



RAK Dash Button User Guide

Final

Ashok Chilukuri

7/6/2018



Contents

1	Revision History	1
2	Acronyms	1
3	Introduction	2
4	RAK Dash Button Development Environment setup	2
4.1	Hardware Requirement	2
4.2	Software Requirement.....	2
4.3	Development environment.....	2
4.3.1	Download the source code	2
4.3.2	Installing driver	2
4.3.3	Set up Arduino IDE	3
4.3.4	The Hardware JTAG connection see below:	6
4.3.5	Run the RDB Device Application code example:.....	7
5	Device configuration in Web Application	15

1 Revision History

Version	Date	Author	Remarks
1.0	06/07/2018	Ashok	Initial release for RDB device

2 Acronyms

RDB – RAK DASH Button device
CREATER pro – CREATER pro device



3 Introduction

The document explains the requirement and procedure to setup RAK DASH button to connect to Tantiv4 Server.

4 RAK Dash Button Development Environment setup

4.1 Hardware Requirement

- 1) RAK Dash button
- 2) System
- 3) CREATR pro

4.2 Software Requirement

- 1) Standard ARM MBED CMSIS DAP driver (USB driver for CREATR pro.)
- 2) Arduino IDE (From version 1.6.5)
- 3) GitHub account

4.3 Development environment

The Dash Button use the CREATR pro to download the program, so the development environment is same as the CREATR pro.

4.3.1 Download the source code

Please download the complete source code package from GitHub from the following link
<https://github.com/Tantiv4/RakButton.git>

4.3.2 Installing driver

Step 1: First, Connect CREATR pro to the computer via Micro USB.

Step 2: If this is the first time you connect CREATR pro to your computer, you have to install the USB driver for CREATR pro. CREATR pro uses the standard ARM MBED CMSIS DAP driver, you can get the installation file and related information in the following website:

<https://developer.mbed.org/handbook/Windows-serial-configuration>

The screenshot shows the ARM mbed website with the following details:

- Header:** ARM mbed, Developer Resources, Partners, Cloud, Search mbed...
- Navigation Bar:** Hardware ▾, Documentation ▾, Code, Questions, Forum, Log In/Signup, Compiler (highlighted).
- Breadcrumbs:** Handbook » Windows serial configuration
- Section Title:** Windows serial configuration
- Text:** The mbed serial port works by default on Mac and Linux, but Windows needs a driver. These instructions explain how to setup the mbed Microcontroller to use the USB serial port on Windows.
- Step 1:** 1. Download the mbed Windows serial port driver
 - Download the installer to your PC, e.g. your desktop.
 - Download latest driver
- Note:** Note: Not Required for Windows 10!
- Step 2:** 2. Run the installer
- Text:** With your mbed plugged in, and no explorer drive windows open, run the installer.
- Table of Contents:** 1.1. Download the mbed Windows serial port driver, 1.2. Run the installer
- Recent changes:**
 - Firmware FRDM K64F
 - Debugging
 - mbed Compiler Getting Started
 - mbed SDK
 - USBMSD
 - USBDevice

Step 3: In "Download latest driver" download and install MbedWinSerial_16466.exe.



Step 4: If you are using the WIN10 system, when installing the mbed Serial Port driver, the installation fails. Please re-power the module and install the mbed Serial Port driver again. If the device driver does not display the mbed Serial Port driver after installing the driver, update the module DAP firmware.

4.3.3 Set up Arduino IDE

Step 1) From version 1.6.5, Arduino IDE supports third-party hardware. Therefore, we can use Arduino IDE to develop applications on CREATOR pro, and the examples of Arduino can run on CREATOR pro too. Arduino IDE can be downloaded in the Arduino website:

<https://www.arduino.cc/en/Main/Software>

Download the Arduino IDE

ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

HOURLY BUILDS LAST UPDATE 8 Mar 2018 20:43:37 UTC BETA BUILDS CO BETA

Step 2) When the installation is finished, open up Arduino IDE. To set up **Realtek Ameba** correctly in Arduino IDE, go to "File" -> "Preferences".

Tantiv4_RAK_Dash_button_code | Arduino 1.8.5

File Edit Sketch Tools Help

New Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q

#define RED 0
#define GREEN 1
#define BLUE 2
#define OFF 3

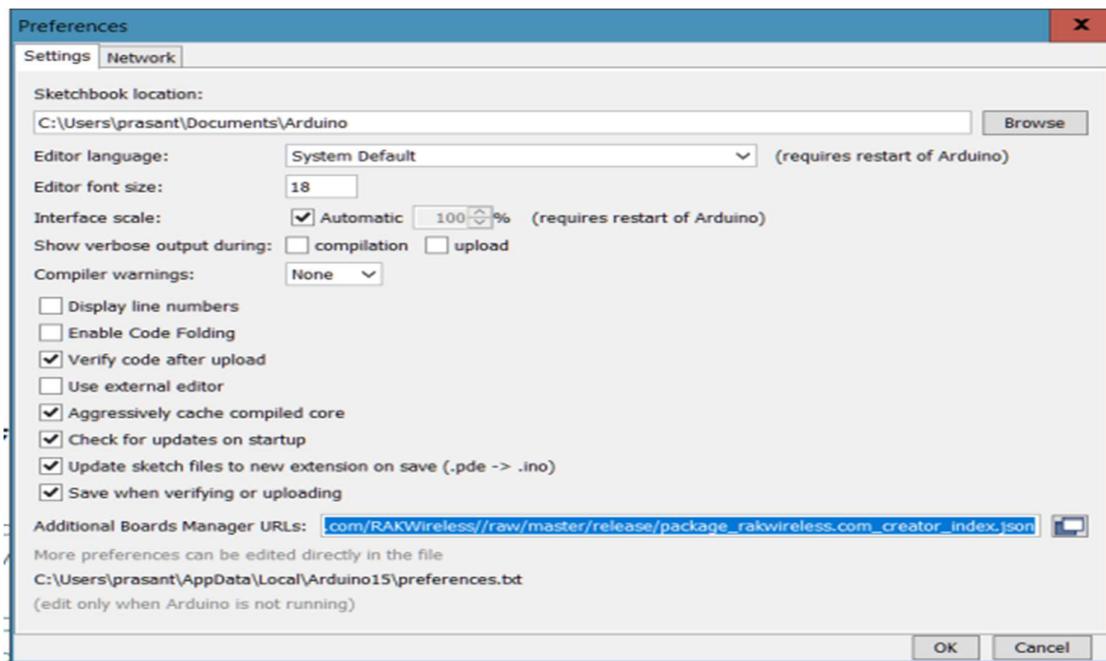
void printWifiStatus();
void led_off();
void led_ctrl(uint8_t led_num, uint8_t rgb);
void trigger_action();
char ssid[] = "Ashok_JIO"; // your network SSID (name)
char pass[] = "Tanvi@2016"; // your network password (use for WPA, or use as key for WEP)
char server[] = "api-dev.tantiv4.com"; // Name of the server we want to connect to
unsigned char test_ca_cert[] = \n"-----BEGIN CERTIFICATE-----\n" \n"MIIDSjCCAjKgAwIBAgIQRK+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/\n" \n

Done compiling

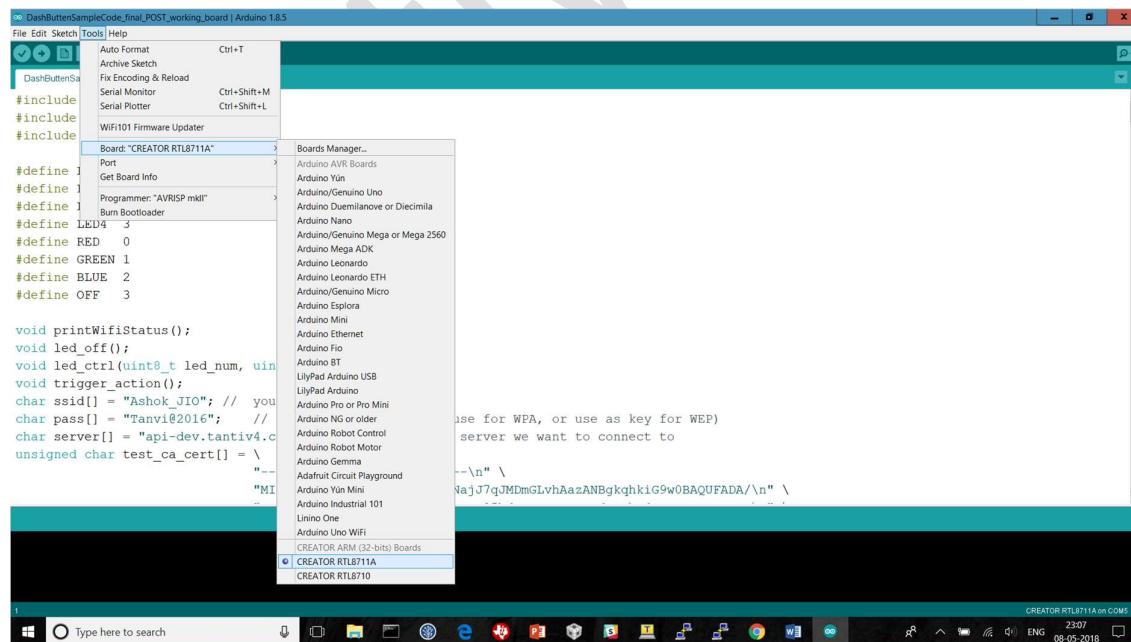


Step 3) And paste the following URL into "Additional Boards Manager URLs" field:

https://github.com/RAKWireless/CREATOR-Arduino-SDK/raw/master/release/package_rakwireless.com_creator_index.json



Step 4) Next, go to "Tools" -> "Board" -> "Boards Manager":

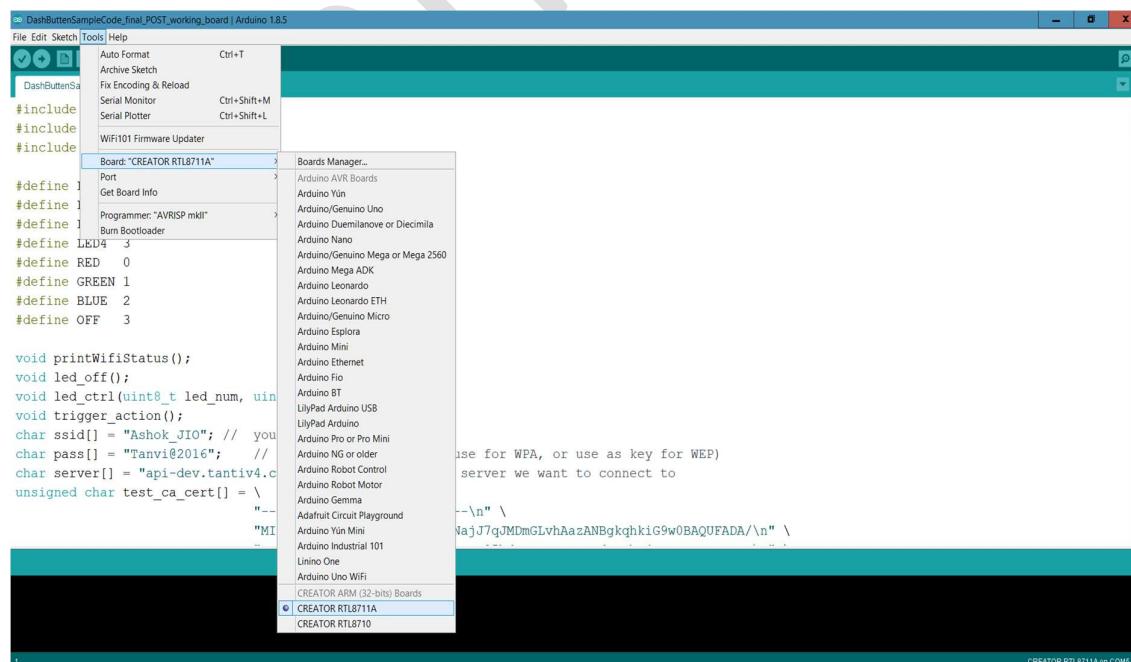




Step 5) The "Boards Manager" requires 10 seconds to refresh all hardware files (if the network is in bad condition, it may take longer). Every time the new hardware is connected, we need to reopen the Board Manager. So, we close the Boards Manager, and then open it again. Find "RAK CREATOR Boards" in the list, click "Install", then the Arduino IDE starts to download required files for CREATOR pro.



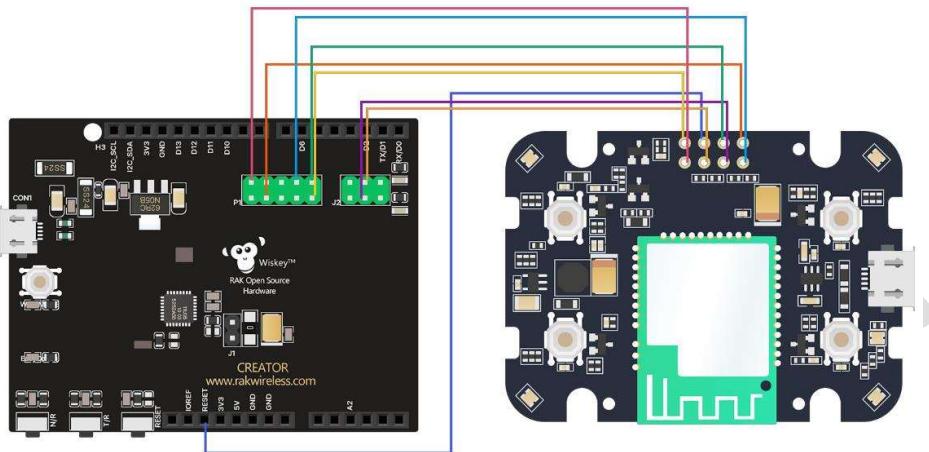
Step 6) Finally, we select CREATOR pro as current connected board in "tools" -> "Board" -> "CREATOR RTL8711A": (NOTE: RTL8195 and RTL8711 are compatible)



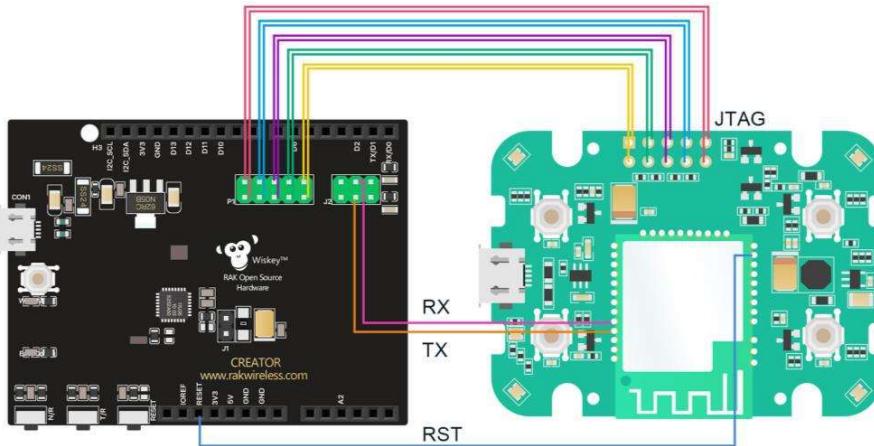


4.3.4 The Hardware JTAG connection see below

DashButton_V12 version:



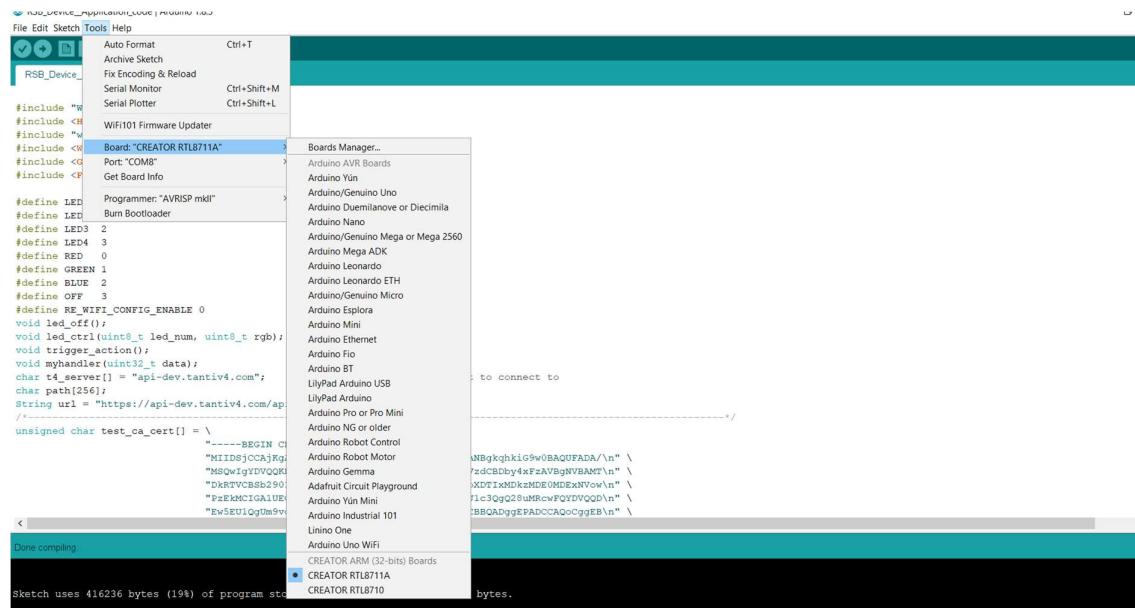
DashButton_V1.1 version:



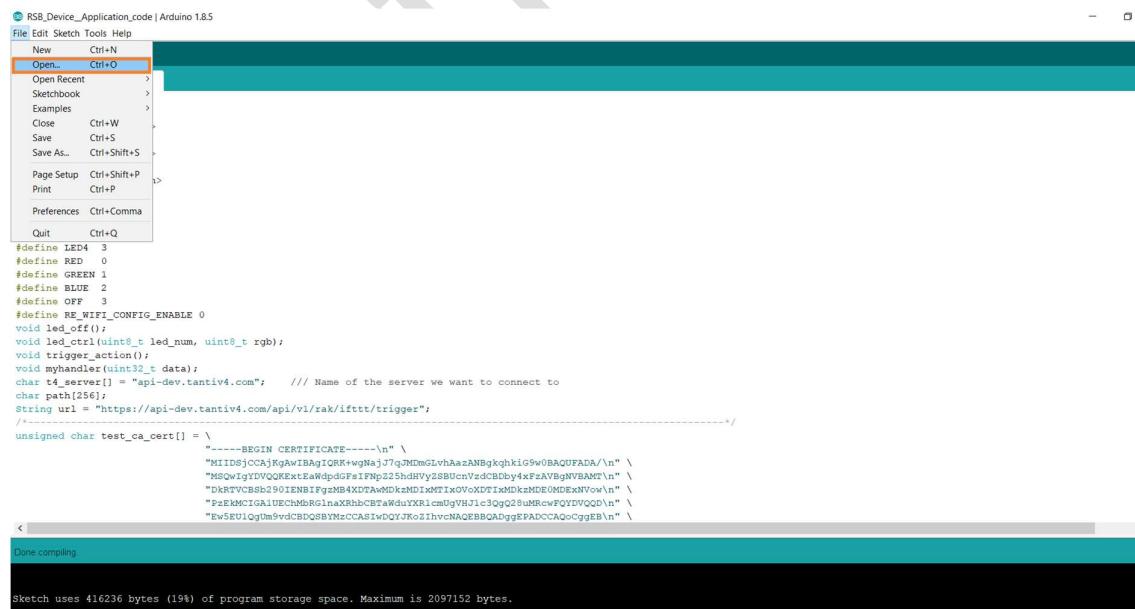


4.3.5 Run the RDB Device Application code example

Step 1) First, make sure CREATOR pro is selected in Arduino IDE: "Tools" -> "Board" -> "CREATOR RTL8711A"(NOTE:RTL8195 and RTL8711 are compatible).



Step 2) Next, Open the **RDB Device Application code** example, "File" -> "Open" -> "<RDB Device Application code file path from section 4.1>" and select "RDB_Device_Application_code" file and click on open option.





A screenshot of the Arduino IDE interface. The title bar says "RDB_Device_Application_code | Arduino 1.8.5". A modal dialog box titled "Open an Arduino sketch..." is in the foreground. It shows a list of files under "Look in: RDB_Device_Application_code". There is one item: "RDB_Device_Application_code" (07-06-2018 17:11, Arduino file). Below the list are dropdown menus for "Object name:" (set to "RDB_Device_Application_code") and "Objects of type:" (set to "All Files (*)"). To the right of these are "Open" and "Cancel" buttons. The background shows the main workspace with some code visible at the bottom.

Step 3) Modify the above source file opened with proper DeviceID and WiFi Access Point details

DeviceID acts as a signature to grant access to Tantiv4 server. WiFi Access Point details like SSID and Password needed to connect RDB to internet.

- 1) Update Device ID
 - 2) Update WiFi configuration

1) Update Device ID:

Device id is generated in Tantiv4 server. The steps are beyond scope of this document. Please contact us support@tantiv4.com for to generate device id.

In the place marked below, you can modify the device id for to communicate Tantiv4 server.



2) Update WiFi configuration:

There are two ways to configure WiFi network,

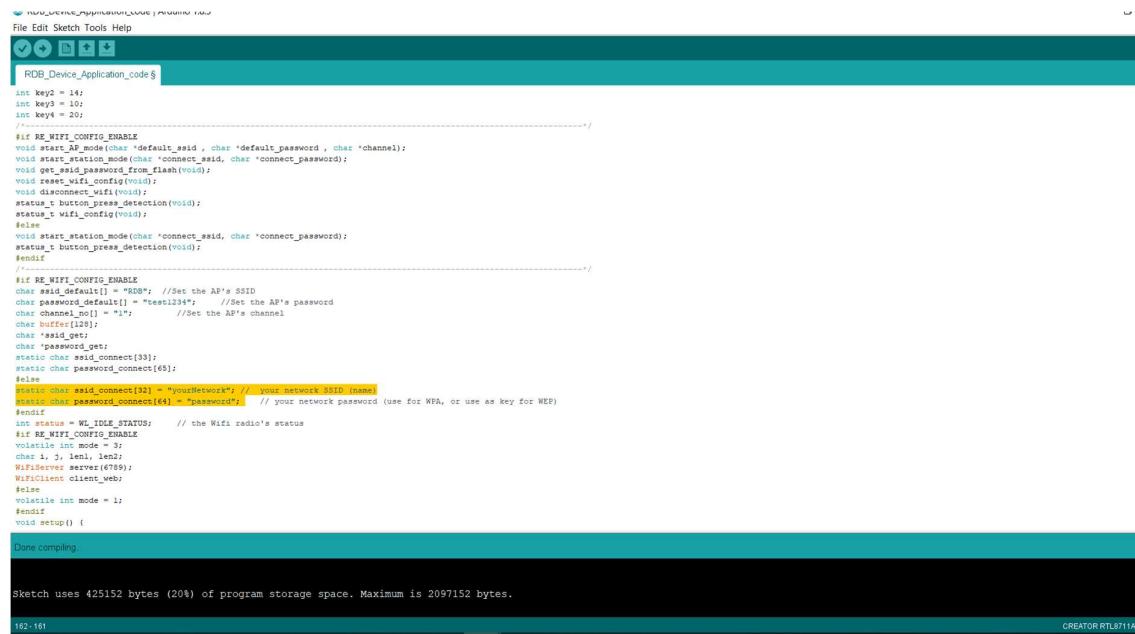
- a. Use Predefined WiFi network SSID and PASSWORD
- b. Use WiFi provisioning over HTTP

a) Use Predefined WiFi network SSID and PASSWORD

1) Make sure RE_WIFI_CONFIG_ENABLE set to 0 in code.

```
#define RE_WIFI_CONFIG_ENABLE 0
```

In the place marked below, you can modify the device needs to connect the router's SSID and password. We need to reflash/reprogramming device for to change SSID and PASSWORD in device.



The screenshot shows the Arduino IDE interface with the file 'RDB_Device_Application_code.ino' open. The code is written in C and defines WiFi configuration parameters. A large watermark 'Tantiv4' is overlaid across the code area. The code includes comments for defining WiFi mode (AP or station), default SSID, password, and channel, along with functions for connecting and disconnecting.

```
int key2 = 1;
int key3 = 10;
int key4 = 20;
/*-----*/
#ifndef RE_WIFI_CONFIG_ENABLE
void start_AP_mode(char *default_ssid, char *default_password, char *channel);
void start_station_mode(char *connect_ssid, char *connect_password);
void get_ssid_password_from_flash(void);
void reset_wifi_config(void);
void disconnect_wifi(void);
status_t button_press_detection(void);
status_t wifi_config(void);
else
void start_station_mode(char *connect_ssid, char *connect_password);
status_t button_press(void);
#endif
/*-----*/
#ifndef RE_WIFI_CONFIG_ENABLE
char ssid_def[32] = "RDB"; //Set the AP's SSID
char password_def[32] = "test1234"; //Set the AP's password
char channel_def[3] = "1"; //Set the AP's channel
char buffer[128];
char *ssid_get;
char *password_get;
static char password_connect[33];
static char password_connect165;
else
static char ssid_connect[32] = "yourNetwork" // your network SSID [name]
static char password_connect165 = "password"; // your network password (use for WPA, or use as key for WEP)
#endif
int status = WL_IDLE_STATUS; // the WiFi radio's status
#ifndef RE_WIFI_CONFIG_ENABLE
volatile int mode = 3;
char *ssid; int len;
WiFiServer server(4789);
WiFiClient client_web;
#else
volatile int mode = 1;
#endif
void setup() {
Done compiling.
Sketch uses 425152 bytes (20%) of program storage space. Maximum is 2097152 bytes.
102-161
CREATOR RTL8711A
```

b) Use WiFi provisioning over HTTP

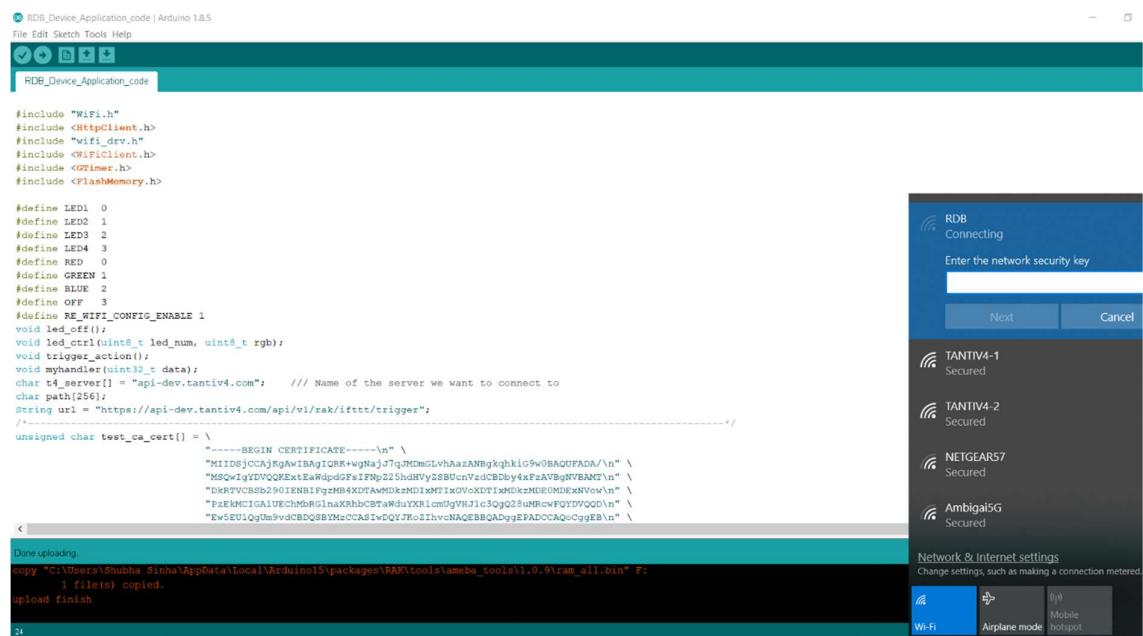
1) Make sure RE_WIFI_CONFIG_ENABLE set to 1 in code.

```
#define RE_WIFI_CONFIG_ENABLE 1
```



How to configure WiFi:

- 1) Connect any Mobile or System to WiFi network with SSID “RDB” available. Enter the “test1234” as password when prompted.



```

#include "WiFi.h"
#include <HTTPClient.h>
#include "wifi_drv.h"
#include <WiFiClient.h>
#include <GTTimer.h>
#include <FlashMemory.h>

#define LED1 0
#define LED2 1
#define LED3 2
#define LED4 3
#define RED 0
#define GREEN 1
#define BLUE 2
#define OFF 3
#define RE_WIFI_CONFIG_ENABLE 1
void led_off();
void led_ctrl(uint8_t led_num, uint8_t rgb);
void trigger_action();
void myhandler(uint32_t data);
char t4_server[] = "api-dev.tantiv4.com"; // Name of the server we want to connect to
char path[256];
String url = "https://api-dev.tantiv4.com/api/v1/rak/ifttt/trigger";
/*-----BEGIN CERTIFICATE-----\n" \
"MIIDJSCCAjBgkqhkiBQHNaJzTqJNDmLlvhaazANBgkqhkiG9w0BAQUFADw/\n" \
"MSQwIgYDQKEextEadpdgPeTFnpZ25hdWVzSBUDnVzdCB0by4xPxAVBqgVBAM/\n" \
"DRRTVCBSd290tENB1PgM84X7Am0kEMD1MTIxOVxDT1xMDkzMDR0NDEXNvow/\n" \
"PExEKNCIGA1UEChbGRlnaxRhbCERtaWduYXRLcm9gVHJic3QoQ28uMRgwFOYDVQOD/\n" \
"EwSEU1QgUm9vdCBQSBVmzCCAS1wDQYJKoZIhvCNQEBBQAQdgEPADCCAQcCggEB\n" \
"-----END CERTIFICATE-----\n"

```

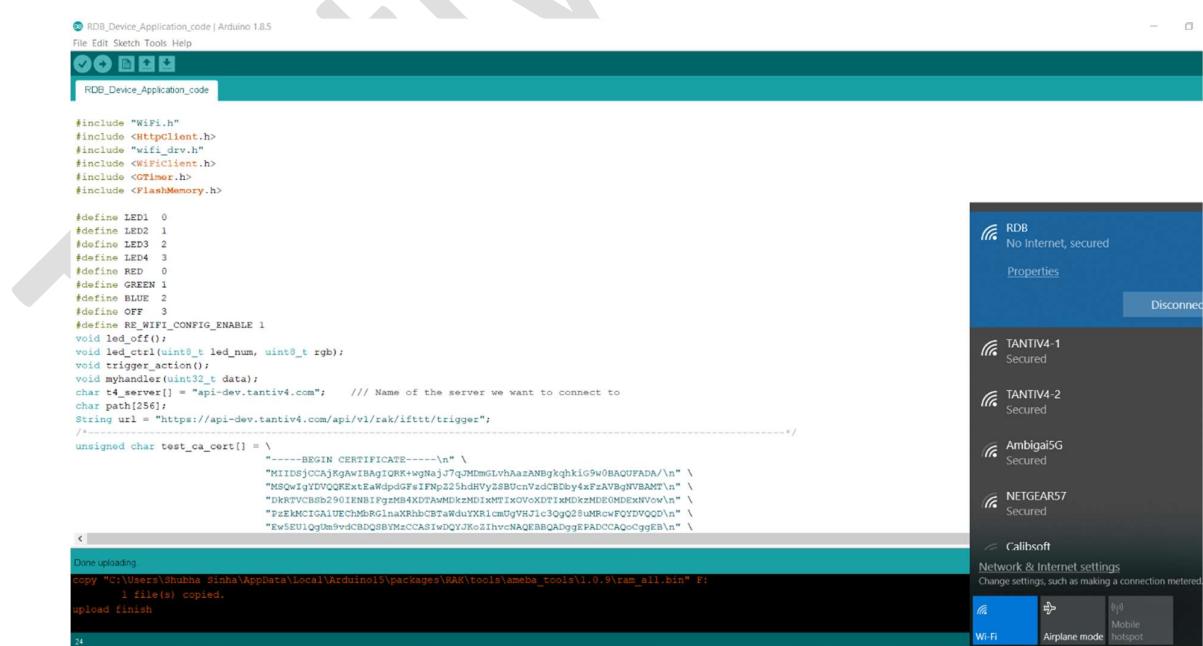
Done uploading
copy "C:\Users\Shubha_Sinha\AppData\Local\Arduino15\packages\RAA\tools\ameba_tools\1.0.9\ram_all.bin" F:
1 file(s) copied.
upload finish

24

RDB
Connecting
Enter the network security key
Next Cancel

TANTIV4-1 Secured
TANTIV4-2 Secured
NETGEAR5 Secured
Ambigal5G Secured

Network & Internet settings
Change settings, such as making a connection metered.
Wi-Fi Mobile
Airplane mode hotspot



```

#include "WiFi.h"
#include <HTTPClient.h>
#include "wifi_drv.h"
#include <WiFiClient.h>
#include <GTTimer.h>
#include <FlashMemory.h>

#define LED1 0
#define LED2 1
#define LED3 2
#define LED4 3
#define RED 0
#define GREEN 1
#define BLUE 2
#define OFF 3
#define RE_WIFI_CONFIG_ENABLE 1
void led_off();
void led_ctrl(uint8_t led_num, uint8_t rgb);
void trigger_action();
void myhandler(uint32_t data);
char t4_server[] = "api-dev.tantiv4.com"; // Name of the server we want to connect to
char path[256];
String url = "https://api-dev.tantiv4.com/api/v1/rak/ifttt/trigger";
/*-----BEGIN CERTIFICATE-----\n" \
"MIIDJSCCAjBgkqhkiBQHNaJzTqJNDmLlvhaazANBgkqhkiG9w0BAQUFADw/\n" \
"MSQwIgYDQKEextEadpdgPeTFnpZ25hdWVzSBUDnVzdCB0by4xPxAVBqgVBAM/\n" \
"DRRTVCBSd290tENB1PgM84X7Am0kEMD1MTIxOVxDT1xMDkzMDR0NDEXNvow/\n" \
"PExEKNCIGA1UEChbGRlnaxRhbCERtaWduYXRLcm9gVHJic3QoQ28uMRgwFOYDVQOD/\n" \
"EwSEU1QgUm9vdCBQSBVmzCCAS1wDQYJKoZIhvCNQEBBQAQdgEPADCCAQcCggEB\n" \
"-----END CERTIFICATE-----\n"

```

Done uploading
copy "C:\Users\Shubha_Sinha\AppData\Local\Arduino15\packages\RAA\tools\ameba_tools\1.0.9\ram_all.bin" F:
1 file(s) copied.
upload finish

24

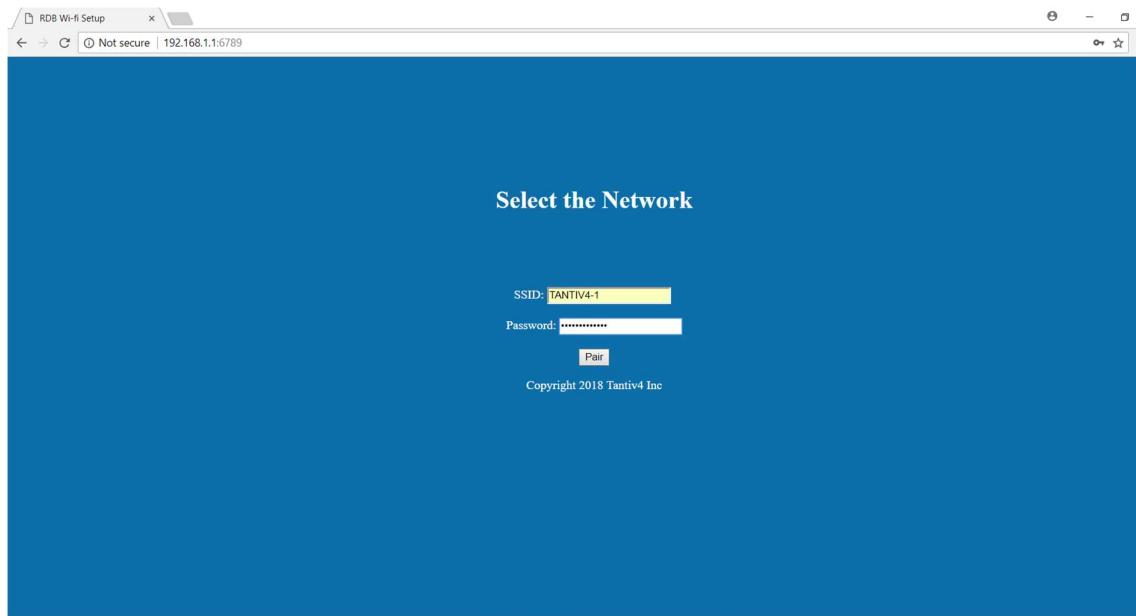
RDB
No Internet, secured
Properties
Disconnected

TANTIV4-1 Secured
TANTIV4-2 Secured
Ambigal5G Secured
NETGEAR5 Secured
Calibsoft

Network & Internet settings
Change settings, such as making a connection metered.
Wi-Fi Mobile
Airplane mode hotspot



- 2) After connection to WiFi network is established, open any web browser and type "**192.168.1.1:6789**" and enter, a web page as shown in the screenshot.



- 3) Enter working Wi-Fi network SSID and password in the box available as shown in the screenshot above.
- 4) Click on the "Pair" button to send the Wi-Fi network details to device. At this time the web page will be closed.
- 5) Device will try to connect to given WiFi network and get IP from router. All button LED's will turn GREEN for half second to indicate device has got IP from router.

Note:

- 1) Device supports short press only and will send trigger messages to server.
- 2) There is no LED indication to show device in AP mode or station mode.
- 3) If Device is in Access Point mode, button press won't work until access point is configured.

LEDs behavior

- 1) All LED's turn Blue for half second - Indicate device is in AP mode.
- 2) All LED's turn Green for half second – Indicate device got IP from router.



Step 4) Next, we compile the code directly, click "Sketch" -> "Verify/Compile", Or you can click the icon in the upper left corner.

RDB_Device_Application_code | Arduino 1.8.5

File Edit Sketch Tools Help

Sketch

- Verify/Compile Ctrl+R
- Upload Ctrl+U
- Upload Using Programmer Ctrl+Shift+U
- Export compiled Binary Ctrl+Alt+S
- Show Sketch Folder Ctrl+K
- Include Library >
- Add File...

```

if (c == '\n' && currentLineIsBlank()) {
    while (client_web.available()) {
        char c = client_web.read();
        // if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
    }

    if (c == '\n' && currentLineIsBlank()) {
        Serial.println("Sending response");
        // send a standard http response header
        client_web.println("HTTP/1.1 200 OK");
        client_web.println("Content-Length: 2048 - 167");
        client_web.println("Content-Type: text/html\r\n\r\n");
        client_web.println("<!DOCTYPE html>\r\n");
        client_web.println("<html>\r\n<head>\r\n");
        client_web.println("<meta rel='icon' href='data:base64," + icon);
        client_web.println("<meta charset='UTF-8'>\r\n");
        client_web.println("<meta name='viewport' content='width=device-width, initial-scale=1'>\r\n");
        client_web.println("<title>RDB Wi-Fi Setup</title>\r\n</head>\r\n");
        client_web.println("<body style='background-color:#00d6ff; color:ffff; text-align:center;padding: 10px;'>\r\n");
        client_web.println("<br><br><br><br><br>\r\n");
        client_web.println("<div>Select the Network</div>\r\n");
        client_web.println("<form class='form' action='http://192.168.1.1:6789/' method='POST'>\r\n<div>\r\n");
        // client.println("<label>Network Available:</label>\r\n");
        // client.println("<select id='ssid'>\r\n");
        client_web.println("<div>\r\n<label>SSID:</label>\r\n");
        client_web.println("<input type='text' id='ssid' name='ssid'>\r\n");
        client_web.println("<div>\r\n<label>Select:</label>\r\n");
        client_web.println("<div>\r\n<label>Password:</label>\r\n");
        client_web.println("<input type='password' name='password' id='password'>\r\n");
    }
}

```

Done compiling

Step 5) Afterwards, we will upload the compiled code to RDB device via CREATOR pro. Please make sure CREATOR pro and RDB device is connected to your system, then click "Sketch" -> "Upload" Or You can also click the shortcut icon.

RDB_Device_Application_code | Arduino 1.8.5

File Edit Sketch Tools Help

Sketch

- Verify/Compile Ctrl+R
- Upload Ctrl+U
- Upload Using Programmer Ctrl+Shift+U
- Export compiled Binary Ctrl+Alt+S
- Show Sketch Folder Ctrl+K
- Include Library >
- Add File...

```

if (c == '\n' && currentLineIsBlank()) {
    while (client_web.available()) {
        char c = client_web.read();
        // if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
    }

    if (c == '\n' && currentLineIsBlank()) {
        Serial.println("Sending response");
        // send a standard http response header
        client_web.println("HTTP/1.1 200 OK");
        client_web.println("Content-Length: 2048 - 167");
        client_web.println("Content-Type: text/html\r\n\r\n");
        client_web.println("<!DOCTYPE html>\r\n");
        client_web.println("<html>\r\n<head>\r\n");
        client_web.println("<meta rel='icon' href='data:base64," + icon);
        client_web.println("<meta charset='UTF-8'>\r\n");
        client_web.println("<meta name='viewport' content='width=device-width, initial-scale=1'>\r\n");
        client_web.println("<title>RDB Wi-Fi Setup</title>\r\n</head>\r\n");
        client_web.println("<body style='background-color:#00d6ff; color:ffff; text-align:center;padding: 10px;'>\r\n");
        client_web.println("<br><br><br><br><br>\r\n");
        client_web.println("<div>Select the Network</div>\r\n");
        client_web.println("<form class='form' action='http://192.168.1.1:6789/' method='POST'>\r\n<div>\r\n");
        // client.println("<label>Network Available:</label>\r\n");
        // client.println("<select id='ssid'>\r\n");
        client_web.println("<div>\r\n<label>SSID:</label>\r\n");
        client_web.println("<input type='text' id='ssid' name='ssid'>\r\n");
        client_web.println("<div>\r\n<label>Select:</label>\r\n");
        client_web.println("<div>\r\n<label>Password:</label>\r\n");
        client_web.println("<input type='password' name='password' id='password'>\r\n");
    }
}

```

Sketch uses 425152 bytes (20%) of program storage space. Maximum is 2097152 bytes.

Done compiling



Step 6) Again, during the uploading procedure the IDE prints messages. Uploading procedure requires respectively longer time (depends on system - about 1 minute to 2 minutes). When upload completed, the "upload finish" message is printed.

```
RDB_Device_Application_code
...
if (client_web.available()) {
    Serial.println("Client connected");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
}

if (client_web.connected()) {
    while (client_web.available()) {
        char c = client_web.read();
        // if you receive a carriage return at the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
        if (c == '\n' & currentLineIsBlank) {
            Serial.println("Sendin response");
            // send a standard http response header
            client_web.println("HTTP/1.1 200 OK");
            client_web.println("Content-Length: 2048 - 167");
            client_web.println("Content-Type: text/html\r\n\r\n");
            client_web.println();
            client_web.println();
            client_web.println("<!-- Wi-Fi Setup --&gt;\r\n&lt;html&gt;\r\n&lt;head&gt;\r\n&lt;meta charset='UTF-8'&gt;\r\n&lt;meta name='viewport' content='width=device-width, initial-scale=1'&gt;\r\n&lt;/head&gt;\r\n&lt;body style='background-color:#00fear; color:#fff; text-align: center; padding: 10px;'&gt;\r\n&lt;h1&gt;Select the Network&lt;/h1&gt;\r\n&lt;form class='form' action='http://192.168.1.1:6789' method='POST'&gt;\r\n&lt;div&gt;\r\n&lt;label for='ssid'&gt;Select the network&lt;/label&gt;\r\n&lt;select id='ssid'&gt;\r\n&lt;/select&gt;\r\n&lt;/div&gt;\r\n&lt;div&gt;\r\n&lt;label for='password'&gt;Enter the password&lt;/label&gt;\r\n&lt;input type='password' name='password' id='password'&gt;\r\n&lt;/div&gt;\r\n&lt;/form&gt;\r\n&lt;/body&gt;\r\n&lt;/html&gt;");
            client_web.println("Content-Type: application/x-www-form-urlencoded\r\n");
            client_web.println("Content-Length: 167\r\n");
            client_web.println("Connection: close\r\n");
            client_web.println("Cache-Control: no-store, no-cache, must-revalidate, pre-check=0, post-check=0, max-age=0\r\n");
            client_web.println("Pragma: no-cache\r\n");
            client_web.println("Date: Sun, 31 Dec 2020 23:59:59 GMT\r\n");
            client_web.println("Expires: Sun, 31 Dec 2020 23:59:59 GMT\r\n");
            client_web.println("Last-Modified: Sun, 31 Dec 2020 23:59:59 GMT\r\n");
            client_web.println("Server: Apache/2.4.41 (Ubuntu)\r\n");
            client_web.println("Content-Type: text/html\r\n\r\n");
            client_web.println("<!-- Wi-Fi Setup --&gt;\r\n&lt;html&gt;\r\n&lt;head&gt;\r\n&lt;meta charset='UTF-8'&gt;\r\n&lt;meta name='viewport' content='width=device-width, initial-scale=1'&gt;\r\n&lt;/head&gt;\r\n&lt;body style='background-color:#00fear; color:#fff; text-align: center; padding: 10px;'&gt;\r\n&lt;h1&gt;Select the Network&lt;/h1&gt;\r\n&lt;form class='form' action='http://192.168.1.1:6789' method='POST'&gt;\r\n&lt;div&gt;\r\n&lt;label for='ssid'&gt;Select the network&lt;/label&gt;\r\n&lt;select id='ssid'&gt;\r\n&lt;/select&gt;\r\n&lt;/div&gt;\r\n&lt;div&gt;\r\n&lt;label for='password'&gt;Enter the password&lt;/label&gt;\r\n&lt;input type='password' name='password' id='password'&gt;\r\n&lt;/div&gt;\r\n&lt;/form&gt;\r\n&lt;/body&gt;\r\n&lt;/html&gt;");
            client_web.println();
            client_web.println();
        }
    }
}
}

Done uploading.
copy "C:\Users\Subhra\Downloads\TANTIV4\Arduino\5\packages\RAK\tools\raakme_tools\1.0.0\ram_all.bin" F:
1 file(s) copied.
upload finish
401</pre>
```

Step7) Finally, Repower the Dash button, and you can see related information shown in serial monitor.

```
=====
Check boot type form eFuse
SPI calibration
Image length: 0x1a88, Image Addr: 0x10000bc8
Image Validate OK, Going jump to Image1
BOOT from Flash:YES
SPI calibration
First available window
Baud(0): auto length:0 Delay startt:0 Delay end:63
[SPIF Err]Spi0tWNCalStore: The flash memory@0x9000 = 0x0) is not able to be write, Erase it first!===== Enter Image 1 ====
SPI calibration
First available window
Baud(2): auto length:31 Delay startt:0 Delay end:63
[SPIF Err]Spi0tWNCalStore: The flash memory@0x9000 = 0x0) is not able to be write, Erase it first!
SR Controller Init
Test: 0 != No match addrs 0xac8d38 => 0xf != 0xc
Test: 0 != No match addrs 0x1ff7c0 => 0x5 != 0x5
0x5 addrs 0x0 INVALID
0x5 addrs 0x0 INVALID

Image 1
Flash Image1 Addr: 0xb000, Len 213000, Load to SRAM 0x10006000
Image length: 0x1c088, Image Addr: 0x30000000
Img Sign: RTIMain, InfStart @ 0x1000049
===== Enter Image 2 =====
interface 0 is initialized
interface 1 is initialized

Initializing WiFi ...
WIFI initialized ...
Attempting to start AP with SSID: RDB
LwIP_DNCM: drop stop.
Deinitializing WiFi ...
WIFI deinitialized
Initializing WiFi ...
WIFI initialized

Starting AP ...
AP start
IP Address: 192.168.1.1
NetMask: 255.255.255.0
Gateway: 192.168.1.1
SSID: RDB
SSID: RDB
IP Address: 192.168.1.1
signal strength (RSSI):10 dBm
To see this page in action, open a browser to http://192.168.1.1
```



```

COM6

DIOV2 - EMMC FIRMWARE
SPI calibration
Find the available window
Baud:3; auto_length:0; Delay start:0; Delay end:63
[SPIF Err]SpicNVMeCalStore: The flash memory(0x9080 = 0x0) is not able to be write, Erase it first!===== Enter Image 1 ====
SPI calibration
Find the available window
Baud:2; auto_length:2; Delay start:0; Delay end:63
[SPIF Err]SpicNVMeCalStore: The flash memory(0x90b0 = 0x0) is not able to be write, Erase it first!
SDR Controller Init
Test 0: No match addr 0xc9d38 => 0xf != 0xc
Test 0: No match addr 0x1f7c10 => 0x5 != 0x8
OTA addr 0xd0 INVALID

load NEW fw: 0
Flash Image0:Addr 0xb000, Len 205700, Load to SRAM 0x10006000
Image3 length: 0x2638, Image3 Addr: 0x30000000
Img2 Sign: RTKWin, InfraStart @ 0x10006049
===== Enter Image 2 ====
interface 0 is initialized
interface 1 is initialized

Initializing WIFI ...
WIFI initialized
Attempting to connect to SSID: TANTIV4-1

RTL8195A[Driver]: set ssid [TANTIV4-1]

RTL8195A[Driver]: start auth to 6c:72:20:f9:f6:b4

RTL8195A[Driver]: auth success, start assoc

RTL8195A[Driver]: association success(res=5)

RTL8195A[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4)

RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4) keyid:1

Interface 0 IP address : 192.168.0.110SSID: TANTIV4-1
IP Address: 192.168.0.110
signal strength (RSSI):-49 dBm
192.168.0.110

 Autoscroll
 No line ending
 9600 baud
 Clear

```



```

COM6
|_
Verify requested for (Depth 0):
ceThis certificate has no flags
connected to server
Serial.printlnlnbr_no =
1
requesting URL: HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
content-type: application/json
content-length: 20
set-cookie: connect.sid=s%3AgREJFNUkeOROsNkguZOLYbDwjnUm4Kk.BNJoAh80aYqM%2FVfJ5vS45BLrjrwA145i2n1Mb0x4M; Path=/; Expires=Thu, 07 Jun 2018 11:16:44 GMT; HttpOnly
Date: Thu, 07 Jun 2018 11:15:04 GMT
Connection: close

>{"status":"success"
Success
key3
Verify requested for (Depth 2):
ceThis certificate has no flags

Verify requested for (Depth 1):
ceThis certificate has no flags

Verify requested for (Depth 0):
ceThis certificate has no flags
connected to server
Serial.printlnlnbr_no =
3
requesting URL: HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
content-type: application/json
content-length: 20
set-cookie: connect.sid=s%3AJE2PcbxpAdCerhT1CV0r2zs07AasoUL8.M6Gbkcle9Vw%2PHRC3Yvyxp%2ByNhSMzhCzu9892Y6G07Fc; Path=/; Expires=Thu, 07 Jun 2018 11:17:10 GMT; HttpOnly
Date: Thu, 07 Jun 2018 11:15:30 GMT
Connection: close

>{"status":"success"
Success
 Autoscroll
 No line ending
 9600 baud
 Clear

```

Step 8) Device LED behavior to notify device action

Button press:

If any device button is configured and pressed the corresponding LED will turn BLUE until it successfully sends the trigger to Tantiv4 server or timeout. The LED will turn GREEN for one second if it is successful. Otherwise LED will turn RED for one second.

If device buttons are not configured and if any button is pressed, the corresponding LED will turn BLUE until it successfully sends the trigger to Tantiv4 server or timeout. LED will turn RED for one second in both success or failure cases.



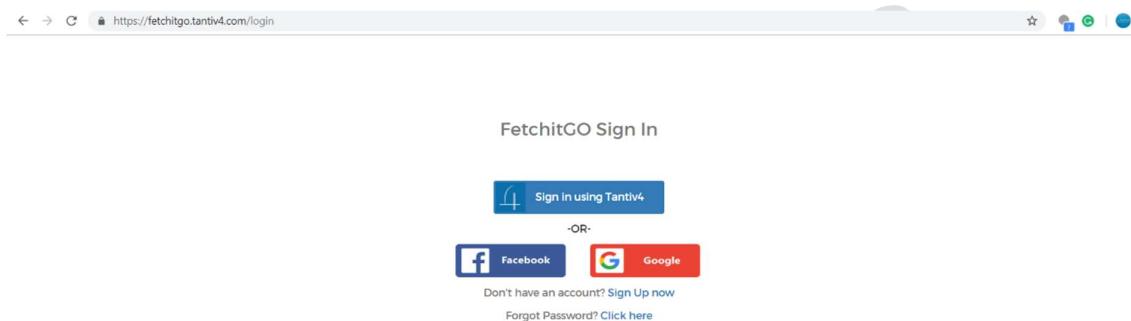
WiFi reconfiguration:

If WiFi provisioning is enabled, pressing and holding button “2” for 10 seconds will reset WiFi configuration detail from device flash memory. All button LEDs will turn BLUE for half second to indicate device is in Access Point mode.

Step 9) If you encounter any problem, please contact support@tantiv4.com.

5 Device configuration in Web Application

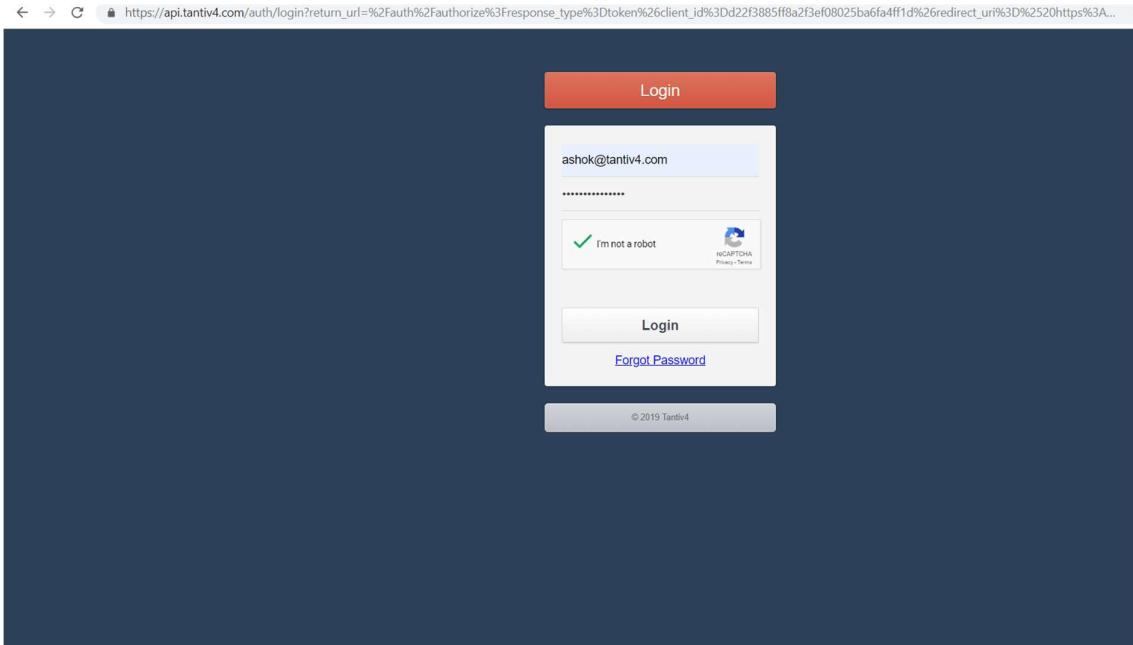
Step 1: Signup



Create your account at fetchitgo.tantiv4.com and **Sign Up** through Tantiv4



Step 2: SignIn

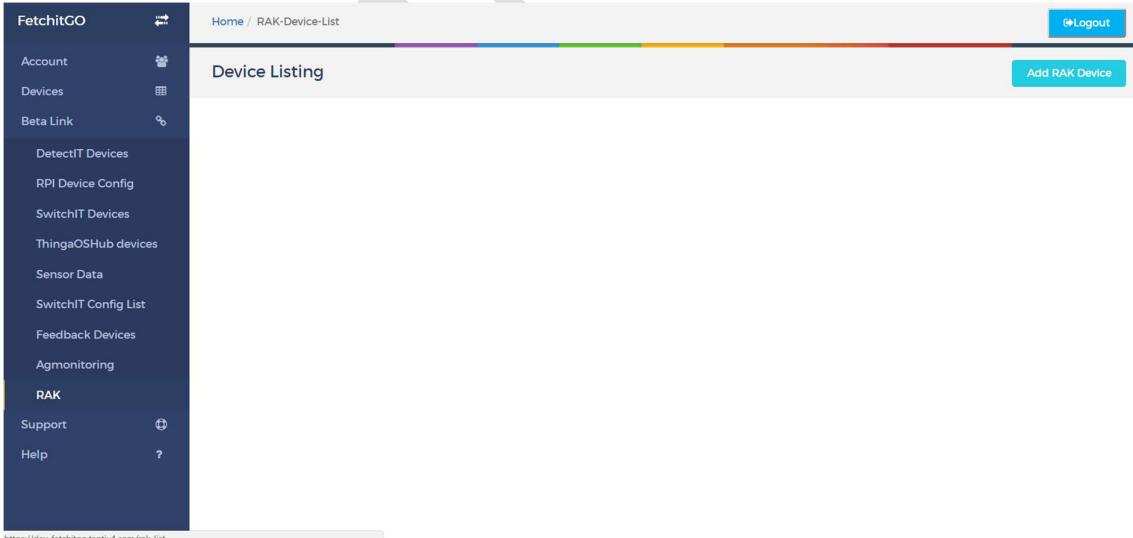


The screenshot shows a login page with a dark blue background. At the top right is a large orange 'Login' button. Below it is a white input field containing 'ashok@tantiv4.com'. Underneath is a password field with several dots. A reCAPTCHA checkbox is checked, showing 'I'm not a robot'. To its right is the reCAPTCHA logo and the text 'reCAPTCHA Privacy - Terms'. Below the form is a 'Login' button and a 'Forgot Password' link. At the bottom center is a small grey footer with the text '© 2019 Tantiv4'.

Once you have signed up, go head and login

Step 3: Add RAK Device

-> Click on **Beta Link** and click on **RAK** option it will show **Device List page**.



The screenshot shows a navigation sidebar on the left with the title 'FetchitGO'. It includes links for Account, Devices, Beta Link, DetectIT Devices, RPI Device Config, SwitchIT Devices, ThingaOSHub devices, Sensor Data, SwitchIT Config List, Feedback Devices, Agmonitoring, RAK, Support, and Help. The 'RAK' link is highlighted with a yellow background. The main content area has a header 'Home / RAK-Device-List' with a 'Logout' button on the right. Below the header is a 'Device Listing' section with a 'Add RAK Device' button. The URL at the bottom of the page is 'https://dev-fetchitgo.tantiv4.com/rak-list'.



-> In **Devices listing page** by clicking on “**Add RAK Device**” tab you can add device either by entering serial number/customer ID or by Scanning QR code.

FetchitGO

Home / Device / Scan

Logout

Device QR Scan

Scan the QR-Code of Device

OR

Enter Serial Number (OR) Customer ID and Device Friendly Name

Serial Number/Customer ID : 243484E030418000025

Device Friendly Name : Enter Device Friendly Na

Save

-> Click on **save** button after entered Serial number/customer ID or by Scanning QR code. Device will be added to your account.

FetchitGO

Home / RAK-Device-List

Logout

Add RAK Device

Device Listing

Name : rak-1	Battery : 100%
Device ID : 54463245484E030418000025	
IFTTT	Delete

-> When devices get added successfully device ID and device name is shown.



Step 4: Device-Configuration

-> Click on the **IFTTT** option at the bottom of the added device box. You will be directed to the Device-Action viewing page with 4 buttons.

The screenshot shows the 'Device Action Viewing' section for a device named 'rak-1'. It displays four buttons, each with a note field. Buttons 1, 2, and 3 have the note 'Select a recipe'. Button 4 has the note 'IFTTT' and a dropdown menu set to 'Set IFTTT Action Notes'. A 'Save' button is located at the bottom right of the form.

->Click on **button** it will re-direct to **IFTTT** page.

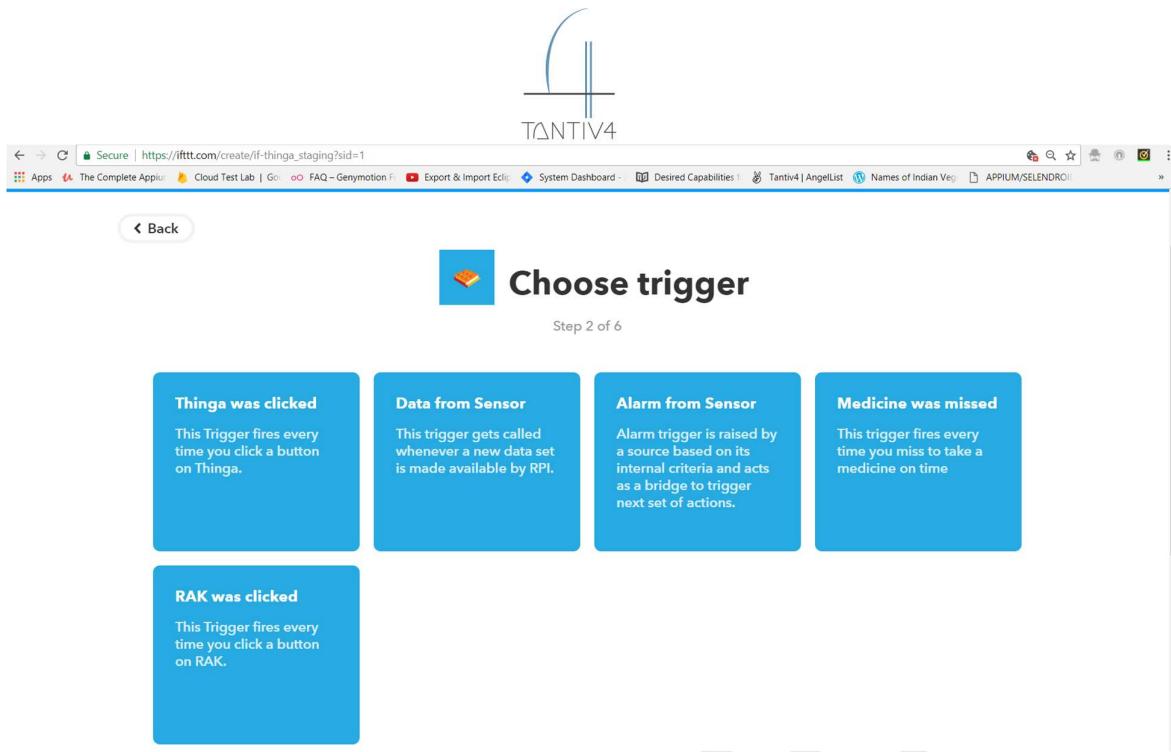
->Create **New Applet** in IFTTT

The screenshot shows the IFTTT website with a blue header bar containing the text 'New Applet'. The main area is white with a large, bold, black text 'if +this then that'.

Want to do more? Partner with IFTTT, work with everyone. Build and publish your service and Applets on our platform.

[Partner with IFTTT](#)

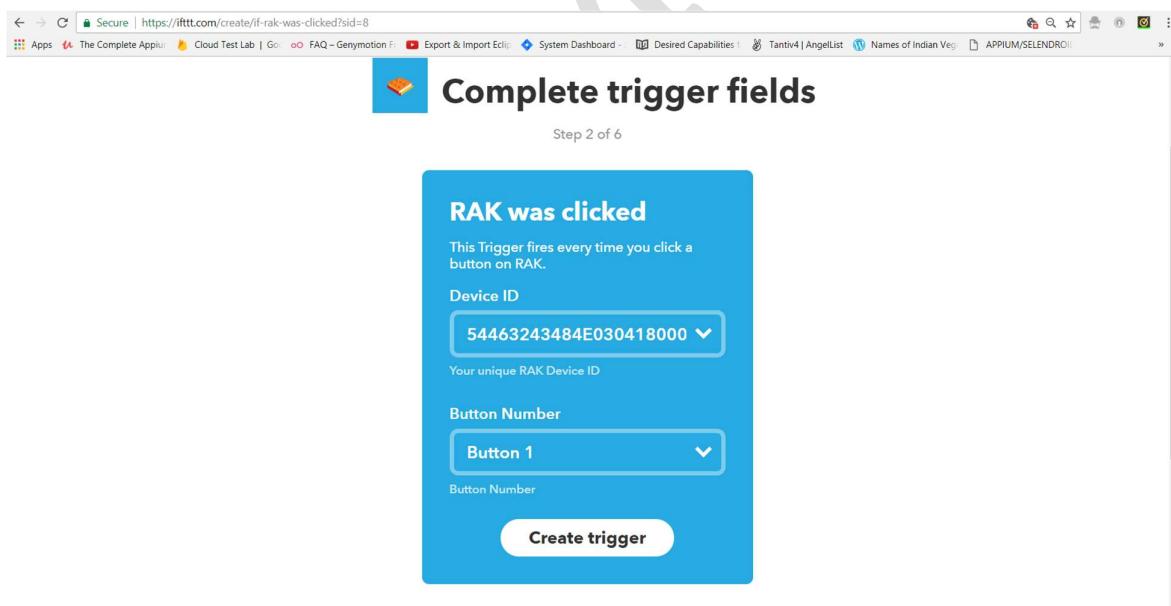
-> Click on **+ this** and choose service as **Thinga** it will show **Choose Trigger page**.



The screenshot shows the 'Choose trigger' step of an IFTTT workflow. The title 'Choose trigger' is at the top, followed by 'Step 2 of 6'. There are five trigger options listed:

- Thinga was clicked**: This Trigger fires every time you click a button on Thinga.
- Data from Sensor**: This trigger gets called whenever a new data set is made available by RPI.
- Alarm from Sensor**: Alarm trigger is raised by a source based on its internal criteria and acts as a bridge to trigger next set of actions.
- Medicine was missed**: This trigger fires every time you miss to take a medicine on time.
- RAK was clicked**: This Trigger fires every time you click a button on RAK.

->Click on **RAK was clicked** it will show **Complete Trigger fields** page with device id and button number.



The screenshot shows the 'Complete trigger fields' step of the IFTTT workflow, specifically for the 'RAK was clicked' trigger. The title 'Complete trigger fields' is at the top, followed by 'Step 2 of 6'. The form contains the following fields:

- RAK was clicked**: A summary of the trigger: 'This Trigger fires every time you click a button on RAK.'
- Device ID**: A dropdown menu showing '54463243484E030418000'.
- Button Number**: A dropdown menu showing 'Button 1'.
- Create trigger**: A button at the bottom right.

->Configure the device with Device ID, Button number and click on **Create Trigger**.

->Select +that it will show **Choose Action Services** page and Select any services like **Gmail, Slack** etc.



References

Tantiv4 Inc.