

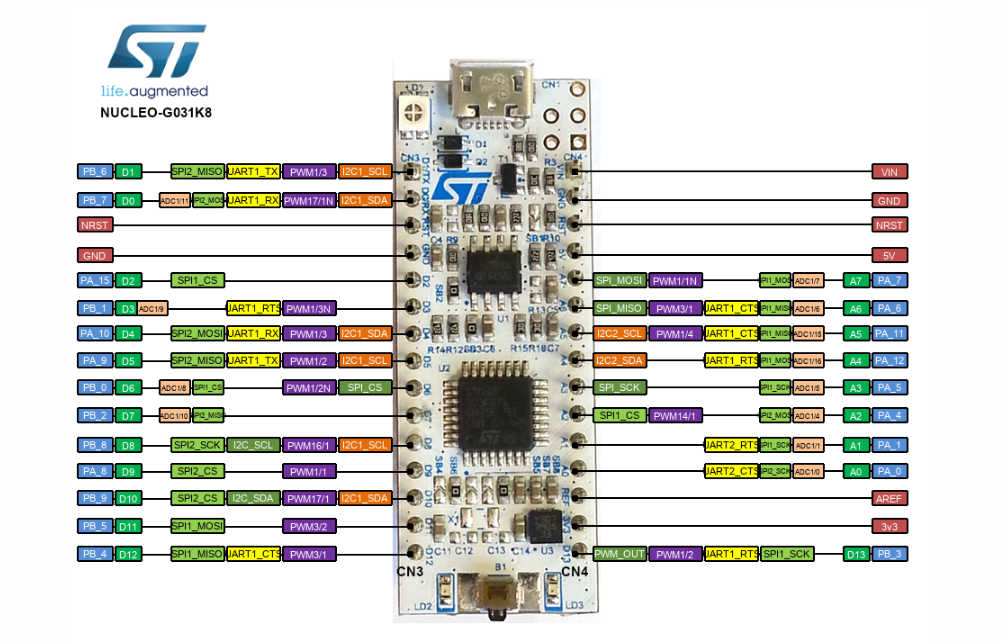
GEBZE TECHNİCAL UNIVERTİSY

ELEC 335

LAB 1 REPORT

BERKAY TÜRK

171024024



• STM32G031K8T6U microcontroller (Arm® Cortex®-M0+ at 64 MHz) in LQFP32 package, featuring

64 Kbytes of Flash memory and 8 Kbytes of SRAM

• 1 user LED

• 1 RESET or user push-button

• Board connectors:

– Arduino™ Nano V3 expansion connector

– USB with Micro-B

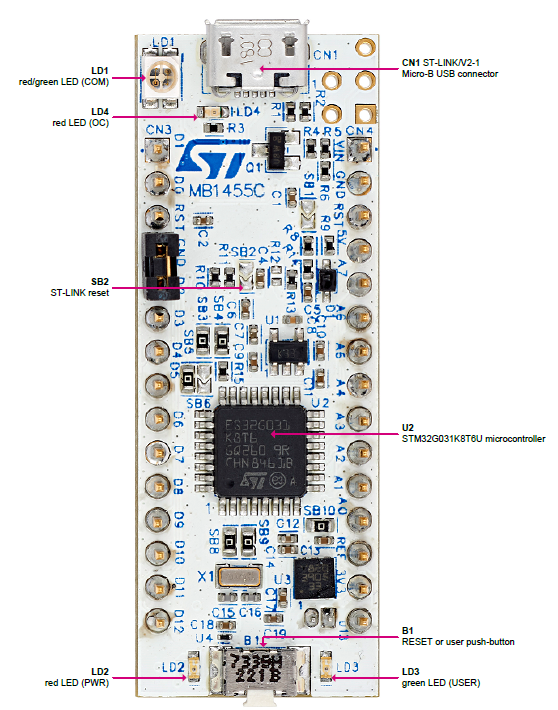
• Flexible power-supply options: ST-LINK USB VBUS or external sources

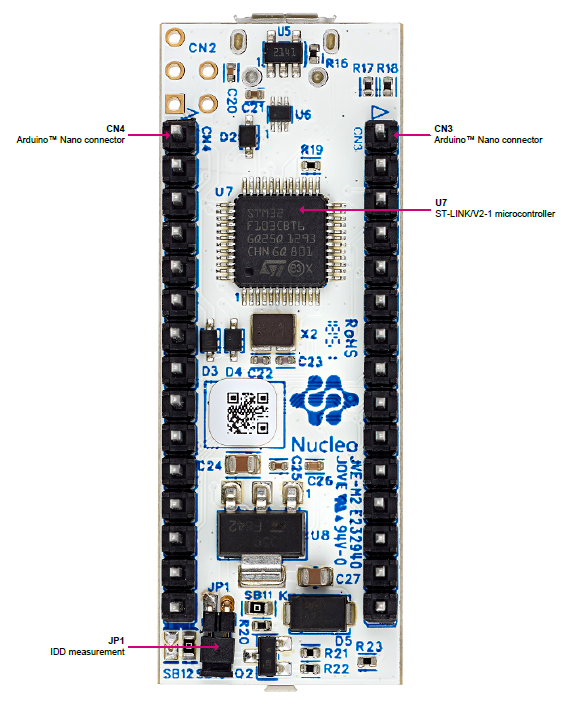
• On-board ST-LINK/V2-1 debugger/programmer with USB re-enumeration capability: mass storage, Virtual

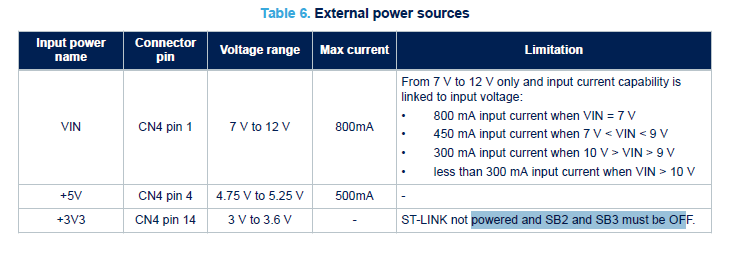
COM port, and debug port

**ICs:**

**Microcontroller (CPU):** The STM32G031x4/x6/x8 mainstream microcontrollers are based on high-performance Arm® Cortex®-M0+ 32-bit RISC core operating at up to 64 MHz frequency. Offering a high level of integration, they are suitable for a wide range of applications in consumer, industrial and appliance domains and ready for the Internet of Things (IoT) solutions. The devices incorporate a memory protection unit (MPU), high-speed embedded memories (8 Kbytes of SRAM and up to 64 Kbytes of Flash program memory with read protection, write protection, proprietary code protection, and securable area), DMA, an extensive range of system functions, enhanced I/Os, and peripherals.







**Pin Connections:**

**VIN:** Input voltage on any other pin. Min=-0.3V,Max=4.0V

**GND:** (GROUND )

**NRST:**It resets chip in other words.İn short, In short, NRST pin is used to activate bootloader when you need to flash new program using one of the communication channels and to reset MCU to default state when you want to flash new program using SWD interface and SWD pins are not available.

**5V:**It provides the system’s power.

It manages the Flash memory reprogramming through USART on pins PA9/PA10 or PA2/PA3, or through I2C-bus on pins PB6/PB7 or PB10/PB11.

**+3V3 power supply**

Using the +3V3 (CN4 pin 14) directly as power input is interesting, for instance if the 3.3 V is provided by a shield

board. In this case, the ST-LINK is not powered, thus programming and debugging features are not available.

When the board is powered by +3V3 (CN4 pin 14), solder bridges SB2 (NRST) and SB3 must be OFF.

**LD1 ST-LINK COM LED**

The bicolor LED LD1 (green, red) provides information about ST-LINK communication status. LD1 default color is

red. LD1 turns to green to indicate that communication is in progress between the PC and the ST-LINK, with the

following setup:

• Blinking red: the first USB enumeration with the PC is taking place

• Red on: the initialization between the PC and ST-LINK is complete

• Blinking red or green: programming and debugging with target

• Green on: communication finished and successful

• Orange on: communication failure

**LD2 PWR**

The red LED indicates that the STM32G0 part is powered and 5 V power is available on CN4 pin 4.

**LD3 USER**

The LD3 USER green LED is connected to the following STM32G031K8T6 I/O:

• PB3, if the configuration is SB12 ON, and SB13 OFF

• PC6, if the configuration is SB12 OFF, and SB13 ON (default configuration)

It is also connected to the Arduino™ D13 signal when SB12 is ON. To light this LED, a high-logic state “1” must be

written in the corresponding GPIO PB3 or PC6. A transistor drives the LED, therefore its consumption does not

affect the STM32G0 power measurement.

**LD4 USB power fault (OC, overcurrent)**

The LD4 red LED indicates that the board power consumption on USB ST-LINK exceeds 500 mA. Therefore, the

user must check the root cause of the overconsumption, and consequently power the STM32G0 Nucleo-32 board

with an external power supply if needed.

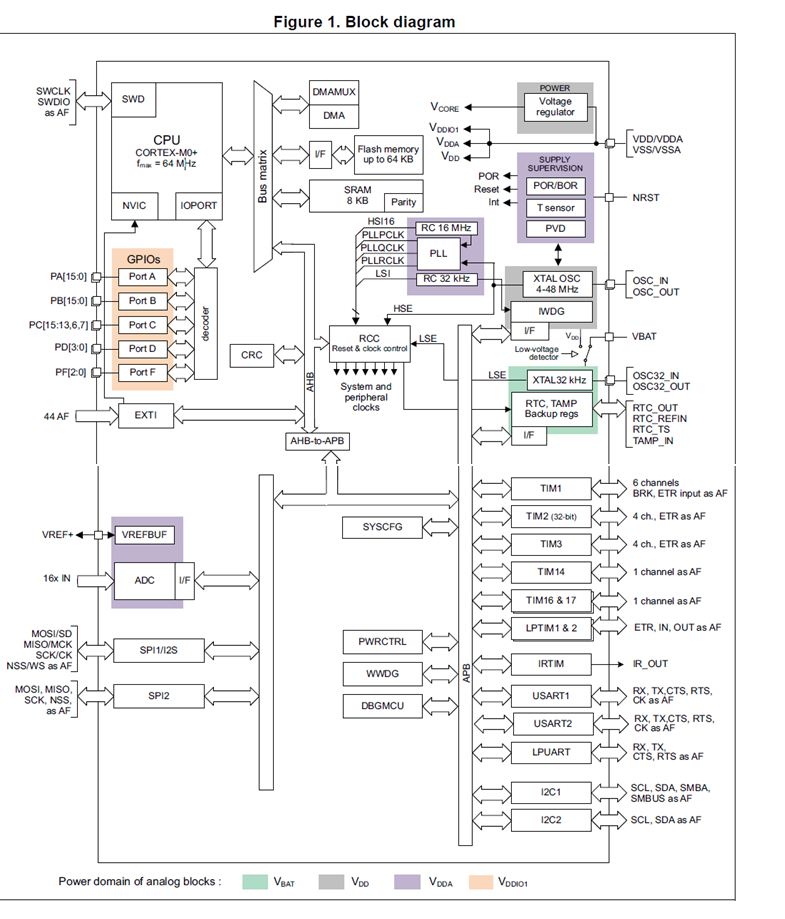
**B1 RESET/USER**

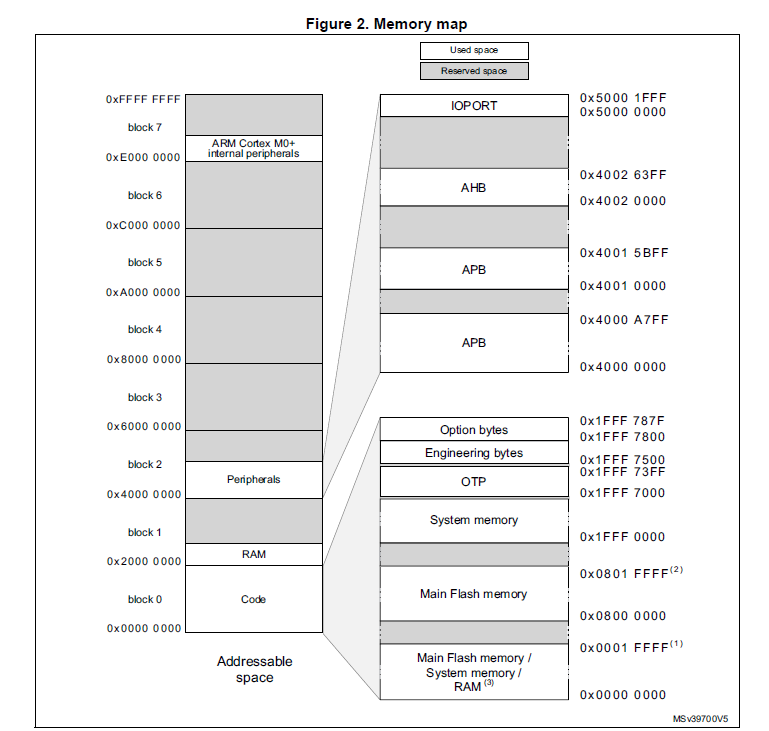
This push-button is connected to NRST (PF2-NRST) and is used to reset the STM32G0 microcontroller or to

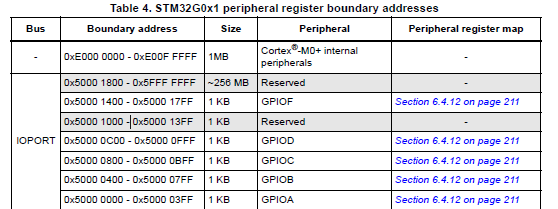
generate a USER event.

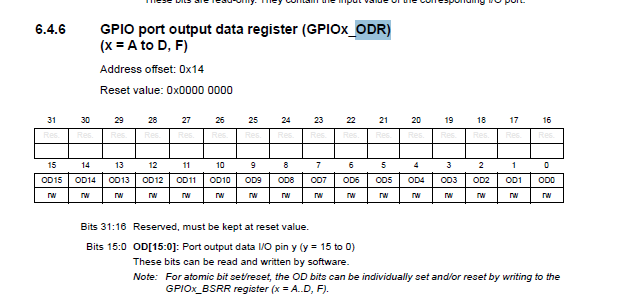
**Virtual COM port (VCP): USART**

The STM32G0 Nucleo-32 board offers the possibility to connect a USART interface to the ST-LINK/V2-1.









**CODE 1:**

Stack\_Size EQU 0x00004000

AREA STACK, NOINIT, READWRITE, ALIGN=3

Stack\_Mem SPACE Stack\_Size

\_\_initial\_sp

THUMB

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD \_\_initial\_sp ; Top of Stack

DCD Reset\_Handler ; Reset Handler

DCD NMI\_Handler ; NMI Handler

DCD HardFault\_Handler ; Hard Fault Handler

AREA |.text|, CODE, READONLY

; nmi handler

NMI\_Handler PROC

EXPORT NMI\_Handler

B .

ENDP

; hardfault handler

HardFault\_Handler PROC

EXPORT HardFault\_Handler

B .

ENDP

; entry function

Reset\_Handler PROC

EXPORT Reset\_Handler

; Edit below this line

GPIOA\_ODR equ 0x50000014

LDR r1 ,=GPIOA\_ODR

LDR r3,[r1]

LDR r2 ,=0x100

ORRS r3,r3 ,r2

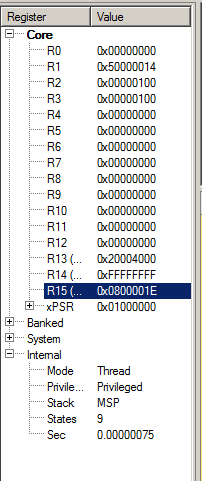
STR r3,[r1]

; Edit above this line

B .

ENDP

END



**CODE 2:**

Stack\_Size EQU 0x00004000

AREA STACK, NOINIT, READWRITE, ALIGN=3

Stack\_Mem SPACE Stack\_Size

\_\_initial\_sp

THUMB

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD \_\_initial\_sp ; Top of Stack

DCD Reset\_Handler ; Reset Handler

DCD NMI\_Handler ; NMI Handler

DCD HardFault\_Handler ; Hard Fault Handler

AREA |.text|, CODE, READONLY

; nmi handler

NMI\_Handler PROC

EXPORT NMI\_Handler

B .

ENDP

; hardfault handler

HardFault\_Handler PROC

EXPORT HardFault\_Handler

B .

ENDP

; entry function

Reset\_Handler PROC

EXPORT Reset\_Handler

; Edit below this line

GPIOA\_ODR equ 0x50000014

GPIOB\_ODR equ 0x50000414

LDR r0 ,=GPIOA\_ODR

LDR r3,[r0]

LDR r1 ,=0x800

ORRS r3,r3 ,r1

STR r3,[r0]

LDR r2 ,=GPIOA\_ODR

LDR r3,[r2]

LDR r1 ,=0x1000

ORRS r3,r3 ,r1

STR r3,[r2]

LDR r5 ,=GPIOB\_ODR

LDR r3,[r5]

LDR r1 ,=0x10

ORRS r3,r3 ,r1

STR r3,[r5]

LDR r6 ,=GPIOB\_ODR

LDR r3,[r6]

LDR r1 ,=0x20

ORRS r3,r3 ,r1

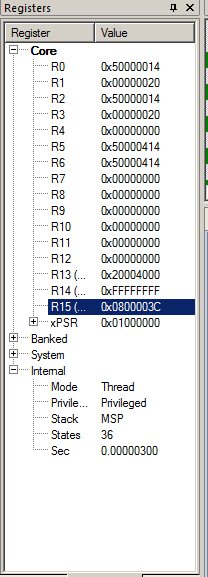
STR r3,[r6]

; Edit above this line

B .

ENDP

END



**CODE 3:**

Stack\_Size EQU 0x00004000

AREA STACK, NOINIT, READWRITE, ALIGN=3

Stack\_Mem SPACE Stack\_Size

\_\_initial\_sp

THUMB

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD \_\_initial\_sp ; Top of Stack

DCD Reset\_Handler ; Reset Handler

DCD NMI\_Handler ; NMI Handler

DCD HardFault\_Handler ; Hard Fault Handler

AREA |.text|, CODE, READONLY

; nmi handler

NMI\_Handler PROC

EXPORT NMI\_Handler

B .

ENDP

; hardfault handler

HardFault\_Handler PROC

EXPORT HardFault\_Handler

B .

ENDP

; entry function

Reset\_Handler PROC

EXPORT Reset\_Handler

; Edit below this line

GPIOA\_ODR equ (0x50000014)

loop1

LDR r1 ,=GPIOA\_ODR

LDR r3,[r1]

LDR r2 ,=0x100

ORRS r3,r3 ,r2

STR r3,[r1]

LDR r1 ,=GPIOA\_ODR

LDR r3,[r1]

LDR r2 ,=0x0

ANDS r3,r3,r2 // RESET VALUE

STR r3,[r1]

;delay:

MOVS r0,#100

loop2

SUBS r0,#1

BNE loop2

MOVS r4,#10 ;LED 10 KERE 1 ER SANIYE ARALIKLARLA YANACAKTIR.

SUBS r4,#1

BNE loop1

; Edit above this line

B .

ENDP

END