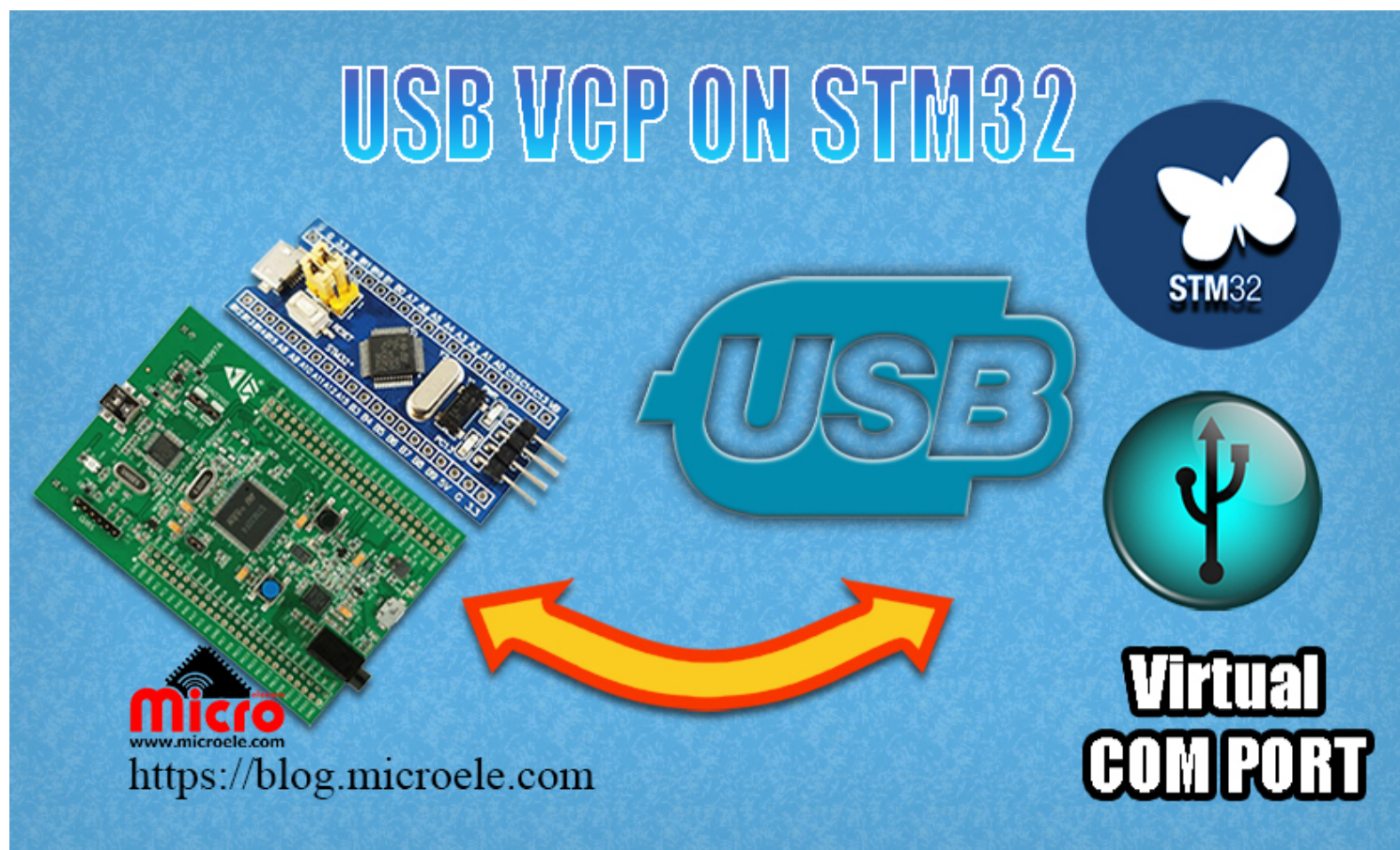




راه اندازی USB و ایجاد پورت سریال مجازی در میکروکنترلر های STM32



تاریخ انتشار ۲۴ فروردین، ۱۴۰۰ توسط آرش فتاحی

در این مطلب قصد داریم با استفاده از پورت USB در میکروکنترلرهای STM32، یک پورت سریال مجازی بر روی این میکروکنترلرها ایجاد کنیم. اگر قبلاً با بردهای آردینو کار کرده باشید، احتمالاً با تنظیمات مربوط به پورت COM در این نوع برد ها آشنایی دارید. در **بردهای آردینو** یک آیسی **مبدل USB به سریال** وجود دارد که علاوه بر پروگرام کردن میکروی روی برد، امکان برقراری ارتباط بین کامپیوتر و میکروکنترلر را از طریق پروتکل UART و درگاه USB امکان



پذیر می‌کند. سپس می‌توان با اجرای یک ترمینال سریال مانند Serial monitor و یا Serial plotter، داده‌های دریافتی را مشاهده کرده و یا داده‌های مد نظر خود را به برد ارسال کنیم.

در اکثر میکروکنترلرهای STM32، پورت USB به صورت پیش فرض وجود داشته و می‌توان با تبدیل آن به پورت سریال مجازی، اطلاعات را به صورت سریال و بدون نیاز به آرسی مبدل بین کامپیوتر و میکرو انتقال داد. همچنین شما می‌توانید [آموزش کامل نحوه اختصاص بخشی از حافظه FLASH به EEPROM در میکروکنترلر سری STM32](#) را نیز مطالعه کنید.

Device و Host، USB FS

در برد دیسکآوری F407 که در این آزمایش استفاده شده است، کانکتور USB تعبیه شده بر روی برد، به پایه‌های مربوط به USB_FS متصل شده است. USB FS مخفف USB Fast Speed می‌باشد که سرعت انتقال داده در آن به 12 مگابیت بر ثانیه می‌رسد.

Host یا میزبان، یک دستگاه الکترونیکی است که ارتباط بین خود و دیوایس slave را کنترل می‌کند. میزبان معمولاً یک رایانه شخصی یا لپ تاپ است.

دیوایس، یک دستگاه الکترونیکی است که توسط میزبان کنترل می‌شود. دیوایس می‌تواند صفحه کلید یا ماوس، PC Disk drive، Flash drive، یا تلفن هوشمند باشد.

وسایل الکترونیکی که قابلیت OTG دارند، می‌توانند هم به صورت Host و هم به صورت Device عمل کنند. به عنوان مثال یک تلفن هوشمند هنگام اتصال به لپ تاپ Device بوده و هنگامی که یک حافظه فلش به آن گوشی متصل شود، برای آن حافظه Flash تبدیل به میزبان یا Host خواهد شد.

تعریف پروژه

در این جا قصد داریم بر روی برد دیسکآوری F407، پایه‌های مربوط به LED های روی برد را از طریق درگاه USB کنترل کنیم. بعد از ارسال کاراکتر '1'، تمام LED ها روشن شده و پیام وضعیت روشن بودن آن‌ها به ترمینال جهت نمایش ارسال خواهد شد. همچنین با ارسال کاراکتر '0' تمام LED ها خاموش شده و پیام خاموش شدن LED ها بر روی ترمینال نمایش داده خواهند شد. علاوه بر این، یک شمارنده بر روی میکرو ایجاد خواهیم کرد که با هر بار فشردن



کلید User (کلید آبی)، مقدار آن افزایش یافته و بر روی ترمینالِ سریال نمایش داده شود.

وسایل مورد نیاز

برد دیسکآوری F407

یک عدد کابل USB mini

یک عدد کابل USB Micro

برنامه Serial terminal که در اینجا ما از Hercules استفاده می‌کنیم.

عزیزان توجه داشته باشند که می‌توانند از این مطلب، برای بردها و سری‌های دیگر میکروکنترلرهای STM32 نیز استفاده کنند.

مراحل انجام آزمایش

ابتدا نرم افزار STM32CUBEMX را اجرا کرده و بعد از انتخاب گزینه ACCESS TO MCU SELECTOR، میکروکنترلر STM32F407VG را انتخاب می‌کنیم.



MX New Project from a MCU/MPU

MCU/MPU Selector Board Selector Example Selector Cross Selector

MCU/MPU Filters

Part Number

Core >

Series >

Line >

Package >

Other >

Peripheral >

Features Block Diagram Docs & Resources Datasheet Buy

STM32F4 Series

STM32F407VG High-performance foundation line, Arm Cortex-M4 core with DSP and FPU, 1 Mbyte of Flash memory, 168 MHz CPU, ART Accelerator, Ethernet, FSMC

ACTIVE Active Product is in mass production

Unit Price for 10kU (US\$) : 5.623

Board: STM32F407G-DISC1

LQFP100

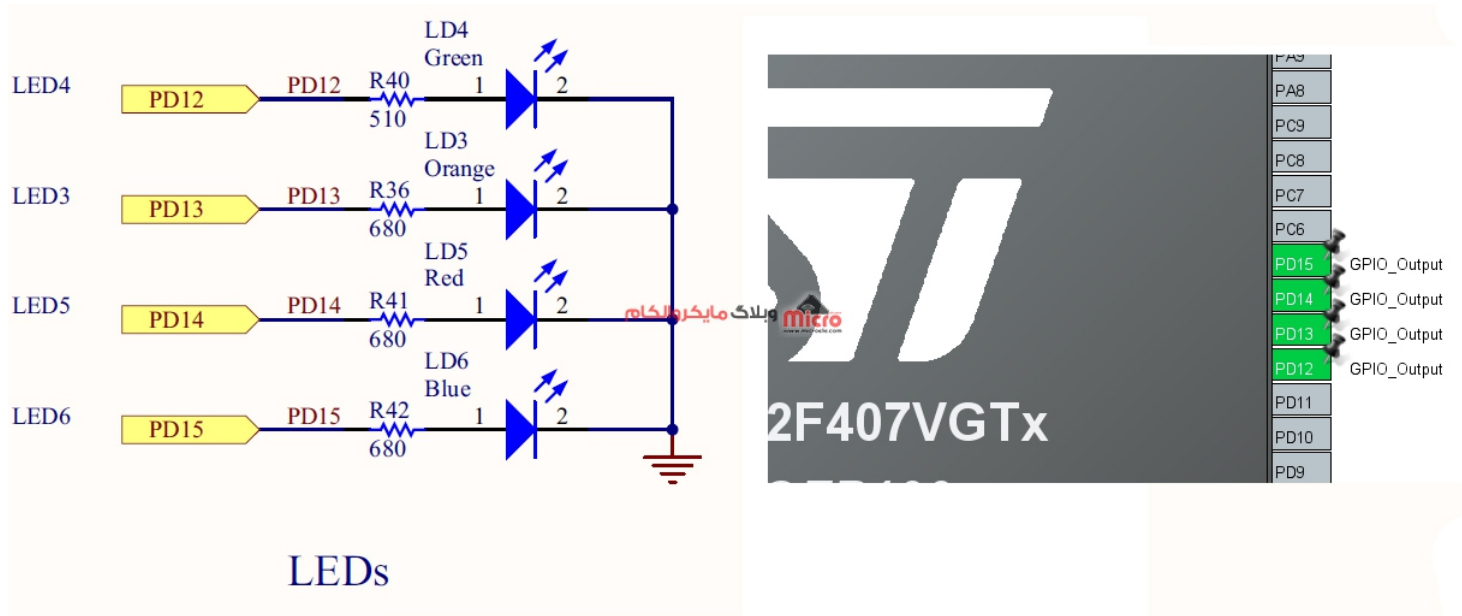
The STM32F405xx and STM32F407xx family is based on the high-performance Arm® Cortex® -M4 32-bit

MCUs/MPUs List: 1 item + Display similar items Export

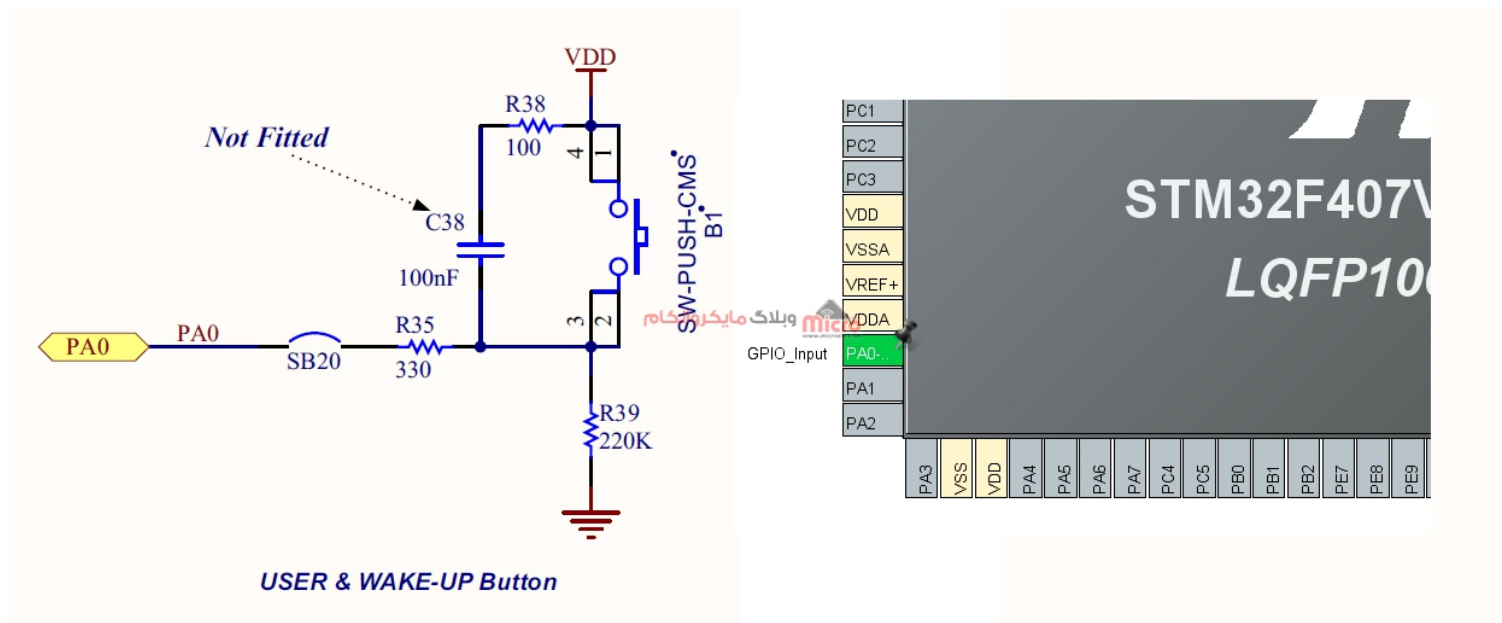
	Part No.	Referen.	Unit Pri.	Board	Package	Flash	RAM	IO	Freq.
☆	STM32F407VG	STM32... A...	5.623	STM32F407G-DISC1	LQFP100	1024 k...	192 kB...	82	168 ...

نرم افزار STM32CUBEMX

بر اساس شماتیک برد دیسکاور F407، چهار عدد LED بر روی پایه‌های PD12 تا PD15 وجود دارد که آن‌ها را خروجی و یک کلید که بر روی PA0 قرار دارد را به صورت ورودی تعریف می‌کنیم (توجه شود که این کلید به صورت سخت افزاری، Pull-Down شده است).



شماتیک LED ها در برد دیسکاواری F407



کلید در برد دیسکاواری برای راه اندازی F407

در نرم افزار Cube، از بخش Categories به قسمت Connectivity رفته و USB_OTG_FS را انتخاب می‌کنیم. از آنجایی که قرار است برد ما به کامپیوتر متصل شود، کامپیوتر، میزبان (HOST) بوده و برد دیسکاواری باید در حالت DEVICE قرار



گیرد. بنابراین Mode را بر روی Device_Only تنظیم می‌کنیم. در آخر برای راه اندازی بخش تغذیه USB در برد دیسکآوری، تیک Activate_VBUS را زده و آن را فعال می‌کنیم. با این کار PA9 به صورت اتوماتیک روی USB_OTG_FS_VBUS تنظیم می‌گردد.



MX STM32CubeMX Untitled*: STM32F407VGTx



File

Window

Help

Home

STM32F407VGTx

Untitled - Pinout & Configuration

Pinout & Configuration

Clock Configuration

Software Packs

Pinout

Search



Categories

A->Z

Connectivity

1

CAN1

CAN2

ETH

FSMC

I2C1

I2C2

I2C3

SDIO

SPI1

SPI2

SPI3

UART4

UART5

USART1

USART2

USART3

USART6

USB_OTG_FS

2

USB_OTG_HS

USB_OTG_FS Mode and Configuration

Mode

Mode Device_Only

☐ Activate_SOF

☒ Activate_VBUS

4

3

وبلاگ مایکروالکام 

Configuration

Reset Configuration

☒ User Constants

☒ NVIC Settings

☒ GPIO Settings

☒ Parameter Settings

Configure the below parameters :

Search (Ctrl+F)



Speed

Device Full Speed 12MBit/s

MCUs Selection

Output

Series	Lines	Mcu
STM32F4	STM32F407/417	STM32F407VGTx

تنظیمات بخش USB در CubeMX



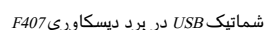
در نرم افزار Cube از قسمت Categories، به بخش Middleware رفته و USB_DEVICE را انتخاب کنید. مانند تصویر زیر در قسمت Mode منوی Class For FS IP را باز کرده و آن را بر روی Communication Device Class (Virtual Port Com) قرار دهید تا در هنگام تولید پروژه پیکربندی نرم افزاری مربوط به USB CDC به کتابخانه‌های مورد نیاز جهت کار با پورت سریال مجازی ایجاد گردند.

The screenshot shows the STM32CubeMX software interface. The 'Pinout & Configuration' tab is active. In the left sidebar, 'Middleware' is selected (1). Below it, 'USB_DEVICE' is selected (2). In the main area, under 'USB_DEVICE Mode and Configuration', the 'Class For FS IP' dropdown is set to 'Communication Device Class (Virtual Port Com)' (3). The 'Configuration' section shows 'Parameter Settings' selected. The bottom table lists the selected MCU and package.

Series	Lines	Mcu	Package	Required Peripherals
STM32F4	STM32F407/417	STM32F407VGTx	LQFP100	None

تنظیمات Middleware در نرم افزار CubeMX

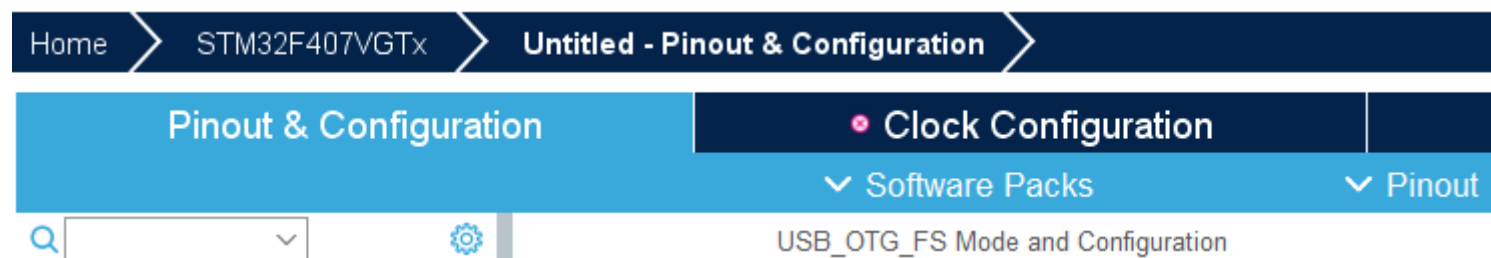
بر اساس شماتیک برد دیسکاور F407، برای فعال سازی VBUS، نیاز است که پایه PC0 صفر شود تا پایه EN در آیدی پاور سویچ روی برد فعال گردد. بنابراین مجدد به Pin Configuration رفته و PC0 را نیز به صورت خروجی قرار می‌دهیم.





تعریف پایه PC0 به عنوان خروجی

همانطور که در تصویر قبل نیز مشاهده شد، بخش Clock configuration دارای یک ضربدر قرمز رنگ می باشد. این ضربدر بعد از فعال کردن بخش USB ایجاد شده که نشان می دهد، باید تغییراتی را در بخش فرکانس اعمال کنیم.



CubeMX در Clock configuration

ابتدا از قسمت Categories، به شاخه System Core رفته و RCC را انتخاب می کنیم. در قسمت Mode و HSE، گزینه Crystal/Ceramic Resonator را انتخاب کرده تا امکان استفاده از کریستال خارجی فراهم گردد.



MX STM32CubeMX Untitled*: STM32F407VGTx



File

Window

Help

Home

STM32F407VGTx

Untitled - Pinout & Configuration

Pinout & Configuration

Clock Configuration

Software Packs

Pinout

Search



Categories

A->Z

System Core

1

DMA

GPIO

IWDG

NVIC

✓ RCC

⚠ SYS

WWDG

Analog

Timers

Connectivity

CAN1

CAN2

⚠ ETU

RCC Mode and Configuration

Mode

High Speed Clock (HSE)

Crystal/Ceramic Resonator

3

Low Speed Clock (LSE)

Disable

☐ Master Clock Output 1

☐ Master Clock Output 2

☐ Audio Clock Input (I2S, CKIN)

Configuration

Reset Configuration

✓ User Constants

✓ NVIC Settings

✓ GPIO Settings

✓ Parameter Settings

Search Signals

Search (Ctrl+F)

Pin Name

Signal on Pin

GPIO output...

GPIO mode

GPIO Pull-u...

Maxi

MCUs Selection

Output

Series

Lines

Mcu

✓ STM32F4

STM32F407/417

STM32F407VGTx

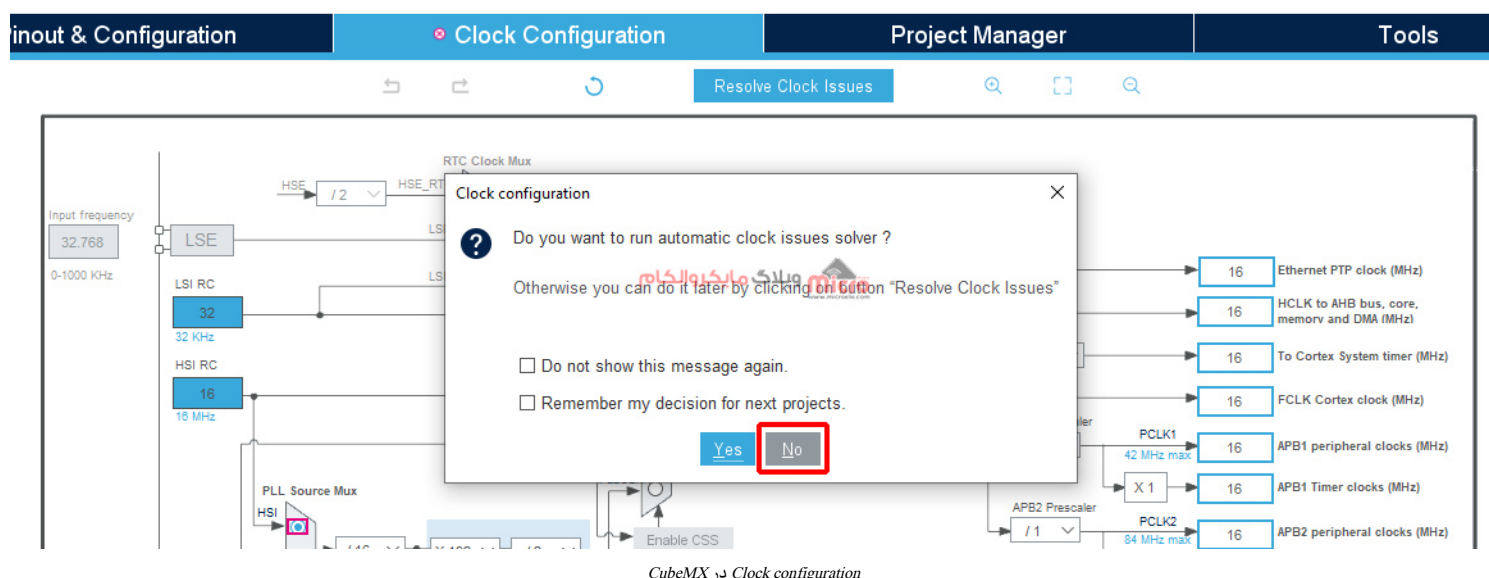
تنظیمات RCC در CubeMX



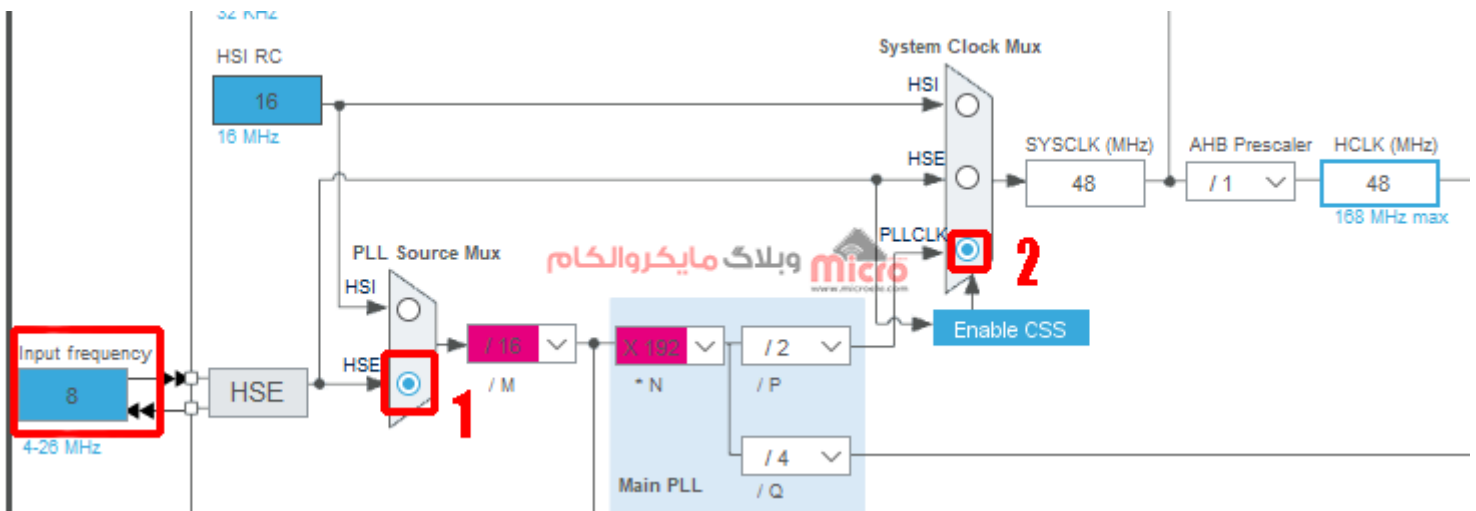
حال به سر برگ Clock Configuration می رویم.



بعد از رفتن به سربرگ Clock Configuration، پیغام تنظیم مجدد کلاک نمایش داده می شود که در این جا No را انتخاب کنید.

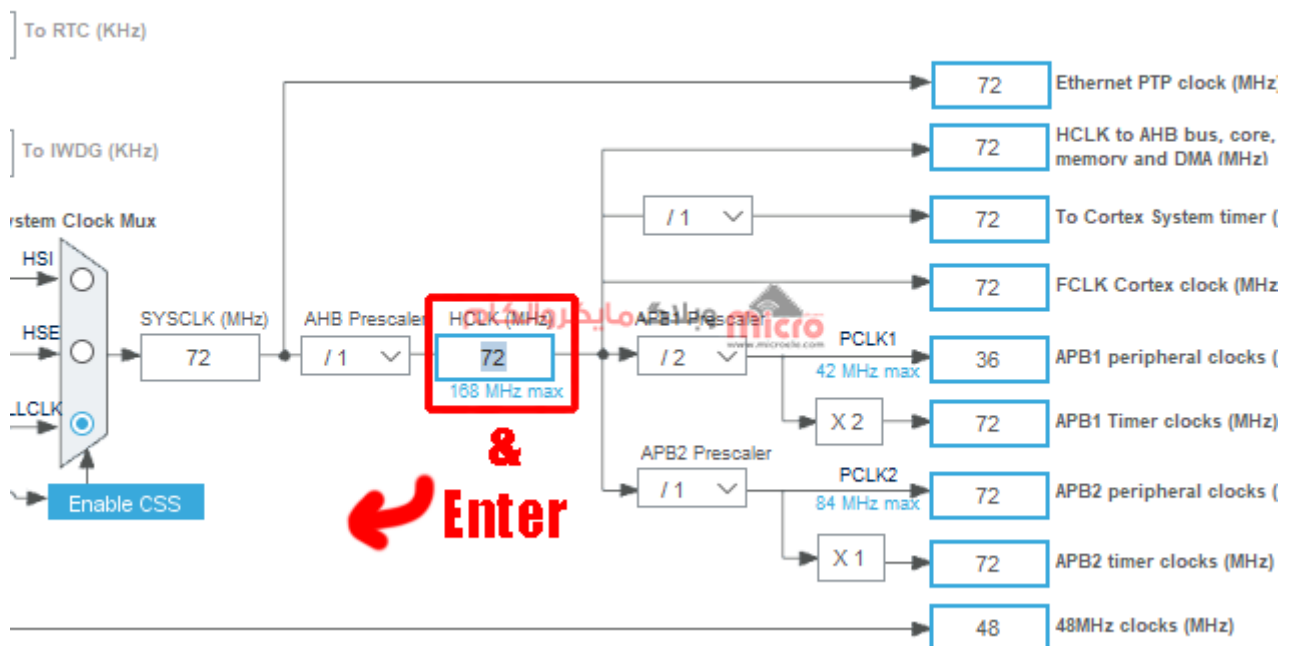


از آن جایی که بر روی برد دیسکاواری، کریستال 8 مگاهرتز قرار دارد، Input frequency را بر روی 8 قرار می دهیم. PLL Source Mux را بر روی HSE قرار داده و سپس System Clock Mux را روی PLLCLK تنظیم می کنیم.



CubeMX در Clock configuration

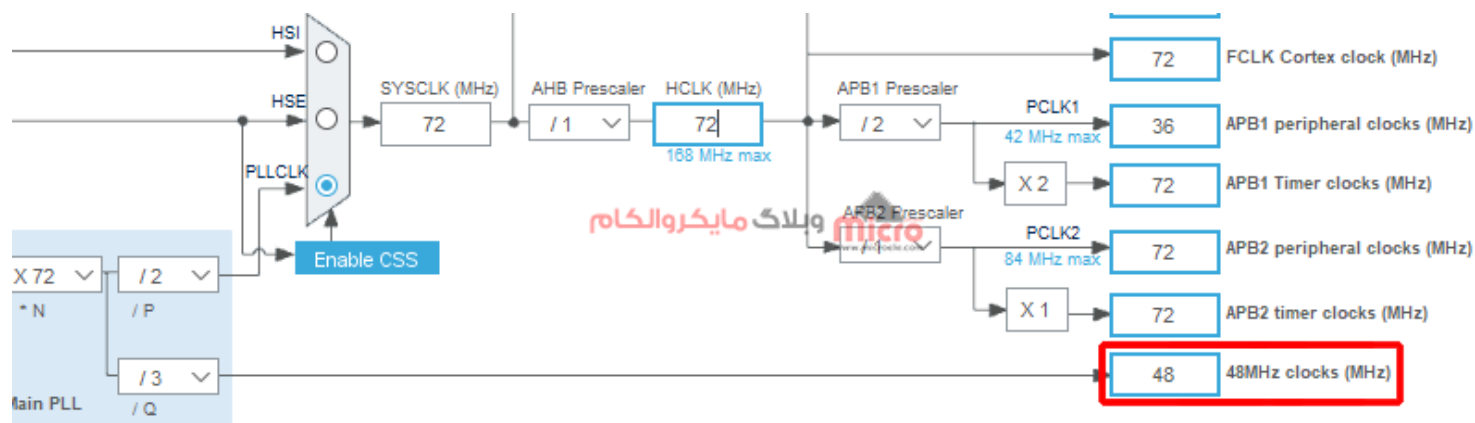
در نهایت (HCLK(MHz) را بر روی فرکانس دلخواه مثلاً 72Mhz تنظیم کرده و کلید Enter را بر روی کیبورد کامپیوتر زده تا تنظیمات مربوط به کلاک میکرو به صورت اتوماتیک توسط نرم افزار Cube محاسبه و اعمال گردد.



CubeMX در Clock configuration



برای اطمینان از درست بودن تنظیماتِ مقادیرِ مربوط به کلاک، باید فرکانس بخش USB بر روی 48MHz تنظیم شده باشد.



CubeMX در Clock configuration

برای راه اندازی بخش USB لازم است که در بخش Project Manager، سائز Heap و Stack را افزایش دهیم. مقدار Heap را به 0x2000 و Stack را به 0x4000 تغییر دهید. نام پروژه، محل ذخیره سازی و IDE توسعه نرم افزار، که در اینجا برای ما Keil است را انتخاب کرده و بر روی Generate Code کلیک کنید تا پروژه تولید گردد (در میکرو های سری F1 و F3 معمولا نیاز به تغییر سائز Heap و Stack نمی باشد).



STM32CubeMX Untitled*: STM32F407VGTx

STM32CubeMX File Window Help

Home STM32F407VGTx Untitled - Project Manager GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Project

Code Generator

Advanced Settings

Project Settings

Project Name
USBCDC

Project Location
F:\microele\USBCDC\PRJ\ Browse

Application Structure
Advanced ☐ Do not generate the ma...

Toolchain Folder Location
F:\microele\USBCDC\PRJ\USBCDC\

Toolchain / IDE
MDK-ARM

Min Version
V5.27 ☐ Generate Under ...

Linker Settings

Minimum Heap Size 0x2000

Minimum Stack Size 0x4000

تنظیم مقادیر Heap و Stack در Project manager

کد نویسی - بخش اول

بعد از بالا آمدن محیط Keil، از قسمت Project، فایل main.c را از پوشه Application/User/Core و فایل usbd_cdc_if.c را از پوشه Application/User/USB_DEVICE/App باز کنید.



F:\microele\USBCDC\PRJ\USBCDC\MDK-ARM\USBCDC.uvprojx - μVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project

- Project: USBCDC
 - USBCDC
 - Application/MDK-ARM
 - Application/User/Core
 - main.c**
 - stm32f4xx_it.c
 - stm32f4xx_hal_msp.c
 - Application/User/USB_DEVICE/App
 - usb_device.c
 - usb_desc.c
 - usbdc_cdc_if.c**
 - Application/User/USB_DEVICE/Target
 - Drivers/STM32F4xx_HAL_Driver
 - Drivers/CMSIS
 - Middlewares/USB_Device_Library
 - CMSIS

main.c

```
8      *
9      * <h2><center>
10     * All rights reserved
11     *
12     * This software is
13     * SLA0044, the
14     * the License.
15     *
16     *
17     *
18     */
19     /* USER CODE END
20     /* Includes ----
21     #include "main.h
22     #include "usb_desc
23
24     /* Private include
25     /* USER CODE BEG
26     #include "usbdc_c
27     /* USER CODE END
28
29     /* Private typedef
30     /* USER CODE BEG
31
32     /* USER CODE END
33
34     /* Private defin
35     /* USER CODE BEG
36     /* USER CODE END
```

Build Output

Programming Done.
Verify OK.
Flash Load finished at 02:43:11

باز کردن فایل های c در محیط Keil



در فایل `usbdc_cdc_if.c`، کتابخانه `usbdc_cdc_if.h` را کپی کرده و آن را در فایل `main.c` پیست کنید.

The screenshot shows the µVision IDE interface. The Project Explorer on the left displays the project structure for 'USB CDC'. The main editor window shows the file `main.c` with the following code:

```
15  * the License. You may obtain a copy of the License at:
16  * www.st.com/SLA0044
17  *
18  *
19  */
20  /* USER CODE END Header */
21
22  /* Includes -----
23  #include "usbdc_cdc_if.h"
24
25  /* USER CODE BEGIN INCLUDE
26
27  /* USER CODE END INCLUDE
28
29  /* Private typedef -----
30  /* Private define -----
31  /* Private macro -----
32
33  /* USER CODE BEGIN PV */
34  /* Private variables -----
35
36  /* USER CODE END PV */
37
38  /** @addtogroup STM32_USB_CDC_DRIVER
39  * @brief Usb device library
40  * @{
41  *
42  */
```

A context menu is open over the `#include "usbdc_cdc_if.h"` line, showing options like 'Split Window horizontally', 'Open document "usbdc_cdc_if.h"', 'Insert \'#include file\'', 'Toggle Header/Code File', 'Insert/Remove Breakpoint', 'Enable/Disable Breakpoint', 'Insert/Remove Bookmark', 'Undo', 'Redo', 'Cut', 'Copy', 'Paste', 'Select All', 'Outlining', and 'Advanced'. The 'Copy' option is highlighted.

The Build Output window at the bottom shows the following messages:

```
Programming Done.
Verify OK.
Flash Load finished at 02:43:11
```

کپی کردن کتابخانه `usbdc_cdc_if.h`



F:\microele\USBCDC\PRJ\USBCDC\MDK-ARM\USBCDC.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project: USBCDC

- Application/MDK-ARM
- Application/User/Core
 - main.c
 - stm32f4xx_it.c
 - stm32f4xx_hal_msp.c
- Application/User/USB_DEVICE/App
 - usb_device.c
 - usbd_desc.c
 - usbd_cdc_if.c
- Application/User/USB_DEVICE/Target
- Drivers/STM32F4xx_HAL_Driver
- Drivers/CMSIS
- Middlewares/USB_Device_Library
- CMSIS

```
8  *
9  * <h2><center>&copy; Copyright (c) 2021 STMicroelectronics
10 * All rights reserved.</center></h2>
11 *
12 * This software component is licensed by ST under Ultralight
13 * SLA0044, the "License"; You may not use this file except in
14 * accordance with the License. You may obtain a copy of the License at
15 *
16 * www.st.com/SLA0044
17 *
18 */
19 /* USER CODE END Header */
20 /* Includes -----
21 #include "main.h"
22 #include "usb_device.h"
23 -----
24 /* Private includes -----
25 /* USER CODE BEGIN Includes */
26 #include "usbd_cdc_if.h"
27 /* USER CODE END Includes */
28 -----
29 /* Private typedef -----
30 /* USER CODE BEGIN PTD */
31 -----
32 /* USER CODE END PTD */
33 -----
34 /* Private define -----
35 /* USER CODE BEGIN PD */
36 -----
37 /* USER CODE END PD */
```

Paste

Build Output

Programming Done.
Verify OK.
Flash Load finished at 02:43:11

کپی کردن کتابخانه usbd

بعد از پیست کردن فایل کتابخانه در main.c، حال نوبت به نوشتن برنامه می‌رسد. در قدم اول می‌خواهیم یک شمارنده ایجاد کنیم که با فشردن کلیدی که به صورت پیش فرض در برد دیسکاواری به پایه PA0 متصل است، مقدار افزایش یافته و در نهایت این مقدار از طریق پورت USB به کامپیوتر جهت نمایش منتقل شود.



ابتدا کتابخانه stdio.h را به فایل main.c اضافه کرده و متغیر cnt از نوع int و آرایه s از نوع uint8_t را تعریف می‌کنیم.

F:\microele\USBCDC\PRJ\USBCDC\MDK-ARM\USBCDC.uvprojx - μVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project

Project: USBCDC

- USBCDC
 - Application/MDK-ARM
 - Application/User/Core
 - main.c
 - stm32f4xx_it.c
 - stm32f4xx_hal_msp.c
 - Application/User/USB_DEVICE/App
 - usb_device.c
 - usbd_desc.c
 - usbd_cdc_if.c
 - Application/User/USB_DEVICE/Target
 - Drivers/STM32F4xx_HAL_Driver
 - Drivers/CMSIS
 - Middlewares/USB_Device_Library
 - CMSIS

main.c

```
13  * SLA0044, the "License"; You may not use this f
14  * the License. You may obtain a copy of the Lice
15  *
16  *
17  *
18  */
19  /* USER CODE END Header */
20  /* Includes -----
21  #include "main.h"
22  #include "usb_device.h"
23
24  /* Private includes -----
25  /* USER CODE BEGIN Includes */
26  #include "stdio.h"
27  #include "usbd_cdc_if.h"
28  /* USER CODE END Includes */
29
30  /* Private typedef -----
31  /* USER CODE BEGIN PTD */
32
33  /* USER CODE END PTD */
34
35  /* Private define -----
```

Build Output

Build target 'USBCDC'

"USBCDC\USBCDC.axf" - 0 Error(s), 0 Warning(s).

Build Time Elapsed: 00:00:00

افزودن کتابخانه stdio.h در فایل main



F:\microele\USBCDC\PRJ\USBCDC\MDK-ARM\USBCDC.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

USBCDC

Project

- Project: USBCDC
 - USBCDC
 - Application/MDK-ARM
 - Application/User/Core
 - main.c
 - stm32f4xx_it.c
 - stm32f4xx_hal_msp.c
 - Application/User/USB_DEVICE/App
 - usb_device.c
 - usb_desc.c
 - usbd_cdc_if.c
 - Application/User/USB_DEVICE/Target
 - Drivers/STM32F4xx_HAL_Driver
 - Drivers/CMSIS
 - Middlewares/USB_Device_Library
 - CMSIS

main.c

```
35  /* USER CODE BEGIN PD */
36  /* USER CODE END PD */
37
38  /* Private macro -----
39  /* USER CODE BEGIN PM */
40
41  /* USER CODE END PM */
42
43  /* Private variables -----
44
45  /* USER CODE BEGIN PV */
46  int cnt=0;
47  uint8 t s[10];
48  /* USER CODE END PV */
49
50  /* Private function prototypes -----
51  void SystemClock_Config(void);
52  static void MX_GPIO_Init(void);
53  /* USER CODE BEGIN PFP */
54
55  /* USER CODE END PFP */
56
57  /* Private user code -----
```

Build Output

Programming Done.
Verify OK.
Flash Load finished at 02:43:11

تعریف متغیر ها در فایل main

اضافه کردن کتابخانه:

```
#include "stdio.h"
```



تعریف متغیر ها:

```
int cnt=0;  
uint8_t s[10];
```

قبل از حلقه اصلی، پایه PC0 را صفر می‌کنیم تا بخش مربوط به تغذیه USB_FS در برد دیسکاوری F407 فعال شود. در داخل حلقه While، شرط مربوط به فشردن کلید و بالا رفتن شمارنده را نوشته و از طریق sprintf مقادیر را برای ارسال، به رشته تبدیل می‌کنیم. با استفاده از دستور CDC_Transmit_FS می‌توانیم رشته را از برد به USB و در نهایت به کامپیوتر ارسال کنیم.



F:\microele\USBCDC\PRJ\USBCDC\MDK-ARM\USBCDC.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project: USBCDC

- USBCDC
 - Application/MDK-ARM
 - Application/User/Core
 - main.c
 - stm32f4xx_it.c
 - stm32f4xx_hal_msp.c
 - Application/User/USB_DEVICE/App
 - usb_device.c
 - usb_desc.c
 - usb_cdc_if.c
 - Application/User/USB_DEVICE/Target
 - Drivers/STM32F4xx_HAL_Driver
 - Drivers/CMSIS
 - Middlewares/USB_Device_Library
 - CMSIS

```
86  /* USER CODE END SysInit */
87
88  /* Initialize all configured peripherals */
89  MX_GPIO_Init();
90  MX_USB_DEVICE_Init();
91  /* USER CODE BEGIN 2 */
92  HAL_GPIO_WritePin(GPIOC,GPIO_PIN_0,GPIO_PIN_RESET);
93  /* USER CODE END 2 */
94
95  /* Infinite loop */
96  /* USER CODE BEGIN WHILE */
97  while (1)
98  {
99      /* USER CODE END WHILE */
100
101      /* USER CODE BEGIN 3 */
102      if (HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0) == 1)
103      {
104          cnt++;
105          sprintf(s,"%04d\r\n",cnt);
106          CDC_Transmit_FS(s,strlen(s));
107          HAL_Delay(300);
108      }
109  }
110  /* USER CODE END 3 */
```

Build Output

Build target 'USBCDC'

"USBCDC\USBCDC.axf" - 0 Error(s), 0 Warning(s).

Build Time Elapsed: 00:00:06

کدنویسی بخش شمارنده در فایل main

در دستور CDC_Transmit_FS عبارت STR، رشته و یا متغیر رشته‌ای و L طول رشته آن رشته می‌باشد.

```
CDC_Transmit_FS(STR,L);
```

به عنوان مثال برای آن که یک کلمه مانند microele که 8 کاراکتر دارد را ارسال کنیم، تابع به صورت زیر تکمیل می‌گردد:



```
CDC_Transmit_FS(microele,8)
```

همچنین برای ارسال آرایه رشته‌ای می‌توانیم به شکل زیر عمل کنیم:

```
uint8_t str1[]="microele";  
CDC_Transmit_FS(str1,strlen(str1));
```

توجه شود که با استفاده از دستور strlen، تعداد کاراکترهای رشته مستقیماً استخراج می‌شوند.

```
/* USER CODE BEGIN 2 */  
HAL_GPIO_WritePin(GPIOC,GPIO_PIN_0,GPIO_PIN_RESET);  
/* USER CODE END 2 */  
/* Infinite loop */  
/* USER CODE BEGIN WHILE */  
while (1)  
{  
/* USER CODE END WHILE */  
/* USER CODE BEGIN 3 */  
if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0) == 1)  
{  
cnt++;  
sprintf(s,"%04d\r\n",cnt);  
CDC_Transmit_FS(s,strlen(s));  
HAL_Delay(300);  
}  
}  
/* USER CODE END 3 */
```

تا این جای کار کدهای مربوط به شمارنده و ارسال آن‌ها به USB جهت نمایش در کامپیوتر را انجام دادیم. حال می‌خواهیم LED های روی برد را از طریق کامپیوتر و پورت USB کنترل کنیم و بخش دریافت داده‌ها را تکمیل کنیم.



کد نویسی - بخش دوم

حال می‌خواهیم LED های روی برد را از طریق کامپیوتر و پورت USB کنترل کنیم و بخش دریافت داده‌ها را تکمیل کنیم. به فایل usbd_cdc_if.c رفته و تابع CDC_Receive_FS را در آن پیدا کنید.

```
257 *
258 * @param Buf: Buffer of data to be received
259 * @param Len: Number of data received (in bytes)
260 * @retval Result of the operation: USB_OK if all operati
261 */
262 static int8_t CDC_Receive_FS(uint8_t* Buf, uint32_t *Len)
263 {
264     /* USER CODE BEGIN 6 */
265
266     USBDCDC_SetRxBuffer(&hUsbDeviceFS, Buf[0]);
267     USBDCDC_ReceivePacket(&hUsbDeviceFS);
268     return (USB_OK);
269     /* USER CODE END 6 */
270 }
271
272 /**
273 * @brief CDC_Transmit_FS
274 * Data to send over USB IN endpoint are sent over
275 * through this function.
276 * @note
277 *
```

Build target 'USBCDC'
"USBCDC\USBCDC.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:06

دریافت داده در فایل usbd_cdc_if

با هر بار ارسال دیتا از کامپیوتر به برد، این تابع به صورت اتوماتیک فراخوانی شده و کاراکترهای دریافتی درون آرایه Buf قرار خواهند گرفت. با خواندن مقدار Buf[0] می‌توانیم کاراکتر مورد نظر را خوانده و دستورات مربوط به آن را کد نویسی کنیم.



F:\microele\USBCDC\PRJ\USBCDC\MDK-ARM\USBCDC.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project: USBCDC

- USBCDC
 - Application/MDK-ARM
 - Application/User/Core
 - main.c
 - stm32f4xx_it.c
 - stm32f4xx_hal_msp.c
 - Application/User/USB_DEVICE/App
 - usb_device.c
 - usbd_desc.c
 - usbd_cdc_if.c
 - Application/User/USB_DEVICE/Target
 - Drivers/STM32F4xx_HAL_Driver
 - Drivers/CMSIS
 - Middlewares/USB_Device_Library
 - CMSIS

```
261  /*
262  static int8_t CDC_Receive_FS(uint8_t* Buf, uint32_t *Len)
263  {
264      /* USER CODE BEGIN 6 */
265      if(Buf[0] == '1')
266      {
267          HAL_GPIO_WritePin(GPIOD, 0xf<<12, GPIO_PIN_SET);
268          CDC_Transmit_FS("\r\nLEDS ON!\r\n", 12);
269      }
270      if(Buf[0] == '0')
271      {
272          HAL_GPIO_WritePin(GPIOD, 0xf<<12, GPIO_PIN_RESET);
273          CDC_Transmit_FS("\r\nLEDS OFF!\r\n", 13);
274      }
275      USBD_CDC_SetRxBuffer(&hUsbDeviceFS, &Buf[0]);
276      USBD_CDC_ReceivePacket(&hUsbDeviceFS);
277      return (USBD_OK);
278      /* USER CODE END 6 */
279  }
```

Build Output

FromELF: creating hex file...
"USBCDC\USBCDC.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:04

کنترل LED ها در فایل usbd_cdc_if با تابع CDC_Receive_FS

```
static int8_t CDC_Receive_FS(uint8_t* Buf, uint32_t *Len)
{
    /* USER CODE BEGIN 6 */
    if(Buf[0] == '1')
    {
        HAL_GPIO_WritePin(GPIOD, 0xf<<12, GPIO_PIN_SET);
        CDC_Transmit_FS("\r\nLEDS ON!\r\n", 12);
    }
}
```

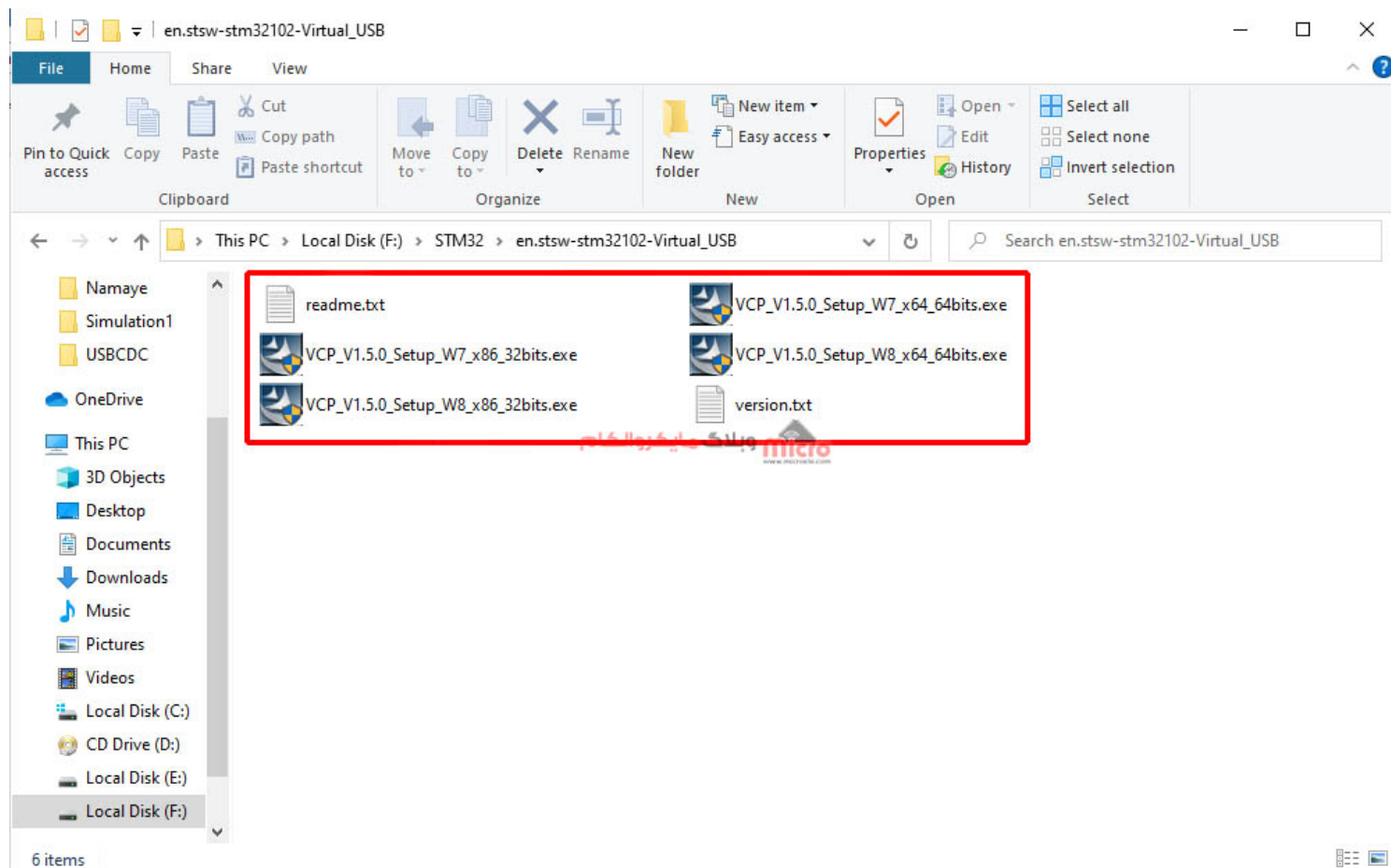


```
if(Buf[0] == '0')
{
    HAL_GPIO_WritePin(GPIOD,0xf<<12,GPIO_PIN_RESET);
    CDC_Transmit_FS("\r\nLEDS OFF!\r\n",13);
}
USBD_CDC_SetRxBuffer(&hUsbDeviceFS, &Buf[0]);
USBD_CDC_ReceivePacket(&hUsbDeviceFS);
return (USBD_OK);
/* USER CODE END 6 */
}
```

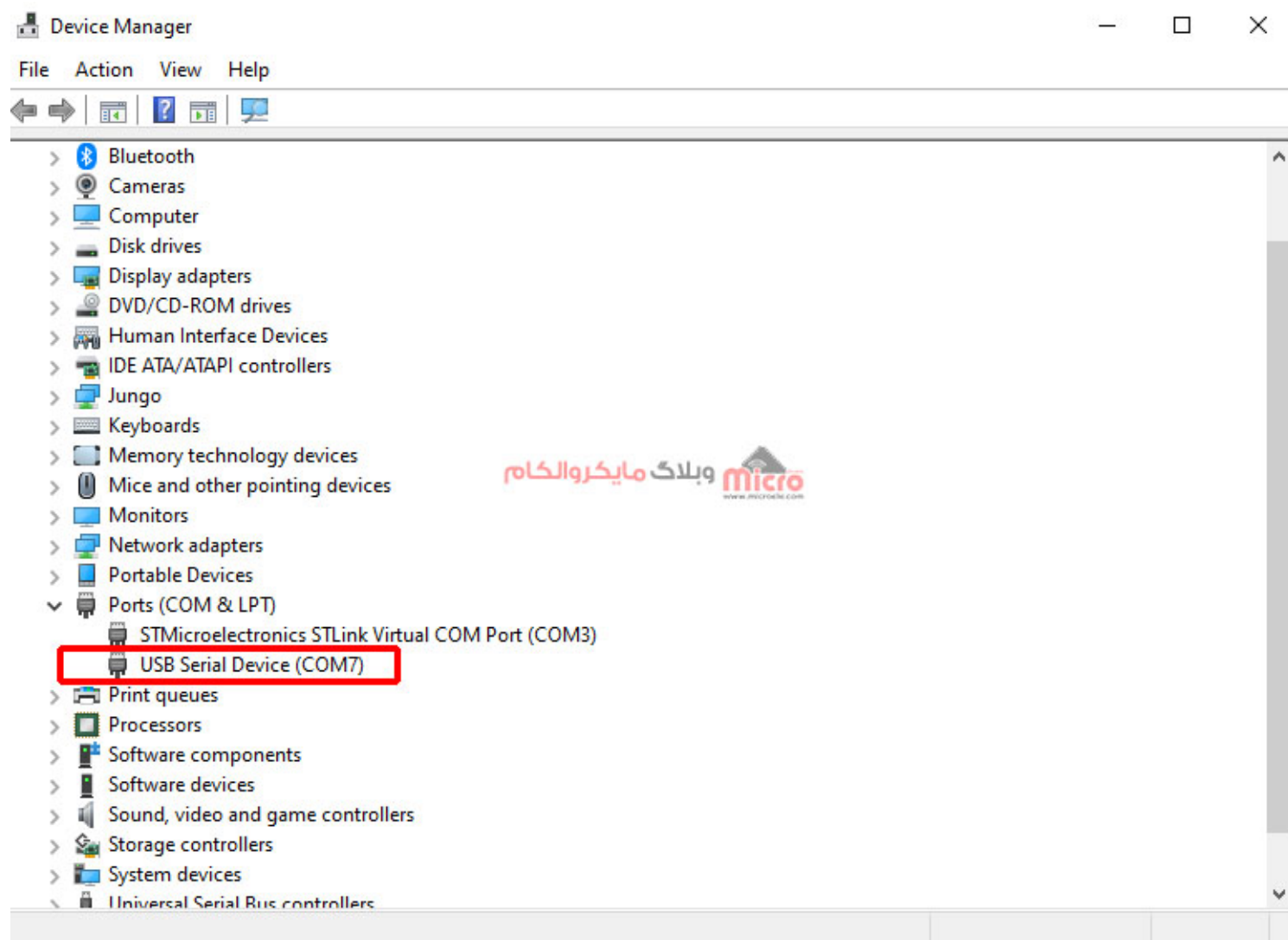
اگر کاراکتر '1' از کامپیوتر به برد ارسال شود، تمام LED ها روشن شده و پیام LEDS ON! از طرف برد به کامپیوتر جهت نمایش ارسال می‌گردد و اگر کاراکتر '0' ارسال شود، تمام LED ها خاموش شده و پیام LEDS OFF! نمایش داده خواهد شد.

برنامه را Build و بر روی برد دیسکاواری LOAD می‌کنیم. برای اتصال برد دیسکاواری F407 به کامپیوتر و استفاده از USB_FS، لازم است که هر دو درگاه USB که بر روی برد دیسکاواری وجود دارد را به پورت USB کامپیوتر متصل کنیم. بر روی برد دیسکاواری یک درگاه USB-MINI برای ST-LINK وجود دارد که تغذیه برد نیز از همین درگاه تامین می‌شود. و دیگری یک درگاه USB-MICRO است که مربوط به USB-FS بوده که پورت سریال مجازی نیز بر روی همین درگاه ایجاد شده است.

اگر به Device manager در ویندوز بروید ممکن است که پورت USB میکرو در کامپیوتر شما شناخته نشده باشد. به همین منظور لازم است که درایور های USB VCP را از [این لینک](#) دانلود کرده و با توجه به نوع سیستم عامل خود، اقدام به نصب درایور کنید.



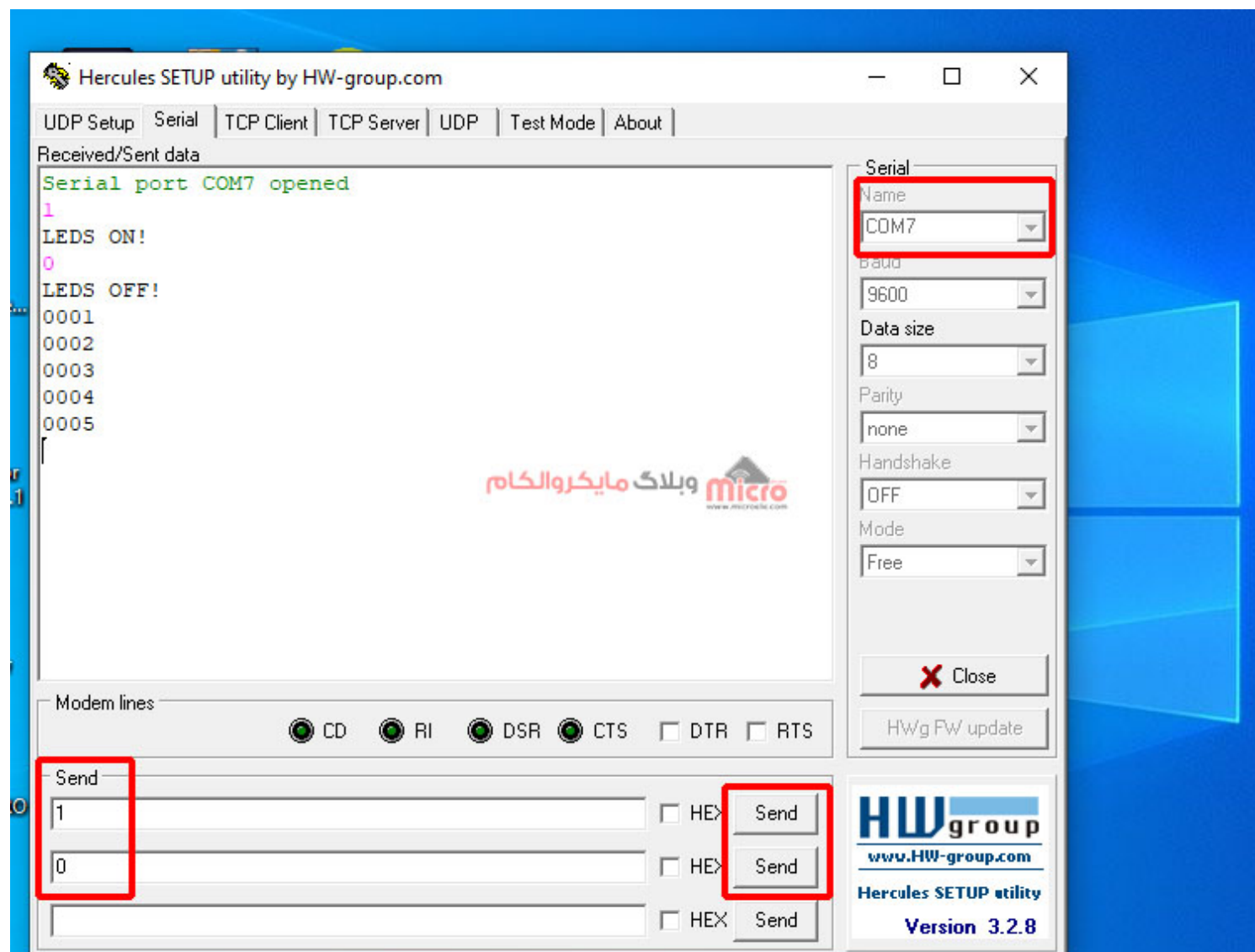
نصب درایور USB STM32 VCP



شناسایی پورت مجازی توسط ویندوز در Device Manager

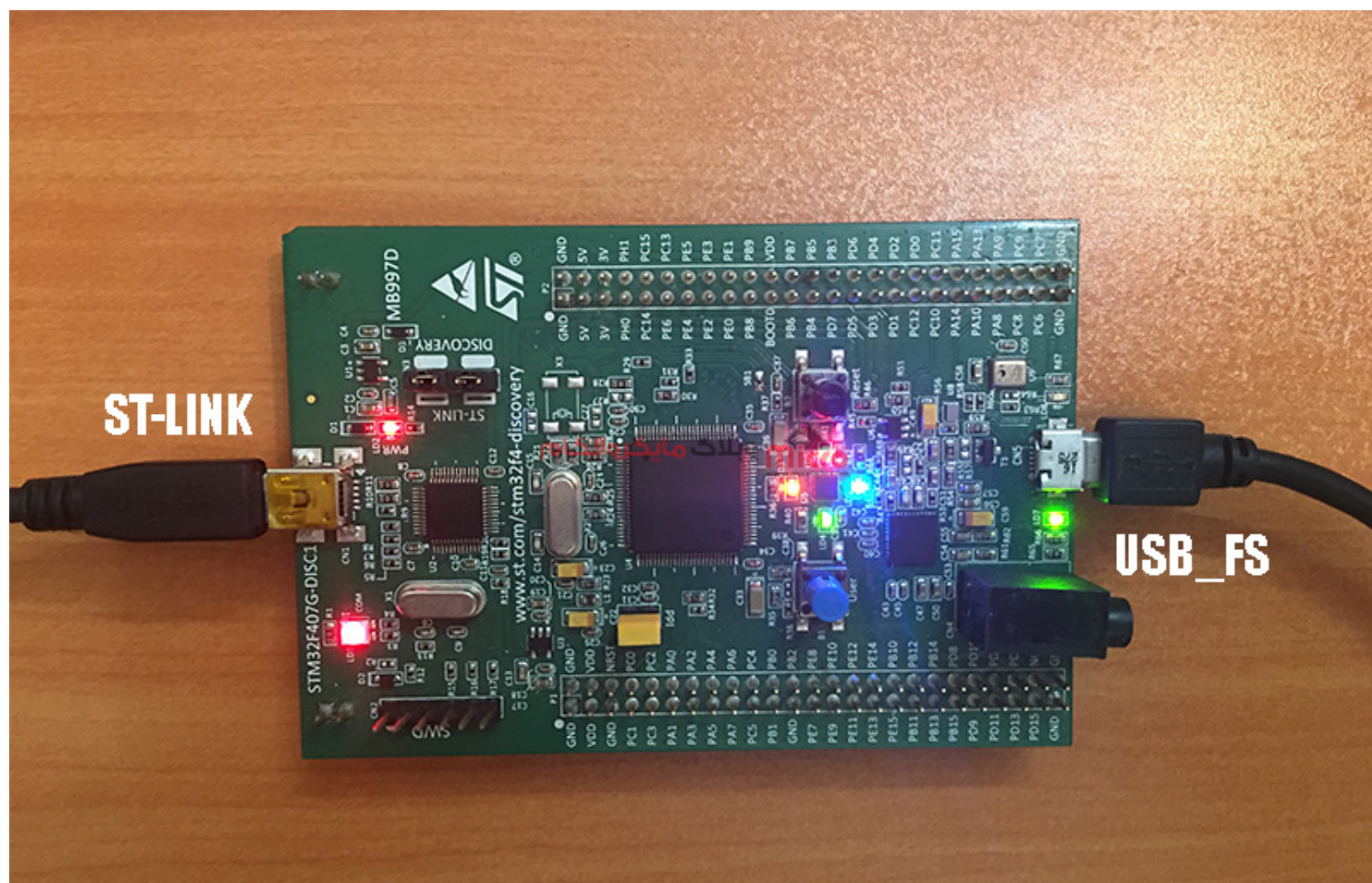
برای من پورت سریال مجازی به صورت پیش فرض بر روی COM7 تنظیم شده است. حال کافیهست که با برنامه سریال ترمینال، برنامه نوشته شده را تست کنیم. در این آموزش از ترمینال Hercules استفاده شده است. که می‌توانید آن را از [اینجا](#) دانلود کنید.

تنظیمات پورت را بر روی COM7 قرار داده و Baud rate را به صورت دلخواه تنظیم کرده و در آخر بر روی Open برای باز کردن پورت کلیک کنید.



ارسال و دریافت دیتا با برنامه Hercules

با ارسال 1 و 2، LED ها روشن و خاموش شده و پیام مربوط به وضعیت آن‌ها در ترمینال نشان داده می‌شود. همچنین با زدن کلید روی برد، مقادیر شمارنده افزایش پیدا کرده و بر روی ترمینال نمایش داده می‌شود.



برد Discovery F407

نتیجه آموزش

نتیجه گیری

در این مطلب توانستیم با راه اندازی USB در میکروکنترلر STM32F407، بین کامپیوتر و میکروکنترلر، بدون استفاده از آیسایهای واسط مبدل USB به سریال، داده ها را ارسال و دریافت کنیم. در نرم افزار CubeMX تنظیمات مربوط به



USB_OTG_FS را انجام داده و از طریق ترمینال سریال، LED ها و کلید روی برد دیسکاواری را کنترل کردیم. از این روند می توان در میکروکنترلرها و سری های دیگر STM32 نیز استفاده نمایید.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورتی که هرگونه نظر یا سوال داشتید درباره این آموزش لطفاً اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتماً با دوستانتون به اشتراک بگذارید. همینطور میتونید اون رو پس از اجرای عملی توی اینستاگرام با هشتگ #microelecom به اشتراک بگذارید و [پیج مایکروالکام](#) (@microelecom) رو هم منشن کنید.