

BEÁGYAZOTT RENDSZEREK (C)

3. forduló



A kategória támogatója: Robert Bosch Kft.

Ismertető a feladathoz



RENDELKEZÉSRE ÁLLÓ IDŐ:

60:00

Ezután a forduló után automatikusan jár a [kitartóknak szóló garantált ajándékunk](#), érdemes kitöltened a feladatlapot! :)

Ha kifutsz az adott feladatlap kitöltésére rendelkezésre álló időből, a felület **automatikusan megpróbálja beküldeni** az addig megadott válaszokat.

A kérdésekre **mindig van helyes válasz**, olyan kérdés viszont nincs, amelyre az összes válasz helyes!

Egyéb információkat a [versenyszabályzatban](#) találsz!

Harmadik forduló

Ájóti János ezen a héten folytatja megkezdett projektjét. Miután már be tudja állítani a kívánt hőmérsékletet, szeretné, ha meg is tudná mérni azt. A méréshez egy TMP102-es hőmérséklet szenzort választ, melyből I2C protokoll segítségével olvassa ki az aktuálisan mért hőmérsékletet. Az SDA és SCL lábakat rendre a NUCLEO-F091RC board PB9 és PB8 lábaira köti, az ADD0 pinto GND-re. A default beállításokhoz a továbbiakban nem nyúl.

Segítség megválaszolni Ájót Jánosban felmerülő kérdéseket a projekttel kapcsolatban!

A megoldásban a következő adatlapok lesznek segítségetekre:

TMP102 adatlap:

<https://www.ti.com/lit/ds/symlink/tmp102.pdf>

STM32F091RC adatlap:

<https://www.st.com/resource/en/datasheet/stm32f091rc.pdf>

NUCLEO-F091RC adatlap:

https://www.st.com/resource/en/data_brief/nucleo-f091rc.pdf

HAL API dokumentáció:

https://www.st.com/resource/en/user_manual/dm00122015-description-of-stm32f0-hal-and-lowlayer-drivers-stmicroelectronics.pdf

Kontroller reference manual:

https://www.st.com/resource/en/reference_manual/rm0091-stm32f0x1stm32f0x2stm32f0x8-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

Felhasznált idő: 00:54/60:00

Elért pontszám: 0/25

1. feladat 0/2 pont

Mi lesz így az eszköz 7 bites címe? Az eredményt hexadecimális számként add meg, elhagyva a 0x előtagot, pl. 0xF3 esetén a válasz F3.

Válasz

A helyes válasz:

48
0x48

Magyarázat

Az adatlap 11. oldalának tetején található információ szerint az ADD0 pin GND-re történő kötése esetén az eszköz címe binárisan 1001000, vagyis 48 hexadecimálisan.

2. feladat 0/10 pont

Melyik kód valósítja meg a szenzor helyes kiolvasását? Feltételezhetjük, hogy az I2C periféria helyesen lett felkonfigurálva, és elérhető a *hi2c1* handleren keresztül. Az utolsó else ágban fog történni a későbbiekben a kiolvasott adatok konverziója.

Válasz



```
I2C_HandleTypeDef hi2c1;
static const uint8_t TMP102_ADDR = 0x48;
static const uint8_t REG_TEMP = 0x00;
void readTemp(void *argument)
{
    HAL_StatusTypeDef ret;
    uint8_t buf[12];
    /* Infinite loop */
    for(;;)
    {
        buf[0] = REG_TEMP;
        ret = HAL_I2C_Master_Transmit(&hi2c1, TMP102_ADDR, buf, 1, HAL_MAX_DELAY);
        if(HAL_OK != ret){
            strcpy((char*)buf, "Error Tx\r\n");
        } else {
            ret = HAL_I2C_Master_Receive(&hi2c1, TMP102_ADDR, buf, 2, HAL_MAX_DELAY);
            if(HAL_OK != ret){
                strcpy((char*)buf, "Error Rx\r\n");
            } else {
                /* ... */
            }
        }
        osDelay(1000);
    }
}
```



```
I2C_HandleTypeDef hi2c1;
static const uint8_t TMP102_ADDR = 0x48 << 1;
static const uint8_t REG_TEMP = 0x00;
void readTemp(void *argument)
{
    HAL_StatusTypeDef ret;
    uint8_t buf[12];
    /* Infinite loop */
    for(;;)
    {
        buf[0] = REG_TEMP;
        ret = HAL_I2C_Master_Transmit(&hi2c1, TMP102_ADDR, buf, 1, HAL_MAX_DELAY);
        if(HAL_OK != ret){
            strcpy((char*)buf, "Error Tx\r\n");
        } else {
            ret = HAL_I2C_Master_Receive(&hi2c1, TMP102_ADDR, buf, 2, HAL_MAX_DELAY);
            if(HAL_OK != ret){
                strcpy((char*)buf, "Error Rx\r\n");
            } else {
                /* ... */
            }
        }
        osDelay(1000);
    }
}
```



```

I2C_HandleTypeDef hi2c1;
static const uint8_t TMP102_ADDR = 0x48 << 1;
static const uint8_t REG_TEMP = 0x01;
void readTemp(void *argument)
{
    HAL_StatusTypeDef ret;
    uint8_t buf[12];
    /* Infinite loop */
    for(;;)
    {
        buf[0] = REG_TEMP;
        ret = HAL_I2C_Master_Transmit(&hi2c1, TMP102_ADDR, buf, 1, HAL_MAX_DELAY);
        if(HAL_OK != ret){
            strcpy((char*)buf, "Error Tx\r\n");
        } else {
            ret = HAL_I2C_Master_Receive(&hi2c1, TMP102_ADDR, buf, 2, HAL_MAX_DELAY);
            if(HAL_OK != ret){
                strcpy((char*)buf, "Error Rx\r\n");
            } else {
                /* ... */
            }
        }
        osDelay(1000);
    }
}

```



```

I2C_HandleTypeDef hi2c1;
static const uint8_t TMP102_ADDR = 0x48;
static const uint8_t REG_TEMP = 0x01;
void readTemp(void *argument)
{
    HAL_StatusTypeDef ret;
    uint8_t buf[12];
    /* Infinite loop */
    for(;;)
    {
        buf[0] = REG_TEMP;
        ret = HAL_I2C_Master_Transmit(&hi2c1, TMP102_ADDR, buf, 1, HAL_MAX_DELAY);
        if(HAL_OK != ret){
            strcpy((char*)buf, "Error Tx\r\n");
        } else {
            ret = HAL_I2C_Master_Receive(&hi2c1, TMP102_ADDR, buf, 2, HAL_MAX_DELAY);
            if(HAL_OK != ret){
                strcpy((char*)buf, "Error Rx\r\n");
            } else {
                /* ... */
            }
        }
        osDelay(1000);
    }
}

```



Egyik sem.

Magyarázat

A HAL dokumentációja alapján a 7 bites címet shiftelni kell eggyel, valamint a szenzor adatlapja szerint a 0x00 címről kell kiolvasni 2 bájtot, hogy megkapjuk a hőmérsékletet.

3. feladat 0/5 pont

Kiolvasva a regisztert binárisan a következő értéket kapja: **0001 1101 1101** (MSB-től LSB-ig). Mekkora a hőmérséklet Celsiusban, decimálisan?

Az eredmény elé tégy egy - jelet, ha a hőmérséklet negatív, pozitív esetben megadhatod előjel nélkül, vagy + jellel is a választ. Tizedesvesszőként pontot és vesszőt is egyaránt használhatsz. Például: **13.5**, **+8,5** vagy **-5,6875**

Válaszok

A helyes válasz:

29,8125

29.8125

+29,8125

+29.8125

Magyarázat

Az adatlap alapján ha a kapott érték pozitív, egyszerűen be kell szorozni 0,0625°C-kal:

0001 1101 1101 -> 0x1DD -> 477 -> $477 \cdot 0,0625^\circ\text{C}$ -> 29,8125°C

4. feladat 0/8 pont

Kiolvasva a regisztert binárisan a következő értéket kapja: **1111 0011 1110** (MSB-től LSB-ig). Mekkora a hőmérséklet Celsiusban, decimálisan?

Az eredmény elé tégy egy - jelet, ha a hőmérséklet negatív, pozitív esetben megadhatod előjel nélkül, vagy + jellel is a választ. Tizedesvesszőként pontot és vesszőt is egyaránt használhatsz. Például: **13.5**, **+8,5** vagy **-5,6875**

Válasz

A helyes válasz:

-12,125

-12.125

Magyarázat

A negatív hőmérsékletet a kettes komplementum miatt először át kell váltani:

1111 0011 1110 -> 0000 1100 0010 -> 0xC2 -> 194 -> $194 \cdot 0,0625^\circ\text{C}$ -> -12,125°C

5. feladat 0/0 pont

Egyik alkalommal a konverzió után 22,1875°C hőmérsékletet kap. Kíváncsi lesz, mi is történik pontosan az I2C buszon, ezért logikai analízátorral megvizsgálja. Melyik timing diagramot láthatta János?

Megjegyzés: a képek túl nagyok lennének, ezért egy kódot adunk csak meg a válaszok alatt, melyet illesszettek be ebbe a waveform editorba, hogy lássátok a diagramokat:

<https://wavedrom.com/editor.html>

Válasz



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1.....h|ph1.h1..h1...h1h.1...h.1...h1h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1..h1.....h|ph1.h1.....h1h.1...h.1...h1h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1..h1.....h|ph1.h1.....h1h.1...h.1....h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1.....h|ph1.h1..h1...h1h.1...h.1....h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1.....h|ph1.h1..h1..h.1.....h1h.1h1h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1.....h|ph1.h1..h1..h.1.....h1h.1..h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1..h1.....h|ph1.h1.....h.1.....h1h.1..h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```



```
{signal: [  
  {name: 'SDA', wave: 'hph1.h1..h1.....h|ph1.h1.....h.1.....h1h.1h1h'},  
  {name: 'SCL', wave: 'h.n.....h|.n.....h'},  
]}
```

Magyarázat

Szervezői információk:

a feladatot nem fogjuk beleszámítani az értékelésbe, ugyanis sajnos nem lehet másolni a válaszokat, ez pedig nagyban megnehezítette a megoldását. A feladatlap kitöltésére szánt időt ebben a fordulóban ebben a kategóriában nem vesszük figyelembe.

A 22,1875°C értéke binárisan **0001 0110 0011**, ez 2 byte formájában érkezik az eszközből, az adatlap 8-as táblázata szerint először a **0001 0110**, majd a **0011 0000**.

Az I2C kommunikáció felfutó élre mintavételez. Először lehúzza az SDA lábat (0), ezzel jelezve, hogy el szeretné kezdeni a kommunikációt. Kiküldi a megcímzendő eszköz 7 bites címét (100 1000), egy WRITE bitet (0), majd a slave jelez egy ACK bittel (0). Ezután rögtön kiküldi a regiszter címét, amiből olvasni akar (0000 0000), majd a slave ismét jelez egy ACK-kal (0), majd következik egy STOP bit (0). Ezután a master a START bit után újra kiküldi az eszköz címét (100 1000), ezúttal egy READ bit (1) követi, majd ACK (0). Innentől a master már csak "kiclockolja" a slave eszközből az olvasandó 2 byte-ot (0001 0110 és 0011 0000), közöttük szintén egy ACK bittel jelez (0), hogy sikeres volt az átvitel. A végén jelezvén, hogy végzett a kommunikációval ACK bit (0) helyett NACK-ot küld (1) a master, majd következik egy STOP bit (0).