

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ»

Институт математики, физики, информатики и технологии

Кафедра информатики, информационных технологий и методики обучения
информатике

**Системы обеспечения безопасности жилья на основе микроконтроллера
и датчиков**

Выпускная квалификационная работа

Квалификационная работа
допущена к защите
зав.кафедрой
« ____ » _____ 20 ____ г

(подпись)

Выполнил:
Бархатов Алексей Михайлович
Обучающийся группы ПИвС-1501z

(подпись)

Руководитель:
Кудрявцев Александр Владимирович
к.п.н., доцент

(подпись)

Екатеринбург 2020

Оглавление

Введение.....	3
Глава 1. Анализ существующих электронных охранных систем.....	5
1.1. Основные принципы систем безопасности на основе микроконтроллеров.....	5
1.2. Выбор методов разработки, материалов и компонентов.....	11
1.3. Техническое задание.....	28
Глава 2. Практическая часть	32
2.1. Проектирование и установка системы.....	32
2.2. Настройка, программирование и применение системы.....	39
2.3. Результаты апробации, техническая документация.....	50
Заключение	55
Список информационных источников.....	58
Приложения	63
Приложение 1	63
Приложение 2	66
Приложение 3	77
Приложение 4	78

Введение

Проблема обеспечения безопасности объектов недвижимости, будь это гаражи, различные помещения, жилые частные и многоквартирные дома остается актуальной задачей современного мира.

Системы электронных пожарных сигнализаций являются одним из главных компонентов предотвращения или минимизации пожара. Охранные сигнализации служат для обнаружения несанкционированного проникновения на охраняемый объект, формирования, передачи различных извещений, управления звуковыми и световыми оповещателями. Существуют так же и различные специализированные датчики обнаружения протечки воды, и содержание вредных, смертоносных и взрывоопасных газов.

В связи с широким использованием современных электронных компонентов и цифровых методах обработки информации происходит «интеллектуализация» технических средств. Современные технические средства охраны могут быть использованы в виде полностью автономной, интегрированной системы или системы, состоящей из функционально независимых компонентов.

Объект исследования — система обеспечения безопасности жилья на основе микроконтроллера и датчиков.

Цель работы: разработать систему, управляемую микроконтроллером, которая обеспечивает безопасность жилого помещения.

В связи с поставленной целью определены задачи:

- проанализировать существующие системы безопасности, и на основе современных решений, обосновать необходимость выбора тех или иных компонентов системы;
- описать процесс разработки системы, основываясь на определенных критериях;

- собрать систему обеспечения безопасности на основе микроконтроллера, необходимых датчиков и модулей;
- составить программу для микроконтроллера, обрабатывающую сигналы с датчиков и посылающую команды на управляемые модули.

Для решения поставленных задач мы рассмотрели статьи, описывающие различные особенности разработки управляемых микроконтроллером систем, документацию на выбранную плату и другие компоненты, учебные и методические пособия по данному направлению.

Разрабатываемая система должна быть проста в установке, настройке и эксплуатации, и не менее важна гибкость в конфигурации и в логике работы.

Такая система может быть модифицирована под конкретные нужды и успешно применена для обеспечения безопасности любого жилого помещения.

Глава 1. Анализ существующих электронных охранных систем

1.1. Основные принципы систем безопасности на основе микроконтроллеров

Охранная сигнализация — система, отслеживающая посторонние движения на охраняемом объекте: в квартире, дома, в офисе, на складе. Противопожарная охрана — сигнализирует о возникновении пожара. Современное охранное оборудование защищает от несанкционированного доступа извне, а также от повреждений объектов бытовыми угрозами (затопление, пожар, утечка газа). Основное назначение — предупредить, по возможности предотвратить или способствовать предотвращению ситуаций, в которых будет нанесён вред людям или материальным и не материальным ценностям.

Извещатели, являются основными средствами обнаружения несанкционированного проникновения в системах охранной сигнализации. Современные производители выпускают множество устройств, имеющих различный принцип действия. Каждый способ обнаружения имеет свои преимущества и недостатки. Очень важно при проектировании системы безопасности обозначить типы нарушений и нарушителей, от которых охраняется объект.

Охранная система подразумевает как предотвращение нарушения посредством вызова специальной группы людей, так и уведомление владельца о всевозможных нарушениях, предоставляя выбор дальнейших действий. При слабой защите одного места, например, окна, охрана других мест теряет смысл. Поэтому необходимо грамотно определить: в каких местах охрана наиболее необходима и эффективна.

Безопасность может обеспечиваться любым набором устройств и свойств, выполняющих определённые функции, состав определяется каждым производителем самостоятельно. Электронные системы безопасности могут варьироваться от небольшого автономного устройства, работающего на

аккумуляторе до многокомпонентных систем, обеспечивающих безопасность на огромной территории с дорогостоящими многофункциональными составляющими. Многие имеют управляющее устройство с клавиатурой или ключом, или управление может происходить с помощью телефона.

Так же необходимо уделить внимание не только выбору микроконтроллера и датчиков, но и защите кода программы от изменения или несанкционированному доступу к программному обеспечению.

Системы безопасности значительно отличаются друг от друга различными свойствами, областями применения и другими атрибутами. Можно подобрать определённую систему безопасности под конкретные задачи и объекты. Можно выделить три цели защиты:

- информация;
- человек;
- имущество.

Независимо от цели, можно выделить несколько видов систем безопасности, классифицируя по набору используемых датчиков:

- система контроля обхода;
- система контроля и управления доступом;
- охранно-пожарная сигнализация;
- видеонаблюдение;
- охранный сигнализация;
- прочие.

По взаимодействию с угрозой:

- пассивные: комплекс средств и действий, направленных на привлечение внимания владельца или охранных служб;
- активные: для предотвращения проникновения на охраняемый объект.

По способу передачи информации можно разделить на проводные и беспроводные. Проводные в свою очередь делятся на аналоговые (по

средствам измерения потребляемого тока) и адресные (каждый извещатель имеет свой адрес, что позволит определить точное место начала возгорания или проникновения). Беспроводные делятся на системы с обратной связью и без обратной. Недостатками последней являются возможность подавления радиосигнала специальными «глушилками» и «шумилками», а иногда и обычными бытовыми приборами. Системы с обратной связью позволяют проводить непрерывный мониторинг системой всех датчиков.

Некоторые системы позволяют передавать сигналы по GSM-сети на пульт охранной компании или собственнику охраняемого объекта информируя о различных событиях (тревога, пожар, затопление) [30]. Так же в место GSM-сетей могут использоваться протокол Wi-Fi, передавая данные на специальный ресурс в сети Internet, или на мобильное приложение.

Беспроводные системы обычно применяются, когда нет возможности провести проводку.

Специалисты советуют комбинировать как пассивные системы с активными, так и проводные системы с беспроводными.

Простая система может представлять собой камеру, записывающую видео на жёсткий диск или карту памяти.

Сложные же системы могут иметь множество разнообразных датчиков. Для обработки данных с датчиков обычно используется специальный микроконтроллер.

Разберём некоторые из систем безопасности.

Охранные системы ГОЛЬФСТРИМ.

«ГОЛЬФСТРИМ охранные системы» на протяжении 24 лет является лидером российского рынка охранных систем и предоставляет услуги по охране квартир, домов и бизнеса. Наши преимущества:

Беспроводное оборудование: безопасно для ремонта

Установка всего 2 — 3 часа!

Мгновенный прием и обработка сигналов

Гарантия скорости реагирования

Дополнительные датчики дыма, протечки воды, утечки газа и пропадания электричества 220В.

Из минусов стоит выделить высокую стоимость, наличие абонентской платы и ложные срабатывания.

Системы безопасности ДЕЛЬТА.

Недорогая охранная система. На сайте компании легко выбрать систему для охраны, частного дома, бизнеса, квартиры, ресторана, магазина, банка или других объектов особой важности.

Установка сигнализации специалистами, занимает не более трёх часов. На сайте всегда можно найти контактные номера, для связи с сотрудниками организации или задать интересующие вопросы, а ещё можно создать собственный кабинет и следить за своим объектом самостоятельно в любое доступное время.

Сотрудники службы реагирования внимательно осматривают объект. Обнаружив следы проникновения, охранники блокируют все выходы до приезда клиента и полиции. Если злоумышленник предпримет попытку побега, то сотрудники остановят и обезвредят преступника.

Из недостатков стоит выделить работающее с перебоями онлайн приложение, стоимость подключения и абонентскую плату.

Группа частных охранных предприятий «Реванш-охрана».

Удобное управление охраной вашего предприятия через мобильное приложение MyAlarm

- iOS и Android
- взятия \ снятия единым кодом пользователя
- лента событий
- живое видео с камер наблюдения
- тревожная кнопка
- датчик T° и контроль температуры
- видеоподтверждения тревог.

Дополнительные технологические датчики, предупредят вас о бытовых проблемах: пожаре, прорыве трубы, падении температуры или утечке газа.

Основные недостатки: очень высокая стоимость базового комплекта, абонентская плата, ложные срабатывания.

ЧОП «Талион»

Среди наших основных преимуществ можно выделить:

- Стабильность, тщательный подход к изучению задания, организованность являются принципами нашей деятельности;
- Использование современной технической базы — мы всегда следим за рынком оборудования;
- Доступная ценовая политика — формирование цены происходит после обсуждения всех деталей с клиентом, выбора оптимального варианта, исходя из его пожеланий, требований и возможностей;
- Наличие лицензии на проведения охранной деятельности;
- Сотрудничество с правоохранительными органами, благодаря чему осуществляется возможность оперативного решения экстренных проблем;
- Профессиональный штат сотрудников — каждый охранник прошел специальное обучение.

Из отрицательных сторон хотелось бы отметить стоимость минимального комплекта оборудования и наличие абонентской платы.

Расположим получившиеся свойства и атрибуты систем в порядке возрастания важности. Плюсы: большое разнообразие функций, сборка и монтаж обычным пользователем, доступная цена. К минусам можно отнести: невозможность изменения структуры, сложность, плохо разработанное программное обеспечение, высокая цена, абонентская плата, плата за вызов охраны при ложном срабатывании.

Таким образом, рассмотренные нами системы безопасности хоть и позволяют эффективно охранять жилые объекты, но всё же имеют ряд недостатков.

Подводя итог, наиболее важными критериями системы являются цена и простота использования и монтажа.

Определим свойства разрабатываемой охранной системы на основе потребностей большинства потребителей. Составим список функций, основываясь рассмотренных выше систем, учитывая, как положительные, так и отрицательные атрибуты:

- сборка, монтаж;
- простота в использовании;
- мониторинг состояния квартиры;
- определение проникновения;
- возможность модификации под конкретные нужды;
- определение пожара, протечки воды, утечки газа;
- видеонаблюдение;
- контроль температуры.
- анализ данных с датчиков;
- активное реагирование на сигналы с датчиков;
- добавление и изменение набора датчиков;

Учитывая главные недостатки охранных систем, а именно цена и сложность монтирования и управления, нужно разработать понятную инструкцию по монтированию и использованию системы. Вся информация с датчиков будет обрабатываться микроконтроллером, и представляться пользователю в обычном и понятном виде.

В результате должна получиться простая в установке, настройке и модификации охранная система, подходящая под нужды конкретных пользователей.

1.2. Выбор методов разработки, материалов и компонентов

С ростом популярности Arduino количество производителей плат начало исчисляться сотнями. Перечислим и приведём характеристики некоторых из них.

Seeed. Плата Seeeduino Mega

Seeeduino Mega. Это плата, полностью совместимая с платой Arduino Mega. Совместима с большинством модулей расширения (shield), разработанных для Arduino Duemilanove или Diecimila. Он оснащен процессором ATmega2560, который имеет большое количество выводов ввода-вывода, до 70 цифровых входов / выходов, 16 аналоговых входов, 14 ШИМ (шиотно-импульсная модуляция) и 4 аппаратных последовательных порта. Полностью совместим с продуктами Seeed Shield.

Raspberry Pi

Разрабатывается английской компанией Raspberry Pi Foundation во главе с Эбеном Аптоном. Raspberry Pi, это одноплатный компьютер [17] размером с банковскую карту, изначально разработанный как бюджетная система для обучения информатике, но позже получивший более широкое применение и известность. Большая часть моделей одноплатных компьютеров Raspberry Pi распространяется полностью собранным на четырёхслойной печатной плате размером примерно с банковскую карту.

Одной из самых интересных особенностей Raspberry Pi является наличие портов GPIO (general purpose input/output). Благодаря этому его можно использовать для управления различными устройствами [23, с. 170].

Iskra JS

Компания Амперка выпустила устройство Iskra JS, которое управляется при помощи популярного языка Javascript. Iskra JS сочетает в себе лучшие качества Arduino и Espurino, предлагая пользователям расширенные возможности разработки и применения дополнительных модулей.

Arduino

Торговая марка аппаратно-программных средств для построения простых систем автоматики и робототехники, ориентированная на непрофессиональных пользователей. Плата «Arduino» — это самостоятельный процессор, который обладает памятью и снабжен множеством стандартных вводов и выводов. К нему легко подключаются самые разные устройства и механизмы, сенсоры, датчики, моторы и прочее.

Основные серии Arduino: Uno, Nano, Lilypad, Mega, Leonardo, Micro, Due. Самой распространённой всё же является Uno. Все они отличаются не только размерами, но и формами, характеристиками и соответственно ценой. Каждая из этих плат отлична от других и имеет преимущества для разных ситуаций [1]. Из этого следует вывод: под конкретные проекты или приложения больше подойдёт та или иная плата на Arduino.

Сравнение

Среди большого выбора плат и их модификаций нужно выбрать одну более точно подходящую под заданные критерии, такие как простота настройки, стоимость, масштабируемость, а также стоимость компонентов и датчиков.

Сравним основные характеристики плат и добавим Arduino Uno как самую распространённую плату [22, с. 27].

Характеристики	Seeeduino Mega	Raspberry Pi 4B	Iskra JS	Arduino Uno	Arduino Mega 2560 Rev3
Цена, руб	3000	5500	1400	500	1000
Размеры, мм	71x53	85,6x56,5	69x53	69x53	101,6x53
Частота, МГц	16	1500	168	16	16
Микроконтроллер	ATmega2560	ARMv8 Cortex-A72	STM32F405 RG	ATmega328	ATmega2560
FLASH память, кб	256	SD-карта	1024	32	256
SRAM память, кб	8	1024-4096	192	2	8

Аналоговые входы	16	Нужен переходник	12+2	6	16
Цифровые входы/выходы	54 (15 с шим)	Нужен переходник	26(22 с шим)	20	54 (15 с шим)

Таблица 1. Сравнительная таблица характеристик плат

Можно увидеть из таблицы 1, что Raspberry Pi является самым мощным решением, но так как главными критериями являются цена и простота, а размеры не главное, от него придётся отказаться.

Seeeduino Mega практически идентична Arduino Mega, но за исключением большей цены и меньших габаритов.

Iskra JS построенная на микроконтроллере STM32F405RG [10] превосходит по характеристикам платы на ATmega, но имеет меньшее количество входов/выходов чем Mega, а для разрабатываемой системы необходима гибкость и масштабируемость системы.

Для разрабатываемой системы была выбрана Arduino Mega, так как она является мощнее Uno, и имеет наибольшее количество аналоговых и цифровых входов. Поэтому при необходимости можно масштабировать систему путём добавления датчиков и плат расширений. Это позволит более гибко разработать систему под конкретные нужды.

Для работы с Mega необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или аккумуляторной батареей. Arduino Mega совместима со всеми платами расширения, разработанными для платформ Arduino Uno. «Arduino Mega 2560 Rev3 — это последняя модель, пришедшая на смену предыдущим Arduino Mega 2560 и Mega 1280. Версия Rev3 включает в себя чип Atmega16U2 (в версиях Rev1 и Rev2 был Atmega8U2)[18] с программной прошивкой конвертера «USB-последовательный порт», вместо используемых в более ранних версиях микросхем FTDI. Микроконтроллер ATmega16u2 позволяет использовать контроллер Arduino Mega2560 как USB HID устройство». HID (human interface device class) — класс устройств USB для

взаимодействия с человеком. При перепрошивке этого микроконтроллера контроллер можно использовать как клавиатуру, мышь, игровой джойстик и другие устройства, взаимодействующие с компьютером.

Датчики

Большинство датчиков имеют сложную процедуру преобразования измеряемой величины в электрический сигнал. Например, в тензорезисторном датчике давления измеряемая величина воздействует на чувствительный элемент, изменяя его сопротивление. После подачи сигнала возбуждения, падение напряжения на резисторе позволит косвенно определить его сопротивление и, на основании зависимости сопротивления от давления, вычислить измеряемую величину.

Для разработчика охранной системы датчик представляет собой черный ящик с известными соотношениями сигналов между входами и выходами.

Определим слабые места в доме, тем самым составим список датчиков, необходимых для системы безопасности жилья. Такими местами является входная дверь, уборная комната, кухня с газовой плитой.

Таким образом, составим примерный список необходимых компонентов защиты:

- датчик открытия двери, датчик движения;
- датчик протечки воды, влажности;
- датчик природного газа, датчик задымлённости.

В качестве реагирования системы можно выбрать звуковой сигнал. Для оповещения владельца будем использовать GSM сеть. Оповещение владельца с помощью SMS будет намного практичнее, чем через интернет, т.к. покрытие мобильной сети намного больше, чем покрытие мобильного интернета. К тому же подключение такой системы к сети интернет может обернуться большими проблемами, такими как: компрометация, изменение, нарушение всего кода или части системы.

Датчики определения проникновения

Обнаружение проникновения — один из наиболее важных аспектов безопасности для жилья или иных объектов. Так обеспечивается охрана имущества, ценных вещей от порчи или кражи. Периметральная сигнализация обеспечивает круглосуточный контроль внешних границ территории или выходов помещения с помощью специальных датчиков, генерирующих лучи или сенсорных проводов и кабелей. Когда луч прерывается или кабель реагирует на воздействие, формируется сигнал о несанкционированном проникновении на охраняемый объект.

Для этих задач существует огромное количество датчиков (рисунок 1).

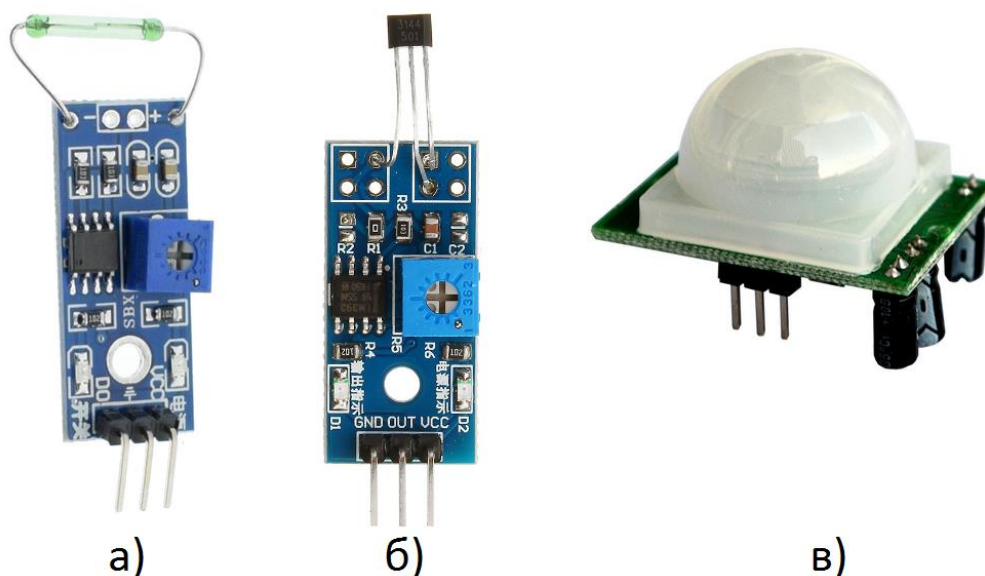


Рисунок 1. Датчики: а) - с герконом, б) - Холла, в) - движения

Слово «геркон» образовано от словосочетания «герметичный контакт». Устройство датчика довольно простое, он представляет собой два разомкнутых или замкнутых контакта в колбе с инертным газом или с вакуумом. Контакты меняют свое состояние при поднесении магнита, при этом на выходе модуля появляется низкий уровень напряжения [16]. В работе представлен нормально разомкнутый геркон, при поднесении магнита контакты замыкаются, сам модуль символизирует о наличии питания и о срабатывании датчика.

Более простые датчики Холла [4]. Это твердотельные устройства, которые генерируют электрические сигналы, пропорциональные приложенному к нему магнитному полю. Эти электрические сигналы затем дополнительно обрабатываются специальной электронной схемой пользователя для получения желаемого выхода.

В наши дни эти магнитные датчики способны реагировать на широкий спектр магнитных полей. Одним из таких магнитных датчиков является датчик Холла, выход которого (напряжение) зависит от плотности магнитного поля.

Рассмотрим еще один датчик, связанный с обеспечением безопасности для умного дома. Это модуль датчика присутствия HC-SR501 [12] на основе пирозлектрического эффекта. «Состоит из самого PIR-датчика (Pyroelectric (Passive) InfraRed sensor) и схемы управления. Такие датчики часто используются в охранных системах и в быту для обнаружения движения в помещении» [26, с. 76].

Модуль имеет два переменных резистора и переключку для настройки режима. Пассивный он потому, что для обнаружения движения не используется какая-либо дополнительная энергия, кроме той, что испускается самими объектами.

500BP состоит из двух чувствительных элементов. Управляющая микросхема модуля регистрирует изменения сигналов от обоих элементов и по характеру их изменения обнаруживает движение объектов, испускающих инфракрасные сигналы (живых организмов).

Подводя итог остановимся на выборе геркона и датчика движения HC-SR501 для включения света, так как они проще в подключении и обладают малой ценой.

Бытовые датчики

В функционал бытовых датчиков входят такие задачи как: обнаружение протечки воды, утечки газа, регистрация температуры и давления, обнаружение концентрации ядовитых газов – рисунок 2.

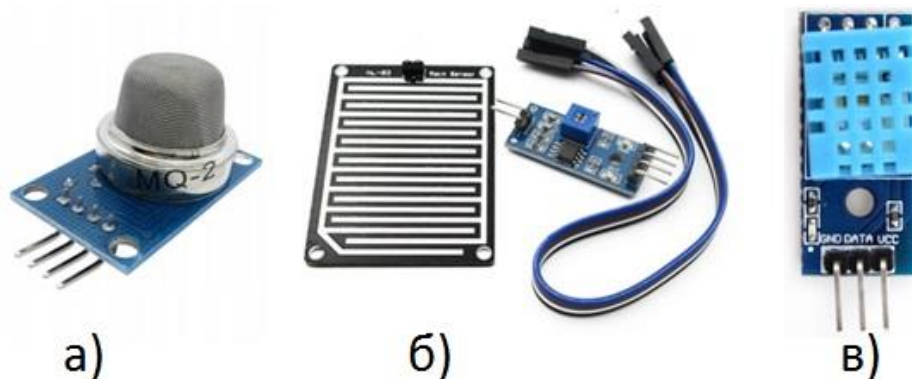


Рисунок 2. Датчики: а) - газа и дыма, б) - протечки воды, в) - температуры

Датчик MQ-2 определит концентрацию углеводородных газов (пропан, метан, н-бутан), дыма (взвешенных частиц, являющихся результатом горения) и водорода в окружающей среде. Сама плата датчика состоит из 6-пинового датчика, который для точных показаний должен некоторое время постоять работающим и нагреться до необходимой температуры. Потенциометр для изменения чувствительности датчика. Подключается к Arduino с помощью 4 пинов, 2 из которых +,- и 2 пина: один Аналоговый, другой цифровой. Подключать нужно либо к аналоговому, либо к цифровому пину. Существует целая серия не дорогих датчиков MQ и в таблице 2 представлены некоторые из них [14, с. 104].

Датчик	Определение газа
MQ-2	Дым и другие газы, возникающие в процессе горения
MQ-3	Пары алкоголя (этиловый спирт)
MQ-4	Метан (CH ₄)
MQ-5	Горючие газы
MQ-6	Сжиженные нефтяные газы
MQ-7	Угарный газ (CO)
MQ-9	Угарный газ, метан, природный газ (пропан или бутан)
MQ-131	Озон (O ₃)

Таблица 2. Датчики серии MQ и газы, которые они определяют

В связи с тем, что мы проектируем систему безопасности жилья отберём необходимый датчик. Для определения газа возьмём специальный для этого датчик MQ-4. «Датчик MQ-2 определит концентрацию

углеводородных газов (пропан, метан, н-бутан), дыма (взвешенных частиц, являющихся результатом горения) и водорода в окружающей среде. Датчик можно использовать для обнаружения утечек газа и задымления» [26, с. 55]. Для определения дыма выберем MQ-2, кроме того с его помощью можно будет определять газ, тем самым подстраховывая MQ-4.

Датчик протечки и дождя в проектах Arduino позволяет определить появление капель влаги и вовремя отреагировать на это, например, включив оповещение. Модуль датчика состоит из двух частей:

«Сенсорная» плата обнаружения капель. Она отслеживает количество попавшей на неё влаги. Представляет собой простой переменный резистор, замыкаемый водой в разных местах, что вызывает изменение сопротивления.

Вторая часть датчика – сдвоенный компаратор (как правило, LM393, но возможны варианты LM293 и LM193). Его главная задача – преобразование значения с сенсора в аналоговый сигнал от 0 до 5 вольт.

Так как одна из самых распространённых бытовых катастроф – потоп наличие датчика воды обоснованно.

Для комфорта, и контроля микроклимата жилья установим датчик температуры и влажности [29]. «В повседневной жизни влажность выступает немаловажным параметром, от степени влажности воздуха немало зависит наше самочувствие. Особенно чувствительными к влажности являются метеозависимые люди, а также люди, страдающие гипертонической болезнью, бронхиальной астмой, заболеваниями сердечнососудистой системы» [26, с. 28].

Датчик DHT11 [3] состоит из двух частей – емкостного датчика температуры и гигрометра. Первый используется для измерения температуры, второй – для влажности воздуха. Находящийся внутри чип может выполнять аналого-цифровые преобразования и выдавать цифровой сигнал, который считывается посредством микроконтроллера. Так же из серии DHT существуют и другие датчики, их характеристики представлены в

таблице 3. Погрешность влажности измеряется в RH (Relative Humidity) — величина, показывающая, насколько далёк пар от насыщения.

Датчик	Определение влажности в % и погрешность в RH	Определение температуры и погрешности в °C	Частота опроса в секунду
DHT11	20-80, ± 5	от 0 до 50, ± 2	1
DHT21	0-100, ± 2	от -40 до +80, $\pm 0,5$	1, 2
DHT22	0-100, ± 2	от -40 до +125, $\pm 0,5$	1, 2

Таблица 3. Характеристики датчиков DHT

Так как в жилом помещении температура воздуха обычно не ниже 10 и не выше 40 °C, а влажность не меняется от 0 до 100%, можно выбрать не дорогой и простой DHT11.

Радиомодули

«Существует огромное количество способов передачи данных по радиоканалу. С некоторыми из них вы сталкиваетесь очень часто: это прежде всего мобильная связь и Wi-Fi. Несомненно, знаком вам также и интерфейс Bluetooth — беспроводная замена последовательного порта в мобильных и планшетах» [28, с. 179].

Например, полудуплексный, последовательный приёмопередатчик HC-12. Модуль работает со 100 каналами в диапазоне 433.4–473.0 МГц, и способен передавать данные на расстояние до 1 км на прямой видимости со скоростью 5000бит/сек. С помощью модуля nRF24L01 [19], можно соединить две Arduino или отправить команды, например, роботу. Работает такой передатчик на частоте 2,4ГГц, имеет 128 каналов с шагом 1МГц и может передавать данные, со скоростью от 0,25 до 2Мбит/сек. Радиус же действия: в помещении до 30м, на открытой местности до 100м. Нередко возникает необходимость дистанционного управления или передачи данных с телефона или другого устройства. Одним из удобных способов является Bluetooth (от слов англ. blue — синий и tooth — зуб, являющейся спецификацией беспроводных персональных сетей). Самыми популярными модулями

являются устройства на основе чипа BC417. Эта серия называется HC. Модули HC-03 и HC-05 могут быть и сервером соединения, и клиентом, они обладают широким набором AT команд (от англ. attention — «внимание», это короткий набор текстовых команд в специальном формате). Спецификация Bluetooth – v2.0 и EDR. Максимальная дальность связи 30м.

Несомненно, самым распространённым каналом передачи данных является Wi-Fi (от англ. Wireless Fidelity, что означает «беспроводная передача данных» или «беспроводная точность») [2]. Имеется множество библиотек, в том числе и пользовательских, а также поддерживается работа через Serial шины и простейшие AT и AT+ команды. Все модули построены на базе популярного чипсета ESP8266 [32, с. 13]. Некоторые модули представлены на рисунке 3.

Все модули различаются не только компоновкой и расположением антенны, но и характеристиками. Дальность связи будет зависеть от погодных условий, препятствий, типа антенны (внешняя, внутренняя) и обычно может составить от 30м до 1км.

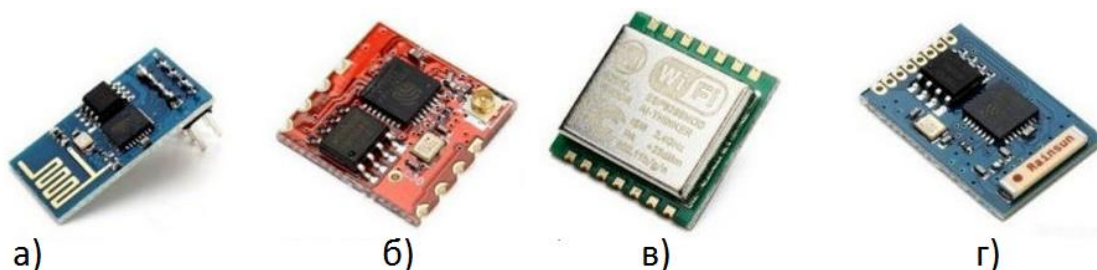


Рисунок 3. Wi-Fi ESP модули: а) - 01, б) - 02, в) - 08, г) - 11

Существуют так же GSM и GPRS модули, которые позволяют подключаться устройствам по средствам сотовых сетей. GSM (от названия группы Groupe Special Mobile) — глобальный стандарт цифровой мобильной сотовой связи с разделением каналов по времени (TDMA) и частоте (FDMA). Стандарт GPRS (от англ. General Packet Radio — пакетная радиосвязь общего использования). Это надстройка над технологией мобильной связи GSM, осуществляющая пакетную передачу данных. GPRS предполагает

тарификацию по объёму переданной/полученной информации, а не по времени, проведённому онлайн. Самыми распространёнными являются модули SIM800L и Neoway M590 представленные на рисунке 4.

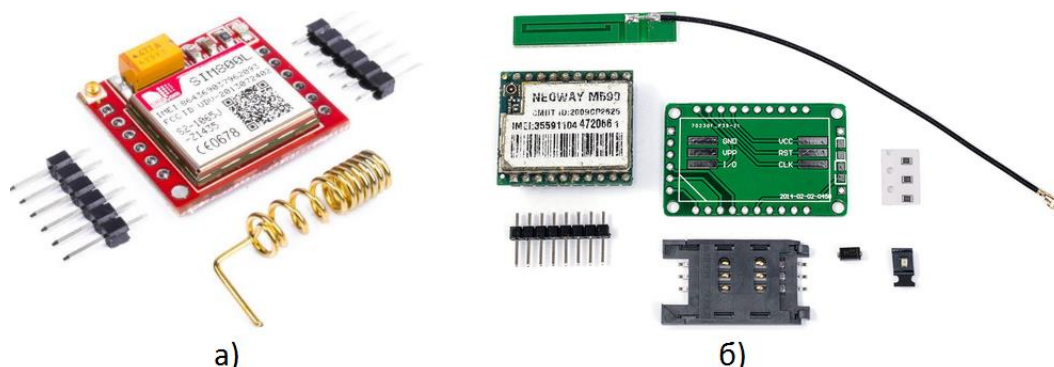


Рисунок 4. GSM/GPRS модули: а) - SIM800L, б) - Neoway M590

Основа модуля – компонент SIM800L с реализацией обмена данными по каналам GSM и GPRS с помощью дуплексного режима. В модуль устанавливается SIM-карта, есть встроенная антенна и выход под еще одну антенну. Это более продвинутый чип, имеющий вход микрофона и выход на динамики, т.е. можно принимать передавать голос. Сердцем модуля SIM800L является чип Mediatek ARM MT6261. За GSM/GPRS-связь отвечает 4 диапазонный (GSM850/GSM900/DCS1800/PCS1900) приемопередатчик RF7198. Neoway M590 — предназначен в основном для устройств телеметрии, поэтому не поддерживает звуковые функции — микрофон и динамики [15]. Имеет два диапазона сетей. Рабочая частота 900 / 1800 МГц. Так как система будет только отправлять и принимать SMS, а голосовая связь не нужна, выберем самый простой и дешёвый, а именно Neoway m590.

GSM модуль станет оптимальным вариантом для создания бюджетной охранной системы, которую можно настроить под конкретный объект. Нет необходимости в использовании Wi-Fi роутера и интернета [11].

Другие

Так же существуют и другие датчики и устройства. Например, различные джойстики и клавиатуры, представленные на рисунке 5.



а)



б)



в)

Рисунок 5. а) - джойстик, б) и в) - клавиатуры

Джойстики являются одним из устройств для удобной передачи информации от человека к компьютеру или микроконтроллеру. Они используются для управления движением роботов, мобильных платформ и прочих механизмов. Клавиатуры используются там, где необходим пользовательский ввод информации. Для этого очень часто используются кнопки. Количество кнопок обычно варьируется от 4 до 16.

Существуют различные модули повышения/уменьшения и стабилизации напряжения. Для устройств, работающих в режимах с относительно большими токами существуют реле. «Реле — электрическое или электронное устройство (ключ), предназначенное для замыкания и размыкания различных участков электрических цепей» [25, с. 213]. Реле в Arduino включает или выключает внешние устройства, определенным образом замыкая или размыкая отдельную электрическую сеть, в которую они подключены. Например, для подключения вентиляторов, позволяющих проветрить помещения при большой концентрации углекислого и других газов. Или же можно подключить освещение и таким образом создать проект для автоматического включения света при нахождении человека в помещении.

С помощью различных двигателей можно двигать предметы и даже собрать 3D принтер. Или автоматически открывать/закрывать двери и окна.

Как правило, датчики используются не сами по себе, а входят в состав системы управления, обеспечивая сигнал обратной связи.

Благодаря несложной конструкции, простоте языка программирования, а также использования открытых кодов даже непрофессионал сможет самостоятельно сделать многофункциональную сигнализацию для охраны своего дома, дачи, квартиры или гаража.

План работы системы

Система будет состоять из самой платы Arduino, нескольких датчиков и GSM-модуля на основе Neoway m590. Датчик газа и дыма устанавливается на кухне, рядом с плитой, датчик движения в зале, датчик открытия на входную дверь, а датчик воды в ванную комнату. Все датчики соединены с платой, которая будет считывать с них показания. При срабатывании определённых датчиков отправляется сообщение по GSM-модулю с информацией о проблеме или опасности.

При срабатывании датчика движения активируется реле включающее свет в комнате, при срабатывании датчика дыма или газа активируется реле включающее вентилятор или вытяжка.

При определённых командах по SMS будут совершаться действия: например, отправка температуры и влажности, что позволит эффективнее мониторить состояние жилого объекта.

Рассмотрим подробнее возможности платы Arduino Mega 2560 r3: входы, выходы, алгоритмы подключения модулей и способы подключения питания.

Arduino Mega на базе ATmega2560 [32, с. 11] — рисунок 6, имеет 54 порта ввода вывода, 15 из которых могут работать, как источник ШИМ сигнала, для плавного регулирования мощности, тока, скорости, яркости, в общем, всего, что можно регулировать с помощью широтно-импульсной модуляции, плюс к этому 16 аналоговых портов могут обрабатывать сигналы с датчиков, использоваться, как цифровой выход.

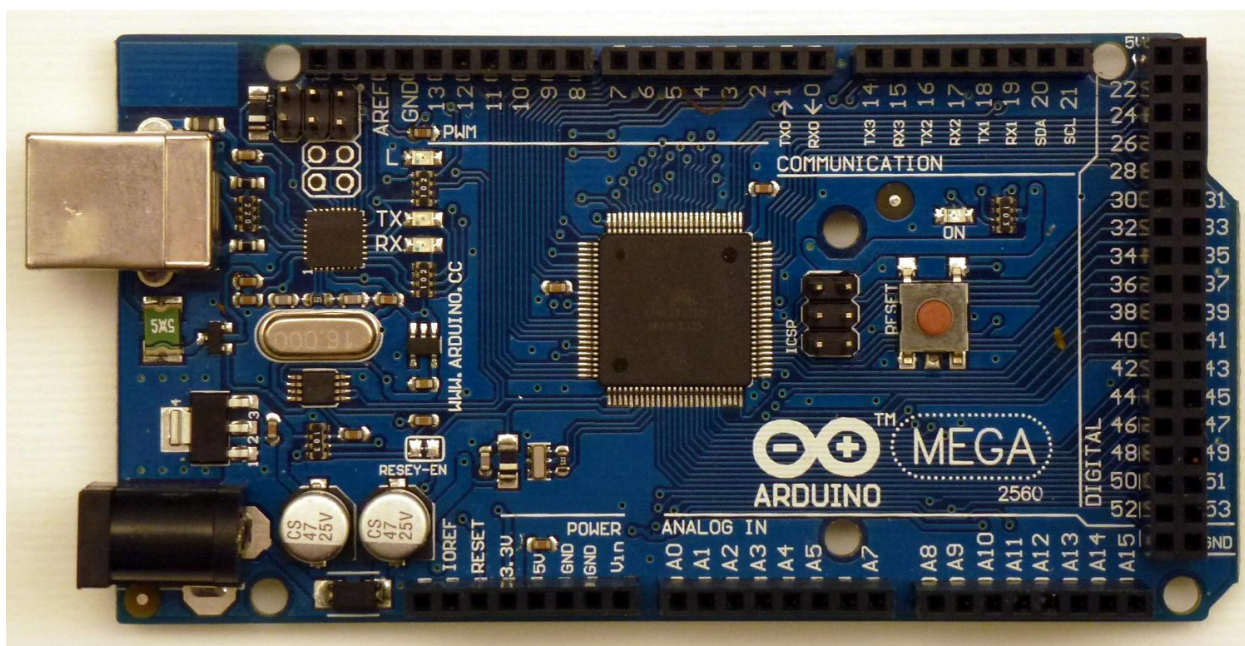


Рисунок 6. Arduino Mega 2560

Для связи между разными устройствами предусмотрено целых 4 UART интерфейса, в их роли выступают выводы 0, 1, 14–19. Один из портов направлен на USB через микроконтроллер ATmega8U2 — он здесь применен вместо привычного по младшим платам USB-TTL контроллера, а его прошивка доступна для свободного скачивания. Для связи с различными дисплеями и другими исполнительными устройствами предусмотрена SPI и I2C технологии [21].

Микроконтроллер ATmega16U2 обеспечивает связь микроконтроллера ATmega2560 с USB-портом компьютера. При подключении к ПК Arduino Mega 2560 определяется как виртуальный COM-порт. Порт USB Type-B для питания и прошивки платформы Arduino Mega 2560 с помощью компьютера. Разъём для подключения внешнего питания от 7 В до 12 В. Кнопка сброса — аналогия кнопки RESET обычного компьютера, перезапускает Arduino.

ICSP-разъём предназначен для внутрисхемного программирования микроконтроллера ATmega2560. Также с применением библиотеки SPI данные выводы могут осуществлять связь с платами расширения по интерфейсу SPI. Линии SPI выведены на 6-контактный разъём, а также продублированы на цифровых пинах 50(MISO), 51(MOSI), 52(SCK) и 53(SS).

Breadboard (макетная (монтажная) беспаячная плата). Основное назначение такой платы — конструирование и отладка прототипов различных устройств. Состоит данное устройство из отверстий-гнезд с шагом 2,54мм либо кратным ему. «Внутри каждого отверстия находится металлический зажим. Когда соединительный провод компонента вставляется в отверстие, зажим захватывает его, и удерживает на месте. Как только мы его вставили, с помощью электрического соединения компонент будет подключен ко всем другим компонентам, что находятся в этом ряду» [31, с. 31]. Макетные платы бывают различных размеров, но в большинстве случаев они состоят из одинаковых блоков, рисунок 7. Слева и справа находится по две линии питания: здесь все отверстия в столбце соединены между собой. Прорезь по середине предназначена для установки и удобного извлечения микросхем в DIP-корпусах. DIP корпус (англ. dual in-line package, также DIL) — применяется для микросхем, микросборок и некоторых других электронных компонентов. Корпуса такого типа отличаются прямоугольной формой и наличием двух рядов выводов по длинным сторонам.

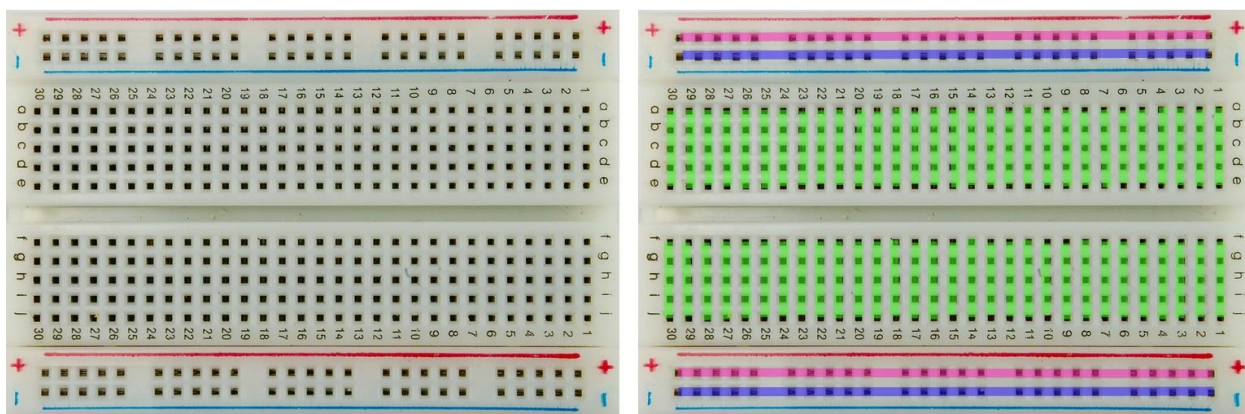


Рисунок 7. Breadboard

Беспаячные монтажные платы используют как для создания простеньких электросхем, так и для сложных прототипов.

Еще одна сфера применения breadboard'ов — проверка новых деталей и компонентов — например, микросхем (ICs).

Как и всем электронным устройствам, Arduino тоже нужно питание. Arduino Mega может быть запитан либо от USB, либо от внешнего источника

питания — тип источника выбирается автоматически. В качестве внешнего источника питания (не USB) может использоваться любой сетевой AC/DC-адаптер или аккумулятор/батарея. Штекер адаптера (диаметр — 2.1мм, центральный контакт — положительный) необходимо вставить в соответствующий разъем питания на плате. В случае питания от аккумулятора/батареи, ее провода необходимо подсоединить к выводам Gnd и Vin разъема POWER.

Напряжение внешнего источника питания может быть в пределах от 6 до 20 В. Однако, уменьшение напряжения питания ниже 7В приводит к уменьшению напряжения на выводе 5V, что может стать причиной нестабильной работы устройства. Использование напряжения больше 12В может приводить к перегреву стабилизатора напряжения и выходу платы из строя. С учетом этого, рекомендуется использовать источник питания с напряжением в диапазоне от 7 до 12В.

Выводы питания:

— VIN: вход для внешнего питания, используется при отсутствии питания от разъема USB. Напряжение, поступающее в Arduino непосредственно от внешнего источника питания. Через этот вывод можно как подавать внешнее питание, так и потреблять ток, когда устройство запитано от внешнего адаптера.

— 5V: используется для питания МК и других компонентов. На этот вывод поступает напряжение 5 В от стабилизатора напряжения на плате, вне зависимости от того, как запитано устройство: от адаптера (7–12 В), от USB (5В) или через вывод VIN (7–12 В). Запитывать устройство через выводы 5V или 3V3 не рекомендуется, поскольку в этом случае не используется стабилизатор напряжения, что может привести к выходу платы из строя. Максимальный выходной ток составляет 800 мА.

— 3V3: вывод 3.3 В, через него поступает напряжение от стабилизатора напряжения на плате. Максимальный выходной ток составляет 150 мА.

— GND: выводы земли.

Так же в Arduino Mega 2560 есть восстанавливаемые предохранители, защищающие USB-порт компьютера от коротких замыканий и перегрузок. Даже, 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.

Так как GSM модуль Neoway m590 может потреблять большой ток, до 2-х ампер в пике, поэтому для надёжной работы необходимо организовать дополнительное питание. Для этого может подойти зарядка для телефона на 5В и 2А. В таком случае соединим землю Arduino и модуля.

1.3. Техническое задание

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку информационной системы обеспечения безопасности жилья на основе микроконтроллера и датчиков.

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1.Название организации-заказчика.

Некоммерческая организация обеспечения безопасности жилых домов «Безопасный дом».

1.2.Название продукта разработки (проектирования).

Системы обеспечения безопасности жилья на основе микроконтроллера и датчиков.

1.3.Назначение продукта.

Обеспечение безопасности жилья на основе микроконтроллера и датчиков. Обнаружение: несанкционированного доступа, возгорания, утечки газа и потопы.

1.4.Плановые сроки начала и окончания работ.

Начало работ по созданию системы – 10 ноября 2019. Конец работ по созданию системы – 10 февраля 2020.

2. Характеристика области применения продукта.

2.1.Процессы и структуры, в которых предполагается использование продукта разработки.

Охрана и безопасность жилья, частных и многоквартирных домов

2.2.Характеристика персонала (количество, квалификация, степень готовности).

Персонал должен состоять из одного человека квалификации пользователь. (Программист – при изменении, модификации программы.

После провести тестирование). Сохранять целостность и не допускать повреждения микроконтроллера, датчиков и соединительных проводов при установке. Знать правила пользования и эксплуатации электроприборов.

3. Требования к продукту разработки.

3.1.Требования к продукту в целом.

Простота установки, настройки. Цена. Гибкость и масштабируемость.

3.2.Аппаратные требования.

Микроконтроллер и датчики: газа и дыма, протечки воды, температуры и влажности, а также открытия/закрытия двери. Модуль взаимодействующий с сотовыми сетями, пьезоизлучатель для звукового оповещения. Наличие реле для установки и разрыва соединений в электросетях.

3.3.Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

Windows, Arduino IDE.

3.4.Указание программного обеспечения, используемого для реализации.

Windows, Arduino IDE.

3.5.Для сетевых систем – особенности реализации серверной и клиентской частей.

При срабатывании датчика, микроконтроллер производит определённые действия и при необходимости отправляет SMS-сообщение. Возможность отправки специальных команд по средству SMS-сообщений.

3.6.Форматы входных и выходных данных

Входные данные: показания датчиков дыма и газа — целое число означающее концентрацию дыма и газа соответственно. Датчик движения — HIGH (замечено движение) и LOW (движение не обнаружено). Показания датчика воды — целое число от 0 (потоп) до 1023 (сухо). Показания температуры и влажности — дробное число. Команды владельца — SMS-

сообщение содержащее определённое слово. Значение датчика двери — 0 (геркон замкнут — двери закрыты) и 1 (геркон разомкнут — двери открыты).

Выходные данные: SMS-сообщения с текстом: сработавшего датчика (при превышении определённого порога срабатывания), показаний температуры и влажности в виде десятичной дроби, активные (on — включен, off — выключен) и сработавшие датчики (true — датчик сработал, иначе — false), баланс сим-карты в денежном формате.

Формат SMS-сообщений: текстовый режим с GSM кодировкой (латиница), номер телефона в виде цифр.

3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Круглосуточный мониторинг обстановки жилья датчиками. Источники данных — сами датчики. Код скетча хранится в постоянно запоминающем устройстве (ПЗУ) — flash-памяти. Во время работы устройства, значения датчиков хранятся в переменных, которые в свою очередь находятся в оперативно запоминающем устройстве (ОЗУ) — SRAM в виде нулей и единиц (двоичная система исчисления). После перезагрузки микроконтроллера такая память полностью очищается.

3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Приём и отправка SMS-сообщений

3.9. Меры защиты информации.

Код микроконтроллера изменяется только непосредственно при физическом контакте с подключением платы к компьютеру.

Сам микроконтроллер будет находиться в закрытом помещении исключая повреждение от погодных условий или несанкционированного доступа.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

Мобильное устройство с функцией приёма SMS-сообщения.

4.2.Размещение информации на экране, дизайн экрана.

SMS сообщение на экране телефона.

4.3.Особенности ввода информации пользователем, представление
выходных данных.

Определённые команды в виде SMS с телефона владельца.

5. Требования к документированию.

5.1.Перечень сопроводительной документации.

Правила работы с системой (Приложение № 1).

5.2.Требования к содержанию отдельных документов.

В правилах работы с системой должны указаны схемы подключения, настройки и способах проверки (тестирования). Общий перечень датчиков и используемых компонентов системы (состав).

6. Порядок сдачи-приемки продукта.

Проверка работы. Передача продукта и документации.

Глава 2. Практическая часть

2.1. Проектирование и установка системы

Датчики, подключаемые к Arduino, позволяют системе «общаться» и взаимодействовать с окружением: анализировать, отмечать, фиксировать изменение показаний и совершать определённые действия.

Для наглядности и моделирования схем будем использовать программу Fritzing [13, с. 56]. Подключим, самый популярный датчик DHT11 – датчик температуры и влажности.

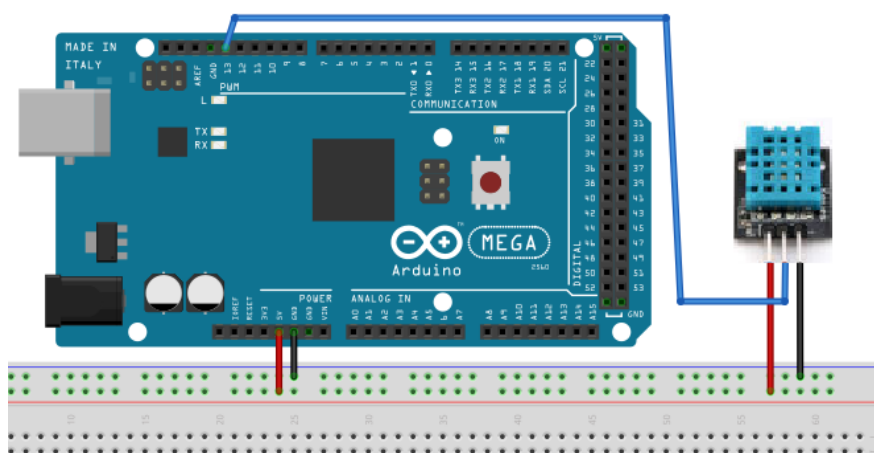


Рисунок 8. Схема подключения датчика DHT11

Если имеется только сам датчик, то для подключения DHT11 понадобится подтягивающий резистор номиналом в 10 кОм. Если же готовый модуль, то подключение его к Arduino предельно упрощается: VCC к +5В, GND – к земле, третий контакт – к любому свободному пину на плате Arduino, на рисунке 8, датчик подключен к цифровому пину 13, но у нас к 13 и 12 пину будет подключен модуль GSM. Подключим DHT11 к 11 пину. Номер пина затем укажем в скетче (`#define DHTPIN 11`). Таким образом, можно подключить датчика к разным платам Arduino: Arduino Uno, Mega, Nano, Pro Mini и другим.

Микроконтроллер выступает в роли ведущего устройства шины и, следовательно, отвечает за инициирование связи (т. е. чтение). Датчик влажности и температуры DHT11 всегда остаётся в качестве подчиненного устройства и отвечает данными, когда микроконтроллер опрашивает его.

DHT11 состоит из 3 основных компонентов. Датчик влажности резистивного типа, термистор NTC и 8-разрядный микроконтроллер, который преобразует аналоговые сигналы и отправляет одиночный цифровой сигнал.

При подключении геркона установим подтягивающий резистор на 10 кОм. Схема подключения указана на рисунке 9. Подключим его к 10 цифровому пину.

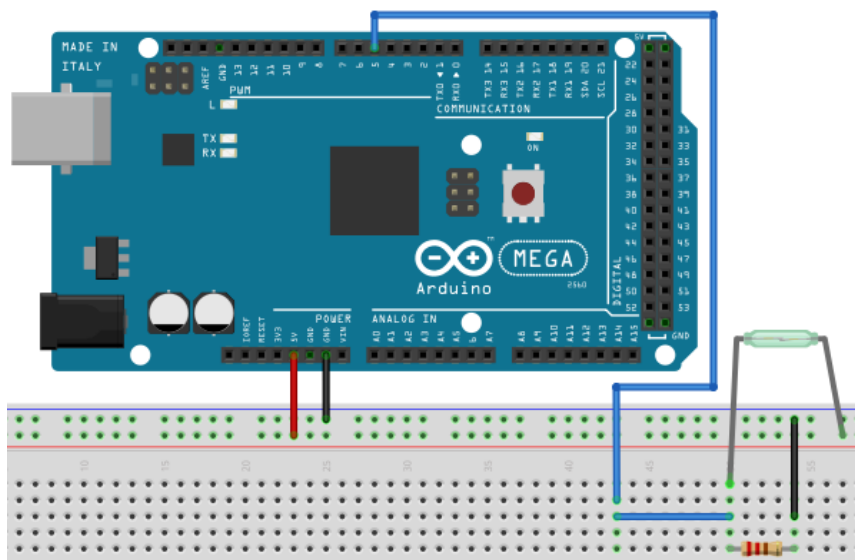


Рисунок 9. Подключение геркона

Аналогично устанавливаем следующие датчики. При подключении будет меняться только пин данных: аналоговый или цифровой. Некоторые сенсоры требуют подключение с резистором, другие же могут потребовать дополнительного питания. Другие состоят из двух частей, например датчик протечки и дождя (рисунок 10). «У него имеются четыре выхода: VCC, GND, аналоговый и цифровой» [7, с. 93]. Подключим его к аналоговому пину A0 (water = A0).

Модуль датчика состоит из двух частей:

- «сенсорная» плата обнаружения капель. Она отслеживает количество попавшей на неё влаги. По сути, сенсор представляет собой простой переменный резистор, замыкаемый водой в разных местах, что вызывает изменение сопротивления.

- сдвоенный компаратор (как правило, LM393, но возможны варианты LM293 и LM193). Его главная задача – преобразование значения с сенсора в аналоговый сигнал от 0 до 5 вольт.

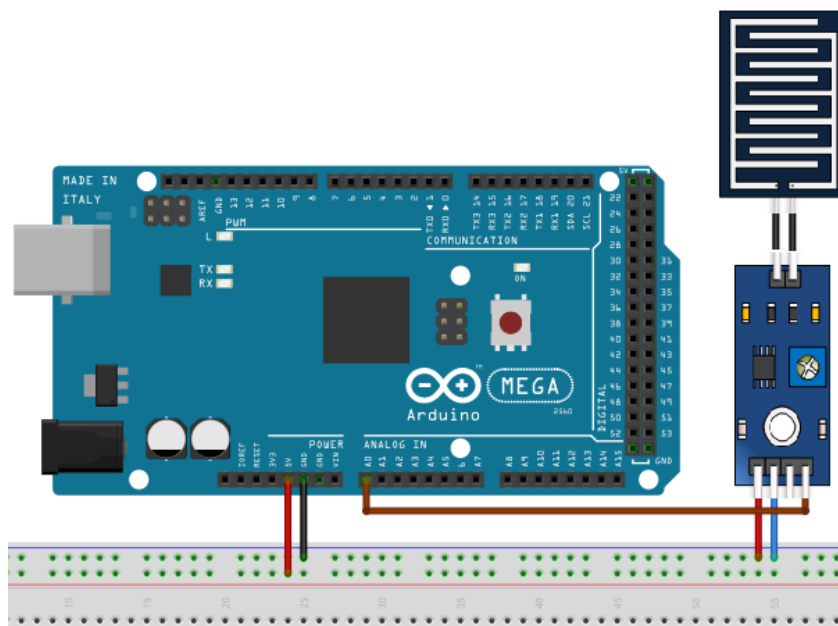


Рисунок 10. Датчик протечки и дождя

Как правило, у модуля датчика доступно два выхода. Один аналоговый. Значение, получаемое контроллером, будет варьироваться от 0 до 1023. Где 0 — все затопило или идет ливень, сенсор очень влажный, 1023 — сухая погода, сенсор. Второй цифровой. Выдает высокое (5В) или низкое напряжение в случае превышения некоторого порога. Уровень порога срабатывания регулируется с помощью подстроечного резистора.

Далее, на рисунке 11, приведены схемы подключения других датчиков.

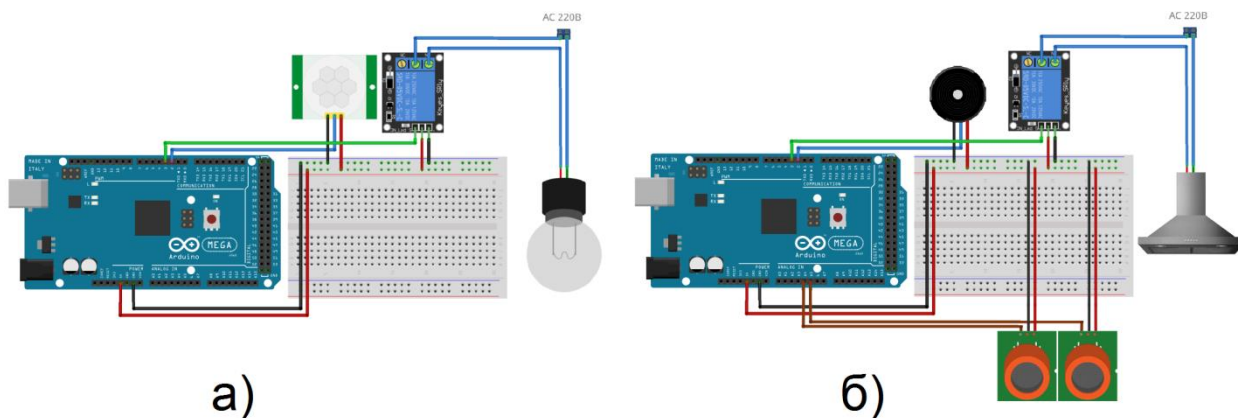


Рисунок 11. Подключение реле и датчиков: а) - движения, б) датчики газа и дыма

Подключаем реле к 5 и 6 цифровому пину (relay1 = 6, relay2 = 5). Пятый пин будет отвечать за включение света, шестой — включение вытяжки. С помощью реле можно управлять высоковольтными электронными устройствами. Реле на самом деле является переключателем, который электрически приводится в действие электромагнитом. Электромагнит активируется низким напряжением, например, 5В от микроконтроллера, и он тянет контакт, чтобы создать или разорвать цепь высокого напряжения. В данной системе безопасности жилья будем использовать 2-канальный релейный модуль HL-52S, который имеет 2 реле с номиналами 10 А при 250 и 125 В переменного тока и 10 А при 30 и 28 В постоянного тока. Выходной разъем высокого напряжения имеет 3 контакта, средний является общим контактом, и, как видно из маркировки, один из двух других контактов предназначен для нормально разомкнутого соединения, а другой — для нормально замкнутого соединения.

Таким образом, возможно управлять любым высоковольтным устройством с помощью Arduino или любого другого микроконтроллера [20, с. 103].

Установка GSM модуля

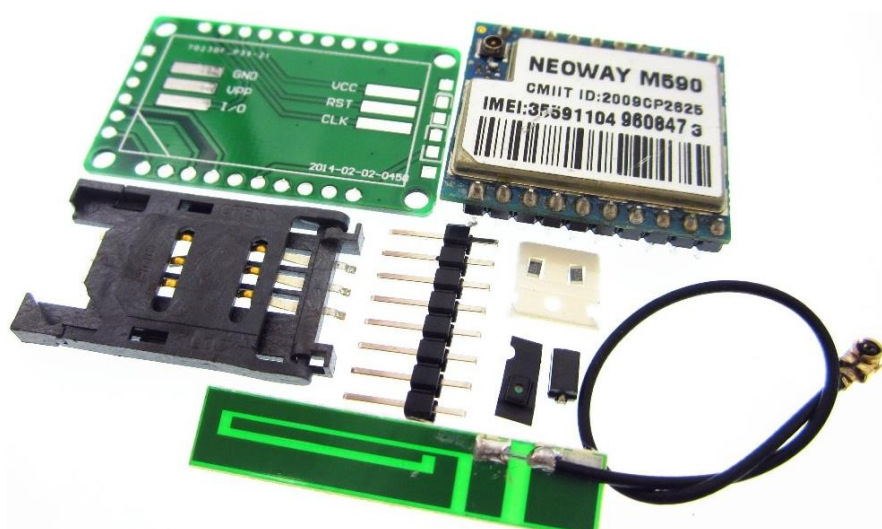


Рисунок 12. Neoway m590

В отличие от подключения выше описанных датчиков, модуль GSM подключается совсем по-другому. Он представляет собой беспроводное

коммуникационное устройство (модем) для приема/передачи данных в сетях мобильной связи. В любом мобильном устройстве, позволяющем подключить сим-карту, есть такой модуль.

GSM модуль Neoway m590 может продаваться как в разобранном, так и в собранном виде, рисунок 12.

Для начала модуль необходимо собрать. Для этого с одной стороны нужно припаять SMD компоненты: резистор R2, и диод, который нужно установить анодом к контакту +5V. Затем сверху припаиваем сам модуль, совмещая белую точку на модуле и белый круг на плате. На другой стороне припаиваем светодиод таким исходя из того, что зелёная точка – катод. Далее припаиваем держатель для сим-карты. После этого остаётся припаять два контакта для питания, шесть контактов для обмена данными и подключить антенну.

Выводы модема:

I — При поступлении входящего вызова на линии появляются импульсы с низким логическим уровнем длительностью 30 мс;

T — TX, transmit (отправка);

R — RX, receive (получение);

U — + напряжение питания 3,3–4,5 В;

K — Включение модуля при соединении с землей;

G — Земля.

Для проверки работоспособности подключим модуль к компьютеру через TTL конвертер (от англ. transistor-transistor logic — транзисторно-транзисторная логика) — упрощённой физической разновидностью стандартного интерфейса RS-232, используемого, например, для программирования простейших плат Arduino, а также для подключения различных устройств с данным интерфейсом к ПК [32, с. 23]. Часто по данному интерфейсу подключаются роутеры, сотовые телефоны, некоторые компьютерные устройства для восстановления прошивки, рисунок 13.

Посмотрим в диспетчере задач, какой COM-порт у TTL конвертера. Это порт 13. Открываем в PuTTY serial порт 13 и видим сообщение о запуске модема «MODEM: STARTUP», затем через несколько секунд надпись «+PBREADY», означающая, что была прочтена телефонная книга, даже если она пуста.

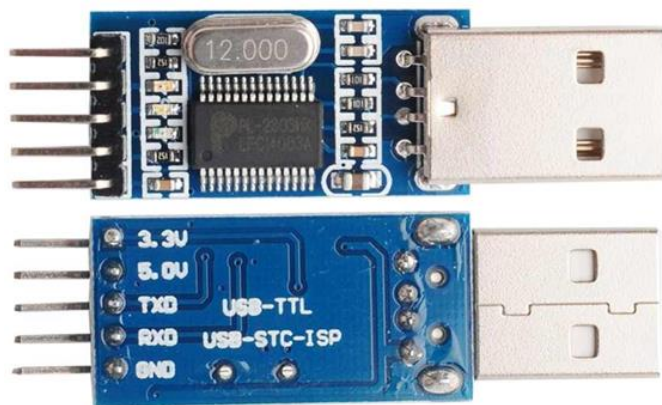


Рисунок 13. RS232 TTL конвертер

Neoway m590 можно сразу подключить к Arduino, предварительно запитав отдельно и соединив GND контакты.

Схему подключения можно увидеть на рисунке 14.

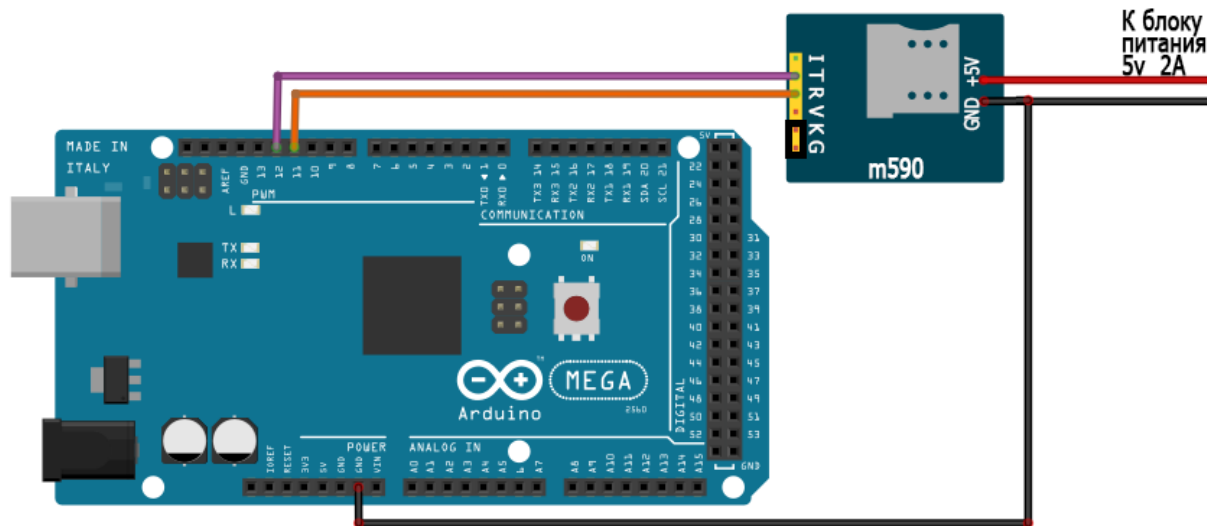


Рисунок 14. Подключение Neoway m590

Для включения модема необходимо закоротить контакты К и Г перемычкой на плате. Соответствие подключения выводов модема, Arduino и блока питания указаны в таблице 4 [32, с. 24].

Вывод модема m590	Куда подключаем
T (TX)	Arduino RXD (12 пин)
R (RX)	Arduino TXD (13 пин)
GND	GND Arduino и блока питания
+5V	Блок питания

Таблица 4. Подключение выводов модема

Затем нужно вставить SIM карту без PIN кода и подключить питание.

GSM модуль можно назвать самой главной частью системы безопасности. Ведь с помощью него будет происходить приём команд и отправка сообщений о сработавших датчиках владельцу. Для этого напишем программы, обрабатывающие показания сигналов с датчиков и совершаемые действия. Из таких мини программ и будет состоять итоговая система безопасности жилья.

2.2. Настройка, программирование и применение системы

Arduino IDE – это среда программирования, в ней есть все необходимое для написания скетча для нашей охранной системы. Для программирования будем применять адаптированный вариант языка Processing» [32, с. 16].

С помощью простого интерфейса можно быстро создавать новые скетчи. «Скетч для Arduino — это набор инструкций, определяющий пути решения стоящей перед вами задачи; иными словами, скетч — это программа» [9, с. 47].

Примеры скетчей находятся во вкладке «Файл». Например, стандартный скетч Blink [8, с. 39].

Процесс программирования платы состоит из двух этапов: написание программы-скетча. Затем загрузка скетча в контроллер нажатием одной кнопки. Скетч загружается и записывается в специальную область памяти контроллера и запускается автоматически каждый раз при включении платы. Загрузив код программы один раз, он останется там пока не перезапишется другой программой. Если памяти для хранения данных будет не хватать, есть возможность подключить дополнительный модуль памяти для Arduino.

Программа на Arduino содержит две одинаковые функции: `void setup()` и `void loop ()`. В начале кода объявляются константы и используемые порты. В разделе `void setup()` указываются команды, которые выполняются один раз [27, с. 332]. Раздел `void loop()` содержит команды, которые выполняются, всё время пока включена плата Arduino. Начав выполнение с первой команды, микроконтроллер дойдет до конца и сразу же перейдёт в начало, чтобы повторить ту же последовательность. И так бесконечное число раз (до тех пор, пока на плату подано электричество) [8, с. 40].

Так же есть возможность тестирования с помощью вывода результата работы или сообщений для отладки в последовательный порт или монитор порта [5, с. 229]. Запустим программу из примера — Примеры 08.Strings

CharacterAnalysis и откроем монитор порта, результат мы видим на рисунке 15.

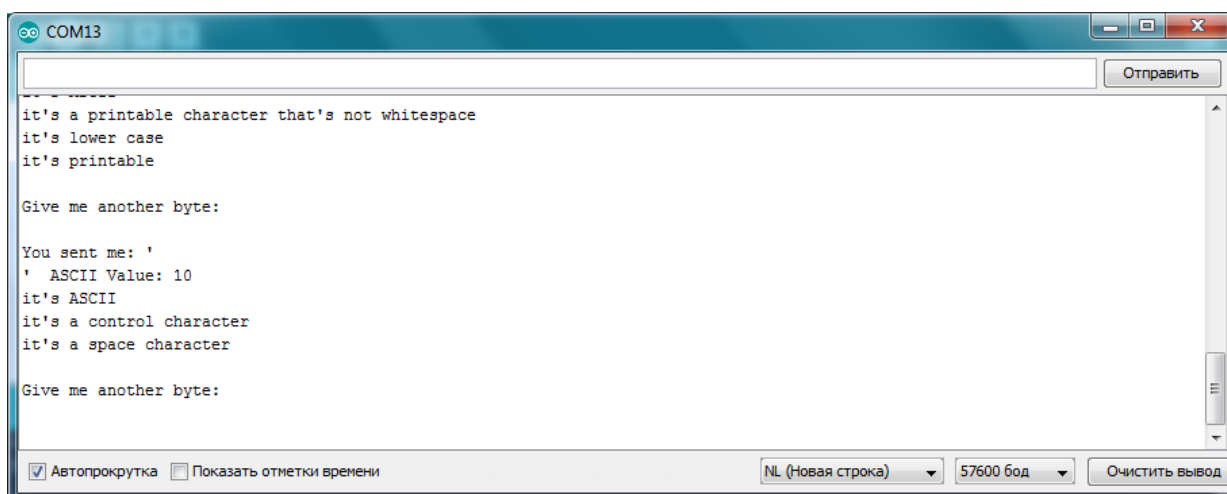


Рисунок 15. Монитор порта

Для того, чтобы установить соединение в скетче, а именно в функции `setup`, нужно вызвать метод `begin` класса `Serial` и указать скорость соединения. Выберем скорость `Serial.begin(57600)`. Что бы информация отображалась корректно, нужно выбрать такую же скорость в мониторе порта. Она указывается в «бод'ах» [8, с. 69]. Бод — скорость передачи данных по последовательному интерфейсу в битах в секунду. Для взаимодействия с компьютером следует использовать одну из предустановленных скоростей обмена: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 или 115200. Тем не менее, можно задавать и другие скорости — например, для взаимодействия с каким-либо специфичным компонентом посредством выводов 0 и 1.

«Дополнительная функциональность может быть добавлена разрабатываемым скетчам с помощью библиотек, представляющих собой специальным образом оформленный программный код, реализующий некоторый функционал, который можно подключить к создаваемому проекту. Специализированных библиотек существует множество. Обычно библиотеки пишутся так, чтобы упростить решение той или иной задачи и скрыть от разработчика детали программно-аппаратной реализации. Среда Arduino IDE поставляется с набором стандартных библиотек: `Serial`,

EEPROM, SPI, Wire и др» [24, с. 14]. Для удобной работы и установки присутствует менеджер библиотек, который позволяет прямо из меню программы скачать, установить и подключить в свой скетч множество библиотек – рисунок 16. Подключаем две необходимые библиотеки: SoftwareSerial.h — для работы GSM модуля и DHT.h — для датчика DHT11.

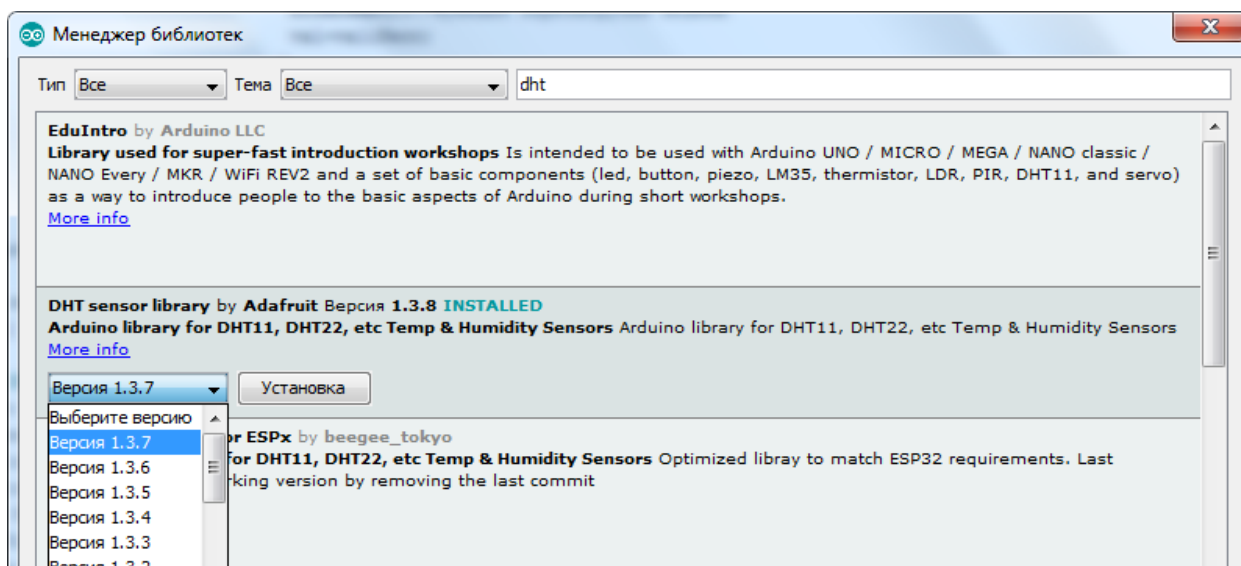


Рисунок 16. Менеджер библиотек

Для большинства библиотек Arduino есть возможность выбора версии и присутствует описание и примеры использования. Это поможет понять принцип работы библиотеки. Примеры можно доработать под свои нужды и использовать для реализации своих устройств.

Функции обработки сигнала

Как уже отмечалось выше Arduino Mega имеет 54 цифровых и 16 аналоговых портов ввода вывода. Это позволяет подключить огромное множество датчиков.

Рассмотрим обработку сигналов датчика газа (MQ-4). Вывод сигнала от платы датчика газа подключается к одному из аналоговых портов — А4. При превышении определённого уровня загазованности (который подбирается опытным путём [7, с. 106]) программа выводит сообщение «Утечка газа!» в монитор порта и подаёт звуковой сигнал пьезоизлучателем. Укажем для него цифровой пин 8, рисунок 17.

```

const int gassensorpin = A4; //Объявляем переменную для датчика газа
int gassensorvalue = 0; //Объявляем переменную для хранения значения
int buzzer = 8; //Объявляем переменную для пьезоизлучателя, указываем пин
void setup() {
    Serial.begin(57600);
}

```

Рисунок 17. Инициализация переменных

Функция loop() используется для определения загазованности и подаче сигнала при превышении порога – рисунок 18.

```

void loop()
{

gassensorvalue = analogRead(gassensorpin);
if (gassensorvalue >= 300) // задаем порог уровня загазованности
{
    tone(buzzer, 900, 200); //Подаем сигнал пьезоизлучателем
    Serial.println("Утечка газа!");
    delay(5000);
}
else
{
    noTone(buzzer);
} //Выключаем сигнал пьезоизлучателя

}

```

Рисунок 18. Обработка сигнала с датчика газа

Таким образом, при превышении порога содержания газа в монитор с периодичностью в 5 секунд будет выводиться текст «Утечка газа!» и будет происходить писк с частотой 900 герц и продолжительностью 200 миллисекунд. На рисунке 19 блок-схема срабатывания датчиков: газа и дыма.

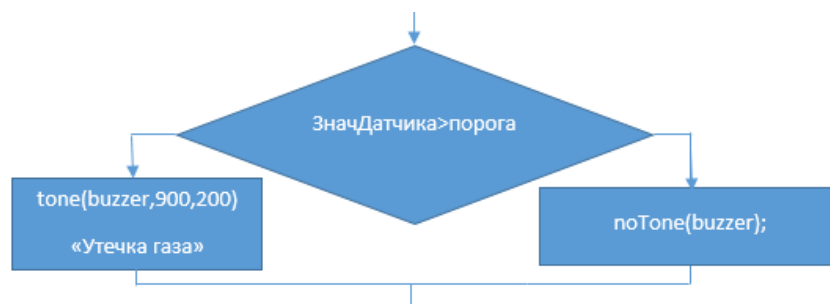


Рисунок 19. Блок-схема работы датчиков: газа и дыма

Для всех остальных датчиков логика будет такая же, будет меняться только пин: аналоговый или цифровой. При превышении определённого порога, будет отправляться сообщение, идентифицирующее датчик. Датчик

дыма подключим к А5, датчик влажности – А0, датчик движения – 9, геркон – 10, двухканальное реле 5 и 6, датчик влажности и температуры – 11, а ТХ и RX модема – 12 и 13 соответственно. Датчик газа практически схож с датчиком дыма, за исключением порога срабатывания, поэтому и код обработки будет такой же. Датчик дыма (MQ-2), может выступать в роли датчика газа, что даёт дополнительную подстраховку системе безопасности.

```
#define pirPin 9 // Объявляем переменную для датчика движения и указываем пин
#include "DHT.h" //Подключаем библиотеку для работы с датчиками температуры и влажности
#define DHTPIN 11 // Номер пина температуры
DHT dht(DHTPIN, DHT11); // Сообщаем на каком порту будет датчик
int relay1 = 6; // Объявляем переменную для реле 1 и указываем пин
int relay2 = 5; // Объявляем переменную для реле 2 и указываем пин
int water = A0; //Объявляем пин для подключения датчика дождя
unsigned int waterValue = 0; //Создаем переменную для хранения значений с датчика воды=0
int Door_sensor = 10; // Объявляем пин для подключения датчика открытия двери
int Door_val = 0; // Объявляем переменную для хранения состояния датчика
int dooroff01=0; //Переменная включения/выключения датчика двери. Датчик двери включен
int buzzer = 8;
void setup() {
  pinMode(pirPin, INPUT); // Назначаем пин датчика движения как вход
  pinMode(Door_sensor, INPUT); //Назначаем пин датчика открытия двери как вход
  pinMode(relay1, OUTPUT); //назначаем пин как реле как выход
  pinMode(relay2, OUTPUT); //назначаем пин как реле как выход
  delay(2000); //время на инициализацию модуля
  dht.begin(); //запускаем датчик DHT11
  Serial.begin(57600); //скорость порта
}
```

Рисунок 20. Инициализация датчиков: движения, двери и воды

При срабатывании датчика движения — активируется реле, которое включает свет. Показания датчика воды зависят от степени погружения датчика в жидкость и от параметров, влияющих на коэффициент передачи напряжения, например, проводимость жидкости. При полностью сухом датчике выходное напряжение и показания на аналоговом входе будут равны нулю, чем больше датчик будет погружен в воду, тем меньше будут его показания (от 0 до 1023). Если он сработал (значение меньше 300) или сработал датчик двери, тогда вызовется функция отправки СМС и раздастся писк. На рисунке 20 и 21 приведена инициализация и действия при срабатывании этих датчиков соответственно.

```

int pirVal = digitalRead(pirPin);
if(pirVal == HIGH) // Если движение
{
  digitalWrite(relay2, LOW); //Выключаем реле
  delay (10000); //хз убрать delay
}
else
{
  digitalWrite(relay2, HIGH); //Выключаем
}
}

a)

waterValue = analogRead(water); //Считываем
if(waterValue<300) //задаем порог
{
  sms("Water!", MASTERTEL); // смс
  tone(buzzer, 900, 200); //Подаем сигнал
  delay (2000);
}
else
{
  noTone(buzzer); //Выключаем сигнал
}

б)

Door_val = digitalRead(Door_sensor); //Значение
if(Door_val!=0){
  //Если не равно нулю, магнит удалён=дверь открыта
  tone(buzzer, 900, 200); //Подаем сигнал
  sms("Door open!", MASTERTEL); //смс
  delay (2000);
}
else //Иначе 0, магнит к геркону, дверь закрыта
{
  noTone(buzzer); //Выключаем сигнал
};

в)

```

Рисунок 21. Действия датчиков: а) - движения, б) - воды, в) - двери

Если, например, случится протечка воды, датчик будет постоянно срабатывать и как следствие отправлять смс-сообщения. Что бы этого не происходило, добавим новые переменные типа boolean, так называемые флаги срабатывания датчиков. Их первоначальное состояние будет “false” (т.е. датчик не сработал). Если датчик сработал и флаг имеет значение false, тогда будет выполняться код. Под конец флагу будет присвоено значение true и второй раз код датчика выполняться, не будет.

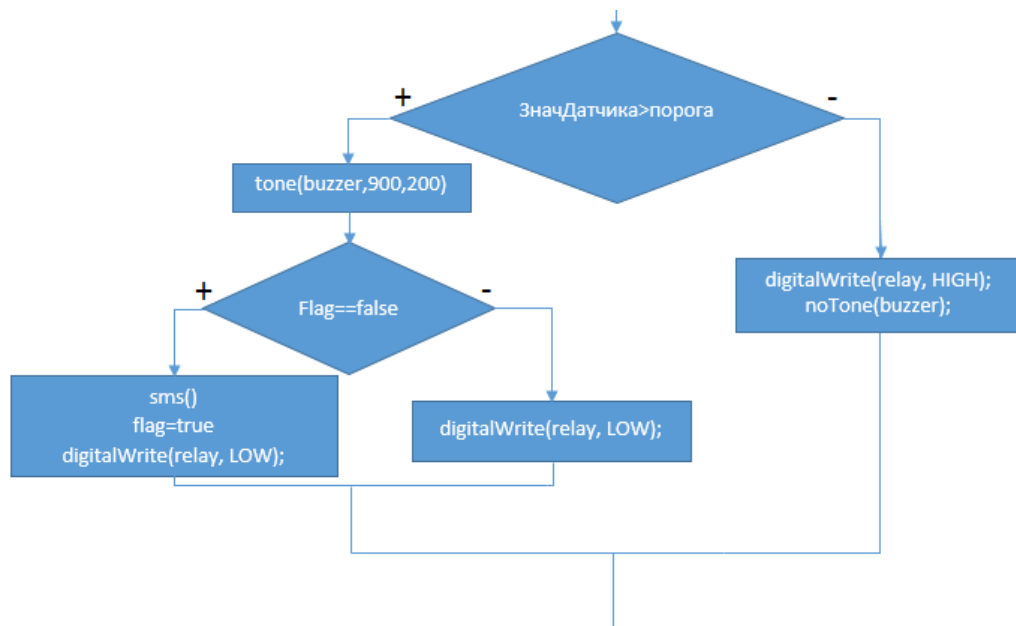


Рисунок 22. Подробная блок-схема датчиков: газа и дыма

У датчиков газа и дыма также есть флаги срабатывания, только они будут, отвечают за разовое уведомление владельца. На рисунке 22 представлена блок-схема срабатывания датчиков газа и дыма. Код этих датчиков будет выполняться, пока обнаружен газ или дым – подача звукового сигнала и активация реле, которое в свою очередь включает

вентилятор или вытяжку. При втором срабатывании SMS уже не будет опправляться, т.к. флаг будет иметь значение true.

Так же нет необходимости постоянно отслеживать температуру и влажность. Уберём весь код в функцию. На рисунке 23 представлен код функции температуры и влажности – tah() (сокращение от англ. temperature and humidity – температура и влажность)

```
void tah() //функция отправки смс с температурой и влажностью
{
    delay(2000); //ждём датчики
    float hh = dht.readHumidity(); //Измеряем влажность
    float tt = dht.readTemperature(); //Измеряем температуру
    String sms_string=""; //Объявляем строку со значениями
    sms_string="Statistic:"+'\n'+sms_string + "Temperature= " +
    String(tt)+'C'+'\n'+ "Humidity= "+String(hh)+'%'; //Формируем текст
    sms(sms_string, MASTERTEL); //Отвечаем смской
}
```

Рисунок 23. Функция tah()

Если нужно будет узнать температуру и влажность, достаточно просто вызвать функцию tah(). При вызове измеряются показания, затем формируется текст сообщения и наконец, отправляется смс-уведомление.

В некоторых случаях нужно будет отключить датчик двери, например, при входе и выходе из дома. Объявим переменную типа boolean – “dooroff” и присвоим значение false. Когда dooroff примет значение true код датчика двери не будет выполняться. Это выражение можно прочитать как «дверьотключена = истина (да) или ложь (нет).

```
if (!dooroff&&!doorflag) { //Если dooroff=false датчик включен и выполняется код
    Door_val = digitalRead(Door_sensor); //Считываем значения с датчика открытия двери
    if(Door_val!=0){ //При данных с датчика НЕ равным нулю, магнит удалён = дверь открыта
        tone(buzzer, 900, 200); //Подаем сигнал пьезоизлучателем
        sms("Door open!", MASTERTEL); //Вызов функции отправки смс
        doorflag=true; //Один раз сработал датчик
        delay (2000);
    }
    else //При данных с датчика 0, магнит прижимает к геркону и дверь закрыта
    { noTone(buzzer); //Выключаем сигнал
    };
}
else { noTone(buzzer); //Если dooroff=true датчик выключен (не обрабатывается)
};
```

а)

```
void fdooroff() //функция отключения датчика двери
{
    dooroff=true;
    sms("Door sensor off!", MASTERTEL); // смс отключен
}

void fdooron() //функция включения датчика двери
{
    dooroff=false;
    sms("Door sensor on", MASTERTEL); //смс включен
}
```

б)

Рисунок 24. Обработка двери: а) - код, б) - функции отключения и включения

Датчик двери срабатывает, если dooroff и doorflag имеют значение false – рисунок 24. То есть если датчик двери включен, и ни разу не сработал.

Функции отключения/включения датчика двери будут срабатывать, если поступило смс-сообщение с определённой командой, в ответ должно придти подтверждение: «Door sensor off!» или «Door sensor on».

Благодаря GSM модулю владелец охранной системы будет не только получать смс-уведомления от сработавших датчиков, но и отправлять определённые команды. К Arduino можно подключить несколько GSM модулей [32, с. 27].

Программирование GSM модуля

Библиотека SoftwareSerial позволяет реализовать последовательный интерфейс на любых цифровых выводах Ардуино с помощью программных средств, дублирующих функциональность UART (отсюда и название "SoftwareSerial") [9, с. 385]. Библиотека позволяет программно создавать несколько последовательных портов, работающих на скорости до 115200 бод. Для устройств, работающих с инвертированным сигналом, в библиотеке предусмотрен соответствующий параметр, включающий инвертирование.

В общем виде схема будет выглядеть так:

RX — цифровой вывод 12 (соединяем с выводом TX модуля);

TX — цифровой вывод 13 (соединяем с выводом RX модуля).

```
#include <SoftwareSerial.h> //Подключаем библиотеку для программного Rx Tx
SoftwareSerial mySerial(12,13); // RX, TX
#define MASTERTEL "+7xxxxxxxxxx" //Номер телефона владельца
void setup() {
    delay(2000); //время на инициализацию модуля
    Serial.begin(57600); //скорость порта
    mySerial.begin(57600);
    mySerial.println("AT+CMGF=1"); //режим кодировки СМС - обычный (для англ.)
    delay(100);
    mySerial.println("AT+CSGS=\"GSM\""); //режим кодировки текста
    delay(100);
    mySerial.println("AT+CMGD=1,4"); //удалить все смс
    delay(100);
    mySerial.println("AT+CNMI=2,2"); //для приёма смс
    delay(100);
    mySerial.println("AT+ENPWRSAVE=0"); //Отключить спящий режим
}
```

Рисунок 25. GSM модуль и его инициализация

Объявляем пины 12 и 13, номер телефона владельца и начинаем инициализацию GSM модуля. На рисунке 25 — код инициализации GSM модуля Neoway m590. Командой `mySerial.begin(57600)` — установим скорость передачи данных последовательного порта.

Объявляем переменную номера телефона владельца — `define MASTERTEL «+7xxxxxxxxxxx»`. Теперь можно оперировать одной переменной и в случае замены сим-карты нет необходимости искать и менять номер телефона во всей программе.

Важно понимать, что сигнализация должна знать и определять только телефон владельца, иначе любой человек может отключить такую систему.

```
if (mySerial.available()) {           //Если есть данные от GSM модуля
    comm=0;val="";delay(200); //Обнуляем переменную команд и строку val.
    //Подождём, чтобы строка успела попасть в порт целиком раньше чем будет считана
    while (mySerial.available()) {     //Сохраняем входную строку в переменную val
        soobshenie = mySerial.read();  //Читаем символы сообщения
        val += char(soobshenie);delay(10); //Собираем принятые символы в строку
    };
    }
//----- Определение факта приема смс и сравнение номера с заданным
    if (val.indexOf("+CMT") > -1) {
//если обнаружена смс
        if (val.indexOf(MASTERTEL) > -1) {
//-----если CMC от МАСТЕРТЕЛЕФОНА
            Serial.println("--- MASTER SMS DETECTED ---");
        }
    }
}
```

Рисунок 26. Определение факта и владельца смс

На рисунке 26 проиллюстрирован код определения факта приёма смс-сообщения и сравнение номера телефона с номером владельца. Если сообщение от владельца, тогда в монитор порта посылается сообщение, иначе программа ничего не делает.

Определение команд владельца осуществляется с помощью простой конструкции: переменная команд равна нулю, если в сообщении найдено одно из кодовых слов, ей присваивается номер команды. Далее идёт конструкция `switch(comm)`. Оператор `switch` сравнивает значение переменной со значением, определенном в операторах `case`. Когда найден оператор `case`, значение которого равно значению команды, выполняется программный код в этом операторе. Например, при смс-сообщении от владельца с текстом

«Tah» значение comm равно единице и в операторе switch выполняется код case 1, который в свою очередь вызывает функцию tah().

Если кодовое значение не найдено переменная comm так же остаётся нулём и в конструкции switch(comm) не найдя case 0, выполняется программный код оператора default. В нём находится функция отправки смс с текстом «Unknown command!».

```
void sms(String text, String phone) //функция отправки смс
{
    Serial.println("SMS send started");//Уведомляем в терминале о начале передачи сообщения
    mySerial.println("AT+CMGS=\"" + phone + "\"");//Указываем телефон отправки
    delay(500);
    mySerial.print(text);//Указываем сообщение
    delay(500);
    mySerial.print((char)26);//Символ завершающий передачу
    delay(500);
    Serial.println("SMS send complete");//Уведомление о завершении передачи
    delay(2000);
}
```

Рисунок 27. Функция отправки смс-сообщения

Так как отправка смс – одно из повторяющихся действий, уберём её в функцию и будем вызывать по необходимости. На рисунке 27 представлен код этой функции. Так же можно увидеть команду AT+CMGS и номер телефона. Без Arduino для отправки смс пришлось бы вводить AT+<номер телефона><CR><текст сообщения>. Символ <CR> – возврат каретки (шестнадцатеричное значение 0x0D, в языках программирования “\r”), этот символ должен присутствовать в конце каждой команды.

Вызывается же функция очень просто:

sms(<текст сообщения или данные>, <номер телефона>).

Во время выполнения функции отправки в монитор порта отсылаются служебные сообщения, уведомляющие о начале и конце процесса отправки сообщения.

Что бы исключить ситуацию с отсутствием денег на сим-карте в GSM модуле, предусмотрена команда запроса баланса, запрашиваемая на обычных мобильных телефонах командой «*105#».

При получении GSM модулем команды «Money» переменная comm принимает значение 3 и в операторе switch выполняется программа case 3, в

которой вызывается функция smsBalance(). Код этой функции можно увидеть на рисунке 28.

```
void smsBalance() //функция отправки по смс баланса
{
    mySerial.println("ATD*105#");delay(5000);//Отправляем команду запроса баланса
    if (mySerial.available()) { //Если есть данные от GSM модуля
        delay(200);//Подождём, чтобы строка успела попасть в порт целиком раньше чем будет считана
        while (mySerial.available()) { //сохраняем входную строку в переменную val
            soobshenie = mySerial.read(); //Читаем сообщение
            val += char(soobshenie); //И присваиваем переменной val
        }
        delay(10);}
    //Убираем лишнее кроме слов ОСТАТОК и сам баланс
    String moneyBalance=val.substring(val.indexOf("ОСТАТОК"),val.indexOf("p.));
    sms(moneyBalance, MASTERTEL);//Отправка баланса СМС
} else sms("Error. No sms balance!", MASTERTEL);//Если нет смс с балансом
}
```

Рисунок 28. Функция запроса баланса

Когда выполняется функция smsBalance() GSM модуль посылает команду «*105#» и ждёт сообщение с балансом. Затем с помощью функции substring() берётся часть сообщения от слова «ОСТАТОК» и, заканчивая «р.». Наконец вызывается функция отправки смс-сообщения с балансом на телефон владельца. Если сообщение с балансом не пришло, отправляется сообщение об отсутствии смс с балансом.

Бывают случаи, когда нужно включить датчики единожды сработавшие. Для этого есть команда «Resf» от словосочетания reset flag. При получении этой команды переменным срабатывания датчиков присваивается false (ни разу не сработали), а затем отправляется смс-сообщение об успешном сбросе флагов срабатывания.

Бывают случаи, когда пользователь охранной системы не знает статуса флагов срабатывания датчиков или отслеживается ли открытие двери. Для этого присутствует команда «Test». Результатом будет смс-сообщение со статусом флагов и включён ли сенсор двери (выполняется ли код).

Есть дополнительная простая команда «Sms», результатом которой будет сообщение «Command Sms – OK».

С помощью простого интерфейса и гибкого кода можно без труда добавить дополнительные команды, тем самым изменяя программу под определённые требования и условия. Например, команда открытия окна или

включение/отключение, какого-либо устройства. Шаблон для добавления новой команды очень прост:

```
if (val.indexOf("<Команда>") > -1) comm=<Номер команды>;
```

Затем в операторе switch добавить case <Номер команды>: программный код, который должен выполняться или функция.

Все команды и их описание представлены в Приложении 3.

Вообще можно напрямую «общаться» с GSM модулем с помощью AT команд, предварительно подключив его к компьютеру с помощью TTL конвертера. Затем открыть терминал (например, Putty) выбрать COM порт и скорость и нажать кнопку «open». Можно вводить AT команды. С их помощью можно узнать версию прошивки модуля, позвонить или отправить смс и настроить под определённые требования. Основные команды представлены в Приложении 4.

Все датчики проверяются по очереди в бесконечном цикле loop(). Если от датчика поступит сигнал, будет вызвана функция отправки смс-сообщения с текстом идентифицирующем датчик.

Исходный код программы представлен в Приложении 2.

С помощью Arduino, датчиков и GSM модуля можно построить неплохую бюджетную охранную систему для жилья и других объектов.

2.3. Результаты апробации, техническая документация

В итоге, собрав охранную систему и написав код программы, подключаем Arduino к компьютеру и нажимаем на кнопку загрузки скетча в микроконтроллер.

Общая схема представлена на рисунке 29.

Для проверки работы системы нужно воссоздать условия, в которых система будет эксплуатироваться. Так же нужно проверить работу всех датчиков и корректную отработку функций.

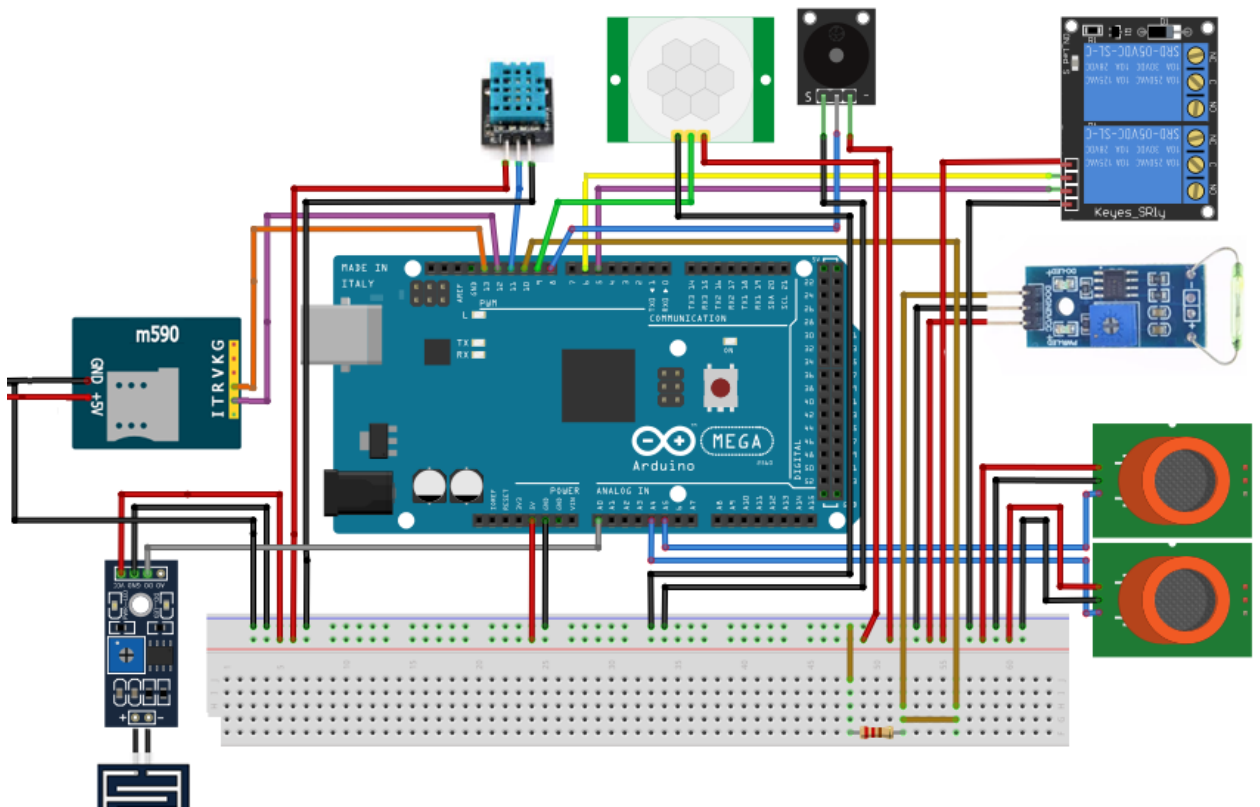


Рисунок 29. Общая схема

Что бы понять, что и когда происходило, напомним функцию вывода в монитор порта времени. Назовём такую функцию TimePrint() – рисунок 30. Здесь используется функция millis() которая возвращает количество миллисекунд с момента начала выполнения текущей программы [6, с. 266]. Это количество сбрасывается на ноль, вследствие переполнения значения, приблизительно через 50 дней, что для отладки должно хватить.

```
void TimePrint()//функция отсчёта времени от старта Arduino в формате часы:минуты:секунды
{
  unsigned long time1=millis()/1000;//Общее количество секунд, millis()/1000
  if (time1/60/60<10) { Serial.print ("0"); }//Считаем целые часы и выводим их значение
  Serial.print (time1/60/60);//Вывод кол-ва часов
  Serial.print (":");
  if (time1/60%60<10) { Serial.print ("0"); }//Высчитываем минуты, их остаток уже за вычетом часов
  Serial.print ((time1/60)%60);//Вывод минут
  Serial.print (":");
  if (time1%60<10) { Serial.print ("0"); }//Высчитываем секунды
  Serial.println (time1%60);//Вывод секунд
}
```

Рисунок 30. Функция отсчёта времени

Для того что бы понять, как формируется sms-сообщение в Arduino и формируется ли оно вообще, добавим в код вывод переменной val, которая

будет содержать текст сообщения со служебной информацией, и выводом времени – рисунок 31.

```
if (mySerial.available()) {Serial.println("-----Сформированное сообщение начало"); //Если данные от GSM модуля
comm=0;val="";delay(200); //Обнуляем переменную команд и строку val. Подождём, чтобы строка успела попасть в порт целиком
while (mySerial.available()) { //Сохраняем входную строку в переменную val
    soobshenie = mySerial.read(); //Читаем символы сообщения
    val += char(soobshenie); //Собираем принятые символы в строку
    delay(10);
}TimePrint();Serial.println(val); //ТРАССИРОВКА СООБЩЕНИЙ
Serial.println("-----Сформированное сообщение конец");
}
```

Рисунок 31. Трассировка сообщений

Для начала следует проверить датчик открытия двери. Для этого добавим вывод сообщения в монитор при его срабатывании. Затем запустим программу. Как можно увидеть из рисунка 32 — в мониторе порта сначала прошла инициализация всех компонентов, затем на третьей секунде работы программы успешно применились настройки кодировки текста и отправки сообщений. При размыкании контакта датчика двери в монитор порта было отправлено сообщение «Сработал датчик двери!», затем начался процесс отправки сообщения и успешный ответ от модуля «OK». Под буквой б) – принятое смс-сообщение.

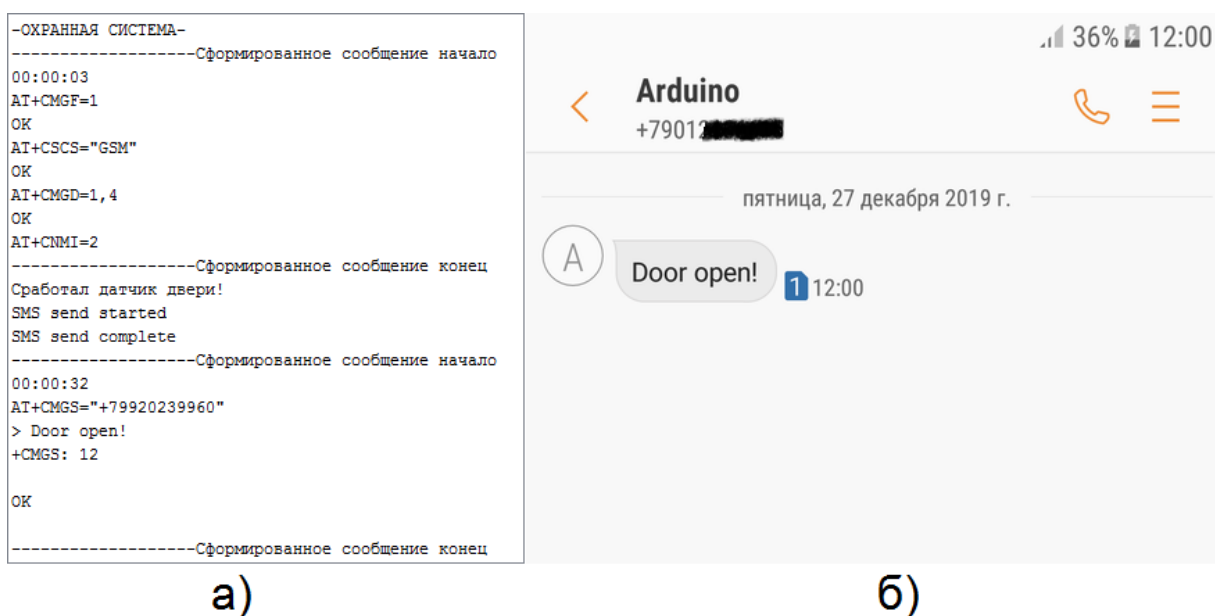


Рисунок 32. Работа охранной системы: а) - монитор порта, б) - сообщение

Следующими тестами будут проверка датчиков: газа, дыма, воды. Для проверки симитируем ситуацию, когда датчиками обнаружилась утечка газа, воды и случилось затопление. В результате в порт монитора стали

появляться сообщения о сработавших сенсорах и как следствие отправка сообщений на телефон. Так же следует проверить команду Test и Resf – результат можно увидеть на рисунке 33.

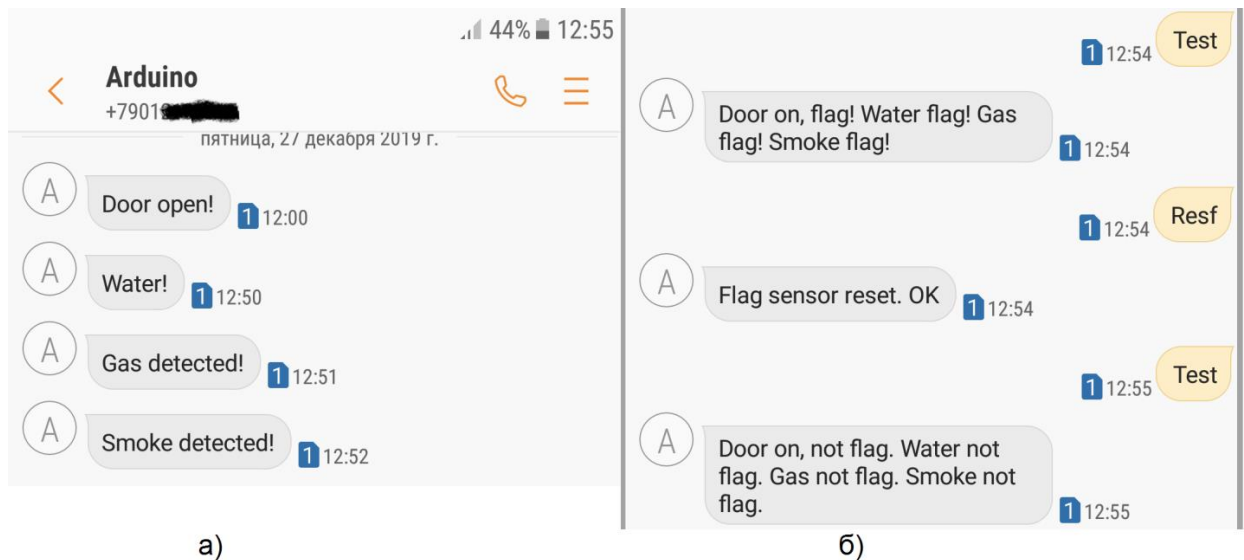


Рисунок 33. Тестирование: а) - датчиков, б) - команд

Проведём тест команды Tah – температуры и влажности – рисунок 34.

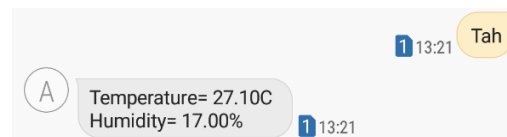


Рисунок 34. Проверка датчика температуры и влажности

Следующим этапом тестирования системы безопасности будут:

- отключение и включение сенсора двери;
- отправка любой неизвестной команды или сообщения;
- запрос оставшихся денег на сим-карте, работающей в модуле.

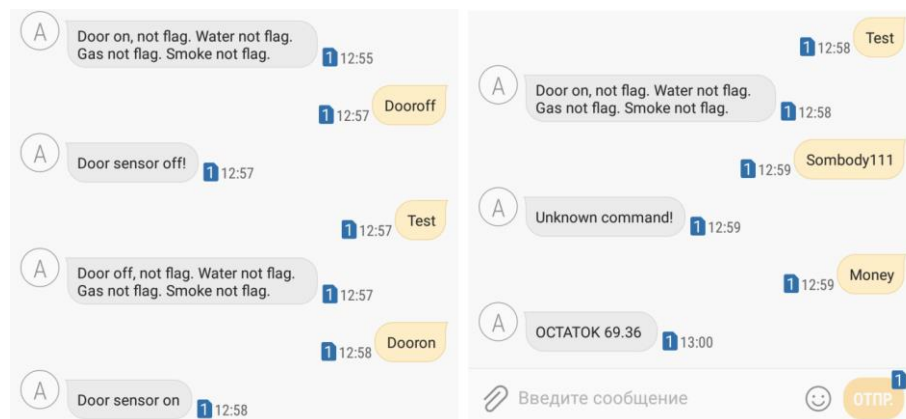


Рисунок 35. Результат выполнения команд

Как можно увидеть на рисунке 35, все команды успешно принимаются и обрабатываются. Отключив датчик двери, можно будет не беспокоиться о ложном срабатывании при входе/выходе из жилья. Если отправить команду Test, то получим сообщение о сработавших датчиках, а отправив неправильную команду, система пришлёт «Unknown command!». Так же можно не беспокоиться о ситуации, когда у системы не хватит денег на отправку сообщения, ведь баланс можно проверить командой Money.

Следующим и заключительным тестом будет проверка датчика движения и как следствие активация реле, для включения света. Когда в комнату зайдёт человек в комнате включится свет.

Тестирование было пройдено успешно и совпало с ожидаемыми результатами. Система может быть введена в эксплуатацию.

Заключение

В соответствии с поставленными задачами были получены следующие результаты:

- исследованы теоретические основы разработки систем безопасности. Определены атрибуты систем безопасности и выведены основные критерии для учета при разработке;
- отобраны компоненты системы: рассмотрены популярные платформы разработки – Seeeduino Mega, Raspberry Pi, Iskra JS, Arduino. Обоснован выбор микроконтроллера Arduino Mega, подобраны необходимые модули, в которые входят датчики, GSM-модуль и его питание;
- изучены способы подключения датчиков в сопровождении схем и описаний, подключение через макетную плату;
- подробно разобрана сама Arduino Mega, способы её питания, а также описаны цифровые и аналоговые входы/выходы;
- разработана и описана работа GSM модуля, его сообщений и команд;
- разработана программа для микроконтроллера обрабатывающая сигналы с датчиков, а в Arduino IDE написан и отлажен код для отдельных сенсоров и GSM модуля, которые в последствии объединены в единую программу.

GSM-сигнализация — это охранная система, которая использует в качестве канала передачи мобильную связь. При срабатывании датчиков тревоги владельцу приходит смс с сообщением о тревоге. Для защиты дома чаще выбирают расширенные комплекты, которые включают кроме датчиков движения, датчики размыкания, разбития стекла, противопожарные датчики и газоанализаторы. Однако датчики разбития стекла не обязательно включать в комплект, т.к. при проникновении в квартиру практически никогда не бьют стёкла. Стеклопакеты обычно выдавливают, а датчик реагирует именно на

звук бьющегося стекла. Они больше подходят для установки в помещения с витринами, которые могут разбить при попытке кражи содержимого.

GSM-сигнализацию часто выбирают из-за сравнительно низкой стоимости и простоты установки. Устройство GSM-сигнализации довольно простое. В её состав всегда входит GSM-модуль и датчики. Как правило есть возможность дополнительно подключить к системе датчики дыма и газа. Принцип работы такой охранной сигнализации заключается в прямой передаче сигнала тревоги на телефон владельца в виде смс. Можно настроить систему так, чтобы сообщения о проникновении, пожаре, утечке газа отправлялись сразу нескольким людям из заранее подготовленного списка.

GSM-сигнализации считаются автономными, потому что они не связаны с мониторинговой службой и в них не входят услуги по обслуживанию оборудования и реагированию службы безопасности. При получении тревожного сигнала владелец самостоятельно принимает решение о дальнейших действиях и единственное, что можно предпринять – вызвать полицию или спасательные службы и надеяться, что они успеют вовремя.

Помимо отправки смс и звонков, модуль M590 умеет принимать смс-сообщения и звонки, это можно обрабатывать в скетче и как-то реагировать, выполняя какие-либо действия. В довершении ко всему модуль способен подключаться к GPRS.

В более продвинутых и как следствие дорогих сигнализациях, как правило существует контрольная панель отправляющая сигнал тревоги на станцию мониторинга и в мобильное приложение, а также позволяющее управлять всей охранной системой – отслеживать активность системы, ставить/снимать объект на охрану/с охраны, управлять ключами, добавлять пользователей и т.д.

Квартирная охранная сигнализация GSM – инновация в мире услуг безопасности. Она позволяет установить систему без Интернета и телефонной линии. Подключить сигнализацию последним способом может быть дешево, но это повлечет за собой минусы. Например, преступник

сможет ее легко обрезать, и система не сработает. В GSM данные передаются через сим-карту, а это значит — сигнал тревоги не потеряется по пути на станцию мониторинга или на мобильный телефон владельца. Информация пересылается за считанные секунды, как обычное смс-сообщение. При покупке охранной системы для защиты квартиры – ориентироваться стоит только на новые технологии.

Мы изучили микроконтроллеры, и что с помощью них можно сделать. Получившаяся система обеспечения безопасности жилья на основе микроконтроллера и датчиков выгодно отличается от конкурентов стоимостью масштабируемостью, простотой установки, а также, что не менее важно, отсутствием абонентской платы.

Платформа Arduino активно развивается, и в настоящее время доступно большое число аналоговых и цифровых датчиков, предназначенных для оценки самых разнообразных аналоговых величин. К ним относятся датчики магнитного поля, температуры, влажности, освещенности, ультразвуковые датчики для измерения расстояния и многие другие.

В результате разработанная нами система безопасности – недорогая, простая в установке, настройке и эксплуатации, и что самое главное очень гибкая, что позволяет модифицировать с наименьшими затратами под практически любую задачу обеспечения безопасности объектов.

Список информационных источников

1. Аджиев Р.А. Микроконтроллеры. Arduino и IDE среда разработки / Р.А. Аджиев, Д.В. Картавцев // Проблемы обеспечения безопасности при ликвидации последствий чрезвычайных ситуаций. 2015. №2. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/mikrokontrollery-arduino-i-ide-sreda-razrabotki> (дата обращения: 19.12.19).
2. Альбекова З.М. Анализ эволюции технологии беспроводных сетей и прогнозы развития инфокоммуникационных сетей в России / З.М. Альбекова, В.О. Квашурин, Н.А. Тутик // ИВД. 2016. №4 (43). [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/analiz-evolyutsii-tehnologii-besprovodnyh-setey-i-prognozy-razvitiya-infokommunikatsionnyh-setey-v-rossii> (дата обращения: 18.12.19).
3. Багутдинов Р.А. Проектирование модульной мультисенсорной системы для задач мониторинга окружающей среды на базе arduino // Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика. 2019. №1. — С. 173-180. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/proektirovanie-modulnoy-multisensornoy-sistemy-dlya-zadach-monitoringa-okruzhayuschey-sredy-na-baze-arduino> (дата обращения: 16.12.19).
4. Бакытбек Е.Б. Датчики для измерения слабых магнитных полей // Наука и образование сегодня. — 2016. — №4(5). — С. 30-31. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/datchiki-dlya-izmereniya-slabyh-magnitnyh-poley> (дата обращения: 20.12.19)
5. Белов А.В. Arduino. От азов программирования до создания практических устройств. — СПб.: Наука и Техника, 2018. - 480 с.
6. Белов А.В. Программирование ARDUINO. Создаем практические устройства. — СПб.: Наука и Техника, 2018. — 272 с.
7. Бирюков А.А. Умные устройства безопасности на микроконтроллерах Atmel. М.: ДМК Пресс, 2017. - 162 с.

8. Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ. — СПб.: БХВ-Петербург, 2015. — 336 с.

9. Бокселл Дж. Изучаем Arduino. 65 проектов своими руками. — СПб.: Питер, 2017. — 400 с.

10. Диковицкий В.В. Программно-аппаратный комплекс удаленного присутствия / В.В. Диковицкий, М.М. Панфилов, А.А. Менькова // Труды Кольского научного центра РАН. 2017. №3-8 (8). [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/programmno-apparatnyy-kompleks-udalennogo-prisutstviya> (дата обращения: 19.12.19).

11. Довгаль В.А. Построение IoT-системы безопасности на базе Arduino / В.А. Довгаль, Д.В. Довгаль // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2018. №3 (226). — С. 142-147. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/postroenie-iot-sistemy-bezopasnosti-na-baze-arduino> (дата обращения: 19.12.19).

12. Иванов В.Н. Методика эффективного обучения робототехнической программно-элементной базе в школе / В.Н. Иванов, А.В. Иванов // Ped.Rev.. 2018. №1 (19). — С. 157-166. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/metodika-effektivnogo-obucheniya-robototekhnicheskoy-programmno-elementnoy-baze-v-shkole> (дата обращения: 20.12.19).

13. Иго Т. Arduino, датчики и сети для связи устройств: Пер. с англ. — 2-е изд. — СПб.: БХВ-Петербург, 2015. — 544 с.

14. Карвинен Т., Карвинен К., Валтокари В.. Делаем сенсоры: проекты сенсорных устройств на базе Arduino и RaspBerry Pi.: Пер. с англ. — М.: ООО «И.Д. Вильямс»: 2015. — 432 с.

15. Князьков А.В. Система онлайн-контроля значений показателей здоровья человека с передачей по каналу GSM / А.В. Князьков, В.С. Королев, В.М. Чайковский // НиКа. 2017. №. [Электронный ресурс]. — Режим доступа:

<https://cyberleninka.ru/article/n/sistema-onlayn-kontrolya-znacheniy-pokazateley-zdorovya-cheloveka-s-peredachej-po-kanalu-gsm> (дата обращения: 19.12.19).

16. Козлов А.С. Применение оптоволоконных и термочувствительных элементов в системах охранной и пожарной сигнализации / А.С. Козлов, Р.Я. Лабковская, В.Л. Ткалич, О.И. Пирожникова // Известия высших учебных заведений. Приборостроение. — 2018. — №4. — С. 356-359. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/primenenie-optovolokonnyh-i-termochuvstvitelnyh-elementov-v-sistemah-ohrannoy-i-pozharnoy-signalizatsii> (дата обращения: 20.12.19)

17. Кузнецов А.А. Создание суперкомпьютера на основе одноплатных вычислительных устройств Raspberry Pi / А.А.Кузнецов, Г.А. Кузнецов // Решетневские чтения. — 2017. — С. 80-81. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/sozдание-superkompyutera-na-osnove-odnoplattnyh-vychislitelnyh-ustroystv-raspberry-pi> (дата обращения: 20.12.19)

18. Ларина Е. А. Методика проектирования цифровых систем на базе AVR – микроконтроллеров // Международный научно-исследовательский журнал. — 2016. — №6. — С. 87–94. [Электронный ресурс]. — Режим доступа: <https://research-journal.org/technical/metodika-proektirovaniya-cifrovyyh-sistem-upravleniya-na-baze-avr-mikrokontrollerov/> (дата обращения: 16.12.19)

19. Лихолетова М.В. Технический обзор микросхемы приемопередатчика nRF24L01+ / М.В. Лихолетова, В.А. Устюгов // Juvenis scientia. 2016. №6. — С. 4-5. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/tehnicheskiy-obzor-mikroshemy-priemoperedatchika-nrf24l01> (дата обращения: 16.12.19).

20. Макаров С.Л. Arduino Uno и Raspberry Pi 3: от схемотехники к интернету вещей. — М. : ДМКПресс, 2018. — 204 с.

21. Михалин Е.С. Разработка внутренней системы мониторинга окружающей среды с использованием недорогостоящей сенсорной сети / Е.С. Михалин, М.В. Лунев, А.В. Проскурин // Инновационная наука. 2018. №2. - С. 11-13. [Электронный ресурс]. — Режим доступа:

<https://cyberleninka.ru/article/n/razrabotka-vnutrenney-sistemy-monitoringa-okruzhayushey-sredy-s-ispolzovaniem-nedorogostoyashey-sensornoj-seti> (дата обращения: 20.12.19).

22. Монк С. Програмируем Arduino: Основы работы со скетчами. 2-е изд. — СПб.: Питер, 2017. — 208 с.

23. Монк С. Электроника. Сборник рецептов: готовые решения на базе Arduino и Raspberry Pi. : Пер. с англ. — СПб.: ООО "Диалектика": 2019. — 480 с.

24. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. — СПб.: БХВ-Петербург, 2016. — 320 с.

25. Петин В.А. Проекты с использованием контроллера Arduino. — 3-е изд., СПб.: БХВ-Петербург, 2019 - 496 с.

26. Петин В.А. Создание умного дома на базе Arduino. — М.: ДМК Пресс, 2018. — 180 с.

27. Платт Ч. Электроника для начинающих: Пер. с англ. — 2-е изд. — СПб.: БХВ-Петербург, 2017. — 416с.

28. Ревич Ю.В. Азбука электроники. Изучаем Arduino / Ю. Ревич. — Москва: Издательство АСТ: Кладезь, 2017. — 224 с.

29. Салтыкова М.М. Новый подход к анализу влияния погодных условий на организм человека / М.М. Салтыкова, И.П. Бобровницкий, М.Ю. Яковлев, А.Д. Банченко, С.Н. Нагорнев // Гигиена и санитария. 2018. №11. — С. 1038-1042. [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/novyuy-podhod-k-analizu-vliyaniya-pogodnyh-usloviy-na-organizm-cheloveka> (дата обращения: 16.12.19).

30. Харченко А.В. Разработка цифровых устройств мониторинга и контроля на базе микропроцессора / А.В. Харченко, А.О. Пасюк, П.И. Поленичкин // Вестник Волгоградского государственного университета. Серия 10: инновационная деятельность. — 2017. — №3. — С. 35-44. [Электронный ресурс]. — Режим доступа:

<https://cyberleninka.ru/article/n/razrabotka-tsifrovyyh-ustroystv-monitoringa-i-kontrolya-na-baze-mikroprotsessora> (дата обращения: 16.12.19)

31. Ярнольд С. Arduino для начинающих : самый простой пошаговый самоучитель / Стюарт Ярнольд ; [пер. с англ. М. Райтман]. — М.: Эксмо, 2017. — 256 с.

32. Яценков В.С. Здоровье, спорт и окружающая среда в проектах Arduino. — СПб.: БХВ-Петербург, 2020. — 336 с.

Приложения

Приложение 1

Правила работы с системой

1. Перед установкой системы необходимо:

1.1. Убедиться в отсутствии повреждений микроконтроллера, датчиков и соединительных проводов.

1.2. Поместить микроконтроллер в закрытое помещение исключаящее повреждение от погодных условий или несанкционированного доступа.

2. Непосредственно перед включением охранной системы:

2.1. Осмотреть жилой объект с целью выявления и удаления с территории посторонних лиц и животных.

2.2. Закрыть окна, двери и решетки на запирающие устройства (защелки, замки), если такие имеются.

2.3. Осмотреть средства сигнализации, обратив особое внимание на то, чтобы извещатель и датчики охранной сигнализации не были закрыты посторонними предметами.

3. При включении сигнализации необходимо:

3.1. Закрыть входную дверь и прекратить движение в зоне действия датчика движения.

3.2. Убедиться в работоспособности системы, отправив смс-команду «Test».

3.3. При необходимости сбросить флаги срабатывания датчиков командой «Resf».

3.4. Убедиться в наличии средств на сим-карте охранной системы, отправив команду «Money».

4. Отключение системы:

4.1. Отправить смс-команду «Dooroff».

4.2. Дождаться подтверждения отключения датчика двери смс ответом «Door sensor off!».

ВАЖНО ПОМНИТЬ КОД МИКРОКОНТРОЛЛЕРА ИЗМЕНЯЕТСЯ
ТОЛЬКО НЕПОСРЕДСТВЕННО ПРИ ФИЗИЧЕСКОМ КОНТАКТЕ С
ПОДКЛЮЧЕНИЕМ ПЛАТЫ К КОМПЬЮТЕРУ.

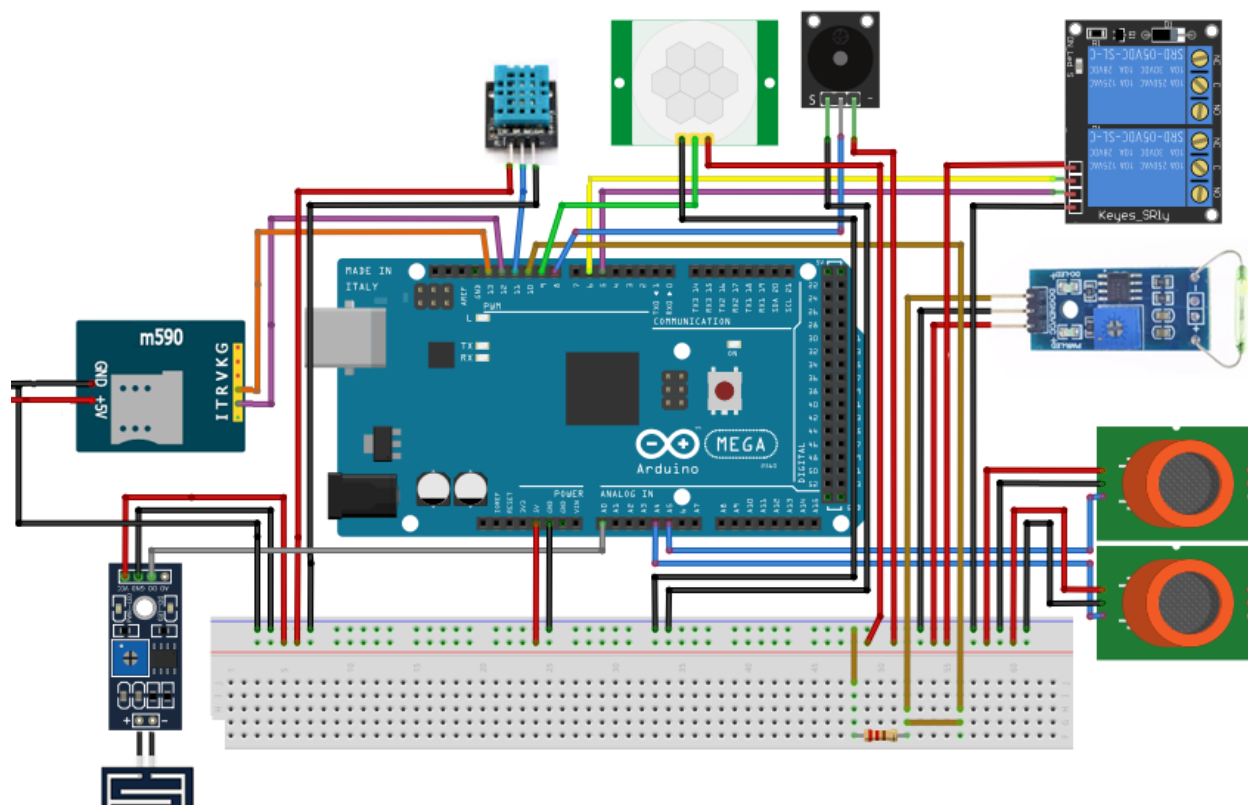
Способы настройки системы:

1. Изменение кода программы под свои нужды.
2. Изменение порога срабатывания датчиков в коде программы.
3. Изменение чувствительности датчиков подстроечными резисторами.

Общий перечень датчиков и компонентов:

1. Arduino Mega 2560 r3 — 1шт.
2. Датчик дыма MQ-2 — 1шт.
3. Датчик газа MQ-4 — 1шт.
4. Модуль геркона с подстроечным резистором — 1шт.
5. Датчик протечки воды — 1шт.
6. Модуль GSM Neoway m590 — 1шт.
7. 2-канальный релейный модуль HL-52S — 1шт.
8. Датчик температуры и влажности DHT11 — 1шт.
9. Датчик движения HC-SR501 — 1шт.
10. Резистор 10 кОм — 1шт.
11. Пьезоизлучатель — 1шт.
12. Соединительные провода — в зависимости от площади и удалённости датчиков.
13. Макетная плата (bradboard), при необходимости — 1шт.

Общая схема подключения



Приложение 2

```
#include <SoftwareSerial.h> //Подключаем библиотеку для программного Rx Tx
SoftwareSerial mySerial(12,13); // RX, TX

#define pirPin 9 // Объявляем переменную для датчика движения и указываем пин

#include "DHT.h" //Подключаем библиотеку для работы с датчиками температуры и
влажности

#define DHTPIN 11 // Номер пина температуры

DHT dht(DHTPIN, DHT11); // Сообщаем на каком порту будет датчик

#define MASTERTEL "+79920239960" //Номер телефона владельца

int soobshenie = 0; //Переменная символа сообщения

String val = ""; //Переменная сообщения

int comm=0; //Переменная команд по смс

int relay1 = 6; // Объявляем переменную для реле 1 и указываем пин

int relay2 = 5; // Объявляем переменную для реле 2 и указываем пин

//int relay3 = 4; // Объявляем переменную для дополнительного реле 3 и указываем пин

//int relay4 = 7; // Объявляем переменную для дополнительного реле 4 и указываем пин

const int smoksensornpin = A5; //Объявляем переменную для сенсора дыма, указываем пин
MQ-2

const int gassensornpin = A4; //Объявляем переменную для датчика газа(метана), указываем
пин MQ-4

int smoksensornvalue = 0; //Объявляем переменную для хранения значения с датчика газа и
дыма приравниваем ее к 0

int gassensornvalue = 0; //Объявляем переменную для хранения значения с датчика
природного газа приравниваем ее к 0

int water = A0; //Объявляем пин для подключения датчика дождя

unsigned int waterValue = 0; //Создаем переменную для хранения значений с датчика воды
и приравниваем его к 0

int Door_sensor = 10; // Объявляем пин для подключения датчика открытия двери

int Door_val = 0; // Объявляем переменную для хранения состояния датчика

int buzzer = 8; //Объявляем переменную для пьезоизлучателя, указываем пин
```

```

bool dooroff=false;//Переменная включения/выключения датчика двери. Датчик двери
включен

//Флаги срабатывания датчиков

bool doorflag=false;//Флаг срабатывания датчика двери

bool waterflag=false;//Флаг срабатывания датчика воды

bool gasflag=false;//Флаг срабатывания датчика газа MQ-4

bool smokeflag=false;//Флаг срабатывания датчика дыма MQ-2

void setup() {

  pinMode(pirPin, INPUT); // Назначаем пин датчика движения как вход

  pinMode(Door_sensor, INPUT); //Назначаем пин датчика открытия двери как вход

  pinMode(relay1, OUTPUT); //назначаем пин как реле как выход

  pinMode(relay2, OUTPUT); //назначаем пин как реле как выход

  // pinMode(relay3, OUTPUT); //назначаем пин как доп.реле как выход

  // pinMode(relay4, OUTPUT); //назначаем пин как доп.реле как выход

  delay(2000); //время на инициализацию модуля

  dht.begin();//запускаем датчик DHT11

  Serial.begin(57600); //скорость порта

  Serial.println("-ОХРАННАЯ СИСТЕМА-");

  mySerial.begin(57600);

  mySerial.println("AT+CMGF=1"); //режим кодировки СМС - обычный (для англ.)

  delay(100);

  mySerial.println("AT+CSCS=\"GSM\""); //режим кодировки текста

  delay(100);

  mySerial.println("AT+CMGD=1,4"); //удалить все sms

  delay(100);

  mySerial.println("AT+CNMI=2,2"); //для приёма смс

  delay(100);

  mySerial.println("AT+ENPWRSAVE=0"); //Отключить спящий режим

```

```

}

void loop()

{
//Начало дым MQ-2

smoksensorvalue = analogRead(smoksensorpin);//Считываем значение с датчика газа и дыма
if (smoksensorvalue >= 600) // задаем порог уровня загазованности
{Serial.print("Smoke: ");

Serial.print(smoksensorvalue);

tone(buzzer, 900, 200); //Подаем сигнал пьезоизлучателем

    if(!smokeflag) {sms("Smoke detected!", MASTERTEL);smokeflag=true;}//Один раз
сработал датчик. Вызов функции отправки смс

digitalWrite(relay1, LOW); // Включаем реле, к которому подключен вентилятор
delay(5000);
}
else
{
digitalWrite(relay1, HIGH); //Выключаем реле с вентилятором
noTone(buzzer); //Выключаем сигнал пьезоизлучателя
}

//Конец дым. MQ-2


//Начало метан. MQ-4

gassensorvalue = analogRead(gassensorpin);
if (gassensorvalue>= 300) // задаем порог уровня загазованности
{Serial.print("Gas: ");

Serial.print(gassensorpin);

tone(buzzer, 900, 200); //Подаем сигнал пьезоизлучателем

```

```
    if(!gasflag) {sms("Gas detected!", MASTERTEL);gasflag=true;}//Один раз сработал датчик. Вызов функции отправки смс
```

```
    digitalWrite(relay1, LOW); // Включаем реле, к которому подключен вентилятор
```

```
    delay(5000);
```

```
    }
```

```
    else
```

```
    {digitalWrite(relay1, HIGH);
```

```
        noTone(buzzer);
```

```
    } //Выключаем сигнал пьезоизлучателя
```

```
//Конец метан
```

```
//Начало движение
```

```
int pirVal = digitalRead(pirPin);
```

```
if(pirVal == HIGH) // Если происходит движение
```

```
{
```

```
    //sms("Motion!", MASTERTEL);//Вызов функции отправки смс
```

```
    digitalWrite(relay2, LOW); //то Включаем реле, к которому подключено освещение
```

```
    //digitalWrite(relay3, LOW); // включаем доп.реле, к которому подключено освещение
```

```
    //digitalWrite(relay4, LOW); // включаем доп.реле, к которому подключено освещение
```

```
    delay (10000);//хз убрать delay
```

```
}
```

```
else
```

```
{
```

```
    digitalWrite(relay2, HIGH); //Выключаем реле
```

```
    //digitalWrite(relay3, HIGH); //Выключаем доп.реле
```

```
    //digitalWrite(relay4, HIGH); //Выключаем доп.реле
```

```
}
```

```
//Финиш движение
```

```

//Начало вода. Маленькая капля уже - 200. По обычному около 1000

waterValue = analogRead(water); //Считываем значения с датчика дождя

if(waterValue<300&&!waterflag) //задаем пороговое значение срабатывания сигнализации
{
    sms("Water!", MASTERTEL); //Вызов функции отправки смс
    tone(buzzer, 900, 200); //Подаем сигнал пьезоизлучателем
    waterflag=true; //Один раз сработал датчик
    delay (2000);
}
else
{
    noTone(buzzer); //Выключаем сигнал
}

//конец вода


//Начало дверь.

if (!dooroff&&!doorflag) { //Если dooroff=false датчик включен и выполняется код
    Door_val = digitalRead(Door_sensor); //Считываем значения с датчика открытия
двери
    if(Door_val!=0){ //При данных с датчика НЕ равным нулю, магнит удалён =
дверь открыта. Датчик сработал
        tone(buzzer, 900, 200); //Подаем сигнал пьезоизлучателем
        sms("Door open!", MASTERTEL); //Вызов функции отправки смс
        doorflag=true; //Один раз сработал датчик
        delay (2000);
    }
}

```

```

else//При данных с датчика 0, магнит примыкает к геркону и дверь закрыта

{noTone(buzzer); //Выключаем сигнал

};

}

else {noTone(buzzer);//Если dooroff=true датчик выключен (не обрабатывается)

};

//Конец дверь

if (mySerial.available()) {Serial.println("-----Сформированное сообщение
начало");      //Если есть данные от GSM модуля

comm=0;val="";delay(200);//Обнуляем переменную команд и строку val. Подождём,
чтобы строка успела попасть в порт целиком раньше чем будет считана

while (mySerial.available()) {    //Сохраняем входную строку в переменную val

soobshenie = mySerial.read(); //Читаем символы сообщения

val += char(soobshenie); //Собираем принятые символы в строку

delay(10);

        }TimePrint();Serial.println(val);//ТРАССИРОВКА СООБЩЕНИЙ

        Serial.println("-----Сформированное сообщение конец");

    }

//----- Определение факта приема смс и сравнение номера с заданным

if (val.indexOf("+CMT") > -1) {

//если обнаружена смс (для определения звонка вместо "+CMT" можно вписать "RING",
можно реагировать только на факт звонка, трубку Neoway m590 не поднимет)

    if (val.indexOf(MASTERTEL) > -1) {

//-----если СМС от МАСТЕРТЕЛЕФОНА

        Serial.println("--- MASTER SMS DETECTED ---");

//----- Поиск кодового слова в смс если МАСТЕРТЕЛЕФОН

```

//Поиск слова. Если слово найдено выражение val.indexOf("<Слово>") будет больше -1. Если не найдено результат -1, переменная comm не меняется и равна 0 (неизвестная команда)

if (val.indexOf("Tah") > -1) comm=1;//Если обнаружено кодовое слово Tah. Команда 1. Отправка температуры и влажности

if (val.indexOf("Sms") > -1) comm=2;//Если обнаружено кодовое слово Door. Команда 2. Обработка двери

if (val.indexOf("Money") > -1) comm=3;//Если обнаружено кодовое слово Money. Команда 3. Отправка баланса сим-карты

if (val.indexOf("Test") > -1) comm=4;//Если обнаружено кодовое слово Test. Команда 4. Отправка сообщения состояния датчиков

if (val.indexOf("Dooroff") > -1) comm=5;//Если обнаружено кодовое слово Dooroff. Команда 5. Отключение датчика двери. Отправка сообщения

if (val.indexOf("Dooron") > -1) comm=6;//Если обнаружено кодовое слово Dooron. Команда 6. Включение датчика двери. Отправка сообщения

if (val.indexOf("Resf") > -1) comm=7;//Если обнаружено кодовое слово Resf. Команда 7. Сброс срабатывания всех датчиков

//Шаблон для новой команды: if (val.indexOf("<Кодовое слово>") > -1) comm=<Номер команды>;

mySerial.println("AT+CMGD=1,4");//Очищаем все Sms

delay(1000);//Ждём 1 сек

switch(comm) { //Выбор действий по командам. Шаблон: case <Номер команды>:<Код или вызов функции>;break;

case 1: tah();//Вызов функции температуры и влажности
break;

case 2: sms("Command Sms - OK", MASTERTEL);//Отправка смс "Command Sms - OK" на команду Sms
break;

case 3: smsBalance();//Вызов функции отправки баланса сим-карты по смс
break;


```

case 4:          ftestd();//Вызов функции опроса срабатывания датчиков
                break;

case 5:          fdooroff();//Вызов функции отключения датчика двери
                break;

case 6:          fdooron();//Вызов функции включения датчика двери
                break;

case 7:          fresf();//Вызов функции сброса срабатывания датчиков
                break;

default:         sms("Unknown command!", MASTERTEL);//Если другое слово или
вообще пустое сообщение. НЕ РАСПОЗНАНА КОМАНДА.

};

//----- конец если смс от МАСТЕРТЕЛЕФОНА

} else {

    //СМС Обнаружена, но не от МАСТЕРТЕЛЕФОНА

    Serial.println("--- NO MASTER SMS ---");

    val = "";//на всякий случай, ХЗ ПОМОЖЕТ НЕТ, скорее не нужно т.к. вал очищается
когда ардуино видит смс

}

}

//-----смс не обнаружена, ничего не делать


} //Конец loop() {

***Раздел функций***

```

```

void sms(String text, String phone) //Функция отправки смс
{
    Serial.println("SMS send started");//Уведомляем в терминале о начале передачи сообщения
    mySerial.println("AT+CMGS=\"" + phone + "\"");//Указываем телефон отправки
    delay(500);
    mySerial.print(text);//Указываем сообщение
    delay(500);
    mySerial.print((char)26);//Символ завершающий передачу
    delay(500);
    Serial.println("SMS send complete");//Уведомление о завершении передачи
    delay(2000);
}

```

```

void tah() //Функция отправки смс с температурой и влажностью
{
    delay(2000);//ждём датчики

    float hh = dht.readHumidity(); //Измеряем влажность

    float tt = dht.readTemperature(); //Измеряем температуру

    String sms_string="";//Объявляем переменную, куда присвоим значение
    температуры и влажности

    sms_string = "Statistic:" + '\n' + sms_string + "Temperature="
    "+" + String(tt) + 'C' + '\n' + "Humidity=" + String(hh) + "%";//Формируем текст сообщения

    sms(sms_string, MASTERTEL); //Отвечаем смской
}

```

```

void smsBalance() //Функция отправки по смс баланса
{
    mySerial.println("ATD*105#");//Отправляем команду запроса баланса
}

```

```

if (mySerial.available()) {      //Если есть данные от GSM модуля

    delay(200);//Подождём, чтобы строка успела попасть в порт целиком
раньше чем будет считана

    while (mySerial.available()) {    //сохраняем входную строку в
переменную val

        soobshenie = mySerial.read();    //Читаем сообщение

        val += char(soobshenie);        //И присваиваем переменной val

        delay(10);}

    String
moneyBalance=val.substring(val.indexOf("ОСТАТОК"),val.indexOf("р."));//Убираем лишнее
кроме слов ОСТАТОК и сам баланс

    sms(moneyBalance, MASTERTEL);//Отправка баланса СМС

    } else sms("Error. No sms balance!", MASTERTEL);//Если нет
смс с балансом

}

```

```

void fdooroff() //Функция отключения датчика двери

{

    dooroff=true;

    sms("Door sensor off!", MASTERTEL);//Отправка смс. Датчик двери отключен

}

```

```

void fdooron() //Функция включения датчика двери

{

    dooroff=false;

    sms("Door sensor on", MASTERTEL);//Отправка смс. Датчик двери включен

}

```

```

void fresf() //Функция сброса срабатывания всех датчиков

{

```

```

    doorflag=false;//Флаг срабатывания датчика двери

    waterflag=false;//Флаг срабатывания датчика воды

```

```

gasflag=false;//Флаг срабатывания датчика газа MQ-4
smokeflag=false;//Флаг срабатывания датчика дыма MQ-2
sms("Flag sensor reset. OK", MASTERTEL);//Отправка смс. Флаги сброшены
}

void TimePrint()//Функция отсчёта времени от старта Arduino в формате
часы:минуты:секунды
{
unsigned long time1=millis()/1000;//Общее количество секунд, millis()/1000
if (time1/60/60<10) { Serial.print ("0"); }//Считаем целые часы и выводим их значение
Serial.print (time1/60/60);//Вывод кол-ва часов
Serial.print (":");
if (time1/60%60<10) { Serial.print ("0"); }//Высчитываем минуты, их остаток уже за
вычетом целых часов
Serial.print ((time1/60)%60);//Вывод минут
Serial.print (":");
if (time1%60<10) { Serial.print ("0"); }//Высчитываем секунды
Serial.println (time1%60);//Вывод секунд
}

void ftestd() {
String test_string="";
if (!dooroff) test_string+="Door on, "; else test_string+="Door off, ";
if (!doorflag) test_string+="not flag. "; else test_string+="flag! ";
if (!waterflag) test_string+="Water not flag. "; else test_string+="Water flag! ";
if (!gasflag) test_string+="Gas not flag. "; else test_string+="Gas flag! ";
if (!smokeflag) test_string+="Smoke not flag."; else test_string+="Smoke flag!";
sms(test_string, MASTERTEL);      //Отвечаем смской
}

```

Приложение 3

Tah — Отправка температуры и влажности на телефон владельца.

Sms — Отправка смс "Command Sms - ОК".

Money — Отправка баланса сим-карты находящейся в охранной системе.

Test — Отправка сообщения состояния датчиков.

Dooroff — Отключение датчика двери. Отправка сообщения.

Dooron — Включение датчика двери. Отправка сообщения.

Resf — сброс срабатывания всех датчиков (флагов срабатывания).

Приложение 4

- **АТ+КОМАНДА=?** — Синтаксис запроса списка параметров и диапазонов их значений поддерживаемых командой (shield вернёт строку с корректными параметрами для отправки данной команды).
- **АТ+КОМАНДА?** — Синтаксис запроса данных (shield вернёт текущие значения параметров команды).
- **АТ+КОМАНДА=ПАРАМЕТР(Ы)** — Синтаксис установки данных (shield установит указанные значения параметров команды).
- **АТ+КОМАНДА** — Синтаксис запроса/установки данных определяемых внутренними процессами модуля.
- Если синтаксис команды неверен, то возвращается **ERROR**.
- Если синтаксис команды верен, но допущены ошибки в параметрах, то возвращается **+CME ERROR** или **+CMS ERROR**.
- Если команда выполнена успешно, то в большинстве случаев возвращается **OK**.

Основные команды:

АТ&F — Восстановить заводские настройки.

АТ&W — Сохранить текущие настройки в области EEPROM.

АТ+СМЕЕ=ЧИСЛО — Установить формат вывода сообщений об ошибках мобильного оборудования:

ЧИСЛО — значение от 0 до 2 определяющее формат вывода ошибок:

0 — При возникновении ошибок возвращать только текст «ERROR», без указания кода ошибки.

1 — При возникновении ошибок возвращать код ошибки.

2 — При возникновении ошибок возвращать текст описывающий возникшую ошибку.

АТ+CSCS=ИМЯ — Установить набор символов терминального оборудования:

ИМЯ — название набора символов:

"GSM" — 7-битный алфавит GSM (3GPP TS 23.038).

"HEX" — шестнадцатеричный режим, без кодировки.

"PCSR936" — таблица символов PC страница 437.

"UCS2" — 16-битная таблица символов ISO / IEC10646.

AT+CMGF=ФОРМАТ — установка формата SMS сообщений (текстовый / PDU):

ФОРМАТ — представлен цифрой 0 или 1:

0 — SMS сообщения принимаются и отправляются в формате PDU.

1 — SMS сообщения принимаются и отправляются в текстовом формате.

AT+CMGD=ИНДЕКС,СТАТУС — Команда удаляет сообщения из области памяти указанной командой CPMS.

ИНДЕКС — представлен числом указывающим на номер (ячейку) удаляемого SMS сообщения из предпочтительной области памяти.

СТАТУС — сообщений представлен цифрой от 0 до 4:

0 — Одно сообщение находящееся в ячейке ИНДЕКС.

1 — Все прочитанные сообщения.

2 — Все прочитанные и отправленные сообщения.

3 — Все прочитанные, отправленные и неотправленные сообщения.

4 — Все сообщения.

AT+CNMI=РЕЖИМ,СООБЩЕНИЯ — Индикация новых сообщений:

РЕЖИМ — представлен цифрой от 0 до 3:

0 — Все незапрашиваемые результирующие коды буферизируются, но не отображаются.

1 — Незапрашиваемые результирующие коды отображаются только в режиме offline (командный режим), без буферизации.

2 — Незапрашиваемые результирующие коды буферизируются и отображаются. При работе в режиме online (передача данных), коды не отображаются, но будут отображены после выхода из режима online в режим offline (командный режим).

3 — Незапрашиваемые результирующие коды не буферизируются и отображаются вне зависимости от режима работы (online/offline).

СООБЩЕНИЯ — параметр представлен цифрой от 0 до 3, он управляет индикацией входящих SMS сообщений:

0 — Незапрашиваемые результирующие коды индикации получения новых входящих SMS сообщений не возвращаются.

1 — Если входящее SMS сообщение сохранено в память, то возвращается незапрашиваемый результирующий код +CMTI с указанием индекса (номера ячейки памяти в которую сохранено сообщение).

2 — Входящие SMS сообщения выводятся как незапрашиваемый результирующий код +CMT с указанием номера отправителя, текста сообщения или блока PDU.

3 — В настоящее время модули A6 не поддерживают данное значение.

Значение по умолчанию 0.

AT+ENPWRSAVE=ЧИСЛО — Включение/отключение спящего режима.

ЧИСЛО — Цифра 0 или 1.

0 — Выключить спящий режим.

1 — Включить спящий режим.