

Использование Arduino для создания комплектующих «Умного дома»

Автор работы: Денисевич Гордей Владимирович, 10 К класс  
Государственное учреждение образования «Средняя школа №7 г. Новогрудка»  
Секция: информатика  
Руководитель работы: Подлипский Андрей Викторович, учитель информатики  
Государственное учреждение образования «Средняя школа №7 г. Новогрудка»

## ОГЛАВЛЕНИЕ

|                                                                                                |   |
|------------------------------------------------------------------------------------------------|---|
| ВВЕДЕНИЕ                                                                                       | 3 |
| ОСНОВНАЯ ЧАСТЬ                                                                                 | 4 |
| ГЛАВА 1. АППАРАТНО-ПРОГРАММНАЯ СРЕДА ARDUINO                                                   | 4 |
| 1.1 Общая характеристика Arduino                                                               | 4 |
| 1.2 Аппаратная часть Arduino                                                                   | 4 |
| 1.3 Программная часть Arduino                                                                  | 4 |
| ГЛАВА 2. КОМПЛЕКТУЮЩИЕ УМНОГО ДОМА                                                             | 5 |
| 2.1 Понятие умного дома и его техническое оснащение                                            | 5 |
| 2.2 Механизмы, созданные с помощью Arduino, которые можно использовать в оснащении умного дома | 5 |
| 2.2.1 Пожарный извещатель                                                                      | 5 |
| 2.2.2 Умные жалюзи                                                                             | 6 |
| 2.2.3 Умный светильник                                                                         | 7 |
| ЗАКЛЮЧЕНИЕ                                                                                     | 8 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ                                                               | 9 |

## ВВЕДЕНИЕ

В связи с большим развитием технологической сферы появляются всё новые и новые механизмы и программы, с помощью которых можно создавать простые схемы и целых роботов. Поэтому важно знать не только начальный курс информатики, но и сторонние программы и языки программирования. Познакомившись с некоторыми программами (Scratch 2, Lego Wedo), я пришёл к выводу, что наиболее удобной и доступной для меня является Arduino.

Изучением языка программирования Arduino я занялся около 1 года назад, так как я понимал, что в этой сфере я буду работать в будущем, когда стану самостоятельным и войду во взрослую жизнь. На сегодняшний день я изучил несколько десятков механизмов и их полную работу. Эти занятия способствуют развитию не только специальных навыков, но и помогают в изучении других предметов, таких как физика, математика. В представленном исследовании по информатике исследуется аппаратно-вычислительная платформа Arduino, и её возможности по созданию «Умного дома».

Объект: наборы Arduino.

Предметом нашего исследования являются механически созданные с помощью аппаратно-программных средств Arduino механизмы, которые можно использовать при создании «Умного дома».

Гипотеза исследования: с помощью аппаратно-программных средств Arduino создать комплектующие для «Умного дома».

Целью исследовательской работы является изучение возможности использования аппаратно-программных средств Arduino для создания элементов системы «Умный дом».

Задачи работы:

- изучить особенности аппаратно-программного обеспечения Arduino;
- определить отличительные особенности комплексной системы «Умный дом»;
- сконструировать механизмы, которыми можно укомплектовать «Умный дом»;
- создать демонстрационные макеты.

Методы исследования: теоретический анализ источников, обобщение, моделирование, программирование, эксперимент, описание.

Практическое применение: материал данного исследования может применяться на факультативных занятиях по робототехнике.

## ОСНОВНАЯ ЧАСТЬ

### ГЛАВА 1. АППАРАТНО-ПРОГРАММНАЯ СРЕДА ARDUINO

#### 1.1 Общая характеристика Arduino

Arduino – это аппаратно-программное средство, для построения простых систем автоматики и робототехники, ориентированная на непрофессиональных пользователей. Это мини-компьютер, имеющий свой собственный процессор, некоторое количество памяти и набор входов-выходов для управления внешними устройствами и получения данных с этих устройств.

Преимущества Arduino по сравнению с другими сферами: не нужен программатор, не нужны особые знания в программировании. Arduino – это целый конструктор для создания и управления электронными устройствами.

Arduino включает: аппаратную часть (Arduino и набор электронных компонент) и программную часть (среда программирования Arduino).[4]

#### 1.2 Аппаратная часть Arduino

Набор Arduino комплектуется платой Arduino Uno и достаточно большим количеством электронных датчиков и компонент. (Приложение 1, рис.1)

Главной частью программирования на Arduino является материнская плата Uno. По периметру платы Arduino размещены электрические выводы. Есть выводы цифровые, а есть аналоговые.

Arduino позволяет подключать к нему огромное количество различных периферийных устройств, например: кнопки, светодиоды, микрофоны и динамики, электродвигатели и сервоприводы, ЖК-дисплеи, считыватели SD карт. А также десятки различных датчиков: освещённости, датчики дыма и состава воздуха, температуры и влажности и многое другое. Всё это превращает Arduino в универсальное ядро системы, которое может быть сконфигурировано совершенно разнообразными способами. [1]

#### 1.3 Программная часть Arduino

Существует язык для общения с микроконтроллером, упрощённый и адаптированный специально для Arduino. (Приложение 1, рис.2)

Написание сообщений для Arduino называется программирование. И для упрощения этого процесса разработана специальная программная среда -Arduino IDE.[1]

## ГЛАВА 2. КОМПЛЕКТУЮЩИЕ УМНОГО ДОМА

### 2.1 Понятие умного дома и его техническое оснащение

Комплексная система умный дом – это полная автоматизация управления устройствами во всех помещениях дома квартиры или офиса. Она позволяет управлять как отдельными блоками, так и всем в целом. Различные аудио-, свето- и видео- сценарии заранее прописываются в программе, благодаря которой работает контроллер, то есть мозг Умного дома. Это существенно экономит время хозяев жилища. Также исключается возможность возникновения аварийных ситуаций. Умный дом самостоятельно отключает электроприборы, переводит их в спящий режим при отсутствии людей.[2]

В наше время любому человеку хочется жить в комфорте и безопасности. Для этого создаются различные комплектующие для дома, которые помогают облегчить жизнь, сэкономить время, защитить от различных бед, например: автоматическое освещение, климат-контроль, управление электроприводами, сбор информации, управление техникой в доме, контроль и оповещение об аварийных ситуациях.

### 2.2 Механизмы, созданные с помощью Arduino, которые можно использовать в оснащении умного дома

#### 2.2.1 Пожарный извещатель

В наше время, у каждого дома есть пожарный извещатель. Он может не только предупредить о возникновении возгорания, но дать сигнал к его автоматическому тушению. В качестве чувствительных элементов в таких модулях используются следующие датчики: тепловые датчики, дымовые датчики, датчики пламени.

В современных пожарных модулях могут быть использованы несколько видов датчиков сразу. Я работал с датчиком пламени и создал настоящий пожарный извещатель, который будет посылать сигнал о возникновении открытого пламени в зоне его видимости.

Для сборки устройства на макетной плате понадобятся следующие компоненты: Arduino Uno; USB-кабель для подключения к компьютеру; безопасная макетная плата; 1 датчик пламени; 1 светодиод любого цвета; 1 пьезодинамик; 1 резистор, номиналом 10 кОм; 1 резистор, номиналом 220 Ом; 7 соединительных проводов разных цветов.

Последовательность подключения: (Приложение 1, рис.3)

1. Длинная ножка датчика подключается к земле через резистор в 10 кОм, по схеме простейшего делителя напряжения. С этой же ноги и снимается сигнал, который получаем на пине A0;
2. Питание подключается ко второй – короткой ножке. В качестве питания можно использовать как пин на 5 В, так и пин на 3,3 В;
3. Светодиод и пьезодинамик подключены стандартным образом. К каким конкретно пинам их подключать – не важно. У меня светодиод подключён к 9-му цифровому пину Arduino, а пьезодинамик – к 8-му;

Для того чтобы увидеть разницу между фоновым сигналом и сигналом, который датчик выдаст, в присутствии открытого огня, задаётся порог чувствительности. Нужно посмотреть на фоновые значения датчика, для этого посылается значение сигналов датчика в последовательный порт компьютера, а далее их видно на экране в мониторе последовательного порта.

Для передачи сигнала в последовательный порт компьютера, используется блок [в последовательный порт], который находится в разделе [Коммуникации]. (Приложение 1, рис.4) Передаю в порт значение с пина A0, к которому подключен датчик пламени, поэтому часть блока с надписью [message] не понадобится, на её место ставлю блок [аналоговый порт], с номером нашего пина – A0. (Приложение 1, рис.5) Для того чтобы склеить блоки, можно использовать блок [клей] нужной формы, находящийся в разделе [Коммуникации]. (Приложение 1, рис.6) Добавляю полученную конструкцию к блоку [Программа] и ставлю задержку в 1000 мс, чтобы значения датчика посылались в порт 1 раз в секунду. (Приложение 1, рис.7)

Алгоритм готов, подключаю собранное устройство к Arduino и загружаю в него программу. Теперь смотрю на полученные данные нажатием кнопки [Монитор последовательного порта]. (Приложение 1, рис.8) Откроется окно, в которое будут приходить данные с последовательного порта, к которому подключена ваша Arduino. Для того, чтобы порт работал корректно, значение скорости передачи данных должно стоять на [9600 бод]. Если скорость передачи указана правильно, то 1 раз в секунду получаю значения сигнала с порта A0 платы Arduino. (Приложение 1, рис.9) Значения, которые посылает датчик в данный момент, являются фоновыми. А теперь, нужно проверить – как датчик реагирует на открытый огонь. Можно увидеть, как изменяются данные датчика при зажигании спички в непосредственной близости от него. Зажжённая спичка находится на расстоянии в 30 см, 60 см, 1 м и 1,5 м, 2 м. (Приложение 1, рис.10) В момент зажигания спички, значения, посылаемые датчиком, многократно возрастали. Для расстояния в 30 см, получил значение 834, для 60 см – 394, для 1 метра – 220, для 1,5 метров – 148, для 2 метров – 81.

Определил порог срабатывания датчика. Фоновые значения в “спокойном” состоянии не превышали 10-ти, для надёжности увеличиваю наш порог чувствительности до 20-ти. Таким образом, при аналоговом сигнале с датчика со значением 20 и выше, Arduino будет давать команду на включение сигнализации. Для проверки условия достижения порога, использую блок [если]. (Приложение 1, рис.11)

Условие будет выполняться, если значения с датчика, подключённого к пину A0, будут больше или равны 20. Теперь составляю алгоритм для сигнализации. Нужно включать и выключать светодиод и издавать звуковой сигнал пьезодинамиком. (Приложение 1, рис.12)

Программа готова, загружаю её в Arduino и проверяю её работу. Теперь, при обнаружении открытого пламени на расстоянии до 2-х метров, сигнализация должна мигать светодиодом и пищать с задержками в 0,2 секунды.

### 2.2.2 Умные жалюзи

Человеку, работающему в помещении, необходим свет. Для того чтобы избавиться от слепящего солнечного света используют жалюзи. Но солнце не стоит на месте, следовательно, жалюзи должны открываться и закрываться, в зависимости от интенсивности солнечного света, падающего на них.

Для сборки устройства на макетной плате понадобятся следующие компоненты: Arduino Uno; USB-кабель для подключения к компьютеру; беспаячная макетная плата; 1 сервопривод; 1 потенциометр; 1 фоторезистор; 1 резистор, номиналом 10 кОм; 9 соединительных проводов разных цветов.

Последовательность подключения: (Приложение 1, рис.13)

1. Сервопривод имеет 3 пина для подключения, вынесенные на длинных проводах. Самый тёмный провод (коричневый) обозначает пин, подключаемый к земле (-), красный провод – пин питания (+5 В), а жёлтый или оранжевый провод является сигнальным. Подключить сигнальный провод сервопривода можно к любому цифровому пину;

2. Фоторезистор подключён с пина, идущего на землю через резистор номиналом 10 кОм, одновременно буду снимать сигнал, а вторую ногу фоторезистора подключаю к питанию (+5 В);

3. Сигнальные провода, идущие от фоторезистора и потенциометра подключаются к цифровым пинам Arduino (пины A0 и A1).

Для начала, нужно повернуть сервопривод на какой-либо угол. Это делается с помощью блока [Серво], который находится в разделе [Общее оборудование]. (Приложение 1, рис.14) Чтобы повернуть сервопривод последовательно на угол: 0°, 90° и 180°, а затем снова вернуть в положение 0°, нужно задать соответствующие углы. (Приложение 1, рис.15) Задержка после каждого из блоков поворота нужна для того, чтобы вал сервопривода успевал поворачиваться на нужный угол. Время необходимое сервоприводу, для того, чтобы повернуть свой вал на 180°, составляет около 1 секунды. Если поворачивать сервопривод на 180° из положения 0° и поставить при этом недостаточное время задержки, то сервопривод просто не успеет сделать полный оборот.

Чтобы управлять сервоприводом в ручную нужен потенциометр. Для этого использую алгоритм, который будет поворачивать вал сервопривода вслед за вращением ручки потенциометра в одну и другую сторону. Задать угол сервопривода с помощью блока [Карта], т.к. необходимо преобразовать диапазон [0 ... 1023] аналогового сигнала потенциометра, в диапазон [0 ... 180] угла поворота сервопривода. (Приложение 1, рис.16)

Чтобы настроить рукоятку «умных» жалюзи, необходимо экспериментально найти углы поворота. Эти углы могут варьироваться от 90° до 180°.

В ходе того, как вращается сервопривод, и, наблюдая за закрытием жалюзи, нужно ещё и знать – на какой угол повернут сервопривод на данный момент, для этого, угол поворота, который задаёт сервоприводу блок [Карта], продублируется в последовательный порт компьютера. (Приложение 1, рис.17)

Вращая потенциометр, добиться полного закрытия наших жалюзи. Когда это произошло, можно открыть [Монитор последовательного порта] и посмотреть на значение угла, в котором на данный момент находится вал сервопривода. Например, следующие значения. (Приложение 1, рис.18)

Полностью открытые жалюзи соответствуют углу поворота сервопривода на 90°, а полностью закрытые углу 170°. Осталось определить значения сигнала с фоторезистора, при котором жалюзи будут полностью открыты, а при котором – полностью закрыты. Закрывать жалюзи по мере роста значения сигнала на фоторезисторе, т.е. когда солнце начинает светить ярче, и наоборот: открывать их по мере того, как солнце будет уходить, и яркость внешнего освещения будет падать. Получаю данные с фоторезистора в последовательный порт компьютера. (Приложение 1, рис.19)

Загрузив программу в Arduino, запускаю [Монитор последовательного порта] и направляю фоторезистор на солнце или яркую лампу. В мониторе порта видны “светлые” значения сигнала с фоторезистора, которые соответствуют яркому солнцу. (Приложение 1, рис.20)

Округляю полученные значения до [800]. Теперь нужно получить “тёмные” значения, для этого прикрываю фоторезистор рукой и наблюдаю за тем – как изменились значения в мониторе порта. (Приложение 1, рис.21)

Эти значения можно округлить до [300]. Всё что осталось, автоматически поворачивать сервопривод от 90° до 170°, при изменении значений сигнала на фоторезисторе от [300] до [800]. (Приложение 1, рис.22) В пасмурные дни или ночью, значения сигнала с фоторезистора могут опускаться гораздо ниже [300], а в солнечные дни подниматься гораздо выше [800], при этом сервопривод будет пытаться поворачивать свой вал, получая команды углов меньших 90° и больших 170°, поэтому необходимо жёстко задать крайние значения в 90° и 170°. Для того, чтобы ограничить диапазон значений, использую блок [Constrain], который находится в

разделе [математические операторы], и ограничиваю с его помощью диапазон значений углов поворота сервопривода. (Приложение 1, рис.23)

### 2.2.3 Умный светильник

При наличии необходимого программного обеспечения, с помощью звуков, можно не только управлять своим собственным компьютером, но и подключёнными к нему устройствами. Я попробую управлять “мини-компьютером” Arduino и подключёнными к нему устройствами с помощью звука. Для этого понадобится специальный датчик – датчик звука. Сделаю модель умного светильника и попробую зажечь и погасить светодиод по хлопку наших рук.

Для сборки устройства на макетной плате понадобятся следующие компоненты: (Приложение 1, рис.24) Arduino Uno; USB-кабель для подключения к компьютеру; безопасная макетная плата; 1 светодиод любого цвета; 1 резистор, номиналом 220 Ом; 1 датчик звука; 5 соединительных проводов разных цветов.

Для начала нужно настроить порог чувствительности таким образом, чтобы микрофон реагировал лишь на громкие звуки и не обращал внимания на фоновый постоянный шум.

Подключаю собранную схему у Arduino, а Arduino подключаю к USB-порту компьютера. На датчике звука должны зажечься один или два индикатора – тот, что справа сигнализирует о работе датчика, а тот, что слева (он может быть потушен) является индикатором шума и загорается лишь тогда, когда громкость внешнего звука достигла порога чувствительности микрофона. Над этим индикатором располагается подстроенный резистор, с помощью которого будут задаваться значения этого самого порога. Для этого нужна маленькую плоскую отвёртку и выполняются следующие инструкции:

1. Если индикатор шума потушен, аккуратно вращать регулировочный винт подстроенного резистора по часовой стрелке до тех пор, пока индикатор не загорится. Как только индикатор загорится – отвернуть винт немного назад, чтобы он снова потух;

2. Если индикатор шума зажжён, вращать регулировочный винт подстроенного резистора против часовой стрелки до тех пор, пока индикатор не потухнет. Вращать винт следует очень аккуратно и как только индикатор погаснет – прекратить регулировку.

Сделав правильно, индикатор шума будет находиться в потушенном состоянии и вспыхивать лишь при громких звуках. Хлопнув в ладоши рядом с датчиком звука, индикатор шума вспыхнет, если всё так, то настройка проведена правильно. В момент вспыхивания, датчик посылает высокий (HIGH) сигнал на свой цифровой пин. Этот сигнал будет принимать на 2-м пине Arduino. Таким образом, получился как-бы аналог обычной кнопки, которая работает не по нажатию, а от громких звуков.

Алгоритм программы: если на 2-м пине Arduino появляется высокий (HIGH) сигнал, а светодиод подсоединённый к 9-му пину выключен, то включаю его (HIGH). Если же светодиод уже включён, то выключаю (LOW). В разделе [Порты] как раз есть блок, который может не просто включать и выключать пин, посылая на него высокий или низкий сигнал, но проверять состояние пина в данный момент и если на пин уже подан высокий сигнал (HIGH), переключать его в низкий (LOW) и наоборот, если на пине – низкий сигнал (LOW), переключать его в высокий (HIGH). Этот блок является своеобразным цифровым переключателем и называется [Переключить цифровой выход]. (Приложение 1, рис.25) Загружаю программу в Arduino и проверяю работу светодиода. Тем не менее программа ещё не готова, так как светодиод не всегда корректно реагирует на хлопки. Он может включаться и выключаться, а может просто мигать. Всё дело в том, что звук хлопка не затихает сразу и датчик звука успевает считывать не один, а несколько цифровых сигналов и может включить и сразу выключить светодиод. Для того, чтобы избавиться от такого мигания, ставится задержка в 200мс после команды переключения пина. Это даст звуку время затихнуть, чтобы датчик не считал сигнал дважды. (Приложение 1, рис.26)

## ЗАКЛЮЧЕНИЕ

В результате исследования я многое узнал о программно-аппаратной платформе Arduino, о сферах её применения и пришёл к следующим выводам:

1. Arduino – это аппаратно-программное средство, для построения простых систем автоматики и робототехники, ориентированная на непрофессиональных пользователей.
2. Аппаратно-программное средство Arduino состоит из программной и аппаратной части.
3. Arduino позволяет подключать к нему огромное количество различных периферийных устройств, поэтому с помощью этого аппаратно-программного средства можно проектировать механизмы, например: индикатор SOS, светофор, фонарик, кодовые замки, термометры, часы.
4. Комплексная система «Умный дом» – это полная автоматизация управления устройствами во всех помещениях дома квартиры или офиса, созданная для комфорта и безопасности человека.
5. Arduino позволяет сконструировать и использовать в быту механизмы: умные жалюзи, пожарный извещатель, умный светильник, аналоги которых используются в комплектующих умного дома.

Таким образом, наша гипотеза подтвердилась: с помощью аппаратно-программных средств Arduino можно создать комплектующие «Умного дома».

Поскольку Arduino – это достаточно простое и интересное средство для реализации своих идей, в дальнейшем я планирую реализовать свои идеи в быту, расширить мои знания и заняться программированием Arduino на языке C++.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Графический язык программирования ArduBloch для Arduino [Электронный ресурс] – Режим доступа: <https://arduinomaster.ru/program/visual-ardublock-arduino/> – Дата доступа: 12.01.2019.
2. Умный дом. Википедия [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BD%D1%8B%D0%B9\\_%D0%B4%D0%BE%D0%BC](https://ru.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B4%D0%BE%D0%BC) – Дата доступа: 06.02.2019.
3. Уроки и проекты Arduino [Электронный ресурс] – Режим доступа: <https://arduino-kit.ru/blogs/blog/uroki-i-proekty> – Дата доступа: 12.01.2019.
4. Что такое Arduino? [Электронный ресурс] – Режим доступа: <http://arduino.ru/about> – Дата доступа: 20.01.2019.

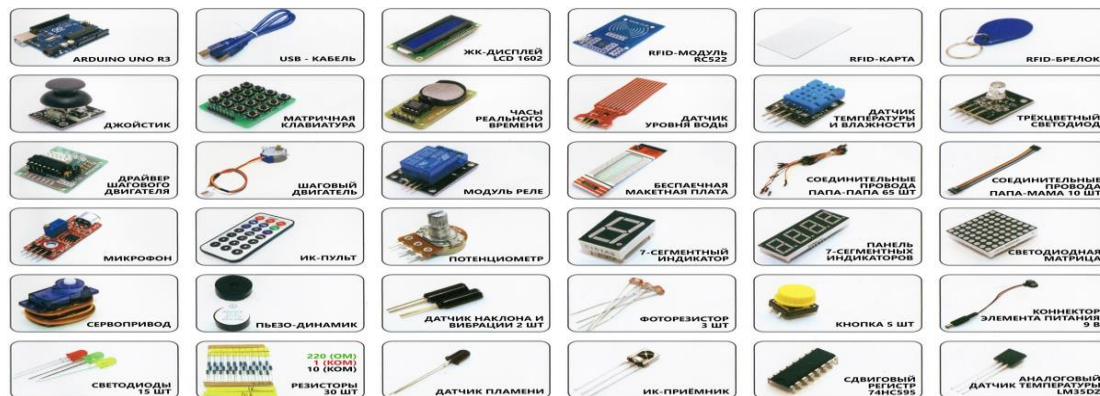


Рис.1 – Набор Arduino

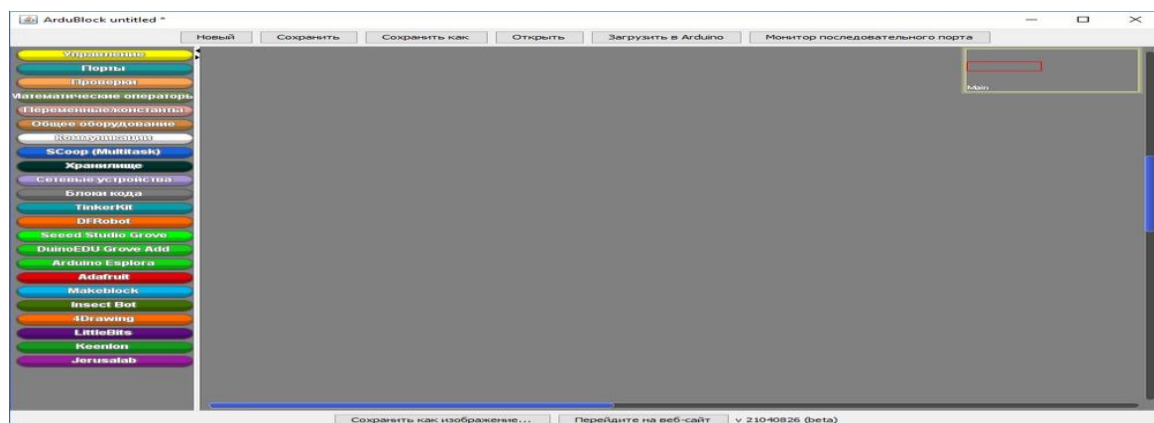


Рис.2 – Программа ArduBlock

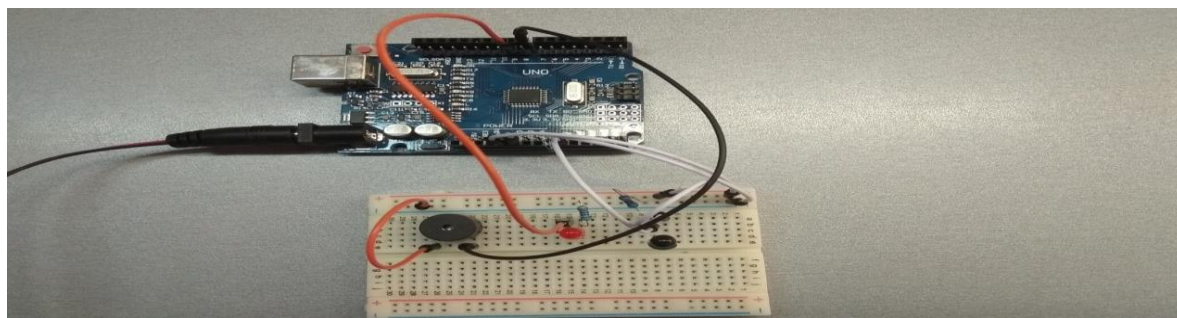


Рис.3 – Модель пожарного извещателя



Рис.4 – Блок передачи сигнала в последовательный порт



Рис.5 – Блок передачи сигнала в последовательный порт с пином A0



Рис.6 – Использование блока [клей]



Рис.7 – Программа для вывода на монитор последовательного порта



Рис.8 – Запуск монитора последовательного порта

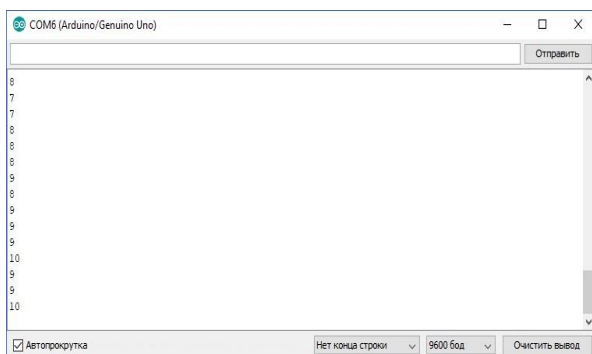


Рис.9 – Значения сигнала с порта A0

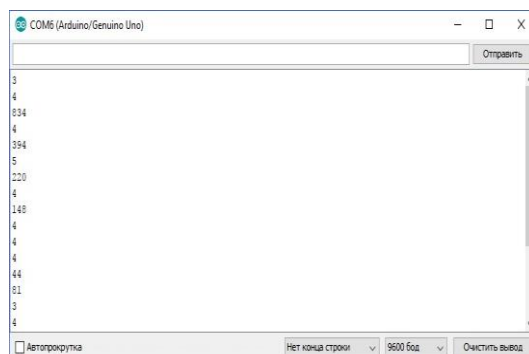


Рис.10 – Значения сигнала с порта A0 при зажжённой спичке



Рис.11 – Использование блока [если]



Рис.12 – Алгоритм для сигнализации

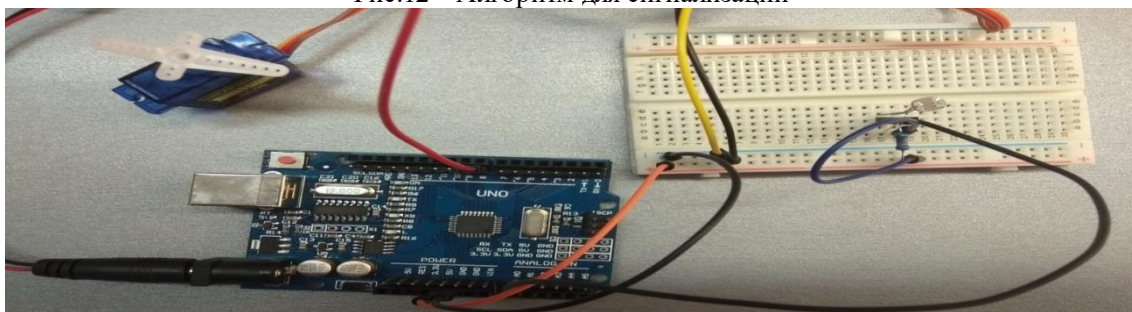


Рис.13 – Модель умных жалюзи



Рис.14 – Использование блока [Серво]

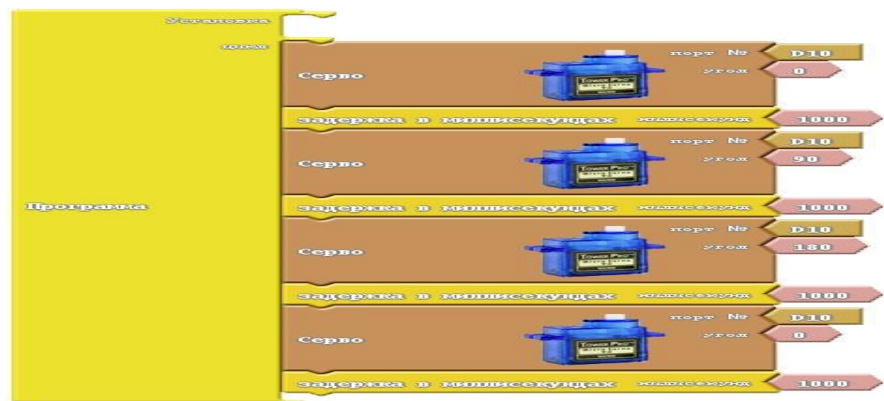


Рис.15 – Поворот сервопривода на различные углы



Рис.16 – Поворот вала сервопривода за ручкой потенциометра

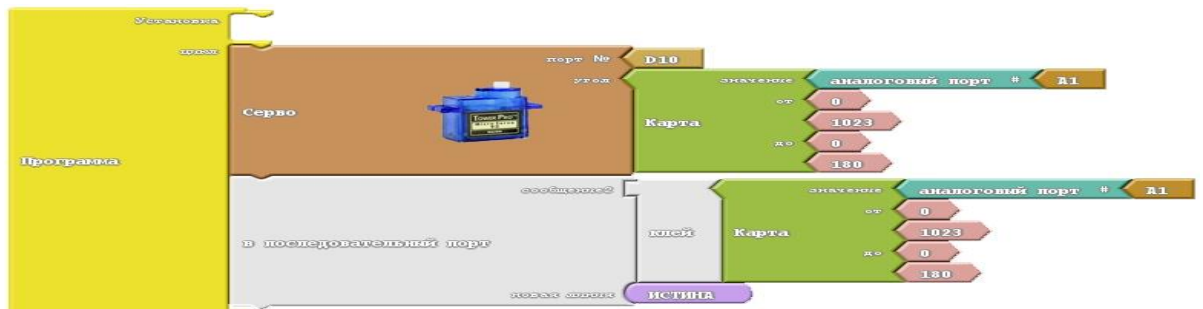


Рис.17 – Подключение последовательного порта для определения углов



Рис.18 – Значения угла сервопривода



Рис.19 – Получение данных с фоторезистора

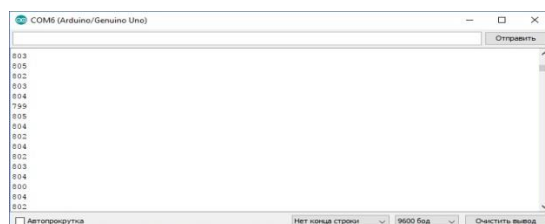


Рис.20 – Значения сигнала с

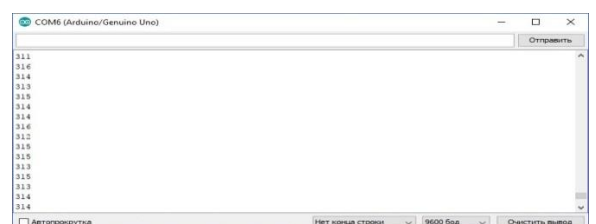


Рис.21 – Значения сигнала с

фоторезистора при включённой  
лампе



Рис.22 – Программа для поворота  
поворота умных жалюзи с  
постоянными значениями

фоторезистора при выключенной  
лампе

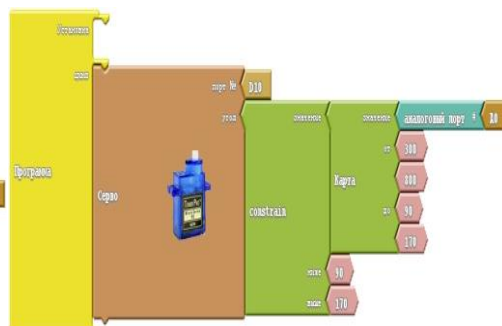


Рис.23 – Программа для  
умных жалюзи

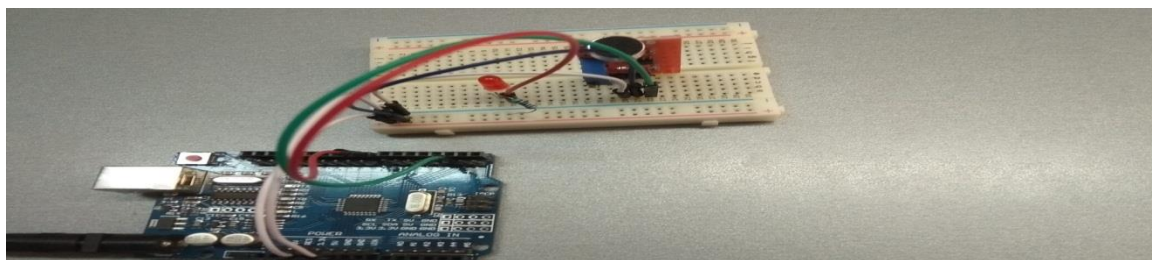


Рис.24 – Модель умного светильника



Рис.25 – Работа с цифровым переключателем

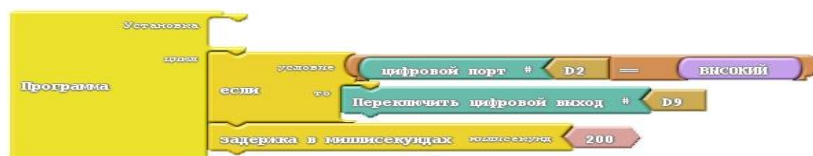


Рис.26 – Программа для работы умного светильника