



## Введение

Развитие микроэлектроники и ее широкое применение в автомобильном производстве, а конкретнее в системах управления самыми разнообразными способами и процессами. В настоящее время это актуальное из основных направлений научно-технического прогресса.

В наше время большинство автовладельцев время от времени сталкиваются с проблемами, такими как: ДТП, конфликтами с другими участниками дорожного движения, а также с нехваткой дополнительных функций для комфортного вождения. На сегодняшние дни автопроизводители придумали множество помощников при вождении, системы автоматической парковки, системы автоматической отключения цилиндров, помощники при торможении, поворотники и аварийный сигнал и многие другие помощники. Но так и не изобрели устройство, которое дает знать водителю идущим позади вас, конкретные, дальнейшее ваши действия на участках дорожного движения.

Целью моего курсового проекта является создание такого помощника для водителей, едущих за вашим транспортным средством и дающий информацию о твоих намерениях в движении. С использованием программированных микроконтроллеров на базе Arduino.

Задачи:

- Изучить существующие аналогичные устройства
- Сделать обзор существующих типов платформ Arduino
- Сделать обзор элементов автоматизации
- Разработать структурную схему
- Обосновать выбор элементной базы
- Разработать электрическую принципиальную схему
- Написать программу для устройства

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

## 1 Общая часть

В настоящее время, в связи с технологическим прорывом в изготовлении светодиодов, наличием высокопроизводительных электронных комплектующих, активно развивается производство ярких, экономичных и динамичных светодиодных информационных табло.

Электронное табло - устройство отображения визуальной информации, изображение, на поверхности которого формируется электронным способом. Наибольший интерес представляют устройства, которые позволяют оперативно изменять изображение (рис. 1.1).



Рисунок 1.1 – Пример электронного табло

Его используют в общественных местах, на вокзалах, в аэропортах, магазинах, крупных торговых объектах для простого и быстрого информирования клиентов. Как и светодиодные экраны, электронные табло привлекают к себе внимание без особого труда, они функционируют за счет светодиодов, работа которых ориентирована на определенные программы. Чаще всего электронное табло используется для отображения различного рода показателей – времени, температуры, даты, атмосферного давления. В ресторанах, барах и кафе на нем отображают приветствия, меню, режим работы. На вокзалах их используют для указания номера поездов и время их отправления. Также из достоинств электронные табло, стоят они относительно недорого и могут иметь различные параметры.

Электронное табло имеет в своем арсенале несколько разных видов:

- многоцветное использование
- одноцветное использование
- применяется на табло в помещениях, транспорте, на улице
- функционирует, как в ограниченном режиме, так и в обширном (память, спецэффекты, анимация и т.д.).

Существуют множество различных типов электронного табло для различного рода деятельности обладающими своими характеристиками и особенностями.

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

## 1.1 Анализ существующих аналогичных устройств

### Рекламная Бегущая строка

Бегущая строка (рис. 1.2) - в своем роде предоставляет медийную рекламу, она же определяется рекламными баннерами (графическими или текстовыми). Реклама с цифровым дисплеем бывает разных форм, но по своей сути она основана на том же принципе. Светодиодная бегущая строка имеет в основе формирования изображения матрицу из светодиодов одного или нескольких цветов. Каждый светодиод в таком табло управляется контроллером по отдельности. Современные бегущие строки обладают впечатляющим функционалом для показа рекламы и объявлений. Наборы красивых шрифтов, большое количество спецэффектов показа рекламы. Даже монохромные, не говоря уже про цветные табло, позволяют выводить не только тексты, но и анимацию, а цветные — фото и даже видео. Продуманное программное обеспечение позволяет программировать вывод рекламных роликов по дням недели и времени суток. В наружном применении, бегущая строка предоставляет собой электронное устройство, предназначенное для отображения большого объема текстовой и графической информации, которая все время бежит горизонтально, двигаясь, справа налево.



Рисунок 1.2 – Бегущая строка

### Табло валют

Табло валют необходимо для информирования о текущих обменных курсах валюты в банках и пунктах обмена валюты. Электронное табло валют выполняется в виде законченного изделия, либо в виде отдельных элементов, встраиваемых в рекламный щит с логотипом банка, вывеску или консоль.

Законченное табло валют (рис.1.3) выполняется в виде щита с расположенными в нем отдельными полями стоимости покупки и продажи валюты. Отдельные поля стоимости валюты отображается символьным табло, а наименование валюты, логотип и адрес пункта обмена валюты или банка - пленкой с подсветкой или без. Контроллер управления табло валют расположен внутри корпуса.

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6



### Табло для АЗС

Табло для АЗС (рис. 1.5) - это стела, имеющая встроенные блоки светодиодных табло, для обозначения цены на топливо в формате «рубль и копейки». В принципе - это стандартное, минималистичное исполнение.

Важным моментом является то, что ценовое табло для АЗС обычно располагается на открытом воздухе, поэтому, оно обязательно должно быть изготовлено из таких материалов, которые смогут успешно противостоять влаге и пыли, вредному воздействию окружающей среды (включая, выцветание от солнца), химическому загрязнению от выхлопных газов автомобилей. Оптимальным вариантом для табло читаются цены для заправочных станций с высотой символа 270 и 350 мм – обеспечивающие видимость с большого расстояния. В модулях для стел АЗС применены супер яркие светодиоды красного, белого, желтого, зеленого или синего цвета свечения, предназначенные для работы при солнечном освещении. В табло реализована система регулировки яркости светодиодов: автоматический ночной и дневной режим работы в зависимости от времени суток и по расписанию, регулируется с ПДУ. В стеле все модули объединяются в единую систему и управляются одним ПДУ. Цвет светодиодов на стоимость табло не влияет.



Рисунок 1.5 – Табло для АЗС

## 1.2 Обзор существующих типов платформ Arduino

Arduino – это инструмент для проектирования электронных устройств (электронный конструктор) более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Это платформа, предназначенная для «physical computing» с открытым программным кодом, построенная на простой печатной плате с современной средой для написания программного обеспечения.

Arduino применяется для создания электронных устройств с возможностью приема сигналов от различных цифровых и аналоговых датчиков, которые могут быть подключены к нему, и управления различными исполнительными устройствами (рис. 1.6).

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

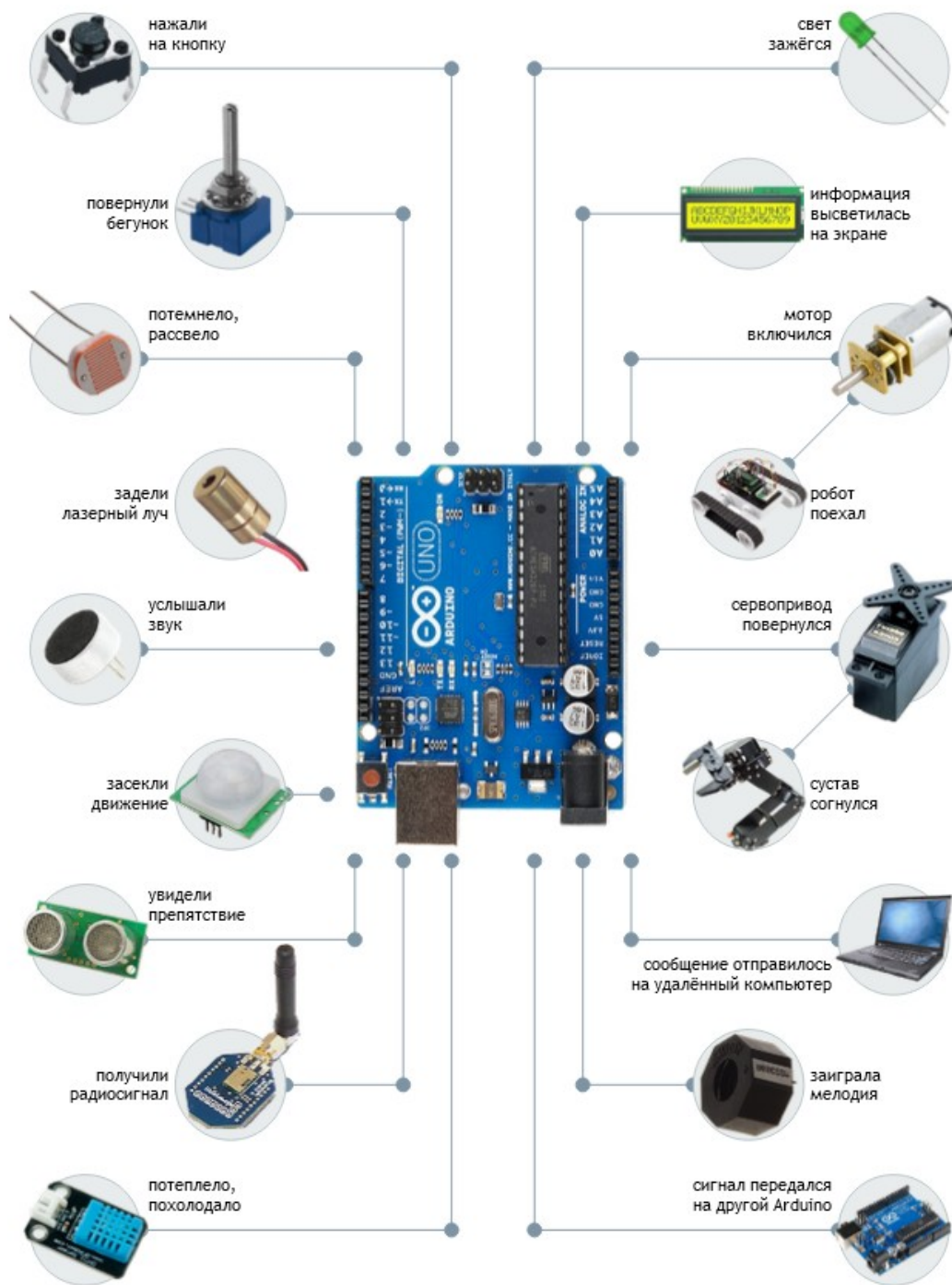


Рисунок 1.6 – первичное представление об Arduino

### Простота и доступность

Платформа Arduino приобрела бешеную популярность благодаря простоте и дружелюбности. Даже полный ноль в программировании и схемотехнике может освоить основы работы с Arduino за пару часов. Этому способствуют тысячи публикаций, учебников, заметок в интернете и отличная серия видео уроков по Arduino (рис. 1.7).

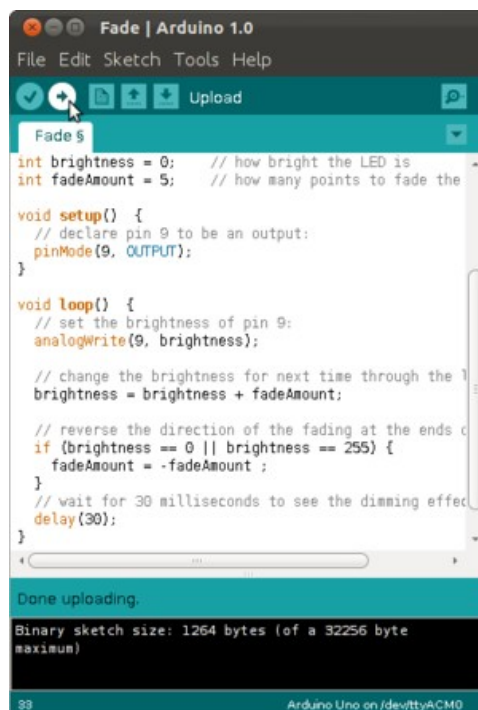


Рисунок 1.7 – Пример кода на Arduino

Программы для Arduino пишутся на обычном C++, дополненным простыми и понятными функциями для управления вводом-выводом на контактах. Если вы уже знаете C++, Arduino станет дверью в новый мир, где программы не ограничены рамками компьютера, а взаимодействуют с окружающим миром и влияют на него. Если же вы новичок в программировании — не проблема, вы с лёгкостью научитесь, это просто. Кросс-платформенность – программное обеспечение Arduino работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows. Также стоит отметить низкую стоимость – платы Arduino относительно дешевы по сравнению с другими платформами. Самая недорогая версия модуля Arduino может быть собрана в ручную, а некоторые даже готовые модули стоят меньше 50 долларов.

Аппаратные средства с возможностью расширения и открытыми принципиальными схемами – микроконтроллеры ATMEGA8 и ATMEGA168 являются основой Arduino. Схемы модулей выпускаются с лицензией Creative Commons, а значит, опытные инженеры имеют возможность создания собственных версий модулей, расширяя и дополняя их. Даже обычные пользователи могут разработать опытные образцы с целью экономии средств и понимания работы.

#### "Голова" Arduino

"Голова" Arduino - это микроконтроллер серии Atmega. Микроконтроллер представляет собой микропроцессор с памятью и различными периферийными устройствами, выполненной на одной микросхеме. По факту это однокристальный микрокомпьютер, который способен выполнять различные задачи. Разные модели из семейства Arduino оснащены разными микроконтроллерами, более и менее мощными и функциональными (рис. 1.8).

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10



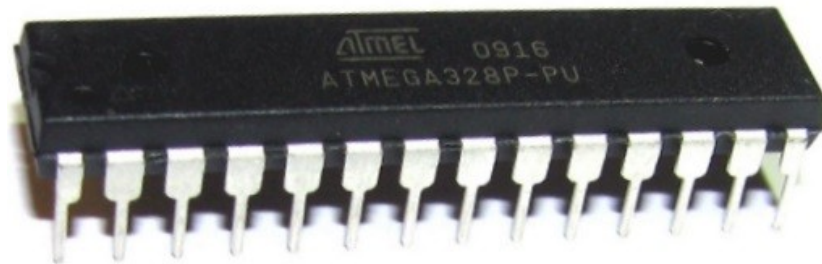


Рисунок 1.8 – Микроконтроллер серии Atmega

Команда Arduino выпустила множество плат, с различным функционалом, количеством входов/выходов, тактовой частотой, оперативной памятью, дополнительными функциями и тому подобным. Ниже перечислены актуальные, на данный момент платформы Arduino, такие, как:

Arduino Uno (рис. 1.9) является стандартной платой Arduino и возможно наиболее распространенной. Она основана на чипе ATmega328, имеющем на борту 32 КБ флэш-памяти, 2 Кб SRAM и 1 Кбайт EEPROM памяти. На периферии имеет 14 дискретных (цифровых) каналов ввода / вывода и 6 аналоговых каналов ввода / вывода, это очень разносторонне-полезные девайсы, позволяющие покрывать большинство любительских задач в области микроконтроллерной техники. Данная плата контроллера является одной из самых дешевых и наиболее часто используемых.



Рисунок 1.9 – Arduino Uno

Arduino UNO имеет следующие характеристики:

- микроконтроллер ATmega328;
- напряжение питания 5В;
- входное напряжение (рекомендуемое) 7-12В;
- входное напряжение (предельное) 6-20В;
- цифровой ввод-вывод 14 линии (из них 6 поддерживают ШИМ);
- аналоговый ввод 6 линий;

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

- постоянный ток на линиях ввода-вывода 40мА;
- постоянный ток на линии 3.3В 50мА;
- Flash-память 32кб, 0.5 кб из них использованы для загрузчика;
- SRAM-память 2кб;
- EEPROM-память 1кб;
- тактовая частота 16МГц.

Arduino Leonardo (рис. 1.10) - контроллер на базе ATmega32u4. Платформа имеет 20 цифровых вход/выходов (7 из которых могут использоваться как выходы ШИМ и 12 как аналоговые входы), кварцевый генератор 16 МГц, разъем микро-USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

В отличие от всех предыдущих плат ATmega32u4 имеет встроенную поддержку для USB соединения, это позволяет задать как Leonardo будет виден при подключение к компьютеру, это может быть клавиатура, мышь, виртуальный серийный / COM порт.



Рисунок 1.10 – Arduino Leonardo

Arduino Leonardo имеет следующие характеристики

- Микроконтроллер ATmega32u4
- Рабочее напряжение 5В
- Входное напряжение (рекомендуемое) 7-12 В
- Входное напряжение (предельное) 6-20 В
- Цифровые Входы/Выходы 20 (7 из которых могут использоваться как выходы ШИМ)
- Аналоговые каналы 12
- Постоянный ток через вход/выход 40 мА
- Постоянный ток для вывода 3.3 В 50 мА
- Флеш-память 32 Кб (ATmega32u4) из которых 4 Кб используются для загрузчика

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

- ОЗУ 2 Кб (АТmega32u4)
- EEPROM 1 Кб (АТmega32u4)
- Тактовая частота 16 МГц

#### Arduino Nano

Arduino Nano (рис. 1.11) – это отладочная плата небольшого размера, которая входит в тройку лидеров по популярности среди радиолюбителей-программистов. Несмотря на свой скромный размер, она практически ничем не уступает нашумевшей - Arduino Uno по функционалу и может использоваться в проектах, где габариты играют существенную роль. Платформа Nano, построенная на микроконтроллере АТmega328 (Arduino Nano 3.0) или АТmega168 (Arduino Nano 2.x), Она имеет схожую с Arduino Duemilanove функциональность, однако отличается сборкой. Отличие заключается в отсутствии силового разъема постоянного тока и работе через кабель Mini-B USB. Nano разработана и продается компанией Gravitech.

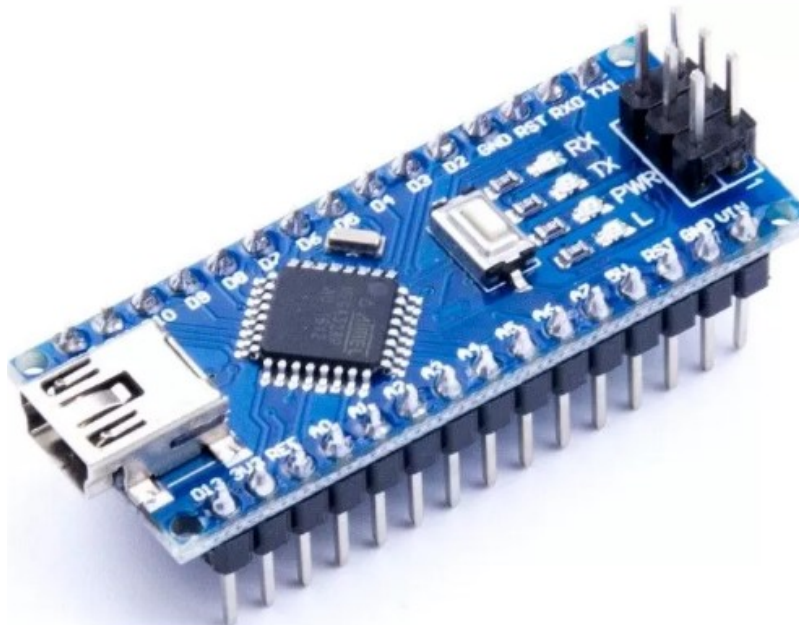


Рисунок 1.11 – Arduino Nano

Nano имеет такие характеристики

- Микроконтроллер Atmel АТmega168 или АТmega328
- Рабочее напряжение (логическая уровень) 5 В
- Входное напряжение (рекомендуемое) 7-12 В
- Входное напряжение (предельное) 6-20 В
- Цифровые Входы/Выходы 14 (6 из которых могут использоваться как выходы ШИМ)
- Аналоговые входы 8
- Постоянный ток через вход/выход 40 мА
- Флеш-память 16 Кб (АТmega168) или 32 Кб (АТmega328) при этом 2 Кб используются для загрузчика
- ОЗУ 1 Кб (АТmega168) или 2 Кб (АТmega328)

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

- EEPROM 512 байт (ATmega168) или 1 Кб (ATmega328)
- Тактовая частота 16 МГц
- Размеры 1.85 см x 4.2 см

#### Arduino Mega

Arduino Mega (рис. 1.12) построена на микроконтроллере ATmega2560. Плата имеет: 54 цифровых входа/выходов (14 из которых могут использоваться как выходы ШИМ), 16 аналоговых входов, 4 последовательных порта UART, кварцевый генератор 16 МГц, USB коннектор, разъем питания, разъем ICSP и кнопка перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или аккумуляторной батареей.

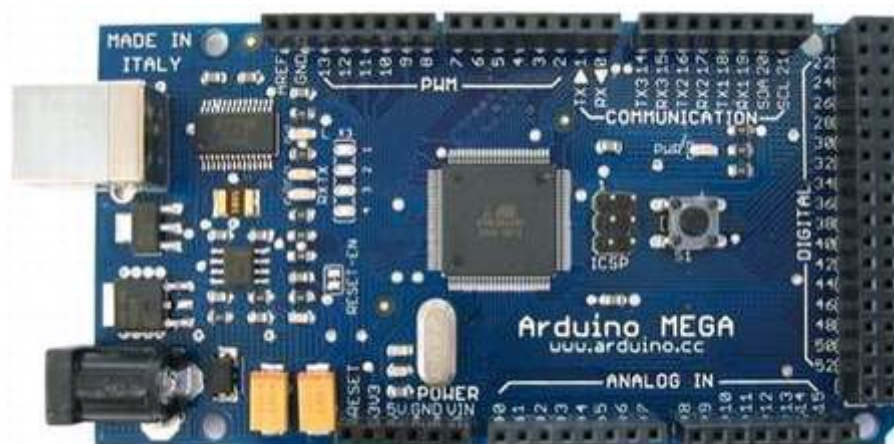


Рисунок 1.12 – Arduino Mega

Arduino Mega имеет следующие характеристики

- Микроконтроллер: ATmega2560
- Тактовая частота: 16 МГц
- Напряжение: 5 В
- Предельные напряжения: 5-20 В
- Рекомендуемое напряжение питания: 7-12 В
- Макс. сила тока с одного вывода: 40 мА
- Цифровые пины: 54
- Цифровые пины с поддержкой ШИМ: 15
- Аналоговые входы: 16
- Flash-память: 256 КБ (8 из них используются загрузчиком)
- SRAM: 8 КБ
- EEPROM: 4 КБ

#### Arduino Mini

Arduino Mini (рис. 1.13) построена на микроконтроллере ATmega168 (технические данные) и предназначена для использования в лабораторных работах и проектах, где пространство является критическим параметром. Платформа содержит 14 цифровых входов и выходов (6 из которых могут использоваться как выходы ШИМ), 8 аналоговых входов и кварцевый

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

генератор 16 МГц. Программируется при помощи адаптера Mini USB или любого преобразователя USB или RS232 в TTL.



Рисунок 1.13 – Arduino Mini

#### Характеристики Arduino Mini

- Микроконтроллер – ATmega168/328;
- Рабочее напряжение – 3.3В/5В;
- Напряжение питания – 3.35-12В/5 - 12В;
- Цифровые входы/выходы – 14;
- Аналоговые входы – 8;
- Flash-память – 16/32 КБт;
- SRAM – 1/2 КБт;
- EEPROM – 512/1024 байт;
- Тактовая частота – 8/16 МГц;
- Размеры – 33x18 мм;
- Вес – 5 г.

#### 1.3 Обзор элементов автоматизации

Для работы любого механического проекта нужны элементы автоматизации. Автоматизация в современном обществе – необходимая мера, в цифровом веке крайне важно исключить человеческий фактор в различных производствах, чтобы стандартизировать и улучшить качество продукции. Существуют и сферы, где человеку просто не подвластно делать то, на что способны роботы, например, производство нано-материалов и микроплат.

Но до глобальной автоматизации человечеству, пока еще далеко и все же взаимодействие между техникой и человеком все так же востребованы, поэтому и существуют устройства как: ввода, различные кнопки, рычаги, так и устройства вывода - световые индикаторы (лампочки), различные звуковые сигнализаторы и наконец дисплеи.

Дисплеи можно разделить на:

- семисегментные (такие, как на цифровых часах);

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

- алфавитно-цифровые;
- графические;

Своё название семисегментные индикаторы получили в связи с тем, что изображение символа формируется с помощью семи отдельно управляемых (подсвечиваемых светодиодами) элементов - сегментов. Эти элементы позволяют отобразить любую цифру 0..9, а также некоторые другие символы, например: '-', 'A', 'b', 'C', 'd', 'E', 'F' и другие. Это даёт возможность использовать индикатор для вывода положительных и отрицательных десятичных и шестнадцатеричных чисел и даже текстовых сообщений. Обычно индикатор имеет также восьмой элемент - точку, используемую при отображении чисел с десятичной точкой. Сегменты индикатора обозначают буквами a, b, ..., g (a - верхний элемент, далее буквы присваиваются сегментам по часовой стрелке; g - центральный сегмент; dp - точка).

В настоящее время для отображения информации всё чаще используются графические дисплеи, однако, семисегментные индикаторы также не утратили своего значения. Если требуется лишь отображение чисел, то они могут стать более предпочтительным вариантом, т.к. просты в управлении и могут использоваться совместно с любым микроконтроллером с достаточным количеством выводов.

Поэтому семи сегментные индикаторы используют только для отображения простейших сообщений (рис. 1.14).



Рисунок 1.14 – Семисегментный индикатор

Алфавитно-цифровые дисплеи (рис 1.15) представляют собой активные устройства отображения информации, предназначенные для организации диалога оператора с ЭВМ.

По степени сложности и полноте выполняемых функций алфавитно-цифровые дисплеи делятся на простые с аппаратной реализацией основных функций и интеллектуальные дисплеи, в которых функции обработки информации выполняется программно-управляемым дисплейным процессором



Рисунок 1.15 – Алфавита-Цифровой дисплей

Графический дисплей – дисплей, обеспечивающий создание на экранах матриц точек, высвечивающих изображения и тексты.

Графические дисплеи находят широкое применение в системах управления технологическими процессами, контрольно-измерительной, медицинской, телекоммуникационной аппаратуре, устройствах ограничения доступа и многих других приложениях.

Графический дисплей размером 64x16 см. показан на рисунке 1.16

Одна из наиболее популярных бюджетных модификаций графических дисплеев малого размера. Чаще всего монтируется в окне и работает на улице или же эксплуатируется в торговых помещениях расположенных на крытых рыночных либо выставочных площадях.

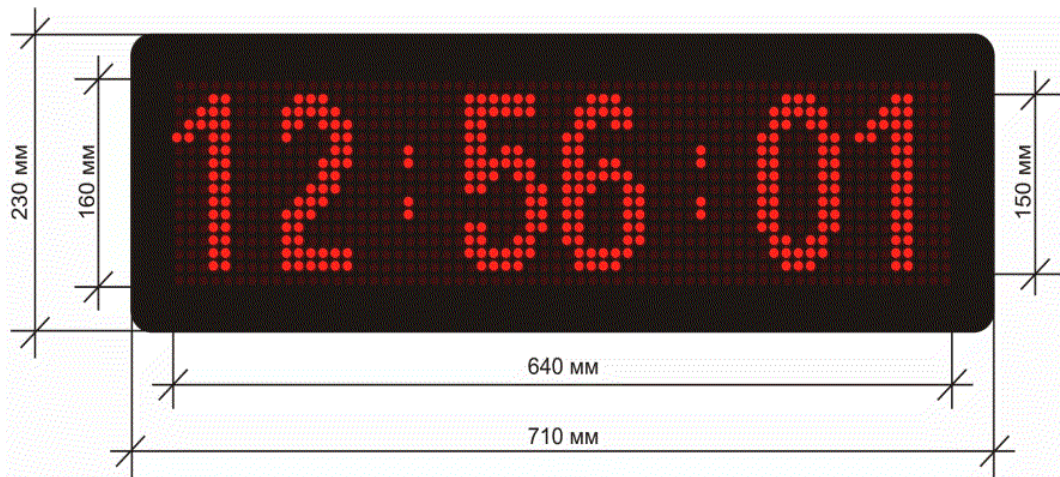


Рисунок 1.16 – Графический дисплей

К счастью, инженеры давно уже разработали специализированные микросхемы для управления разного рода индикаторами. В этом проекте я рассмотрел матричный модуль с микросхемой MAX7219.

MAX7219 (рис. 1.17) – микросхема для управления 7-сегментными индикаторами и матрицами 8x8. Благодаря встроенной динамической индикации и настройке тока этот чип в десятки раз упрощает работу и с теми, и с другими

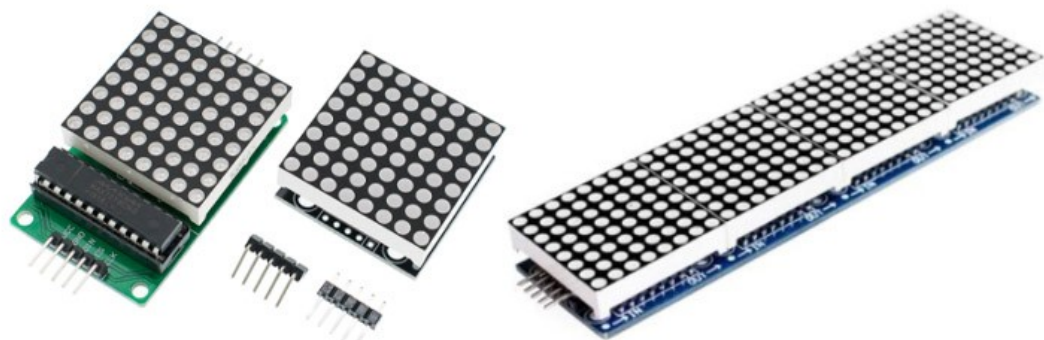


Рисунок 1.17 – Матрица MAX7219

Модуль или адаптер Bluetooth представляет собой миниатюрное (размером не более флешки) электронное устройство, предназначенное для дистанционной передачи информации между компьютерами, внешними устройствами, смартфонами и их аксессуарами. Он часто используется для удалённого управления (например, беспроводными наушниками, телевизором и бытовыми приборами).

На фоне остальных протоколов Bluetooth выделяется, прежде всего, помехоустойчивостью и простотой. От не менее распространённого Wi-Fi, Bluetooth отличается в первую очередь низким энергопотреблением, что делает его доступным для автономных устройств. Благодаря этому Bluetooth получил повсеместное распространение.

Чаще всего для самоделок используются Bluetooth -модули HC-05 и HC-06. Их легко найти в продаже. HC-05 отличается от HC-06 тем, что ему доступны оба режима работы: ведомый (master) и ведущий (slave). А вот HC-06 работает только ведомым, то есть он не способен находить другие устройства и самостоятельно устанавливать с ними связь, один из модулей показан на рисунке 1.18.

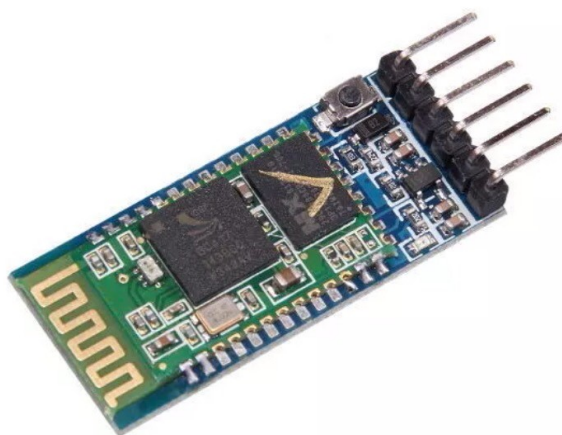


Рисунок 1.18 – Bluetooth -модуль HC-05

Также существует аналог Bluetooth - модулю HC-05 и HC-06, это Bluetooth модуль JDY-3. Модуль JDY-31(рис 1.19) позволяют изменять режим работы slave/master с помощью AT-команд, т.е. может работать как ведомым SLAVE, так и MASTER который способен находить другие устройства и самостоятельно устанавливать с ними связь.

						1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата			18



Основные характеристики модуля JDY-31:

- Рабочая частота: 2.4 GHz
- Интерфейс: UART
- Напряжение питания: 3.6.. 6V
- Ток потребления: 5 мА в режиме поиска, ~8 мА в режиме передачи
- Логический уровень: 3.3V, но вроде как не боится 5V
- Дальность связи: 30м
- Версия Bluetooth: 3.0 SPP

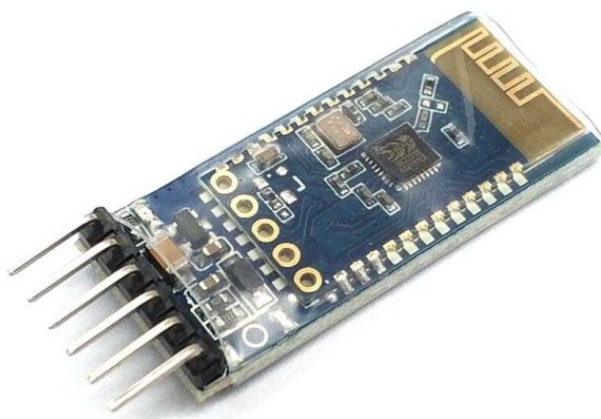


Рисунок 1.19 – Модуль JDY-31

В Arduino проектах требуется питать разные устройства разным уровнем напряжения. Если использовать стабилизатор, то при высоких значениях тока схема будет иметь заметно сниженное КПД. А при использовании нескольких источников питания схема приобретёт громоздкий и неэстетичный вид. Именно для этих случаев лучше использовать DC-DC преобразователи напряжения.

Понижающий преобразователь – это важная часть в любом проекте Arduino, поскольку он позволяет добиться заданных значений напряжения для питания любых компонентов, преобразуя различное входное напряжение

Понижающий модуль постоянного тока на базе LM2596S позволяет преобразовать входящее напряжение от 3.6 – 46 В до 1.25 – 35 В с высоким КПД (до 90%). Сила тока на входе — до 3А. Предельная нагрузка – до 15 Вт, его можно посмотреть на рисунке 1.20

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19



## 2 Специальная часть

### 2.1 Разработка структурной схемы

Для дальнейшей разработки устройства, была разработана структурная схема устройства, показана на рисунке 2.22

Проектируемый поворотно-остановочный сигнал содержит следующие компоненты и выполняют свои функции:

- Питание автомобиля, обеспечивает питанием платформу
- Bluetooth модуль, для связи с мобильным телефоном
- Платформа Arduino Nano с микроконтроллером ATmega328, выполняет команды
- Преобразователь DC-DC Converter, преобразует напряжение
- Табло 16x64, используется для отображения результата

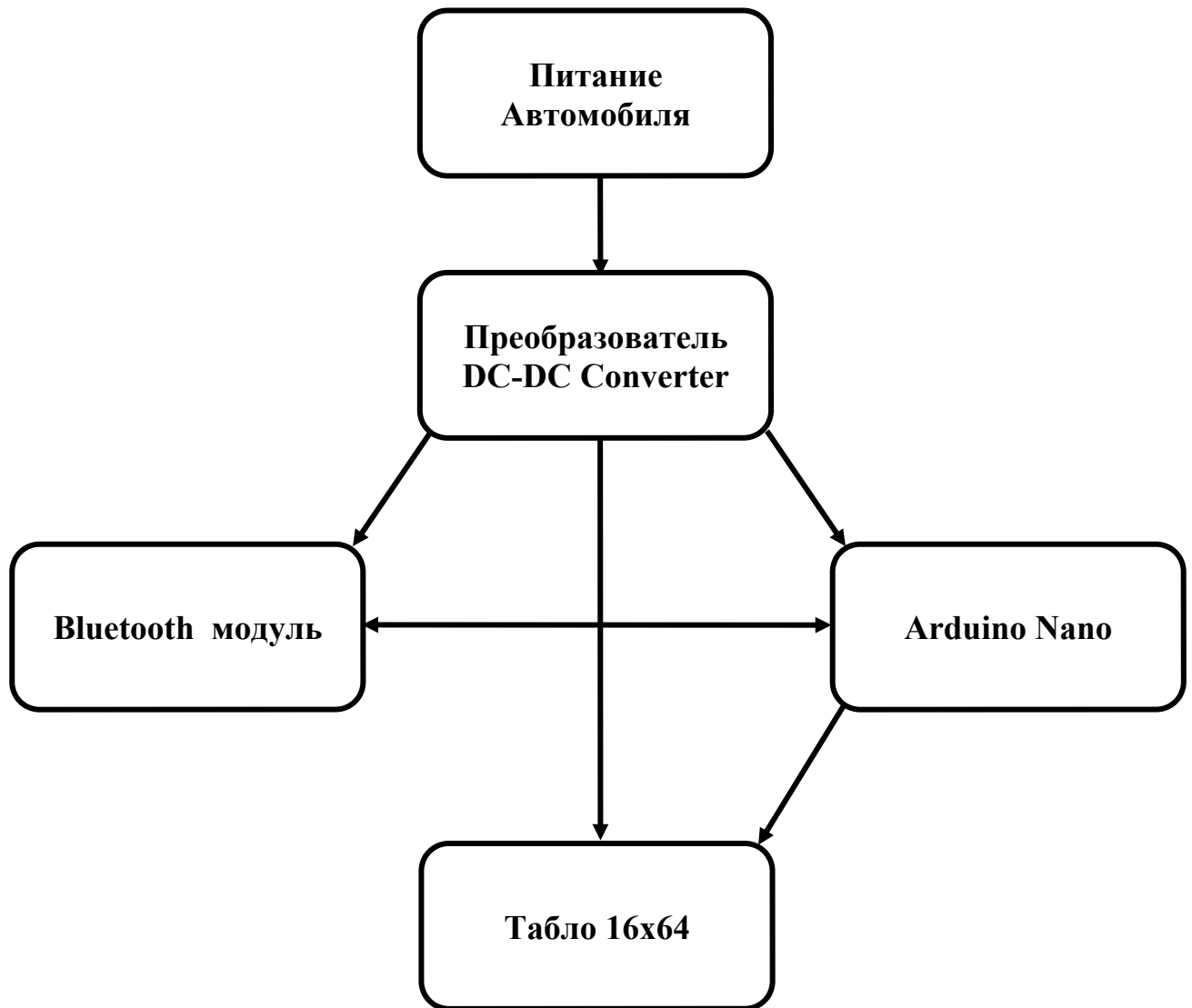


Рисунок 2.22 - Структурная схема

## 2.2 Обоснование выбора элементной базы

На из важнейших этапов разработки любого устройства это выбор элементной базы, так как от правильного выбора зависит будет-ли в конечной итоге устройство отвечать всем требованиям, предъявляемый к ней.

На сегодняшних день схожих себе помощников еще не существует. Для этого мне пришлось детально подойти к реализации проекта.

Платформа Arduino Nano(рис. 2.23), это одноплатный контроллер с открытыми начальными кодами, который возможно использовать во множестве различных приложений. Это - самый простой и наиболее дешевый вариант из микроконтроллеров. В платах Arduino используются либо микроконтроллеры ATmega168, либо его ближайший собрат микроконтроллер ATmega328P, а в некоторых из версий имеется интерфейс USB. Обладают шестью или более выводами аналоговых входов и четырнадцатью или более выводами цифровых входов и выходов, использующихся для возможности подключения к микроконтроллеру датчиков, различных приводов и иных периферийных схем.

В Arduino Nano есть восстанавливаемые предохранители, защищающие USB-порт компьютера от коротких замыканий и перегрузок. Несмотря на то, что большинство компьютеров имеют собственную защиту, такие предохранители обеспечивают дополнительный уровень защиты. Если от USB-порта потребляется ток более 500 мА, предохранитель автоматически разорвет соединение до устранения причин короткого замыкания или перегрузки.

ATmega328P поддерживается различными программными средствами и интегрированными средствами разработки, такими как компиляторы C++, макроассемблеры, программные симуляторы, внутрисхемные эмуляторы и ознакомительные наборы.

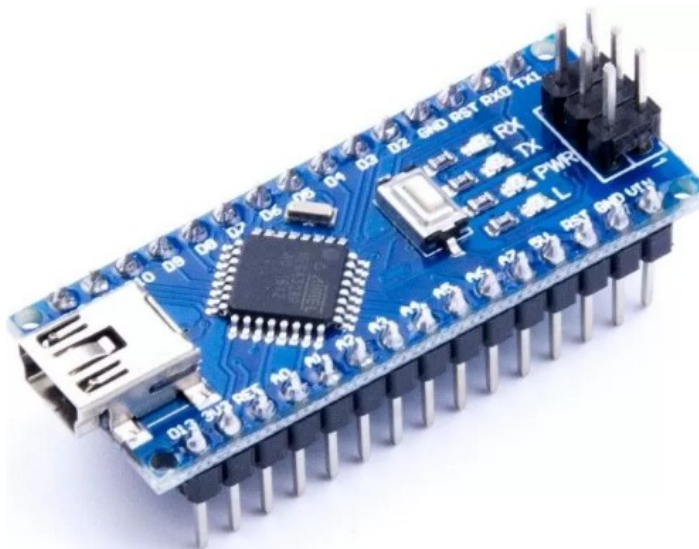


Рисунок 2.23 – Arduino Nano

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

Основные преимущества Arduino Nano:

- Отличная платформа для начинающих.
- Крошечный размер делает его идеальным для компактных проектов.
- Функционально так же, как их более крупные аналоги.
- удобная среда разработки;
- возможности расширения;
- широкая периферия.
- открытый исходный код;

Платформа имеет штырьковые контакты, что позволяет легко устанавливать её на breadboard. Используйте Arduino Nano там, где важна компактность, а возможностей Arduino Mini либо не достаточно, либо не хочется заниматься пайкой.

Характеристики:

- Микроконтроллер: ATmega328P;
- Тактовая частота: 16 МГц;
- Напряжение логических уровней: 5 В;
- Входное напряжение питания: 7...12 В
- Портов ввода-вывода общего назначения: 22;
- Портов с поддержкой ШИМ: 6;
- Портов, подключённых к АЦП: 8;
- Flash-память: 32 КБ;
- Габариты: 18×45 мм.

Распиновка платформы Arduino Nano(рис 2.24)

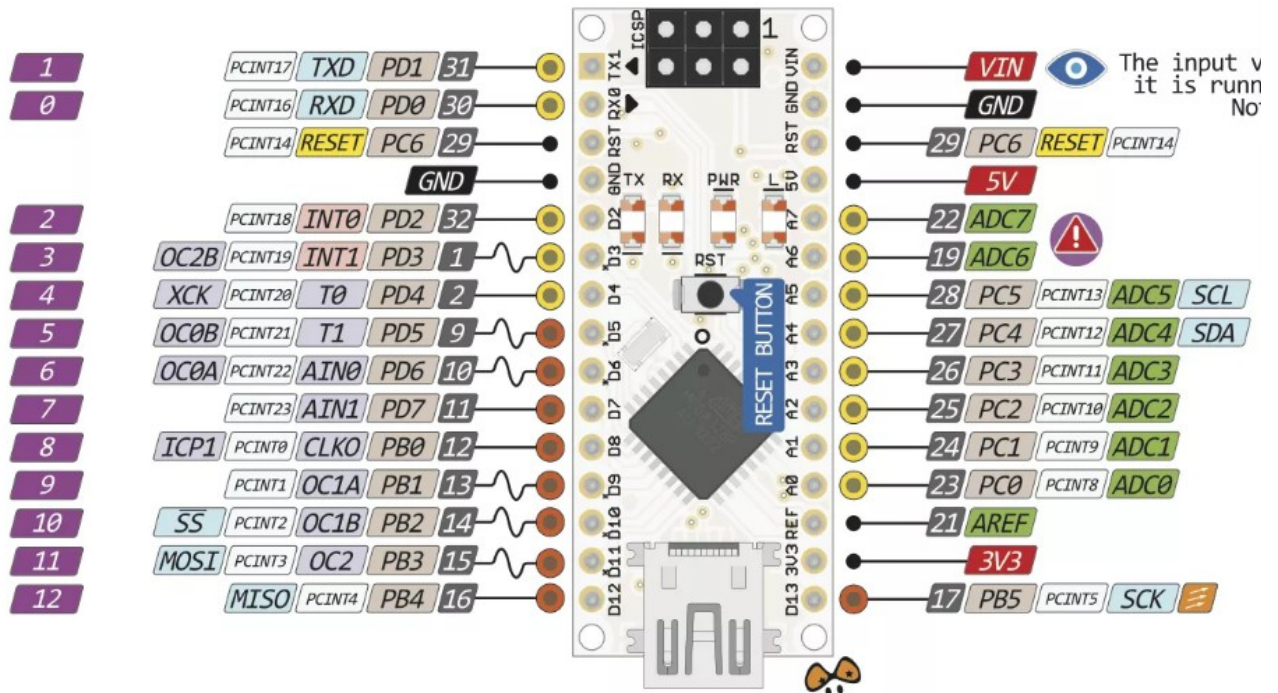


Рисунок 2.24 – Распиновка платформы Arduino Nano

Обозначение выводов модуля показана на рис. 2.25

Пины питания:

- VIN: Входной пин для подключения внешнего источника питания с напряжением в диапазоне от 7 до 12 вольт.
- 5V: Выходной пин от регулятора напряжения на плате с выходом 5 вольт и максимальных током 800 мА. Питать устройство через вывод 5V не рекомендуется — вы рискуете спалить плату.
- 3.3V: Выходной пин от стабилизатора микросхемы FT232R с выходом 3,3 вольта и максимальных током 50 мА. Питать устройство через вывод 3V3 не рекомендуется — вы рискуете спалить плату.
- GND: Выводы земли.
- AREF: Пин для подключения внешнего опорного напряжения АЦП относительно которого, происходят аналоговые измерения при использовании функции analogReference() с параметром «EXTERNAL».

Порты ввода/вывода

- Цифровые входы/выходы: пины 0–13 Логический уровень единицы — 5 В, нуля — 0 В. Максимальный ток выхода — 40 мА. К контактам подключены подтягивающие резисторы, которые по умолчанию выключены, но могут быть включены программой.
- ШИМ: пины 3,5,6,9,10 и 11 Позволяет выводить аналоговые значения в виде ШИМ-сигнала. Разрядность ШИМ не меняется и установлена в 8 бит.
- АЦП: пины A0–A7 Позволяет представить аналоговое напряжение в цифровом виде. Разрядность АЦП не меняется и установлена в 10 бит. Диапазон входного напряжения от 0 до 5 В. При подаче большего напряжения — вы убьёте микроконтроллер.
- TWI/I<sup>2</sup>C: пины A4(SDA) и A5(SCL) Для общения с периферией по интерфейсу I<sup>2</sup>C. Для работы используйте библиотеку Wire.
- SPI: пины 11(MOSI), 12(MISO), 13(SCK) и 10(SS) Для общения с периферией по интерфейсу SPI. Для работы — используйте библиотеку SPI.
- UART: пины 0(RX) и 1(TX) Используется для коммуникации платы Arduino с компьютером или другими устройствами по последовательному интерфейсу

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24



Рисунок 2.25 Обозначение выводов модуля

Элементы платы (рис 2.26).

Микроконтроллер ATmega328P

Сердцем платформы Arduino Nano является 8-битный микроконтроллер семейства AVR — ATmega328P с тактовой частотой 16 МГц. Контроллер предоставляет 32 КБ Flash-памяти для хранения прошивки, 2 КБ оперативной памяти SRAM и 1 КБ энергонезависимой памяти EEPROM для хранения данных.

Микросхема FT232R

Микросхема FTDI FT232R обеспечивает связь микроконтроллера ATmega328P с USB-портом компьютера. При подключении к компьютеру Nano определяется как виртуальный COM-порт.

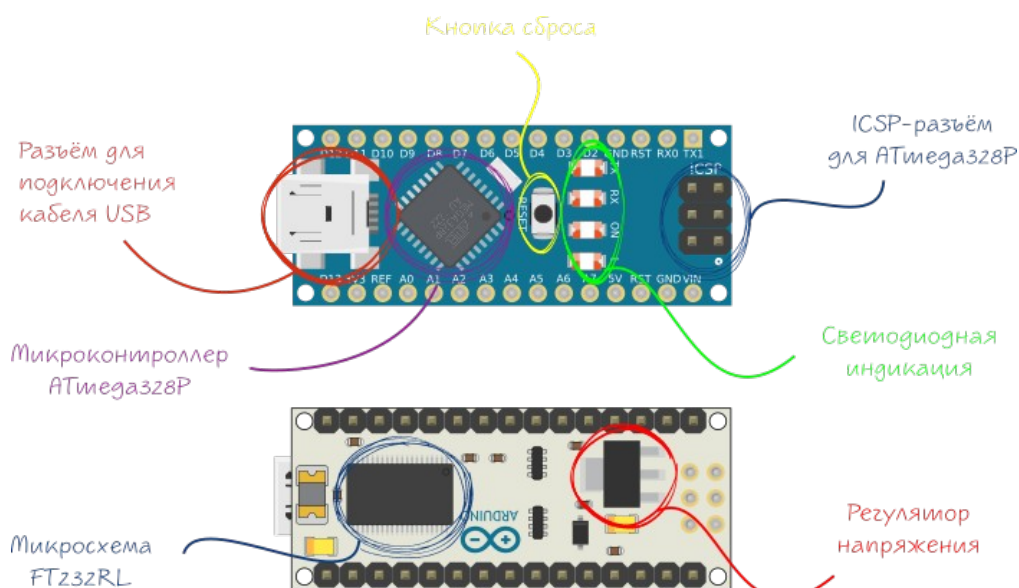


Рисунок 2.26 – Элементы платы





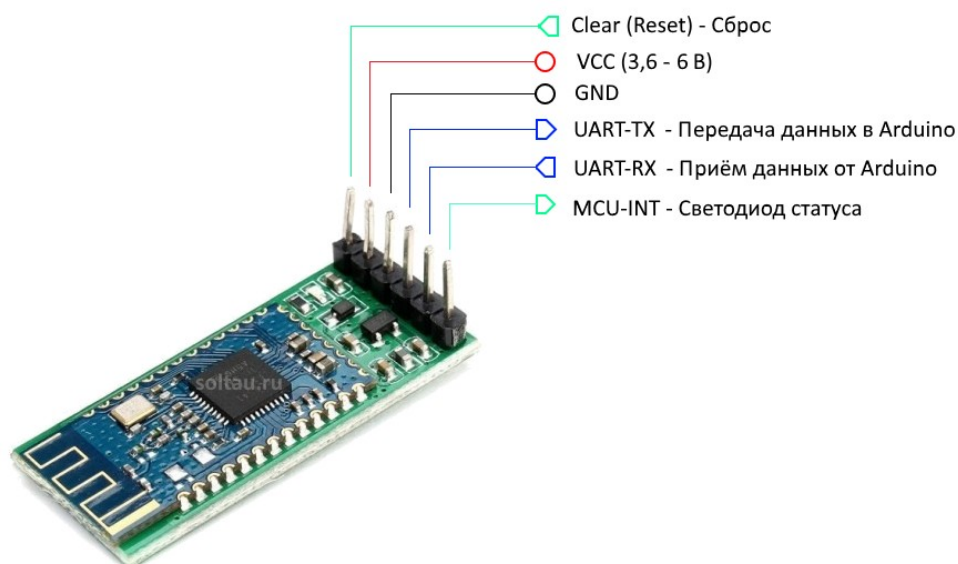


Рисунок 2.28 – Bluetooth HC 06

Контакты модуля:

- VCC , GND - плюс и минус питания;
- RX и TX - приемник и передатчик;
- MCU-INT —выводит статус;
- Clear (Reset) - сбрасывание и перезагрузка модуля. Последние два

вывода обычно не задействованы в работе, поэтому сейчас производятся модули без этих контактов.

- UART-TX - Передача данных Arduino
- UART-RX - Прием данных Arduino

По итогам выбора Bluetooth модуля свойства и характеристики данного модуля с лихвой хватят мне для реализации устройства.

И для того, чтоб устройство безошибочно работало, использую преобразователь (DC-DC Converter), показан 2.29 на рисунке

Высокоэффективный импульсный источник питания, обладающий малыми габаритами.

Многооборотный переменный резистор расположенный на плате, позволяет максимально точно настроить требуемое выходное напряжение. Переменный резистор, позволяет установить максимальный ток защиты.

Высокий КПД импульсного преобразователя позволяет при минимальных размерах получить высокий выходной ток.

Характеристики преобразователя

- Диапазон входного напряжения: 4-38В (38В — максимальный предел)
- Диапазон выходного напряжения: 1.25V-36В (понижающий режим, вход должен быть больше, чем на выходе на 1,5 и более вольт)
- Выходной ток: 4.5А непрерывный, 5А — непродолжительное время.
- Максимальная выходная мощность: 75 Вт (требуется дополнительное охлаждение), 50 Вт — с компактным радиатором охлаждения на чип или при кратковременном использовании.

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

- Защита от перегрева и от короткого замыкания.
- Защиты от переплюсовки входного напряжения — нет.
- Эффективность преобразования: до 96%
- IN +: вход Положительный; IN-: вход отрицательный;
- OUT +: выход Положительный; OUT-: выход отрицательный
- Индикатор входного / выходного напряжения — 4 разряда  
(необходимо входное напряжение не менее 4,5 В, максимальное измеряемое напряжение — 40 В).
- Погрешность индикации — 0,05 В.
- Частота преобразования — 180 кГц.



Рисунок 2.29 DC-DC Преобразователь

### 2.3 Разработка электрической принципиальной схемы

Принципиальная схема, схема в которой каждая деталь отображается условно-графическим обозначением (УГО). Принципиальные схемы позволяют понять, как работает устройство, как его детали соединены друг с другом. Кроме того, схема электрическая принципиальная, является исходным заданием для конструктора.

Схема была разработана в онлайн сервисе EasyEDA и представлена на рисунке 2.30

EasyEDA - кросс-платформенная, веб-ориентированная среда автоматизации проектирования электроники включающая в себя редактор принципиальных схем, редактор топологии печатных плат, также средства заказа изготовления печатных плат

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

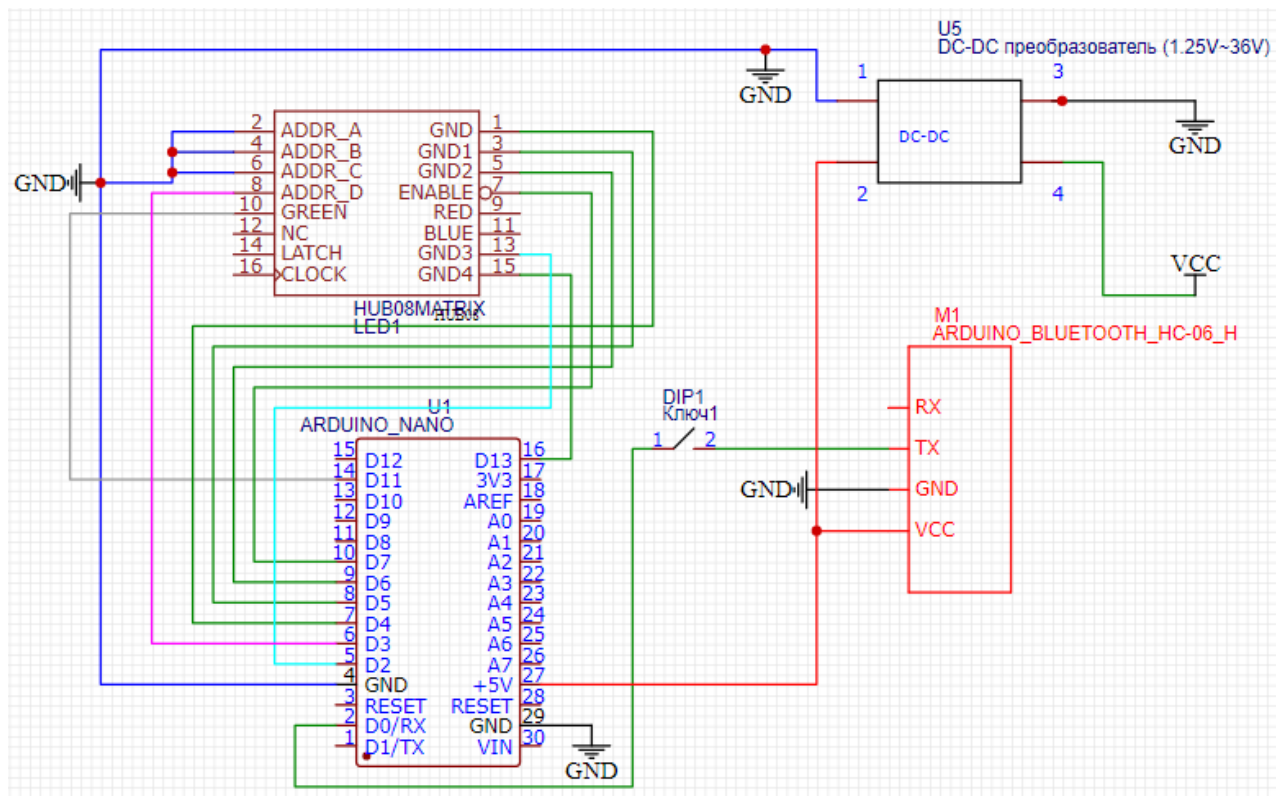


Рисунок 2.30 - Принципиальная схема

## 2.4 Написание программы для устройства

Программирование микроконтроллеров Arduino осуществляется на языке программирования C++. Этот язык является низкоуровневым, поэтому считается сложным и имеет высокий порог вхождения. Но для программирования Arduino используется упрощенная версия этого языка программирования. Так же для упрощения разработки прошивок существует множество функций, классов, методов и библиотек. Благодаря этому работать с этими микроконтроллерами очень удобно и легко.

Программный код для данного устройства представлен в приложении А.

### Заключение

В ходе данной курсовой работе я изучил основы работы с Arduino, также познал технологию создания программ на данной платформе.

Результатом курсового проекта является разработанный проект поворотного-остановочного сигнала на базе Arduino Nano. Также были разработаны структурные и принципиальные схемы и программа на языке Arduino.

Благодаря минимальному количеству элементов в устройстве, а также их дешевизны, и знаниям по микропроцессорным системам и базовых знаний в электротехнике, то данное устройство очень просто в реализации. Благодаря несложному, удобному и настраиваемому управлению всего устройства, пользователь может легко его использовать.

В течение выполнения курсового проекта был приобретен опыт работы и умение проектировать цифровые микропроцессорные устройства.

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

### Список использованных источников

- Гуров, В. В. Микропроцессорные системы: учебник / В.В. Гуров. — Москва: ИНФРА-М, 2019. — 336 с. + Доп. материалы [Электронный ресурс; Режим доступа: <http://new.znaniium.com>]. — (Среднее профессиональное образование).
- Булатов В.Н. Микропроцессорная техника. Схемотехника и программирование [Электронный ресурс]: учебное пособие для СПО/ Булатов В.Н., Худорожков О.В.— Электрон. текстовые данные.— Саратов: Профобразование, 2020.— 376 с.— Режим доступа: <http://www.iprbookshop.ru/91893.html>.— ЭБС «IPRbooks».
- Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст: электронный. - URL: <https://znaniium.com/catalog/product/1172261>
  - [https://ampermarket.kz/base/arduino\\_family/?ysclid=lahyb6h5lp247647430](https://ampermarket.kz/base/arduino_family/?ysclid=lahyb6h5lp247647430)
  - <http://wiki.amperka.ru/продукты:arduino-nano>

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		31

## Приложение А. Листинг программы

```

#include <avr/pgmspace.h>
#include <SPI.h>
#include "HUB08SPI.h"
#include <TimerOne.h>
//выбираем шрифт
//#include "Font_RUS1.h" //English широкий + русский узкий
#include "Font_RUS2.h" //English средний + русский средний
//#include "Font_RUS3.h" //English средний + русский узкий
#define WIDTH 64
#define HEIGHT 16
#include <EEPROM.h> // подключаем библиотеку EEPROM
uint8_t displaybuf[WIDTH * HEIGHT / 8];

HUB08SPI display;
//далее рисунок на дисплей меняем под себя в приложении EXCEL
//если не нужен декларируем три строчки ниже в void setup()
/* стереть справа -> */
uint8_t display_risunok[WIDTH * HEIGHT / 8] = {...}; //*/
unsigned long time_2znaka=0, time_telf=0, time_inver=0, time_povrt=0;
//временные таймеры
boolean tormaz=0, nazad=0, pravo=0, levo=0, avariik=0, rezt=0, //флаги //
flag_tel=0, flag_len=0, new_fraza=0, flag_nazd, flag_prv=0, flag_stop=0; //флаги //
uint8_t len=0, invers_N=0, A=3, T=4; //переменные, менять тут ничего не надо

int length = WIDTH; //переменная для длины фраз
String Str_telf; //объект для конвертирования массива телефонных фраз
char * msg_tel; //резервируем память под массив фраз с телефона
char * msg_avto[] = // буквы "Е" нет! // массив автомобиля
{////////сервисные фразы. ////////////:
"\x0c6\x0c7", //0// знак !/
"\x0c0\x0c1", //1// знак шины "Ш"// или, что то своё
"\x0c6\x0c7\x020\x020\x020\x0c6\x0c7", //2// 2 знака "/! //
"\x0c2\x0c3\x0c4\x0c5", //3// СТОП крупным шрифтом
"НАЗАД", //4//
"СПАСИБО" //5//
};
//-----предварительные настройки-----
// Библиотека увеличивает частоту Timer2 так что ШИМ работает на 32 кГц. время в
скетче бежит быстрее :)
#define time_invers 1000 // время между инверсиями СТОП.//с телефона не меняется
#define time_povorot 800 // время между морганиями поворотов//с телефона НЕ
меняется//зависит от авто, подобрать при необходимости
#define time_spasibo 8000 // время горения "спасибо" //с телефона НЕ меняется
unsigned long time_sbros = 15000; //время горения фраз с телефона///потом меняем с
телефона
uint16_t Skorost=150; //от 30 до 300//0-быстро, 700 медленно//потом меняем с телефона
uint8_t Jarkost=250; //0-256//256 max//потом с телефона меняем
//-----пины приема сигналов с машины (аналоговые входы)-----
#define trm A1 //тормоз // присваиваем пины входов от ламп машины
#define nzd A2 //назад //

```

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

```

#define prv A4 //вправо //
#define lev A3 //налево //
//-----переменные для хранения в EEPROM -----
// тут тип переменной "int" для корректной работы функции "sscanf()
// все меняется с телефона потом
//обязательно хоть раз "сохранить" нажать, а то в памяти(до первой записи) может быть
мусор
int adrees=0;//адрес ардуино, не меняется//29284 прописано в программе телефона (от
фонаря придумал)
int skorost=70;//скорость(10-99)//99 быстро 10-медленно//
int jarkost=70;//яркость(10-99)
int invers=6;// инверсия//0-9
int reklam=1; // режим когда фразы не уходят сами//1-уходят,2- постоянно горит последняя
фраза, до выкл табло
int vremya=15;//время горения фраз с телефона// сек//2-99
int soxran=1;//сохранить еером//1-нет//2- да
int begun=1;// бежит//1-нет//2- да//
int shipu=1;//1-знак !/, 2-буква "Ш"

//
=====
=====//
void setup()
{ Serial.begin(9600); // для блютуз CH-06
// Serial.begin(38400);// для блютуз CH-05
//-----подключаем входы от машины -----
digitalWrite(trm,LOW); // тормоз//
digitalWrite(nzd,LOW); // назад // ВКЛ ВНУТРЕННИЕ РЕЗИСТОРЫ
digitalWrite(prv,LOW); // поворот право
digitalWrite(lev,LOW); // поворот лево//*/

//-----чтение сохраненных переменных их энергонезависимой памяти
-----
skorost=EEPROM.read(51); jarkost=EEPROM.read(52); reklam=EEPROM.read(53);
vremya=EEPROM.read(54); begun = EEPROM.read(55); shipu=EEPROM.read(56);
invers=EEPROM.read(57);
// применяем сохраненные переменные
Skorost=map(skorost, 10, 99, 300, 30); //инвертирует диапазон , после новой команды.
Jarkost=map(jarkost, 10, 99, 40, 255); //инвертирует диапазон , после новой команды
time_sbros=(long)vremya*1000; /// переводим сек в млсек*/
//-----прерывания-----
Timer1.initialize(200); //частота обновления
Timer1.attachInterrupt(refresh); //скорость обновления
//-----печатаем рисунок---если не нужен рисунок, то закомментировать 3 строки
ниже-----
/*стереть справа -> */
display.begin(display_risunok, WIDTH, HEIGHT);//
delay(1000); //рисунок горит 1(1000мс) секунд при старте
display.clear();//чистим дисплей */
//-----
display.begin(displaybuf, WIDTH, HEIGHT);//выводим из функции рисунок, прописываем
в функцию буфер для текста

```

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

```

}

//-----
volatile uint16_t Nx = 0; //переменная позиции печати на табло//не разрешаем
оптимизировать переменную//

void refresh() //функция обновления строки,(прерывания таймера)
{ static uint16_t count = 0; count++;
if (count > Skorost)
{ count= 0; Nx++;
if(Nx > length) Nx = 0; //reset после xxx pixels
}
display.scan();//refresh//обновить одну строку дисплея
}

//
=====
void loop()
{
unsigned long new_taimer = millis();//текущее время работы скетча в мл.сек
//----- прием по блютуз фраз и команд -----
if (Serial.available() !=0) /// если данные пришли по блютуз
{ Str_telf = Serial.readString();//читаем буфер порта
display.clear(); new_fraza=0;Nx=0;//чистим табло//
flag_tel=1; //опрокидываем флаг и запрещаем работу от вх. машины
time_telf = new_taimer;///запоминаем время когда пришла новая фраза с телефона
}
if (flag_tel==1) //обрабатываем то, что пришло с телефона
{
String str = Str_telf; // создаем объект
len= Str_telf.length() ;//считаем длину
msg_tel = str.c_str(); //преобразовываем массив
sscanf(msg_tel, "%d,", &adres);///берем 5 символов
//-----управление-----
if (adres == 29284 && len>22&&len<26) //если пришла команда разгбераем её на части
отделенные запятой
{
sscanf(msg_tel, "%d,%d,%d,%d,%d,%d,%d,%d,%d",//берем в десятичном значении
&adres, &skorost, &jarkost,&invers,&begut, &vremya, &soхран, &reklam, &shipu );//
переменные
//--1-----2-----3-----4-----5-----6-----7-----8-----9--- № п/п
//29284----70-----70-- ----0-----1-----15-----2-----1-----1--- значения(пример)
//29284,70,70,0,1,15,2,1,1 // ПРИМЕР для записи EEPROM через порт//29284-прописан
в программе телефона (меняется только в app.invertor2)
//хоть разок вставляем ПРИМЕР в порт и отправляем,дабы перезаписать мусор в ячейках
памяти//разово делается//можно с телефона
Skorost=map(skorost, 10, 99, 300, 30); //инвертирует диапазон , после новой команды.
Jarkost=map(jarkost, 10, 99, 40, 255); //инвертирует диапазон , после новой команды
time_sbros=(long)vremya*1000; /// переводим сек в мл.сек*/
display.setBrightness(Jarkost); //устанавливаем яркость
if(сохран==2) //если пришла команда записать//седьмая позиция в команде

```

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34



```

// если значения новые, перезаписываем их.
EEPROM.update(51, skorost);EEPROM.update(52, jarkost); EEPROM.update(53, reklam);
EEPROM.update(54, vremya); EEPROM.update(55, begun); EEPROM.update(56, shipu);
EEPROM.update(57, invers);//инверсия стопа
} //51,52..56 от фонаря взял //лишь бы не первые//

flag_tel=0; adrees = 0;//выходим из записи
display.clear();//стираем фразу
length = WIDTH;//делаем длину "64", для фраз от машины
} //
else //если это была не команда- печатаем текст,
{ if (!new_fraza && flag_len) //разово-только при новой фразе
  {printString(msg_tel,WIDTH,0);//помещаем фразу в буфер
  new_fraza=1; flag_len=0; //считаем длину фразы до старта анимации.
  display.clear();//стираем, ранний старт
  } //
//БЕЖАТЬ, ИЛИ НЕ БЕЖАТЬ? ВОТ, В ЧЕМ ВОПРОС!
if (length <= WIDTH && begun!=2) printString(msg_tel,0,0);//статика
else printString(msg_tel,WIDTH-Nx,0);//бегущая строка
}
if (new_taimer - time_telf >= time_sbros && flag_tel && !Nx && reklam !=2)//если не вкл
реклама
  {flag_tel=0; length = WIDTH ; display.clear();//стираем фразу телефона если время горения
истекло
  rezt=0;//флаг готовности. для вкл знака треугольник(или шипы)
  } // если вкл режим "реклама" , то ниже по скетчу мы не опускаемся :)//тут остаемся т.к.
флаг "flag_tel" не сбрасывается
}

//=====машина=====
=====//
else if (!flag_tel && !tormaz && !pravo && !levo&& !nazad && !avariik &&
PINC<2)//ничего не нажато и флаги сброшены
  {
  if (!Nx && !rezt) //по окончании старой фразы//даем фразе добежать
  {display.setBrightness(Jarkost);//устанавливаем яркость//
  display.clear(); rezt=1; // разочек стираем буфер
  invers_N=0;// готовим инверсии стоп
  }
  if (rezt && shipu !=2) printString(msg_avto[0],20,0); // пишем, знак !/
  else if (rezt && shipu ==2) printString(msg_avto[1],20,0); // пишем букву "Ш" (крупную).
  }
//если флаги не сброшены то скетч пробегает до включенного флага
//если что то нажато (PINC>=2), вначале проверяем тормоз, потом остальное
// конструкция "else if" делает приоритет 1-тормоз.2-аварийка.3-назад.4,5-
повороты.//блоки можно переставить
//-----стоп -----
else if (!tormaz && digitalRead(trm)) //если нажат тормоз
  {tormaz=1,display.clear(); //очищаем дисплей
  printString(msg_avto[A],T,0); //пишем С Т О П или !/
  time_inver=new_taimer;//запоминаем время нажатия
  }

```

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

```

else if (tormaz)
{
  if (!invers) printString(msg_avto[3],4,0); // пишем сразу С Т О П, если нет инверсий
  else //переменная "invers_N" сбрасывается, когда загорится треугольник
  { //Т.е. если(например)горит поворот до СТОП и после, повторно СТОП не будет
инвертироваться//не бужет раздражать лишний раз
    if (invers_N<=invers && new_taimer-time_inver > time_invers)//до заданного числа
инверсий(с телефона выбираем)
    { tormaz=0; invers_N++; flag_ctop= !flag_ctop;//считаем нажатия//инверсия флага
    if (flag_ctop) A=0,T=20;// знак !/
    else A=3,T=4;//СТОП//
    }
    else if (invers_N > invers) {A=3;T=4;printString(msg_avto[A],T,0);} //пишем С Т О П
  }
  if (digitalRead(trm)==LOW) {tormaz=0;display.clear();A=3,T=4;flag_ctop=0;}//выходим
стираем,если отпущен
}
//-----аварийка-----
else if (!avariik && digitalRead(prv)&& digitalRead(lev) ) //заработала аварийка
  { avariik=1; time_povrt = new_taimer; time_2znaka = new_taimer;
display.clear();//запоминаем время
  }
else if (avariik) //спасибо горит XX сек, даже если было однократное моргание //до
time_sbros//меняется с телефона
  {
    if (digitalRead(prv) && digitalRead(lev)) time_povrt=new_taimer;//2поворота снова
моргнули, переводим таймер
    if (new_taimer - time_2znaka >= time_spasibo)// стираем "спасибо" после xx сек
"time_spasibo"
    {
      if (new_taimer - time_povrt >= time_povorot) avariik=0; //больше не моргает
аварийка,выходим отсюда
      else
      {
        if (digitalRead(prv) && digitalRead(lev)) display.clear();//черное табло когда
горят повороты ,а треугольники когда не горят
        else printString(msg_avto[2],2,0); //вкл !/ !/ моргают 2 знака если долго горит
аварийка
      }
    }
    else printString(msg_avto[5],0,0);//печатаем "спасибо" в первые(time_spasibo) секунды
вкл , аварийки
  }
//-----назад-----
else if (!nazad && digitalRead(nzd)) //если нажато НАЗАД
  { //такого эффекта нет в библиотеке//вот изгалялся как мог :)
  static int8_t Nz=-16; static unsigned long pauz_nazd;
  if(!flag_nazd )
  { pauz_nazd = new_taimer; flag_nazd=1; display.clear();Nz++;//наращиваем
переменную позиции по вертикали
  printString(msg_avto[4],8,Nz);//8- это отступ от правого края//печатаем
  }
  }

```

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

```

else if (Nz >= 16) Nz= -16;//начинаем сначала

if(new_taimer - pauz_nazd>= 100&& Nz!=0) flag_nazd=0;//прокручиваем вниз
else if (new_taimer - pauz_nazd>= 1000) flag_nazd=0;// держим в статике в центре 1
сек(1000мс)
}
//-----нажат поворот влево-----
else if (!levo && !avariik && !digitalRead(prv) && digitalRead(lev)) levo=1,length =
WIDTH+44;// вкл поворот влево
else if (levo) //сразу не приступаем к анимации, скетч пробегает снова сверху вниз - А не
нажат ли тормоз?
{
if (!digitalRead(prv) && digitalRead(lev)) time_povrt = new_taimer;//пере взвод
таймера//ждем следующего моргания поворота
if (!digitalRead(lev) &&(new_taimer - time_povrt >= time_povorot)) levo=0; //выкл
поворот лево//не дождались
else printString("<<<< <",WIDTH-Nx,0);//длина строки тут не пересчитывается, поэтому
44 вручную считал
}
//-----нажат поворот вправо-----=-----
else if (!pravo && !avariik && digitalRead(prv) && !digitalRead(lev)) pravo=1,length =
WIDTH+44;// вкл поворот вправо
else if (pravo) //сразу не приступаем к анимации, скетч пробегает снова сверху вниз - А не
нажат ли тормоз?
{
static int8_t Ny=-44; static unsigned long pauz_prv;//"-44" старт за пределами табло
слева
if (digitalRead(prv)) time_povrt = new_taimer;//перевзвод таймера//ждем следующего
моргания поворота
if (!digitalRead(prv)&&(new_taimer - time_povrt >= time_povorot)) pravo=0;//выкл
поворот вправо//не дождались
if (!flag_prv ) //пока флаг не сброшен продолжаем анимацию
{ pauz_prv = new_taimer; flag_prv=1; display.clear();//периодически стираем табло(а то
накладки идут)
Ny++;printString("> >>>",Ny,0);//длина строки тут не пересчитывается, поэтому 44
вручную считал
}
if (Ny >= 64) Ny= -44;//начинаем новый пробег вправо
if(new_taimer - pauz_prv>= 20) flag_prv=0;//"20" величина от которой зависит
скорость строки вправо
}
//-----
if (rezt&&PINC>=2) rezt=0;// готовимся вкл знак !/ если, что то нажималось
}

```

					1.033.00.00	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37