

**Автор:**

**Волков Егор Олегович**

ученик 10 класса

МБОУ «СОШ № 24»

г. Череповец, Вологодская область

**Руководитель:**

**Ситников Павел Леонидович**

учитель физики, информатики и ИКТ

г. Череповец, Вологодская область

## **Создание метеостанции и контролера светодиодной ленты на основе микрокомпьютера Arduino**

В наше время стремительно развиваются системы умного дома, а также устройства, которые управляются не только с помощью пульта, а через другие гаджеты, такие как компьютеры, смартфоны и планшеты. Мы выбрали данную тему и решили создать подобное устройство, потому что на рынке практически отсутствуют подобные устройства с такой функциональностью за разумную плату. Поэтому нами было решено создать собственное устройство с необходимым уровнем функций. Создание нашего устройства несет практическое значение, так как был создан практически готовый продукт. Наша работа выполнена в соответствии с современными тенденциями, и поэтому актуальна в наше время.

Целью нашей работы стало создание устройства с декоративной подсветкой зеркала с эффектом туннеля, а также с измерением температуры, влажности и давления в помещении с удаленным управлением через компьютер. Для выполнения поставленной цели перед нами стояли следующие задачи: в первую очередь необходимо было познакомиться с аппаратно-вычислительной платформой Arduino и ее периферии, далее надо было рассмотреть средства разработки, как для создания самой прошивки для микрокомпьютера, так и для создания клиентских приложений, соответственно рассматривались и языки программирования для создания программного обеспечения. Перед нами стояла задача создать многофункциональное устройство с RGB-контроллером и комнатной метеостанцией с помощью набора сенсоров и прочих устройств.

### *Аппаратная платформа Arduino*

Мы с помощью различных, включая иностранных, источников рассмотрели аппаратную платформу Arduino. Arduino — аппаратная вычислительная платформа, основными компонентами которой являются простая плата ввода-вывода [1]. Существует несколько версий плат. Они отличаются размерами, количеством доступной памяти, количеством разъемов и, соответственно, ценой. Для своего устройства мы выбрали микрокомпьютер Arduino Uno, который имеет оптимальное соотношение цены функций для нашего проекта. Сердцем этой платы является 8-битовый процессор ATmega328 с частотой 16 МГц. Всего на плате 14 цифровых и 6 аналоговых разъемов для

подключения периферии. Доступная память для программ составляет 32 КБ, а также имеется 2 КБ оперативной памяти и килобайт для энергонезависимого хранения данных [2]. Удобство платы в том, что в ее нет необходимости заливать загрузчик программатором, инициализирующий оборудование. Порт USB типа В используется для питания устройства, заливки прошивки, а также для связи с компьютером.

Для хранения данных с датчиков и настроек мы установили плату расширения для SD карты «SeeedStudio SD Card Shield v 1.0», который использует четыре цифровых контактов Arduino с 10-13. К данной плате можно подключить так полноразмерную SD карту, так и microSD карту [3]. Для своего проекта мы использовали SD карту объемом 1 гигабайт.

Для создания устройства мы воспользовались двумя четырёхразрядными дисплеями «GROVE 4-Digit Display», которые использовались для вывода информации с устройства. Данный экран имеет возможность изменять яркость сегментов [4].

Мы воспользовались «Grove Led Strip Driver» для соединения светодиодной ленты с блоком питания к плате, он позволяет регулировать цвет для каждого ее канала. Для питания светодиодной ленты мы использовали блок питания на 20 ватт, что достаточно для метровой полноцветной светодиодной ленты, наклеенной по периметру углубления зеркала.

Для определения давления и высоты использовался модуль барометра на базе высокоточного сенсора BMP085 [5], выходные штыри которого были пропаяны на GROVE-разъём, а также сенсор температуры и влажности на базе DHT11 [6]. Для удобного соединения данных устройств мы воспользовались «GROVE Base Shield» [7]. Используя, официальную документацию и библиотеки мы подключили все эти устройства к основной плате.

#### *Среда разработки и языки программирования*

Для разработки мы воспользовались средой Visual Studio от компании Microsoft [8], к которой был установлено расширение Visualmicro [9], которое связывало ее со средой разработки Arduino. Для написания прошивки используется язык программирования C++ с урезанной стандартной библиотекой [11]. Данный язык имеет широкое применение и является основным языком программирования от простых программ до операционных систем. У данного языка широкие возможности, но из-за этого он имеет сложный синтаксис для их реализации [12]. Для некоторых низкоуровневых операций использовался язык машинных команд – ассемблер. Для написания клиентского приложения для Windows использовался язык программирования C#, имеющий похожий на C++ синтаксис, то есть набор правил написания программ. Программа использует возможности .NET Framework [13]. Для приложений мы воспользовались мощной системой Windows Presentation Foundation (WPF), использующий язык XAML для разметки, основанный на XML – удобном формате для представления иерархических структур [14].

#### *Написание прошивки*

По умолчанию Arduino не поддерживает многопоточность, а нам она потребовалась для реализации проекта. Поэтому нами была добавлена

реализация вытесняющей многопоточности, когда по прерыванию сохраняется или загружается контекст потока. Ее реализация потребовала изменения основного файла прошивки `main.cpp`, который скрыт от программиста, а ему доступен только скетч, который не является файлом программы, а потом объединяется с `main.cpp`. Реализация данной функции потребовало ассемблеровских вставок, так как ввелась низкоуровневая работа с оборудованием. Создана структура, которая содержит информацию об активном потоке, а также о флагах выполнения, с помощью которых можно приостанавливать, возобновлять или выходить из потока.

Нами также созданы классы для даты и времени. Он позволяет инициализировать дату и время с точностью до микросекунды, за это отвечает конструктор класса `DateTime`, добавлять к данной дате другую дату, а также вычитать из нее, вызвав определенные операторы сложения и вычитания, изменять дату на введенное значение, вызвав метод `Change`, а также выводить дату и время в строку, как для CSV файла, так и для общепринятого формата. Особенность нашей реализации в том, что в любом методе класса мы можем указывать значения, выходящие за пределы допустимых, например, при установке секунд выбрать 62, то в секундах будет 2, а в минуты на одну больше. Наибольшие трудности возникли для реализации подобного функционала для месяцев, так как они имеют различное количество дней в году, эти данные хранятся в неизменяемом массиве.

Для представления цвета мы также создали собственный класс `Color`. По основным принципам он построен аналогично классу даты и времени, также существует преобразование в строку, сложение и вычитание.

Мы отредактировали библиотеку для дисплея, чтобы можно было выводить текст на него, а не только цифры, как в оригинальной библиотеке. Надо понимать, что сам дисплей не знает ничего о символах для отображения, он лишь может включать сегменты с помощью команд. Эти команды мы получили для 255 символов кодировки `Win1251` с помощью написанной нами программы, где можно визуально редактировать «шрифт» сегментного дисплея. Нами реализована функция разделения текста на подстроки для обработки строковых данных. Сами данные хранятся на карте памяти в формате CSV для каждого состояния, временный файл показаний сенсоров и файл текущих настроек.

### *Написание клиентского приложения для компьютера*

Как ранее было сказано, для отображения информации на дисплей использовался шрифт, который генерировался через написанную нами вспомогательную программу. Она позволяет визуально задавать шрифт дисплея. Данные на компьютер передаются через USB-порт `Arduino`, их принимает и отправляет со стороны компьютера компактная служба, которая записывает данные в базу данных `SQLite`, к которой подключается программа с графическим интерфейсом на основе технологии `WPF`. Это программное обеспечение позволяет наглядно представлять показания датчиков с помощью графиков и индикатора, который был создан нами. Для этого нами созданы пользовательские элементы управления. Для датчика мы предварительно его

нарисовали в векторном редакторе Expression Design, которое преобразовалось в файл разметки. Данные пользовательские элементы выполнены со всеми требованиями платформы по оформлению кода пользовательского элемента. Одинаковые для каждого показателя колонки вынесены, также в отдельный элемент управления для большей читаемости программы и уменьшения дубликатов. Для управления светодиодной ленты мы написали элемент управления выбора цвета, путем перетаскивания ползунков по цветовому полю. В программе можно создавать собственные предустановки, как для простого свечения цвета, так и для более сложного алгоритма подсветки. Полученные данные можно обрабатывать, экспортировать в файл CSV, который обрабатывают электронные таблицы.

**Выводы.** С помощью нашей метеостанции и контроллера светодиодной ленты на основе микроконтроллера Arduino мы создали действительно интересное решение, которое удовлетворяет современным требованиям к подобным устройствам. Станция является законченным, готовым к применению продуктом. Созданные нами комнатная метеостанция и регулируемое светодиодное освещение с эффектом туннеля, позволяют измерять температуру, влажность и давление в помещении, а также удаленно управлять устройствами через компьютер.

#### **Список литературы**

1. Andrew Troelsen Pro C# 5.0 and the .NET 4.5 Framework. - 6-е изд. Apress, 2012. – 1560 с.: ил.
2. Arduino - Compare // Arduino URL: <http://arduino.cc/en/Products.Compare> (дата обращения: 18.11.2013).
3. Arduino — Википедия // Википедия — свободная энциклопедия URL: <http://ru.wikipedia.org/wiki/Arduino> (дата обращения: 29.12.2014).
4. Arduino Ide for Visual Studio URL: <http://www.visualmicro.com> (дата обращения: 08.01.2013).
5. Brian Evans Beginning Arduino Programming. - Apress, 2011. - 272 с.: ил.
6. Grove 4-Digit Display Wiki // SeeedStudio URL: [http://www.seeedstudio.com/wiki/Grove\\_-\\_4-Digit\\_Display](http://www.seeedstudio.com/wiki/Grove_-_4-Digit_Display) (дата обращения: 26.12.2013).
7. Grove-Base Shield V1.3 Wiki // SeeedStudio URL: [http://www.seeedstudio.com/wiki/Grove\\_-\\_Base\\_Shield\\_V1.3](http://www.seeedstudio.com/wiki/Grove_-_Base_Shield_V1.3) (дата обращения: 26.12.2013).
8. Grove-Temperature and Humidity Sensor Wiki // SeeedStudio URL: [http://www.seeedstudio.com/wiki/Grove\\_-\\_Temperature\\_and\\_Humidity\\_Sensor](http://www.seeedstudio.com/wiki/Grove_-_Temperature_and_Humidity_Sensor) (дата обращения: 26.12.2013).
9. Matthew MacDonald Pro WPF 4.5 in C#. - 4-е изд. Apress, 2012. - 1112 с.: ил.
10. SD Card Shield V3.0 Wiki // SeeedStudio URL: [http://www.seeedstudio.com/wiki/SD\\_Card\\_Shield\\_V3.0](http://www.seeedstudio.com/wiki/SD_Card_Shield_V3.0) (дата обращения: 26.12.2013).
11. SparkFun Electronics URL: <https://www.sparkfun.com/datasheets/Components/General/BST-BMP085-DS000-05.pdf> (дата обращения: 04.01.2013).
12. Visual Studio - Home // Visual Studio - Home URL: <http://www.visualstudio.com> (дата обращения: 08.01.2013)
13. Платт Ч. Электроника для начинающих/ Пер. с англ. - Спб.: БХВ-Петербург. 2012. - 480 с.: ил.
14. Соммер У. Программирование компьютерных плат Arduino/Freduino. - Спб.: БХВ-Петербург. 2012. - 256 с.: ил.