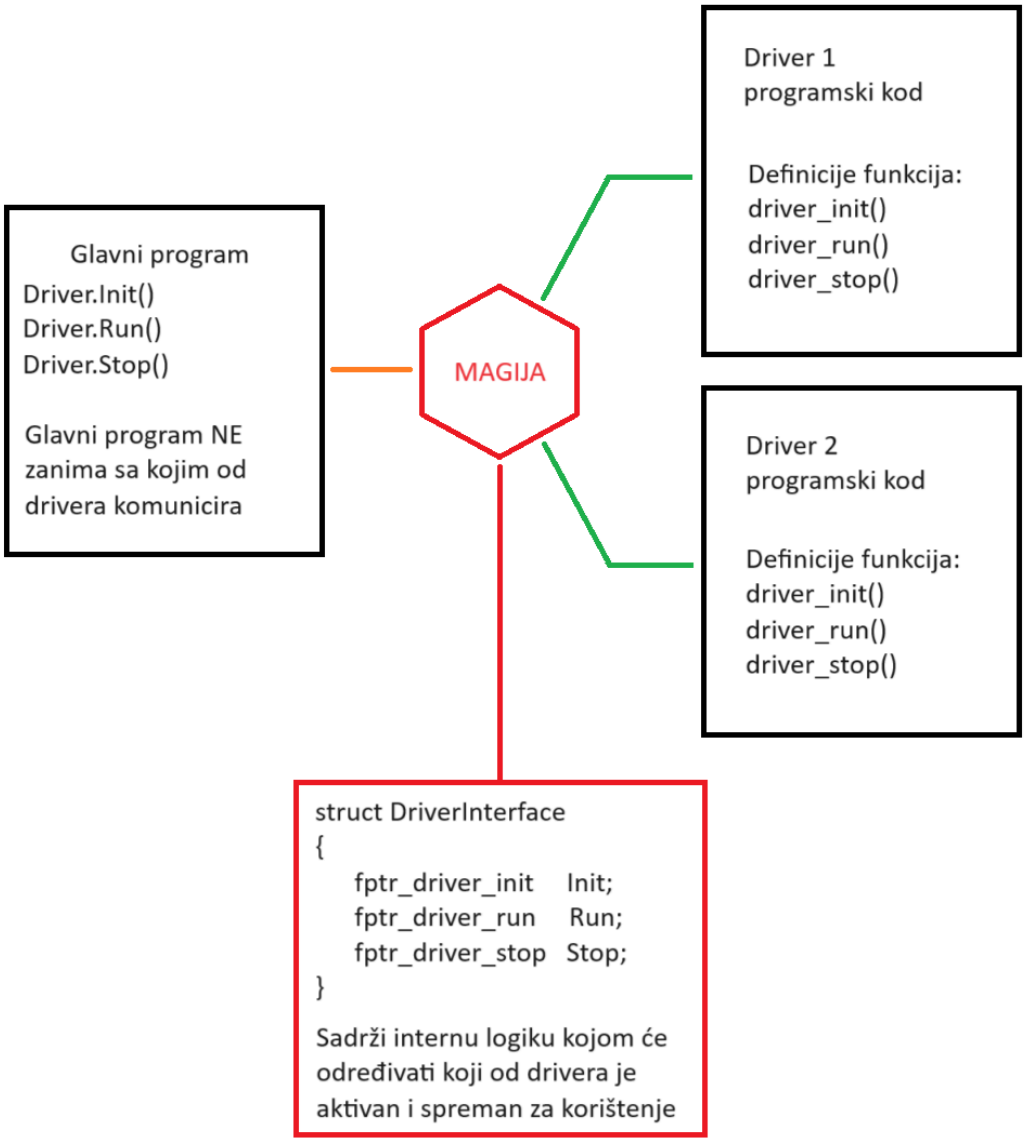
Studentske prakse - Interface i I2C

Interface

- Design pattern u C programskom jeziku
- Možemo ih smatrati kao krnje klase (nije bas najtočnija terminologija)
- Koristimo ih kako bi mogli imati vise definicija jedne ili vise funkcija odnosno metoda (polimorfizam)
- C programski jezik NIJE OOP, stoga način na koji se ostvaruje polimorfizam nije “lijep” kao u nekim višim programskim jezicima
- U C programskom jeziku interface nam služi kao spojnica 2 ili vise dijelova koda
- Pregledan i strukturiran programski kod
- Modularnost programskog koda



Kako kreirati interface u C programskom jeziku:

- Trebamo poznavati rad sa strukturama i funkcijskim pointerima !
- Struktura predstavlja krnju klasu koja sadrži funkcijske pointere (može sadržavati i ostale tipove varijabli)
- Funkcijski pointeri služe za odabir metoda koje će interface posjedovati
- Postavljamo si pitanje što mi zapravo želimo želimo modularizirati (spojiti)
- Kreiramo "footprint" za svaku od metoda koju znamo da će interface sadržavati, odnosno kreiramo tipove funkcijskih pointera:
  typedef <return_type> (<name_of_method_type>)(<method_arguments>)*

- Kreiramo "footprint" za interface, odnosno kreiramo tip interface-a:

```
typedef struct <name_of_interface_type>
{
   <name_of_method_1_type>  <method_1_name>;
   <name_of_method_2_type>  <method_2_name>;
   ⋮
  <name_of_method_n_type>  <method_n_name>;
} <name_of_interface_type>;
```

Primjer:

Definiranje tipova funkcijskih pointera:

```
typedef   uint32_t   (*fnptr_driver_init_t)    (uint8_t *p_buffer, uint16_t len);
typedef   uint32_t   (*fnptr_driver_run_t)     (void);
typedef   uint32_t   (*fnptr_driver_stop_t)    (void);
```

Definiranje tipa interface-a:

```
typedef struct   driver_if_t
{
   fnptr_driver_init_t      Init;
   fnptr_driver_run_t       Run;
   fnptr_driver_stop_t      Stop;
}   driver_if_t;
```

Kreiranje MyDriverIF interface varijable:

```
driver_if_t    MyDriverIF;
```

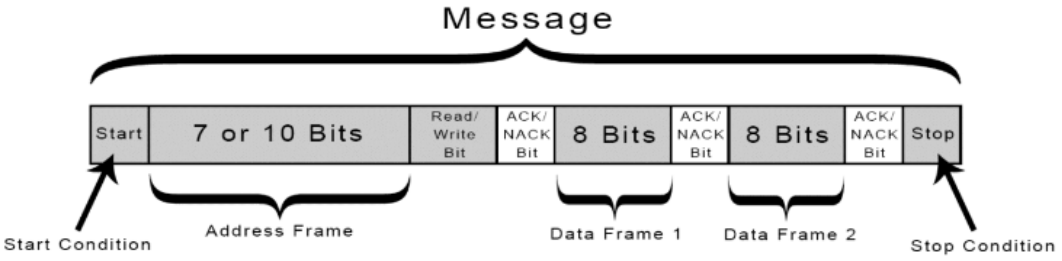
Interface dio koda sadrži i logiku kojom će odrediti da funkcijski pointeri interface-a *MyDriverIF* pokazuju na funkcije(metode) koje želimo da interface sadrži. To bi značilo da interface dio koda ima informacije o oba drivera i njihovim implementacijama ali ih ne prosljeđuje dalje u ostatak koda. Ostatak koda nije nužan znati sto se zapravo događa u pozadini, jedino što njega interesira je rad sa driverom preko interface-a.

I2C komunikacijski protokol

- I2C - Inter Integrated Circuit
- Mogucnost posjedovanja vise “slave” i/ili “master” uređaja (jedan master vise slave-ova, vise mastera jedan slave)
- Koristi dvije linije:
    - SCL - Serial Clock - linija kojom se prenose pulsevi clock-a
    - SDA - Serial Data - linija preko koje se šalju podaci (oba smjera)
- Riječ je o serijskom komunikacijskom protokolu što znači da se podaci prenose bit po bit
- Sinkronizirani komunikacijski protokol - izlaz podataka je sinkroniziran sa clock-om periferije. Clock uvijek kontrolira master uređaj

Podaci koji se šalju između dva uređaja dolaze u paketima koji se mogu podijeliti na dijelove (frame-ove):

- **Start** - prijelaz linije sa High na Low (prvo SDA pa zatim SCL)
- **Address** - 7 ili 10 bitna adresa kojom master uređaj kontaktira slave uređaj
- **Read/Write** - bit koji označuje da li master šalje podatke prema slave-u ili master prima podatke od slave-a
- **ACK/NACK** - bit koji označuje ako je dio paketa (frame) uspješno poslan
- **Data** - podaci koji se šalju (byte to byte)
- **Stop** - prijelaz linije sa Low na High (prvo SCL pa zatim SDA)



Master uređaj šalje adresu slave-a s kojim želi komunicirati svakom slave-u spojenom na njega. Svaki slave zatim uspoređuje adresu koju je poslao glavni uređaj sa svojom vlastitom adresom. Ako se adresa podudara, šalje niskonaponski ACK bit natrag masteru. Ako se adresa ne podudara, podređeni uređaj ne radi ništa i SDA linija ostaje visoka.

Prednosti:

- koristi mali broj žica (dvije), samim time i malo HW
- može imati vise master-a slave-ova
- dobiva potvrdu o slanju svakog data frame-a

Mane:

- velicina data frame-a je ograničena na 1B
- sporiji od SPI protokola