



**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**Факултет Електронна Техника и Технологии**

**Катедра Микроелектроника**

**Маг. инж. Николай Петев Браянов**

**АВТОМАТИЗАЦИЯ НА ПРОЦЕСА НА РАЗРАБОТКА И ИНТЕГРИРАНЕ  
НА HARDWARE-IN-THE-LOOP СИСТЕМИ ЗА АВТОМОБИЛНАТА  
ИНДУСТРИЯ**

**А В Т О Р Е Ф Е Р А Т**

на дисертация за придобиване на образователна и научна степен  
**"ДОКТОР"**

Област: 5. Технически науки

Професионално направление: 5.2 „Електротехника, електроника и  
автоматика“

Научна специалност: Микроелектроника

**Научен ръководител: Проф. д-р Анна Стойнова**

СОФИЯ, 2019 г.

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „Микроелектроника“ към Факултет „Електронна Техника и Технологии“ на ТУ-София на редовно заседание, проведено на 28.10.2019 г..

Публичната защита на дисертационния труд ще се състои на 10.03.2020 г. от 13:00 часа в Конферентната зала на БИЦ на Технически университет – София на открито заседание на научното жури, определено със заповед № ОЖ-5.2-115/11.11.2019 г. на Ректора на ТУ-София в състав:

1. Доц. д-р Георги Ангелов – ТУ-София, София –председател
2. Проф. д-р инж. Анна Стойнова – ТУ-София, София – научен секретар
3. Проф. д-р инж. Рачо Иванов – ТУ-София, София
4. Проф. д-р инж. Иван Евстатиев – Русенски университет „Ангел Кънчев“, Русе
5. Доц. д-р инж. Димитър Токмаков – Пловдивския университет „Паисий Хилендарски“, Пловдив

Рецензенти:

1. Проф. д-р инж. Иван Евстатиев
2. Доц. д-р Георги Ангелов

Материалите по защитата са на разположение на интересуващите се в канцеларията на Факултет „Електронна Техника и Технологии“ на ТУ-София, блок №1, кабинет № 1355.

Дисертантът е редовен докторант към катедра „Микроелектроника“ на факултет „Електронна техника и технологии“. Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Николай Петев Браянов

Заглавие: Автоматизация на процеса на разработка и интегриране на НІІ системи за автомобилната индустрия

Тираж: 30 броя

Отпечатано в ИПК на Технически университет – София

# **I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД <sup>1</sup>**

---

## **Актуалност на проблема**

След индустриалната революция, последните пет десетилетия в живота на хората са белязани от навлизането на изчислителната техника. Беше изминат дългия път между скъпи, големи и недотам ефективни платформи до съвременните бързи, многоядрени, с висока степен на интеграция системи, вградени в един чип.

Комбинирането на софтуер, хардуер и механика в мехатронна комплексна система, работеща в реално време, превърна валидирането на съвременните устройства в сложна, а понякога и невъзможна операция, заемаща голяма част от процеса на разработка. Прилагането на такъв тип системи в индустрии с изисквания за доказано ниво на безопасност, като самолетостроене, автомобилостроене и други, и невъзможността те да бъдат пълнофункционално тествани в реални условия, превърна проблема в първостепенен. Единственото реално и приложено решение към момента са така наречените *hardware-in-the-loop* (HIL) системи. В пряк превод, HIL означава „хардуер в контур“ или „хардуер в цикъл“, като този термин визуализира възможността за изпитване на хардуерни системи, чрез симулиране на входните системни сигнали на база управляващите изходни сигнали и модел на средата, максимално доближавайки истинския режим на работа на системата.

Липсата на стандартен метод за реализиране и интегриране на HIL системи, води до специфични изисквания към персонала, неизвестно дълъг период на интеграция, и в резултат висока цена - значими фактори, затрудняващи масовата интеграция. В тази връзка, към момента HIL системите са използвани изключително за системи, за чието тестване не съществува друга методика.

Широкото прилагане на HIL би довело до автоматизиране и намаляване на цената на тестването, както и до повишаване на качеството на продуктите. В тази връзка е необходим автоматизиран подход за разработване и интегриране на HIL системи

## **Цел на дисертационния труд, основни задачи и методи за изследване**

Целта на дисертационния труд е да се създаде методология за автоматизиране на процеса на разработка и интегриране на *hardware-in-the-loop* системи за автомобилната индустрия.

Основните задачи за постигане на поставената цел са:

1. Да се анализира възможността за автоматично генериране на програмен код, както и неговото използване за тестване на системи, в съответствие със стандартите в автомобилната индустрия.
2. Да се създаде универсална системна архитектура, способна да реализира неограничено комплексна HIL система чрез алгоритъм, позволяващ автоматизирането на разработката.
3. Да се създаде универсална софтуерна архитектура, позволяваща автоматично реализиране на HIL система, без ограничения във връзка с нейната комплексност
4. Да се разработи методология, позволяваща теоретичното изследване на работоспособността на системата по автоматизиран начин.
5. Експериментално изследване на приложимостта на предложената система и нейните ограничения

## **Научна новост**

В дисертационния труд е изследвана възможността за автоматизиране на процеса на разработване и интеграция чрез моделно базирания подход.

*Разглеждана е възможността да бъде използван модел на HIL система, на база на който автоматично да се генерира програмен код, управляващ HIL система за автомобилната индустрия.*

*Създадена е универсална и с възможност за разширяване системна архитектура. В този контекст е създадена софтуерна архитектура, отговаряща на изискванията за HIL система и даваща възможност за универсален и автоматизиран подход за разработка.*

*Създаден е модел на работоспособността на участващите в решението подсистеми. На тази база е съставен автоматизиран подход за изследване на работоспособността.*

### **Практическа приложимост**

*Предложената система е приложима за автоматизиране на процеса за разработване и интегриране на HIL системи за автомобилната индустрия.*

### **Апробация**

*Приложимостта на предложената система е доказана чрез интеграция във връзка с тестване на отоплител на седалки за автомобилната индустрия.*

*Решението не е ограничено по отношение индустрия на приложение. В тази връзка то е интегрирано за изследване на микроконтролерна система, посредством методите на инфрачервената термография.*

*Резултатите от изследванията са представени на научни форуми с международно участие.*

### **Публикации**

*Основните постижения и резултати от дисертационния труд са публикувани в 8 научни статии, от които 2 самостоятелни и 6 в колектив с други автори. Шест от статиите са на английски език, другите на български. Две от статиите са публикувани в списания, шест са представени на научни конференции.*

### **Структура и обем на дисертационния труд**

*Дисертационният труд е в обем от 132 страници, като включва увод, 5 глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията и използвана литература. Цитирани са общо 173 литературни източници, всички са на латиница. Работата включва общо 84 фигури и 21 таблици. Номерата на фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.*

## II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

### ГЛАВА 1 Съвременно състояние на процеса на разработка и интегриране на HIL системи

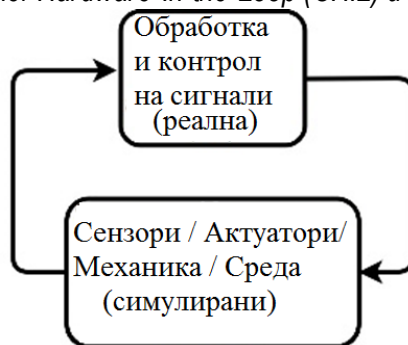
В обзорната първа глава е разгледан Hardware-in-the-loop като един съвременен метод за валидиране на комплексни системи. Той представя незаменима възможност за извършване на симулация, максимално близка до реалността, в лабораторни условия. Анализът на публикациите, разглеждащи този подход показва, че въпреки ползата от него и незаменимостта му при изследване на комплексни системи, предназначени за използване в недостъпни или животозастрашаващи условия, той все още не е унифициран. Липсват стандартен метод за реализиране и интегриране на такъв тип системи, което води до специфични изисквания към персонала, неизвестно дълъг период на интеграция и в резултат висока цена - значими фактори, затрудняващи масовата интеграция. Липсва и консенсус при дефиницията за HIL, както и съпътстващата терминология.

Анализирана е използваната терминология, като подробно са разгледани различията и причините за съществуването им. В резултат е предложена обобщена дефиниция за hardware-in-the-loop системите с типизиране на възможните интеграции.

„HIL система е неразрушителен подход за тестване, съдържащ физически контролер свързан в open- или close-loop с виртуална или частично виртуална подсистема, предоставяща много близка до действителността реплика на физическата среда и изпълняваща black / grey / white box тестове върху тестваната система.“

#### 1.5.2 HIL на сигнално ниво

Най-често sHIL фиг.1.8 се използва в случаи, в които истинските трансдюсери не са налични или тестовете не могат да бъдат изпълнени заради физически ограничения. Литературата е в консенсус относно дефиницията на този тип система, състояща се от физически реален контролер комуникиращ със симулирана подсистема. sHIL е реферирана и с други термини, като Controller-In-the-Loop (CIL) , Controller Hardware-In-the-Loop (CHIL) и дори component level HIL.

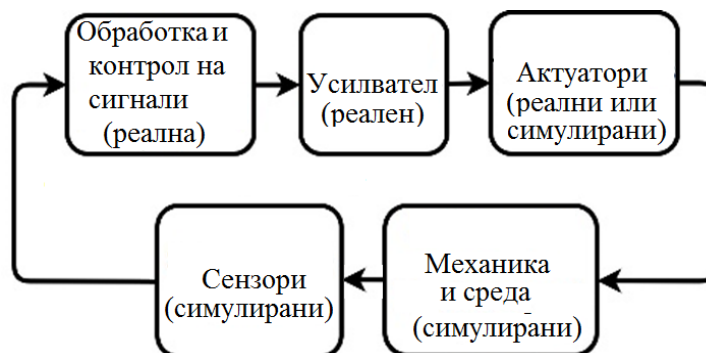


Фиг.1.8. HIL на сигнално ниво (sHIL)

#### 1.5.3 Силнотоксов HIL

rHIL фиг.1.9 предоставя възможност за по-обективно и близко до реалната система тестване. Приложението му е насочено към верификацията на мощни усилватели и силова електроника. Силнотоксовите HIL системи правят възможно тестването на реални физически силови електронни блокове в реалните условия, за които те са проектирани.

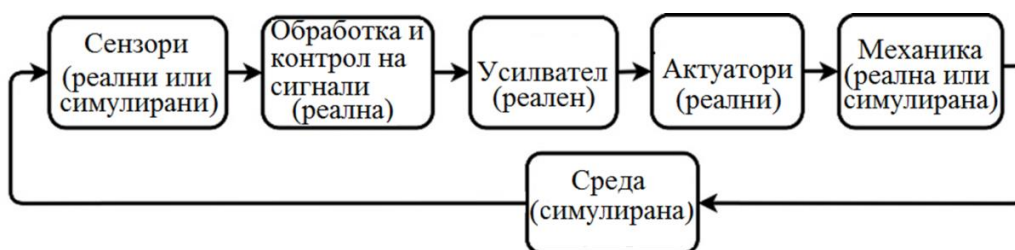
Приложението на rHIL с цел валидиране на силовите електронни блокове, задължава интегрирането на реални товари. В някои случаи могат да бъдат извлечени ползи от използването на по-малки товари, също и на мащабирани такива. Този подход е необходим при много мощни товари, чиято неправилна експлоатация може да доведе до унищожаването им. В такива случаи следва да се използва товар със сходни характеристики.



**Фиг.1.9.** Силнотоков HIL (pHIL)

#### 1.5.4 HIL на ниво механика

Максимално доближаване на тест до реалността предлага mHIL фиг.1.10. Подходът изследва поведението на реалните трансдюсери и предлага последна възможност за тестване преди продуктивния тест.

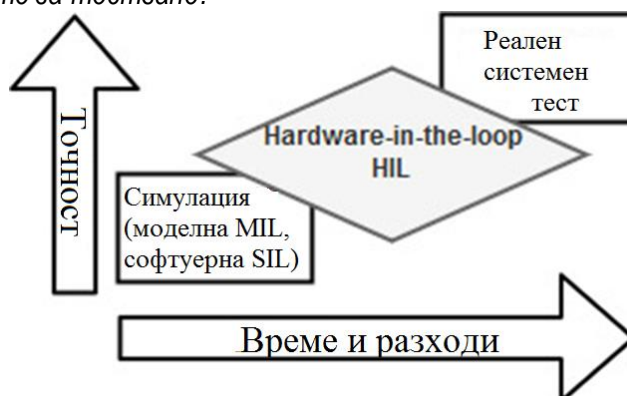


**Фиг.1.10.** HIL на ниво механика (mHIL)

Чрез анализ на резултатите в публикациите са обобщени всички използвани интеграции на HIL системи, техните преимущества и недостатъци. Във връзка с липсата на универсалност, като основен недостатък, и нуждата HIL системите да бъдат лесно добавяни в процеса на работа и интегрирани с вече използваните помощни програми, се оформи тезата, че моделно базираният подход може да даде решение на тези проблеми.

#### 1.7 Предимства и недостатъци на HIL системите

В общия смисъл HIL е метод, който дава възможност да бъдат тествани различни ситуации, запазвайки контекста максимално близо до реалността. Заемайки мястото между симулация и реална система фиг.1.16, HIL представлява компромис с точността, допринасящ за значително намаляване на разходите за тестване.



**Фиг.1.16.** Методите за тестване като комбинация от необходимо време и разходи за реализиране и точност на изпълнение

Поради полу-виртуалната си същност, HiL има многобройни предимства:

- *Повторяемост и стабилност* – За разлика от полевия тест, HiL дава възможност за тестване на пълно функционална система в лаборатория, без допълнително влияние на външни системи.
- *Ефективност от гледна точка на разходи* – Скъпите и изтощителни тестови периоди могат да бъдат значително намалени, когато HiL системата е приложена в сложни системи. Възможността за тестове в лабораторията намалява необходимото количество полеви тестове.
- *Сходни с реалността* – Комбинираната интеграция на реални и виртуални компоненти дава възможност за изследване на динамичните характеристики на подсистемите, което е невъзможно за чистите симулации.
- *Недеструктивно тестване* - HiL симулациите дават възможност за недеструктивно изследване на системи, елиминирайки възможността за повреда на скъпи компоненти.
- *Широкообхватност* - HiL системата дава възможност тестовата система да бъде симулирана чрез многобройни комбинации от интерфейсни сигнали, които трудно могат да бъдат пресъздадени в реалността при стандартното прототипиране.
- *Гъвкавост* – HiL системата може да бъде лесно модифицирана в случай на промяна на поведението на виртуалната част на системата или ако тестваната система трябва да бъде верифицирана чрез няколко различни подхода.
- *Възможност за изучаване на динамичните характеристики на системата, изследване на чувствителността и оптимизация чрез лесна промяна на параметрите*.
- *Безопасност* - HiL симулатори могат да бъдат използвани за обучение на оператори на критични системи в безопасни условия.
- *Конкурентна разработка на системи* – HiL дава възможност за тестване на подсистеми, дори и в случаите, когато цялата система не е налична, запазвайки реалността на интеграционните тестове.
- *Подпомага shift left доктрината* - HiL дава възможност за използване на последователна верига от помощни програми през целия жизнен цикъл на изделието, използвайки едни и същи тестове.
- *Намалява разходите за персонал и оборудване* – HiL автоматизацията на тестовете води до намаляване на броя на служителите, и намалява разходите за унищожено оборудване.

Заради липсата на алтернативи за такъв тип устройство, то трудно може да бъде критикувано. В публикации са посочени следните несъвършенства:

- *Не разполага с информация, относно вътрешното състояние на тестваната система.*
- *Скоростта на симулиране е константна* - HiL симулацията не може да бъде временно спряна или забавена, нито пък ускорена.
- *Няма стандартно решение, и в резултат има нужда от дълго време за интеграция* – в повечето случаи интегрирането на HiL симулатор изисква значителен обем първоначални настройки.
- *Конкретни проблеми, свързани с несъвършенство на системите за визуализация или хардуерни особености, като дължина на кабела, терминиращи резистори, обратни напрежения.*

## 1.8 Изисквания и предизвикателства пред HiL системите

Минималните функционалности, които една HiL система съдържа са изчислителна симулация и хардуерен интерфейс, работещи в ограничено реално време. Те следва да отговарят на значителен брой изисквания:

- *Изчислителна мощ* – параметърът е неизменна част от публикациите. Често той се

свързва с възможността да бъде интегрирана симулация на средата с висока точност и бързодействие подходящо за тестваната система. Честотата на изпълнение на симулацията следва да бъде под 1ms . Операционна система, работеща в реално време е посочена като част от средствата необходими за осигуряване на правилната работа на системата. Обсъжда се и факта, че увеличаването на точността на симулация е в пряка връзка с намаляването на скоростта .

- Гъвкавост – хардуерната платформа и по-специално интерфейсите следва да бъдат гъвкави, за да могат да осигурят използването на една HIL система и за нуждите на RCP. Допълнителна гъвкавост дава възможността системите да се използват за тестване на различни приложения. Възможност за това е използването на специални входно/изходни модули, например за PWM, CAN и други .
- Висока степен на автоматизация – нуждата от автоматизация е на всяко ниво на симулацията, от инжектирането на грешки до създаването на цялата HIL система .
- Цена на закупуване и интегриране на HIL система – като параметър, който винаги е от значение

Настрана от тези изисквания, съществуват редица други, свързани по-скоро с различните видове интеграции на системата или различно концептуално разбиране:

- Изчислителната платформа следва да бъде с възможност за разширяване– това би подпомогнало повторното използване на системата и/или нейните подсистеми.
- Модулна интерфейсна част би позволила ценово-оптимизирани решения .
- Интеграция с различни помощни програми – архитектурата трябва да позволява интеграцията на различни външни помощни програми, както и модели и код от други области .
- Конфигуриране чрез скриптове – дава възможност за повтаряне на тестовете чрез скрипт и свързване с различни външни програми .
- Използваните товари да бъдат същите или много подобни .
- Модулност, стандартизация и възможност за разширяване .
- Възможност за бърза интеграция, и по-важното – прогнозируемост на периода на интеграция .
- Компетентност на доставчика – познания в конкретните индустрии и технологии, опит от гледна точка интегриране и автоматизация на HIL системи [80].
- Open- and close- loop тестване .
- Функционално и нефункционално тестване .
- Налични допълнителни помощни програми за debugging на системата .
- Обвързаност на тестовете и изискванията, с цел достигане на по-добро покритие на функционалността; Възможност за извършване на регресивни тестове, с цел по-бърза и качествена валидация по време на дизайна и интеграциите .

## **ГЛАВА 2. Автоматично генериране на код за автомобилни HIL системи.**

Изследването анализира възможността за автоматично генериране на продуктов код и използването му за тестване на системи, в контекста на стандартите за безопасност в автомобилната индустрия.

Направената оценка на необходимостта за сертифициране на генерирания код във връзка с интегрирането му в HIL система показва, че такова не е приложимо. Помощната програма не е част от крайния продукт и не може да въздейства върху него, поради което във връзка със стандарта за безопасност в автомобилната индустрия ISO26262, таблица 2.1, е определено ниво

на доверие към нея TCL1. Същият стандарт определя необходимите средства за квалификация на помощна програма, като за TCL1 не е предвидено квалифициране (таблица 2.2). Потвърждение за това е факта, че разпространените към момента индустриални решения не изпълняват този критерий.

**Таблица 2.1.** Установяване нужното ниво на доверие към помощната програма

|                                   |     | Вероятност за откриване на грешките на помощната програма |            |           |
|-----------------------------------|-----|-----------------------------------------------------------|------------|-----------|
|                                   |     | TD1 голяма                                                | TD2 средна | TD3 други |
| Въздействие на помощната програма | TI1 | TCL1                                                      | TCL1       | TCL1      |
|                                   | TI2 | TCL1                                                      | TCL2       | TCL3      |

**Таблица 2.2.** Избор на методи за квалификация на помощна програма

| Методи                                                                                       | Препоръчани методи |            |
|----------------------------------------------------------------------------------------------|--------------------|------------|
|                                                                                              | за TCL2            | за TCL3    |
| 1. Доверие, придобито от употреба                                                            | ASIL A, B и C      | ASIL A и B |
| 2. Проверка на процеса на разработване на помощната програма                                 | ASIL A, B и C      | ASIL A и B |
| 3. Валидация на софтуера на помощната програма                                               | ASIL D             | ASIL C и D |
| 4. Разработване на помощната програма според стандартите за безопасност, за които се прилага | ASIL D             | ASIL C и D |

Извършена е моделно базирана разработка на код и такава чрез класическия, ръчен подход, като в резултат са оценени качествата на генерирания код, предимствата и недостатъците, които носи употребата му. За обективност на изследването, всички експерименти са извършени по три пъти, използвайки различни компилатори, след което получените резултатите са усреднени.

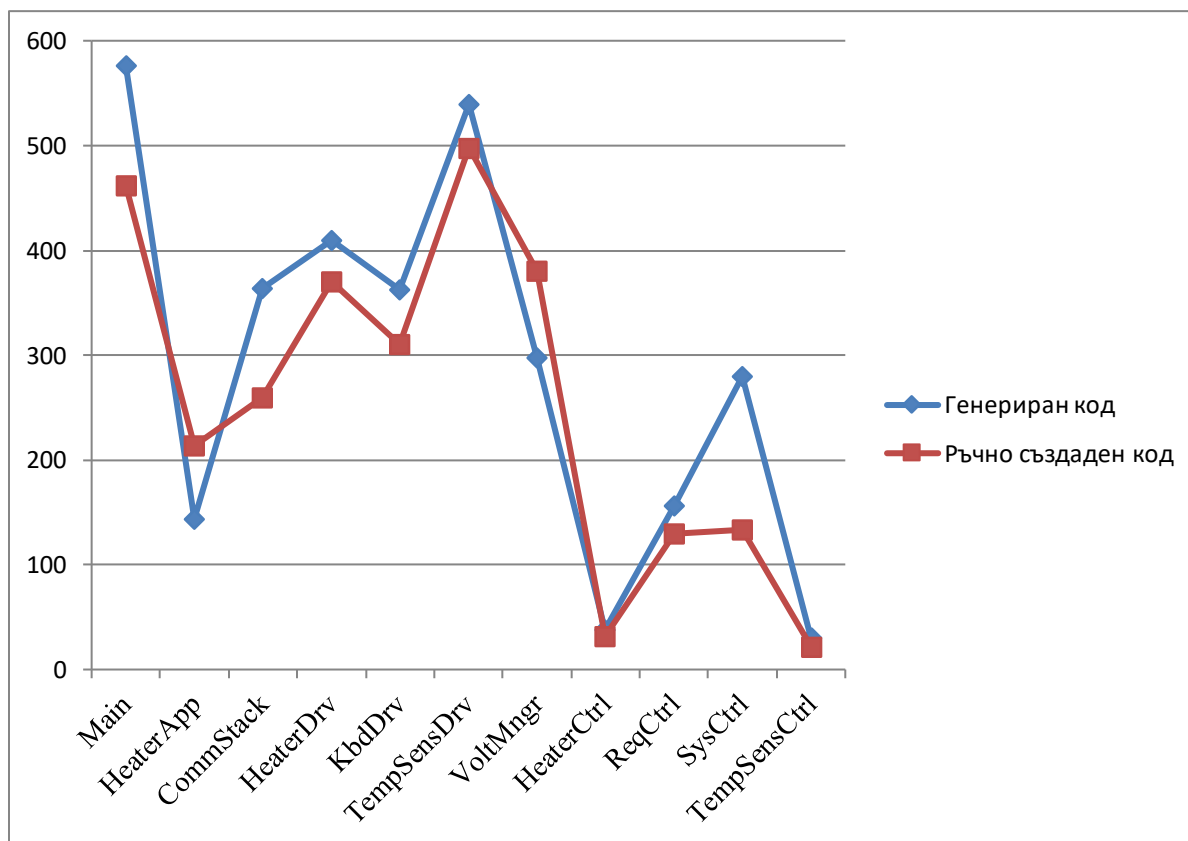
Изследването установи предимството на моделно базирана разработка на код по отношение на гъвкавост, възможност за повторно използване и интегриране на различни платформи. Надеждност и тестваемост са съпоставими и до голяма степен зависят от конкретната реализация. Езиците от високо ниво предоставят по-добра възможност за цялостно разбиране на алгоритъма, генерираният код е със задоволителна четимост, а основното му достоинство е хомогенност.

Изследването установи сходни параметри по отношение на ефективността. Установено е, че генерираният код заема 7% повече памет (таблица 2.5) и е 12% по-бърз (таблица 2.7). Тези показатели варират според реализираната функционалност и стил на разработката, както е видно за размера на кода (фиг.2.13), размера на данните (фиг.2.14) и времето за изпълнение (фиг.2.15).

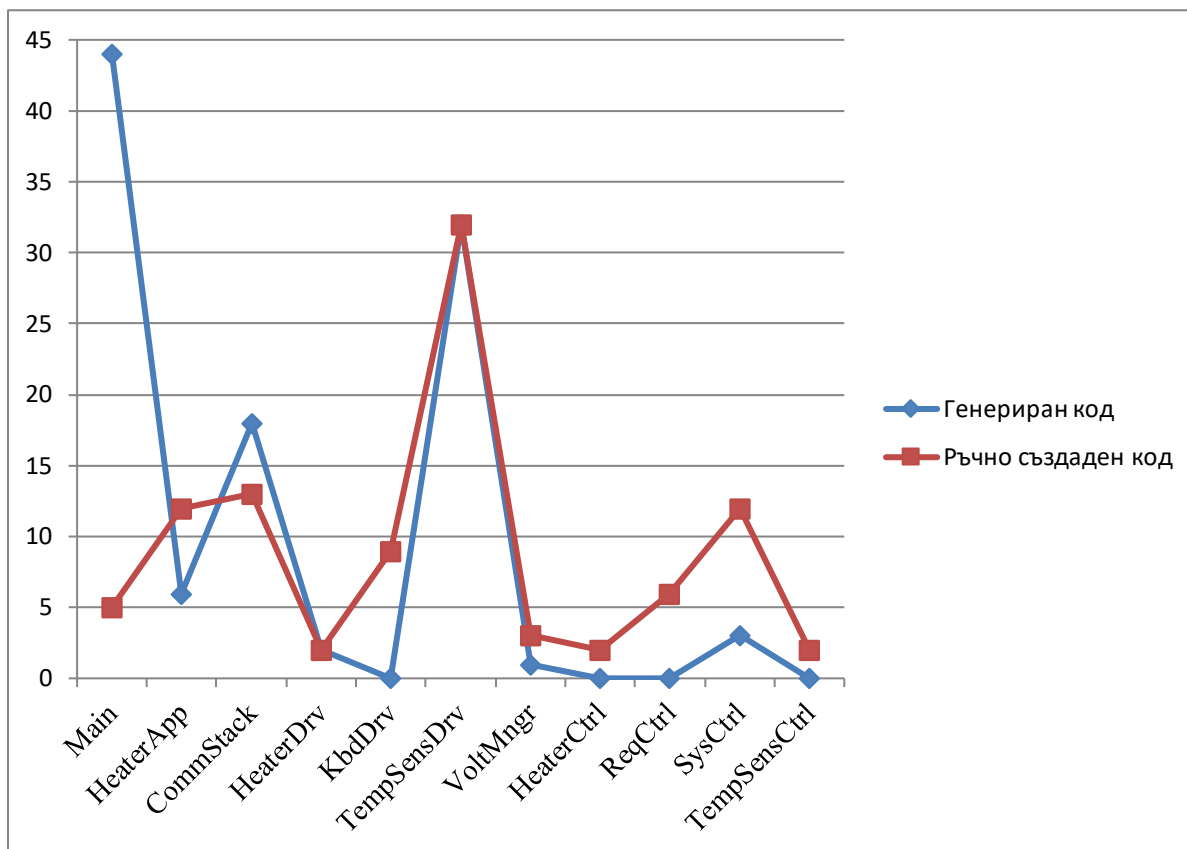
Резултатите следва да се считат за индикатор за съизмеримостта на подходите.

**Таблица 2.5.** Използвана памет за автомобилен отоплител за седалки в bytes – обобщени данни

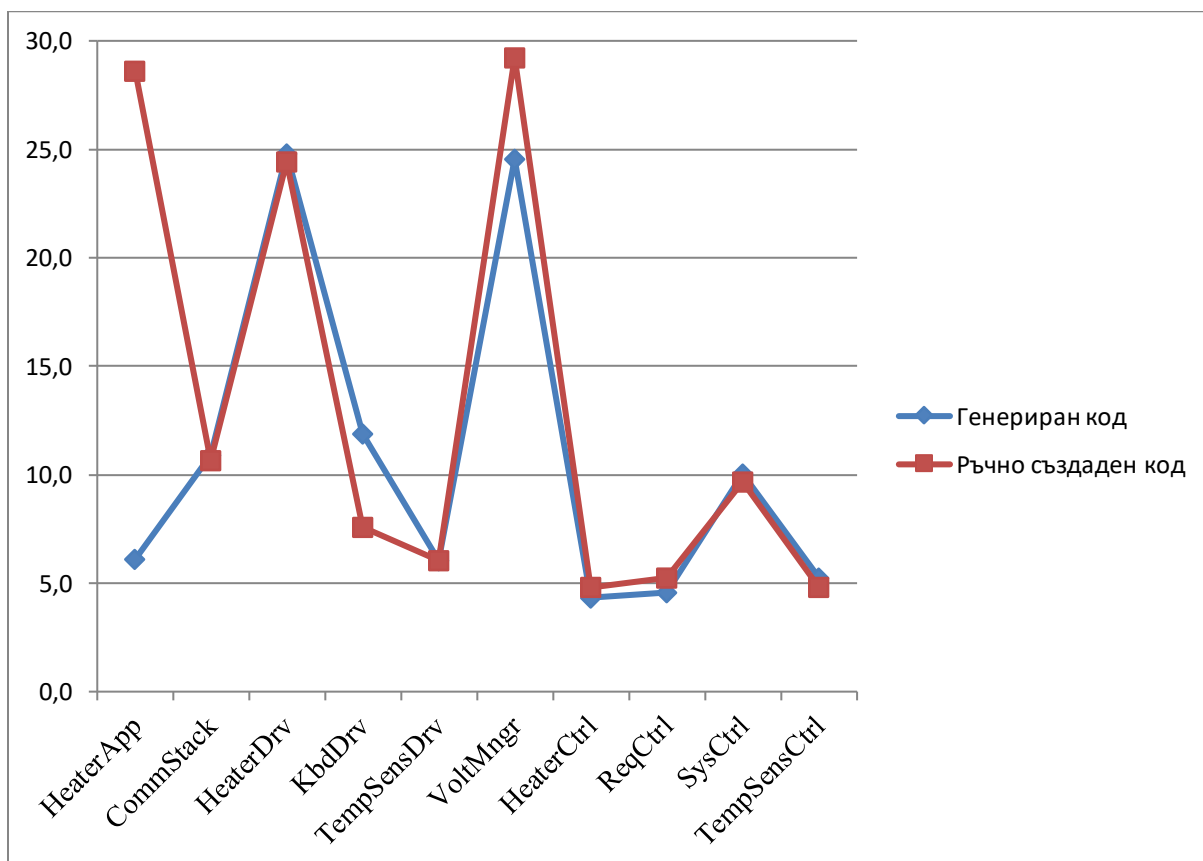
| Стил на кодиране | Вид памет       | TI v16.9.4.LTS | GNU v6.2.1.16 (SOMNIUM) | IAR C/C++ Compiler for MSP430 7.11.1 | Средно      |
|------------------|-----------------|----------------|-------------------------|--------------------------------------|-------------|
| Ръчен            | Код + константи | 4928           | 5333                    | 5308                                 | <b>5190</b> |
|                  | Данни + стек    | 182            | 163                     | 150                                  | <b>165</b>  |
| Генериран        | Код + константи | 5446           | 5971                    | 5694                                 | <b>5704</b> |
|                  | Данни + стек    | 186            | 174                     | 171                                  | <b>177</b>  |
| Разлика          | Код + константи | <b>518</b>     | <b>638</b>              | <b>386</b>                           | <b>514</b>  |
|                  | Данни + стек    | <b>4</b>       | <b>11</b>               | <b>21</b>                            | <b>12</b>   |
| Разлика, %       | Код + константи | <b>10,5</b>    | <b>12,0</b>             | <b>7,3</b>                           | <b>9,9</b>  |
|                  | Данни + стек    | <b>2,2</b>     | <b>6,7</b>              | <b>14,0</b>                          | <b>7,3</b>  |



Фиг.2.13. Среден размер на кода на модулите, bytes



Фиг.2.14. Среден размер на данните за всеки от модулите, bytes



Фиг.2.15. Осреднено най-голямо време за изпълнение на всеки от модулите,  $s \cdot 10^{-6}$

**Таблица 2.7.** Най-голямо време за изпълнение на системния софтуер, s.10<sup>-6</sup>

|          |            | TI<br>v16.9.4.LTS | GNU<br>v6.2.1.16<br>(SOMNIUM) | IAR C/C++<br>Compiler<br>for MSP430<br>7.11.1 | Средно       |
|----------|------------|-------------------|-------------------------------|-----------------------------------------------|--------------|
| Сумарно  | Ръчен      | 151.4             | 130.1                         | 111.6                                         | <b>131.0</b> |
|          | Генериран  | 121.6             | 108.4                         | 95.6                                          | <b>108.5</b> |
|          | Разлика, % | <b>-19.7</b>      | <b>-16.7</b>                  | <b>-14.3</b>                                  | <b>-17.2</b> |
| Измерено | Ръчен      | 119.7             | 95.3                          | 77.8                                          | <b>97.6</b>  |
|          | Генериран  | 110.5             | 82.2                          | 63.3                                          | <b>85.3</b>  |
|          | Разлика, % | <b>-7.7</b>       | <b>-13.7</b>                  | <b>-18.6</b>                                  | <b>-12.6</b> |

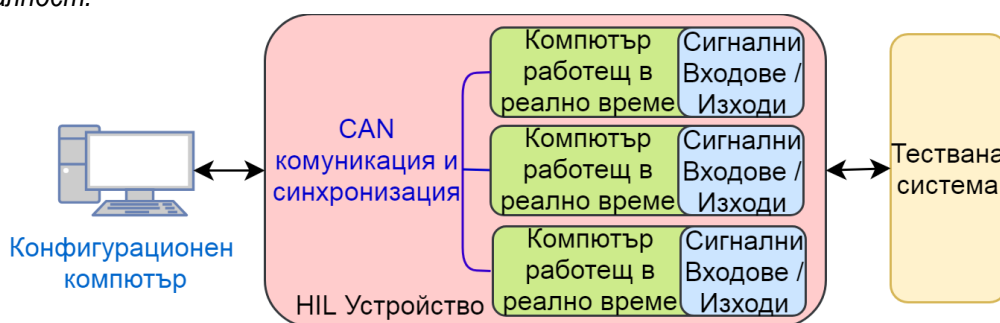
Извършените изследвания дават аргументация за потвърждение на възможността за използване на автоматично генериран продуктов код за тестване на системи, в контекста на стандартите за безопасност в автомобилната индустрия.

### ГЛАВА 3. Дефиниране на системна архитектура

Изследването цели дефиниране на хардуерна платформа и системна архитектура, които биха дали възможност за автоматизиране на процеса на разработка и интеграция на HIL системи. Системата следва да гарантира манипулирането на тестовата система в реално време, реализирайки среда, сходна с реалната.

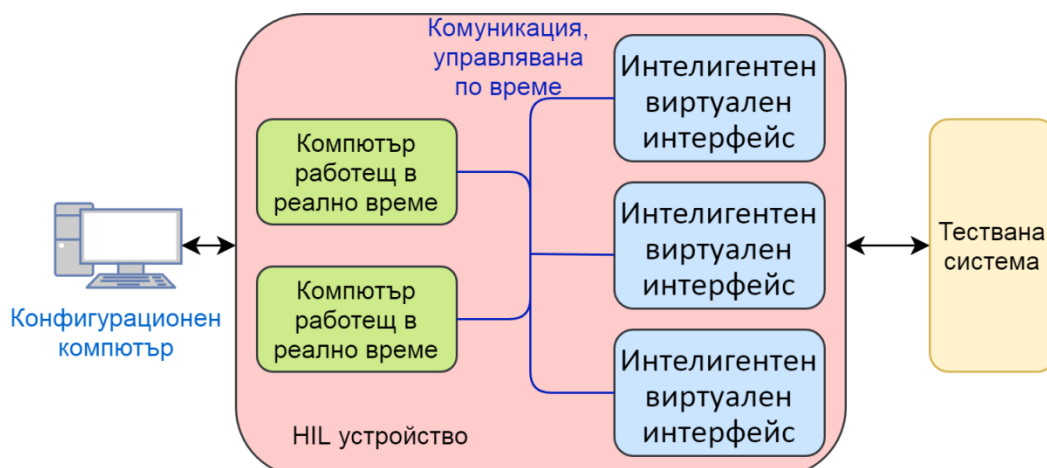
Подробно са разгледани публикуваните HIL решения и са анализирани от гледна точка на системното архитектурно решение, което те предлагат. Анализирани са приложимите изчислителни платформи и са изведени зависимостите между типа архитектура и нейните преимущества и недостатъци във връзка с процеса на разработване и интегриране на такъв тип системи.

Най-често разпределените HIL системи (фиг.3.1) са базирани на протоколи реализиращи синхронизация по време (Time-Triggered Protocol, TTP). Универсалността на този вид протоколи и възможността за синхронизиране на системата чрез тях ги прави много добър избор при разпределени синхронизирани системи. Използването на такъв протокол води до ограничение по отношение на приложимите хардуерни платформи, които следва да разполагат с тази функционалност.



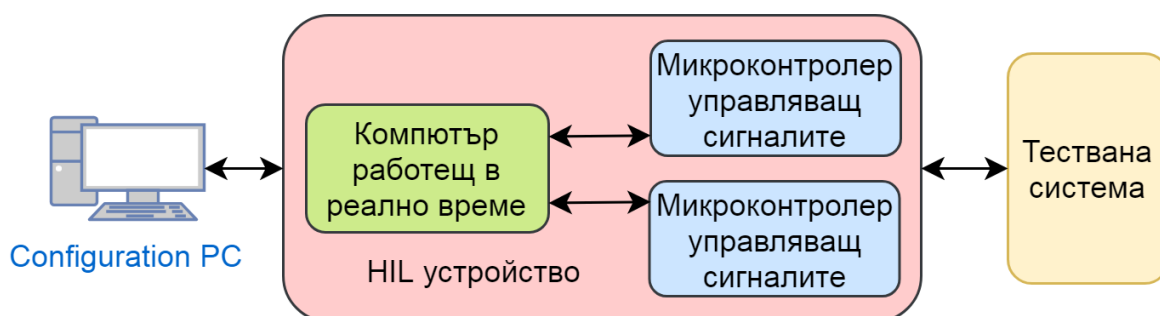
**Фиг.3.1.** TTP базирана HIL система

Алтернативно, архитектурата може да разпредели изчислителната и интерфейсна част на системата. Такъв подход представлява съвкупност от изчислителни машини и микроконтролерни системи, комуникиращи и синхронизирани чрез TTP. Предложението е базирано на интелигентни виртуални симулатори (Smart Virtual Transducers) и предлага възможност за оптимизация на ресурсите и използване на конкретно необходимите изчислителни и интерфейсни конфигурации (фиг.3.2)



**Фиг.3.2.** HIL система, базирана на интелигентни виртуални симулатори

Разпределена HIL система с централизирана изчислителна част може да бъде реализирана без ограничения в комуникационните протоколи фиг.3.3. Решението се състои от мощна процесорна платформа свързана с неограничен брой подсистеми отговорни за прочитане или симулиране на сигнали. Последните комуникират с изчислителната платформа без ограничения от гледна точка на комуникационните протоколи, и могат да бъдат реализирани на различни хардуерни платформи. Синхронизация не е необходима, понеже честотата на опресняване на входно/изходните сигнали е много по-висока от тази с която тестовата система взема решения.



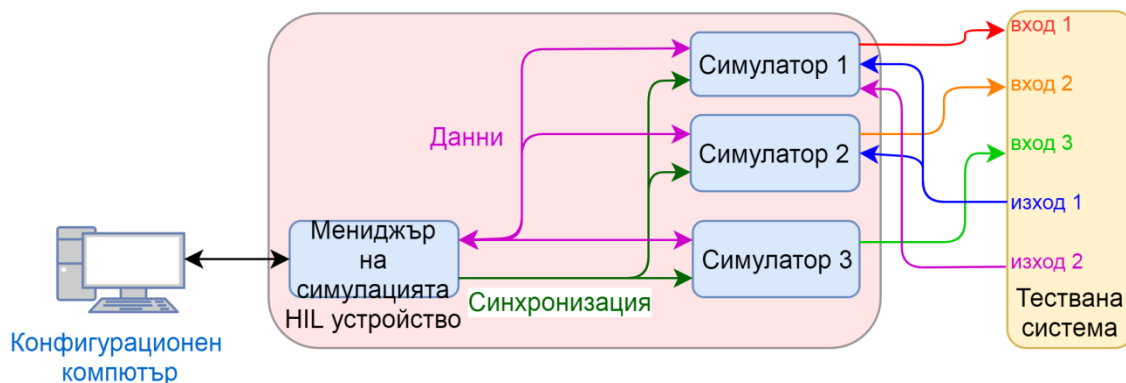
**Фиг.3.3.** Разпределена HIL система с централизирана изчислителна част

Създадената архитектура следва да отговаря на значителен брой условия, подредени приоритетно, спрямо генералните изисквания към HIL системите и целите на изследването:

- Унификация и възможност за автоматизация на структурата, както и на разпределението на функционалностите
- Работа в реално време
- Синхронност между входовете и изходите на системата
- Липса на ограничения за използваната платформа

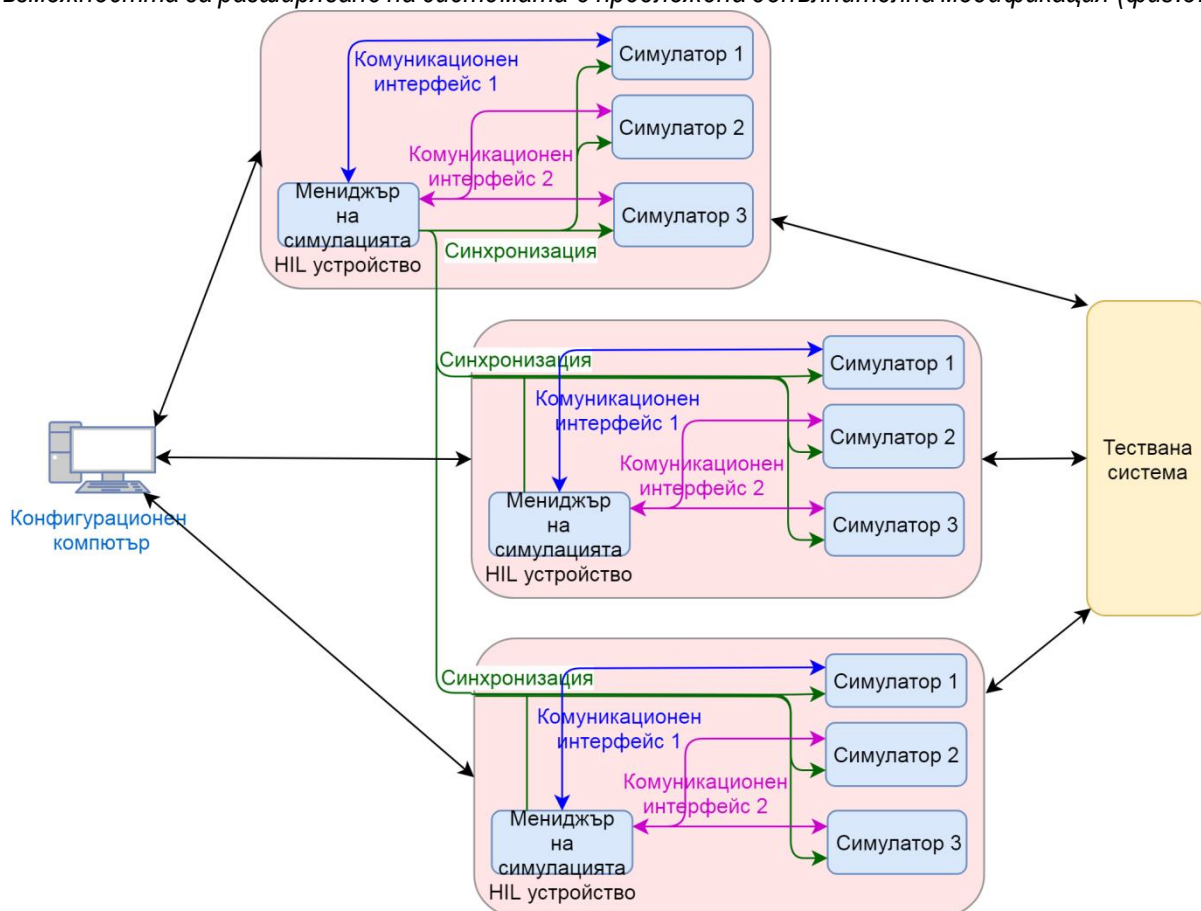
Установено е, че най-често използваните хардуерни платформи FPGA и CPU трудно биха могли да бъдат включени в автоматизиран процес за разработка и интеграция на HIL системи. В тази връзка е избран микроконтролерът, като сравнително гъвкав, универсален, масов и евтин. Липсата на естествен паралелизъм на операциите в тази платформа дефинира необходимостта от разпределена системна архитектура.

В резултат е предложена уникална системна архитектура, базирана на микроконтролери свързани в разпределена, синхронизирана, безкрайно разрастваща се мрежа (фиг.3.6).



**Фиг.3.6.** Предложена интеграция на HIL система

Ограниченията на реализираната системна архитектура се свързват главно с използваните хардуерни платформи, тяхната изчислителна мощност и интерфейси. Във връзка с възможността за разширяване на системата е предложена допълнителна модификация (фиг.3.8).



**Фиг.3.8.** Разширяване на системата чрез добавяне на синхронизирани HIL подсистеми

## ГЛАВА 4. Софтуерна архитектура за HIL система

Изследването анализира съвременните решения за софтуерна архитектура. Генерално, изискванията към софтуерната архитектура на предложената HIL система са свързани с:

- Унификация и възможност за автоматизация на структурата, както и разпределението на функционалностите
- Работа в реално време
- Синхронизация между входовете и изходите на системата

Предложената системна архитектура се реализира с два типа подсистеми.

- ❖ **Мениджър на симулацията:**
  - осигурява синхрон в симулацията, чрез генериране на синхронизиращ електрически цифров сигнал
  - комуникира с конфигуриращия компютър посредством общ интерфейс(USB, Ethernet и др.)
  - комуникира със симулаторите посредством различни стандартни интерфейси (I2C, SPI, UART и др.)
  - управлява данните между конфигуриращия компютър и симулаторите, гарантирайки навременното им доставяне.
- ❖ **Симулатор**
  - базира времето на изпълнение на задачите си на синхронизиращ външен цифров сигнал
  - комуникира с мениджъра на симулацията чрез стандартен интерфейс (I2C, SPI, UART или др.)
  - комуникира с конфигурационния компютър чрез мениджъра на симулацията, приемайки данни от които зависи симулацията и изпращайки данни за състоянието на системата
  - синхронно прочита различни типове управляващи електрически сигнали(цифрови, аналогови, широчинно-импулсна модулация, комуникационни и др.) на тестовата система
  - синхронно генерира конкретен симулационен сигнал (цифров, аналогов, широчинно-импулсна модулация, комуникационен или друг) базиран на модел на средата приложен върху прочетени управляващи сигнали и данни от конфигурационния компютър

На база на по-горе дефинираните изисквания за двата типа подсистеми, съставляващи предложението HIL (мениджър на симулацията и симулатор), се предлага нестандартно решение, способно да реализира описаната функционалност съответно за мениджър на симулацията (фиг.4.6) и симулатор (фиг.4.11).



**Фиг.4.6.** Софтуерна архитектура на мениджър на симулацията

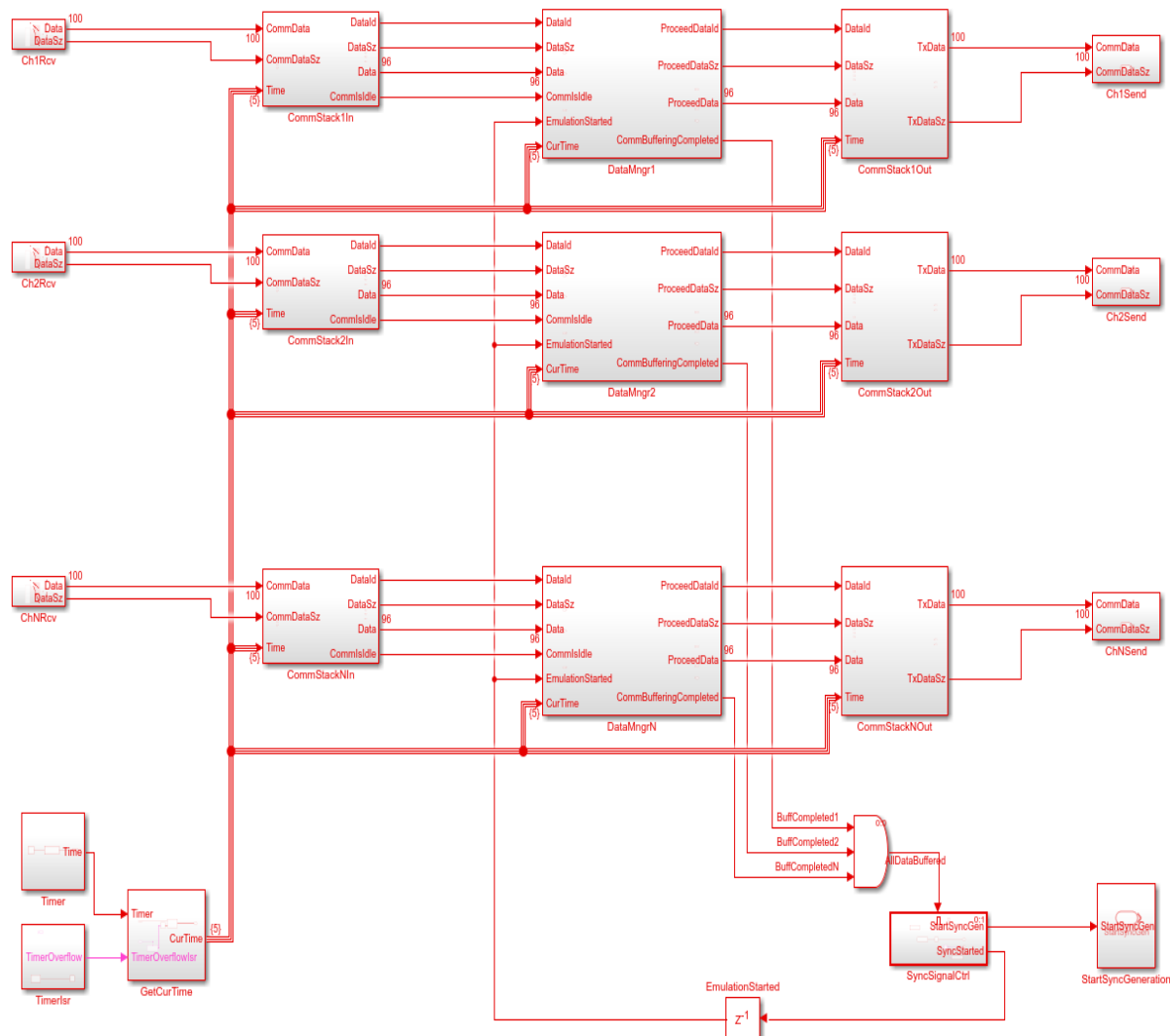


**Фиг.4.11.** Архитектура на симулатор

Основна разлика между двете архитектури са от една страна по-големия брой различни интерфейси, използвани от симулатора, а от друга – по-големия брой архитектурни слоеве, осигуряващи абстракция на данните и необходимостта от операционна система, управляваща различните функционалности във връзка с приоритета им.

Създаденият модел на системата на мениджъра на симулация се отличава със специфичност

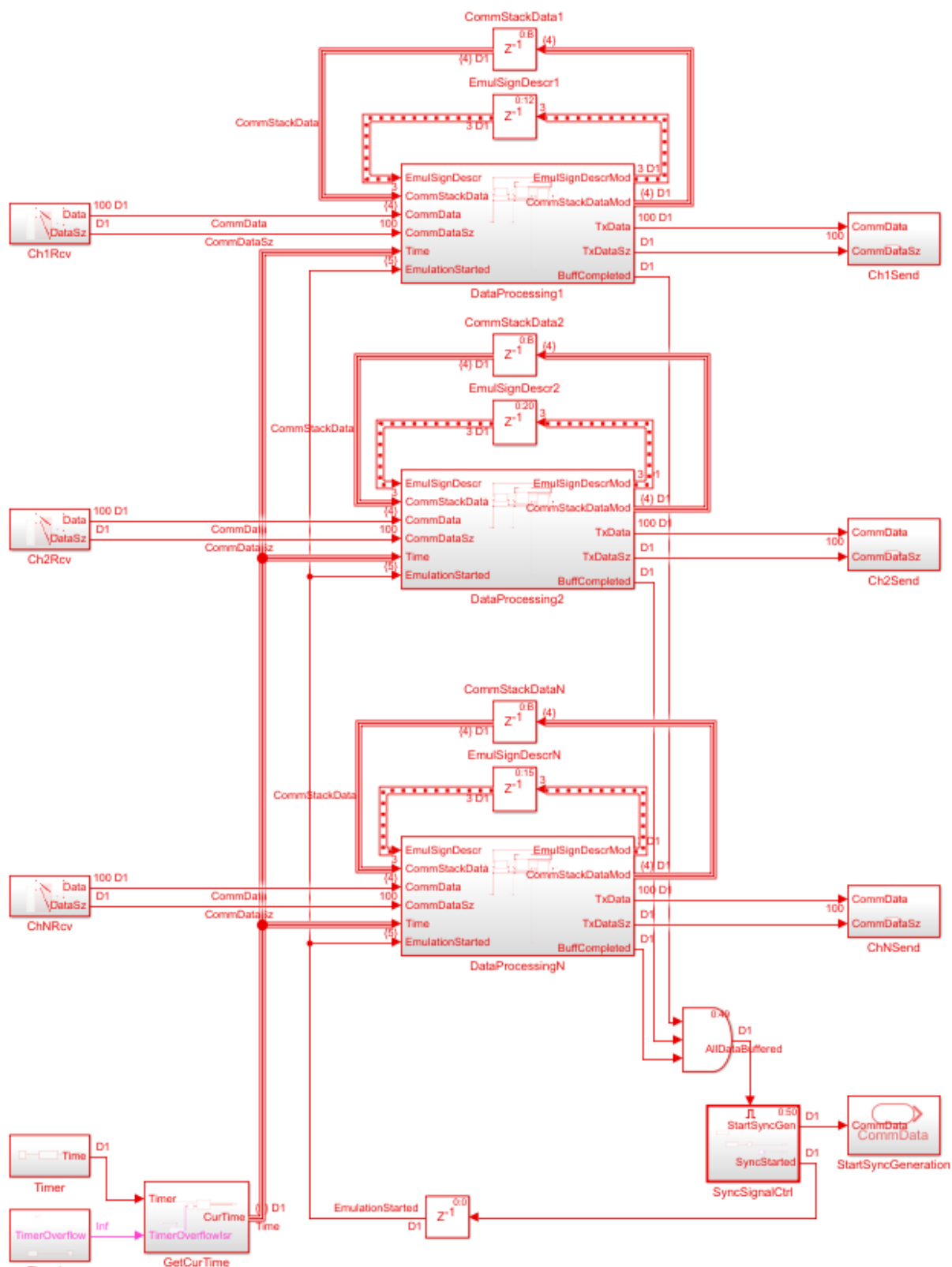
при работа с различните комуникационни интерфейси (фиг.4.7).



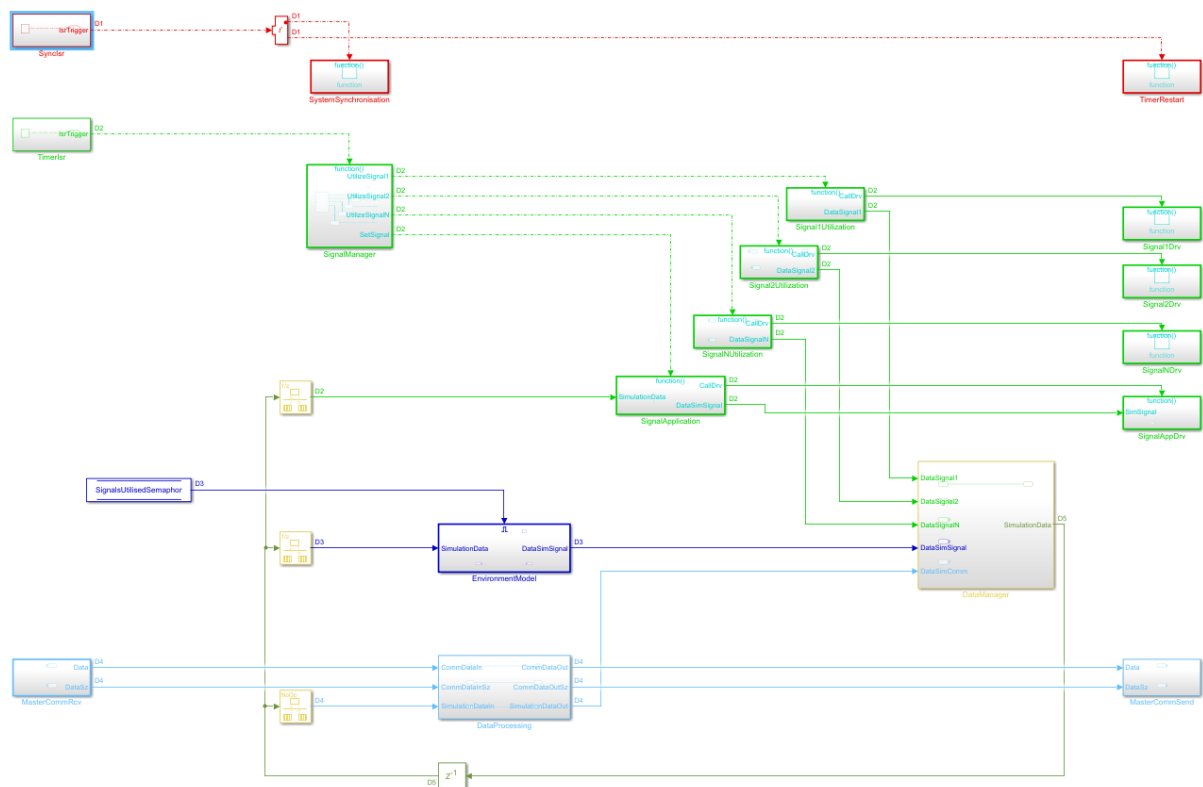
Фиг.4.7. Детайлен софтуер дизайн на мениджър на симулацията

След оптимизация на решението се достига до универсална форма, състояща се от стандартни модули и конфигурация, която може да бъде формално създадена, т.е. генерирана (фиг.4.9).

Моделът на системата на симулатора (фиг.4.12) се характеризира с по-голяма сложност. Системата се състои от мениджър на сигналите, управляван чрез таймер и отговорен за прочитане и установяване на електрически сигнали, комуникационен стек и модел на симулираната среда. Данните в системата се управляват чрез мениджър на данни а синхронизацията ѝ с останалите подсистеми се осигурява от синхронизиращо прекъсване.



Фиг.4.9. Обобщен, унифициран детайлен софтуер дизайн на мениджър на симулацията



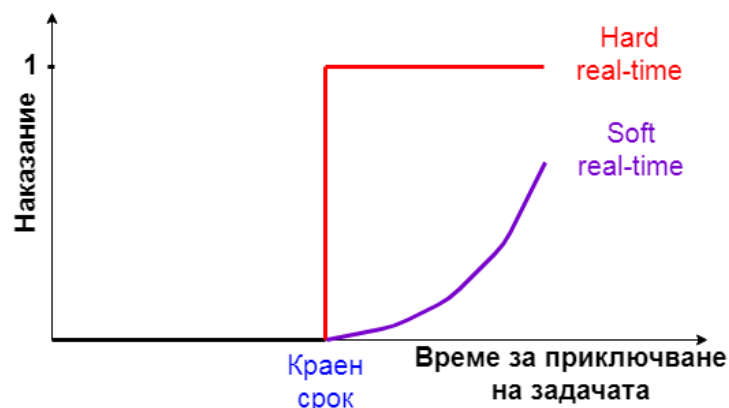
Фиг.4.12. Детайлен дизайн на симулатор

Представеното решение е уникално и представлява възможност за автоматизиран процес на разработка.

## ГЛАВА 5. Оценка на работоспособността на архитектурата

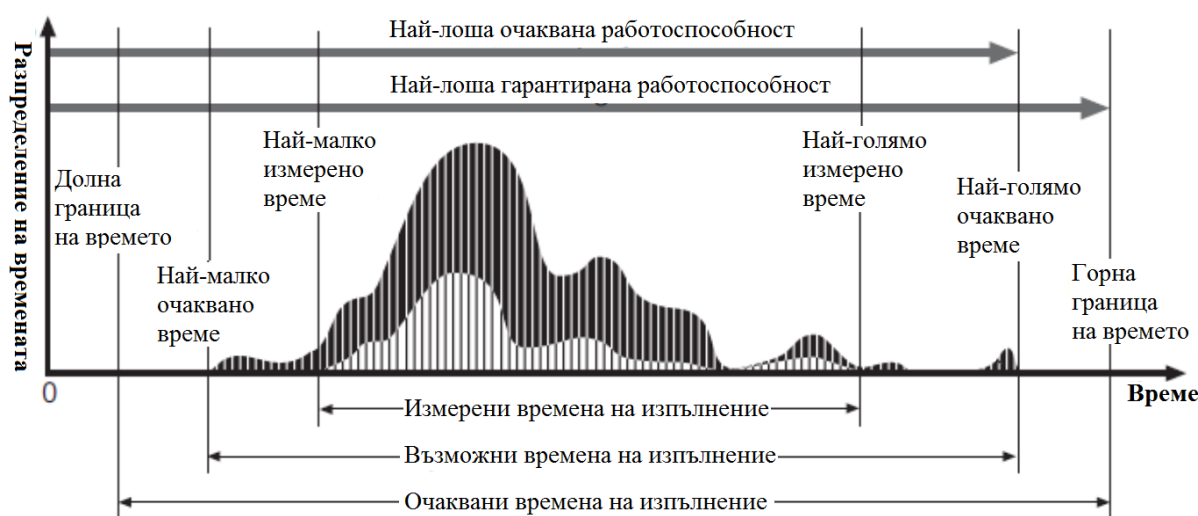
В тази част на дисертационния труд са създадени модели на работоспособността на двата типа подсистеми, участващи в предложената HIL реализация. Методът се базира на вече известни уравнения за натоварването на системата, приложени в контекста на разработката, и специфични за конкретната система изисквания и ограничения.

HIL предоставя възможност за симулиране на работата на част от подсистемите на една комплексна система. Това налага бързодействието ѝ да е съобразено с това на тестваната система, т.е. тя да работи в реално време. В зависимост от типа системи, работата в реално време може да бъде разделена на такава с твърди и меки ограничения на времето на изпълнение. Разликата между двете е, че докато за първата *hard real time* система всяко преминаване на срока в който задачата трябва да бъде завършена е пагубно, то при *soft real time* то е нежелано, но не критично фиг. 5.1.



Фиг.5.1. Сравнение на *hard* и *soft* системи за работа в реално време

Смята се, че една система, работеща в реално време, се състои от няколко задачи които реализират необходимата функционалност. Обикновено времето за изпълнение на задачите варира, съобразно входните данни и състоянието на системата. Това води до променливо натоварване на системата фиг.5.2. Възможни са значителни разлики между най-малкото и най-голямото време за изпълнение. В допълнение за комплексни системи, възможните комбинации от параметри са твърде много за да бъдат изследвани, което води до невъзможност за обективно определяне на времената чрез изследване на реалната работа на системата. В резултат се наблюдава подценяване на най-голямото измерено време и надценяване на най-малкото. По-голяма степен на обективност може да бъде постигната чрез методи, оценяващи всички възможни пътища за изпълнение на системата, статистически методи. Те не са базирани на изпълнението на реалния код върху реалната хардуерна платформа или симулация. Тяхното статистически базирано очакване е въз основа на използвания код и допълнителна информация, изследвани чрез откриване на пътищата на изпълнение и комбинацията им с абстрактни модели на хардуера.



Фиг.5.2. Разпределение на времето за изпълнение

### 5.3 Характеризиране на системата и нейните зависимости

$n_{зад} n_{прек}$  Работоспособността на една система може да бъде изчислена на база настъпилите събития прекъсвания и задачи в конкретен времеви прозорец  $t$  (5.1).

$$\sum_{j=1}^{n_{прек}} \frac{C_j + 2 * \Delta_{прек}}{t} * \left[ \frac{t}{T_j} \right] + \sum_{j=n_{прек}+1}^{n_{прек}+n_{зад}} \frac{C_j + 2 * \Delta_{зад}}{t} * \left[ \frac{t}{T_j} \right] < 1 \quad (5.1)$$

В случай, че прекъсванията и задачите се разглеждат като еднозначни модули, и базирани на факта, че в практиката  $\Delta_{зад} > \Delta_{прек}$ , (5.1) може да бъде опростена до (5.2)

$$\sum_{j=1}^n \frac{C_j + 2 * \Delta_{зад}}{t} * \left[ \frac{t}{T_j} \right] < 1 \quad (5.2)$$

Изследваната по-долу система се базира на периодични събития с константен период и честота значително по-голяма от 1Hz. Това дава възможност за допълнително опростяване на използваното уравнение, чрез извършване на изчисленията за период  $t = 1s$ . Тогава (5.2), добива вида (5.3) използван в изчисленията по-долу.

$$F_j T_j = \frac{1}{F_j} \sum_{j=1}^n \frac{C_j + 2 * \Delta_{зад}}{T_j} < 1 \quad (5.3)$$

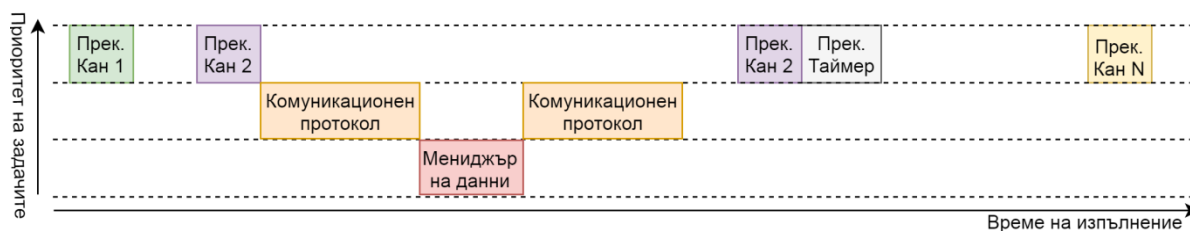
$\sum_{j=1}^n F_j * (C_j + 2 * \Delta_{зад}) < 1$  (5.4) След заместване с , където е честотата на изпълнение, уравнението добива формата

### 5.3.1 Мениджър на симулацията

Чрез времедиagramата на работа (фиг.5.3) е изведена зависимост за натоварването на системата на мениджъра на симулацията, във връзка с работата на различните ѝ подсистеми и параметрите от които тя зависи.

Условието за работоспособност (неравенство 5.11) е средното натоварване в системата за период от време 1 s да бъде по-малко от 1.

В допълнение са предложени допълнителни зависимости на работоспособността на системата. Неравенство 5.12 регламентира необходимата скорост за обмен на данни и невъзможността тя да надхвърли капацитета на комуникационния канал. Неравенство 5.12 проверява възможността за работа на интегрираната система, предвид че архитектурата ѝ реализира пренос на данни между симулаторите и конфигуриращия компютър чрез мениджъра на симулацията. Поради тази причина, сумата от необходимите скорости за обмен на данни по всички комуникационни канали симулатор – мениджър на симулацията следва да е по-малка от капацитета на канала мениджър на симулацията – конфигуриращ компютър (неравенство 5.13).



Фиг.5.3. Времедиagramа на работата на мениджър на симулацията

$$\sum_{j=1}^{n_{ком}} \frac{R_{комj}}{D_{прекj}} * t_{ком\_прекj} + \sum_{j=1}^{n_{ком}} \frac{R_{комj}}{D_{пакетj}} * t_{ком\_стекj} + \sum_{j=1}^{n_{ком}} \frac{R_{комj}}{D_{пакетj}} * t_{мен\_данниj} < 1 \quad (5.11)$$

$$\forall R_{комj} < R_{коммаксj} \quad (5.12)$$

$$\sum_{j=1}^{n_{ком}-1} R_{комj} < R_{коммакс\_n} \quad (5.13)$$

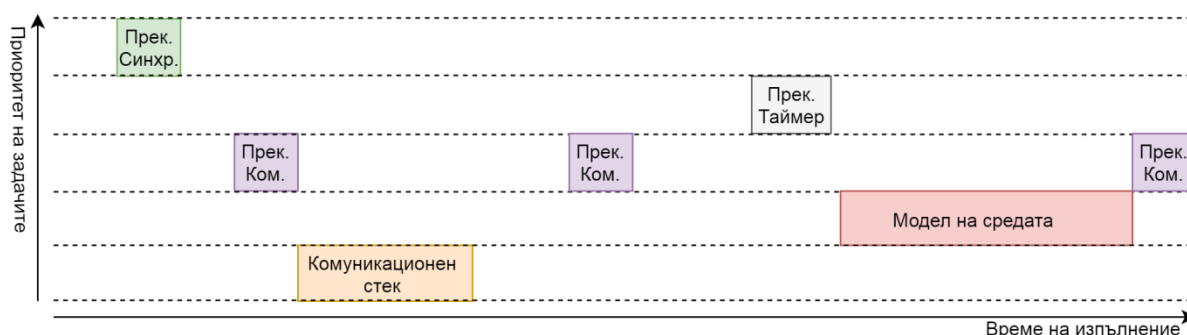
### 5.3.2 Симулатор

Чрез времедиagramата на работа (фиг.5.4) е изведена зависимост за натоварването на системата на симулатора, във връзка с работата на различните ѝ подсистеми и параметрите от които тя зависи.

Условието за работоспособност (неравенство 5.18) е средното натоварване в системата за период от време 1 s да бъде по-малко от 1.

В допълнение е предложена допълнителна зависимост на работоспособността на системата, характеризираща работата ѝ като HIL система, работеща в реално време. Изчислена е максималната асинхронност на симулираните изходни сигнали, във връзка с предложената

софтуерна архитектура. Системата е работна при стойност по-малка от изисканата  $\Delta_{\text{сим}}$  (неравенство 5.21).



Фиг.5.4. Времедиаграма на симулатор

$$\frac{R_{\text{ком}}}{D_{\text{прек}}} * t_{\text{ком\_прек}} + f_{\text{синхр}} * t_{\text{синхр\_прек}} + f_{\text{сим\_прек}} * t_{\text{сим\_прек}} + \frac{R_{\text{ком}}}{D_{\text{пакет}}} * t_{\text{ком\_стек}} + \frac{R_{\text{ком}}}{D_{\text{пакет}}} * t_{\text{мен\_данни}} + f_{\text{сим}} * t_{\text{модел\_сим}} < 1 \quad (5.18)$$

$$t_{\text{сим\_прек}} + \max(t_{\text{ком\_прек}}, t_{\text{синхр\_прек}}) + \max(t_{\text{ком\_прек}}, t_{\text{сим\_прек}}) < \Delta_{\text{сим}} \quad (5.21)$$

## 5.5 Резултати

Предложените зависимости са изследвани, чрез постепенно натоварване на изучаваните системи до достигане на границата на тяхната работоспособност.

За мениджъра на симулацията се увеличава необходимата скорост за комуникация с 1000 bytes/s на стъпка. Експериментът не успя да достигне границата на работоспособността, дори чрез максимално допустимите скорости на комуникация. Изчисленията също доказаха, че това не е достатъчно за постигане на максимално натоварване на системата. Поради тази причина е добавено закъснение от 1 милисекунда в мениджъра на данни, което да доведе до допълнително натоварване. Вследствие в три експеримента с различна конфигурация са измерени различни прагове на работоспособност. Същите са изчислени чрез предложената методика. Резултати от сравнението са представени на фиг.5.5.

Таблица 5.5. Сравнение между теоретичната и измерената максимална скорост

|                                                 | Опит 1 | Опит 2 | Опит 3 |
|-------------------------------------------------|--------|--------|--------|
| Изчислена макс. скорост, byte/s                 | 30746  | 21549  | 12232  |
| Максимално достигната скорост на обмен на данни | 31000  | 21000  | 12000  |
| Разлика, %                                      | -0,8   | 2,6    | 1,9    |

За симулатора се увеличава времето, необходимо за изпълнение на модела на средата с 100 микросекунди на стъпка. Вследствие в три експеримента с различна конфигурация са измерени различни прагове на работоспособност. Същите са изчислени чрез предложената методика. Резултати от сравнението са представени на фиг.5.9.

Таблица 5.9 Сравнение между измереното максимално време за изпълнение на модела и теоретично изчисленото, базирано на (5.22), s.10<sup>-6</sup>

| Максимално време за изпълнение на модела | Опит 1 | Опит 2 | Опит 3 |
|------------------------------------------|--------|--------|--------|
|------------------------------------------|--------|--------|--------|

|                   |            |             |            |
|-------------------|------------|-------------|------------|
| <b>Изчислено</b>  | 9311,1     | 9153,9      | 8956,0     |
| <b>Измерено</b>   | 9200       | 9200        | 8400       |
| <b>Разлика, %</b> | <b>1,2</b> | <b>-0,5</b> | <b>6,6</b> |

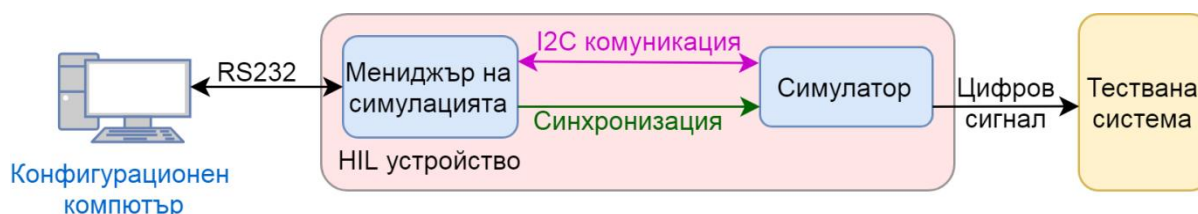
Експериментите доказаха грешка под 3% за мениджъра на симулация и под 7% за симулатора. Грешката за втората система е значително по-голяма. Това се дължи на по-комплексния характер на системата – по-голям брой обслужвани събития, повече функционалности, управление от система, работеща в реално време. Въпреки това, получените резултати са в заложените граници и това дава възможност предложения подход да бъде използван за пресмятане на работоспособността на предложената HIL система.

Достойнство на метода е доказаната му точност, даваща възможност той да бъде използван за надеждна оценка на работоспособността. В допълнение, в изследваните публикации не беше открито предложение за подобна методика, способна не само автоматично да разработи управлението на система, но и да оцени работоспособността ѝ. В тази връзка тя би могла да се нарече иновативна.

## ГЛАВА 6. Изследване на приложимостта на системата.

Тази част от дисертационния труд предлага изследване на три интеграции на HIL система, реализирани чрез предложения подход.

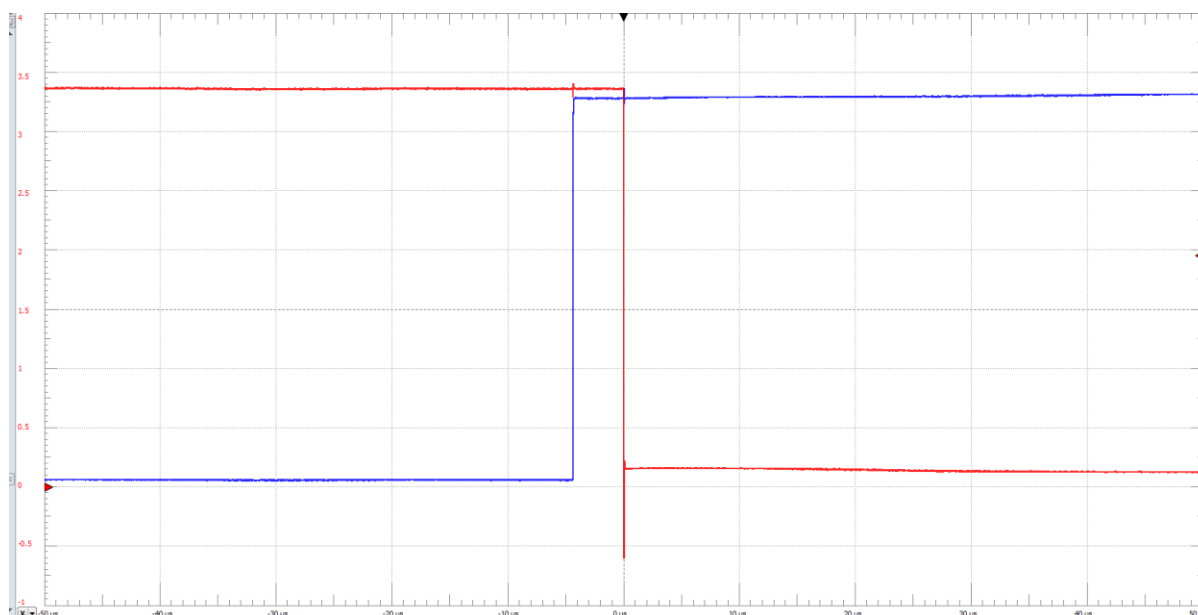
В първата реализация предложената HIL система е интегрирана за термично, инфрачервено изследване на работата на микроконтролерна система. За тази реализация е необходимо генерирането на един цифров симулиран сигнал. За целта се използва един симулатор, свързан с мениджъра на симулацията чрез I2C комуникационен канал. От своя страна, мениджърът комуникира с конфигуриращия компютър чрез UART интерфейс. В резултат е съставена HIL системата изобразена на фиг.6.1.



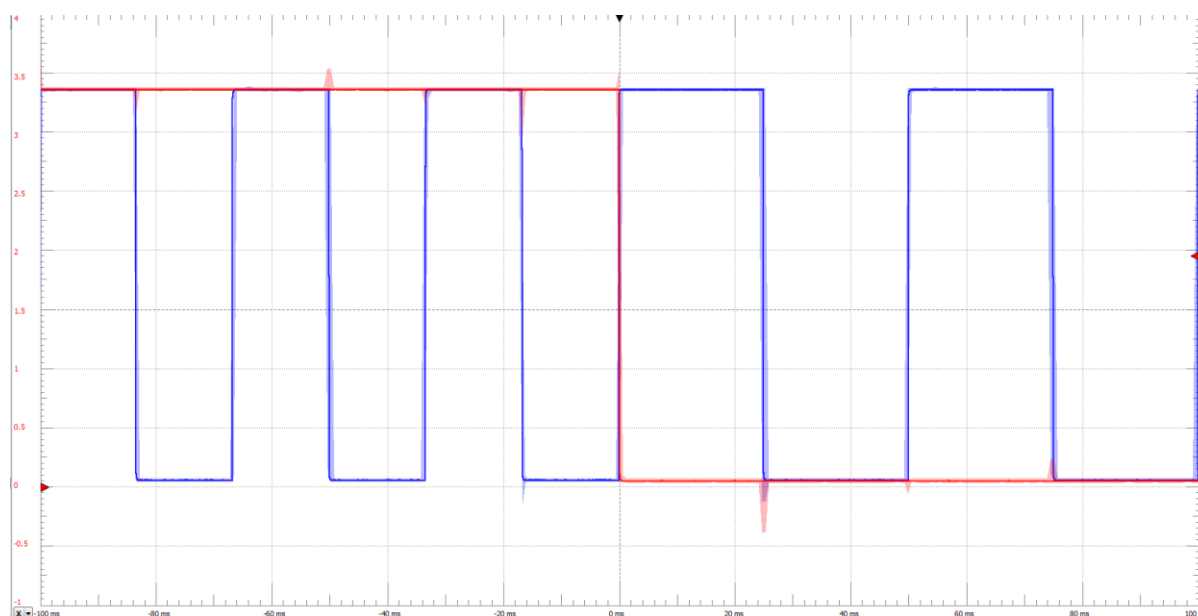
**Фиг.6.1.** Система на интегриран HIL за стимулиране на вградена система с цел изследване посредством инфрачервена термография

Същата е реализирана както е показано на фиг.6.4.





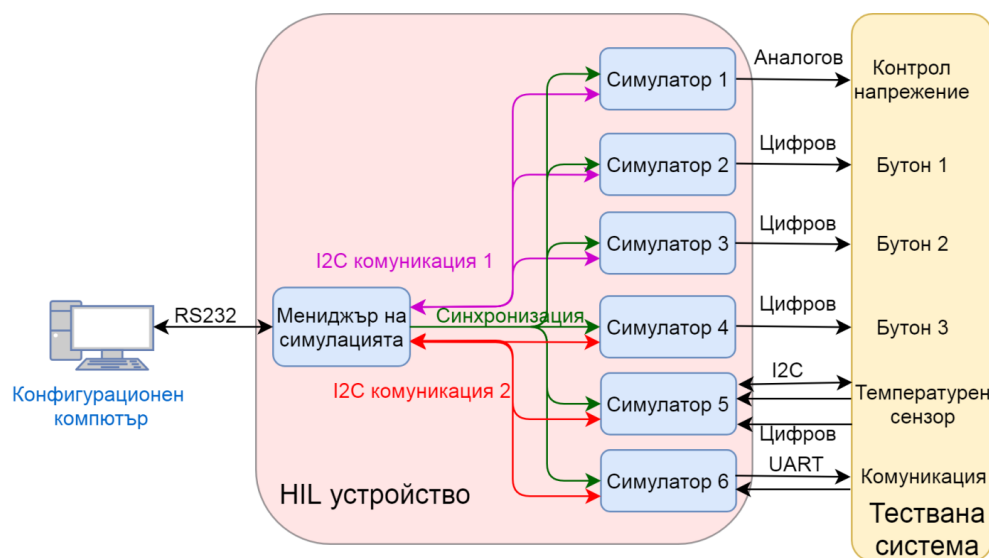
**Фиг.6.6.** Максимално отклонение на симулация спрямо синхронизиращия сигнал



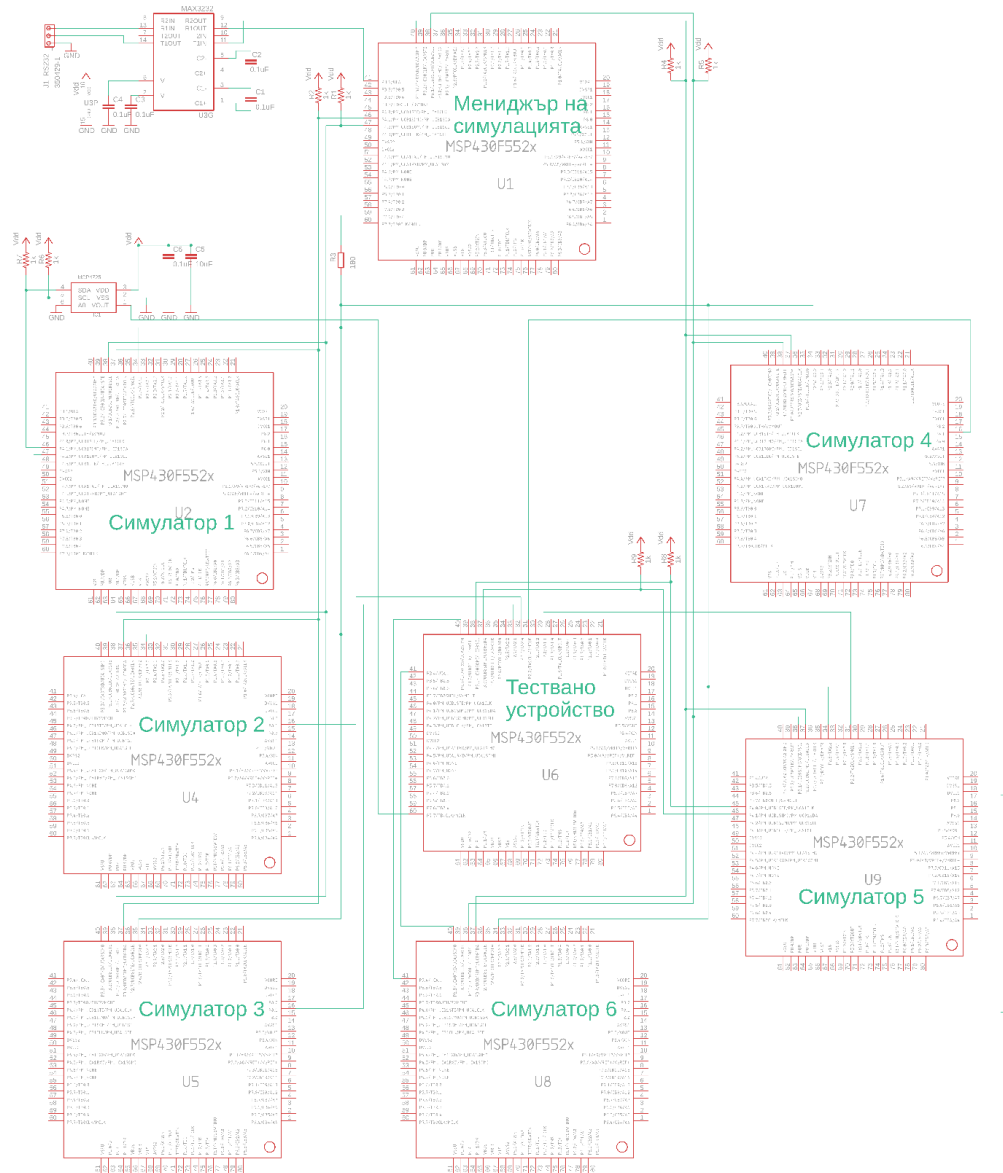
**Фиг.6.7.** Симулиран сигнал (лилаво), синхронизиращ сигнал (червено) и прехода от 30Hz – 20Hz

Вторият експеримент изследва приложимостта на предложената система за валидиране на автомобилен отоплител на седалки. Тази типична за съвременните автомобили функция е показателен пример за система без изисквания за безопасност, която рядко бива тествана с HIL система, поради изброените по-горе нейни недостатъци. В допълнение тя съдържа типични за индустрията интерфейси и функционалности.

Реализацията на предложената HIL система е изобразена на фиг.6.8. На нея всеки от входните за тестваната система сигнали е свързан с отделен симулатор. Симулаторите, реализирани чрез MSP430F5529 микроконтролери, са синхронизирани чрез сигнал, генериран от мениджъра на комуникацията. Обмяната на информация със същия се реализира посредством два I2C комуникационни канала. Комуникацията между конфигуриращия компютър и така предложената HIL система се реализира чрез RS232. Принципната схема на реализация е предложена на фиг.6.12.

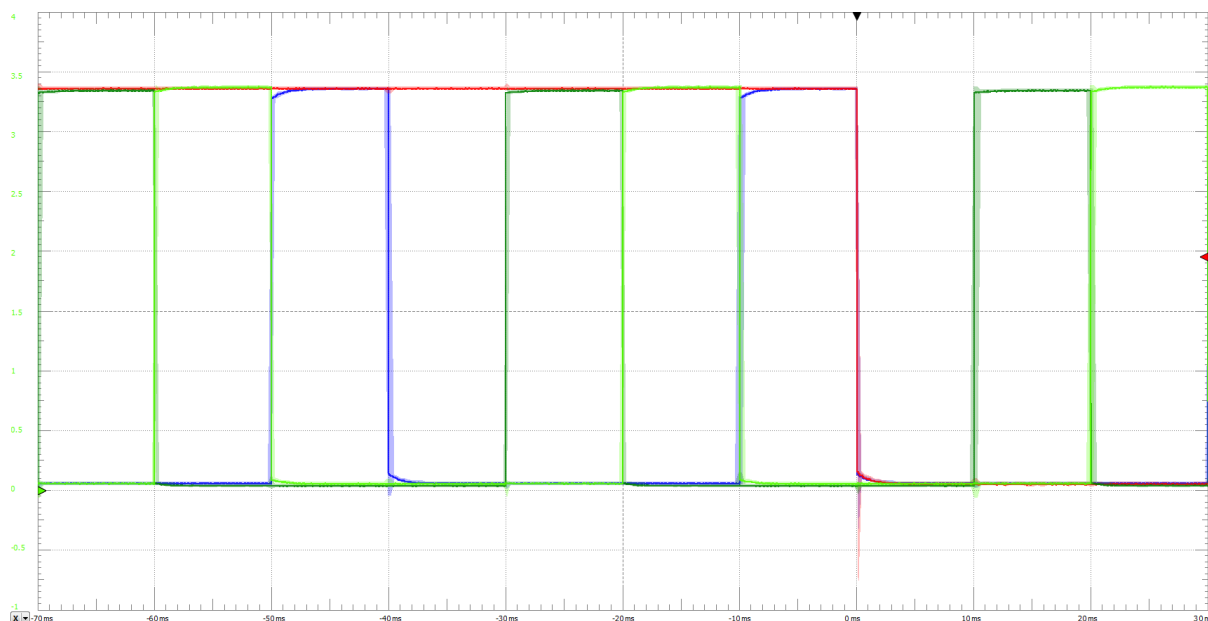


Фиг.6.8. HiL система за тестване на автомобилен отоплител на седалки



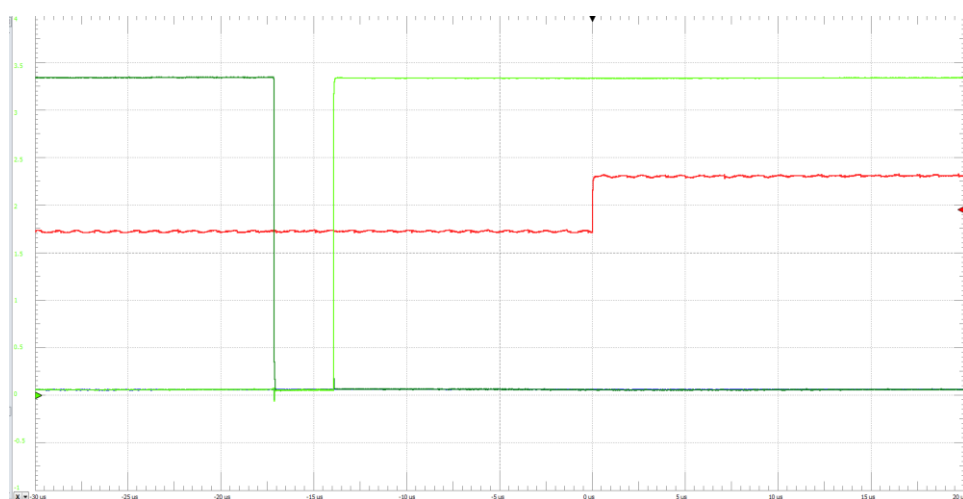
Фиг.6.12. Принципна схема на свързване на MSP430F5xxx демонстрационни платки в интегрирана HiL система за валидиране на автомобилен отоплител за седалки

В резултат от реализацията са измерени генерираните цифрови сигнали (фиг.6.13). Те са синхронни и със зададената стойност.



**Фиг.6.13.** Синхронността на симулираните сигнали на трите бутона (тъмно зелено, синьо, светло зелено) спрямо синхронизирания сигнал (червено)

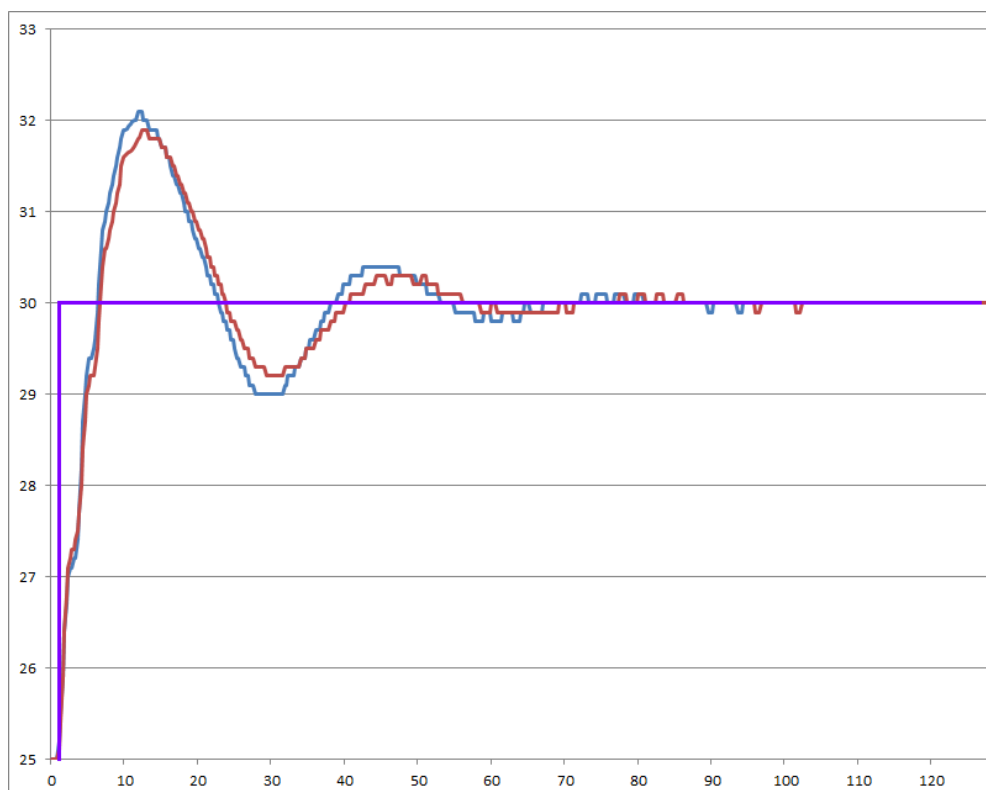
Установено е несъответствие във връзка със симулатор 1, за чието реализиране е необходимо използването на външен ЦАП, поради липсата на тази функционалност в използвания микроконтролер. Предложеният метод изисква използването на платформи, отговарящи на заданието, т.е. съдържа интерфейси съвместими с тези на тестваната система както по тип, така и по логическо ниво. В тази връзка е измерено до 17 микросекунди закъснение на симулирания аналогов сигнал(в червено) спрямо цифровия такъв(тъмно зелено и светло зелено). Това закъснение значително превишава изчисленото, което е показателно за необходимостта предложената методология да бъде стриктно спазвана.



**Фиг.6.14.** Синхронността на симулираните сигнали - два цифрови (тъмно зелено и светло зелено) и аналогов (червено)

За изследване на коректността на валидация, базирана на предложената HIL интеграция, са сравнени температурата като функция от времето(фиг.6.15) за тест с реална седалка

(червената крива) и такъв, реализиран чрез предложената HIL система (синята крива). Резултатните криви се различават, поради използването на модел за симулацията, който реализира грешка. Въпреки това те са сходни и може да се счита, че предложената система реализира точно копие на реалната среда.

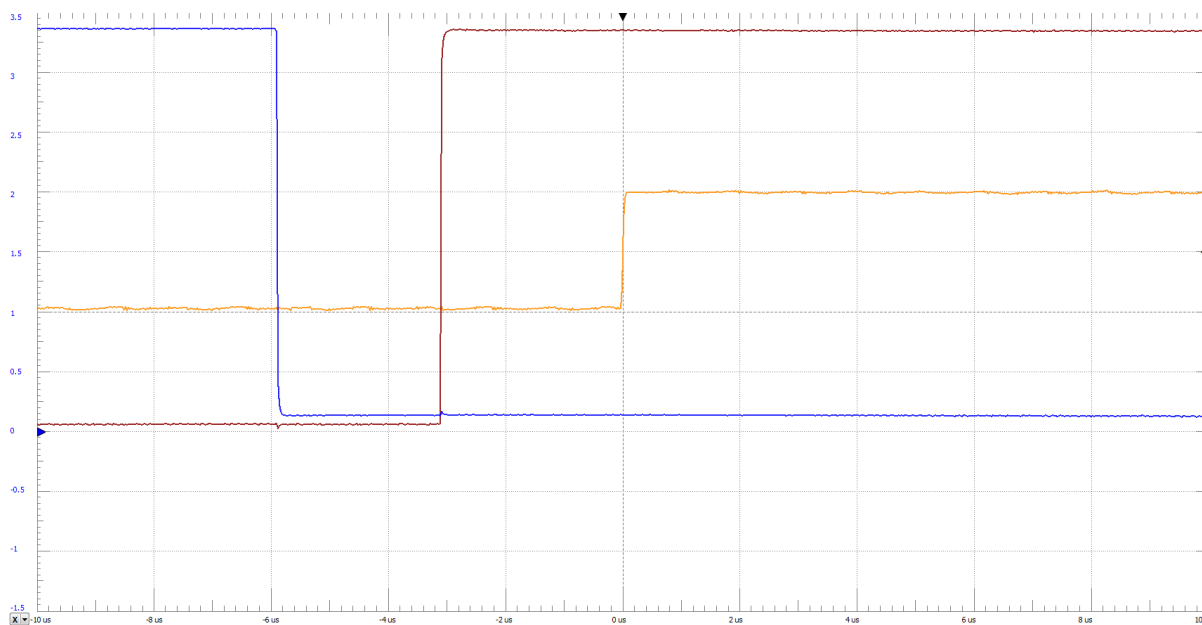


**Фиг.6.15.** Промяна на температура на реална седалка (червено) и симулация (синьо) след промяна на желаната температура (лилаво)

Изследвана е възможността за подобряване на работа на системата и постигане на по-високи честоти на симулация чрез заменяне на най-натоварените платформи. Същата методология е използвана за заменяне на 16 битов микроконтролер с максимална честота 40 MHz с 32 битов, работещ на 168 МН. За целта предложеното на фиг.6.8 решение е реализирано, като за Симулатор 1 е използвана хардуерна платформа, разполагаща с ЦАП. Най-натоварения Симулатор 5 е реализиран чрез STM32F4. В резултат е реализиран експеримент, базиран на принципна схема от фиг.6.16.

За симулираните сигнали е измерено максимално закъснение около 6 микросекунди, значително по-малко от измерените в предходния експеримент 17 микросекунди. Това закъснение съвпада с изчисленото от предложената методология.





**Фиг.6.18.** Максимални закъснения на симулираните цифрови сигнали (син, кафяв) и симулирания аналогов сигнал (оранжев).

Използването на по-голяма изчислителна мощност доведе до повишаване на максималната скорост на симулация на системата от 1250 на 5500 Hz, т.е. над 4 пъти по-бързо. По-големи изчислителни мощности ще доведат до още по-големи честоти на симулация, но достигнатите от индустриалните HIL системи 7.8 MHz никога няма да бъдат реализирани. Причина за това е ограничената скорост на комуникация с конфигуриращия компютър, ограничената изчислителна мощност, но най-вече допустимото максималното закъснение на симулирания сигнал, дефинирано в (5.18), което за период на симулация  $T_{\text{сим}} = 13 \text{ s} \cdot 10^{-9}$  е непостижимо малък. Сложни, специфични изчисления, като симулация на автомобилния трафик, също няма да бъдат по силите на микроконтролер. За това се използват бързи и специфични изчислителни машини.

Предложената методология преодолява три от най-основните проблеми на този тип системи – висока цена, невъзможност за повторно използване и трудна и бавна интеграция.

Настрана от нестандартната реализация от втория експеримент, HIL системата е разработена и интегрирана бързо, благодарение на описаната формална методика.

Всички експерименти показват, че предложената архитектура е в състояние да бъде използвана за разработване и имплементиране на различни като тип и комплексност HIL системи.

## НАУЧНО-ПРИЛОЖНИ И ПРИЛОЖНИ ПРИНОСИ

### • НАУЧНО-ПРИЛОЖНИ

1. Предложено и е дефинирано всеобхватно определение за HIL система. Същата е типизирана и параметризирана във връзка с различните си реализации и интеграции. Формулирана и доказана е теза, че универсалност и прогнозируемо и оптимално време за интеграция на HIL система, могат да бъдат постигнати единствено чрез формални методи за разработка.
2. Предложена е методика за параметризиране и изследване на вграден код. Изследвана е оптималността на генерирания от Embedded Coder код, като е изведена връзката между оптималността на кода и алгоритъма, който той описва. Доказана е липсата на необходимост автоматично генериран код за реализация на тестова система да бъде сертифициран според стандартите за безопасност в автомобилната индустрия.
3. Предложена е нова, обобщена и с възможности за автоматично разширение системна архитектура, способна да осигури необходимите за работата на HIL система параметри. Дефинирани са изискванията към подсистемите и Предложено е решение за разширяването и според нуждите на приложението.
4. Предложена е опростена модулна софтуерна архитектура, която дава възможност за пълното автоматизиране на процеса на разработка на hardware-in-the-loop системи на сигнално ниво. Постигнато е унифициране на софтуерната структура, което дава възможност за всеобхватна реализация на HIL системи и автоматичното пресмятане на работоспособността.
5. Създадена е методология за определяне на работоспособността на hardware-in-the-loop системи на сигнално ниво, реализирани чрез разпределена архитектура, чрез описващи системата зависимости и теоретично определяне на времето за изпълнение на подсистемите.

Тези приноси се отнасят до използване на съвременни методи за автоматизация на процеса на разработка и интегриране на hardware-in-the-loop системи за автомобилната индустрия чрез моделно базирания метод за разработка и генериране на код.

### • ПРИЛОЖНИ

1. Доказана е приложимостта на предложената методология за създаване и интегриране на hardware-in-the-loop системи на сигнално ниво, по универсален, независим от вида на тестовото устройство, начин.
2. Доказана е възможността предложената система да бъде разширявана, обхващайки различен брой и тип интерфейси, както и да бъде повишена нейната производителност във връзка с честота на симулация.
3. Изследвани са ограниченията на предложената система, свързани с ограничената изчислителна мощност и липсата на хардуерно реализиран паралелизъм.

## СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИОННИЯ ТРУД

1. **Nikolay Brayanov**, Anna Stoyanova, *Review of hardware-in-the-loop - a hundred years progress in the pseudo-real testing*, ELECTROTECHNICA & ELECTRONICA, ISSN 0861-4717, Vol. 54. No 3-4, 2019, pp. 70-84.
2. **Nikolay Brayanov**, Anna Stoyanova, *Evaluation of Model-based Code Generation for Embedded Systems – Mature Approach for Development in Evolution*, World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering, Vol:13, No:8, 2019, pp. 463-468, <https://publications.waset.org/10010703/pdf>, doi: 10.5281/zenodo.3455705
3. **Николай Браянов**, Ана Стойнова, *Анализ на методи за сертифициране и квалифициране на помощни програми за генериране на код в автомобилната индустрия*, Сб. Доклади от XIII Национална конференция с международно участие ЕЛЕКТРОНИКА 2016, 12–13 май 2016, София, стр. 76 – 81
4. **Николай Браянов**, *Сертифицираност и квалифицираност на помощни програми за генериране на код за автомобилната индустрия*, SCIENTIFIC TECHNICAL UNION OF MECHANICAL ENGINEERING, Year XXIV, Issue 8 (194), June 2016, ISSN: 1310 – 3946, стр. 45-48
5. **Nikolay Brayanov**, Arno Eichberger, *Automation in Hardware-in-the-loop Units Development and Integration*, Proceedings of the 19th IEEE International conference on Software Quality, Reliability, and Security (QRS 2019), 22-26 Jul., Sofia, Bulgaria, 2019, pp. 191 – 197 doi: 10.1109/QRS-C.2019.00047.
6. **Nikolay Brayanov**, *Evaluation of Modern Model Driven Architecture Software Development*, 2018 IEEE XXVII International Scientific Conference Electronics – ET, 13-15 Sept. 2018, Sozopol, Bulgaria, pp. 1-4, DOI: 10.1109/ET.2018.8549669
7. George Popov, **Nikolay Brayanov**, Alexander Balevsky, *Estimation of Hard Real-Time System Workability*, 2018 International Conference on High Technology for Sustainable Development (HiTech), 11-14 June 2018, Sofia, Bulgaria, pp. 1-3, DOI: 10.1109/HiTech.2018.8566346
8. **Nikolay Brayanov**, Anna Stoyanova, *Multi-Microcontroller Emulation of Sensors, Actuators and Systems for Flexible Integration of Complex Devices*, 41st International Spring Seminar on Electronics Technology (ISSE), 16-20 May 2018, Zlatibor, Serbia, pp. 1-4, DOI: 10.1109/ISSE.2018.8443703

# **AUTOMATION OF THE PROCESS OF DEVELOPMENT AND INTEGRATION OF HARDWARE-IN-THE-LOOP SYSTEMS FOR THE AUTOMOTIVE INDUSTRY**

**Nikolay Petev Brayanov**

*The last five decades in people's lives have been marked by the advent of computer technology. It has come a long way between expensive, large and inefficient platforms to today's fast, multi-core, highly integrated systems. Combining software, hardware and mechanics in a real-time cyber-physical system has made validating modern devices a complex, and sometimes impossible, operation that takes up much of the development process. The only real and implemented solution so far is the hardware-in-the-loop (HIL) system. The lack of a standard method for implementation and integration leads to specific staffing requirements, an unknown integration period and, as a result, high cost, significant factors hampering mass integration. The widespread use of HIL would lead to automation and cost reduction of testing, as well as product quality improvement. Automated approach is needed to develop and integrate HIL systems.*

*The assessment of the need for certification of the generated code in connection with its integration into an automotive HIL system has shown that it is not applicable. This is confirmed by the fact that currently available industrial solutions do not meet this criterion.*

*Model based development's advantage in terms of flexibility, reusability and integration across platforms has been identified. A study found similar performance parameters of manually done and generated code. The generated is found to be 12% faster and take up 7% more memory. This indicator varies according to the implemented functionality and style of development and the results are indicative. The studies carried out provide evidence to support the possibility of using automatically generated program code to test systems in the context of safety standards in the automotive industry.*

*Regarding the definition of a system architecture that enables the automation of the process of development and integration of HIL systems, it has been found that the most commonly used FPGAs and CPUs could hardly be integrated into an automated process for the development and integration of HIL systems. In this regard, the microcontroller is chosen as a flexible, versatile, mass and cheap. The lack of natural concurrency of operations in this platform defines the need for a distributed system architecture.*

*A unique system architecture based on microcontrollers connected in a distributed, synchronized, infinitely expanding network is proposed. The limitations of the implemented system architecture are mainly related to the hardware platforms used, their processing power and interfaces.*

*Modern software architecture solutions are proposed for the two types of subsystems part of the proposed HIL. Optimization leads to a universal form consisting of standard modules and a configuration that can be formally created, ie. generated.*

*Performance models for the two types of subsystems involved in the proposed HIL implementation have been created. The method is based on already known system load equations applied in the context of development and system-specific requirements and constraints. The merit of the method is its proven accuracy, enabling it to be used for reliable performance assessment. The proposed innovative methodology is able not only to automatically develop, but also to evaluate the proposed system.*

*All experiments show that the proposed architecture is capable of being used to develop and integrate HIL systems different in type and complexity. The ability to improve system performance and achieve higher simulation frequencies by replacing the busiest platforms has been proven. Thus the simulation frequency was increased from 1250Hz to 5500Hz. Higher processing power will lead to even higher simulation frequencies, but the 7.8 MHz industrial HIL systems will never be realized. The reason for this is the limited communication speed with the configuring computer, the limited computing power, but most of all the maximum delay of the simulated signal defined in (5.18), which for a simulation period of 13 us is insanely small. Complex, specific calculations, such as simulation of car traffic, will also not be beyond the power of a microcontroller. Fast and specific computing machines are used for this.*

*The proposed methodology overcomes three of the major problems of this type of system - high cost, inability to reuse, and difficult and slow integration.*