KAMAMI

KAmodESP32 POW 485 (PL)

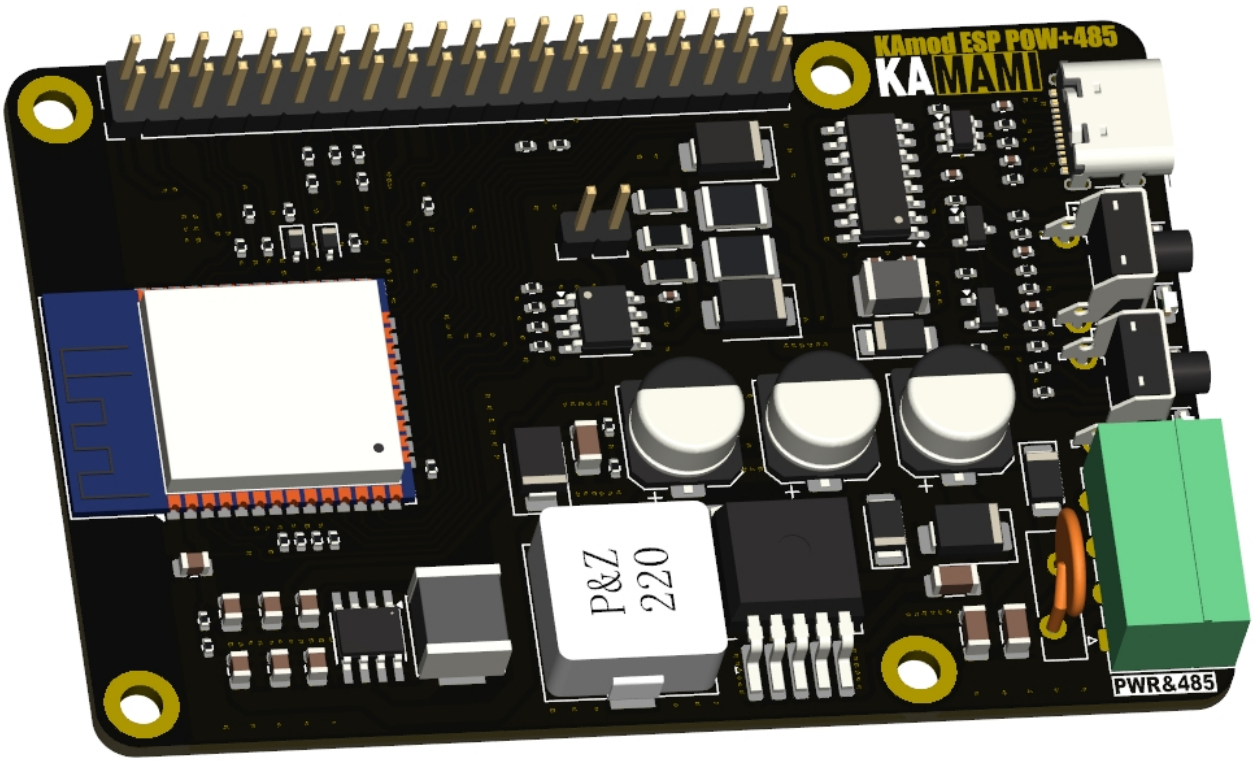
Rev. 20240813062928
Źródło: https://wiki.kamamilabs.com/index.php/KAmodESP32_POW_485_(PL)

Spis treści

Opis

Płytka ewaluacyjna z modułem Wi-Fi typu ESP32-WROOM, interfejsem RS485 oraz wydajnym układem zasilania.

Na płytce KAmod ESP32 POW+RS485 znajduje się moduł ESP32-WROOM umożliwiający komunikację w sieci bezprzewodowej Wi-Fi 2,4 GHz. Do programowania ESP32 zastosowano konwerter USB-UART ze złączem USB-C. Moduł jest połączony z interfejsem RS485 - popularnym rozwiązaniem w automatyce domowej i przemysłowej, umożliwiającym komunikację poprzez 2-żyłową skrętkę np. w standardzie MODBUS. Płytkę uzupełnia wydajny układ zasilania, który pracuje przy napięciu wejściowym z zakresu od 8 do 32 V, a dostarcza stabilizowanych napięć 5 V oraz 3,3 V o znacznej wydajności prądowej. Konstrukcja płytki odpowiada SBC rodziny Raspberry Pi - ma wymiary 81x56 mm, a na charakterystycznym, 40-stykowym złączu zostały wyprowadzone wszystkie istotne porty I/O oraz napięcia zasilające.



Podstawowe parametry

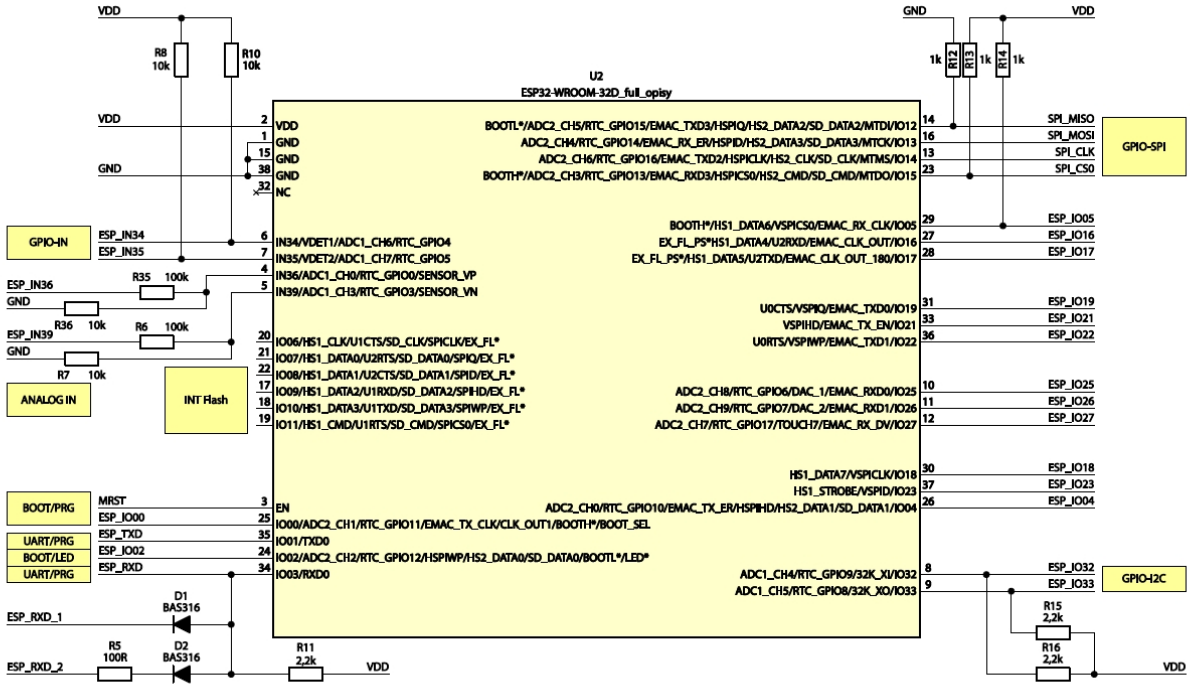
- Moduł ESP32-WROOM umożliwiający komunikację w sieci Wi-Fi w paśmie 2,4 GHz
- Zintegrowany konwerter UART-USB ze złączem USB-C umożliwiający programowanie układu ESP32
- Interfejs RS485 do komunikacji Half-duplex z maksymalną szybkością 1 Mbps
- Maksymalna liczba urządzeń dołączonych do magistrali RS485: 64
- Dostosowany do napięcia zasilającego z zakresu 8...32 V
- Dostarcza stabilizowanego napięcia 5,1 V o prądzie ciągłym do 3 A i krótkotrwałym do wartości 5 A
- Dostarcza stabilizowanego napięcia 3,3 V o prądzie ciągłym do 1 A i krótkotrwałym do wartości 2 A
- Zabezpieczenie przepięciowe, przeciążeniowe oraz termiczne
- Na 40-stykowe złącze w standardzie Raspberry Pi zostały wyprowadzone wszystkie istotne porty I/O oraz napięcia zasilające
- Wymiary płytki: 81x56 mm, wysokość ok. 20 mm

Wyposażenie standardowe

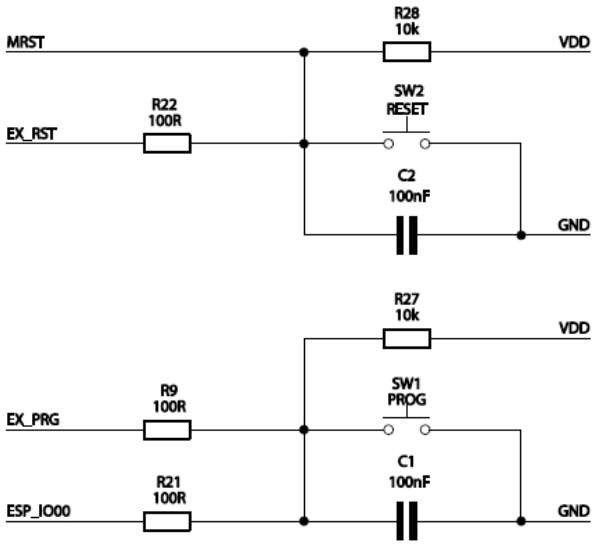
Kod	Opis
KAmod ESP32 POW+RS485	Zmontowany i uruchomiony moduł

Schemat elektryczny

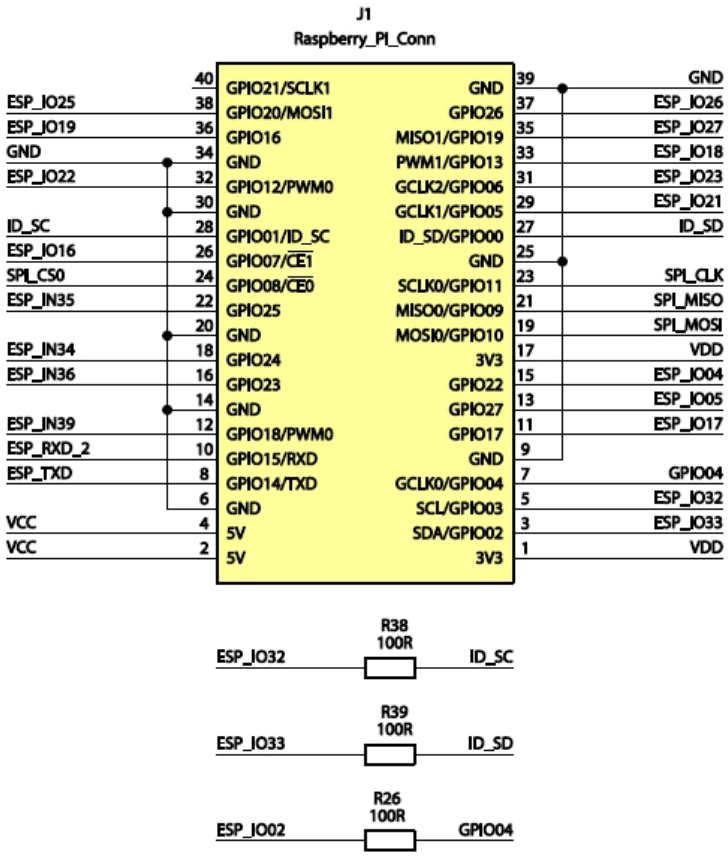
Moduł ESP32



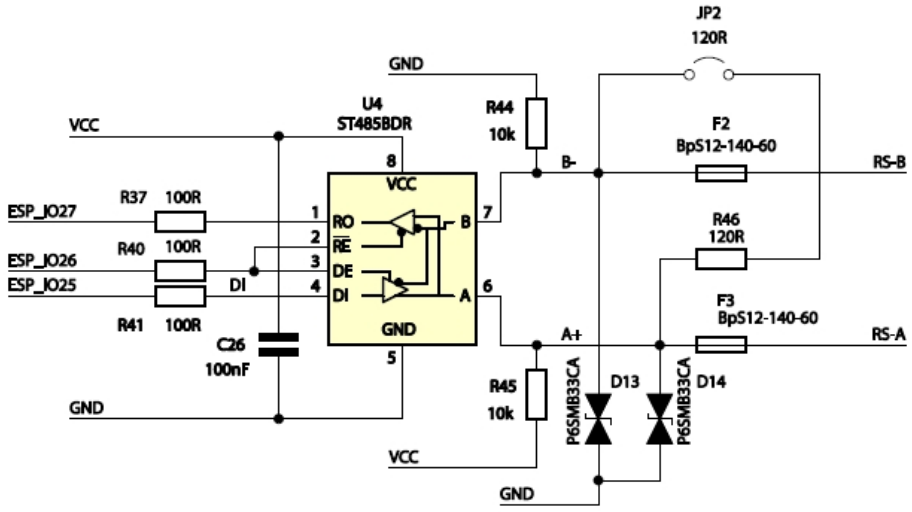
Elementy odpowiedzialne za funkcje resetu i programowania



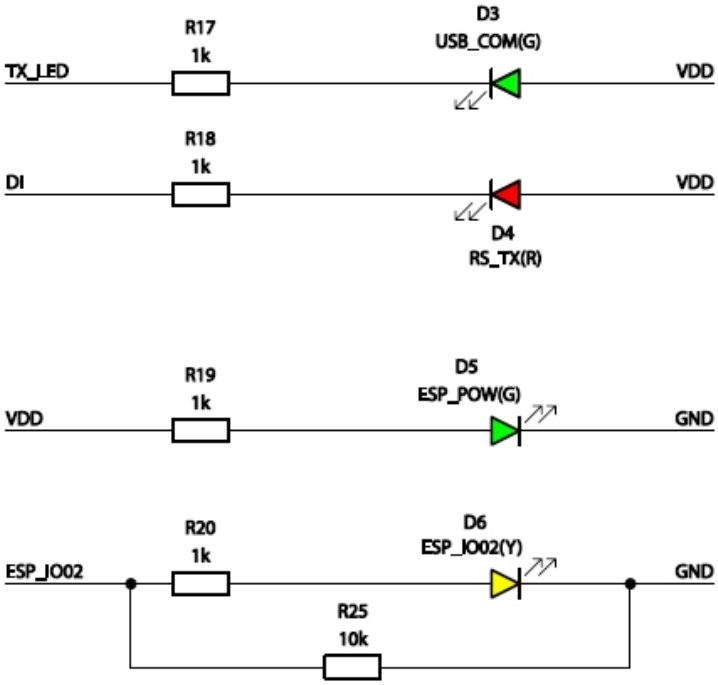
Złącze GPIO



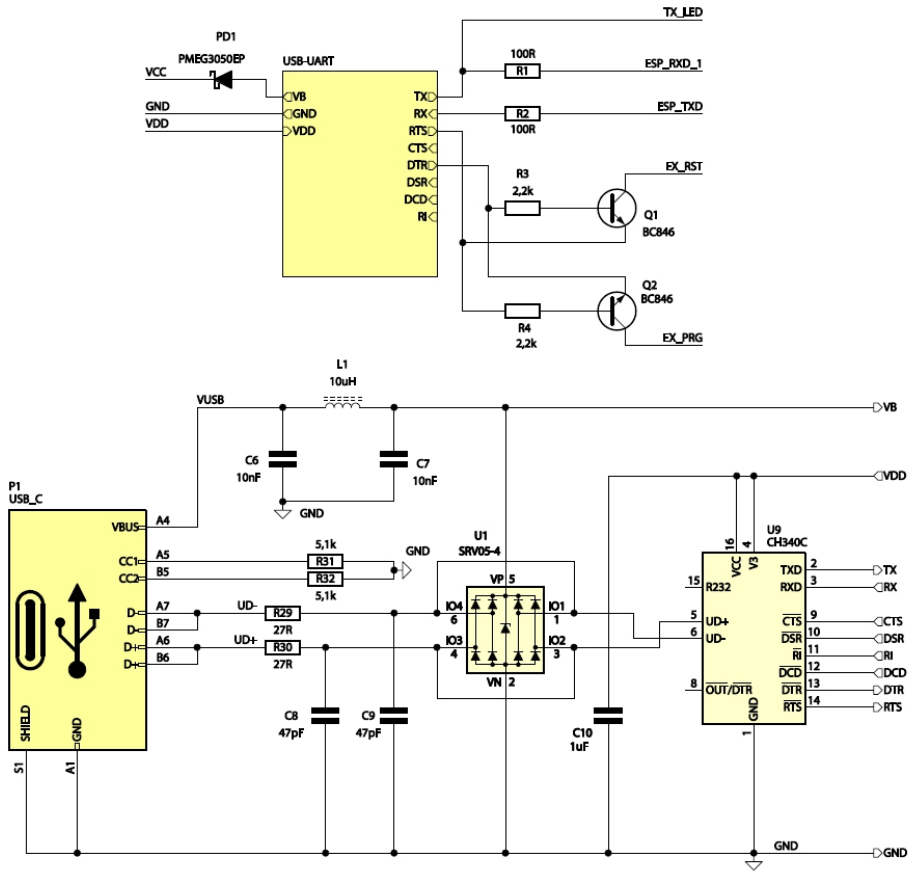
Interfejs RS485



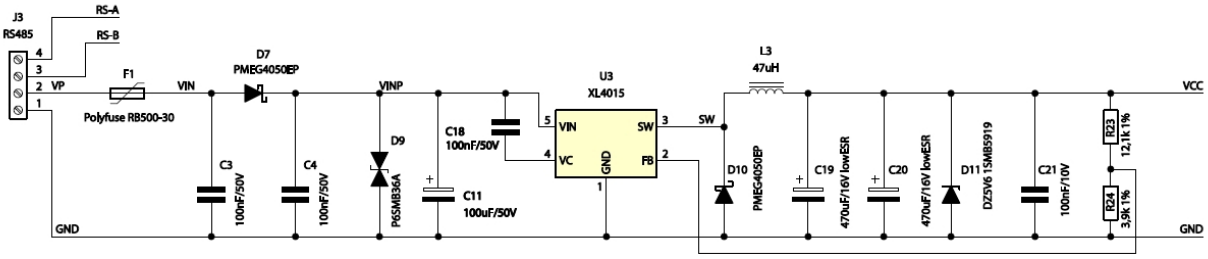
Diody sygnalizacyjne



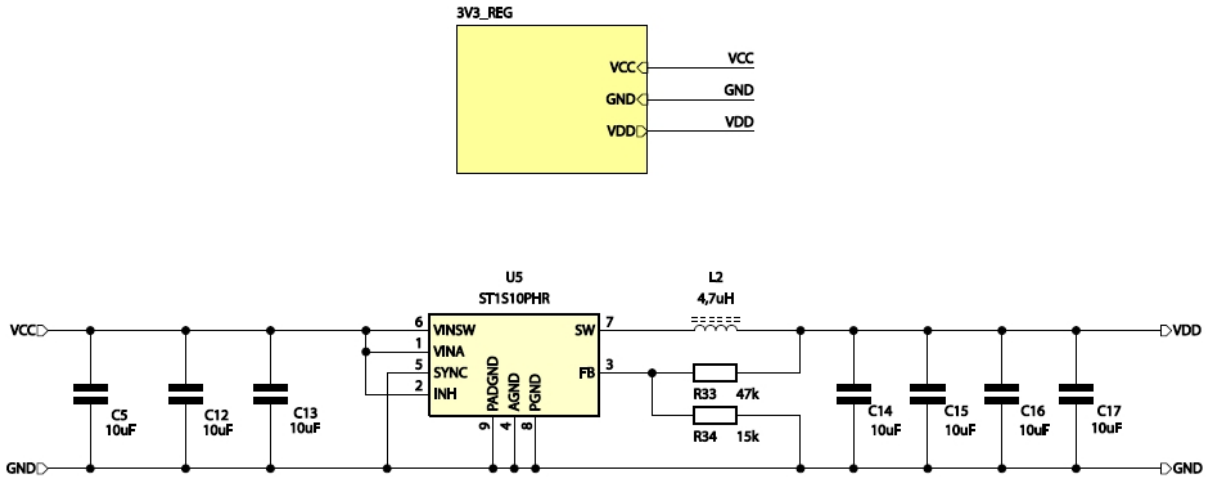
Interfejs USB-UART



Blok zasilania o napięciu 5 V



Blok zasilania o napięciu 3 V



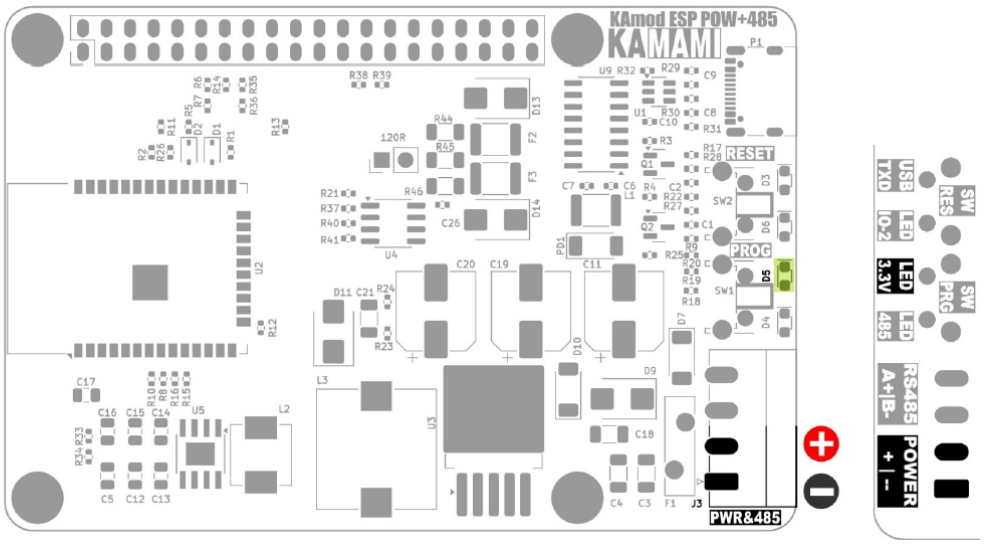
Złącze zasilania

Złącze	Funkcja
POW&485 Phoenix MC3,81 mm	• Doprowadza zasilanie do modułu • Złącze interfejsu RS485

Złącze **POW&485** (J3) pozwala na dołączenie napięcia stałego (DC) z zakresu 8...32 V, z którego jest wytwarzane napięcie 5,1 V do zasilania wszystkich komponentów płytki. Dołączając napięcie do złącza POW&485 należy zwrócić uwagę na jego prawidłową polaryzację. Oznaczenie na spodzie płytki: **POWER --|+** wskazuje prawidłową biegunowość zasilania:

- **--** - styk nr 1 to masa, ujemny biegun zasilania (GND),
- **+** - styk nr 2 jest wejściem dodatniego bieguna zasilania.

Dołączenie zasilania o parametrach wystarczających do uzyskania napięcia 3,3 V będzie sygnalizowane świeceniem diody LED D3. Nie należy dołączać napięcia o wartości powyżej 34 V. Moduł zawiera zabezpieczenie przeciw-przepięciowe, które odłączy zasilanie przy napięciu wyższym od ok. 34 V. Dołączone źródło zasilania powinno mieć odpowiednią moc. Aby moduł mógł działać z zachowaniem wszystkich parametrów (5,1 V/ 5 A), moc źródła zasilania nie powinna być mniejsza niż 30 W. Dołączenie zasilania o niższej mocy będzie skutkowało niższą wartością maksymalnego prądu w obwodzie napięcia 5 V oraz 3,3 V.



Interfejs RS485

Złącze	Funkcja
POW&485 Phoenix MC 3,81 mm	• Doprowadza zasilanie do modułu • Złącze interfejsu RS485

Na płytce KAmod ESP32 POW RS485 został zaimplementowany kompletny obwód interfejsu RS485 wraz z elementami zabezpieczającymi przed przepięciami. W roli sterownika zastosowano układ ST485, który umożliwia dołączenie maksymalnie 64 urządzeń, pracuje w trybie half-duplex, realizuje komunikację z maksymalną prędkością 1 Mbps. Wysyłanie danych na magistralę jest sygnalizowane miganiem diody LED D4.

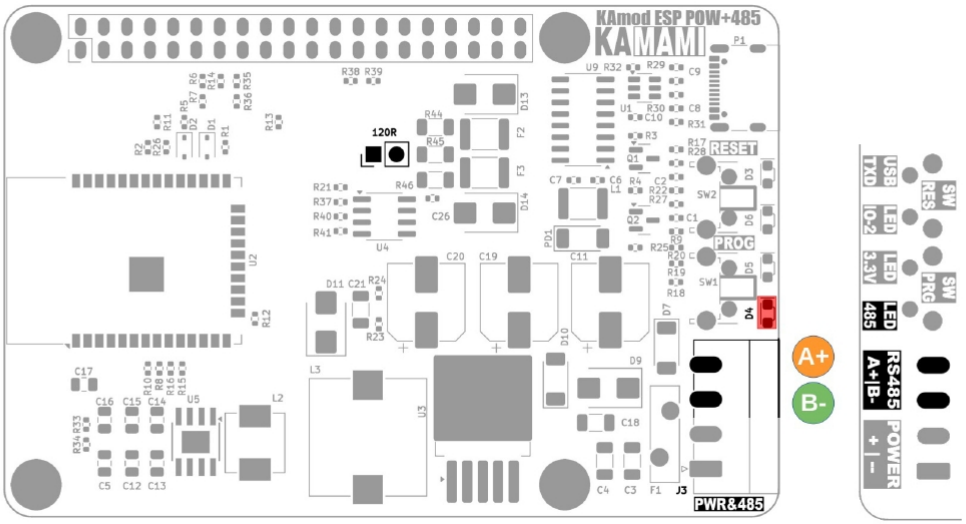
Połączenie interfejsu RS485 z modułem ESP32 jest następujące:

Sygnał RS485	Wyprowadzenie modułu ESP32
• RO (odczyt z magistrali RS485) • DI (wysyłanie na magistralę RS485) • DE/RE (sterowanie kierunkiem komuniakcji;H – nadawanie/L - odczyt)	• GPIO27 • GPIO25 • GPIO26

Medium transmisyjnym jest 2-żyłowa skrętka, którą należy dołączyć do złącza **POW&485** (J3):

- **B-**, styk nr 3, ujemny biegun magistrali RS485,
- **A+**, styk nr 4, dodatni biegun magistrali RS485.

Należy także zapewnić połączenie wszystkich urządzeń w magistrali do wspólnej masy/uziemienia. Specyfikacja magistrali RS485 wymaga, aby wszystkie połączenia tworzyły linię bez rozgałęzień i pętli, a na jej końcach należy dołączyć rezystory 120 Ω, które pełnią rolę tzw. terminatorów magistrali. Na płytce KAmod ESP32 POW RS485 znajduje się odpowiedni rezystor, który można dołączyć do magistrali zakładając zworkę na szpilki oznaczone **120R**.



Interfejs USB

Złącze	Funkcja
P1 USB-C	• Realizuje funkcję konwertera USB-UART • Umożliwia programowanie modułu ESP32 • Jest alternatywnym wejściem zasilania

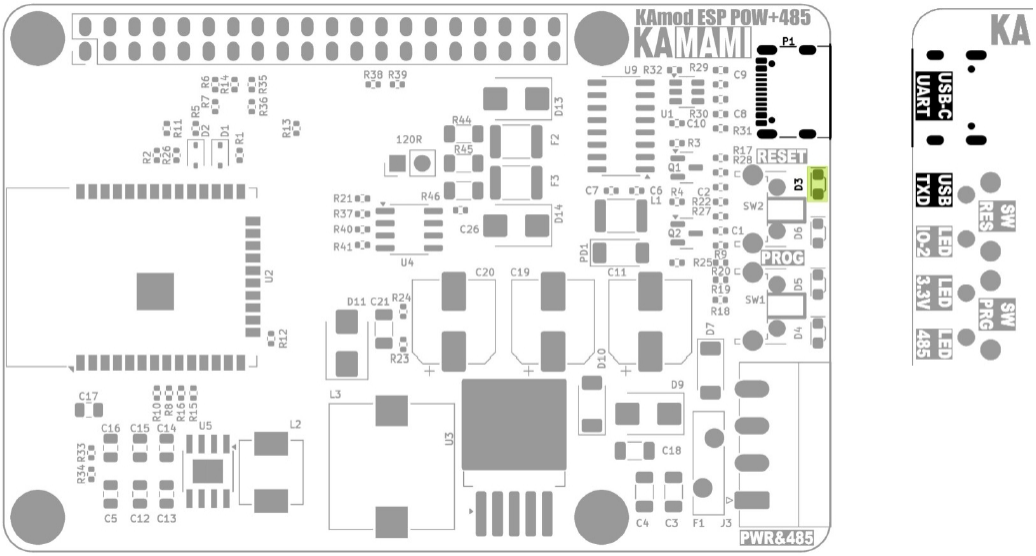
Złącze P1 typu USB-C jest połączone z kontrolerem typu CH340, który realizuje funkcje konwertera USB-UART. Interfejs UART może być używany w docelowej aplikacji, ale służy także do programowania modułu ESP32. Proces programowania może przebiegać całkowicie automatycznie, ponieważ kontroler CH340 steruje kluczowymi wyprowadzeniami modułu ESP32 (**GPIO0** – *Boot Select* oraz **EN** - *Chip Power-up*).

Połączenia sygnałów pomiędzy CH340 i ESP32 są następujące:

Sygnał kontrolera CH340	Wyprowadzenie modułu ESP32
• TXD (wyjście danych) • RXD (wejście danych) • DTR (wyście kontroli transmisji) • RTS (wyjście kontroli transmisji)	• GPIO03 (UART0 RXD) • GPIO01 (UART0 TXD) • EN (Chip Power-up) • GPIO0 (Boot Select)

Do linii TXD jest dołączona dioda LED (D3), która sygnalizuje odbieranie danych z interfejsu USB. W przypadku użycia w docelowej aplikacji konwertera USB-UART należy zadbać o to, aby linie DTR oraz RTS pozostały nieobsługiwane (*Handshaking: None*).

Złącze USB-C może służyć jako alternatywne wejście zasilania dla płytki KAmod ESP32 POW RS485, jednak wtedy parametry obwodów zasilania nie będą spełnione. Napięcie na linii 5 V, będzie niższe i będzie wynosiło ok. 4,5 V; napięcie na linii 3,3 V nie powinno się zmienić; wydajność prądowa napięć 5 V oraz 3,3 V będzie dużo niższa i będzie zależała od zastosowanego zasilania na złączu USB-C.

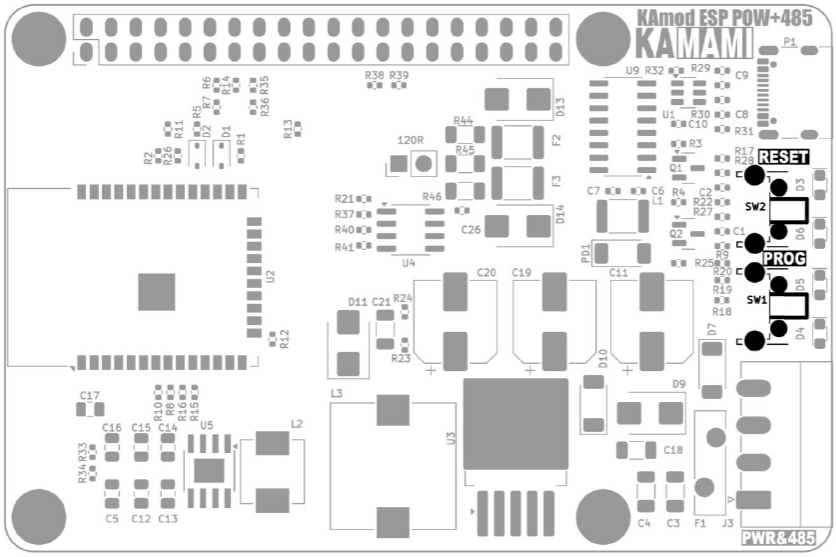


Przyciski resetowania i programowania

Komponent	Funkcja
• Przycisk SW1 – **PROG** • Przycisk SW2 – **RESET**	• Uruchamia tryb programowania poprzez UART (tylko w momencie restartu modułu ESP32) • Powoduje restart modułu ESP32

Przycisk RESET umożliwia wykonanie restartu modułu ESP32. Jest połączony z linią EN (*Chip Power-up*). modułu ESP32.

Przycisk PROG pozwala wprowadzić moduł ESP32 w tryb programowania. Należy wtedy nacisnąć przycisk RESET, następnie, trzymając wciśnięty RESET, przytrzymać przycisk PROG i wtedy zwolnić RESET, jednocześnie trzymając jeszcze przez chwilę wciśnięty PROG. Funkcjonalność ta może być przydatna, gdy z jakiegoś powodu tryb programowania nie będzie uruchamiany automatycznie poprzez konwerter USB-UART.

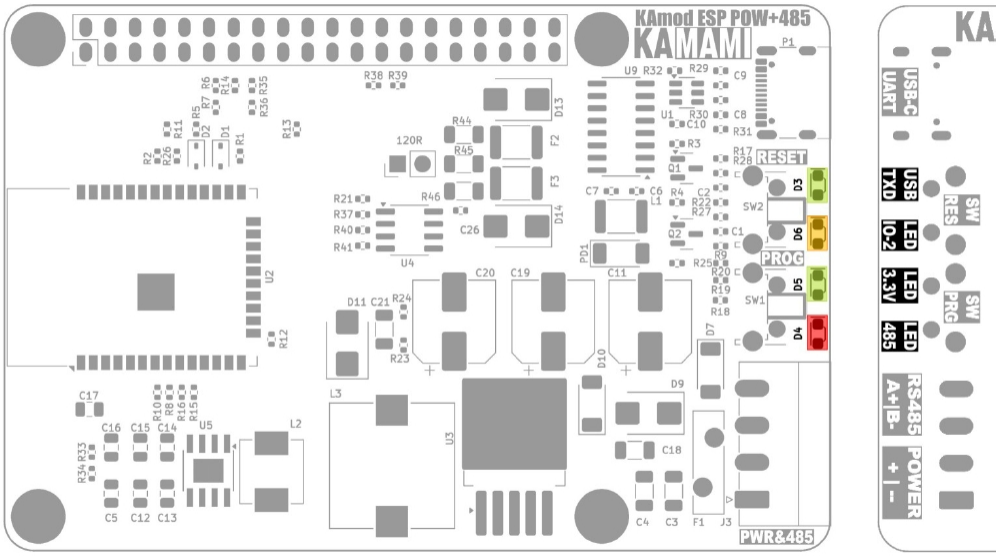


Kontrolki sygnalizacyjne

Komponent	Funkcja
• D3- **USB TXD** • D4- **LED 485** • D5- **LED 3.3V** • D6- **LED IO-2**	• Miganie diody D3 oznacza przesyłanie danych z USB do modułu ESP32 • Miganie diody D4 oznacza wysyłanie danych na magistralę RS485 • Świecenie diody D5 oznacza działanie obwodu zasilania i obecność napięcia 3,3 V • Dioda D6 jest dołączona do wyprowadzenia GPIO2 modułu ESP32 i jej świecenie może być wyzwalane programowo

Na płytce KAmod ESP32 POW RS485 znajdują się 4 diody LED, które sygnalizują działanie różnych komponentów – zgodnie z powyższą tabelą.

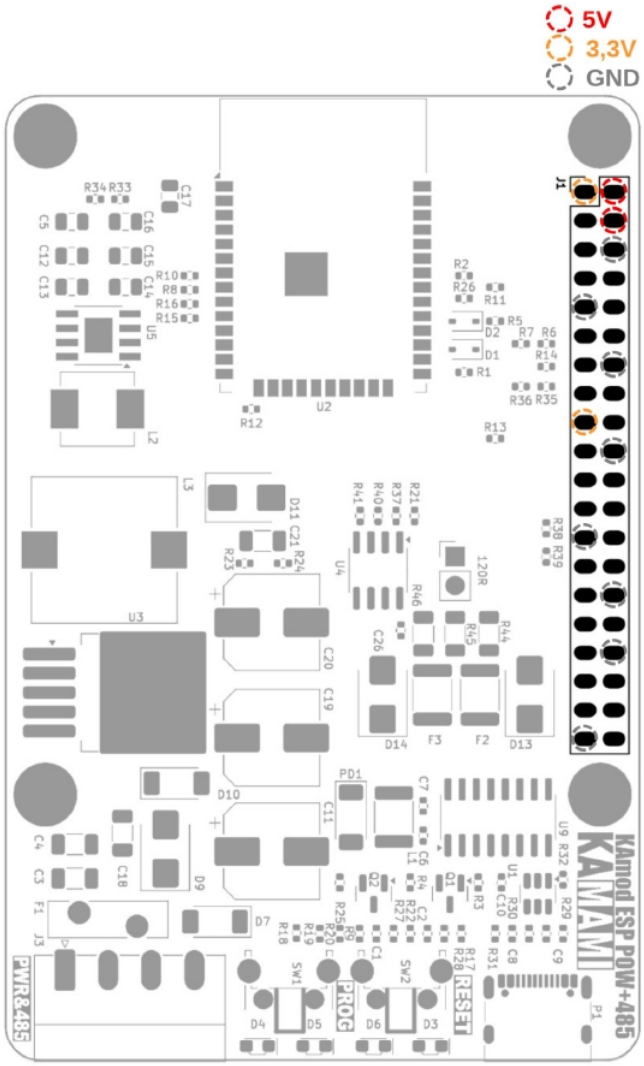
Dioda D6 (LED IO-2) jest dołączona do wyprowadzenia GPIO2 modułu ESP32. Jej zaświecenie wymaga programowego ustawienia stanu wysokiego na wyprowadzeniu GPIO2



Złącze GPIO w standardzie RPi

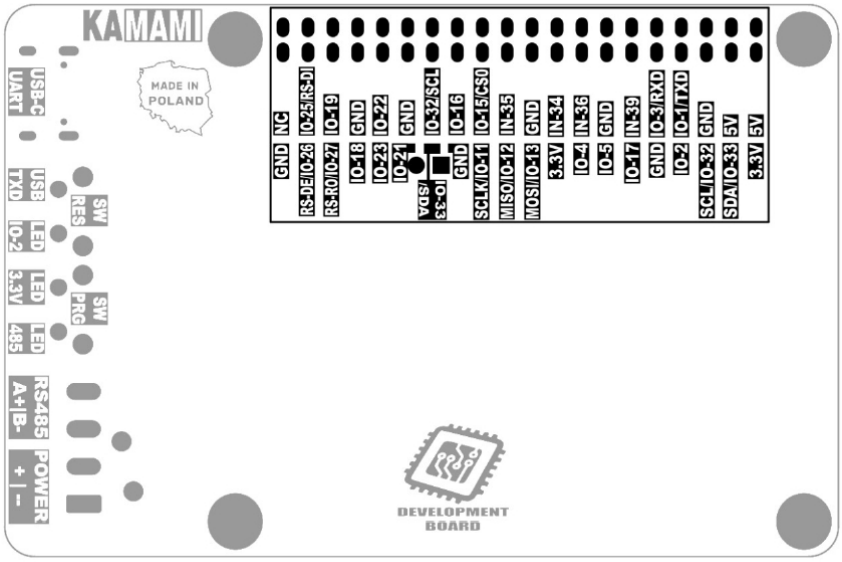
Złącze GPIO (J1) w standardzie Raspberry Pi zawiera 40 szpilek, do których doprowadzone są linie zasilania 5 V, 3,3 V, GND oraz wyprowadzenia GPIO modułu ESP32. Wyprowadzenia interfejsów UART (TXD, RXD), I2C (SDA, SCL) oraz SPI (MOSI, MISO, SCLK, CS0) zostały rozmieszczone tak, jak ma to miejsce w płytkach rodziny Raspberry Pi.

Dokładny opis wyprowadzeń oraz ich funkcje pokazuje rysunek i tabela poniżej:



J1	
3,3 V	5 V
GPIO33 (SDA)	5 V
GPIO32 (SCL)	GND
GPIO02 (LED D6)	GPIO1 (TXD)
GND	GPIO3 (RXD)
GPIO17	GPIO39 (IN)
GPIO5	GND
GPIO4	GPIO36 (IN)
3,3 V	GPIO34 (IN)
GPIO13 (MOSI)	GND
GPIO12 (MISO)	GPIO35 (IN)
GPIO11 (SCLK)	GPIO15 (SPI CS0)
GND	GPIO16
GPIO33 (SDA)	GPIO32 (SCL)
GPIO21	GND
GPIO23	GPIO22
GPIO18	GND
GPIO27 (RS485 RO)	GPIO19
GPIO26 (RS485 DE)	GPIO25 (RS485 DI)
GND	NC (niepodłączony)

Opis wyprowadzeń został również naniesiony na spodzie płytki KAmod ESP32 POW RS485:

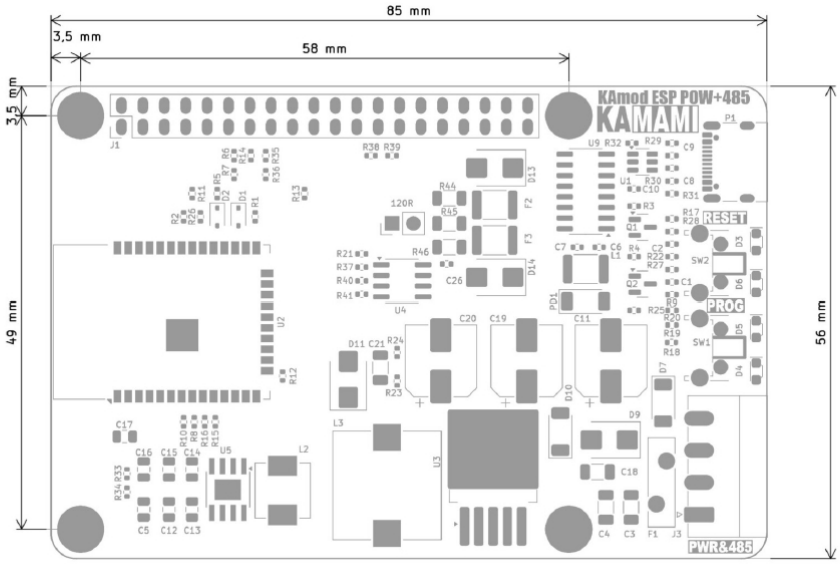


Uwagi dotyczące sygnałów wyprowadzonych na złącze GPIO

- Porty GPIO 34, 35, 36 i 39 modułu ESP32 mogą pracować wyłącznie jako wejścia cyfrowe lub analogowe – zostały one oznaczone symbolem IN.
- Porty GPIO 34 i 35 zostały wyposażone w rezystory podciągające pull-up 10k.
- Porty GPIO 36 i 39 zostały wyposażone w dzielniki napięcia (100k/10k), dzięki czemu można do nich dołączyć napięcie o maksymalnej wartosci 35 V.
- Porty GPIO 32 i 32 zostały dostosowane do funkcjonalności magistrali I2C i zawierają rezystory podciągające pull-up 2,2k.
- Porty GPIO1 oraz GPIO3 pełnią funkcję interfejsu UART i zostały połączone do modułu konwertera USB-UART oraz równolegle do złącza GPIO J2. Interfejs UART wysyła/odczytuje dane do/z złącza GPIO J2 oraz konwertera USB-UART jednocześnie.

Wymiary

Wymiary płytki KAmod ESP32 POW RS485 to 81x56 mm. Wysokość maksymalna wynosi ok. 20 mm. Na płytce znajdują się 4 otwory montażowe o średnicy 3 mm rozmieszczone podobnie jak na płytkach z rodziny Raspberry Pi.



Program testowy

Kod programu testowego znajduje się poniżej, można go skompilować w środowisku Arduino.

```
#include <WiFi.h>
#include <NetworkClient.h>
#include <WiFiAP.h>
#include <HardwareSerial.h>

#define LED_PIN 2

//Server
const char *ssid = "KAmodESP32";

const char *password = "12345678";
NetworkServer server(80);


IPAddress AP_LOCAL_IP (192, 168, 1, 1);
IPAddress AP_GATEWAY_IP (192, 168, 1, 254);
IPAddress AP_NETWORK_MASK (255, 255, 255, 0);


//RS485
#define RS485_RXD 27 //RO PIN
#define RS485_TXD 25 //DI PIN
#define RS485_RE_DE 26 //RE & DE PIN
#define RX_BUFF_SIZE 64
HardwareSerial RS485Port(2);


void setup() {

 //Serial
 Serial.begin (115200);
 Serial.println ("Hello. Configuring access point...");

 //Server
WiFi.softAPConfig(AP_LOCAL_IP, AP_GATEWAY_IP, AP_NETWORK_MASK);
 if (!WiFi.softAP(ssid, password)) {
   log_e("Soft AP creation failed.");
   while (1);
 }
 IPAddress myIP = WiFi.softAPIP();
 Serial.print("AP IP address: ");
 Serial.println(myIP);
 Serial.print("AP MAC address: ");
 Serial.println(WiFi.softAPmacAddress());
 server.begin();

 //RS485
 //pinMode(RS485_RE_DE, OUTPUT);
 //digitalWrite(RS485_RE_DE, LOW);
 RS485Port.setPins(RS485_RXD, RS485_TXD, RS485_RE_DE);
 RS485Port.setHwFlowCtrlMode(UART_HW_FLOWCTRL_CTS);
 RS485Port.setMode(UART_MODE_RS485_HALF_DUPLEX);
 RS485Port.begin(115200, SERIAL_8N1, RS485_RXD, RS485_TXD);
 RS485Port.println("RS485 Hello");

 //LED
```

```
 pinMode(LED_PIN, OUTPUT);
 for(int i=0; i<10; i++){
   digitalWrite(LED_PIN, HIGH);
   delay(200);
   digitalWrite(LED_PIN, LOW);
   delay(200);
 }

} void loop() {

Networkclient client = server.accept();

if (client) {
 Serial.println("********New client********");

 String currentLine = "";
 while (client.connected()) {
  if (client.available()) {
   char c = client.read();
   Serial.write();
   if (c == '\n') {
if (currentLine.length() == 0) {
     client.println("HTTP/1.1 200 OK");
     client.println("Content-type:text/html");
     client.println();
     client.println("Click <a href=\"/H\">here</a> to turn ON the LED.
");
     client.println("Click <a href=\"/L\">here</a> to turn OFF the LED.
");
     client.println("Click <a href=\"/RS485\">here</a> to wite to RS485.
");
     client.println();
     break;
    } else {
     currentLine = "";
    }
   } else if (c != '\r') {
    currentLine += c;
   }
   //if (currentLine.endsWith("GET /H")) {
   if (currentLine.indexOf("GET /H") >= 0) {
    digitalWrite(LED_PIN, HIGH);
   }
   //if (currentLine.endsWith("GET /L")) {
   if (currentLine.indexOf("GET /L") >= 0) {
    digitalWrite(LED_PIN, LOW);
   }
   if (currentLine.endsWith("GET /RS485")) {
    RS485Port.println("You send message via RS485");
   }
  }
 }
 client.stop();
 Serial.println("********Client Disconnected********");
} else {
 RS485Port.println("Tick...");
 delay(1000);
}

}
```

Program testowy konfiguruje sprzętowy interfejs UART 2 do pracy jako interfejs RS485 ze sprzętowym sterowaniem kierunkiem transmisji:

```
#define RS485_RXD 27 //RO PIN
#define RS485_TXD 25 //DI PIN
#define RS485_RE_DE 26 //RE & DE PIN
#define RX_BUFF_SIZE 64
HardwareSerial RS485Port(2);
...
RS485Port.setPins(RS485_RXD, RS485_TXD, RS485_RE_DE);
RS485Port.setHwFlowCtrlMode(UART_HW_FLOWCTRL_CTS);
RS485Port.setMode(UART_MODE_RS485_HALF_DUPLEX);
RS485Port.begin(115200, SERIAL_8N1, RS485_RXD, RS485_TXD);
RS485Port.println("RS485 Hello");
```

<pre>
Ponadto program testowy uruchamia moduł ESP32 w trybie access pointa oraz w roli serwera www:

```
const char *ssid = "KAmodESP32";
const char *password = "12345678";
NetworkServer server(80);
IPAddress AP_LOCAL_IP(192, 168, 1, 1);
IPAddress AP_GATEWAY_IP(192, 168, 1, 254);
IPAddress AP_NETWORK_MASK(255, 255, 255, 0);

WiFi.softAPConfig(AP_LOCAL_IP, AP_GATEWAY_IP, AP_NETWORK_MASK);
  if (!WiFi.softAP(ssid, password)) {
    log_e("Soft AP creation failed.");
    while (1);
  }
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  Serial.print("AP Mac Address: ");
  Serial.println(WiFi.softAPmacAddress());
  server.begin();
```

Po uruchomieniu programu testowego powstanie sieć Wi-Fi o nazwie (SSID): KAmodESP32 z hasłem: 12345678. Za pomocą smartfona należy dołączyć się do tej sieci, a następnie w przeglądarce internetowej wpisać adres IP strony: 192, 168, 1, 1. Wyświetli się bardzo prosta strona www, która pozwoli sterować diodą LED D6 lub wysłać prosty komunikat poprzez interfejs RS485:

Click here to turn ON the LED.
Click here to turn OFF the LED.
Click here to write to RS485.

=====Linki=====

BTC Korporacja
05-120 Legionowo
ul. Lwowska 5
tel.: (22) 767-36-20
faks: (22) 767-36-33
e-mail:
sprzedaz@kamami.pl
https://kamami.pl

Zastrzegamy prawo do wprowadzania zmian bez uprzedzenia.
Oferowane przez nas płytki drukowane mogą się różnić od prezentowanej w dokumentacji, przy czym zmianom nie ulegają jej właściwości użytkowe.
BTC Korporacja gwarantuje zgodność produktu ze specyfikacją.
BTC Korporacja nie ponosi odpowiedzialności za jakiekolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.
BTC Korporacja zastrzega sobie prawo do modyfikacji niniejszej dokumentacji bez uprzedzenia.