

**ФГАОУ ВПО «Национальный исследовательский университет
«Высшая школа экономики»
Министерство образования Пензенской области
ГАОУ ДПО «Институт регионального развития Пензенской области»
Управление образования города Пензы
МБОУ «Лицей современных технологий управления № 2» г. Пензы
МБОУ финансово-экономический лицей № 29 г. Пензы
Портал поддержки Дистанционных Мультимедийных Интернет-Проектов «ДМИП.рф»**

**VII открытый региональный конкурс
исследовательских и проектных работ школьников
«Высший пилотаж - Пенза» 2025**

ДАТЧИК ПРОТИВ ПРОТЕЧКИ ВОДЫ И ГАЗА

Выполнил: учащийся

ГБНОУ Пензенской области «Губернский лицей»

7 ФМИ2 класса

Годин Сергей

Научный руководитель: учитель информатики

Гуляйкина Алина Игоревна

Пенза, 2025

Содержание

Введение.....	3
Робототехника.....	4
Arduino.....	6
Конструирование системы обнаружения и устранения утечек.....	7
Программирование контроллера.....	9
Результат.....	15
Вывод.....	15
Список используемой литературы.....	16
Приложение 1.....	17

Введение

В современном мире практически каждая техническая специальность тесно связана с робототехникой. Общество становится всё более автоматизированным, и множество задач, ранее выполняемых людьми, берут на себя машины. В связи с этим, в будущем, ученики, которые хотят обучаться техническим направлениям, неизбежно столкнутся с робототехникой на следующем этапе своего образования. Таким образом, изучение основ робототехники предоставляет учащимся возможность более эффективно подготовиться к будущей профессиональной деятельности, освоить начинающий уровень программирования и научиться сборке простейших роботов.

Я познакомился с этим направлением на занятиях «Робототехника», которые проходят в нашем лицее. На этих занятиях мы учимся основам робототехники на Arduino, создаем роботов и программируем их на определённые задачи.

Цели проекта:

Изучение основ разработки роботизированных систем на платформе Arduino, разработка системы контроля наличия утечек воды и природного газа в квартире, а, так же, управления исполнительными механизмами для остановки утечек.

Задачи:

1. Знакомство с робототехникой (где применяется, что в себя включает)
2. Знакомство с Arduino
3. Пошаговая сборка системы
4. Пошаговое программирование системы
5. Подведение итогов

Объект исследования:

Платформа Arduino и язык программирования C++

Методы исследования:

1. Теоретические: изучение литературы, посвящённой проектированию роботизированных систем с помощью Arduino
2. Практические: проектирование и разработка роботизированной системы, проверка ее работоспособности

Arduino — это всемирно известная платформа, на которой удобно осваивать программирование, микроэлектронику и робототехнику. В процессе работы над проектом я хочу приобрести навыки по разработке алгоритмов и реализовать их на языке программирования Си ++ на основе Arduino.

Мой проект ориентирован на популяризацию робототехники, которая является одной из самых динамично развивающихся областей науки в современном мире. Почти каждый

желает создать что-то уникальное своими руками. Освоение робототехники на платформе Arduino предоставляет возможность реализовать это стремление всем желающим. Вы сможете провести время с увлечением и создать множество оригинальных и развлекательных роботов, а более глубокое изучение этой области может привести к достижениям, которые окажутся полезными в любой сфере деятельности.

Arduino — это одна из самых доступных платформ в данной области, что позволяет людям с различными финансовыми возможностями приобрести этот электронный «конструктор».

Робототехника

Робототехника представляет собой область прикладной науки, сосредоточенную на проектировании, производстве и эксплуатации автоматизированных технических систем, известных как роботы. Робот — это программируемое механическое устройство, способное функционировать без вмешательства человека.

В последнее время наблюдается значительный рост технического прогресса, и робототехника становится одной из ключевых научных дисциплин. Эта сфера достигла впечатляющих результатов, а техническая революция предоставила роботам важное место в жизни человека как в промышленности, так и в бытовой сфере. Всё больше предприятий внедряют современные автоматизированные решения.

Робототехника не является отдельной отраслью; она интегрирует последние достижения технических, естественных наук и информационных технологий. В этой области пересекаются механика, системы управления и искусственный интеллект, что делает её важным направлением научно-технического прогресса. В робототехнике сочетаются такие дисциплины, как электроника, механика, кибернетика, телемеханика, мехатроника, информатика, радиотехника и электротехника. Таким образом, робототехник, в отличие от узкого специалиста, должен обладать обширными знаниями и системным подходом к решению задач.

Законы робототехники предписывают машинам следующее:

1. Робот не может навредить человеку или позволить нанести ему вред в результате бездействия.
2. Робот обязан подчиняться командам человека за исключением тех, что не соответствуют Первому закону.
3. Робот должен следить за собственной сохранностью, если это не идет вразрез с Первым или Вторым законом.

Основная цель создания роботов - это помощь человеку в выполнении тяжелой,

опасной или монотонной работы. Устройства должны освободить человека от неквалифицированного труда. Это позволит человеку сосредоточиться на разработке уникальных проектов и выполнении задач, где требуется не только точность, но и интуиция, индивидуальный вкус и опыт, а роботу - помочь достичь максимальных показателей эффективности и производительности.

В современном мире роботы присутствуют почти в каждой сфере деятельности. Существуют разные системы классификации.

Основная классификация роботов.

1. Классификация роботов по типу управления:

1) Автономные — совершают работу без вмешательства человека. Запрограммированный цикл их действий в зависимости от обстоятельств изменяется. Автономными роботами являются: роботы-пылесосы, газонокосилки, мойщики окон и т. д.

2) Полуавтономные — выполняют задачу без вмешательства человека, по заранее определённому алгоритму.

Человек контролирует основные движения робота, но в это же время бортовой процессор может измерять и реагирует на различные внешние воздействия.

2. Управляемые — контролируются человеком непосредственно дистанционно.

Классификация по типу позиционирования:

1) Стационарные — монтированные в фундамент, к несущим стенам или потолку по отношению к обслуживаемому оборудованию.

2) Передвижные — способные перемещаться в пространстве с помощью шасси, либо по ограниченной траектории по рельсам или индуктивным и оптическим трассам.

3. Классификация по типу назначения:

1) Промышленные — участвующие в производственном процессе изготовления изделий и деталей.

2) Бытовые — предназначены для помощи человеку повседневной жизни.

3) Медицинские — предназначенные для работы в медицинской отрасли.

4) Военные — предназначенные для ведения боевых действий и обороны различных стран мира.

4. Классификация по способу передвижения:

1) Подземные — соответственно перемещающиеся под землёй. Это могут быть исследовательские дроны.

2) Подводные — перемещающиеся под водой. Это могут быть подводный батискаф или торпеда.

- 3) Надводные — перемещающиеся над водой. Это могут быть лодки или катера.
- 4) Наземные — передвигающиеся по суше. Это самоходные машины на гусеничном или колёсном ходу. Некоторые модели перемещаются при помощи механических ног.
- 5) Летательные — перемещающиеся по воздуху над землёй. Это так называемые беспилотники и квадрокоптеры.

Это основная классификация роботов, но далеко не вся. Активно развивающаяся робототехника позволяет конструировать огромное множество различных роботов, поэтому просто невозможно перечислить их всех.

Arduino

Arduino - торговая марка аппаратно-программных средств для построения простых систем, моделей и экспериментов в области электроники, автоматике, автоматизации процессов и робототехники. Она обладает аппаратной и программной частями для простой разработки электроники. Аппаратная часть включает в себя большое количество видов плат Arduino со встроенными программируемыми микроконтроллерами, а также дополнительные модули. Программная часть состоит из среды разработки (программы для написания скетчей и прошивки микроконтроллеров Arduino), упрощенного языка программирования, огромного множества готовых функций и библиотек. История её начинается с начала 2000-х. С помощью Arduino каждый сможет создать своего робота.

Все Arduino платы обладают своими личными характеристиками. В своём проекте я использовал Arduino Uno. Большинство программистов начинают именно с платы UNO. Контроллер Uno является самым наилучшим вариантом для начала работы с платформой: она имеет удобный размер, достаточно доступна из-за массового выпуска всевозможных клонов, под нее написано огромное количество бесплатных уроков и скетчей.

На Рисунке 1. Представлено схематичное изображение платы Arduino Uno.

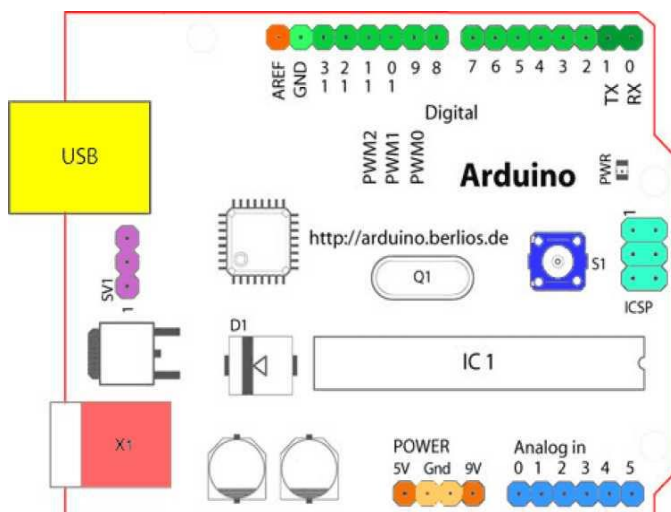


Рис. 1 – Arduino Uno

Основные компоненты платы начиная по часовой стрелке от верхнего центра:

- Аналоговый опорный контакт (оранжевый)
- Цифровая земля (светло-зеленый)
- Цифровые контакты 2-13 (зеленые)
- Цифровые контакты 0-1/Serial In/Out — TX/RX (темно-зеленые) — эти контакты нельзя использовать для цифрового ввода-вывода (`digitalRead` и `digitalWrite`), если вы также используете последовательную связь (например, `Serial.begin`).
- Кнопка сброса — S1 (темно-синяя)
- Внутрисхемный последовательный программатор (сине-зеленый)
- Аналоговые входы 0-5 (голубые)
- Контакты питания и заземления (питание: оранжевый, заземление: светло-оранжевый)
- Вход внешнего источника питания (9-12 В постоянного тока) — X1 (розовый)
- Переключает внешнее питание и питание USB (поместите перемычку на два контакта, ближайšie к нужному источнику питания) — SV1 (фиолетовый)
- USB (используется для загрузки скетчей на плату и для последовательной связи между платой и компьютером; может использоваться для питания платы) (желтый).

Конструирование системы обнаружения и устранения утечек

Чтобы процесс изучения Arduino и языка программирования был максимально практичен, я решил разработать систему контроля утечек воды и природного газа в квартире. Система должна распознавать наличие протечек, оповещать о них с помощью надписи на дисплее и сигнала светодиода, а так же, путем отправки СМС-уведомления о протечке. Так же, в случае обнаружения протечки воды, система должна с помощью серводвигателя перекрыть кран подачи воды в квартиру. В случае обнаружения утечки газа – перекрыть кран подачи газа и открыть створку окна для проветривания помещения.

Компоненты, которые понадобятся мне для конструирования системы:

- 1) Плата Arduino Uno
- 2) Датчик наличия воды Arduino
- 3) Датчик газа MQ-2 Arduino
- 4) Дисплей LCD1602
- 5) GSM модуль SIM800L
- 6) Блок реле для управления сервомоторами и линейным приводом.

- 7) Сервомотор управления шаровым краном – 2 шт.
- 8) Линейный привод
- 9) Модуль преобразования напряжения регулируемый LM2596S – 2 шт.
- 10) Светодиод – 2 шт.
- 11) Резистор 10 кОм 4 шт.
- 12) Блок питания 12 Вольт
- 13) Провода для соединения компонентов
- 14) Гнездо и выключатель питания

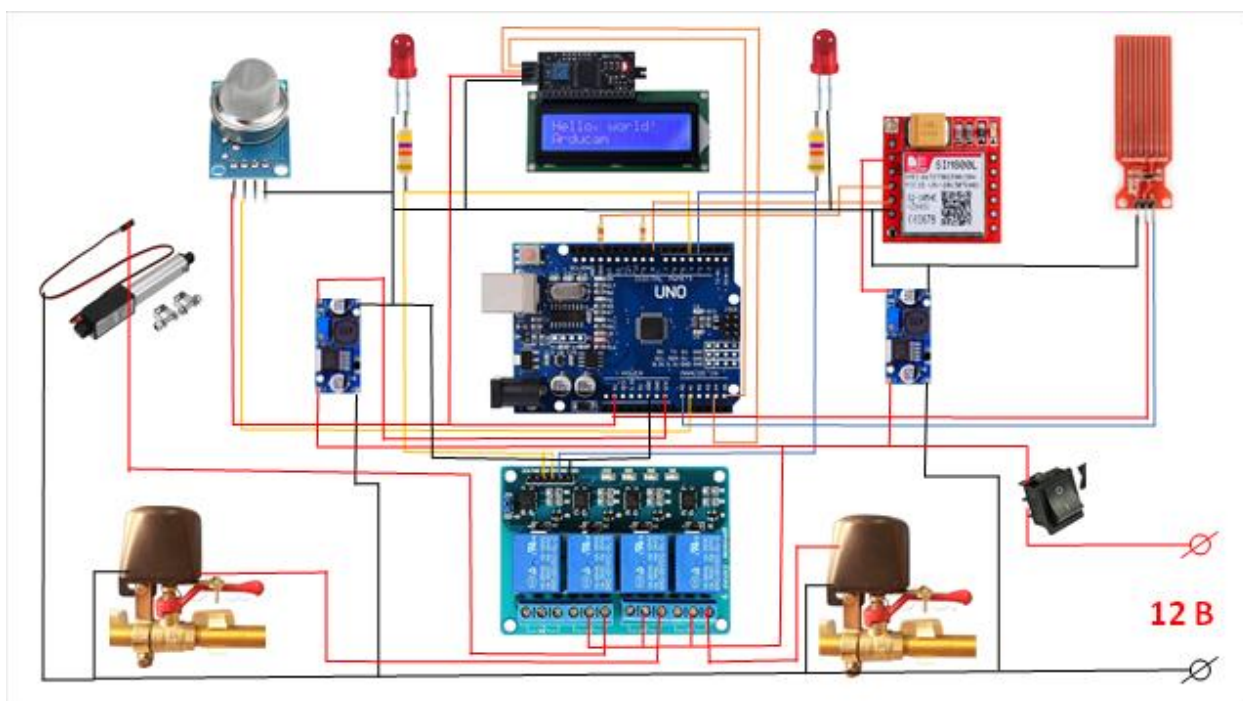


Рис.2. Схема соединения модулей и компонентов.

Основой нашего проекта стал контроллер Uno CH340G – аналог Arduino Uno.

К аналоговому входу A1 подключен датчик газа MQ-2 Arduino. К аналоговому входу A0 подключен датчик наличия воды.

К цифровому выходу 4, через резистор, подключен светодиод, который будет сигнализировать о наличии утечки газа. Параллельно, с этого же выхода, идет сигнал управления на модуль реле, с помощью которого управляется двигатель перекрывающий кран подачи газа и линейный привод, который крепится на открывающуюся створку окна и, в случае утечки газа, открывает окно для проветривания помещения.

К цифровому выходу 3, через резистор, подключен светодиод, который будет сигнализировать о наличии утечки воды. Параллельно, с этого же выхода, идет сигнал управления на модуль реле, с помощью которого управляется двигатель перекрывающий

кран подачи воды.

Так же, контакт 8 контроллера Uno соединен с контактом TXD GSM-модуля SIM800L. Контакт 9 контроллера Uno должен быть соединен с контактом RXD GSM-модуля SIM800L, но так как максимальное напряжение на входе RXD модуля SIM800L должно быть в диапазоне от 2,1В до 3,1 В, а на выходе с платы Uno на контакте 9 – 5В, то мы применим равноплечий делитель напряжения из двух резисторов номиналом 10 кОм. В результате, мы получим напряжение на выходе – 2,5 В. Это напряжение корректно вписывается в допустимый диапазон. Схема подключения GSM-модуля SIM800L изображена на рисунке 2.

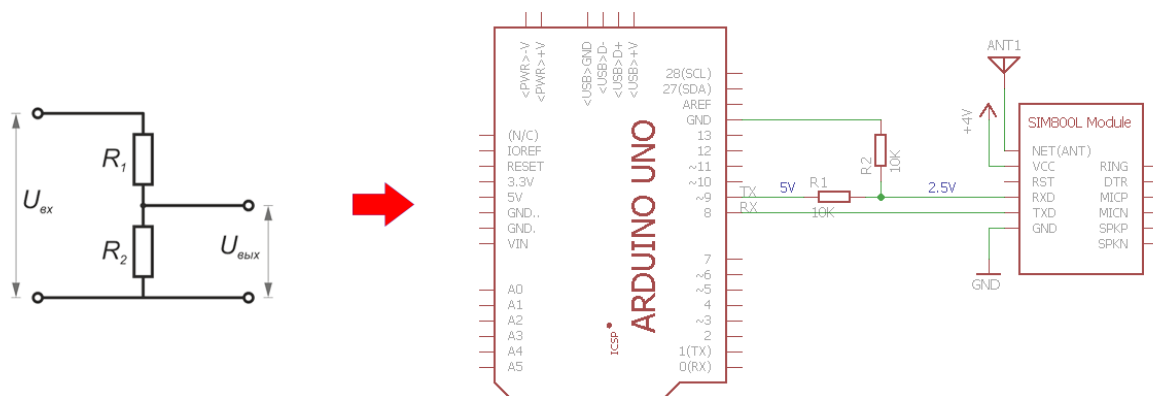


Рис. 3 – Подключение контроллера Uno к SIM800L

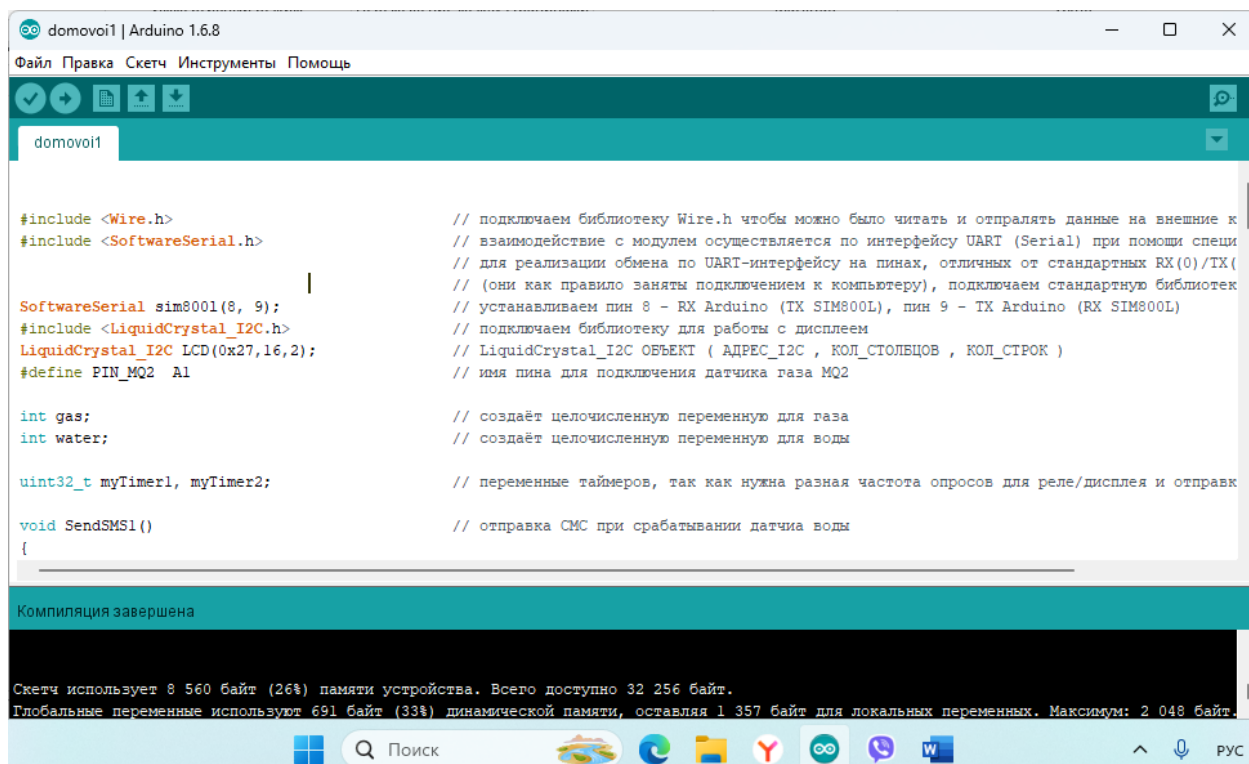
К аналоговым контактам А4 и А5 подключен модуль дисплея, на котором будет отображаться информация о наличии утечек воды или газа.

Так как управляемые контроллером двигатели и линейный привод работают от напряжения 12В, то питание демонстрационного стенда моего проекта осуществляется от блока питания, который преобразует переменное напряжение 220В в постоянное 12В с максимальной силой тока 2А. При этом, питание контроллера Uno осуществляется напряжением 5В, а питание модуля SIM800L напряжением 3.4-4.5 В, рекомендованное 4 В. Для того чтобы получить все требуемые мне номиналы напряжения, в проекте я использовал два регулируемых преобразователя напряжения LM2596S. На вход питания этих преобразователей поступает 12В. На одном преобразователе с помощью переменного резистора я установил выходное напряжение 5В для питания контроллера Uno, а на другом – 4 В, для питания SIM800L.

Программирование контроллера

Самая сложная, но и самая интересная часть - это написание программы. На эту часть работы ушло больше всего времени, поскольку я ещё неопытный программист.

Программа (скетч) для Arduino пишется и компилируется в среде разработки Arduino IDE. Эта среда подсвечивает синтаксис языка, позволяет загружать и устанавливать сторонние библиотеки к всевозможным платам расширения, позволяет скомпилировать проект из исходного кода и записать полученный машинный при помощи загрузчика платы Arduino непосредственно в саму плату. Основой для написания скетча является язык программирования C++. После этого плата автоматически перезагружается, и новая прошивка сразу начинает работать. Это очень удобно, т.к. все действия, связанные с программированием и запуском программы на устройстве, выполняются в одной программе.



```

#include <Wire.h> // подключаем библиотеку Wire.h чтобы можно было читать и отправлять данные на внешние к
#include <SoftwareSerial.h> // взаимодействие с модулем осуществляется по интерфейсу UART (Serial) при помощи спеш
// для реализации обмена по UART-интерфейсу на пинах, отличных от стандартных RX(0)/TX(
// (они как правило заняты подключением к компьютеру), подключаем стандартную библиотек
// устанавливаем пин 8 - RX Arduino (TX SIM800L), пин 9 - TX Arduino (RX SIM800L)

SoftwareSerial sim8001(8, 9); // подключаем библиотеку для работы с дисплеем
#include <LiquidCrystal_I2C.h> // LiquidCrystal_I2C ОБЪЕКТ ( АДРЕС_I2C , КОЛ_СТОЛБЦОВ , КОЛ_СТРОК )
LiquidCrystal_I2C LCD(0x27,16,2); // имя пина для подключения датчика газа MQ2
#define PIN_MQ2 A1

int gas; // создаёт целочисленную переменную для газа
int water; // создаёт целочисленную переменную для воды

uint32_t myTimer1, myTimer2; // переменные таймеров, так как нужна разная частота опросов для реле/дисплея и отправок

void SendSMS1() // отправка СМС при срабатывании датчика воды
{

```

Компиляция завершена

Скетч использует 8 560 байт (26%) памяти устройства. Всего доступно 32 256 байт.
Глобальные переменные используют 691 байт (33%) динамической памяти, оставляя 1 357 байт для локальных переменных. Максимум: 2 048 байт.

Рис 4. Arduino IDE

Логика работы устройства должна быть такой:

- при обнаружении воды датчиком наличия воды должен загореться светодиод, на дисплее должно появиться сообщение, что обнаружена утечка воды и с помощью GSM-модуля SIM800L устройство должно отправить SMS о наличии протечки на указанный в программе номер телефона. Так же, контроллер должен подать напряжение на соответствующий контакт модуля реле к которому подключен двигатель, который должен перекрыть кран подачи воды.

- при обнаружении наличия газа в воздухе с помощью датчика газа MQ-2 должен загореться светодиод, на дисплее должно появиться сообщение, что обнаружена утечка газа и с помощью GSM-модуля SIM800L устройство должно отправить SMS о наличии утечки на указанный в программе номер телефона. Так же, контроллер должен подать напряжение

на соответствующий контакт модуля реле к которому подключен двигатель, который должен перекрыть кран подачи газа и линейный привод для открытия окна на проветривание.

1 этап

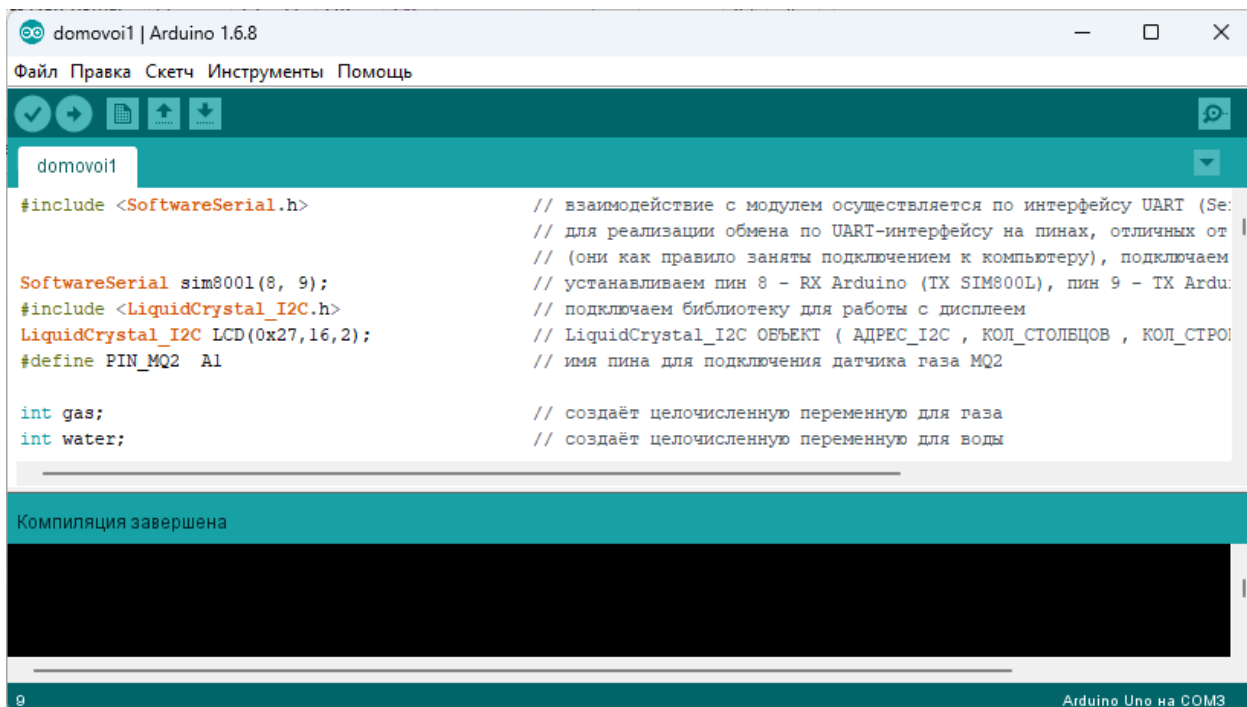
Изучением программной платформы Arduino я занимаюсь на уроках по робототехнике. При первоначальном запуске на поле, где мы в дальнейшем будем писать программу, мы видим только две функции `void setup()` и `void loop()`, ограниченных фигурными скобками.

Сразу же хочется отметить, что фигурными скобками отделяются все структуры программы, циклы, функции, различные инструкции. `void setup()` дает нашему коду возможность поучаствовать в инициализации системы.

Проще говоря, здесь мы пропишем программы, которые выполняются только один раз - при запуске.

`void loop()` используется в качестве непрерывного цикла. Сюда помещается вся основная программа, то есть здесь прописываются все действия, которые мы хотим получить от робота.

Первым этапом я подключил все нужные нам библиотеки для работы дополнительных модулей.



```
domovoi1 | Arduino 1.6.8
Файл Правка Скетч Инструменты Помощь
domovoi1
#include <SoftwareSerial.h> // взаимодействие с модулем осуществляется по интерфейсу UART (Serial)
// для реализации обмена по UART-интерфейсу на пинах, отличных от RX и TX,
// (они как правило заняты подключением к компьютеру), подключаем
// устанавливаем пин 8 - RX Arduino (TX SIM800L), пин 9 - TX Arduino
SoftwareSerial sim8001(8, 9);
#include <LiquidCrystal_I2C.h> // подключаем библиотеку для работы с дисплеем
LiquidCrystal_I2C LCD(0x27,16,2); // LiquidCrystal_I2C ОБЪЕКТ ( АДРЕС_I2C , КОЛ_СТОЛБЦОВ , КОЛ_СТРОК )
#define PIN_MQ2 A1 // имя пина для подключения датчика газа MQ2

int gas; // создаёт целочисленную переменную для газа
int water; // создаёт целочисленную переменную для воды

Компиляция завершена
9 Arduino Uno на COM3
```

Рис. 5. Библиотеки

Вторым этапом я назначил переменные для воды, газа и таймеров. Нам нужно чтобы программа опрашивала датчики воды и газа каждые 0,5 с и, если значения, передаваемые с

датчиков, превысят заданный нами порог, то контроллер должен подать сигнал на соответствующий светодиод и на пин управления реле. Для отправки СМС будем использовать таймер на каждые 10 секунд.

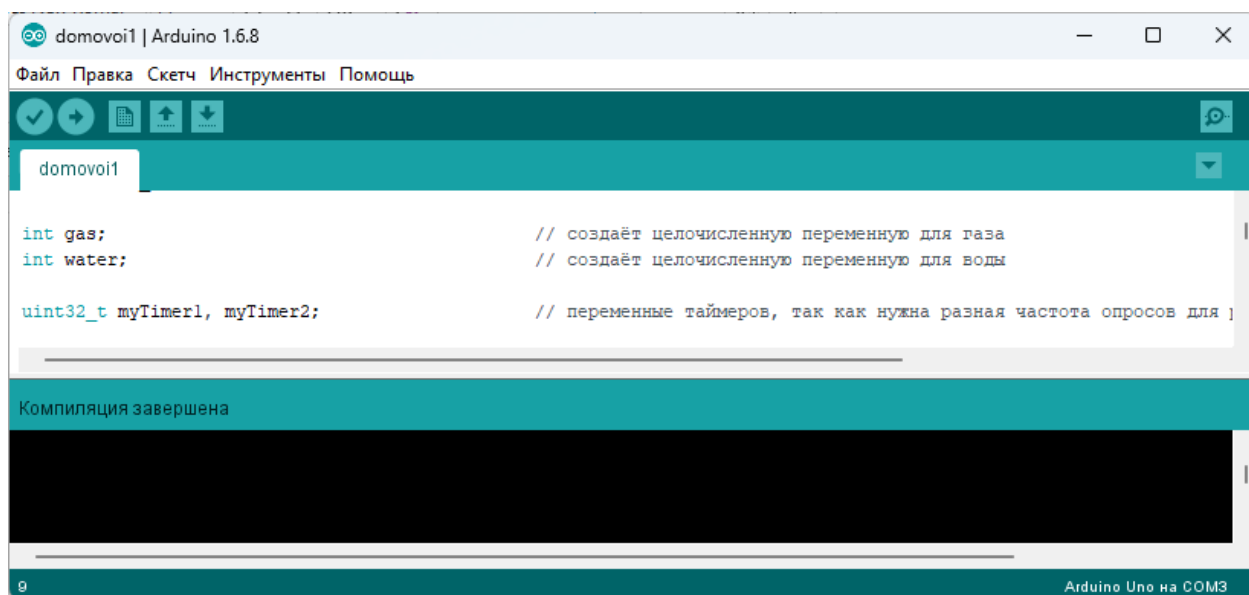


Рис. 6. Переменные

Третьим этапом я написал код для отправки СМС GSM-модулем SIM800L. Отдельно написан код для отправки СМС в случае протечки воды (SendSMS1), отдельно для газа (SendSMS2)

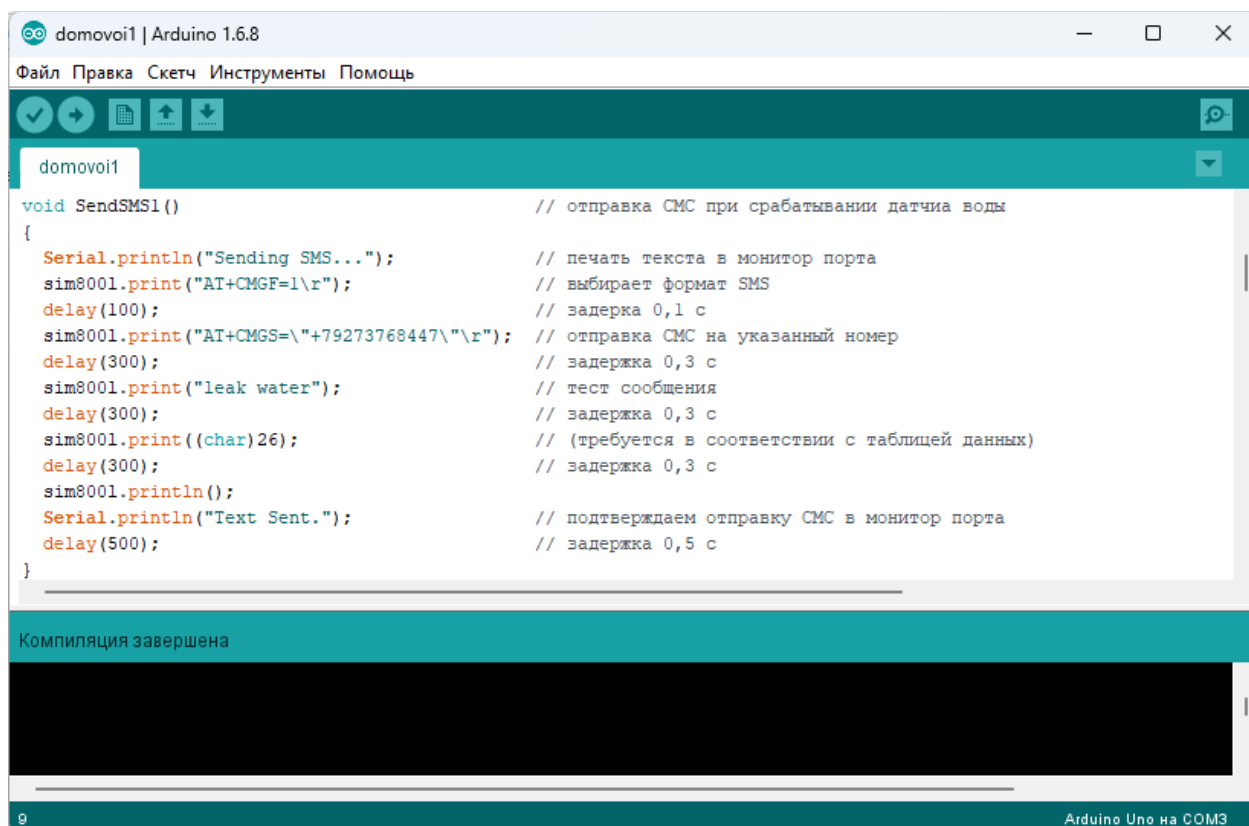
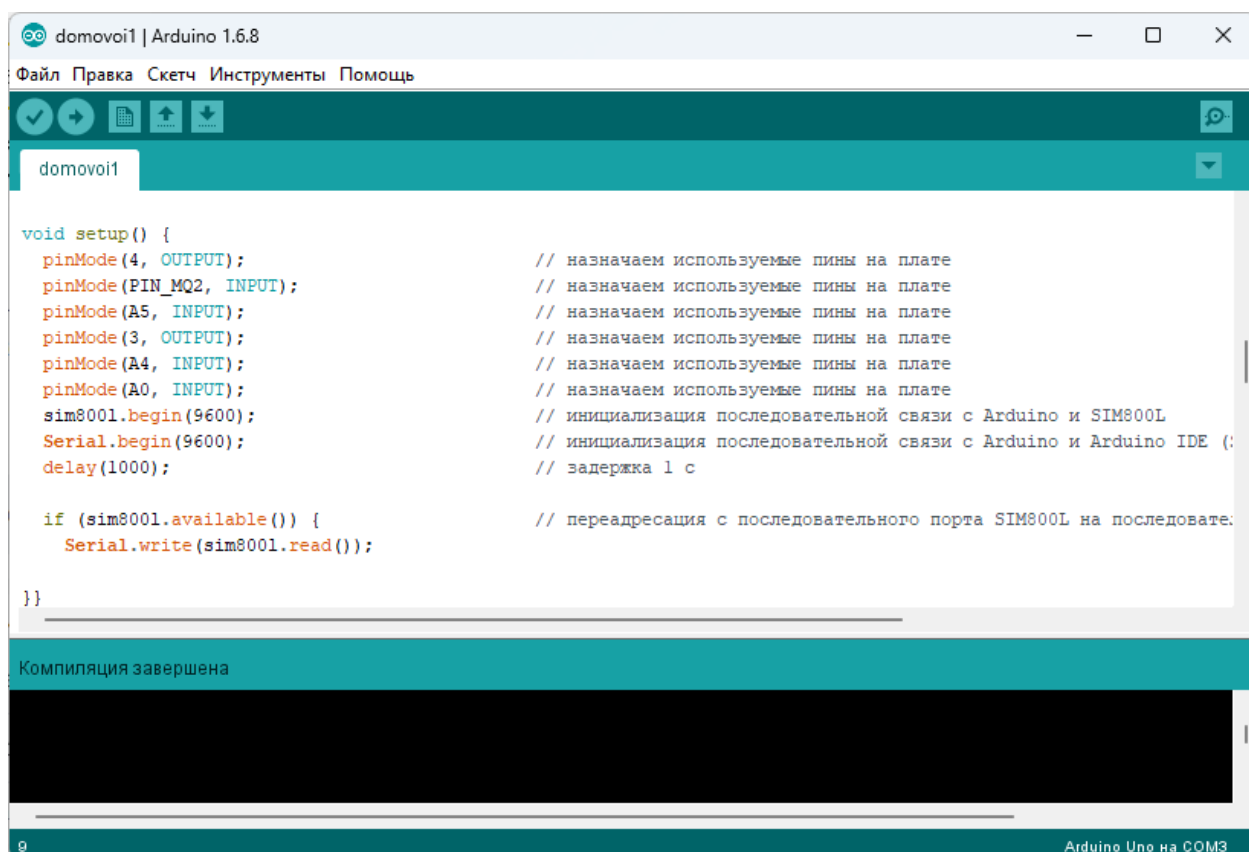


Рис. 7. Отправка СМС

Четвертым этапом я назначил все нужные нам пины на плате UNO, инициализировал

связи UNO с модулем SIM800L и Arduino IDE.



```
void setup() {
  pinMode(4, OUTPUT);           // назначаем используемые пины на плате
  pinMode(PIN_MQ2, INPUT);     // назначаем используемые пины на плате
  pinMode(A5, INPUT);          // назначаем используемые пины на плате
  pinMode(3, OUTPUT);          // назначаем используемые пины на плате
  pinMode(A4, INPUT);          // назначаем используемые пины на плате
  pinMode(A0, INPUT);          // назначаем используемые пины на плате
  sim8001.begin(9600);         // инициализация последовательной связи с Arduino и SIM800L
  Serial.begin(9600);          // инициализация последовательной связи с Arduino и Arduino IDE (COM)
  delay(1000);                 // задержка 1 с

  if (sim8001.available()) {   // переадресация с последовательного порта SIM800L на последовательный порт
    Serial.write(sim8001.read());
  }
}
```

Компиляция завершена

9 Arduino Uno на COM3

Рис 7. void setup

Пятым этапом по первому таймеру на 0,5 с MyTimer1 мы начинаем получать данные с датчиков и описывать условия выполнения действий при разных значениях поступающих с этих датчиков. Опытным путем, получая значения с датчиков в монитор порта я определил, что для датчика воды, при отсутствии на нем влаги выдаются значения меньше 100. Для датчика газа, при отсутствии воздействия газа – меньше 200. Буду считать, что превышение этих показателей будет говорить о протечке. В этой части программы мы выводим надписи о наличии протечек на монитор и направляем сигнал на соответствующий пин платы UNO для включения моторов шаровых кранов через блок реле.

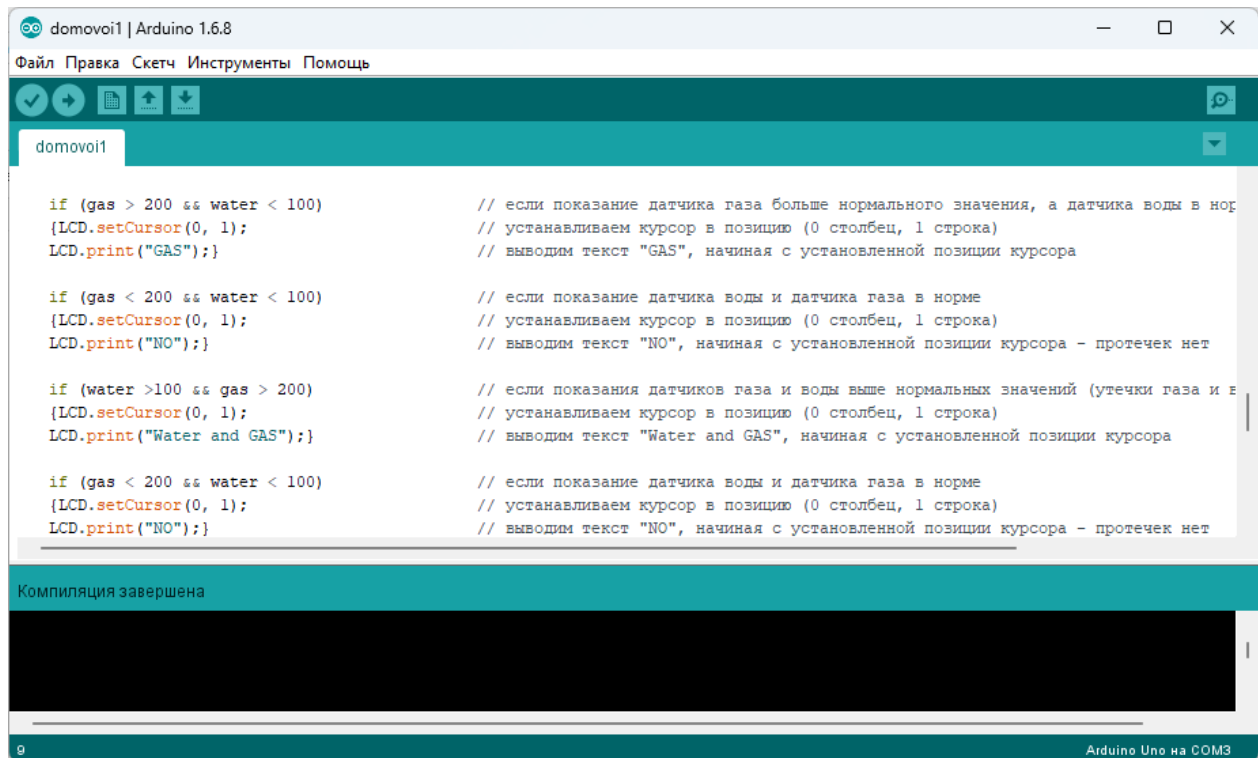


Рис. 8. Блок условий.

Последним этапом, по второму таймеру MyTimer2 на 10 секунд мы запускаем, при наличии превышения значений с датчиков воды и газа, соответствующие программы отправки СМС.

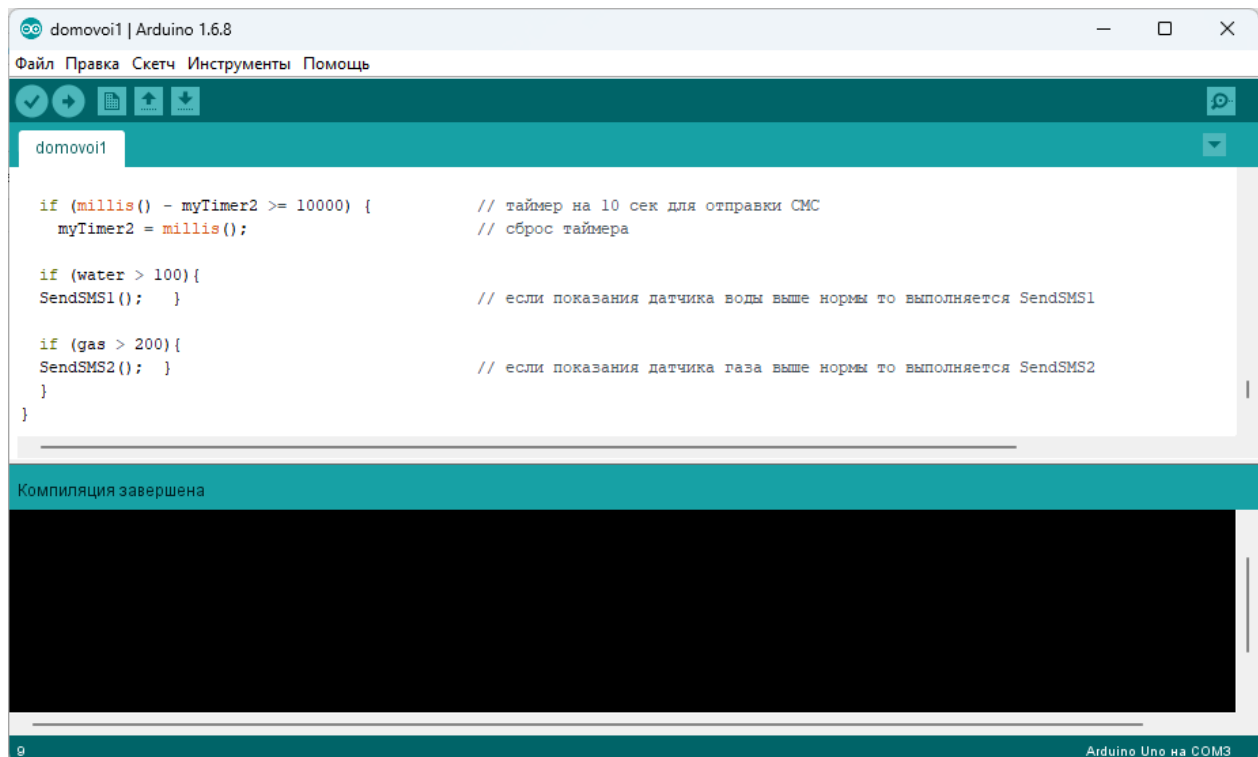


Рис. 9. Отправка СМС по условиям.

Получившийся в итоге код программы полностью представлен в Приложении 1.

Результат

После загрузки программы в контроллер UNO, я включил устройство и проверил его работу. Я имитировал срабатывание датчика утечки воды, поместив несколько капель на датчик. На устройстве загорелся соответствующий светодиод, на дисплее появилась информация Leak: Water, двигатель закрыл кран подачи воды. Через несколько секунд я получил СМС «Leak water» на свой телефон, номер которого я указал в программе.

Далее, я имитировал наличие газа в воздухе, поднеся к датчику газа зажигалку и нажав на кнопку подачи газа. На устройстве загорелся соответствующий светодиод, на дисплее появилась информация Leak: GAS, двигатель закрыл кран подачи газа, линейный привод начал выдвижение штока для открытия окна на проветривание. Через несколько секунд я получил СМС «Leak gas» на свой телефон, номер которого я указал в программе.

Вывод

В ходе выполнения проекта мною были решены все планируемые задачи и в итоге достигнута основная цель: научиться основам робототехники на Arduino. Я создал устройство, которое контролирует наличие воды на датчике воды и газа в воздухе и запрограммировал его на определённые задачи.

На изучение и создание проекта мне потребовалось 3 месяца. На основе полученной действующей модели можно создавать готовые устройства для работы в помещениях. Для этого необходимо поместить все компоненты в более компактный корпус, удлинить кабели исполнительных механизмов и предусмотреть резервное питание в виде аккумулятора на случай отключения подачи электроэнергии.

Надеюсь, мой проект будет интересен для всех, кто хочет начать изобретать и станет примером в начальном изучении Arduino.

Список используемой литературы

1. Новые сложные задачи на плате Ардуино/ Саттер, Герб, Вильямс, 2017.-272 с.
2. Изучаем Arduino. Инструменты и методы технического волшебства/ Джереми Блум, 2020.- 544 с.
3. Программирование микроконтроллерных плат Arduino/Freduino/ Улли Соммер, 2016.- 256 с.
4. GSM-модуль SIM800L: самый полный мануал (на примерах с Arduino Uno) [Электронный ресурс].- Режим доступа:
https://codius.ru/articles/GSM_модуль_SIM800L_часть_1
5. Arduino [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Arduino>

Приложение 1.

```
#include <Wire.h> // подключаем библиотеку Wire.h чтобы можно было читать и  
отправлять данные на внешние компоненты
```

```
#include <SoftwareSerial.h> // взаимодействие с GSM-модулем sim800l осуществляется  
по интерфейсу UART (Serial) при помощи специальных AT-команд. для реализации обмена  
по UART-интерфейсу на пинах, отличных от стандартных RX(0)/TX(1) (они как правило  
заняты подключением к компьютеру), подключаем стандартную библиотеку  
SoftwareSerial.h
```

```
SoftwareSerial sim800l(8, 9); // устанавливаем пин 8 - RX Arduino (TX SIM800L), пин 9  
- TX Arduino (RX SIM800L)
```

```
#include <LiquidCrystal_I2C.h> // подключаем библиотеку для работы с дисплеем
```

```
LiquidCrystal_I2C LCD(0x27,16,2); // LiquidCrystal_I2C ОБЪЕКТ ( АДРЕС_I2C ,  
КОЛ_СТОЛБЦОВ , КОЛ_СТРОК )
```

```
#define PIN_MQ2 A1 // имя пина для подключения датчика газа MQ2
```

```
int gas; // создаёт целочисленную переменную для газа
```

```
int water; // создаёт целочисленную переменную для воды
```

```
uint32_t myTimer1, myTimer2; // переменные таймеров, так как нужна разная частота  
опросов для реле/дисплея и отправки СМС
```

```
void SendSMS1()// отправка СМС при срабатывании датчика воды
```

```
{
```

```
Serial.println("Sending SMS..."); // печать текста в монитор порта
```

```
sim800l.print("AT+CMGF=1\r"); // выбирает формат SMS
```

```
delay(100); // задержка 0,1 с
```

```
sim800l.print("AT+CMGS="+79273768447"\r"); // отправка СМС на указанный  
номер
```

```
delay(300); // задержка 0,3 с
```

```

sim8001.print("leak water"); // тест сообщения

delay(300); // задержка 0,3 с

sim8001.print((char)26); // (требуется в соответствии с таблицей данных)

delay(300); // задержка 0,3 с

sim8001.println();

Serial.println("Text Sent."); // подтверждаем отправку СМС в монитор порта

delay(500); // задержка 0,5 с

}

void SendSMS2() // отправка СМС при срабатывании датчика газа

{

Serial.println("Sending SMS..."); // печать текста

sim8001.print("AT+CMGF=1\r"); // выбирает формат SMS

delay(100); // задержка 0,1 с

sim8001.print("AT+CMGS=\"+79273768447\"\r"); // отправка СМС на указанный
номер

delay(300); // задержка 0,3 с

sim8001.print("leak gas"); // тест сообщения

delay(300); // задержка 0,3 с

sim8001.print((char)26); // (требуется в соответствии с таблицей данных)

delay(300); // задержка 0,3 с

sim8001.println();

Serial.println("Text Sent."); // подтверждаем отправку СМС в монитор порта

delay(500); // задержка 0,5 с

}

void setup() {

```

```

pinMode(4, OUTPUT); // назначаем используемые пины на плате

pinMode(PIN_MQ2, INPUT); // назначаем используемые пины на плате

pinMode(A5, INPUT); // назначаем используемые пины на плате

pinMode(3, OUTPUT); // назначаем используемые пины на плате

pinMode(A4, INPUT); // назначаем используемые пины на плате

pinMode(A0, INPUT); // назначаем используемые пины на плате

sim800l.begin(9600); // инициализация последовательной связи с Arduino и SIM800L

Serial.begin(9600); // инициализация последовательной связи с Arduino и Arduino IDE
(Serial Monitor)

delay(1000); // задержка 1 с

    if (sim800l.available()) { // переадресация с последовательного порта SIM800L на
последовательный порт Arduino IDE

        Serial.write(sim800l.read());

    }}

void loop() {

    water = analogRead(A0); // записываем полученные данные с датчика воды

    gas = analogRead(A1); // записываем полученные данные с датчика газа

    Serial.println("GAS - " + String(gas)); // выводим информацию на монитор порта

    Serial.println(" ");

    if (millis() - myTimer1 >= 500) { // таймер на 0,5 с для дисплея и реле

        myTimer1 = millis(); // сброс таймера

            // работа с дисплеем

        LCD.init(); // инициализируем работу с LCD дисплеем

```

```
LCD.backlight(); // включаем подсветку дисплея

LCD.setCursor(0, 0); // устанавливаем курсор в позицию (0 столбец, 0 строка)

LCD.print("Leak:"); // выводим текст "Leak:", начиная с установленной позиции
курсора

    if (water > 100 && gas < 200) // если показание датчика воды больше нормального
значания, а датчика газа в норме (утечка воды)

        {LCD.setCursor(0, 1); // устанавливаем курсор в позицию (0 столбец, 1 строка)

LCD.print("Water");} // выводим текст "Water", начиная с установленной позиции
курсора

    if (gas > 200 && water < 100) // если показание датчика газа больше нормального
значания, а датчика воды в норме (утечка газа)

        {LCD.setCursor(0, 1); // устанавливаем курсор в позицию (0 столбец, 1 строка)

LCD.print("GAS");} // выводим текст "GAS", начиная с установленной позиции
курсора

    if (water >100 && gas > 200) // если показания датчиков газа и воды выше
нормальных значений (утечки газа и воды)

        {LCD.setCursor(0, 1); // устанавливаем курсор в позицию (0 столбец, 1 строка)

LCD.print("Water and GAS");} // выводим текст "Water and GAS", начиная с
установленной позиции курсора

    if (gas < 200 && water < 100) // если показание датчика воды и датчика газа в норме

        {LCD.setCursor(0, 1); // устанавливаем курсор в позицию (0 столбец, 1 строка)

LCD.print("NO");} // выводим текст "NO", начиная с установленной позиции
курсора - протечек нет
```

// управление реле

if (water < 100) { digitalWrite(3, LOW); } // если показания датчика воды в норме -
пин 3 нет сигнала

Serial.println(water); // выводим информацию по датчику воды на монитор порта

LCD.noDisplay();

LCD.display();

delay(500); // задержка 0,5 с

if (water > 100) { digitalWrite(3, HIGH); } // если показания датчика воды выше нормы
- сигнал на пин 3

else { digitalWrite(3, LOW); } // иначе - пин 3 нет сигнала

if (gas > 200) { digitalWrite(4, HIGH); } // если показания датчика газа выше нормы
- сигнал пин 4

else { digitalWrite(4, LOW); } // иначе - пин 4 нет сигнала

delay(1000); // задержка 1 с

}

if (millis() - myTimer2 >= 10000) { // таймер на 10 сек для отправки СМС

myTimer2 = millis(); // сброс таймера

if (water > 100){

SendSMS1(); } // если показания датчика воды выше нормы то
выполняется SendSMS1

if (gas > 200){

SendSMS2(); } // если показания датчика газа выше нормы то

выполняется SendSMS2

}

}

Рецензия

На проект ученика 7 ФМИ2 класса Година Сергея

На тему: «Датчик против протечек воды и газа»

В представленном проекте производится изучение основ работы с платформой Arduino через создание системы для обнаружения и устранения утечек воды и газа. Автор предлагает использовать платформу Arduino, которая широко применяется в области электроники и программирования.

Достоинства проекта:

1. **Актуальность.** Проблема утечек воды и газа является серьёзной угрозой безопасности жилых и промышленных помещений. Проект направлен на разработку системы, способной своевременно обнаруживать и устранять утечки, что может предотвратить аварии и спасти жизни людей.
2. **Использование платформы Arduino.** Arduino — это популярная платформа, которая позволяет создавать различные электронные устройства. Использование этой платформы в проекте делает его доступным для понимания и реализации.
3. **Практическая значимость.** Проект имеет практическую значимость, так как может быть использован для создания реальных систем обнаружения и устранения утечек. Это делает проект ценным для специалистов в области электротехники, программирования и безопасности.
4. **Перспективы развития.** Проект может стать основой для дальнейшего изучения возможностей платформы Arduino, а также разработок в области систем безопасности и автоматизации. Автор может продолжить работу над проектом, добавив новые функции и улучшив существующие.

В целом, проект представляет собой интересную идею, которая может быть реализована в виде реальной системы обнаружения и устранения утечек воды и газа.

Учитель информатики

ГБНОУ Пензенской области «Губернский лицей»



Гуляйкина А.И.