**1 Лабораторна робота №1**

**РОЗРОБКА ПРИСТРОЇВ “ІНТЕРНЕТУ РЕЧЕЙ” В СЕРЕДОВИЩІ МОДЕЛЮВАННЯ PROTEUS**

**Мета роботи**

Ознайомитися із функціональними можливостями середовища віртуального моделювання Proteus. Ознайомитися із віртуальними моделями пристроїв “Інтернету речей”. Навчитися створювати схеми IoT-пристроїв і виконувати їх симуляцію. Отримати практичні навички відлагодження програм, написаних для мікроконтролерів AVR в середовищі Proteus.

**1.1 Короткі теоретичні відомості**

Розробником середовища віртуального моделювання Proteus є компанія Labcenter Electronics (Великобританія). Сайт розробника - http://www.labcenter.co.uk/. Відмінність цього середовища від аналогічних за призначенням (наприклад, Electronics Workbench, Multisim Ultiboard, MicroCap, Tina) полягає у розвиненій системі симуляції (інтерактивного налагодження як в режимі реального часу, так і покроково) для різних сімейств мікроконтролерів: Intel 8051, AVR і PIC (компанії Microchip), ARM (різних виробників) та ін.

Proteus має в своєму складі велику кількість як бібліотек окремих компонентів, так і периферійних пристроїв: світлодіодних і рідкокристалічних індикаторів, датчиків температури, годинників реального часу (RTC), інтерактивних елементів вводу/виводу (кнопок, перемикачів, віртуальних портів і віртуальних вимірювальних приладів), інтерактивних графічних об’єктів, які не завжди присутні в інших подібних програмах.

Середовище Proteus VSM складається із двох самостійних частин: ISIS і ARES. Основним призначенням підсистеми ARES є робота із проектування, трасування та виготовлення друкованих плат (PCB – Printed Circuit Board). Підсистема ISIS, що має безпосередній зв'язок із ARES, призначена для створення електричних принципових схем, написання та редагування коду програмованих пристроїв, а також моделювання і відлагодження їх роботи.

У спрощеному вигляді роботу в середовищі моделювання Proteus ISIS можна представити наступним алгоритмом:

1. Вибір необхідної елементної бази, створення схеми електричної принципової пристрою або системи, які розробляються.

2. Налаштування необхідних початкових параметрів компонентів для їх функціонування.

3. Розміщення необхідних віртуальних пристроїв (вимірювання та/або візуалізації), налаштування режимів їх роботи.

4. Написання та відлагодження програм для мікроконтролерів.

5. Симуляція або проведення спеціалізованого аналізу.

Більш детальне ознайомлення із можливостями середовища моделювання Proteus слід починати із ознайомлення із проектами-прикладами. Зокрема, в Proteus VSM реалізовано підтримку наборів інструментів для роботи із платформою Arduino. Це дозволяє розробляти прототипи проектів для Arduino безпосередньо в самому середовищі Proteus, тобто, редактор коду є вбудованим і не вимагає використання, наприклад, Arduino IDE. Така функція проекту є дуже корисною, оскільки проекти можна легко створити для плат Arduino різних версій.

**1.1 Результати виконання завдання**

Завдання для лабораторної роботи зображено в таблиці 1.1.1

Таблиця 1.1.1 – Завдання

|  |  |  |
| --- | --- | --- |
| № варіанту (бригади) | Конфігурація пристрою (компонент / кількість) | Примітки |
| 4 | BME280 / 1;  Servo / 3 | Кут повороту першого сервоприводу (-90°...+90°) визначається пропорційно виміряній температурі (-25°С...+65°С); другого – пропорційно виміряній відносній вологості (0%...90%); третього – пропорційно виміряному тиску (600 гПа...1050 гПа) |

Код програми наведено в лістингу 1.1.2

|  |
| --- |
| Лістинг 1.1.2 |
| #include <Adafruit\_I2CDevice.h> |
| #include <SPI.h> |
| #include <Wire.h> |
| #include <Adafruit\_Sensor.h> |
| #include <Adafruit\_BME280.h> |
| #include <Servo.h> |
|  |
|  |
| Adafruit\_BME280 bme; |
|  |
| Servo servo1; |
| Servo servo2; |
| Servo servo3; |
|  |
| int servo1Pin = 3; // Пін для першого сервоприводу |
| int servo2Pin = 5; // Пін для другого сервоприводу |
| int servo3Pin = 6; // Пін для третього сервоприводу |
|  |
| void setup() { |
| // Встановлюємо швидкість передачі даних через інтерфейс I2C |
| Wire.begin(9600); |
|  |
| bme.begin(0x76); // Адреса пристрою |
|  |
| // Налаштування пінів для сервоприводів |
| servo1.attach(servo1Pin); |
| servo2.attach(servo2Pin); |
| servo3.attach(servo3Pin); |
|  |
| } |
|  |
| void loop() { |
| // Зчитування значень температури, відносної вологості та тиску |
| float temperature = bme.readTemperature(); |
| float humidity = bme.readHumidity(); |
| float pressure = bme.readPressure() / 100.0F; |
|  |
| Serial.println(temperature); |
| Serial.println(humidity); |
| Serial.println(pressure); |
| // Обмеження значень температури, відносної вологості та тиску відповідно до вимог завдання |
| int servo1Angle = map(temperature, -25, 65, -90, 90); |
| int servo2Angle = map(humidity, 0, 90, 0, 90); |
| int servo3Angle = map(pressure, 600, 1050, 0, 180); |
|  |
| // Керування сервоприводами |
| servo1.write(servo1Angle); |
| servo2.write(servo2Angle); |
| servo3.write(servo3Angle); |
| delay(1000); // Затримка 1 секунда між кожним оновленням значень |
| } |

На рисунку 1 зображено схему пристрою

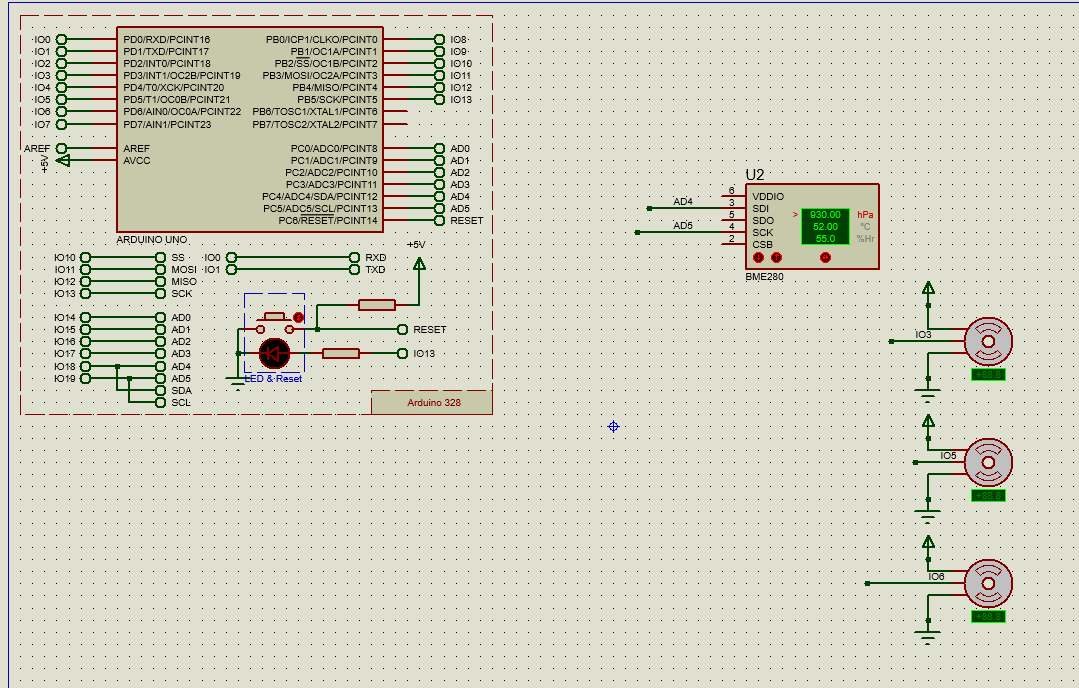
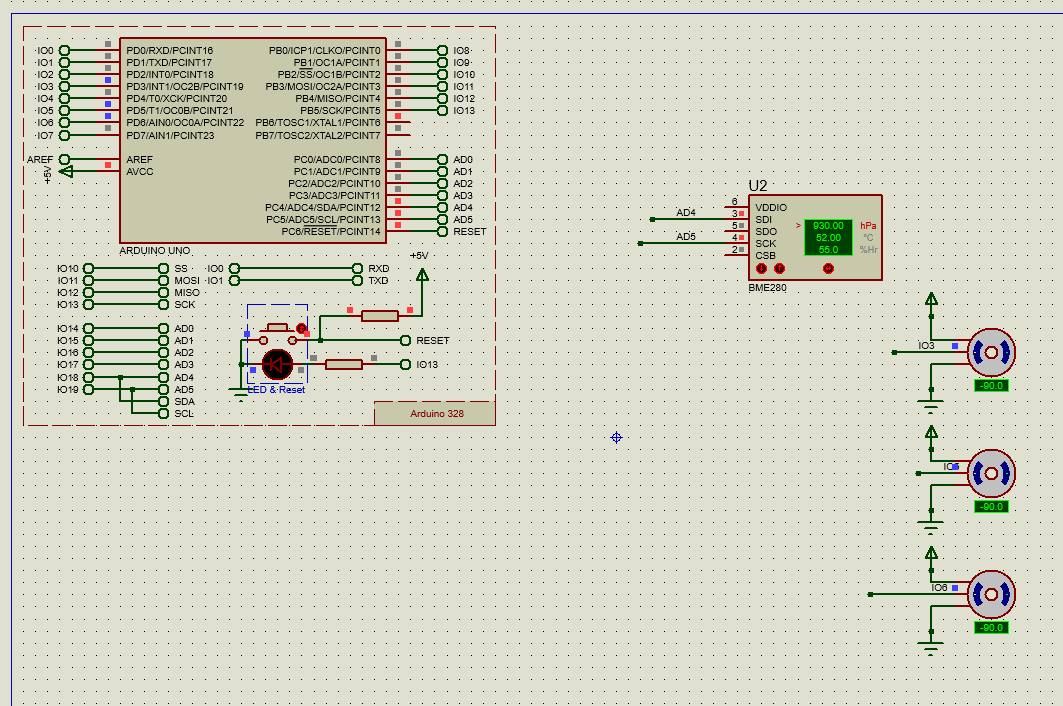


Рисунок 1 – Схема пристрою

На рисунку 2 та 3 зображено роботу пристрою.

пусто

Рисунок 2 – Поворот сервоприводу до +90

Рисунок 2 – Поворот сервоприводу до -90

**ВИСНОВКИ**

Ознайомився із функціональними можливостями середовища віртуального моделювання Proteus. Ознайомився із віртуальними моделями пристроїв “Інтернету речей”. Навчився створювати схеми IoT-пристроїв і виконувати їх симуляцію. Отримав практичні навички відлагодження програм, написаних для мікроконтролерів AVR в середовищі Proteus.