

Problema 1 (3 punti)

Su un PIC32MX3xx, sia data una linea digitale condivisa con un convertitore Analogico-Digitale; in che modo è possibile configurare la linea come **input-analogico**?

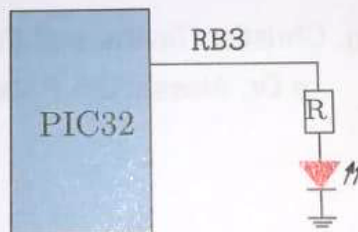
Barrare tutti i punti appropriati:

- ☐ Impostando a '1' il relativo bit del registro TRIS e a '1' il relativo bit del registro LAT
- ☐ Impostando a '0' il relativo bit del registro PORT e a '0' il relativo bit del registro ANSEL
- ☐ Impostando a '1' il relativo bit del registro PORT e a '1' il relativo bit del registro LAT
- ③ ☒ Impostando a '1' il relativo bit del registro TRIS e a '1' il relativo bit del registro ANSEL
- ☐ Impostando tutti i registri della porta a '0'

Problema 2 (3 punti)

Su un PIC32MX3xx, sia la porta RB3 collegata come in figura, quali delle seguenti impostazioni permettono l'accensione del LED?

Barrare tutti i punti appropriati:



- ☐ TRISbits.TRISB3 = 0; LATBbits.LATB3 = 0
- ② ☒ TRISbits.TRISB3 = 0; LATBbits.LATB3 = 1
- ☐ TRISB = 0xFAB7; LATBSET = 0xF8A8
- ☐ TRISbits.TRISB3 = 1; LATBbits.LATB3 = 0
- ☒ TRISBCLR = 0x0000; LATBSET = 0xFFFF
- ☐ TRISbits.TRISB3 = 1; LATBbits.LATB3 = 1
- ☐ TRISbits.TRISB3 = 1; PORTBbits.RB3 = 0

Problema 3 (3 punti)

Sia data una UART configurata per trasmettere e ricevere a 8 bit con parità dispari. Oltre gli 8 bit del messaggio (ogni carattere), quanti bit di "supporto alla trasmissione" vengono trasmessi almeno?

Barrare tutti i punti appropriati:

☐ 8

☐ 10

☐ 2

☒ 3

☐ 5

Problema 4 (3 punti)

Supponiamo che al pin RB5 sia connesso un pulsante verso massa che in un certo istante di tempo viene pigiato.

Sulla base del seguente *snippet code*, quale sarà il valore di count alla fine?

Barrare tutti i punti appropriati:

```
int i, j, count = 0;

TRISBbits.TRISB5 = 1;
for(i = 0; i < 5; i++)
{
    if (PORTbits.RB5 == 0)
        count++;
    for(j = 0; j < 1000; j++);
}
```

☐ 0

☐ 5

☒ Dipende dal tempo di pressione del pulsante e dall'istante in cui viene pigiato

☐ 1

Problema 5 (8 punti)

Sia dato un PIC32MX3xx con $SOSCIN = 8\text{ MHz}$, $PBCLK = 10\text{ MHz}$ e **TIMER1** (il cui schema è in figura) configurato nel modo seguente:

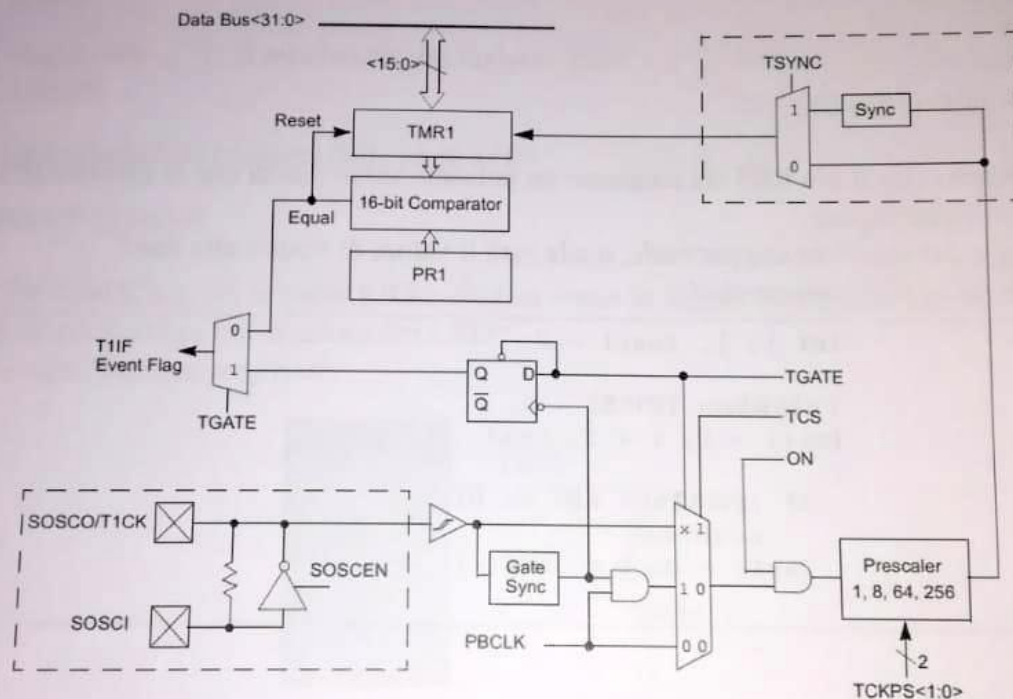
$TGATE = 0$

$TCS = 0$

$ON = 1$

$TCKPS = 0b01$

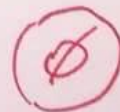
$TSYNC = 0$



Supposto che $PR1 = 0x0064$ e che il timer venga fatto partire con $TMR1 = 0$

1) Dopo quanto tempo verrà settato il primo EVENTFLAG? 44,44 ms

*mancano i calcoli
per poter valutare il
procedimento*



Problema 6 (3 punti)

Su un PIC32MX3xx, quale delle seguenti istruzioni permette di impostare le porte RB2 e RB3 come output e RB0 e RB1 come input?

Barrare tutti i punti appropriati:

✓ ☒ $\text{TRISB} = 0x03$

☐ $\text{TRISB} = (\text{TRISB} | 0x03) \& 0xF3$

☐ $\text{TRISB} = (\text{TRISB} \& 0x03) | 0xF3$

✓ ☒ $\text{TRISB} = 0xF3$

Problema 7 (3 punti)

Sia data una UART configurata a 8 bit senza parità e 1 stop-bit. Da una misura risulta che un carattere completo viene trasmesso, sulla linea TX, in circa 1.04 ms.

Quale tra questi è il baud-rate che si avvicina di più ?

☐ 19200 baud

☒ 7600 baud

☐ 9600 baud

☐ 57600 baud



Problema 8 (8 punti)

Viene fornito il seguente codice. Disegnare il *flowchart* corrispondente.

```
char flg_tx;
int count;

void main(void)
{
    gpioInit();           // general purpose I/O setting
                          // (all inputs and all outpus we need)
    Timer2Init();         // timer2 configuration
    UartInit();           // UART peripheral configuration and
                          // Receveing/Trasmitting functions

    while(getString() != "start"); // exec getString and wait
                                   // if cmd is not "start"
    Timer2InterruptEnable(); // Enable interrupt

    while(1)
    {
        if(flg_tx) // check if custom flag is set by ISR
        {
            putString("Hello"); // send a message
            flg_tx = 0;         // reset custom flag
        }

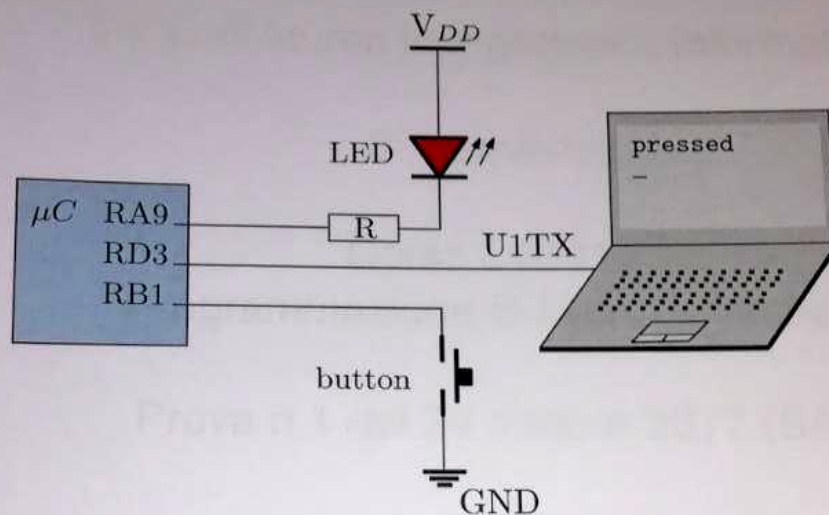
    } //end while
} // end main

void _ISR(_TIMER_2_VECTOR, ip12)
Timer2IntHandler(void) // Interrupt Service Routine
{
    count++;           // increment a variable
    if(count == 10)    // check if "count" reaches "10"
    {
        count = 0;     // reset "count"
        flg_tx = 1;    // set custom flag for endless while
    }

    IFS0bits.T2IF = 0; // Clear interrupt flag event
}
```

Problema 9 (16 punti) - Consegnare "NomeCognome.zip" con *source files* e *header files* sulla piattaforma

Scrivere un programma su μC in grado di leggere lo stato di un pulsante (*external interrupt*) che, quando premuto, accende un led ed invia un messaggio su UART di notifica "pressed". Lo schema generale del sistema è il seguente:



Mappa I/O:

- RA9 per pilotaggio LED
- RB1 per lettura pulsante (external interrupt 4)
- RD3 linea di trasmissione UART1

Funzioni:

La pressione del pulsante "scatena" un evento di interrupt (*external interrupt 4*). Il servizio all'interruzione accende il LED e invia la stringa "pressed" al terminale.

La comunicazione seriale asincrona deve essere configurata in modo da avere:

- 8-bit
- no parity
- 1 stop bit
- 19200 baud
- No HW control