

Министерство образования Республики Беларусь
Учреждение образования
«Гомельский государственный технический университет
имени П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ПРАКТИЧЕСКИЕ РАБОТЫ
ПО ДИСЦИПЛИНЕ
«ПРОГРАММИРОВАНИЕ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ
НА ОСНОВЕ ОДНОПЛАТНЫХ КОМПЬЮТЕРОВ»
для студентов специальности
1-40 05 01 «Информационные системы и технологии
(по направлениям)»
Соболев Д.В.

Гомель 2019

Содержание

1	Знакомство с Arduino IDE.	2 ч.
2	Знакомство с составом набора LEGO MINDSTORMS, блоком EV3 и средой разработки	4 ч.
3	Программирование роботов из стандартных наборов на Lego EV3.	4 ч.
4	Программирование роботов из стандартных наборов на Arduino.	4 ч.
5	Установка операционной системы на одноплатный компьютер	4 ч.

Практическая работа № 1

«Знакомство с Arduino IDE»

(2 часа)

1. Цель работы

Познакомиться со средой разработки Arduino IDE.

2. Знакомство с ARduino IDE

2.1 Скачать и установить утилиту Arduino IDE.

2.2 Изучить состав и соновные функции среды:

Знакомство с Arduino начинается со среды разработки Arduino IDE.

Давайте найдём ответ на вопрос: зачем нам всё таки нужна эта IDE? Ответ достаточно прост. Среда Arduino IDE необходима для разработки программ и их записи в модули Arduino (и не только).

С предназначением IDE разобрались, теперь давайте изучим, что же она из себя представляет.

Чтобы подробнее разобраться со средой Arduino IDE, дорогой читатель, её нужно скачать и установить (если её нет на компьютере).

Среда разработки Arduino IDE состоит из следующих компонентов:

- текстовый редактор кода программы (1);
- область сообщений (2);
- консоль (3);
- панель инструментов (4);
- панель с часто используемыми командами (5).

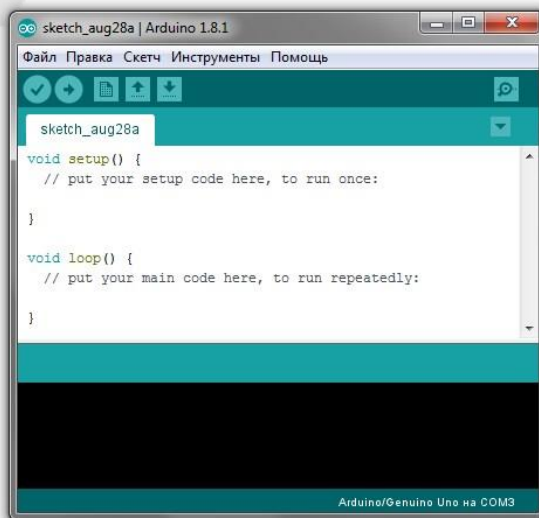
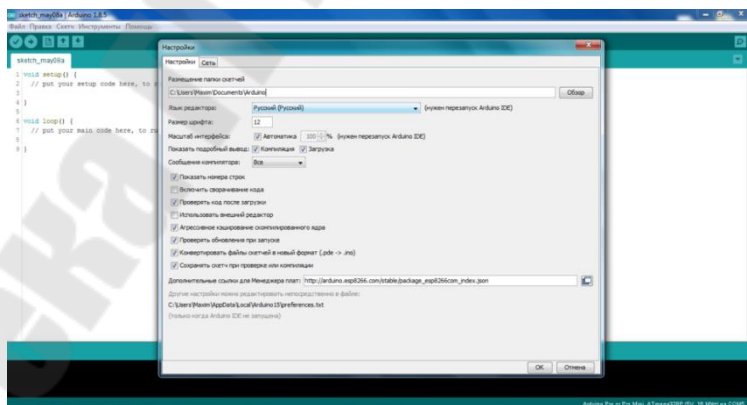


Рисунок 1 – Внешний вид Arduino IDE

Подробно останавливаться на предназначении каждого пункта меню мы не будем, а рассмотрим только самые необходимые функции. Более подробно о каждом пункте меню можно прочитать [здесь](#).

Фактически нет особого языка программирования Arduino. Модули Arduino программируются на C/C++. Особенности программирования сводятся к тому, что существует набор библиотек, включающий некоторые функции (pinMode, digitalWrite и т.д.) и объекты (Serial), которые значительно облегчают процесс написания программы.

Открыть настройки среды (рисунок 2) можно из пункта меню **Файл -> Настройки** или с помощью комбинации клавиш **Ctrl + ,**



В меню настроек можно изменить путь сохранения программ по умолчанию, изменить размер шрифтов и т.д.

Одной из особенностей Arduino IDE является довольно обширная база различных примеров, что очень удобно для начинающих. Открыть пример можно из пункта меню **Файл -> Примеры**.

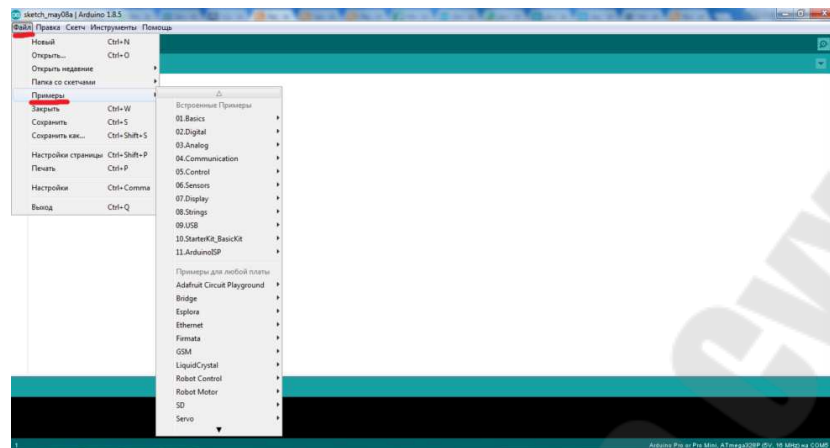


Рисунок 3 – Размещение примеров в Arduino IDE

Создание новой программы:

Итак, с настройками среды мы разобрались. Что теперь? А сейчас нужно создать новую программу. Это можно сделать несколькими способами:

- из пункта меню **Файл -> Новый**;
- при помощи комбинации клавиш **Ctrl + N**;
- из панели с часто используемыми командами/
-

Сохранение программы:

После того, как написали программу, нужно сохранить. Сделать это можно несколькими способами:

- из пункта меню **Файл -> Сохранить**;
- при помощи комбинации клавиш **Ctrl + S**;
- из панели с часто используемыми командами.

Теперь нужно только ввести имя вашей программы (**оно не должно содержать русских символов!**) и выбрать место, куда её сохранить. Сохранённая программа автоматически помещается в одноимённую папку, которая создаёт сама Arduino IDE.

Открытие программы:

Теперь мы научились создавать и сохранять программу. Но как теперь открыть сохранённую программу? Для этого в Arduino IDE предусмотрено несколько способов:

- из пункта меню **Файл -> Открыть**;
- при помощи комбинации клавиш **Ctrl + O**;
- из панели с часто используемыми командами.

В открывшемся окошке нужно выбрать папку, в которой находится нужная программа.

Редактирование текста программы:

Удобный редактор текста программы очень важен при разработке какой-либо программы. В Arduino IDE он довольно неплох, однако уступает конкурентам, таким как Eclipse, Visual Studio и т.д. Однако, его вполне достаточно. В редакторе присутствуют все основные команды, необходимые при редактировании кода. Они находятся в меню **Правка**. Для самых часто используемых команд (копировать, ставить и т.д.) существуют комбинации, способствующие быстрому доступу к нужной команде правки, что очень удобно. Другими отличительными особенностями встроенного редактора кода являются возможность копирования кода для форумов и в html формате, что позволяет делиться Вашими программами, сохраняя наглядность разметки в виде BB кодов или html разметки соответственно.

Давайте остановимся на основных командах, необходимых для редактирования программы:

- копировать. Пункт меню **Правка** -> **Копировать** или комбинация клавиш **Ctrl + C**;
- вставить. Пункт меню **Правка** -> **Вставить** или комбинация клавиш **Ctrl + V**;
- вырезать. Пункт меню **Правка** -> **Вырезать** или комбинация клавиш **Ctrl + X**;
- выделить всё. Пункт меню **Правка** -> **Выделить всё** или комбинация клавиш **Ctrl + A**;
- найти. Пункт меню **Правка** -> **Найти** или комбинация клавиш **Ctrl + F**;
- отменить. Пункт меню **Правка** -> **Отменить** или комбинация клавиш **Ctrl + Z**;
- вернуть. Пункт меню **Правка** -> **Вернуть** или комбинация клавиш **Ctrl + Y**;
- форматировать текст программы. Пункт меню **Инструменты** -> **АвтоФорматирование** или комбинация клавиш **Ctrl + T**. Эта команда позволяет исправить неточности в разметке программы и привести её в более читабельный вид.

Подключение библиотеки:

Давайте разберёмся, что же такое библиотека. **Библиотека** — это набор функций, предназначенных для того, чтобы максимально упростить работу с различными модулями, датчиками и т.д. Например, библиотека *LowPower* позволяет легко управлять режимами энергосбережения модулей Arduino. Существует огромное количество различных модулей и датчиков. Но как ими управлять? Для этого разработаны специальные библиотеки, которые значительно облегчают работу. Но перед тем, как

использовать дополнительные библиотеки, необходимо установить и подключить их. А как подключить библиотеку, спросите Вы? Есть несколько способов:

– через среду Arduino IDE. Для этого перейдите в меню **Скетч -> Подключить библиотеку**.

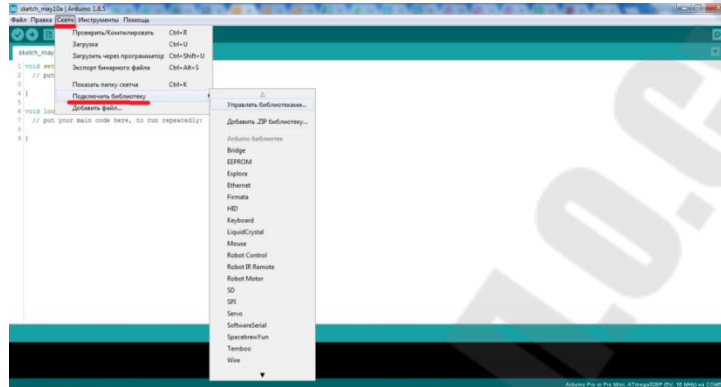


Рисунок 4 – Подключение библиотеки

Если нужная библиотека есть в списке, то необходимо просто кликнуть на неё мышкой и она автоматически подключится к Вашей программе. А что же делать, если нужной библиотеки нет в списке? В таком случае перейдите во вкладку **Скетч -> Подключить библиотеку -> Управление библиотеками**. Перед Вами откроется следующее окошко:

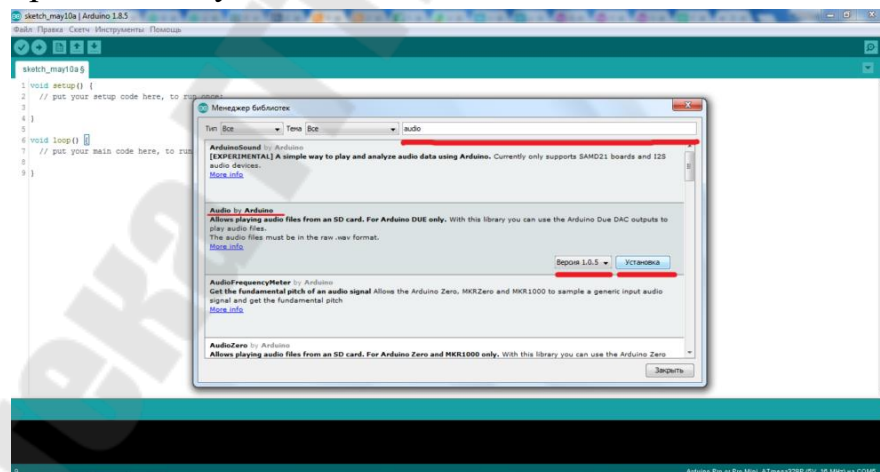


Рисунок 5 – Поиск библиотек

В правом верхнем углу необходимо ввести название нужно библиотеки. Далее, из списка надо выбрать нужную библиотеку, её версию и нажать кнопку **установить**. Обратите внимание, что есть возможность отсортировать результаты поиска (вкладки **Тип** и **Тема**). Теперь она появится в списке установленных библиотек и её можно подключить через меню **Скетч -> Подключить библиотеку**.

– Предварительно скачав нужную библиотеку её также можно подключить к своей программе. Если она находится в zip архиве, то её можно подключить следующим способом. Для этого нужно перейти в меню **Скетч -> Подключить библиотеку -> Подключить .ZIP библиотеку** и в открывшемся окошке выбрать zip архив с библиотекой. Теперь эта библиотека появится в списке установленных и её можно подключить через меню **Скетч -> Подключить библиотеку**. Если у Вас открыта Arduino IDE, то её нужно перезапустить, чтобы изменения были применены.

– Подключить библиотеку можно и без использования Arduino IDE. Для этого необходимо её скачать и скопировать папку с библиотекой по следующему пути: **X:\Пользователи\<Имя пользователя>\Документы\Arduino\libraries**.

Если у Вас открыта Arduino IDE, то её нужно перезапустить, чтобы изменения были применены. Теперь эта библиотека появится в списке установленных и её можно подключить через меню **Скетч -> Подключить библиотеку**. Однако можно и вручную в программе подключить библиотеку. Для этого перед функцией **setup()** нужно прописать следующую конструкцию **#include <имя заголовочного файла.h>** или **#include «имя заголовочного файла.h»**.

В каждой библиотеке есть различные примеры использования функционала библиотеки. Имя заголовочного файла можно найти там. Дорогой читатель, рекомендуем внимательно изучать примеры, ведь любая, даже большая программа, состоит из кусочков простых примеров.

Выбор платы:

Существует довольно большое количество модулей Arduino. Загружать написанную программу нужно именно в тот тип модуля, который подключён к компьютеру. Выбрать модуль можно в меню **Инструменты -> Плата**.

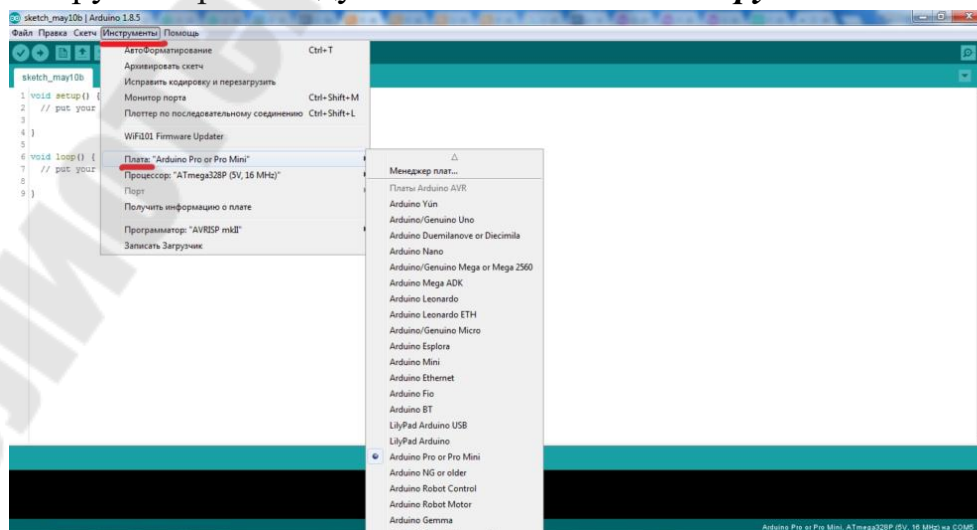


Рисунок 6 – Идентификация платы

В списке находятся все официальные версии модулей Arduino. Если нужного модуля нет в списке, то его можно добавить.

Далее необходимо выбрать тип контроллера, который установлен на модуле Arduino (на каждом контроллере есть маркировка). Это можно сделать в меню **Инструменты -> Процессор**.

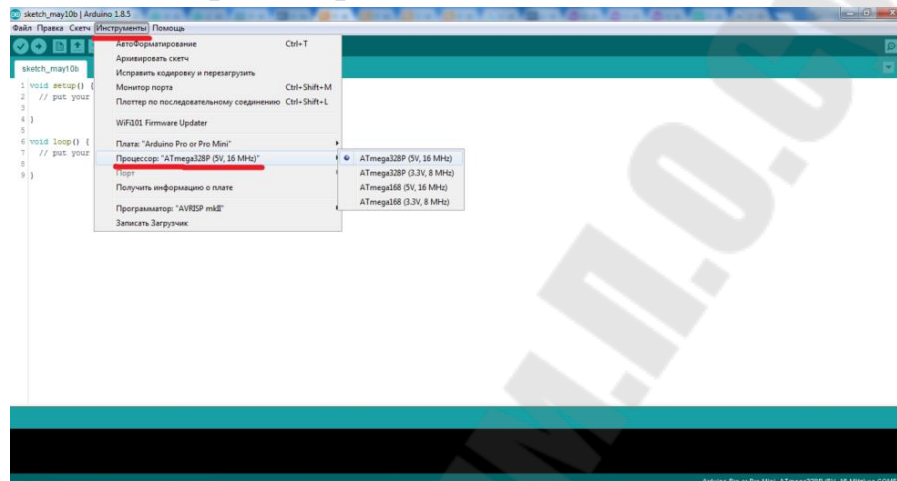


Рисунок 7 – Выбор процессора

Компиляция программы:

Теперь, когда выбран конкретный модуль Arduino, можно переходить к компиляции написанной программы. Давайте разберёмся, что же такое компиляция. Если говорить простым языком и касательно среды Arduino IDE, то компиляция — это перевод написанной в IDE программы в эквивалентную, но в машинных кодах. Программа записывается в микроконтроллер именно в машинных кодах, а не в том виде, в котором она написана в IDE. Компиляция также помогает найти ошибки в программе, т.к. компиляция не будет выполнена, если в программе есть ошибки.

С понятием компиляции мы разобрались. А как теперь открыть скомпилированную написанную программу? Для этого в Arduino IDE предусмотрено несколько способов:

- из пункта меню **Скетч -> Проверить/Компилировать**;
- при помощи комбинации клавиш **Ctrl + R**;
- из панели с часто используемыми командами.

После успешной компиляции будет выведено сообщение об этом. Также в области сообщений можно найти информацию о том, сколько памяти занимает написанная программа. Если в программе есть ошибки, тогда в области сообщений будет выведено сообщение с указанием конкретной строки и ошибки в ней.

Выбор программатора:

Пункт меню **Инструменты -> Программатор** используется для выбора аппаратного программатора, если программирование модуля или микроконтроллера осуществляется *не* при помощи встроенного USB-последовательного соединения. Как правило, эта команда используется довольно редко, однако может пригодиться, например, при записи загрузчика в новый микроконтроллер.

Т.к. в модулях Arduino уже есть свой встроенный программатор, то в качестве программатора в меню **Инструменты -> Программатор** нужно оставить стандартный **AVRISP mkII**.

Запись загрузчика:

При помощи команды **Инструменты -> Записать загрузчик** в микроконтроллер можно записать загрузчик. При использовании Arduino этого не требуется, однако эта команда может пригодиться если Вы хотите прошивать обычный микроконтроллер фирмы Atmel (именно микроконтроллеры фирмы Atmel стоят в модулях Arduino и как правило, они продаются без встроенного загрузчика) аналогично Arduino.

Загрузка программы:

Теперь, когда почти со всеми элементами среды Arduino IDE мы разобрались, можно приступить к финальному этапу — загрузке программы в модуль Arduino.

Прежде чем загружать программу, нужно выбрать порт, к которому подключён Ваш модуль Arduino. Выбрать его можно в меню **Инструменты -> Порт**.

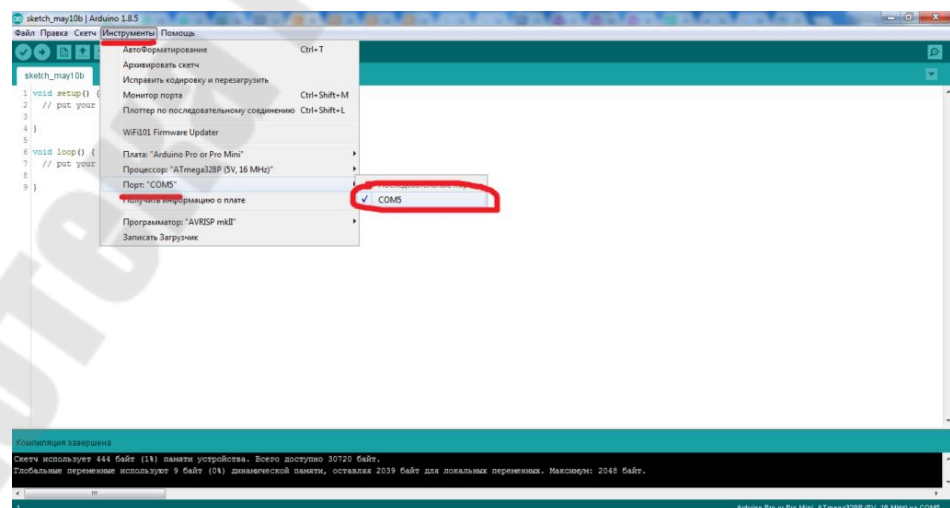


Рисунок 8 – Выбор порта

Это не обязательно должен быть COM5, как на рисунке. Имя порта у Вас, скорее всего, будет другим (COM3, COM 10 и т.д.).

Когда выбран соответствующий модуль Arduino, нужный порт, процессор и программатор, можно приступить к загрузке программы.

из пункта меню *Скетч* -> *Загрузка*;

- В процессе загрузки программы первым этапом код будет скомпилирован, а только потом, если не возникло ошибок в ходе компиляции, будет записан в модуль Arduino. При успешной загрузке программы в области сообщений появится сообщение об этом.

[illegible]

Между Arduino и компьютером мо

Открыть окно монитора последовательного порта можно несколькими способами:

- После открытия монитора последовательного порта появится следующее
ошибка:

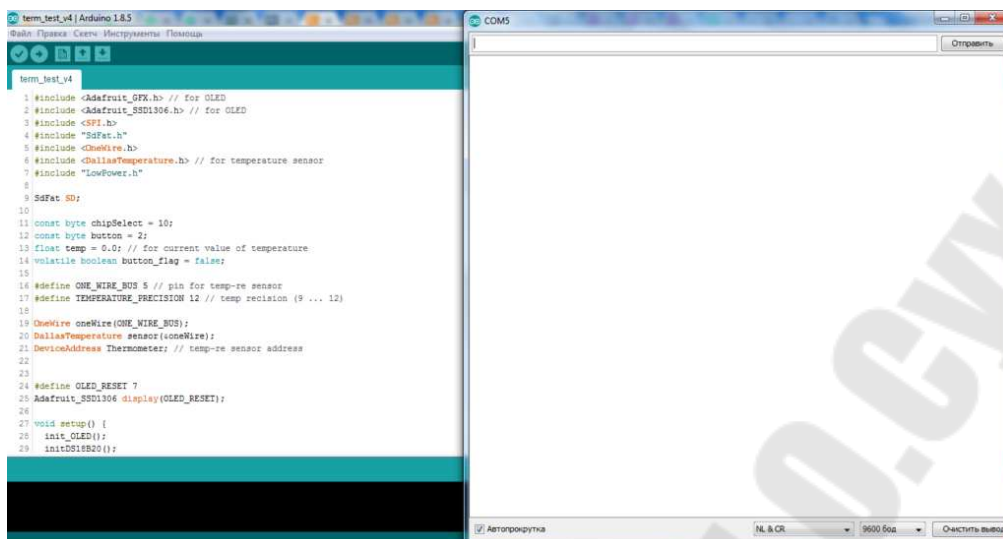


Рисунок 10 – Монитор последовательного порта

В самом низу этого окошка можно изменить скорость работы порта (*она должна совпадать с той, которая указана в программе!*) очистить окно и т.д.

Пример полного цикла разработки программы в Arduino IDE:

Начнём мы не со стандартного мигания светодиодом, а с классического примера из программирования — вывода на экран «Hello World!».

Запускаем Arduino IDE

Откроется следующее окошко:



Рисунок 11 – Стартовое окно программы

Ждём открытия среды. Когда она открылась, создаём новую программу путём нажатия на комбинацию клавиш **Ctrl + N**.

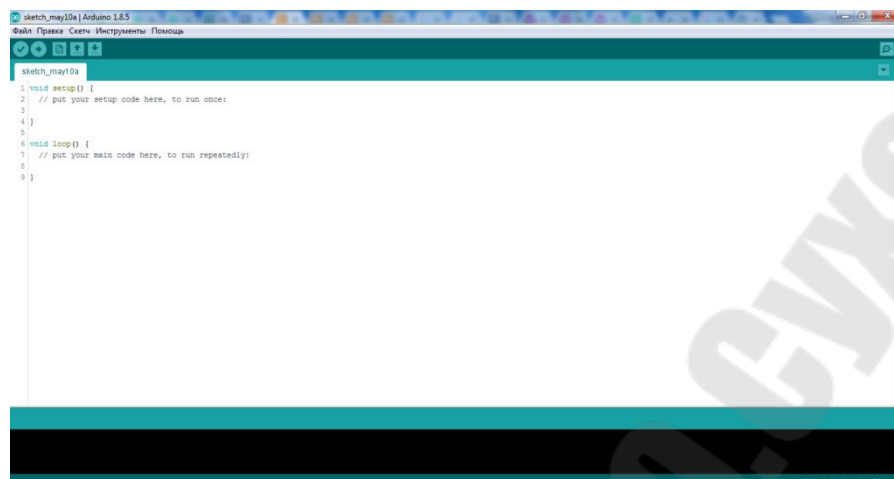


Рисунок 12 – Создание новой программы

Теперь давайте сохраним эту программу под именем `Hello_World`, для этого нажимаем на комбинацию клавиш **Ctrl + S**, вводим путь, куда сохраним программу и её имя.

В указанной директории появится папка с программой `Hello_World`.

Теперь давайте напишем сам код. Мы хотим с периодичностью в 5 секунд выводить сообщение `Hello World!` на экран, т.е. в монитор последовательного порта. Никаких сторонних библиотек для этого нам не понадобится. Скопируйте следующий код в IDE.

ТЕКСТ программы

```

/*-----*/
void setup() {
  Serial.begin(9600); // конфигурируем последовательный порт на
  скорость 9600 бод
}
/*-----*/
void loop() {
  Serial.println("Hello World!"); // выводим в каждый раз в новую строку
  Hello World!
  delay(5000); // ждём 5 секунд (1 с = 1000 мс)
}
/*-----*/

```

Комментарии к программе:

Функция [`setup\(\)`](#) запускается однократно при запуске программы. В ней инициализируется последовательный порт и его скорость (9600 бод). Функция [`loop\(\)`](#) вызывается после функции [`setup\(\)`](#). Функция представляет собой бесконечный цикл, в котором выполняется пользовательская программа. В ней при помощи функции `Serial.println(«Hello World!»)` мы каждый раз в

новую строку выводим сообщение *Hello World!* она порт. Функция *delay* (5000) устанавливает задержку при выполнении программы на 5000 мс, т.е. сообщение выводится раз в 5 секунд. Таким образом, мы выводим наше сообщение с периодичностью 5 секунд.

Теперь сохраните его при помощи комбинации клавиш **Ctrl + S**.

Когда программа написана, подключите модуль Arduino к компьютеру (в данном примере используется Arduino Nano).

В меню **Инструменты -> Плата** выберите ваш модуль Arduino (у нас — Arduino UNO).

Далее в меню **Инструменты -> Процессор** выберите процессор (микроконтроллер), который установлен на модуле Arduino. В нашем примере в модуле UNO стоит микроконтроллер Atmega328.

Теперь, проверьте, какой программатор установлен в меню **Инструменты -> Программатор**, он должен быть **AVRISP mkII**.

Давайте скомпилируем нашу программу и узнаем, сколько же она занимает места в памяти микроконтроллера. Для этого нажмём комбинацию клавиш **Ctrl + R**.

Наша программа занимает 5% памяти программ (флеш памяти) и 9% оперативной памяти.

Всё готово к загрузке программы. Для этого нажмите комбинацию клавиш **Ctrl + U** и наша программа загрузится в модуль. После успешной загрузки появится сообщение об этом.

Теперь давайте откроем монитор последовательного порта (в правом верхнем углу IDE)

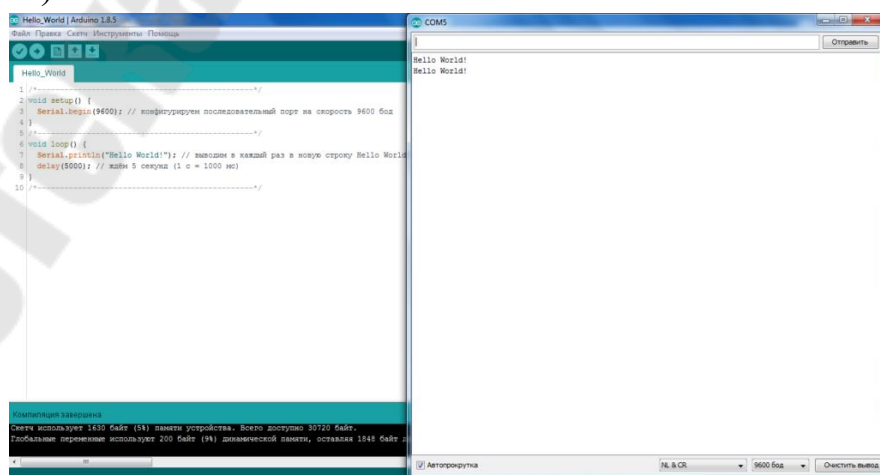


Рисунок 13 – Результаты работы программы

После загрузки программы в модуль Arduino, в монитор последовательного порта с периодом в 5 секунд будет выводиться сообщение Hello World!.

Практическая работа № 2

«Знакомство с составом набора LEGO MINDSTORMS, блоком EV3 и средой разработки»

(4 часа)

1 Цель работы

Ознакомление с составом набора Lego Mindstorms, изучение функциональных клавиш блока EV3, знакомство со средой разработки Lego Mindstorms.

2 Практическая часть

2.1 Изучить документацию по контроллеру Lego EV3;

Микрокомпьютер EV3

Микрокомпьютер EV3 начали выпускать в 2013 году. Он пришел на смену микрокомпьютеру NXT и является более современной моделью.

Микрокомпьютер EV3 часто называют микроконтроллер EV3, блок управления EV3, модуль EV3, интеллектуальный блок. Также часто можно встретить в литературе название «Р — кирпич» или «Р — brick».

Это название является сокращением от Programmable brick — то есть программируемый блок (кирпич).



Рисунок 1 – Внешний вид контроллера EV3

Микрокомпьютер EV3 входит в состав робототехнических конструкторов Lego Mindstorms Education EV3 образовательной и домашней версии. Микроконтроллер EV3 является компьютерным мозгом робота. Операционной системой микрокомпьютера является Linux. Датчики опрашиваются со скоростью в 1 Гигагерц.

Интерфейс управления микрокомпьютера EV3

В модуле EV3 реализован шестикнопочный интерфейс управления. Этот интерфейс имеет программируемую визуальную индикацию состояний — подсветка различных цветов.

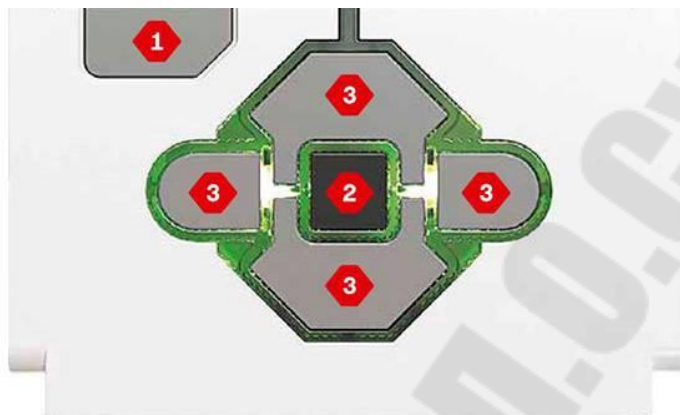


Рисунок 2 – Кнопочный интерфейс EV3

Ход выполнения программы может контролироваться при помощи подсветки модуля. В этом случае подсветка выполняет роль индикатора состояния контроллера EV3. Индикатор может приобретать зеленый, оранжевый или красный цвет. Также индикатор может мигать. Индикатор состояния модуля имеет следующие коды:

- красный цвет означает запуск. Также он сигнализирует об обновлении или показывает завершение работы;
- если красный цвет мигает, то микроконтроллер занят;
- оранжевый цвет — это оповещение, блок работает;
- когда зеленый цвет пульсирует это значит, что происходит выполнение программы.

Для лучшего понимания на каком этапе находится программа можно отдельно программировать индикатор блока EV3. Тогда при выполнении различных условий индикатор сможет гореть различными цветами и пульсировать.

Порты модуля EV3

С одной стороны, блок EV3 оснащен четырьмя портами входа для подключения различных датчиков.



Рисунок 3 – Порты модуля EV3

Также сбоку есть дополнительный USB — порт, который можно использовать для чтения USB-flash накопителей, подключения Wi-Fi приемника Wi-Fi. Адаптер нужно приобретать отдельно. Flash память адаптера 16 Мб. При помощи этого порта можно создать последовательную цепь микрокомпьютеров EV3. Всего можно последовательно соединить до четырех программируемых блоков управления EV3.

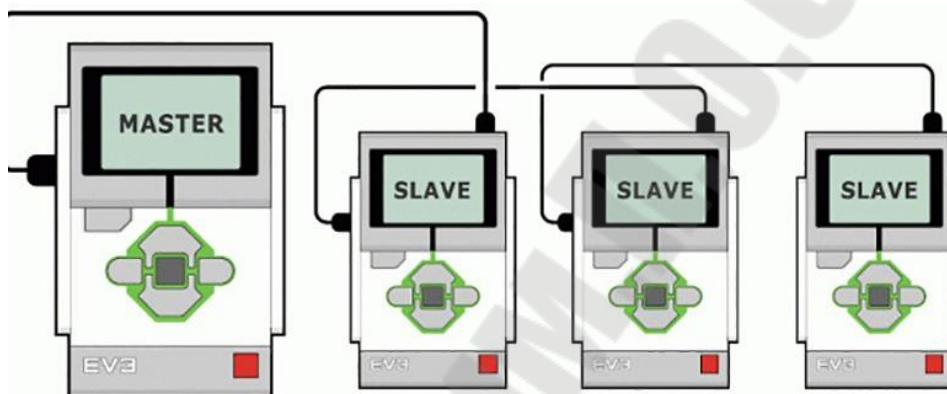


Рисунок 4 – Соединение модулей EV3

Порты входа обозначаются цифрами от одного до четырех. На другой стороне модуля есть четыре порта выхода, куда происходит подключение моторов. Это позволяет создавать сложные и функциональные модели роботов. Порты выхода обозначаются буквами А, В, С, D. Также со стороны портов выхода имеется мини – USB порт. Он позволяет при помощи USB кабеля соединять модуль EV3 с компьютером.

С этой же стороны блока есть слот для чтения карт формата микро-USB. Этот слот позволяет увеличить объем памяти доступной для хранения данных микрокомпьютера EV3 до 32 гигабайт. SD карта в комплект не входит и покупается отдельно.

Динамик EV3

С другой стороны блока расположен высококачественный встроенный динамик. Этот динамик может воспроизводить любые звуки программируемого блока EV3. Качество звука всегда будет хорошим если динамик микроконтроллера не будет закрыт.

Динамик позволяет эффективно использовать звуковые функции микроконтроллера EV3. Благодаря дополнительному объему памяти и улучшенным характеристикам встроенного динамика увеличились возможности использования звука и вышли на новый качественный уровень.

Программирование с помощью интерфейса микрокомпьютера EV3

В микрокомпьютере есть возможность программирования при помощи самого микрокомпьютера. Есть двенадцать программируемых блоков при помощи которых можно создавать программы.

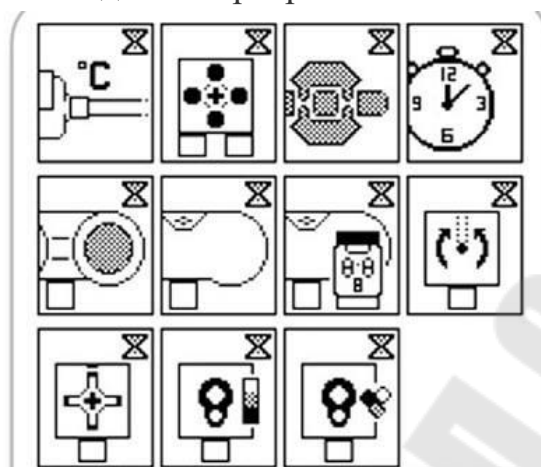


Рисунок 5 – Программирование на модуле EV3

Можно писать сложные программы без использования компьютера. Эту программу можно экспортировать на компьютер. На компьютере программу можно улучшить при помощи программного обеспечения EV3. После этого ее снова можно загрузить на микрокомпьютер и проверить на практике. Это является очень полезной функцией.

Как соединить микроконтроллер EV3 с компьютером

Всего существует три способа установить соединение микроконтроллера с компьютером:

1. Через USB – кабель;
2. С помощью встроенного модуля Bluetooth;
3. Через Wi — Fi приемник.

Установка и зарядка батарей блока EV3

На нижней стороне модуля находится литиево — ионный аккумулятор. Конструкция аккумулятора выполнена таким образом, что позволяет плотно устанавливать блок и детали модели робота. Когда батарея используется в первый раз или в случае полностью разряженной батареи нужно в течении как минимум двадцати минут произвести зарядку батареи.

Производить зарядку батареи можно, не вынимая ее из собранной модели. Это значит, что не нужно тратить время на разборку и сборку робота, чтобы заменить аккумулятор.

Батарея заряжается при помощи стандартного адаптера Lego EV3, который входит в комплект. Когда батарея разряжена, то при включении в розетку через адаптер индикатор начинает гореть красным светом. После завершения зарядки индикатор начинает гореть зеленым цветом, а индикатор красного цвета гаснет.

При использовании программируемого микроконтроллера EV3 во время зарядки аккумулятора время зарядки увеличивается. Для зарядки аккумулятора нужно три – четыре часа. Когда аккумуляторная батарея используется в первый раз, рекомендуется, чтобы она имела полную зарядку.

При этом модуль при использовании аккумулятора работает дольше, чем при использовании батареек. Чтобы батарея дольше сохраняла работоспособность есть несколько простых правил:

- если батарея не используется, то ее лучше на время вынуть;
- каждый комплект батарей должен храниться в отдельном контейнере для совместного использования;
- громкость динамик лучше уменьшить;
- произвести правильные настройки параметров спящего режима;
- если вы не используете Bluetooth и Wi-Fi, то их нужно отключить;
- стараться избегать ненужного износа моторов;

Технические характеристики микрокомпьютера EV3

Микрокомпьютер Lego EV3 имеет следующие характеристики:

- Процессор — ARM9
- Микрокомпьютер EV3 имеет операционную систему Linux
- Контроллер блока программирования — ARM9 с частотой 300 МГц
- Объем flash-памяти составляет 16 МБ
- Объем оперативная память — 64 МБ
- Модуль EV3 имеет монохромный черно-белый экран с разрешением 178×128 пикселей
- При осуществлении связи с главным компьютером через шину USB 2.0 скорость передачи данных достигает 480 Мбит/с
- При осуществлении связи с главным компьютером через шину USB 1.1 скорость передачи данных до 12 Мбит/с
- Карта памяти типа Micro SD с поддержкой SDHC, версии 2.0. Максимальный объем памяти карты 32 ГБ
- По 4 входных и выходных порты для подключения моторов и датчиков
- Соединение осуществляется при помощи кабелей с коннекторами типа RJ12
- Поддерживается автоматическая идентификация сенсоров и моторов
- Электропитание от 6 батареек типа AA (батарейки можно заменить на литиевые аккумуляторы) или от аккумуляторной батареи

EV3 является третьей версией конструктора Lego Mindstorms Education. Вторая версия носила название NXT, первая версия – RCX.

Как включить микрокомпьютер EV3

Для включения микрокомпьютера EV3 нужно нажать на кнопку, находящуюся в центре шестикнопочного интерфейса микрокомпьютера EV3. После нажатия индикатор статуса модуля загорится красным цветом.

При этом будет видно окно запуска. После того, как индикатор загорится зеленым цветом, модуль EV3 будет готов к работе. Чтобы выключить модуль EV3 нужно нажимать кнопку «Назад» до тех пор, пока не будет видно окно завершения работы.

По умолчанию будет выбран вариант «Прервать X». При помощи кнопки «Вправо» нужно выбрать вариант «Принять». После этого нажмите центральную кнопку (OK). Теперь микрокомпьютер EV3 выключен. Если нажать OK, то при выбранном варианте X, произойдет возврат в окно «Запустить последнюю».

2.2 Собрать робота-тележку:

Получить у преподавателя инструкцию и комплект деталей, и собрать первого робота-тележку.

2.3 Знакомство со средой Lego Mindstorms

Первым делом загружаем среду программирования Lego mindstorms EV3. В главном меню программы выбираем: "Файл" - "Новый проект" или нажимаем "+", показанный на рисунке стрелкой.

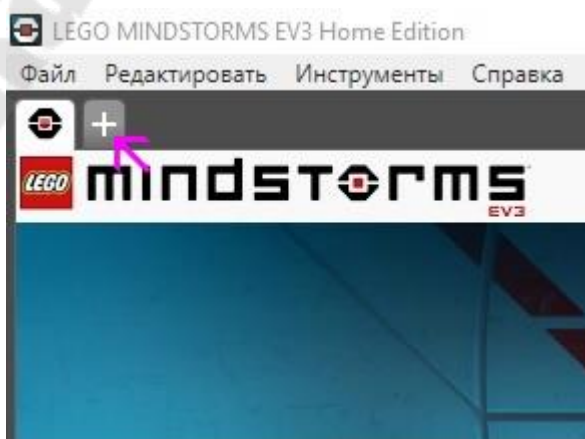


Рисунок 6 – Создание нового проекта

В одном проекте может находиться множество программ. Для того, чтобы проект корректно загружался в нашего робота необходимо в названии проекта и программ использовать только буквы латинского алфавита! Давайте назовем

наш проект lessons (уроки), а первую программу - lesson-1 (урок-1). Для того, чтобы дать название проекту, воспользуемся главным меню программы: "Файл" - "Сохранить проект как..." Чтобы изменить название программы - следует сделать двойной щелчок мышью на её названии (program) и вписать свое название.

Включим центральный блок нашего робота. Для этого нажмем на центральную (самую темную) кнопку блока. С помощью USB-кабеля, идущего в комплекте с конструктором, подключим робота к компьютеру. Успешное подключение робота отразится на вкладке аппаратных средств программного обеспечения EV3 в правом нижнем углу программы.



Рисунок 7 – Индикатор подключения блока EV3

Если подключение робота прошло успешно, то приступим к программированию и создадим нашу первую программу.

Научим нашего робота двигаться вперед на определенное расстояние. В нижней части экрана находится палитра программирования, каждому цвету палитры соответствуют различные группы программных блоков. Выберем зеленую палитру "Действие". Она содержит блоки управления моторами, блок вывода информации на экран, блок управления звуком и кнопками контроллера EV3 (главного блока). Выберем блок "Рулевое управление" и перетащим его в область программирования (центральная область программы).

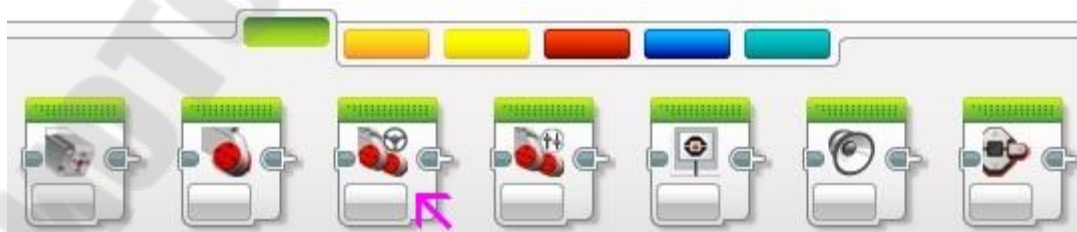


Рисунок 8 – Палитры программирования

Каждая программа состоит из цепочки блоков, задающих определенное действие или проверяющих различные условия. Каждый блок имеет множество различных параметров. Первый, оранжевый блок с зеленым треугольником внутри называется - "Начало". Именно с него начинается любая программа для

нашего робота. Второй блок установили мы. Повторю - он называется "Рулевое управление". Его назначение - одновременное управление двумя моторами.



Рисунок 9 – Блок «Рулевое управление»

Но, если вы собирали робота по инструкции, предложенной выше, то, наверное, обратили внимание, что в ней отсутствует схема подключения моторов и датчиков. Настало время с этим разобраться. Блок EV3 имеет 4 порта, обозначенных цифрами: 1, 2, 3, и 4. Эти порты служат для подключения только датчиков. Для подключения моторов служат порты, обозначенные буквами: A, B, C и D. Можно подключать моторы в любые свободные порты, предназначенные для них. Но в случае управляемой тележки рекомендовано подключать моторы в порты: B и C. Давайте сейчас возьмем два соединительных кабеля длиной 25 см, левый мотор подключим к порту B, а правый - к порту C. Именно это подключение выбрано по умолчанию в блоке "Рулевое управление". Специальная кнопка, обозначенная стрелкой, отвечает за режим работы блока. Для первой программы выберем режим: "Включить на количество оборотов". Значение 0 под черной стрелочкой на блоке означает прямолинейное движение, когда оба мотора крутятся с одинаковой скоростью. Число 75 задает мощность моторов, чем больше это значение, тем быстрее поедет наш робот. Цифра 2 задает количество оборотов каждого из моторов, на которое они должны повернуться.

Итак, наша первая программа готова. Загружаем ее в нашего робота. Для этого нажимаем кнопку "Загрузить" на вкладке аппаратных средств и отсоединяем USB-кабель от робота.

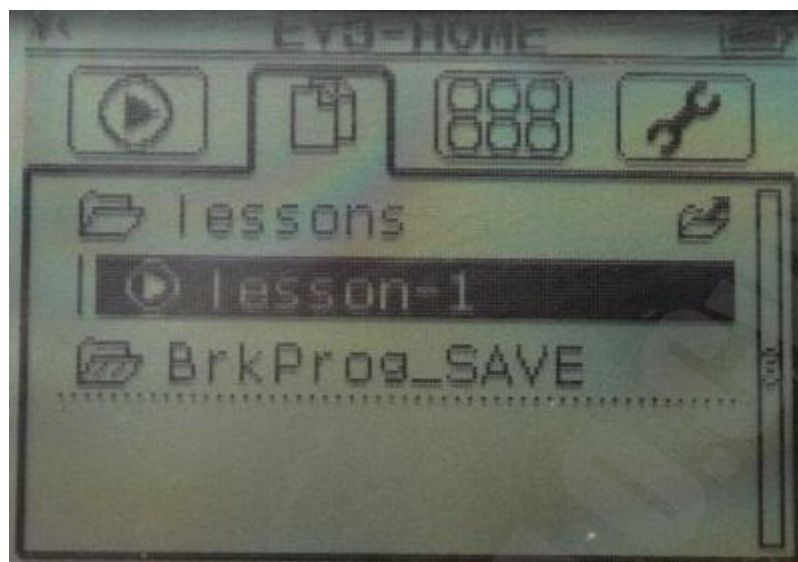


Рисунок 10 – Запуск программы на выполнение

Устанавливаем робота на ровную поверхность. С помощью стрелок на блоке EV3 заходим в папку нашего проекта, выбираем программу lesson-1 и центральной кнопкой блока EV3 запускаем ее на выполнение.

Практическая работа № 3

Программирование роботов из стандартных наборов на Lego EV3.

(4 часа)

1 Цель работы

Получить навыки сборки роботов на базе Lego EV3 в соответствии с инструкцией.

2 Практическое задание:

2.1 Изучить на практическом занятии техническую документацию по роботу в соответствии с вариантом (робот-змея, робот=воин, робот-скорпион)

2.2 Собрать робота (задание выдается преподавателем).

2.3 Загрузить стандартный для данного набора пакет программ, убедиться в правильности работы, модифицировать функционал робота в соответствии с заданием преподавателя.

2.4 Продемонстрировать результаты

Практическая работа № 4

Программирование роботов из стандартных наборов на Arduino.

(4 часа)

1 Цель работы

Получить навыки программирования роботов на базе Arduino в соответствии с инструкцией.

2 Практическое задание:

2.1 Изучить на практическом занятии техническую документацию по роботу в соответствии с вариантом (Sparky, Hexy, PopBot).

2.2 Познакомиться с соответствующей роботу средой разработки программ IDE.

2.3 Загрузить стандартный для данного набора пакет программ, убедиться в правильности работы, модифицировать функционал робота в соответствии с заданием преподавателя.

2.4 Продемонстрировать результаты

Практическая работа № 5

Установка операционной системы на одноплатный компьютер

(4 часа)

1 Цель работы:

Научиться разворачивать операционную систему на одноплатном компьютере и подключать к нему периферийное оборудование.

2 Практическое задание:

2.1 Получить от преподавателя комплект одноплатного компьютера (Raspberry PI, Orange PI, Banana PI, Asus ThinkerBoard).

2.2 Изучить техническую документацию соответствующей модели одноплатного ЭВМ (Raspberry PI, Orange PI, Banana PI, Asus ThinkerBoard).

2.3 В соответствие с вариантом подготовить образ и развернуть операционную систему на соответствующем одноплатном компьютере (Raspberry PI, Orange PI, Banana PI, Asus ThinkerBoard)

2.4 Изучить в технической документации к одноплатному компьютеру назначение выходов порта GPIO.

2.5 Продемонстрировать результаты (рабочая машина) преподавателю.