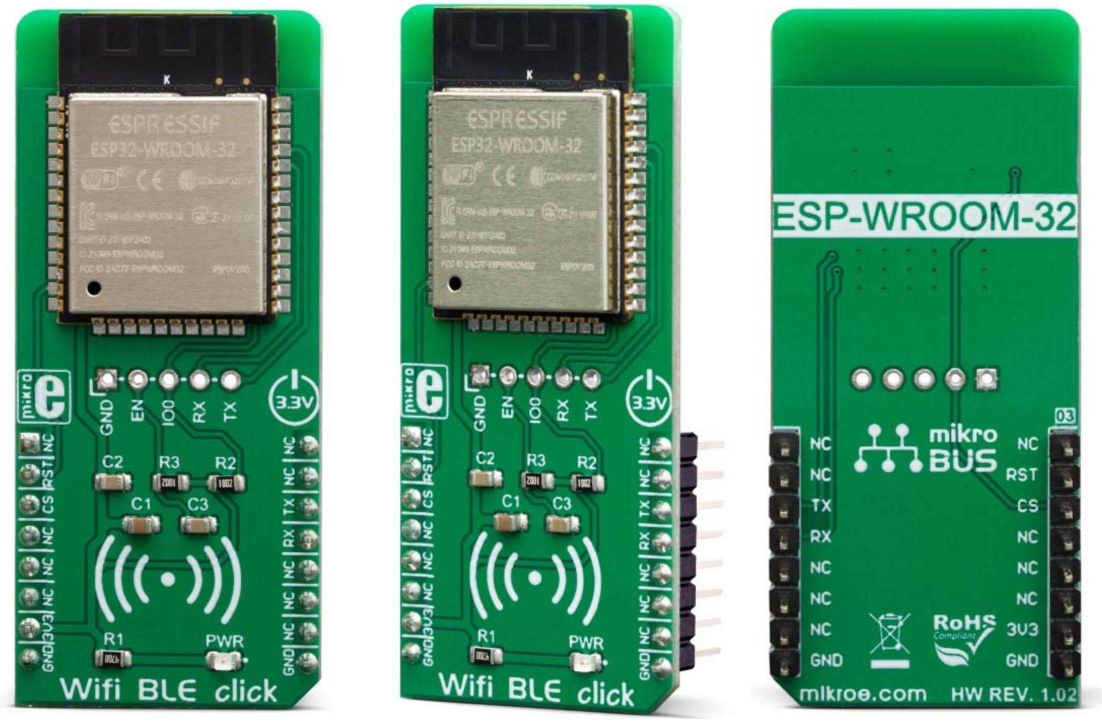
MIKROE



WIFI BLE CLICK

PID: MIKROE-3542
Weight: 22 g

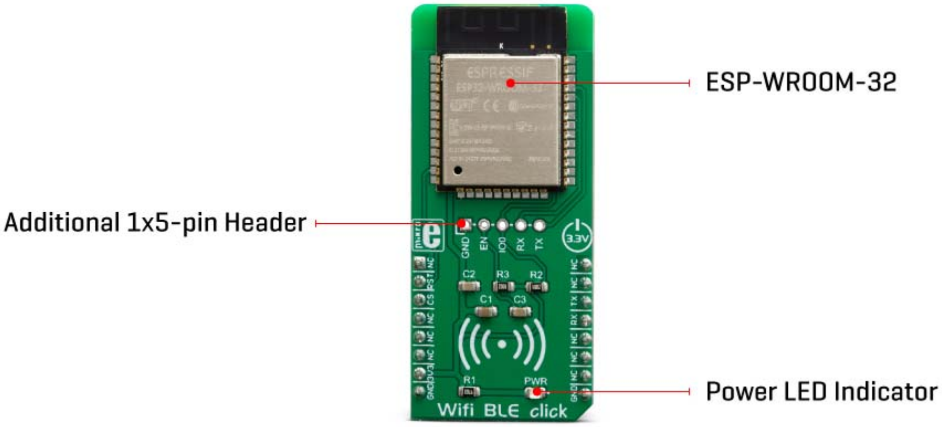
WiFi BLE click is a Click board™ which provides WiFi and BT/BLE connectivity for any embedded application. It features the ESP32-WROOM-32, an integrated wireless connectivity solution. The ESP32-WROOM-32 module itself is based on the ESP32-D0WDQ6 chip, a powerful 32-bit SOIC that runs RTOS featuring LwIP and TLS 1.2 with hardware acceleration. Armed with such a powerful processor, the ESP32-WROOM-32 can achieve data rates of up to 150 Mbps. The ESP32-WROOM-32 can achieve up to 20 dBm of TX power, ensuring a reliable connection and good signal coverage. It offers an UART communication interface allowing it to be operated by using simple AT commands.

WiFi BLE click is supported by a mikroSDK compliant library, which includes functions that simplify software development. This Click board™ comes as a fully tested product, ready to be used on a system equipped with the mikroBUS™ socket.

Over time, ESP WiFi modules became a synonym for IoT WiFi applications. The ESP32-WROOM-32 module is built on the success of its predecessors, namely ESP8266, bringing more processing power (up to 240MHz), more peripherals, and lowered power consumption (down to 5 µA in Deep-Sleep mode). The ESP32-WROOM-32 combines both WiFi, and BT/BLE connectivity options, allowing both long and short-range wireless connectivity, depending on specific requirements. Reliable and easy to use, WiFi BLE click is a perfect solution for development of various IoT applications, smart home applications, WiFi enabled toys, advanced robotics, and other similar applications.

HOW DOES IT WORK?

WiFi BLE click is equipped with the ESP32-WROOM-32, a generic WiFi and BT/BLE module produced by Espressif Systems, which is based on the ESP32-D0WDQ6 chip, a powerful 32-bit wireless SOIC, running RTOS. It is operated by a set of AT commands, over the UART interface, which makes the WiFi BLE click very easy to use. By integrating most of the critical components on the chip, the ESP32-D0WDQ6 allows the module to overcome any imperfections of external discrete components, allowing signal transmission power of up to 20dBm, and -97 dBm sensitivity for the BT NZIF receiver.



The ESP32-WROOM-32 supports 802.11 b/g/n WiFi standards at 2.4GHz, allowing up to 150Mbps for the 802.11n. Security options including WEP, WPA/WPA2 PSK/Enterprise, as well as hardware accelerated encryption including the AES/SHA2/Elliptical Curve Cryptography/RSA-4096, are also present on the ESP32-WROOM-32 module. It supports Sniffer, Station, SoftAP, and Wi-Fi direct modes. The ESP32-WROOM-32 module also complies with the Bluetooth v4.2 BR/EDR, and BLE specifications.

By combining both WiFi and BT technologies, the ESP32-WROOM-32 module offers a choice between using WiFi for large area coverage and Internet connectivity, or point-to-point and PAN BT connectivity. WiFi can be power demanding, while BT, on the other hand, ensures minimalistic power consumption rates.

WiFi BLE click offers the basic WiFi and BLE functionality, not exposing all the pins of the module. As such, it is meant to be used as an add-on to an existing application. However, it is equipped with the programming header, which exposes UART programming pins, along with the boot mode selection pin (IO0) and the Reset/Enable pin (EN). If the IO0 stays at a LOW logic level after the module is reset, it will boot into the serial bootloader mode. The firmware can be then programmed over the RX0 and TX0 pins, located at the additional 1x5-pin header on the Click board™.

The EN pin is also routed to the mikroBUS™ CS pin, and it is used to reset the Click board™ by the host MCU. Again, by setting the CS pin to a LOW logic level, the ESP32-WROOM-32 module will be reset. Most of the ESP-WROOM-32 module pins are configurable and multiplexed with other functions. One such pin is provided for a reference (IO14) and it is routed to the RST pin of the mikroBUS™.

The communication with the host MCU is done over a dedicated UART interface. RX and TX pins of the mikroBUS™ are routed to the appropriate pins of the module. The host MCU can send commands and receive responses over the UART interface, allowing the ESP32-WROOM-32 to perform various operations. The default baud rate of the ESP32-WROOM-32 module is 115200.

The real power behind the ESP-WROOM-32 module is in the usage simplicity. By sending a simple macro over the UART, the WiFi BLE click can be easily used to establish an internet connection and do some additional tasks. By utilizing its GPIOs (multiplexed with some other functions), the WiFi BLE click can perform some additional tasks on its own. Besides the official mikroSDK compatible library and its and functions, there is a comprehensive programming guide available for download, which explains each AT command in more details. It can be used for application development, along with the provided demo example.

SPECIFICATIONS

Type	Wi-Fi
Applications	WiFi BLE click is a perfect solution for development of various IoT applications, smart home applications, WiFi enabled toys, advanced robotics, and other similar applications.
On-board modules	ESP32-WROOM-32, a generic WiFi and BT/BLE module made by Espressif Systems.

Key Features	Easy to use, rich with features, both WiFi and Bluetooth/BLE connectivity options, operated by a simple set of AT commands, etc.
Interface	GPIO,UART
Input Voltage	3.3V
Click board size	L (57.15 x 25.4 mm)

PINOUT DIAGRAM

This table shows how the pinout on **WiFi BLE click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin	mikro™ BUS				Pin	Notes
	NC	1	AN	PWM	16	NC	
ESP32 IO14	**RST**	2	RST	INT	15	NC	
ESP32 Enable	**EN**	3	CS	RX	14	**TX**	UART Transmit
	NC	4	SCK	TX	13	**RX**	UART Receive
	NC	5	MISO	SCL	12	NC	
	NC	6	MOSI	SDA	11	NC	
Power Supply	**3.3V**	7	3.3V	5V	10	NC	
Ground	**GND**	8	GND	GND	9	**GND**	Ground

ONBOARD SETTINGS AND INDICATORS

Label	Name	Default	Description
PWR	PWR	-	Power LED Indicator

ADDITIONAL PINS

Pin	Description
GND	Reference Ground
EN	ESP32-WROOM-32 Enable/Reset
IO0	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
RX	GPIO3, U0RXD, CLK_OUT2
TX	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2

SOFTWARE SUPPORT

We provide a library for the **WiFi BLE click** on our LibStock page, as well as a demo application (example), developed using MikroElektronika compilers. The demo can run on all the main MikroElektronika development boards.

Library Description
Library carries generic command parser adopted for AT command based modules. Generic parser.

Key functions:

- `wifible_cmdSingle` - Sends provided command to the module.
- `wifible_setHandler` - Handler assignation to the provied command.
- `wifible_modulePower` - Turn on module.

Examples description

The application is composed of the three sections :

- System Initialization - Initializes all necessary GPIO pins, UART used for the communcation with WiFi-BLE module and UART used for infromation logging.
- Application Initialization - Initializes driver, power on module and sends few command for the default module configuration.
- Application Task - Waits for valid user input and executes functions based on set of valid commands.

Commands : '+' - Next command 's' - Start command 'e' - change example (WiFi example / BLE example)

Additional Functions :

- void _packageBeaconData( ) - Package iBeacon data
- void _WiFi_TCPServer( ) - WiFi example / command list
- void _BLE_iBeacon( ) - BLE example / command list

Timer initialization and default handler.

Note:

- For click test we've created two examples which you can change by sending the command [e] to the terminal.
- In the WiFi example, the module connects to TCP server (we used application called "TCP Server" for Android) and sends the message "MikroE" to it.
- BLE example demonstrates how WiFi BLE Click acts as iBeacon - with the help of the right commands we've created a beacon which allows us to scan it using the app called Beacon scanner.
- Only some of the commands that WiFi BLE click supports were used in the example
- for the rest of the available commands check the documentation
- Before starting the program it is necessary to provide valid parameters for connecting to the router (SSID, password) as well as the IP address and port of the TCP server which the Click board is going to connect to.

```
void applicationTask()

{

    uint8_t dataReady_;

    char receivedData_;


    dataReady_ = UART_Rdy_Ptr( );

    wifible_process();


    if (dataReady_ != 0)
    {
        receivedData_ = UART_Rd_Ptr( );

        switch (receivedData_)
        {
            case '+' :
            {
                command_cnt++;
                if(command_cnt > 5)
                {
                    command_cnt = 0;
                }
                fCommand = _WIFIBLE_NEXT_COMMAND;
                break;
            }
            case 's' :
            {
                fCommand = _WIFIBLE_START_COMMAND;
                break;
            }
            case 'e' :
            {
                if(fExample == _WIFIBLE_BLE_EXAMPLE)
                {
                    fExample = _WIFIBLE_WIFI_EXAMPLE;
                    mikrobus_logWrite("- Example: WiFi - TCP example ", _LOG_LINE);
                    mikrobus_logWrite("- [+] First command // [e] Next example", _LOG_LINE);
                    command_cnt = -1;
                    fCommand = _WIFIBLE_NO_COMMAND;
                }
                if(fExample == _WIFIBLE_WIFI_EXAMPLE)
                {
                    fExample = _WIFIBLE_BLE_EXAMPLE;
                    mikrobus_logWrite("- Example: BLE - iBeacon example ", _LOG_LINE);
                    mikrobus_logWrite("- [+] First command // [e] Next example", _LOG_LINE);
                    command_cnt = -1;
                    fCommand = _WIFIBLE_NO_COMMAND;
                }
                break;
            }
        }

        if(fExample == _WIFIBLE_WIFI_EXAMPLE)
        {
            _WiFi_TCPServer(command_cnt);
        }
        if(fExample == _WIFIBLE_BLE_EXAMPLE)
        {
            _BLE_iBeacon(command_cnt);
        }
    }
}
```

Alongside with the demo application timer initialization functions are provided. Note that timer is configured acording to default develoment system and MCUs, changing the system or MCU may require update of timer init and timer ISR functions.

The full application code, and ready to use projects can be found on our LibStock page. Other mikroE Libraries used in the example:

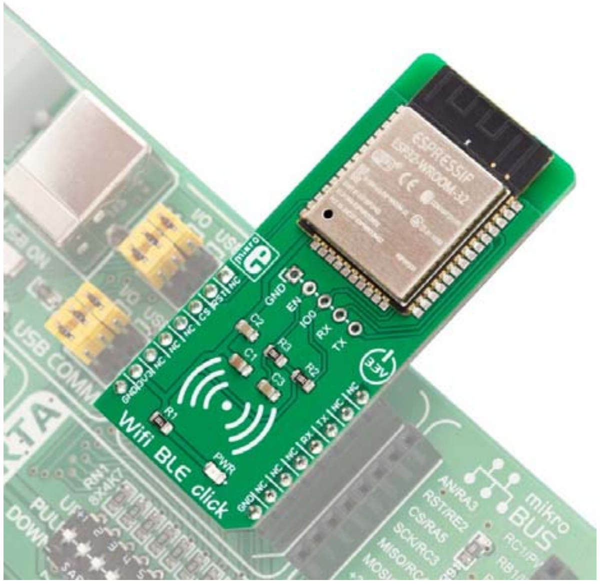
- `String`
- `Conversion`

Additional notes and informations
Depending on the development board you are using, you may need USB UART click, USB UART 2 click or RS232 click to connect to your PC, for development systems with no UART to USB interface available on the board. The terminal available in all MikroElektronika compilers, or any other terminal application of your choice, can be used to read the message.

MIKROSDK

This click board is supported with mikroSDK - MikroElektronika Software Development Kit. To ensure proper operation of mikroSDK compliant click board demo applications, mikroSDK should be downloaded from the LibStock and installed for the compiler you are using.

For more information about mikroSDK, visit the official page



https://www.mikroe.com/wifi-ble-click/5-14-19