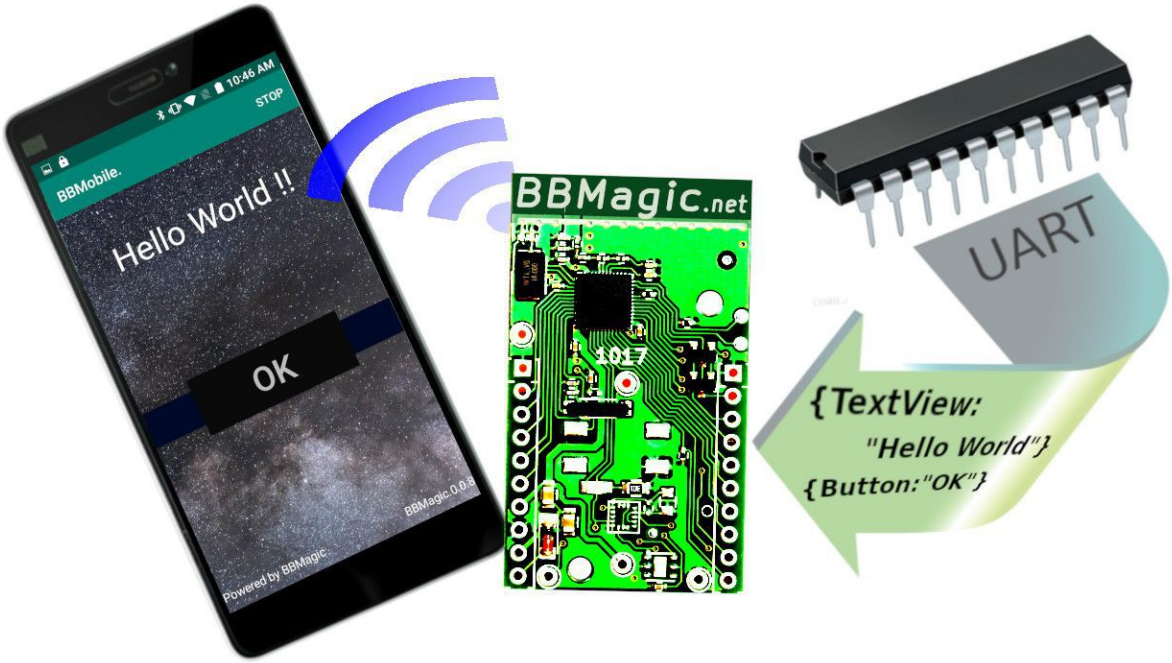
www.bbmagic.net

BBMobile

BEZPRZEWODOWE APLIKACJE MOBILNE TWORZONE PRZEZ UART MIKROKONTROLERA



specyfikacja

moduł F_01.00 | aplikacja 0.2.5

Moduł BBMobile umożliwia tworzenie aplikacji dla urządzeń mobilnych z systemem Android i Bluetooth od wersji 4.0 (Bluetooth Low Energy – BLE).

Aplikacje tworzone i obsługiwanie są przy pomocy dowolnego mikrokontrolera z interfejsem UART, 9600bps, 8N1.

Opis interfejsu aplikacji przesyłany jest w postaci struktury danych w popularnym formacie JSON (JavaScript Object Notation).

Mikrokontroler steruje zachowaniem kontrolek wysyłając krótkie komendy sterujące.

Informacje o interakcji użytkownika (tapnięcie guzika, przełączenie switcha, itp.) przesyłane są zwrotnie do mikrokontrolera,

BBMobile i...

- **zapomnij o problemach z estetycznym interfejsem użytkownika – z BBMobile stworzysz i obsłużysz estetyczny HMI prostymi tekstowymi komendami,**
- **spraw, aby następny projekt działał bezprzewodowo,**
- **twórz piękne graficzne interfejsy dla swoich projektów,**
- **naucz się jak korzystać z lekkiego i popularnego formatu wymiany danych JSON (JavaScript Object Notation),**
- **zobacz możliwości najnowszej technologii Bluetooth Low Energy (BLE),**

Spis treści

Jak najszybciej opanować BBMobile....4

Parametry modułu BBMobile....5

Wyprowadzenia modułu....5

Komunikacja UART....7

Komunikacja w trybie komend – brak połączenia BLE....7

Format komunikacji....7
Komendy i odpowiedzi modułu BBMobile....8

Komunikacja w trybie połączenia BLE....9

Struktura JSON tworząca interfejs użytkownika....10
Komunikaty sterujące zachowaniem kontrolek....10
Informacje zdarzeniowe z aplikacji BBMobile....11
Komunikat TOAST....11

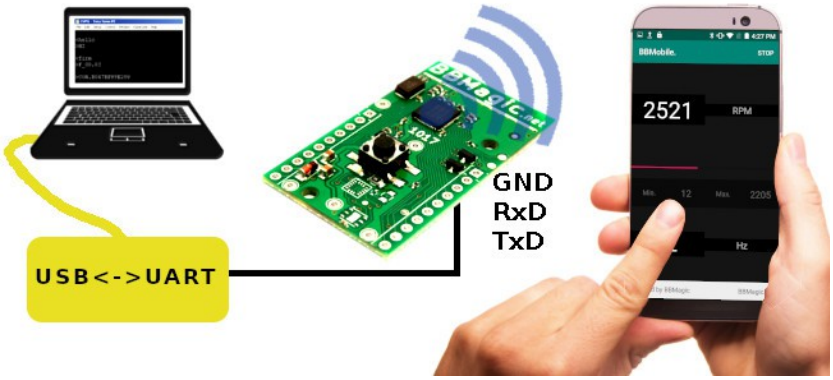
Kontrolki....11

Layouty....15

Przykładowe projekty interfejsów użytkownika....16

TextView „Hello world” i Button....16

Jak najszybciej opanować BBMobile



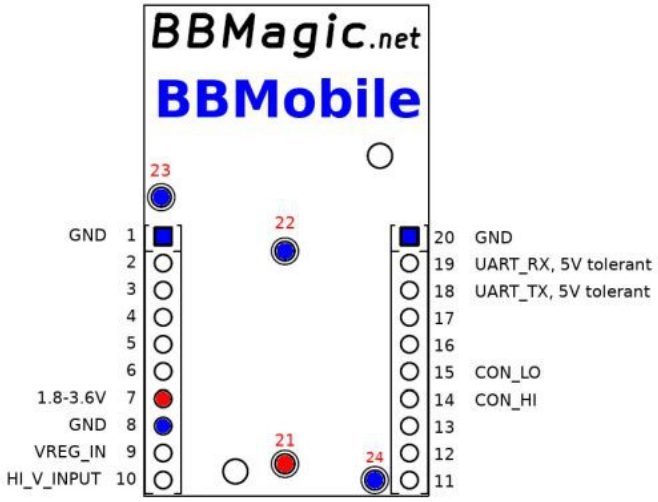
- zainstaluj w systemie Android aplikację BBMobile,
- zainstaluj i uruchom na PC dowolny program terminala, np. TeraTerm,
- podłącz moduł BBMobile do komputera PC wykorzystując konwerter UART<->USB,
- włącz zasilanie modułu BBMobile,
- uruchom aplikację BBMobile, nawiąż połączenie Bluetooth z modułem i zaprojektuj pierwszy interfejs mobilny.

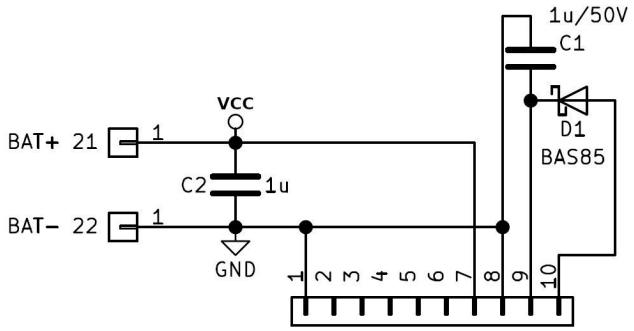
http://bbmagic.net/bbmobile-jak-zaczac/

Parametry modułu BBMobile

- Zasilanie bezpośrednie 1,8V do 3,6V (pady 7,8),
- Zasilanie z baterii CR2030 (pady dla uchwytu 21, 22),
- Zasilanie do 30V po podłączeniu do padów 7,8,9 stabilizatora np.: 78L33, LP2950CZ-3,3RAG, LM2936Z-3,3, LT1121CZ-3,3,
- Pobór prądu przy zasilaniu 3,3V – 1mA,
- Komunikacja przewodowa z mikrokontrolerem – UART:
  - tolerancja 5V,
  - domyślnie 9600 bps, możliwe 4800 bps,
  - 8N1 – 8 bitów danych, bez bitu parzystości, 1 bit stopu,
- Komunikacja bezprzewodowa z urządzeniem mobilnym – BLE (Bluetooth Low Energy) – Bluetooth od wersji 4.0.
- Bezpłatna aplikacja działająca z Android od wersji 4.4.2
- Wymiary: 27 x 46 mm,

Wyprowadzenia modułu

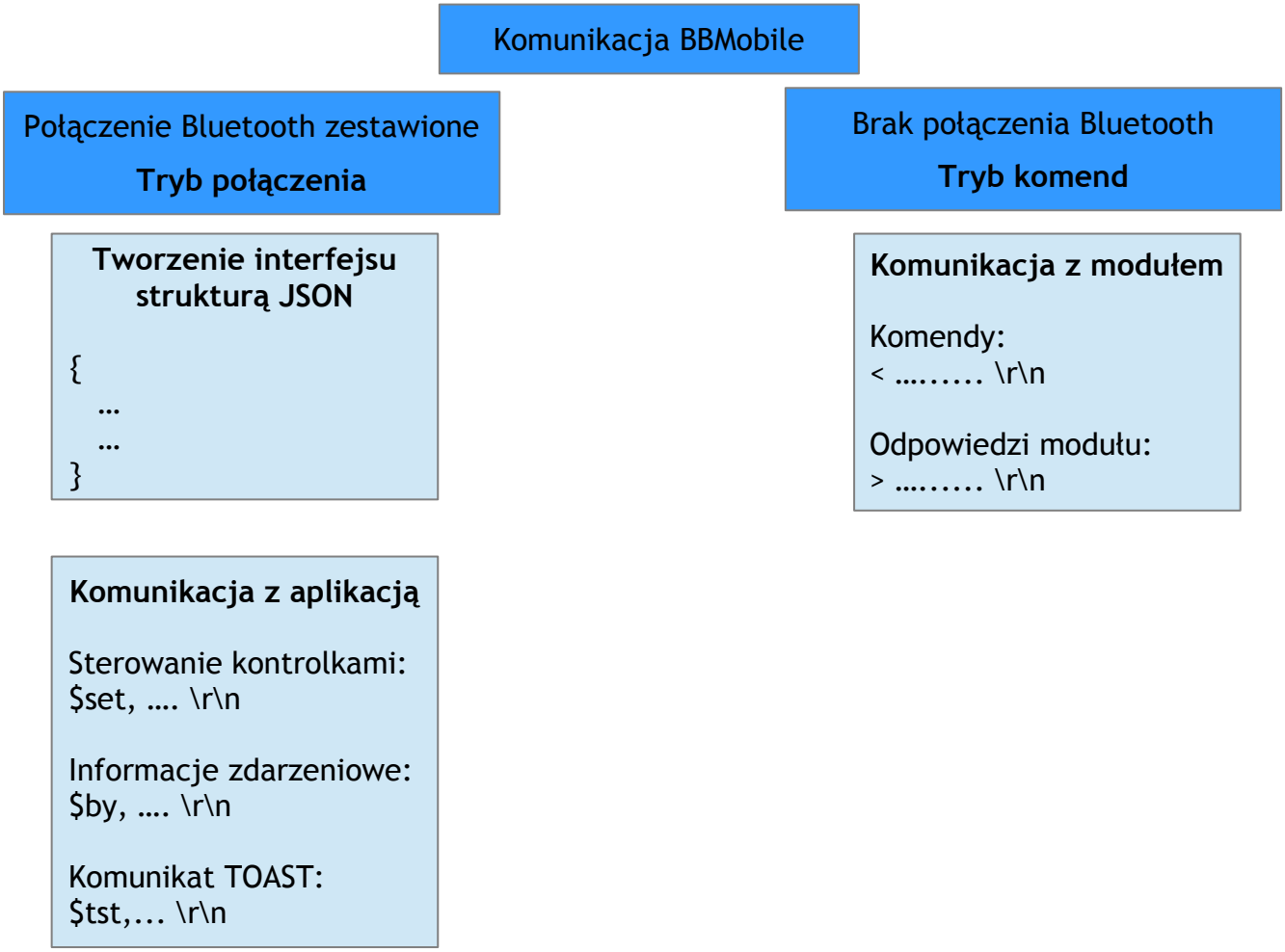




Numer Pinu	Nazwa	Opis
1, 8, 20, 23, 24	GND	Masa
22	GND	Masa. Miejsce do wlutowania uchwytu baterii CR2030.
7	VCC	Zasilanie. Napięcie w granicach od 1,8V do 3,6V.
21	VCC	Zasilanie. Miejsce do wlutowania uchwytu baterii CR2030.
9	VREG_IN	Wejście stabilizatora w przypadku zasilania wyższym napięciem podanym do pinu 10. Stabilizator należy podłączyć do pinów 7–wyjście, 8–gnd, 9–wejście.
10	HI_V_INPUT	Wejście napięcia zasilania dla stabilizatora podłączonego do pinów 7,8,9. (zabezpieczone przed odwrotną polaryzacją)
14	CON_HI	Wyjście sygnalizujące stanem wysokim nawiązanie połączenia Bluetooth. Sygnał może być użyty jako przerwanie budzące lub uruchamiające kontroler nadrzędny.
15	CON_LO	Wyjście sygnalizujące stanem niskim nawiązanie połączenia Bluetooth. Sygnał może być użyty jako przerwanie budzące lub uruchamiające kontroler nadrzędny.
18	UART_TX	Wyjście nadajnika UART. Toleruje napięcia 5V.
19	UART_RX	Wejście odbiornika UART. Toleruje napięcia 5V.

http://bbmagic.net/bbmobile-w-projekcie-pinologia-i-zasilanie/

Komunikacja UART



Komunikacja w trybie komend – brak połączenia BLE

Moduł znajduje się w tym trybie gdy połączenie z aplikacją BBMobile nie jest zestawione. Po włączeniu zasilania BBMobile znajduje się w trybie komend.

Format komunikacji

Każda komenda wysyłana do modułu gdy połączenie BLE nie jest zestawione rozpoczyna się znakiem '<'. Zawiera znaki ASCII i kończy znakami powrót karetki i nowej linii. (Carriage Return + Line Feed – „\r\n” – kody ASCII 13 i 10 dec).

<	Znaki ASCII	„\r\n”

Każdy komunikat otrzymywany od modułu BBMobile gdy połączenie BLE nie jest zestawione rozpoczyna się znakiem '>'. Zawiera znaki ASCII i kończy znakami powrót karetki i nowej linii. (Carriage Return + Line Feed – „\r\n” – kody ASCII 13 i 10 dec).

>	Znaki ASCII	„\r\n”

Komendy i odpowiedzi modułu BBMobile

Komendy realizowane są przez moduł tylko wtedy gdy połączenie Bluetooh SMART nie jest zestawione. Po wykonaniu żądanej akcji odsyłana jest odpowiedź lub informacja o błędzie. W poniższej tabeli zestawiono komendy i możliwe odpowiedzi modułu. Dla zachowania przejrzystości pominięto znaki kończące komunikat: „\r\n”.

Komenda	**Odpowiedź BBMobile**
<hello	Pozwala sprawdzić poprawność komunikacji UART. >HI
<baud,bbbb	Zmienia prędkość transmisji UART. Domyślna po włączeniu zasilania to 9600bps. bbbb – prędkość transmisji 4800 lub 9600. >OK,bbbb – zmieniono prędkość transmisji UART >ERR005.parameter – niepoprawny parametr.
<name,nnn	Zmienia nazwę urządzenia Bluetooth SMART. nnn – nowa nazwa ; maksymalnie 25 znaków ; pierwszy znak musi być literą lub cyfrą. >OK,nnn – zmieniono nazwę >ERR003.too_long – podano zbyt długą nazwę. >ERR005.parameter – pierwszy znak nazwy powinien być literą lub cyfrą.
<pin,nnn	Ustawia kod dostępu do modułu – PIN. PIN może zawierać od jednej do dziesięciu cyfr. Jego wpisanie w aplikacji BBMobile będzie konieczne do nawiązania połączenia z modułem. >OK,nnn – pin ustawiony >ERR003.too_long – podany PIN zawiera więcej niż 10 cyfr i jest zbyt długi. >ERR005.parameter – nieprawidłowy format PINu.
<pin,0	Usuwa zabezpieczenie modułu kodem PIN. Od tej chwili podanie PINu przy zestawianiu połączenia nie będzie wymagane.

	>OK,0 – zabezpieczenie PIN zostało usunięte.
<firm	Podaje wersję firmware'u modułu. >F_00.01 – numer wersji firmware'u.
	>CON.pppppppppppp Komunikat informujący o zestawieniu połączenia Bluetooth. 'pppppppppppp' – to sześciobajtowy adres urządzenia, które nawiązało połączenie z modułem BBMobile. Po wysłaniu tego komunikatu moduł przechodzi w tryb połączenia.
	>DCON Komunikat informujący o zamknięciu połączenia Bluetooth. Po wysłaniu tego komunikatu moduł przechodzi w tryb komend.
	>ERR001.connected Komenda trybu bezpołączeniowego nie może być zrealizowana ponieważ połączenie BLE jest zestawione.
	>ERR002.not_connected Komenda trybu połączeniowego nie może być zrealizowana, ponieważ połączenie BLE nie jest zestawione
	>ERR004.no_cmd Komenda nie jest obsługiwana.
	>ERR007.tout Komunikat informujący, że przyjmowanie kodu JSON zostało zakończone z powodu zbyt długiej przerwy w jego nadawaniu – timeout.

Komunikacja w trybie połączenia BLE

Po zestawieniu połączenia BLE:

- jako pierwsze dane należy przesłać poprawny obiekt JSON opisujący interfejs użytkownika. Moduł akceptuje kod '$ok' lub odrzuca w przypadku wykrycia błędu '$nok'.
- sterowanie kontrolkami odbywa się poprzez wysyłanie poleceń: $set... ,
- komunikat TOAST jest wyświetlany po przesłaniu komendy $tst,... ,
- informacje od kontrolek przesyłane są komunikatami $by,... ,

Struktura JSON tworząca interfejs użytkownika

Format JSON

Skrótowiec JSON pochodzi od angielskiego JavaScript Object Notation. Jest prostym, lekkim, tekstowym formatem wymiany danych bazującym na języku JavaScript.

- ✓ Obiekty stryktury zamknięte są w nawiasach '{' i '}',
- ✓ Ciągi znaków zamknięte są w cudzysłowach,
- ✓ Podstawowym budulcem JSONa jest struktura składająca się z pary ”nazwa” : ”wartość” ; np. { "paliwo":"benzyna","przebieg":"50000km" }
- ✓ Obiekty rozdzielone są przecinkami ; np.: { "płeć":"kobieta", "wzrost":"169" }
- ✓ Obiekty mogą być zorganizowane w tablicy. Elementy tablicy zamknięte są w nawiasach kwadratowych ; np.: { "komponenty":[{"koła":"zimowe"}, {"nadwozie":"sedan"}, {"silnik":"disel"}] }

http://bbmagic.net/co-to-jest-json-i-jak-go-uzyc/

Komunikaty sterujące zachowaniem kontrolek

Komunikat ustawiający parametr kontrolki rozpoczyna się znakiem '$'. Zawiera znaki ASCII i kończy znakami powrót karetki i nowej linii. (Carriage Return + Line Feed – „\r\n” – kody ASCII 13 i 10 dec).

$set,	Informacje zmieniające parametry kontrolek	„\r\n”

- Komunikat ustawiający w kontrolce o nazwie 'k1' pole 'p1' na wartość 'w1' ma postać: $set,k1:p1=”w1”.
- Komunikat ustawiający w kontrolce o nazwie 'k1' pole 'p1' na wartość 'w1', a pole 'p2' na wartość 'w2' ma postać: $set,k1:p1=”w1”p2=”w2”.
- Komunikat ustawiający w kontrolce o nazwie 'k1' pole 'p1' na wartość 'w1', a w kontrolce 'k2' pole 'p2' na wartość 'w2' ma postać: $set,k1:p1=”w1”,k2:p2=”w2”.

Informacje zdarzeniowe z aplikacji BBMobile

Aplikacja przesyła informację za każdym razem gdy użytkownik:

- naciśnie kontrolkę Button,
- przestawi kontrolkę ToggleButton.

Komunikat informujący o zdarzeniu rozpoczyna się znakiem '$'.Zawiera znaki ASCII i kończy znakami powrót karetki i nowej linii. (Carriage Return + Line Feed – „\r\n” – kody ASCII 13 i 10 dec).

$by,	Informacje od kontrolek	„\r\n”

Po wykonaniu akcji przez użytkownika wysyłany jest stan wszystkich kontrolek wejściowych znajdujących się w interfejsie począwszy od tej, która wywołała akcję.

Komunikat TOAST

TOAST to pojawiający się u dołu ekranu komunikat, który po chwili samoczynnie znika. Komunikat ten można wyświetlić tylko wtedy, gdy moduł znajduje się w trybie połączenia.

Komenda wyświetlająca komunikat TOAST rozpoczyna się znakiem '$'.Zawiera znaki ASCII i kończy znakami powrót karetki i nowej linii. (Carriage Return + Line Feed – „\r\n” – kody ASCII 13 i 10 dec).

$tst,	”Komunikat do wyświetlenia zawarty w cudzysłowie”	„\r\n”

Aby wyświetlić komunikat 'Hi user' wprowadzamy do UARTa modułu, który jest w trybie połączenia komendę:

```
$tst,"Hi user"
```

Kontrolki

Kontrolki są podstawowymi cegiełkami budującymi interfejs użytkownika na ekranie urządzenia mobilnego. Każda kontrolka posiada swoje cechy/parametry, które mogą przyjmować określone wartości. Kontrolki dzielimy na:

- wejściowe – przyjmują informację od użytkownika: EdiText, Button, TogButoon, Spinner,

– wyjściowe – prezentują informacje użytkownikowi: TextView, ProgBar,

Poniższe tabele zawierają cechy kontrolek wraz z ich opisem. Definiując przykładowo kontrolkę Button o nazwie 'b1', w kolorze czerwonym z napisem GO piszemy kod:

```
{
        ”ty”:”Button”,
        ”n”:”b1”,
        ”te”:”GO”,
        ”bg”:”255,0,0”
}
```

Kontrolka: Button		"ty":"Button"
NAZWA POLA	MNEMONIK	OPIS
Type	ty	Określa typ kontrolki. W tym przypadku jest to 'Button' – przycisk jednostanowy.
Name	n	Umożliwia nadanie unikalnej nazwy kontrolce służącej do jej identyfikacji w interfejsie. Np.: ”b1”
Text	te	Ustawia wyświetlany na guziku tekst. Np.: ”OK”
Text Color	tc	Definiuje składowe RGB koloru tekstu. np.: ”255.255.255” określa kolor biały.
Text Size	ts	Określa rozmiar tekstu. Np.: ”30”
Text Style	tl	Ustawia styl wyświetlanego tekstu. Np.: ”bold”, ”italic”, ”bolditalic”
Background	bg	Definiuje składowe RGB koloru tła kontrolki. np.: ”255.0.0” określa kolor czerwony.
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku. Gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' wszystkich kontrolek znajdujących się w layoucie.
Enabled	en	0 – kontrolka nieaktywna (nie reaguje na dotknięcie) 1 lub brak parametru – kontrolka aktywna
UpDown	ud	0 lub brak parametru – przycisk wysyła informację tylko przy jego naciśnięciu. 1 – wysyłanie komunikatów przy naciśnięciu i puszczeniu przycisku.

Kontrolka: TogButton		"ty":"TogButton"
NAZWA POLA	MNEMONIK	OPIS
Type	ty	Określa typ kontrolki. W tym przypadku jest to 'TogButton' – przełącznik o dwóch stanach stabilnych ON i OFF.
Name	n	Umożliwia nadanie unikalnej nazwy kontrolce służącej do jej identyfikacji w interfejsie. Np.: ”tb1”
Mode	m	0 lub brak parametru – kontrolka przyjmie styl przycisku.

		1 – kontrolka przyjmie styl przełącznika.
Text ON	ton	Ustawia tekst, który jest wyświetlany na przełączniku gdy jest on włączony. Np.: ”ON”
Text OFF	toff	Ustawia tekst, który jest wyświetlany na przełączniku gdy jest on wyłączony. Np.: ”OFF”
Text Color	tc	Definiuje składowe RGB koloru tekstu. np.: ”255.255.255” określa kolor biały.
Text Size	ts	Określa rozmiar tekstu. Np.: ”30”
Text Style	tl	Ustawia styl wyświetlanego tekstu. Np.: ”bold”, ”italic”, ”bolditalic”
Background	bg	Definiuje składowe RGB koloru tła kontrolki. np.: ”255.0.0” określa kolor czerwony.
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku. Gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' wszystkich kontrolek znajdujących się w layoucie.
Enable	en	0 – kontrolka jest nieaktywna (nie reaguje na dotknięcie) 1 lub brak parametru – kontrolka jest aktywna
Initial value	iv	Początkowy stan kontrolki: 0 lub brak parametru – wyłączona. 1 – włączona.

Kontrolka: TextView		`"ty":"TextView"`
NAZWA POLA	MNEMONIK	OPIS
Type	ty	Określa typ kontrolki. W tym przypadku jest to 'TextView' – pole nieedytowalne.
Name	n	Umożliwia nadanie unikalnej nazwy kontrolce służącej do jej identyfikacji w interfejsie. Np.: ”t1”
Text	te	Ustawia wyświetlany w polu tekst. Np.: ”Hello World”
Text Color	tc	Definiuje składowe RGB koloru tekstu. np.: ”255.255.255” określa kolor biały.
Text Size	ts	Określa rozmiar tekstu. Np.: ”30”
Text Style	tl	Ustawia styl wyświetlanego tekstu. Np.: ”bold”, ”italic”, ”bolditalic”
Background	bg	Definiuje składowe RGB koloru tła kontrolki. np.: ”255.0.0” określa kolor czerwony.
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku. Gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' wszystkich kontrolek znajdujących się w layoucie.

Kontrolka: EditText		`"ty":"EditText"`
NAZWA POLA	MNEMONIK	OPIS
Type	ty	Określa typ kontrolki. W tym przypadku jest to 'EditText' – pole edytowalne.

Name	n	Umożliwia nadanie unikalnej nazwy kontrolce służącej do jej identyfikacji w interfejsie. Np.: ”e1”
Field Type	ft	0 lub brak parametru – pole tekstowego 1 – pole liczbowe
Text	te	Ustawia wyświetlany w polu tekst. Np.: ”Hello World”
Text Color	tc	Definiuje składowe RGB koloru tekstu. np.: ”255.255.255” określa kolor biały.
Text Size	ts	Określa rozmiar tekstu. Np.: ”30”
Text Style	tl	Ustawia styl wyświetlanego tekstu. Np.: ”bold”, ”italic”, ”bolditalic”
Background	bg	Definiuje składowe RGB koloru tła kontrolki. np.: ”255.0.0” określa kolor czerwony.
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku. Gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' wszystkich kontrolek znajdujących się w layoucie.

Kontrolka: Spinner		`"ty":"Spinner"`
NAZWA POLA	MNEMONIK	OPIS
Type	ty	Określa typ kontrolki. W tym przypadku jest to 'Spinner' – pole umożliwiające wybór jeden z wielu.
Name	n	Umożliwia nadanie unikalnej nazwy kontrolce służącej do jej identyfikacji w interfejsie. Np.: ”s1”
Mode	m	0 – kontrolka będzie miała postać okna dialogowego 1 lub brak parametru – kontrolka będzie miała postać listy
Items	is	Lista zawierająca elementy wyboru rozdzielone przecinkami. Np.: „jeden,dwa,trzy,cztery”
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku. Gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' wszystkich kontrolek znajdujących się w layoucie.

Kontrolka: ProgBar		`"ty":"ProgBar"`
NAZWA POLA	MNEMONIK	OPIS
Type	ty	Określa typ kontrolki. W tym przypadku jest to 'ProgBar' – pasek wizualizacji postępu.
Name	n	Umożliwia nadanie unikalnej nazwy kontrolce służącej do jej identyfikacji w interfejsie. Np.: ”p1”
Determine	d	0 lub brak parametru – kontrolka sygnalizuje postęp bez określania stopnia wykonania. 1 – kontrolka pokazuje postęp określający stopień wykonania.
Progress	p	Aktualny postęp w procentach: przedział od 0 do 100.
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku. Gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' wszystkich kontrolek

		znajdujących się w layoucie.

Kontrolki mogą mieć nieco różny wygląd zależny od wersji systemu Android.

Layouty

Layout to kontener określający sposób ułożenia obiektów znajdujących się w jego wnętrzu. Istnieją dwa rodzaje layoutów: horyzontalny, w którym obiekty sąsiadują ze sobą w poziomie oraz wertykalny, gdzie komponenty układane są jeden pod drugim. Obiekt Layout posiada cechy przedstawione w poniższej tabeli.

Obiekt: Layout		`"ty":"Layout"`
NAZWA POLA	MNEMONIK	OPIS
type	ty	Określa typ obiektu. W tym przypadku jest to 'Layout' – kontener organizujący ułożenie obiektów.
orientation	or	Określa sposób ułożenia obiektów wewnątrz layoutu: ”or”:”v” - ułożenie wertykalne ”or”:”h” - ułożenie horyzontalne
background	bg	Definiuje składowe RGB koloru tła layoutu. np.: ”0.0.255” określa kolor niebieski.
image	img	Powoduje użycie jednej z wbudowanych grafik jako tła layoutu. ”img”:”1” - używa grafiki numer 1 jako tła.
components	cs	Jest tablicą zawierającą wewnętrzne obiekty layoutu. Symbolicznie jako: „cs”:[{obiekt_1},{obiekt_2},{obiekt_3},{obiekt_4}]. Szczegóły w przykładowym projekcie poniżej.

Poniżej przedstawione są dwa layouty (opisy interfejsów), w których obiekty przedstawiono symbolicznie jako Obiekt_1 … Obiekt_n.

```
1    {
2    "ty": "lout",
3    "or":"h",
4    "bg":"0,0,0",
5    "cs":[
6        {Obiekt_1},
7        {Obiekt_2},
8        {Obiekt_3},
9        ... ,
10       {Obiekt_n}
11       ]
12   }
```

```
1    {
2    "ty": "lout",
3    "or":"v",
4    "img":"3",
5    "cs":[
6        {Obiekt_1},
7        {Obiekt_2},
8        {Obiekt_3},
9        ... ,
10       {Obiekt_n}
11       ]
12   }
```

Cała struktura layoutu jako obiekt JSON rozpoczyna sie i kończy nawiasami – linie 1 i 12. W linii 2 określono, że obiekt ten to layout zawierający wyświetlany interfejs użytkownika. W linii 3 zdefiniowano układ komponentów znajdujących sie w jego wnętrzu: w kodzie po lewej horyzontalny,

a po prawej wertykalny. Linia 4 definiuje tło layoutu: po lewej jest to kolor czarny (RGB=0,0,0), a po prawej użyto wbudowanej grafiki o numerze 3. Kolejna, linia 5 rozpoczyna definicję tablicy komponentów znajdujących się wewnątrz layoutu. Elementy tablicy zawarte są w nawiasach kwadratowych i rozdzielone przecinkami. Linie od 6 do 10 definiują symbolicznie kolejne komponenty. Mogą nimi być kontrolki lub inne layouty. Linia 11 zamyka tablicę komponentów layoutu.

Przykładowe projekty interfejsów użytkownika

TextView „Hello world” i Button

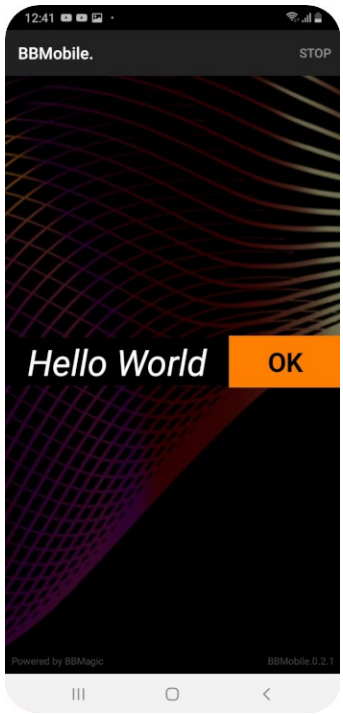
Jako pierwszy przykład zbudujmy najprostszy interfejs horyzontalny złożony z dwóch kontrolek:

- pola tekstowego z czarnym tłem i białym tekstem
- pomarańczowego przycisku z czarnym napisem 'OK'.

Oto struktura JSON opisująca ten interfejs i jej reprezentacja na ekranie:

```
{
    "ty": "lout",
    "or":"h",
    "img":"3",
    "cs":[
    {
        "ty":"TextView",
        "n":"t1",
        "w":"2",
        "te":"Hello World",
        "tc":"255,255,255",
        "bg":"0,0,0",
        "ts":"40",
        "tl":"italic"
    },
    {
        "ty":"button",
        "n":"b1",
        "te":"OK",
        "w":"1",
        "tc":"0,0,0",
        "bg":"254,127,0",
        "ts":"30",
        "tl":"bold"
    }
    ]
}
```

Struktura JSON tworząca pole tekstowe i przycisk.



Zbudowana aplikacja

Pierwszy nawias '{' i ostatni nawias '}' określają granice obiektu jakim jest nadrzędny layout reprezentujący cały widok ekranu. Kolejne rozdzielone przecinkami pary ”cecha”:”wartość” określają parametry budowanego interfejsu:

- "ty": "lout", – oznacza, że typ obiektu jako layout
- "or":"h", – ustala sposób układania kolejnych komponentów widoku: poziomy,
- "img":"3" – określa tło layoutu: będzie nim wbudowana grafika numer 3.

Następnie pojawia się tablica z komponentami widoku, która ma postać:

”cs”:[{komponent_1}, {komponent_2}, … , {komponent_n}]

”cs”to skrót od ”components”, a komponent_x opisują kolejne obiekty umieszczane kolejno na ekranie. W omawianym przykładzie mamy dwa komponenty:
pole tekstowe opisane jako:

- "ty":"TextView" – komponent jest polem tekstowym,
- "n":"t1" – ustalona unikalna nazwa kontrolki: t1,
- "w":"2" – określa wielkość kontrolki jako ułamek rozmiaru wszystkich komponentów widoku. W tym przykładzie pole tekstowe zajmie 2/3 szerokości całego widoku, ponieważ wielkość buttona wynosi 1, a pola tekstowego 2: 2/(1+2),
- "te":"Hello World" – definicja napisu znajdującego się w polu tekstowym,
- "tc":"255,255,255" – definiuje składowe RGB koloru tekstu – biały,
- "bg":"0,0,0" – definiuje składowe RGB koloru tła komponentu – czarny,
- "ts":"40"– ustala rozmiar wyświetlanego tekstu,
- "tl":"italic" – definiuje styl czcionki wyświetlanego tekstu,

i przycisk opisany tak.

- "ty":"button" – komponent jest typu button,
- "n":"b1" – unikalna nazwa kontrolki to: b1,
- "te":"OK" – napis na przycisku to OK,
- "w":"1" – określa wielkość kontrolki jako ułamek rozmiaru wszystkich komponentów widoku. W tym przykładzie button zajmie 1/3 szerokości całego widoku, ponieważ wielkość pola tekstowego wynosi 2, a buttona 1: 1/(2+1),
- "tc":"0,0,0" – określa składowe RGB koloru tekstu buttona – czarny,
- "bg":"254,127,0" – definiuje składowe RGB koloru tła buttona – pomarańczowy,
- "ts":"30" – ustala rozmiar tekstu wyświetlanego na buttonie,
- "tl":"bold" – określa styl czcionki tekstu wyświetlanego na buttonie,

Jeśli przesłany format struktury JSON nie zawierał błędów, to przez UART nadejdzie potwierdzenie otrzymania danych: **`$ok`**. Jeśli struktura zawierała błędy i nie jest możliwe wyświetlenie interfejsu zostanie wysłany komunikat: **`$nok`**.

Naciśnięcie guzika OK spowoduje wysłanie przez UART komunikatu zakończonego znakami „\r\n”:

$by,b1"1"

Polecenia sterujące interfejsem przedstawia tabela. Każde polecenie musi kończyć się znakami '\n'

Polecenie	Wynik działania
$set,b1:bg=”0,255,0”	Zmiana koloru tła buttona b1 na zielony
$set,b1:te="GO"	Zmienia tekstu na buttonie b1 na GO
$set,b1:ts="20"	Zmniejszenie rozmiaru tekstu na buttonie b1 o 10 punktów.
$set,b1:tc="255,255,255"	Zmiana koloru tekstu wyświetlanego na buttonie b1 na biały.
$set,b1:en="0"	Dezaktywacja przycisku. Naciśnięcia nie będą generować komunikatów UART.
$set,t1:te="Hi BBMobile"	Zmiana tekstu wyświetlanego w polu tekstowym t1 na: Hi BBMobile.
$set,t1:bg="150,75,0"	Zmiana koloru tła pola tekstowego t1 na brązowy.

Polecenia modyfikujące kilka parametrów jednej kontrolki można połączyć w:

$set,b1:te="GO"ts="20"bg="0,255,0"

Możliwa jest też zmiana kilku parametrów różnych kontrolek jednym poleceniem, np.:

$set,b1:te="GO"ts="20"bg="0,255,0",t1:te="Hi BBMobile"bg="150,75,0"

Producent

Stings Inwestycje Sp. z o.o.
Snycerska 34/30
30-817 Kraków