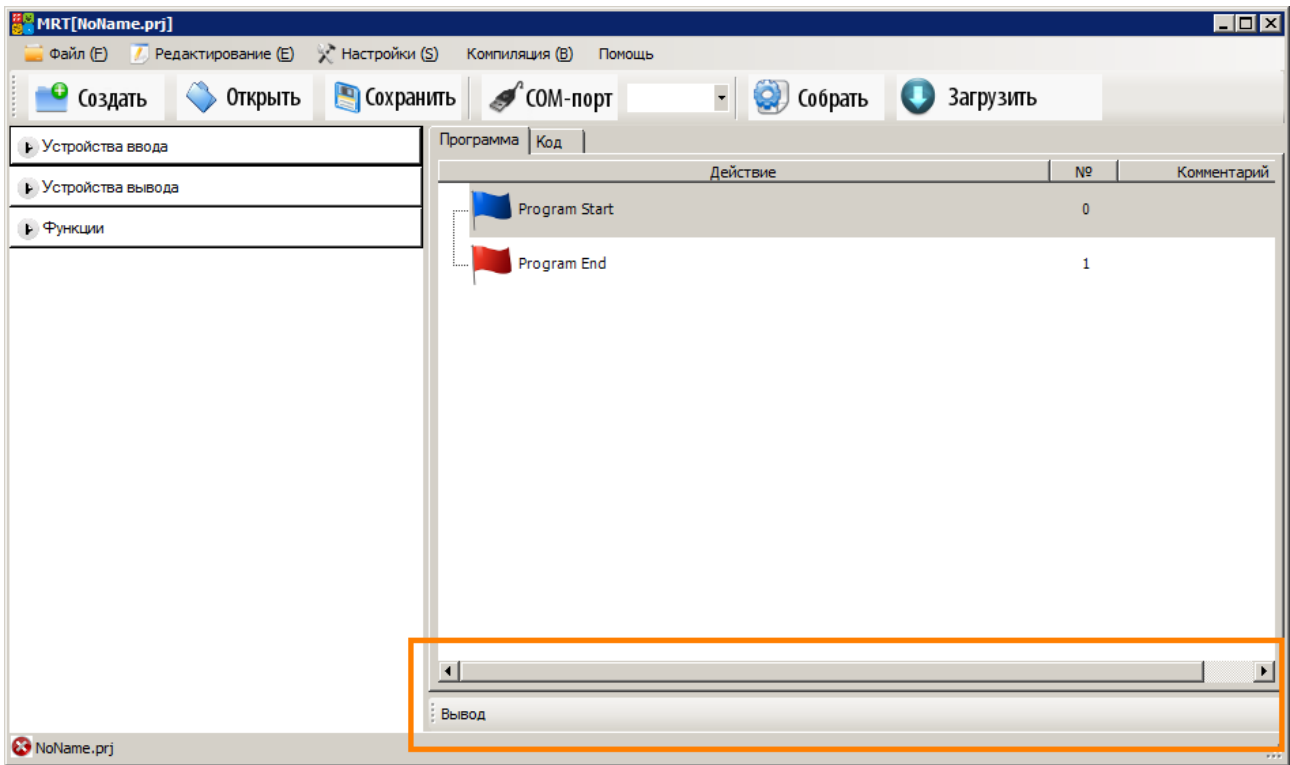


После запуска открывается главное окно программы, состоящее из нескольких рабочих областей:



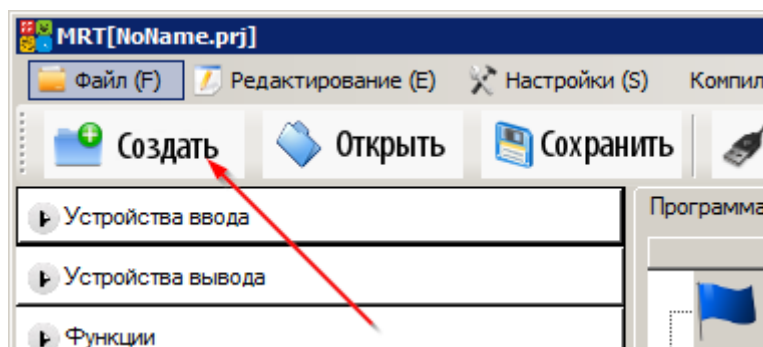
1. Строка меню и часто используемые пункты меню, вынесенные в виде отдельных кнопок;
2. Панель стандартных функций и операторов для составления программы в визуальном режиме;
3. Окно программы, в котором непосредственно отображается создаваемая программа для платы в графическом виде с комментариями (вкладка «Программа») или в виде кода на языке СИ (вкладка «Код»);
4. Окно вывода сообщений компилятора, в котором отображается ход компилирования, а также ошибки и предупреждения, возникшие в процессе компиляции;
5. Окно вывода информации, получаемой с платы при использовании функции «Отладка».

Так как окна 4 и 5 используются достаточно редко, особенно на порах освоения ПО, то панель с ними можно сдвинуть вниз для увеличения области 3:

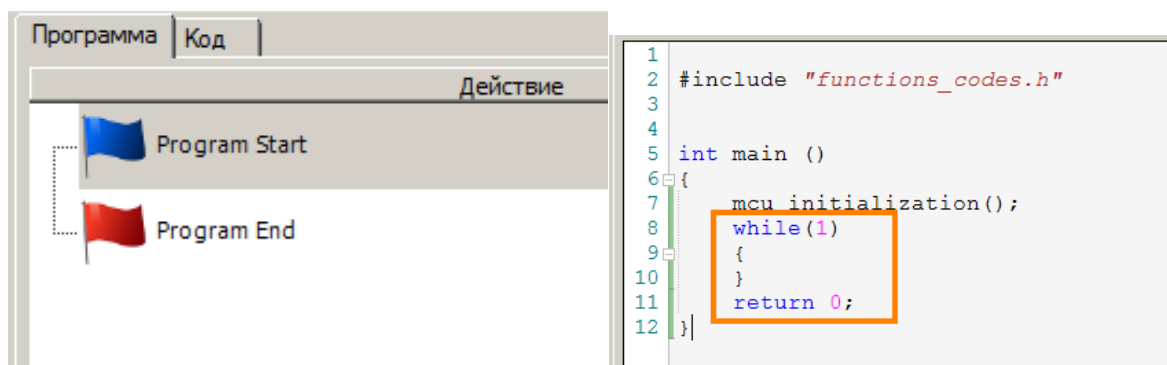


3.1. Создание и редактирование программы

Новая пустая программа создается автоматически при запуске ПО. Для создания новой программы вы можете в любой момент нажать кнопку «Создать» в панели меню:



Пустая программа отображается в виде двух флагов: «Program Start» и «Program End», и пустого бесконечного цикла в коде:




Необходимо помнить, что микроконтроллеры всегда выполняют программу циклически, поэтому после того, как выполниться последняя команда перед флагом «Program End», микроконтроллер перейдет к флагу «Program Start» и начнет выполнять программу с начала, и так бесконечное количество раз, пока включено питание платы. Для принудительной остановки выполнения программы в каком-либо месте используется функция «Стоп».

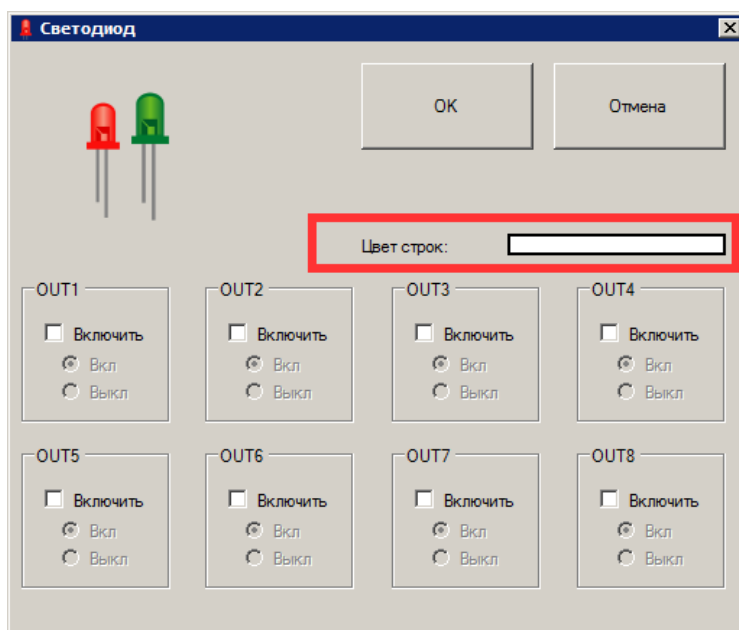
3.1.1. Визуальный режим.

При работе в визуальном режиме программа представляется в виде таблицы идущих друг за другом команд (действий), каждая из которых располагается в своей строке. Номер строки указан в столбце «№». К каждой строке для удобства можно (а начинающим программистам – нужно!) добавлять комментарий. Для этого необходимо один раз кликнуть в поле «Комментарий» необходимой строки и ввести текст комментария:



Действие	№	Комментарий
 Program Start	0	Начало программы
 Program End	1	Конец программы, возврат в начало

Любую строку для удобства можно выделить любым цветом. Для этого во всех окнах редактирования параметров стандартных функций предусмотрено соответствующее поле:



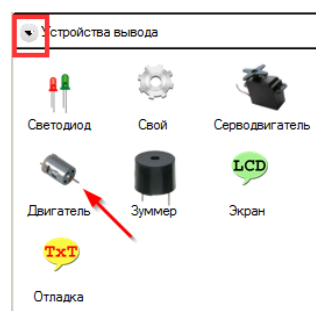
Программа выполняется всегда сверху вниз вдоль серой пунктирной линии. После выполнения последнего действия выполняется первое и так далее:

Программа	Код	Действие	№	Комментарий
		Program Start	0	Начало программы
		LED : [OUT1]=[On]	1	Включение светодиода
		Delay:[1 sec]	2	Задержка 1 с
		LED : [OUT1]=[Off]	3	Выключение светодиода
		Delay:[1 sec]	4	Задержка 1 с
		Program End	5	Конец программы, возврат в начало

Необходимо помнить, что микроконтроллер работает на частоте 16МГц, то есть выполняет 16000 операций в секунду. Т.О. выполнение действия и переход к следующему происходит практически моментально. Именно поэтому в предыдущем примере моргания светодиодом вставлены функции задержки, на которых ход выполнения программы временно останавливается. Без них бы светодиод моргал бы со скоростью, неразличимой для человеческого глаза, и выглядело бы это, как будто светодиод горит вполсилы.

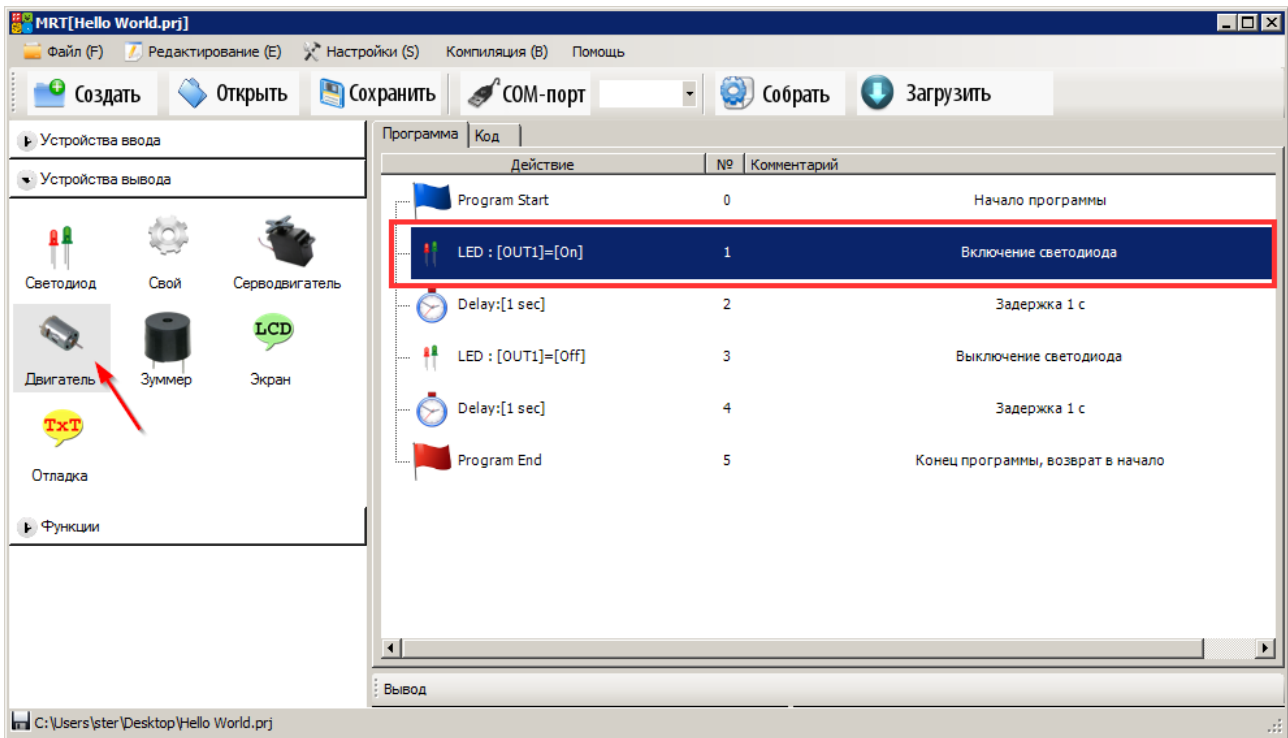
Добавление действий

Для добавления необходимого действия или функции необходимо установить курсор (сделать активной) строку, после которой вы хотите вставить действие, раскрыть необходимую группу устройств и кликнуть на значок устройства, которому вы хотите послать команду:

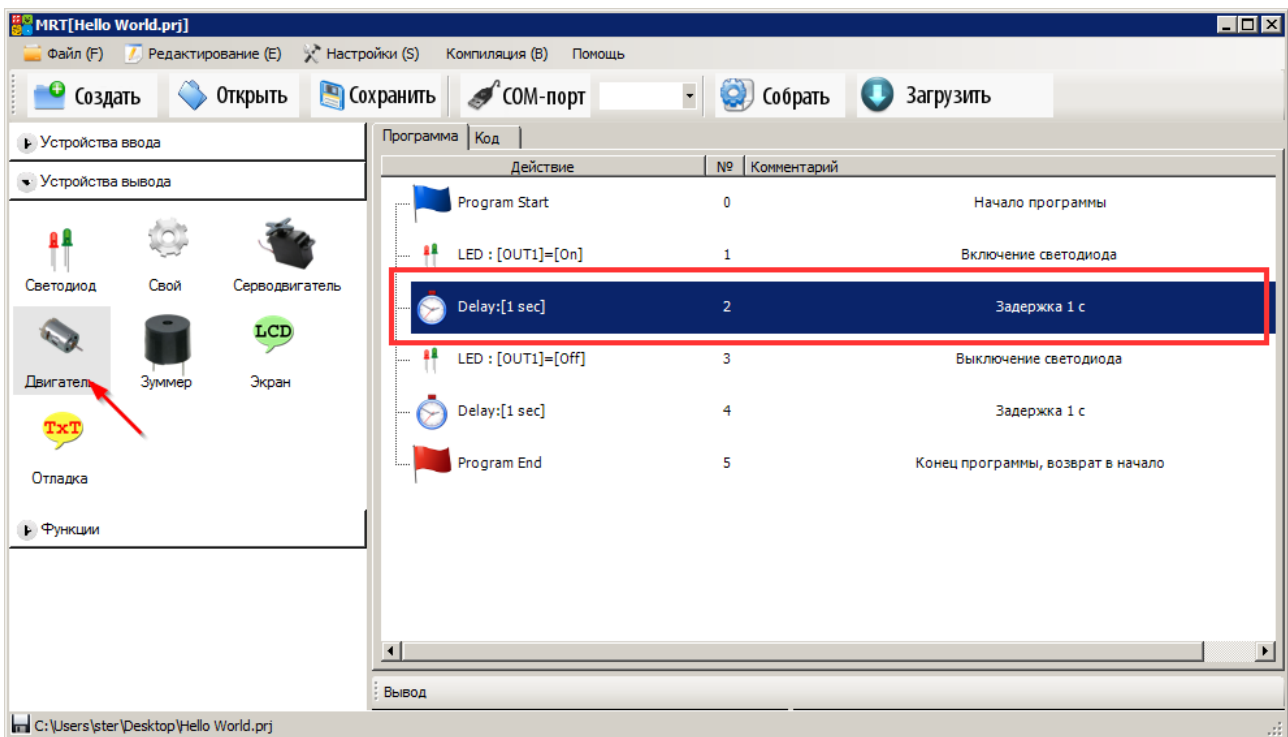


После этого откроется отдельное окно настройки параметров устройства. После нажатия на «ОК» в этом окне новое действие добавится в программу.

ВАЖНО: при добавлении действий добавляемое действие вставляется ПОСЛЕ строки, на которой установлен курсор. Например:



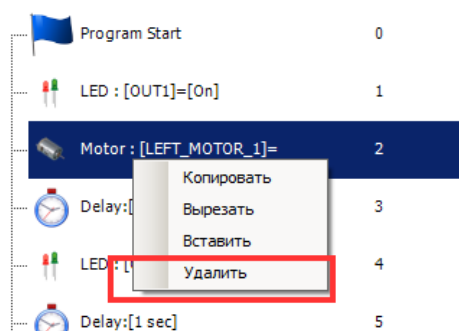
При добавлении двигателя его включение произойдет сразу после включения светодиода, но:



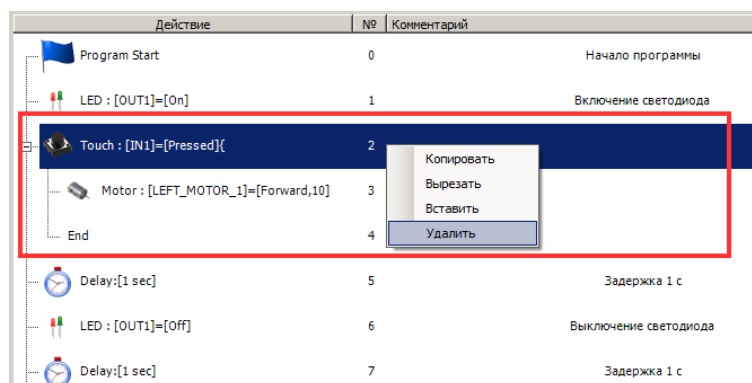
В данном случае команда включения двигателя добавится после задержки, т.е. двигатель фактически включится вместе с выключением светодиода.

Удаление действий

Для удаления необходимой строки сделайте активной строку, которую хотите удалить, и нажмите на клавиатуре клавишу Delete, либо «Удалить» в контекстном меню:



При выборе функций, добавляющих в программу ветвление (устройств ввода), удаляется вся подпрограмма:

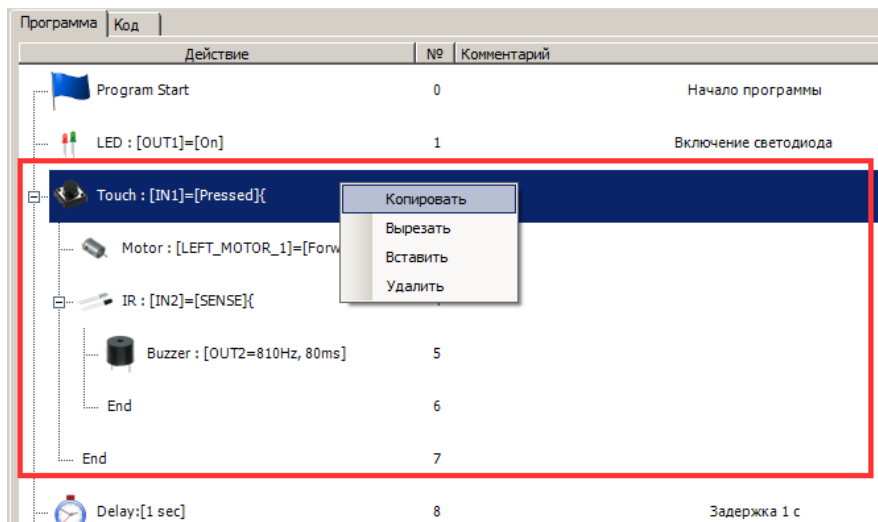


Подробнее о ветвлениях и устройствах ввода читайте в главе «Устройства ввода»

Копирование, вставка, перемещение

К сожалению, на данный момент ПО не поддерживает функцию «Drag-and-Drop», то есть строку нельзя переместить в другое место программы перетаскиванием мышью. Однако это можно осуществить с помощью функций копирования, вырезания и вставки. Их можно выполнять как с помощью контекстного меню, так и помощью стандартных клавиатурных сокращений Control+C (копировать), Control+X (вырезать), Control+V (вставить).

Копировать, вырезать и вставлять можно только ОДНУ строку за раз, несмотря на то, что выбрать их можно несколько. Исключение, по аналогии с удалением, составляют подпрограммы: при копировании (вырезании, вставке) *строки с условием выполнения подпрограммы* копируется (вырезается, вставляется) *вся подпрограмма* вместе со всеми вложенными действиями и подпрограммами:



При удалении выбранной строки удалятся все обозначенные строки

Редактирование параметров функций

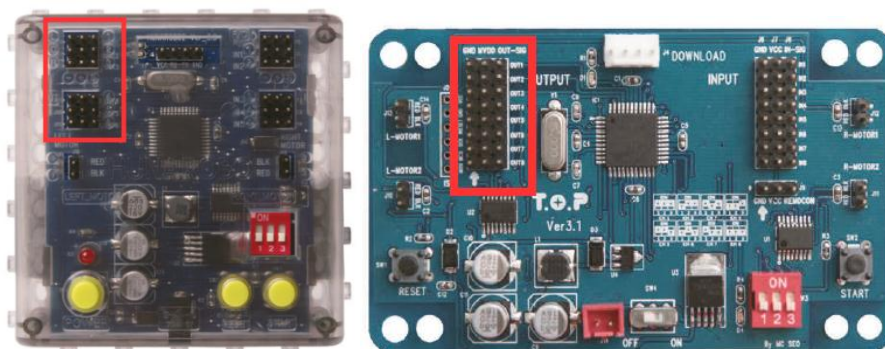
Для того, чтобы отредактировать параметры какой-либо уже добавленной строки, необходимо дважды кликнуть мышью на нужной строке, после чего откроется окно редактирования параметров.

Устройства вывода

В ПО можно посылать команды на следующие устройства вывода:

- Светодиод;
- Двигатель постоянного тока;
- Серводвигатель;
- Зуммер (пищалка);
- Экран (временно не используется);
- Свой (включение\выключение произвольного порта);
- Отладка (посылание данных на компьютер);

Все устройства вывода, кроме Двигателей постоянного тока и «Отладки», подключаются в любой из портов «OUT»:



Расположение портов OUT

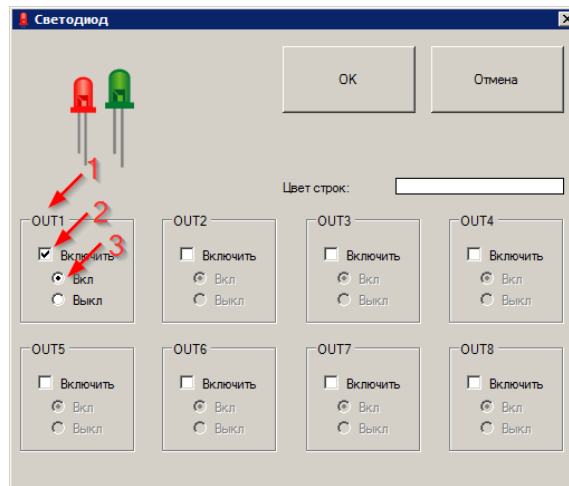
Не забывайте соблюдать полярность и правильно подключать периферию!

Добавление в программу команд устройствам вывода не добавляет в программу никаких ветвлений.



Модули светодиодов

Данная функция позволяет включать и выключать светодиоды:



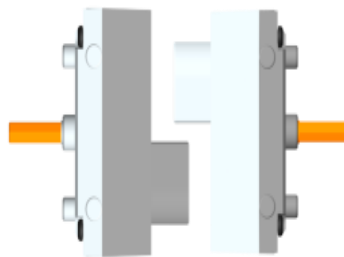
Окно настройки параметров функции управления светодиодами

1. Обозначение порта, в который включен светодиод. Как видно, можно подключить до 8 светодиодов одновременно, и всеми можно управлять одним действием.
2. Флажок «Включить» можно читать как «отправить команду». Можно выбрать только те светодиоды, которые необходимо переключить. Недействующим светодиодам не отправляется никакой команды и они остаются в том же состоянии, что были ранее.
3. Выбор состояния светодиода.

Пример простейшей программы мигания одним светодиодом уже представлен ранее в §2.1.1.

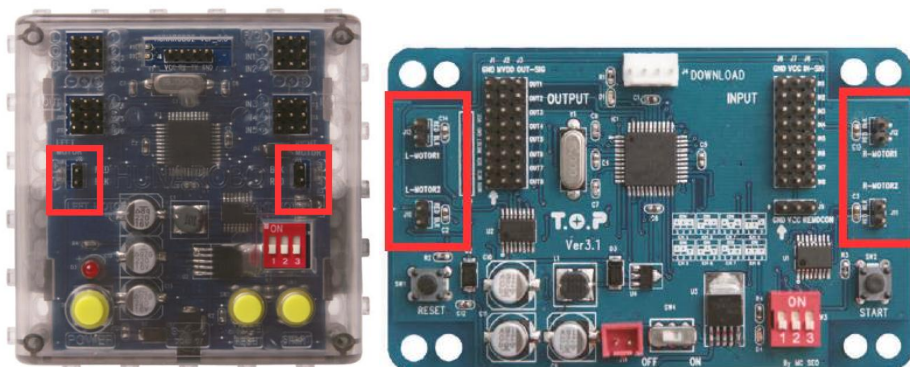


Двигатель **Двигатель постоянного тока**



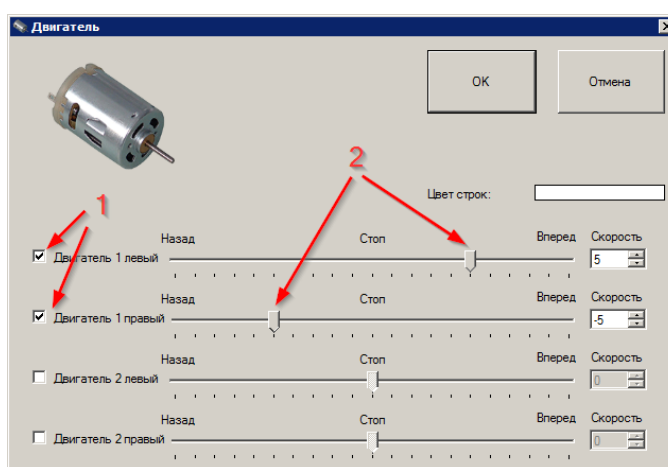
Двигатели постоянного тока

Двигатели постоянного тока включаются в специальные порты двигателей:



Расположение портов двигателей

В платах HUNAROBO и MRT можно управлять двумя двигателями (1 левый и 1 правый), в плате T.O.P – четырьмя (1 левый, 1 правый, 2 левый и 2 правый).



Окно настройки параметров функции управления двигателями











1. Галочки выбора двигателей, состояние которых необходимо изменить.
2. Ползунки установки скорости.

Можно регулировать скорость каждого двигателя независимо. Например, выполнить команду на картинке выше на двуосной тележке, она начнет вращаться на месте вокруг своего центра.

Для **остановки** двигателя необходимо установить галочку выбора двигателя, но скорость оставить «0».

Необходимо понимать, что данной командой мы устанавливаем состояние двигателя, которое не будет меняться, пока не будет установлено другое состояние. Данное правило применимо ко всем устройствам вывода.

Пример программы с использованием двигателя (алгоритм программы понятен из комментариев):

	Program Start	0	
	Motor : [LEFT_MOTOR_1]=[Forward,5],[RIGHT_MOTOR_1]=[Forward,5]	1	Включение 2х двигателей вперед со скоростью = 5
	Delay:[1 sec]	2	Ожидание 1 секунда перед остановкой
	Motor : [LEFT_MOTOR_1]=[Stop,0],[RIGHT_MOTOR_1]=[Stop,0]	3	Остановка двигателей
	Delay:[3 sec]	4	Ожидание 3 с перед началом движения назад
	Motor : [LEFT_MOTOR_1]=[Backward,-5],[RIGHT_MOTOR_1]=[Backward,-5]	5	Включение 2х двигателей назад со скоростью = 5
	Delay:[1 sec]	6	Ожидание 1 секунда перед остановкой
	Motor : [LEFT_MOTOR_1]=[Stop,0],[RIGHT_MOTOR_1]=[Stop,0]	7	Остановка двигателей
	Delay:[3 sec]	8	Ожидание 3 с перед переходом к началу программы
	Program End	9	

Пример программы управления двигателями



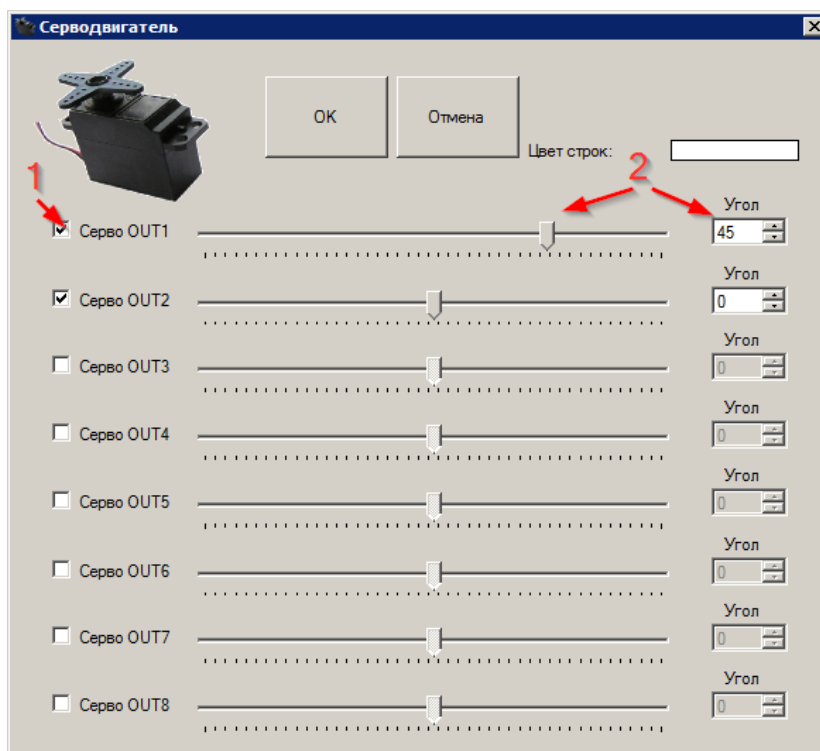
Серводвигатель **Серводвигатель**

Данная функция устанавливает серводвигатель на определенный угол в диапазоне от -90 до +90 градусов.



Серводвигатель

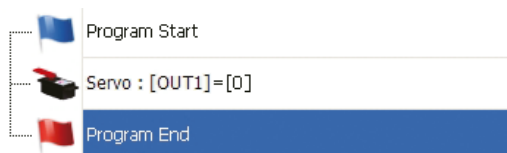
Серводвигатели включаются в любой порт OUT.



Окно настройки параметров функции управления серводвигателями

1. Выбор портов, на которых необходимо изменить угол установки серводвигателя.
2. Ползунок и окошко установки угла поворота оси серводвигателя относительно её нулевой позиции.

У оси серводвигателя существует фиксированное нулевое положение, от которого отсчитывается его рабочий диапазон. Однако серворожок можно закрепить относительно оси как угодно, и «ноль» серворожка может как угодно относительно корпуса. Перед закреплением серворожка болтом нужном положении относительно оси серводвигателя необходимо программно установить ось серводвигателя в нулевую позицию:



Программа, устанавливающая ось серводвигателя в нулевое положение

ВАЖНО: с момента включения платы и до момента получения первой команды серводвигатель не активен и может свободно вращаться. После получения команды установки на какой-либо угол серводвигатель будет продолжать сохранять этот угол и сопротивляться повороту, пока не получит команды установки на другой угол, который так же будет сохранять. Перевести серводвигатель обратно в неактивное состояние программно невозможно.



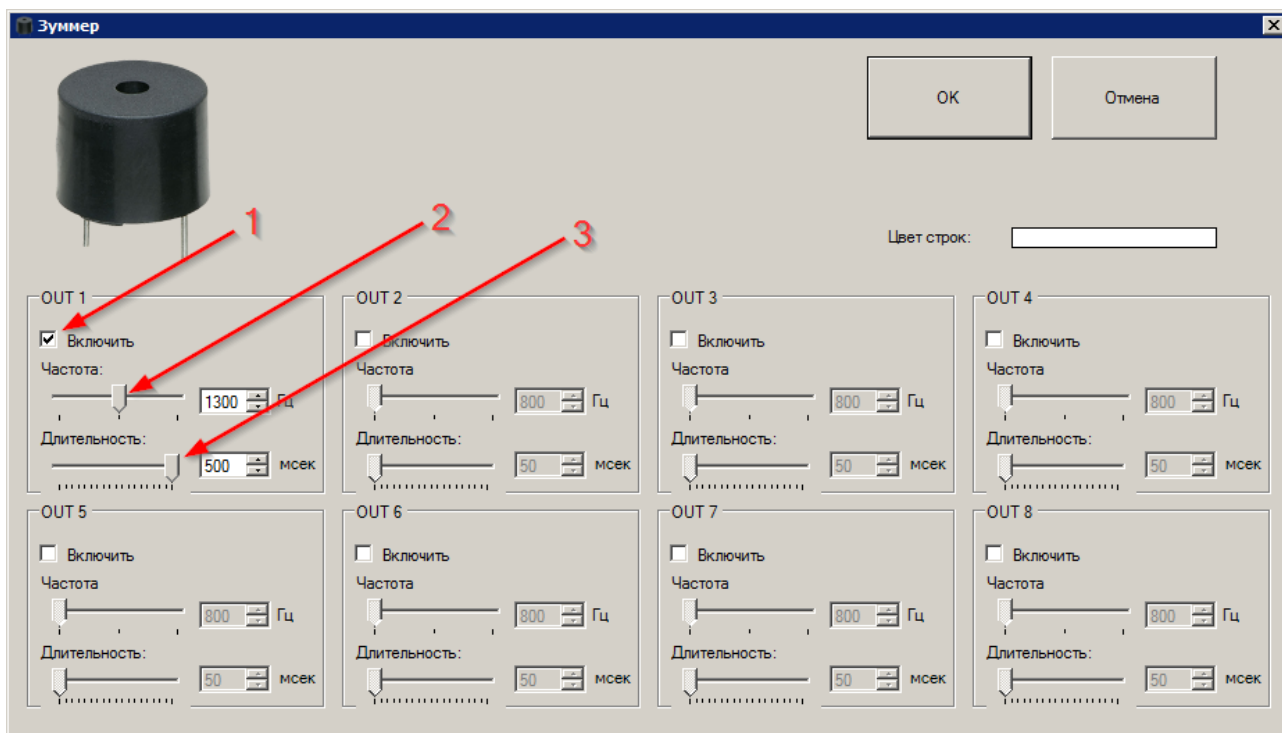
Зуммер (пищалка)



Модуль зуммера

Данная функция включает и выключает подачу на напряжения на определенный порт с заданной частотой в течение заданного промежутка времени, что позволяет издавать звук слышимой частоты с помощью пьезо-пищалки.

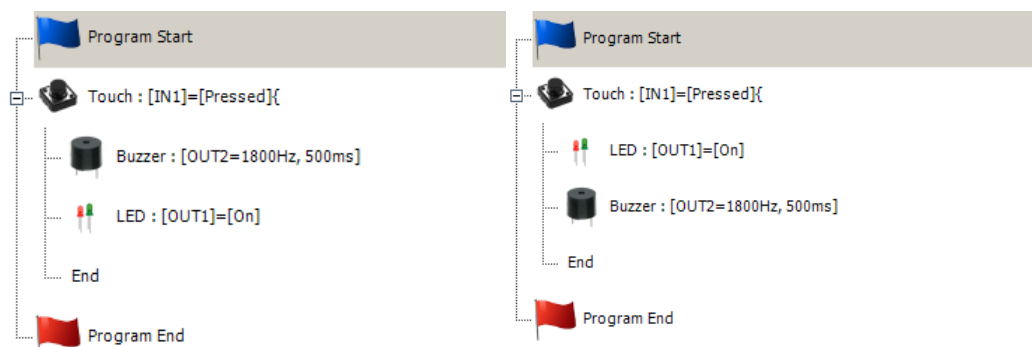
Зуммеры включаются в любой из портов OUT.



Окно настройки параметров функции управления зуммерами

1. Выбор портов, к которым подключены пищалки и на которые необходимо подать сигнал. Можно издавать аккорды, используя сразу несколько зуммеров одновременно.
2. Выбор частоты (высоты) издаваемого звука в Герцах.
3. Выбор длительности издаваемого звука от 50 до 500 мс.

ВАЖНО: во время извлечения звука программа останавливается и не выполняет последующих команд, пока не закончится извлечение звука. Этот факт необходимо учитывать при составлении программ, например:



Светодиод включится только после того, как закончит пищать зуммер, т.е. через 500мс после нажатия кнопки

Светодиод включится вместе с началом звучания зуммера



Экран **Экран (временно не используется)**

Данная функция временно не используется.



Свой **Свой (включение \ выключение произвольного порта)**

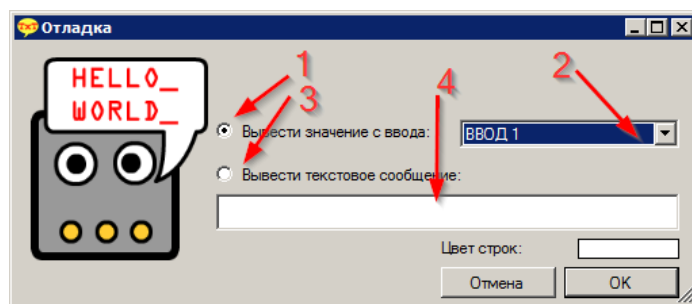
Данная функция предназначена для тех, кто хочет подключить к плате дополнительные устройства, не входящие в набор конструктора, и управлять ими. Например, транзисторами, реле и т.п. По сути функция работает аналогично функции включения\выключения светодиода, т.е. замыкает контакт «S» порта (к которому подключается белый провод) на землю через микроконтроллер.

ВАЖНО: Максимальный ток, который может пропустить через себя один порт – 40 мА. Превышение этого значения навсегда выведет из строя этот порт или даже весь микроконтроллер. Подключайте собственные устройства на свой страх и риск, предварительно обязательно вспомнив закон Ома.



Отладка **Отладка (отправка данных с платы на компьютер)**

Данная функция применяется для отладки программы и мониторинга показаний датчиков. При выполнении строки с этим действием микроконтроллер отправляет по последовательному интерфейсу (тому же проводу, по которому вы загружаете программу в плату) набор символов, которые отображаются в области 5 главного окна программы.



Окно настройки параметров функции отладки

1. Переключение на режим вывода показаний датчика на произвольном порте ввода. В этом случае на компьютер посылается текущее значение напряжения на порте «S», приведенное к диапазону от 0 до 1024, при этом в случае цифровых датчиков (кнопок) будет отправляться 0 в отпущенном состоянии и 1024 в нажатом, а в случае цифровых датчиков (ИК, освещенности, микрофона) будет отображаться его текущее показание.
2. Выпадающий список выбора порта ввода, показания с которого следует посылать.
3. Переключение на режим вывода произвольно текста.

Подсказка: Для перехода на новую строку выведете как текст команду «\r\n» (без кавычек). Программа может выводить текст, содержащий только латинские буквы, цифры и знаки. Вывод русского текста невозможен.

ВАЖНО: не забывайте добавлять паузы между командами вывода сообщений, иначе сообщения будут посылаться и меняться с огромной скоростью и вы не сможете извлечь из них нужную информацию.

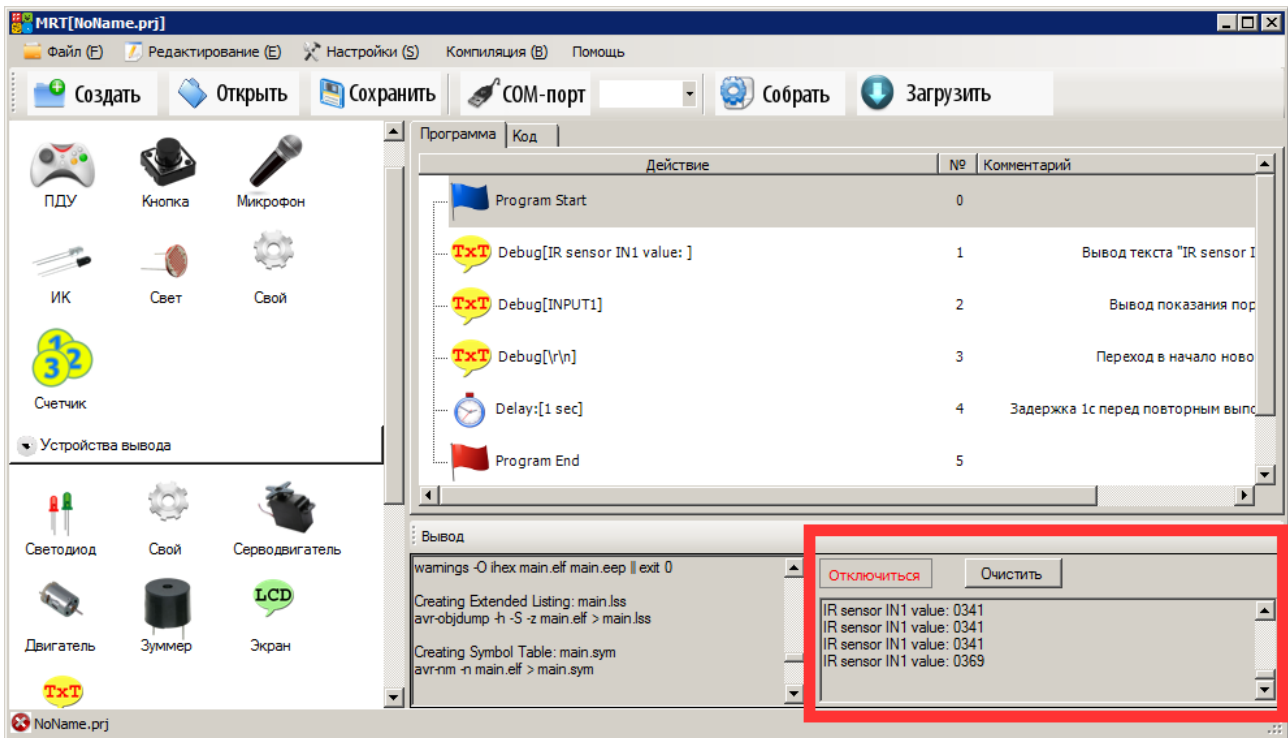
Действие	№	Комментарий
Program Start	0	
Debug[IR sensor IN1 value:]	1	Вывод текста "IR sensor IN1 value: "
Debug[INPUT1]	2	Вывод показания порта IN1
Debug[\r\n]	3	Переход в начало новой строки
Delay:[1 sec]	4	Задержка 1с перед повторным выполнением программы
Program End	5	

Пример программы вывода на экран текста и показаний датчика

Для просмотра посылаемых с платы данных на компьютере необходимо:

1. Скомпилировать и загрузить программу, содержащую функции отладки, в плату (см. раздел «Загрузка»);
2. Перезагрузить плату, не отсоединяя от нее загрузочного провода;
3. В области 5 главного окна нажать кнопку «Подключиться», после чего в окне ниже начнут отображаться данные, приходящие с платы. По завершении работы перед

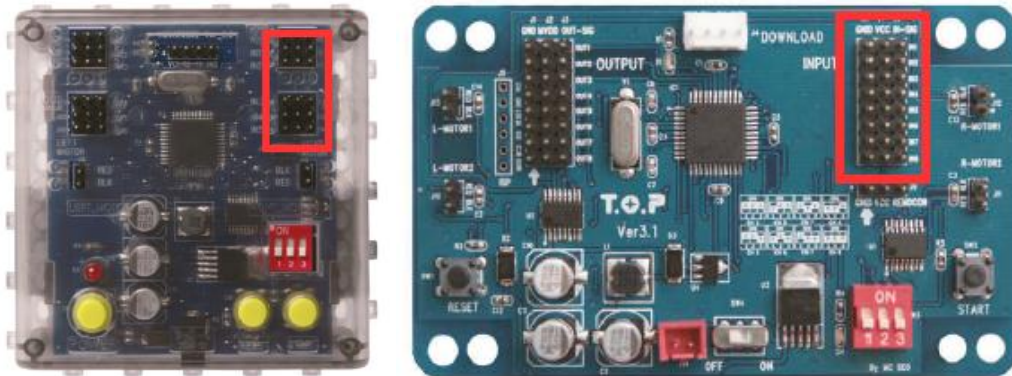
выключением платы нажать кнопку «Отключиться». Кнопка «Очистить» очищает содержимое области 5.



Область вывода информации, полученной с платы

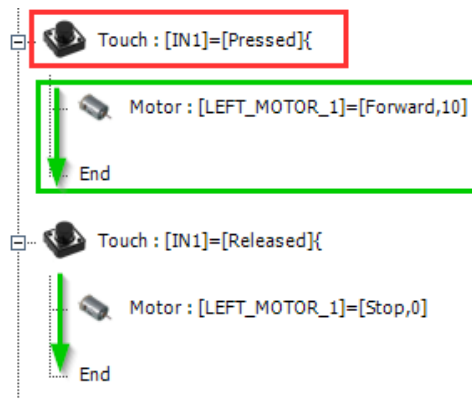
Устройства ввода

Все устройства ввода (кроме ПДУ) подключаются в разъемы INPUT (IN):

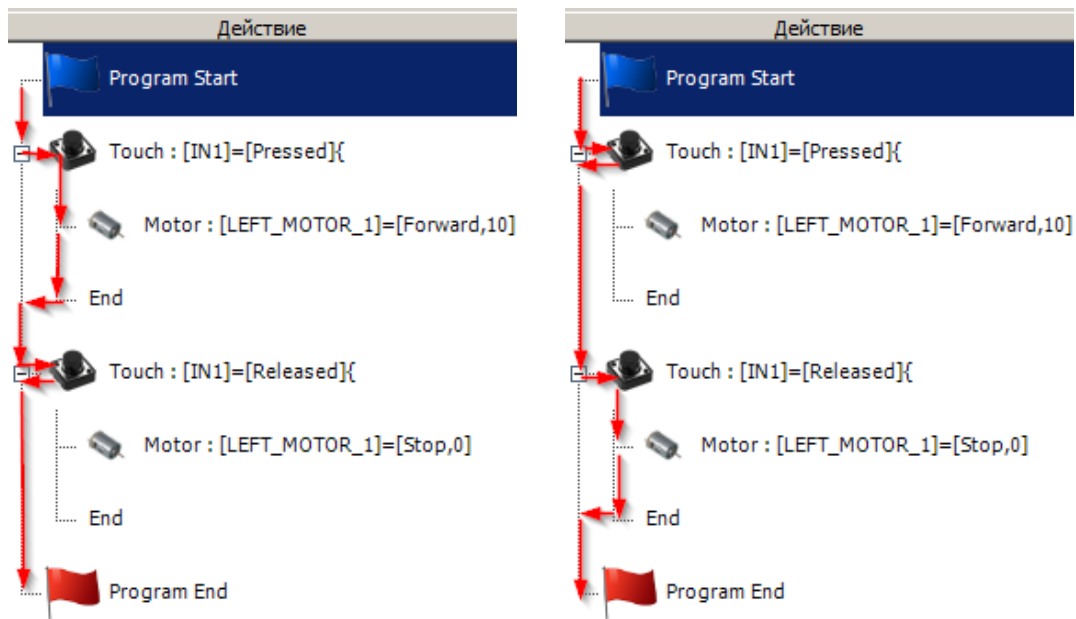


Расположение портов IN

Добавление функции опроса любого из устройств ввода добавляет в программу ветвление (оператор *if()*...), которое можно рассматривать как подпрограмму, в которую ход выполнения программы будет заходить только в случае выполнения определенного условия (в нашем случае – определенных показаний датчика). Опрос датчика происходит в момент, когда программа находится на строке с условием. В случае невыполнения этого условия программа пропустит подпрограмму и начнет выполнять действия основной программы. Путь, по которому будет двигаться программа, можно отследить по серой пунктирной линии в графическом режиме:

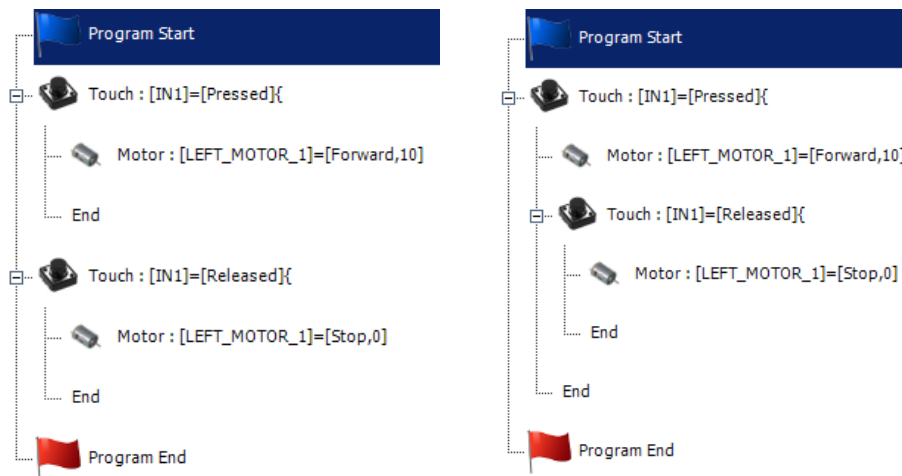


Условие выполнения подпрограммы и подпрограмма



Путь выполнения программы: а) при нажатии кнопки; б) при отпуске кнопки

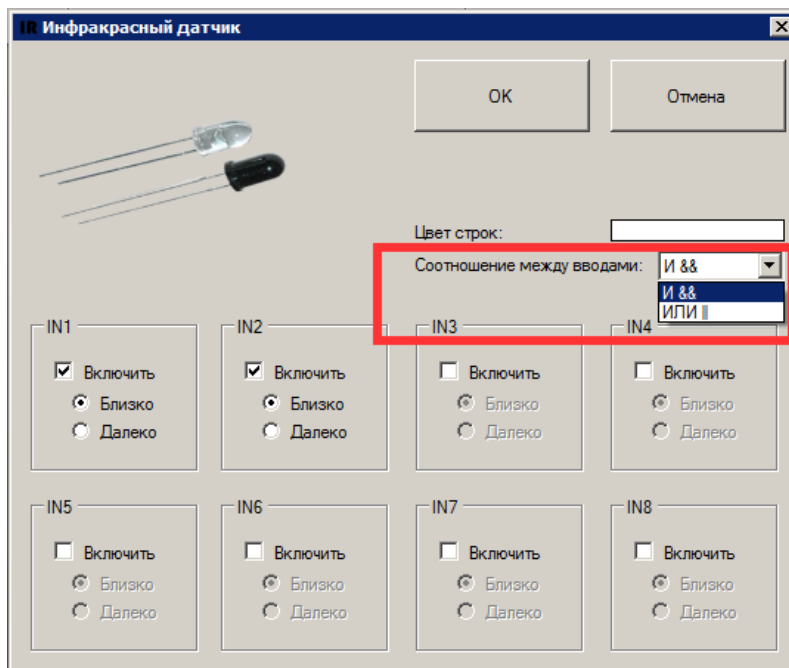
При программировании для выхода из подпрограммы и добавления следующего действия в основную программу необходимо перед добавлением действия установить курсор и сделать активной строку «End» подпрограммы, иначе, к примеру, опрос датчика может происходить только в подпрограмме:



Правильно

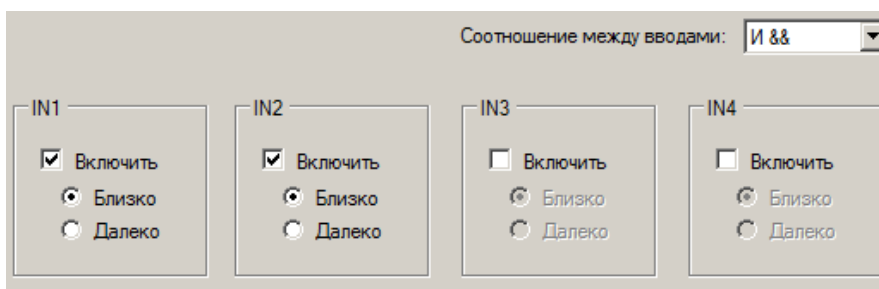
Неправильно

При опросе нескольких датчиков одновременно можно добавлять в условие выполнения подпрограммы логические «И» и «ИЛИ». Соответствующие выпадающие списки есть во всех окнах настройки функций опроса датчиков (кроме ПДУ):

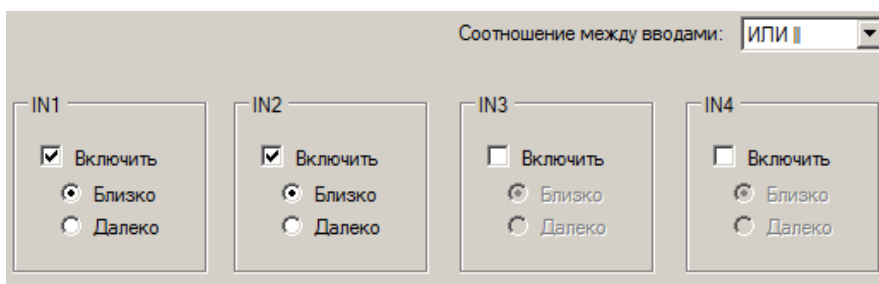


Выбор соотношения между вводами

С работой и назначением этих логических элементов проще всего разобраться на примерах:



Подпрограмма будет выполняться если И первый, И второй датчик будут чувствовать «Близко»



Подпрограмма будет выполняться если ИЛИ первый, ИЛИ второй датчик будут чувствовать «Близко»

Соотношение между вводами: И &&

IN1	IN2	IN3	IN4
<input checked="" type="checkbox"/> Включить	<input checked="" type="checkbox"/> Включить	<input type="checkbox"/> Включить	<input type="checkbox"/> Включить
<input checked="" type="radio"/> Близко	<input type="radio"/> Близко	<input checked="" type="radio"/> Близко	<input checked="" type="radio"/> Близко
<input type="radio"/> Далеко	<input checked="" type="radio"/> Далеко	<input type="radio"/> Далеко	<input type="radio"/> Далеко

Подпрограмма будет выполняться только если первый датчик будет чувствовать «Близко», а второй – «Далеко»

Соотношение между вводами: ИЛИ I

IN1	IN2	IN3	IN4
<input checked="" type="checkbox"/> Включить	<input checked="" type="checkbox"/> Включить	<input type="checkbox"/> Включить	<input type="checkbox"/> Включить
<input type="radio"/> Близко	<input type="radio"/> Близко	<input checked="" type="radio"/> Близко	<input checked="" type="radio"/> Близко
<input type="radio"/> Далеко	<input checked="" type="radio"/> Далеко	<input type="radio"/> Далеко	<input type="radio"/> Далеко

Подпрограмма будет выполняться при выполнении хотя бы одного из условий: ИЛИ если первый датчик будет чувствовать «Близко», ИЛИ второй – «Далеко»

ВАЖНО: при работе с несколькими кнопками существуют особенности, см. соответствующий параграф.

В ПО можно работать со следующими устройствами ввода

- Пульт дистанционного управления («ПДУ»)
- Кнопка
- Микрофон
- Инфракрасная оптопара («ИК»)
- Датчик освещенности («Свет»)
- Свой
- Счетчик



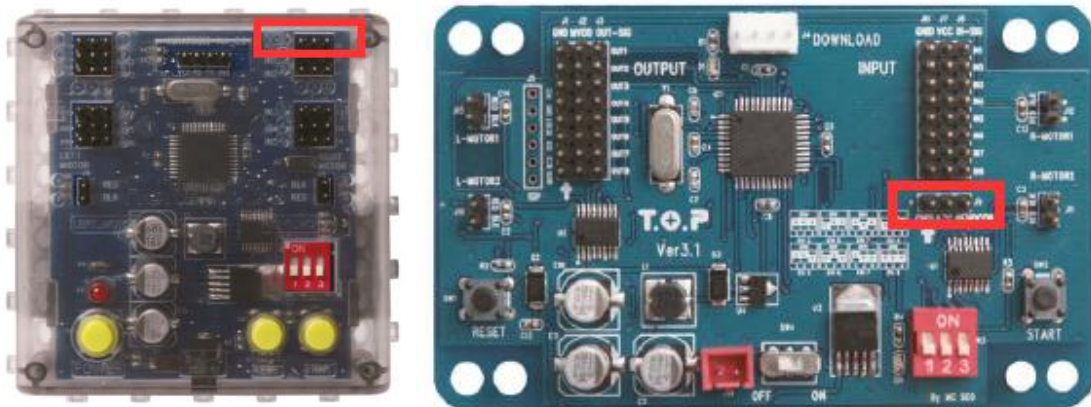
пду

Пульт дистанционного управления («ПДУ»)

Для приема инфракрасных сигналов с ПДУ необходимо подключить приемник ПДУ к разъему «R/C» на плате:



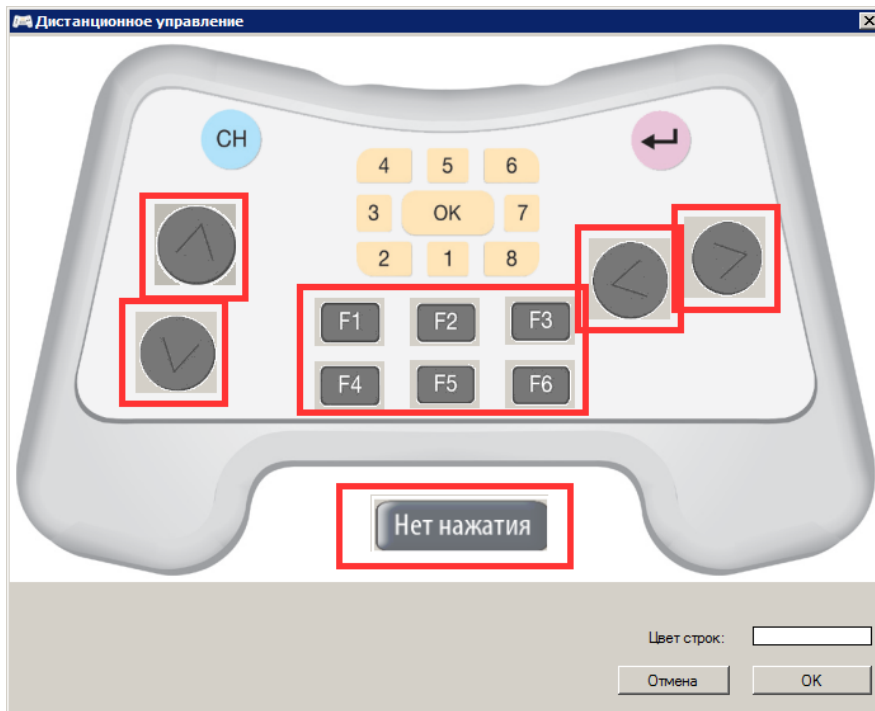
Приемник сигналов ПДУ



Расположение портов подключения приемника сигналов ПДУ

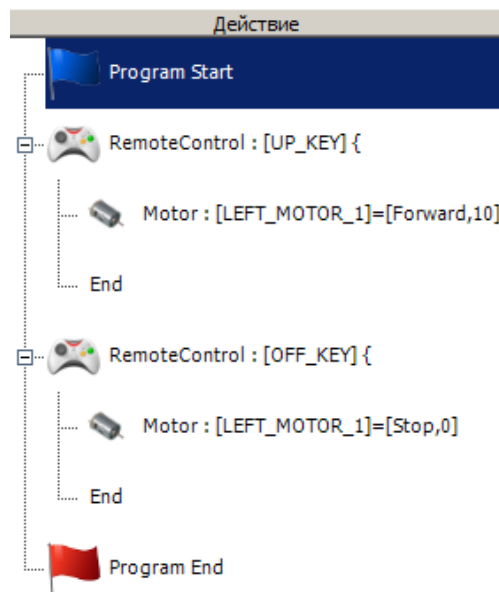
После нажатия на значок «ПДУ» в панели функций откроется окно с изображением ПДУ, на котором нужно мышкой выбрать кнопку, действие при нажатии которой вы хотите запрограммировать. Можно создать до 15 подпрограмм для нажатия различных кнопок на ПДУ:

- Отсутствие нажатия
- Вперед
- Назад
- Влево
- Вправо
- Одновременное нажатие вперед и влево
- Одновременное нажатие вперед и вправо
- Одновременное нажатие назад и влево
- Одновременное нажатие назад и вправо
- Нажатие кнопок F1 – F6



Кнопки, для которых можно создать подпрограммы

Если вы включаете, например, двигатели при нажатии кнопки «вперед», не забывайте их останавливать при отпуске кнопки (отсутствии нажатия):



Пример подпрограммы для кнопки «Нет нажатия»



Кнопка **Кнопка**

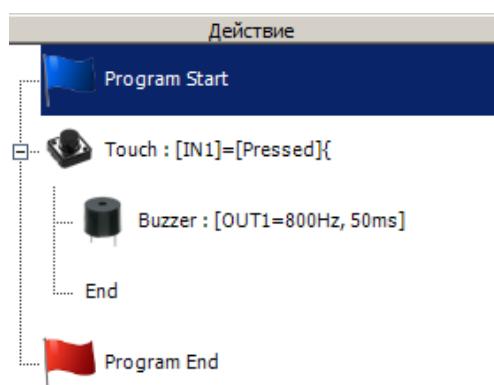


Модуль кнопки

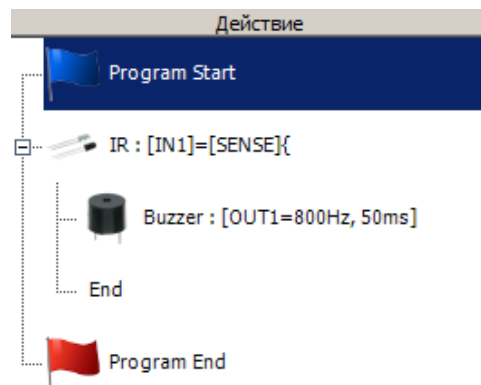
У кнопок есть особенность: функция, которая проверяет состояние кнопки, реализована таким образом, что подпрограмма, назначенная на нажатие кнопки, будет выполняться только один раз при нажатии кнопки. Аналогично с отпусканием кнопки. По сути программа не постоянно опрашивает состояние кнопки, а реагирует на 2 вида событий:

- Нажатие кнопки (переход кнопки из отпущенного состояния в нажатое) и
- Отпускание кнопки (переход кнопки из нажатого состояния в отпущенное).

Данная особенность есть только у кнопки. Например:



Зуммер пикнет один раз при нажатии кнопки и будет молчать, пока кнопка не будет отпущена и нажата снова



Зуммер будет пищать постоянно, пока ИК-датчик чувствует «Близко»

Поэтому нужно внимательно анализировать программу при использовании нескольких кнопок и логического отношения «И» между ними, т.к. совершенно одновременно изменить состояние кнопок практически невозможно.



Микрофон **Микрофон**



Модуль микрофона

Микрофон (датчик громкого звука) включается в любой из портов IN. Можно опрашивать несколько микрофонов одновременно, например, для реализации алгоритма, определяющего направление, откуда идет звук.

Микрофон работает как аналоговый датчик, однако рассматривается программой как цифровой. Порог перехода из состояния «тишина» в состояние «звук» соответствует хлопку или достаточно громко сказанной фразе на расстоянии не более метра от микрофона.



ИК **Инфракрасная оптопара («ИК»)**



Модуль ИК-датчика

ИК-датчики подключаются в любой из портов IN платы.

ИК-датчики измеряют уровень отраженного ИК-света, выдают высокое значение («Близко») при нахождении близко к какой-либо светлой (хорошо отражающей свет) поверхностью и низкое («Далеко») при нахождении над черной (плохо отражающей свет) поверхностью или когда рядом нет никакой поверхности вообще.

ИК-датчик работает как аналоговый датчик, однако в визуальном режиме с ним можно работать только как с цифровым. Порогу между «Близко» и «Далеко» соответствует показание датчика, равное 950. В текстовом режиме с ним можно работать как с аналоговым датчиком.

Можно опрашивать сразу несколько ИК-датчиков и создавать разные подпрограммы на разные сочетания показаний датчиков с использованием логического «И». Например, используя только 2 датчика можно составить 4 разных варианта срабатывания, используя 3

ИК-датчика – 8, и так далее. В отличие от кнопок, ИК-датчики не создают никаких событий и постоянно опрашиваются при нахождении программы на строке опроса датчика.



Свет Датчик освещенности («Свет»)



Модуль датчика освещенности

Датчики освещенности подключаются в любой из портов IN платы.

Датчики освещенности измеряют уровень общей освещенности (яркость падающего на фотозлемент света) в видимом диапазоне.

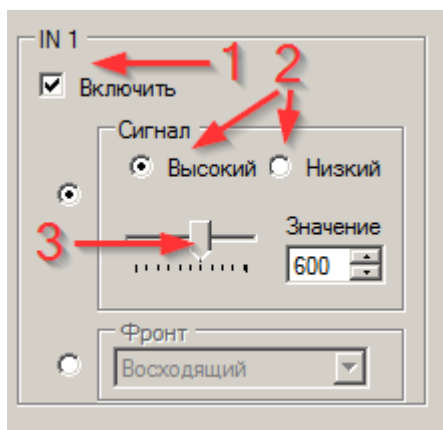
ИК-датчик работает как аналоговый датчик, однако в визуальном режиме с ним можно работать только как с цифровым. Порогу между «Светло» и «Темно» соответствует показание датчика, равное 950. В текстовом режиме с ним можно работать как с аналоговым датчиком.

Логика работы с датчиком освещенности полностью соответствует логике ИК-датчика.



Свой Свой датчик

Данная функция позволяет использовать с платой в визуальной среде любые дополнительные датчики, работающие по резистивной схеме. В визуальной среде с ними можно работать только как с цифровыми. Порог срабатывания можно устанавливать самостоятельно:



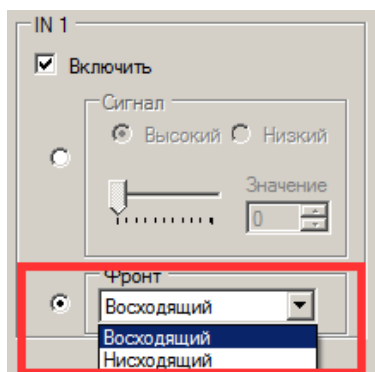
1 – включение\отключение опроса датчика на определенном порту;

2 – выбор параметров условия выполнения подпрограммы – если показания датчика выше\ниже порогового значения;

3 – Установка порогового значения.

При этом уровень 1024 соответствует сопротивлению датчика = 0, а 0 – сопротивлению порядка 10кОм.

Помимо использования пороговых значений можно работать с характером изменений показаний датчика. Микроконтроллер будет дважды измерять показания датчика с небольшой задержкой и определять характер их изменения:



При этом подпрограмма, связанная с этим датчиком, будет выполняться, если:

- второе измерение оказалось больше первого («Восходящий фронт»), или
- второе измерение оказалось меньше первого («Нисходящий фронт»).



Счетчик **Счетчик**

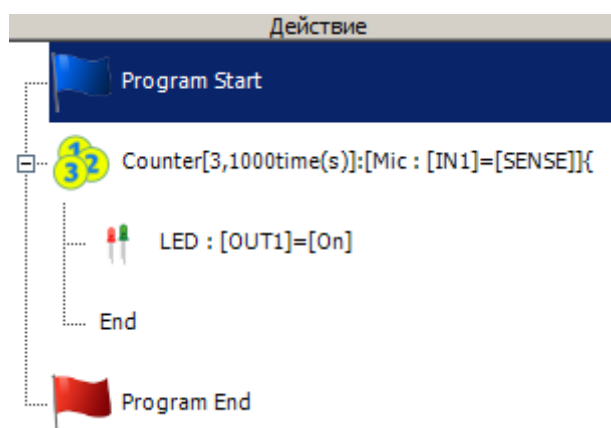
Функция «Счетчик» позволяет создать такое условие выполнения подпрограммы, которое выполняется только при срабатывании какого-либо датчика некоторое количество раз в течение заданного времени:



Окно настройки параметров функции счетчика

1. Выбор датчика, срабатывания которого будем считать;
2. Установка количества срабатываний для выполнения условия;
3. Установка максимального временного промежутка, между первым и последним срабатыванием.

Например:



Пример программы, использующей счетчик

В этой программе светодиод загорится, только если микрофон услышит хлопок (громкий звук), и в течение 1000мс услышит еще два. На 1001 миллисекунде программа перестает считать хлопки, поэтому если третий хлопок произойдет на 1002 миллисекунде, он будет для программы уже не третьим, а первым, и программа начнет ждать еще двух.

Важно: Нужно помнить, что пока программа считает и ждет хлопки (или срабатывания других датчиков), она останавливается на строке с счетчиком и не выполняет никаких других действий.

Программа не считает срабатывания датчиков, произошедшие менее чем в 100мс друг от друга. Поэтому если в качестве датчика выбрать ИК-датчик и 3 срабатывания, а потом

зажать датчик пальцем, то условие выполнится через 200мс после того, как вы поднесете палец к датчику.

Функции

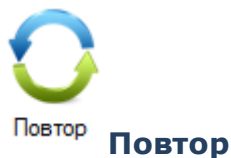
ПО позволяет добавлять в программу 4 вида дополнительных функций, связанных не с устройствами ввода-вывода, а с процессом выполнения самой программы:

- Стоп
- Повтор
- Задержка
- Переход

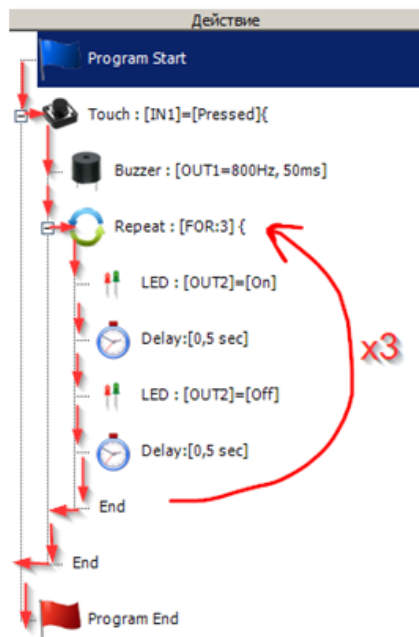


Команда «Стоп» останавливает работу микроконтроллера до перезагрузки платы. Программно возобновить выполнение программы нельзя. По сути при выполнении этой команды программа «зависает» в бесконечном цикле, не выполняя никаких действий.

ВАЖНО: При остановке прекращается выполнение команд программы, но не меняется состояние портов вывода. Поэтому если до команды «Стоп» крутились двигатели или горели светодиоды, они будут продолжать работать и гореть бесконечно долго, пока не выключится питание платы.



Данная функция создает цикл «*For()*...», с подпрограммой, которая будет выполняться заданное количество раз (или бесконечно). Например:

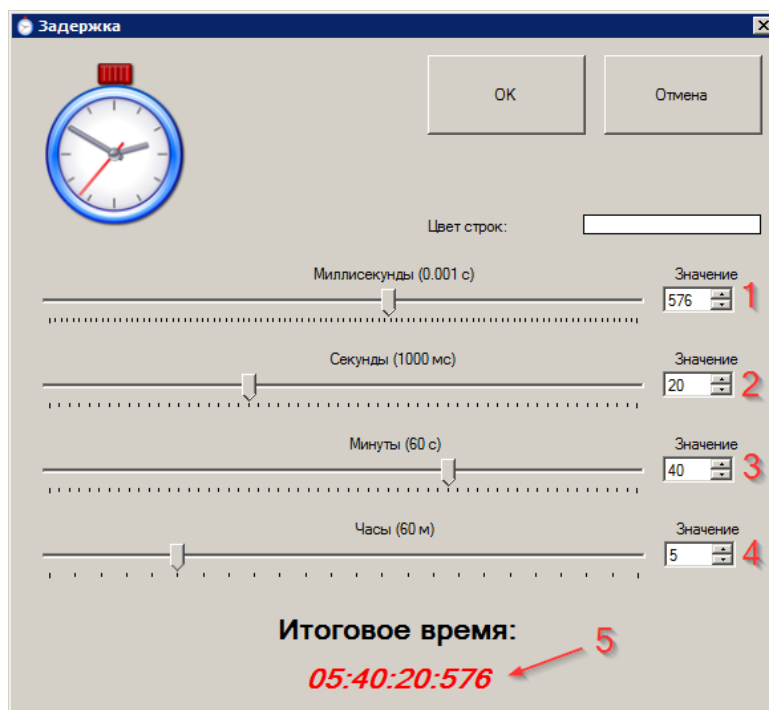


Ход выполнения программы при использовании функции повтора



Задержка **Задержка**

Данная функция приостанавливает выполнение программы на заданное количество времени. Во время выполнения функции задержки команда не выполняет никаких других действий. По окончании времени задержки выполняется следующее действие, идущее за ней в программе.



Окно настройки параметров функции задержки

- 1 – установка количества миллисекунд задержки;
- 2 – установка количества секунд задержки;
- 3 – установка количества минут задержки;
- 4 – установка количества часов задержки;
- 5 – суммарное время задержки;

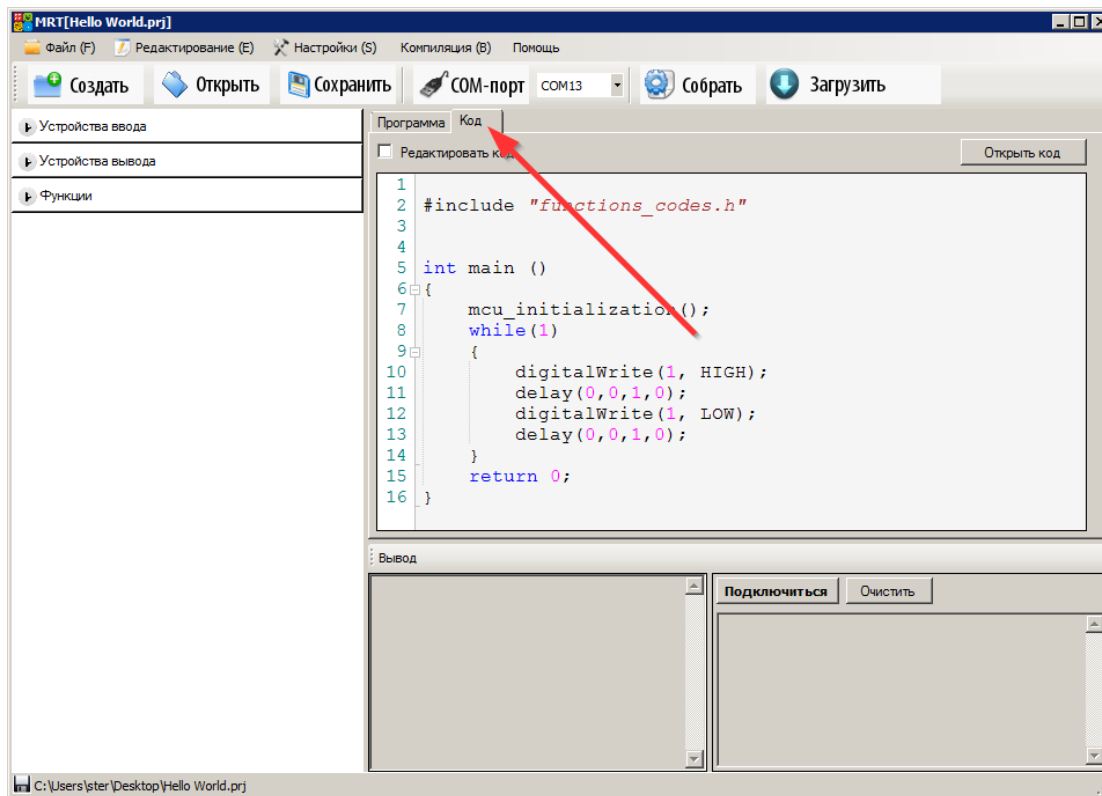
С помощью одной функции можно заставить микроконтроллер «зависнуть» на время от 1 миллисекунды до 23ч 59м 59с 999мс. Устанавливая несколько задержек подряд, микроконтроллер можно заставить ждать бесконечно долго.



Данная команда эквивалентна «goto» и заставляет микроконтроллер прекратить последовательное выполнение программы «сверху вниз» и «перепрыгнуть» в заданное место программы и продолжить выполнение программы с этого места. В визуальной среде можно запрограммировать переход только к первой строке какой-либо подпрограммы, в том числе основной программы. В текстовом режиме можно указать любую строку.

3.1.2. Текстовый режим.

Программа, составленная в визуальном режиме, автоматически транслируется на язык СИ. Для просмотра и редактирования программы на языке СИ предусмотрен текстовый режим, на который можно переключиться, нажав на вкладку «Код» области редактирования программы:



Переключение в режим работы с кодом

Во вкладке «Код» открывается текстовый редактор, заточенный на программирование на языке СИ – с подсветкой синтаксиса, подсказками, автоформатированием и т.д. В данном режиме можно:

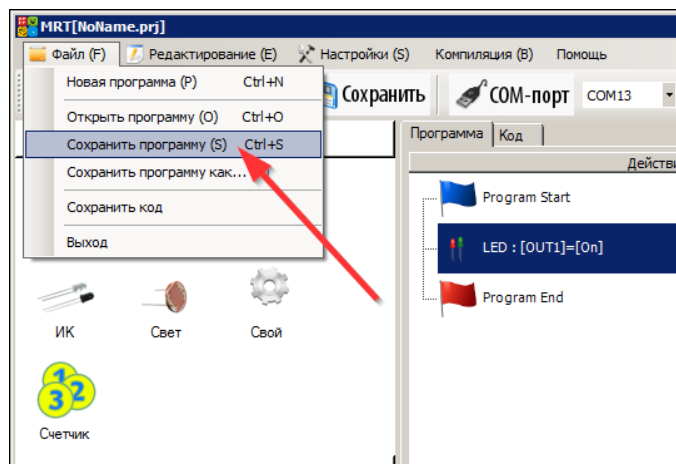
- Просматривать код созданной в визуальном режиме программы, постепенно знакомясь с тем, как выглядят стандартные функции и какие параметры используют (полезно для постепенного перехода с визуального программирования к текстовому);
- **Поставив галочку «Редактировать код»** - получить полный контроль над кодом, создавать собственные переменные, функции, подключать библиотеки и т.д.

ВАЖНО: Пока не стоит галочка «Редактировать код», изменение визуальной программы приводит к изменению кода, однако вручную код изменить нельзя. При переходе в **режим редактирования** кода код возможно редактировать, но эти изменения **не влияют** на визуальную программу. При включенном режиме редактирования кода вы можете удалять строки и изменять параметры в визуальном режиме (и эти изменения будут отражаться в коде), но не можете добавлять новые действия. При выходе из режима редактирования кода (снятии галочки «Редактировать код») **все изменения** в коде, сделанные в текстовом режиме, **теряются**, и в код транслируется оставшаяся во вкладке «Программа» визуальная программа.

3.2. Сохранение созданных программ и их открытие

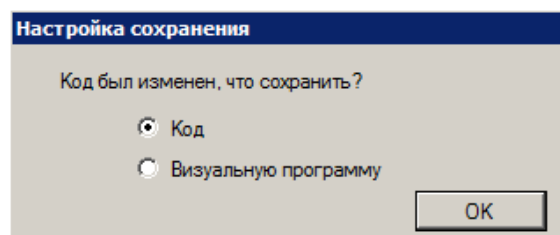
3.2.1. Сохранение

После того, как Вы составили свою программу в визуальном или текстовом режиме, рекомендуется её сохранить на жестком диске. Для этого необходимо нажать на «Файл» - «Сохранить программу» в строке меню или кнопку «Сохранить»:



При первом сохранении будет предложено выбрать имя и место сохранения файла, при последующих сохранениях текущей программы будет перезаписываться уже сохраненный файл. Для того, чтобы сохранить измененную программу под другим именем или в другом месте, необходимо нажать на «Файл» - «Сохранить программу как».

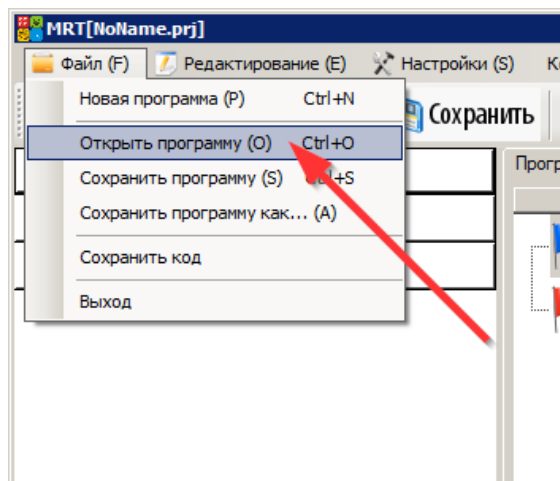
Если Вы изменяли код программы в текстовом режиме, то при первом сохранении программа спросит вас, что сохранить: оригинальную визуальную программу или измененный код:



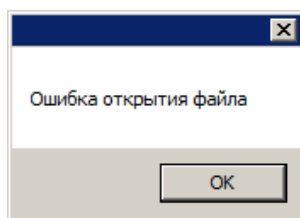
При этом визуальная программа сохраняется в формате «.prj», а код – в формате «.еро».

3.2.2. Открытие

Для **открытия** сохраненных файлов необходимо нажать «Файл» - «Открыть программу» в строке меню и выбрать необходимый файл в открывшемся окне:



ВАЖНО: на некоторых компьютерах может не работать открытие файлов .prj из Проводника Windows (открытие двойным кликом по файлу). При этом выводится сообщение об ошибке и открывается пустая программа:



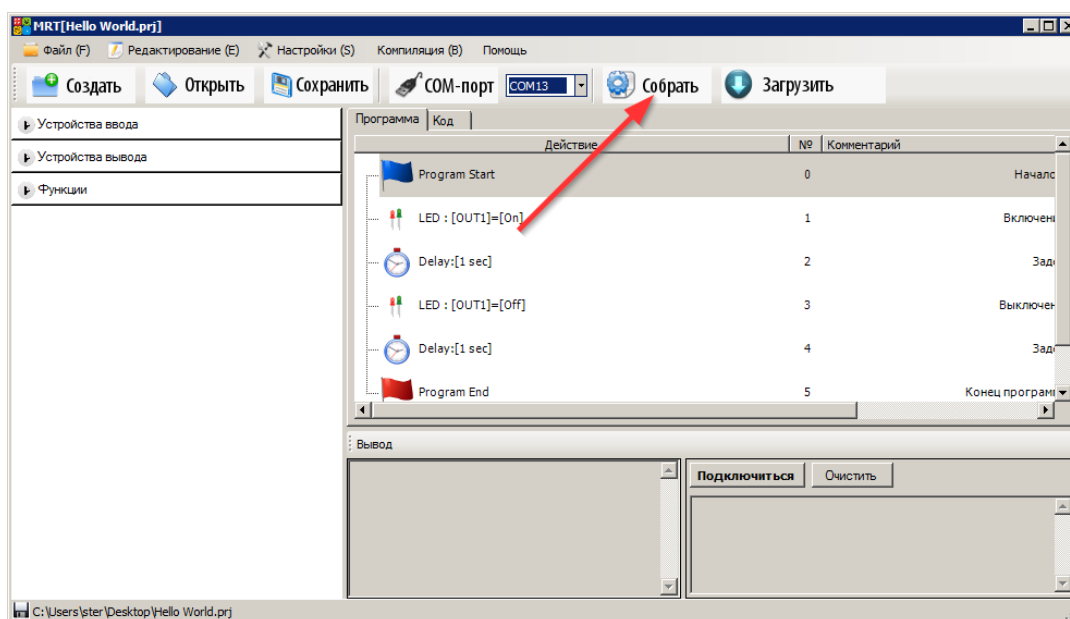
В этом случае необходимо открывать сохраненные файлы из программы, как описано выше.

3.3. Компилирование и загрузка программы в плату

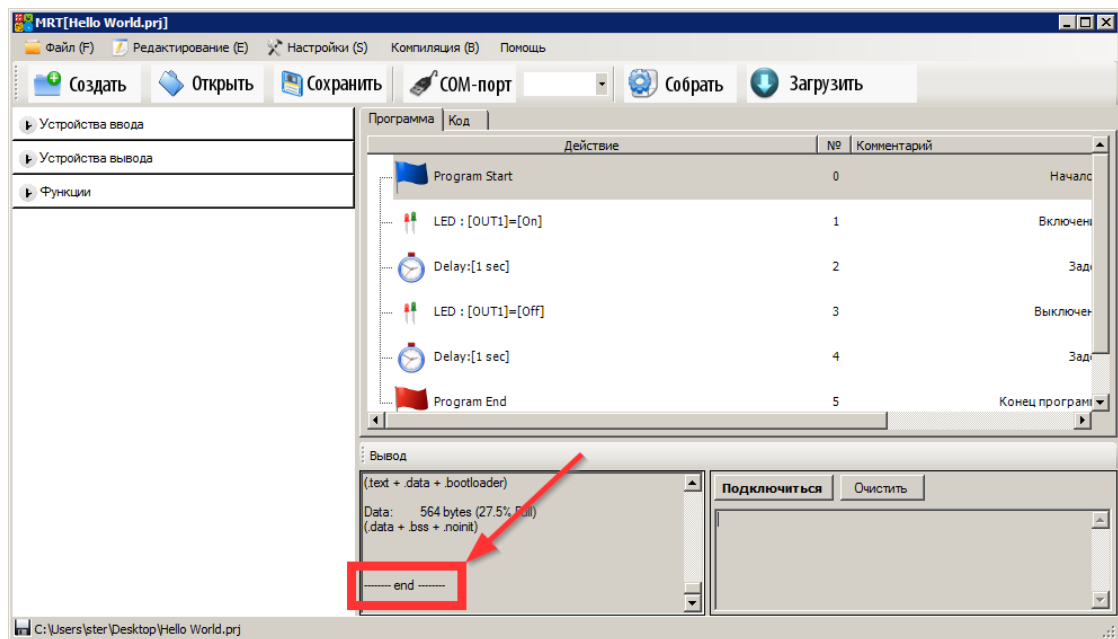
В процессе **компилирования** программа, созданная в визуальном или текстовом редакторах, переводится в шестнадцатеричные машинные коды, которые выполняет микроконтроллер в плате (перезаписывается файл main.hex в папке C:/MRT/). В процессе **загрузки** созданный файл байт за байтом передается и записывается в память микроконтроллера по загрузочному кабелю. Именно поэтому компилирование («сборку») программы необходимо выполнять перед каждой загрузкой программы в плату, иначе загрузится скомпилированный ранее файл, оставшийся, возможно, от другой программы.

3.3.1. Компилирование

Для запуска процесса компиляции необходимо нажать кнопку «Собрать» в панели меню:

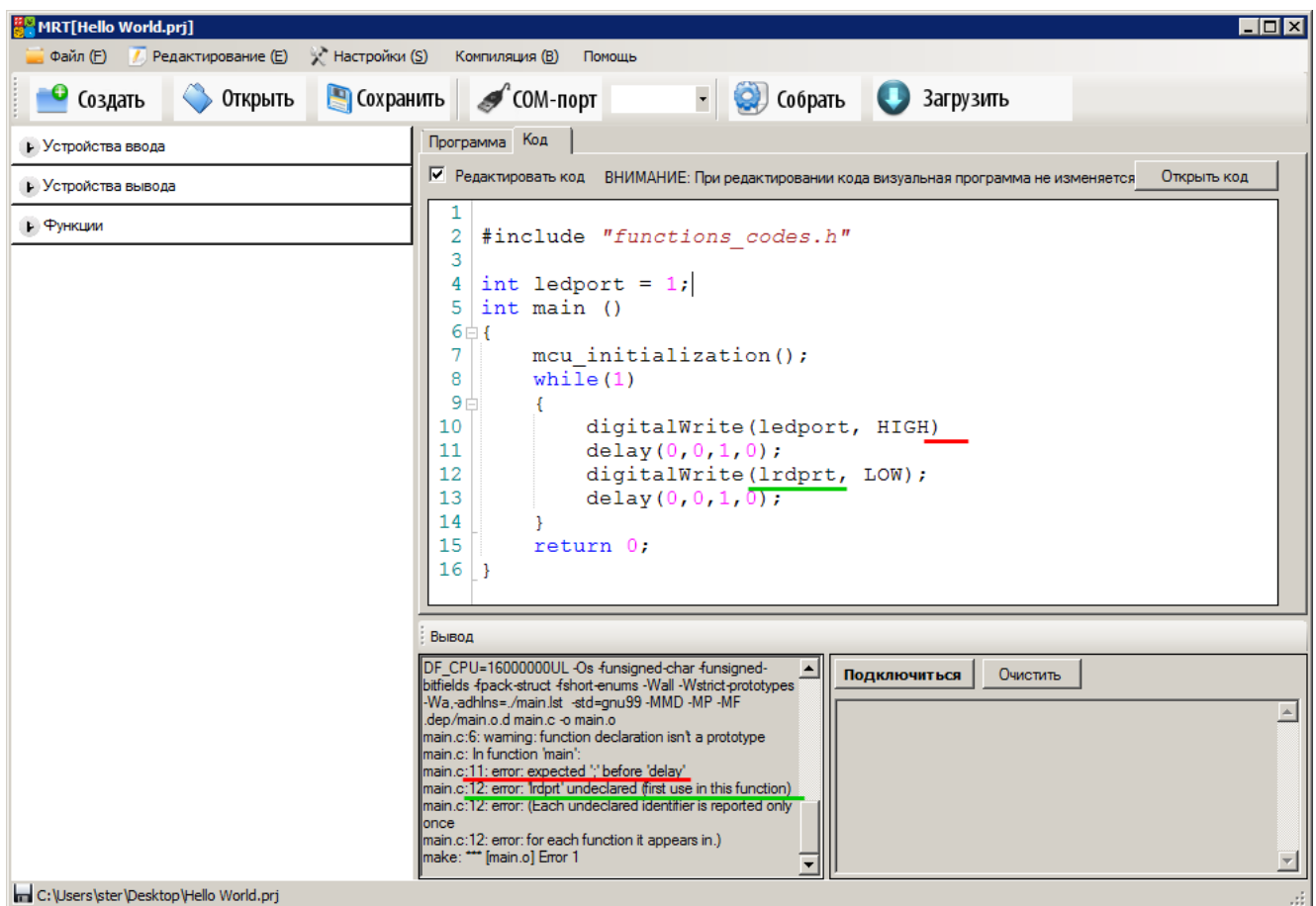


При этом в области 4 главного окна начнут выводиться сообщения компилятора о процессе компиляции, и, если не обнаружено никаких ошибок в программе, через 5 – 20 секунд выведется сообщение «----- end -----», что говорит об успешном завершении компиляции:



Прошивка успешно скомпилировалась, можно загрузить её в плату

Если вместо «----- end -----» появилось сообщение «make: *** [main.o] Error 1», возможно, вы допустили ошибки при редактировании кода (если программа создавалась в визуальном режиме, ошибок быть не может). О предполагаемых ошибках и их месте компилятор сообщает в этом же окне вывода сообщений (область 4):



*Перед строкой её ожидается «;» - в строке 10 пропущена запятая
В строке 12 используется необъявленная переменная «lrdprt» - допущены опечатки в имени переменной*

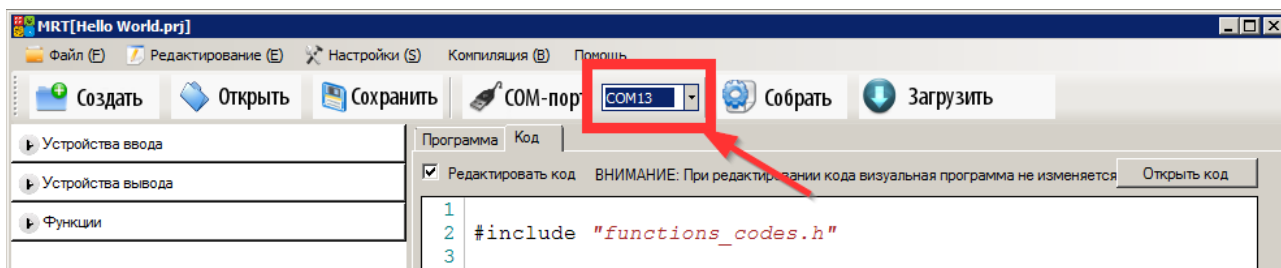
Если ни одно из сообщений не выводится более 30 секунд, необходимо сохранить созданную программу, перезапустить ПО и попробовать запустить компиляцию снова.

3.3.2. Выбор COM-порта

После успешного завершения процесса компиляции можно начинать загрузку программы в плату. Загрузка выполняется с помощью загрузочного устройства (загрузчика) (USB-COM адаптера), вставленного в порт USB и успешно распознанного операционной системой.

Примечание: рекомендуется вставлять загрузчик в компьютер ДО запуска ПО.

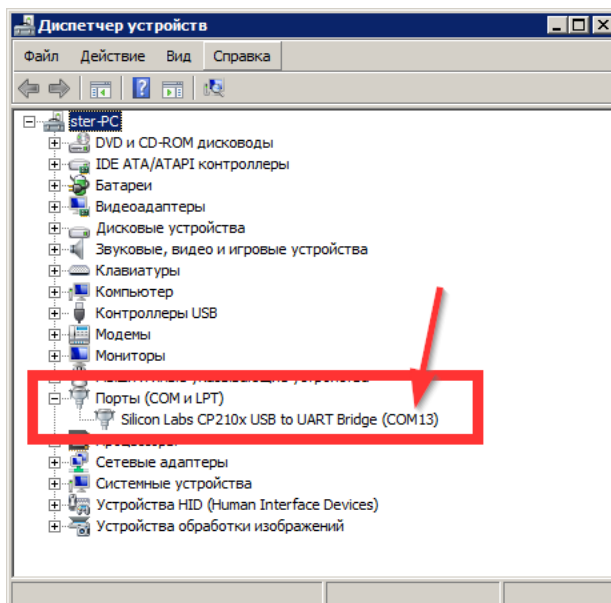
Перед началом загрузки необходимо выбрать в соответствующем списке COM – порт, который создал в системе загрузчик:



Список выбора COM-порта, созданного загрузчиком

Если в выпадающем списке только один порт, смело выбирайте его и начинайте процесс загрузки.

Если же портов несколько, вам необходимо узнать, какому именно порту соответствует ваш загрузчик. Для этого необходимо открыть диспетчер устройств Windows (Пуск – Панель управления – Оборудование и звук – Диспетчер устройств), раскрыть группу «Порты (COM и LPT)»:



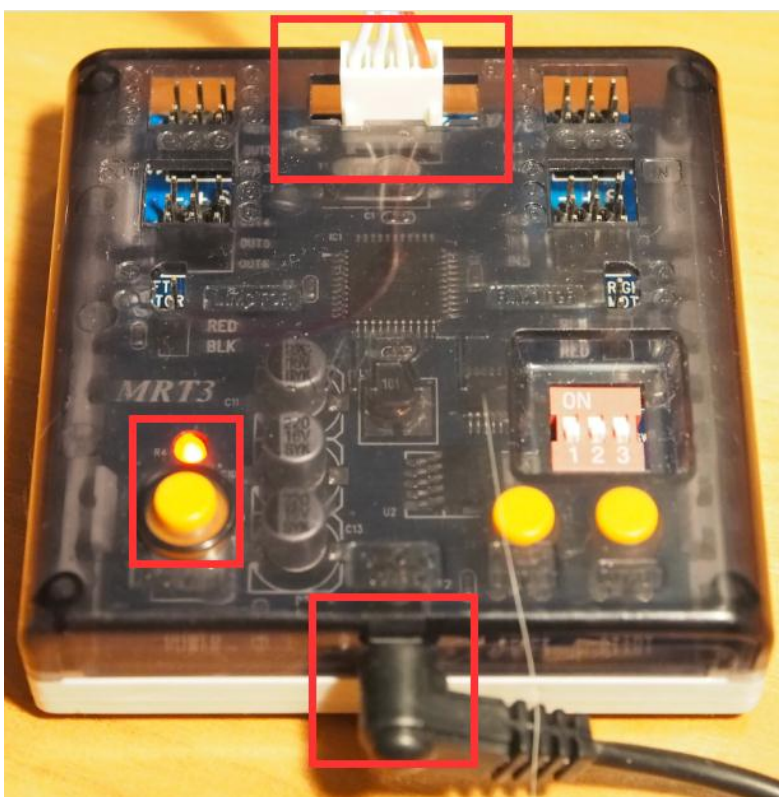
Вам необходимо запомнить номера портов, присутствующих на данный момент в системе, а затем вытащить загрузчик из порта USB. При этом список устройств обновиться и в группе «Порты (COM и LPT)» станет на одно устройство меньше, именно это и есть ваш загрузчик. Снова подключите загрузчик к порту USB, список обновится, убедитесь, что устройству назначен тот же COM-порт, что был до этого. Запомните номер COM-порта, соответствующего загрузчику, вернитесь в ПО MRT 1.2, и выберете в списке необходимый COM-порт.

После каждой загрузки поле «COM-порт» очищается, однако чаще всего заново указывать порт не обязательно, так как при следующей загрузке программа будет использовать тот же порт, что использовала при предыдущей загрузке. Обязательно указывайте программе COM порт только если в системе несколько портов при запуске программы и при возникновении проблем при загрузке (см. далее).

3.3.3. Загрузка

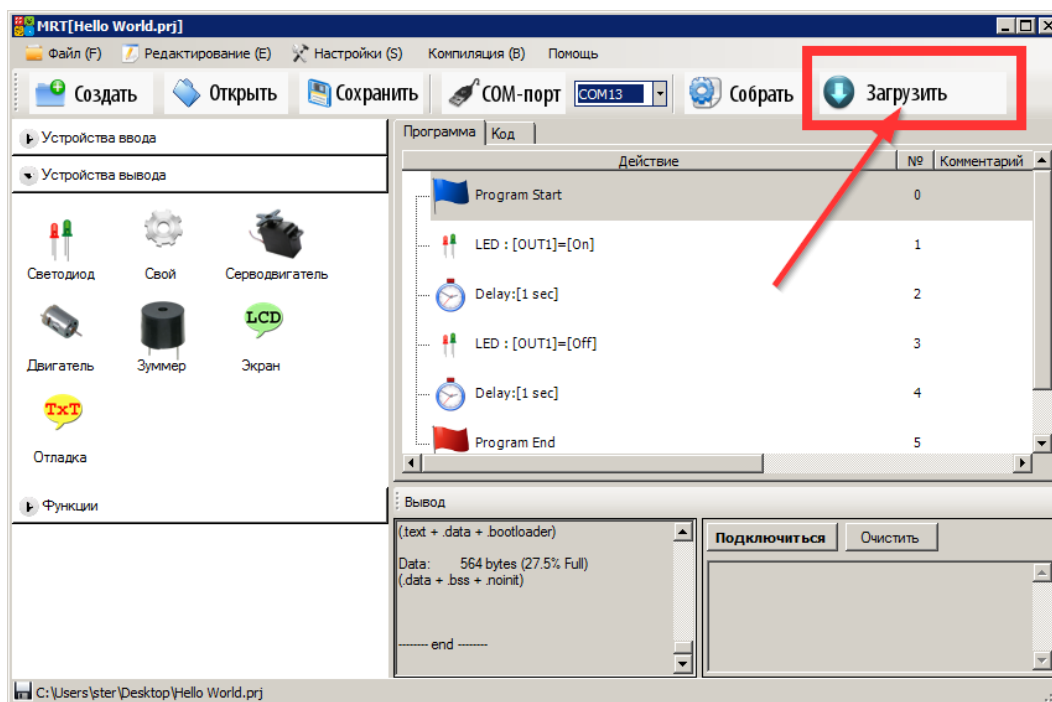
Перед началом загрузки еще раз проверьте, что:

- К плате подключены блоки с заряженными батарейками, питание платы отключено (отжата кнопка «Power»);
- Плата корректно подключена к загрузчику (см. фото ниже), а загрузчик вставлен в порт USB (на загрузчике должен гореть светодиод, на плате тускло гореть светодиод «Power»);
- В ПО указан COM-порт, соответствующий загрузчику.



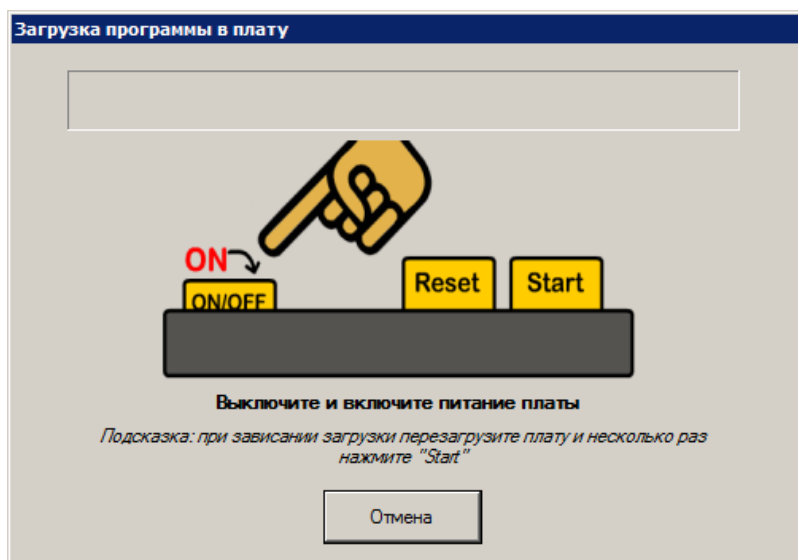
Плата готова к загрузке: разъемы кабеля попали в соответствующие пазы, кнопка «Power» выключена, но светодиод тускло горит, батарейки подключены

После успешной компиляции и определения COM-порта можно начинать процесс загрузки программы в плату, нажав на кнопку «Загрузить»:

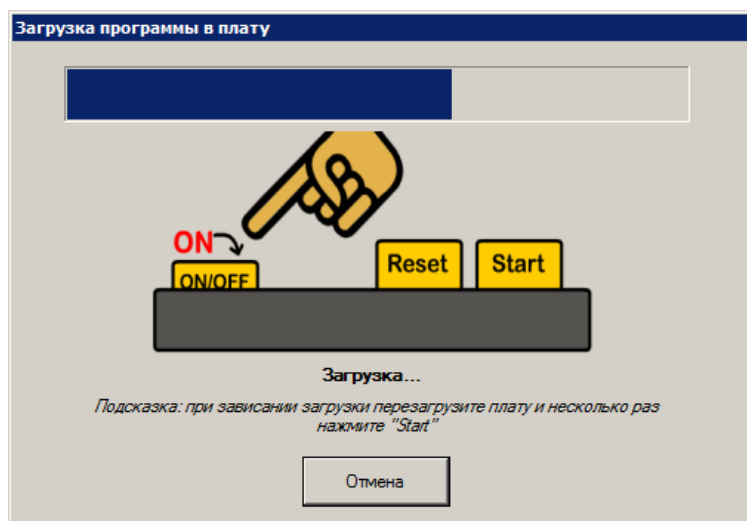


Если ранее программа не сохранялась, то откроется стандартное окно сохранения программы. Сохранение программы можно пропустить, нажав «Отмена». Если программа сохранялась ранее, то файл перезапишется и сохранится текущая версия программы.

После этого откроется окно, в котором будет предложено включить питание платы:



Включите питание платы, нажав кнопку «Power» на плате, или нажмите кнопку «Reset», если плата была включена до этого. Начнется процесс загрузки, прогресс которого будет отображаться в этом же окне:

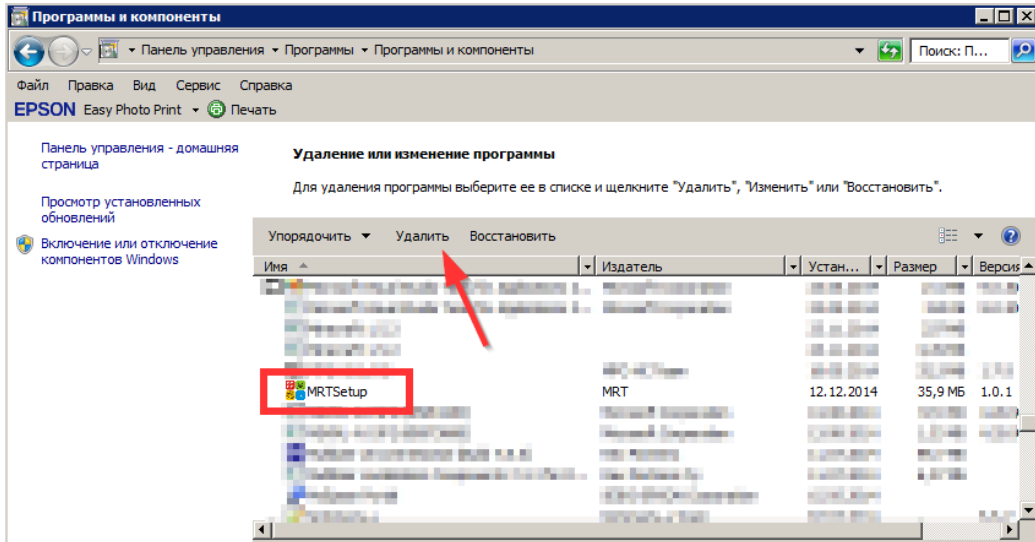


Если при включении питания нажать кнопку не до конца, выключить питание или отсоединить загрузочный кабель в процессе загрузки, то загрузка «зависнет» и окно не будет реагировать на нажатие кнопки «Отмена». В этом случае воспользуйтесь подсказкой: выключите и включите питание платы и несколько раз нажмите кнопку «Start» на плате. Окно закроется и загрузку можно будет попробовать повторить.

При **успешном** завершении загрузки индикатор прогресса дойдет до конца и окно автоматически закроется. Можно отключать кабель загрузчика и перезагружать плату. После перезагрузки плата начнет выполнять загруженную программу.

4. Удаление программы

Для удаления программы из системы воспользуйтесь стандартным средством удаления программ Windows (Пуск – Панель управления – Программы – Удаление программ). Найдите строку «MRTSetup» и нажмите «Удалить»:



Помимо этого можно удалить драйверы загрузчика:

