Домашняя ТехноСфера своими руками

Скетч для ESP-12F WeMos D1 WiFi

(версия 1.00 от 23.05.2023)

Оглавление

1 Общая информация

В этом Документе приведен исходный текст (и его краткое описание) шаблона скетча (далее − скетч) для «ESP-12F WeMos D1 WiFi» (далее – ESP).

В шаблоне реализован минимальный функционал, необходимый для обеспечения работы ESP и сопряженного с ним Arduino UNO (далее − Arduino) в рамках домашней ТехноСферы.

Важно!

На Arduino UNO должен быть установлен соответствующий скетч (совместимый с тем, что приведен в разделе «5 Текст скетча для ESP-12F WeMos D1 WiFi»).

1.1 Назначение скетча для ESP.

1. Обеспечение информационного обмена:
   1.1 ESP с внешней программой через Wi-Fi (по определенным правилам);
   1.2 ESP с сопряженным Arduino через "Tx/Rx";
   1.3 Внешней программы с сопряженным Arduino через ESP.
2. Выполнение (по запросу от внешней программы) дополнительных, специфичных (ориентированных на конкретную задачу) функций, определенных в скетче программистом.

1.2 Нюансы информационного обмена с внешней программой

Важно!

Для того, чтобы не связываться с нюансами парсинга HTTP-запросов принято, что в строке HTTP-запроса вместо символа «**точка с запятой**» используется цепочка символов **z9z**.

Пример:

```
http://192.168.0.103/serial_readln?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z pin_12 z9z pin_13 z9z pin_A2
```

1.3 Нюансы дополнительных, специфичных функций

Нюансы дополнительных, специфичных функций, определенных в скетче программистом.

1.3.1 Формат HTTP-запроса

HTTP-запрос имеет следующий формат (примеры):

`http://192.168.0.104/spec_func?func01=50`
`http://192.168.0.104/spec_func?func01=30,1000`
`http://192.168.0.104/spec_func?func01`

где

`http://192.168.0.104/` – адрес ESP (IP);

`spec_func` – команда на выполнение специфичной функции;

`func01` – имя (идентификатор) специфичной функции (должна быть определена в скетче);

`50` – входной параметр для функции **`func01`**;

`30,1000` – перечень входных параметров для функции **`func01`**;

1.3.2 Структура специфичной функции

Пример (шаблон) специфичной функции.

```
bool func01(String sParamIn)
{
  bool res=false; //значение, возвращаемое функцией

  //-----------------------------
  //Парсинг входных параметров sParamIn
  unsigned long dt;
  sParamIn.trim();
  dt = sParamIn.toInt();
  if (dt<=0) { dt = 50; }
  if (dt > 5000) { dt = 5000; }
  //-----------------------------

  //-----------------------------
  //Реализация функционала
  digitalWrite(D12, LOW);
  pinMode(D12, OUTPUT);
  digitalWrite(D12, LOW);
  digitalWrite(D12, HIGH);
  delay(dt);
  digitalWrite(D12, LOW);
  res = true;  //признак, что функция выполнена успешно
  //-----------------------------

  return res; //возвращаемое функцией значение
}
```

2 Перечень и назначение функций скетча

2.1 Предопределенные (обязательные) функции

2.1.1 Функции, вызываемые по запросу от внешней программы

В таблице ниже приведены функции, вызываемые по запросу от внешней программы.

Таблица 1 − Функции, вызываемые по запросу от внешней программы

Имя (идентификатор) функции	Назначение	Заголовок функции
1	2	3
`handleRoot`	Обработчик запроса клиента по корневому адресу. Возвращает список команд (функций), выполняемых по запросу от внешней программы	`void handleRoot()`
`who_is_it`	Возвращает уникальный идентификатор этого ESP	`void who_is_it()`
`serial_readln`	Получение данных от сопряженного Arduino	`void serial_readln()`
`serial_println`	Отсылку данных на сопряженный Arduino	`void serial_println()`
`serial_readln_delay`	Получение или изменение значения переменной `SERIAL_READLN_DELAY` (время ожидания ответа от Arduino (мсек). Минимальное значение = 500). См., также, раздел «5 -12»	`void serial_readln_delay()`
`pin_mode`	Установка режима заданного Пина (Input или Output)	`void pin_mode()`
`pin_read`	Получение состояния/значения заданного Пина	`void pin_read()`
`pin_write`	Изменение состояния/значения заданного Пина	`void pin_write()`
`analog_read`	Получение аналогового значения заданного Пина	`void analog_read()`
`analog_write`	Запись аналогового значения заданного Пина	`void analog_write()`
`led_builtin_sys`	Изменение значения переменной `LED_BUILTIN_SYS`. См., также, раздел «5 -12»	`void led_builtin_sys()`
`pins_vals_get`	Получение состояний/значений всех Пинов	`void pins_vals_get()`
`spec_func`	Выполнение спец. функции	`void spec_func()`

2.1.2 Вспомогательные функции

В таблице ниже приведены вспомогательные функции.

Таблица 2 − Вспомогательные функции

Имя (идентификатор) функции	Назначение	Заголовок функции
1	2	3
`setup0`	"Стандартные" Действия, связанные с инициализацией этого ESP	void setup0()
`serial_wait`	Ожидание ответа по последовательному соединению заданное время (dt) в миллисекундах	bool serial_wait(unsigned long dt)
`StringToPin`	Конвертация строкового значения в номер Пина	int StringToPin(String s)
`LED_BUILTIN_SYS_ON_OFF`	Включает/выключает Пин `LED_BUILTIN` в том случае, если переменная `LED_BUILTIN_SYS` равна `true`	void LED_BUILTIN_SYS_ON_OFF(bool itsON)
`handleNotFound`	Возвращает во внешнюю программу HTML-текст: `Ошибка 404 (некорректный URL)` в том случае, если в запросе была указана несуществующая функция (см. таблицу 1)	void handleNotFound()
`pin_val_get`	Получить значение заданного Пина	int pin_val_get(int pn, bool its_digit)

2.2 Специфичные функции, определенные программистом

См. раздел «1.3 Нюансы дополнительных, специфичных функций».

3 Обобщенный алгоритм работы скетча

На рисунке ниже представлен обобщенный алгоритм работы скетча.

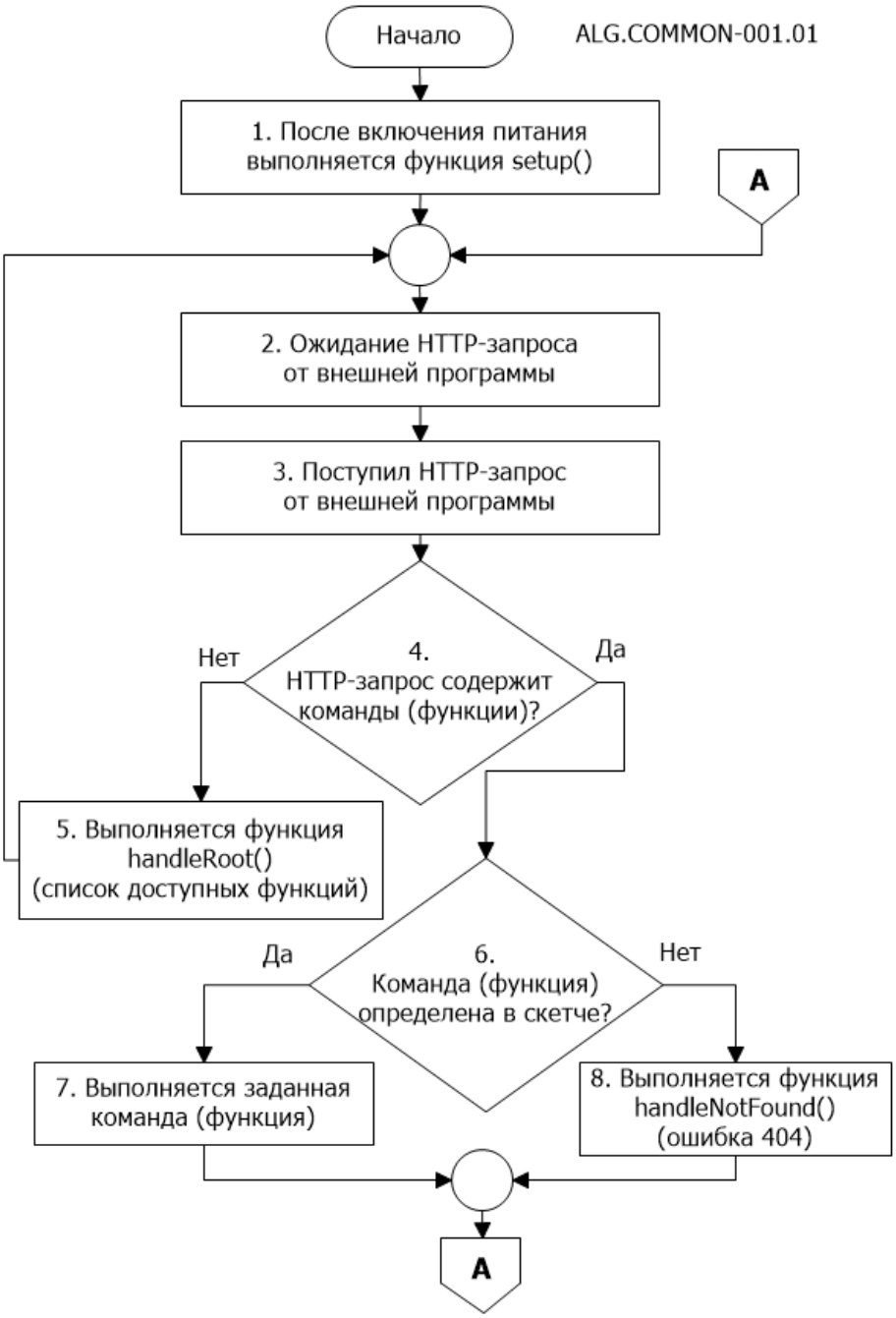

Рисунок 1 – Обобщенный алгоритм выполнения скетча: ALG.COMMON-001

4 Примеры вызова функций из внешней программы

4.1 ESP-12F WeMos D1 WiFi

1. Функция `handleRoot`
   `http://192.168.0.103`

2. Функция `who_is_it`
   `http://192.168.0.103/who_is_it`

3. Функция `led_builtin_sys`
   `http://192.168.0.103/led_builtin_sys?1`
   `http://192.168.0.103/led_builtin_sys?TRUE`
   `http://192.168.0.103/led_builtin_sys?0`
   `http://192.168.0.103/led_builtin_sys?FALSE`

4. Функция `pin_mode`
   `http://192.168.0.103/pin_mode?LED_BUILTIN=OUTPUT`
   `http://192.168.0.103/pin_mode?D8=OUTPUT`
   `http://192.168.0.103/pin_mode?D10=INPUT`

5. Функция `pin_read`
   `http://192.168.0.103/pin_read?LED_BUILTIN`
   `http://192.168.0.103/pin_read?D8`
   `http://192.168.0.103/pin_read?A0`

6. Функция `pin_write`
   `http://192.168.0.103/pin_write?LED_BUILTIN=LOW`
   `http://192.168.0.103/pin_write?LED_BUILTIN=HIGH`
   `http://192.168.0.103/pin_write?D8=HIGH`
   `http://192.168.0.103/pin_write?A0=183`

7. Функция `analog_read`
   `http://192.168.0.103/analog_read?A0`
   `http://192.168.0.103/analog_read?D12`

8. Функция `analog_write`
   `http://192.168.0.103/analog_write?D12=250`
   `http://192.168.0.103/analog_write?D11=150`
   `http://192.168.0.103/analog_write?D10=100`
   `http://192.168.0.103/analog_write?A0=155`

9. Функция `pins_vals_get`
   `http://192.168.0.103/pins_vals_get`

10. Функция `spec_func`

```
http://192.168.0.103/spec_func?kb_key_press=1000
http://192.168.0.103/spec_func?kb_key_press
http://192.168.0.103/spec_func?svetofor_RYG_blink
http://192.168.0.103/spec_func?svetofor_RYG_blink=500
http://192.168.0.103/spec_func?led_rgb_light=D12
http://192.168.0.103/spec_func?sound_alarm=1
http://192.168.0.103/spec_func?display_this=1,7,3,2,true
```

4.2 ESP-12F WeMos D1 WiFi + Arduino UNO

Важно!

Указанные ниже команды актуальны только в том случае, если на Arduino UNO установлен соответствующий скетч (совместимый с тем, что приведен в разделе «5 Текст скетча для ESP-12F WeMos D1 WiFi»).

1. Функция **serial_readln_delay**

```
http://192.168.0.103/serial_readln_delay
http://192.168.0.103/serial_readln_delay?val=2000
http://192.168.0.103/serial_readln_delay?val=500
```

2. Функция **serial_readln**

```
http://192.168.0.103/serial_readln
http://192.168.0.103/serial_readln?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z
pin_12 z9z pin_13 z9z pin_A2
http://192.168.0.103/serial_readln?device_all z9z info_all
http://192.168.0.103/serial_readln?device_all z9z pin_12 z9z pin_13 z9z
pin_A2
```

3. Функция **serial_println**

```
http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z
pin_12=HIGH z9z pin_13=1
http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z
pin_12=LOW z9z pin_13=0
http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z
pin_A0=117 z9z pin_A3=99
http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z
pin_12=1 z9z pin_A3=99
```

5 Текст скетча для ESP-12F WeMos D1 WiFi

Ниже приведен скетч для ESP-12F WeMos D1 WiFi:

```
/*
   Дата актуальности: 23.05.2023
   Шаблон скетча для "ESP-12F WeMos D1 WiFi" (далее - ESP).
   Назначение - обеспечение информационного обмена:
     1. ESP с внешней программой через Wi-Fi;
     2. ESP с сопряженным Arduino UNO через "Tx/Rx";
     3. Внешней программы с сопряженным Arduino UNO через ESP.
   Важно!
     Для того, чтобы не связываться с нюансами парсинга HTTP-запросов принято,
     что в строке HTTP-запроса вместо символа "точка с запятой" используется цепочка символов z9z
 */

#include <ESP8266WebServer.h>

//.................................
//Идентификация wi-fi сети и этого ESP
const char* ssid = "ИмяТвоейСети"; // Указываем имя существующей точки доступа (wi-fi сеть)
const char* password = "ПарольКСети"; // Указываем пароль существующей точки доступа (wi-fi сеть)
const String id_this = "ED0EBECDA1A74DDB8B666351F57734D2"; // Уникальный ID этого ESP
//.................................
//.................................
//"Настроечные" переменные
bool ItsMode_Debug = false;      //Режим отладки. Значение true актуально только, если этот ESP подсоединен к ПК через
USB (COM-порт)
bool LED_BUILTIN_SYS = true;     //Флаг, определяющий режим использования встроенного светодиода
int SERIAL_READLN_DELAY = 5000;  //Время ожидания ответа от Arduino (мсек). Минимальное значение = 500 (см. ниже)
//.................................
//.................................
//Вспомогательные переменные (используются в оперативных целях)
String param0_name = "";
String param0_val = "";
String sInfo = "";
int pin_num = 0;
bool itsok=false;
int va = 0;
int v = LOW;
```

```
//................................

ESP8266WebServer server(80);

void setup(void)
{
  // ----------------------------------------
  //"Стандартные" Действия, связанные с инициализацией устройства
  setup0();
  // ----------------------------------------

  // ----------------------------------------
  // Обработчики HTTP-запросов
  //Отработка запроса на получение общей информации
  server.on("/", handleRoot);
  //Отработка запроса на получение ID этого ESP
  server.on("/who_is_it", []() { who_is_it(); });
  //Отработка запроса на изменение значения переменной LED_BUILTIN_SYS (см. выше)
  server.on("/led_builtin_sys", []() { led_builtin_sys(); });
  //Отработка запроса на установку режима заданного Пина (Input или Output)
  server.on("/pin_mode", []() { pin_mode(); });
  //Отработка запроса на получение состояния всех Пинов
  server.on("/pins_vals_get", []() { pins_vals_get(); });
  //Отработка запроса на изменение состояния заданного Пина
  server.on("/pin_write", []() { pin_write(); });
  //Отработка запроса на получение состояния заданного Пина
  server.on("/pin_read", []() { pin_read(); });
  //Отработка запроса на запись знечения для заданного Пина (аналоговый)
  server.on("/analog_write", []() { analog_write(); });
  //Отработка запроса на считывание знечения для заданного Пина (аналоговый)
  server.on("/analog_read", []() { analog_read(); });
  //Отработка запроса на выполнение спец. функции
  server.on("/spec_func", []() { spec_func(); });
  //Отработка запроса на получение или изменение значения переменной SERIAL_READLN_DELAY  (см. выше)
  server.on("/serial_readln_delay", []() { serial_readln_delay(); });
  //Отработка запроса на отсылку данных на сопряженный Arduino
  server.on("/serial_println", []() { serial_println(); });
  //Отработка запроса на получение данных от сопряженного Arduino
  server.on("/serial_readln", []() { serial_readln(); });
  // ----------------------------------------
```

```
  // Вызывается, когда обработчик не назначен
  server.onNotFound(handleNotFound);

  // Запуск сервера
  server.begin();
}

void setup0()
{
  /*
    "Стандартные" Действия, связанные с инициализацией этого ESP
  */
  //Установка режима встроенного светодиода: OutPut
  pinMode(LED_BUILTIN, OUTPUT);
  //"Выключить" встроенный светодиод
  digitalWrite(LED_BUILTIN, HIGH);

  //Инициализация последовательного соединения со скоростью 9600
  Serial.begin(9600);

  //Инициализация Wi-Fi модуля
  WiFi.mode(WIFI_STA); // Установка Wi-Fi модуля в режим клиента (STA)
  WiFi.begin(ssid, password); // Подключение к сети
  // Ожидание подключения к сети
  while (WiFi.status() != WL_CONNECTED) { delay(500); }

  //...................
  //Если включен режим отладки, то по последовательному соединению
  //отправляется соотв. информ. блок
  if (ItsMode_Debug==true)
  {
     Serial.println("");
     Serial.println("-----------------------");
     Serial.print("WI-FI network: ");
     Serial.println(ssid);
     Serial.print("IP address: ");
     Serial.println(WiFi.localIP());
     Serial.print("My ID: ");
     Serial.println(id_this);
     Serial.println("-----------------------");
  }
```

```
  //...................
}

void loop(void)
{
  server.handleClient();
}

void handleRoot()
{
  // Обработчик запроса клиента по корневому адресу (возвращается список команд)
  server.send(200, "text/plain", id_this+":\n"+"Commands: \n who_is_it \n pin_mode \n pin_write \n pin_read \n
analog_write \n analog_read \n pins_vals_get \n led_builtin_sys \n spec_func \n serial_println \n serial_readln \n
serial_readln_delay");
}


//*****************************************************
//*****************************************************
//*****************************************************
// С П Е Ц И Ф И К А
void spec_func()
{
    //Запрос на выполнение спец. функции
    LED_BUILTIN_SYS_ON_OFF(true);
    param0_name = (String) server.argName(0);
    param0_val = (String) server.arg(0);
    itsok=false;

    //*****************************************************************
    //Вызовы спец.функций
    //-----------------------------------------
    //Пример (команда на выполнение функции func01).
    //http://192.168.0.104/spec_func?func01=1000
    //http://192.168.0.104/spec_func?func01
    if (param0_name == "func01")
    {
       itsok=true;
       if (func01(param0_val) == true)
       {
           server.send(200, "text/plain", id_this + ": "+"OK; "+"spec_func"+"; "+param0_name+";");
```

```
       }
       else
       {
          server.send(200, "text/plain", id_this + ": "+"Error. spec_func. "+param0_name+". "+"Здесь - сообщение об
ошибке");
       }
    }
    //---------------------------------
    //*****************************************************************

    //*****************************************************************
    //Ответ "ESP-12F WeMos D1 WiFi",  если требуемая спец.функция не найдена
    if (itsok == false)
    {
       server.send(200, "text/plain", id_this + ": "+"Error. spec_func. "+param0_name+". Undefined");
    }
    //*****************************************************************

    LED_BUILTIN_SYS_ON_OFF(false);
}

//---------------------------------
bool func01(String sParamIn)
{  //Пример спец.функции ("мигнуть" светодиодом на D12)
  bool res=false;
  unsigned long dt;
  sParamIn.trim();
  dt = sParamIn.toInt();
  if (dt<=0) { dt = 100; }
  if (dt > 1000) { dt = 1000; }
  digitalWrite(D12, LOW);
  pinMode(D12, OUTPUT);
  digitalWrite(D12, LOW);
  digitalWrite(D12, HIGH);
  delay(dt);
  digitalWrite(D12, LOW);
  res = true;
  return res;
}
//---------------------------------
//*************************************************
```

```
//*****************************************************
//*****************************************************


//*****************************************************
bool serial_wait(unsigned long dt)
{ //Ожидание ответа по последовательному соединению заданное время (dt) в миллисекундах
  bool res=false;
  if (dt < 0) { dt = 0; }
  unsigned long t0 = millis();
  bool yes_exit = false;
  while (yes_exit == false)
  {
    if (Serial.available() > 0)
    {
       res = true;
       yes_exit = true;
    }
    else
    {
       if ((millis() - t0) > dt)
       {
       yes_exit = true;
       }
    }
  }
  return res;
}
//*****************************************************

//*****************************************************
int StringToPin(String s)
{ //Конвертация строкового значения в номер Пина
  int res;
  res = -1;
  s.trim();
  s.toUpperCase();
  if (s == "D0") { res = D0; };
  if (s == "D1") { res = D1; };
  if (s == "D2") { res = D2; };
  if (s == "D3") { res = D3; };
```

```
  if (s == "D4") { res = D4; };
  if (s == "D5") { res = D5; };
  if (s == "D6") { res = D6; };
  if (s == "D7") { res = D7; };
  if (s == "D8") { res = D8; };
  if (s == "D9") { res = D9; };
  if (s == "D10") { res = D10; };
  if (s == "D11") { res = D11; };
  if (s == "D12") { res = D12; };
  if (s == "D13") { res = D13; };
  if (s == "D14") { res = D14; };
  if (s == "D15") { res = D15; };
  if (s == "LED_BUILTIN") { res = LED_BUILTIN; };
  if (s == "2") { res = LED_BUILTIN; };
  if (s == "A0") { res = A0; };
  return res;
}
//*****************************************************

//*****************************************************
void LED_BUILTIN_SYS_ON_OFF(bool itsON)
{
  /*
    В начале каждого обработчика "зажжем" встроенный светодиод (itsON = true).
    А в конце каждого обработчика - "погасим" его (itsON = false).
  */
  if (itsON == true)
  {
     if (LED_BUILTIN_SYS == true) {  digitalWrite(LED_BUILTIN, LOW); }
  }
  else
  {
     if (LED_BUILTIN_SYS == true)
     {
        delay(200);
        digitalWrite(LED_BUILTIN, HIGH);
     }
  }
}
//*****************************************************
```

```
//*****************************************************
//*****************************************************
//*****************************************************
// Обработчики...
//=======================================
void handleNotFound()
{ // Ошибка 404 (некорректный URL)
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i = 0; i < server.args(); i++)
  {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", id_this+":\n"+message);
}
//=======================================

//=======================================
void who_is_it()
{ //Запрос на получение ID этого ESP
    //Пример:   http://192.168.0.102/who_is_it
    LED_BUILTIN_SYS_ON_OFF(true);
    server.send(200, "text/plain", id_this);
    LED_BUILTIN_SYS_ON_OFF(false);
}
//=======================================

//=======================================
void serial_readln()
{ //Запрос на получение данных от сопряженного Arduino
    /*
     Примеры:
        http://192.168.0.103/serial_readln
        http://192.168.0.103/serial_readln?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z pin_12 z9z pin_13 z9z pin_A2
        http://192.168.0.103/serial_readln?device_all z9z info_all
```

```
    http://192.168.0.103/serial_readln?device_all z9z pin_12 z9z pin_13 z9z pin_A2
*/
LED_BUILTIN_SYS_ON_OFF(true);
String scomm = "";
for (uint8_t i = 0; i < server.args(); i++)
{
  param0_name = (String) server.argName(i);
  param0_name.trim();
  if (param0_name != "") { param0_name += "="; } else { param0_name = " "; }
  param0_val = (String) server.arg(i);
  param0_val.trim();
  scomm += " " + param0_name + param0_val;
}
scomm.trim();
if (scomm != "")
{
   scomm.replace("=", " ");
   scomm.replace(" z9z ", "; ");
   scomm.trim();
}
if (scomm == "") {scomm = "device_all; info_all"; }
Serial.println("GET: "+scomm); //передача команды на сопряженное устройство (Arduino)
serial_wait(SERIAL_READLN_DELAY); //Ожидаем ответ Arduino заданное время
String sInfo = "";
while (Serial.available() > 0)
{
  sInfo += (char)Serial.read();
  delay(2);
}
sInfo.trim();
if (sInfo.length() < 3)
{
    sInfo="";
}
else
{
   String s = sInfo.substring(0, 2);
   if (s != "A:") { sInfo=""; }
}
if (sInfo.length() > 3)
{
```

```
      String s2 = sInfo.substring(sInfo.length()-2, sInfo.length());
      if (s2 != "]>") { sInfo=""; }
    }
    if (sInfo == "")
    {
      sInfo = "NO INFO";
    }
    else
    {
      sInfo.setCharAt(0, ' ');
      sInfo.setCharAt(1, ' ');
      sInfo.trim();
    }
    server.send(200, "text/plain", id_this + ": "+sInfo);
    LED_BUILTIN_SYS_ON_OFF(false);
}
//=========================================

//=========================================
void serial_println()
{ //Запрос на отсылку данных на сопряженный Arduino
    /*
     Примеры:
      http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z pin_12=HIGH z9z pin_13=1
      http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z pin_12=LOW z9z pin_13=0
      http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z pin_A0=117 z9z pin_A3=99
      http://192.168.0.103/serial_println?098BB20A0CEE4FBBAB7BDC71EB2A8778 z9z pin_12=1 z9z pin_A3=99
    */
    LED_BUILTIN_SYS_ON_OFF(true);
    String scomm = "";
    for (uint8_t i = 0; i < server.args(); i++)
    {
      param0_name = (String) server.argName(i);
      param0_name.trim();
      if (param0_name != "") { param0_name += "="; } else { param0_name = " "; }
      param0_val = (String) server.arg(i);
      param0_val.trim();
      scomm += " " + param0_name + param0_val;
    }
    scomm.trim();
    if (scomm != "")
```

```
    {
       scomm.replace(" z9z ", "; ");
       Serial.println("SET: "+scomm); //передача команды на сопряженное устройство (Arduino)
       server.send(200, "text/plain", id_this + ": "+"OK; "+"serial_println"+"; "+scomm);
    }
    else
    {
       server.send(200, "text/plain", id_this + ": "+"Error. serial_println. Command is bad");
    }
    LED_BUILTIN_SYS_ON_OFF(false);
}
//=========================================

//=========================================
void serial_readln_delay()
{ //Запрос на получение или изменение значения переменной SERIAL_READLN_DELAY (см. выше)
    /*
     Примеры:
       http://192.168.0.103/serial_readln_delay
       http://192.168.0.103/serial_readln_delay?val=2000
       http://192.168.0.103/serial_readln_delay?val=500 - минимально возможное значение
    */
    LED_BUILTIN_SYS_ON_OFF(true);
    param0_name = (String) server.argName(0);
    param0_val = (String) server.arg(0);
    param0_val.trim();
    if (param0_val == "")
    {
       server.send(200, "text/plain", id_this + ": "+"OK; "+"serial_readln_delay"+";
SERIAL_READLN_DELAY="+SERIAL_READLN_DELAY);
    }
    else
    {
       va = param0_val.toInt();
       if (va <500) { va = 500; }
       SERIAL_READLN_DELAY = va;
       server.send(200, "text/plain", id_this + ": "+"OK; "+"serial_readln_delay"+";
SERIAL_READLN_DELAY="+SERIAL_READLN_DELAY);
    }
    LED_BUILTIN_SYS_ON_OFF(false);
}
```

```
//========================================

//========================================
void pin_mode()
{ //Запрос на установку режима заданного Пина (Input или Output)
    /*
     Примеры:
       http://192.168.0.103/pin_mode?LED_BUILTIN=OUTPUT
       http://192.168.0.103/pin_mode?D8=OUTPUT
       http://192.168.0.103/pin_mode?D10=INPUT
    */
    LED_BUILTIN_SYS_ON_OFF(true);
    param0_name = (String) server.argName(0);
    param0_val = (String) server.arg(0);
    pin_num = StringToPin(param0_name);
    itsok=false;
    if (pin_num>=0)
    {
       param0_val.toUpperCase();
       //if (param0_val == "OUTPUT")
       if ((param0_val == "OUT") || (param0_val == "OUTPUT"))
       {
          pinMode(pin_num, OUTPUT);
          itsok=true;
       }
       //if (param0_val == "INPUT")
       if ((param0_val == "IN") ||  (param0_val == "INPUT"))
       {
          pinMode(pin_num, INPUT);
          itsok=true;
       }
       if (itsok == true)
       {
          server.send(200, "text/plain", id_this + ": "+"OK; "+"pin_mode"+"; "+pin_num+"; "+param0_val);
       }
       else
       {
          server.send(200, "text/plain", id_this + ": "+"Error. pin_mode. "+pin_num+". +"+"Pin mode is bad");
       }
    }
    else
```

```
    {
      server.send(200, "text/plain", id_this + ": "+"Error. pin_mode. "+param0_name+". "+"Pin number is bad");
    }
    LED_BUILTIN_SYS_ON_OFF(false);
}
//=========================================

//=========================================
void pin_read()
{ //Запрос на получение состояния заданного Пина
    /*
     Примеры:
       http://192.168.0.103/pin_read?LED_BUILTIN
       http://192.168.0.103/pin_read?D8
       http://192.168.0.103/pin_read?A0
    */
    LED_BUILTIN_SYS_ON_OFF(true);
    param0_name = (String) server.argName(0);
    pin_num = StringToPin(param0_name);
    if (pin_num>=0)
    {
       if (pin_num == A0)
       {
          va = analogRead(pin_num);
          server.send(200, "text/plain", id_this + ": "+"OK; "+"pin_read"+"; "+pin_num+"; "+va);
       }
       else
       {
          v = digitalRead(pin_num);
          server.send(200, "text/plain", id_this + ": "+"OK; "+"pin_read"+"; "+pin_num+"; "+v);
       }
    }
    else
    {
      server.send(200, "text/plain", id_this + ": "+"Error. pin_read. "+param0_name+". "+"Pin number is bad");
    }
    LED_BUILTIN_SYS_ON_OFF(false);
}
//=========================================

//=========================================
```

```
void pin_write()
{ //Запрос на изменение состояния заданного Пина
    /*
     Примеры:
       http://192.168.0.103/pin_write?LED_BUILTIN=LOW
       http://192.168.0.103/pin_write?LED_BUILTIN=HIGH
       http://192.168.0.103/pin_write?D8=HIGH
       http://192.168.0.103/pin_write?A0=183
    */
    LED_BUILTIN_SYS_ON_OFF(true);
    param0_name = (String) server.argName(0);
    param0_val = (String) server.arg(0);
    pin_num = StringToPin(param0_name);
    itsok=false;
    if (pin_num>=0)
    {
       param0_val.toUpperCase();
       if (pin_num == A0)
       {
           va = param0_val.toInt();
           analogWrite(pin_num, va);
           server.send(200, "text/plain", id_this + ": "+"OK; "+"pin_write"+"; "+pin_num+"; "+va);
       }
       else
       {
          if ((param0_val == "LOW") || (param0_val == "0"))
          {
             digitalWrite(pin_num, LOW);
             itsok=true;
             server.send(200, "text/plain", id_this + ": "+"OK; "+"pin_write"+"; "+pin_num+"; "+LOW);
          }
          if ((param0_val == "HIGH") || (param0_val == "1"))
          {
             digitalWrite(pin_num, HIGH);
             itsok=true;
             server.send(200, "text/plain", id_this + ": "+"OK; "+"pin_write"+"; "+pin_num+"; "+HIGH);
          }
          if (itsok == false)
          {
             server.send(200, "text/plain", id_this + ": "+"Error. pin_write. "+pin_num+". "+"Value is bad");
          }
```

```
       }
    }
    else
    {
      server.send(200, "text/plain", id_this + ": "+"Error. pin_write. "+param0_name+". "+"Pin number is bad");
    }
    LED_BUILTIN_SYS_ON_OFF(false);
}
//========================================

//========================================
void analog_read()
{ //Запрос на считывание значения заданного Пина
    /*
     Примеры:
       http://192.168.0.103/analog_read?A0
       http://192.168.0.103/analog_read?D12
    */
    LED_BUILTIN_SYS_ON_OFF(true);
    param0_name = (String) server.argName(0);
    pin_num = StringToPin(param0_name);
    if (pin_num>=0)
    {
       va = analogRead(pin_num);
       server.send(200, "text/plain", id_this + ": "+"OK; "+"analog_read"+"; "+pin_num+"; "+va);
    }
    else
    {
      server.send(200, "text/plain", id_this + ": "+"Error. analog_read. "+param0_name+". "+"Pin number is bad");
    }
    LED_BUILTIN_SYS_ON_OFF(false);
}
//========================================

//========================================
void analog_write()
{ //Запрос на запись знечения для заданного Пина
    /*
     Примеры:
       http://192.168.0.103/analog_write?D12=250
       http://192.168.0.103/analog_write?D11=150
```

```
       http://192.168.0.103/analog_write?D10=100
    */
    LED_BUILTIN_SYS_ON_OFF(true);
    param0_name = (String) server.argName(0);
    param0_val = (String) server.arg(0);
    pin_num = StringToPin(param0_name);
    itsok=false;
    if (pin_num>=0)
    {
       param0_val.toUpperCase();
       va = param0_val.toInt();
       analogWrite(pin_num, va);
       server.send(200, "text/plain", id_this + ": "+"OK; "+"analog_write"+"; "+pin_num+"; "+va);
    }
    else
    {
      server.send(200, "text/plain", id_this + ": "+"Error. analog_write. "+param0_name+". "+"Pin number is bad");
    }
    LED_BUILTIN_SYS_ON_OFF(false);
}
//========================================

//========================================
void led_builtin_sys()
{ //Запрос на изменение значения переменной LED_BUILTIN_SYS (см. выше)
    /*
     Примеры:
       http://192.168.0.103/led_builtin_sys?1
       http://192.168.0.103/led_builtin_sys?TRUE
       http://192.168.0.103/led_builtin_sys?0
       http://192.168.0.103/led_builtin_sys?FALSE
    */
    digitalWrite(LED_BUILTIN, LOW);
    param0_name = (String) server.argName(0);
    itsok=false;
    param0_name.toUpperCase();
    if ((param0_name == "FALSE") || (param0_name == "0"))
    {
       LED_BUILTIN_SYS = false;
       itsok=true;
    }
```

```
    if ((param0_name == "TRUE") || (param0_name == "1"))
    {
       LED_BUILTIN_SYS = true;
       itsok=true;
    }
    if (itsok == true)
    {
       server.send(200, "text/plain", id_this + ": "+"OK; "+"led_builtin_sys"+"; "+LED_BUILTIN_SYS);
    }
    else
    {
       server.send(200, "text/plain", id_this + ": "+"Error. led_builtin_sys. "+"Value is bad");
    }
    delay(200);
    digitalWrite(LED_BUILTIN, HIGH);
}
//========================================

//========================================
int pin_val_get(int pn, bool its_digit)
{ //Получить значение заданного Пина
  int res=-1;
  if (pn >= 0)
  {
     if (its_digit == true)
     {
        res = digitalRead(pn);
     }
     else
     {
        res = analogRead(pn);
     }
     String s = "";
     s = "pin_" + String(pn) + "=" + String(res) + "; ";
     sInfo = sInfo + s;
  }
  return res;
}
//========================================

//========================================
```

```
void pins_vals_get()
{
  //Отработка запроса на получение состояния всех Пинов
  //Пример:   http://192.168.0.102/pins_vals_get
  LED_BUILTIN_SYS_ON_OFF(true);
  sInfo = "";
  pin_val_get(D0, true);
  pin_val_get(D1, true);
  pin_val_get(D2, true);
  pin_val_get(D3, true);
  pin_val_get(D4, true);
  pin_val_get(D5, true);
  pin_val_get(D6, true);
  pin_val_get(D7, true);
  pin_val_get(D8, true);
  pin_val_get(D9, true);
  pin_val_get(D10, true);
  pin_val_get(D11, true);
  pin_val_get(D12, true);
  pin_val_get(D13, true);
  pin_val_get(D14, true);
  pin_val_get(D15, true);
  //pin_val_get(LED_BUILTIN, true);
  pin_val_get(A0, false);
  sInfo.trim();
  String sEndData = "]>";
  String sStartData = "<[";
  sInfo = sStartData + id_this+"; " + sInfo  + sEndData;
  server.send(200, "text/plain", id_this + ": "+sInfo);
  LED_BUILTIN_SYS_ON_OFF(false);
}
//=======================================

//*****************************************************
//*****************************************************
//*****************************************************
```