

# Avtomatsko testiranje avtomobilskih diagnostičnih naprav

Matej Rakovec, Boštjan Murovec

ComTrade, d. o. o., Letališka cesta 29b, 1000 Ljubljana, Slovenija,  
Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija  
E-pošta: matej.rakovec@comtrade.com, bostjan.murovec@fe.uni-lj.si

**Povzetek.** Članek opisuje sistem za avtomatizirano testiranje avtomobilskih diagnostičnih naprav, ki vključuje skriptni jezik ETSL (Embedded Testing Script Language) za avtomatizacijo postopkov testiranja, programski paket UETS (Universal Embedded Testing System), ki izvaja skripte in krmili celotno napravo, ter emulator referenčnih veličin, ki napravo vzbuja na enak način, kot bi to počeli senzori, s katerimi naprava meri parametre delovanja avtomobilskega motorja. Avtomatsko testiranje poteka pod nadzorom osebnega računalnika. Testiranje naprave poteka tako, da emulator referenčnih veličin vzbuja napravo na predpisani način, sistem pa primerja dobljene odzive z referenčnimi. Testira se tako strojna kot tudi programska oprema naprave.

Opisani sistem se je izkazal za zelo uspešnega pri izvajanju regresijskih in stresnih testov med razvojem naprave, kot tudi štiriminutnih testov med proizvodnjo. Še zlasti je nepogrešljiva uporaba sistema pri testiranju časovnih odvisnosti merjenih veličin avtomobilske diagnostične naprave. Tega dela testa ročno ni mogoče izvesti, saj človeško oko ni zmožno zaznati hitrih sprememb merjene veličine (napetosti, toka, tlaka itd.), medtem ko opisani sistem avtomatskega testiranja zazna in ustrezno dokumentira že eno samo napačno točko v časovni odvisnosti merjene veličine. O koristnosti razvitega sistema nam dovolj pove tudi podatek, da z njim v štirih minutah izvedemo nabor testov, ki bi jih operater ročno izvajal devetnajst ur, kar je v proizvodnji nedopustno.

**Ključne besede:** avtomatsko testiranje, avtomobilsko testiranje, skriptni jezik, regresijski test, testiranje v proizvodnji

## Automatic testing of automotive diagnostic devices

The paper focuses on automatic testing of automotive diagnostic devices. The test system consists of ETSL (Embedded Testing Script Language) script language used for writing test procedures, UETS (Universal Embedded Testing System) software executing the ETSL code and controlling the automotive diagnostic device, and reference-value emulator emulating the engine parameters. The test execution runs under control of a personal computer. During testing the automotive diagnostic device sends the results measurement to UETS to have them compared with the expected values. The system tests the firmware and hardware of the device under test and successfully runs regression and stress tests during development, as well as four-minute quick tests during production of automotive diagnostic devices. The performed evaluation shows indispensability of the system since it comprehensively tests the automotive diagnostic devices and enables the test scenarios that a human operator cannot execute manually due to limitations of the human eyes. The main achievement is shortening of the quick test in production line to 4 minutes instead of 19 hours when run manually.

## 1 UVOD

Avtomobilske diagnostične naprave merijo temperaturo motornega olja, tlak oljne tlačilke motorja, električne

tokove in napetosti v motorju ter druge njegove parametre. Delujejo kot digitalni osciloskop ali digitalni multimeter, pri čemer je v prvem primeru iz meritev razviden časovni potek merjenih veličin, medtem ko je v drugem primeru uporabniku na voljo zgolj presek merjenega stanja v določenem trenutku. Izmerjene vrednosti in časovni poteki signalov so podlaga za sklepanje o stanju in napakah motorja, zato so te naprave nepogrešljive pri servisiranju avtomobilov.

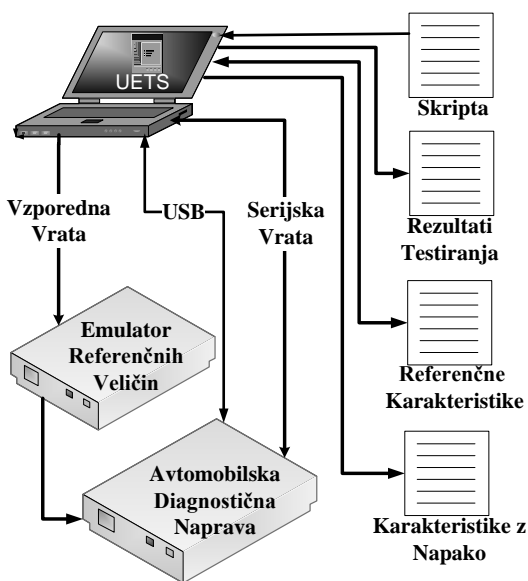
Avtomobilske diagnostične naprave je treba testirati tako med njihovo utečeno proizvodnjo kot pri samem razvoju. V ta namen smo razvili sistem njihovega avtomatskega testiranja, ki vključuje širok spekter testnih scenarijev.

Pri testiranju proizvedenih diagnostičnih naprav sistem izvede ustrezne teste v štirih minutah, kot zahteva naročnik, pri čemer je bil naš cilj, da je v omenjenem časovnem intervalu naprava izpostavljena čim večjemu številu predpisanih testnih scenarijev.

Pri testiranju diagnostičnih naprav med njihovim razvojem preverjamo delovanje funkcionalnosti, dodanih v določenem koraku razvoja, ter izvajamo regresijsko in stresno testiranje. Pri zadnjem je med 72-urnim neprekinjenim delovanjem diagnostična naprava izpostavljena hitremu preklapljanju merilnih območij,

priklapljanju in odklapanju merilnih sond ter drugim operacijam.

Teste izvaja sistem (slika 1), ki vsebuje uporabniško prijazno in nadgradljivo programsko opremo UETS (Universal Embedded Testing System), emulator referenčnih veličin in skripte v jeziku ETSL (Embedded Testing Script Language), ki smo ga razvili sami. Paket UETS komunicira z avtomobilsko diagnostično napravo prek serijskih vrat in USB-vmesnika, z emulatorjem referenčnih veličin pa prek vzporednih vrat Centronics. Zasnovan je tako, da se po potrebi lahko dodajajo novi vmesniki. Omogočeno je tudi testiranje s ključnimi besedami [7], pri čemer ključna beseda določa ime skripte, ki se pri tem izvede. Testiranje s ključnimi besedami zniža stroške vzdrževanja skript in pohitri implementacijo, ker dele skripte, kjer se test ponovi, zapakiramo v svojo skripto in jo kličemo z več mest v skriptah, podobno klicanju funkcij v programskih jezikih C/C++ in Java. Tako imamo preglednejše in krajše skripte ter jih je tudi lažje in ceneje vzdrževati. Tak pristop omogoča implementacijo na tisoče testov z le nekaj sto skript [7].



Slika 1: Sistem avtomatskega testiranja avtomobilskih diagnostičnih naprav

Za lasten razvoj programske opreme UETS smo se odločili zaradi fleksibilnosti, neodvisnosti od drugih razvijalcev in možnosti njenih hitrih popravkov, s čimer manjšamo verjetnost izpadov proizvodnje avtomobilskih diagnostičnih naprav. Razvili smo tudi napravo za emulacijo vklopa in izklopa sond ter emulacijo referenčnih veličin, s čimer smo avtomatizirali celoten postopek testiranja, kar omogoča popolnoma avtomatizirano izvajanje testov diagnostične naprave brez interakcije operaterja.

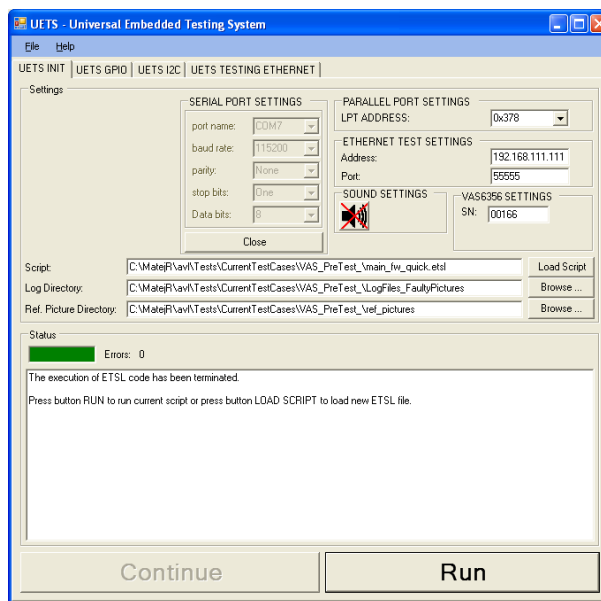
Razviti sistem opravi proizvodni test diagnostične naprave v štirih minutah, medtem ko ekvivalentno testiranje, ki ga izvede operater, traja 19 ur. Stresni test,

ki zdaj traja 72 ur, bi operater brez naše naprave izvajal 11 let pri osemurnem delovniku in petih delovnih dneh v tednu.

V nadaljevanju so pregledani gradniki opisanega sistema. Sledi razlaga metod primerjanja odzivov, beleženja rezultatov in avtomatizacije, ter zaključki in spoznanja pri uporabi naprave v realnem okolju.

## 2 PROGRAMSKI PAKET UETS

Na sliki 2 vidimo prikazovalni obrazec paketa UETS. Zgornji del vsebuje nastavitve, spodnji del pa prikazuje trenutni status testa, ki se izvaja.



Slika 2: Prikazovalni obrazec paketa UETS

V nastavitvah (slika 2) lahko nastavimo parametre za serijska vrata, vzporedna vrata, povezavo Ethernet in zvok. Če vključimo zvok, UETS po končanem izvajanju testov zapiska. Omenjena funkcionalnost je uporabljena v proizvodnji, kjer operater med izvajanjem testov dela kaj drugega. V prikazovalni obrazec vpišemo tudi serijsko številko diagnostične naprave, ki jo trenutno testiramo, izberemo pot do skripte, ki jo želimo izvajati, direktorij, kamor naj se shranjujejo rezultati testiranja, in direktorij z referenčnimi karakteristikami.

Statusni del prikazovalnega obrazca vsebuje zeleno lučko, ki signalizira odsotnost napak, in rdečo lučko, ki zasveti ob vsaj eni odkriti napaki. Pomembna sta še števec napak in statusna vrstica, ki opisuje trenutno dogajanje.

Prikazovalni obrazec paketa UETS vsebuje tudi tipko RUN, s katero zaženemo izvajanje testnih scenarijev, tipka CONTINUE pa postane aktivna, ko program izvaja ukaz WAIT, ki ga lahko uporabimo za prekinitveno točko.

Primer:  
wait 1000000

Zgornji ukaz čaka tisoč sekund. Če v tem intervalu pritisnemo tipko CONTINUE, program preneha s čakanjem in nadaljuje izvajanje. Z dolgimi premori simuliramo prekinitvene točke, ki jih izkoristimo za iskanje napak na avtomobilski diagnostični napravi, opravljanje meritev itd.

Paket UETS izvaja naslednje aktivnosti:

- Dinamično analizo (dynamic analysis [7, 2]), kjer ocenjuje diagnostično napravo med njenim testiranjem. Primer dinamične analize je ocenjevanje odtokanja pomnilnika (memory leak).
- Testiranje zmogljivosti avtomobilске diagnostične naprave (performance testing [7, 2]) prek meritev odzivnih časov diagnostične naprave, časov obhoda sporočil itd.
- Testiranje obremenjenosti (load testing [7, 2]) s povečanim prometom sporočil z diagnostično napravo, hitrejšim preklapljanjem med merilnimi območji itd. Tako se preverijo meje zmogljivosti diagnostične naprave.
- Izvajanje testnih scenarijev in primerjave rezultatov z referenčnimi vrednostmi (execution and comparison [7]).

Paket UETS tolmači in izvaja ukaze, zapisane v skripti jezika ETSL (sekcija 3). Z diagnostično napravo komunicira prek vodila USB in serijskih vrat, z emulatorjem referenčnih veličin pa prek vzporednih vrat (slika 1). Drugih vmesnikov pri testiranju avtomobilskih diagnostičnih naprav ne uporabljamo.

Paket UETS omogoča modularno pisanje preglednih testnih scenarijev v jeziku ETSL. Ime skripte, ki jo kličemo, je ključna beseda. Testiranje s ključnimi besedami [7] zniža stroške vzdrževanja skript, pohitri implementacijo in omogoča implementacijo na tisoče testov z le nekaj sto skript. Glavna skripta določa, katere skripte naj paket UETS izvaja.

Primer testa, kjer so ključne besede *set\_scope\_channel\_A\_dso1\_range\_16\_dc*, *CheckError* in *ReadSerialPort*, je naslednji:

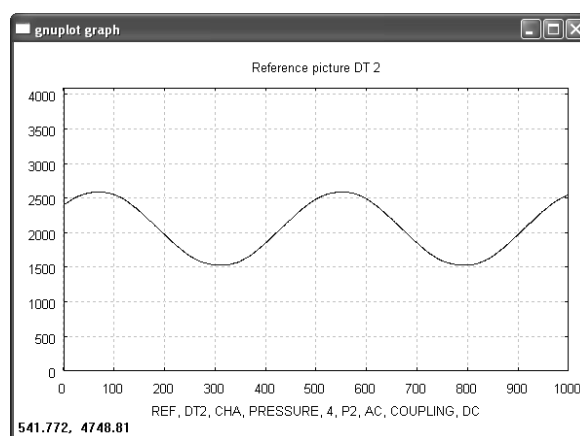
1. execute "set\_scope\_channel\_A\_dso1\_range\_16\_dc.etsl"
2. execute "CheckError.etsl"
3. execute "ReadSerialPort.etsl"
4. wait 100

V prvi vrstici izberemo merilno območje, v drugi vrstici preverimo napake, v tretji preberemo znakovni niz prek serijskih vrat, nakar v četrti vrstici počakamo 100 ms.

Sistem omogoča tudi, da pri analizi vzroka napake s paketom UETS ponovno izvedemo le tisto podskripto, ki vzbudi napako v vdelani programski ali strojni opremi diagnostične naprave, ne pa glavne skripte, s katero paket UETS izvede celoten test. S tem skrajšamo čas, potreben za ponovitev nastopa napake.

Po vsaki spremembi vdelane programske ali strojne opreme diagnostične naprave, ki naj bi napako odpravila, s paketom UETS poženemo izvajanje ustrezne podskripte, da ovrednotimo uspeh spremembe. Kot primer navedimo, da en cikel regresijskega testa diagnostične naprave traja približno 15 minut, skripta, ki ponovi napako, pa v povprečju traja 30 sekund. Če vzrok napake odpravimo npr. v četrtem poskusu, bi štirikratna ponovitev celotnega regresijskega testa trajala 60 minut, štirikratna ponovitev podskripte pa le eno minuto.

Paket UETS uporablja dva tipa pričakovanih odzivov. Prvi tip je časovna odvisnost izmerjenih vrednosti (slika 3) in ga UETS generira sam. Drugi tip pričakovanega odziva je izmerjena vrednost, ki jo razvijalec testnih scenarijev ročno vpiše v skripto.



Slika 3: Primer referenčne karakteristike meritve tlaka

Pričakovane odzive prvega ali drugega tipa pripravimo z delujočo in kalibrirano diagnostično napravo. Pričakovane odzive, ki jih generira paket UETS, nato ročno skrbno pregledamo.

### 3 SKRIPTNI JEZIK ETSL

Skriptni jezik ETSL omogoča hitro in učinkovito pisanje testnih scenarijev. Za komunikacijo z diagnostično napravo in emulatorjem referenčnih veličin ni treba poznati tehničnih podrobnosti uporabljenih vodil. Za razvoj lastnega skriptnega jezika ETSL smo se odločili, ker nam to omogoča dodajanje novih ukazov, ki jih potrebujemo za učinkovito testiranje avtomobilске diagnostične naprave.

V nadaljevanju podajamo primere nekaterih ukazov.

- Pošiljanje treh zlogov avtomobilski diagnostični napravi prek vmesnika USB:  
**usb write 0x01 0x02 0x03.**
- Branje znakovnega niza prek serijskih vrat in shranjevanje vrednosti v spremenljivko *sv\_data*:  
**sport read sv\_data.**

- Shranjevanje znakovnega niza iz spremenljivke sv\_data v datoteko:  
`log "%s" sv_data.`

Tolmač skriptnega jezika pred njeno izvedbo preveri napake v kodi, ki nam jih sporoči v obliki deskriptivnih sporočil. Če je koda brez napak, se v prikazovalnem obrazcu paketa UETS prikaže gumb RUN, s katerim poženemo izvajanje programa.

#### 4 EMULATOR REFERENČNIH VELIČIN

Sonde, ki jih uporablja avtomobilska diagnostična naprava, so pretvorniki merjene veličine v tok, napetost ali upornost. Na tej podlagi smo razvili emulator referenčnih veličin, ki generira izmenične in enosmerne napetosti in tokove ter različne vrednosti upornosti. Tako smo emulirali realne pogoje delovanja avtomobilske diagnostične naprave (merjenje temperature, tlaka, visokih napetosti do velikostnega reda 1000 V in tokove do 100 A).

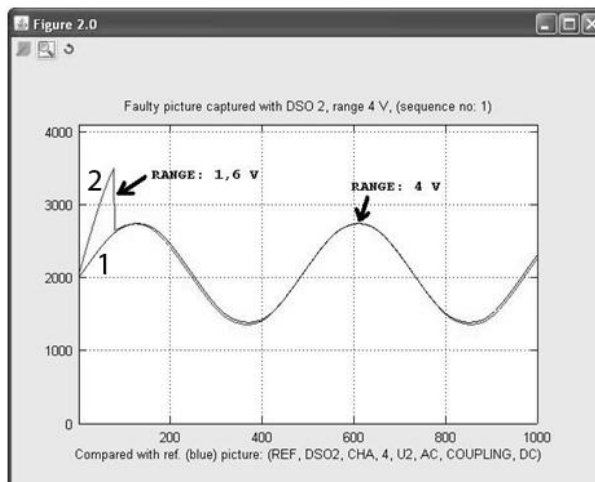
Največji prihranek časa in sredstev smo pridobili z emulacijo temperature motornega olja, saj je mogoče emulirati odziv sonde na poljubno temperaturo motornega olja v manj kot 1 s, kar bi bilo z resničnim segrevanjem in ohlajanjem olja nemogoče doseči. Poleg tega sistemu ni treba dodati grelnikov in hladilnikov za segrevanje in ohlajanje olja.

Vsaka sonda ima svojo karakteristično upornost, prek katere jo diagnostična naprava prepozna. Te upornosti emulator referenčnih veličin z releji prikloplja in odkloplja, s čimer emulira vklope in izklope sond. Na diagnostično napravo prikloplja senzor temperature, senzor tlaka, 100 A tokovno sondo, 1000 V sondo, dve napetostni sondi in sondo za proženje.

#### 5 TESTIRANJE AVTOMOBILSKE DIAGNOSTIČNE NAPRAVE

Paket UETS prek vzporednih vrat krmili emulator referenčnih veličin, s katerim emulira priklope in izklope ustreznih sond ter vzbuja (tok, napetost, tlak in temperaturo). Prek povezave USB sproži delovanje avtomobilske diagnostične naprave v načinu digitalnega osciloskopa ali digitalnega multimetra.

V načinu osciloskopa diagnostična naprava pošilja karakteristike (merjena veličina po času) paketu UETS, ki jih primerja z referenčnimi karakteristikami in v primeru prevelikega odstopanja med njimi shrani napačne karakteristike. Slika 4 prikazuje primer referenčne karakteristike (krivulja 1) in napačne karakteristike (krivulja 2). V prikazanem primeru je avtomobilska diagnostična naprava prepozno preklpila z merilnega območja 1,6 V na merilno območje 4 V.



Slika 4: Primer referenčne karakteristike (krivulja 1) in napačne karakteristike (krivulja 2)

V načinu digitalnega multimetra diagnostična naprava paketu UETS pošilja informacije o izmerjeni veličini, ki jo UETS primerja z referenčno veličino, in v primeru napake, ki je večja od dopustne, odstopanje zabeleži v datoteko.

#### 6 PRIMERJAVA REZULTATOV MERITEV

Avtomobilska diagnostična naprava deluje kot digitalni osciloskop ali kot digitalni multimeter. V prvem primeru paket UETS izvaja primerjavo karakteristik z računanjem metrike (1), v drugem primeru pa paket UETS izvaja primerjavo izmerjenih vrednosti z uporabo evklidske metrike [4–6].

##### 6.1 Primerjava karakteristik

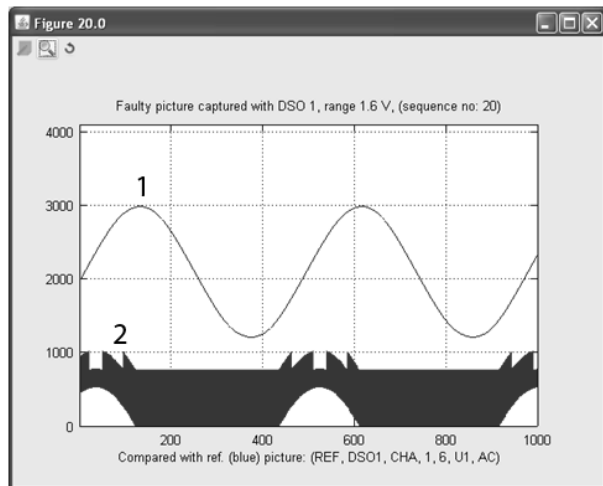
Pri primerjavi karakteristik gre za primerjavo dveh nizov merjenih veličin po času (trenutnega z referenčnim). Paket UETS izvaja primerjavo omenjenih karakteristik z računanjem metrike v funkcijskem prostoru [4–6] med referenčno in dobljeno karakteristiko po enačbi (1).

$$d(f, g) = \sup_{x \in [a, b]} |f(x) - g(x)| \quad (1)$$

$d(f, g)$  je metrika med funkcijami  $f$  in  $g$ , pri čemer je  $f$  funkcija referenčne karakteristike in  $g$  funkcija dejanske karakteristike. Metrika je definirana na zaprtem intervalu od  $a$  do  $b$ . Če je razdalja med karakteristikama  $d(f, g)$  večja od pričakovane, paket UETS shrani napačno karakteristiko. Primer take karakteristike je krivulja 2 na sliki 5.

Število točk, iz katerih je sestavljena karakteristika, je mogoče spreminjati, vendar smo se pri naših testnih scenarijih omejili na tisoč točk. Vsaka točka je 12-bitni podatek, torej metrika primerja tisoč 12-bitnih podatkov na karakteristiko. Če je npr. že tretji podatek napačen, je to za UETS že napačna karakteristika, zato preneha z nadaljnjo primerjavo točk in napačno karakteristiko

shrani. Tako pridobimo na hitrosti izvajanja testnega scenarija. Nato UETS nadaljuje s primerjavo naslednje karakteristike. Na en test se preveri od 10 do 20 karakteristik, vendar se v primeru napačnih karakteristik shrani le prvih 10 napačnih, da se v primeru napak na disk ne bi shranilo preveč podatkov. Krivulja 1 na sliki 5 pomeni referenčno karakteristiko, medtem ko krivulja 2 pripada napačni karakteristiki.



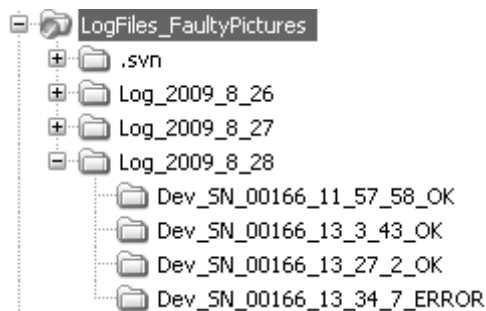
Slika 5: Primer referenčne karakteristike (krivulja 1) in napačne karakteristike (krivulja 2)

## 6.2 Primerjava izmerjenih vrednosti

Za primerjavo izmerjene vrednosti z referenčno vrednostjo paket UETS računa metriko (razdaljo) med izmerjeno vrednostjo in referenčno vrednostjo z uporabo evklidske metrike [4–6] v enodimenzionalnem prostoru realnih števil. Če je razdalja med izmerjeno vrednostjo in referenčno vrednostjo večja od pričakovane, paket UETS zabeleži opis napake v datoteko.

## 7 BELEŽENJE REZULTATOV

Slika 6 prikazuje organizacijo map, ki vsebujejo rezultate avtomatskega testiranja. Mapo na najvišjem nivoju izbere uporabnik v prikazovalnem obrazcu paketa UETS (na sliki 6 je to mapa LogFiles\_FaultsPictures). Druge nivoje map (slika 6) generira paket UETS sam. En nivo nižje je mapa z datumom. Ob vsakem novem testu paket UETS generira mapo na najnižjem nivoju. Ime te mape vsebuje serijsko številko diagnostične naprave, čas začetka testiranja in informacijo o uspešno opravljenem testu. Paket UETS v to mapo shrani datoteke, ki vsebujejo informacijo o postopku in rezultatih testiranja.



Slika 6: Organizacija map, ki vsebujejo datoteke z rezultati

Imamo dva tipa shranjenih datotek, pri čemer prvi tip vsebuje okvirne, drugi pa podrobne podatke o testiranju. Slika 7 prikazuje primer datoteke, v katero paket UETS beleži okvirne rezultate testiranja. V vrstici 1 UETS pošlje ukaz SCOPE START avtomobilski diagnostični napravi, s čimer se merjenje začne. Naprava začne pošiljati karakteristike paketu UETS (vrstice 3–20). V vrstici tri vidimo DSO ERROR, ker je druga karakteristika napačna (razlika med referenčno in prejeta karakteristiko je večja od pričakovane). Podobno imamo napačno karakteristiko v vrsticah 11 in 14. V vrstici 21 ukaz SCOPE STOP povzroči prenehanje merjenja.

```

1 26.03.2012 07:44:05 SCOPE START ->
2 26.03.2012 07:44:05 <- OK
3 26.03.2012 07:44:05 <- MSGID_SCOPE_DATA_AVERAGE (152)
4 26.03.2012 07:44:05 <- MSGID_SCOPE_DATA_AVERAGE (153)
5 26.03.2012 07:44:05 DSO ERROR (socket_id: 5, sensor_id: 2)
6 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (154)
7 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (155)
8 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (156)
9 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (157)
10 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (158)
11 26.03.2012 07:44:06 DSO ERROR (socket_id: 5, sensor_id: 2)
12 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (159)
13 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (160)
14 26.03.2012 07:44:06 DSO ERROR (socket_id: 5, sensor_id: 2)
15 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (161)
16 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (162)
17 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (163)
18 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (164)
19 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (165)
20 26.03.2012 07:44:06 <- MSGID_SCOPE_DATA_AVERAGE (166)
21 26.03.2012 07:44:06 SCOPE STOP ->

```

Slika 7: Primer datoteke z rezultati testiranja

## 8 EMPIRIČNI TESTI IN REZULTATI

Avtomatizirali smo dve aktivnosti testiranja, in sicer izvajanje in primerjavo. Podajamo primer štiriminutnega testnega scenarija, ki se izvaja v proizvodnji diagnostičnih naprav.

Na začetku testni scenarij shrani:

- strojno verzijo glavne plošče,
- strojno verzijo čelne plošče,
- strojno verzijo napajalnika,
- strojno verzijo prve in druge napetostne sonde,

- verzijo vdelane programske opreme glavne plošče in napajalnika,
- serijsko številko avtomobilske diagnostične naprave,
- verzijo TPU (Time Process Unit),
- verzijo FPGA,
- verzijo EEPROM,
- leto proizvodnje,
- serijsko številko napajalnika,
- leto izdelave in status napajalnika,
- informacije o priklopljenih sondah ter njihove verzije in serijske številke.

Sledi test, ki preveri pravilnost FPGA vezja.

Nato napravo preizkusimo v režimu digitalnega pomnilniškega osciloskopa, kjer se preverijo karakteristike 100 A tokovne sonde, dveh napetostnih sond, dveh temperaturnih senzorjev, dveh senzorjev tlaka in 1000 V napetostne sonde.

Testi karakteristik za zgoraj omenjene sonde so si med seboj podobni, zato opišimo le test karakteristik za senzor tlaka. Paket UETS pošlje ukaz:

- emulatorju referenčnih veličin, naj emulira zeleni tlak,
- avtomobilski diagnostični napravi, naj izbere kanal za merjenje karakteristike tlaka,
- avtomobilski diagnostični napravi, naj nastavi nivo proženja, tip sklopitve in izbere merilno območje,
- avtomobilski diagnostični napravi, naj nastavi frekvenco vzorčenja in število odčitanih točk,
- avtomobilski diagnostični napravi, naj začne meriti in pošiljati karakteristike paketu UETS, ki jih začne primerjati,
- avtomobilski diagnostični napravi, naj preneha meriti.

Paket UETS rezultate zgoraj omenjenega poteka shranjuje v datoteko. Nato se test ponovi z drugo vrednostjo tlaka. Enak postopek se izvede tudi za senzor na drugem kanalu.

Sledi test, kjer avtomobilska diagnostična naprava deluje kot digitalni multimeter. Primer testa z napetostno sondo je naslednji. Paket UETS pošlje ukaz:

- emulatorju referenčnih veličin, naj priklopi zeleno izmenično napetost,
- avtomobilski diagnostični napravi, naj izbere kanal in način meritve (npr. merjenje izmenične napetosti),
- avtomobilski diagnostični napravi, naj požene meritev in primerja dobljeni rezultat z referenčno vrednostjo.

Paket UETS izvajanje zgoraj omenjenega poteka shranjuje v datoteko.

Paket UETS nato ponovi test z drugo vrednostjo napetosti. Enak postopek izvede tudi za enosmerne napetosti in druga merilna območja. Podobni so testi za preostale sonde.

Meritve časa izvajanja testov kažejo, da smo dosegli faktor produktivnosti [1] 282. To pomeni, da bi štiriminutni avtomatski test avtomobilske diagnostične naprave operater izvajal  $282,5 \cdot 4 = 1130$  minut (19 ur), kar je v proizvodnji nedopustno.

$$fp = \frac{\text{čas\_ro\cno\_testiranja}}{\text{čas\_avtomatskega\_testiranja}} \quad (2),$$

kjer je fp faktor produktivnosti.

Na leto se proizvede okvirno 2400 avtomobilskih diagnostičnih naprav. Čas, ki ga prihranimo pri testiranju ene avtomobilske diagnostične naprave, je približno 1130 minut. Če bi te teste izvajal operater z urno postavko 10 EUR/h, bi to na leto pomenilo približno 500.000 evrov.

Med razvojem avtomobilske diagnostične naprave smo opravljali tudi 72-urne stresne teste. Omenjeni test bi operater izvajal  $282,5 \cdot 72 \cdot 60 = 1220400$  minut, kar je približno pri 8-urnem delovniku in petih dneh na teden 10 let. Če bi operaterja plačali 10 EUR/h, bi stresni test pomenil strošek okoli 200.000 evrov. Tudi v tem primeru se je testna avtomatizacija izkazala za nepogrešljivo.

## 9 SKLEP IN RAZPRAVA

Paket UETS omogoča dinamično analizo in testiranje zmogljivosti avtomobilske diagnostične naprave, izvajanje testov, primerjanje dobljenih odzivov s pričakovanimi odzivi ter beleženje poteka testov in rezultatov testiranja na disk. Kot prikazuje slika 6, so rezultirajoče datoteke organizirane tako, da je omogočeno hitro iskanje po preteklih rezultatih testiranja.

Ker proizvodnja in razvoj diagnostičnih naprav zahtevata pogosto izvajanje enakih testov in primerjavo dejanskih odzivov s pričakovanimi odzivi, smo ti dve aktivnosti avtomatizirali, kar priporočajo tudi v [7]. V proizvodnji traja izvajanje avtomatskega testiranja posamezne avtomobilske diagnostične naprave štiri minute. Z ročnim testiranjem bi iste teste izvajali več ko 19 ur, kar je nesprejemljivo.

Sistem avtomatskega testiranja se je izkazal za nepogrešljivega tudi pri testiranju časovnih odvisnosti merjenih veličin. Tega dela testa ročno ni mo izvesti, ker človeško oko ne zaznava hitrih sprememb merjene veličine.

Razvoj lastnega sistema avtomatskega testiranja nam omogoča fleksibilnost, neodvisnost od drugih razvijalcev programske opreme in možnost hitrih popravkov testnega sistema, s čimer preprečimo izpade

proizvodnje avtomobilskih diagnostičnih naprav. Razvoj emulatorja referenčnih veličin je bil nujen, saj smo le z njim lahko izvedli popolno avtomatizacijo, brez človeških interakcij med izvajanjem testov, in tako dosegli visok faktor produktivnosti [1]. Torej lahko povzamemo, da je avtomatizacija testiranja avtomobilskih diagnostičnih naprav popolnoma upravičena.

### LITERATURA

- [1] Linda G. Hayes, The Automated Testing Handbook.
- [2] Bart Broekman, Edwin Notenboom, Testing Embedded Software, Addison – Wesley, GreatBritain 2003, ISBN: 0 321 15986 1.
- [3] Michael L. Bushnell, Vishwani D. Agrawal, Essentials of Electronic Testing, Kluweracademic publishers, New York 2002, ISBN: 0 306 47040 3.
- [4] Michael O. Searcoid, Metric Spaces, Springer, London 2007, ISBN: 978 84628 627 8.
- [5] Conrad Colin Adams, David Robert Franzosa, Introduction to Topology: pure and applied, Pearson Prentice Hall, Upper Saddle River 2008, ISBN: 978 0 13 184869 6.
- [6] Mariano Giaquinta, Giuseppe Modica, Mathematical Analysis, Birkhäuser, Boston 2007, ISBN: 978 0 8176 4514 4.
- [7] Mark Fewster, Dorothy Graham, Software Test Automation: effective use of test execution tools, ACM Press, New York 1999, ISBN: 0 201 33140 3.

**Matej Rakovec** je diplomiral leta 2007 na Fakulteti za elektrotehniko v Ljubljani. Zaposlen je kot razvojni inženir v podjetju ComTrade, d. o. o., v Ljubljani. Njegova raziskovalna zanimanja vključujejo razvoj testnih in vgrajenih sistemov.

**Boštjan Murovec** je izredni profesor na Fakulteti za elektrotehniko Univerze v Ljubljani. Diplomiral, magistriral in doktoriral je v letih 1996, 1999 in 2002, vse na Fakulteti za elektrotehniko. Njegova raziskovalna področja so strojni vid, bioinformatika, kombinatorične optimizacije in vgrajeni sistemi.