

РОЗРОБЛЕННЯ МОБІЛЬНОЇ РОБОТОТЕХНІЧНОЇ СИСТЕМИ НА ОСНОВІ FPGA

© Теслюк В. М., Матвійчук К. В., Романюк А. Б., 2016

Розроблено структуру проєктованої мобільної робототехнічної системи на базі FPGA. Розроблено VHDL-моделі підсистем керування МРТС, програмне забезпечення системи та описано особливості фізичної моделі мобільної РТС на основі FPGA. Реалізовано макетний взірць на базі FPGA, що дає змогу дослідити розроблене програмне забезпечення.

Ключові слова: мобільні робототехнічні системи, ПЛІС, VHDL-моделі, UML діаграма, RTL схема, підсистема керування.

V. Teslyuk, K. Matviichuk, A. Romaniuk
Lviv Polytechnic National University

DEVELOPING OF MOBILE ROBOTIC TECHNICAL SYSTEM BASED ON FPGA

© Teslyuk V., Matviichuk K., Romaniuk A., 2016

In the article the structure of the designed FPGA based mobile robot technical system is described. For the design the block-hierarchical approach was used. This system includes the following elements: - the MicroBlaze soft, which is a 32-bit programmable RISC processor with Harvard architecture, in which the processor has separated memory commands and data memory, motor control subsystem, video processing subsystem, radio module control subsystem, sensor control subsystem, the subsystem of obstacle identification. The structure that was built is based on a modular principle that allows you to quickly modify and develop MRTS.

To solve the task of the system functioning in the conditions of a rapidly changing environment hardware implementation FPGA based MRTS subsystems were used. Complex VHDL models of control subsystems of MRTS for the schemotechnical level of design were developed. Obstacle detection and engine control subsystems based on RTL circuits and modeled by means of the VHDL languages and their working principles, and detailed descriptions of the individual valves were presented.

System software is divided into client and server parts. The client part is placed on a computer, this is a program written in C++ programming language, which communicates with a mobile RTS. The main component of the program for developed model is an RTS control subsystem, which is provided in UML diagram. The features of the physical model of mobile RTS based on the FPGA are described. The sample model is implemented in FPGA, which gives the opportunity to explore the developed software. In the process of implementation was used Spartan 3E, 500 K capacity logical element in the FG320 package. Proposed the implementation of a

subsystem of the MRTS in the future will develop the RTS which will have high performance at low weight and size and a low price.

Key words: mobile robotic technical systems, FPGA, VHDL-models, UML diagram, RTL circuit, control subsystem.

Вступ

Сьогодні активно розвиваються мобільні робототехнічні системи (МРТС), які дають змогу замінити людину в агресивному, несприятливому чи небезпечному для життя навколишньому середовищі [1–3]. Особливості функціонування в таких умовах характеризуються не тільки наявністю нечітких та неструктурованих даних від підсистеми давачів, але й діями, які можуть спричинити помилки в роботі системи або її знищення. Отже, такі системи повинні забезпечити високу швидкодію реакції РТС на швидкозмінні умови навколишнього середовища та запобігти небезпечним ситуаціям.

Проведений аналіз наявних МРТС показує, що наведені рішення дорогі [11–14]. У цій роботі для розв’язання таких задач використовуємо апаратну реалізацію підсистем МРТС на основі ПЛІС, яка дає змогу покращити масогабаритні параметри та ціну. Відповідно роботи з розроблення моделей та засобів підсистем для РТС на основі ПЛІС є актуальними.

Зазвичай у процесі розроблення МРС використовують блочно-ієрархічний підхід [4]. Зокрема, на системному рівні проектування використовуються моделі на основі мереж Петрі [7], систем масового обслуговування та інші [7, 8]. Для побудови моделей, які дають змогу врахувати нечіткі та неструктуровані дані підсистеми давачів, використовують моделі на основі штучних нейронних мереж [9–11].

Для проектування на компонентному рівні використовуються такі системи, як ANSYS, COMSOL, SolidWorks та інші [13], а для схемотехнічного – VHDL-мова для опису цифрових елементів системи [1–2, 6]

Побудова структури системи МРС

Для реалізації МРТС розроблено структуру системи, яку зображено на рис. 1. Ця схема містить такі компоненти:

- Ядро – MicroBlaze soft, що являє собою 32-розрядний RISC програмований процесор з гарвардською архітектурою, за якої процесор має розділені пам’ять команд і пам’ять даних. Його основним завданням є виконання загальних завдань, таких як управління рештою підсистем, приймання команд від панелі керування та перенаправлення їх на підсистеми МРТС.
- Підсистема керування двигунами. Відповідає за керування швидкістю двигунів та поворотами. На вхід системи надходять команди та значення швидкості, які обробляються та у вигляді імпульсів передаються до драйвера двигунів постійного струму.
- Підсистема опрацювання відеозображення. Відповідає за зв’язок та керування відеокамерою. Виконує стиснення “сирого” відеопотоку з камери для подальшого передавання через радіоканал.
- Підсистема керування радіомодулем. Відповідає за формування пакетів даних, що містять виміри значень акселерометра, гіроскопа та магнітометра, для подальшого

надсилання цих даних через радіомодуль до мікропроцесора та передавання команди з панелі керування до ядра.

- Підсистема управління датчиками та обчислення відстані до перешкоди на основі даних з ультразвукового датчика відстані. Виконує перетворення часових відліків на відстань.
- Підсистема визначення перешкод на основі даних з ультразвукового віддалеміра. Формує можливі варіанти обходу перешкоди.

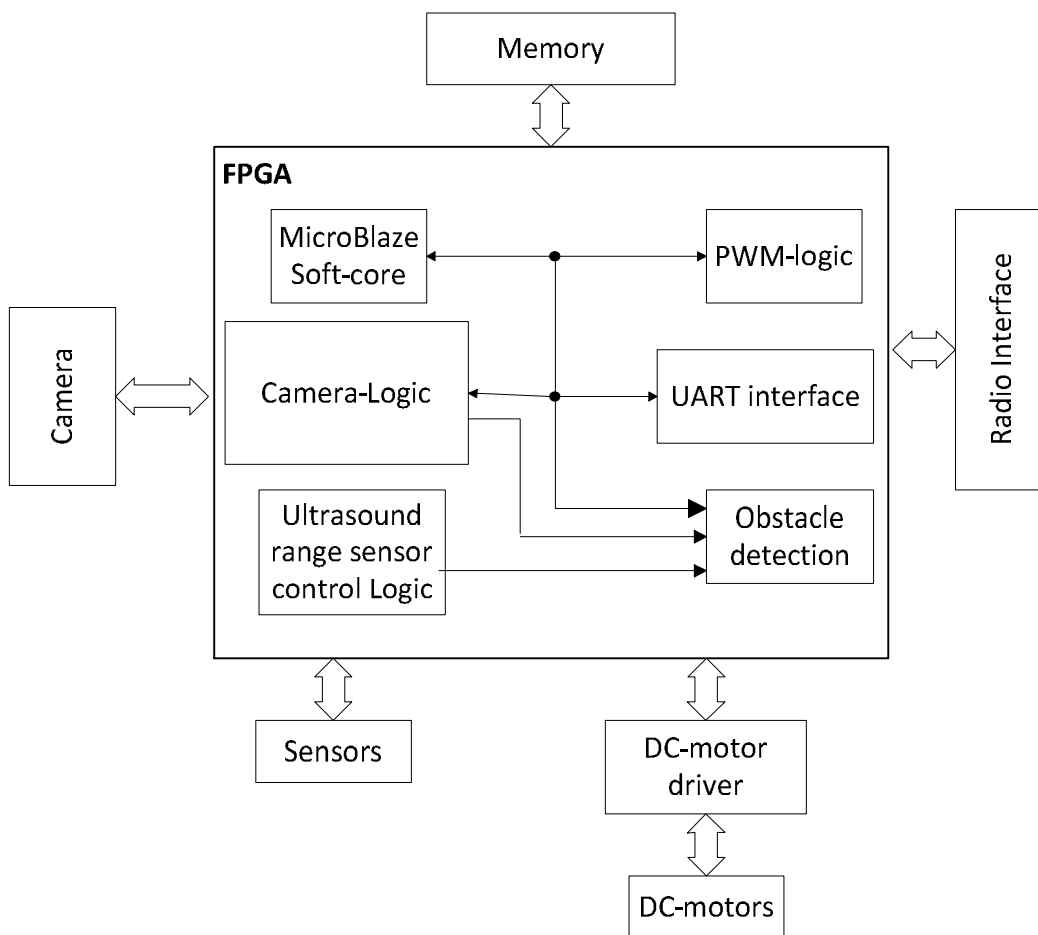


Рис. 1. Структура системи МРТС

Побудована структура ґрунтується на модульному принципі, що дає змогу швидко модифікувати та вдосконалювати МРТС.

Програмне забезпечення макета підсистеми керування на основі ПЛІС

Використовуючи описану вище структуру системи, було спроектовано макетний зразок для функціонування якого розроблено програмне забезпечення, яке поділено на клієнтську та серверну частини. Клієнтська частина розміщена на комп'ютері, це програма, написана на С++, яка взаємодіє з мобільною РТС. Взаємодіючи через порт вводу/виводу, вона запускає роботу системи та викликає необхідні для виконання завдань компоненти підсистем. Основним компонентом програми для розроблюваного макета РТС є підсистема керування, представлена на UML діаграмі класом CarController (рис. 2). Клас CarController інкапсулює в собі основні механізми взаємодії із апаратним забезпеченням мобільної робототехнічної системи. Цей клас використовує клас QTextSerialPort для взаємодії з послідовним портом

вводу–виводу, що забезпечує ефективний зв’язок. Для реалізації зв’язку вибрано послідовний інтерфейс обміну байтовою інформацією, оскільки він дешевий, не потребує проміжних вузлових компонентів, які формують пакети даних для передавання, що є недоліком USB, простий для передавання команд у реальному часі.

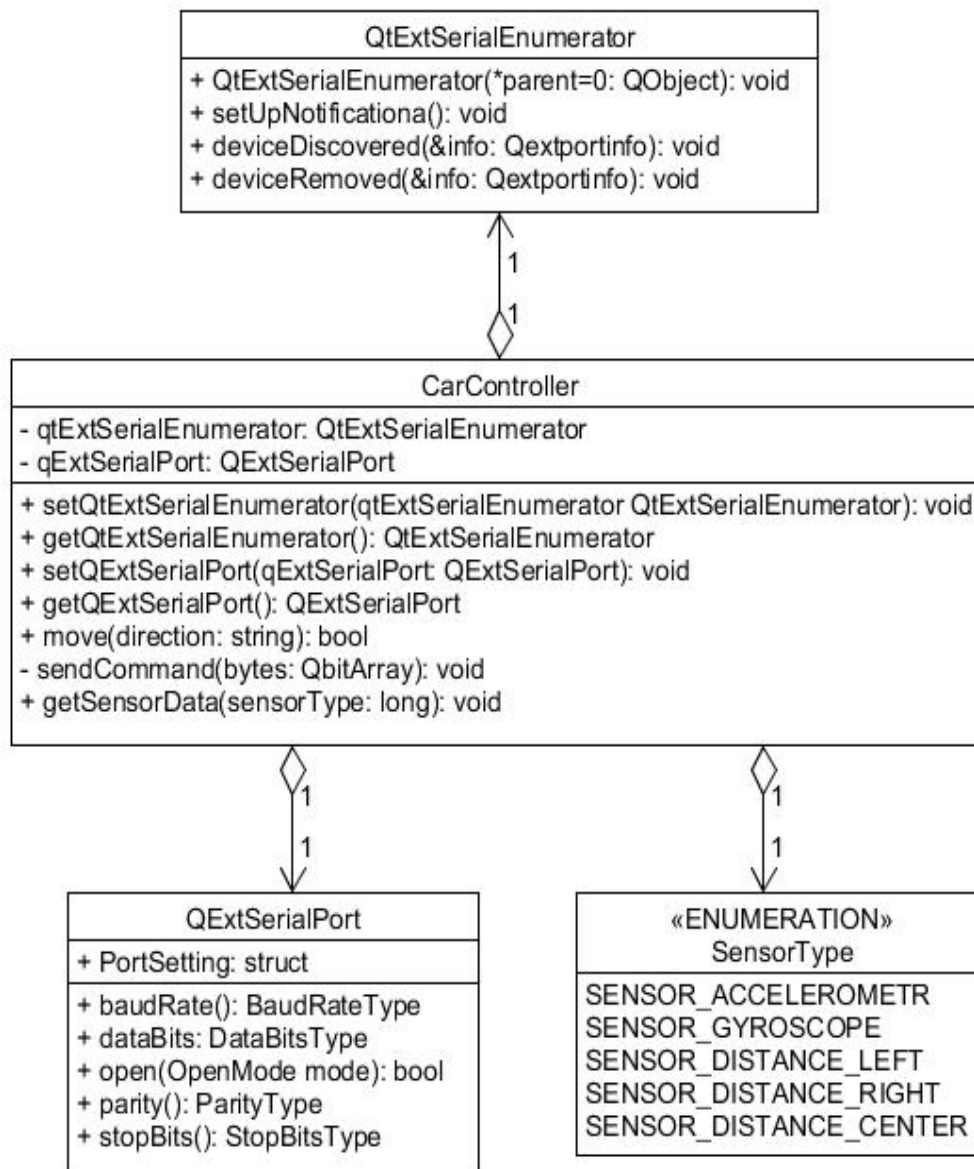


Рис. 2. Діаграма класів підсистеми керування PTC

VHDL-моделі підсистем керування MPC на основі ПЛІС

Для моделювання роботи проєктованих підсистем MPTC використано мову VHDL, яка дає можливість застосовувати “функціональний” опис, без конкретизації структури схеми [4, 5]. І навіть більше, описано поведінку системи та деталізовано опис окремих вентилів та алгоритмів роботи.

Для прикладу підсистема визначення перешкод та управління давачами працює на основі RTL схеми, зображеної на рис. 3, яка змодельована засобами мови VHDL (рис. 4) За визначення перешкод відповідає блок distance_calculation. Цей блок взаємодіє з блоком trigger_generator, який напряму підключений до входу давача відстані.

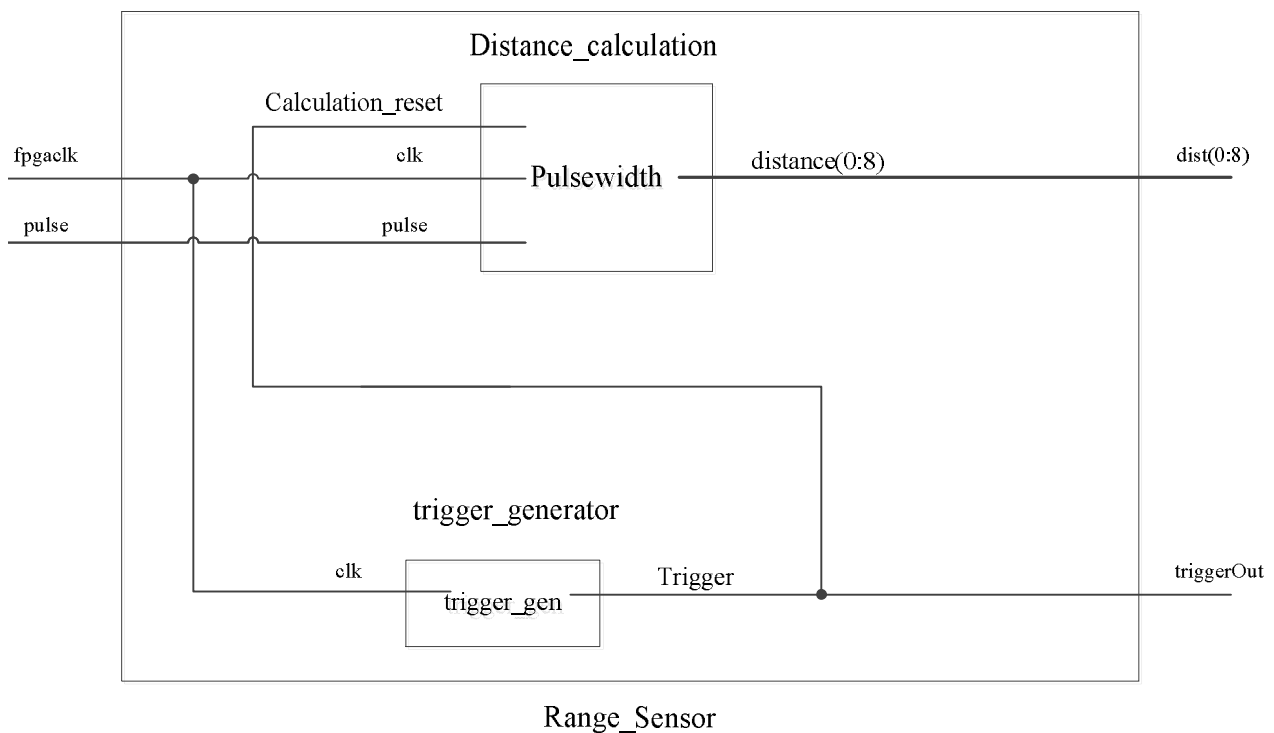


Рис. 3. RTL – схема підсистеми визначення перешкод

На вхід clk системи подається тактовий сигнал частотою 50 МГц, на вхід ультразвукового давача – імпульс тривалістю 10 мс, водночас значення лічильника скидається. По передньому фронту сигналу ЕЧНО з давача запускається лічильник, який обчислює тривалість імпульсу. На виході лічильника отримуємо значення тривалості імпульсу в мілісекундах. На основі тривалості імпульсу обчислюється відстань до перешкоди. Отримане значення встановлюється на виході distance і відображає відстань до об'єкта в міліметрах.

Представлений фрагмент коду, що описує підсистему мовою VHDL (рис. 4), виконує такі дії: визначає тривалість імпульсу на виході ультразвукового давача, проводить налаштування лічильника та здійснює пошук відстані, на якій розміщена перешкода, зберігаючи в пам'яті результат [3, 5–7].

Підсистема керування двигунами складається із двох блоків PWM та Control_motor, що зображено на рис. 5. На вхід підсистеми подається тактовий сигнал clk, частотою 50 мГц. На вхід speedcontrol подається 8-бітне значення швидкості. Блок PWM має внутрішній лічильник, який з кожним тактом збільшується на одиницю, на виході PWMout маємо логічну одиницю. В момент, коли значення лічильника та швидкості стануть рівними, на виході PWMout отримаємо логічний нуль. На блок контролю двигунів подається вхідна команда, яка містить інформацію про подальший напрям руху. Команда складається з чотирьох бітів:

- 1 – рух вперед;
- 2 – рух назад;
- 3 – повернути наліво;
- 4 – повернути направо.

На виході маємо згенеровану команду, яка подається на драйвер двигунів. Перші два біти формуються за допомогою операції логічного множення між значеннями PWMout та input_comand, наступні два залишаються без змін.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;
entity Distance_calculation is
  Port ( clk : in STD_LOGIC;
        Calculation_reset : in STD_LOGIC;
        pulse : in STD_LOGIC;

        distance : out STD_LOGIC_VECTOR (8
            downto 0));
end Distance_calculation;
architecture Behavioral of
Distance_calculation is
component Counter is
  generic(n : POSITIVE:=10);
  Port ( clk : in STD_LOGIC;
        enable : in STD_LOGIC;
        reset : in STD_LOGIC;
        Counter_output : out
            STD_LOGIC_VECTOR (n-1 downto 0));
end component;
signal Pulse_width:
STD_LOGIC_VECTOR(21 downto 0);
begin
CounterPulse: Counter generic map(22) port
  map(clk,pulse,not
  Calculation_Reset,Pulse_width);
Distance_calculation : process(pulse)
  variable Result : integer;
  variable multiplier :
STD_LOGIC_VECTOR(23 downto 0);
  begin
    if(pulse = '0') then
      multiplier := Pulse_width * "11";

      Result :=
to_integer(unsigned(multiplier(23 downto 13))

      if (Result > 458) then
        distance <= "11111111";
      else
        distance <=
STD_LOGIC_VECTOR(to_unsigned(Result,9));
      end if;
    end if;
  end process Distance_calculation;
end Behavioral;

entity Counter is
  generic(n : POSITIVE:=10);
  Port ( clk : in STD_LOGIC;
        enable : in STD_LOGIC;
        reset : in STD_LOGIC;
        counter_output : out
            STD_LOGIC_VECTOR (n-1 downto 0));
end Counter;
architecture Behavioral of Counter is
signal count: STD_LOGIC_VECTOR(n-1
downto 0);
begin
process (clk, reset)

  begin
    if(reset='0')
      then
        count <= (others => '0');
      else
        if(clk'event and clk = '1') then
          if (enable = '1') then
            count <= count + 1;
          end if;
        end if;
      end process;
      counter_output <= count;
    end Behavioral;

```

Рис. 4. Приклад фрагмента коду VHDL моделі підсистеми

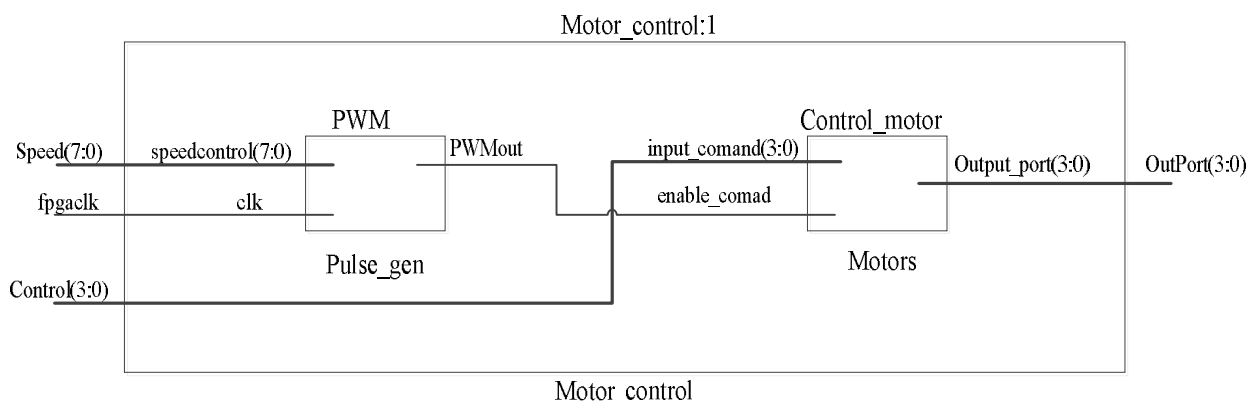


Рис. 5. RTL – схема підсистеми керування двигунами

Особливості фізичної моделі підсистеми керування МРС на основі ПЛІС

Розроблені моделі підсистем МРТС реалізовано на базі ПЛІС. Вони дали змогу забезпечити реалізацію складних функцій, водночас зменшити витрати та терміни виробництва макетного зразка. Окрім того, системи на основі вбудованих мікроконтролерів та архітектур, що перебудовуються, забезпечують фізичний рівень передавання даних. Необхідно додати, що функціональність системи може бути задана безпосередньо на місці відповідно до вимог замовника-користувача.

У процесі реалізації макетного зразка (рис. 6.) використано плату Spartan 3E ємністю 500K Logic Element в корпусі FG320, яка має низьку вартість, сумісна з MicroBlaze Embedded від Xilinx та має USB і JTAG інтерфейси паралельного програмування [6]. Зразок працює за таким алгоритмом. Після запуску програми і встановлення зв'язку з мікропроцесором, який надсилає команди до підсистем МРТС, отримавши команду, система розпочинає свій рух та обробляє дані що надходять від давачів. Підсистема визначення перешкод отримує дані від трьох сенсорів, які розташовані спереду МРТС. Опрацювавши отримані значення, визначає подальший маршрут пристрою. Якщо перешкоди не виявлено, рух продовжується у заданому напрямку. За наявності перешкоди підсистема знаходить найкращий варіант обходу та надсилає сформовану команду до мікропроцесора. Опрацювавши дані, мікропроцесор надсилає підтвердження виконання команди, або, у разі значного відхилення від маршруту, заборону на виконання команди, після чого підсистема здійснює пошук іншого варіанта обходу перешкоди.



Рис. 6. Фізична модель системи

Паралельно підсистема керування радіомодулями зчитує та опрацьовує дані від акселерометра, гіроскопа та магнітометра, для орієнтації в просторі, з яких формує пакети даних та передає до мікропроцесора.

На основі отриманої інформації мікропроцесор формує команди та надсилає їх до підсистеми керування двигунами, задаючи подальший маршрут пересування.

Висновок

Розроблено структуру підсистеми визначення перешкод для МРТС на основі ПЛІС. Побудовано VHDL-моделі цієї підсистеми та наведено фізичну модель, яка дає змогу перевірити правильність та коректність розробленої підсистеми. Запропонована реалізація підсистеми МРТС в майбутньому дасть змогу розробити РТС, яка матиме високу швидкодію за малих масо-габаритних параметрів та низьку ціну.

1. Цмоць І. Г, Теслюк В. М., Ваврук І., Апаратні та програмні засоби для управління рухом рухомої робототехнічної системи // в: матер. 12-ї Міжнародної конференції The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM'2013. Львів-Поляна.

Лютый 19–23. 2013. 368 с. 2. Денисюк П., Матвійчук К., Дуда М., Теслюк Т., Кобилух Я. Технічне забезпечення для мобільної робототехнічної системи RoboCAD // Матер. 12-ї Міжнародної конференції *The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM'2013*, Львів–Україна. Лютый 19–23. 2013. С. 431–432. 3. Чень Юн-Хі, Ван-Вей. Проектування та реалізація робота для порятунку в аварійних ситуаціях, з використанням відкритого коду, прогресивних комп'ютерних інформаційних технологій та інженерії. Тези доп. в області електротехніки. 2015. № 352. С. 53–60. 4. Лінсетт Роджер, Шефер Карл Ф. та Асері Керрі. *VHDL: Опис та проектування обладнання*. Springer. Наука та медіа-бізнес. 2012. 5. Сепульведа Роберто та ін. Методика тестування і перевірки на VHDL механізму логічного висновку про наявність 2-го типу, через генератор системи Xilinx // *Еволюційне проектування інтелектуальних систем в області моделювання і контролю*. Springer Берлін Хейдельберг. 2009. С. 295–308. 6. Чу Понг Р. Прототипування FPGA на основі прикладу VHDL: Xilinx Spartan-3 version. John Wiley & Sons. 2011. 7. Тан Ай-Хві. FALCON: Об'єднана архітектура для навчання пізнання і навігації // *IJCNN-04. IEEE Міжнародна об'єднана конференція по нейронних мережах*. 2004. № 4. С. 3297–3302. 8. Монмасн Е. Ідкхайне Л. Кірстеа М. Н. Бахрі І. Тісан А. і Наоар М. В. . *FPGAs в додатках промислового управління. Ведення промислової інформатики IEEE*. 2011. 7(2). С. 224–243. 9. Шудха Н., Мохан А. Р. Апаратно-продуктивне планування шляху робота на основі зображень в динамічному середовищі і його реалізація FPGA. *Промислова електроніка // IEEE Transactions on* 58.5. 2011: С. 1907–1920. 10. Проектування на основі ПЛІС контролера змінного режиму для робота-маніпулятора / Фарзін Пільтан та ін. // *Міжнародний журнал з робототехніки та автоматизації (IJRA)*. 2.3 2011. С. 173–194. 11. Кастіло Оскар. Програмні обрахунки для інтелектуального управління і мобільної робототехніки. Vol. 318. Springer. Наука та медіа-бізнес. 2010. 12. Вивчення дій сейсмічного робота за допомогою COMSOL Multiphysics / Файребог Самара Л. та ін. // *NAVAL ACADEMY ANNAPOLIS MD*. 2010. 13. Цмоць І., Ваврук І., Ткаченко Р., Теслюк В. Проектування і моделювання системи управління рухом мобільного робота на основі нечіткої логіки // *Міжнародний журнал перспективних досліджень в прикладних обчислювальній техніці і технологіях (IJARCET)*. 2014. № 3, Issue 9. P. 3143–3147. 14. Головатий А., Теслюк В., Кривий Р. VHDL-Амс комплексна модель мембранного типу мікро-акселерометра з дельта-сигма ($\Delta\sigma$) аналогово-цифрового перетворювача для схемотехнічного рівня проектування // *ECONTECHMOD*. 2015. № 4. no. 2. P. 65–70. 15. Часткове налаштування Xilinx для програмування систем: довідкове керівництво, Розділ 5 [Електронний ресурс]. URL: <http://www.xilinx.com/itp/xilinx8/de/dev/partial.pdf>. 16. Головатий А., Теслюк В., Лобур М. Verilog-AMS модель гребінчатого приводу чутливого елемента інтегрованого ємнісного мікроакселерометра для поведінкового рівня комп'ютерного проектування // *ECONTECHMOD*. 2014. Vol. 3, no. 4. P. 49–53.

References

1. I. G. Tsmots, V. M. Teslyuk, I. Vavruk, “Hardware and software tools for motion control of mobile robotic system”, in *Proc. of the 12-th International Conference on The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM'2013, Lviv-Polyana, Ukraine, Feb. 19–23, 2013*. 2. Pavlo Denysyuk, Kateryna Matviichuk, Marta Duda, Taras Teslyuk, Yaroslav Kobyliuk. *Technical Support For Mobile Robot System RoboCAD. / // in Proc. of the 12-th International Conference on The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM'2013, Lviv-Polyana, Ukraine, Feb.19–23, 2013* С. 431–432. 3. Yung-Hui Chen, Jyu-Wei Wang, *The Disaster Rescue Robot Design and Implementation Using Open Source, Advanced Multimedia and Ubiquitous Engineering, Lecture Notes in Electrical Engineering, Volume 352, 2015, 53–60*. 4. Lipsett, Roger, Carl F. Schaefer, and Cary Ussery. *VHDL: Hardware description and design*. Springer Science & Business Media, 2012. 5. Sepúlveda, Roberto, et al. “Methodology to test and validate a vhdl inference engine of a type-2 fis, through the xilinx system generator”. *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*. Springer Berlin Heidelberg, 2009. 295–308. 6. Chu, Pong P. *FPGA prototyping by VHDL examples: Xilinx Spartan-3 version*. John Wiley & Sons, 2011. 7. Ah-Hwee Tan. *FALCON: A fusion*

architecture for learning, cognition and navigation. // IJCNN-04. IEEE International Joint Conference on Neural Networks. 2004. Vol. 4. 3297–3302.

8. Monmasson, E., Idkhajine, L., Cirstea, M. N., Bahri, I., Tisan, A., & Naouar, M. W. (2011). FPGAs in industrial control applications. *Industrial Informatics, IEEE Transactions on*, 7(2), 224–243.

9. Sudha, N., and Arupukottai Rajarathinam Mohan. “Hardware-efficient image-based robotic path planning in a dynamic environment and its FPGA implementation”. *Industrial Electronics, IEEE Transactions on* 58.5 (2011): 1907–1920.

10. Piltan, Farzin, et al. “Design of FPGA-based Sliding Mode Controller for Robot Manipulator”. *International Journal of Robotics and Automation (IJRA)* 2.3 (2011): 173–194.

11. Castillo, Oscar. *Soft Computing for Intelligent Control and Mobile Robotics*. Vol. 318. Springer Science & Business Media, 2010.

12. Firebaugh, L.Samara, et al. *A Study of Seismic Robot Actuation Using COMSOL Multiphysics*. NAVAL ACADEMY ANNAPOLIS MD, 2010.

13. Ivan Tsmots, Iryna Vavruk, Roman Tkachenko, Vasyl Teslyuk *Designing and simulation a motion control system of mobile robot based on fuzzy logic // International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*. – 2014, Vol. 3, Issue 9. – P. 3143–3147.

14. A.Golovatyj, V.Teslyuk, R.KRYVYY *VHDL-Ams Model of Integrated Membrane-Type Micro-Accelerometer with Delta-Sigma ($\Delta\sigma$) Analog-To-Digital Converter for Schematic Design Level // ECONTECHMOD*. – 2015, vol. 4, no. 2. – P. 65–70.

15. *Partial Reconfiguration, Xilinx, Development System Reference Guide, Chapter 5*. <http://www.xilinx.com/itp/xilinx8/de/dev/partial.pdf>.

16. A. Holovaty., V.Teslyuk, M.Lobur *Verilog-AMS model of comb-drive sensitive element of integrated capacitive microaccelerometer for behavioral level of computer-aided design // ECONTECHMOD*. – 2014, vol. 3, no. 4. – P. 49–53.