

# **Laborator 4 - PIT**

Coca Mihai  
Ioana Dragos

# Tabelă de Conținut

1. Concepte teoretice
2. Configurare
3. Exemplu practic

# Concepțe teoretice

- ❖ **Periodic Interrupt Timer**
- ❖ Acest periferic este reprezentat printr-un tablou de *timere* care pot configura programatic pentru generarea de întreruperi mascabile la perioade de timp specifice.
- ❖ **Timer** - numărător digital a cărui valoare configurată prin intermediul regiștrilor se modifică decremental la fiecare semnal de ceas
- ❖ **Semnal de ceas**
  - ❖ **Perioada semnalului de ceas** -  $\frac{1}{\text{Frecvență BUS CLOCK}}$
  - ❖ **Frecvență BUS CLOCK**  $\sim 10.485.760\text{Hz}$
- ❖ **Resetare timer** - Valoare configurată în registrul asociat \* perioada semnalului de ceas.

# Capabilități

- ***Periodic Interrupt Timer***
- Măsurarea unei perioade de timp
- Numărarea unor evenimente
- Generarea unor evenimente la perioade fixe de timp
- Generarea unor forme de undă

# Mod de funcționare

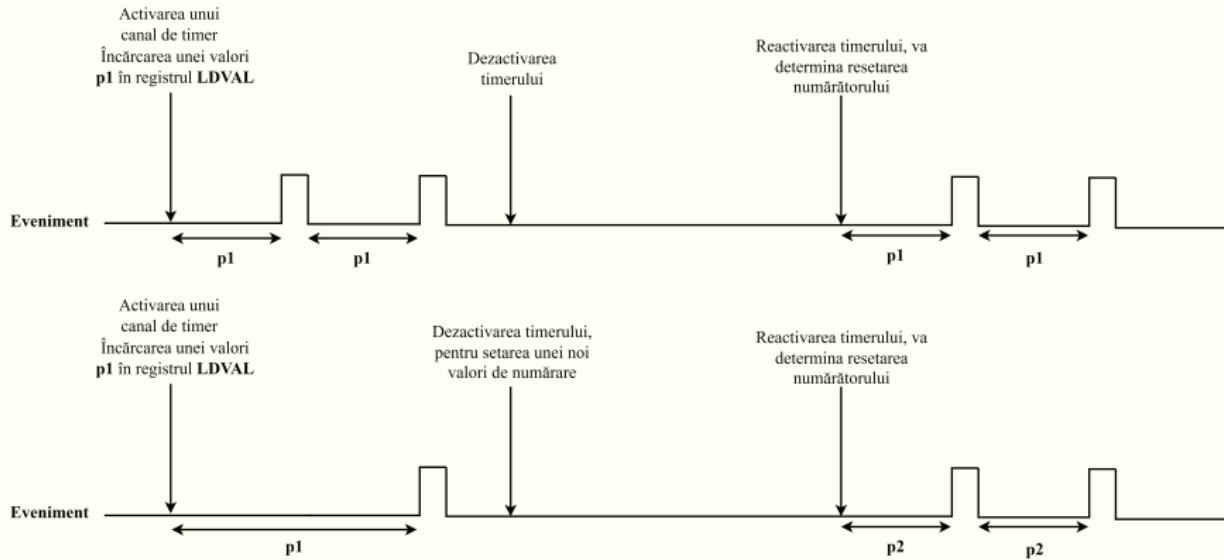


Figura: Generarea semnalelor prin intermediul perifericului PIT

# Noțiuni introductive

## ► **PIT FRDM-KL25Z**

- Existența a două canale folosite pentru timere  
**(Channel 0 și Channel 1)**
- Fiecare timer are asociată o valoare de numărare independentă.
- Accesul la un canal se face cu ajutorul sintaxei  
**PIT->CHANNEL[x]**
- Perifericul PIT nu interfațează cu pini externi.

# Configurare

# Setarea regiștrilor

- ❖ Atunci când dorim să utilizăm un timer, pornirea acestuia, setarea și citirea valorii numărătorului se pot face într-o manieră programatică.
- ❖ Regiștrii care pun la dispoziție aceste setări:
  - ❖ **PIT\_MCR** - Module Control Register
  - ❖ **PIT\_LDVALx** - Timer Load Value Register
  - ❖ **PIT\_CVALx** - Current Timer Value Register
  - ❖ **PIT\_TCTRLx** - Timer Control Register
  - ❖ **PIT\_TFLGx** - Timer Flag Register

# Setarea regiștrilor

- Activarea semnalului de ceas pentru a putea configura modulele folosite
  - **SIM\_SCGC6** - activarea modulului periferic **PIT**
- **PIT\_MCR** - utilizarea semnalului de ceas pentru timerele puse la dispoziție de PIT (**MDIS**) și configurarea comportamentului timerelor la debug (**FRZ**)
- **PIT\_LDVAL** - configurarea valorii numărătorului se face pe baza formulei

$$\text{Load Value} = \text{Nr. sec.} * \text{BUS CLOCK Freq.} - 1$$

- **PIT\_TCTRL** - permite activarea/dezactivarea timerului(**TEN**) de pe canalul asociat și a întreruperilor generate(**TIE**).

# Setarea regiștrilor

- **PIT\_TFLG** - verificarea câmpului **TIF** pentru apariția unei întreruperi pe un anumit canal
- **PIT\_LTMR** - PIT Lifetime Timer Register permite crearea unui timer pe 64 de biți (separat pe doi registri) și prin utilizarea flag-ului **CHN** din cadrul registrului **PIT\_TCTRL** care permite înlănuirea timerelor.
  - **PIT\_LTMR64H** - HIGH, corespunde cu  
**PIT->CHANNEL[1].CVAL**
  - **PIT\_LTMR64L** - LOW, corespunde cu  
**PIT->CHANNEL[0].CVAL**
  - Pentru a fi decrementat canalul **n** este nevoie de resetarea numărătorului de pe canalul **n-1**.

# Exemplu practic

# Experiment

- Dorim activarea ambelor canale PIT, pentru a folosi 2 numărătoare în paralel, fiecare cu o valoare diferită.
- **Obiectiv:** La interval de o secundă, să fie actualizat un cronometru afișat prin intermediul perifericului **UART** la terminal. La interval de 10 secunde să aprindem un LED cu ajutorul perifericului **GPIO**.
- Încărcarea valorilor corespunzătoare în registrul **LDVAL** pe baza formulei prezentate în laborator.
- Avem nevoie de activarea interuperilor pentru perifericul **PIT** și suprascrierea handler-ului de intreruperi **PIT\_IRQHandler** pentru a avea o logică separată pentru cele două canale.
- Realizarea conexiunii cu **GND**

# Rezultat

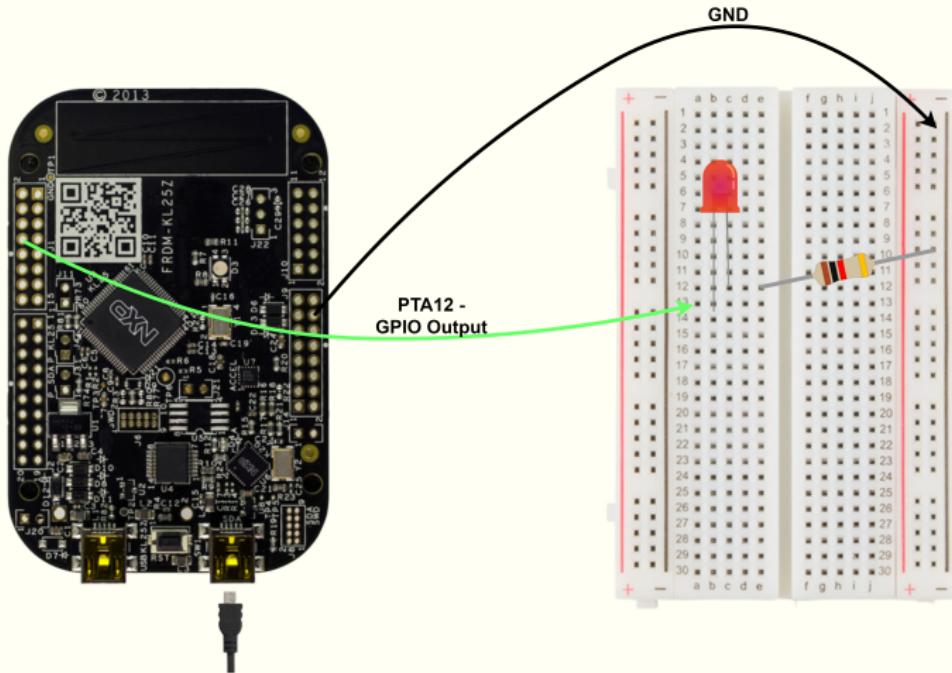


Figura: Circuit electric final