

Д.И. Муромцев, В.Н. Шматков

**Интернет Вещей: Введение
в программирование на Arduino**



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО

Д.И. Муромцев, В.Н. Шматков

Интернет Вещей: Введение в программирование на Arduino

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
09.03.04 Программная инженерия в качестве учебного пособия для реализации
основных профессиональных образовательных программ высшего
образования бакалавриата

 **УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2018

Муромцев Д.И., Шматков В.Н. «Интернет Вещей: Введение в программирование на arduino» – СПб: Университет ИТМО, 2018. – 36 с.

Рецензенты: Лаздин А.В., к.т.н, доцент кафедры информатики и прикладной математики Университета ИТМО, начальник студенческого вычислительного центра.

В учебно-методическом пособии приводится базовая теория и примеры различных микроконтроллеров, датчиков и сенсоров. Рассматривается микрокомпьютер и операционные системы для работы и настройки лабораторной установки. Рассмотрены основные методы коммуникаций между устройствами в рамках занятий, посвященных направлению интернета вещей (Internet of Things, IoT). Приведён пример реализации проекта на базе лабораторной установки.



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2018

©Муромцев Д.И., Шматков В.Н., 2018

Оглавление

Введение.....	4
1. Краткая информация о лабораторной установке.....	5
1.1 Что такое Arduino	5
1.1.1 Семейство Arduino	6
1.1.2 Датчик, актуаторы, модули	8
1.2. Микрокомпьютер Raspberry Pi.....	11
1.2.1 Raspberry Pi	11
1.2.2 Операционная система и программное обеспечение	12
2. Интернет вещей. Архитектуры и методы коммуникаций	13
2.1 Интернет вещей	13
2.2 Архитектура интернета вещей	14
2.3 Технологии коммуникации между устройствами.....	16
3. Примеры реализации	18
3.1 Лабораторная установка	18
3.2 Пример проекта «Умный музей»	18
Лабораторная работа 1: «Первая программа на Arduino».....	20
Лабораторная работа 2: «Подключение кнопки управления светодиодом к Arduino».....	23
Лабораторная работа 3: «Получение данных с датчиков Arduino.»	29
Литература	32

Введение

В учебно-методическом пособии приводятся базовая теория и примеры различных микроконтроллеров, датчиков и сенсоров. Рассматривается микрокомпьютер и операционные системы для работы и настройки лабораторной установки. Рассмотрены основные методы коммуникаций между устройствами в рамках занятий, посвященных направлению интернета вещей (Internet of Things, IoT). Приведён пример реализации проекта на базе лабораторной установки.

В пособии представлены три лабораторные работы начального уровня. В качестве практической задачи рассмотрен проект «Умный музей», основная цель проекта создать интеллектуальную систему контроля освещения и температуры экспонатов, собирать информацию для анализа и автоматического прогнозирования потребностей посетителей музея в режиме реального времени.

1. Краткая информация о лабораторной установке

1.1 Что такое Arduino

Arduino – общее название аппаратно-программных средств для построения систем автоматики и робототехники, а также решений в области Интернета вещей. Отличительной особенностью данной платформы является простота программирования, подробная документация и реализация аппаратной платформы в различных вариациях.

Для программирования микроконтроллеров используется Arduino IDE (Рис. 1) с поддержкой множества языков программирования, но фактически все программы пишутся на C/C++, а компилируются и собираются с помощью avr-gcc.

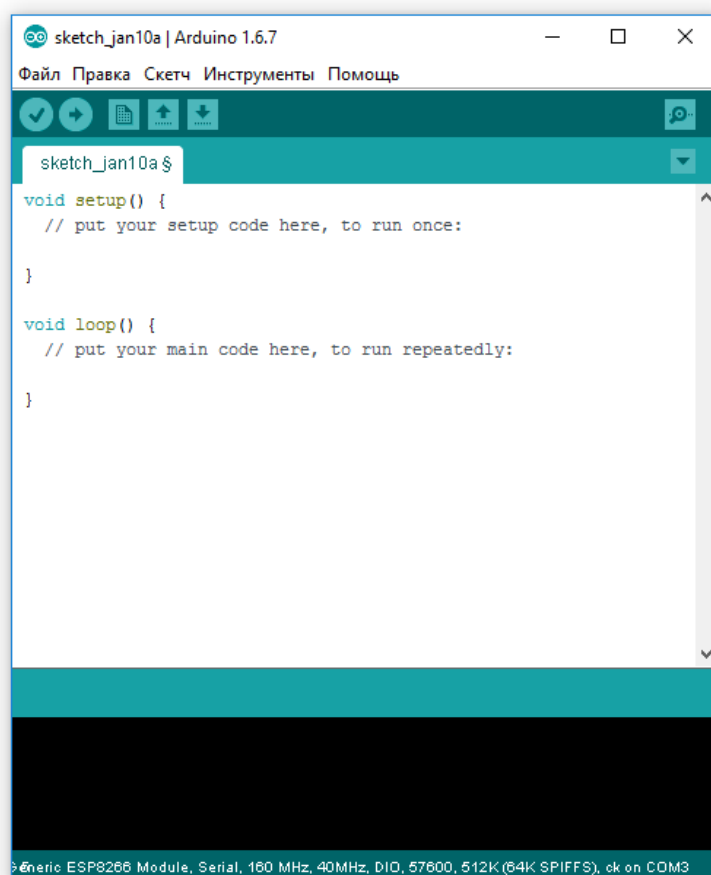


Рисунок 1 – Среда разработки Arduino IDE

В Arduino IDE по умолчанию уже содержатся примеры различных программ (Sketch) для проверки работоспособности микроконтроллера или быстрого создания своей программы на базе существующей, файлы с программой сохраняются с расширением «.ino».

Простейшая программа состоит из двух функций:

- `setup()`: функция вызывается однократно при старте микроконтроллера.
- `loop()`: функция вызывается после `setup ()` в бесконечном цикле в течение всего времени работы микроконтроллера.

Стоит отметить, что существуют неофициальные, созданные сторонними производителями, различные плагины к различным средам разработки, например: Eclipse plugin AVR-eclipse, Visualmicro для Microsoft Visual Studio и др.

1.1.1 Семейство Arduino

Существует много разновидностей микроконтроллеров, это связано с открытой лицензией оригинальной версии Arduino, единственное условие которой – не использовать официальное название Arduino в своих микроконтроллерах без разрешения, чтобы не запутать конечного потребителя в множестве существующих микроконтроллеров.

Официально существует несколько видов микроконтроллеров, отличающихся между собой размером и производительностью.

Arduino Mini — представляет собой небольшую плату, первоначально основанную на ATmega168, но теперь поддерживается ATmega328. Предназначен для использования на макетах и уже готовых продуктах. Он имеет 14 цифровых входных / выходных пинов (из которых 6 могут использоваться как выходы PWM), 8 аналоговых входов и 16 МГц кварцевый генератор. Его можно запрограммировать с помощью USB Serial адаптера или другого последовательного адаптера USB или RS232-TTL (Рис. 2).

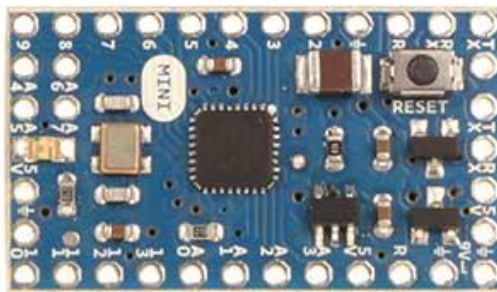


Рисунок 2 – Плата Arduino Mini

Arduino Nano — компактная плата, следующая по размеру после Mini. Основана на ATmega328P. Работает с кабелем USB-mini. Имеет 22 цифровых входа / выхода (из которых 6 могут использоваться как выходы PWM), а также 8 аналоговых входов (Рис. 3).

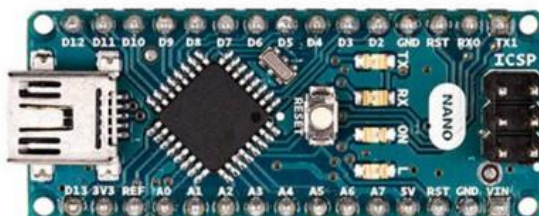


Рисунок 3 – Плата Arduino Nano

Arduino Uno — средняя по размеру плата, основана на ATmega328P. Имеет отдельный разъем питания и полноценный USB AB порт, выбор питания происходит автоматически. На плате находятся 14 цифровых входов / выходов (из которых 6 могут использоваться как выходы PWM), 6 аналоговых входов. Считается наиболее подходящей платой для знакомства с Arduino (Рис. 4).

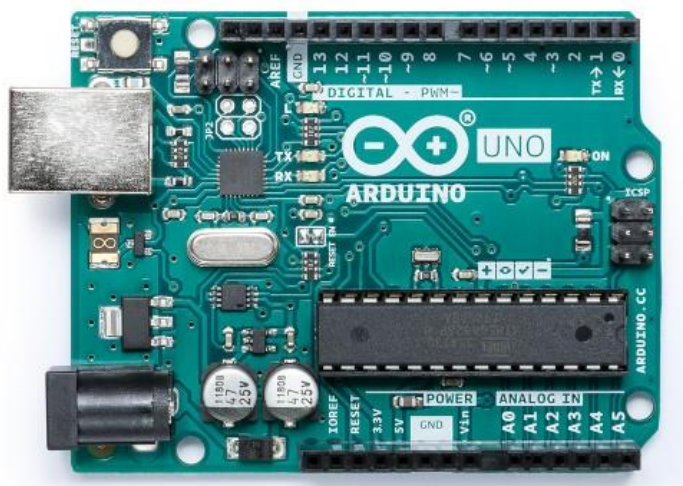


Рисунок 4 – Плата Arduino Uno

Arduino Mega 2560 Rev3 — самая большая плата, которая позволяет создать множество сложных проектов. Плата оснащена 54 цифровыми

входами / выходами, 16 аналоговыми входами и достаточно большим объемом памяти. Имеет полноценный USB AB порт, отдельный разъем питания, выбор питания происходит автоматически. Данная плата рекомендована для создания 3D принтеров, роботов или других сложных проектов (Рис. 5).



Рисунок 5 – Плата Arduino MEGA 2560

1.1.2 Датчик, актуаторы, модули

Для Arduino и множества других платформ производится большое количество различных датчиков, сенсоров и различных модулей расширяющих функционал устройства. Условно их можно разделить на три группы: датчики, актуаторы и модули.

Датчик — средство измерений, предназначенное для выработки сигнала измерительной информации в форме, удобной для передачи, дальнейшего преобразования, обработки и (или) хранения, но не поддающейся непосредственному восприятию наблюдателем [1].

Другими словами, датчики (сенсоры) позволяют получать информацию об окружающей среде и передавать её в удобном виде. Примером датчика может служить обыкновенный электронный термометр DS18B20 или LM35. Так как датчиков существует очень много, то выбор конкретного датчика зависит от решаемой задачи.

Продолжим рассмотрение на примере датчика температуры. Каждый датчик занимает порты подключения на плате, это не критично если мы работаем с большими платами наподобие Arduino Uno или MEGA,

но если предпочтительно использовать плату меньшего размера, такую как Arduino Nano или Mini, обладающую существенно меньшим количеством портов подключения, то необходимо рассмотреть более продвинутые (и дорогие) датчики, совмещающие в себе несколько датчиков. Примером совмещенного датчика может служить датчик GY-68 (Рис. 6) (датчик атмосферного давления и температуры BMP180). Хотя данный датчик не показывает высокую точность измерения, однако демонстрирует компактность решения задачи. Дорогие датчики имеют высококачественные компоненты, позволяющие получать более точные измерения.



Рисунок 6 – Датчик GY-68 (Датчик атмосферного давления и температуры BMP180)

Стоит отметить, что к датчикам относятся различные устройства для измерения света, движения, тепловой сигнатуры или различного рода сигналов и т.д. В совокупности данные устройства формируют группу датчиков.

Актуатор — устройство автоматического управления или регулирования, воздействующее на процесс в соответствии с получаемой командной информацией. Состоит из двух функциональных блоков: исполнительного устройства (если исполнительное устройство механическое, то его часто называют исполнительным механизмом) и регулирующего органа, например, регулирующего клапана, и может оснащаться дополнительными блоками [2].

Это устройство позволяет совершать какое-либо действие при определенном сигнале. Примером такого устройства могут служить различные сервоприводы, например, сервопривод MG995 (Рис. 7).



Рисунок 7 – Сервопривод MG995

Модуль — достаточно обширная группа, позволяющая расширить функционал плат Arduino. К ним относятся не только внешние модули, которые можно подключить к имеющимся разъемам, но и внутренние, такие как микроконтроллер, ОЗУ, флэшпамять и т.д.

К внешним подключаемым устройствам можно отнести модули связи: WiFi (ESP8266 (Рис.8)), Bluetooth (HC-06), LoRa/LoRaWAN (RN2483), GPS и т.д.

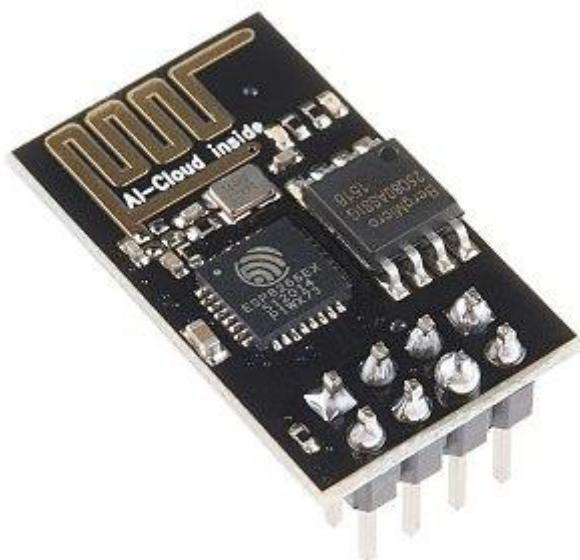


Рисунок 8 – WiFi модуль ESP8266

К внутренним относятся все модули, размещенные на плате в стандартном исполнении Arduino, но благодаря открытой лицензии можно разработать свою плату под необходимые характеристики. Конечно такая плата не будет называться Arduino (так как это название торговой марки).

1.2. Микрокомпьютер Raspberry Pi

1.2.1 Raspberry Pi

Микрокомпьютер — термин, обозначающий компьютер, выполненный на основе микропроцессора [3]. На данный момент существует множество исполнений микрокомпьютеров, но в рамках данной работы рассматривается Raspberry Pi.

Raspberry Pi – одноплатный компьютер размером с кредитную карту, изначально разрабатывался для обучения информатике, но впоследствии получил широкое применение в различных областях.

Архитектура платы Raspberry Pi 3 b+ (Рис. 9) похожа на Arduino, но имеет более мощный 64 битный процессор с частотой 1.4GHz, полноценный модуль WiFi и Lan, Bluetooth 4.2/BLE, а также полноценные два USB порта, выход HDMI и аудиовыход. То есть перед нами полноценный микрокомпьютер со всеми необходимыми для полноценной работы выходами.

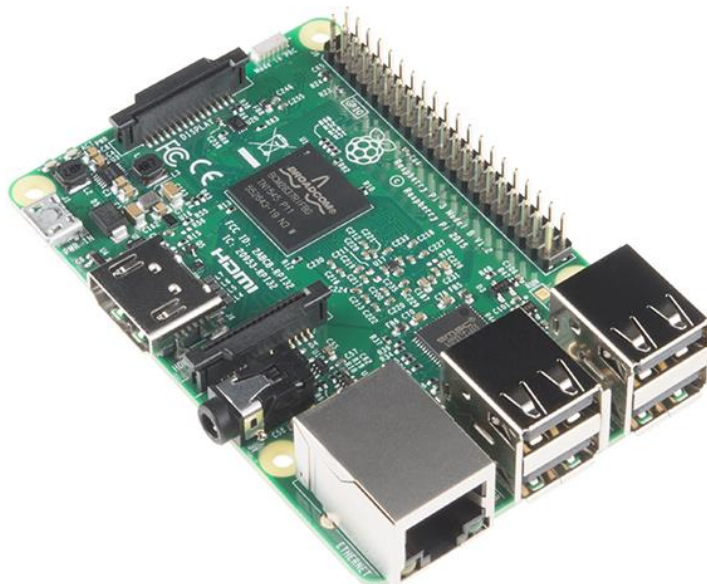


Рисунок 9 – Плата Raspberry Pi 3 b+

Отличительной особенностью Raspberry Pi от других микрокомпьютеров является наличие пинов для подключения внешних модулей и датчиков, как на Arduino. Хотя функционал микрокомпьютера более широкий, к нему можно подключить дисплей или использовать в качестве сервера или управляющего устройства. На базе микрокомпьютера «строят» «умные системы» такие как «SmartHome».

Чертежи Raspberry Pi распространяются по открытой лицензии и любой может собрать свой микрокомпьютер по уже имеющимся чертежам или купить модифицированную модель под собственные нужды. На рынке достаточно моделей микрокомпьютеров, например, с большим количеством ОЗУ или размещенной на плате флэш-памятью, более мощным процессором по сравнению с Raspberry Pi.

1.2.2 Операционная система и программное обеспечение

Для микрокомпьютеров существует множество вариантов операционных систем на базе Linux и одна операционная система, разработанная компанией Microsoft, специально для микрокомпьютеров, Windows 10 Internet of Things.

Таким образом имеются два варианта работы с микрокомпьютером, с использованием бесплатных решений на базе Linux или платных от Microsoft. В рамках данной работы рассмотрим систему на базе Linux.

На сайте Raspberry Pi официально поддерживается два дистрибутива Linux на базе Debian:

1. Noobs — рассчитана на начинающих пользователей. Имеет все необходимые базовые программы для начала программирования (в том числе Arduino) и множество предустановленных программ для комфортной работы с компьютером, в том числе и браузер.
2. RASPBIAN — операционная система для опытных пользователей созданная и оптимизированная специально для Raspberry Pi, имеет базовый набор программ и утилит.

Сторонние разработчики выпускают свои дистрибутивы, различающиеся включенными программами и пакетами, а также используемыми дистрибутивами, например:

- Ubuntu MATE for the Raspberry Pi 2 and Raspberry Pi 3 — имеет полностью настроенную оболочку, предустановленные драйвера и программы. Дистрибутив основан на Ubuntu.
- OSMC (open source media center) — дистрибутив для построения на базе Raspberry Pi домашнего мультимедийного центра.

- PiNet — централизованный центр для работы в классе (лаборатории), построен так, чтобы хранить всю информацию о пользователях и их системе во время обучения программированию на микроконтроллерах.

Рассмотренные операционные системы демонстрируют широкую область применения микрокомпьютеров и гибкость их настройки под различные задачи.

2. Интернет вещей. Архитектуры и методы коммуникаций

2.1 Интернет вещей

Интернет вещей — это, согласно определению Роба Ван Краненбурга, «концепция пространства, в котором все из аналогового и цифрового миров может быть совмещено. Это переопределит наши отношения с объектами, а также свойства и суть самих объектов».

Другими словами, это не просто множество сенсоров, датчиков и приборов, объединённых в одну сеть. Главная особенность данной сети — это однозначное идентифицирование каждого объекта, а также более тесная интеграция реального и «виртуального» миров, где взаимодействие происходит между людьми и устройствами.

Развитие «умных вещей» по мнению Роба Ван Краненбурга состоит из четырех уровней:

1 уровень: идентификация каждого объекта в сети.

2 уровень: определение и обслуживание потребностей потребителя.

3 уровень: Глобальная урбанизация и автоматизация систем города.

4 уровень: Внедрение интернета вещей достигает планетарных масштабов, появление концепции «Сенсорная планета».

На данный момент концепция развития интернета вещей претерпевает множественные изменения, уровней становится больше, дробление решаемых проблем становится узкоспециализированным. Это связано с развитием новых стандартов, более четким пониманием сути решаемых проблем, а также с развитием технологий в целом, как, например: появлением новых технологий передачи энергии, развитие носимых вещей и т.д. Все это расширяет сферы применение интернет вещей в различных областях (Рис. 10).

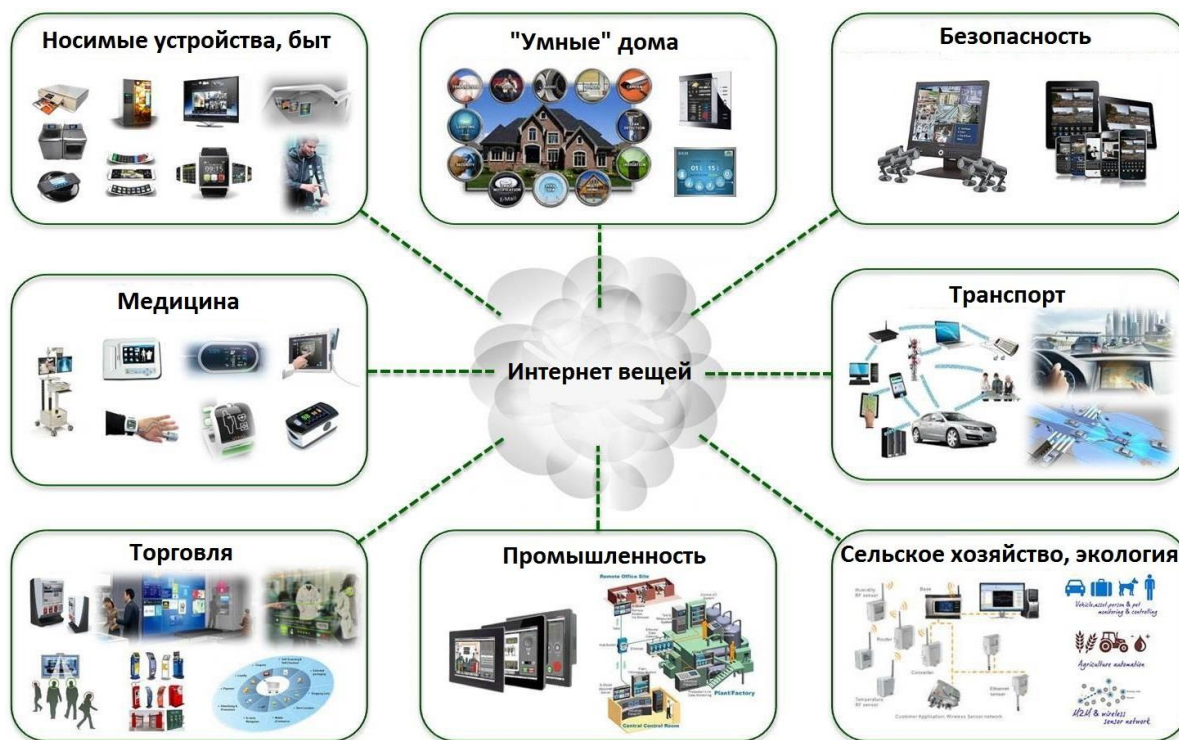


Рисунок 10 – Пример применение Интернета вещей в различных сферах

2.2 Архитектура интернета вещей

Стандартизация — это важный момент любой области. На данный момент существует очень много видов архитектурных решений, но типовой (основной) не существует, так как первые попытки стандартизировать данную область были предприняты всего 4 года назад, и данный процесс еще не закончился.

Отсутствие общепринятых стандартов привело к созданию множества разрозненных систем, такие, как «умный дом» от различных компаний, что приводит к ряду проблем:

- Создание избыточного количества стандартов.
- Чрезмерное регламентирование наиболее простых объектов и процессов.
- Большое количество организаций и стандартов приводит к лоббированию интересов отдельных компаний в ущерб общих задач стандартизации.
- Долгие сроки разработки стандартов приводят к их моральному устареванию, так как они не успевают за развитием технологий, особенно на их раннем этапе.

Таким образом стоит рассмотреть две основные архитектуры: программную и физическую.

Условно программная архитектура состоит из 7 уровней:

1. **Физический уровень** — это датчики и электронные устройства, которые способны подключаться к «вещам» и получать данные от них.

2. **Датчики** собирают данные, но нам необходимо преобразовать их в понятный формат и подключить это устройство к системе, используя протокол обмена данными, который нужно настроить.

3. **Сетевое подключение.** Подключение устройства к беспроводной или проводной сети. Эта возможность подключения изменяется на основе контекста и домена.

4. **Аккумуляция данных.** Можно сказать, что данный слой отвечает за уровень безопасности и доступ к данным. Данный слой должен быть достаточно «мобилен» для внесения необходимых изменений.

5. **Абстрагирование данных.** Собранные данные используются для принятия решения или для целей отчетности. Это важный слой, в который входит фактический создаваемое решение и бизнес-логика.

6. **Уровень приложения.** На данном уровне происходит контроль, анализ и представление отчетов системы. Основываясь на этих данных, мы можем отображать отчеты или применять машинное обучение, какую-то специальную логику или использовать интеллектуальное решение и посылать сигнал обратно на датчики.

7. Последний уровень, это слой **пользовательского интерфейса.**

Это краткое описание организации программной части Интернета вещей. Программная архитектура представлена на Рисунке 11.

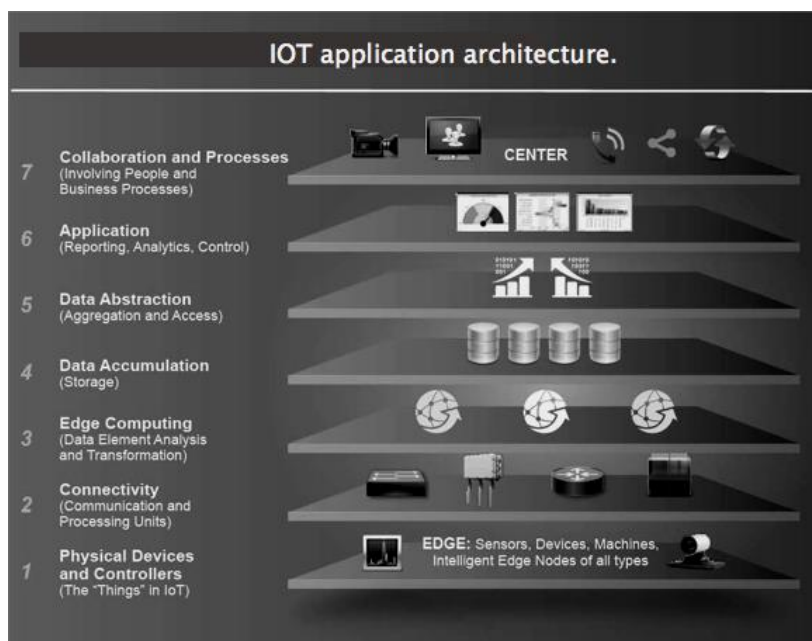


Рисунок 11 – Пример программной архитектуры Интернета вещей

Простая архитектура системы в физической реализации представляет собой взаимодействия всех участников сети (датчиков, устройств и т.д.), а также обработка полученной информации и принятие решений. Одна из таких реализаций представлена на Рисунке 12.

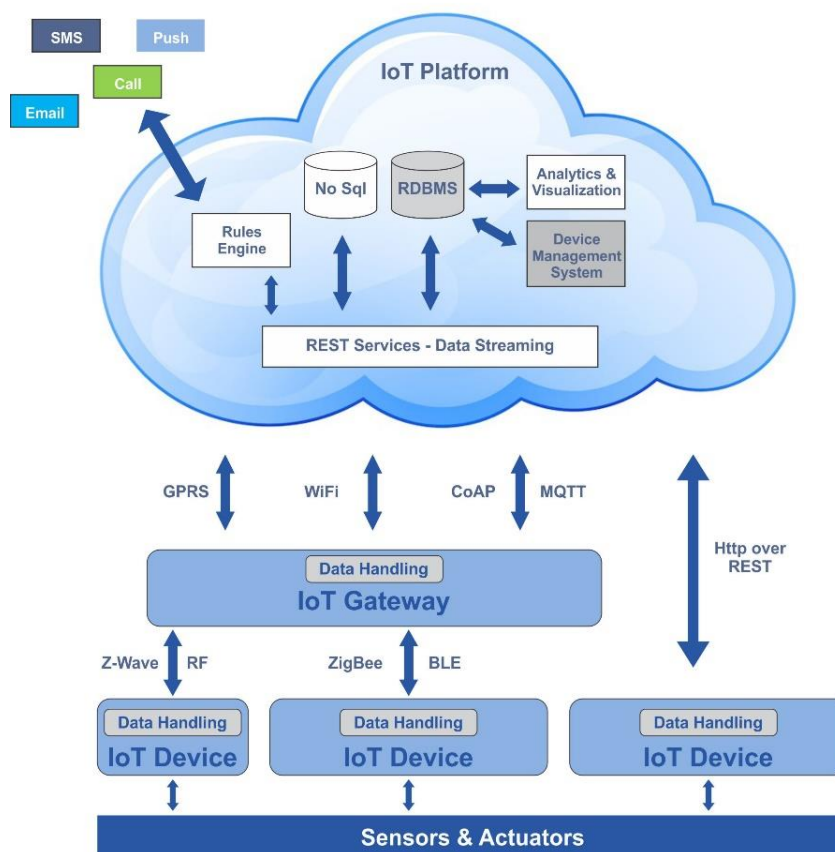


Рисунок 12 – Пример реализации архитектуры Интернета вещей

В зависимости от решаемых задач и используемых устройств архитектура системы будет изменяться. При её создании необходимо учитывать текущее развитие стандартов и возможность дальнейшей поддержки системы и её развития.

2.3 Технологии коммуникации между устройствами

Все датчики и устройства должны быть связаны в единую сеть для обмена информацией между собой. Решение какой тип связи выбрать основывается на решении конкретной задачи и окружения. Рассмотрим основные технологии коммуникации между устройствами.

WiFi — одна из самых распространённых беспроводных технологий передачи данных. Соответствует стандарту IEEE 802.11 (n/b/g)

и функционирует на частотах 2,4ГГц и 5ГГц в зависимости от использованного оборудования. Обеспечивает высокую скорость передачи данных, но имеет ряд недостатков применимых к интернет вещам, такие как:

- высокое энергопотребление;
- в диапазоне 2,4 ГГц множество устройств и пересекающихся технологий (Bluetooth);
- различные эксплуатационные ограничения для разных стран;
- слабая взломостойкость стандарта шифрования WEP, новые стандарты WPA и WPA2 имеют большую надежность.

Bluetooth — позволяет обмениваться информацией устройствам находящимся в радиусе 10 метров (новые версии устройств работают до 16 метров). Имеет существенный недостаток в безопасности, а также на качество связи сильно влияют внешние факторы. К плюсом можно отнести низкое энергопотребление и постоянное улучшение стандарта.

Технологии сотовой связи (2g, 3g, 4g) — данная технология имеет ряд преимуществ по сравнению с рассмотренными ранее. Большое покрытие и достаточно быстрая скорость передачи данных (для сетей с технологией 3g/4g) позволяет охватить большую территорию при реализации проектов без необходимости поиска отдельной точки доступа к сети. Основной проблемой считается обязательная привязка к сотовому оператору, возможная нагрузка на вышки сотового оператора (или их недоступность), высокое энергопотребление при передаче данных (особенно при слабом покрытии сети).

LoRaWan — открытый протокол для высокочастотных (до 1 млн устройств в одной сети) сетей с большим радиусом действия и низким энергопотреблением. Протокол обеспечивает двустороннюю связь с шифрованием для всех классов устройств. Архитектура протокола разрабатывалась в том числе и для того, чтобы легко найти мобильные объекты для отслеживания передвижений. Это наиболее быстрорастущее направление приложений интернета вещей. Основным недостатком является скорость передачи данных (до 5 КБ/с), но для передачи данных с датчиков этого вполне достаточно.

Local Area Network, LAN — компьютерная сеть, обычно покрывает относительно небольшую территорию. Соединение осуществляется с помощью сетевого кабеля. Имеет высокую скорость передачи данных, высокую помехоустойчивость, определенную адресацию внутри сети. Из минусов стоит отметить необходимость наличия маршрутизаторов, физической доступности для подключения устройства к сети кабелем.

Приведенные технологии не являются исчерпывающим описанием всех существующих решений, но являются основными при работе с интернет вещами.

3. Примеры реализации

3.1 Лабораторная установка

Лабораторная установка состоит из микрокомпьютера Raspberry PI 3 b+, Arduino Mega и комплекта датчиков и актуаторов, для удобства соединения используется специальная макетная плата (breadboard) (Рис. 13).

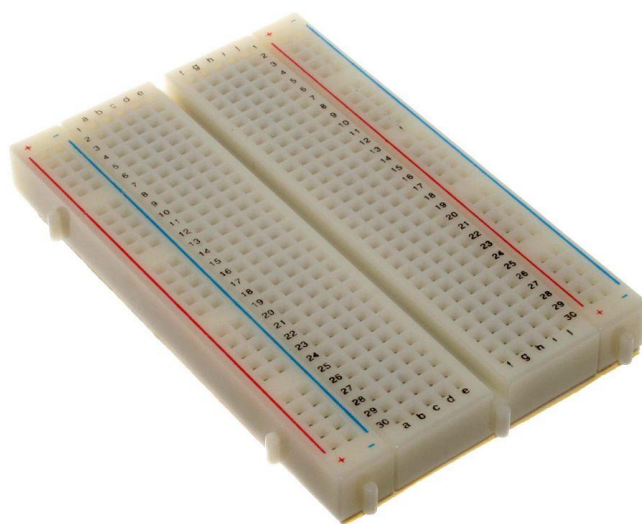


Рисунок 13 – Макетная плата (Breadboard)

Микрокомпьютер выполняет роль платформы для программирования Arduino, а также выступает в роли сервера работающего проекта. Обмен данными с Arduino происходит с использованием технологии WiFi, USB или Bluetooth, в зависимости от задач проекта.

3.2 Пример проекта «Умный музей»

Рассмотрим проект «Умный музей», основная цель проекта создать интеллектуальную систему контроля освещения и температуры экспонатов, собирать информацию для анализа и автоматического прогнозирования потребностей посетителей музея в режиме реального времени.

Рассмотрим задачи решаемые данным проектом непосредственно для музея:

- Влажность — определяется не только для помещения, но и для каждого экспоната рядом с которым расположен датчик.

- Температура — контроль температуры для каждого экспоната и помещения.
- Датчик приближения — определяет близость посетителя к экспонату.
- Управление освещением — позволяет создать профили освещенности экспоната (помещения) и поддерживать его в реальном времени.

Для реализации данного проекта и построения лабораторной установки потребуется: микрокомпьютер Raspberry Pi, Arduino, макетная плата, WiFi модуль, датчик температуры, датчик освещенности, датчики влажности и приближения.

Взаимодействие между микрокомпьютером и Arduino осуществляется с использованием WiFi технологии, подключение датчиков осуществляется с использованием макетной платы. Архитектура проекта представлена на рисунке 14.

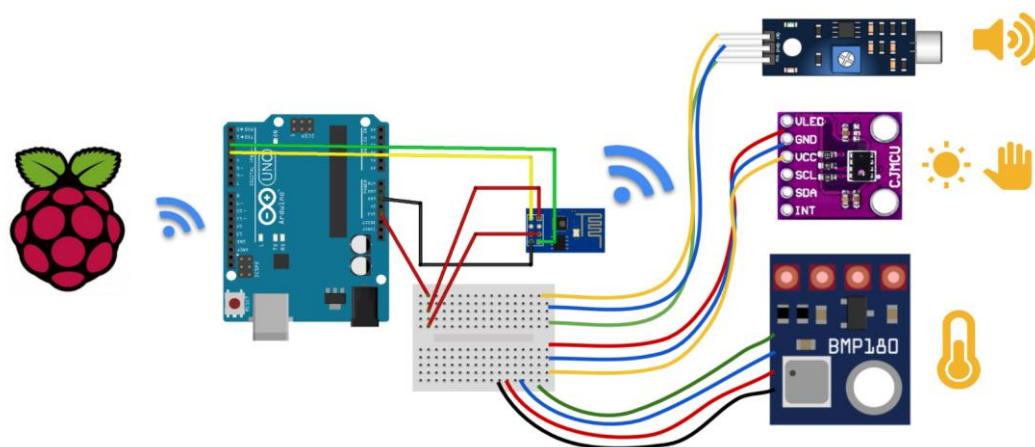


Рисунок 14 – Архитектура проекта «Умный музей»

В данном проекте Arduino выступает в роли платформы для размещения датчиков и передачи данных на управляющий компьютер (сервер). Каждая отдельная Arduino с набором датчиков представляет собой узел в сети («ноду»), количество которых ограничено только производительностью системы в целом.

Микрокомпьютер Raspberry Pi выступает в роли управляющего сервера для каждого узла в сети («ноды»), он получает, обрабатывает и хранит информацию с каждого датчика, а также в случае необходимости оповещает о необходимых изменениях (в рамках выполнения данного проекта).

Для изучения различных методов коммуникаций между устройствами, есть возможность заменить WiFi модуль на Bluetooth или подключить дополнительный микроконтроллер на основе Arduino с иным набором датчиков, что позволит протестировать работу различных методов коммуникации в реальном времени.

Лабораторная работа 1: «Первая программа на Arduino»

Цель работы: ознакомиться с программой Arduino IDE и изучить ее интерфейс на примере создания и загрузки в Arduino простейшей программы.

Общее описание программы Arduino IDE

Arduino IDE — это среда разработки программ для платформы Arduino. Она позволяет взаимодействовать с Arduino как для передачи данных, так и для прошивки кода (Рис. 15).

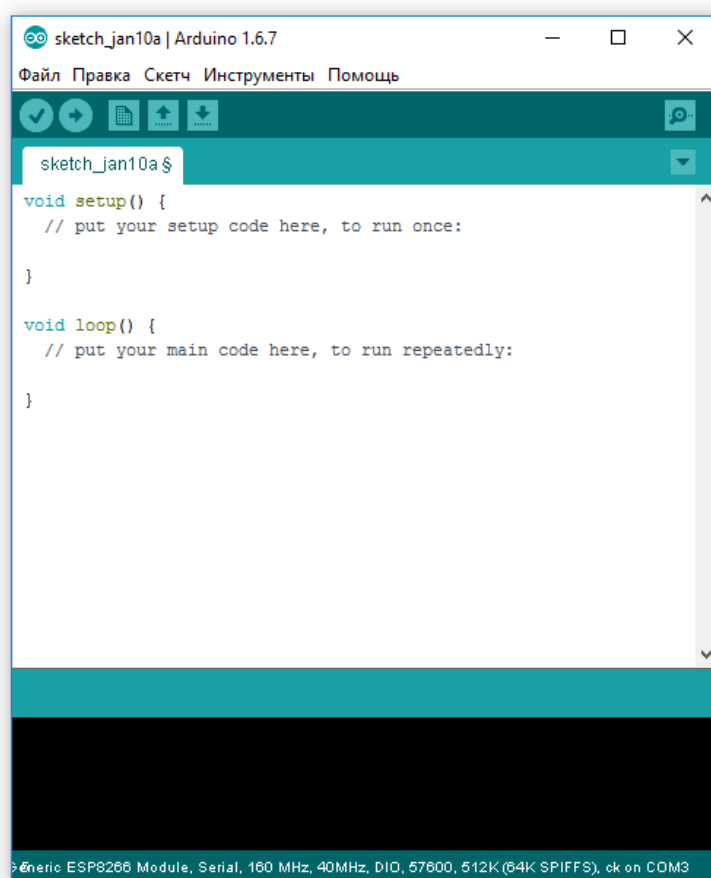


Рисунок 15 – Интерфейс программы Arduino IDE 1.6.7

Интерфейс имеет простую структуру. В верхней части расположена панель навигации, ниже кнопки для сохранения и загрузки программы в Arduino, следом текстовый редактор для написания программ, зеленая область под текстовым редактором отображает информацию о загрузке программы на Arduino, последняя часть — это консоль для вывода

служебной информации. В нижнем правом углу показывается информация о порте, к которому подключена Arduino.

Программы, разрабатываемые для Arduino, иногда называют «Скетчами». Скетчи пишутся в текстовом редакторе и сохраняются в файлах с расширением «.ino».

Попробуем загрузить свою первую программу (мигание светодиодом) на Arduino.

Шаг 1. Как выбрать программу?

Среда разработки Arduino IDE уже содержит базовый набор программ (Скетчей), это удобно для обучения или разработки собственных проектов используя их как шаблоны. Попробуем загрузить программу «Blink», которая позволит нам «мигать» светодиодом на плате Arduino. Данную программу удобно использовать для проверки работоспособности платы.

Для этого рассмотрим панель управления, она представлена кнопками (Рис. 16).



Рисунок 16 – Панель управления программами (Скетчами)

Первая кнопка, в виде «галочки», осуществляет проверку написанного кода, следующая (стрелочка) тоже проверяет код, но сразу после проверки осуществляет его загрузку на микроконтроллер. Следующие три кнопки по порядку: Создать новый документ, открыть существующий, сохранить текущий.

Нажмём на кнопку «открыть», выберите пункт «01.Basics» и программу «Blink». Откроется новое окно, но с уже готовой программой (Рис. 17).

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

Рисунок 17 – Стандартная программа Blink

Шаг 2. Подключить Arduino к компьютеру

Подключите Arduino к компьютеру. Обратите внимание, если всё подключено правильно и питания для платы достаточно, то загорится зеленый светодиод напротив надписи «On» (Рис. 17).

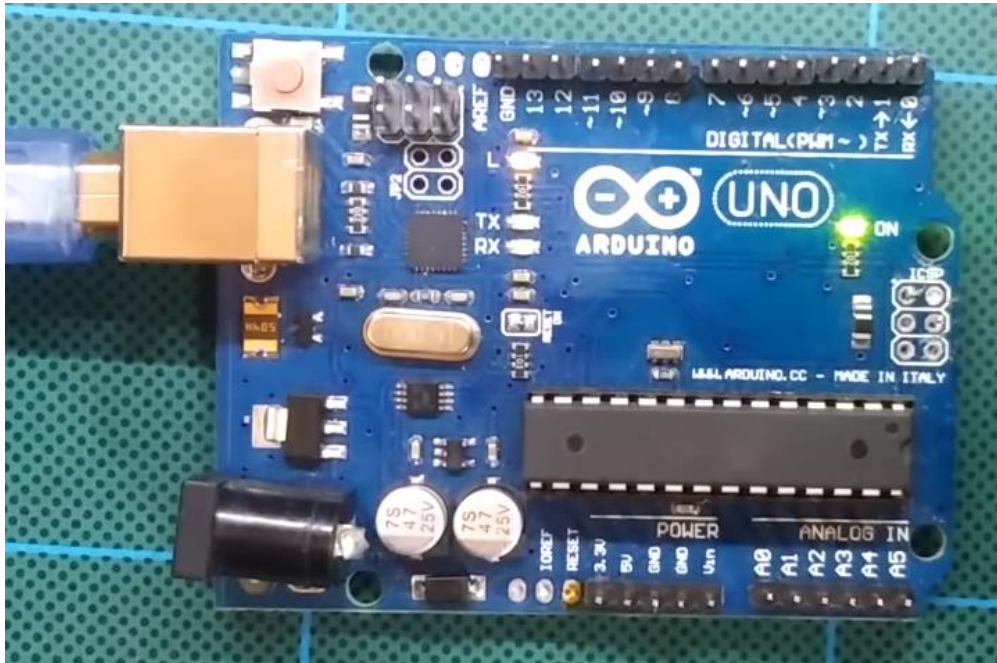


Рисунок 17 – Подключение платы Arduino

Шаг 3. Проверить подключение Arduino на компьютере

Если всё настроено верно, то в Arduino IDE в левом нижнем углу определится платформа и порт, к которому она подключена (Рис. 18).

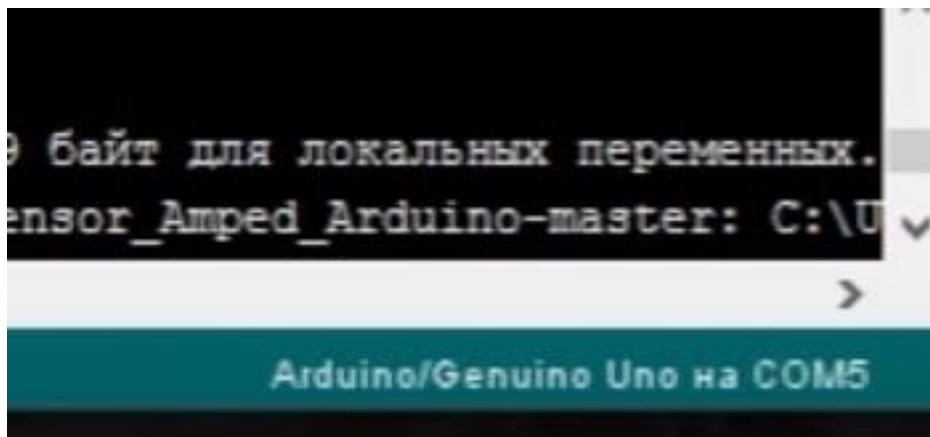


Рисунок 18 – Определение платформы и порта при подключении Arduino

Шаг 4. Загрузите программу на Arduino

Нажмите на кнопку «загрузка». Строка сообщений уведомляет о начале компиляции и завершении компиляции, затем происходит загрузка кода на плату Arduino (Рис. 18).



Рисунок 18 – Уведомление о начале компиляции и загрузке программы на плату Arduino

Шаг 5. Проверка выполнения программы

Как только программа будет загружена светодиод на плате начнет мигать с задержкой в 1 секунду (Рис. 19). Это говорит нам о том, что плата работает нормально и готова для создания новых проектов.

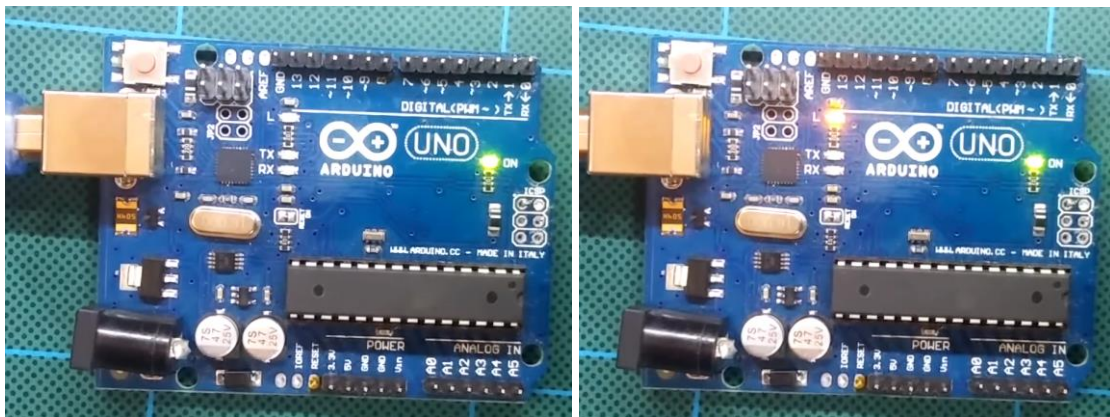


Рисунок 19 – Моргание светодиода на плате Arduino

Лабораторная работа 2: «Подключение кнопки управления светодиодом к Arduino»

Цель работы: ознакомиться с подключением устройств к микроконтроллеру и написанием программ для управления ими.

В данной лабораторной работе вы научитесь подключать устройства к микроконтроллеру и управлять миганием светодиода с кнопки.

Запустите Arduino IDE, в текстовом редакторе вы увидите уже написанные две основные функции:

- `setup()`: функция вызывается однократно при старте микроконтроллера.
- `loop()`: функция вызывается после `setup ()` в бесконечном цикле в течение всего времени работы микроконтроллера.

Стоит отметить, что на Arduino Uno уже имеется светодиод на плате и подключен он к 13 PIN (Pin – вывод микроконтроллера), каждый PIN на микроконтроллере может настраиваться как на вход (прием сигнала), так и на выход (подачу сигнала) (Рис. 20). В нашем случае необходимо подавать на светодиод сигнал, поэтому необходимо указать микроконтроллеру, что один из его выводов должен работать как выход (у нас это 13 pin).

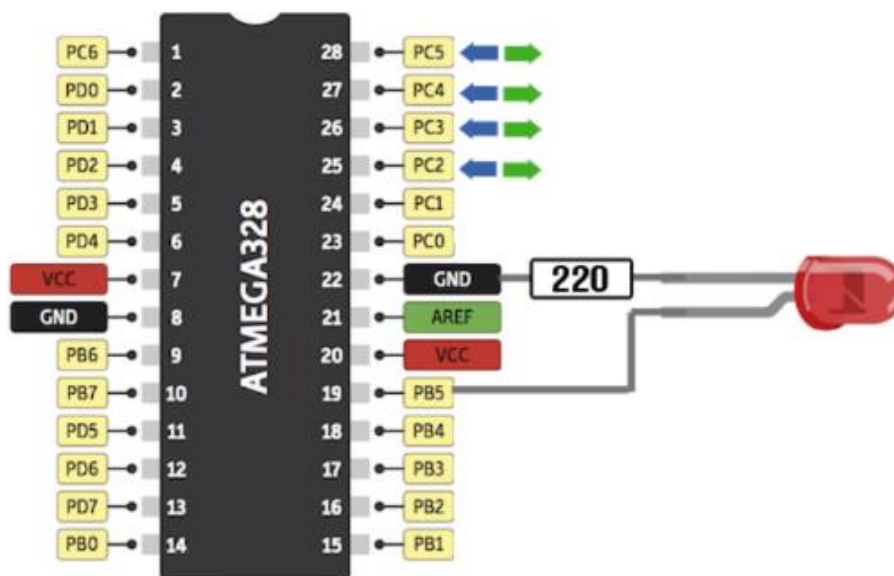


Рисунок 20 – Схем-а входов/выходов микроконтроллера Arduino

Шаг 1. Обзор базовых команд

Наша программа состоит из двух функций: `setup` и `loop`.

Как было сказано выше, функция `setup` выполняется один раз при запуске микроконтроллера, здесь выполняется настройка выходов микроконтроллера (Рис. 21). В данном случае указывается, что мы

настраиваем константу LED_BUILTIN (по умолчанию 13 pin и замена константы на цифру ничего не изменит) как выход.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Рисунок 21 – Настройка выхода микроконтроллера

Так как мы уже указали вывод (как выход) для управления светодиодом, то нам необходимо добавить вывод (как вход) для отправки сигнала с кнопки. Пусть это будет 3 pin (Рис. 22).

```
void setup() //функция установки/настройки, запускается один раз
{ //открывающая фигурная скобка функции
  pinMode(13, OUTPUT); //настройка 13-го пина на выход (светодиод)
  pinMode(3, INPUT); //настройка 3-го пина на вход (сигнал с кнопки)
} //закрывающая фиг. скобка
```

Рисунок 22 – Настройка 3 пина на вход (сигнал с кнопки)

Шаг 2. Функция loop

Данная функция выполняется циклически после запуска микроконтроллера. Здесь описывается сама программа. Но предыдущем шаге мы определили код мигания светодиодом, теперь нам необходимо добавить логику мигания светодиодом, а мигать он должен при появлении на 3 pin положительного сигнала, подаваемого кнопкой. Поэтому сначала мы считываем состояние этого входа и затем, если оно равно единице (кнопка нажата), выполняем условие мигание светодиодом.

Для этого нам необходимо добавить в коде условный оператор «if» и выполнить условие:

if (digitalRead(3) == 1) (Рис. 23).

```

void loop() //циклическая функция, работает бесконечно после функции настройки
{ //открывающая скобка функции
  if(digitalRead(3) == 1) //условный оператор if (если считанное с порта 3 значение равно 1, то..)
  { //открывающая скобка оператора
    digitalWrite(13, HIGH); //подать на выход 13 сигнал высокого уровня
    delay(1000); //задержка работы нк на 1000мс = 1с
    digitalWrite(13, LOW); //подать на выход 13 сигнал низкого уровня
    delay(1000); //задержка работы нк на 1с
  }
}

```

Рисунок 23 – Добавление условия проверки микроконтроллером 3 pin

Шаг 3. Подключение кнопки к микроконтроллеру

Порты Arduino реагируют на малейшие изменения электромагнитного поля, что может привести к ложному срабатыванию. Чтобы этого не происходило необходимо постоянно подавать отрицательный потенциал (когда кнопка выключена) используя подтягивающий резистор на 10К Ом (Рис. 24).

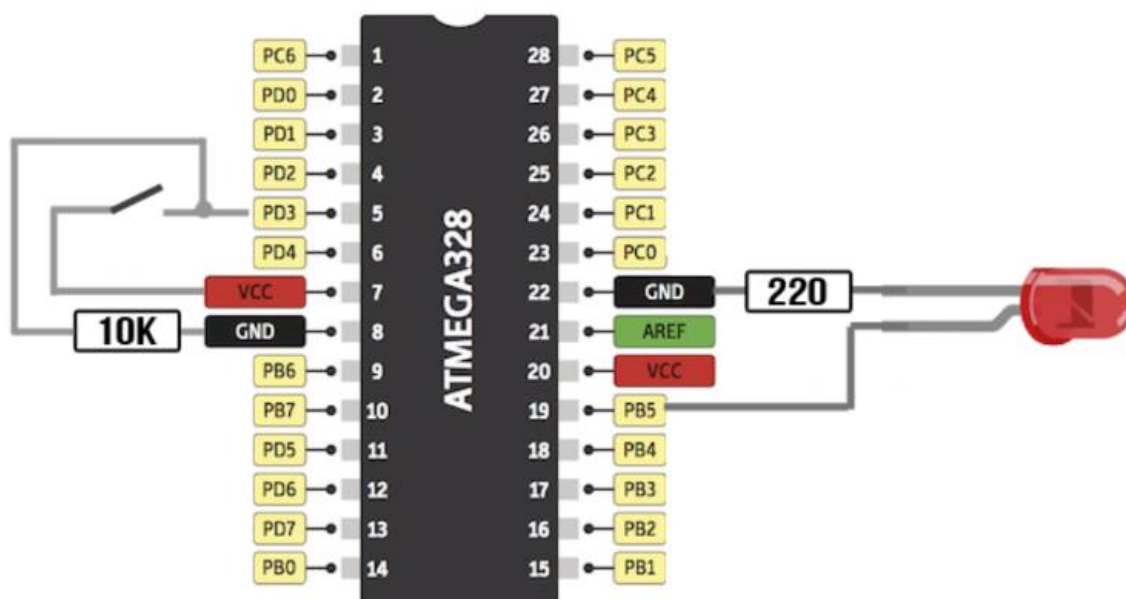


Рисунок 24 – Схема подключения кнопки к микроконтроллеру Arduino

Теперь необходимо собрать наш проект подключив все необходимые элементы используя макетную плату (breadboard). Внутренние соединения макетной платы состоят из нескольких зон:

Самые крайние линии (слева и справа) имеют параллельную структуру и отмечают за питание: синий — минус, красный — плюс.

Центральные гнезда пронумерованы и имеют перпендикулярные линии питания.

Общая схема структуры макетной платы и подключения кнопки к Arduino представлена на рисунке 25.

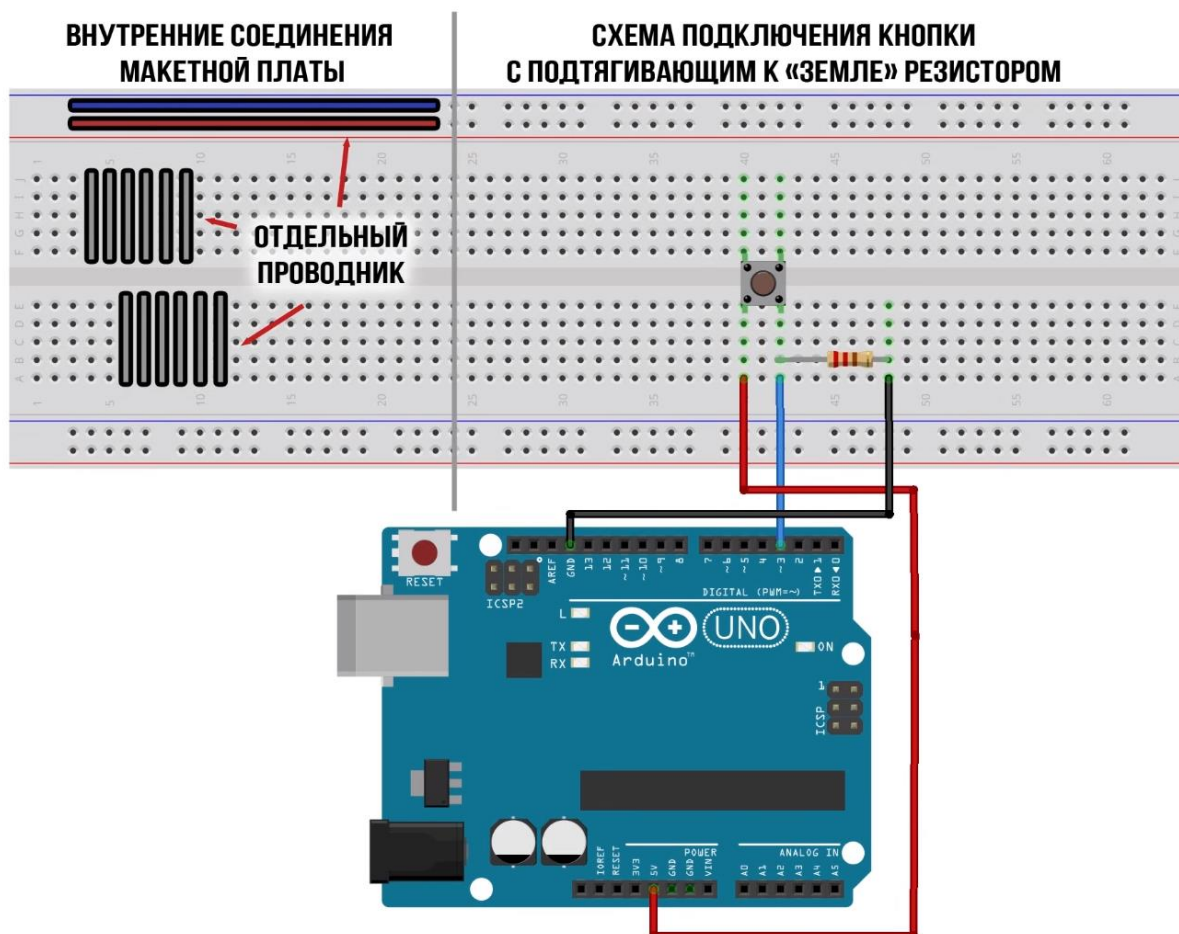


Рисунок 25 – Структура макетной платы и схема подключения кнопки к Arduino

Шаг 5. Загрузка кода в микроконтроллер

Подключите Arduino к компьютеру и убедитесь, что Arduino IDE правильно определила плату и порт подключения.

Нажмите кнопку загрузить и дождитесь положительного результата.

Шаг 6. Проверка собранного проекта

Если всё сделано правильно, то светодиод не мигает как раньше, так как микроконтроллер ждёт пока нажмут кнопку (Рис. 26).

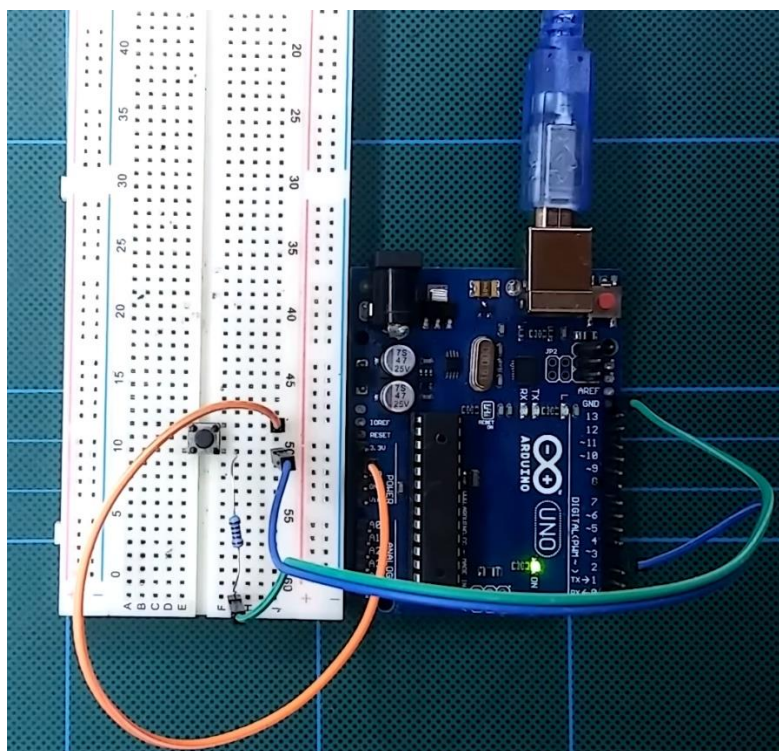


Рисунок 26 – Микроконтроллер ждёт сигнала с кнопки

При нажатии на кнопку происходит подача на 3 pin сигнала высокого уровня и светодиод начинает мигает с частотой 1 раз в 2 секунды (Рис. 27).

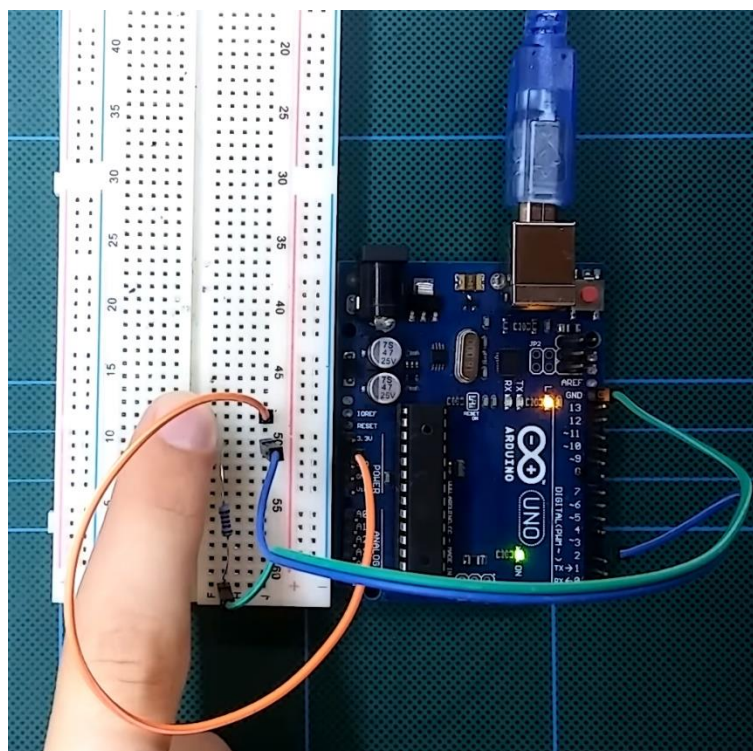


Рисунок 27 – Включение светодиода нажатием на кнопку

Мы можем изменить частоту мигания светодиода на любое другое просто исправив на нужное нам значение в поле «delay».

Лабораторная работа 3: «Получение данных с датчиков Arduino.»

Цель работы: ознакомиться с методом получения данных с Arduino используя среду разработки Arduino IDE.

В ходе выполнения данной работы Вы научитесь получать данные с датчиков, подключенных к Arduino. Для выполнения данной работы нам потребуется Arduino и совмещенный датчик атмосферного давления и температуры BME/BMP280 (Рис. 28).

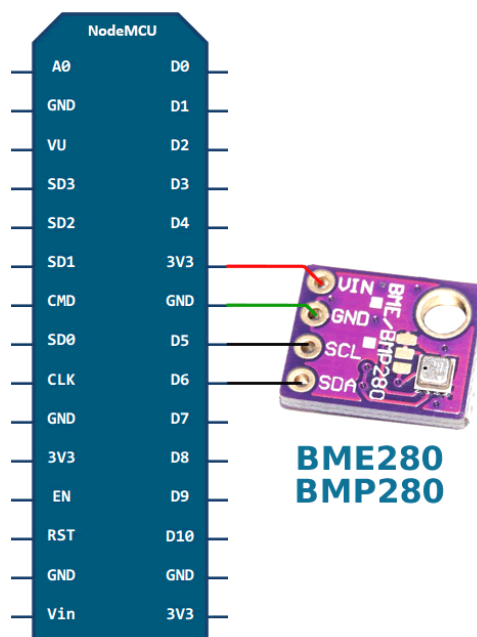


Рисунок 28 – Датчик атмосферного давления и температуры BME/BMP 280 со схемой подключения

Шаг 1. Подключение датчика к Arduino

При сборке проекта используется макетная плата и подключение по шине I2C. Схема соединения продемонстрирована на рисунке 29.

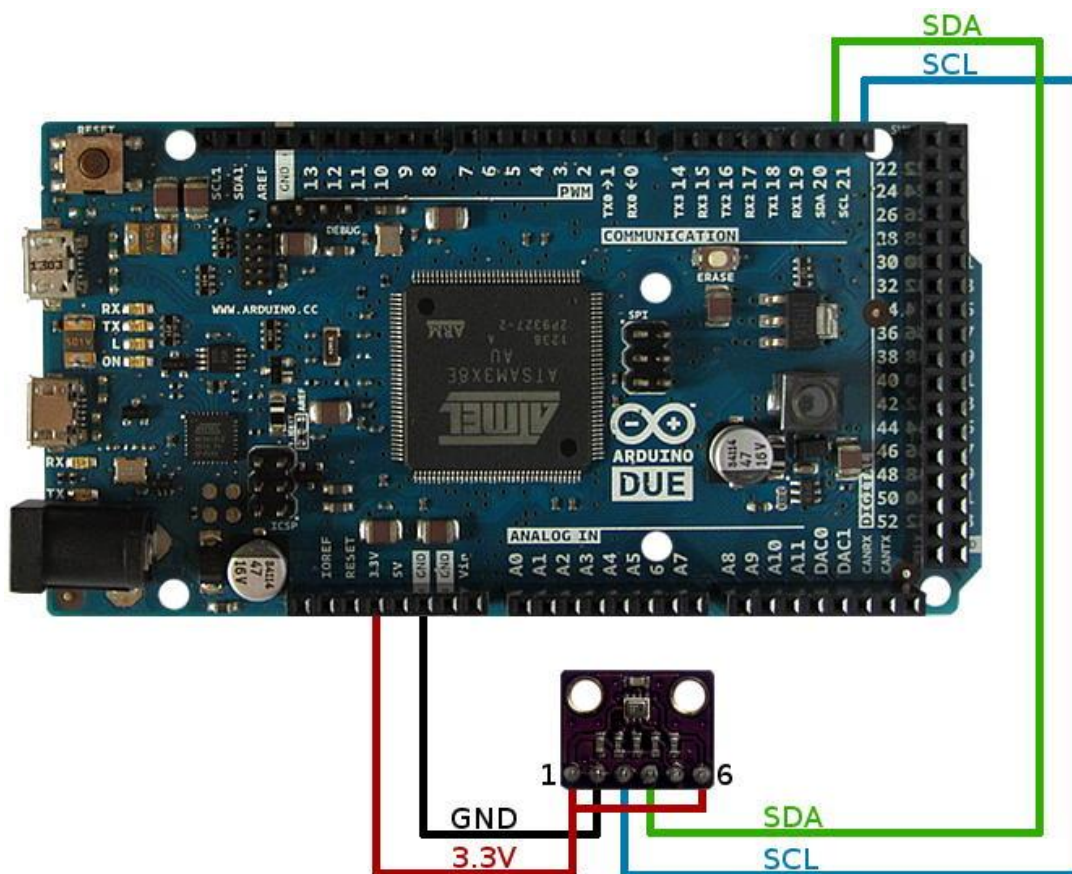


Рисунок 29 – Схема подключения датчика BME/BMP 280 к Arduino

Шаг 2. Выбор кода программы

Нам понадобится стандартная библиотека работы с датчиком.

Запустите Arduino IDE, выберите «файл» → «примеры» → Adafruit BMP280 Library → bmp280test

Шаг 3. Загрузка программы на Arduino

Подключите Arduino к компьютеру и убедитесь, что Arduino IDE правильно определила модель микроконтроллера и порт подключения. Нажмите кнопку «загрузка» и дождись положительного результата.

Шаг 4. Считывание данных с датчика

Откройте Arduino IDE и нажмите кнопку «монитор порта», которая находится справа вверху (Рис. 30).



Рисунок 30 – Кнопка «монитор порта» выделена красным кружком

Откроется окно с консолью с автоматическим определением Arduino и активным портом передающим данные (Рис. 30).

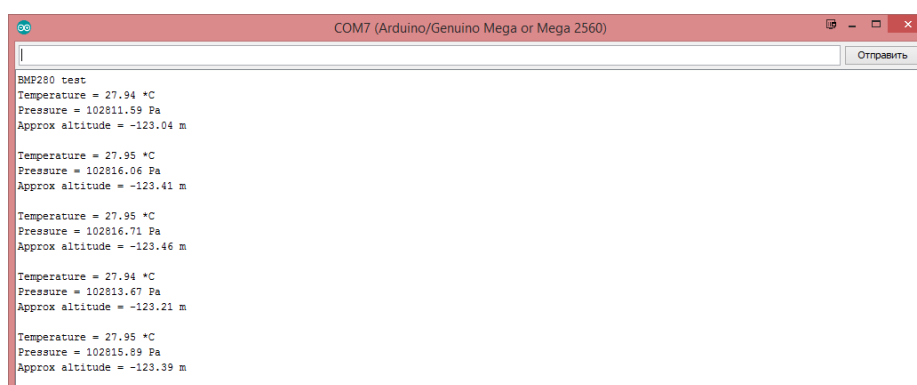


Рисунок 30 – Считывание данных с датчика в реальном времени

Литература

1. ГОСТ Р 51086-97. Датчики и преобразователи физических величин электронные
2. ГОСТ 14691-69 «Устройства исполнительные для систем автоматического регулирования».
3. Ред. кол.: С. И. Адян, Н.С. Бахвалов, В.И. Битюцков, А. П. Ершов, Л. Д. Кудрявцев, А. Л. Онищик, А. П. Юшкевич. Математический энциклопедический словарь / Гл. ред. Ю. В. Прохоров. — М.: Советская энциклопедия, 1988-01-01. — 845 с.
4. Adrian McEwen. Designing the Internet of Things 1st Edition – 2013 – 336 с.
5. Jerker Delsing, IoT Automation: Arrowhead Framework - CRC Press, 2017 – 366 с.

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ИНФОРМАТИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ

К середине 70-х годов в ЛИТМО активно развиваются работы по созданию систем автоматизации проектирования. Резко возрастает потребность в кадрах, способных эффективно применять вычислительную технику в различных областях наук и производства. В связи с этим в 1976 году из кафедры Вычислительной техники выделяется кафедра Прикладной математики, на которую возлагается задача по подготовке специалистов в области программирования и методов вычислений. Кафедру возглавляет д.т.н., проф. О.Ф. Немолочнов, работающий в области систем автоматизации проектирования ЭВМ. На кафедре работают: д.т.н., проф. Я.М. Цейтлин, специализирующийся в области машинного эксперимента.

Научное направление кафедры состояло в разработке регулярных методов проектирования тестов для логических схем ЭВМ на основе аппарата исчисления кубических комплексов. В результате выполнения ряда НИР были созданы САПР тестов для плат бортовых ЦВМ, стендовое оборудование для контроля, диагностики и наладки цифровых схем в виде двухслойных и многослойных плат ЭВМ. Сотрудничество с академическими организациями в участии в ежегодных школах-семинарах по технической диагностике, проводимых под руководством члена-корреспондента АН СССР Пархоменко П.П. (ИПУ г. Москва), отраслевыми НИИ (НИЦЭВТ г. Москва), конструкторскими бюро (КБЭ г. Харьков) и промышленными предприятиями (НПО ВТ г. Минск) позволило интегрировать научные исследования с последующим внедрением

результатов в производственные технологии и учебный процесс в единое целое.

В результате научных исследований были подготовлены кадры высшей квалификации: были подготовлены и защищены 5 докторских и несколько десятков кандидатских диссертаций. После защиты сотрудники НИЧ нашего института, как правило, переходили на преподавательскую работу. В частности, по кафедре ИПМ защитили диссертации и стали преподавателями: Шипилов П.А., Голованевский Г.Л., Блохин В.Н., Усвятский А.Е., Звягин В.Ф., Голыничев В.Н., Щупак Ю.А., Кукушкин Б.А., Раков С.В., Слоеу Б.А., Павловская Т.А., Денисова Э.В. и ряд других.

Кафедра занимается разработкой теоретических основ программирования в области моделирования, верификации, тестирования и диагностики вычислительных процессов программ. Вычислительные процессы, порождаемые программами, представляются и описываются графо-аналитическими моделями (ГАМ) в виде множества вершин и дуг связи между ними. ГАМ строится на основе концептуальной двухконтурной итерационно-рекурсивной модели (IRM), позволяющей описывать как ациклические, так и циклические вычислительные процессы.

Математическое описание ГАМ строится в виде кубических покрытий с использованием алгебро-топологического аппарата исчисления кубических комплексов. При построении кубических покрытий вычислительный процесс декомпозируется на множество параллельных структур с любым уровнем вложенности их друг в друга. Переход от программ к вычислительным процессам позволяет решать задачи проектирования программного продукта через верификацию в общем виде, т.е. без учета конкретных особенностей языков программирования, операционных систем и процессоров, которые могут быть как реальными, так и виртуальными.

Разрабатываемые методы являются детерминированными и являются составной частью любой технологии проектирования программного продукта. На основе разрабатываемых методов кафедра планирует создание учебно-исследовательской системы (УИС) в виде

САПР, позволяющей унифицировать лабораторный практикум путем создания единой базы знаний и базы данных, повысить объективность и качество оценки знаний студентов, повысить производительность работы преподавателей.

В настоящее время кафедрой руководит доцент, к.т.н. Д.И.Муромцев. Список проектов и научных направлений опубликован на сайте кафедры

http://iam.ifmo.ru/ru/listnaprav/show_all_naprav.htm .