



ZÁKLADY BEZDRÔTOVÝCH SIETÍ

KOMUNIKAČNÉ TECHNOLOGIE PRE SYSTÉMY IOT (2/4)

ING. LUKÁŠ FORMANEK, PHD.

Vytvorené v rámci projektu KEGA 026TUKE-4/2021

ESP32 DEVELOPMENT BOARD

ESP32 DEV KIT V1 PINOUT

PWR LED **USER LED (GPIO2)**

EN	36	GPIO23	V_SPI_D	MOSI
Input only	39	GPIO22	V_SPI_WP	SCL
RTC GPIO00	41	GPIO1	TXD 0	CLK3
SensVP	40	GPIO3	RXD 0	CLK2
ADC1_0	42	GPIO21	VSPI_HD	SDA
GPIO36	38	GPIO19	V_SPI_Q	MISO
5	35	GPIO18	V_SPI_CLK	SCK
Input only	34	GPIO5	V_SPI_CS0	SS
RTC GPIO03	27	GPIO17	TXD 2	
SensVN	25	GPIO16	RXD 2	
ADC1_3	24	GPIO4	ADC2_0	HSPI_HD
GPIO39	22	GPIO2	ADC2_2	HSPI_WP0
8	21	GPIO15	ADC2_3	HSPI_CS0
Input only	20			
RTC GPIO04	17			
ADC1_6	16			
GPIO34	15			
10	14			
Input only	13			
RTC GPIO05	12			
ADC1_7	11			
GPIO35	10			
11	9			
RTC GPIO09	8			
Xtal32P	7			
Touch9	6			
ADC1_4	5			
GPIO32	4			
12	3			
RTC GPIO08	2			
Xtal32N	1			
Touch8	0			
ADC1_5				
GPIO33				
13				
DAC 1				
RTC GPIO06				
ADC2_8				
GPIO25				
14				
DAC 2				
RTC GPIO07				
ADC2_9				
GPIO26				
15				
RTC GPIO17				
Touch7				
ADC2_7				
GPIO27				
16				
RTC GPIO16				
Touch6				
HSPI_CLK				
ADC2_6				
GPIO14				
17				
RTC GPIO15				
Touch5				
HSPI_Q				
ADC2_5				
GPIO12				
18				
RTC GPIO14				
Touch4				
HSPI_ID				
ADC2_4				
GPIO13				
20				
GND				
VIN				
3.3v				

UART

RST **BOOT (GPIO0)**

www.mischianti.org CC BY-NC-ND

TLAČIDLO (BOOT – GPIO0)

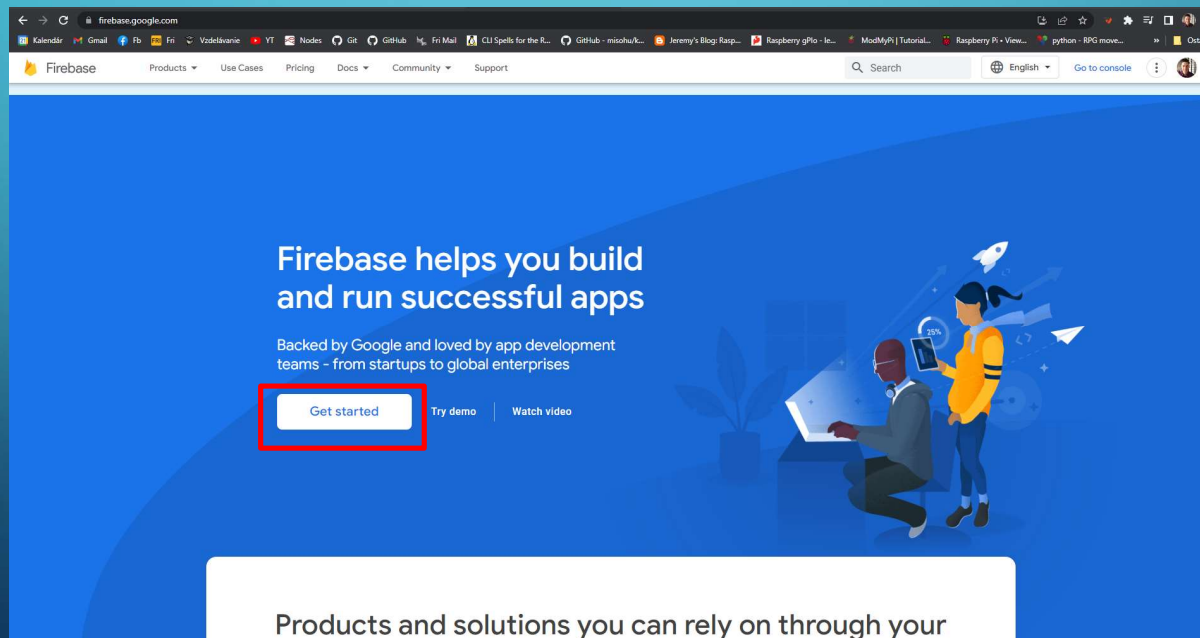
- <https://docs.micropython.org/en/latest/esp32/quickref.html#pins-and-gpio>
- Špeciálna funkcia (firmware update / používateľské tlačidlo)

TLAČIDLO (BOOT – GPIO0) - PRERUŠENIE

```
button.py ×
1 from machine import Pin
2
3 class Button:
4     button_pin_default = 0 # Boot pin
5
6     def __init__(self, pin_num = button_pin_default):
7         self.pin = Pin(pin_num, Pin.IN, Pin.PULL_UP)
8         self.pin.irq(trigger=Pin.IRQ_FALLING, handler=self.interrupt_handler)
9         self.is_button_pressed = False
10
11     def interrupt_handler(self, pin):
12         self.is_button_pressed = True
13         # print("Btn is pressed")
```

GOOGLE FIREBASE

- <https://firebase.google.com/>
- Realtime NoSQL databáza.



VYTVORENIE PROJEKTU

The screenshot shows the Firebase console interface. At the top left is the Firebase logo. In the top right corner, there are links for 'Go to docs', a notification bell, and a user profile icon. The main heading is 'Your Firebase projects'. Below this heading, there are three white cards. The first card on the left is highlighted with a red border and contains a blue plus sign and the text 'Add project'. The second card is titled 'test' with the ID 'test-898e2'. The third card is titled 'Test' with the ID 'test-48dff'. Below the project cards, there are two promotional boxes. The left one is titled 'Explore a demo project' and features an 'iOS+' icon. The right one is titled 'Firebase projects are containers for your apps' and includes a 'Learn more' link. At the bottom left, there is a language dropdown menu set to 'English (United Kingdom)'. At the bottom right, there are links for 'Support', 'Terms', and 'Privacy policy'.

VYTVORENIE PROJEKTU

× Create a project(Step 1 of 3)

Let's start with a name for
your project[®]

Project name

IoT-ESP32

iot-esp32-5c3f0

I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft or profession.

Continue



VYTVORENIE PROJEKTU

× Create a project(Step 2 of 3)

Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, in-app messaging, Remote Config, A/B Testing and Cloud Functions.

Google Analytics enables:

-  A/B testing ⓘ
-  User segmentation and targeting across Firebase products ⓘ
-  Crash-free users ⓘ
-  Event-based Cloud Functions triggers ⓘ
-  Free unlimited reporting ⓘ

Enable Google Analytics for this project
Recommended

[Previous](#)

[Continue](#)




VYTVORENIE PROJEKTU

× Create a project(Step 3 of 3)

Configure Google Analytics

Choose or create a Google Analytics account ⓘ

 Default Account for Firebase ▾

Automatically create a new property in this account ✎

Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

[Previous](#)

Create project



VYTVORENIE DATABÁZY

The screenshot displays the Firebase console interface for a project named 'IoT-ESP32'. The left-hand navigation menu is visible, with the 'Realtime Database' option highlighted in a red box. The main content area features the heading 'Realtime Database' and the subtext 'Store and sync data in real time'. A 'Create Database' button is prominently displayed and also highlighted with a red box. To the right of the text, there is an illustration of three server racks connected by lines. Below the main heading, a white banner contains the text 'Is Realtime Database right for you?' followed by a link to 'Compare Databases'. Underneath, a 'Learn more' section includes two links: 'How do I get started? View the docs' and 'How much will Realtime Database cost? View pricing'. A video player is embedded in the 'Learn more' section, showing a video titled 'Introducing Firebase Realtime Database' with a play button overlay. The video player includes standard controls like a clock icon and a share icon. At the bottom of the video player, there is a small 'Prehľad na YouTube' (Preview on YouTube) button and the 'firebase' logo.

VYTVORENIE DATABÁZY

The screenshot displays the Firebase console interface for a project named 'IoT-ESP32'. The main heading is 'Realtime Database' with the subtitle 'Store and sync data in real time'. A 'Create Database' button is visible. A modal dialog titled 'Set up database' is open, showing two steps: '1 Database options' and '2 Security rules'. The 'Database options' step is active, displaying the text 'Your location setting is where your Realtime Database data will be stored.' Below this, there is a dropdown menu for 'Realtime Database location' which is currently set to 'United States (us-central1)'. At the bottom right of the dialog, there are 'Cancel' and 'Next' buttons, with the 'Next' button highlighted by a red box. The background of the console shows a sidebar with navigation options like 'Project Overview', 'Build', 'Release and monitor', 'Analytics', 'Engage', and 'Extensions', along with a main content area with informational cards.

VYTVORENIE DATABÁZY

Set up database

1 Database options — 2 Security rules

Once you have defined your data structure, you will have to write rules to secure your data.
[Learn more](#)

Start in locked mode
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

Start in test mode
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{  
  "rules": {  
    ".read": false,  
    ".write": false  
  }  
}
```

All third party reads and writes will be denied

Cancel **Enable**

NASTAVENIE DATABÁZY

Firestore Realtime Database

ioT-ESP32

Go to docs

Realtime Database

Data **Rules** Backups Usage

Edit rules Monitor rules

Rules Playground

⚠ Your security rules are defined as public, so anyone can steal, modify or delete data in your database [Learn more](#) [Dismiss](#)

```
1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }
```

Database location: United States (us-central1)

URL LINK A ZOBRAZENIE DÁT

The screenshot displays the Firebase Realtime Database interface. On the left is a navigation sidebar with categories like 'Build', 'Release and monitor', 'Analytics', and 'Engage'. The main content area shows the 'Realtime Database' page for a project named 'IoT-ESP32'. A red box highlights the URL in the browser's address bar: `https://iot-esp32-5c3f0-default-rtdb.firebaseio.com`. A red arrow labeled 'URL' points to this box. Below the address bar, a red box highlights a data snapshot: `test: "1234"`. A red arrow labeled 'Dáta' points to this data. A security warning banner is visible between the URL and the data, stating: 'security rules are defined as public, so anyone can steal, modify or delete data in your database'. At the bottom, it indicates the database location is 'United States (us-central1)'.

MICROPYTHON-FIREBASE KNIŽNICA

- <https://github.com/ckoever/micropython-firebase-realtime-database>
- Příkazy:
 - get,
 - getfile,
 - put,
 - patch,
 - addto,
 - delete.

POSTUP

1. Nahratie knižnice `ufirebase.py` do ESP32.
2. Pripojenie ESP32 k AP s konektivitou na internet (použite napr. `[ssid/pswd]: wifri/-` alebo `KIS-Guest/hatatitla`).
3. Import knižnice : `import ufirebase as firebase.`
4. Nastavenie URL : `firebase.setURL("URL").`
5. Zaslanie (upload) dát: `firebase.put("test", "1 234").`
6. Stiahnutie (download) dát: `firebase.get("test", "testVariable")`
`print(firebase.testVariable).`

ÚLOHY

- Vyskúšajte funkčnosť tlačidla boot (aj s prerušením).
- Vytvorte si vlastnú Firebase databázu.
- Otestujte všetky metódy knižnice `ufirebase.py`.

ZADANIE IOT 1 (10B)

- Vytvorte program, ktorý automaticky po reštarte zariadenia (ESP32) zobrazí dostupné WiFi siete (SSID) + RSSI. Ak je sieť nezabezpečená, zobrazí pri danej sieti „open“. Následne vyzve používateľa, aby zadal SSID + heslo siete, ku ktorej sa chce pripojiť. Po pripojení zobrazí IP adresu, masku siete, default gateway. Využite objektový prístup a navrhnete vlastnú/é triedu/y.
 - Použite triedu: `network.WLAN`.

Ukážka výstupu:

```
NETGEAR83          -77
Eduroam            -80
Wifri              -81      open
...
Enter the SSID:    NETGEAR83
Enter the PSWD:    123pswd321
Connected!
IP address:        192.168.1.23
mask:              255.255.255.0
Default gateway:   192.168.1.1
```

- Po stlačení tlačidla (boot) sa do vašej Firebase databázy pridajú nasledovné hodnoty (použite knižnicu esp32 a metódy : `raw_temperature()`, `hall_sensor()`):
 - Interná teplota MCU v °C.
 - Interná hodnota z hallovho senzora.

Ukážka výstupu:

Zdrojové súbory nahrajte na moodle vo formáte : Priezvisko1_Priezvisko2.zip

Termin: do 07.05.2023

The screenshot shows the Firebase Realtime Database interface. The left sidebar contains navigation options: Project Overview, Build (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release and monitor, Analytics, Engage, and Extensions. The main area displays the Realtime Database for the project 'IoT-ESP32'. A data viewer window is open, showing the following JSON structure:

```
https://iot-esp32-5c3f0-default-rtdb.firebaseio.com/  
- Measured_values  
  - N8VC02K-BfDK-8HNDwJ  
    hall: 57  
    temp: 50.55556  
  - N8VCbxz-Igr7VDfGIb9  
    hall: 63  
    temp: 50  
  - hall: 60  
    temp: 50.55556
```

At the bottom of the data viewer, it indicates 'Database location: United States (us-central1)'.

ĎAKUJEM ZA POZORNOST



Vytvorené v rámci projektu KEGA 026TUKE-4/2021