

# DIZAJN I IMPLEMENTACIJA MREŽNIH UREĐAJA KOJI KORISTE WIRELESSHART PROTOKOL DESIGN AND IMPLEMENTATION OF NETWORK DEVICES THAT USE WIRELESSHART PROTOCOL

Miroslav Kostadinović, *Saobraćajni fakultet, Doboj*  
Božidar Popović, *Elektrotehnički fakultet, Istočno Sarajevo*  
Nataša Popović, *Elektrotehnički fakultet, Istočno Sarajevo*

**Sadržaj** - U ovom radu opisano je korištenje TrueTime simulatora baziranog na Matlab/Simulinku za dizajn i implementaciju mrežnih uređaja koji koriste WirelessHART protokol. TrueTime simulator je sposoban da simulira kompleksne mrežne sisteme i upravljačke parametre u realnom vremenu. Model WirelessHART mreže u ovom radu sadrži četiri različita uređaja koja su inicirana pisanjem izvršnog koda u odvojenim M-fajlovima i to za kontroler, gateway, senzor i aktuator.

**Abstract** - This paper describes the use of TrueTime simulator based on Matlab / Simulink for the design and implementation of network devices using WirelessHART protocol. TrueTime simulator is able to simulate complex network systems and control parameters in real time. WirelessHART network model in this paper contains four different devices that are initiated by writing the executive code in separate M-files to the controller, gateway, sensor and actuator.

## 1. UVOD

Novi uslužni program koji ćemo opisati u ovom radu je implementiran da poboljša ponašanje i upotrebljivost Wireless Network bloka u TrueTime. Originalno su implementirane dvije vrste komunikacijskog protokola, 802.11b/g (WLAN) i 802.15.4 (ZigBee), a mogućnost korištenja WirelessHart-a je nahnadno dodana. Cilj razvoja WirelessHART-a je da se uspostavi standard za bežičnu komunikaciju za primjenu u procesnoj automatici [1].

HCF (engl. *Hart Communication Foundation*) je dizajnirala specifikaciju wirelessHART mreže da bi se osigurala jednostavna, pouzdana i sigurna komunikacija između bežičnih uređaja za automatizovane procese kod industrijskih postrojenja, kao zamjena postojeće HART mreže.

WirelessHART protokol radi na frekvenciji od 2.4 GHz ISM opsegu fizičkog sloja, a koristi IEEE 802.15.4 standard koristeći TDMA (engl. *Time Division Multiple Access*) za pristup komunikacionom mediju [2]. Kompletno vrijeme komunikacije se izvršava unutar unaprijed određenog vremenskog slota od 10 ms. Niz vremenskih slotova formira superfrejm za prenos podataka i WirelessHart omogućava tehniku skakanja između komunikacionih kanala da bi se izbjegla interferencija i redukovao *multi-path fading* efekat.

Jedan ili više uređaja koji šalju podatke i jedan ili više uređaja koji primaju podatke, mogu biti određeni da komuniciraju međusobno u jednom unaprijed predviđenom vremenskom slotu. Vremenski slot može biti rezervisan za komunikaciju samo jednog uređaja ili može podržavati djeljenje prenosa podataka preko jednog vremenskog slota. WirelessHart je komunikacioni protokol koji je organizovan kao ISO/OSI 7 komunikacioni model [3].

WirelessHART je dizajniran da bude kompatibilan sa postojećim HART uređajima, koji rade na žičanim komunikacijskim sistemima.

## 2. KARAKTERISTIKE PROTOKOLA

### 2.1. Pouzdanost

Da bi se postigla pouzdana komunikacija pri gustom okruženju industrijskih postrojenja, kao i zbog pomjeranja velikih vozila i opreme, te usljed promjene uslova i elektromagnetnih smetnji, wirelessHART uključuje nekoliko funkcija koje su opisane u nastavku:

**Redundatno rutiranje** je veoma važno u realnom svijetu RF okruženja. Promjene uslova u industrijskim postrojenjima najprije zbog vremenski uslova, prepreka u komunikacijskim putanjama, dovode to toga da redundantno mrežno rutiranje ponovo rutira prenos ukoliko je došlo do dužeg prekida u komunikaciji mreže.

**Tehnika skakanja između prenosnih kanala** koristi se za povećanje pouzdanosti, a dozvoljava dinamičko biranje bilo kojeg od 16 definisanih kanala prema standardu IEEE 802.15.4. Smetnje od strane drugih izvora mreža, jer u industriji postoji više od jedne mreže koja koristi isti 2.4 GHz ISM opseg, WirelessHART izbjegava ove probleme preko tehnike skakanja između prenosnih kanala [4].

**Vremenski sinhronizirana komunikacija** nudi pouzdanu i efikasnu komunikaciju omogućavajući interoperabilnost i jednostavnost upotrebe. Definiše prioritet za svaku poruku, te osigurava kvalitetan servis isporuke. Fiksni vremenski intervali omogućavaju kreiranje optimalne mreže za bilo koju aplikaciju sa samo konfigurirajućim pristupom bez intervencije korisnika.

### 2.2. Sigurnost

WirelessHART nudi visoko sigurnu mrežu i uključuje:

- *Enkripciju*. Sve poruke u mreži se enkriptuju preko 128-bitnog enkripcionog ključa prije nego što se izvrši prenos.

- *Verifikacija*. Kodovi integrisani u poruke verifikuju svaki paket
- *Robusnost operacije*. Preskakanje kanala i mrežna infrastruktura ublažava učinak ometanja te odbijanja usluga
- *Ključevi*. WirelessHART mreže osiguravaju različite ključeve za svaki uređaj, kao što su mrežni ključ, pristupni ključ, ključ za pristup i komunikaciju unutar mreže. Ovi ključevi se periodično rotiraju tako da se izbjegne neovlaštenim uređajima da pristupe u mrežu.
- *Autorizacija*. Uređaji mogu da pristupe u mrežu samo sa odgovarajućim ID ili pristupnim ključem.

### 2.3. Napajanje

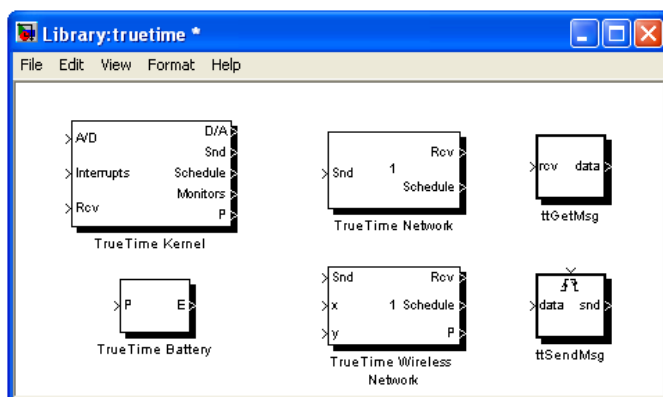
Bežični uređaji nude veliku fleksibilnost i nižu cijenu instalacije, nastojeći da budu „istinski wireless“, što znači da ima baterijsko napajanje čiji životni vijek zavisi od potrošnje energije [5].

Wireless HART koristi dve funkcije kako bi smanjili potrošnju energije kod komunikacije između uređaja kao i između uređaja i korisnika, a to su *Smart Data Publishing* i *Notification by Exception* tako što:

- Prva omogućava prenos poruka samo kada su promjenjeni uslovi u procesu ili informacije zahtjevane od strane korisničkih aplikacija. Ovim se izbjegava nepotrebno slanje poruka a samim tim povećavaju se komunikacijski zahtjevi te efikasnost.
- Druga, kada dođe do promjene u statusu procesa, automatski generiše poruku obavještenja prema aplikaciji.

### 3. TRUETIME SIMULATOR

Ovo poglavlje opisuje upotrebu originalnog simulatora TrueTime baziranog na Matlab/Simulinku, koji simulira upravljački sklop u izvršavanju zadataka u real-time sistemima, mrežama (žičane ili bežične mreže) i dinamičkim pogonima.



Sl. 1. TrueTime biblioteka Simulink blokova

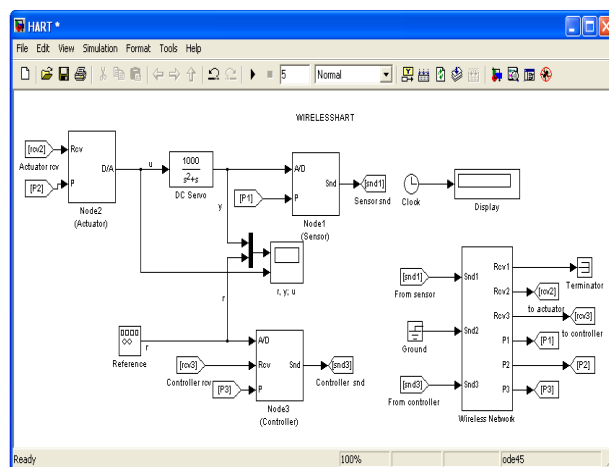
TrueTime ne sačinjavaju samo bibliotečni blokovi (slika 1), već i kolekcije C++ funkcija sa odgovarajućim MATLAB MEX-interfejsima. Neke funkcije omogućavaju pomoću simulacije kreiranje zadataka, ručno prekidanje, nadzor, tajmere, itd. Druge funkcije su real-time koje se pozivaju kodom za vrijeme izvršenja zadatka i omogućuju AD/DA konverziju, slanje i primanje poruka, itd. TrueTime je

razvijen na Simulinku, koji vodi brigu o upravljačkom sistemu u smislu performansi, stabilnosti i izdržljivosti, a prvenstveno je namijenjen korištenju zajedno s MATLAB/Simulinkom [6].

Wireless Network blok simulira pristup mediju i prenošenje paketa. Originalno u TrueTime su implementirane dvije vrste komunikacijskog protokola, 802.11b/g (WLAN) i 802.15.4 (ZigBee), dok je mogućnost korištenja WirelessHart-a je nahnadno dodana.

### 4 DIZAJN I IMPLEMENTACIJA MODELA

Model upravljačkog sistema sadrži tri uređaja tj.čvorova jer HART protokol objedinjuje funkciju kontrolera i gateway-a te su prikazani kao jedan čvor na slici 2., a smješteni su na međusobnoj razdaljini od 20 metara. Svaki od uređaja je predstavljen sa TrueTime kernelom, a povezani su jednim Wireless Network blokom [7].



Sl. 2. Model upravljačkog sistema

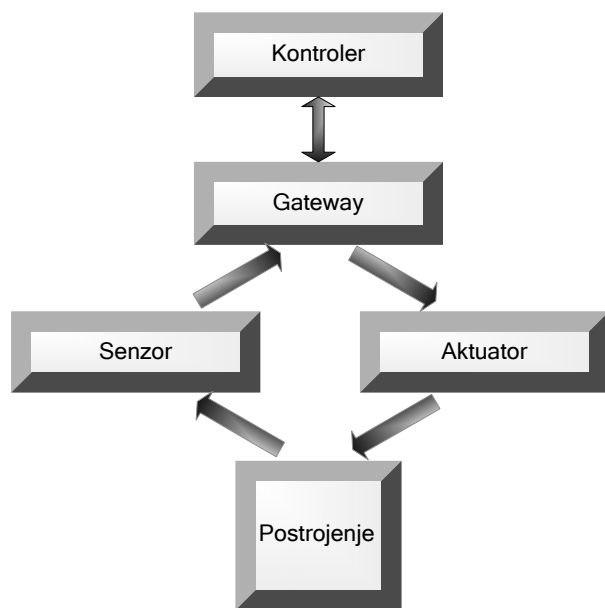
U ovom upravljačkom sistemu postoji samo jedna petlja u kojoj senzor periodično konvertuje analogni signal iz procesa u digitalnu vrijednost i šalje ga kontroleru. Kontroler, nakon što primi poruku od senzora, izračunava izlaz u skladu sa odgovarajućim upravljačkim algoritmom i šalje upravljački signal aktuatoru koristeći WirelessHart mrežu, kao na slici 2. Aktuator konvertuje primljeni upravljački signal u analogni i biva pokrenut.

#### 4.1. Arhitektura korisničkog interfejsa

Na samom početku implementacije moramo postaviti neka ograničenja u sistemu, tako što predpostavimo da koristimo topologiju zvijezde u mreži i da imamo samo jedan aktivni kanal. To znači da je bežičnim uređajima dovoljan samo jedan skok do gateway-a.

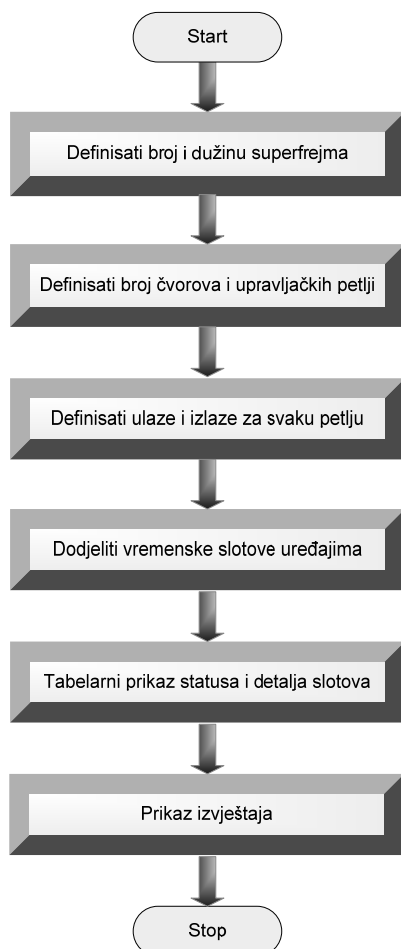
Slika 3. prikazuje osnovni blok dijagram za model Simulink blokova kod WirelessHART simulatora. Sensorski čvor očitava vrijednosti varijable procesa koje treba da su kontrolisane od strane procesa postrojenja, te ih zatim transmituje do gateway uređaja. Gateway uređaji primaju podatke od senzora, a zatim se vrši zapisivanje vrijednosti na kontrolere, da bi se proračunao upravljački signal zahtjevan za izvođenje operacija u procesima postrojenja. Kontroler izračunava vrijednost upravljačkog signala i zapisuje ga na gateway. Gateway očitava vrijednost i šalje ga prema aktuatoru. Aktuator, prima vrijednost upravljačkog signala

od gateway-a i zapisuje ga u proces postrojenja. Procesi u postrojenju izvedu operaciju na osnovu primjenjenih vrijednosti upravljačkog signala.



Sl. 3. Šema komunikacije između bežični uređaja

Prije simuliranja rada mreže, treba da se osiguraju korisnički unosi u mrežu preko korisničkog interfejsa. Slika 4. prikazuje niz komunikacijskih veza ostvarenih između uređaja u jednom komunikacijskom krugu.



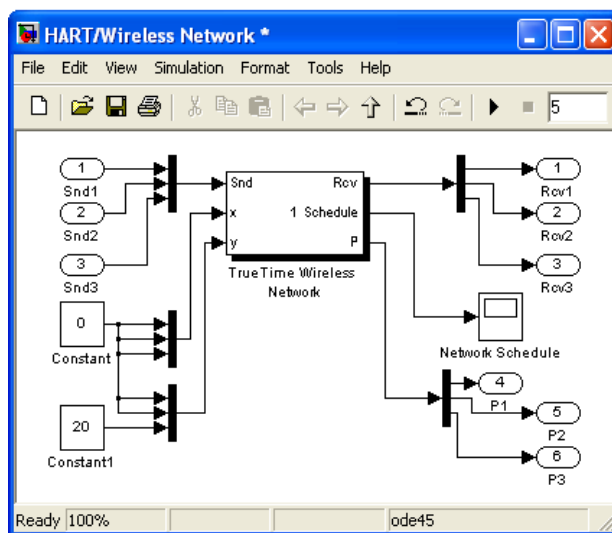
Sl. 4. Arhitektura korisničkog interfejsa

Značenje pojedinih blokova na slici 4. je sledeće:

- Prvi procesni blok dozvoljava korisniku da definiše broj superfrejmova i dužinu svakog od njih u mreži. Svi superfrejmovi su različite dužine.
- Drugi blok dozvoljava korisniku da definiše broj senzora, aktuatora u mreži kao i broj upravljačkih petlji u mreži.
- Treći blok dozvoljava definisanje zavisnog ulaza i zavisnog izlaza za svaku upravljačku petlju. Zavisan ulaz je podatak dobijen od strane senzorskog uređaja, za koji upravljački signali treba da budu izračunati. Zavisan izlaz predstavlja za bežični uređaj izračunati upravljački signali koji treba da bude odaslan.
- Četvrti blok raspoređuje svakom uređaju u mreži određeni vremenski slot u odgovarajućem superfrejmu.
- Peti blok dozvoljava tabelarni prikaz statusa i detalje o svakom vremenskom slotu svih superfrejmova.
- Šesti blok dozvoljava prikaz informacije o rasporedu izvršavanja zadataka pojedinih uređaja u mreži, kao i o greškama i upozorenjima.

#### 4.2. Konfiguracija mrežnih uređaja

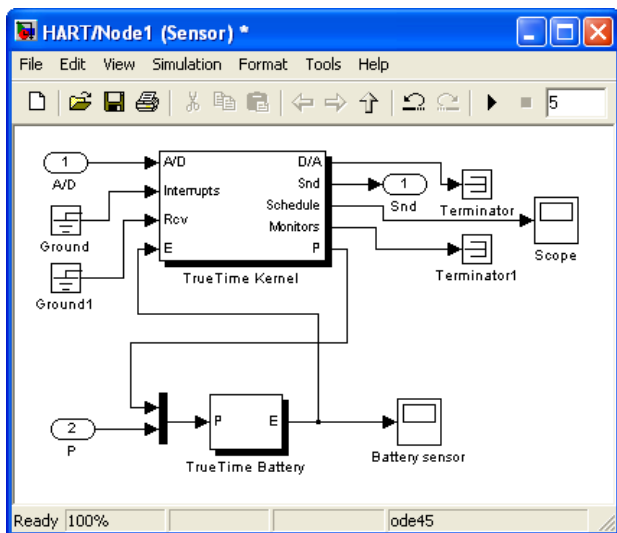
U TrueTime simulacionom modelu koji je prikazan na slici 2., dizajniran je umreženi upravljački sistem, gdje su povezana tri mrežna uređaja korištenjem komunikacije putem WirelessHART mreže [5,6,7]. Dvostrukim klikom na Wireless Network blok na slici 5. otvara se dijaloški prozor u koji unosimo broj uređaja/čvorova i izabiremo WirelessHart protokol.



Sl. 5. TrueTime Wireless Network bloka

U ovom sistemu postoji četiri različita uređaja senzori, gateway, kontroler i aktuator, međutim pri simulaciji HART protokola objedinjena je funkcija kontrolera i gateway-a u jedan čvor. Svaki čvor se prije upotrebe u mreži mora inicijalizirati.

**Inicijalizacija senzora.** Mrežni uređaji/čvorovi su simulirani u podsistemu sa TrueTime kernel blokom. Detalji podsistema za senzor/čvor1 su dati na slici 6. Senzor/čvor1 koristi jedan A/D konvertor u ulaznom dijelu i jedan mrežni izlaz (Snd) na izlazu.



Sl. 6. TrueTime kernel blok senzora

Senzor/čvor1 periodično konvertuje informacije iz analognog izlaza i šalje informacije mreži, koristeći funkciju `sensor_init` koja je pozvana u TrueTime kernelu, a prikazana je na slici 7.

```

1 function sensor_init
2
3 % Inicijalizacija TrueTime kernel
4 ttInitKernel(1, 0, 'prioFP'); % broj ulaza, broj izlaza, fiksni prioritet
5 ttSetKernelParameter('energyconsumption', 0.0100); %10 mW
6
7 % Kreiranje mailboxes
8 ttCreateMailbox('control_signal', 10)
9 ttCreateMailbox('power_ping', 10)
10 ttCreateMailbox('power_response', 10)
11
12 % Kreiranje sensor task
13 data.y = 0;
14 offset = 0;
15 period = 0.010;
16 prio = 1;
17 ttCreatePeriodicTask('sens_task', offset, period, prio, 'senscode', data);
18
19 % Kreiranje power controller task
20 offset = 5;
21 period = 0.025;
22 prio = 2;
23 power_data.transmitPower = 20;
24 power_data.name = 1; % Čvor broj 1 u mreži
25 power_data.receiver = 3; % Komunikacija sa čvorom 3
26 power_data.haverun = 0;
27 ttCreatePeriodicTask('power_controller_task', offset, period, prio, 'powctrlcode', power_data);
28
29 % Kreiranje power response task
30 deadline = 100;
31 prio = 3;
32 ttCreateTask('power_response_task', deadline, prio, 'powrscpcode');
33
34 % Inicijalizacija mreže
35 ttCreateInterruptHandler('hw_handler', prio, 'msgRcvSensor');
36 ttInitNetwork(1, 'hw_handler');

```

Sl. 7. Funkcija `sensor_init`

Inicijalizacija uređaja može se izvršiti pisanjem dijela koda u odvojenim M-fajlovima za svaki uređaj, tako što moramo da definišemo u mreži broj analognih ulaza i izlaza za svaki čvor i planirati izvršavanje zadataka. Ovo se može postići pisanjem funkcije `ttInitKernel`, u sledećem obliku: `ttInitKernel(nbrOfInputs, nbrOfOutputs, scheduling policy)`. Na slici 7. imamo prikazan M-fajl inicijalizacije senzora gdje je implementirana prethodna funkcija.

Od strane korisnika definiše se offset kanala, određena adresa, pravac komuniciranja, karakteristika veze, ID okvir za svaki vremenski slot i kreira se tabela podataka koja sadrži sve informacije o čvorovima. Nakon čega se kreira periodični zadaci koji su aktivni svakih 0,01s, što se postiže pisajući funkcije `ttCreatePeriodicTask`. Kada pozivamo ovu

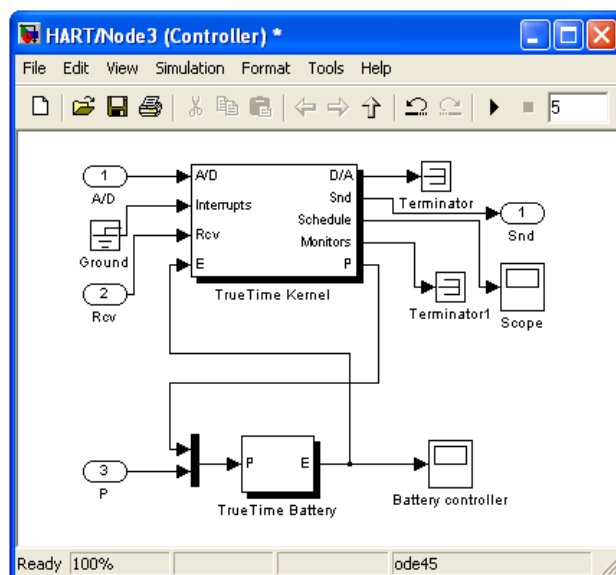
funkciju moramo definisati parametre prikazane u sledećem obliku: `ttCreatePeriodicTask ('task name', offset, period, prio, 'function name', data)`. U nastavku svaki od navedenih parametara biće posebno objašnjen:

- *Task name* parametar traži da se osigura ime za svaki zadatak, tako da ne postoji mogućnost istih imena zadataka u mreži.
- *Offset* predstavlja vrijeme koje se koristi za odlaganje odnosno kašnjenje izvršavanja zadatka posle buđenja.
- *Period* je definisan u sekundama i koristi se za periodično pobudjivanje.
- *Function name* parametar koristi se za davanje imena funkciji koja treba da bude pozvana prilikom aktiviranja zadatka.
- *Data* se koristi kada korisnik želi prosljediti neke parametre za pozivanje funkcije i tada moramo pozvati funkciju koja upravlja prekidima za čvorove.

Da bi koristili funkciju prekida moramo kreirati prekidni task, definisati prioritet funkcije i ime funkcije koja treba da se izvrši pri prekidu, a što se može postići pisanjem funkcije: `ttCreateInterruptHandler (handler name, priority, function name)`.

Svaki uređaj/čvor koji sadrži blok TrueTime kernela mora biti pokrenut od strane funkcije `ttInitNetwork (identifikacioni broj uređaja, ime funkcije mrežnog prekida)`. Postavlja mrežnu adresu uređaja/čvora i također definiše koji će prekid biti izveden kada stigne nova poruka. Ukoliko je potrebno da se periodično izvodi zadatak na uređaju/čvoru, koristiti se funkcija `ttCreatePeriodicTask (period, prioritet, naziv m – fajla, lokalne informacije za uređaj)`.

**Inicijalizacija kontrolera/gateway.** Kontroler prima poruke od senzora i šalje podatke natrag na mrežu tj. aktuatoru. Detalji podsistema za kontroler/čvor3 su dati na slici 8. gdje koristi jedan A/D konvertor i jedan mrežni ulaz (Rcv) na ulaznom dijelu, jedan mrežni izlaz (Snd) na izlaznom dijelu.



Sl. 8. TrueTime kernel blok kontrolera

Nakon podešavanja parametara Kernel bloka, potrebno je kreirati ctrlcode M-fajl koji je prikazan na slici 9. Ovaj napisani kod izračunava upravljački izlazni signal tako što koristi jednačine za PD kontrolere [5,7].

```

1 function [exectime, data] = ctrlcode(seg, data)
2
3 switch seg,
4
5 case 1,
6     temp = ttTryFetch('sensor_signal');
7     while ~isempty(temp),
8         y = temp;
9         temp = ttTryFetch('sensor_signal');
10    end
11    % Cita referentna vrijednost
12    r = ttAnalogIn(1);
13    P = data.R*(r-y);
14    D = data.ad*data.dold + data.bd*(data.yold-y);
15    data.u = P + D;
16    data.dold = D;
17    data.yold = y;
18    exectime = 0.0005;
19
20 case 2,
21     msg.msg = data.u;
22     msg.type = 'control_signal';
23     ttSendMsg(2, msg, 80); % Slanje poruke cvoru 2 (actuator)
24     exectime = -1;
25 end

```

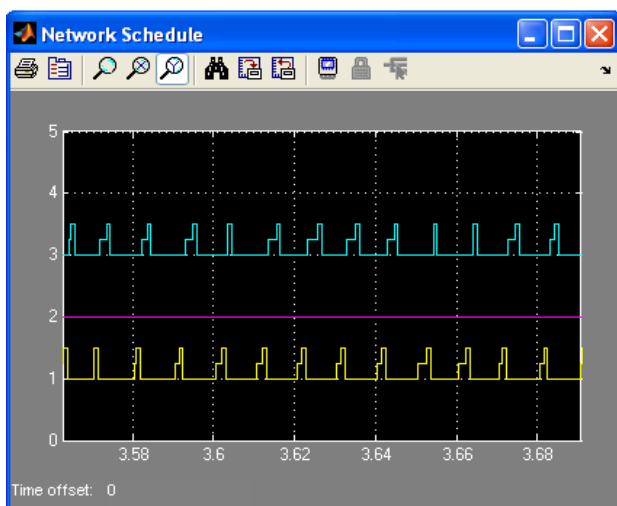
Sl. 9. Funkcija ctrlcode

Analogijom i za preostale Simulink blokove u modelu upravljačkog sistema na slici 1., potrebno je napisati M-fajlove kako bismo testirali model bežične mreže u kojoj je komunikacija između uređaja ostvarena koristeći WirelessHART protokol [3], [4].

## 5. SIMULACIJA PLANIRANJA ZADATAKA U MREŽI

Ukoliko simulator ne prijavi grešku prilikom testiranja koda uređaja, tada možemo startovati simulaciju modela sa slike 1. Nakon pokretanja simulacije otvara se prozor kao na slici 10. koji omogućava da pratimo planiranje izvršavanja zadataka uređaja koji ostvaruju komunikaciju koristeći WirelessHART protokol [2].

Dobijeni rezultati tj. dijagrami modela upravljačkog sistema prikazuju planiranje zadataka uređaja i imaju sledeće značenje: visok signal znači slanje, srednji predstavlja čekanje i nizak da je uređaj besposlen.



Sl. 10. Planiranje izvršavanja zadataka u mreži

Također, dobijeni dijagrami prikazuju početno vrijeme kao i ukupno vrijeme za izvršavanje zadataka svakog uređaja. Vreme za izvršenje svakog uređaja nije definisano od strane korisnika nego je fiksno u ovom slučaju. Zadaci gateway/aktuator za prijem podataka od strane senzor/gateway uređaja je definisano. Tako da se zadatak izvršava poslije završetka izvršavanja odgovarajućeg odašiljanja uređaja.

## 6. ZAKLJUČAK

Automatizovano upravljanje predstavlja savremeni način proizvodnje u procesnoj industriji. Danas postoji trend zamjene žičanih komunikacijski sistema sa novim razvijenim bežičnim komunikacijskim sistemima, zbog njihovog efikasnijeg te jeftinijeg načina priključenja i korištenja. Bežični sistemi za komunikaciju su veoma fleksibilni i osiguravaju sigurnu i pouzdanu komunikaciju između mrežnih uređaja.

HCF nudi specifikacije bežičnog komunikacijskog mrežnog sistema poznatog kao WirelessHART protokol. Većina industrija vrši zamjenu postojeći žičanih komunikacijski sistema sa bežičnim komunikacijskim sistemom baziranim na WirelessHART-u. Glavni razlog njegovog uvođenja pored navedenoga, jeste postojanje više od 26 miliona HART uređaja u industrijskoj primjeni sa više od 1000 različitih tipova uređaja od 220 proizvođača. WirelessHART je kompatibilan sa HART uređajima, što smanjuje cijenu instaliranja novih uređaja i vremena potrebnog za implementaciju bežičnog sistema. Pa prema tome, WirelessHART nudi jednostavan, siguran i pouzdan način komunikacije između uređaja.

## LITERATURA

- [1] HART Communication Foundation: "Control with WirelessHART", HCF LIT-127, Revision 1.0, June30, 2008.
- [2] M. Kostadinović, M. Stojčev, Z. Bundalo, D. Bundalo: "Upravljanje WirelessHART mrežom", Međunarodni naučno-stručni simpozijum INFOTEH-JAHORINA 2009., Jahorina, 18.-20. marta 2009.
- [3] M. Kostadinović, M. Stojčev, Z. Bundalo, D. Bundalo: "Design, Implementation and Simulation of WirelessHart Network", 9th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services - TELSIS 2009., Niš, 7-9.10. 2009.
- [4] M. Kostadinović, Z. Bundalo, D. Bundalo, B. Popović: "PROBLEM OF PACKET LOSS IN WIRELESSHART NETWORK", 13<sup>th</sup> International Research/Expert Conference "Trends in the Development of Machinery and Associated Technology" TMT 2009, Hammamet, Tunisia, 16 -21 October 2009.
- [5] Andersson Martin, Dan Henriksson, Anton Cervin and Karl-Erik Årzén: "Simulation of wireless networked control systems", in Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005. Seville, Spain, December 12-15 2005.
- [6] A. Cervin, D. Henriksson, M. Ohlin: "TrueTime 2.0 beta-Reference Manual", Department of Automatic Control, Lund University, Lund, Sweden, January 2009.
- [7] M. Kostadinović, M. Stojčev, Z. Bundalo, D. Bundalo: "APPLICATION OF MODIFIED TRUETIME SIMULATOR IN CONTROL OF WIRELESSHART NETWORKS", 15<sup>th</sup> INTERNATIONAL SYMPOSIUM on POWER Electronics – Ee2009, Novi Sad, Republic of Serbia, October 28<sup>th</sup> – 30<sup>th</sup>, 2009.