



07. Сетевое взаимодействие

Практические задания.

Цель: научиться разрабатывать программы, обменивающиеся данными по сети.

Задача: Разработать программу чтения данных в формате JSON по протоколу HTTP.

№ 07.0

1. Запустите редактор **Geany** из раздела «Программирование» в главном меню.
2. Откройте в редакторе **Geany** пример программы, читающей данные в формате JSON по протоколу HTTP (у модуля ESPEasy на базе ESP8266) `~/CodeClub-IoT/samples/http_json.rb`.
3. Уточните у преподавателя IP-адрес модуля и измените его в исходном тексте программы.
4. Запустите программу на выполнение из раздела меню «Сборка», пункт «Execute» (выполнить) и проверьте её работу. Закройте терминальное окно после завершения программы.

№ 07.1 - JSON

1. Доработайте программу **MeteoStation**, чтобы она запрашивала недостающие данные об атмосферном давлении от модуля ESPEasy на базе ESP8266 и выводила их на экран вместе с показаниями локального датчика DHT11.
2. Протестируйте программу.
3. Предусмотрите в программе обработку ситуации, когда удалённый источник данных (ESP8266) по какой-то причине недоступен.
4. Пример обработки таймаута — в программе `samples/http_timeout.rb`.

№ 07.2 - REST

1. Откройте в редакторе **Geany** программу `~/CodeClub-IoT/samples/http_server.rb`. Это пример простого сервера, который в цикле принимает запросы по протоколу HTTP и возвращает текущее время, оформляя его в виде правильного HTTP-ответа.
2. Запустите эту программу и проверьте, как она работает: обратитесь из браузера по адресу `http://localhost:5000`.
3. На основе этого примера доработайте программу **MeteoStation**: добавьте в неё метод `http_server`, который будет принимать запросы в соответствии с рекомендациями REST и выдавать текущие значения указанных датчиков:
`GET /sensor/temperature` → значение датчика температуры
`GET /sensor/humidity` → значение датчика влажности
`GET /sensor/pressure` → значение датчика атмосферного давления
4. Для выделения из запроса адреса (URL), удобно применить метод `split()`, который разделит строку запроса на части и поместит их в элементы массива:
`strings = request.split(" ")`
`# "GET /a/b HTTP/1.1\r\n" → ["GET", "/a/b", "HTTP/1.1"]`
5. Протестируйте, как поведёт себя ваша программа, если к ней обратятся с несуществующим адресом URL.
6. Измените программу **MeteoStation**, чтобы она принимала и другие управляющие команды. Например:
`PUT /station/stop` → закончить выполнение программы
7. Какие ещё команды полезно предусмотреть для управления метеостанцией?