

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

Кафедра информационных систем и технологий

В.В. Извозчикова

ИСПОЛЬЗОВАНИЕ МИКРОКОНТРОЛЛЕРА ARDUINO В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Рекомендовано к изданию Редакционно-издательским советом федерального государственного бюджетного образовательного Учреждения высшего профессионального образования «Оренбургский Государственный университет» в качестве методических указаний для студентов, обучающихся по программам высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии

Оренбург
2015

УДК 004.3/.4'23(075.8)

ББК 32.973-018.я73

ИЗ4

Рецензент – доцент, кандидат технических наук, А.С. Боровский

- ИЗ4 **Извозчикова, В.В.**
Использование микроконтроллера Arduino в информационных системах: методические указания к лабораторным работам /В.В. Извозчикова. – Оренбургский гос. ун.-т. – Оренбург: ОГУ, 2015. – 29 с.

Лабораторный практикум состоит из одной лабораторной работы и посвящен изучению вопросов сопряжения нестандартных устройств к ПК и программированию микроконтроллеров. Лабораторная работа содержит подробную методику выполнения работы с наглядными иллюстрациями.

Методические указания к лабораторным работам позволяют получить навыки по сопряжению нестандартных устройств к персональному компьютеру, познакомиться с возможностями микроконтроллера Arduino и научиться программировать микроконтроллер в составе с датчиком расстояния HC-SR04. Предназначены для выполнения лабораторных работ по дисциплине «Основы микропроцессорной техники» для бакалавров по направлению 09.03.02 Информационные системы и технологии очной и заочной форм обучения.

УДК 004.3/.4'23(075.3)

ББК 32.973-018.я73

© Извозчикова В.В., 2015

© ОГУ, 2015

Содержание

Введение.....	4
1 Микроконтроллер Arduino.....	5
1.1 Апператная часть платформы Arduino.....	7
1.2 Версии платформы Arduino.....	7
1.3 Платы расширения.....	9
1.4 Характеристики микроконтроллера ArduinoPro Micro.....	10
1.5 Распиновка контактов ArduinoPro Micro.....	11
1.6 Схема подключения датчика HC-SR04 к микроконтроллеру Arduino.....	12
2 Разработка драйвера для сопряжения датчика HC-SR04 с микроконтроллером Arduino.....	14
2.1 Установка среды разработки Arduino.....	14
2.2 Описание среды разработки Arduino.....	15
2.3 Пример написания драйвера для сопряжения.....	16
2.3.1 Написание драйвера без использования библиотек.....	16
2.3.2 Написание драйвера с использованием библиотек.....	16
2.4 Загрузка и настройка программного обеспечения Gobetwino.....	19
2.4.1 Загрузка ПО Gobetwino.....	19
2.4.2 Настройка ПО Gobetwino.....	19
2.5 Разработка драйвера для ПО Gobetwino.....	24
2.5.1 Модификация кода без использования библиотеки для вывода в файл.....	25
2.5.2 Модификация кода с использованием библиотеки для вывода в файл.....	25
2.6 Пример работы системы с выводом информации на экран монитора и с записью в файл.....	26
2.6.1 Запись в файл.....	26
3 Задание к лабораторной работе.....	28

Введение

В настоящее время наблюдается бурный прогресс в развитии персональных компьютеров и их стали внедрять в такие области, где раньше вычислительная техника не использовалась. Особенное распространение получили ЭВМ типа IBM PC или совместимые с ними. Но компьютер работает только с цифровой информацией, а существуют ещё и аналоговые сигналы, которые нужно принимать, обрабатывать, хранить и выдавать пользователю. В связи с этим, важное значение приобретает проблема связи объектов, выдающих аналоговые сигналы, с персональными компьютерами.

Для обработки аналоговых и цифровых сигналов разработана большая номенклатура микросхем, среди которых можно отметить аналого-цифровые и цифро-аналоговые преобразователи, компараторы, элементы выборки и хранения, устройства управления вводом/выводом, программируемые параллельные и последовательные интерфейсы и т.д. Это сильно упрощает разработку систем сопряжения с аналоговыми объектами. Если раньше нужно было разрабатывать целый проект, от травления платы до написания разных драйверов, то сейчас существуют унифицированные микроконтроллеры, с помощью которых можно в очень короткие сроки, с минимальными затратами, зная основы языка C++, построить интересующую нас систему.

В данном методическом пособии рассмотрены аппаратно-программные средства для построения простых систем автоматики на базе микроконтроллера Arduino, которые реализованы в виде небольшого устройства. Устройство работает совместно с персональным компьютером, и способно обрабатывать и хранить данные принятые с датчика.

Система автоматизации сбора информации состоит из датчика, контроллера и компьютера типа IBM PC. Система измеряет расстояние до препятствия и выводит полученные данные на экран компьютера или записывает их в файл для последующей обработки.

1 Микроконтроллер Arduino

Arduino – это инструмент для проектирования электронных устройств (электронный конструктор) более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Внешний вид микроконтроллера представлен на рисунке 1.1. Это платформа, предназначенная для «physical computing» с открытым программным кодом, построенная на простой печатной плате с современной средой для написания программного обеспечения.

Arduino применяется для создания электронных устройств с возможностью приема сигналов от различных цифровых и аналоговых датчиков, которые могут быть подключены к нему, и управления различными исполнительными устройствами. Проекты устройств, основанные на Arduino, могут работать самостоятельно или взаимодействовать с программным обеспечением на компьютере (например: Flash, Processing, MaxMSP). Платы могут быть собраны пользователем самостоятельно или куплены в сборе. Среда разработки программ с открытым исходным текстом доступна для бесплатного скачивания.

Язык программирования Arduino является реализацией Wiring, схожей платформы для «physical computing», основанной на мультимедийной среде программирования Processing.

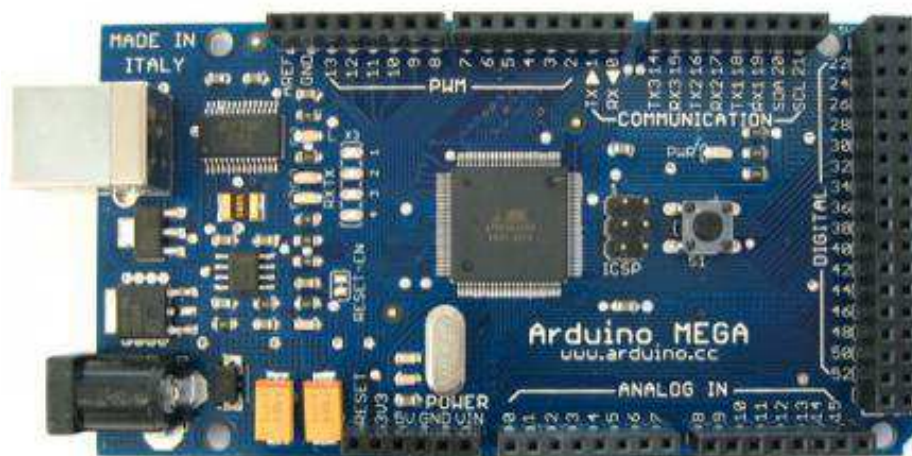


Рисунок 1.1 - Микроконтроллер Arduino Mega

Существует множество микроконтроллеров и платформ для осуществления «physical computing». Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard и многие другие предлагают схожую функциональность. Все эти устройства объединяют разрозненную информацию о программировании и заключают ее в простую в использовании сборку. Arduino, в свою очередь, тоже упрощает процесс работы с микроконтроллерами, однако имеет ряд преимуществ перед другими устройствами, которые перечислены ниже.

1. Низкая стоимость – платы Arduino относительно дешевы по сравнению с другими платформами. Самая недорогая версия модуля Arduino может быть собрана в ручную, а некоторые даже готовые модули стоят меньше 50 долларов.

2. Кросс-платформенность – программное обеспечение Arduino работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows.

3. Простая и понятная среда программирования – среда Arduino подходит как для начинающих пользователей, так и для опытных. Arduino основана на среде программирования Processing, что очень удобно для преподавателей, так как студенты работающие с данной средой будут знакомы и с Arduino.

4. Программное обеспечение с возможностью расширения и открытым исходным текстом – ПО Arduino выпускается как инструмент, который может быть дополнен опытными пользователями. Язык может дополняться библиотеками C++. Пользователи, желающие понять технические нюансы, имеют возможность перейти на язык AVR C на котором основан C++. Соответственно, имеется возможность добавить код из среды AVR-C в программу Arduino.

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

1.1 Аппаратная часть платформы Arduino

Существует несколько версий платформ Arduino. Последняя версия (на момент написания пособия) Due, базируется на микроконтроллере Atmel SAM3X8E ARM Cortex-M3. Это первая плата Arduino на основе 32-битного микроконтроллера с ARM ядром. Предпоследняя версия Leonardo, базируется на микроконтроллере ATmega32u4. Uno, как и предыдущая за ней, версия Duemilanove построены на микроконтроллере Atmel ATmega328. Старые версии платформы Diecimila и первая рабочая Duemilanoves были разработаны на основе Atmel ATmega168, более ранние версии использовали ATmega8. Arduino Mega2560, в свою очередь, построена на микроконтроллере ATmega2560.

1.2 Версии платформы Arduino

Due — самая новая (на момент написания) плата на базе ARM микропроцессора 32bit Cortex-M3 ARM SAM3U4E.

Leonardo — предпоследняя версия платформы Arduino на ATmega32u4 микроконтроллере. Имеет разъем micro-USB, по размерам совпадает с UNO.

Yun (описание на англ.) - новая плата, с встроенной поддержкой WiFi на базе ATmega32u4 and the Atheros AR9331.

Micro — новое компактное решение на базе ATmega32u4.

Uno — самая популярная версия базовой платформы Arduino USB. Uno имеет стандартный порт USB. Arduino Uno во многом схожа с Duemilanove, но имеет новый чип ATmega8U2 для последовательного подключения по USB и новую, более удобную маркировку вход/выходов. Платформа может быть дополнена платами расширения, например, пользовательскими платами с различными функциями.

Arduino Ethernet — контроллер со встроенной поддержкой работы по сети и с опциональной возможностью питания по сети с помощью модуля POE (Power over Ethernet).

Duemilanove — является предпоследней версией базовой платформы Arduino USB. Подключение Duemilanove производится стандартным кабелем USB. После подключения она готова к использованию. Платформа может быть дополнена платами расширения, например, пользовательскими платами с различными функциями.

Diecimila — предыдущая версия базовой платформы Arduino USB.

Nano — это компактная платформа, используемая как макет. Nano подключается к компьютеру при помощи кабеля USB Mini-B.

Mega ADK - версия платы Mega 2560 с поддержкой USB host интерфейса для связи с телефонами на Android и другими устройствами с USB интерфейсом.

Mega2560 – новая версия платы серии Mega. Построена на базе Atmega2560 и с использованием чипа ATMega8U2 для последовательного соединения по USB порту.

Mega – предыдущая версия серии Mega на базе Atmega1280.

Arduino BT платформа с модулем Bluetooth для беспроводной связи и программирования. Совместима с платами расширения Arduino.

LilyPad– платформа, пурпурного цвета, разработанная для переноски, может зашиваться в ткань.

Fio – платформа разработана для беспроводных применений. Fio содержит разъем для радио XBee, разъем для батареи LiPo и встроенную схему подзарядки.

Mini – самая маленькая платформа Arduino. Прекрасно работает как макетная модель, или, в проектах, где пространство является критическим параметром. Платформа подключается к компьютеру при помощи адаптера Mini USB.

Адаптер Mini USB – плата, конвертирующая подключение USB в линии 5 В, GND, TX и RX для соединения с платформой Arduino Mini или другими микроконтроллерами.

Pro – платформа, разработанная для опытных пользователей, может являться частью большего проекта. Она дешевле, чем Diecimila и может

питаться от аккумуляторной батареи, но в тоже время требует дополнительной сборки и компонентов.

Pro Mini – как и платформа Pro разработана для опытных пользователей, которым требуется низкая цена, меньшие размеры и дополнительная функциональность.

Serial – базовая платформа с интерфейсом RS232 для связи и программирования. Плата легко собирается даже начинающими пользователями. (включает схемы и файлы CAD)

Serial Single Sided – платформа разработана для ручной сборки. Она обладает чуть большим размером, чем Diecimila, но совместима с платами расширения Arduino.

USB Serial Light Адаптер - адаптер, позволяющий подключать платы Arduino к компьютеру для обмена данными и заливки скетчей. Удобен для программирования таких плат, как Arduino Mini, Arduino Ethernet и других, не имеющих своего разъема USB.

1. 3 Платы расширения

Платы расширения устанавливаемыми на платформы, являются платы, расширяющие функциональность Arduino для управления различными устройствами, получения данных и т.д. Пример платы приведен на рисунке 1.2.

1. Плата расширения WiFi, используется для соединения с беспроводными сетями стандарта 802.11 b/g.

2. Плата расширения Xbee Shield, обеспечивает при помощи модуля Maxstream Xbee Zigbee беспроводную связь, нескольким устройствам Arduino в радиусе до 35 метров (в помещении) и до 90 метров (вне помещения).

3. Плата расширения Motor Shield, обеспечивает управление двигателями постоянного тока и чтение датчиков положения.

4. Плата расширения Ethernet Shield, обеспечивает подключение к интернету.



Рисунок 1.2 - Плата расширения WiFi

1. 4 Характеристики микроконтроллера Arduino Pro Micro

Arduino Pro Micro, упрощённая версия Arduino Leonardo. Она отличается, меньшим количеством разъёмов, и меньшим размером платы. Создана на базе процессора ATmega32u4. Микроконтроллер представлен на рисунке 1.3.

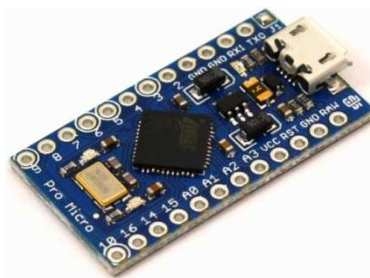


Рисунок 1.3 - Arduino Pro micro

Плата имеет 18 цифровых вход/выходов (из них 5 могут использоваться в качестве выходов ШИМ и 4 - как аналоговые входы), кварцевый генератор с частотой 16 МГц, разъем micro-USB.

Чтобы запустить Arduino Pro Micro, нужно подключить его к ПК с помощью кабеля micro-USB. Форм-фактор контроллера позволяет легко разместить его на макетной плате.

Micro Pro схож с Arduino Leonardo тем, что ATmega32u4 имеет встроенную поддержку USB-соединения, благодаря чему не требуется вспомогательный процессор. Это позволяет Pro Micro определяться на подключенном компьютере в качестве мыши или клавиатуры в дополнение к виртуальному (CDC) последовательному COM порту. Рассмотрим основные характеристики.

1. Рабочее напряжение: 5 В.
2. Входное напряжение (рекомендуемое): 7-12 В.

3. Входное напряжение (предельное): 6-20 В.
4. Цифровые Входы/Выходы: 18 (5 из которых ШИМ).
5. Аналоговые каналы: 12.
6. Постоянный ток через вход/выход: 40 мА.
7. Постоянный ток для вывода: 50 мА.
8. Флеш-память: 32 Кб (ATmega32u4) из которых 4 Кб используются для загрузчика.
9. ОЗУ: 2 Кб (ATmega32u4).
10. EEPROM: 1 Кб (ATmega32u4).
11. Тактовая частота: 16 МГц.

1.5 Распиновка контактов Arduino Pro Micro

На рисунке 1.4 представлен микроконтроллер Arduino Pro Micro с обозначением выводов, а в таблице 1 приведено их описание.

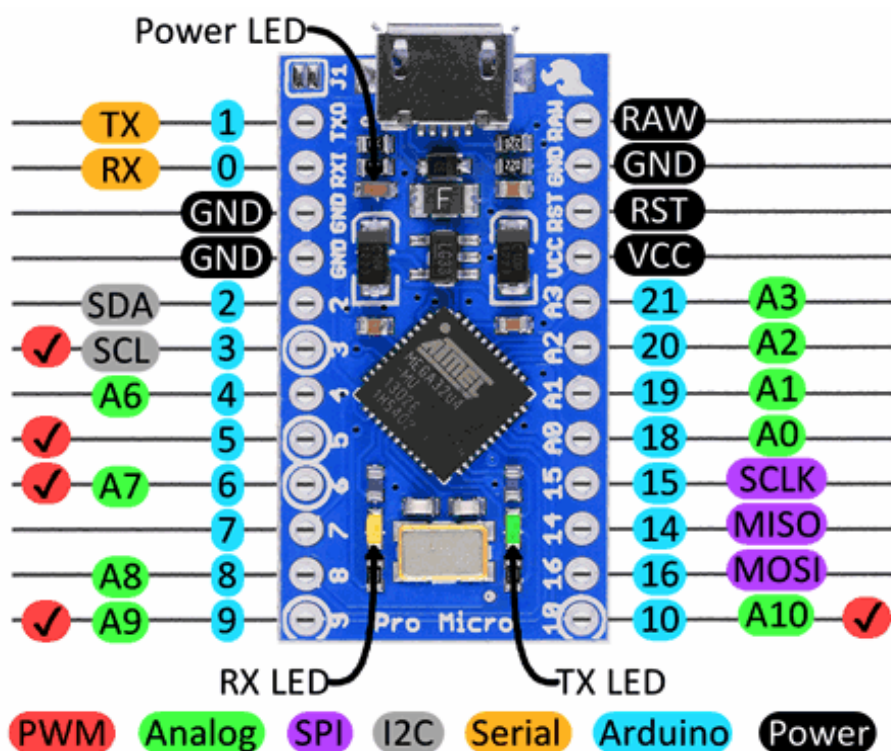


Рисунок 1.4 - Распиновка контактов

Таблица 1 - Описание выводов МК

Вывод	Описание
TX	Передача данных TTL
RX	Получение данных TTL
GND	Земля
RAW	Внешний источник питания следует подключать через этот вывод, допустимое напряжение - до 12 В
RST	Вывод сброса микроконтроллера
VCC	Напряжение 5 В, подаваемое на микроконтроллер
A0	Аналоговый вывод
A1	Аналоговый вывод
A2	Аналоговый вывод
A3	Аналоговый вывод
2	вывод SDA, интерфейс I2C / TWI
3	вывод SCL, интерфейс I2C / TWI, а так же ШИМ
4	Аналоговый вывод, а так же ШИМ
5	ШИМ вывод
6	Аналоговый вывод, а так же ШИМ
7	Аналоговый вывод
8	Аналоговый вывод, а так же ШИМ
9	Аналоговый вывод, а так же ШИМ
10	Аналоговый вывод, а так же ШИМ
14	MISO, интерфейс SPI
15	SCLK, интерфейс SPI
16	MOSI, интерфейс SPI

1.6 Схема подключения датчика HC-SR04 к микроконтроллеру Arduino

На рисунке 1.5 представлена схема подключения датчика расстояния HC-SR04 к микроконтроллеру Arduino.

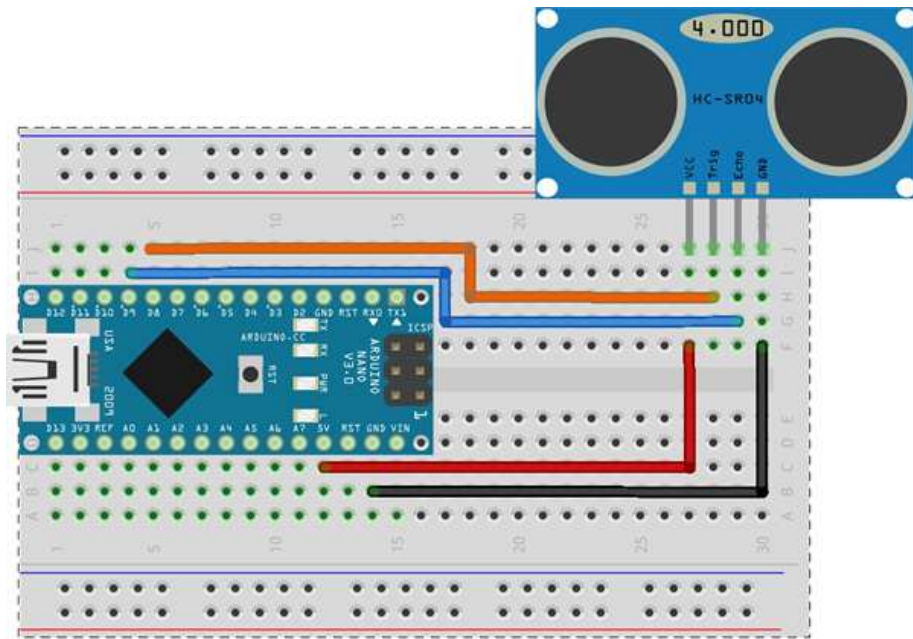


Рисунок 1.5 - Схема подключения

2 Разработка драйвера для сопряжения датчика HC-SR04 с микроконтроллером Arduino

2.1 Установка среды разработки Arduino

Для того что бы начать работу, нужно установить среду разработки Arduino. Загрузить ее нужно по ссылке <https://www.arduino.cc/en/Main/Software>.

Когда мы перейдем по ссылке, увидим часть такой страницы, как показано на рисунке 2.1.

Download the Arduino Software



Рисунок 2.1 - Часть страницы, для загрузки ПО Arduino

На данный момент самая актуальная версия ПО, 1.6.4. На выбор, для ОС Windows, нам предоставляется, два варианта загрузки. Первый Windows Installer, и второй Windows ZIP file. Как видно из названия, первый вариант, загружается в установочном пакете, а второй ZIP-архивом, который впоследствии можно разархивировать в удобную для нас директорию.

После того как мы установили ПО, нужно установить драйвер для микроконтроллера. Для этого нужно, подключить микроконтроллер к ПК посредством Micro-USB кабеля. Затем зайти в «Диспетчер устройств», выбрать не установленное устройство, как показано на рисунке 2.2, правой кнопкой мыши нажать на «Обновить драйвер». Указать «Поиск в ручном режиме», выбрать папку с драйверами, находящуюся в директории среды разработки Arduino. После этого, МК, установится в систему, и он готов к работе.

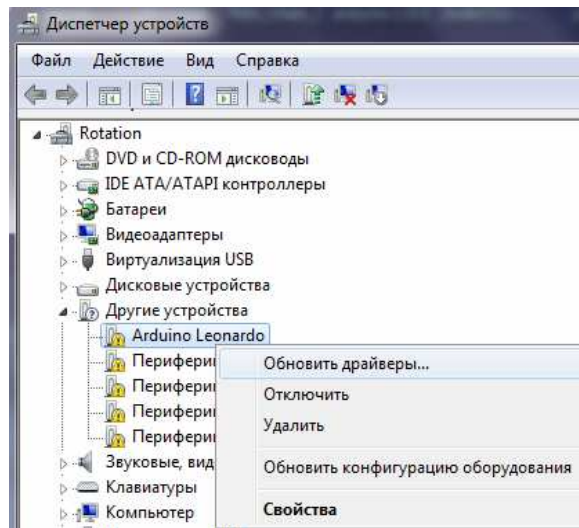


Рисунок 2.2 - Установка драйвера

2.2 Описание среды разработки Arduino

После установки драйвера, запустим среду разработки, согласно рисунку 2.3. Среда разработки Arduino, состоит из встроенного текстового редактора, программного кода, области сообщений, окна вывода текста(консоли), панели инструментов с кнопками, часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino. Язык для написания программ, основывается на C++. Что имеет большой плюс, для разработки драйверов.

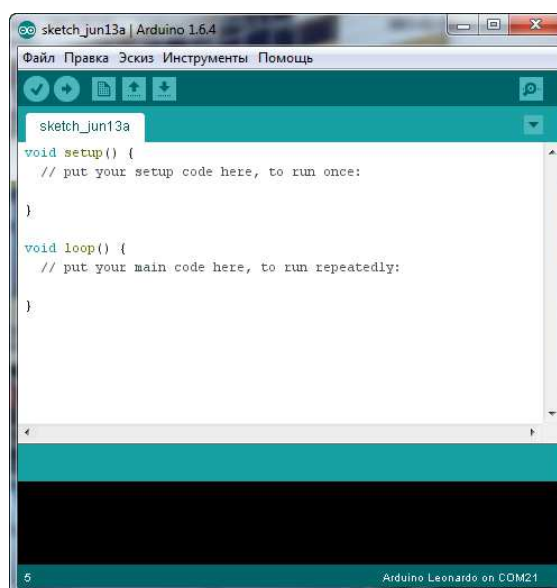


Рисунок 2.3 - Среда разработки Arduino

2.3 Пример написания драйвера для сопряжения

Для облегчения написания программного обеспечения (ПО), существуют готовые библиотеки, под распространённые датчики, и разные платы расширения. Рассмотрим два случая:

- 1) сами пишем ПО;
- 2) используем готовую библиотеку.

2.3.1 Написание драйвера без использования библиотек

Рассмотрим частично программный код. Программа инициализация контактов в МК приведена на рисунке 2.4.

Функция для подсчета расстояния приведена на рисунке 2.5.

```
int echoPin = 9; Назначаем контактам, переменные, для  
удобства ориентирования в коде  
int trigPin = 8;  
  
void setup() {  
  Serial.begin (9600); Инициализируем серийный порт, на  
скоростях 9600 Бод  
  pinMode(trigPin, OUTPUT); Указываем контроллеру, что  
переменная trigPin, то есть 8 контакт, будет использоваться  
как выход  
  pinMode(echoPin, INPUT); 9 контакт, будет использовать-  
ся как вход.  
}
```

Рисунок 2.4 - Фрагмент драйвера без использования библиотек

`void loop() {` это стандартный и обязательный метод при программировании под `Arduino`. Он циклически запускается (после окончания всегда начинает с начала).

`int duration, cm;` Вводим переменные длины, сантиметров.

`digitalWrite(trigPin, LOW);` В начале мы выставляем на нашем триггере логический ноль. Делается это для того, чтобы можно было быть уверенным, что никакая другая часть программы не установила (быть может по случайности) на этот вывод логическую единицу.

`delayMicroseconds(2);` После этого мы ждем две микросекунды (очень-очень мало) но этого более чем достаточно, чтобы "начать с нуля", так сказать.

`digitalWrite(trigPin, HIGH);` После этого, мы устанавливаем логическую единицу.

`delayMicroseconds(10);` На 10 микросекунд (Согласно спецификации сенсора это именно та минимальная продолжительность импульса на триггере, после которой генерируется ультразвуковой сигнал).

`digitalWrite(trigPin, LOW);` Выключаем. После этого мы начинаем "слушать эфир".

`duration = pulseIn(echoPin, HIGH);` Функция `pulseIn(...)` возвращает `duration` значение в микросекундах через то время, когда на указанном входе появится логическая единица (в нашем случае это вход номер 9). Как только на входе 9 появляется логическая единица - этот метод возвращает количество микросекунд за которое сигнал вернулся.

`cm = duration / 58;` Полученное время, делим на 58 (константа, посчитанная производителем датчика, и взятая из технической документации к датчику) и получаем в переменной `cm` значение расстояния.

Рисунок 2.5 - Фрагмент драйвера функции подсчета расстояния

Теперь, полученное значение осталось вывести на монитор серийного порта, или сохранить в файл. Если выйти за рамки данных указаний, то вывод возможен, на отдельный ЖК дисплей, флэш - карту формата MicroSD, или на веб-сайт.

2.3.2 Пример написания драйвера с использованием библиотек

Метод с использованием библиотеки намного проще. На рисунке 2.6 приведен пример программного кода.

```
#include <Ultrasonic.h>    Подключение библиотеки
Ultrasonic ultrasonic(8,9);  Подключение контактов
void setup() {
  Serial.begin(9600);      Открытие серийного порта на скорости
9600 бод
}
void loop()
{
  Serial.print(ultrasonic.Ranging(CM));  Функция передает дан-
ные через последовательный порт, в данном случае в этой пере-
менной возвращается, посчитанное в библиотеке, расстояние.
  Serial.println("cm");
  delay(1000);  Пауза 1с
}
```

Рисунок 2.6 - Фрагмент драйвера с использованием библиотек

После запуска вашей программы на экране монитора будет выведен результат, пример которого показан на рисунке 2.7.

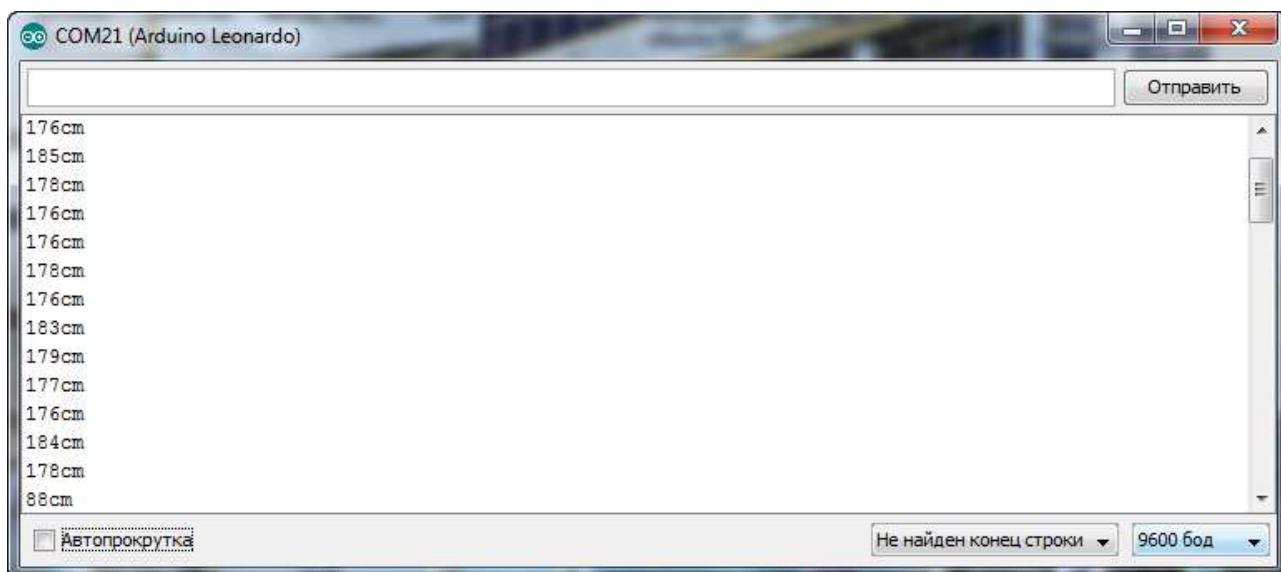


Рисунок 2.7- Пример работы системы

Далее будет рассмотрен способ записи в файл, полученных данных

2.4 Загрузка и настройка программного обеспечения Gobetwino

Стандартная среда разработки имеет последовательный серийный монитор только с возможностью просмотра содержимого, пересылаемого через этот порт. Поэтому нам понадобится стороннее ПО Gobetwino. Это монитор последовательного порта, с возможностью записи в файл.

2.4.1 Загрузка ПО Gobetwino

Данный дистрибутив, загружается по ссылке <http://www.mikmo.dk/gobetwino.html>

Открыв страницу, приведенную на рисунке 2.8, нажимаем на ссылку с надписью «download here», загружаем архив в нужную директорию. Разархивируем в папку и приступаем к настройке.



Рисунок 2.8 - Часть страницы загрузки ПО Gobetwino

2.4.2 Настройка ПО Gobetwino

Запустим программу согласно рисунку 2.9.

Интерфейс программы достаточно простой, он состоит из верхнего меню, и двух окон вывода сообщений. Верхнее окно выводит данные поступающие в серийный порт, а нижнее уведомляет о командах записи. Gobetwino, имеет

возможность сохранять данные не только в файл, но и отсылать полученные данные на электронную почту, веб - сайт.

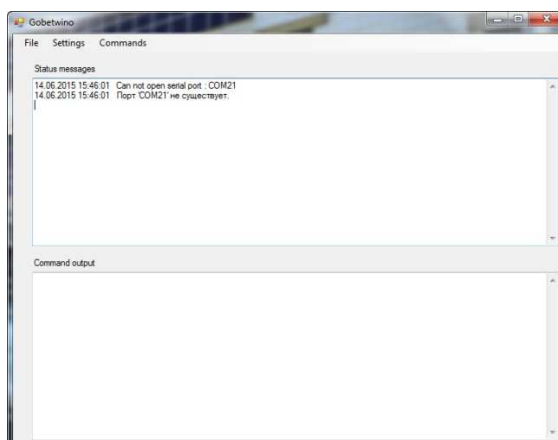


Рисунок 2.9 - Интерфейс программы Gobetwino

Для настройки ПО, нужно нажать на меню «Settings» и откроется следующее окно, показанное на рисунке 2.10.

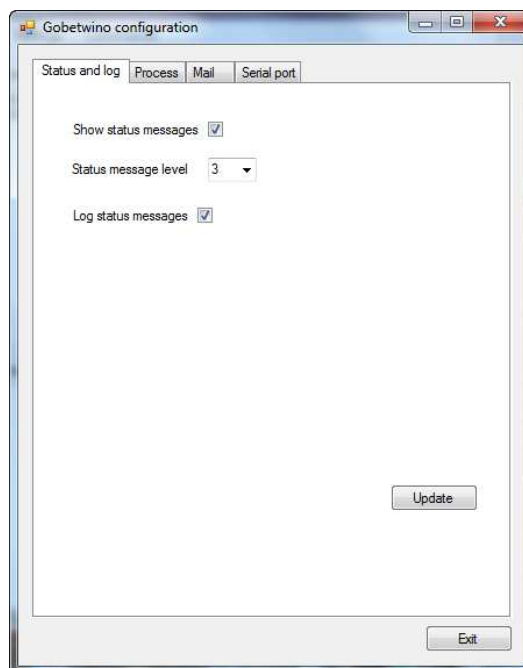


Рисунок 2.10- Окно конфигураций

Переходим на вкладку «Serial port» получаем окно, показанное на рисунке 2.11 . Выбираем в строке «Serial port» тот COM порт на котором находится микроконтроллер Arduino. В строчке «Baud rate» убеждаемся, что прописано значение 9600. Нажимаем кнопку «Update», а затем «Exit».

Перейдем к настройке команды записи в файл. Для этого в главном меню программы, приведенном на рисунке 2.9, выберем меню «Commands». Откроется следующее окно, как показано на рисунке 2.12.

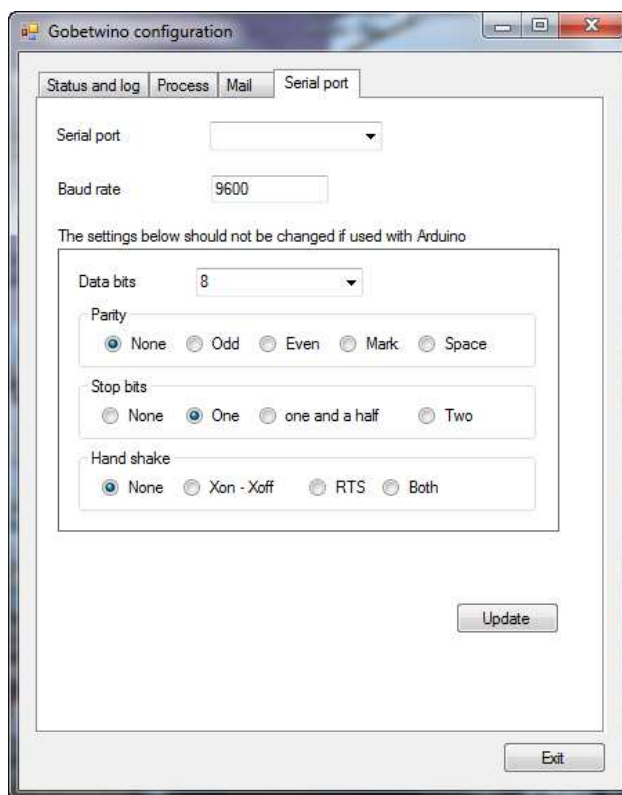


Рисунок 2.11 - Вкладка Serial port

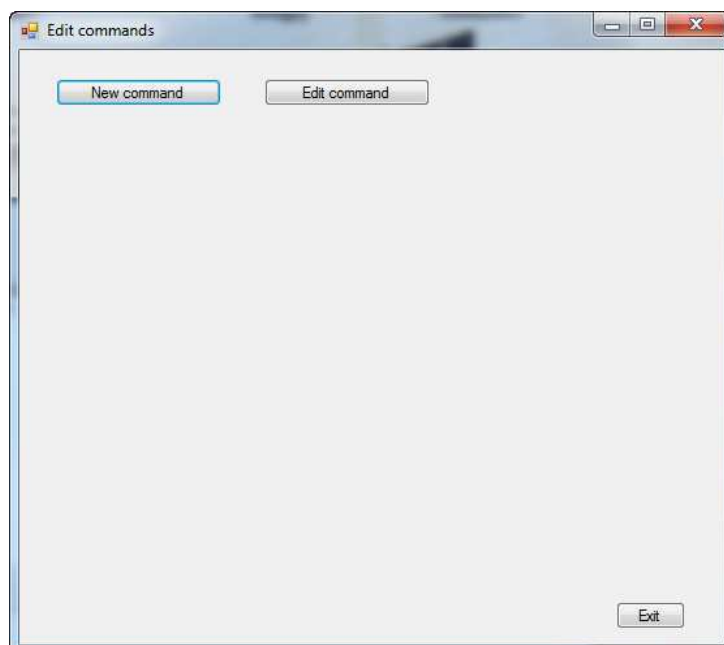


Рисунок 2.12 - Окно редактирования команд

Нажимаем кнопку New command, активируется скрытое меню, приведенное на рисунке 2.13.

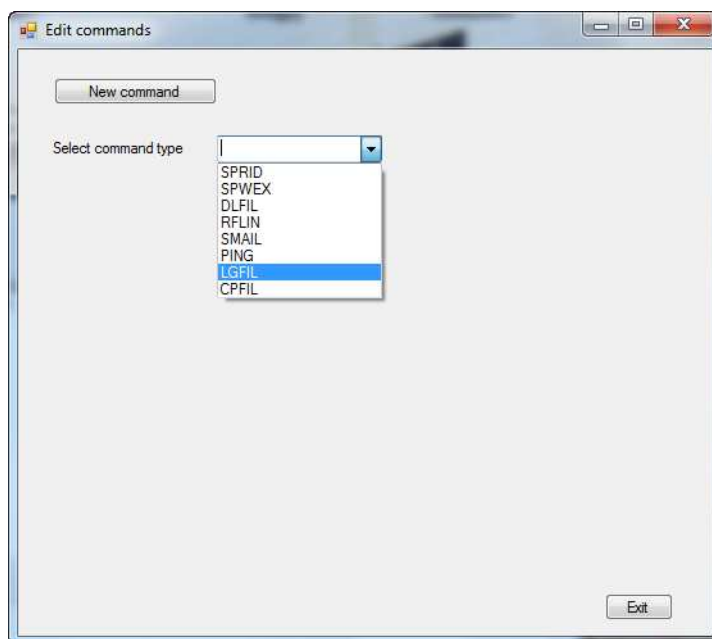


Рисунок 2.13 - Меню выбора команды

Из всего списка выбираем «LGFIL» – запись данных в файл. Пример настройки параметров приведен на рисунке 2.14.

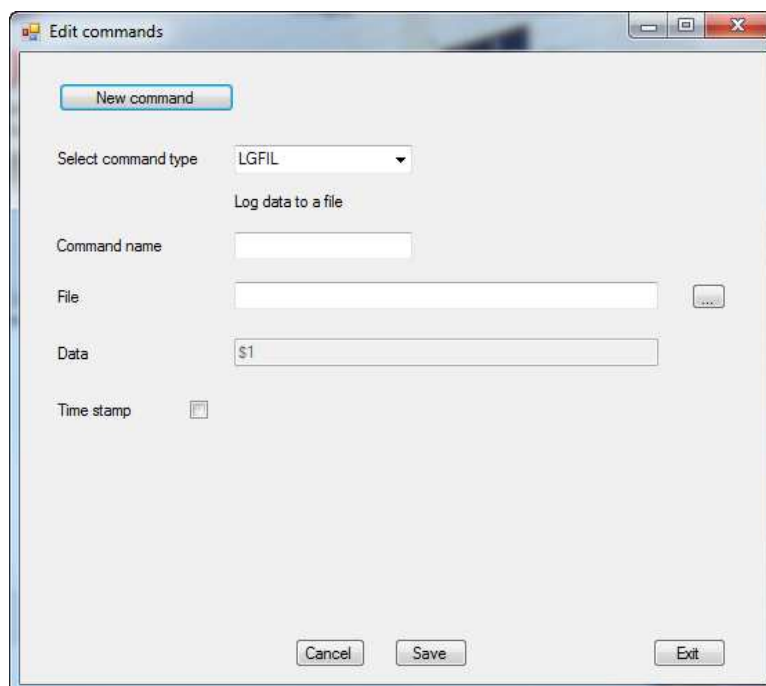


Рисунок 2.14 - Параметры настройки

В строчке «Command name», указываем имя, которое будет использоваться в драйвере МК. Назовем его, LOGON. В строке «File» указываем путь к нашему файлу для записи. Создаем пустой файл, к примеру с расширением .txt, и указываем к нему путь. Ставим галочку напротив «Time stamp» – при записи значений, будет указываться дата и время. Нажимаем кнопку «Save», потом «Exit».

Что бы убедиться в правильности настроек, а также иметь возможность редактировать название или путь к файлу, перейдем в меню, приведенному на рисунке 2.15. Нажмем «Edit command» и выбираем нашу команду LOGON.

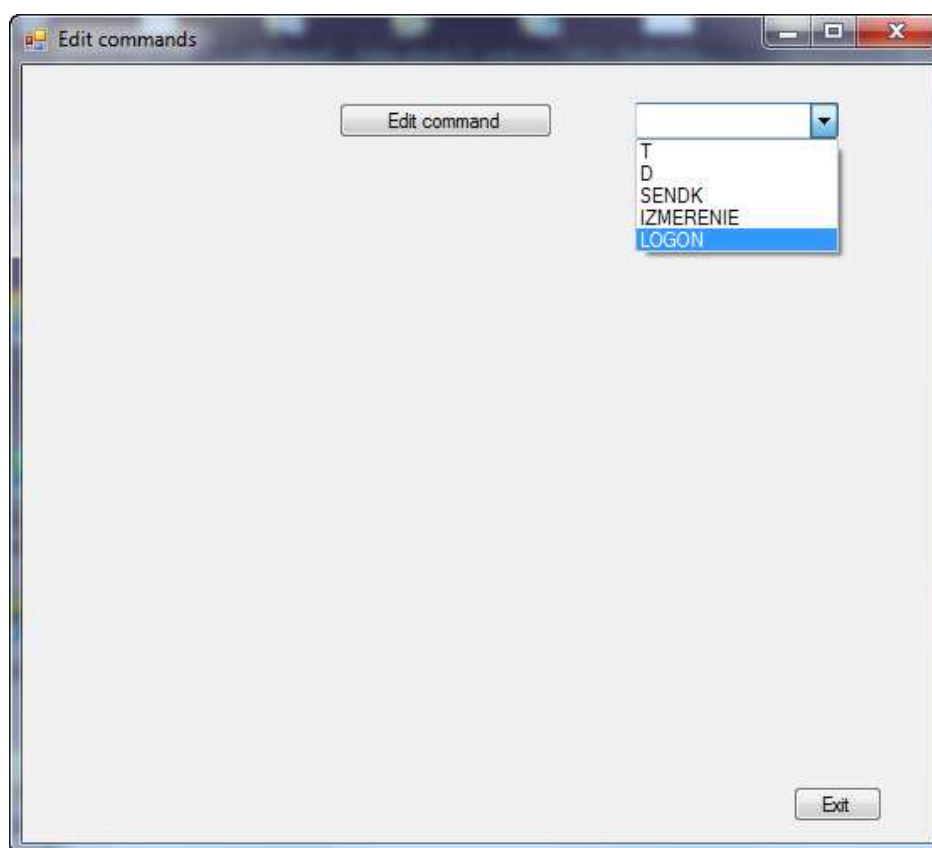


Рисунок 2.15 - Выбор команды LOGON

После нажатия, появляется окно, представленное на рисунке 2.16. Теперь, если нужно, мы можем изменить имя файла, расположение, а так же убрать указание даты и времени.

Настройка ПО Gobetwino закончена, перейдем к написанию драйвера.

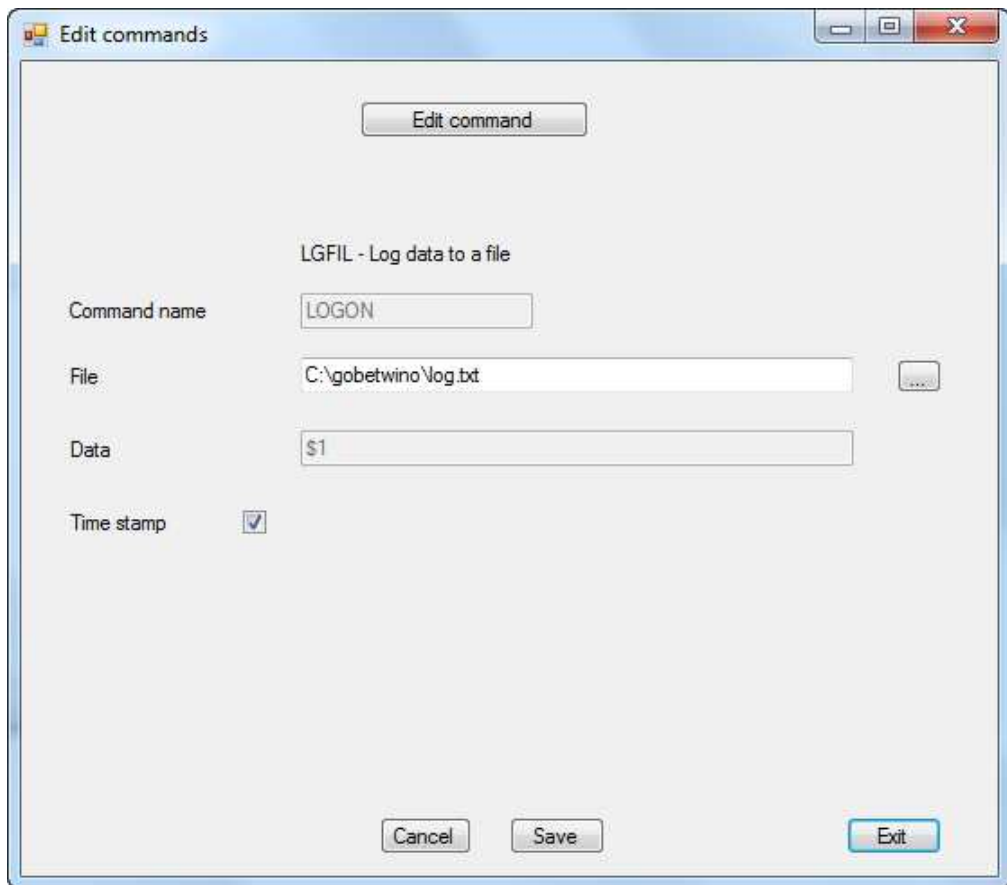


Рисунок 2.16- Команда LOGON

2.5 Разработка драйвера для ПО Gobetwino

Вернемся к среде разработки Arduino, и модифицируем наш код. Для того чтобы Gobetwino начал записывать поступающие данные в файл через серийный порт, ему нужно указать так называемые “метки” начала и конца записи. Поскольку в микроконтроллере Arduino выводом данных на серийный порт служит функция `Serial.print()`, то внутри нее мы и добавим нужный нам код.

Само тело команды представляет следующий вид.

```
#S|Название команды|[Имя переменной, которую нужно сохранить]#
```

Далее рассмотрим модификацию кода для двух версий драйверов, рассмотренных выше. Напомним, что первый написан вручную, а второй с использованием библиотеки.

2.5.1 Модификация кода без использования библиотеки для вывода в файл

Произведем модификацию кода путем указания `Gobetwino` где начинать, а где заканчивать запись, как показано на рисунке 2.17.

```
Serial.print("#S|LOGON|["); Указываем нашу команду LOGON, созданную в разделе 2.6, а также ставим символ квадратных скобок [ , тем самым указывая начало записи.  
Serial.print(cm); Переменная содержащая в себе значения расстояния, будет записываться в файл  
Serial.print(" cm"); величина переменной, тоже будет записываться в файл  
Serial.println("]#"); квадратная скобка ] указывает на конец записи в файл, # указывает на конец команды LOGON.
```

Рисунок 2.17 - Фрагмент кода без использования библиотеки

2.5.2 Модификация кода с использованием библиотеки для вывода в файл

Применим те же модификации кода, что и ранее, только с использованием переменной из библиотеки, согласно рисунку 2.18.

```
Serial.print("#S|LOGON|["); Указываем нашу команду LOGON, созданную в разделе 2.6, а также ставим символ квадратных скобок [ , тем самым указывая начало записи.  
Serial.print(ultrasonic.Ranging(CM)); Переменная содержащая в себе значения расстояния, будет записываться в файл  
Serial.print(" cm"); величина переменной, тоже будет записываться в файл  
Serial.println("]#"); квадратная скобка ] указывает на конец записи в файл, # указывает на конец команды LOGON.
```

Рисунок 2.18 - Фрагмент кода с использованием библиотеки

2.6 Пример работы системы с выводом информации на экран монитора и с записью в файл

Пример работы системы через монитор серийного порта Gobetwino приведен на рисунке 2.19.

В окне «Status messages», выводится информация о состоянии открытия порта, а также информация о найденной функции в драйвере для старта записи. Выводятся сообщения об успешном выполнении команды, указывается путь до места хранения файла-лога.

В нижнем окне «Command output» выводится информация об ошибках или предупреждениях. В данном примере указан 0, значит все в порядке.

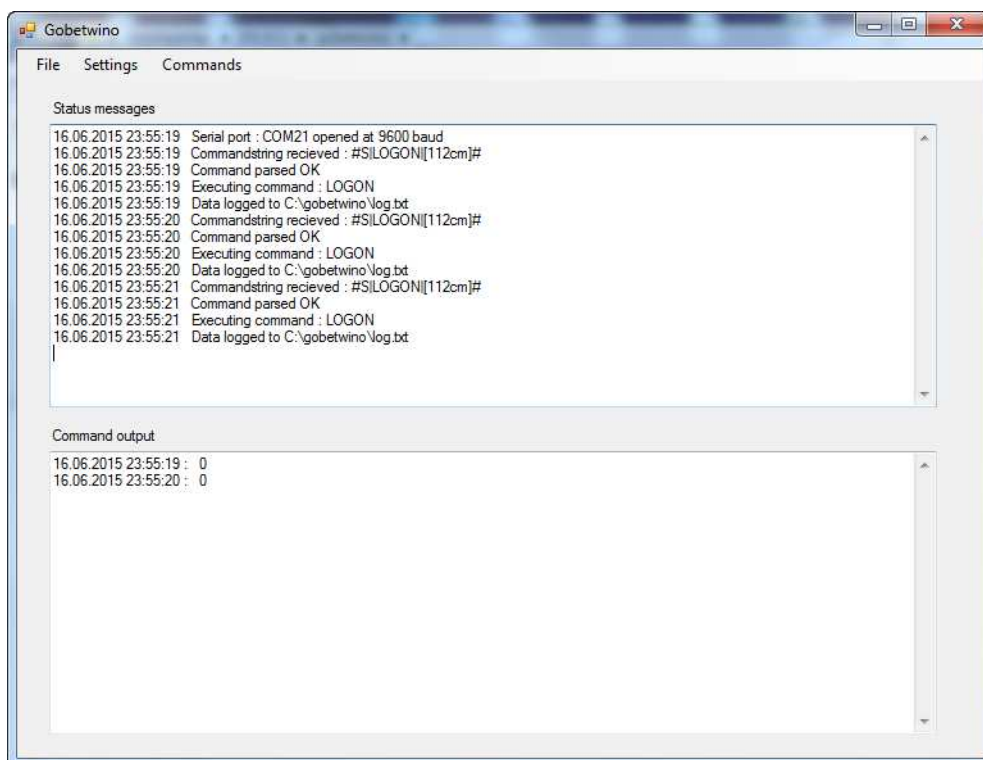


Рисунок 2.19 - Работа серийного порта Gobetwino

2.6.1 Запись в файл

Структура файла представлена на рисунке 2.20. В таком формате данные из файла, могут быть импортированы в Microsoft Excel для дальнейшей обработки.

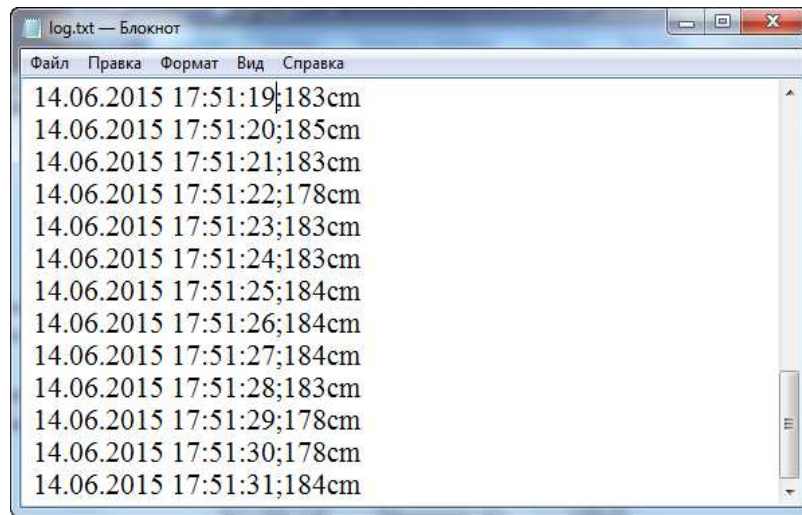


Рисунок 2.20 - Запись в файл

3 Задание к лабораторной работе

Цель работы: Получить навыки по сопряжению нестандартных устройств к персональному компьютеру, познакомиться с возможностями микроконтроллера Arduino и научиться программировать микроконтроллер в составе с датчиком расстояния HC-SR04.

Задание 1. Подключить к системе светодиод согласно приведенной схеме на рисунке 3.1, и написать программу по управлению работой светодиода, задавая различные значения частоты.

Так как питание светодиода не должно превышать 2-3В (в зависимости от цвета), а на выводе может быть сигнал логической единицы, что соответствует значению 5В, то нужно обязательно подключить последовательно резистор примерно на 220 Ом.

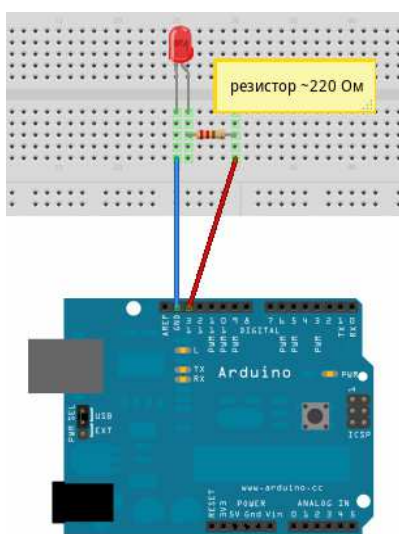


Рисунок 3.1 - Схема подключения светодиода

Задание 2. Напишите программу для управления датчиком расстояния и выведите показания на экран через монитор среды разработки Arduino.

Задание 3. Напишите программу для управления датчиком расстояния и выведите показания на экран через монитор серийного порта Gobotwino .

Задание 4. Напишите программу для управления датчиком расстояния и запишите показания в файл.

Список использованных источников

- 1 Кашкаров, А.П.. Датчики в электронных схемах: от простого к сложному/ А.П. Кашкаров, Издательство: ДМК Прес , 2013. - 200 с.
- 2 Соммер, Улли. Программирование микроконтроллерных плат Arduino /Freeduino / Улли Соммер. Издательство: БХВ-Петербург, 2012. - 256 с. – ISBN 978-5-9775-0727-1.
- 3 Петин, В.А.. Проекты с использованием контроллера Arduino / В.А. Петин. Издательство: БХВ-Петербург, 2014. - 448 с. - ISBN 978-5-9775-3550-2.
- 4 Том, Иго. Arduino, датчики и сети для связи устройств / Иго Том. Издательство: БХВ-Петербург, 2015, -544с. – ISBN 978-5-9775-3566-3.