

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2894

**Ostvarenje upravljačkog programa za  
RTL8139 mrežnu karticu**

Juraj Selanec

Zagreb, lipanj 2013.



## Sadržaj

Uvod .....	4
1. Mrežna kartica RTL8139.....	5
1.1. Mehanizam slanja paketa.....	8
1.2. Mehanizam primanja paketa.....	12
2. Ostvarenje upravljačkog programa.....	16
2.1. Ostvarenje funkcije <code>init</code> .....	17
2.2. Ostvarenje funkcije <code>send</code> .....	18
2.3. Ostvarenje funkcije <code>recv</code> .....	19
Zaključak .....	20
Literatura .....	21
Sažetak.....	22

# Uvod

Komunikacija računalnih sustava u današnjem društvu je sveprisutna te čini glavni alat u globalizaciji svijeta. Internet, bez kojeg je danas teško zamisliv poslovni i privatni život skoro svakog pojedinca, se upravo oslanja na komunikaciju računalnih sustava.

Mrežne kartice su ključan dio u komunikaciji računala i računalnih sustava te danas čine sastavni dio svakog računala. Uz samu karticu dolazi i upravljački program koji je zadužen za pravilan rad mrežne kartice u određenom operacijskom sustavu. Upravljački program sadrži informacije kako se sklop kontrolira i kako mu pristupiti, te predstavlja vezu između operacijskog sustava i sklopovlja.

Zadatak ovog završnog rada je ostvarenje jednostavnog upravljačkog programa za mrežnu karticu RTL8139, te njegova ugradnja u pojednostavljeni operacijski sustav Benu.

U poglavlju *Mrežna kartica RTL8139* ćemo se upoznati sa samim sklopom, njenim proizvođačem, njenim glavnim prednostima, te gdje se danas najviše koristi. Također dobit ćemo detaljan uvid u tehničke specifikacije sklopa, način rada kartice i mehanizme slanja odnosno primanja paketa.

U poglavlju *Ostvarenje upravljačkog programa* ćemo se upoznati sa funkcijama koje se koriste u upravljačkom programu kroz njihov pseudokod i njihovu međusobnu komunikaciju.

# 1. Mrežna kartica RTL8139

Mrežnu karticu RTL8139 proizvodi tvornica Realtek Semiconductor Corp [2] sa sjedištem u Hsinchu, Taiwan. Tvornica se bavi dizajniranjem i prodajom računalnog sklopovlja, najviše na području mrežne komunikacije (mrežne kartice, usmjereni (eng. router), komutatori (eng. switch)) i multimedije (LCD kontroleri, čitači kartica). Sama kartica je dobila „Best Component“ i „Best of Show“ nagradu na Computex-u 97' (*Taipei International Information Technology Show*)



Slika 1.1 Primjerak mrežne kartice sa sklopom RTL813

Iako izvorni sklop nema obilježja novijih Gigabit uređaja, često se koristi upravo zbog svoje jednostavnosti i niske cijene. Također jedna od prednosti je i besplatna dokumentacija o sklopu koju pruža sam proizvođač te brojni korisni materijali koji se nalaze na Webu. Vrlo često se simulira u virtualnim okruženjima upravo zbog svoje jednostavnosti i podrške kod brojnih operacijskih sustava. Također, njegov jednostavan upravljački program ga čini čestim izborom kod programera početnika te ljudi koji se programiranjem operacijskih sustava bave iz hobija.

RTL8139 sklop je visoko integrirani i ekonomičan *single chip Fast Ethernet* kontroler koji pruža 32-bitne performanse, te punu usklađenost sa IEEE 802.3u 100Base-T specifikacijama i IEEE 802.3x Full Duplex Flow Control. Sklop je opremljen sa PCI i Boot ROM sučeljem, koje je opremljeno sa EPROM (*Erasable Programmable Read – only Memory*) i flash memorijom. To omogućuje sklopu da pojednostavi upravljačke procedure i optimizira mrežnu sigurnost. Također podržava ACPI (*Advanced Configuration Power managment Interface*), OSPM (*Operating System Directed Power Managment*) kako bi se postiglo što efikasnije upravljanje energijom [1].

RTL8139 ima 86 registara preko kojih se sklop konfigurira za sve svoje mogućnosti. U našoj izvedbi jednostavnog upravljačkog programa koji omogućava jednostavnu komunikaciju, korišteni su registri prikazani tablicom Tablica 1.1.

Tablica 1.1 Registri korišteni u upravljačkom programu [3]

Ofset	Oznaka	Ime registra
0037h	CR	<i>Command Register</i>
0052h	CONFIG1	<i>Configuration Register</i>
0010h-0013h	TSD0	<i>Transmit Status Register of Descriptor 0</i>
0014h-0017h	TSD1	<i>Transmit Status Register of Descriptor 1</i>
0018h-001Bh	TSD2	<i>Transmit Status Register of Descriptor 2</i>
001Ch-001Fh	TSD3	<i>Transmit Status Register of Descriptor 3</i>
0020h-0023h	TSAD0	<i>TransmitStart Address of Descriptor 0</i>
0024h-0027h	TSAD1	<i>Transmit Start Address of Descriptor 1</i>
0028h-002Bh	TSAD2	<i>Transmit Start Address of Descriptor2</i>
002Ch-002Fh	TSAD3	<i>Transmit Start Address of Descriptor3</i>
003Ch-003Dh	IMR	<i>Interrupt Mask Register</i>
003Eh-003Fh	ISR	<i>Interrupt Status Register</i>
0040h-0043h	TCR	<i>Transmit (Tx) Configuration Register</i>
0030h-0033h	RBSTART	<i>Receive (Rx) Buffer Start Address Register</i>
0038h-0039h	CAPR	<i>Current Address of Packet Read Register</i>
0044h-0047h	RCR	<i>Receive (Rx) Configuration Register Register</i>
005Ch-005Dh	MULINT	<i>Multiple Interrupt Select Register</i>
004Ch-004Fh	MPC	<i>Missed Packet Counter Register</i>

Mehanizmi za slanje i primanje paketa se razlikuju te su u nastavku zasebno objašnjeni.

## 1.1. Mehanizam slanja paketa

Prilikom slanja paketa koriste se 4 opisnika određena svojim ulazno izlaznim adresama. Svaki opisnik se sastoji od 2 registra : statusnog registra (TSD) i početne adrese za slanje (TSAD), čineći ukupno 8 registara (TSD0-3, TSAD0-3) [3].

- Registri TSAD0-3 pohranjuju početne fizičke adrese paketa koji će se slati.
- Registri TSD0-3 su statusni registri u kojima se nalaze podaci o slanju i greškama tijekom slanja pojedinog deskriptora.

Tablica 1.2 TSD registar

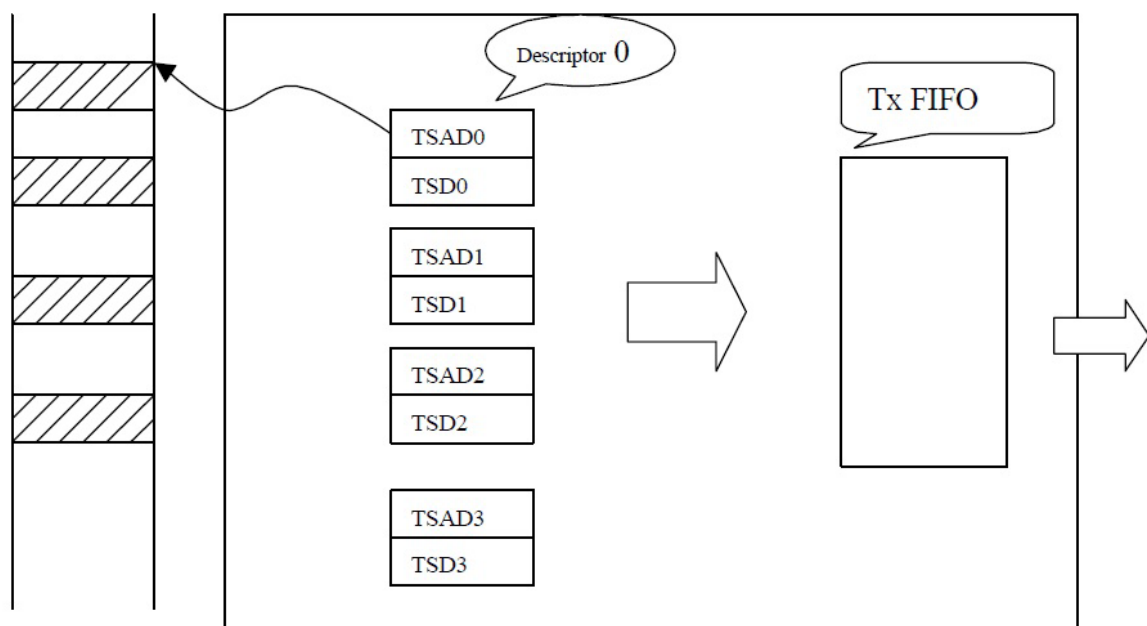
Bit	Oznaka	Opis
31	CRS	Prijenosni signal izgubljen pri prijenosu ( <i>Carrier Sense Lost</i> )
30	TABT	Slanje prekinuto ( <i>Transmit Abort</i> )
29	OWC	Kolizija paketa ( <i>Out of Window Collision</i> )
28	CDH	Ne koristi se kod prijenosa pri brzini od 100Mbps.
27-24	NCC3-0	Brojač kolizija tijekom slanja ( <i>Number of Collision Count</i> )
23-22	-	Rezervirano
21-16	ERTXTH5-0	Prag veličine podataka ( <i>Early Transmit Threshold</i> )
15	TOK	Slanje paketa uspješno završilo ( <i>Transmit OK</i> )
14	TUN	Ispražnjen FIFO spremnik ( <i>Transmit FIFO Underrun</i> )
13	OWN	Završena DMA operacija slanja opisnika.
12-0	SIZE	Veličina opisnika u bajtovima ( <i>Descriptor Size</i> )

Kod slanja koristi se kružni obilazak opisnika. Primjerice, prvo se šalje paket iz opisnika 0, pa tek onda iz opisnika 1, pa 2 pa 3 te ponovno 0 i tako dalje. Ukoliko je opisnik, koji je na



redu za korištenje, zauzet nekim paketom, slanje se privremeno zaustavlja dok se opisnik ne oslobodi. Jedan opisnik omogućuje slanje samo jednog paketa ne većeg od 1792 okteta.

Pri slanju podataka (paketa) oni se najprije iz spremnika (adresa zadana u nekom od TSAD registra) kopiraju u međuspremnik sklopa (*Transmit FIFO*) pa tek onda na fizički medij (npr. paricu), prema slici Slika 1.2. Slanje na fizički medij zbiva se kada veličina podataka u međuspremniku dosegne vrijednost zapisanu u TSD registru (*Early Transmit Threshold* dio).

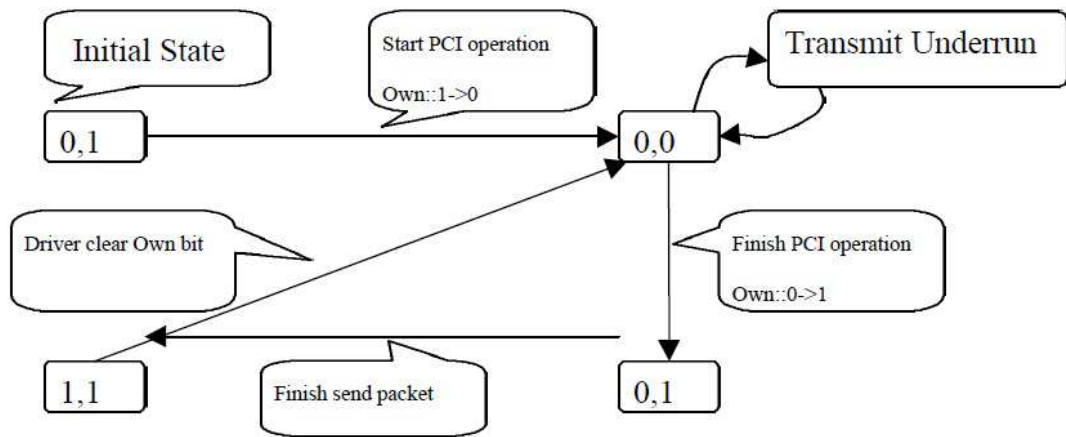


Slika 1.2 Prikaz prijenosa podataka iz memorije u *Transmit FIFO* registar

Proces slanja paketa može se opisati u nekoliko koraka. Neke od njih treba napraviti programski [P], dok se drugi izvode pod upravljanjem sklopa[S]:

1. [P] Kopiranje paketa u radni spremnik (kontinuirani dio)
2. [P] Definiranje opisnika s vrijednostima:
  - a. adresa paketa u registar TSAD te
  - b. veličina paketa, *Early Transmit Threshold*, bit OWN=0 u TSD registar

3. [S] Samo slanje na fizički medij zbiva se kada broj podatak u međuspremniku dosegne broj zadan u TSD registru.
4. [S] Kada se cijeli paket prebaci u međuspremnik sklopa, OWN bit TSD registra se postavlja na 1.
5. [S] Kada se cijeli paket pošalje, TOK bit TSD registra se postavlja na 1.
6. [S] Ako su postavljeni TOK bitovi u IMR i ISR registrima dolazi do prekida.
7. [P] Poziva se prekidna rutina - upravljački program resetira vrijednost TOK(ISR).



Slika 1.3 Dijagram stanja TOK i OWN zastavica registra TSD tijekom slanja paketa [4]

Osim TSD i TSAD registara pri slanju se koriste i registri:

- IMR(*Interrupt Mask Register*) – definira događaje koji trebaju izazvati prekid
- ISR (*Interrupt Status Register*) – stanje prekidne linije.

Registri IMR i ISR [3] prikazani su tablicama Tablica 1.3 i Tablica 1.4.

Tablica 1.3 IMR registar

<i>Bit</i>	<i>Oznaka</i>	<i>Opis</i>
15	<i>SERR</i>	Greška sustava ( <i>System Error InterruptI</i> )
14	<i>TimeOut</i>	Istek vremena ( <i>Time Out Interrupt</i> )
13	<i>LenChg</i>	Promjena duljine kabela ( <i>Cable Length Change Interrupt</i> )
12-7	-	Rezervirano
6	<i>FOVW</i>	Preljev prijemnog FIFO međuspremnik ( <i>Rx FIFO Overflow Interrupt</i> )
5	<i>PUN/LinkChg</i>	Nepotpuni paket / Promjena prijenosne linije ( <i>Packet Underrun/ Link Change Interrupt</i> )
4	<i>RXOVW</i>	Preljev prijemnog međuspremnik ( <i>Rx Buffer Overflow Interrupt</i> )
3	<i>TER</i>	Pogreška pri slanju ( <i>Transmit Error Interrupt</i> )
2	<i>TOK</i>	Slanje obavljeno ( <i>Transmit OK Interrupt</i> )
1	<i>RER</i>	Pogreška pri primanju ( <i>Receive Error Interrupt</i> )
0	<i>ROK</i>	Primljen paket ( <i>Receive OK Interrupt</i> )

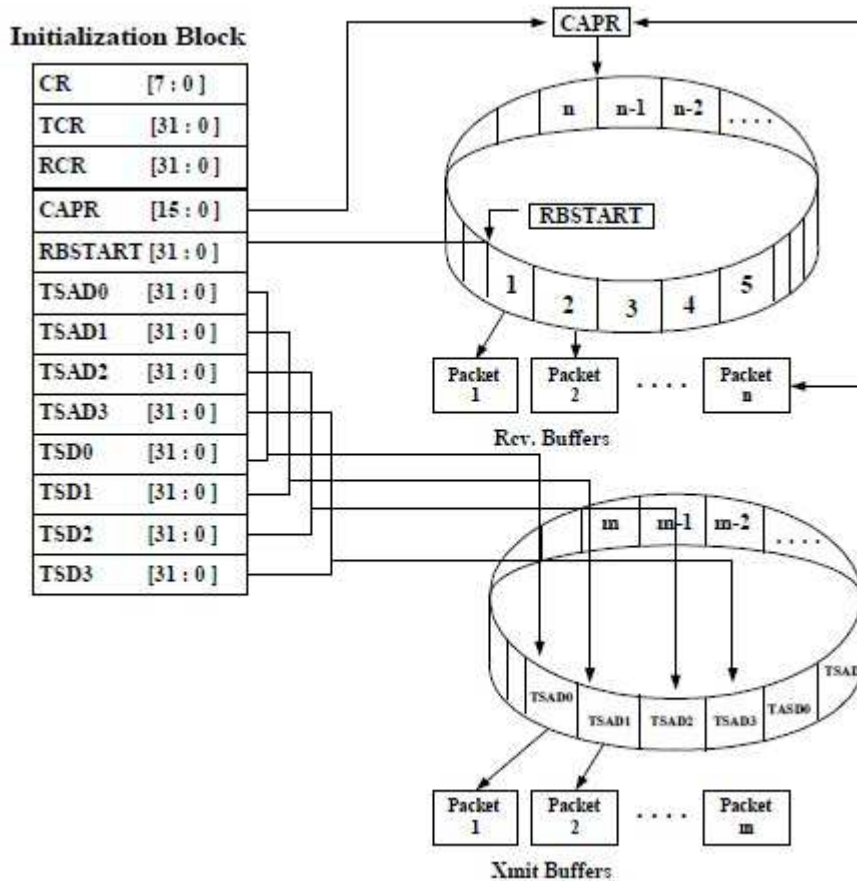
Tablica 1.4 ISR Registar

<i>Bit</i>	<i>Oznaka</i>	<i>Opis</i>
15	<i>SERR</i>	Greška sustava ( <i>System Error Interrupt</i> )
14	<i>TimeOut</i>	Istek vremena ( <i>Time Out Interrupt</i> )
13	<i>LenChg</i>	Promjena duljine kabela ( <i>Cable Length Change Interrupt</i> )
12-7	-	Rezervirano
6	<i>FOVW</i>	Preljev prijemnog FIFO međuspremnik ( <i>Rx FIFO Overflow Interrupt</i> )
5	<i>PUN/LinkChg</i>	Nepotpuni paket / Promjena prijenosne linije ( <i>Packet Underrun/ Link Change Interrupt</i> )
4	<i>RXOVW</i>	Preljev prijemnog međuspremnik ( <i>Rx Buffer Overflow Interrupt</i> )
3	<i>TER</i>	Pogreška pri slanju ( <i>Transmit Error Interrupt</i> )
2	<i>TOK</i>	Slanje obavljeno ( <i>Transmit OK Interrupt</i> )
1	<i>RER</i>	Pogreška pri primanju ( <i>Receive Error Interrupt</i> )
0	<i>ROK</i>	Primljen paket ( <i>Receive OK Interrupt</i> )

## 1.2. Mehanizam primanja paketa

Pri primanju paketa kod sklopa RTL8139 prijemna međumemorija (eng. *Receive Buffer*) je organiziran kao linearna memorija organizirana u prsten (eng. *Ring Buffer*). Prilikom primanja paketa sa fizičkog medija, podaci se prvo spremaju u međuspremnik sklopa (*Receive FIFO* registar) te se prenose u prijemnu međumemoriju kada broj podataka dosegne prag zadan u RCR registru (*Early Rx Threshold*- dio). Sklop puni prijemnu

međumemoriju sve dok ne potroši svu memoriju. Ako se popuni sva memorija, sklop ponovno počinje spremati podatke od početka linearne memorije, čineći je tako memorijom organiziranom u prsten.



Slika 1.4 Prikaz *Receive i Transmit Buffera* [4]

Pri primanju paketa CBA registar sadrži trenutnu adresu podataka koji se prenose u međumemoriju. CAPR registar je pokazivač koji pokazuje na adresu podataka koji je upravljački program pročitao. Statusni podaci o primanju pojedinog paketa su zapisani u zaglavlju samog paketa (Tablica 1.5) [3].

Tablica 1.5 Zaglavlje paketa

Bit	Oznaka	Opis
15	MAR	<i>Multicast Address Received-1</i> => primljen multicast paket.
14	PAM	<i>Physical Address Matched-1</i> => određena adresa odgovara vrijednosti zapisani u ID registrima.
13	BAR	<i>Broadcast Address Received-1</i> => primljena broadcast adresa.
12-6	-	Rezervirano
5	ISE	<i>Invalid Symbol Error-1</i> => nađen krivi simbol tijekom primanja paketa.
4	RUNT	<i>Runt Packet Received-1</i> => primljeni paket je manji od 64 bajta.
3	LONG	<i>Long Packet-1</i> => duljina paketa je veća od 4k bajta.
2	CRC	<i>CRC Error-1</i> => dogodila se CRC( <i>cyclic redundancy check</i> ) pogreška.
1	FAE	<i>Frame Alignment Error-1</i> => dogodila se <i>frame alignment</i> pogreška.
0	ROK	<i>Receive OK-1</i> => uspješno primljen paket.

Proces primanja paketa u koracima. Koraci su podijeljeni na programske [P] i sklopovske [S]:

1. [S] Primljeni podatak se sprema u međuspremnik (*Receive FIFO*).
2. [S] Kada broj podataka u međuspremniku dosegne broj zadan u RCR registru započinje prijenos u prijemnu međumemoriju (*Receive Buffer*).
3. [S] Kada je cijeli paket premješten iz međuspremnika u prijemnu međumemoriju, zaglavlje paketa se zapisuje na početak paketa te se osvježava vrijednost CBA registra (adresa kraja paketa).
4. [S] U registru CR se postavljaju bit *Buffer Empty* te se u ISR registru postavlja bit TOK.

5. [P] Zove se ISR rutina i upravljački program resetira bit TOK u ISR registru, šalje paket i osvježava vrijednost CAPR registra.

Ostali registri koji se koriste pri primanju paketa:

- ISR registar.
- IMR registar.
- RBSTART registar –startna adresa prijemne međumemorije.
- CR (*Command Register*) – registar koji služi za izdavanje naredbi sklopu.
- RCR (*Receive Configuration Register*)- registar preko kojeg se konfiguriraju postavke za primanje paketa.

## 2. Ostvarenje upravljačkog programa

Upravljački program sklopa RTL8139 je pisan za jednostavni operacijski sustav Benu, te je pisan u programskom jeziku C kao i sam sustav. Operacijski sustav sam sklop vidi kao sučelje koje je zastupljeno strukturom `device_t` (Kod 2.1). To znači da operacijski sustav komunicira sa sklopom preko `device_t` strukture, a upravljački program prenosi te naredbe samom sklopu.

```
struct _device_t_
{
    /* device name */
    char    dev_name[DEV_NAME_LEN];

    /* which IRQ is using */
    int     irq_num;

    /* interrupt handler function */
    void    (*irq_handler) ( int irq_num, void *device );

    /* callback function (to kernel) */
    int     (*callback) ( void *param );
    void    *callback_param;
    /* device interface */
    int     (*init)(uint flags, void *params, device_t *dev );
    int     (*destroy) (uint flags, void *params, device_t *dev );
    int     (*send)(void *data, size_t size, uint flags,      device_t *dev
);
    int     (*recv)( void *data, size_t size, uint flags, device_t *dev );

    /* various flags and parameters specific to device */
    int     flags;
    void    *params;
};
```

Kod 2.1 `device_t` struktura



Glavno sučelje prema sklopu čine funkcije:

1. `int(*init)(uint flags, void *params, device_t *dev )`
2. `int(*destroy)(uint flags, void *params, device_t *dev )`
3. `int(*send)(void *data, size_t size, uint flags, device_t *dev)`
4. `int(*recv)( void *data, size_t size, uint flags, device_t *dev)`
5. `void (*irq_handler)(int irq_num, void *device)`

## 2.1. Ostvarenje funkcije `init`

Funkcija `init` (Kod 2.2) se brine za početnu inicijalizaciju sklopa i struktura koje su potrebne za njen ispravan rad. Funkciju poziva operacijski sustav preko strukture `device_t`, koja sadrži pokazivač na funkciju, kako bi se sklop inicijalizirao za normalan rad. Funkcija alocira memoriju za strukture sklopa, dohvaća podatke i inicijalizira registre za rad sa sklopom tj. slanje i primanje podatka.

```

init(){
    alociraj memoriju za privatne podatke uređaja;
    dohvati bazne adrese uređaja;
    dohvati MAC adresu uređaja;
    postavi funkciju za prekide;
    alociraj memoriju za opisnik;
    alociraj memoriju za pakete koji se primaju;
    dodjeli memoriju opisnicima;
    resetiraj uređaj;
    omogući slanje i primanje paketa;
    upiši veličinu paketa i duljinu buffer-a
    upiši početne adrese opisnika u TSAD registre;
    upiši početnu adresu paketa koji se primaju
    omogući prekide;
    javi operacijskom sustavu da je sklop inicijaliziran;
}

```

Kod 2.2 Pseudokod init funkcije

## 2.2. Ostvarenje funkcije send

Funkciju send (Kod 2.3) poziva operacijski sustav preko `device_t` strukture kada želi poslati određene pakete podataka. Sama funkcija je trivijalna te ima zadatak da odredi i inicijalizira prvi slobodni opisnik za slanje paketa. Kada je opisnik inicijaliziran, funkcija prosljeđuje opisnik sa odgovarajućim podacima na slanje.

```

send(){
    uzmi prvi prazan opisnik;
    ako je broj podataka manji od minimalnog broja za ethernet paket(60
okteta)
        dopuni paket sa nulama;
    proslijedi opisnik;
    upisi duljinu paketa u registar sklopa;
    odredi „novi“ prazan opisnik;
    ako je (novi prazan opisnik == opisnik čiji sadržaj još nije
poslan)
        zaustavi slanje;
}

```

Kod 2.3 Pseudokod send funkcije

## 2.3. Ostvarenje funkcije `recv`

Funkciju `recv` (Kod 2.4) poziva prekidna funkcija `irq_handler` ukoliko je došlo do prekida koji javlja uspješno primljen paket. Funkcija čita primljene podatke, uklanja višak informacija koji se nalaze u paketu i šalje ih na daljnju obradu višim slojevima.

```

recv(){
    dok je(broj pročitanih podatak != broj primljenih podatak)
        pročitaj podatke;
        ukloni višak informacija;
        pošalji podatke višem sloju;
    ažuriraj adresu za pisanje pristiglih paketa;
}

```

Kod 2.4 Pseudokod `recv` funkcije

## Zaključak

U okviru ovog završnog rada izrađen je upravljački program za mrežnu karticu RTL8139 koji omogućuje inicijalizaciju samog uređaja, te slanje i primanje paketa. Upravljačku program je pisan u programskom jeziku C, kao i operacijski sustav Benu kojem je upravljački program namijenjen.

Mehanizam slanja se temelji na 4 opisnika pomoću kojih se paketi iz spremnika, preko sklopovskog međuspremnik, šalju na fizički medij.

Kod primanja paketa sa fizičkog medija, paketi se preko međuspremnik sklopa, prenose u prijemnu međumemoriju iz koje se paketi prosljeđuju višim slojevima. Sama konfiguracija sklopa za primanje i slanje paketa se vrši preko brojnih registara sklopa u koje se pohranjuju određeni podaci.

Funkcionalnost mrežne kartice je ostvarena kroz funkcije `init`, `send` i `recv` koje operacijski sustav poziva preko `device_t` strukture koja predstavlja sučelje prema kartici.

## Literatura

- [1] Realtek, RTL8139C(L),  
<http://152.104.125.41/products/productsView.aspx?Langid=1&PFid=6&Level=5&Conn=4&ProdID=16>
- [2] Wikipedia, Realtek, <http://en.wikipedia.org/wiki/Realtek>
- [3] Realtek, RTL8139 Data Sheet,  
[http://www.cs.usfca.edu/~cruse/cs326f04/RTL8139D\\_DataSheet.pdf](http://www.cs.usfca.edu/~cruse/cs326f04/RTL8139D_DataSheet.pdf), 9.11.2001.
- [4] Realtek, RTL8139 Programming guide,  
[http://www.cs.usfca.edu/~cruse/cs326f04/RTL8139\\_ProgrammersGuide.pdf](http://www.cs.usfca.edu/~cruse/cs326f04/RTL8139_ProgrammersGuide.pdf),  
15.1.1999.
- [5] Mohan Jal Jangir, Writing Network Drivers for Linux,  
<http://linuxgazette.net/156/jangir.html>, studeni 2008.

# Sažetak

**Naslov:** Ostvarenje upravljačkog programa za RTL8139 mrežnu karticu

**Ključne riječi:** Mrežna kartica, Realtek, RTL8139, Benu, Slanje paketa, Opisnik, Primanje paketa

Cilj rada bio je izrada upravljačkog programa za mrežnu karticu RTL8139. Upravljački program je namijenjen operacijskom sustavu Benu kojem omogućuje slanje, primanje i samu inicijalizaciju uređaja. Upravljački program je pisan u programskom jeziku C i glavna funkcionalnost sadržana je u funkcijama `init`, `send` i `recv`.