

# BEÁGYAZOTT RENDSZEREK (C)

4. forduló



A kategória támogatója: Robert Bosch Kft.

Ismertető a feladatlaphoz

Közeleg az 5. forduló, figyelj az időpontokra!

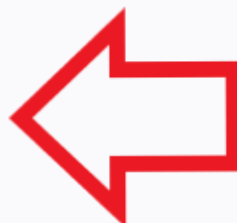
Használd a naptárat:

## KATEGÓRIÁIM

Összesen 10 kategóriára jelentkezted



**Versenynaptár letöltése**



Vagy figyeld kategóriánként az időpontokat (íme egy MINTA, hol találod):




## ● 3. FORDULÓ

A lezárult fordulókban eddig megszerzett pontok:

0/105 pont

**BOSCH**  
Invented for life

## Fordulók

Forduló	Pontok, időtartam	Feladat megoldható	Státusz
7. forduló	23 pont 25:00	 2023.11.28. 20:00-tól 2023.11.28. 20:35-ig	Feladatlap
6. forduló	23 pont 30:00	 2023.11.21. 20:00-tól 2023.11.21. 20:40-ig	Feladatlap
5. forduló	28 pont 25:00	 2023.11.14. 20:00-tól 2023.11.14. 20:35-ig	Feladatlap

Amennyiben olyan kategóriában játszol, ahol van csatolmány, de hibába ütközel a letöltésnél, ott valószínűleg a vírusirtó korlátoz, annak ideiglenes kikapcsolása megoldhatja a problémát. (Körülbelül minden 3000. letöltésnél fordul ez elő.)

Jó versenyzést kívánunk!

Mekk Mester úgy döntött, hogy készít egy saját távirányítót az előzőleg megépített audio lejátszójához. Ehhez a régi cuccai között talált egy PIC24EP512GP202 mikrovezérlőt, ami alkalmas lehet a feladatra. Segíts neki a távirányító elkészítésében!

Adatlapok:

PIC24EP512GP202: <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU16/ProductDocuments/DataSheets/dsPIC33EPXXXGP50X-dsPIC33EPXXXMC20X-50X-and-PIC24EPXXXGP-MC20X-Family-Data-Sheet-DS70000657J.pdf>

## 1. feladat 5 pont

Mekk Mester szeretné a mikrovezérlő órajelét (FCY) a belső oszcillátor segítségével 3275555Hz-re állítani.

Melyik kód állítja be helyesen az órajelet, ha feltételezzük, hogy megfelelően vannak beállítva a PIC konfigurációs regiszterei?

# Válasz

```
void DeviceInit() {
    PLLFBD = 158;
    CLKDIVbits.PLLPOST = 0;
    CLKDIVbits.PLLPRE = 7;
    CLKDIVbits.DOZEN = 0;

    __builtin_write_OSCCONH(0x01);
    __builtin_write_OSCCONL(OSCCON | 0x01);
    while (OSCCONbits.COSC != 0b001);
    while (OSCCONbits.LOCK != 1);
}
```

```
void DeviceInit() {
    PLLFBD = 158;
    CLKDIVbits.PLLPOST = 0;
    CLKDIVbits.PLLPRE = 7;
    CLKDIVbits.DOZEN = 0;

    __builtin_write_OSCCONH(0x00);
    __builtin_write_OSCCONL(OSCCON | 0x01);
    while (OSCCONbits.COSC != 0b001);
    while (OSCCONbits.LOCK != 1);
}
```

```
void DeviceInit() {
    PLLFBD = 160;
    CLKDIVbits.PLLPOST = 2;
    CLKDIVbits.PLLPRE = 9;
    CLKDIVbits.DOZEN = 0;

    __builtin_write_OSCCONH(0x01);
    __builtin_write_OSCCONL(OSCCON | 0x01);
    while (OSCCONbits.COSC != 0b001);
    while (OSCCONbits.LOCK != 1);
}
```

```
void DeviceInit() {
    PLLFBD = 158;
    CLKDIVbits.PLLPOST = 2;
    CLKDIVbits.PLLPRE = 7;
```

```
CLKDIVbits.DOZEN = 0;
```

```
__builtin_write_OSCCONH(0x01);  
__builtin_write_OSCCONL(OSCCON | 0x01);  
while (OSCCONbits.COSC != 0b001);  
while (OSCCONbits.LOCK != 1);  
}
```

## 2. feladat 5 pont

Mekk Mester szeretné az OC1 perifériát felhasználni a PWM jel generálására. Azt szeretné, hogy az OC1 órajel forrása az Peripheral Clock (Fp) legyen és a PWM kimenete a PORTB RB7-es lába legyen.

Válaszd ki a helyes megoldást!

### Válasz

☐

```
void OCInit() {  
    __builtin_write_OSCCONL(OSCCON & ~(1 << 6));  
    RPOR2bits.RP39R = 0b010000;  
    __builtin_write_OSCCONL(OSCCON | (1 << 6));  
    OC1CON1 = 0;  
    OC1CON2 = 0;  
    OC1CON2bits.SYNCSEL = 0x1F;  
    OC1CON1bits.OCTSEL = 0b111;  
}
```

☐

```
void OCInit() {  
    __builtin_write_OSCCONL(OSCCON & ~(1 << 6));  
    RPOR2bits.RP38R = 0b010000;  
    __builtin_write_OSCCONL(OSCCON | (1 << 6));  
    OC1CON1 = 0;  
    OC1CON2 = 0;  
    OC1CON2bits.SYNCSEL = 0x1F;  
    OC1CON1bits.OCTSEL = 0b111;  
}
```

☐

```
void OCInit() {  
    __builtin_write_OSCCONL(OSCCON & ~(1 << 6));
```

```

RPOR2bits.RP39R = 0b010000;
__builtin_write_OSCCONL(OSCCON | (1 << 6));
OC1CON1 = 0;
OC1CON2 = 0;
OC1CON2bits.SYNCSEL = 0x1F;
OC1CON1bits.OCTSEL = 0b100;
}

```

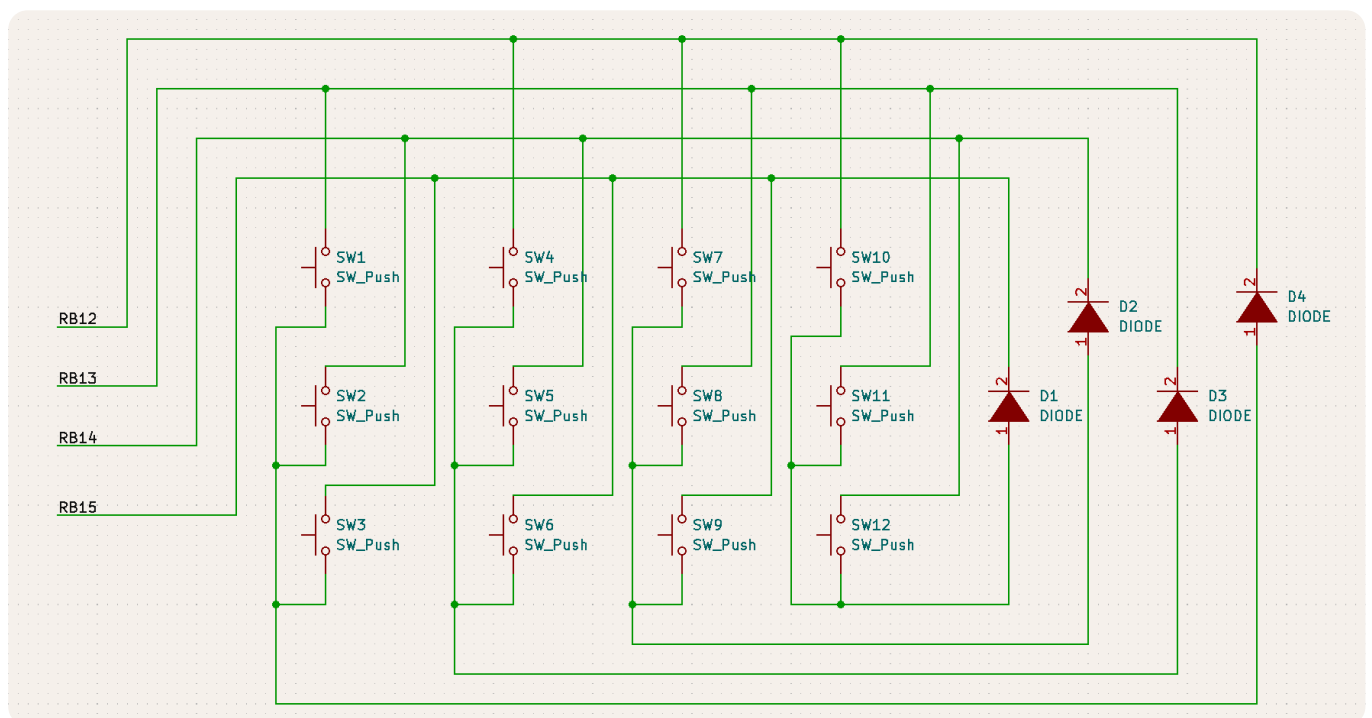
```

void OCInit() {
    __builtin_write_OSCCONL(OSCCON & ~(1 << 6));
    RPOR2bits.RP38R = 0b01000;
    __builtin_write_OSCCONL(OSCCON | (1 << 6));
    OC1CON1 = 0;
    OC1CON2 = 0;
    OC1CON2bits.SYNCSEL = 0x1F;
    OC1CON1bits.OCTSEL = 0b111;
}

```

### 3. feladat 15 pont

Mekk Mester szeretne az IO portokból minél kevesebbet elhasználni a gombokhoz, ezért az alábbi ábrán látható kapcsolást választotta. A rajzon a mikrovezérlőn felhasznált bemenetek is jelölve vannak.



Válaszd ki azokat a megoldásokat, amik megfelelően térnek vissza a megnyomott gomb sorszámával!

# Válaszok



```
uint8_t ScanButtons()
{
    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b1110000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b1110000000000000;
    LATBbits.LATB12 = 0;
    __delay_us(10);
    if(PORTBbits.RB13 == 0) return 1;
    if(PORTBbits.RB14 == 0) return 2;
    if(PORTBbits.RB15 == 0) return 3;

    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b1101000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b1101000000000000;
    LATBbits.LATB13 = 0;
    __delay_us(10);
    if(PORTBbits.RB12 == 0) return 4;
    if(PORTBbits.RB14 == 0) return 5;
    if(PORTBbits.RB15 == 0) return 6;

    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b1011000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b1011000000000000;
    LATBbits.LATB14 = 0;
    __delay_us(10);
    if(PORTBbits.RB12 == 0) return 7;
    if(PORTBbits.RB13 == 0) return 8;
    if(PORTBbits.RB15 == 0) return 9;

    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b0111000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b0111000000000000;
    LATBbits.LATB15 = 0;
    __delay_us(10);
    if(PORTBbits.RB12 == 0) return 10;
    if(PORTBbits.RB13 == 0) return 11;
    if(PORTBbits.RB14 == 0) return 12;

    return 0;
}
```



```
uint8_t ScanButtons() {
    LATB = LATB & 0b0000111111111111;
    TRISB = (TRISB & 0b0000111111111111) | 0b0111000000000000;
```

```

CNPDB = (CNPDB & 0b0000111111111111) | 0b0111000000000000;
LATBbits.LATB15 = 1;
__delay_us(10);
if(PORTBbits.RB12 == 1) return 3;
if(PORTBbits.RB13 == 1) return 6;
if(PORTBbits.RB14 == 1) return 9;

LATB = LATB & 0b0000111111111111;
TRISB = (TRISB & 0b0000111111111111) | 0b1011000000000000;
CNPDB = (CNPDB & 0b0000111111111111) | 0b1011000000000000;
LATBbits.LATB14 = 1;
__delay_us(10);
if(PORTBbits.RB12 == 1) return 2;
if(PORTBbits.RB13 == 1) return 5;
if(PORTBbits.RB15 == 1) return 12;

LATB = LATB & 0b0000111111111111;
TRISB = (TRISB & 0b0000111111111111) | 0b1101000000000000;
CNPDB = (CNPDB & 0b0000111111111111) | 0b1101000000000000;
LATBbits.LATB13 = 1;
__delay_us(10);
if(PORTBbits.RB12 == 1) return 1;
if(PORTBbits.RB14 == 1) return 8;
if(PORTBbits.RB15 == 1) return 11;

LATB = LATB & 0b0000111111111111;
TRISB = (TRISB & 0b0000111111111111) | 0b1110000000000000;
CNPDB = (CNPDB & 0b0000111111111111) | 0b1110000000000000;
LATBbits.LATB12 = 1;
__delay_us(10);
if(PORTBbits.RB13 == 1) return 4;
if(PORTBbits.RB14 == 1) return 7;
if(PORTBbits.RB15 == 1) return 10;
return 0;
}

```



```

uint8_t ScanButtons() {
    LATB = LATB & 0b0000111111111111;
    TRISB = (TRISB & 0b0000111111111111) | 0b0111000000000000;
    CNPDB = (CNPDB & 0b0000111111111111) | 0b0111000000000000;
    LATBbits.LATB15 = 1;
    __delay_us(10);
    if(PORTBbits.RB12 == 1) return 1;
    if(PORTBbits.RB13 == 1) return 2;
    if(PORTBbits.RB14 == 1) return 3;
}

```

```

LATB = LATB & 0b0000111111111111;
TRISB = (TRISB & 0b0000111111111111) | 0b1011000000000000;
CNPDB = (CNPDB & 0b0000111111111111) | 0b1011000000000000;
LATBbits.LATB14 = 1;
__delay_us(10);
if(PORTBbits.RB12 == 1) return 4;
if(PORTBbits.RB13 == 1) return 5;
if(PORTBbits.RB15 == 1) return 6;

LATB = LATB & 0b0000111111111111;
TRISB = (TRISB & 0b0000111111111111) | 0b1101000000000000;
CNPDB = (CNPDB & 0b0000111111111111) | 0b1101000000000000;
LATBbits.LATB13 = 1;
__delay_us(10);
if(PORTBbits.RB12 == 1) return 7;
if(PORTBbits.RB14 == 1) return 8;
if(PORTBbits.RB15 == 1) return 9;

LATB = LATB & 0b0000111111111111;
TRISB = (TRISB & 0b0000111111111111) | 0b1110000000000000;
CNPDB = (CNPDB & 0b0000111111111111) | 0b1110000000000000;
LATBbits.LATB12 = 1;
__delay_us(10);
if(PORTBbits.RB13 == 1) return 10;
if(PORTBbits.RB14 == 1) return 11;
if(PORTBbits.RB15 == 1) return 12;
return 0;
}

```



```

uint8_t ScanButtons() {
    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b1110000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b1110000000000000;
    LATBbits.LATB12 = 0;
    __delay_us(10);
    if(PORTBbits.RB13 == 0) return 3;
    if(PORTBbits.RB14 == 0) return 6;
    if(PORTBbits.RB15 == 0) return 9;

    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b1101000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b1101000000000000;
    LATBbits.LATB13 = 0;
    __delay_us(10);
    if(PORTBbits.RB12 == 0) return 2;
    if(PORTBbits.RB14 == 0) return 5;
}

```



```

    if(PORTBbits.RB15 == 0) return 12;

    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b1011000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b1011000000000000;
    LATBbits.LATB14 = 0;
    __delay_us(10);
    if(PORTBbits.RB12 == 0) return 1;
    if(PORTBbits.RB13 == 0) return 8;
    if(PORTBbits.RB15 == 0) return 11;

    LATB = (LATB & 0b0000111111111111) | 0b1111000000000000;
    TRISB = (TRISB & 0b0000111111111111) | 0b0111000000000000;
    CNPUB = (CNPUB & 0b0000111111111111) | 0b0111000000000000;
    LATBbits.LATB15 = 0;
    __delay_us(10);
    if(PORTBbits.RB12 == 0) return 4;
    if(PORTBbits.RB13 == 0) return 7;
    if(PORTBbits.RB14 == 0) return 10;

    return 0;
}

```

Megoldások beküldése