



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) CREATER pro

4.2 Software Requirement

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

4.3 Development environment

The Dash Button use the CREATOR pro to download the program, so the development environment is same as the CREATOR 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 CREATOR pro to the computer via Micro USB.

Step 2: If this is the first time you connect CREATOR pro to your computer, you have to install the USB driver for CREATOR pro. CREATOR 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>

A screenshot of the ARM mbed website. The top navigation bar includes links for 'Developer Resources', 'Partners', 'Cloud', 'Search mbed...', 'Hardware', 'Documentation', 'Code', 'Questions', 'Forum', 'Log In/Signup', and 'Compiler'. Below the navigation is a breadcrumb trail: 'Handbook » Windows serial configuration'. The main content area has a teal header 'Windows serial configuration'. The text explains that the mbed serial port works by default on Mac and Linux, but Windows needs a driver. It provides instructions to download and run the mbed Windows serial port driver. A red box highlights the 'Download latest driver' button. To the right is a sidebar titled 'Recent changes' with links to 'Firmware FRDM K64F', 'Debugging', 'mbed Compiler Getting Started', 'mbed SDK', 'USBMSD', and 'USBDevice'. A table of contents on the right lists steps 1.1 and 1.2.

Windows serial configuration

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.

1. Download the mbed Windows serial port driver

Download the installer to your PC, e.g. your desktop.

[Download latest driver](#)

Note: Not Required for Windows 10!

2. Run the installer

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
2. 2. Run the installer

Recent changes

Firmware FRDM K64F

cmsis, K64F-Firmware

Debugging

mbed Compiler Getting Started

compiler, editor, tutorial

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>

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

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

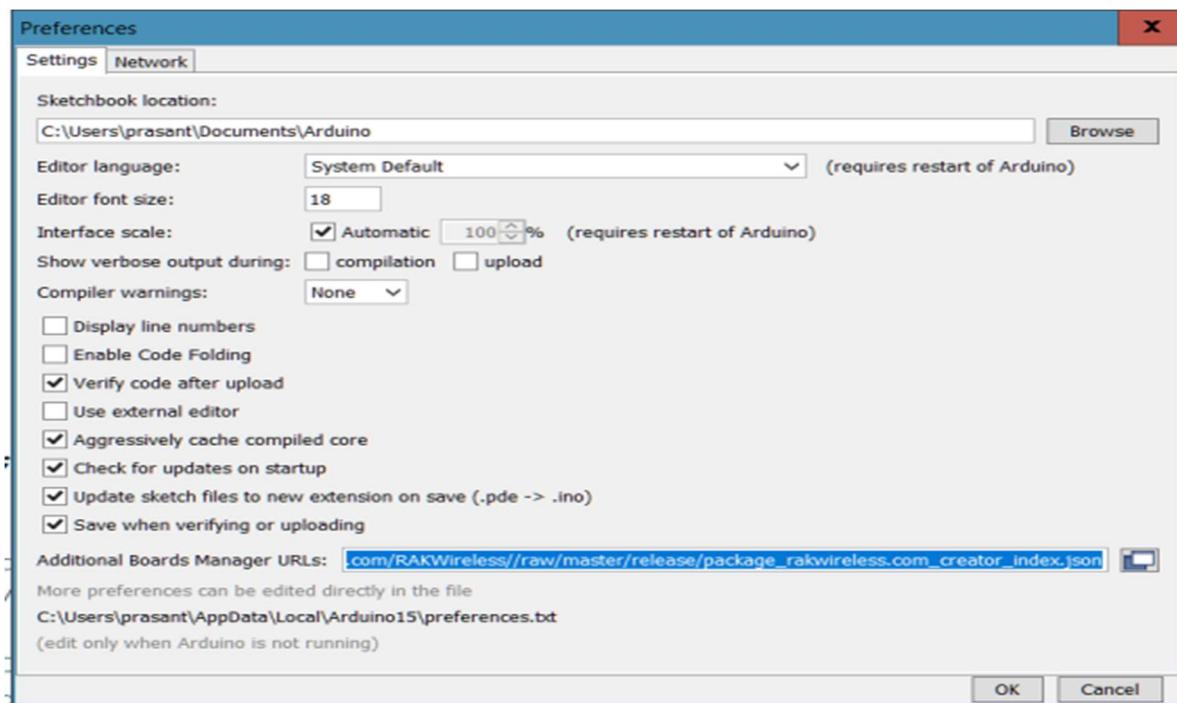
void printWifiStatus();
void led_off();
void led_ctrl(uint8_t led_num, uint8_t rgb);
void trigger_action();
char ssid[] = "Ashok_JIO"; // your network SSID (name)
char pass[] = "Tanvi02016"; // your network password (use for WPA, or use as key for WEP)
char server[] = "api-dev.tantiv4.com"; // Name of the server we want to connect to
unsigned char test_ca_cert[] = \
"-----BEGIN CERTIFICATE-----\n" \
"MIIDSjCCAjKgAwIBAgIQRK+wqNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/\n" \
"-----END CERTIFICATE-----\n"

Done compiling.
```

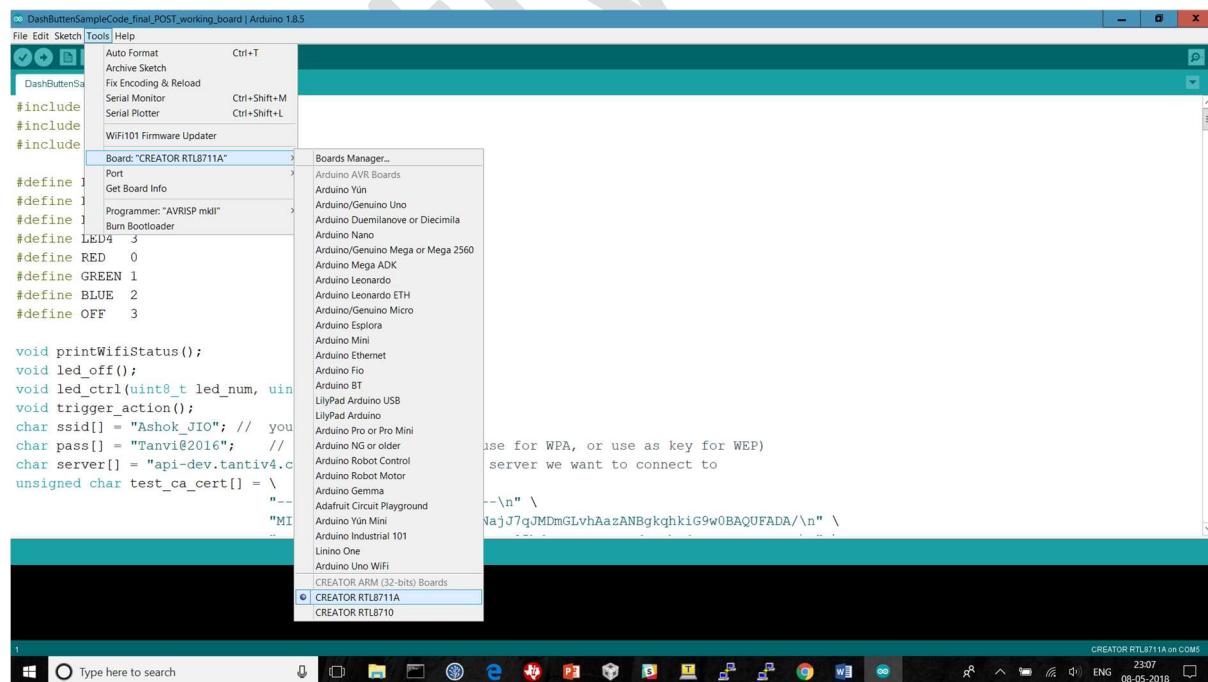


Step 3) And paste the following URL into "Additional Boards Manager URLs" field:

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":

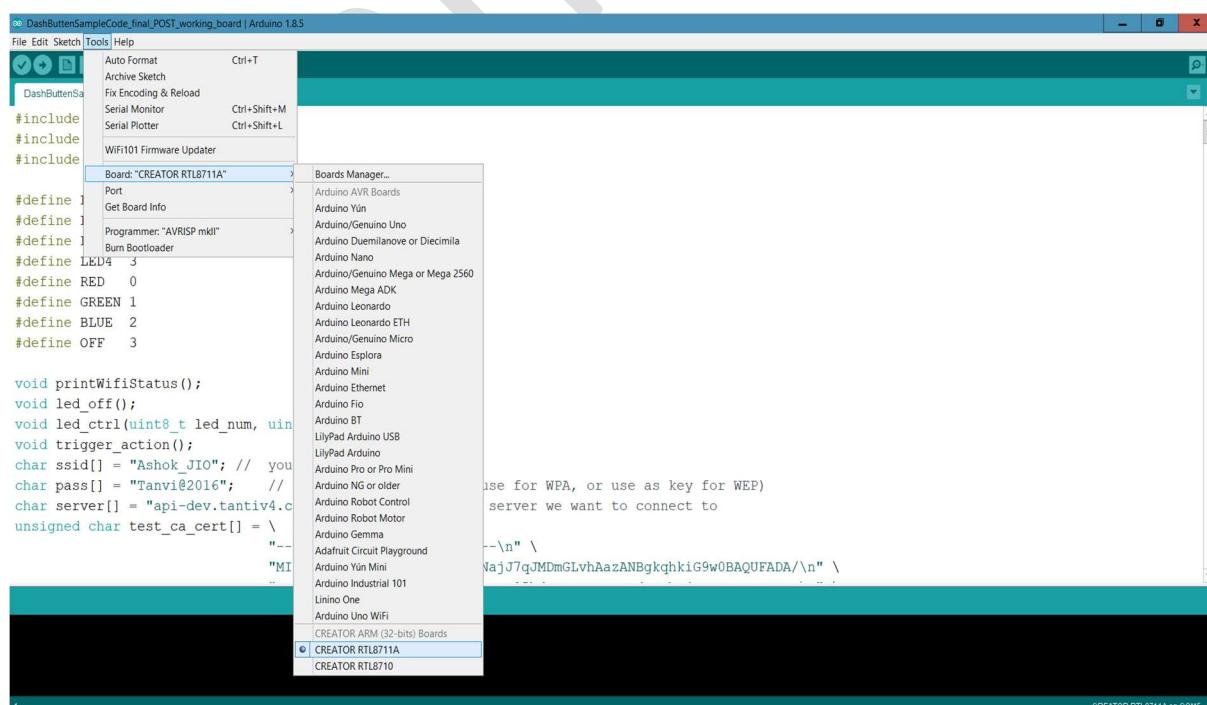




Step 5) The "Boards Manager" requires 10 seconds to refresh all hardware files (if the network is in bad condition, it may take longer). Every time the new hardware is connected, we need to reopen the Board Manager. So, we close the Boards Manager, and then open it again. Find "RAK CREATOR Boards" in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



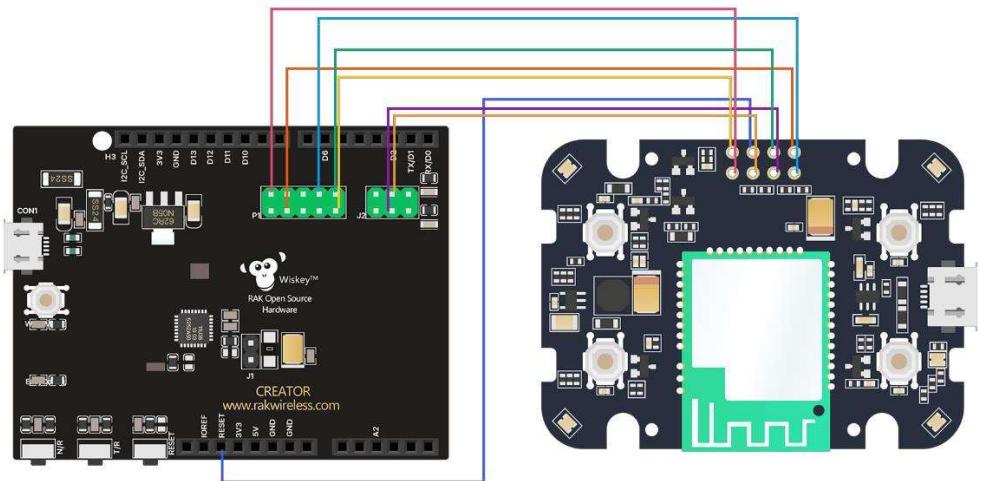
Step 6) Finally, we select CREATOR pro as current connected board in "tools" -> "Board" -> "CREATOR RTL8711A": (NOTE: RTL8195 and RTL8711 are compatible)



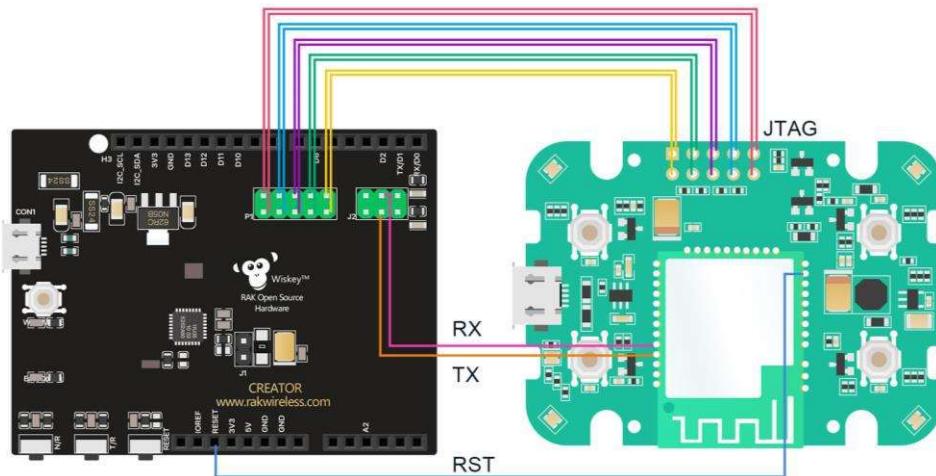


4.3.4 The Hardware JTAG connection see below

DashButton_V12 version:



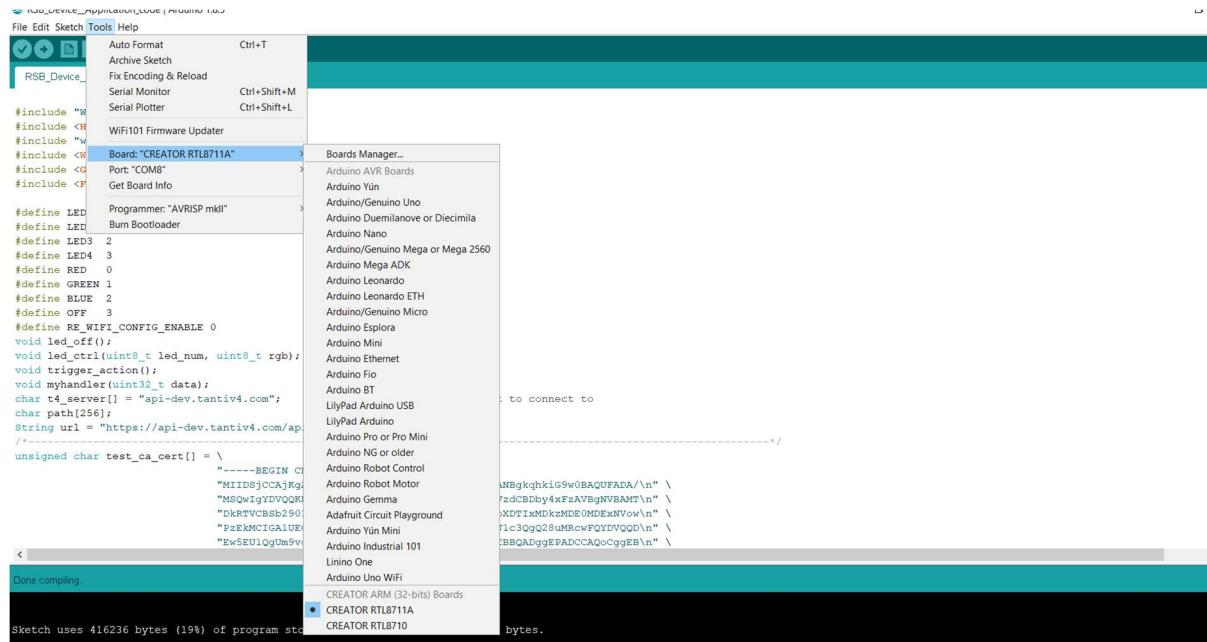
DashButton_V1.1 version:



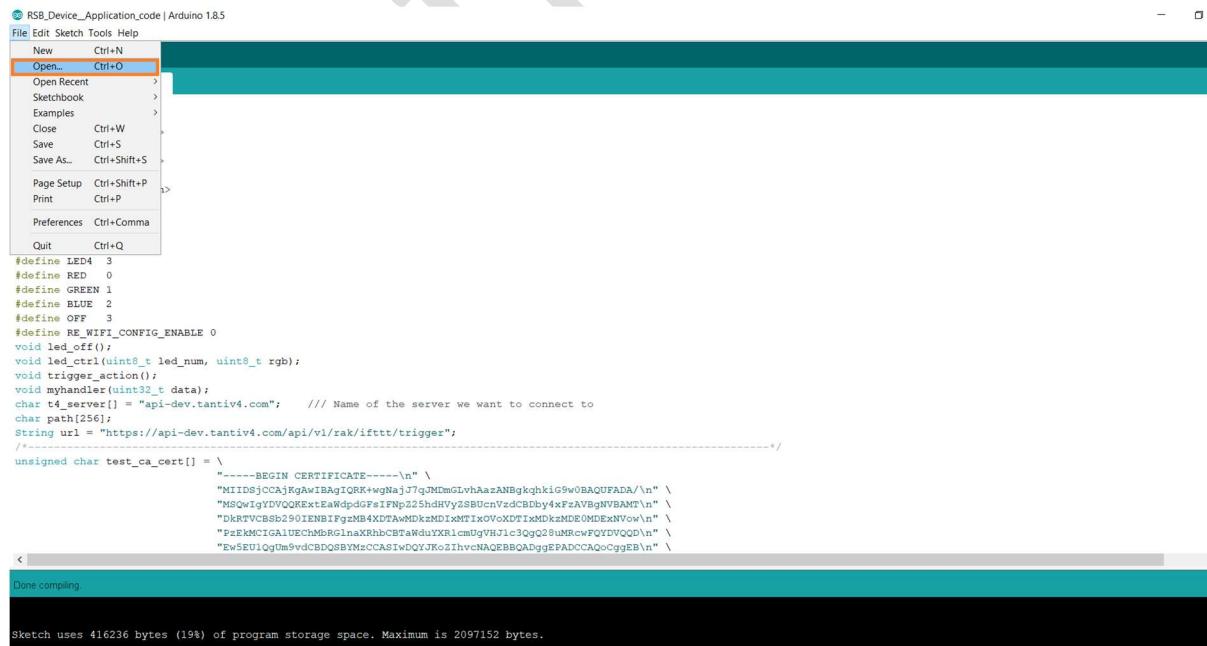


4.3.5 Run the RDB Device Application code example

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.





 RDB Device Application code | Arduino 1.8.5

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,

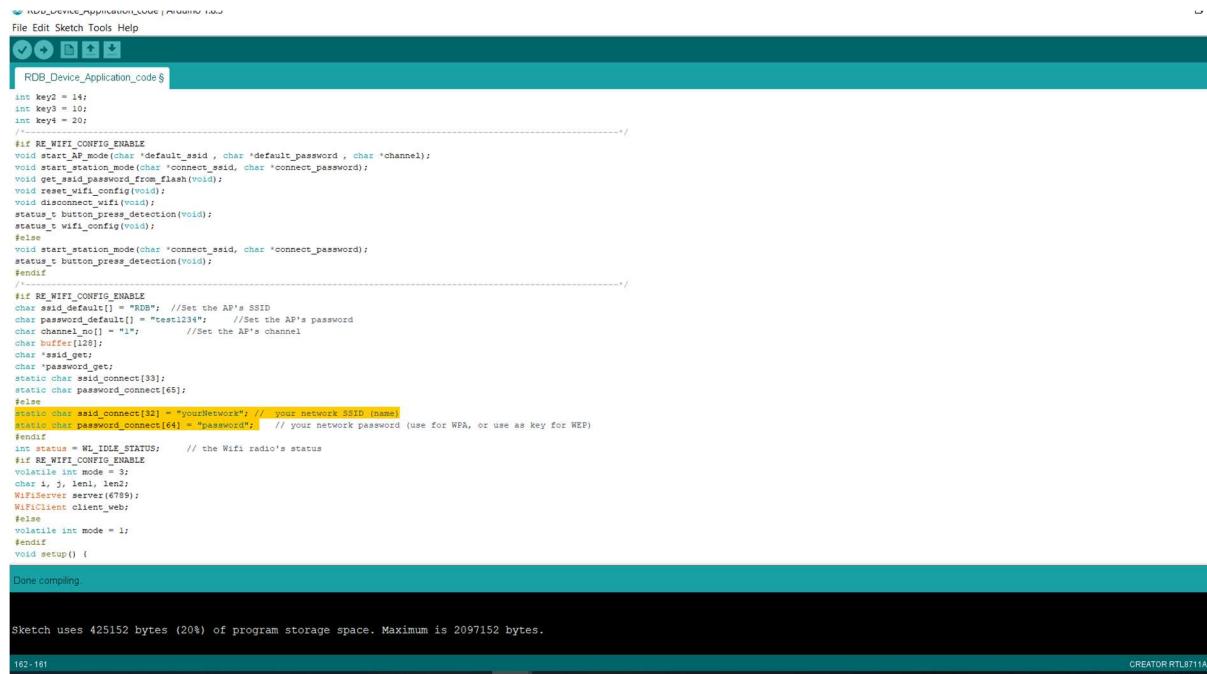
- Use Predefined WiFi network SSID and PASSWORD
- Use WiFi provisioning over HTTP

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 'R0B_Device_Application_code.ino' open. The code is written in C++ and defines WiFi configuration parameters. A specific line of code is highlighted in yellow, indicating where to modify the SSID and password:

```
char ssid_connect[32] = "yourNetwork"; // your network SSID (name)
char password_connect[64] = "password"; // your network password (use for WPA, or use as key for WEP)
```

Below the code, the status bar indicates 'Sketch uses 425152 bytes (20%) of program storage space. Maximum is 2097152 bytes.'

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.

RDB_Device_Application_code | Arduino 1.8.5

File Edit Sketch Tools Help

RDB_Device_Application_code

```
#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" \
"MIIDsjCCAJBgAwIBAgTORKw+wNaJ7gJM0mGLvhazANBgkqhkg9w0BAQUFADA/\n" \
"MSQWiqgDVQExTEawdgdFeIFnwZ25hdHVyZSBUcnVsdCDBdyxRfzAxBgNTBAMT\n" \
"DRKTVBSe290IEBRIPgzbM84X0TAwMDkEMDlMXMTIxOVXNDTlXMDXzMDNEDNEvNvow\n" \
"PEzKMCiGA1UEChMBRGlnaXhhbCBTAwDyXK1cmUgVHJl3QgQ28uRcwFQ2DyQODn\n" \
"EwEzUQ0Q0m9vdCDBQSByMzCCAS1wDQYJKoZIhvncNAQEBBQAQdgEPADOCRAQoCggEBn\n" \
"/-----END CERTIFICATE-----\n"

unsigned char test_ca_cert[] = \
```

Done uploading.

copy "C:\Users\Shubha Sinha\AppData\Local\Arduino15\packages\RAR\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

NETGEAR57 Secured

Ambigal5G Secured

Network & Internet settings

Change settings, such as making a connection metered

Wi-Fi Airplane mode hotspot

RDB_Device_Application_code | Arduino 1.8.5

File Edit Sketch Tools Help

RDB_Device_Application_code

```
#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" \
"MIIDsjCCAJBgAwIBAgTORKw+wNaJ7gJM0mGLvhazANBgkqhkg9w0BAQUFADA/\n" \
"MSQWiqgDVQExTEawdgdFeIFnwZ25hdHVyZSBUcnVsdCDBdyxRfzAxBgNTBAMT\n" \
"DRKTVBSe290IEBRIPgzbM84X0TAwMDkEMDlMXMTIxOVXNDTlXMDXzMDNEDNEvNvow\n" \
"PEzKMCiGA1UEChMBRGlnaXhhbCBTAwDyXK1cmUgVHJl3QgQ28uRcwFQ2DyQODn\n" \
"EwEzUQ0Q0m9vdCDBQSByMzCCAS1wDQYJKoZIhvncNAQEBBQAQdgEPADOCRAQoCggEBn\n" \
"/-----END CERTIFICATE-----\n"

unsigned char test_ca_cert[] = \
```

Done uploading.

copy "C:\Users\Shubha Sinha\AppData\Local\Arduino15\packages\RAR\tools\ameba_tools\1.0.9\ram_all.bin" F:
1 file(s) copied.
upload finish
24

RDB

No Internet, secured

Properties

Disconnect

TANTIV4-1 Secured

TANTIV4-2 Secured

Ambigal5G Secured

NETGEAR57 Secured

Calibsoft

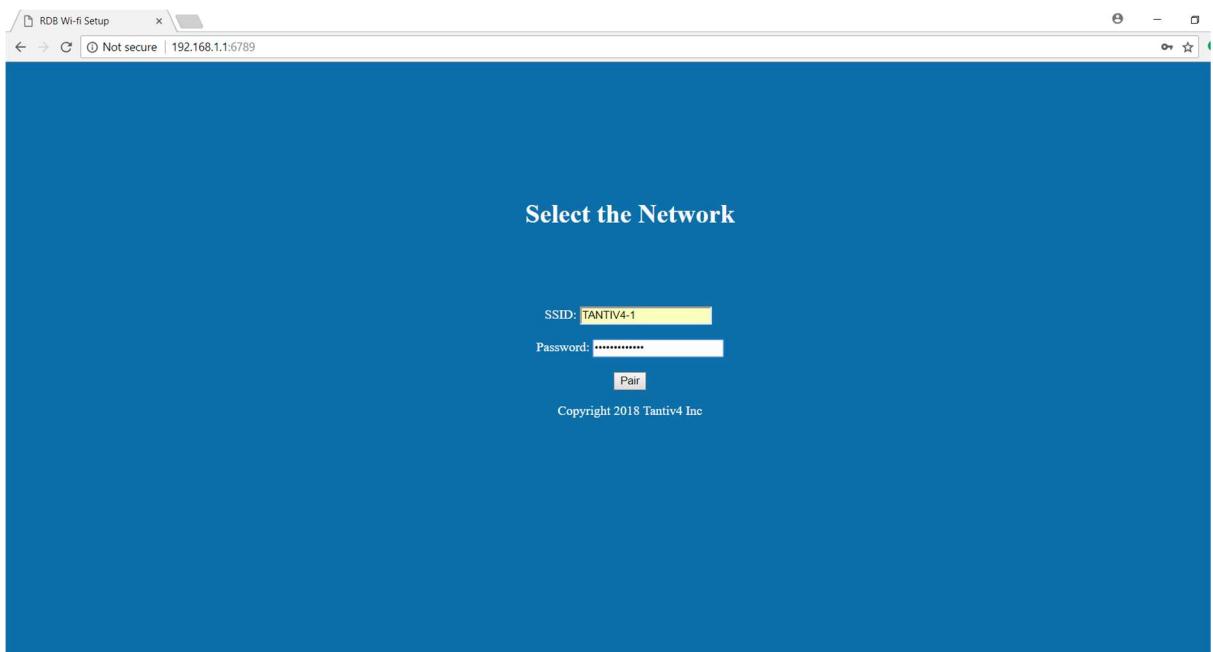
Network & Internet settings

Change settings, such as making a connection metered

Wi-Fi Airplane mode hotspot



- 2) After connection to WiFi network is established, open any web browser and type "**198.162.1.1:6789**" 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
RDB_1
  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 (client_web.connected()) {
  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\nExpires: Sun, 31 Dec 2028 23:59:59 GMT\r\nPragma: no-cache\r\n\r\n";
      client_web.println("<!DOCTYPE html>\r\n");
      client_web.println("<html>\r\n<head>\r\n");
      client_web.println("<link rel='icon' href='data:base64,=\'>\r\n");
      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>WIFI Settings</title>\r\n");
      client_web.println("<body style='background-color:#00ffaa; color:#fff; text-align: center;padding: 10px;\r\n");
      client_web.println("<h1>Select the Network</h1>\r\n");
      client_web.println("<h2>\r\n");
      client_web.println("<form class='formVal' action='http://192.168.1.1:16789' method='POST'\r\n");
      // client.println("<!-- No CORS available --&gt;"); //||\r\n");
      // client.println("&lt;select name='ssid' id='ssid'\r\n");
      client_web.println("&lt;/select&gt;\r\n");
      client_web.println("&lt;label&gt;SSID:&lt;/label&gt;\r\n");
      client_web.println("&lt;input type='text' name='ssid' id='ssid'\r\n");
      client_web.println("&lt;div&gt;\r\n");
      client_web.println("&lt;label&gt;Password:&lt;/label&gt;\r\n");
      client_web.println("&lt;input type='password' name='password' id='password'\r\n");
      client_web.println("&lt;/div&gt;\r\n");
      client_web.println("&lt;/form&gt;\r\n");
      client_web.println("&lt;/body&gt;\r\n");
      client_web.println("&lt;/html&gt;\r\n");
    }
  }
}</pre>
```

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
RDB_1
  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 (client_web.connected()) {
  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\nExpires: Sun, 31 Dec 2028 23:59:59 GMT\r\nPragma: no-cache\r\n\r\n";
      client_web.println("<!DOCTYPE html>\r\n");
      client_web.println("<html>\r\n<head>\r\n");
      client_web.println("<link rel='icon' href='data:base64,=\'>\r\n");
      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>WIFI Settings</title>\r\n");
      client_web.println("<body style='background-color:#00ffaa; color:#fff; text-align: center;padding: 10px;\r\n");
      client_web.println("<h1>Select the Network</h1>\r\n");
      client_web.println("<h2>\r\n");
      client_web.println("<form class='formVal' action='http://192.168.1.1:16789' method='POST'\r\n");
      // client.println("<!-- No CORS available --&gt;"); //||\r\n");
      // client.println("&lt;select name='ssid' id='ssid'\r\n");
      client_web.println("&lt;/select&gt;\r\n");
      client_web.println("&lt;label&gt;SSID:&lt;/label&gt;\r\n");
      client_web.println("&lt;input type='text' name='ssid' id='ssid'\r\n");
      client_web.println("&lt;div&gt;\r\n");
      client_web.println("&lt;label&gt;Password:&lt;/label&gt;\r\n");
      client_web.println("&lt;input type='password' name='password' id='password'\r\n");
      client_web.println("&lt;/div&gt;\r\n");
      client_web.println("&lt;/form&gt;\r\n");
      client_web.println("&lt;/body&gt;\r\n");
      client_web.println("&lt;/html&gt;\r\n");
    }
  }
}</pre>
```



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.

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

```
@@ COM6

=====
Check boot type form eFuse
SPIF Initial
Image1 length: 0x4880, Image Addr: 0x10000bc8
Image1 Validate OK, Going jump to Image1
BOOT from Flash: YES
SPIF calibration
Find the available window
Baud13: auto_length10; Delay start:0; Delay end:63
[Baud13]Spin玑NvKcalStore: The flash memory[0x90900 = 0x0] is not able to be write, Erase it first!===== Enter Image 1 =====
SPIF calibration
Find the available window
Baud12: auto_length12; Delay start:0; Delay end:63
[Baud12]Spin玑NvKcalStore: The flash memory[0x90b00 = 0x0] is not able to be write, Erase it first!!
SDR Controller Init
Test 0: No match addr 0x1e7010 == 0x5 != 0x8
Test 0: No match addr 0x1e7010 == 0x5 != 0x8
OTA addr 0x0 INVALIDO

load NEW fw 0
Flash: Target Address 0x0000, Len 213000, Load to SRAM 0x10006000
Image1 length: 0x4880, Image1 Addr: 0x30000000
Img2 Sign: RTWHin, Img2Start @ 0x10006049
===== Enter Image 2 =====
interface 0 is initialized
interface 1 is initialized

Initializing WIFI ...
WIFI initialized
Attempting to start AP with SSID: RDB

LoRa DEBUG: dhcp stop...
Deinitializing WIFI ...
WIFI deinitialized
Initializing WIFI ...
WIFI initialized

Starting AP ...
RDB started
IP Address: 192.168.1.1
NetMask: 255.255.255.0
Gateway: 192.168.1.1

SSID: RDB
BSSID: 60:C5:58:87:FF:FA
Encryption Type:i4

SSID: RDB
IP Address: 192.168.1.1
Signal strength (RSSI):0 dBm
To see this page in action, open a browser to http://192.168.1.1
```



```

COM6

DPOZI 4.0000 MHz
SPI calibration
Find the available window
Baud:3; auto_length:0; Delay start:0; Delay end:63
[SPI Err]SpicNVMCalStore: 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:21; Delay start:0; Delay end:63
[SPI Err]SpicNVMCalStore: The flash memory(@0x9080 = 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 0x0 INVALID

load NEW fw 0
Flash Image2:Addr 0xb000, Len 205700, Load to SRAM 0x10006000
Image3 length: 0x28638, Image3 Addr: 0x30000000
Img2 Sign: RTKWin, Infastart @ 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
  

```

```

COM6
|  

Verify requested for (Depth 0):
ceThis certificate has no flags
Connected to server
Serial.printlnbtn_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%3AgKEJFNUkeORGsNkguzOLYbdwjnUm4Kk.BNJoAh80aYgM%2FVfJ5v845RBLrirWwA145i2n1Mb0x4M; 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
Connected to server
Serial.printlnbtn_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%3AJE2PcbxpAdCerhT1CV0r2zs0AasoUL8.M6Gbkc1e9Vw%2FHRC3Yvyxp%2ByNhSMzhCzu989ZY6GO7Fc; 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
  

```

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.



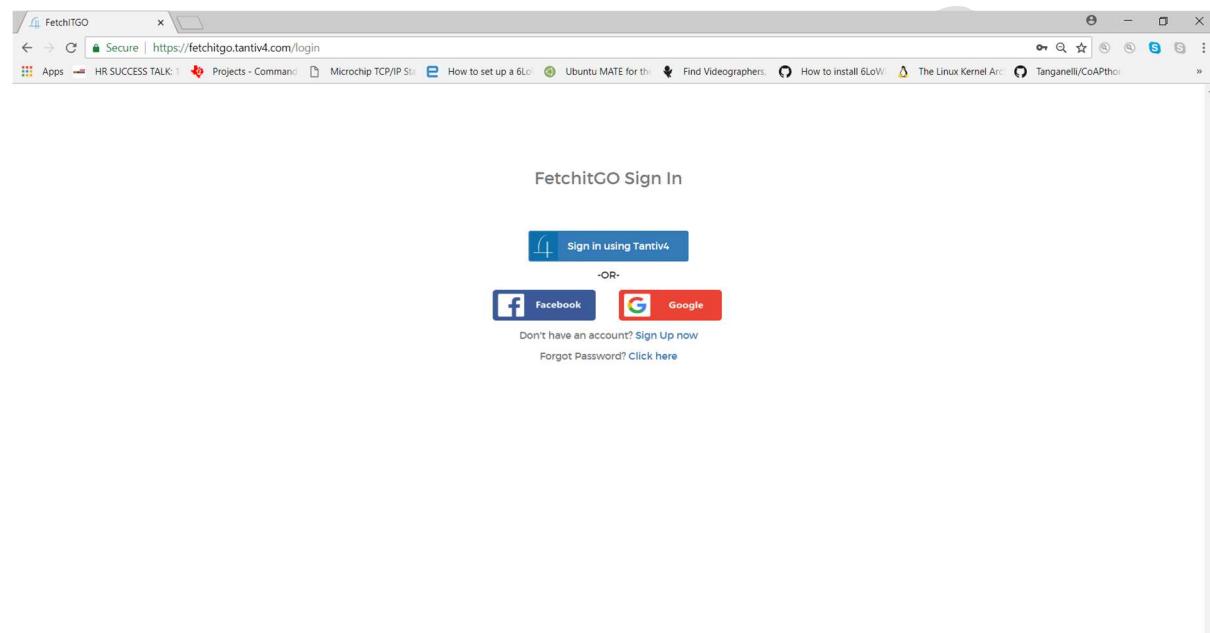
WiFi reconfiguration:

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

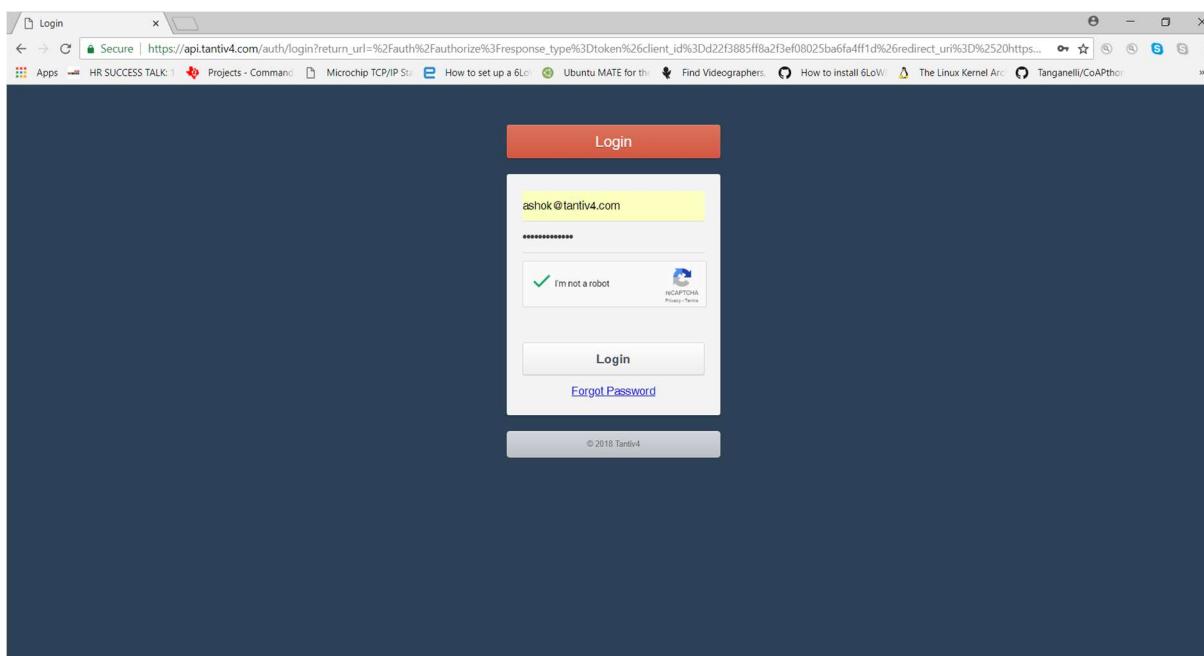
Step 1: Signup



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



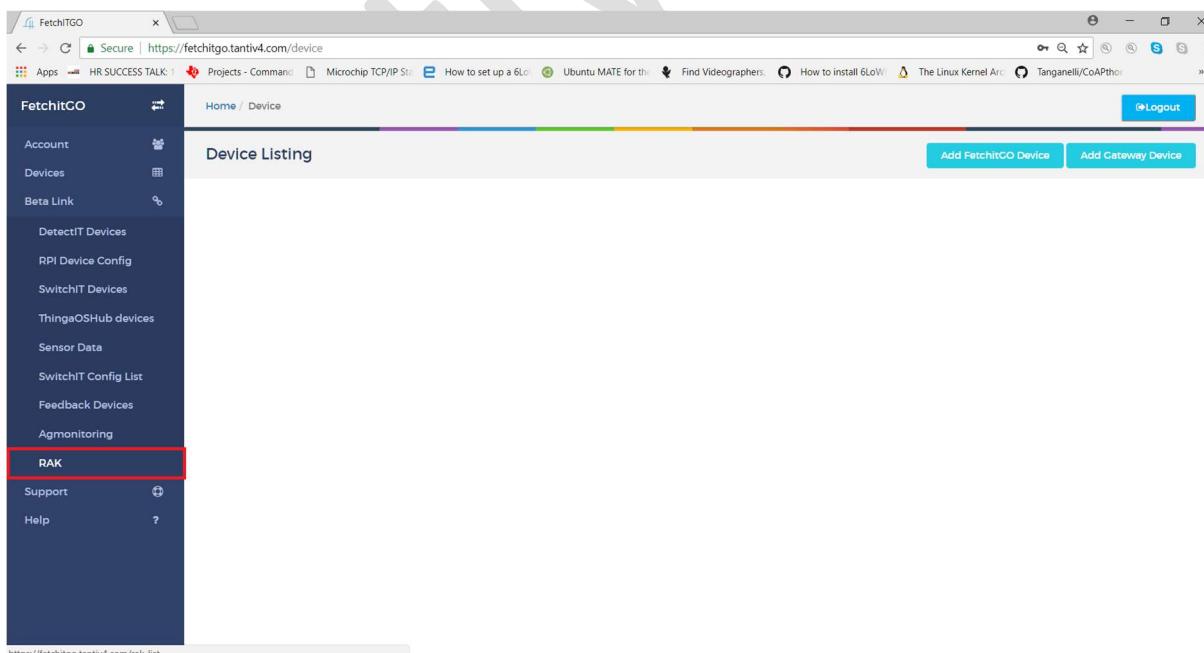
Step 2: SignIn



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**.





-> 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.

A screenshot of a web browser showing the 'Device QR Scan' page of the FetchitGO application. The URL is https://fetchitgo.tantiv4.com/rak/scan/qr-code. The page has a dark blue sidebar on the left with options: Account, Devices, Beta Link, Support, and Help. The main content area has a header 'Device QR Scan'. It contains two sections: 'Scan the QR-Code of Device' (with a large black placeholder box) and 'Enter Serial Number (OR) Customer ID and Device Friendly Name' (with fields for 'Serial Number/Customer ID' containing '54522143484E022518000' and 'Device Friendly Name' containing 'Rak Device'). A 'Save' button is at the bottom right. The top of the browser shows various tabs and icons.

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

A screenshot of a web browser showing the 'RAK-Device-List' page of the FetchitGO application. The URL is https://fetchitgo.tantiv4.com/rak-list. The sidebar is identical to the previous screenshot. The main content area has a header 'Device Listing'. It shows a single device entry: 'Name : Rak Device' and 'Device ID : 54522143484E022518000006'. Below the device details are two buttons: 'IFTTT' (in blue) and 'Delete' (in dark blue). A 'Logout' button is in the top right corner. The top of the browser shows various tabs and icons.

-> 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' page for a device named 'rak-1'. On the left is a sidebar with 'FetchitGO' branding and links for Account, Devices, Beta Link, Support, and Help. The main area has two sections of four buttons each. The first three buttons in both sections have a standard grey background and are labeled 'Select a recipe'. The fourth button in the second section has a white background with a blue IFTTT logo and is labeled 'Set IFTTT Action Notes'. A 'Save' button is located at the bottom right of the grid.

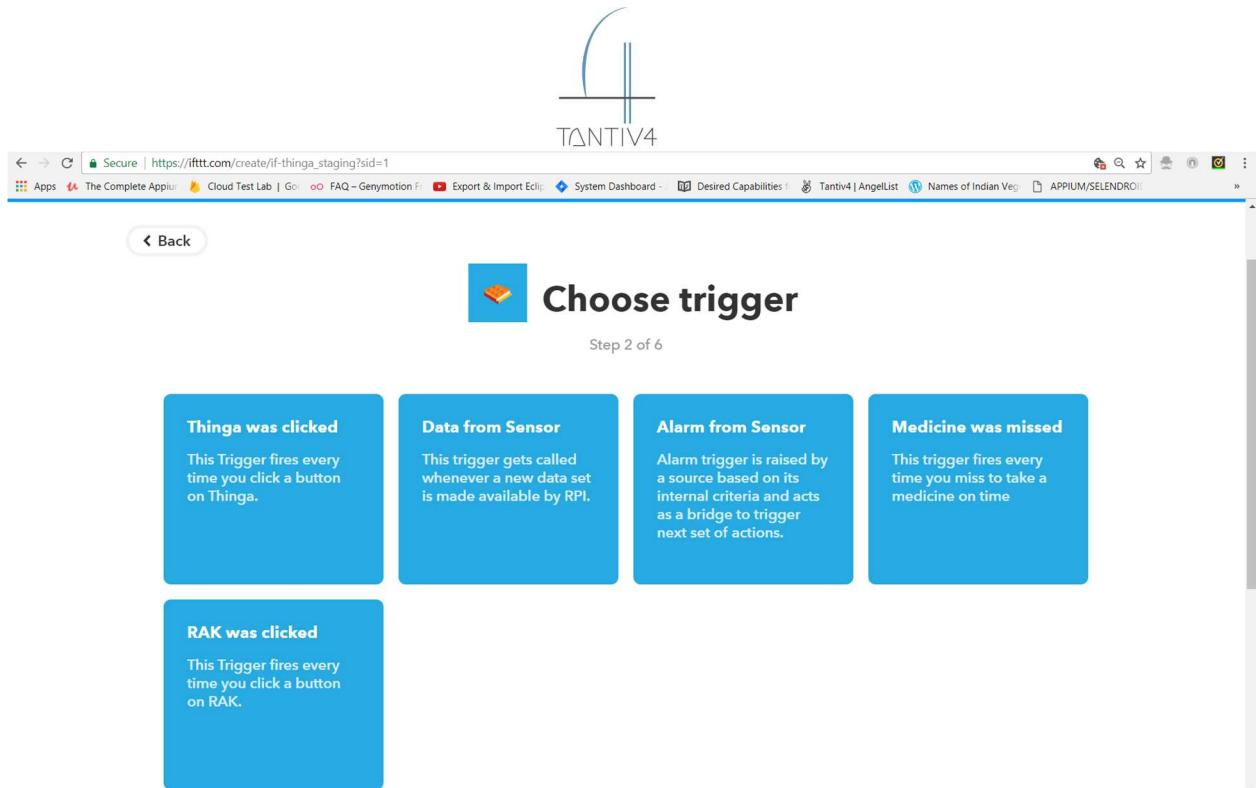
->Click on **button** it will re-direct to **IFTTT** page.

->Create **New Applet** in **IFTTT**

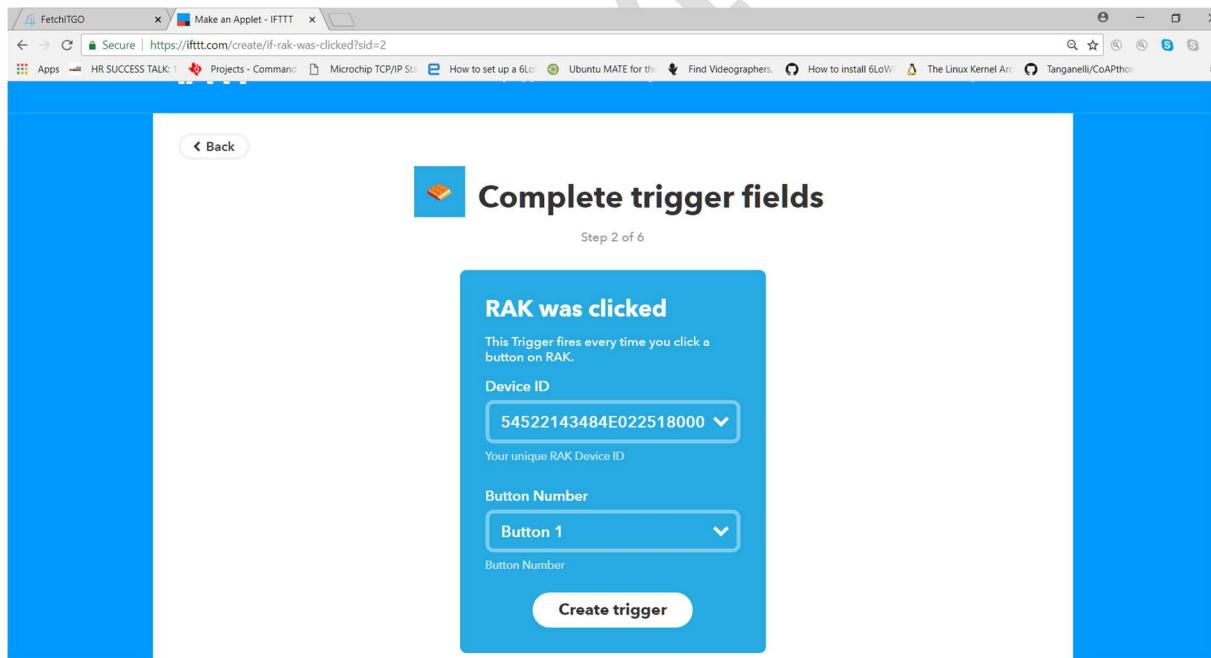
The screenshot shows the IFTTT homepage with a prominent 'New Applet' button in the center. At the top, there's a navigation bar with 'IFTTT', 'Discover', 'Search', 'My Apps', and 'Activity'. A user profile 'vidya6' is visible on the right. Below the main button, there's a green call-to-action box with the text 'Want to do more? Partner with IFTTT, work with everyone. Build and publish your service and Applets on our platform.' and a 'Partner with IFTTT' button.

if +this then that

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



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



->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.

6 References



Tantiv4 Inc.