

1. Подключение мультидатчика к web-приложению посредством USB-порта и BLE

Работа с web-приложением производится посредством подключения мультидатчика как с помощью USB-порта, так и посредством подключения его через BLE.

Для работы с мультидатчиком через USB-порт требуется реализовать подключение с помощью встроенной в Arduino/ESP 32 библиотеки Serial следующим образом:

```
void setup()
{
    ...
    Serial.begin(BAUD_RATE);
    ...
}
```

где **BAUD_RATE** - скорость передачи данных.

Для передачи пакетов данных по BLE нужно создать характеристики для приема и передачи данных. Характеристика, отвечающая за прием данных, должна иметь тип **WRITE**, а характеристика, отвечающая за передачу данных, - **NOTIFY**. Также нужно создать BLE сервер и BLE сервис. Библиотеки для работы с BLE представлены ниже в листинге.

```
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>

void initBle()
{
    // Create the BLE Device
    BLEDevice::init(BLE_NAME_DEVICE);
```

```

// Create the BLE Server
pBleServer = BLEDevice::createServer();
pBleServer->setCallbacks(new BLEServerStateCallbacks());

// Create the BLE Service
pBleService = pBleServer->createService(BLE_SERVICE_UUID);

// Create a BLE Characteristic
pTxCharacteristic = pBleService->createCharacteristic(
    BLE_CHARACTERISTIC_UUID_TX,
    BLECharacteristic::PROPERTY_NOTIFY
);
pTxCharacteristic->addDescriptor(new BLE2902());

pRxCharacteristic = pBleService->createCharacteristic(
    BLE_CHARACTERISTIC_UUID_RX,
    BLECharacteristic::PROPERTY_WRITE
);
pRxCharacteristic->setCallbacks(new BLEReceiveCallbacks());

// Start the service
pBleService->start();

// Start advertising
pBleServer->getAdvertising()->start();
}

```

Для генерации UUID можно использовать ресурс:
<https://www.uuidgenerator.net/>

2. Система команд для работы с мультидатчиком и отправка пакетов

Для работы с датчиком представлен определенный набор команд и правил отправки пакетов. Пакет со значениями, который передается на web-приложение, имеет вид:

KEYch0val0ch1val1...chNvalN\r\n

KEY - уникальный ключ мультидатчика;

| - обязательные пробелы между элементами строки;

chN - нумерация каналов;

valN - показания датчиков типа **double** с двумя знаками после запятой;

\r\n - обязательный элемент в конце строки для корректной обработки запроса web-приложением.

Пример команды:

MultiDatchik|ch0|1.20|ch1|-34.15|ch2|44.00\r\n

Строка должна передаваться как массив **char** символов. Для передачи по USB используется **Serial.print()**.

```
Serial.print(str.c_str()); // если str не является char
Serial.print(str); // если str является char
```

При передаче пакета по BLE нужно реализовать отдельную функцию. Пример реализации:

```
void printStringBle(BLECharacteristic * txCharactic,
std::string value)
{
    uint8_t* data = (uint8_t*)value.c_str();
    int length = value.length();

    txCharactic->setValue(data, length);
    txCharactic->notify();
}
```

Система команд для работы с web-приложением представлена ниже в таблице.

Команда	Описание	Примечание
rcc	Команда чтения конфигурации каналов датчика	Пример команды: rcc\r\n

		<p>Пример ответа: гсс 1 0 0 1 0\r\n 1 - задействован 0 - отключен</p> <p>При ошибке: гсс error -1\r\n (любое отрицательное число)</p>
wcc	Команда записи конфигурации каналов датчика	<p>Пример команды: wcc 0 1 1 0 0\r\n 1 - задействован 0 - отключен</p> <p>При ошибке: wcc error -1\r\n (любое отрицательное число; если 0 - успешно)</p>
tgx	Команда активации передачи устройством неградуированных/градуированных значений	<p>Пример команды: tgx 0\r\n - неград. значения // tgx 1\r\n - град. значения</p> <p>Пример ответа: {tgx 0} MultiSensor11 ch4 4.2 ch6 1.44 ch9 8 ch10 -2.19\r\n // {tgx 1} MultiSensor11 ch4 !4.2 ch6 !1.44 ch9 !8 ch10 !-2.19\r\n</p> <p>При ошибке: tgx error -1\r\n (любое отрицательное число; если 0 - успешно)</p>
rgt	Команда чтения градуировочных таблиц всех активных каналов	<p>Пример команды: rgt\r\n</p> <p>Пример ответа в общем виде: rgt ch0 x0₀ y0₀ x0₁ y0₁ ... ch1 x1₀ y1₀ ... \r\n где x - показание датчика , y - градуированное значение</p> <p>При ошибке: rgt ch1 error-1\r\n</p>

rgt ch <index>	Команда чтения градуировочной таблицы отдельного канала	<p>! Канал для чтения должен быть активным !</p> <p>Пример команды: rgt ch0\r\n</p> <p>Пример ответа: rgt ch0 1.1 1.8 2.4 3.6 4.3 7.8\r\n</p> <p>При ошибке: rgt ch0 error-1\r\n Может возникнуть, если канал не активен</p>
wgt ch <index>	Команда записи градуировочной таблицы канала	<p>Пример команды в общем виде: wgt ch<index> <x₀><y₀> <x₁><y₁> ... <x_{n-1}><y_{n-1}>\r\n где x - показание датчика, y - градуированное значение</p> <p>! Последовательность точек должна быть упорядочена (x - строго в порядке возрастания, y - строго в порядке возрастания или убывания) !</p> <p>При ошибке: wgt ch0 error -1\r\n (0 - если успешно)</p>
cgt	Команда сброса градуированных таблиц всех каналов. Выполняется для калибровки акселерометра.	<p>Пример команды: cgt\r\n</p> <p>Пример ответа: cgt ch0 error 0\r\n cgt ch1 error 0\r\n cgt ch4 error 0\r\n</p>
sm	Команда установки нового режима измерений	<p>Пример команды: sm 2\r\n</p> <p>Пример отчета: sm error 0\r\n</p> <p>При ошибке: sm error-1\r\n</p>
sm ch<index>	Команда установки нового режима измерений у канала	<p>Пример команды: sm ch2 1\r\n</p> <p>! После канала идет порядковый индекс нового режима измерений, начинающийся с нуля !</p> <p>Пример ответа:</p>

		sm ch2 error 0\r\n При ошибке: sm ch2 error-1\r\n
gm	Команда чтения текущего режима измерений	Пример ответа: gm 1\r\n ! После названия команды идет порядковый индекс режима измерений, начинающийся с нуля ! При ошибке: gm error -1\r\n
gm ch<index>	Команда чтения текущего режима измерений у канала	Пример команды: gm ch2\r\n Пример ответа: gm ch2 1\r\n ! После канала идет порядковый индекс режима измерений, начинающийся с нуля ! При ошибке: gm ch2 error-1\r\n
wlg*	Команда для записи измеренных значений в файл логирования * - вид записанных данных должен быть следующим: MultiDatchik t2.18 ch0 1.20 ... \r\n Добавляется значение времени, прошедшее с начала записи	Пример команды: wlg 1\r\n wlg 0\r\n ! Значение 1 соответствует началу записи в лог-файл, 0 - остановке записи ! Пример ответа: wlg error 0\r\n При ошибке: wlg error -1\r\n
rlg	Команда для получения состояния записи в файл логирования	Пример команды: rlg\r\n Пример ответа: rlg 1\r\n rlg 0\r\n ! Значение 1 - запись производится, 0 - запись не производится !

		<div>При ошибке:</div> <div>rlg error -1\r\n</div>
rlg 1(0)	Команда для запуска (1)/досрочной остановки (0) чтения записанных в файл логирования измеренных значений	<div>Пример команды:</div> <div>rlg 1\r\n</div> <div>Пример ответа:</div> <div>-Первое сообщение: rlg SIZE\r\n где SIZE - размер всех передаваемых измерений в байтах</div> <div>-Последующие сообщения: MultiDatchik t0.1 ch0 1.20 ... \r\n MultiDatchik t1.2 ch0 2.00 ... \r\n ...</div> <div>-Последнее сообщение: rlg error 0\r\n</div> <div>При ошибке:</div> <div>rlg error -1\r\n</div>
		<div>Пример команды:</div> <div>rlg 0\r\n</div> <div>Пример ответа:</div> <div>rlg error 0\r\n (при успешной остановке)</div> <div>При ошибке:</div> <div>rlg error -1\r\n</div>
Особые команды		
cmv	Команда для калибровки датчика кистевой силы	<div>Пример команды:</div> <div>cmv\r\n</div> <div>Пример ответа:</div> <div>-После отправки команды должен прийти ответ: cmv start-calibration\r\n</div> <div>-Дождаться команды: cmv place-known-weight\r\n</div> <div>-После этого вводится команда: cmv <вес>\r\n</div> <div>-При успешной калибровке: cmv success\r\n</div>

		При ошибке: cmv error -1\r\n
--	--	---------------------------------

3. Работа с энергонезависимой памятью мультидатчика

При работе с энергонезависимой памятью, в ней следует хранить состояния каналов (активен/неактивен), градуировочные значения каналов, режим чтения градуировочных/неградуировочных значений (tgv), а также режим измерений каналов.

Для работы с энергонезависимой памятью используются библиотеки **<EEPROM.h>** и **<Preferences.h>**. Для записи и чтения из энергонезависимой памяти используются методы **putBytes** и **getBytes** соответственно:

```
Preferences prefs;  
prefs.putBytes("tgv", &tgv, sizeof(uint8_t)); // запись  
prefs.getBytes("tgv", &tgv, sizeof(uint8_t)); // чтение
```