

ARM – RTOS

Čítač

```
// Counter //
__task void Counter_4bit_A(void) {
    for(;;) {
        io_set(ledkyEx[0],1);
        delay_ms(100);
        io_set(ledkyEx[0],0);
        delay_ms(100);
    }
}

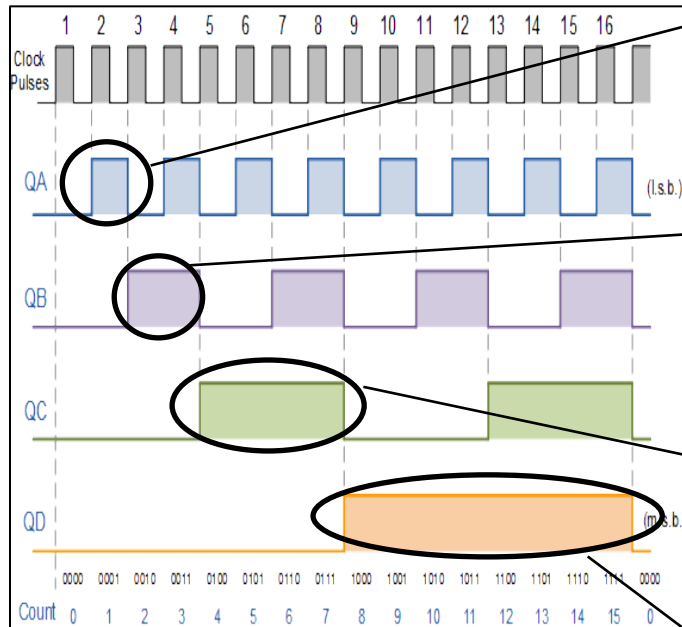
__task void Counter_4bit_B(void) {
    for(;;) {
        io_set(ledkyEx[1],1);
        delay_ms(200);
        io_set(ledkyEx[1],0);
        delay_ms(200);
    }
}

__task void Counter_4bit_C(void) {
    for(;;) {
        io_set(ledkyEx[2],1);
        delay_ms(400);
        io_set(ledkyEx[2],0);
        delay_ms(400);
    }
}

__task void Counter_4bit_D(void) {
    for(;;) {
        io_set(ledkyEx[3],1);
        delay_ms(800);
        io_set(ledkyEx[3],0);
        delay_ms(800);
    }
}
```

- rozdeleno do 4 tasku
- kazdy task se stara o jednu ledku

Časový diagram 4bit čítače



```
task void Counter_4bit_A(void) {
    for(;;) {
        io_set(ledkyEx[0],1);
        delay_ms(100);
        io_set(ledkyEx[0],0);
        delay_ms(100);
    }
}
```

```
task void Counter_4bit_B(void) {
    for(;;) {
        io_set(ledkyEx[1],1);
        delay_ms(200);
        io_set(ledkyEx[1],0);
        delay_ms(200);
    }
}
```

```
task void Counter_4bit_C(void) {
    for(;;) {
        io_set(ledkyEx[2],1);
        delay_ms(400);
        io_set(ledkyEx[2],0);
        delay_ms(400);
    }
}
```

```
task void Counter_4bit_D(void) {
    for(;;) {
        io_set(ledkyEx[3],1);
        delay_ms(800);
        io_set(ledkyEx[3],0);
        delay_ms(800);
    }
}
```

Pro každou LEDku je zodpovedný
jiny task

Na začatku zhasnutá, a zapne sa len
keď uplynul čas

Postupne nastavujeme väčšie
intervaly kedy sa LEDka rozsvieti a
zhasne