

# Esercitazione 06: riconoscitore di sequenze

Corso di  
Progetto e Sviluppo di  
Sistemi in Tempo Reale

Marcello Cinque  
Daniele Ottaviano

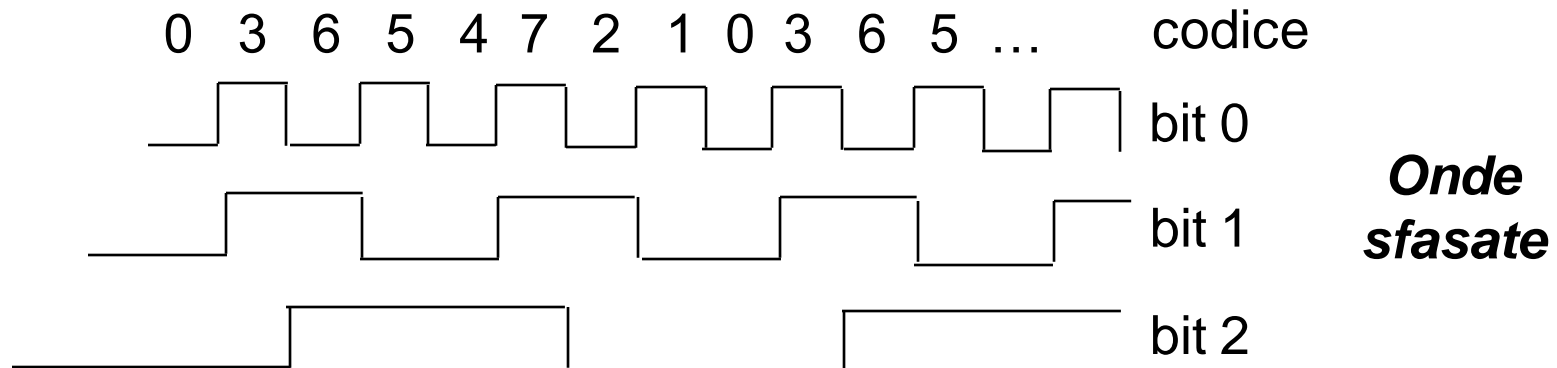
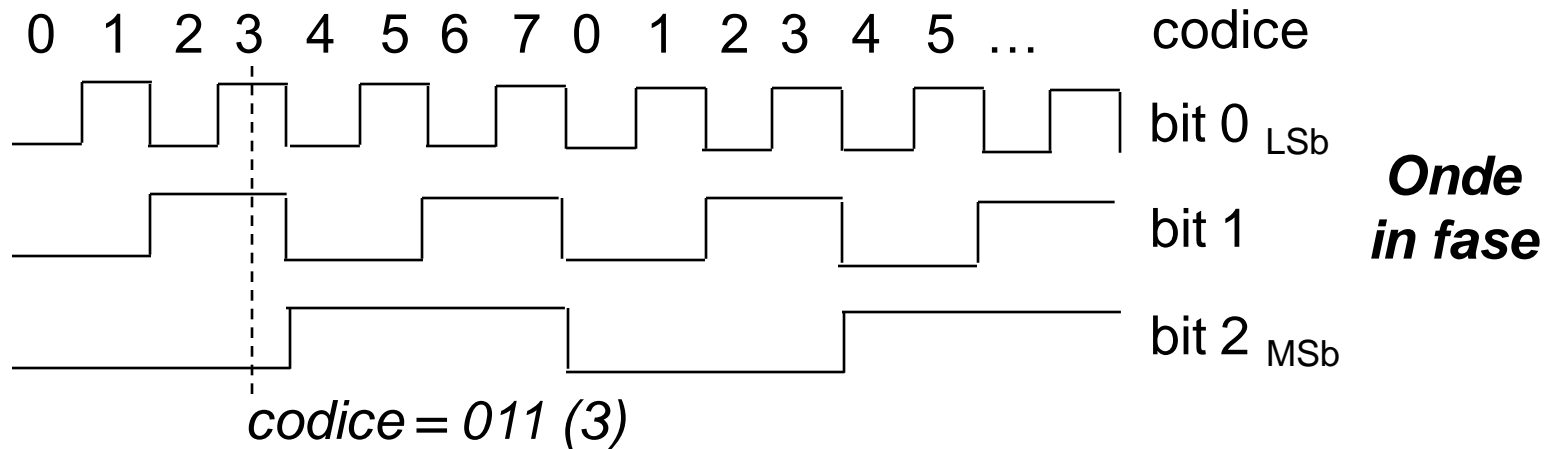
# Traccia

- Si realizzino usando le primitive RT-POSIX
  - 3 task hard real time kernel-level **generatori** di «onde quadre» per la generazione periodica di sequenze numeriche codificate su 3 bit
  - un task hard real time kernel-level **riconoscitore** della sequenza 0,3,6,5 a partire dal codice prodotto dalle onde quadre
  - un task buddy «scope» user-level per la stampa a video dei risultati di riconoscimento
- I task possono far parte dello stesso sorgente e devono essere schedulati con **rate monotonic**

# Task generatori

- Ognuno dei 3 task generatori
  - Genera la propria onda quadra commutando una variabile binaria (da 0 a 1 e viceversa) ogni suo periodo di esecuzione (semiperiodo dell'onda)
  - è caratterizzato da un suo semiperiodo (minimo 10ms) e da una fase di partenza (multipla di 10ms)
- I 3 task generano nel tempo delle sequenze numeriche, considerando le cifre codificate su 3 bit prodotte nel tempo durante l'esecuzione

# Sequenze numeriche da onde quadre binarie



Il codice prodotto dipende dal periodo delle onde e dalla loro fase

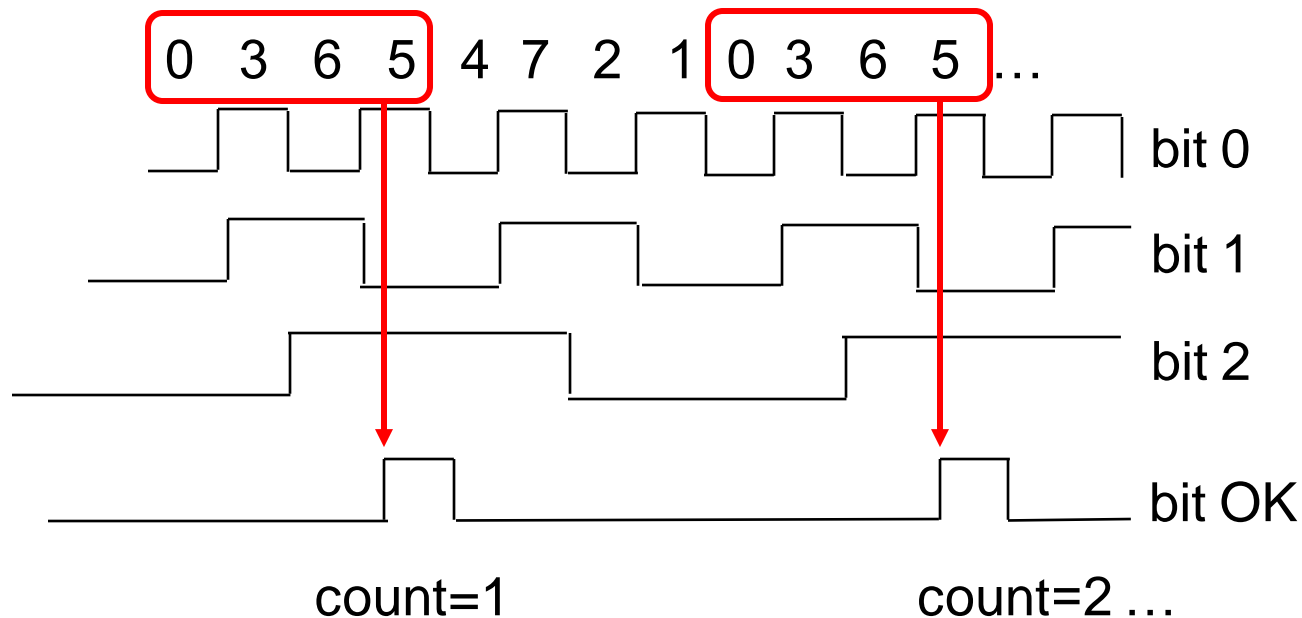
# Configurazione dei task generatori

- I task generatori possono essere configurati specificando per ciascuno il proprio semiperiodo in ms (min 10, max 200) e la propria fase (in multipli di 10ms)
- I parametri possono opzionalmente essere passati in input, oppure possono essere cablati nel codice
- Detto `start` il tempo base di partenza, la fase  $x$  per il task  $j$  ha il significato di far partire il task  $j$  al tempo `start + min_per*x` con `min_per` costante pari al minimo semiperiodo (10 ms)

# Task riconoscitore di sequenza

- Legge periodicamente (ogni 10ms) il valore corrente del codice prodotto dai task generatori e va alla ricerca della sequenza 0, 3, 6, 5
  - Per riconoscere la sequenza, deve possedere uno «stato» delle cifre già riconosciute e della successiva cifra attesa:
    - Per es, ha già riconosciuto 0 e 3 nelle istanze precedenti, attende 6 nell'istanza corrente
- Se riconosce la sequenza, aggiorna i campi di una struttura dati condivisa contenente:
  - Un campo intero «OK» messo a 1 appena la sequenza è riconosciuta, e azzerato successivamente
  - Un contatore «count» del numero di sequenze riconosciute

# Commutazione del bit OK



Il task riconoscitore non deve essere necessariamente in fase con gli altri

# Task buddy

- Il task buddy si limita a stampare continuamente a video, ogni 10ms, i valori OK e count letti da un area di memoria condivisa



# Variabili Condivise

- Le variabili condivise tra i generatori e il task buddy e tra generatori e il ricevitore vanno opportunamente protette tramite dei Mutex

# Varianti e opzioni

- Opzionalmente, si possono realizzare delle varianti, come ad es:
  - Utilizzare scanf/printf per inserire a runtime il periodo e la fase dei generatori.
  - Rendere configurabile anche la sequenza da riconoscere, passandola come parametro
  - Estendere il codice su più bit (aggiungendo uno o più task generatori)

# Consegna

- Da consegnare in settimana su Teams
- Consegna in ritardo possibile entro Martedì 25
- La prova non verrà valutata