

Янішевський Віталій Ігорович, інженер, керівник гуртка робототехніки, **Малиновський Євген Вікторович** завідувач природничо-математичним відділом, керівник гуртка аерофізики та космічних досліджень, **Одинець Катерина** вихованка гуртка аерофізики та космічних досліджень (Рівненська Мала академія наук учнівської молоді)

ОПТИМІЗАЦІЯ ПІДХОДУ ДО КЕРУВАННЯ ПОЗИЦІОНУЮЧИМИ ЕЛЕКТРОМЕХАНІЧНИМИ ПРИСТРОЯМИ НА БАЗІ КРОКОВИХ ДВИГУНІВ

Питання керування кроковими двигунами може здатися доволі простим, але використання крокових двигунів для керування телескопами виявило декілька проблемних місць. А саме, однозначна часова синхронізація команд, та точність мікропозиціонування. Водночас ставиться завдання мінімізації електронно-механічного комплексу для швидкого переміщення та довготривалої роботи в умовах обмеженого живлення.

Одним з напрямів є оптимізація байткоду керування позиціонуючими кроковими двигунами. Існує певний культ «комп'ютер може порахувати все». Це не так. Все що вміє комп'ютер чітко розписано в теорії автоматів. Дана теорія чітко розписує всі обчислювачі на чотири класи. Причому кожен наступний (складніший) завжди складається з простіших за класом, в екстремумі, комп'ютер – це не більше ніж величезна таблиця, в якій прописані всі його теоретично можливі стани обчислення – це послідовність переходів з одного стану, який називається «початкові умови» до іншого, який називається «результатом». Іншими словами, автомат класу «машина Тюрінга», теоретично може бути редукована до автомата «комбінаторна логіка» (таблиця), просто фантастично великих розмірів. При цьому в інформатиці існує закономірність: програеш в пам'яті – виграеш в швидкості.

Взяти з таблиці готове значення за індексом початкових умов – це значно швидше та менш енергозатратно, ніж щоразу проводити покрокові обчислення згідно з заданими алгоритмами. Таким чином, спираючись на твердження, що в машинах все – перелічуване (а значить – скінченне), можна зробити оптимізацію швидкодії, вивівши рутинні обчислення в таблиці.

Це рішення призведе до витрати колосальних об'ємів пам'яті. І все ж ми йдемо на це, в силу певних причин. Існує фізичне обмеження на швидкодію, відоме як «фазова швидкість світла», таким чином ускладнюючи обчислення ми збільшуємо час затримки на обробку. В той же час технічно ми маємо змогу збільшувати об'єм носіїв інформації при збереженні сталої швидкості

обміну інформації. Також в умовах керування локальним пристроєм (телескопом або групою телескопів) існує технічна можливість скласти таблиці на всі можливі положення позиціонування враховуючи роздільну здатність крокових двигунів. Тому, поки габарити носіїв не почали складати проблему з доставкою інформації на велику відстань однієї до іншої частини, машини будуть розвиватись саме в такому напрямку: величезні пристрої вибірки/зберігання на підставі вищезазначеного, можна зробити припущення, що й програмне забезпечення та взагалі метод програмування такого типу машин буде розбитий на два етапи: формування «рутинних таблиць» та їх максимально нескладне комбінування (чим більше етапів перетворення – тим менша швидкодія) в РМАНУМ ця концепція (спочатку все максимально розраховуємо, і лише потім – виконуємо) знайшла своє застосування в архітектурі контролерів керування телескопами.

Ми маємо всі підстави вважати, що найбільш вдалим рішенням буде використовувати UART в якості інтерфейсу керуючого контролера систем позиціонування на базі крокових двигунів. UART (Universal Asynchronous Reseiver-Transmitter). Апаратно реалізований UART присутній майже у кожному мікроконтролері, або відносно незначними зусиллями реалізується програмно [1; 2]. Хоча інтерфейс RS-232 (реалізація UART) вважається застарілим, тому часто відсутній в настільних ПК, та практично повністю відсутній у ноутбуках, існує достатня кількість та широке поширення недорогих адаптерів, що дозволяють приєднувати UART не тільки до настільного ПК через інтерфейс RS-232, а також і до ноутбуків за допомогою USB-UART адаптерів [2]. Для кишенькових та планшетних ПК, смартфонів з'єднання можливе за допомогою USB-OTG кабелів. Окрім того існують спеціальні адаптери, що дозволяють організовувати бездротове з'єднання керуючого пристрою з UART через мережу Bluetooth [3]. Технологія Bluetooth підтримує спеціальний клас пристроїв Bluetooth/Serial, що гарантує підтримку з'єднання UART із будь-яким керуючим пристроєм, який ініціює Bluetooth-з'єднання [4]. Трансляція UART можлива також через бездротову мережу WiFi, що у свою чергу з використанням технології Serial-over-TCP дозволяє трансляцію у мережу Інтернет, і як наслідок, дає можливість дистанційного керування із будь-якої точки світу [5].

UART та адаптери підтримуються усіма операційними системами за замовчуванням як спеціальний клас пристроїв Serial чи tty [6–8]. На рівні API операційної системи для передачі даних UART виглядає як файл послідовним доступом, причому, не має значення яким саме чином програмно чи апаратно здійснено з'єднання. Так, робота з UART, приєданого до послідовного порту ПК безпосередньо, програмно повністю ідентична роботі через віртуальний послідовний порт утворений технологією Serial-

over-TCP, який з'єднаний через сервер-ретранслятор, що через мережу інтернет приєднаний до мобільного пристрою через мережу 2G/3G/LTE, який у свою чергу з'єднано через адаптер USB-UART до того самого UART [3; 4; 6]. Неповний перелік апаратних адаптерів та перетворювачів інтерфейсу для UART зображено на рис. 1.

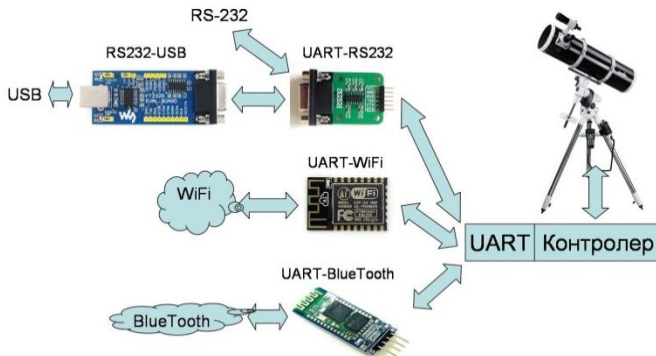


Рис. 1. Інфраструктура з'єднання UART з різними типами інтерфейсів.

Відмова від використання в складі системи засобів із вбудованими сучасними інтерфейсами типу USB продиктована бажанням отримати універсальне та гнучке рішення, досить просте, та достатньо недороге. Саме такими властивостями володіє комбінація UART+адаптер.

Особливості програмно-апаратної реалізації UART дозволяють використовувати його як джерело опорного часу високої точності [2], яке не залежить від платформи реалізації. Таким чином ми отримуємо змогу виконувати керування позиціонуванням з високою точністю в часі без використання додаткових синхронізуючих засобів. Дана властивість дуже корисна для керування у реальному часі, яке необхідне для керування телескопами, та дозволяє здійснювати керування без використання додаткових комп'ютерів, що працюють на операційних системах реального часу (операційна система реального часу vxWorks [9])

Список використаних літератури джерел

1. 044-UART. Применение в электронных проектах. URL: <http://www.getchip.net/posts/044-uart-primenenie-v-ehlektronnykh-proektakh/> (дата звернення: 15.05.2020).
2. TUTORIAL 2141 Determining Clock Accuracy Requirements for UART Communications. URL: <https://pdfserv.maximintegrated.com/en/an/AN2141.pdf> (дата звернення: 15.05.2020).

3. Using the HC06 Bluetooth Module Posted on June 19, 2013 by Erich Styger
технічна документація. URL: <https://cdn.instructables.com/ORIG/FX1N43N/IMXQP4WO/FX1N43NIMXQP4WO.pdf> (дата звернення: 15.05.2020).
4. HC-06 Module Data Sheet, Revision 2.2 технічна документація. URL: <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf> (дата звернення: 15.05.2020).
5. ESP8266EX Datasheet Version 6.0 Espressif Systems Copyright © 2018, технічна документація. URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf (дата звернення: 15.05.2020).
6. MSDN, Communications Resources. Електронний посібник програміста від Microsoft. <https://docs.microsoft.com/en-us/windows/desktop/DevIO/communications-resources-serial-programming> /Serial Linux. Електронний посібник програміста. URL: https://en.wikibooks.org/wiki/Serial_Programming/Serial_Linux (дата звернення: 15.05.2020).
7. Android OS SDK-UART. Електронний посібник програміста. URL: <https://developer.android.com/things/sdk/pio/uart> (дата звернення: 15.05.2020).
8. Experimental Physics and Industrial Control System (EPICS), електронний портал проекту. URL: <https://epics.anl.gov/base/index.php> (дата звернення: 15.05.2020).