## Лабораторная работа 11: Работа с внешними модулями и датчиками

**Цель:**

* Изучить методы подключения и работы с различными датчиками.
* Научиться собирать и анализировать данные с датчиков.
* Ознакомиться с принципами работы и программирования внешних модулей.
* Создать устройство с использованием внешних модулей и датчиков.

## Теоретическая часть

### Использование датчиков

* Датчики - это устройства, преобразующие физические величины (температура, влажность, давление, движение) в электрические сигналы.
* Существует множество видов датчиков, каждый из которых предназначен для измерения определенной величины.
* Для подключения датчиков к микроконтроллерам используются специальные интерфейсы (например, I2C, SPI, аналоговый вход).
* Библиотеки Arduino предоставляют функции для работы с различными типами датчиков.

### Взаимодействие с внешними модулями

* Внешние модули - это устройства, которые расширяют функциональные возможности микроконтроллера.
* Модули могут подключаться к микроконтроллеру через различные интерфейсы (I2C, SPI, UART).
* Библиотеки Arduino предоставляют функции для работы с различными типами модулей.

### Использование датчиков

**Задание:**

* Подключить к Arduino датчик температуры, влажности и давления (например, BME280).
* Считать данные с датчиков и отобразить их на экране монитора.

**Решение:**

#include <Wire.h>  
#include <Adafruit\_BME280.h>  
  
#define BME280\_ADDRESS 0x76  
  
Adafruit\_BME280 bme280;  
  
void setup() {  
 Serial.begin(9600);  
  
 // Initialize BME280 sensor  
 if (bme280.begin(BME280\_ADDRESS)) {  
 Serial.println("Sensor connected.");  
 } else {  
 Serial.println("Sensor not found.");  
 while (1);  
 }  
}  
  
void loop() {  
 // Read sensor data  
 float temperature = bme280.readTemperatureC();  
 float humidity = bme280.readHumidity();  
 float pressure = bme280.readPressurePa();  
  
 // Print sensor data to serial monitor  
 Serial.print("Temperature: ");  
 Serial.print(temperature);  
 Serial.print(" °C");  
 Serial.print(" Humidity: ");  
 Serial.print(humidity);  
 Serial.print(" %");  
 Serial.print(" Pressure: ");  
 Serial.print(pressure);  
 Serial.println(" Pa");  
  
 delay(1000);  
}

### Взаимодействие с внешними модулями

**Задание:**

* Подключить к Arduino LCD-дисплей (например, 16x2 LCD) и клавиатуру.
* С помощью клавиатуры вводить текст, который будет отображаться на LCD-дисплее.

**Решение:**

#include <LiquidCrystal.h>  
  
#define LCD\_RS 12  
#define LCD\_EN 11  
#define LCD\_D4 5  
#define LCD\_D5 4  
#define LCD\_D6 3  
#define LCD\_D7 2  
  
LiquidCrystal lcd(LCD\_RS, LCD\_EN, LCD\_D4, LCD\_D5, LCD\_D6, LCD\_D7);  
  
char keys[] = {'1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '\*', '#', 'A', 'B', 'C', 'D', 'E', 'F'};  
  
char input[16] = "";  
int inputIndex = 0;  
  
void setup() {  
 Serial.begin(9600);  
  
 lcd.begin(16, 2);  
 lcd.clear();  
 lcd.print("Enter text:");  
}  
  
void loop() {  
 char key = readKey();  
  
 if (key) {  
 if (key == '\*' && inputIndex > 0) {  
 inputIndex--;  
 input[inputIndex] = 0;  
 lcd.setCursor(inputIndex, 1);  
 lcd.print(' ');  
 } else if (key == '#' && inputIndex < 16) {  
 lcd.setCursor(inputIndex, 1);  
 lcd.print(input[inputIndex]);  
 inputIndex++;  
 input[inputIndex] = '\0';  
 } else if (key >= '0' && key <= 'F' && inputIndex < 16) {  
 lcd.setCursor(inputIndex, 1);  
 lcd.print(key);  
 input[inputIndex] = key;  
 inputIndex++;  
 }  
 }  
}  
  
char readKey() {  
 for (int i = 0; i < sizeof(keys); i++) {  
 if (digitalRead(i) == LOW) {  
 return keys[i];  
 }  
 }  
 return 0;  
}

**Описание:**

* Подключены LCD-дисплей и клавиатура к определенным пинам Arduino.
* Определен массив keys для символов на кнопках клавиатуры.
* Переменная input хранит вводимый текст, а inputIndex - его текущую длину.
* Функция readKey() сканирует состояние пинов, подключенных к кнопкам клавиатуры, и возвращает соответствующий символ.
* В цикле loop() проверяется наличие нажатой клавиши.
* В зависимости от нажатой клавиши (\*, #, буквы/цифры) происходит управление курсором на LCD, печать символа и обновление переменной input.

## Задачи:

**Задачи:**

**1. Автоматическая система полива:**

* Подключить к Arduino датчик влажности почвы (например, YL-69) и насос.
* Написать программу, которая будет считывать показания датчика влажности и включать насос, когда влажность почвы падает ниже установленного уровня.
* Дополнительно можно реализовать:
  + Отображение информации о влажности почвы на LCD-дисплее.
  + Управление системой полива через веб-интерфейс или мобильное приложение.
  + Использование прогноза погоды для оптимизации режима полива.

**2. Метеостанция:**

* Подключить к Arduino датчики температуры, влажности, давления (например, BME280) и барометр.
* Считывать данные с датчиков и отображать их на LCD-дисплее.
* Сохранять данные в файл на SD-карте.
* Дополнительно можно реализовать:
  + Отображение данных в виде графиков и диаграмм.
  + Отправку данных на сервер для хранения и анализа.
  + Предупреждения о резких изменениях погодных условий.

**3. Система контроля доступа:**

* Подключить к Arduino RFID-модуль (например, RC522) и электромагнитный замок.
* Записать на RFID-метки идентификаторы пользователей.
* Написать программу, которая будет считывать идентификаторы с меток и открывать замок только для авторизованных пользователей.
* Дополнительно можно реализовать:
  + Ведение журнала доступа.
  + Удаленное управление системой контроля доступа через веб-интерфейс или мобильное приложение.
  + Интеграцию с системой видеонаблюдения.

## Примеры кода для лабораторной работы:

### 1. Автоматическая система полива

**Код:**

#include <SoilMoistureSensor.h>  
  
const int soilPin = A0; // Пин подключения датчика влажности  
const int relayPin = 10; // Пин подключения реле  
const int threshold = 40; // Уровень влажности, ниже которого включается насос  
  
SoilMoistureSensor soilSensor(soilPin);  
  
void setup() {  
 Serial.begin(9600);  
 pinMode(relayPin, OUTPUT);  
}  
  
void loop() {  
 // Считывание данных с датчика влажности  
 int moistureValue = soilSensor.readRaw();  
 float moisturePercent = soilSensor.convertRawToPercent(moistureValue);  
  
 // Отображение данных на Serial  
 Serial.print("Moisture: ");  
 Serial.print(moisturePercent);  
 Serial.println("%");  
  
 // Проверка уровня влажности  
 if (moisturePercent < threshold) {  
 // Включение насоса  
 digitalWrite(relayPin, HIGH);  
 Serial.println("Pump turned on.");  
 } else {  
 // Выключение насоса  
 digitalWrite(relayPin, LOW);  
 Serial.println("Pump turned off.");  
 }  
  
 delay(1000);  
}

**В этом решении:**

* Подключен датчик влажности к аналоговому пину A0.
* Реле подключено к пину 10.
* Установлен threshold - уровень влажности, ниже которого включается насос.
* В цикле loop() считываются данные с датчика, отображаются на Serial и проверяется уровень влажности.
* Если уровень влажности ниже threshold, включается насос, иначе - выключается.

### 2. Метеостанция

**Код:**

#include <Wire.h>  
#include <Adafruit\_BME280.h>  
#include <SD.h>  
  
#define BME280\_ADDRESS 0x76  
#define SD\_CHIP\_SELECT 10  
  
Adafruit\_BME280 bme280;  
  
File dataFile;  
  
void setup() {  
 Serial.begin(9600);  
  
 if (bme280.begin(BME280\_ADDRESS)) {  
 Serial.println("Sensor connected.");  
 } else {  
 Serial.println("Sensor not found.");  
 while (1);  
 }  
  
 if (!SD.begin(SD\_CHIP\_SELECT)) {  
 Serial.println("SD card initialization failed.");  
 while (1);  
 }  
  
 dataFile = SD.open("meteo\_data.txt", O\_RDWR | O\_CREAT | O\_TRUNC);  
 if (!dataFile) {  
 Serial.println("Failed to open data file.");  
 while (1);  
 }  
  
 lcd.begin(16, 2);  
 lcd.clear();  
 lcd.print("MeteoStation");  
}  
  
void loop() {  
 float temperature = bme280.readTemperatureC();  
 float humidity = bme280.readHumidity();  
 float pressure = bme280.readPressurePa();  
  
 String dataString = "Temperature: ";  
 dataString += String(temperature);  
 dataString += " °C Humidity: ";  
 dataString += String(humidity);  
 dataString += " % Pressure: ";  
 dataString += String(pressure);  
 dataString += " Pa";  
  
 lcd.setCursor(0, 0);  
 lcd.print(dataString);  
  
 dataFile.println(dataString);  
 dataFile.close();  
  
 delay(1000);  
}

**В этом решении:**

* Подключены датчики BME280, LCD-дисплей и SD-карта.
* Определен dataFile для хранения данных на SD-карте.
* В цикле loop() считываются данные с датчиков, форматируются в строку dataString и выводятся на Serial и LCD.
* dataString записывается в файл meteo\_data.txt на SD-карте.

**Дополнительно:**

* Можно отображать данные в виде графиков и диаграмм на LCD-дисплее.
* Можно отправлять данные на сервер для хранения и анализа.
* Можно реализовать систему оповещения о резких изменениях погодных условий.

### 3. Система контроля доступа

**Код:**

#include <SPI.h>  
#include <MFRC522.h>  
  
#define RFID\_RC522\_SDA 12  
#define RFID\_RC522\_SCK 13  
#define RFID\_RC522\_MOSI 11  
#define RFID\_RC522\_MISO 10  
#define RFID\_RC522\_RST 9  
  
MFRC522 rc522(RFID\_RC522\_SDA, RFID\_RC522\_SCK, RFID\_RC522\_MOSI, RFID\_RC522\_MISO, RFID\_RC522\_RST);  
  
int authorizedIDs[] = {12345, 54321}; // Массив авторизованных ID  
  
void setup() {  
 Serial.begin(9600);  
 SPI.begin();  
 rc522.PCD\_Init();  
}  
  
void loop() {  
 if (rc522.PICC\_IsReady()) {  
 // Считывание UID метки  
 MFRC522::MIFARE\_UID uid;  
 int status = rc522.PICC\_ReadUID(&uid);  
  
 if (status == MFRC522::STATUS\_OK) {  
 // Проверка UID метки  
 for (int i = 0; i < sizeof(authorizedIDs) / sizeof(int); i++) {  
 if (uid.UID[0] == authorizedIDs[i] && uid.UID[1] == authorizedIDs[i] >> 8 && uid.UID[2] == authorizedIDs[i] >> 16 && uid.UID[3] == authorizedIDs[i] >> 24) {  
 Serial.println("Authorized access.");  
 digitalWrite(relayPin, HIGH); // Включение электромагнитного замка  
 delay(1000);  
 digitalWrite(relayPin, LOW); // Выключение электромагнитного замка  
 break;  
 }  
 }  
  
 if (i == sizeof(authorizedIDs) / sizeof(int)) {  
 Serial.println("Unauthorized access.");  
 }  
 }  
 }  
}

**В этом решении:**

* Подключены RFID-модуль RC522 и электромагнитный замок.
* Определен authorizedIDs - массив авторизованных ID.
* В цикле loop() проверяется наличие RFID-метки.
* Считывается UID метки и сравнивается с authorizedIDs.
* Если UID совпадает, открывается электромагнитный замок.