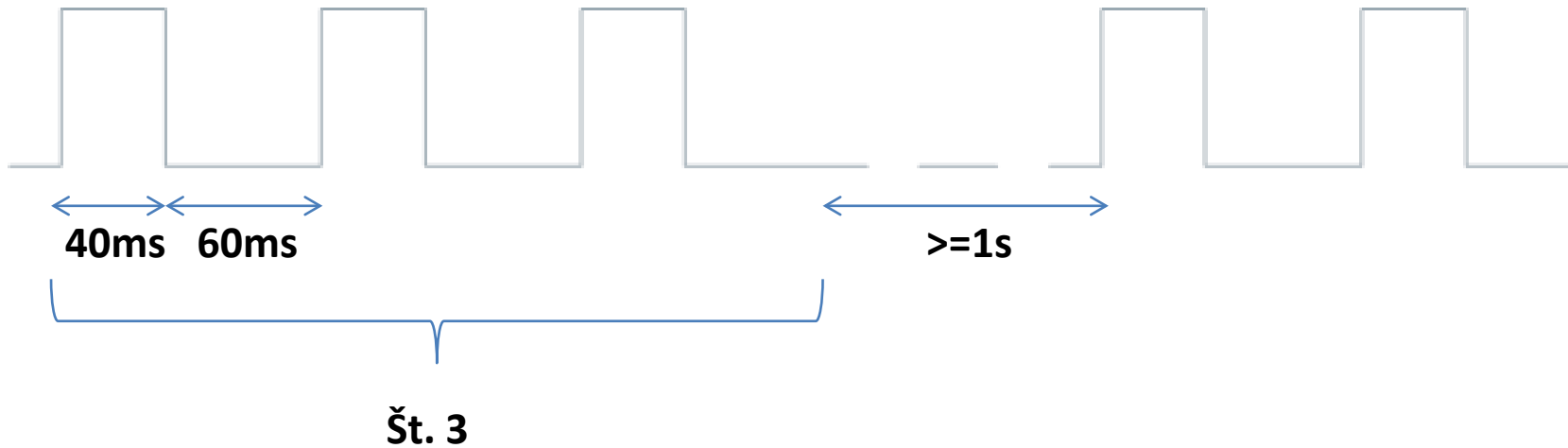


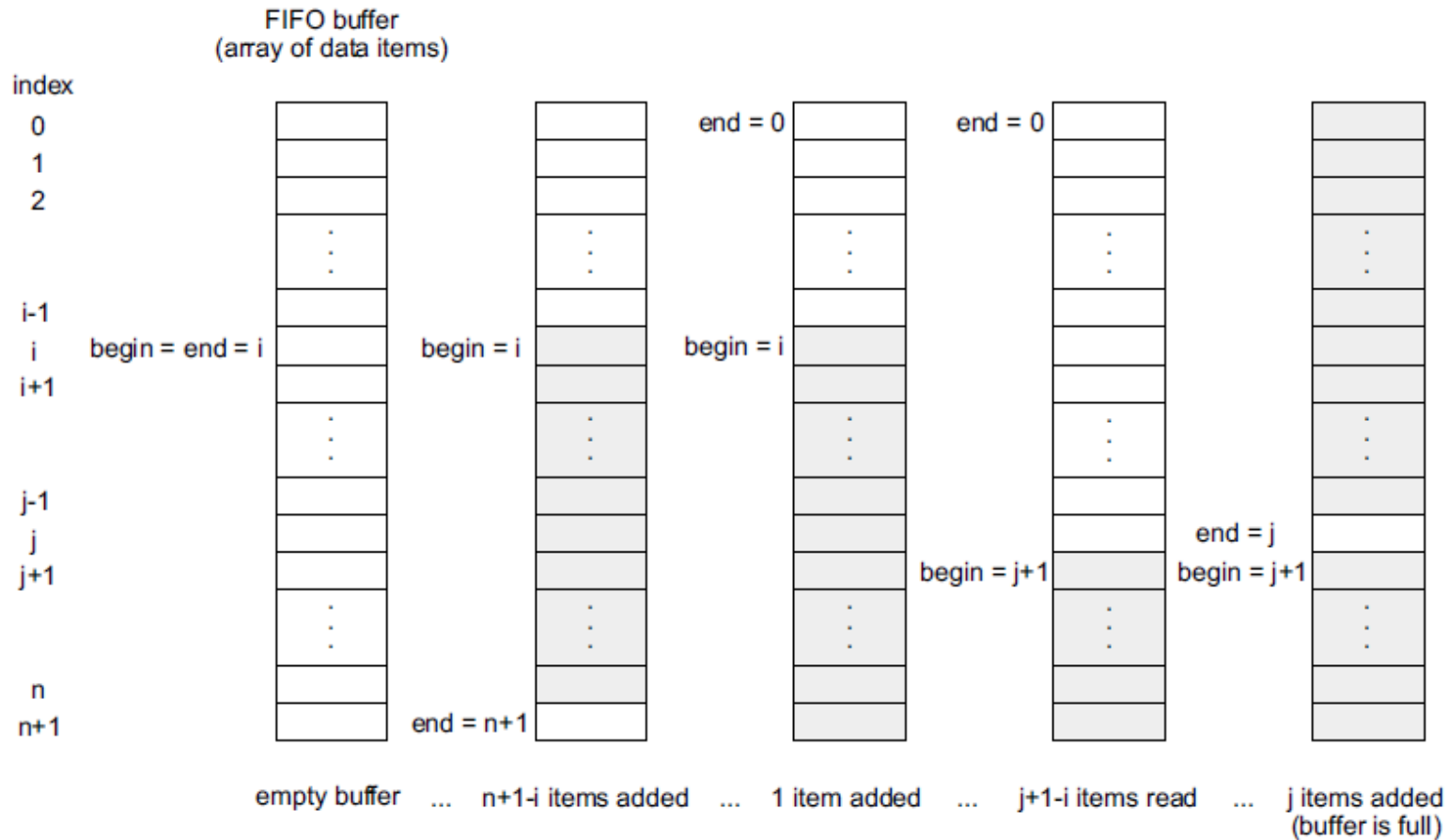
Vaja 7

**Ciklični medpomnilnik, gonilnik za
tipke, pulzno izbiranje telefonske
številke**

- 1. Napišite funkcije za delo s cikličnimi medpomnilniki**
- 2. Napišite gonilnik za tipkovnico**
- 3. Napišite program, ki bo omogočal “pulzno izbiranje telefonske številke” s tipkovnico. Številke, ki čakajo na generiranje vlaka impulzov, naj bodo prikazane na prikazovalniku LCD.**



Ciklični medpomnilnik – tipa FIFO (First In First Out):



Slika 3.8: Ciklični medpomnilnik FIFO

Ciklični medpomnilnik – tipa FIFO (First In First Out):

V projekt vključimo dve novi datoteki (rtos_pipe.h in rtos_pipe.c)

V rtos_pipe.h vpišemo deklaracijo strukture medpomnilnika in prototipe funkcij za delo z njimi (splošne funkcije za vse medpomnilnike, npr. za I2C, SPI, tipke, ...):

```
struct rtos_pipe{
    unsigned int begin;    // bralni index (začetek bufferja, najstarejši podatek)
    unsigned int end;      // pisalni index (konec bufferja, prvo prosto mesto)
    unsigned int size;     // velikost bufferja
    char *buffer;          // naslov, kjer se nahajajo podatki
};                        // definicijo strukture dodamo v rtos_tasks.h

extern int rtos_pipe_read(struct rtos_pipe *pipe, char *data, unsigned int num_bytes);
extern int rtos_pipe_write(struct rtos_pipe *pipe, char *data, unsigned int num_bytes);
```

Parametri:

pipe: naslov strukture, ki vsebuje buffer
data: naslov kjer se nahajajo podatki za pisanje / naslov kamor bomo prebrali podatke iz bufferja
num_bytes: št. byte-ov, ki jih želimo zapisati / prebrati

Ciklični medpomnilnik – tipa FIFO (First In First Out):

V datoteko `rtos_pipe.c` vpišemo kodo obeh funkcij:

```
int rtos_pipe_write(struct rtos_pipe *pipe, char *data, unsigned int num_bytes){  
    ...  
}  
int rtos_pipe_read(struct rtos_pipe *pipe, char *data, unsigned int num_bytes){  
    ...  
}
```

1. preverijo, če je pisanje / branje sploh možno (poln, prazen buffer)
2. vpišejo / preberejo zahtevano število byte-ov ter ustrezno popravijo indexe
3. vrnejo št. uspešno zapisanih / prebranih byte-ov.

Če sta v medpomnilniku **begin** in **end** indexa enaka => buffer je prazen in iz njega ne moremo brati => funkcija `rtos_pipe_read()` vrne 0

Če bi povečali (ciklično) **end** index za 1 in bi ta tako postal enak **begin** => buffer je poln in v njega ne moremo pisati => izgubljen podatek => funkcija `rtos_pipe_write()` vrne 0

En byte v medpomnilniku mora vedno biti prost!

Ciklični medpomnilnik – tipa FIFO (First In First Out):

Medpomnilnik za tipkovnico:

V `rtos_tasks.h` določimo velikost medpomnilnika, dodamo deklaracijo za `keys_pipe`

```
#define KEYS_BUFF_LEN 11  
extern struct rtos_pipe keys_pipe;
```

V `rtos_tasks.c` rezerviramo pomnilnik in ustvarimo ter inicializiramo strukturo `keys_pipe`

```
char keys_buff[KEYS_BUFF_LEN];  
struct rtos_pipe keys_pipe = {0, 0, KEYS_BUFF_LEN, keys_buff};
```

Primer uporabe:

1. ob pritisku tipke T0 moramo v medpomnilnik vpisati znak '0'

```
char c='0';  
int n;  
n = rtos_pipe_write( &keys_pipe, &c, 1); //zapiše znak shranjen v spr. c v prvo prosto  
mesto v medpomnilnik
```

2. Ko želimo prebrati 1 znak s tipkovnice (npr. v opravilu `dial()`):

```
n = rtos_pipe_read( &keys_pipe, &c, 1); //prebere 1 byte iz keys_pipe in ga shrani v c
```

Pulzno izbiranje števil:

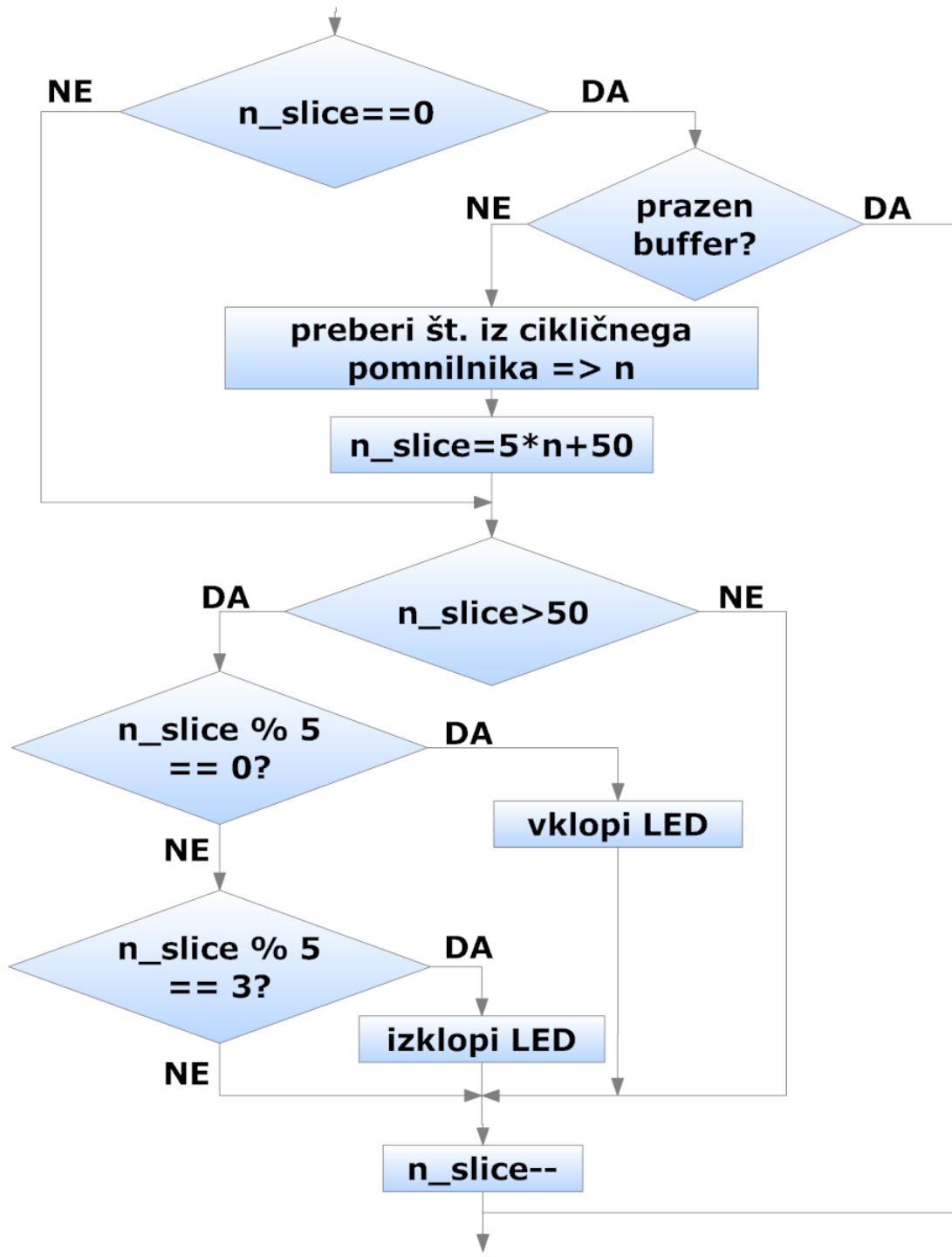
Opravila:

- 1. keys_driver()** : vsaj 10x v sekundi želimo preveriti stanje tipk in morebiten znak vpisati v medpomnilnik keys_pipe
- 2. keys2lcd()** : prebere vsebino keys_pipe in jo oblikuje za lcd_string (prebrane znake pusti v bufferju)
- 3. lcd_driver()** : izpiše vsebino lcd_string na LCD
- 4. dial()** : prebere 1 znak iz keys_pipe (in ga odstrani) in poskrbi za vlak pulzov na LED0

Q: koliko mora biti dolga časovna rezina?

Q: kako zagotoviti, da bodo opravila dovolj pogosto na vrsti?

dial():



Še generiramo pulz?
=> Ne beremo nove št.

Prazen buffer => konec

Toliko rezin traja generiranje
pulzov za trenutno št. (velja za
20ms rezine)

Generiranje pulzov, ali
čakanje na nasl. vlak

LED prižigamo vsakih 5
rezin

LED ugasnemo 2 rezini po
tem ko smo jih prižgali