

Co-funded by the
Tempus Programme
of the European Union



DesIRE

Н.В. Морзе, Л.О. Варченко-Троценко, М.А. Гладун

ОСНОВИ РОБОТОТЕХНІКИ



УДК 004.896(075)
ББК 32.816 я73
М 79

Рецензенти:

О.Ф. Тарасов, доктор технічних наук, професор, завідувач кафедри комп'ютерних інформаційних технологій Донбаської державної машинобудівної академії;

Г.В. Табунцик, кандидат технічних наук, доцент, професор кафедри програмних засобів Запорізького національного технічного університету

*Рекомендовано до друку Вченою радою
Київського університету імені Бориса Грінченка
(протокол № 8 від 29.08.2016 р.)*

Навчальний посібник видано за підтримки проекту
Tempus 544091-TEMPUS- 1-2013-1-BE- TEMPUS-JPCR
«Розробка курсів з вбудованих систем з використанням інноваційних віртуальних підходів для інтеграції науки, освіти та промисловості в Україні, Грузії, Вірменії»

Проект фінансується за підтримки Європейської комісії

Морзе Н.В.

М 79 **Основи робототехніки**: навчальний посібник / Н.В. Морзе, Л.О. Варченко-Троценко, М.А. Гладун. – Кам'янець-Подільський : ПП Буйницький О.А., 2016. – 184 с.

ISBN 978-617-608-063-3

Мета посібника – ознайомлення студентів та вчителів з технологією конструювання роботів та створення програм для керування ними; набуття ними навичок проектування, конструювання та програмування роботизованої техніки.

Зміст матеріалу відображає думку авторів.

Європейська комісія не несе відповідальності за використання інформації, що міститься в навчальному посібнику.

УДК 004.896(075)
ББК 32.816 я73

ISBN 978-617-608-063-3

- © Морзе Н.В., Варченко-Троценко Л.О., Гладун М.А.
- © Київський університет імені Бориса Грінченка, 2016
- © Tempus «DesIRE», 2013-2016
- © Дизайн-студія «Рута», художнє оформлення, макет

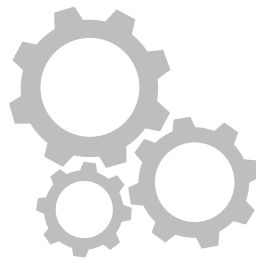
ЗМІСТ

| | |
|--|-----|
| ВСТУП | 5 |
| 1 STEM-ОСВІТА ТА ШЛЯХИ ЇЇ РОЗВИТКУ | 6 |
| 2 ОСНОВНІ ПОНЯТТЯ РОБОТОТЕХНІКИ..... | 18 |
| 2.1. Історія розвитку робототехніки | 18 |
| 2.2. Поняття «робота» та його складові..... | 30 |
| 2.3. Класифікація роботів та галузі їх застосування..... | 34 |
| 3 КОНСТРУКТОР LEGO MINDSTORMS EV3..... | 48 |
| 3.1. Конструктор Lego Mindstorms EV3..... | 48 |
| 3.2. Програмне середовище Mindstorms EV3..... | 53 |
| 3.3. Датчик та сенсори..... | 74 |
| Датчик дотику..... | 74 |
| Сенсор кольору..... | 76 |
| Ультразвуковий сенсор..... | 79 |
| Інші сенсори..... | 82 |
| 4 МІНІ-КОМП'ЮТЕР RASPBERRY PI..... | 86 |
| 4.1. Знайомство з Raspberry Pi..... | 86 |
| 4.2. Запуск Raspberry Pi..... | 97 |
| 4.3. Scratch для Raspberry Pi..... | 103 |
| 5 КОМПЛЕКТ FORMULA FLOWCODE BUGGY | 121 |

| | |
|--|-----|
| 5.1. Особливості комплекту Formula Flowcode Buggy..... | 121 |
| 5.2. Елементи вікна програмного середовища Flowcode | 123 |
| 5.3. Початок роботи у Flowcode..... | 128 |
| 5.4. Проекти у Formula Flowcode Buggy..... | 135 |
| ЛІТЕРАТУРА | 147 |
| СЛОВНИК ТЕРМІНІВ І ПОНЯТЬ..... | 151 |
| ДОДАТКОВІ РЕСУРСИ | 157 |

В багатьох країнах система інженерної, технічної та природничої освіти стоїть перед гострими викликами. Ефективність та вчасність вирішення проблем в галузі технологізації освіти є факторами, які визначають успішність розвитку національної економіки. У відповідь на ці виклики освіта повинна здійснювати реформування галузі. Враховуючи розвиток інформаційно-комунікаційних технологій, робототехніки, нанотехнологій виникає потреба у досвідчених фахівцях технічних та природничо-математичних дисциплін. STEM-освіта (Science Technology Engineering Math) є пріоритетною з причин затребуваності ІТ-фахівців, програмістів, інженерів, фахівців технологічних виробництв. Професії майбутнього пов'язані з технологічним виробництвом на стику з природничими науками. Існує необхідність навчання студентів та вчителів навичкам XXI століття шляхом використання STEM. Проте, більшість програм підготовки педагогічної освіти не націлені на STEM-освіту, а вчителі не мають чіткого розуміння з цього питання.

Одним з прикладів ефективного впровадження STEM-освіти є використання науково-дослідних проектів з використанням робототехніки. Швидко зростаюча потреба створення роботизованих систем, що використовуються в різних сферах суспільної діяльності, на виробництві та в побуті, припускає, що навіть звичайні користувачі повинні володіти знаннями в галузі робототехніки. Навчальний процес повинен бути практико-орієнтованим, мотивувати студентів до створення власних проектів та розробок, з метою розвитку критичного мислення, креативності, поширення проблемного підходу у навчанні та відходу від кліпового мислення. Це дозволить студентам та вчителям отримати досвід пізнавальної і творчої діяльності, зрозуміти сенс основних наукових понять, засвоїти взаємозв'язки між ними. Особлива роль при цьому відводиться шкільній робототехніці. Тому актуальними залишаються питання навчання основам робототехніки: які комп'ютерні плати використовувати, яке програмне забезпечення обрати?



1 STEM-ОСВІТА ТА ШЛЯХИ ЇЇ РОЗВИТКУ

В останні роки в США, Кореї та інших країнах починається процес переходу до нової моделі освіти в галузі науки і технології – STEM. За аббревіатурою STEM стоять чотири слова: science (наука), technology (технологія), engineering (інжиніринг) і mathematics (математика).

Потреба в новій моделі обумовлена тим, що в багатьох країнах спостерігається, з одного боку, нестача кваліфікованих фахівців для STEM-галузей, а з іншого – зниження частки молоді, яка обирає STEM для своєї кар'єри. Намагаючись розібратися в причинах зниження інтересу до вивчення предметів зі STEM-галузей і знайти способи вирішення проблеми, Міністерство підприємництва, інновацій та ремесел Великобританії виконало дослідження [30]. Автори описали ставлення молодих людей до вивчення природничих наук, математики та технології в школі і до майбутньої кар'єри в галузі STEM. Якщо спробувати описати результати трьома короткими фразами, то це буде: «STEM не для мене, занадто важко», «Це не модно», «STEM-освіта не пов'язана з реальним життям».

Автори міжнародного огляду [19] відзначають, що STEM-освіту розробники програм бачать по-різному. Важливим є те, що за будь-якої моделі інтеграції STEM у шкільний процес переважає системний підхід вивчення цих дисциплін. Міждисциплінарні зв'язки і акцент на прикладний характер підкреслюються у визначенні STEM-освіти даними Elaine J. Hom [8]: «STEM – це навчальна програма, заснована на ідеї навчання дітей чотирьом предметами (наука, технологія, інженерія та математика) як прикладним і пов'язаним між собою». STEM відрізняється від традиційного навчання природничих наук і математики впровадженням змішаного середовища навчання і демонстрацією того, як наукові методи можуть використовуватися в повсякденному житті.

Наука. В цій галузі людської діяльності в рамках концепції STEM охоплюються питання лише природничих наук, які можна розділити на декілька категорій, що базуються на тій частині всесвіту, яка вивчається – фізика, хімія, географія, астрономія.

Технології. Використання інформаційно-комунікаційних технологій.

Інженерія. Будування, механіка, електротехніка, обробка матеріалів.

Математика. Виступає як інструмент для технологій та науки, охоплює питання з арифметики, алгебри, початків аналізу, геометрії тощо.

Відповідно до стандартної класифікації професій США, що належать до STEM, є 6 груп:

- менеджмент;
- обчислювальна техніка та математика;
- архітектура та інженерія;
- науки про життя, фізичні та соціальні науки;
- освіта, професійна підготовка і бібліотечна справа;
- торгівля.

Відповідно до доповіді My College Options ринок праці США в галузі STEM у 2018 р. розподілятиметься так [24]:

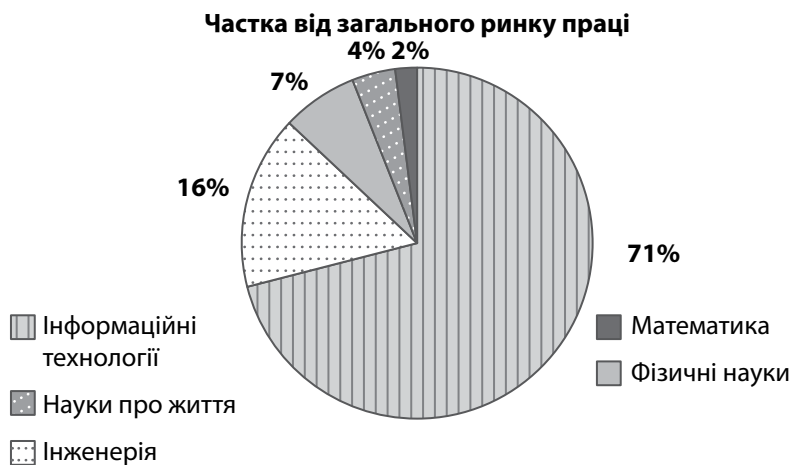


Рис. 1. Розподіл ринку праці США

Не всі робочі місця вимагають для початківців в галузі STEM вищої освіти, більша частина з них не потребують диплома бакалавра, але у свою чергу фахівці з дипломом мають переваги у працевлаштуванні. Попит на спеціалістів в галузях, пов'язаних зі STEM, є характерним і для інших розвинених країн.

У Фінляндії вважають: якщо молода людина вже на виході зі школи володітиме актуальним запасом практичних знань з урахуванням всіх сучасних комп'ютерних технологій та навичками ефективного пошуку інформаційних даних, то можна очікувати, що вона принесе користь не лише собі, а й державі. Тому було запроваджено експеримент навчання молоді з 15 років не окремих наук, а професійних курсів, що мають пряме відношення до реального життя [42].

Однією з альтернатив STEM-освіти, яка зараз стає популярною, є STEAM-освіта (наука, технологія, інженерія, мистецтво та математика). Засновницею цього напрямку вважається G. Yakman, яка висловлює думку про те, що «Наука і техніка інтерпретуються за допомогою інженерії і мистецтва, все це відбувається в розрахунку на математичних елементах» [29].

STEAM-освіта передбачає окрім надання предметно-специфічних знань, пов'язаних з розвитком дедуктивного, індуктивного та логічного мислення – збільшення творчості, розвитку критичного мислення, гнучкості, адаптивності, соціальних та крос-культурних навичок, пов'язаних з мистецтвом. Таким чином, STEAM-освіта спрямована на створення всебічно розвинених фахівців, які є STEM- грамотними, творчими та інноваційними. В Південній Кореї Міністерство освіти, науки і технологій вже прийняли STEAM-освіту як засіб підвищення інтересу студентів до предметів STEM і в результаті країна отримала збільшення впливу науки і технологій у південно корейському суспільстві [5].

Національний науковий фонд (NSF) і Національний фонд мистецтв (NEA) в США після двостороннього обговорення дійшли до думки, що додавання мистецтва (A) до STEM явно недостатньо. Також слід врахувати навички мислення, втілені при читанні і письмі (англ. мовою Reading and wRiting), тому STEAM

трансформується в STREAM. Грамотність є важливим компонентом для науки та технологій, адже фахівці мають вміти писати проекти та експерименти, складати звіти, правильно спілкуватись та співпрацювати.

Як показує досвід штату Квінсленд (Австралія), зусилля щодо проведення реформ в напрямках STEM- та STEAM-освіти сприяє підвищенню наукової грамотності та успішності молоді [23]. Дослідження показують, що учні в молодшій школі можуть розвивати їх обчислювальне мислення за допомогою використання комп'ютерного програмування і робототехніки. Математика надає можливість школярам розвивати свої навички міркувань за допомогою удосконалення алгоритмічного мислення [23].

Для молодшої школи з використанням STEM- та STEAM-освіти характерне поєднання активного навчання та практики під керівництвом вчителя (рис. 2).

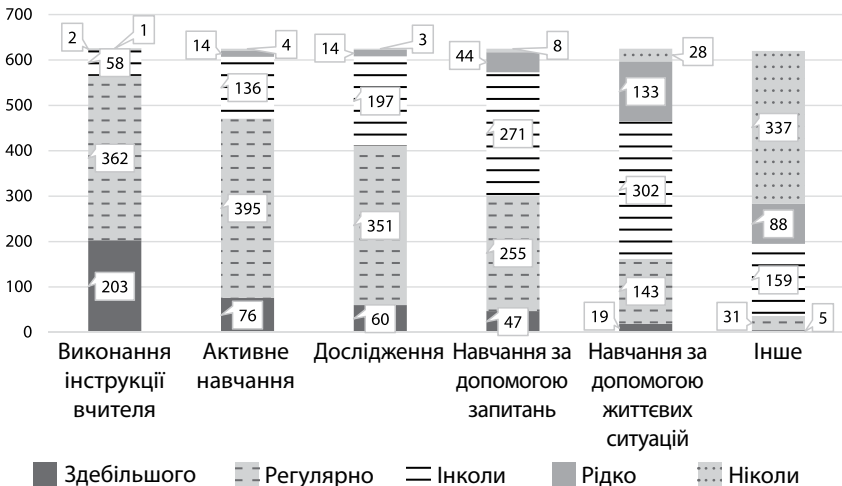


Рис. 2. Поєднання навчання та практики в молодшій школі Квінсленду (за кількістю шкіл)

В середній та старшій школі учні працюють в невеликих групах (рис. 3).

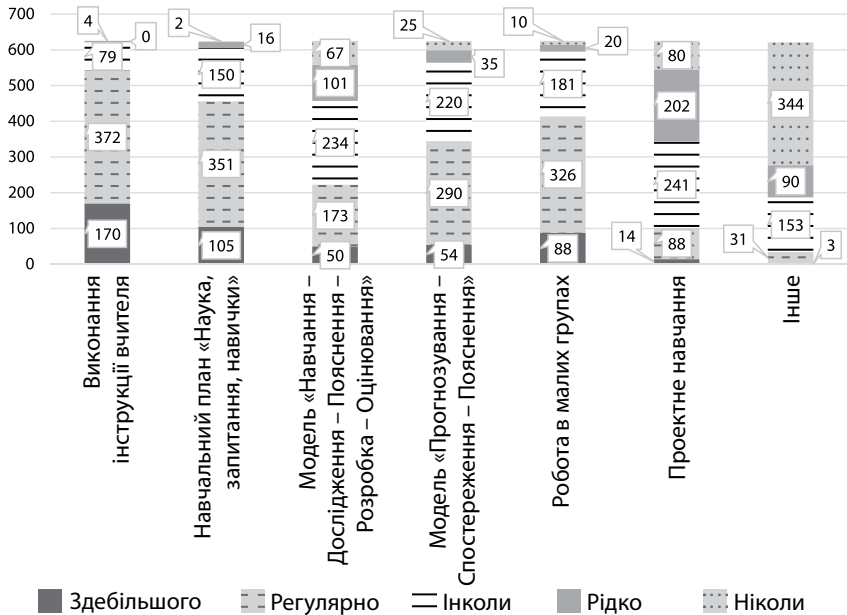


Рис. 3. Робота в малих групах в середній та старшій школі Квінсленду (за кількістю шкіл)

STEM- та STEAM-освіта забезпечує мотивацію до навчання, отримання знань та вмінь на практиці при поєднанні основ науки, технологій, математики, інженерії та мистецтва. До принципів STREAM-освіти належать:

1. Практична інтеграція науки і грамотності: модель, заснована на дослідженнях з використанням інструкцій, що поєднують практичні заняття, аналіз, синтез, лексику, обговорення і організацію даних.
2. Забезпечення можливості для навчання через наукові дослідження, читання, письмо та використання математичних навичок в науковому контексті.

У багатьох країнах світу виділяються засоби для соціальної реклами STEM, підготовки вчителів, організується співпраця між системою освіти та компаніями, що працюють в STEM, учнів долучають до наукової діяльності та розробок, створюються навчальні центри STEM.

Оскільки акронім STEM охоплює чотири предметні галузі (наука, технологія, інженерія та математика) то STEM-освіта передбачає вивчення саме їх.

STEM-освіта може бути представлена як [19]:

- підвищення рівня освіти в рамках кожного окремого предмету, що входить до галузі STEM;
- інтерактивний підхід до освіти, в якому основні концепції науки, технології, інженерії та математики об'єднуються в рамках однієї навчальної програми, яка називається STEM. Важливість подібного інтегративного підходу обумовлена інтегрованим характером сучасних технологій і потребами ринку праці в фахівцях «широкого профілю».

Marlene Hurley [16] запропонувала п'ять рівнів інтеграції:

- послідовне вивчення предметів;
- паралельне вивчення предметів;
- часткова інтеграція;
- розширена інтеграція;
- повна інтеграція.

При всіх рівнях інтеграції STEM-освіта передбачає системний підхід, вона відрізняється від традиційного навчання тим, що має змішане середовище навчання та наповнена демонстрацією того, що наукові методи працюють у повсякденному житті. Перспективним напрямком міжпредметних зав'язків є робототехніка. Перевагами цього напрямку є:

- широкий вибір технологій, які використовуються для створення роботів відповідно до предметів і тем галузі STEM, які вивчаються учнями у проектній діяльності;
- актуальність та привабливість теми для молоді;
- можливість виконання проектів та використання обладнання на різних рівнях.

Робототехніка:

- робить навчання цікавим, привабливим і вмотивованим;
- дає досвід практичної роботи руками;
- заохочує до підготовки до вступу у вищі навчальні заклади;

- розвиває навички критичного мислення і стратегії вирішення проблем;
- дозволяє учням розвивати і виявляти свої творчі здібності;
- розвиває здатності до роботи в команді;
- допомагає зрозуміти на інтуїтивному рівні наукові та математичні концепції;
- сприяє успіхам учнів в науках та математиці, вибору кар'єри в напрямку STEM;
- допомагає учням краще розуміти і реалізовувати технологічні рішення;
- підвищує впевненість у власних силах і самооцінку;
- готує до життя в швидкоплинному конкурентному середовищі.

Робототехніка та автоматизація стають життєвоважливи-ми для розвитку глобальної економіки. Згідно зі звітом 2014 від Allied Market Research, світовий ринок промислової робототехніки (оцінюється в \$ 26780 млн на 2012 р.), як очікується, зростатиме зі швидкістю 5,4 відсотка на рік до \$ 41170 млн до 2020 р. Такий підйом був викликаний швидким зростанням вимог автоматизації в поєднанні зі зниженням мит на реконструйовані товари в пан-азіатському регіоні [4].

В результаті, різні зусилля були зроблені на початковому (діяльність з інформування), середньому (дослідження, кар'єра і інженерна діяльність та програми) та у вищому рівнях освіти (кар'єра і технічні, інженерні технології та інженерні освітні програми) для впровадження робототехніки [21].

Цілі навчальної робототехніки:

- ранній розвиток (дрібна моторика, просторове мислення, конструювання тощо);
- загальний розвиток життєвих компетентностей (командна робота, реалізація проектів, саморефлексія тощо);
- загальнотехнічні навички (прості механізми, міцність тощо);
- спеціальні технічні навички (алгоритмізація, електроніка тощо).

ВПРОВАДЖЕННЯ РОБОТОТЕХНІКИ У ПОЧАТКОВІЙ ТА СЕРЕДНІЙ ОСВІТІ

Petre та Price [28] запропонували для вчителів початкової та середньої школи вивчення робототехніки на рівні навчання програмуванню та основним принципам інженерії. Дослідники стверджують, що мотивація учнів значно зростала під час проектної роботи зі створення власних роботів.

Petre та Price також зазначали, що учні могли б дізнатися про складні для сприйняття предмети (математика, наука, технології) через технологію «via the back door» (через задні двері) під час вивчення основ робототехніки та колективної роботи на позакласних та позашкільних заходах, змаганнях тощо.

Van Lith [33] погоджується з Petre та Price, додаючи, що програми вивчення робототехніки в початковій та середній школі підвищують інтерес до вивчення роботизованих систем, двигунів, датчиків, сенсорів, завдань з їх управління, усунення несправностей та програмування. В США були використані шість тем для вивчення робототехніки (дві власне з робототехніки та чотири міждисциплінарні) в початковій і середній системі освіти:

- механіка і програмування (основи конструювання та програмування роботів);
- робота в команді, вирішення проблем, конструювання робота для самоідентифікації з використанням технологій, які забезпечують додаткові можливості навчання в контексті робототехніки [25].

Також активно використовувались позашкільні заходи різних типів – конкурси, змагання з конструювання та програмування роботів. В Україні представлені два найбільших світових конкурси з науки та робототехніки – FIRST LEGO League (для дітей 9-16 років) та FIRST LEGO League Junior (для дітей 6-10 років), на яких учні можуть виступити командами та продемонструвати свої здібності.

Навчальна робототехніка може використовуватися у різному віці, існує велика кількість різних наборів. Загалом можна виділити 3 основні категорії:



Рис. 4. Типи робототехніки

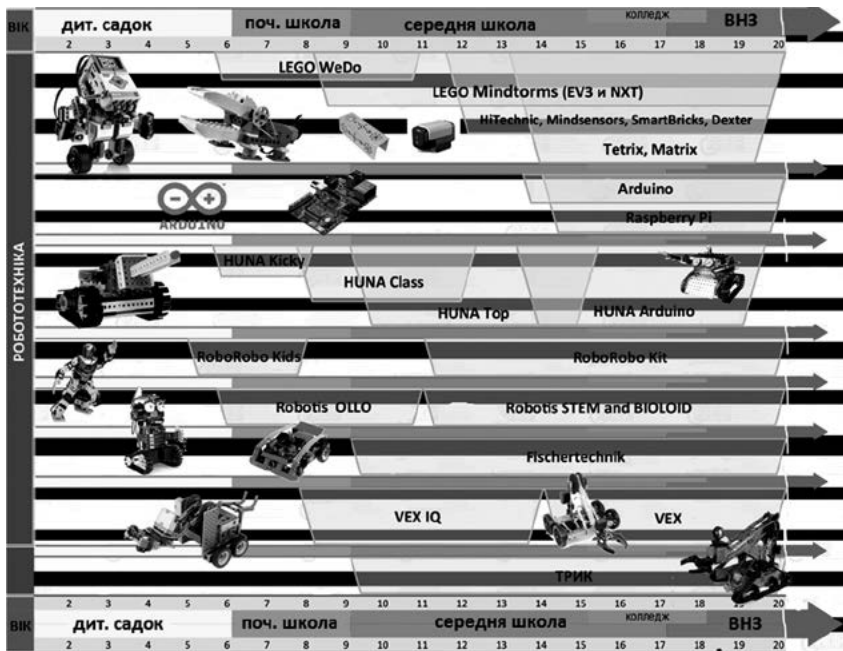








Рис. 5. Розподіл конструкторів за віковими категоріями

Набори роботів, які можуть бути використані в школі:

| № | Назва | Посилання | Фото |
|----|---------------------------------------|---|---|
| 1. | Lego Mindstorms Базовий набір 45544 | http://constructors.com.ua/MINDSTORMS/Bazovyy-nabor-LEGO-MINDSTORMS-Education-EV3-45544 |  A photograph showing the components of the Lego Mindstorms Education EV3 45544 kit. It includes a central processing unit (EV3), several sensors, a motor, and various gears and connectors, all laid out on a white surface next to a tablet displaying the kit's packaging. |
| 2. | Lego Mindstorms Ресурсний набір 45560 | http://constructors.com.ua/Education/Resursnyy-nabor-LEGO-MINDSTORMS-Education-EV3-45560 |  A photograph showing the components of the Lego Mindstorms Education EV3 45560 kit. It includes a central processing unit (EV3), various sensors, a motor, and a large assortment of gears and connectors, all laid out on a white surface next to a tablet displaying the kit's packaging. |
| 3. | LittleBits STEAM Student Set | http://hobbytech.com.ua/shop/littlebits/%D0%BE%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B8%D0%B9-%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80-littlebits-steam-student-set/ |  A photograph showing the components of the LittleBits STEAM Student Set. It includes a variety of colorful modules for sensors, motors, and logic, along with a breadboard and jumper wires, all laid out on a white surface next to the product's packaging. |

| | | | |
|----|----------------------|---|---|
| 4. | LittleBits Synth Kit | http://hobbytech.com.ua/shop/littlebits/%D0%BE%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B8%D0%B9-%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80-littlebits-synth-kit/ |  |
| 5. | ROBOTIS STEM LEVEL 1 | http://hobbytech.com.ua/shop/robotis/%D0%BE%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B8%D0%B9-%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80-robotis-stem-level-1/ |  |
| 6. | ROBOTIS STEM LEVEL 2 | http://hobbytech.com.ua/shop/robotis/%D0%BE%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B8%D0%B9-%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80-robotis-stem-level-2/ |  |

| | | | |
|-----|---------------------|---|---|
| 7. | mBot Ranger | http://hobbytech.com.ua/shop/makeblock/mbot-ranger/ |  |
| 8. | Huna Top full kit 2 | http://www.robot96.ru/catalog/roboty-igrushki/drugie-konstruktory/huna-top-full-kit-2/ |  |
| 9. | TETRIX | http://nanojam.ru/products/tetrix |  |
| 10. | Ozobot | http://ozobot.com/ |  |



2 ОСНОВНІ ПОНЯТТЯ РОБОТОТЕХНІКИ

2.1. ІСТОРІЯ РОЗВИТКУ РОБОТОТЕХНІКИ

Еволюція сучасного суспільства й виробництва обумовила виникнення й розвиток нового класу машин-роботів і відповідного наукового напрямку робототехніки.

За всіх часів наукова й виробнича діяльність людини визначалася, з одного боку, потребою розвитку й удосконалення суспільного виробництва як бази для досягнення економічної могутності, з іншого – необхідністю дослідження й освоєння нових нетрадиційних просторів і галезуй діяльності.

Історія робототехніки сягає глибокої давнини. Уже в ті часи з'явилися ідеї створення технічних засобів, схожих на людину, були зроблені перші спроби з їх створення. Статуї богів з рухомими частинами тіла (руки, голова) з'явилися ще в Стародавньому Єгипті, Вавилоні, Китаї. У III столітті до н. е. римський поет Клавдій згадував про автомат, створений Архімедом. Він мав форму скляної кулі із зображенням небосхилу, на якому відтворювався рух всіх відомих на той час небесних світил. Рух кулі відбувався під впливом води [35].

Грецький винахідник і фізик Ктесібій з Олександрії сконструював водяний годинник. Це був перший автоматичний пристрій для точного хронометрування.

До нас дійшли книги Герона Олександрійського (I століття н.е.), де описано інші автоматичні пристрої, створені у давнину. Як джерело енергії в них використовувалися вода, пара, гравітація (гіри).

У «Театрі автоматів» описано пристрій цілого театру, виставу в якому розігрували фігурки-ляльки, що починали рух за допомогою автоматизованої системи.

За часів Середньовіччя великою популярністю користувалися різного роду автомати, основою яких були годинникові меха-

нізми. Було створено годинники з рухомими фігурами людей, ангелів тощо. До цього періоду належить інформація про створення перших рухомих людиноподібних механічних фігур – андроїдів. Так, андроїд алхіміка Альберта Великого був лялькою в зріст людини, яка, коли стукали в двері, відкривала їх з поклоном.

У XIII столітті Альберт Великий створив автомат, який став згодом відомим як «голова, що розмовляє», здатна відтворювати людський голос.

У 1495 р. Леонардо да Вінчі розробив детальний проект механічної людини, здатної рухати руками і повертати головою. А у 1500 р. він побудував механічного лева, який при в'їзді короля Франції в Мілан висувався, роздирав пазурами груди і демонстрував герб Франції.

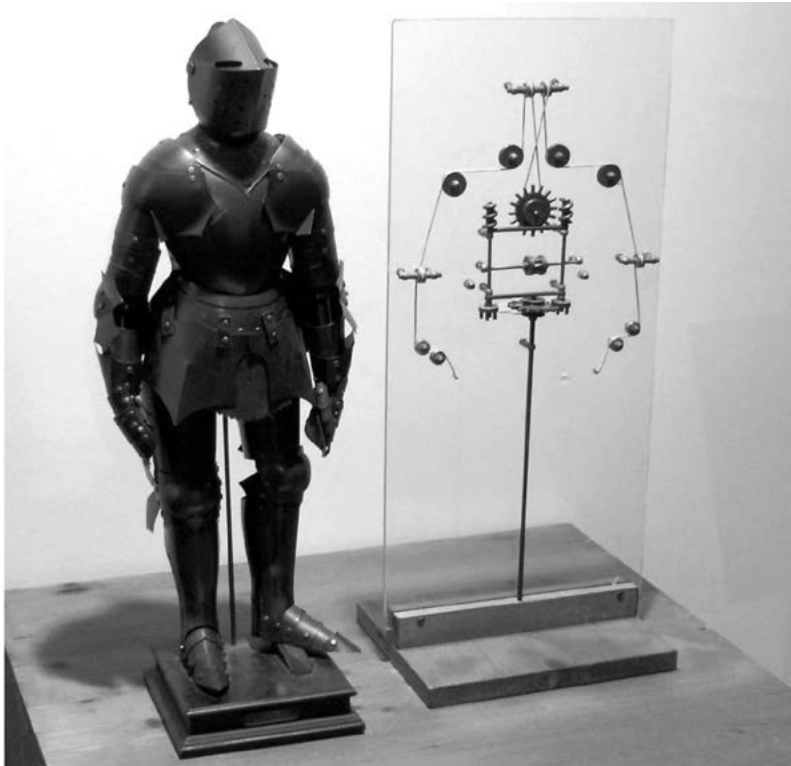


Рис. 6. Механічна людина Леонардо да Вінчі

Роботи зі створення андроїдів досягли найбільшого розвитку в XVIII ст. одночасно з розквітом годинникової майстерності. Французький механік і винахідник Жак де Вокансон створив у 1738 р. перший працюючий людиноподібний пристрій (андроїд), який грав на флейті [37].



Рис. 7. «Флейтисит» Жака де Вокансона

Рухливими пальцями він міг виконувати 11 мелодій за допомогою закладеної в нього програми. Вокансон також створив механічну качку, вкриту справжніми пір'ям, яка могла ходити, рухати крилами, крякати, пити воду, клювати зерно і перемелювати його маленьким внутрішнім млином. Качка складалася з понад 400 рухомих деталей і була однозначно визнана вінцем творіння майстра.



Рис. 8. *Механічна качка
Жака де Вокансона*

Створенням автоматів також займалися швейцарські годинникарі П'єр-Жак Дро (1721-1790 рр.) та його син Анрі Дро (1752-1791 рр.). Від імені останнього пізніше було утворено і поняття «андроїд» [47]. П'єр-Жак Дро створив кілька автоматів, з яких найбільшу популярність здобули писар і художник. Писар був дівчинкою, що сиділа за столом та писала літери, слова, могла намалювати собаку. При цьому вона плавно погойдувала головою і опускала повіки в такт руху руки. Разом з сином вони створили дівчину, що грає на клавесині. Збереглась модель цієї фігури: «Дівчина грає, ворухить губами, груди її піднімаються і опускаються під час «дихання», вона дивиться на клавіші, в ноти, а іноді кидає погляд на публіку, після завершення «номеру» встає і кланяється». Ці людиноподібні іграшки являли собою багатопрограмні автомати з хорошою швидкодією.



Рис. 9. Роботи П'єра-Жака Дро та Анрі Дро

Механік І. Кулібін (1735-1818 рр.) побудував упродовж трьох років фігуру у формі яйця – універсальний годинник.



Рис. 10. Годинник
І. Кулібіна

Всередині годинника відбувалась театралізована вистава і грала музика. Він мав три самостійні механізми і три заводи: вартовий, бойовий та курантовий, а також автоматичні прилади для приведення в дію механізмів, що відтворюють сцени, музику і бій.

Разом з безпосереднім створенням різних автоматичних пристроїв, які виконували функції живих істот, в середні століття були закладені основи різних наукових напрямків.

Леонардо да Вінчі робив спроби встановити відповідність між механізмами і окремими органами людини. А знаменитий французький філософ і математик Рене Декарт говорив, що тіла тварин є нічим іншим, як складнішими машинами.

У XVI-XVII ст. виник новий науковий напрямок на стику фізіології і механіки – ятромеханіка (від грец. *iатros* – лікар). Його видатним представником був Джованні Альфонсо Бореллі (1608-1679 рр.), лікар і механік, професор Мессинського університету. В його праці «Про рух тварин» розглядається робота м'язів серця, кровообіг інших органів тварин і людини на основі механічних аналогій. По суті, ятромеханіка заклала основи сучасних наукових напрямків – біомеханіки і біоніки [46].

На межі XVIII і XIX ст. в працях Лазара Карно, Гаспара Монжа, Хосе Марія Ланца і Августина Бетанкура виникає наука про машини. У 1841 р. Р. Вілліс визначив поняття механізму, і з цього часу до машини починають підходити як до об'єкта, який вимагає наукового дослідження.

Г. Монж став ініціатором викладання курсу «Побудова машин» і сформував основи класифікації механізмів. Л. Карно у 1783 р. опублікував книгу «Досвід про машини взагалі», яка через 10 років була перевидана під назвою «Основні принципи рівноваги і руху». У цій роботі Карно зазначив, що механіка за своєю сутністю є наукою експериментальною, тим самим підтвердив її право на самостійне існування [37].

Російський математик і академік П. Чебишев (1821-1894 рр.) поклав початок новому етапу в дослідженні машин і механізмів. Він пов'язав питання структури і синтезу механізмів в єдине вчення про побудову механізмів на основі математичних методів. У своїй роботі «Теорія механізмів, відомих під назвою паралелограмів» П. Чебишев описав задачі теорії механізмів мовою математики [45].

Промислова революція другої половини XVIII століття, пов'язана з переходом від ручного виробництва до машинного, змушує винахідників створювати нові машини і пристрої. Саме в цей час почали закладатися основи **промислової автоматки**, особливо в текстильній промисловості. Ще у 1725 р. Безіл Бушоні придумав перфоровану паперову стрічку для запису програми, яку в подальшому використовував для програмування ткацьких верстатів для виробництва шовкової тканини з малюнком. В 1728 р. Жан-Батист Фалькон удосконалив винахід Безіла Бушоні. Він замінив перфоровану паперову стрічку картками, з'єднаними в ланцюжок, що дозволило

легко замінювати окремі фрагменти програми. Надалі ці верстати були вдосконалені Вокансоном і Жозефом Марі Жаккардом, у 1805 р. Жаккард створив автоматичний верстат, на якому можна виготовляти тканини із заздалегідь запрограмованим малюнком за допомогою перфокарт. Цей винахід став однією з найважливіших подій, які визначили подальший технічний прогрес промисловості і стали поштовхом розвитку робототехніки.

Ще однією важливою подією в галузі робототехніки стало створення першої обчислювальної машини. На основі способу програмування Жаккарда англійський механік Чарльз Беббідж розробив лічильну «Аналітичну машину», структурні особливості якої на ціле століття визначили напрямок розвитку обчислювальної техніки.

З XIX століття завдяки розвитку електротехніки та електроніки реалізуються потреби суспільства і виробництва в різних автоматичних пристроях. Література і мистецтво в цей час виконували роль каталізатора процесу розвитку робототехніки. Саме в цей період з'являється багато науково-фантастичних творів літератури, в яких роботи-андроїди відігравали головні ролі.

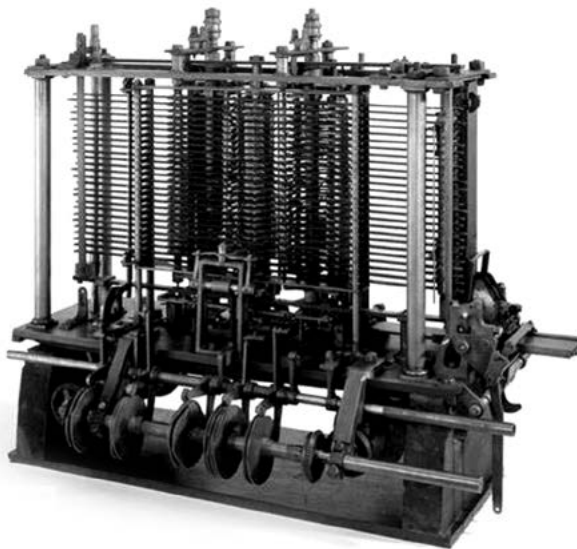


Рис. 11. *Машина Чарльза Беббіджа*

Вперше слово «робот» було введено Карелом Чапеком у 1920 р. у фантастичній п'єсі «РУР» («Рассумські універсальні роботи») [32]. Галузю застосування роботів стали галузі діяльності людини, які є небезпечними для її життєдіяльності. Як правило, це були дистанційно керовані маніпулятори для роботи в атомних реакторах, підводних апаратах і космічних кораблях тощо. У 1947 р. в Арагонській національній лабораторії було вперше розроблено механічні руки для роботи з радіоактивними матеріалами. Уже у 1948 р. такі роботи були оснащені системою відображення сили, щоб оператор мав можливість відчувати силу, що розвивається виконавчим механізмом. Перші місяцеходи та марсоходи були оснащені маніпуляторами для збирання ґрунту. Управління даними маніпуляторами здійснювалося з Землі за командами оператора. У 1963 р. вже була досліджена проблема розпізнавання багатогранних об'єктів, а у 1968 р. створено програмні пристрої, що дозволяють із застосуванням телевізійної камери знаходити предмети, які має брати робот своїм пристроєм для захоплення.

Таким чином, теоретичні основи сучасної робототехніки було закладено ще в 60-і роки, але їх реалізація стримувалася відсутністю відповідних технологій, матеріалів, ресурсів обчислювальних систем. В цей же час письменник-фантаст Айзек Азімов придумує слово «Роботікс» (робототехніка) і вперше формулює три закони робототехніки [34]:

1. Робот не може заподіяти шкоду людині або своєю бездіяльністю допустити, щоб людині було завдано шкоди.
2. Робот повинен виконувати команди людини, якщо ці команди не суперечать першому закону.
3. Робот має бути безпечним так, щоб це не суперечило першому і другому законам.

Ці три закони Айзека Азімова досьгодні залишаються стандартами при проектуванні і розробці роботів.

Завдяки загальному інтересу до роботів винахідникам вдалось розробити оригінальні конструкції роботів-андроїдів.

«Містер Телевокс» (1928 р., американський інженер Дж. Уенслі) – робот, який мав зовнішню схожість з людиною, здатний виконувати елементарні рухи за командами, які подаються голосом, став експонатом Всесвітньої виставки у Нью-Йорку [41]. «Ерік»

(1928 р.) – робот, який на виставці Британської асоціації інженерів з моделювання «виступив» з невеликою промовою [9].

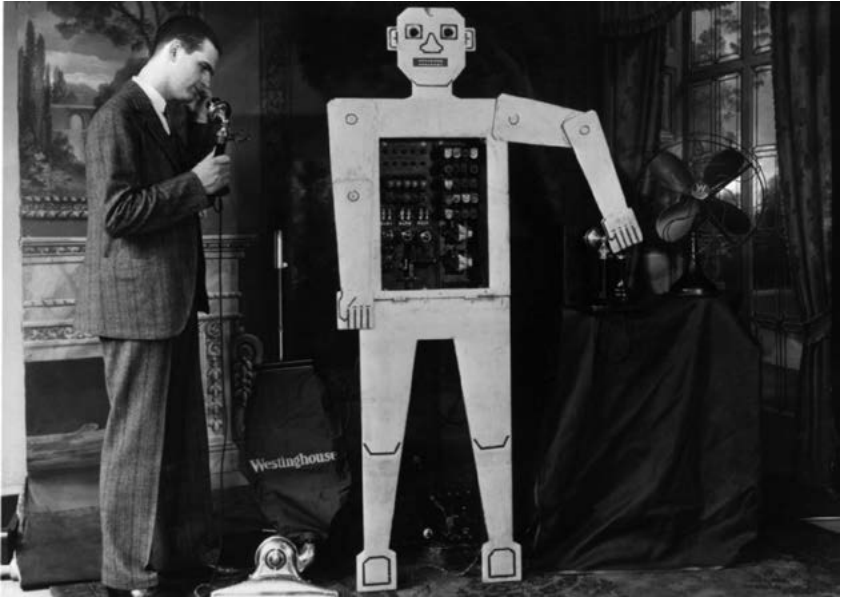


Рис. 12. Містер Телевокс Дж. Уенслі



Рис. 13. Робот «Ерік»

«Натураліст» (1928 р., під керівництвом доктора Нісімура Макота) – японський робот, здатний за допомогою електроприводу маніпулювати руками і головою. Згодом цей андроїд стали вважати початком роботобудування в Японії.

Незважаючи на такий прорив у галузі нової техніки і демонстрації творчих можливостей людини, всі ці роботи мали вкрай вузьке практичне застосування.

Виникнення сучасних роботів слід віднести до 1959 р. У цьому році в США були створені перші промислові маніпулятори з програмним управлінням, які отримали загальноприйнятну назву промислових роботів (ПР) і поклали початок комерційного виробництва.

У 50-х рр. ХХ століття група американських інженерів розпочала роботу над проблемою застосування теорії управління у вирішенні спільних завдань оптимального переміщення обладнання. Піонерами стали Джордж К. Девол (1912-2011 рр.) і Джозеф Ф. Енгельбергер (1925 р.). У 1954 р. Девол запатентував в США спосіб переміщення предметів між різними ділянками підприємства на основі керуючої програми на перфокартах, подібний до запропонованого колись Беббіджем. Винахід був покликаний вирішити, в першу чергу, саме проблему гнучкості, тобто створення універсального транспортувального пристрою, який легко перебудовується для виконання інших операцій [40].

У 1956 р. Девол разом з Енгельбергером організували першу в світі робототехнічну компанію «Unimation», що означає «універсальна автоматизація» – скорочене від «Universal Automation». В лабораторії цієї компанії і був створений перший в світі промисловий робот за патентом Девола, що носив «програмований пристрій для передачі предметів» і став прототипом наступних розробок [3].

На початку 1960-х рр. перші американські промислові роботи з торговельними марками «Unimate» і «Versatran», створені відповідно фірмами «Unimation», «American Machine and Foundry» (AMF) і призначені для обслуговування технологічних процесів, надійшли на промисловий ринок. Вони являли собою вже досить досконалі системи зі зворотним зв'язком і контрольованою траєкторією руху, мали числове програмне управління і пам'ять, як у ЕОМ. Вже в перших роботах «Unimate» і «Versatran» був реалізова-

ний принцип програмованого навчання. Застосування роботів в автомобільній і металургійній промисловості було економічно вигідним: витрати на придбання роботів «Unimate» або «Versatran» окупалися за 1,5 - 2,5 роки.

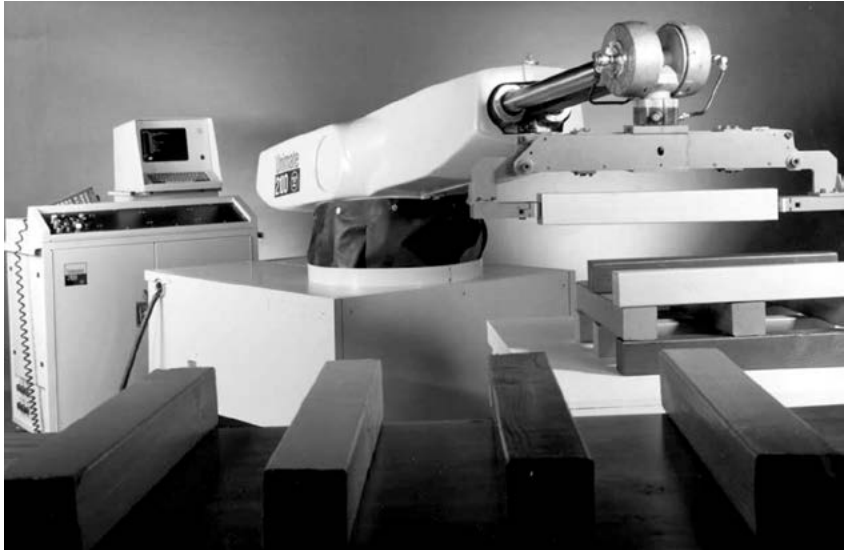


Рис. 14. Програмований пристрій для передачі предметів

Перші комерційні успіхи застосування промислових роботів стали потужним імпульсом для їх подальшого вдосконалення. На початку 1970-х рр. з'являються роботи, керовані комп'ютерами. Перший міні-комп'ютер, керуючий роботом, було випущено в 1974 р. фірмою «Cincinnati Milacron», однією з провідних фірм-виробників роботів у США. Наприкінці 1971 р. американською фірмою «INTEL» було створено перший мікропроцесор, а кількома роками пізніше з'являються роботи з мікропроцесорним управлінням, що зумовило істотне підвищення їх якості при одночасному зниженні вартості [40].

У наступні роки після створення і виходу на промисловий ринок перших роботів у всьому світі почався стрімкий розвиток робототехніки.

Технічний прогрес у розвитку роботів було спрямовано, перш за все, на вдосконалення систем управління. *Промислові*

роботи першого покоління мали програмне управління, в основному запозичене у верстатів. Друге покоління характеризується додаванням чутливості роботам (забезпечено сенсорними системами, головними з яких є системи технічного зору).

Перші промислові роботи з розвинутою сенсорною системою і мікропроцесорним управлінням з'явилися на ринку і отримали практичне застосування у 1980-1981 рр. перш за все на збірці, дугового зварювання, контролю якості для взяття неорієнтованих предметів, наприклад з конвеєра.

Третє покоління – інтелектуальні. Інтелектуальний робот – це робот конкретного призначення, в основних функціональних системах якого використовуються методи штучного інтелекту. Виникнення інтелекту у роботів пов'язане з розвитком ЕОМ. У 1967 р. в США (Стенфордський університет) було створено лабораторний макет робота, оснащеного технічним зором, призначеного для дослідження і відпрацювання системи «око – рука», здатної розпізнавати об'єкти зовнішнього середовища і оперувати ними відповідно до завдання [40].

У 1972-1975 рр. в Київському Інституті кібернетики під керівництвом М. М. Амосова та В. М. Глушкова було створено макет транспортного автономного інтегрального робота (ТАІР). Робот демонстрував цілеспрямований рух в природному середовищі, обхід перешкод. Конструктивно ТАІР являв собою триколісний самохідний візок, забезпечений системою датчиків: оптичним далекоміром, навігаційною системою з двома радіомаяками і компасом, контактними датчиками, датчиками кутів нахилу візка, таймером тощо.

Особливістю, яка відрізняє ТАІР від багатьох інших систем, створених в той час, є відсутність в його складі комп'ютера в тому вигляді, до якого ми звикли. Основу системи управління становить апаратно реалізована нейронна мережа (вузли мережі – спеціальні електронні схеми, зібрані на транзисторах, зв'язки між вузлами – резистори), у якій реалізуються різні алгоритми обробки сенсорної інформації, планування поведінки та управління рухом робота.

Одночасно розгорнулися роботи в новій специфічній галузі робототехніки – крокуючі машини як принципово новий транспортний засіб підвищеної прохідності, зразком для якого є ноги

тварин і людини. Були створені експериментальні зразки чотири- і шестиногих транспортних машин, протезів ніг людини, так званих екзоскелетонів, для паралізованих і тяжкохворих.

Робототехніка як наукова дисципліна формується спільними зусиллями вчених і розробників техніки в цілісний науково-технічний напрям, збагачується величезним досвідом розробки та експлуатації найрізноманітніших роботів, робототехнічних пристроїв і систем.

2.2. ПОНЯТТЯ «РОБОТА» ТА ЙОГО СКЛАДОВІ

Для визначення основного поняття «робот» слід усвідомити головні критерії оцінки його можливостей, оскільки робот за своєю концепцією виник як пристрій, покликаний замінити працю людини в найрізноманітніших видах і галузях.

У широкому розумінні поняття «робот» можна визначити як технічну систему, здатну замінювати людину або допомагати їй у виконанні різних завдань. Однак і до цього часу відсутнє загальноприйняте означення. Різні джерела дають різне тлумачення цього поняття. Велика радянська енциклопедія визначає робота як «машину з антропоморфною (людиноподібною) поведінкою, яка частково або повністю виконує функції людини (іноді тварини) при взаємодії з навколишнім світом» [44], а енциклопедія кібернетики – як «систему, оснащену датчиками, що сприймають інформацію про навколишнє середовище, виконавчими механізмами, які впливають на об'єкти навколишнього середовища, здатну цілеспрямовано поводити себе у ситуаціях, що змінюються. Від інших систем, призначених для опрацювання інформації, яка надходить ззовні, та отримання керуючих впливів (наприклад, систем автоматичного управління технологічними процесами), роботів відрізняє антропоморфізм...» [39].

Професор Токійського університету, доктор Сігеру Ватаата пропонує вважати роботом пристрій, здатний самостійно переміщуватися в просторі, справлятися із завданнями аналізу сцен і розпізнавання образів, що володіє кількома ступенями рухливос-

ті, вміє аналізувати зовнішнє середовище за допомогою зворотного зв'язку, а також прогнозувати ситуації, спираючись на власний досвід і доступну інформацію.

Професор Токійського технологічного інституту, доктор Сейко Морі вважає, що роботом можна назвати пристрій за умови, що він має універсальність, мобільність, є єдиним фізичним об'єктом, працює автоматично, повністю підпорядковується людині, а також здатний до елементарної інтелектуальної діяльності. При цьому універсальність, мобільність, індивідуальність і автоматизм – всі ці характеристики є невід'ємними ознаками всіх роботів без винятку [15].

Професори Інституту обладнання і технологій (Лондон, Великобританія) М.О. Тохі та А.К.М. Азад конкретизують поняття робота. Пристрій, щоб називатися роботом, має мінімальний набір властивостей і можливостей: 1) наявність механічної руки для захоплення; 2) вміння самостійно пересуватися і самостійно керувати своїми діями; 3) наявність виконавчої системи та системи управління, які в сукупності забезпечують реалізацію попередніх властивостей; 4) наявність комп'ютера, здатного запам'ятовувати програми управління для відпрацювання команд, що надходять ззовні, а також програм, які реалізують рішення, що «приймає» робот, виходячи з набору деяких заздалегідь визначених альтернатив; 5) наявність пристроїв і датчиків, здатних визначати торкання роботом будь-яких зовнішніх предметів, вимірювати ступінь гладкості поверхні, твердість матеріалу, розташування заданого предмета, його масу, теплопровідність, температуру, ступінь близькості цього предмета, визначати його форму і розміри, характерні зовнішні ознаки, колір, відстань, запах, знаходити місце розташування «рук і ніг» робота, а також аналізувати звукові сигнали [10].

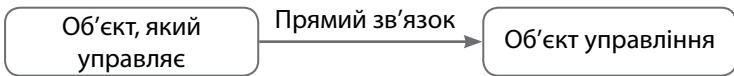
Вікіпедія [Робот. Режим доступу: <https://uk.wikipedia.org/wiki/Робот>] визначає «робота» як автоматичний пристрій, що призначено для виконання виробничих та інших операцій, які зазвичай виконувались безпосередньо людиною. Для опису автоматичних пристроїв, дія яких не має зовнішньої схожості з діями людини, переважно використовується термін «автомат».

Таким чином можемо дати досить загальне і стисле визначення класу машин, що ми називаємо роботами.

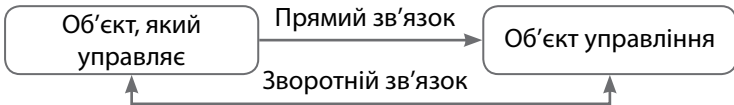
Робот – це автономно функціонуюча універсальна автоматична машина, призначена для відтворення фізичних, рухових і розумових функцій людини, наділена здатністю до адаптації і навчання у процесі активної взаємодії з навколишнім середовищем.

Розглянемо принципи управління роботом:

1. Управління з прямим зв'язком. Об'єкт, який управляє роботом, передає команду роботу, яким управляють. В такому випадку невідомо, виконана команда чи ні.



2. Управління зі зворотнім зв'язком. Об'єкт, який управляє, передає команди об'єкту управління, а той передає по зворотному зв'язку дані про свій стан.



Базові поняття, на яких базується робот:

- **Механізм** – пристрій, за допомогою якого людина може виконувати роботу; внутрішній пристрій машини, що забезпечує її роботу.

Але самостійно механізм зробити нічого не зможе. Для того, щоб механізмом виконувалися команди, необхідна програма.

- **Автомат** – пристрій, який виконує роботу самостійно відповідно до вбудованої програми.
- **Робот** – автомат зі зворотнім зв'язком та елементами штучного інтелекту.

Робот, який активно взаємодіє з навколишнім середовищем, у загальному вигляді повинен мати певні системи: **управляючу, інформаційно-вимірювальну (сенсорну), систему зв'язку, виконавчу (моторну).**

Управляюча або інтелектуальна система – це «мозок» робота, який призначено для управління механізмами виконавчої

системи на основі вбудованої програми з урахуванням сигналів зворотного зв'язку від сенсорної системи.

Інформаційно-вимірювальна або сенсорна система – це штучні «органи» чуття робота, призначені для сприйняття і перетворення інформаційних даних про стан зовнішнього середовища і самого робота.

Система зв'язку – це «мова» робота, призначена для передачі інформаційних сигналів між системами робота, а також для організації обміну інформацією між роботом і людиною або іншими роботами і пристроями з метою здійснення діалогу, надання команд роботу, контролю за функціонуванням його систем, діагностики несправностей, регламентної перевірки тощо.

Виконавча або моторна система – це пристрій, призначений для безпосереднього впливу на об'єкти навколишнього середовища або взаємодії з ними відповідно до сигналів, що формулюються інформаційно-вимірювальною системою або безпосередньо оператором.

Структурна схема робота представлена у вигляді схеми (рис.15).

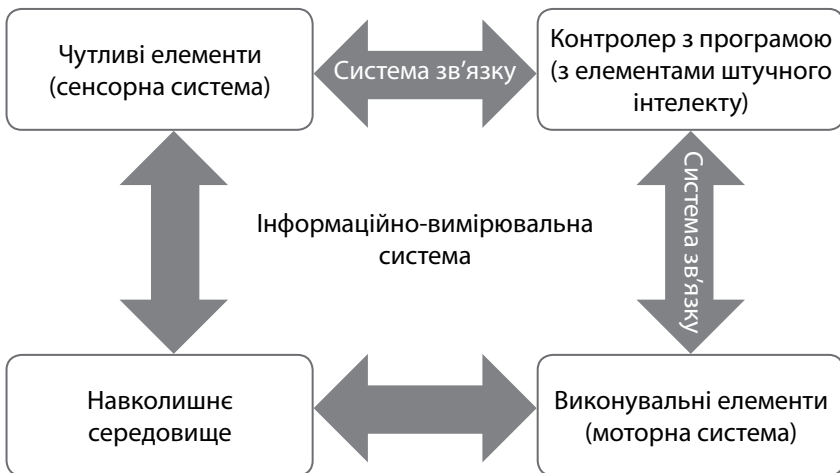


Рис.15. Структурна схема робота

2.3. КЛАСИФІКАЦІЯ РОБОТІВ ТА ГАЛУЗІ ЇХ ЗАСТОСУВАННЯ

Роботи можуть класифікуватися за різними ознаками. Найбільш загальними і змістовними є класифікації за призначенням та класом задач, для розв'язування яких призначений робот; особливостями управління; технологічними ознаками.

Розвиток робототехніки зумовлює формування нових поколінь роботів і впровадження їх в нові галузі. Тому класифікація за призначенням та класом задач, до яких призначено роботів, не є завершеною і певною мірою характеризує шлях розвитку роботів, а тому може бути названа також «еволюційною», але вже зараз можна зазначити, що роботи всіх поколінь можуть бути розділені на дві великі групи: **виробничі** і **дослідницькі**.

Виробничі роботи призначені для виконання важкої, монотонної, шкідливої та небезпечної для здоров'я людей фізичної роботи, а також окремих видів трудомістких, напружених розумових робіт (проектування, інформаційне забезпечення, управління).

Промислові роботи призначені для автоматизації основних і допоміжних операцій в різних галузях промисловості – машинобудуванні і приладобудуванні, гірничодобувній, нафтохімічній, металургійній, атомній тощо.

За методом управління або ступенем безпосередньої участі людини в управлінні роботи поділяють на три групи: **біотехнічні**, **інтерактивні** та **автоматичні**.

Біотехнічні роботи функціонують тільки за безпосередньої участі людини-оператора, яка фактично бере на себе управління виконавчими механізмами.

Інтерактивні роботи на відміну від біотехнічних мають пристрої пам'яті для автоматичного виконання окремих дій і можуть управлятися поперемінно оператором або автоматично.

Автономні роботи можуть функціонувати цілком самостійно без безпосередньої участі в їх управлінні людини-оператора.

За способом переміщення роботів можна поділити на такі види (рис. 16):

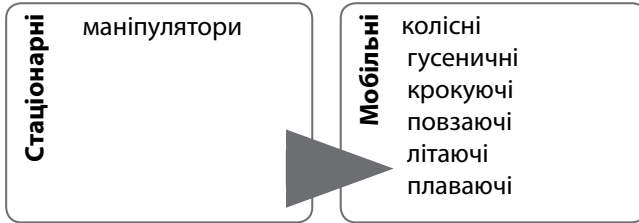


Рис.16. Види роботів за способом переміщення

В сучасному світі роботи користуються великим попитом. Їх використовують у різних сферах життя, про які багато хто може навіть не здогадуватися (рис.17).



Рис.17. Сфери застосування роботів

Використання роботів у медицині. Роботи допомагають рятувати людські долі, а іноді і життя. Сучасні протези кінцівок безпосередньо пов'язані з робототехнікою. Нерухомі штучні руки залишилися в далекому минулому, нинішні протези вміють рухати пальчиками. Їх управління безпосередньо пов'язане з електричними імпульсами, що передаються тілом. Також роботи використовуються для проведення високотехнологічних медичних операцій.

Відомим у медицині є робот-хірург **Da Vinci**, що активно використовується в усьому світі. Поки це не самостійно діючий механізм, а слухняний 500-кілограмовий інструмент в руках лікаря. В операційного модуля чотири «руки». Три з них закінчуються мініатюрними хірургічними інструментами – скальпелем і зажимами, а четверта управляє крихітної відеокамерою. Da Vinci оперує через сантиметрові проколи, тому без камери не обійтися, зате у пацієнта майже не залишається шрамів. Коли робот «чаклує» над хворим, хірург-людина сидить за пультом. Лікар маніпулює джойстиком, які дуже точно передають руху пальців і кисті «рук» da Vinci. Як і у людської кисті, у них сім ступенів свободи, але маніпулятори набагато сильніше, не втомлюються і миттєво завмирають, якщо хірург відпустить джойстики. Свої дії лікар контролює через окуляри, куди надходить збільшена до 12 разів картинка з відеокамери [6].



Рис. 18. Робот-хірург *Da Vinci*

Kirobo – робот-антидепресант для космонавтів, заввишки всього 34 см. Створений спеціально для «живого» спілкування з людиною. Робот розмовляє, розуміє сказане і природно реагує на питання. Штучний інтелект Kirobo відрізняє людську мову (поки лише японську) від оточуючих звуків, виділяє в її потоці окремі слова і визначає сенс фраз. Андроїд запам'ятовує і впізнає конкретних людей, розрізняє емоції, виражені мімікою і жестами. Тіло робота має 20 ступенів свободи, тож Kirobo відповідає людині не тільки словом, а й рухами [20].



Рис. 19. Робот Kirobo

Лікар RP-VITA – робот телеприсутності, за його допомоги лікар може віртуально робити обходи або спостерігати за важким хворим цілодобово, перебуваючи в іншому місці. Лікарняними коридорами за доктора буде кататися робот заввишки близько 1,5 м, який прокладає шлях за допомогою системи лазерних і звукових датчиків. У палаті пацієнт або медсестра бачать обличчя лікаря на екрані і можуть поспілкуватися з ним. RP-VITA возить з собою базовий набір діагностичних інструментів, і, за потреби певних уточнень, медсестра тут же проводить обстеження. Лікаря для спілкування з пацієнтом потрібен лише ноутбук або планшет [31].



Рис. 20. Робот RP-VITA

Екзоскелет HAL – робот-костюм, призначений для того, щоб в буквальному значенні підняти на ноги паралізованих людей. Датчики екзоскелета, прикріплені до поверхні шкіри, зчитують слабкі електричні імпульси, які мозок посилає м'язам, а потім двигуни робота роблять всю роботу. HAL існує в двох варіантах: цілий скелет або тільки «ноги» [12].



Рис. 21. Екзоскелет-HAL

IBM Watson – робот для діагностування онкологічних захворювань. IBM Watson є суперкомп'ютером з 90 серверів по 4 восьми-ядерні процесори в кожному, а його оперативна пам'ять – 16 терабайт. Ватсон – машина зі штучним інтелектом, він самостійно вивчає джерела інформації і робить висновки. Перш ніж приступити до роботи, майбутній діагност проаналізував 605 000 медичних документів. Лікар завантажує в пам'ять робота історію хвороби і через декілька хвилин отримує ймовірний діагноз і курс лікування. Якщо лікарю потрібно щось уточнити, він може задати Ватсону питання в письмовій формі [17].



Рис. 22. IBM Watson

HOSPI – робот-фармацевт, заввишки 130 см, який перевозить до 20 кг ліків і зразків. У пам'ять робота записують вказівки, кому з хворих які препарати прописані, і HOSPI сам обирає оптимальний маршрут. По дорозі він оминає перешкоди, в тому числі рухомі. Прибувши на сестринський пост, робот повідомляє, що і кому він привіз. Персоналу залишається тільки віддати ліки пацієнтам [26].



Рис. 23. HOSPI

Цікавими є роботи-зоотерапевти – вони покликані допомагати людям та тваринам.

Ще однією галуззю використання роботів є космос. Застосування роботів у дослідженнях космосу відбувається вже не перший рік, проте розробники постійно працюють над вдосконаленням вже створених моделей і проектують нові. Вчені не дійшли однозначного висновку з приводу того, кого краще і ефективніше посилати досліджувати космос: людину або все-таки робота. В даному питанні є багато схожого з тим, коли проводилися суперечки про активне використання промислових роботів на різних виробництвах. Головною перевагою космічних роботів є їх економічність. На підготовку такого дослідження не потрібно докладати стільки зусиль і витратити досить багато коштів, як на повноцінну підготовку людини до польоту в космос. Тим більше, часто використовують так званих неповернутих роботів, що додатково та істотно скорочує витрати на подібні космічні дослідження.

Всесвітньо відома розробка NASA і General Motors – проект «Робонавт» – вже декілька років допомагає астронавтам на космічній станції.

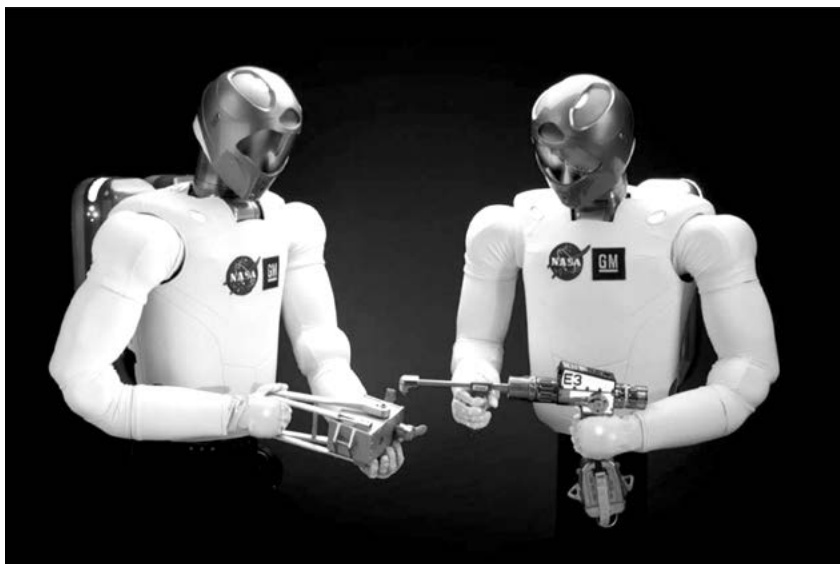


Рис. 24. Робонавт

Також активно використовуються роботизовані системи в **галузі безпеки**. Ці роботи першими виявляють пожежонебезпечні ситуації і успішно запобігають їх виникненню та розповсюдженню.

Сучасні **військові навчання** максимально наближені до умов реальності, завдяки роботам, що імітують противника. Роботи для військових навчань імітують людські імпульси і звички. Також вони здатні проводити тривале стеження за об'єктами, що викликають підозру в органах правопорядку.

Неможливо уявити собі **сучасні заводи** без роботизованої техніки. Роботи виконують безліч найрізноманітніших операцій. В основному – це дії, що вимагають багаторазового повторення і високої точності. Найчастіше застосування роботів рятує цілі галузі промисловості. Адже їх застосування дозволяє значно збільшити продуктивність праці, звільнивши при цьому людські ресурси для вирішення більш важливих завдань.



Рис. 25. Промисловий робот

Роботи застосовуються і в **побуті**. Найвідоміші з них – робот-пилосос і газонокосарка. Також можна зустріти роботів, спеціально розроблених для виконання більш складних побутових завдань. У Японії відкрився готель, в якому прислуга є роботами (рис. 26) [13].

Є роботи, покликані нести людям радість, розважаючи їх своїми вміннями. В більшості своїй, такі роботи представляють світ дитячих іграшок: співаючі і танцюючі тварини, інтерактивні іграш-



Рис. 26. Роботи в готелі Henn-na Hotel

ки, радіокеровані машини і гелікоптери. Втім, роботи для розваги дорослих відрізняються від дитячих, хіба що розмірами.

Настає ера революції та еволюції в галузі робототехніки в напрямку створення людиноподібних роботів-гуманоїдів. Лідером в цій галузі є Японія. Яскравим прикладом є «привабливий робот» Еріка, яка захоплюється театром і анімаційними фільмами та хоче відвідати Південно-Східну Азію. Передовий гуманоїд вийшов завдяки спільним зусиллям університетів Osaka і Kyoto, науково-дослідного інституту сучасних телекомунікаційних систем International (ATR). Автором Еріки є Hiroshi Ishiguro – професор Intelligent Robotics Laboratory в університеті Osaka (розробник адроїда Geminoid HI-1, який є точною копією свого творця). Еріка не в змозі самостійно ходити, але володіє поліпшеною мовою і здатністю розуміти, реагувати на питання, висловлювати свою думку і супроводжувати її відповідною мімікою [14].



Рис. 27. Робот Еріка

ASIMO – Advanced Step in Innovative Mobility, або «Передовий крок в інноваційній мобільності» є найдосконалішим людиноподібним роботом в світі, що пересувається на двох ногах. Нова сучасна технологія руху, розроблена Honda, дозволяє ASIMO пересуватися не тільки вперед і назад, але і повертати при ходьбі, підніматися і спускатися сходами і повертати за кут. Завдяки цьому можна вважати, що ASIMO найближче підійшов до точного відтворення природного руху людини. Основну роль в цьому відіграє інтелектуальна технологія ходьби, розроблена компанією Honda, використовуючи яку, ASIMO може плавно рухатися в режимі реального часу. Ця технологія заснована на тому, що робот наперед планує свої дії і відповідно підлаштовується під них. Наприклад, як людина, огинаючи кут, нахилиється всередину дуги повороту, зміщуючи тим самим свій центр ваги, так і ASIMO планує свій наступний крок і зміщує свій центр ваги необхідним чином. Це дозволяє в реальному часі планувати наступний рух, завдяки чому ASIMO здатний пересуватися, не зупиняючись. Для того, щоб звернути за ріг, йому не потрібно зупинитися і розгортатися; замість цього він здійснює один безперервний плавний рух [2].



Рис. 28. Робот Asimo

Цікавим є робот **NAO**, метою створення якого є позитивний вплив на повсякденне життя людини, починаючи від допомоги по дому і розваги до терапії аутизму. Тому творці сконцентровані на поліпшенні взаємодії робота і людини в двосторонньому порядку за допомогою мови тіла. NAO – дуже емоційний робот. Навіть сидючи, склавши руки, він рухливий: похитується, хитає головою, може розглядати свої «нігті». Це працює дуже ефектно. Завдяки цьому робот здається набагато «людянішим».

Компанія IDC опублікувала звіт під назвою «IDC FutureScape: Прогнози для ринку робототехніки на 2017 рік», де вказує ключові драйвери ринку і розкриває, як вони, ймовірно, впливатимуть на розвиток робототехніки до 2020 року.

«Розвиток технологій штучного інтелекту, машинного зору, навігації, MEMS-сенсорів і виробництва напівпровідників продовжує рухати інновацію щодо можливостей, робочих параметрів, автономності, простоти використання і економічної рентабельності



Рис. 29. Робот Nao

промислових і обслуговуючих роботів», – пише д-р Цзин Бін Чжан (Jing Bing Zhang), автор звіту і директор дослідження, Worldwide Robotics & Asia Pacific Manufacturing Insights, IDC Asia / Pacific [18].

У звіті даються також стратегічні прогнози і вказані головні тренди розвитку робототехніки, які відкривають можливості для ІТ-керівників в 2017 році і далі:

1. **Робот-як-послуга.** До 2019 р. 30% комерційних застосувань обслуговуючих роботів будуть пропонуватися в рамках бізнес-моделі «робот-як-послуга», знижуючи витрати на впровадження роботів.
2. **Директор з робототехніки.** До 2019 р. 30% провідних організацій введуть посаду директора з робототехніки (Chief Robotics Officer) і / або окреслять коло обов'язків, пов'язаних з робототехнікою, у своєму бізнесі.
3. **Мінливий конкурентний ландшафт.** До 2020 р. організації матимуть більший вибір вендорів, оскільки нові гравці вийдуть на ІКТ-ринок з оборотом 80 млрд. дол. з рішеннями щодо впровадження робототехніки.

4. **Брак фахівців робототехніки.** До 2020 р. зростаюче застосування робототехніки збільшить попит на фахівців, у результаті чого 35% пов'язаних з робототехніки вакансій залишаться незаповненими, а середня зарплата таких фахівців зросте як мінімум на 60%.
5. **Робототехніка стане регульованою галуззю.** До 2019 р. держава почне впроваджувати заходи з регулювання ринку робототехніки, щоб зберегти робочі місця і вирішити проблеми захищеності технології, фізичної безпеки.
6. **Програмно-керовані (software-defined) роботи.** До 2020 р. 60% всіх роботів будуть використовувати хмарне ПЗ для здобуття нових вмінь, когнітивних здібностей і як джерело прикладних програм, що призведе до формування ринку хмарних рішень для робототехніки.
7. **Співпрацюючі роботи.** До 2018 р. 30% усіх нових впроваджень робототехніки являтимуть собою інтелектуальних взаємодіючих (collaborative) роботів, що виконують роботу втричі швидше, ніж сьогодні, і є безпечними для роботи серед людей.
8. **Інтелектуальна мережа роботів RoboNet.** До 2020 р. 40% серійного виробництва роботів будуть підключені до мережі «загального інтелекту», що забезпечить 200%-ве підвищення загальної ефективності використання роботів.
9. **Розширення сфери застосування.** До 2019 р. 35% провідних організацій в галузі логістики, охорони здоров'я, комунальних послуг та гірничодобувної промисловості вивчатимуть можливості використання роботів для автоматизації робіт.
10. **Робототехніка для І-комерції.** До 2018 р. 45% з 200 провідних глобальних компаній І-комерції і багатоканальних продажів впроваджуватимуть робототехнічні системи в своїх складських господарствах виконання замовлень і відділах доставки.

ЗАПИТАННЯ ДЛЯ ОБГОВОРЕННЯ

1. Охарактеризуйте поняття робототехніки. Вкажіть, які предмети шкільної програми безпосередньо пов'язані з робототехнікою.
2. Як визначається поняття «робот»? Які найбільш характерні ознаки робота?
3. Яке призначення та структурно-функціональна схема взаємодії основних систем робота?
4. Назвіть галузі застосування роботів. Які основні напрямки розвитку робототехніки?
5. Зробіть прогноз розвитку поколінь роботів на найближче майбутнє.
6. Доведіть необхідність створення штучного інтелекту.
7. Змоделюйте ситуації, в яких роботи могли б виконати Вашу повсякденну роботу. Чи відбувається це сьогодні?

ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Підготуйте презентації на одну з тем:
 - a. Перспективні напрямки розвитку робототехніки.
 - b. Робототехніка і кібернетика.
 - c. Робототехніка і біоніка.
 - d. Застосування роботів.
 - e. Промислові роботи.
 - f. Космічні роботи.
 - g. Військові роботи.
 - h. Історія розвитку робототехніки.
 - i. Штучний інтелект та робототехніка.
2. Використовуючи рекомендовані друковані видання та інтернет-джерела, складіть зведену таблицю з обладнання для вивчення освітньої робототехніки (назва конструктора, освітні завдання, які вирішуються при використанні даного конструктора, передбачуваний вік учнів, можливості використання в навчальному процесі, переваги, недоліки).



3 КОНСТРУКТОР LEGO MINDSTORMS EV3

3.1. КОНСТРУКТОР LEGO MINDSTORMS EV3

LEGO MINDSTORMS Education EV3 – це третє покоління робототехнічних конструкторів серії LEGO Mindstorms. Даний набір розроблявся в Массачусетському технологічному інституті (MIT – Massachusetts Institute of Technology) спільно з компанією LEGO. Набір дає можливість конструювати роботів з безліччю датчиків і моторів або вимірювати відстань, освітлення, температуру, проводячи наукові експерименти. Перший LEGO Mindstorms RCX був випущений в 1998 р., основою для нього став мікроконтролер H8. Наступним поколінням став LEGO MINDSTORMS Education NXT, випущений у 2006 р. Він помітно розширив функціонал першого набору, також в ньому використовувався новий мікроконтролер ARM7 32bit CPU. ROBO LAB розроблено як основу для LabView компанією National Instruments.

EV3 є інтелектуальним, керованим комп'ютером на базі елементів LEGO і системи MINDSTORMS. Система MINDSTORMS називається так завдяки книзі Сеймура Пейперта «Переворот у свідомості: діти, комп'ютери та плідні ідеї», в якій автор аналізує способи формування мислення дітей і роль комп'ютерів у цьому процесі [27].

Конструктор Lego Mindstorms EV3 складається з електронних та механічних елементів. Основа конструктора – програмований блок EV3 (його ще називають «брік»), інтерактивні сервомотори і декілька датчиків.

Склад конструктора **Lego Mindstorms EV3:**



Рис. 30. Конструктор *Lego Mindstorms EV3*
Балки – використовуються як каркаси конструкцій.



Рис. 31. *Балки*

Осі – передають оберти від мотора колесам, шестерням.



Рис. 32. *Осі*

Піни – слугують з'єднаннями для балок, датчиків, сенсорів.



Рис. 33. *Піни*

Конектори – потрібні для з'єднання балок в різних площинах, зміни кутів з'єднання деталей і під'єднання датчиків, сенсорів до робота.



Рис. 34. *Конектори*

Шестерні – призначені для передачі обертання від моторів до інших елементів конструкції робота.



Рис. 35. *Шестерні*

Колеса, шків та гусениці – забезпечують роботу рух у просторі.



Рис. 36. *Колеса, шків та гусениці*

Декоративні деталі – використовують для надання оригінальності зовнішньому вигляду робота.



Рис. 37. Декоративні деталі

Мотори – запускають в дію робота.

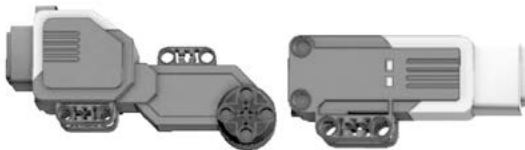


Рис. 38. Мотори

Датчики та сенсори – забезпечують робота «відчуттями».



Рис. 39. Датчики та сенсори

Головний блок EV3 – «мозок» робота.

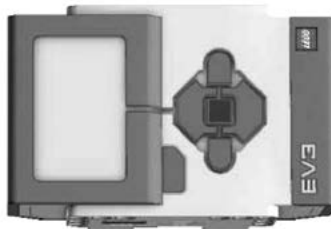



Рис. 40. Головний блок

Конструктивні елементи Lego Mindstorms EV3

| Шестерні (зубчасті колеса) | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| 8-ми зубчасте колесо | 16-ти зубчасте колесо | 24-ти зубчасте колесо | Черв'ячне зубчасте колесо | 4-ох зубчасте колесо | Зубчасте колесо 24 зуба всередині, 56 – зовні |

| Ремені, осі, втулки для осей | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| Ремінь, 33 мм | Втулка, подов- жувач осі | Втулка | Вісь, 5-ти модульна | Вісь, 6-ти модульна | Вісь, 12-ти модульна |

| Колеса з шинами та шківі різного діаметру | | | | | |
|--|--|--|--|--|--|
|  |  |  |  |  |  |
| Ступиця | Колесо | Ступиця | Колесо | Ступиця | Колесо |

| Орієнтуючі та комбіновані з'єднувачі | | | | |
|---|---|---|---|---|
|  |  |  |  |  |
| З'єдну- вальний штифт 2-мо- дульний подвійний | З'єдну- вальний штифт 3-мо- дульний подвійний | Фіксатор 2-модуль- ний | Фіксатор 2-модуль- ний | Кутовий фіксатор |

| Циліндричні фіксатори | | | | |
|---|---|---|---|---|
|  |  |  |  |  |
| З'єдну- вальний штифт піввісь | З'єдну- вальний штифт з виступами / піввісь | З'єдну- вальний штифт з втулкою | З'єдну- вальний штифт з виступами | З'єдну- вальний штифт з виступами |

| Балки | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| Балка 3-мо- дульна | Балка 5-мо- дульна | Вигнута балка 3*5 мо- дульна | Балка з шипамі 1*2 | Балка з шипамі 1*5 | Пластина з отво- рами 2*4 |

3.2. ПРОГРАМНЕ СЕРЕДОВИЩЕ MINDSTORMS EV3

Встановлення ПЗ

Перш ніж встановити середовище програмування, варто перевірити, чи задовольняє комп'ютер системним вимогам. Мінімальні вимоги складають:

- операційна система Windows (XP, Vista, 7, 8) або MacOS (10.6, 10.7, 10.8);
- 2 Гб оперативної пам'яті;
- двоядерний процесор 2,0 ГГц або більш потужний;
- 2 Гб вільного дискового простору;
- Роздільна здатність екрану 1024*768;
- Microsoft Silverlight 5.0;
- Microsoft.NET 4.0.

Завантажити ПЗ можна з офіційного сайту компанії LEGO (<http://www.lego.com/>), в меню *Завантаження*.



Рис. 41. Завантаження програмного забезпечення

Потрібно обрати операційну систему, на яку буде встановлюватися ПЗ, та запустити виконувальний файл.

Слід зазначити, що основним об’єктом, над яким доведеться працювати, буде проект. Проект дозволяє більш комплексно підійти до завдання створення робота.

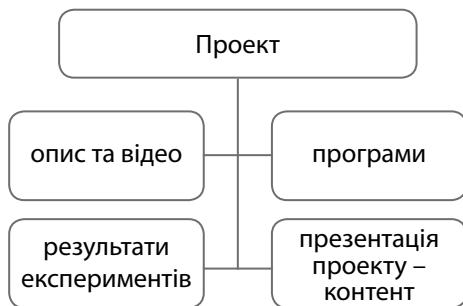


Рис. 42. Склад проекту

Після запуску програмного середовища EV3 на екрані з’являється наступне вікно:



Рис. 43. Головна сторінка програмного забезпечення

1. За замовчуванням активна вкладка **Лобі**.
2. **Додати проект** – використовується для додавання нового проекту, для того, щоб мати можливість почати програмування робота.
3. **Завдання для роботів** – дозволяє будувати і програмувати п'ять основних моделей роботів.
4. **Відкрити останні** – можна здійснити доступ до останніх проектів, з якими працювали на комп'ютері.
5. **Короткий посібник** – допоміжні ресурси (короткі вступні відеоролики, керівництво користувача EV3 і довідка з програмного забезпечення).
6. **Новини** – повідомлення з сайту LEGO.com/Mindstorms.
7. **Інші роботи** – дозволяє будувати і програмувати інші моделі роботів.

Створення нового проекту

Для створення нового проекту потрібно натиснути *Файл/Новий проект* або на плюс поряд з вкладкою *Лобі* (рис. 44).



Рис. 44. Створення нового проекту

З'явиться нова вкладка, яка вказує на те, що було відкрито нову програму, і автоматично створюється файл папки проекту. Всі програми, зображення, звуки, відео, інструкції та інші матеріали, що використовуються в проекті, будуть автоматично зберігатися в цій папці проекту. Це дозволяє легко зберігати проект і ділитися ним з іншими.



Рис. 45. Основні елементи програмного забезпечення

Кожен проект буде відображатися у вигляді вкладки у верхній частині екрану. На рисунку відображаються вкладки програм, які належать до вибраного проекту. Можна додати новий проект або програму, натиснувши кнопку **+** праворуч від інших кнопок. Щоб закрити вкладку, слід натиснути **x**.

Сторінка властивостей проекту

Вкладка *Властивості проекту* дозволяє переміститися на сторінку, яка містить упорядковане відображення вмісту проекту, включаючи всі програми, зображення, звук та інші ресурси. Тут можна описати проект за допомогою тексту, зображень та відео, які визначають, як проект буде відображатися в лобі.

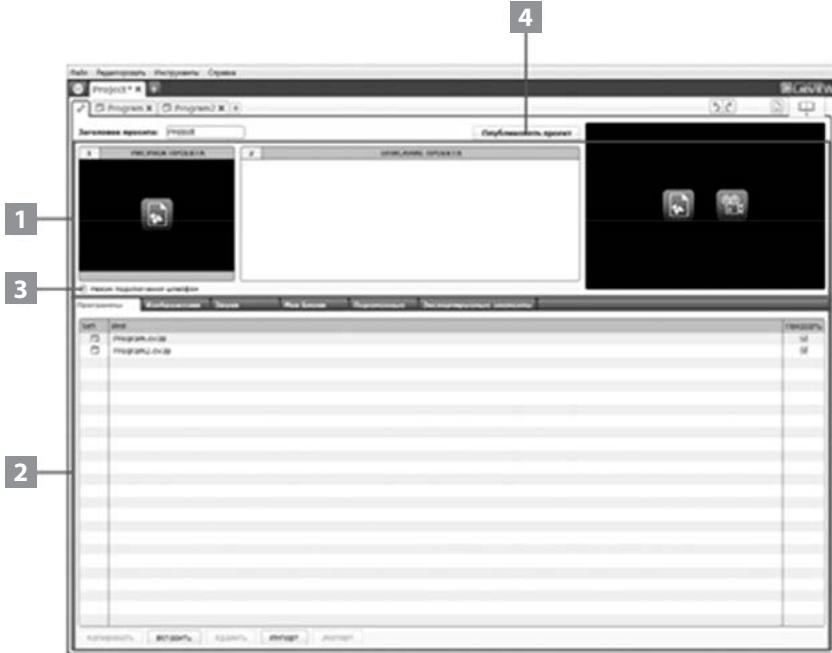


Рис. 46. Параметри проекту

Розрізняють такі властивості проекту:

1. **Опис проекту** – потрібно визначити назву проекту, описати його і вставити зображення і відео, які повинні відображатися в лобі при перегляді проекту.
2. **Огляд вмісту проекту** – містить всі матеріали, що використовуються в проекті, наприклад: програми, зображення, звуки та *Мої Блоки*.
3. **Режим підключення шлейфом** – ця опція включає режим підключення шлейфом, завдяки цьому можна програмувати до чотирьох модулів EV3.

4. **Опублікувати** – дозволяє опублікувати проект на сайті LEGO.com/mindstorms (необхідно підключення до Інтернету).

Опис завдання для роботів

У вкладинці *Лобі* програмного забезпечення EV3 представлено п'ять роботів: TRACK3R, SPIK3R, EV3RSTORM, R3PTAR і GRIPP3R.



Рис. 47. Основні роботи

Зазначених роботів створили конструктори LEGO® для того, щоб показати кілька способів побудови та програмування роботів за допомогою LEGO Mindstorms® EV3. Після вибору робота на екрані відображається розділ *Огляд завдання* для обраного робота. Завдання призначені для навчання основам програмування і ознайомлення з апаратними засобами EV3 та принципами конструювання.

Кожне завдання допомагає побудувати і запрограмувати певну складову робота. Слід почати з першого завдання і після його виконання перейти до наступного. Після виконання всіх завдань, робот буде зібраний і готовий виконувати команди.

Всі завдання складаються з чотирьох кроків:

- + Завдання
- + Створення
- + Команда
- + Вперед!

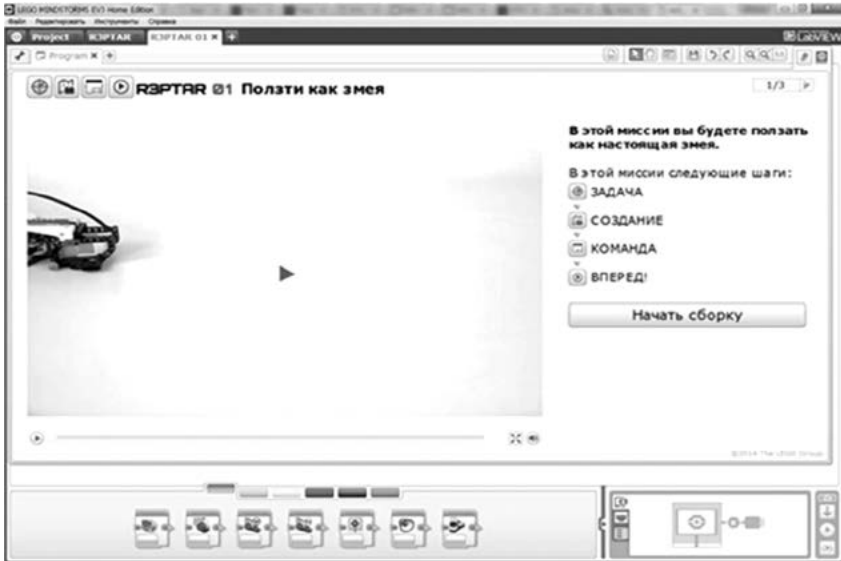


Рис. 48. Проект створення робота

Інтерфейс програми

Створеного робота можна запрограмувати за допомогою інтуїтивно зрозумілого програмного інтерфейсу, в якому використовуються піктограми.

Програмний інтерфейс EV3 складається з таких основних областей:

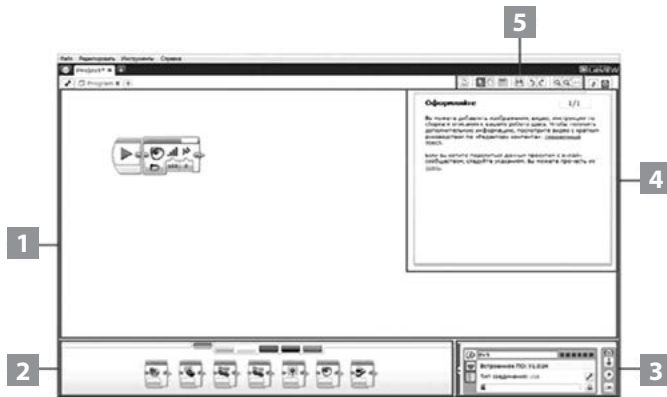


Рис. 49. Інтерфейс програми

1. **Область програмування** – область, де відображається програма, яка створюється.
2. **Вкладки програмування** – містяться конструктивні блоки для програми.
3. **Сторінка апаратних засобів** – використовується для встановлення і керування зв'язком з модулем EV3 і перегляду, як підключені мотори і датчики. Тут можна завантажити програми до модуля EV3.
4. **Редактор контенту** – електронний зошит, вбудований у програмний продукт.
5. **Панель інструментів програмування** – знаходяться основні інструменти для роботи з програмою.




Блоки програмування





Зелені блоки дій



Рис. 50. Блоки дій

Керують діями в рамках програми. Контролюють оберти моторів, зображення, звук та підсвічування модуля EV3.

| Блок | Назва блоку |
|---|-------------------|
|  | Середній мотор |
|  | Великий мотор |
|  | Рульове керування |

| | |
|---|---------------------------------|
|  | Незалежне керування |
|  | Екран |
|  | Звук |
|  | Індикатор стану головного блоку |

Оранжеві блоки виконання програми

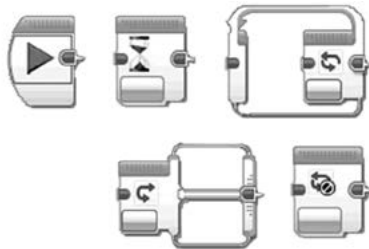



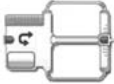



Рис. 51. Блоки виконання програми

Керує процесом виконання програми. Всі програми починаються зі стартового блоку.

| Блок | Назва блоку |
|---|-------------|
|  | Початок |
|  | Очікування |
|  | Цикл |

| | |
|---|------------------|
|  | Перемикач |
|  | Завершення цикла |

Жовті блоки датчиків та сенсорів

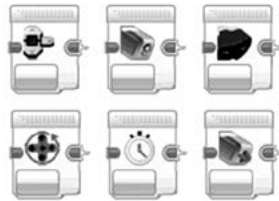







Рис. 52. Блоки датчиків

Блоки датчиків дозволяють програмі отримувати дані, що надходять з датчика кольору, ІЧ датчика, дотику тощо.

| Блок | Назва блоку |
|---|--------------------------|
|  | Кнопки керування модулем |
|  | Сенсор кольору |
|  | Інфрачервоний сенсор |
|  | Обертання мотору |
|  | Таймер |
|  | Датчик дотику |

Червоні блоки операцій з даними

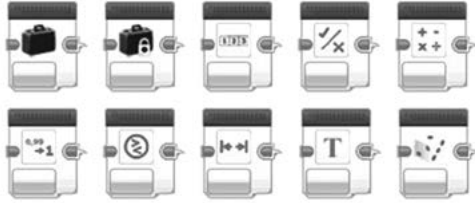












Рис. 53. Блоки дій з даними

Дозволяють додавати та зчитувати змінні величини, порівнювати їх характеристики тощо.

| Блок | Назва блоку |
|---|------------------------------------|
|  | Змінна |
|  | Константа |
|  | Виконання операцій з масивом |
|  | Логічне «і» |
|  | Арифметичні дії |
|  | Округлення значень до цілого числа |
|  | Порівняння |
|  | Значення по модулю |



| | |
|---|--------------------|
|  | Текст |
|  | Випадкове значення |





Сині блоки модернізації



Рис. 54. Блоки модернізації

Блоки модернізації дозволяють працювати з файлами, встановлювати зв'язок по Bluetooth та ін.

| Блок | Назва блоку |
|---|-----------------------------|
|  | Доступ до файлу |
|  | Обмін повідомленнями |
|  | Зв'язок по Bluetooth |
|  | Підтримка в активному стані |
|  | Незворотне значення датчика |

| | |
|---|-------------------------------|
|  | Нерегульований мотор |
|  | Інвертування оборотів двигуна |
|  | Зупинка мотора |
|  | Коментар |

Сторінка апаратних засобів

Вкладка *Інформація про модуль* відображає важливу інформацію про модуль EV3, який підключено в даний момент, таку як назва модуля EV3, рівень заряду батареї, версію вбудованого ПЗ, тип підключення і панель пам'яті. Вона також надає доступ до оглядача пам'яті та інструментів налаштування бездротового підключення.



Рис. 55. Сторінка апаратних засобів

Вкладка *Порт підключення* відображає інформацію про датчики і мотори, підключені до модуля EV3. При підключенні нового модуля EV3 до комп'ютера ця інформація автоматично ідентифікується, що дозволяє бачити поточні значення. Якщо створений мо-

дуль EV3 не підключено, можна, тим не менш, налаштувати вкладку *Порт підключення* вручну. Для цього слід обрати порт, потім обрати належний датчик або мотор зі списку.

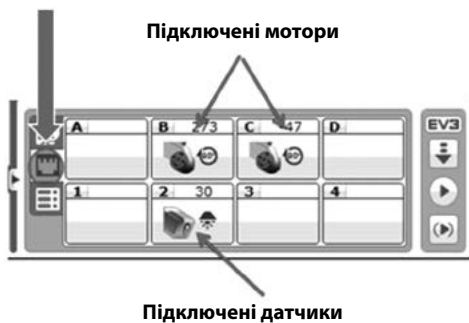


Рис. 56. Відображення підключених датчиків та моторів

Вкладка *Доступні модулі* показує модулі EV3, які в даний момент доступні для підключення. Можна вибрати модуль EV3, який потрібно підключити, і тип зв'язку. Також можна відключити існуюче підключення модуля EV3. Більш детальну інформацію про те, як користуватися сторінкою апаратних засобів, можна знайти в меню *Довідка* програмного забезпечення EV3.

Існують три способи встановлення зв'язку:

- підключення через USB;
- підключення через Bluetooth;
- підключення через Wi-Fi.



Рис. 57. Встановлення зв'язку з роботом

Програмування моторів

Одна з основних функцій робота – рух. Можна забезпечити рух всього робота, так і окремих його складових. Рухом ке-

рують сервомотори (або сервоприводи). У конструкторі Lego Mindstorms сервомотори мають датчики обертів, підраховують кількість градусів повороту головної осі. Важливим компонентом сервомотора є редуктор, який через систему шестерень перетворює надзвичайно швидке обертання внутрішнього електричного двигуна в більш повільне. Наявність датчика оборотів і редуктора дозволяють сервомоторами здійснювати прецизійний рух головної осі. Сервопривод Lego може бути повернутий з точністю 1°.

Прямолінійний рух та повороти робота здійснюються за допомогою великих моторів, що обертаються у заданих напрямках в залежності від встановленого значення потужності: позитивне значення (наприклад, +75) обертатиме мотор за годинниковою стрілкою, а негативне (наприклад, -45) – проти годинникової стрілки. Те ж саме стосується і середніх моторів.

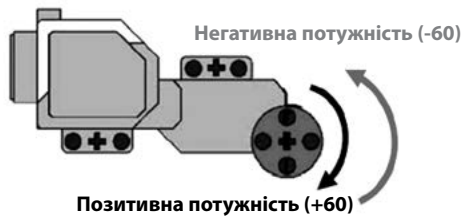


Рис. 58. Оберти двигуна

Програмування руху відбувається за допомогою вкладки *Дія*.

Блок Рульове управління

Блок *Рульове управління* дозволяє роботу рухатися вперед, назад, повертатися або зупинятися. Можна регулювати рульове управління, щоб змусити робота їхати прямо, рухатися за дугою або повертатися на різний кут.

Блок *Рульове управління* використовують для роботизованих транспортних засобів, в яких є два великих мотора, де один мотор управляє лівою стороною транспортного засобу, а другий – правою. Блок *Рульове управління* управляє обома моторами одночасно, щоб транспортний засіб рухався у вибраному напрямку.

Обов'язково слід переконатися, що у робота, направленою вперед, мотор з лівого боку – це перший мотор, зазначений в переліку вибору порту. Інакше робот буде повертатися в неправильному напрямку.

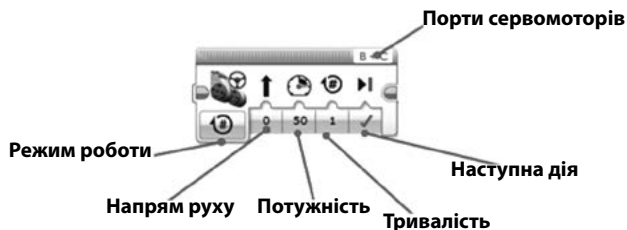


Рис. 59. Параметри блока *Рульове управління*

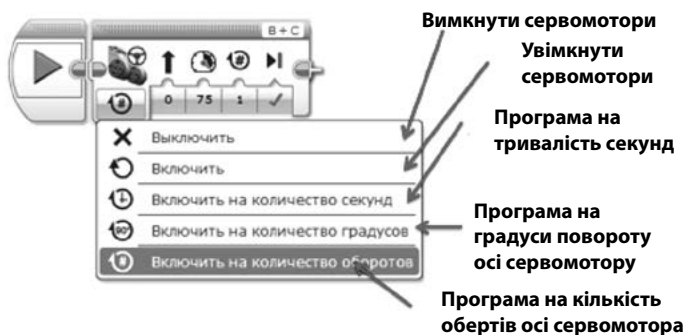


Рис. 60. Параметри запуску двигуна

Блок *Незалежне керування моторами* дозволяє роботу рухатися вперед, назад, повертатися або зупинятися. Можна запрограмувати два мотори на обертання з різними швидкостями або в різних напрямках, щоб робот повертав. Основні функції:

- контролює два мотори та дозволяє вмикати мотори з різною потужністю, включаючи різний напрям обертання для поворотів та кружляння на місці;
- для поворотів: один мотор має потужність 0, інший – позитивну (рух вперед) чи негативну (рух назад) потужність;

- коли встановлена потужність 0, мотор заблоковано та не рухатиметься, що підвищує точність поворотів;
- для кружляння: один мотор має позитивну потужність, інший – негативну.

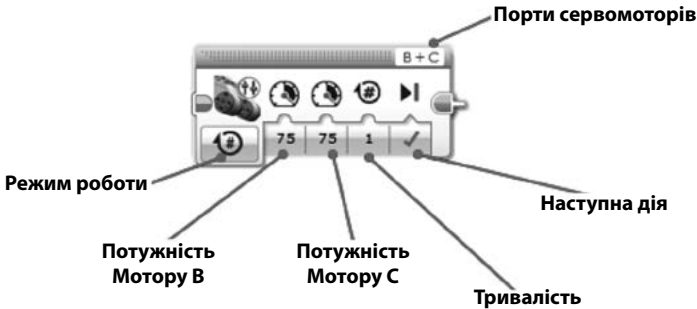


Рис. 61. Параметри блока
Незалежне керування моторами

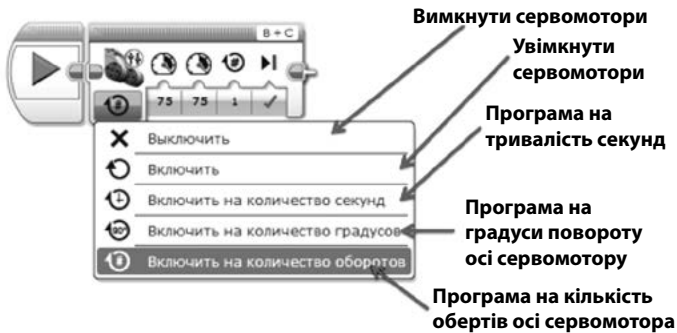


Рис. 62. Параметри запуску двигуна

Блок *Великий мотор* призначено для контролю одного великого мотору. Коли встановлена потужність 0, мотор заблоковано та не рухатиметься, що підвищує точність поворотів.

Блок *Середній мотор* контролює середній мотор, який можна вимкнути/увімкнути, виставити його потужність чи увімкнути на виставлену кількість часу чи обертів. Коли виставлена потуж-

ність 0, мотор заблоковано та не рухається. Використовує позитивну та негативну потужність для контролю напрямку.

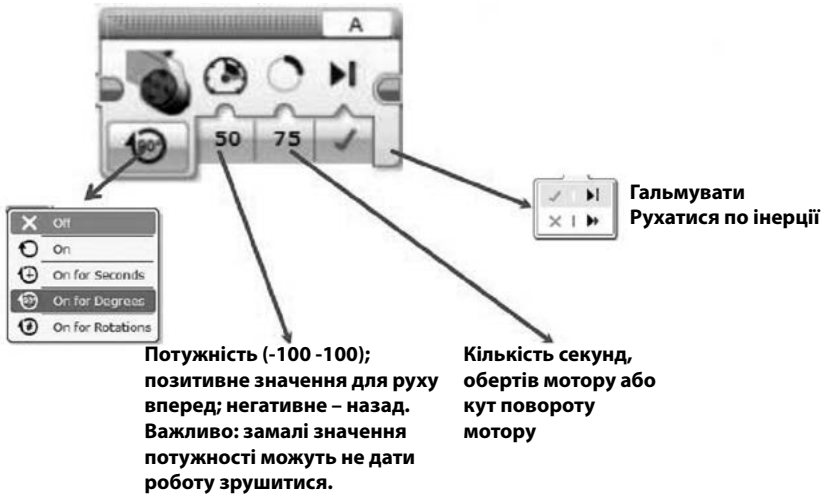


Рис. 63. Параметри блока Великий мотор

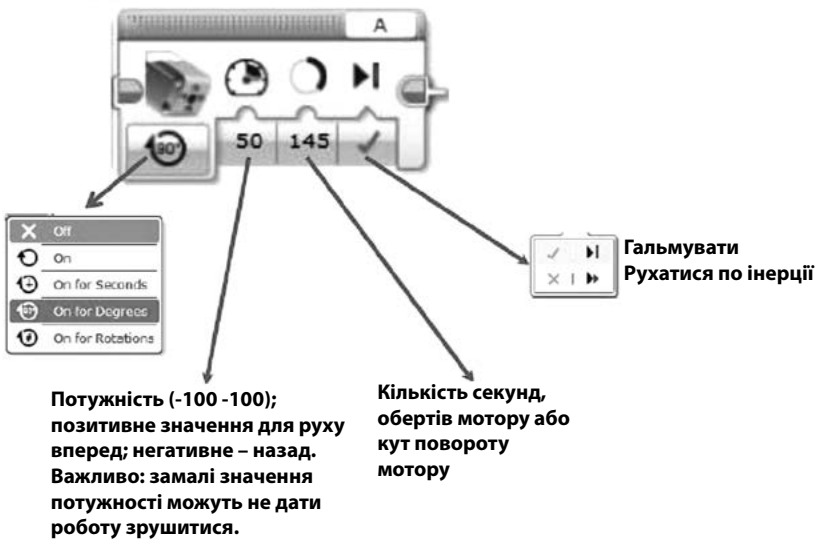


Рис. 64. Параметри блока Середній мотор

Етапи створення програми

Кожна програма складається з ланцюжка блоків, які задають певну дію або перевірку різних умов. Кожен блок має безліч різних параметрів. Перший, оранжевий блок із зеленим трикутником всередині називається – *Початок*.



Рис. 65. Блок Початок

Саме з нього починається будь-яка програма. Інші блоки додаються в потрібному порядку.



Рис. 66. Вибір потрібних блоків

Для створення програми потрібно:

1. Натиснувши та утримуючи блок лівою кнопкою миші, потягнути його нагору.

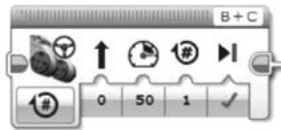


Рис. 67. Блок Рухове управління

2. Відпустити програмний блок, коли з'явиться сірий прямокутник.

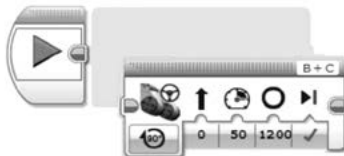


Рис. 68. Фіксування блоку

3. Обрати потрібні параметри.



Рис. 69. Зазначення параметрів

4. Натиснути *Завантажити*, щоб скопіювати та завантажити програму на EV3.

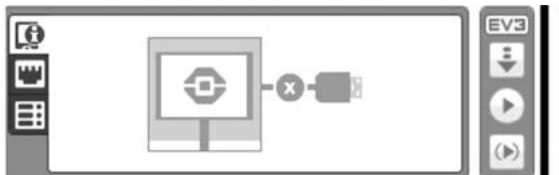


Рис. 70. Завантаження програми у пам'ять робота

Приклади програм

Різні варіанти поворотів робота:

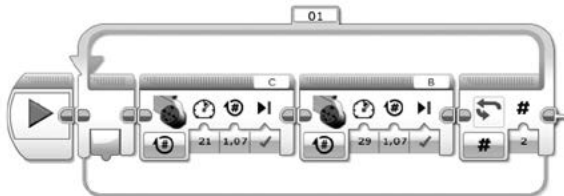


Рис. 71. Програма повороту для окремих моторів



Рис. 72. Програма повороту з використанням рульового керування

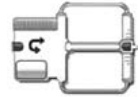


Рис. 73. Програма повороту з використанням незалежного керування

Примітка: Програмний блок Цикл багаторазово виконує програмні блоки, вкладені в його середину, поки не буде виконана умова завершення циклу, задана в настройках блоку.



Програмний блок Перемикач служить для того, щоб в залежності від заданих умов виконати одну послідовність програмних блоків, вкладених в один зі своїх контейнерів.



Робот, що здійснює рух квадратом:

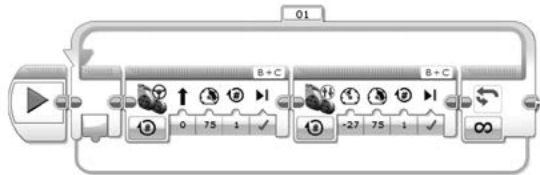


Рис. 74. Програма руху квадратом

На головному блоці робота відбувається підсвічування різними кольорами:

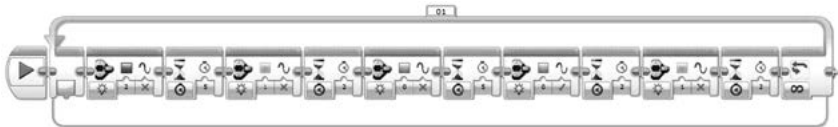


Рис. 75. Програма підсвічування головного блоку

Робот відтворює мелодію:



Рис. 76. Програма запуску мелодії

На екрані головного блока робота висвічується зображення кола:

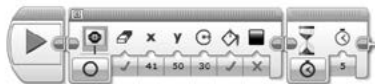


Рис. 77. Програма відображення кола на екрані

3.3. ДАТЧИК ТА СЕНСОРИ

До складу конструктора Lego Mindstorms EV3 входять різні датчики та сенсори. Головне завдання датчиків – представляти інформацію із зовнішнього середовища модулю EV3, а завдання користувача – навчитися отримувати і опрацьовувати цю інформацію, подаючи необхідні команди моторам робота.

Датчик дотику

Аналоговий датчик дотику – простий, але високоточний інструмент, який визначає чи натиснута його кнопка чи ні, а також здатний визначити кількість натискань, як одиночних, так і множинних. Його можна використовувати для побудови систем контролю запуску чи зупинки, для створення роботів, здатних вийти з лабіринту. Все це дає глибоке розуміння технології, що застосовується в цифрових музичних інструментах, комп'ютерних клавіатурах та кухонних пристроях.



Рис.78. Датчик дотику

Цей датчик є спеціальною кнопкою, яка може перебувати в двох станах: *Натискання* (1) або *Звільнення* (2). Послідовний пере-

хід у стан *Натискання*, а потім *Вивільнення* називається *Клацанням* (3) і може опрацьовується програмою як самостійна подія.

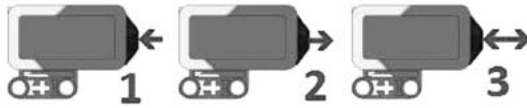


Рис. 79. Стани датчика дотику

Для створення програми з використання *Датчику кольору* потрібно використати блок *Очікування*. Цей блок змушує програму очікувати виконання певної умови або настання деякої події. Поки не виконається умова, встановлена в цьому блоці, програма не перейде до виконання наступних програмних блоків у ланцюжку. Якщо перед тим, як почнеться виконання блоку *Очікування* були включені будь-які мотори, то вони будуть продовжувати обертання зі встановленою швидкістю.

Для створення програми для робота, який має зупинитись при натисканні кнопки (тобто під час зіштовхування з об'єктом), потрібно:

Після блоку *Початок* потрібно додати блок *Рульове управління* для запуску двигунів.

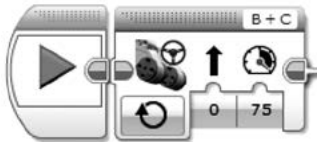


Рис. 80. Додавання блоку *Рульове управління*

Далі додати блок *Очікування* та змінити режим блока на *Датчик дотику – Порівняння*.



Рис. 81. Додавання блоку *Очікування*

Якщо датчик спрацює, то робот має зупинитись, тому потрібно знову додати блок *Рульове управління* та в параметрах вимкнути двигун.

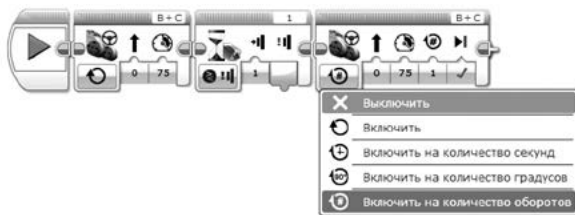


Рис. 82. Додавання блоку *Рульове управління* для вимкнення двигуна

Після цього програму можна скопіювати та завантажити у пам'ять робота.

Сенсор кольору

Цифровий сенсор кольору EV3 здатний визначити 8 різних кольорів. Він також може використовуватися як сенсор освітленості. Завдяки цьому сенсору можна побудувати роботів-сортувальників, що використовують колірні індикатори для сортування. Дає глибоке розуміння технології, широко використовується в переробці відходів, виробництві упаковок і в сільському господарстві.



Рис. 83. Сенсор кольору

Датчик кольору може працювати в трьох різних режимах:

- в режимі *Колір* сенсор може визначити колір піднесеного до нього предмета;

- в режимі *Яскравість відбитого світла* датчик направляє світловий промінь на довколишній предмет і по відбитому пучку визначає яскравість предмета;
- в режимі *Яскравість зовнішнього освітлення* сенсор може визначити наскільки яскраво освітлений простір навколо.

Для створення програми для робота, який має зупинитись при знаходженні чорної лінії:

Після блоку *Початок* потрібно додати блок *Рульове управління* для запуску двигунів.

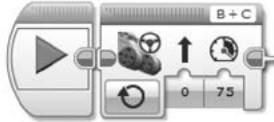


Рис. 84. Додавання блоку *Рульове управління*

Далі – додати блок *Очікування* та змінити режим блока на *Сенсор кольору – Порівняння – Колір*.

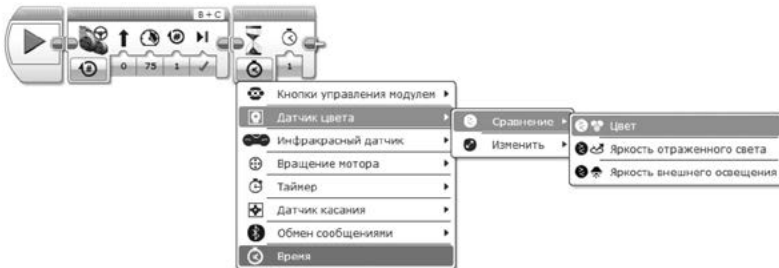


Рис. 85. Додавання блоку *Очікування*

В параметрах сенсора потрібно зазначити колір зупинки – чорний.

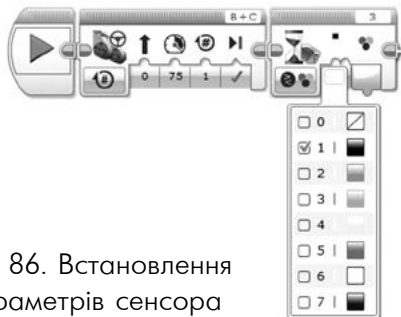


Рис. 86. Встановлення параметрів сенсора

Якщо сенсор спрацьовує, робот має зупинитись, тому потрібно додати знову блок *Рульове управління* та в параметрах вимкнути двигун.

Після цього програму можна скопіювати та завантажити у пам'ять робота.

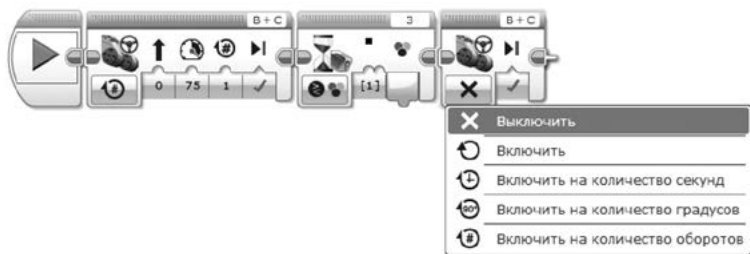


Рис. 87. Додавання блоку *Рульове управління* для вимкнення двигуна

ПРИКЛАДИ ПРОГРАМ

Робот зупиняється після того, як після чорної лінії потрапляє на білу.

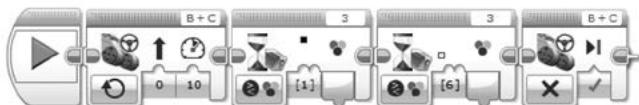


Рис. 88. Програма зупинки робота після чорної лінії

Робот рухається вздовж чорної лінії:

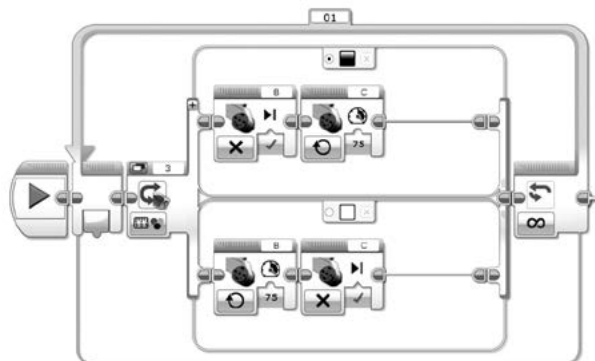


Рис. 89. Програма руху вздовж чорної лінії

Ультразвуковий сенсор

Датчик кольору може працювати в трьох різних режимах: режимі «Яскравість зовнішнього освітлення» датчик може визначити наскільки яскраво освітлено простір навколо. Цифровий ультразвуковий датчик генерує звукові хвилі і фіксує їх відбиття від об'єктів, тим самим вимірюючи відстань до об'єктів. Він також може використовуватися в режимі сонара, випускаючи поодинокі хвилі. Крім того, датчик може вловлювати звукові хвилі, які будуть тригерами для запуску програм. Наприклад, учні можуть використовувати датчик як цифрову лінійку або для побудови системи моніторингу трафіку, вимірювання відстані між автомобілями. Завдяки цьому датчику принципи роботи ультразвукової технології і способи її застосування в автоматичних дверях, машинах і заводських системах стають зрозумілими.



Рис. 90. Ультразвуковий сенсор

Ультразвуковий сенсор може видавати вимірювання відстаней в сантиметрах або в дюймах. Діапазон вимірювань датчика в сантиметрах дорівнює від 0 до 255 см, в дюймах – від 0 до 100 дюймів. Датчик не може виявляти предмети на відстані менше 3 см (1,5 дюймів). Так само він не досить стійко вимірює відстань до м'яких, тканинних і малооб'ємних об'єктів. Крім режимів вимірювання відстані в сантиметрах і дюймах датчик має спеціальний режим «Присутність/слухати». У цьому режимі датчик не випромінює ультразвукові імпульси, але здатний виявляти імпульси іншого ультразвукового датчика.

Для створення програми для робота, який має зупинитись на відстані 30 см від перешкоди.

Примітка: у версії програми, завантаженої з сайту, ультразвуковий сенсор відсутній в списку. Його можна додатково завантажити та підключити. Посилання для завантаження: <https://www.>

lego.com/ru-ru/mindstorms/downloads. Для підключення – Меню *Інструменти/Майстер імпорту блоків*.

Після блоку *Початок* потрібно додати блок *Рульове управління* для запуску двигунів.

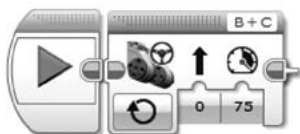


Рис. 91. Додавання блоку *Рульове управління*

Далі потрібно додати блок *Очікування* та змінити режим блока на *Ультразвуковий сенсор – Порівняння – Відстань у сантиметрах*.



Рис. 92. Додавання блоку *Очікування*

В параметрах сенсора потрібно зазначити умову та відстань зупинки – 20 см.

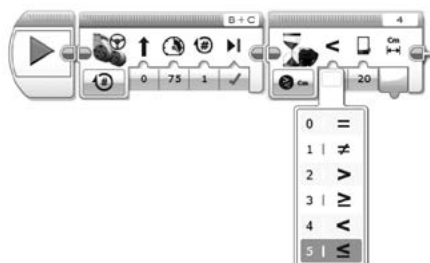


Рис. 93. Зазначення параметрів сенсора

Якщо умова спрацює, то робот має зупинитись, тому потрібно додати знову блок *Рульове управління* та в параметрах вимкнути двигун.

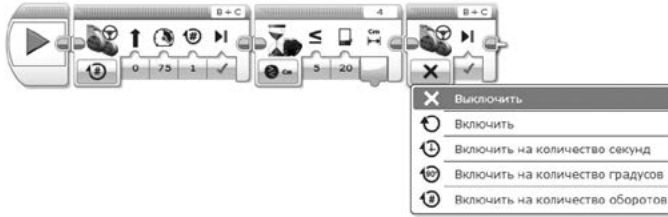


Рис. 94. Додавання блоку *Рульове управління* для вимкнення двигуна

Після цього програму можна скомпілювати та завантажити у пам'ять робота.

Приклади програм

Робот-лінійка – використанням ультразвукового сенсора робот вимірює відстань до об'єктів, значення відображається на екрані.

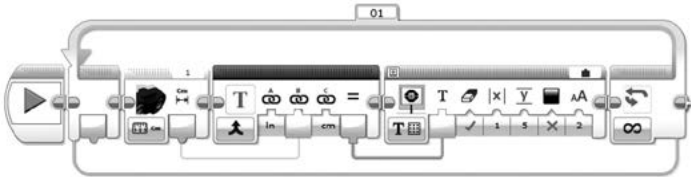


Рис. 95. Програма робота-лінійки

Визначення поблизу іншого робота з включеним ультразвуковим сенсором (режим *Присутність/Слухати*). Якщо поруч є такий робот – світло на головному блоці робота – червоне, немає – зелене.

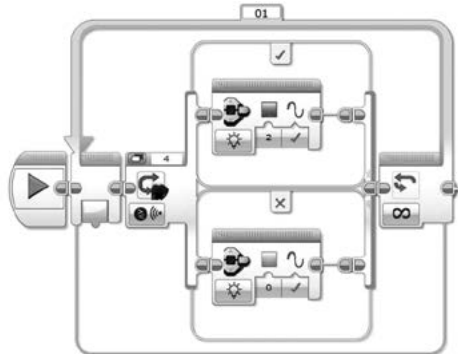


Рис. 96. Програма визначення іншого робота з ультразвуковим сенсором

Сповільнення робота під час наближення до перешкоди.

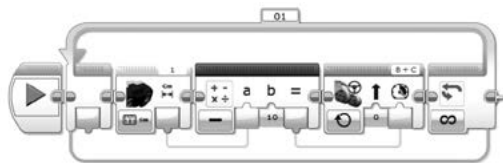


Рис. 97. Програма сповільнення робота

Інші сенсори

Цифровий гіроскопічний сенсор дозволяє вимірювати рух обертання робота, а також вловлювати зміни в його русі і положенні. За допомогою цього сенсора легко можна виміряти кути, створити балансуючого робота і досліджувати технології, які використовуються в справжніх навігаційних системах та ігрових контролерах.



Рис. 98. Гіроскопічний сенсор

Гіроскопічний сенсор призначено для вимірювання кута або швидкості обертання робота. Зверху на корпусі сенсора нанесено дві стрілки, що позначають площину, в якій працює датчик. Тому важливо правильно встановити сенсор на робота. Для більш точного вимірювання кріплення гіроскопічного сенсора має виключати його рухливість щодо корпусу робота. Навіть під час прямолінійного руху робота гіроскопічний сенсор може накопичувати похибку вимірювання кута і швидкості обертання, тому безпосередньо перед вимірюванням слід здійснити скидання в 0 поточного показання сенсора. Обертання робота проти годинникової

стрілки формують негативні значення вимірювань, а обертання за годинниковою стрілкою – позитивні.

Цифровий ІЧ-сенсор здатний визначати наближення робота. Він також здатний відстежити ІЧ-сигнали, що випромінюються ІЧ-маяком, що дозволяє створювати дистанційно керованих роботів, навігаційні системи для подолання перешкод.



Рис. 99. Інфрачервоний сенсор

Інфрачервоний сенсор у своїй роботі використовує світлові хвилі, невидимі людині – інфрачервоні хвилі. Такі ж хвилі використовують, наприклад, дистанційні пульти управління різноманітною сучасною побутовою технікою (телевізорами, відео і музичними пристроями). Інфрачервоний сенсор в режимі *Наближення* самостійно посилає інфрачервоні хвилі і, піймавши відбитий сигнал, визначає наявність перешкоди перед собою. Ще два режими роботи інфрачервоний сенсор реалізує в парі з інфрачервоним маяком. У режимі *Віддалений* інфрачервоний сенсор вміє визначати натискання кнопок інфрачервоного маяка, що дозволяє організувати дистанційне керування роботом. У режимі *Маяк* інфрачервоний сенсор посилає постійні сигнали, за якими інфрачервоний сенсор може визначати зразковий напрям і віддаленість маяка, що дозволяє запрограмувати робота таким чином, щоб він завжди дотримувався руху в сторону інфрачервоного маяка.

ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. Написати програму для робота, що рухається вперед упродовж 5 секунд. Провести випробування, переконатися в працездатності програми і продемонструвати її роботу.
2. Написати програму для робота, що рухається назад упродовж 5 секунд. Провести випробування, переконатися в працездатності програми і продемонструвати її роботу.
3. Написати програму для робота, що рухається вперед на 1 метр. Провести випробування, переконатися в працездатності програми і продемонструвати її роботу.
4. Написати програму для робота, що здійснює поворот на 90° . Подумати, скільки існує можливих варіантів вирішення цієї задачі. Запрограмувати всі. Провести випробування, переконатися в працездатності програми і продемонструвати її роботу.
5. Написати програму, відповідно до якої роботу необхідно повернутися проти годинникової стрілки навколо лівого колеса на 45° . Провести випробування, переконатися в працездатності програми і продемонструвати її роботу.
6. Написати програму для робота, що рухається за квадратною траєкторією. Провести випробування, переконатися в працездатності програми і продемонструвати її роботу.
7. Встановити на рівній поверхні деяку перешкоду (банку, кубик, невелику коробку), відзначити місце старту робота. Створити в проєкті нову програму, що дозволяє роботу об'їхати навколо перешкоди і повернутися до місця старту. Скільки програмних блоків ви використовували?
8. Написати програму для робота, що має проїхати прямолінійно вперед на 4 обороти двигуна, розвернутися, проїхати на 720 градусів.
9. Скласти програму руху робота за довільною складною траєкторією.
10. Написати програму для робота, що може проїхати із заданою потужністю певну відстань вперед, повернутися на

заданий кут за годинниковою стрілкою навколо правого колеса і проїхати ту ж саму відстань назад.

11. Скласти програму робота з двома датчиками дотику, який після зустрічі з перешкодою від'їжджає назад на 20 см, повертає праворуч та продовжує рух.
12. Скласти програму робота з використанням датчика дотику і ультразвукового сенсора, який при появі об'єкта на відстані 1.5 м видає звуковий сигнал, який припиняється при натисканні на датчик дотику. Дії мають повторюватись.
13. Скласти програму робота, який починає їхати повільніше під час перетину жовтої лінії і зупиняється на червоній.

Проект. Сконструювати робота, який буде виконувати основні функції деяких професії, зокрема поліцейського, охоронця тощо з використанням датчика або сенсорів.

4 МІНІ-КОМП'ЮТЕР RASPBERRY PI



4.1. ЗНАЙОМСТВО З RASPBERRY PI

Ще одним з прикладів навчального конструктора є міні-комп'ютер Raspberry Pi, який з'явився декілька років тому, але вже зміг завоювати прихильність всіх користувачів.

Raspberry Pi – невеликого розміру одноплатний комп'ютер, розроблений британським фондом Raspberry Pi Foundation. Raspberry Pi створена на базі процесора з архітектурою ARM і частотою від 700 Мгц. В останніх версіях прошивки офіційно дозволили прискорити процесор до 1,2 Гц, що забезпечує досягнення прийнятної продуктивності при низькому енергоспоживанні. Raspberry Pi – це повноцінний системний блок, за допомогою якого можна навчати роботі з комп'ютером, відтворювати відео, програмувати, користуватися Інтернетом, слухати музику. Одна з головних і привабливих особливостей Raspberry Pi – наявність на платі апаратних портів введення/виведення GPIO (General Purpose Input/Output, входи/виходи загального призначення), що відкриває перспективи використання його в робототехнічних проектах [38].



Рис. 100. *Raspberry Pi* модель B

Перший прототип Raspberry Pi з'явився в 2006 р. Уже тоді він призначався для навчання школярів азам комп'ютерних наук (в англomовній літературі прийнято називати Computer Science), а у нас – інформатики. Засновникам проекту – працівникам і викладачам Комп'ютерної лабораторії Кембриджського університету – не подобався той факт, що з кожним роком рівень підготовки абітурієнтів неухильно знижувався. Якщо в 90-і роки до них, як правило, приходили вчитися молоді люди, які розумілися на програмуванні, то у 2000-х типовий абітурієнт був лише трохи знайомий з веб-дизайном.

Розробників Raspberry Pi можна зрозуміти, адже їхня молодість припала якраз на часи появи і розквіту перших посправжньому домашніх ПК Amiga, BBC Micro, Spectrum ZX і Commodore 64. Користувачі цих ПК деколи просто змушені були займатися програмуванням, якщо не знаходили відповідного програмного забезпечення для своїх завдань. Причому нерідко потрібно було не просто вміти писати код, але і добре розуміти принципи роботи апаратної складової, вміло обходити наявні обмеження і різними хитрощами домагатися максимальної продуктивності. Найбільш досвідчені брали в руки паяльник, адже периферії спочатку теж було небагато. А для деяких це і зовсім стало приводом відкрити власне підприємство.

Втім в 90-х була випущена платформа Wintel, яка стала фактично монополістом на ринку ПК і в результаті позбавила користувачів від необхідності освоювати програмування. У школах теж вирішили переключитися на вивчення основ роботи з одним відомим офісним пакетом і створення простеньких HTML-сторінок. Потім стали масово поширюватися ігрові приставки і персональні комп'ютери. Загалом життя простих користувачів помітно спростилося, а ряди ентузіастів помітно порідшали. Така ситуація не влаштовувала викладачів – і вони загорілися ідеєю створення платформи, яка відродила б інтерес до самостійного вивчення цієї теми. Творці надихалися історією колись легендарного навчального комп'ютера BBC Micro, втім не розраховували докорінно змінити ситуацію з інтересом до обчислювальної техніки.

Завдання для Raspberry Pi та BBC Micro було однаковим: зацікавити підростаюче покоління комп'ютерними технологіями на досить високому рівні. BBC Micro планувалося продати не більше 12 тисяч штук, а в результаті майже за 10 років було продано близько півтора мільйони. Пробна партія Raspberry Pi в десятитисячному обсязі примірників розійшлася за кілька хвилин, причому спочатку діяло правило «одна штука в одні руки». Представник одного з двох офіційних дистриб'юторів попросив користувачів перестати оновлювати сторінку онлайн-магазину, оскільки сервери просто не справлялися з навантаженням. Рік по тому, під час старту продажів в США, історія повторилася. На поточний момент продано понад два мільйона пристроїв, і це, схоже, не межа [1].

У 2009 р. була створена благодійна організація Raspberry Pi Foundation, в завдання якої входить розробка і просування одноименого комп'ютера. Два роки знадобилося на створення апаратної і програмної частин майбутнього пристрою, укладення договорів і дотримання інших формальностей. В певний момент з'явилась ідея зробити міні-ПК у вигляді великої флешки – з одного боку – USB-порт, а з іншого – HDMI-вихід. Подібні пристрої з Android на борту зараз у величезних кількостях випускають китайські компанії. Нарешті, в 2011 році з'явилися перші альфа- і бета-версії плат. Варто врахувати, що Raspberry Pi – некомерційний проект, саме тому на його розробку пішло стільки років.

Технічні характеристики і можливості Raspberry Pi

Raspberry Pi називають одноплатним комп'ютером розміром з кредитну карту. Насправді сама плата трохи крупніше – 85,6x56x21 мм – і не має округлених країв, до того ж деякі порти просто стирчать зовні. Важить пристрій всього 54 грами.

Raspberry Pi доступний у вигляді двох різних моделей, відомих як модель А і модель В. Відмінності полягають у можливості зниження вартості моделі А завдяки обмеження деяких функціональних можливостей. У таблиці представлені технічні можливості та характеристики перших моделей Raspberry Pi А та В [43].

| Одноплатний міні-ПК Raspberry Pi | | |
|---|---|--|
| Характеристика | Модель А | Модель В |
| Ціна, долари США | 35 | 50 |
| System-on-a-Chip | Broadcom BCM 2835 (CPU + GPU) | |
| CPU | 700 МГц ARM 11 (ядро ARM1176JZF-S), можлива швидкість до 1 ГГц | |
| GPU | Broadcom VideoCore IV | |
| Стандарти | Open GL ES 1.1/2.0, Open VG 1.1, Open EGL, Open MAX | |
| Апаратні кодеки | H. 264 (1080 p 30, high-profile) MPEG – 2 u VC - 1 (ліцензія продається окремо) | |
| Пам'ять | 256 Мбайт | 512 Мбайт |
| Порти | 1 | 2 |
| Відеовихід | 1×HDMI 1.3a (CEC) 1×RCA (576i / 480i, PAL – BGHD / M / N, NTSC, NTSC-J) | |
| Аудіовихід | Гніздо 3,5 мм HDMI | |
| Кардрідер | SD / MMC / SIDIO | |
| Мережа | - | Ethernet – nortn RS 45, 10/100 Мбіт/с |
| Інтерфейс | 20×GPIO (SPI, 12 C, UART, TTL) MIPI CSI- 2, MIPI, DSI) | |
| Енерговикористання | 500 мА, (2,5 Вт) | 700 мА, (3,5 Вт) |
| Живлення | 5 В через порт micro – USB або GPIO | |
| Розмір | 85,6×56×21 | |
| Маса | 54 | |

В лютому 2016 року компанія Raspberry Pi в черговий раз перевершила саму себе, випустивши нову модель Raspberry Pi 3. У цьому новітньому Raspberry Pi 3 є два гігантських оновлення. Пер-

ше – це наступне покоління чотириядерного процесора Broadcom BCM2837 64-бітної ARMv8, яке збільшує швидкість процесора від 900 МГц на Raspberry Pi 2 до 1,2 ГГц на Raspberry Pi 3. Другим гігантським оновленням є чіп BCM43143 WiFi, вбудований у Raspberry Pi. Тут уже не потрібні ніякі WiFi-адаптери, цей Raspberry Pi 3 вже готовий до використання Wi-Fi. На платі також є Bluetooth Low Energy (BLE), роблячи Raspberry Pi 3 IoT гарним рішенням (BLE підтримка також реалізована). І нарешті, в цій моделі є модернізоване джерело живлення, що перемикається, та який зріс до 2,5 ампер замість 2 ампер, що дозволяє підключати до Вашого Raspberry Pi навіть більш потужні пристрої через USB-порти.

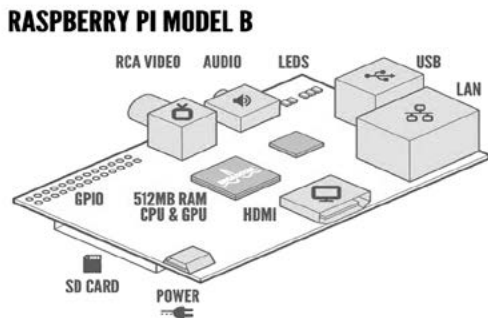


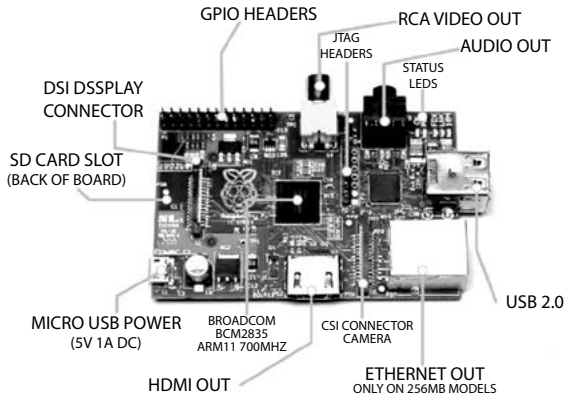
Рис. 101. Схема Raspberry Pi, модель B

Порівняння технічних можливостей моделей Raspberry Pi B

| Технічні характеристики | Raspberry Pi 3 Model B | Raspberry Pi 2 Model B | Raspberry Pi Model B+ | Raspberry Pi Model B |
|-------------------------|---|---|----------------------------------|----------------------------------|
| Процесор | Broadcom 2837 quad-core ARM Cortex-A53 64bit (1,2GHz) | Broadcom BCM2836 quad-core ARMv7 (900MHz) | Broadcom BCM2835, ARMv6 (700MHz) | Broadcom BCM2835, ARMv6 (700MHz) |
| Оперативна пам'ять | 1Gb | 1Gb | 512Mb | 512Mb |

| Відеовихід | HDMI | HDMI | HDMI | HDMI |
|---------------------------|--|----------------------------------|----------------------------------|--|
| A / V вихід | A/V вихід 3.5мм jack 4 pin | A/V вихід 3.5мм jack 4 pin | A/V вихід 3.5мм jack 4 pin | Аудіовихід 3,5мм stereo jack; Композит- ний відеовихід RCA |
| USB-порти | USB 2.0 x 4 | USB 2.0 x 4 | USB 2.0 x 4 | USB 2.0 x 2 |
| Мережа | WiFi 802.11n, 10/100Mb RJ45 Ethernet | 10/100Mb RJ45 Ethernet | 10/100Mb RJ45 Ethernet | 10/100Mb RJ45 Ethernet |
| Bluetooth | Bluetooth 4.1, Bluetooth Low Energy | - | - | - |
| Слот для карти пам'яті | Micro SD | Micro SD | Micro SD | SD |
| GPIO | 40 | 40 | 40 | 26 |

Починаючи з 2013 р. всі Raspberry Pi Model B повинні мати півгібайта RAM, але на складах перекупників могли завалятися більш ранні моделі. Ліцензією на виробництво плат мають компанії Premier Farnell, RS Components і Egoman. Причому остання випускає плати червоного кольору, які можуть пропонуватися тільки на китайських територіях. До першої річниці проекту RS Components випустили ювілейну партію плат синього кольору обсягом 1000 штук. Ці ж компанії мають право продавати Raspberry Pi, а в США поширенням займається Allied Electronics. Тож всі інші магазини просто закуповують великі партії пристроїв у цієї четвірки і перепродують кінцевим споживачам. Обидві моделі плат від різних виробників (складанням займаються заводи Sony, Qisda і Egoman) мають деякі несуттєві відмінності, але зрештою вони ідентичні.

Рис. 102. *Raspberry Pi 2 Model B*

Склад плати Raspberry Pi 3 model B

Процесор: 64-бітний чотириядерний процесор на базі Cortex-A53.

Рис. 103. *Плата Raspberry Pi*

Роз'єм загального призначення введення/виведення (GPIO) – порт для отримання електричних сигналів з Raspberry Pi, таких як датчики для зчитування і управління двигунами. Він складається з двох паралельних рядів штифтів і позначений як P1. Різні моделі Raspberry Pi використовують по-різному через спосіб маршрутизації шрифтів на платі.

Композитний відеовихід: цей роз'єм використовується для підключення Raspberry Pi до композитного відеоз'єднання (стандартне телебачення) за допомогою кабелю RCA.

Аудіовихід – чорний 3,5 мм роз'єм на верхньому правому куті плати.

Вихід HDMI – порт використовується для передачі цифрового відео на монітор комп'ютера. Вихід HDMI також може скерувати аудіо, так що можна не використовувати порт аудіовиходу.

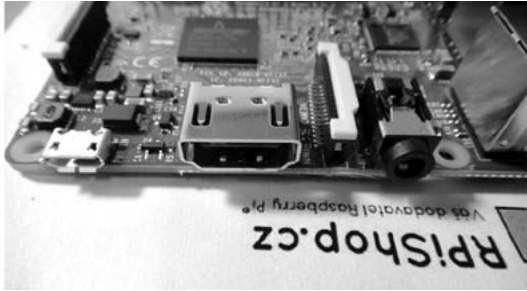


Рис. 104. Вихід HDMI

USB порт(и) – дозволяють підключати USB-аксесуари (такі як клавіатура і миша і зовнішні пристроїв зберігання даних) для системної плати. Модель А має тільки один USB-порт для зниження витрат. Модель В має два порти USB.

Порт Ethernet (тільки для моделі В) – призначений для підключення Raspberry Pi до Ethernet мережі і для доступу в Інтернет.



Рис. 105. Порти

Оперативна пам'ять – об'єм ОЗП не змінився і залишився таким, як і на попередніх моделях – 1 ГБ.



Рис. 106. Оперативна пам'ять

Роз'єм для камери послідовного інтерфейсу (CSI) – тонкий роз'єм чорного кольору між роз'ємом Ethernet і виходом HDMI для підключення невеликої камери, наприклад, веб-камери. CSI роз'єми можна придбати в магазині Raspberry Pi.

Слот microSD

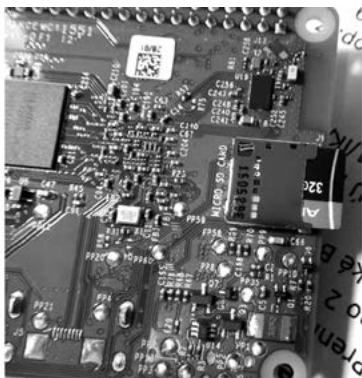
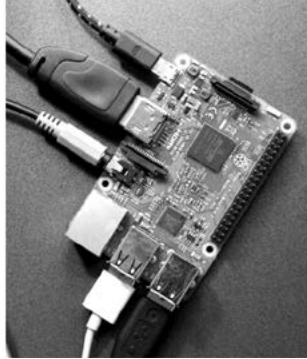


Рис. 107. Слот microSD

Живлення: на лівій нижній стороні – мікророз'єм живлення USB. Живлення забезпечується через мікроджерело живлення USB, яке підключається до цього порту.

Роз'єм дисплею послідовного інтерфейсу (DSI) являє собою тонкий роз'єм для підключення високошвидкісних дисплеїв. Він використовується для підключення невеликої ЖК-панелі безпосередньо до Raspberry Pi. Можна використовувати його як базу для сенсорного введення даних.

Рис. 108. *Raspberry Pi 3 Model B*

Включення Raspberry Pi та вхід у систему

Пристрої необхідні для роботи з Raspberry Pi:

■ Джерело живлення

Живлення плати, як і раніше, здійснюється через роз'єм мікро-USB. Типове значення сили струму становить 800 мА, але може бути підвищено до 1200 мА при використанні більш потужного адаптера (рекомендується з маркуванням 2А). Споживана потужність самої плати не перевищує шести ват і зазвичай утримується в діапазоні 3-4 Вт, що дозволяє використовувати тільки пасивне охолодження. В новій моделі є модернізований перемикач джерела живлення, який виріс до 2,5 ампер замість 2 ампер – це дозволяє підключати до Raspberry Pi навіть більш потужні пристрої через USB-порти.

Рис. 109. *Блок живлення*

■ Карта пам'яті

Операційна система і файли Raspbian зберігаються на SD-карті. Raspberry Pi моделі A і B використовують карти пам'яті SD стандартного розміру, в той час як Pi 2, Pi 3 використовують менші карти microSD. Необхідно не менше 4 ГБ простору (переважно 8 ГБ або більше). Pi Raspberry Foundation продає дуже доступні 8 Гб SD карти з попередньо встановленою операційною системою, і їх можна знайти на сайті <http://swag.raspberrypi.org/>. Якщо ви починаєте з порожньою картою SD, то в цьому посібнику буде описано як правильно встановити ОС.



Рис. 110. Карти пам'яті

■ Пристрої введення-виведення

Для входів будемо використовувати клавіатуру USB і мишу.



Рис. 111. Пристрої введення-виведення

■ Відео

Використовується монітор або телевізор з HDMI або DVI входом, а також відео-кабель, підключений від порту HDMI Raspberry Pi до входу на екрані, як показано на рисунках. Також є можливість приєднати через VGA або композитний роз'єм.



Рис. 112. Монітор, HDMI кабель

■ Мережа

Підключення до Інтернету є дуже корисним, оскільки це дозволяє робити більше роботи безпосередньо на Raspberry Pi. Найпростіший підхід полягає у використанні провідного з'єднання Ethernet. Але в новій моделі є важливе оновлення: додавання чіпа BCM43143 WiFi. Вже не потрібні WiFi-адаптери, Raspberry Pi 3 вже готовий до використання Wi-Fi.

На наступному малюнку показаний роз'єм Ethernet, який використовується для підключення Pi до мережі Інтернет:



Рис. 113. RJ-45 Ethernet роз'єм

4.2. ЗАПУСК RASPBERRY PI

Тепер можна підготувати плату Raspberry Pi, щоб запустити її перший раз. Можна запустити Raspberry Pi, виконавши такі дії:



Рис. 114. Пристрої для підключення Raspberry Pi

I. Вставлення SD-карти у відповідний роз'єм

Карта пам'яті повинна бути із заздалегідь підготовленою операційною системою. Можна купити SD-карту з встановленим програмним забезпеченням, а можна підготувати карту самостійно.

Інструкція з підготовки карти пам'яті

Перед записом SD-карти з операційною системою для завантаження Raspberry Pi необхідно завантажити образ операційної системи. Останні образи можна завантажити за адресою <http://www.raspberrypi.org/downloads>. Існує кілька різних дистрибутивів системи Linux для Raspberry Pi. Найпоширенішим є Raspbian – адаптований варіант дистрибутива Debian Linux. Крім офіційних образів існують адаптовані іншими ентузіастами варіанти Raspbian, з ними можна ознайомитися на сайті системи <http://www.raspbian.org/RaspbianImages>.

Установка отриманого образу на SD-карту проводиться різними способами, в залежності від операційної системи ПК.

Запис образу SD-карти в системі Windows:

1. Завантажте образ SD-карти як описано вище. Файл образу є архів ZIP.

2. Розпакуйте отриманий архів. В середині знаходиться файл з розширенням IMG. Це і є образ SD-карти.
3. Вставте карту в картридер і підключіть його до комп'ютера. Відкрийте програму Windows Explorer і подивіться, яку букву диска присвоєно вашій карті. Крім того, якщо на карті пам'яті вже щось було записано, то краще буде її відформатувати, щоб уникнути проблем на наступних кроках.
4. Завантажте програму Win32DiskImager (<https://sourceforge.net/projects/win32diskimager/>).
5. Розпакуйте отриманий архів і запустіть програму. Краще буде запустити її з правами адміністратора.



Рис. 115. Програма Win32DiskImager

6. Вкажіть програмі, де лежить розпакований на 2-му кроці образ SD-карти.
7. Вкажіть також букву диска SD-карти з 3-го кроку. Дуже важливо тут не помилитися, оскільки при неправильно вказаній початковій літері можна втратити дані на жорсткому диску комп'ютера!
8. Натисніть кнопку *Записати* і почекайте до закінчення процесу запису.
9. Закрийте Win32DiskImager і відключіть картридер за допомогою *Безпечного витягання пристрою*.
10. Вставте карту в Raspberry Pi, увімкніть його і він завантажиться! При першому запуску Raspbian відкриє *Меню налаштувань*, за допомогою якого можна буде розширити розділ на SD-карти до максимально можливого, якщо карта має обсяг понад 4 Гб.

Після запису образу в операційній системі Windows SD-карта буде відображатися з об'ємом всього 75 Мб. Це пов'язано з тим, що розділ карти, що містить Raspbian, не відображається в Windows.

II. Підключення клавіатури і миші до Raspberry Pi

Необхідно підключити ці пристрої до двох доступних USB портів Raspberry Pi.

III. Підключення зовнішнього VGA дисплея

Якщо використовується зовнішній дисплей з VGA інтерфейсом, необхідно спочатку підключити VGA кабель дисплея до перехідника HDMI/VGA, а потім приєднати цей перехідник до HDMI порту Raspberry Pi.

IV. Підключення живлення

Для підключення живлення до Raspberry Pi досить приєднати кабель блоку живлення до мікро USB-роз'єму, а потім включити блок живлення в мережу.

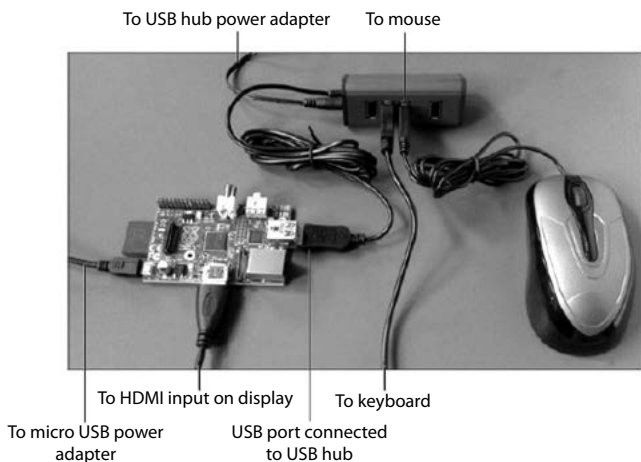


Рис. 116. Підключені пристрої до підключення Raspberry Pi

V. Вхід у систему

Ви повинні бачити вибір операційних систем з коротким описом. Оберіть необхідну і натисніть на кнопку *Встановити*. Завжди можна повернутися та обрати іншу операційну систему пізніше.

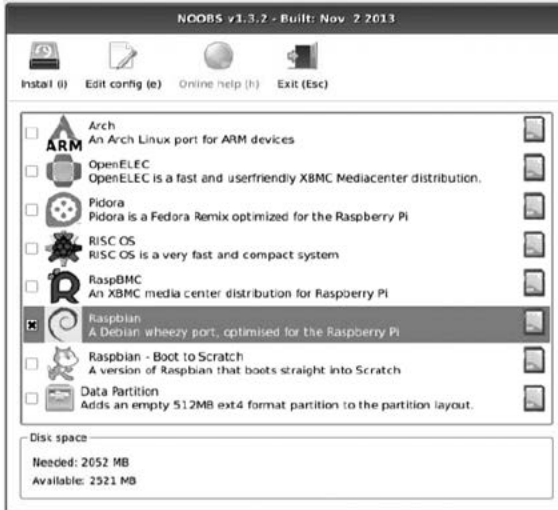


Рис. 117. Вибір ОС

Зачекайте! Операційні системи досить великі, тому встановлення займе декілька хвилин.

Наступне зображення показує процес установки Raspbian:



Рис. 118. Встановлення Raspbian

Коли установка завершена, відобразатиметься синій екран зі списком опцій налаштувань програмного забезпечення Raspberry Pi. Більшість функцій користувач змінює під власні потреби. Зупинимось на двох корисних змінах: за допомогою клавіш зі стрілками перейдіть та оберіть *Enable Boot to Desktop/Scratch* та натисніть *Enter*. Виберіть опцію *Desktop Log* і натисніть *Enter*. Тепер ви повинні повернутися в головне меню. Потім необхідно вибрати *Internationalisation Options*, потрібну мову і розкладку клавіатури. За допомогою клавіші зі стрілкою вправо слід перейти до опції *Finish* і натиснути *Enter*.

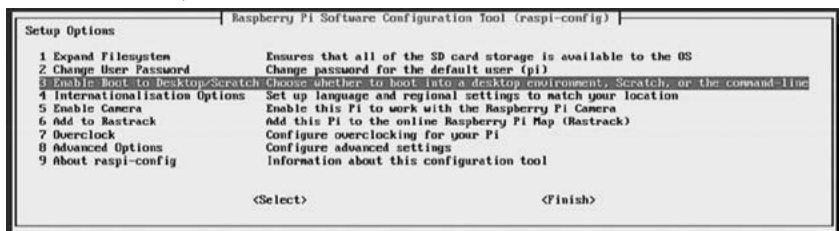


Рис. 119. Опції налаштування

Приблизно через хвилину Raspberry Pi повинен закінчити перезавантаження, і можна побачити робочий стіл Raspberry Pi, як показано на наступному малюнку.

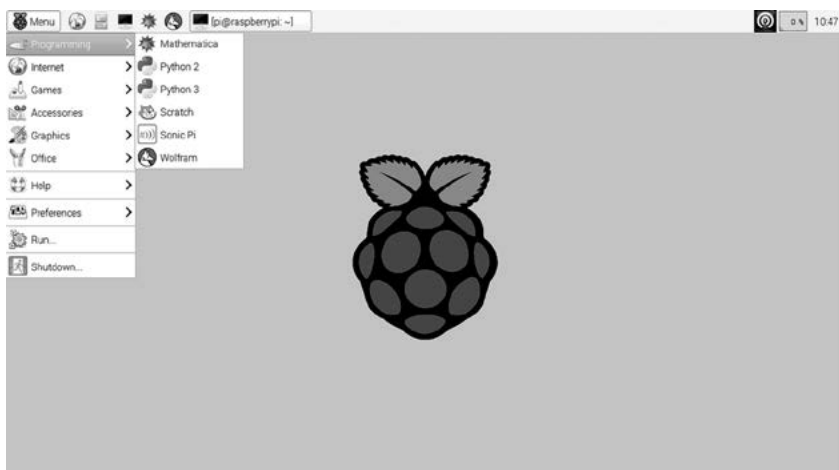


Рис. 120. ОС Raspbian

4.3. SCRATCH ДЛЯ RASPBERRY PI

Можна використовувати Raspberry Pi для вивчення основних концепцій мов програмування, написання команд для виконавців, створення першої анімації в середовищі програмування. У цьому доцільно скористатися Scratch. Це інтерпретована динамічна візуальна мова програмування, основана і реалізована на Squeak. Завдяки динамічності вона дає змогу змінювати код навіть під час виконання. Мова має за мету навчити дітей поняттю програмування і дає змогу створювати ігри, анімації чи музику. Scratch – середовище програмування, яке дозволяє дітям створювати власні анімовані та інтерактивні історії, ігри та інші витвори. Ними можна обмінюватися всередині міжнародної спільноти, яка поступово формується в мережі Інтернет [36].

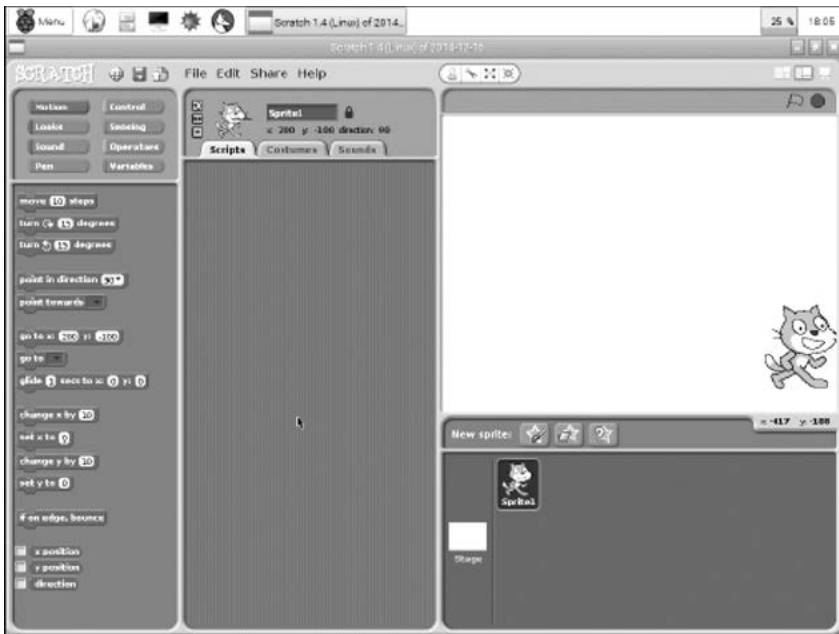


Рис. 121. Scratch

Причини вибору для вивчення Scratch:

1. Кросплатформність – Scratch можна встановлювати на комп'ютерах під управління різних операційних систем Microsoft Windows, Macintosh і Linux, Raspbian та інших.

2. Алгоритмічна повнота – Scratch підтримує концепції об'єкно-орієнтованого програмування, а саме:

- структуру слідування або послідовні процеси;
- структуру повторення або циклічні процеси;
- структуру розгалуження (вибору) або розгалуджені процеси;
- змінні глобальні й локальні, надання і зміна величин змінних;
- типи даних: символічні, числові, логічні, графічні, аудіо;
- вирази (числові, текстові, логічні, порівняння), операції, функції, оператори, операнди;
- введення й виведення даних;
- координування, синхронізування роботи окремих частин програми;
- паралельні процеси – одночасне виконання різних програмних блоків;
- об'єкти (спрайти), властивості об'єктів, методи, події.

3. Наочність створення (запису) алгоритму – відкривши потрібну сукупність вказівок, можна перетягнути потрібну вказівку за допомогою миші у поле скриптів.

4. Наявність інтерфейсів різними мовами – кнопка *Встановити мову* на *Панелі інструментів* відкриває перелік мов інтерфейсу, серед яких є українська.

5. Безоплатність – середовище програмування можна безкоштовно завантажити і вільно використовувати у шкільній чи позашкільній освіті. Посилання для завантаження – офіційний сайт розробників <http://scratch.mit.edu/>. Версія Scratch 1.4 попередньо встановлена на Raspbian ОС.

Елементи інтерфейсу й основні поняття

Scratch – інтерактивне середовище, побудоване на інтуїтивно зрозумілих дитині принципах. Принципово новою є ідея щодо складання програми мишкою з готових блоків-цеглин подібно до того, як діти будують будиночки і машинки з деталей конструктора. Подібний спосіб складання програм унеможливорює проблему синтаксису, що для молодших школярів є важливим.

Головне програмне вікно середовища поділено на декілька частин, згрупованих у три стовпчики:

- лівий стовпчик містить палітру блоків;
- центральний стовпчик містить піктограму активного спрайту з координатами його розташування на сцені та трьома вкладками: *Скрипти*, *Образи*, *Звуки*;
- правий стовпчик містить *Сцену* і *Список спрайтів*.

Спрайт – це об'єкт *Scratch*, що пов'язаний із зображенням, набором змінних і скриптів, які визначають його поведінку.

За замовчуванням використовують спеціального виконавця вказівок – Рудого kota.



Рис. 122. Один зі спрайтів у *Scratch*

Він може рухатися, говорити, змінювати зовнішній вигляд, взаємодіяти з іншими виконавцями на сцені. Інших виконавців можна долучати:

- з бібліотеки *Scratch*;
- з мережі (локальної чи глобальної);
- як об'єкти, створені в інших графічних програмах (наприклад, GIMP);
- як об'єкти, створені у графічному редакторі, вбудованому у *Scratch*.

Скрипт (script, сценарій, метод) – послідовність вказівок, що визначає, які дії і в якому порядку потрібно виконати певному об'єкту (спрайту).

Скрипти створюють методом сполученням окремих блоків: або послідовно, або розташовуючи блок у визначеному місці іншого блоку (структури, що управляє, функції і т.ін.). Один спрайт може мати декілька скриптів, які запускають незалежно дію користувача (натисненням клавіші або кнопки миші), таймером або отриманням повідомлення від іншого спрайту. Скрипт складається зі стеків.

Стек (stack) – набір послідовно сполучених різнокольорових графічних блоків у межах однієї події.

Блок – це мінімальний фрагмент програми у *Scratch*: змінна, оператор, функція або структура, що керує. Блоки згруповано у 8 категорій (див. далі ілюстрації).

Подія – натискання на задану клавішу (наприклад, пропуск), клацання по виконавцю або по сцені, надходження повідомлення від іншого виконавця і т.ін.

Образи (вигляд спрайту) – сукупність зображень одного й того ж об'єкту (спрайту), кожне з яких дещо відрізняється від попередніх.

Звуки – приєднані звукові ефекти й музика.

Сцена – область, в якій діє об'єкт (спрайт) при виконанні програми.

На малюнку нижче подано вигляд вікна програми для категорії *Рух*.

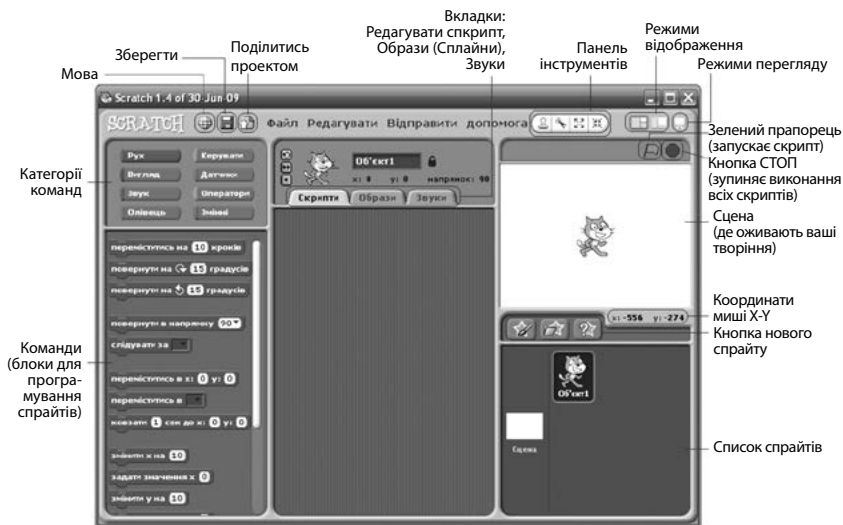


Рис. 123. Інтерфейс програми *Scratch*


Приклади простих скриптів

Вправа 1. Рух спрайта

Завдання. Складіть програму, за якою на сцені кіт буде рухатися. Додайте команди, за якими спрайт повинен рухатись по сцені з одного краю до другого, слідувати за мишкою та змінювати образ.

Виконання

1. Завантажте середовище *Scratch*.

Після завантаження програми чи створення нового проекту, на сцені є єдиний спрайт . Відповідно він виділений у списку спрайтів і все, що буде поміщено на вкладці **Скрипти**, належатиме до цього спрайта.

Увага! Часто користувачі обирають сцену (праворуч знизу) і не розуміють, що в цьому випадку на вкладці *Скрипти* можуть виконуватись команди лише пов'язані зі сценою.

2. Оберіть категорію команд **Рух**.

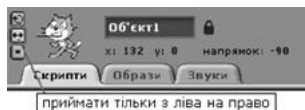
Серед команд знайдіть **переміститись на 10 кроків** та перетягніть на вкладку **Скрипти**. Скрипт поки складається з однієї команди. Щоб спрайт почав виконувати цю команду, достатньо по ній клікнути мишею. Якщо багато разів клікнути, то можна помітити, що спрайт перемістився до краю сцени і далі нічого не виконує.

3. Додайте до скрипта ще одну команду **якщо границя, відбити**. Команда під час перетягування повинна прикріпитись по попередньої. Скрипт має виглядати так:

4. Спрайт вже відбивається від границі сцени, але чомусь

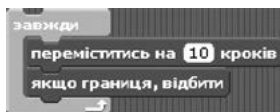


догори ногами. Щоб виправити це, достатньо в налаштуваннях спрайту вказати, що його потрібно відбивати вертикально, а не перевертати. Над скриптом скористайтесь відповідною кнопкою.



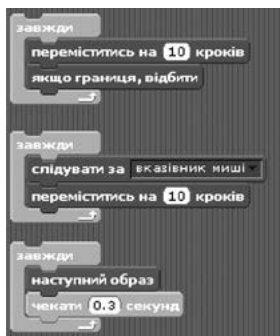
5. Скрипт змушує спрайт рухатись, але хочеться, щоб цей рух відбувався автоматично. Для цього змусимо спрайт виконувати ці

дві команди в циклі, постійно. З категорії **Керувати** слід обрати команду *Завжди*. Необхідно її так перетягнути, щоб попередні команди були у середині циклу. Скрипт повинен виглядати так:



Під час виконання скрипта, спрайт повинен рухатись по сцені з одного краю до іншого.

6. Створіть ще два скрипти, вказані нижче. І подивіться, що вони виконують. Розміщувати їх можна в цьому ж вікні.



Другий скрипт змушує спрайт рухатись за курсором миші. Третій змінює вигляд спрайту за допомогою образів, що містяться у відповідній вкладці.

Вправа 2. Намалюй квадрат

Завдання. Напишіть програму, в якій кіт намалює квадрат розміром 100 x 100 кроків.

Виконання


1. Завантажте середовище *Scratch*.
2. Використовуйте вказівки з категорій керування, руху й малювання.

Потрібні блоки слід перетягнути в область *Скрипти*. Для запуску скрипта – підведіть до блоку *Коли натиснуто Прапорець* вказівник миші і двічі клацніть лівою кнопкою.



Рис. 124. Написання програми в Scratch

3. Складіть алгоритм зображення квадрата:

- запустить скрипт натисканням зеленого прапорця ;
- перемістіть спрайт на позицію $x = -50$, $y = 50$ на сцені;
- поверніть спрайт праворуч;
- очистіть всі сліди олівця та штампи зі сцени;
- встановіть червоний колір олівця;
- встановіть розмір (товщину) олівця 3;
- опустіть олівець, щоб спрайт креслив під час переміщення;
- перемістіть спрайт вперед на 100 кроків;
- поверніть спрайт за рухом годинникової стрілки на 90° ;
- перемістіть спрайт вперед на 100 кроків;
- поверніть спрайт за рухом годинникової стрілки на 90° ;
- перемістіть спрайт вперед на 100 кроків;
- поверніть спрайт за рухом годинникової стрілки на 90° ;
- перемістіть спрайт вперед на 100 кроків;
- поверніть спрайт за рухом годинникової стрілки на 90° .


4. Порівняйте створену програму з поданою нижче:



Рис. 125. Програма «Намалюй квадрат»

Подумайте! Який додатковий блок можна використати, щоб скоротити набір команд програми? Чи відома вам алгоритмічна конструкція *Цикл*? Який блок відповідає за повторення команд?

5. Запустіть на виконання проект, клацнувши на зеленому прапорці у верхньому правому куті вікна середовища.

Результат виконання такого лінійного алгоритму після натискання кнопки  буде мати вигляд.

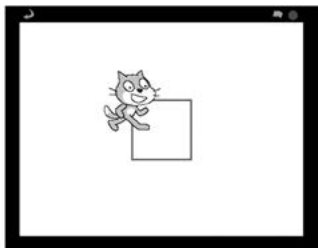


Рис. 126. Результат виконання програми

Вправа 3. Анімація

Завдання. Напишіть програму, в якій будуть взаємодіяти 2 спрайти.

Рекомендації до виконання


1. Для створення нових спрайтів можна скористатися трьома зоряними кнопками.  Список всіх спрайтів проекту розміщений під сценою (рис. 127).



Рис. 127. Сцена і список спрайтів

У програмі є бібліотека готових спрайтів: *Animals* – тварини; *Fantasy* – казкові персонажі; *Letters* – букви і цифри; *People* – люди; *Things* – речі; *Transportation* – транспорт.

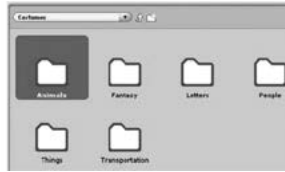


Рис. 128. Бібліотека готових спрайтів

2. Щоб додати новий спрайт з бібліотеки, натисніть на середню кнопку під сценою, яка виглядає як папка.

Спробуйте перейменувати спрайти. Ім'я можна змінити на більш зрозуміле у спеціальному вікні. Щоб видалити спрайт, скористайтеся інструментом *ножиці* над сценою або клацніть по спрайту правою кнопкою миші для вибору команди видалення.



Рис. 129. Способи видалення спрайтів

3. Для повного розуміння та використання можливостей руху спрайтів потрібно ознайомитись із координатами сцени та напрямками руху.

Розмір сцени встановлено виробниками програми *Скретч* і змінити його не можна. Для розуміння координат скористались декартовою системою координат.

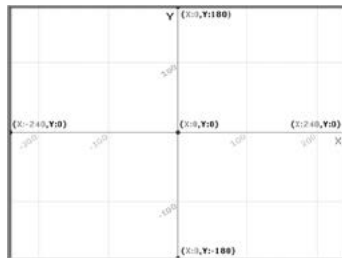
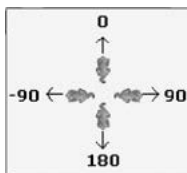


Рис. 130. Координати сцени

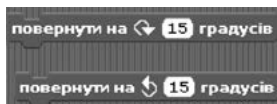
По ширині сцена містить 480 точок (від -240 до 240). По висоті – 360 (-180... 180). По центру сцени знаходиться точка з координатами (0,0).

4. Для переміщення спрайта у відповідні координати користуються командою `идти в x: 0 y: 0`.

Також потрібно розуміти поняття напрямку (в градусах), що дозволяє повернути спрайт.



5. Для встановлення напрямку спрайта використовують команду `повернути в напрямку 90`. Можна змінити кількість градусів у цій команді. Якщо ж потрібно змінити напрямок на декілька градусів, користуються командами:





Вправа 4. Діалог

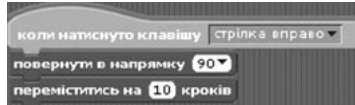
Завдання. Напишіть програму, в якій будуть взаємодіяти два спрайти – котик і песик. Спрайти рухаються назустріч одне одному та ведуть діалог між собою.

Рекомендації до виконання

Для початку розглянемо поняття «подія». **Подія** – зміна властивостей об'єкта, взаємодія між об'єктами, утворення нового об'єкта або знищення існуючого об'єкта. Кожна подія містить оцінку часу, що вказує, коли воно відбувається, і місця, де вона відбувається. Але якщо розглядати події з точки зору комп'ютера, то можна здогадатись, що подією комп'ютер буде вважати всі зовнішні чинники, які він зможе зафіксувати. Наприклад: вставлення в пристрій карти пам'яті, натискання клавіші на клавіатурі, клацання мишею по об'єкту (спрайту чи сцені) тощо. Коли ви граєте в ігри на комп'ютері, то помічаєте, як гра відповідає під час клацан-

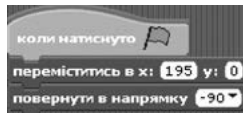
ня мишею чи натисканням клавіш. Саме такі події розглянемо та змусимо спрайт відповідати на них рухами чи іншими діями.

Скористайтесь командою  з категорії . Дана команда змусить виконувати наступні за нею команди, якщо натиснута відповідна клавіша. Змініть клавішу на «стрілка вправо» та додайте ще кілька відомих команд.



Окрім перехоплення подій клавіатури можна в потрібному місті оголосити свою подію. Це дозволить заставити інший спрайт рухатись чи сцену змінюватись у відповідний момент часу.

1. Створіть новий проект. Завантажте новий спрайт, песика. Для початку встановіть дійових осіб історії в потрібні місця. Коли натиснуто прапорець, переміщуємо песика у координати 195,0 та повертаємо у напрямку вліво (-90). Якщо песик відображається догори лапами, пригадайте, що змінюють кнопки властивостей спрайту над вкладкою *Скрипти*.



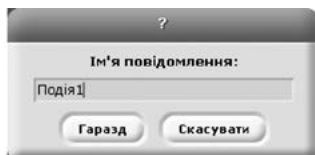
2. Для котика слід створити подібний блок. Обов'язково слід вибрати під сцену спрайт котика. Лише тоді можна створювати скрипт для котика.

При запуску скрипта, котик повинен розміщуватись навпроти песика. Повертати котика не обов'язково, оскільки він знаходиться в потрібному напрямку. Лише додамо рух котика до песика.

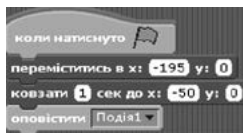


Зверніть увагу на команду *Ковзати*.... З її допомогою спрайт буде плавно рухатись у відповідному напрямку впродовж 1 секунди.

3. Після цих дій потрібно передати керуванню песику, щоб він також рухався у напрямку котика, але лише після того, як котик дійде до вказаних координат. Саме в цьому місці слід скористатися командою **оповістити** з категорії *Керувати*. При натисканні на стрілочку в цій команді слід вибрати категорію *Новий*. Після чого в діалоговому вікні слід ввести назву нової події.



Надалі скрипт для котика виглядатиме так:



Якщо розглядати останню команду як оповіщення повідомлення для песика, то можемо називати її не подія, а повідомлення.

4. Тепер потрібно змусити песика отримати це повідомлення та виконати деякі дії. В скрипті песика додамо новий блок, що починається командою **коли одержую** *Подія1*.

Нехай песик підійде до котика й привітається. Після чого потрібно передати керування котику через наступну подію. Скрипт повинен виглядати так:



Подію *Подія2* повинен отримати котик і також привітатись з песиком. Таким чином можна створити діалог (історію) між песиком та котиком.

5. Спробуйте доробити даний діалог самостійно.

- П.: Привіт, котику!
- К.: Привіт, песику!
- П.: Йдемо до школи!
- К.: Йдемо!
- І одночасно переміщуються вгору.

Вправа 5. Діалог з користувачем

Завдання. Напишіть програму, в якій буде відбуватися діалог з користувачем: запитати ім'я та вік користувача.

Рекомендації до виконання

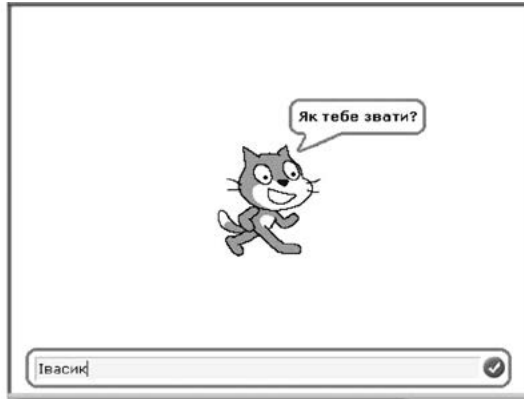
Змінні – це ті категорії, що можуть набувати різних значень (числових або текстових). В математиці ми часто зустрічались з цим поняттям і використовували його при обчисленнях певних завдань. У програмуванні змінні можуть містити не лише числове значення, але й текст. Ось ще одне означення змінної.

Змінна – це іменоване значення певного типу, яке можна змінювати упродовж виконання програми. Тобто, змінна має мати унікальну назву і може містити один з типів даних (ціле число, символ, рядок та інші).

В програмуванні змінні можуть називатись не лише однією буквою *x* чи *y*. *Scratch* дозволяє назвати змінну словом. Наприклад: ол, кб, олівець, разом, кроки тощо.

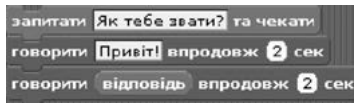
Основне призначення змінних – тимчасово зберігати значення для обчислень у програмі. Також ми можемо задавати значення змінних при запуску скрипта.

1. Примусьте спрайт спитати ім'я користувача, а пізніше привітатись з нами. Для того, щоб учні могли ввести будь-яке значення, використовують команди із категорії **Датчики**. А саме нам потрібна команда **запитати Ваше ім'я? та чекати**. Замість тексту «Ваше ім'я?» напишемо «Як тебе звати?». Під час виконання скрипта, спрайт спитає «Як тебе звати?» і чекатиме, поки учень введе своє ім'я внизу.

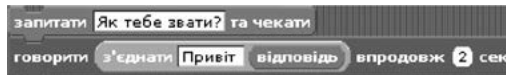


Після вводу імені потрібно натиснути на клавішу *Enter* або клікнути по відповідній кнопці ✓. Ваше ім'я збережеться в тимчасовій області пам'яті, що називається **Відповідь**. Датчик *Відповідь* характерний тим, що зберігає лише останню відповідь (відповідь на останнє питання).

2. Примусьте спрайт привітатись з користувачем. Скористайтеся командою **говорити Привіт! впродовж 2 сек**. Та вдруге використайте цю команду, але замість слова «Привіт» підставте відповідь. Котик привітається з учнем окремими фразами.



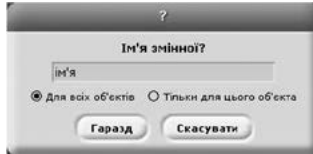
3. Для того, щоб привітання відбувалось одним реченням, потрібно об'єднати слово «Привіт» та датчик *Відповідь* в одну фразу командою **з'єднати Привіт світ** з категорії **Оператори**.



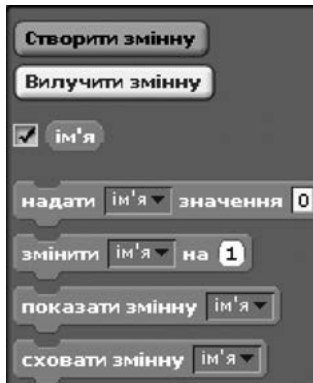
4. Після виконання наступного питання датчик *Відповідь* змінить значення на наступну відповідь. Щоб зберегти це значення довше, потрібно використати змінну.

5. Примусьте спрайт вдруге спитати – «Скільки тобі років?». Після вводу відповіді датчик *Відповідь* змінить значення і не мож-

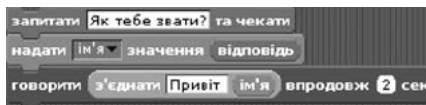
на буде використовувати попереднє значення – ім'я. Отже створимо змінну *Ім'я*. Для цього в категорії **Змінні** скористаємось командою **Створити змінну**. В діалоговому вікні задайте назву змінної *Ім'я* та натисніть кнопку *Гаразд*.



Після чого в категорії з'являться декілька нових команд.



6. Після першого питання потрібно надати змінній *Ім'я* значення *Відповідь*. Та можете змінити наступну команду.

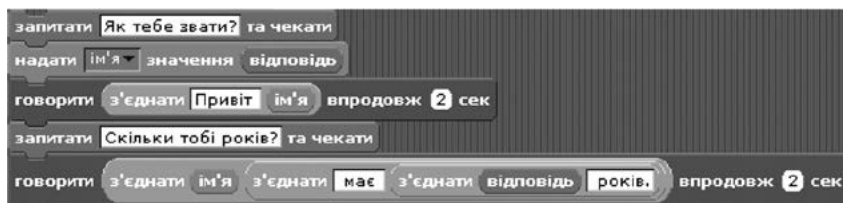


Після виконання скрипта видно, що спрайт виконує ті ж дії, що й попередньо.

7. Тепер примусьте його спитати «Скільки тобі років?» і виведіть фразу «(ім'я) має (відповідь) років». Де відповідно будуть використовуватись змінна *Ім'я* та датчик *Відповідь*. Але для цього потрібно об'єднати чотири слова в одну фразу. Використайте команду *З'єднати* тричі.

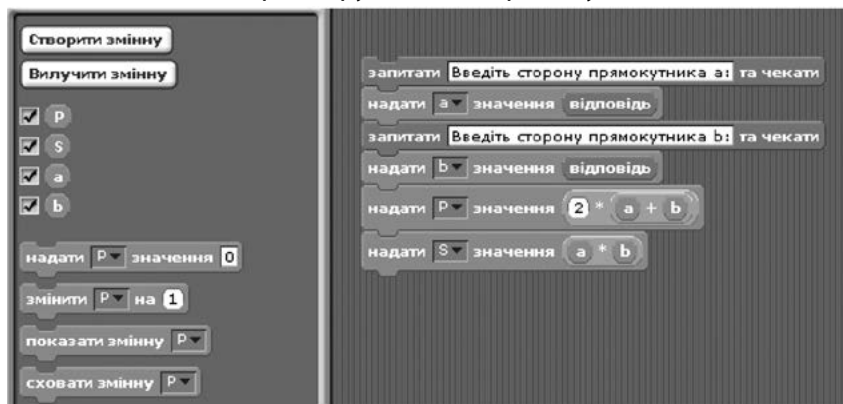


Загалом скрипт виглядатиме так:

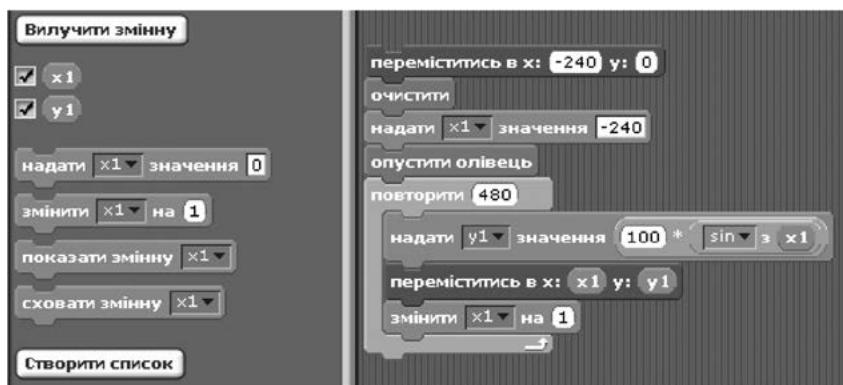


8. Для кращого розуміння змінних спробуйте самостійно проаналізувати наступні скрипти.

Обчислення периметру та площі прямокутника:

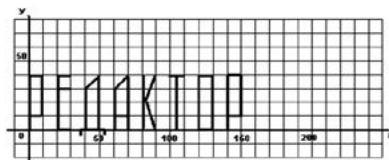
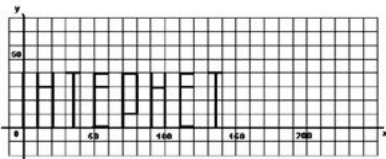


Графік функції $\sin x$:



Завдання для самостійного виконання

1. Скласти алгоритм для виконавця *Рудий Кит*: намалювати квадрат розміром 100x100 кроків, на кожному повороті подумати «Хм...» упродовж однієї секунди (додати команди опустити олівець, встановити колір олівця, встановити розмір олівця).
2. Використовуючи спрайт *Рудого Кота*, зобразити правильний трикутник з довжиною сторін 100 (кроків).
3. Написати на сцені одне зі слів, використовуючи координати сцени.



Поради. Перед складанням програми мовою програмування Скретч доцільно описати команди для побудови літер на аркуші паперу з урахуванням координат ключових точок букв, а також потреби Спрайту залишати слід (малювати – опустити олівець) чи не залишати слід (не малювати – підняти олівець); на початку програми не забуваємо забезпечити «обнуління» сцени.

4. Проект *Анімація*. Змодельювати сюжет «Акваріум» з рухливими водоростями, повітряними бульбашками, рибками.
5. Змодельювати сюжет «Кругообіг води у природі». Передбачити зміну фонів сцени, зміну положення та образу персонажа, який коментує процес кругообігу води.
6. Створити проект, персонаж якого після введення користувачем з клавіатури номера котрогось з місяців року повідомлятиме про відповідну пору року (зима, весна, літо чи осінь) при зміні на належні фон сцени та образу самого персонажа (на фоні сцени напишіть своє прізвище та ім'я).
7. Створити проект *Лабіринт*. Стіни лабіринту мають один колір, двері для переходу на наступний рівень – інший колір. Об'єкт починає рух з визначеної точки сцени. Керування клавішами-стрілками. Об'єкт не може проходити

крізь стіни та виходити за межі сцени. При досягненні виходу з лабіринту повідомляє про це.

- Стіни лабіринту та вихід намалювати як фон сцени.
- Встановити для об'єкту можливість повороту лише справа наліво та зліва направо.
- Сценарій для об'єкту. Розпочинати з визначеного місця у лабіринті. Повторювати, поки об'єкт не торкнеться кольору дверей:
 - якщо натиснута клавіша-стрілка, слід вибрати відповідний напрямок, переміститись на кілька кроків, якщо об'єкт доторкнувся кольору стіни, то:
 - повернути у протилежному напрямку;
 - відступити на ту саму кількість кроків;
 - якщо границя – відбити.

8. Створити проект *Казка* або *Гра*. Для проекту *Казка*: реалізувати взаємодію об'єктів (рух об'єктів до моменту зустрічі, обмін фразами, зміна фону сцени). Після завершення казки вивести повідомлення «Кінець! Казка підготовлена ваше прізвище та ім'я».

Для проекту *Гра*: реалізувати зміну рівнів (зміну фону сцени) при отриманні повідомлення від об'єкта про досягнення виходу. Всього не менше 3-х рівнів.

9. Створити гру за аналогією до відомої *Angry Birds*. Продумати та скласти сценарій проекту, розробити інтерфейс та скрипти.

Досліджуємо

Використовуючи пошукові системи, знайдіть інформацію про те, які проекти можна реалізувати з Raspberry PI.

5 КОМПЛЕКТ FORMULA FLOWCODE BUGGY



5.1. ОСОБЛИВОСТІ КОМПЛЕКТУ FORMULA FLOWCODE BUGGY

Formula Flowcode Buggy створена як платформа для мотивації навчання робототехніці. Робот Formula Flowcode дозволяє користувачам різного віку дізнатися про робототехнічні системи і зрозуміти, як ними управляти. Даний пристрій призначений для задоволення основних потреб технологічної освіти, допомагає розвивати логічне та алгоритмічне мислення.



Рис. 131. Комплект *Formula Flowcode Buggy*

Комплектація та характеристики Formula Flowcode (рис. 131):

1. Машинка (buggy):

- а. шасі: швидкість – 5-20см/с, розмір – 130*80*37 см, мотор – МРМ-GM03 з коробкою передач, акумулятор – AA*4.
- б. плата контролера: процесор – PIC18F4455, виходи – 8 світлодіодів, індикатор живлення, динамік 1 біт, входи – 2 Push-To-Make перемикачі, датчик рівня звуку, мотор – L293D, датчики відстані – TSAL5100, BPV11F трансивери, потужність 4*AA.

2. Стінки для лабіринту (кожна стіна 168*12 *50 мм).
3. Інструкція з використання.
4. Програмне забезпечення – Flowcode [11].

Особливостями даного комплекту є легкість підключення до ПК, наявність готового робота-машинки, можливість імітувати роботу робота в програмному середовищі. Завдяки комплектам *Formula Flowcode* учні можуть легко навчитися першим навичкам програмування, які зводяться до складання алгоритму. Для цього використовують програмне забезпечення *Flowcode* – середовище програмування, в якому можна розробити програму у вигляді простої блок-схеми, змодельовавши її функціональність на екрані, а потім за допомогою натиснення кнопки та за допомогою USB «передати» її до комп'ютера.

Середовище програмування *Flowcode* має простий та зрозумілий графічний інтерфейс, в якому використовується об'єктне програмування (рис. 132). *Flowcode* дозволяє створювати прості програми мікроконтролера за допомогою графічних об'єктів для блок-схем. Створені програми призначені для управління зовнішніми пристроями, підключеними до мікроконтролера, зокрема: AVR, APM і PIC, які є на сьогодні найпоширенішими. Середовище містить вбудовану бібліотеку програм для управління різними периферійними модулями, такими як USART, SPI, ADC, а також компонентами, які, як правило, входять до складу пристроїв на основі мікропроцесорів (світлодіодні індикатори, LCD крокові двигуни тощо). Така можливість дозволяє подати окремі блоки як окремі елементи, які мають відповідні вхід і вихід, що істотно скорочує час на написання та реалізацію програмного коду.

Ще однією корисною функцією даної програми є можливість швидкої перевірки програмного коду на наявність логічних помилок, яка здійснюється за допомогою вбудованого налагоджувача, що дозволяє візуалізувати процеси виведення даних на дисплей, мерехтіння світлодіодних індикаторів і т.д.

Комплект *Formula Flowcode Buggy* можна використовувати для розв'язування як простих завдань, так досить складних проектів.

5.2. ЕЛЕМЕНТИ ВІКНА ПРОГРАМНОГО СЕРЕДОВИЩА FLOWCODE

На рисунку 132 відображено основні елементи вікна програмного середовища [7].

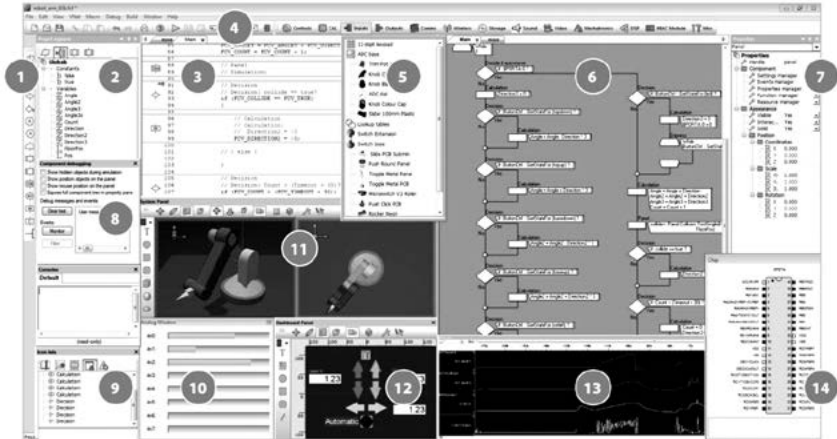


Рис. 132. Елементи вікна Flowcode

1. Панель **Інструментів** – стандартні позначки блок-схем. Для створення блок-схеми слід перетягнути потрібні елементи з цієї панелі інструментів на головне вікно блок-схеми. Як правило, панель розташована в лівому боці екрана. Панель інструментів має такий вигляд:



Рис. 133. Панель Інструментів

2. Панель **Провідник проекту** – кнопки вздовж верхньої частини цієї панелі дозволяють відразу побачити всі порти, макроси, змінні, константи і компоненти у вашому проекті.

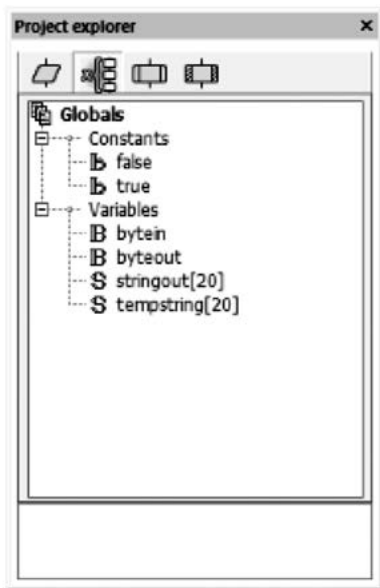


Рис. 134. Панель
Провідник проекту

3. Панель **Програмного коду** призначена для моніторингу С коду, еквівалент блок-схеми.
4. Панель **Управління** – на панелі відображаються зовнішні компоненти, що можуть бути підключені до мікроконтролера, та основні команди панелі малювання. Компоненти згруповано за різними категоріями, які відображаються у вигляді меню. Якщо натиснути на компонент, то його буде додано до мікроконтролера і він з'явиться на панелі. Властивості компонента можна редагувати. Зазвичай ця панель розташована у верхній лівій частині екрану, поруч з *панеллю інструментів*. Панель *Управління* має вигляд:



Рис. 135. Панель Управління

5. **Компонент панелі інструментів** дає можливість обрати електромеханічний компонент з великої бібліотеки деталей; від простого вимикача до модуля *Bluetooth*.
6. **Блок-схема програми** дає можливість перетягувати і редагувати стандартні елементи блок-схеми для створення програми.

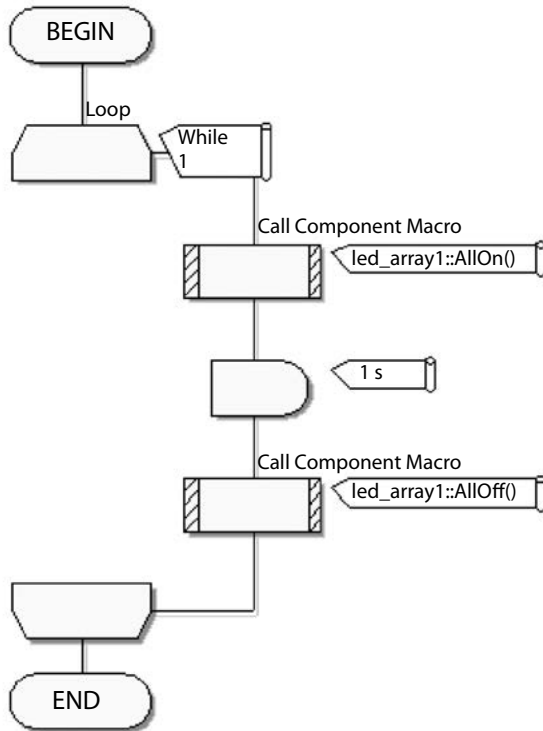


Рис. 136. Блок-схема програми

7. Панель **Властивості** дає можливість переглядати і редагувати властивості всіх компонентів. При виборі елемента панель автоматично стає активною. Деякі властивості призначені тільки для читання, а інші можна редагувати. Панель *Властивості* відображається в правій частині екрану та має вигляд:

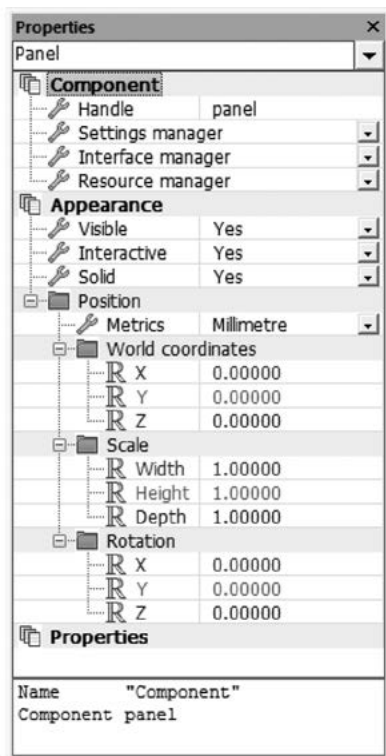


Рис. 137. Панель Властивості

8. **Компонент налагодження** дає можливість переглядати API у програмі користувача і компоненти конструкції.
9. **Список піктограм** використовується для отримання результатів пошуку, повідомлень про помилки, точки зупинки і закладки.
10. **Аналогове вікно** дає змогу подивитися стан аналогових входів у проєкті користувача.
11. **Системна панель** – дизайн системи, що відображається за допомогою системи *MultiView*.
12. Панель **Приладів** містить контроль і моніторинг програми користувача у вигляді моделей об'єктів.
13. Панель **Область застосування** дає змогу показати сигнали, що змінюються, у системі користувача.

14. **Chin** – вікно, що використовується для перегляду та управління станом входів і виходів на активній мікросхемі.

Для розв'язування будь-якої задачі в середовищі *Flowcode* (рис.138) слід виконати такий алгоритм:

1. Створити нову блок-схему, вказавши мікроконтролер, для якого створюється програмний код.
2. Перетягнути необхідні значки елементів з панелі інструментів на блок-схему для програмування додатків.
3. Додати зовнішні пристрої, натиснувши на кнопку в панелі інструментів *Компоненти*, відредагувати їх властивості, вказати як вони пов'язані з мікроконтролером та налаштувати виклики макросів всередині пристрою.
4. Запустити процес моделювання для перевірки роботи створюваного додатку.
5. Перенести створений додаток на мікроконтролер шляхом компілювання блок-схеми на С, потім на асемблері і, нарешті, в об'єктний код.

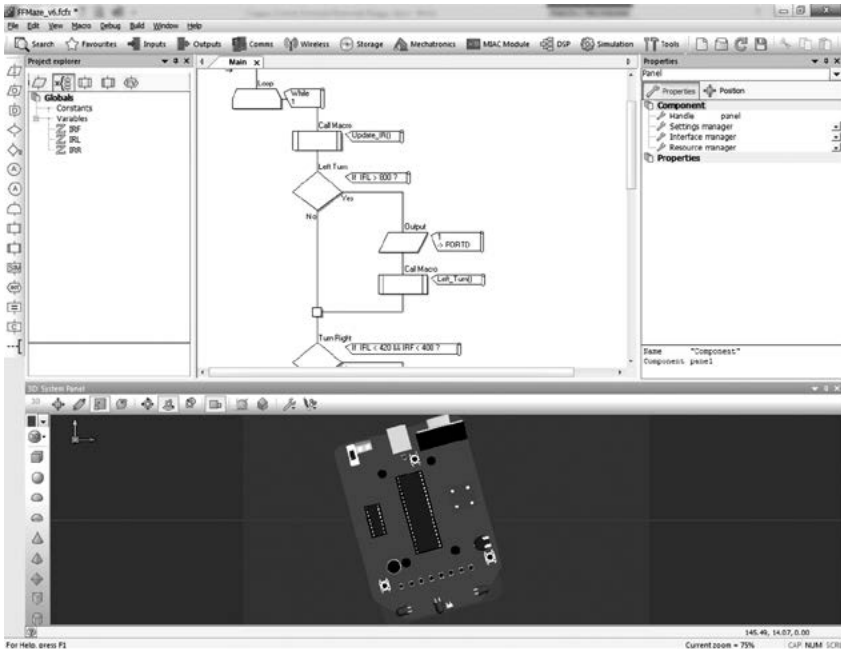


Рис. 138. Створення блок-схеми у *Flowcode*

5.3. ПОЧАТОК РОБОТИ У FLOWCODE

Для швидкої побудови широкого спектру електронних систем можна використовувати Е-блоки: Board-плати та Shield-плати.

Board-плата – пластина, на яку монтуються електронні компоненти, що з'єднуються своїми виводами з елементами провідного рисунка паянням, або, значно рідше, зварюванням, у результаті чого складається електронний модуль – змонтована друкована плата. Готові відкриті платформи дозволяють збирати всілякі електронні пристрої, які можуть працювати як автономно, так і у зв'язці з комп'ютером (рис. 139). Все залежить від ідеї. Board-плата контролює потік інформації в системі. Можна об'єднувати кілька плат одночасно, головне, щоб пристрої не конфліктували за одні і ті ж піни. Для розширення функціональних можливостей Board-плати, використовують плати-розширення, або Shield-плати.

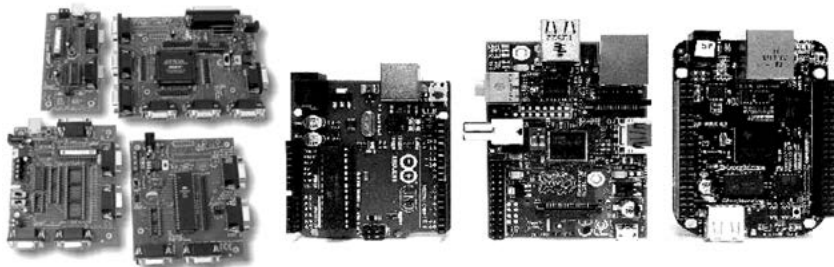
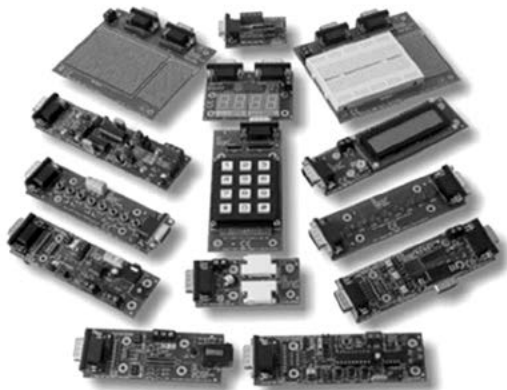


Рис. 139. *Board-плата*

Shield-плата – плата-розширення, готове рішення для реалізації конкретних завдань, що постають перед розробниками апаратного забезпечення (рис. 140). Прикладами таких завдань можуть бути: передача даних за допомогою телефону, робота з Ethernet, управління електронними двигунами. Плати-розширення легко встановлюються на Board-плати, стикуючись з колодками пінів, та перебувають під контролем Board-плати, але інформація може надходити в них і від них.

Рис. 140. *Shield-плата*

Завдання «Засвічування світлодіода». Розглянемо приклад написання програми для засвічування світлодіодів з використанням двох плат у програмі Flowcode.

В таблиці та рисунку 141 представлено обладнання, яке можна використовувати для виконання представленого завдання [22].

| Обладнання | Налаштування |
|--------------------------|-------------------------|
| PIСmicro Device | 16F1937 |
| Блок живлення | 13,5V |
| SW1 (швидкий/ повільний) | повільний |
| SW2 (RC/ XT) | RC |
| XTAL | 19660800 Гц |
| PORTA | N/A |
| PORTB | LED панель (рис. 141) |
| PORTC | N/A |
| PORTD | N/A |
| PORTE | N/A |
| LVP вибір перемикачів | Порт введення/виведення |
| J12-14 (ICD2-USB) | USB |
| J29 (PSU-USB) | PSU |



Рис. 141. Обладнання для виконання завдання

Виконання

Підключення плат та налаштування блок-схеми

1. Під'єднайте LED-панель в порт В як вказано на рисунку:

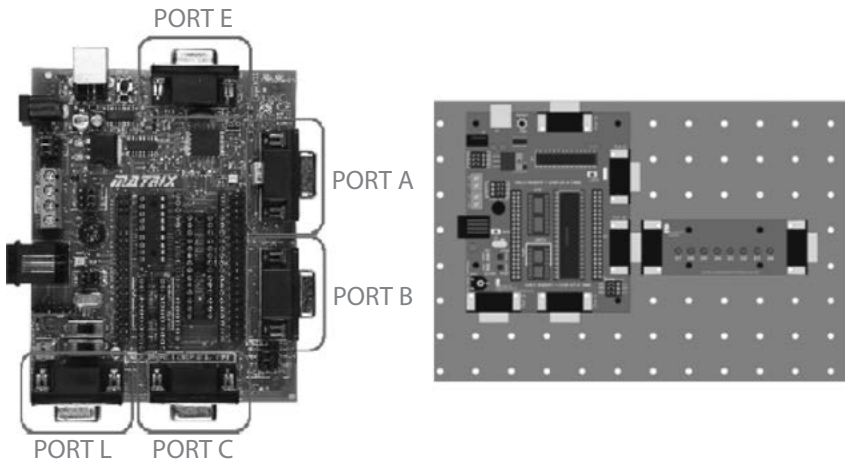


Рис. 142. Підключення LED-панелі

- Створіть новий проект, вибравши *File*, а потім натиснувши кнопку *New*. Вам буде представлений вибір мікроконтролерів, для яких можна розробляти програми. Виберіть 16F1937 зі списку.

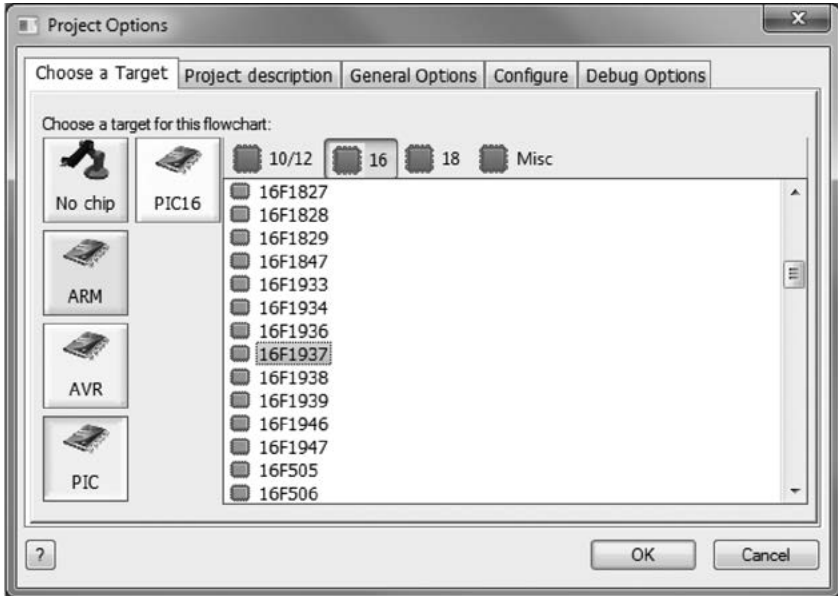



Рис. 143. Вибір мікроконтролера

- Ви побачите, що з'явилися блок-схеми з написами BEGIN END. Всі блок-схеми мають ці елементи за замовчуванням. Наведіть курсор на піктограму *Output*  в панелі інструментів, а потім натисніть і перетягніть її в робочу область. Перемістіть її між BEGIN та END. Коли ви наближаєте елемент до блок-схеми, з'являється стрілка, що вказує куди ваш елемент буде поміщено. Відпустіть кнопку миші, щоб піктограма *Output* стала між BEGIN та END.
- Наведіть курсор на поле *Output*, двічі клацніть лівою кнопкою миші (або клацніть правою кнопкою миші і виберіть *Властивості* з меню, яке з'являється). Тепер ви побачите діалогове вікно:

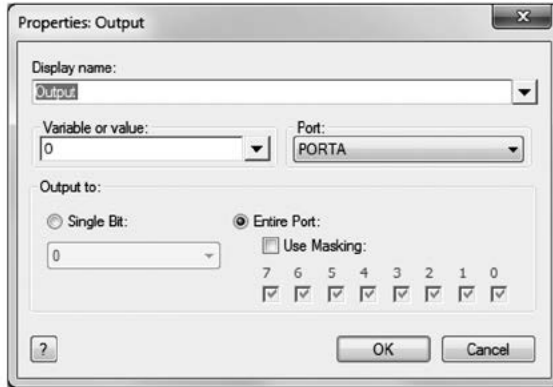



Рис. 144. Налаштування властивостей для порту В

5. Перемістіть курсор на 0 в *Variable or value box* і натисніть кнопку один раз. Це дозволить виділити значення «0». Змініть це значення на «1».
6. Наведіть курсор на стрілку  у вікні *Port*. Натисніть на PORTB в меню, яке відкрилося. Вікно діалогу тепер виглядає як показано нижче.

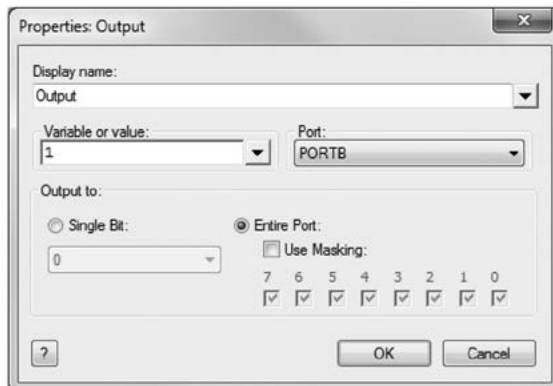



Рис. 145. Налаштування властивостей для порту В

7. Тепер натисніть на кнопку «ОК», щоб закрити вікно. Таким чином, ми вказали на який порт потрібно подавати сигнали.

Додавання світлодіодів

8. Натисніть на кнопку *Outputs* на панелі інструментів і виберіть компонент LED масив. Це виглядає так:  Натисніть, щоб додати набір світлодіодів на системну панель.



9. Натисніть один раз на компоненті в панелі для його вибору та відображення панелі властивостей:

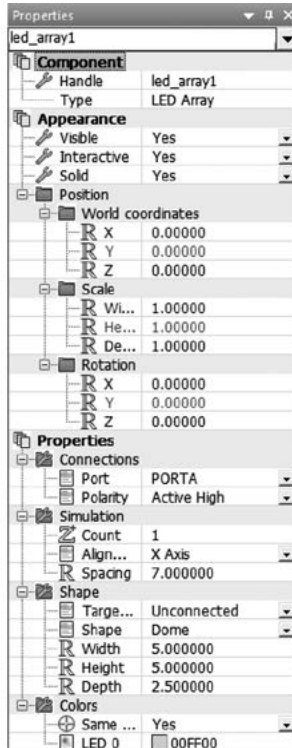


Рис. 146. Налаштування властивостей світлодіодів

10. На панелі властивостей перейдіть у розділ *Simulation* і змініть властивість *Count* на 8, ввівши в поле поруч з ім'ям властивості.



11. На панелі властивостей перейдіть в розділ *Connections* та змініть *Port* на PORTB. Це показує, який порт і біт з'єднаний з чіпом PICmicro.

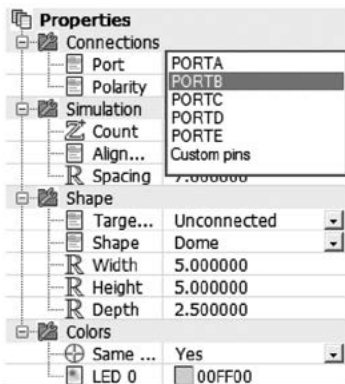




Рис. 147. Налаштування властивостей світлодіодів


12. Ви можете змінити колір світлодіодів, змінивши властивість *Color* в розділі *LED appearance* в панелі Властивостей. Переконайтеся, що *Same Color* має значення *Yes*, щоб змінити весь масив світлодіодів або змінити колір кожного світлодіода окремо.

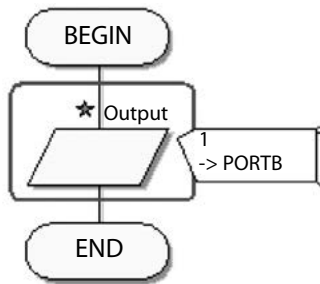


Вітаємо! Ви завершили свою першу програму. Тепер подивимося, що відбувається, коли ви запускаєте виконання цієї програми.

13. Натисніть на значок *Run*  на головній панелі інструментів. Якщо ви моргнули, то ви могли пропустити засвічування світлодіода. Результатом є те, що перший світлодіод загорівся. Це і є повідомленням про те, що моделювання завершено.
14. При натисканні на кнопку *Play*  програма була завершена. Це одна з переваг мікроконтролерів: вони працю-

ють дуже швидко. Зараз ви навчитеся уповільнювати мікроконтролери, керувати затримками тощо.

- Flowcode має функцію, яку можна використовувати для переходу по програмі від елемента до елемента блок-схеми. Якщо виберете кнопку Step Into  на панелі інструментів, то побачите, що Flowcode буде переходити покроково. Це дозволить уповільнити вашу програму і дасть змогу побачити ефект у віртуальній системі. Оскільки ви крокуєте по програмі, то ви побачите червоний прямокутник навколо кожної іконки: це вказує на те, яка іконка буде виконуватися наступного разу при натисканні на Step Into.



- Тепер можна завантажувати програму на мікроконтролер.

5.4. ПРОЕКТИ У FORMULA FLOWCODE BUGGY

Завдання «Засвічування світлодіода на Formula Flowcode Buggy». Запрограмуйте *Formula Flowcode Buggy* таким чином, щоб на роботі засвічувалися світлодіоди.

Рекомендації

- Запустіть *Flowcode*, натисніть *New Project* та оберіть необхідну плату: *Formula Flowcode Buggy* (рис. 148).

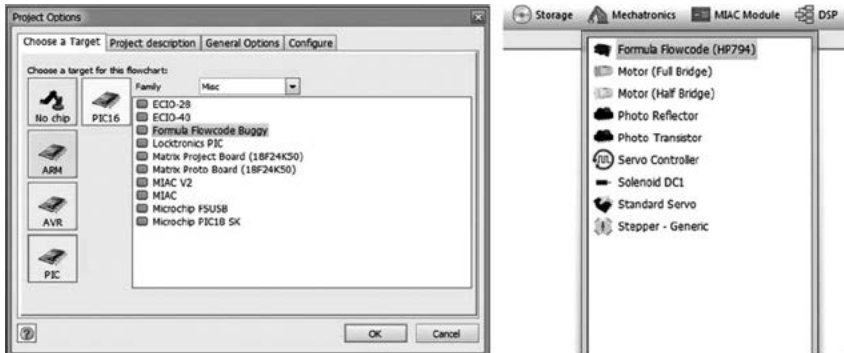


Рис. 148. Вибір плати в програмі Flowcode

2. Модель машинки відобразиться на *Dashboard Panel* (рис. 149).

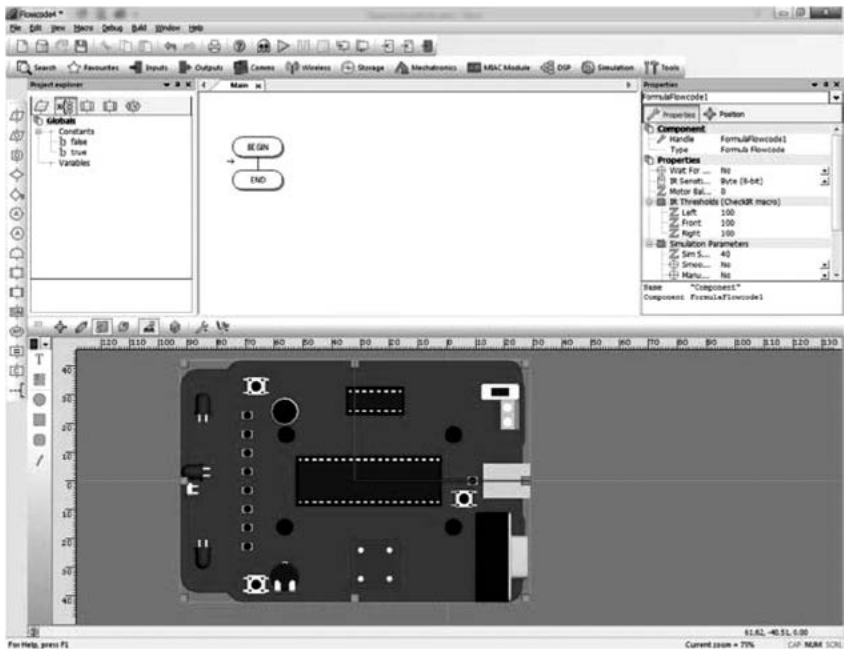


Рис. 149. Плата Flowcode Buggy

3. Використовуючи панель *Інструментів* побудуйте блок-схему та налаштуйте параметри як вказано на рисунку 150.

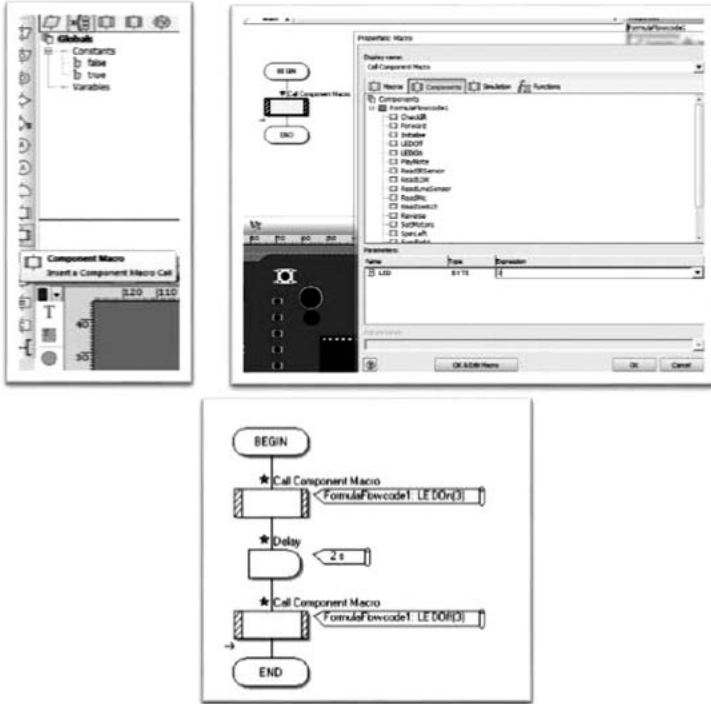


Рис. 150. Налаштування параметрів світлодіода

4. Після збереження програми при натисненні на кнопку *Запуск* (F5) можна спостерігати засвічування світлодіода на декілька секунд та його вимкнення (рис. 151).
5. Завантажте програму на мікроконтролер.

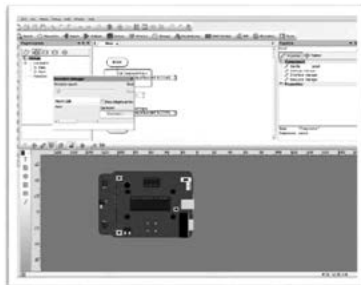


Рис. 151. Виконання програми

Завдання «Танцюючий робот». Запрограмуйте робота таким чином, щоб серія рухів починалася і закінчувалася в одній точці, робот не заїжджав занадто далеко, тобто, танцювальні рухи робота мають бути досить короткими.

Рекомендації. Щоб виконати поставлені умови, слід використати рух вперед, назад та повороти вправо, вліво та коротку затримку після кожного руху. Ви також можете додати деякі паузи всередині ваших танцювальних рухів.

Серію кроків можна об'єднати в макроси (рис. 152), а об'єднавши макроси отримаємо «танець» (рис. 153).

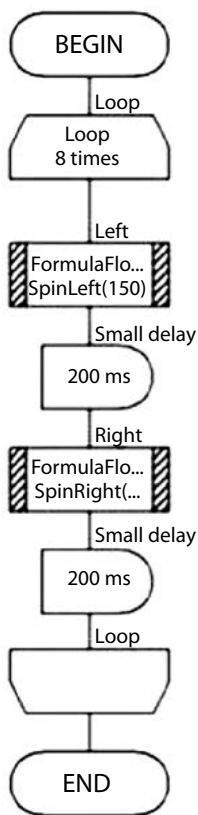


Рис. 152. Створення макросу «Погойдування»

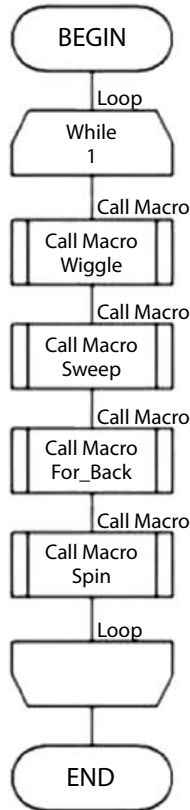


Рис. 153. Блок-схема для «Танцюючого робота»

Завдання «Співаючий робот». Запрограмуйте *Formula Flowcode* таким чином, щоб робот «співав».

Рекомендації. *Formula Flowcode Buggy* – робот, який має зу-мер на борту, а отже, може відтворювати основні звуки, викорис-товуючи макрос *PlayNote*.

Цей макрос приймає 2 параметра:

1. *note* – вибір ноти;
2. *delay_ms* – час звучання (в мілісекундах) обраної ноти.

У таблиці наведені відповідні значення для деяких музичних нот. Поруч наведена програма, за якою робот відтворює перші ноти пісні (рис. 154).

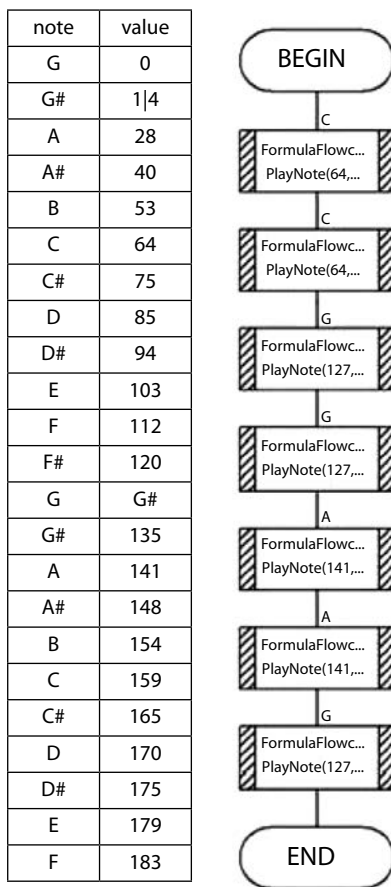



Рис. 154. Значення музичних нот та програма для відтворення нот пісні

Щоб полегшення читання нот у кодї, необхідно ввести змінні, яким присвоїти числове значення нот з таблиці. Для цього необхідно обрати в меню *Edit – Variables*. У діалоговому вікні виберіть пункт *Add new* і введіть ім'я «C», натисніть кнопку ОК (залиште тип змінної «байт»). Аналогічно додайте інші змінні «G», «A» та інші, потім натисніть кнопку *Close*.

Після того, як ми додали ці змінні, ми повинні дати їм відповідні значення, використовуючи піктограму  *Calculation*. Додайте цей елемент до початку вашої програми і двічі клацніть на ньому, щоб ввести значення змінних: C = 64, G = 127, A=141 і т.д. в окремих рядках, а потім натисніть кнопку ОК.

Тепер ви можете використовувати змінні «С», «G» та «А» замість чисел у програмі для «*Співаючого робота*». Ви також можете визначити інші змінні та встановити їм відповідні значення.

Завдання «Під звуки музики». Запрограмуйте *Formula Flowcode* таким чином, щоб робот починав танцювати під справжню музику.

Рекомендації. Для того, щоб «*Танцюючий робот*» виконував рухи під справжню музику, потрібно використовувати датчик мікрофону і макрос *ReadMic*. За основу можна взяти раніше створену танцювальну програму і змінити її так, щоб робот танцював тільки тоді, коли він чує звук.

За замовчуванням, всі показання датчиків є «аналоговими» значеннями і читаються як «байт» (тобто набуває значення в діапазоні від 0 до 255). Отже, спочатку створимо нову змінну *sound*. Щоб прочитати значення датчика мікрофона, використовуйте *ReadMic* макрос і встановіть *return value* для змінної *sound*.

Тепер додайте піктограму *Decision*. Робот повинен танцювати, якщо рівень звуку перевищує певне значення, тому в блоці *if* встановіть *sound > 50*. Якщо рівень звуку перевищує 50, то робот почне танцювати. Якщо ні, він повинен припинити танці.

Блок-схема програми показана на рисунку (рис. 155).

Є ще інші датчики, які розташовані на роботі – LDR (Light Dependant Resistor) для визначення рівня освітленості і три IR (інфрачервоні) датчики, які використовуються для визначення відстані об'єкта до робота. Ці датчики можуть бути використані так само, як і датчик мікрофону.

На додаток до цих аналогових датчиків, є 4 цифрових датчиків: 2 перемикача і 2 лінійно-послідовних. Вони також можуть бути використані як датчики мікрофона, але повертають значення 1 або 0.

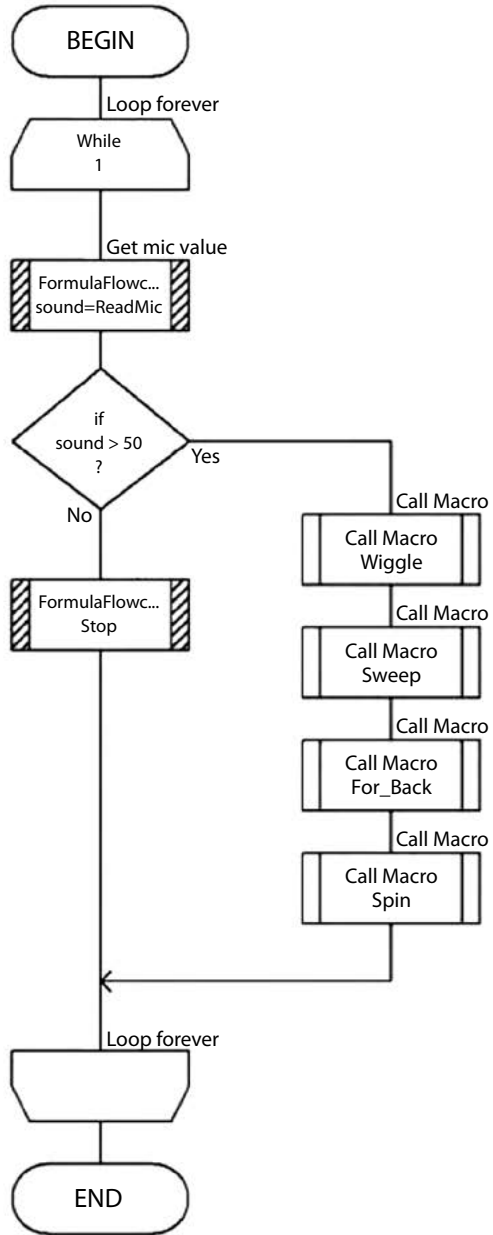



Рис. 155. Блок-схема програми «Під звуку музики»

Завдання «Рух по лінії». Запрограмуйте *Formula Flowcode* таким чином, щоб робот рухався по чорній лінії.

Рекомендації. Датчик лінійного слідування – цифровий датчик, що повертає значення 1, коли датчик знаходиться над чорною поверхнею і 0, якщо знаходиться над білим тлом. Макрос *ReadLineSensor* повертає значення датчика. Він приймає параметр, який вказує лівий датчик (0 або «L») або правий датчик (1 або «R»).

Програма відображає стан обох датчиків лінійного стеження за світлодіодами. Однією з нових можливостей цієї програми є включення коментарів. Ви можете додати їх у програму, розміщуючи піктограму *Comment icon*  на блок-схему. Коментарі дуже важливі в програмах і повинні бути використані в разі потреби, особливо для опису групи елементів. Зверніть увагу, що піктограми можуть мати свої власні коментарі, які можна налаштовувати у властивостях *Display Name*.

У піктограмі *Decision*, вираз «if left < > 0» означає «якщо змінна *left* не дорівнює нулю».

Дотепер ми використовували метод контролю, який називається «*Програмне управління*». Такий метод не дуже добре працює на практиці, тому що не підтримує зворотного зв'язку про положення робота, тому можна тільки здогадуватися у якому напрямку він рухатиметься. У подальшому ми будемо використовувати зворотний зв'язок, щоб створити програму «*Управління роботом зі зворотним зв'язком*».

У наведеній нижче блок-схемі рух вперед існує в нескінченному циклі, який складається з двох частин – перша зчитує значення для кожного датчика, а друга використовує ці значення, щоб вирішити, в якому напрямку має відбуватися переміщення робота.

Макрос *RedLineSensor* зчитує і зберігає данні у змінні *L_Line* і *R_Line*. Таким чином, якщо лівий датчик знаходиться на лінії, то робот рухається вліво, доки датчик не покине лінію. Якщо лівий датчик розташований не над лінією, то ми перевіряємо правий датчик. Якщо він знаходиться на лінії, то ми продовжуємо рухатися вперед. Незважаючи на свою простоту, програма працює добре. Рух робота іноді трохи уривчастий.

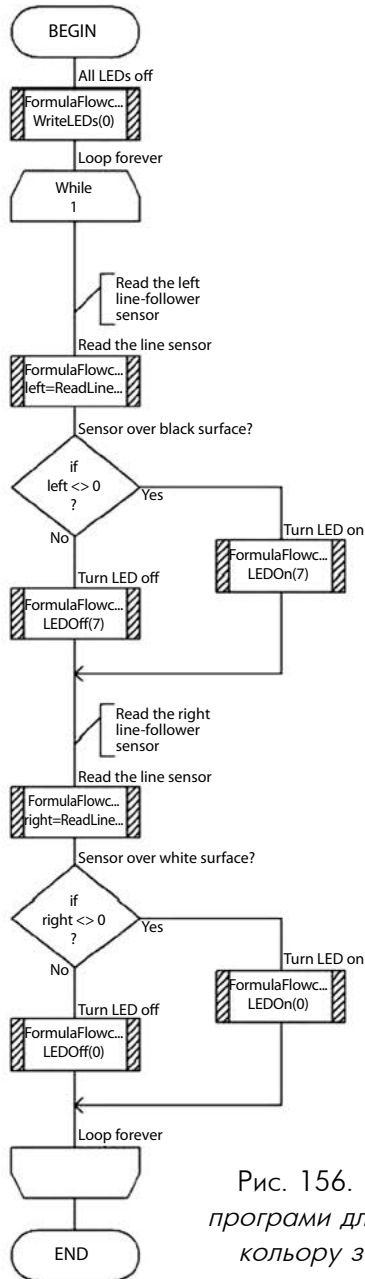


Рис. 156. Блок-схема програми для визначення кольору за датчиком

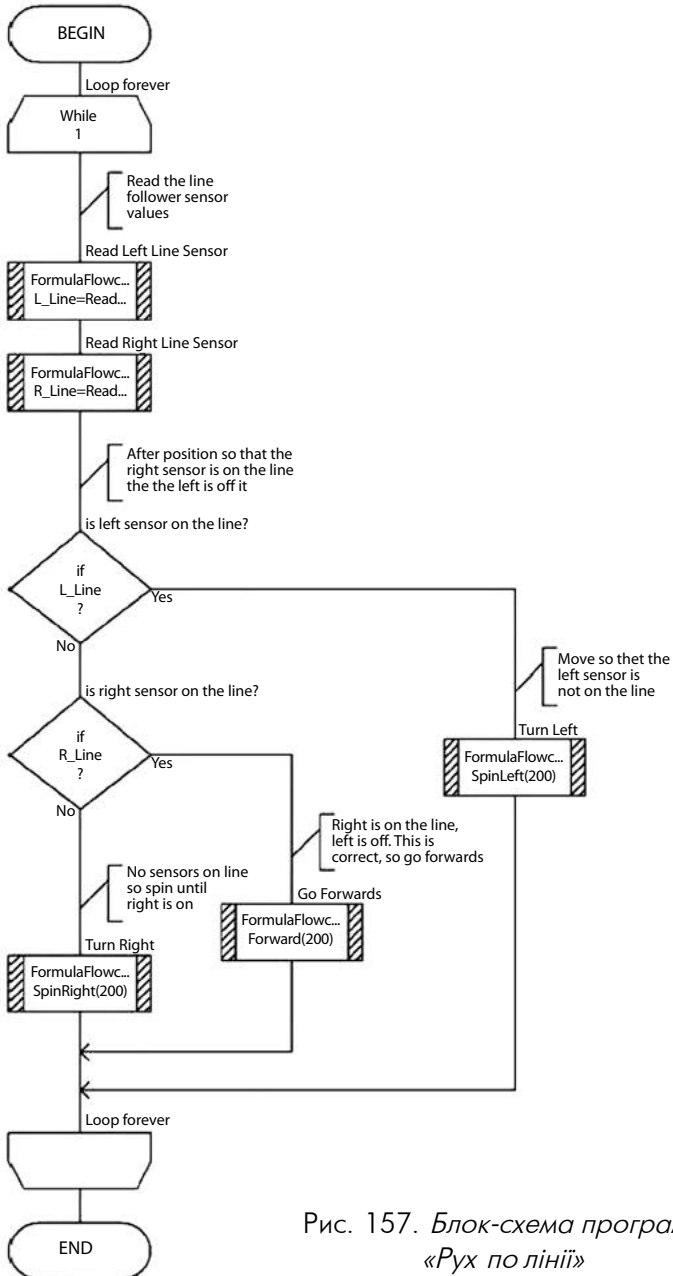


Рис. 157. Блок-схема програми «Рух по лінії»

ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Налаштуйте та протестуйте рухи робота так, щоб він виконував різні рухи в залежності від гучності звуку.
2. Створіть програму, за якою робот рухатиметься вперед, якщо почує хлопок долонь.
3. Створіть програму, за якою робот гратиме залежно від рівня частоти світла LDR.
4. Створіть маршрут, використовуючи білу поверхню та чорну липку стрічку для робота, який рухається по лінії. Змініть або переписіть програму, щоб поліпшити свій час проходження маршруту.
5. Влаштуйте перегони з вашими друзями, щоб побачити, хто написав найбільш ефективний код програми «Рух по лінії». Додайте фари (використовуючи світлодіоди) або сирену (використовуючи зумер).

ЛИТЕРАТУРА

1. 3DNewsDailyDigitalDigest. Режим доступа: <https://3dnews.ru/658701>
2. Asimo. Режим доступа: <http://asimo.honda.com/default.aspx>.
3. Ayres, Robert, and Steve Miller. «Industrial robots on the line.» The Journal of Epsilon Pi Tau 8.2 (1982): 2-10.
4. Chatterjee, N. (2014). Global industrial robotics market (product types, application, technology, end users and geography) - Global share, size, industry analysis, trends, opportunities, growth and forecast, 2013-2020. Portland, OR: Allied Market Research.
5. Consultant Report Securing Australia's Future STEM: Country Comparisons [Online] – Available from: <http://www.acola.org.au/pdf/saf02consultants/consultant%20report%20-%20korea.pdf>
6. da Vinci Surgery. Режим доступа: <http://www.davincisurgery.com/>
7. E-Course Introduction to Microcontroller Programming. Режим доступа: <http://www.matrixtsl.com/courses/itm>
8. Elaine J. Hom. What is STEM Education? – URL: <http://www.livescience.com/43296-what-is-stem-education.html>
6. Симдянкина Е.Е., Никитюк А.М., Спрыгин С.Ф. Межпредметные проекты (физика, химия, биология) как средство развития творческих способностей учащихся. // Успехи современного естествознания. – 2004. – № 5 – С. 80-82 – URL: www.rae.ru/use/?section=content&op=show_article&article_id=7781010
9. Eric the Robot and the Future of Robotics in Industrial Automation. Режим доступа: <http://kingstar.com/eric-robot-humans-robots-living-harmoniously/>
10. Flexible robot manipulators: modelling, simulation and control. – (IET control series) Manipulators (Mechanism) Manipulators (Mechanism) - Automatic control I. Tokhi, A.K.M.

- Azad, Abul III. Institution of Engineering and Technology 629.8'92.
11. Flowcode. Режим доступу: <https://www.matrixtsl.com/flowcode/>
 12. HAL robotics. Режим доступу: <http://www.hal-robotics.com>.
 13. Henn-na Hotel. Режим доступу: <http://www.h-n-h.jp/en/>.
 14. Hiroshi Ishiguro Laboratories. Режим доступу: <http://www.geminoid.jp/en/index.html>.
 15. <http://ieeexplore.ieee.org/abstract/document/569027>.
 16. Hurley, Marlene M. «Reviewing integrated science and mathematics: The search for evidence and definitions from new perspectives.» *School Science and Mathematics* 101.5 (2001): 259-268.
 17. IBM's Watson Has a New Project: Fighting Cybercrime. Режим доступу: <https://www.wired.com/2016/05/ibm-watson-cybercrime/>
 18. IDC FutureScape. Режим доступу: <https://www.idc.com/events/futurescapes>.
 19. International Views of STEM Education. Szu-Chun Chaniel Fan, John M. Ritz. Режим доступу: <http://www.iteea.org/Conference/PATT/PATT28/Fan%20Ritz.pdf>
 20. Introducing Kirobo Mini. Режим доступу: <https://www.toyota-europe.com/world-of-toyota/articles-news-events/introducing-kirobo-mini>.
 21. Koch, D., Ollison, T., Berisso, K., Dalton, A., & Washer, B. (2015). ROBOTICS IN EDUCATION: A postsecondary perspective. *Techniques*, 90(7), 34-37.
 22. Matrix Technology Solutions. Режим доступу: <http://www.matrixtsl.com/eblocks/boards/>
 23. McDonald C. V. International best practice in science, technology, engineering and mathematics (STEM) education / C. V. McDonald. – Brisbane: Griffith Institute for Educational Research. – 2015.
 24. My College Options. Режим доступу: <https://www.mycollegeoptions.org/>

25. Nourbakhsh, I., Hamner, E., Crowley, K., & Wilkinson, K. (2004, April). Formal measures of learning in a secondary school mobile robotics course. Proceedings of the 2004 IEEE International Conference on Robotics and Automation. Retrieved from http://www.ri.cmu.edu/publication_view.html?pub_id=5570
26. Panasonic Revives Hospital Delivery Robot. Режим доступу:
27. Papert, Seymour. Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc., 1980.
28. Petre, M., & Price, B. (2004). Using robotics to motivate 'backdoor' learning. *Education and Information Technologies*, 9(2), 147-158. Retrieved from <http://citeseerx.ist.psu.edu>.
29. Project STEAM Educators [Online] – Available from: <http://steamedu.com/about-us>
30. Project STEM Book of Insights 2014. – 77 p. – URL: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/351433/BIS-14-899-STEM-book-of-insights.pdf
31. RP-VITA ROBOT EXTENDS SPECIALIZED MEDICAL CARE. Режим доступу: <https://medtechboston.medstro.com/blog/2014/08/26/rp-vita-robot-extends-specialized-medical-care>.
32. Valero, R., et al. «Robotic surgery: History and teaching impact.» *Actas Urológicas Españolas (English Edition)* 35.9 (2011): 540-545.
33. van Lith, P. (2007). Teaching robotics in primary and secondary schools. Paper presented at the 2007 ComLab Conference, November 30-December 1, 2007, Radovljica, Slovenia. Retrieved from http://metodika.phy.hr/infiro/Conference-CD/papers_pdf/peter.pdf.
34. Азимов, Айзек. Я, робот. Серия: Библиотека приключений. М: Эксмо (2002).
35. Вікі КДПУ. Режим доступу: <http://wiki.kspu.kr.ua/index.php/%D0%9F%D0%BE%D1%87%D0%B0%D1%82%D0%BE%D0%BA>.
36. Вікіпедія. Режим доступу: [uk.wikipedia.org/wiki/Скретч_\(мова_програмування\)](http://uk.wikipedia.org/wiki/Скретч_(мова_програмування))

37. Горохов, В. Г. Развитие идей теории механизмов и машин в теории автоматического регулирования и нанотехнологии // Наука та наукознавство, 2011.
38. Електронний журнал 3DNews. Режим доступу : <http://www.3dnews.ru/658701>
39. Енциклопедія кібернетики / Відпов. ред. Глушков В.М. – Т.2 (М-Я). – К.: Головна редакція української радянської енциклопедії, 1973. – 576 с.
40. Ковальчук, А. К., et al. История и развитие антропоморфных шагающих роботов как нового класса робототехнических средств для экстремальных условий. Наука и образование: научное издание МГТУ им. НЭ Баумана – 12, 2006.
41. Курников, С. С. РОБОТОТЕХНИКА. ИСТОРИЯ РАЗВИТИЯ // Ответственный за выпуск: ВА Хоник (2016): 136.
42. Открытые технологии в образовании. [Электронный ресурс] – Режим доступа: <http://open-education.net/academic/school/v-finlyandii-budut-uchit-detej-ne-predmetam-a-po-temam>.
43. Петин В. Микрокомпьютеры Raspberry Pi: Практическое руководство, БХВ-Петербург, 2015 – 240 с.
44. Словник. Режим доступу: <https://slovar.cc/enc/bse/2036654.html>.
45. Чебышев, Пафнутий Львович. Теория механизмов, известных под названием параллелограммов // Успехи математических наук 1.2 (12 (1946): 12-37.
46. Черняк, В. С. Д. Борелли и первая форма универсализации механики. И 89 Истина и благо: универсальное и сингулярное. – М., 2002. –: 167.
47. Шевченко, А. И., И. С. Сальников, and Р. И. Сальников. Антропоморфно-гуманоидные роботы: эволюция, проблемы, перспективы» Искусственный интеллект 3 (2006): 39-46 с.

СЛОВНИК ТЕРМІНІВ І ПОНЯТЬ

3D-друк – процес виконання ряду повторюваних операцій (нанесення на робочий стіл установки тонкого шару витратних матеріалів). Цикли друку безперервно слідуєть один за одним: на попередній шар матеріалів наноситься наступний шар. Технологія 3D-друку також відома під назвою **адитивна технологія**.

3D-принтер – спеціальний пристрій для виведення тривимірних даних. 3D-принтер дозволяє виводити тривимірні дані, тобто створювати певні фізичні об'єкти.

Board-плата – пластина, на яку монтуються електронні компоненти, що з'єднуються своїми выводами з елементами провідного рисунка паянням, або, значно рідше, зварюванням, у результаті чого складається електронний модуль – змонтована друкована плата.

Flowcode – середовище програмування, в якому можна розробити програму у вигляді простої блок-схеми, змодельовавши її функціональність на екрані, а потім за допомогою натиснення кнопки та за допомогою USB «передати» її до комп'ютера.

Raspberry Pi – одноплатний комп'ютер, розроблений британським фондом Raspberry Pi Foundation. Його головне призначення – стимулювати навчання базовим комп'ютерним наукам у школах.

Raspbian – спеціалізований дистрибутив Linux для Raspberry Pi (заснований на дистрибутиві Debian).

Scratch – середовище програмування, яке дозволяє дітям створювати власні анімовані та інтерактивні історії, ігри та інші витвори.

Shield-плата – плата-розширення, готове рішення для реалізації конкретних завдань, що постають перед розробниками апаратного забезпечення (передача даних за допомогою телефону, робота з Ethernet, управління електронними дви-

гунами). Плати розширення встановлюються на Board-плати, стикуючись з колодками пінів, та перебувають під контролем Board-плати, але інформація може надходити в них і від них.

USB (англ. Universal Serial Bus, абрєвіатура читається ю-ес-бі) – укр. універсальна послідовна шина, призначена для з'єднання периферійних пристроїв обчислювальної техніки. Символом USB є чотири геометричні фігури: квадрат, трикутник, велике коло та мале коло.

Автомат – пристрій, який виконує роботу самостійно відповідно до вбудованої програми.

Автоматика (грец. αὐτόματος — самодіючий) — галузь науки і техніки, яка розробляє технічні засоби і методи для здійснення технологічних процесів без безпосередньої участі людини. Сукупність механізмів, приладів, що діють автоматично.

Блок – мінімальний фрагмент програми у середовищі програмування *Scratch*: змінна, оператор, функція або структура, що керує. Блоки згруповано у 8 категорій.

Виконавча або моторна система – пристрій, призначений для безпосереднього впливу на об'єкти навколишнього середовища або взаємодії з ними відповідно до сигналів, що формуються інформаційно-вимірною системою або безпосередньо оператором.

Всеукраїнський фестиваль робототехніки Robotica – це масштабна подія у сфері науки, техніки і освіти в якій беруть участь талановиті, творчі, ерудовані діти зі всієї України, які демонструють свої вміння у сфері роботобудування, інженерії і архітектури. Всеукраїнська олімпіада — це частина Світових олімпіад з робототехніки WRO та FLL, яка покликана принести в Україну найкращу філософію навчання та найкращі освітні інструменти. Більше 30000 гостей приєднуються до свята, яке, як завжди проходить під гаслом «Роботи, творчість, майбутнє».

Інтерфейс введення/виведення загального призначення (англ. General-purpose input/output, **GPIO**) – інтерфейс для зв'язку між компонентами комп'ютерної системи, наприклад, мікропроцесором і різними периферійними пристроями.

Контакти GPIO можуть діяти і як входи, і як виходи, і це, як правило, підлягає налаштуванню. GPIO-контакти часто групуються в порти.

Інформаційно-вимірювальна або сенсорна система – це штучні «органи» чуття робота, призначені для сприйняття і перетворення інформації про стан зовнішнього середовища і самого робота.

Кабель (англ. cable, нім. kabel) — один чи більше ізольованих проводів (провідників), поміщених у оболонку, поверх якої залежно від умов прокладання та експлуатації може видозмінюватись захисне покриття, у яке може входити броня (див. кабель броньований).

Кібернетика (англ. cybernetics, нім. Kybernetik) — наука про загальні принципи управління в різних системах: технічних, біологічних, соціальних та ін.

Компіляція – трансляція вихідного коду програми в бінарний код.

Кросплатформність – властивість програмного забезпечення працювати більш ніж на одній програмній (в тому числі – операційній системі) або апаратній платформі, та технології, що дозволяють досягти такої властивості. Кросплатформність дозволяє суттєво скоротити витрати на розробку нового або адаптацію існуючого програмного забезпечення.

Лобі – основне вікно програмного забезпечення LEGO MINDSTORMS EV3, що дозволяє легко орієнтуватися і працювати з програмою.

Машинний зір – це застосування комп'ютерного зору в промисловості та виробництві. В той час як комп'ютерний зір — це загальний набір методів, що дозволяють комп'ютерам бачити, область інтересу машинного зору, як інженерного напрямку, є цифрові пристрої введення/виведення та комп'ютерні мережі, призначені для контролю виробничого обладнання, такого як роботи-маніпулятори чи апарати для вилучення бракованої продукції. Машинний зір є підрозділом інженерії, пов'язаним з обчислювальною технікою, оптикою, машинобудування та промисловою автоматизацією. Одним з найпоширеніших застосувань машинного зору є інспекція промис-

лових товарів, таких як напівпровідникові чіпи, автомобілі, продукти харчування та ліки. Люди, що працюють на складальних лініях, оглядають частини продукції і роблять висновки про якість виконання. Системи машинного зору для цієї мети використовують цифрові та інтелектуальні камери, а також програмне забезпечення обробки зображення для виконання аналогічних перевірок.

Механізм – пристрій, за допомогою якого людина може виконувати роботу; внутрішній пристрій машини, що забезпечує її роботу.

Мікроконтролер – виконана у вигляді мікросхеми спеціалізована мікропроцесорна система, що включає мікропроцесор, блоки пам'яті для збереження коду програм і даних, порти вводу-виводу і блоки зі спеціальними функціями (лічильники, компаратори, АЦП та інші).

Привід – це «м'язи» робота. В наш час, найбільш розповсюдженими двигунами у приводах, є електричні, але застосовуються і інші, що використовують хімічні речовини або стиснене повітря.

Проект – обмежена часовими рамками діяльність, що має визначений початок та кінець, зазвичай обмежений датою, але також може обмежуватися фінансуванням або досягненням результатів, яка здійснюється для реалізації унікальних цілей та завдань зазвичай, щоб призвести до вигідних змін або створення доданої вартості.

Проект – стадія розробки виробу і проектна конструкторська документація, яка містить остаточне технічне рішення і дає повне уявлення про будову розроблюваного виробу.

Процесор – основний компонент комп'ютера, призначений для керування всіма його пристроями та виконання арифметичних і логічних операцій над даними.

Робот – автономно функціонуюча універсальна автоматична машина, призначена для відтворення фізичних, рухових і розумових функцій людини, наділена здатністю до адаптації і навчання у процесі активної взаємодії з навколишнім середовищем.

Робототехніка – прикладна наука, що опікується проектуванням, розробкою, будівництвом, експлуатацією та використанням

роботів, а також комп'ютерних систем для їх контролю, сенсорного зворотного зв'язку і обробки інформації автоматизованих технічних систем (роботів).

Світлодіод – напівпровідниковий пристрій, що випромінює некогерентне світло, при пропусканні через нього електричного струму (ефект, відомий як електролюмінесценція).

Сенсор або датчик (sensor) – пристрій для перетворення деякої фізичної величини в електричний сигнал. Сенсори є по суті нервовою системою робота і служать для забезпечення зворотного зв'язку між контролером і навколишнім світом. Оскільки на поведінку автоматичної системи можуть впливати багато фізичних факторів, існує і безліч різних сенсорів, здатних ці фактори фіксувати.

Сервомотор (сервопривід) – пристрій у системах автоматичного регулювання або дистанційного керування, що за рахунок енергії допоміжного джерела здійснює механічне переміщення регулюючого органу відповідно до отримуваних від системи керування сигналів.

Система зв'язку – «мова» робота, призначена для передачі сигналів інформації між системами робота, а також для організації обміну інформацією між роботом і людиною або іншими роботами і пристроями з метою здійснення діалогу, надання команд роботу, контролю за функціонуванням його систем, діагностики несправностей, регламентної перевірки тощо.

Скрипт (script, сценарій, метод) – послідовність вказівок, що визначає які дії і в якому порядку потрібно виконати певному об'єкту (спрайту).

Слот – щілинний (англ. slot означає «щілину») роз'єм, зазвичай в комп'ютері, з'єднаний із системною шиною і призначений для установки додаткових модулів (карт розширення), що розширюють конфігурацію пристрою.

Сонар – Sound Navigation And Ranging.

Спрайт – об'єкт середовища програмування *Scratch*, пов'язаний із зображенням, набором змінних і скриптів, які визначають його поведінку.

Стек (stack) – набір послідовно сполучених різнокольорових графічних блоків у межах однієї події в середовищі *Scratch*.

Три закони робототехніки – обов'язкові правила поведінки для роботів, вперше сформульовані Айзеком Азімовим в оповіданні «Я, робот» (1941). Типове уявлення робота до Законів Азімова (мультфільм «Супермен»)

У практичному плані наявність лише неповних знань про мозок, про його функціонування не заважає будувати його наближені інформаційні моделі, моделювати на ЕЦОМ найскладніші процеси мислення, у тому числі й творчі.

1. Робот не може заподіяти шкоду людині, або своєю бездіяльністю дозволити, щоб людині була заподіяна шкода;
2. Робот повинен захищати самого себе, якщо тільки його дії не суперечать першому і другому пунктам.
3. Робот повинен підкорятися наказам людини, за винятком тих, котрі суперечать першому пункту;

Управляюча або інтелектуальна система – це «мозок» робота, який призначено для управління механізмами виконавчої системи на основі вбудованої програми з урахуванням сигналів зворотного зв'язку від сенсорної системи.

Штучний інтелект (англ. Artificial intelligence, AI) — розділ комп'ютерної лінгвістики та інформатики, що займається формалізацією проблем та завдань, які нагадують завдання, виконувані людиною. технічна (в усіх сучасних випадках спроб практичної реалізації — комп'ютерна) система, що має певні ознаки інтелекту, тобто здатна: розпізнавати та розуміти; знаходити спосіб досягнення результату та приймати рішення; вчитися.

ДОДАТКОВІ РЕСУРСИ

3D-друк – це виконання ряду повторюваних операцій (нанесення на робочий стіл установки тонкого шару витратних матеріалів). Цикли друку безперервно слідуєть один за одним: на попередній шар матеріалів наноситься наступний шар. Технологія 3D-друку також відома під назвою адитивна технологія.

3D-принтер – це спеціальний пристрій для виведення тривимірних даних. 3D-принтер дозволяє виводити тривимірну інформацію, тобто створювати певні фізичні об'єкти.

Застосування 3D-друку

- створення прототипів;
- швидке виробництво;
- дрібносерійне виробництво;
- навчальні моделі;
- медицина;
- реклама.

Варіанти застосування 3D-друку

Дизайнер Ріккі Ма з Гонконгу створив робота-гуманоїда, який виглядає і рухається як голлівудська актриса Скарлетт Йоханссон. Скелет робота надрукований на 3D-принтері.



Рис. 158. Робот-гуманоїд

ODD Guitars використовують принтер Cubify 3D для друку корпусів для гітар. У середньому вартість подібної гітари складає близько 3000 \$.



Рис. 159. Друкована гітара

Друк металевого моста над одним з каналів Амстердама за допомогою MX3D metal принтера:



Рис.160. Друкований міст

Модель середньовічного замку на площі 3х5 метри.



Рис.161. Друкований замок

3D крісло з губчастою структурою, яке здатне набувати форми тіла людини.



Рис.162. Друковане крісло

Дизайнер з Ізраїлю винайшла каблучки, які не викликають захворювань. Виготовляють їх з нейлону та світлочутливого фотополімерного матеріалу, який не ковзає.



Рис.163. Друковані туфлі

Студія дизайну «Nervous System» в США представила на виставці у Бостоні в Музеї образотворчих мистецтв з 10 березня по 6 липня 2016 р. неповторну яскраво-червону сукню, надруковану на 3D-принтері.

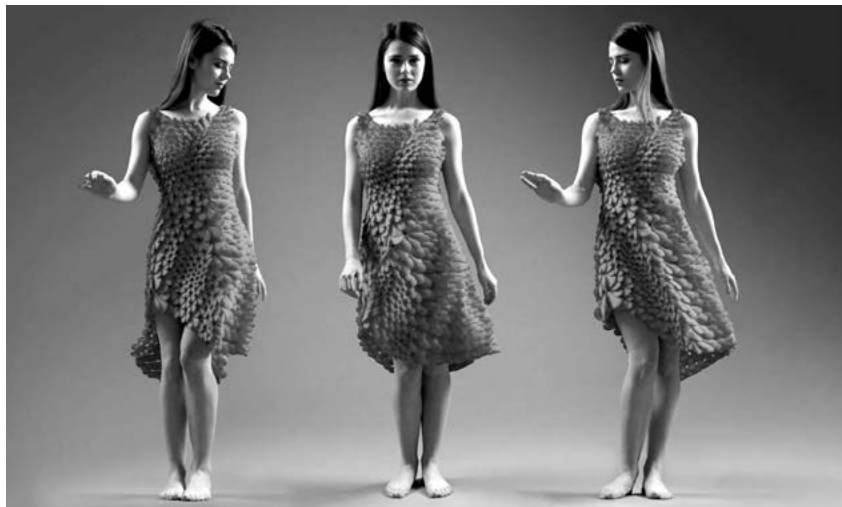


Рис.164. Друкована сукня

Аprecia розробила спосіб індивідуального дозування пігулок для хворих на епілепсію (схвалено FDA – управлінням продовольства і медикаментів), які розчиняються під час контакту з рідиною.



Рис.165. Друковані таблетки

Надрукований за 44 години за допомогою 3D-принтера автомобіль Strati.



Рис.166. Друковане авто

Компанія Boeing запатентувала технологію 3D-друку левітуючих об'єктів. Перший елемент друкарського матеріалу, який називають «самородком», утримується в повітрі за допомогою магнітних полів. Потім 3D-принтери додають до «самородку».



Рис.167. Друкований левітуючий об'єкт

Один з найоригінальніших напрямків 3D-друку – друк власне принтерів - RepRap.

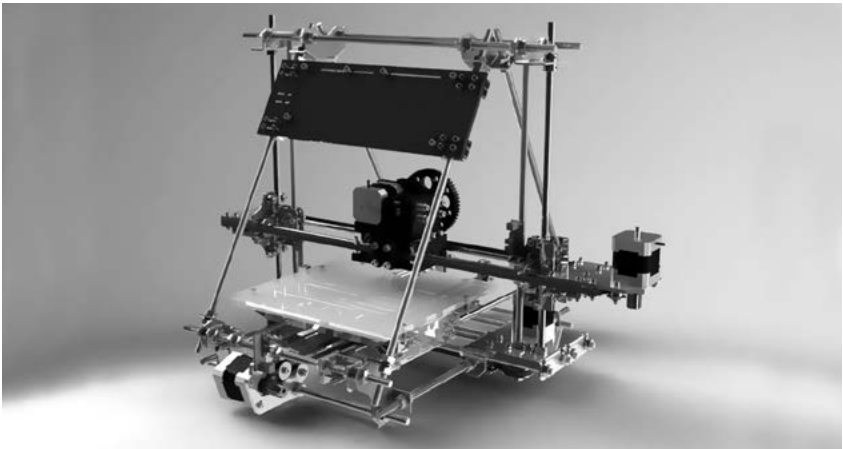


Рис.168. 3D принтер

Технології 3D-друку

Стереолітографія (Stereolithography – SL)

Під дією керованого комп'ютером ультрафіолетового випромінювання відбувається затвердіння шару завтовшки в декілька сотих міліметра, при цьому платформа з майбутньою деталлю опускається вниз і знову покривається рідиною. Далі все повторюється й в результаті ультрафіолетовий промінь «малює» об'ємну фігуру.

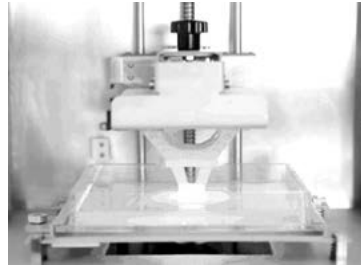


Рис.169. *Stereolithography*

Селективне лазерне спікання (Selective Laser Sintering)

Суть полягає в пошаровому спіканні лазерним променем порошкового матеріалу. По поверхні зони обробки, лазером спікається потрібний контур, далі насипається новий шар, розрівнюється, і процес повторюється.

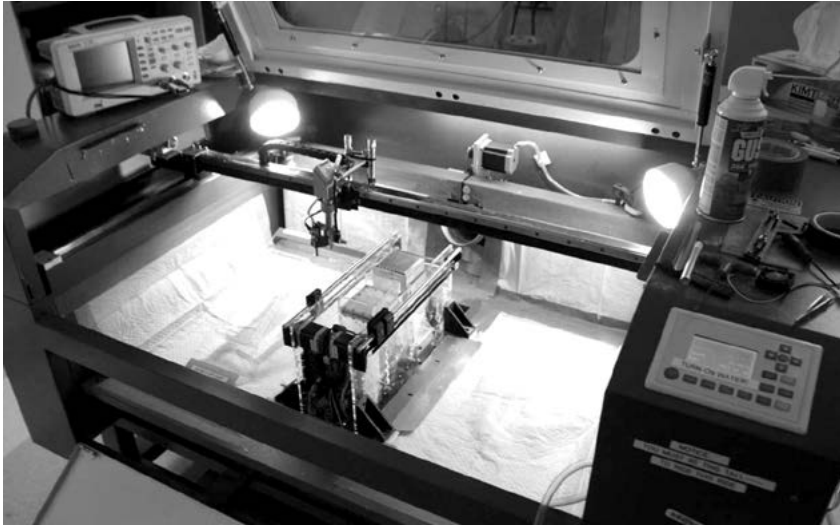


Рис.170. *Selective Laser Sintering*

Пошарове формування об'ємних моделей з листового матеріалу (Laminated Object Manufacturing – LOM)

Листовий матеріал (папір, пластик, кераміка, композити або поліестер) розкроюється за заданим контуром за допомогою CO₂-лазера (можна одночасно розкроювати більше одного аркуша, проте точність при цьому зменшується), а потім нагрівається валик, який здійснює склеювання шарів. При помилці в процесі синтезу об'ємного виробу частину шарів можна видалити.



Рис.171. *Laminated Object Manufacturing*

Струменева полімеризація (Polyjet and Ployjet Matrix)

Процес друку полягає в наступному. На площину побудови згідно з програмним алгоритмом наноситься рідкий фотополімер блоком друкуючих головок. Блок складається з 8 головок – це 768 сопел малого діаметра, що здатні продукувати близько 16 млн крапель на хвилину. На друкуючій голівці розміщені дві ультрафіолетові (УФ) лампи, які замінюють лазер в SLA-установках. Після нанесення фотополімер полімеризується під дію ультрафіолетового світла. Цим завершується побудова одного шару. Далі площину побудови зміщують на дуже малий рівень і головки створюють наступний шар.



Рис.172. *Polyjet and Ployjet Matrix*

Моделювання плавленням (Fused Deposition Modeling – FDM)

Технологія FDM є створенням тривимірних об'єктів за рахунок нанесення послідовних шарів матеріалу, які повторюють контури цифрової моделі. Як правило, в якості матеріалів для друку виступають термопластики, що поставляються у вигляді котушок ниток або прутків.

Оригінальний термін «Fused Deposition Modeling» і аббревіатура FDM є торговими марками компанії Stratasys. Ентузіасти 3D-друку, учасники проекту RepRap, придумали аналогічний термін «Fused Filament Fabrication» («Виробництво методом наплавлення ниток») або FFF для використання в обхід юридичних обмежень. Терміни FDM і FFF еквівалентні за змістом і призначенням.

Технологія друку методом пошарового наплавлення (FDM) була розроблена С. Скоттом Трампом наприкінці 1980-х рр. і представлена на ринку компанією Stratasys, починаючи з 1990 р. На даний момент технологія набуває все більшого поширення серед ентузіастів, що створюють принтери з відкритим вихідним кодом, а також комерційних компаній, у зв'язку із закінченням терміну дії оригінального патенту. У свою чергу, широке поширення техно-

логії призвело до істотного зниження цін на 3D-принтери, що використовують цей метод виробництва.

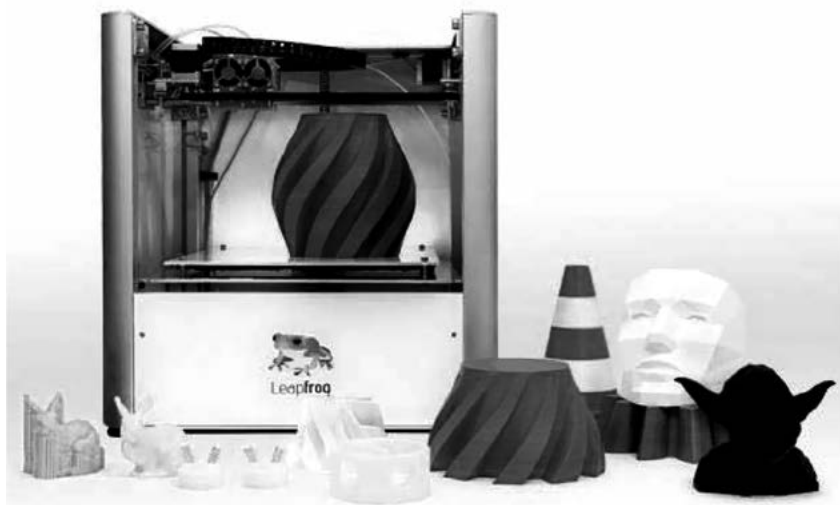


Рис.173. Принтер Leapfrog Creatr HTS

Виробничий цикл починається з обробки тривимірної цифрової моделі. Модель в форматі STL ділиться на шари і орієнтується найбільш підходящим чином для друку. За необхідності генеруються підтримуючі структури, необхідні для друку нависаючих елементів. Деякі пристрої дозволяють використовувати різні матеріали під час одного виробничого циклу. Модель виготовляється видавлюванням («екструзією») і нанесенням мікрокрапель розплавленого пластика з формуванням послідовних шарів, що застигають відразу після екструдкування.

Пластикова нитка розмотується з бобіни і заправляється в екструдер – пристрій, оснащений механічним приводом для подачі нитки, нагрівальним елементом для плавлення матеріалу і соплом, через яке здійснюється безпосередньо екструзія. Нагрівальний елемент служить для нагрівання сопла, яке в свою чергу плавить пластикову нитку і подає розплавлений матеріал на робочий стіл, на якому «вирощується». Верхня частина сопла навпаки охолоджується за допомогою вентилятора для створен-

ня різкого градієнта температур, необхідного для забезпечення плавної подачі матеріалу.

Екструдер переміщується в горизонтальній і вертикальній площинах під контролем алгоритмів – **g-коду**. Сопло переміщується по траєкторії, заданій системою автоматизованого проектування слайсером. Модель будується шар за шаром, від низу до верху. Екструдер приводиться в рух покроковими моторами або сервоприводами. Найбільш популярною системою координат, яка застосовується в FDM, є декартова система, побудована на прямокутному тривимірному просторі з осями X, Y і Z. Альтернативою є циліндрична система координат, яка використовується так званими «дельта-роботами».

В якості витратних матеріалів доступні всілякі термопластики і композити, включаючи ABS, PLA, полікарбонати, поліаміди, полістирол, лігнін та багато інших.

Моделювання методом пошарового наплавлення (FDM) застосовується для швидкого прототипування, швидкого виробництва. FDM є одним з найменш дорогих методів друку, що забезпечує зростаючу популярність побутових принтерів, заснованих на цій технології. У побуті 3D-принтери, що працюють за технологією FDM, можуть застосовуватися для створення різноманітних об'єктів цільового призначення, іграшок, прикрас і сувенірів.

Технологія друку FDM використовується і у навчанні із застосуванням пластику PLA. PLA-пластик є біорозкладаним термопластичним полієфіром, що одержується на основі молочної кислоти – продукту переробки кукурудзи, цукрової тростини, крохмалю, целюлози. При нагріванні PLA пластик виділяє напівсолодкий запах.



Рис.174. Бобіни пластику

Програми створення 3D-моделей

Для створення 3D-моделі використовуються **САПР** (системи автоматизованого проектування) або **CAD** (computer-aided design), що підтримують управління 3D-друком:

- PTC Creo;
- AutoCAD – одна з найбільш потужних і зручних систем автоматизованого 2D і 3D проектування і документування;
- КОМПАС-3D – потужна система тривимірного проектування, що поєднує простоту освоєння і легкість в роботі;
- Blender – безкоштовний професійний пакет для створення тривимірної комп'ютерної графіки;
- SketchUp – програма для моделювання відносно простих тривимірних об'єктів – будівель, меблів, інтер'єру;
- Tinkercad – онлайн програма, забезпечує створення 3D-моделей в додатку, що працює в браузері і передачі їх на 3D-друк;
- Project Shapeshifter;
- 3D редактор;
- Leopoly.

Правила підготовки моделі до 3D-друку (FDM)

Сітка. Межі і ребра, що перетинаються, можуть призвести до непередбачуваних випадків слайсингу. Тому якщо модель складається з декількох об'єктів, то їх необхідно об'єднати в один. Потрібно пам'ятати, що не всі слайсери чутливі до сітки (наприклад, Slic3er). Якщо сітка крива, то виправити її допоможе безкоштовний сервіс netfabb.azurewebsites.net.

Плоска основа. Плоска основа допоможе моделі краще триматися на столі принтера. Якщо модель відклеїться (цей процес називають деламінація), то порушиться геометрія підставки моделі, а це може призвести до зміщення моделі. Якщо модель не має плоску основу або площа основи мала, то її друкують на рафті – надрукованій підкладці. Рафт псує поверхню моделі, з якою стикається. Тому за можливості краще обходитись без нього.

Товщина стінок. Стінки повинні бути рівними або товщими, ніж діаметр сопла. Інакше принтер просто не зможе їх надруку-

вати. Товщина стінки залежить від того, скільки периметрів буде друкуватися. Так, при 3 периметрах і соплі 0,5 мм товщина стінок повинна бути від 0,5, 1, 1,5, 2, 2,5, 3 мм. Тобто товщина стінки повинна бути кратна діаметру сопла, якщо вона менше $N * d$, де N – кількість периметрів, d – діаметр сопла.

Мінімум елементів, які не мають опори. Для кожного елемента, який не має опори, необхідна підтримуюча конструкція – підтримка. Чим менше таких елементів, тим менше підтримок потрібно, тим менше потрібно витратити матеріалу, часу друку. Крім того підтримка псує поверхню, дотичну з нею. Можна друкувати без підтримок стінок, які мають кут нахилу не більше 70 градусів.

Точність. Точність осей XY залежить від механіки принтера. І становить приблизно 0,3 мм для принтерів, створених власноруч. Точність по осі Z визначається висотою шару (0,1-0,4 мм). Звідси і висота моделі буде кратна висоті шару. Також необхідно враховувати, що після охолодження матеріал зменшується, а разом з цим змінюється геометрія об'єкта.

Вузькі місця. Вузькі місця дуже складно обробляти. За можливості необхідно уникати місць, що потребують складної обробки. Можна обробляти поверхню у ванні з розчинником, але тоді оплавлюються дрібні елементи.

Великі моделі. При моделюванні необхідно враховувати максимально можливі габарити друку принтера. Якщо модель більше цих габаритів, то її необхідно розрізати, щоб надрукувати по частинах. Оскільки ці частини будуть склеюватися, необхідно передбачити можливість їх з'єднання.

Розташування на робочому столі. Від того, як розташувати модель на робочому столі, залежить її міцність. Навантаження має розподілятися поперек шарів друку, а не вздовж. Інакше шари можуть розійтися, бо зчеплення між шарами не стовідсоткове. Від того, як прикладена сила щодо шарів, залежить міцність надрукованої деталі. В даному випадку для правої «Г» досить буде невеликої сили, щоб зламати її.

Формат файлу. Слайсери працюють з форматом файлу STL. Тому зберігати модель для друку потрібно саме в цьому форматі.

Налаштування параметрів друку у слайсері

Підготовка моделі до 3D-друку відбувається у слайсері – програмі для перетворення 3D-моделі в керуючий код для 3D-принтера. Модель «ріжеться» шарами. Кожен шар складається з периметра і / або заповнення. Модель може мати різний відсоток заповнення, також його може й не бути (пустотіла модель).

Типи слайсерів

- Simplify3D;
- Slic3r;
- Cura;
- Kisslicer.

Слайсери працюють з форматом файлу *STL*. Тому зберігати модель для друку потрібно саме в цьому форматі. Слайсер перетворює файл *STL* в *g-code*.

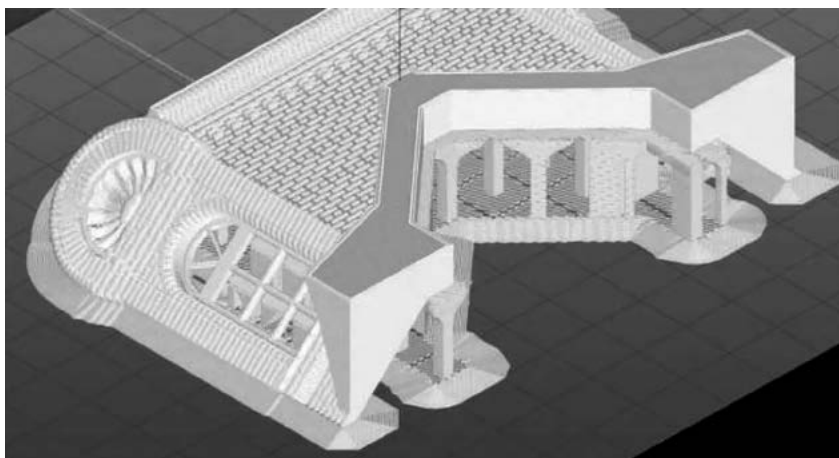
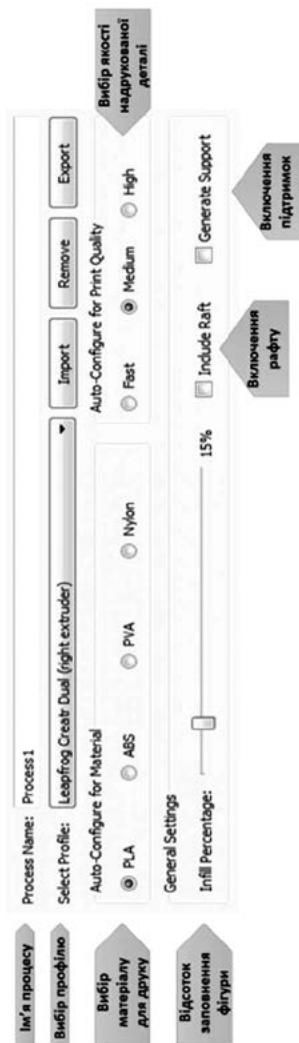


Рис.175. Візуалізація в *g-code*

КОРОТКИЙ ОПИС ВКЛАДОК НАЛАШТУВАННЯ ПАРАМЕТРІВ У ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ «SIMPLIFY3D»

Основні налаштування моделі



Вкладка «Extruder» (екструзія)

Extruder | Layer | Infill | Support | Temperature | Cooling | G-Code | Scripts | Other | Advanced

Left Extruder Toolhead

Overview

Extruder Toolhead Index: Tool 1

Nozzle Diameter: 0,35 mm

Extrusion Multiplier: 0,90

Extrusion Width: Auto Manual 0,40 mm

Діаметр сопла

Множник екструзії

Ширина розплавленого пластику

Втягування пластику

Контроль подачі пластику з соплу

Retraction

Retraction Distance: 4,20 mm

Extra Restart Distance: -0,40 mm

Retraction Vertical Lift: 0,00 mm

Retraction Speed: 1800,0 mm/min

Coast at End

Coasting Distance: 0,20 mm

Wipe Nozzle

Wipe Distance: 5,00 mm

Довжина втягування

Висота підйому сопла для переміщення його з одного місця друку в інше

Швидкість втягування

Включення втягування після закінчення друку

Включення втягування пластику сопла, яке витягло при розігріві платформи

Add Extruder

Remove Extruder

Hide Advanced

Select Models

OK

Cancel

Вкладка «Layer» (шар)

Шар Layer Infill Support Temperature Cooling G-Code Scripts Other Advanced

Layer Settings

Primary Extruder Right Extruder

Primary Layer Height 0,1000 mm

Top Solid Layers 3

Bottom Solid Layers 3

Outline/Perimeter Shells 2

Outline Direction: Inside-Out Outside-In

Print islands sequentially without optimization

Single outline confscrew printing mode (vase mode)

Start Points

Use random start points for all perimeters

Optimize start points for fastest printing speed

Choose start point closest to specific location

X: 0,00 Y: 300,00 mm

Назва екструдера

Висота шару

Кількість верхніх шарів моделі

Кількість нижніх шарів моделі

Кількість «оболонки» бокових частин (стінок) моделі

Напрямок друку шару

Режим послідовного друку без оптимізації

Решка моделі «прозрачна бочка» з товщеною стінкою в один провід сопла - друк по спіралі з безперервним підйомом по вертикальній осі Z.

Початкова точка друку

Довільна точка друку

Оптимальна точка для швидкого друку

Найближча точка до заданої

Layer Settings

First Layer Settings

First Layer Height 90 %

First Layer Speed 50 %

Налаштування першого шару

Висота першого шару

Швидкість друку першого шару

Raft Settings

Include Raft

Exclude Raft

Raft Layers 3

Raft Offset 3,00 mm

Raft Infill 85 %

Disable raft base layers

Включення рафту

Виключення рафту

Висота рафту

Відстань між рафту виступає за основу моделі

Щільність рафту

Skirt Settings

Include Skirt/Brm

Skirt Layers 1

Skirt Offset 4,00 mm

Skirt Outlines 2

Включення ліній обведення

Виключення ліній обведення

Налаштування ліній обведення моделі

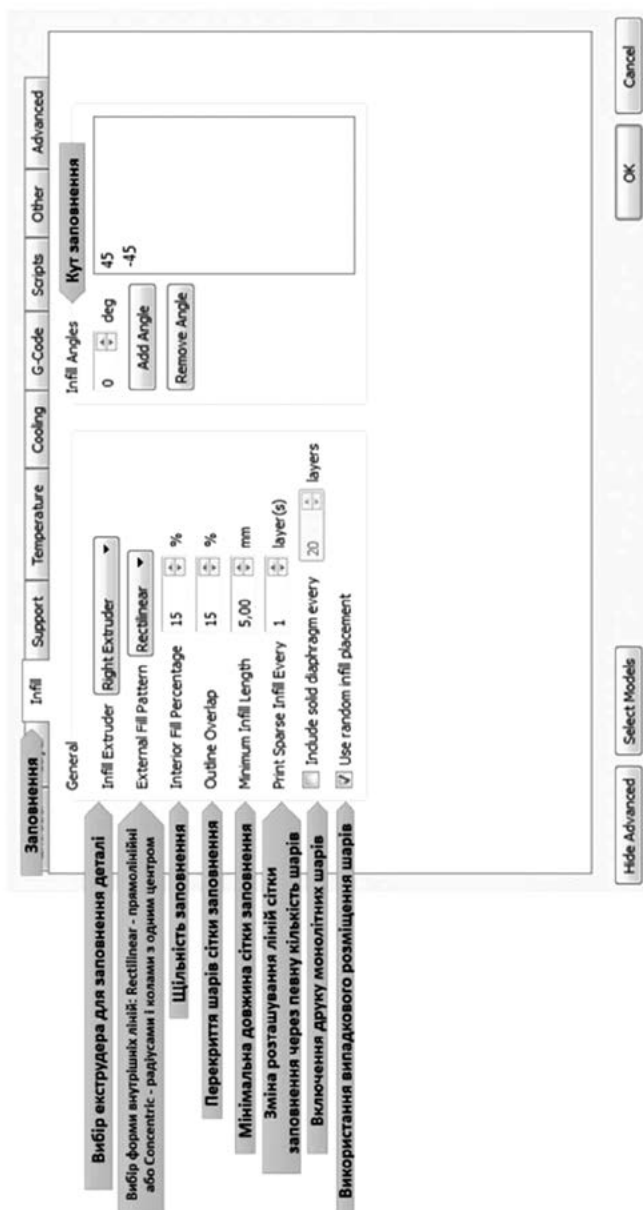
Кількість шарів

Відстань від моделі

Ширина обведення (кількість ліній)

Hide Advanced Select Models OK Cancel

Вкладка «Infill» (заповнення)



Вкладка «Support» (опори)

Включення автоматичного генерування підтримувачів

Вибір екструдера для друку підтримувачів

Щільність підтримувачів

Відстань між підтримувачами та моделлю

Друк шарів підтримки кожен ... шар

Параметри розділення підтримувачів на частини

Відстань через яку можливе горизонтальне розгалуження підтримувачів

Відстань через яку можливе вертикальне розгалуження підтримувачів догори

Відстань через яку можливе вертикальне розгалуження підтримувачів вниз

Автоматичне розміщення підтримувачів
Only used if manual support is not defined

Support Pillar Resolution: 4,00 mm

Max Overhang Angle: 45 deg

Розмір колонок підтримки

Максимальний кут вступу елементів з якого буде генеруватися підтримувач

Support Infill Angles: 0 deg

Кут заповнення підтримувачів

Support Material Generation

Generate Support Material

Support Extruder: Right Extruder

Support Infill Percentage: 30 %

Extra Inflation Distance: 0,00 mm

Print Sparse Support Every: 1 layer(s)

Separation From Part

Horizontal Offset From Part: 0,30 mm

Upper Vertical Separation Layers: 1

Lower Vertical Separation Layers: 1

Hide Advanced

Select Models

OK

Cancel

Вкладка «Temperature» (температура)

Temperature | Cooling | G-Co | Other | Advanced

Heated Bed Temperature

Overview

Temperature Identifier: T2

Temperature Controller Type: Extruder Heated build platform

Relay Temperature Between Each: Layer Loop

Wait for temperature controller to stabilize before beginning build

Per-Layer Temperature Setpoints

| Layer | Temperature | Number | Temperature |
|-------|-------------|--------|-------------|
| 1 | 60 | 1 | 200 °C |

Buttons: Add Setpoint, Remove Setpoint

Buttons: Add Temperature Controller, Remove Temperature Controller

Buttons: Hide Advanced, Select Models, OK, Cancel

Перелік списку налаштувань

Temperature Controller List (click item to edit settings)

Right Extruder
Left Extruder
Heated Bed

Вибір ідентифікатора температурного датчика

Тип температурного контролера: екструдер, стіл

Вибір того як часто буде відбуватися запит контролю температури: Layer - кожен шар або Loop - через кожен цикл

Вибір температури для кожного шару

Вилучення затримки перед початком друку для стабілізації температури

Вкладка «Other» (інше)

The screenshot displays the 'Other' tab of a 3D printing software interface, specifically the 'Filament Properties' section. The interface includes a top navigation bar with tabs: Extruder, Layer, Infill, Support, Temperature, Cooling, G-Code, Scripts, Other, and Advanced. The 'Other' tab is currently selected.

Filament Properties
 Filament diameter: 1.7500 mm (Діаметр пластика)

Speeds

| | | |
|------------------------------|--------|--------|
| Default Printing Speed | 3600,0 | mm/min |
| Outline Underspeed | 50 | % |
| Support Structure Underspeed | 80 | % |
| X/Y Axis Movement Speed | 4800,0 | mm/min |
| Z Axis Movement Speed | 1000,0 | mm/min |

Bridging

| | | |
|-------------------------------|------|-------|
| Unsupported area threshold | 50,0 | sq mm |
| Bridging extrusion multiplier | 100 | % |
| Bridging speed multiplier | 100 | % |

Callouts:

- Налаштування швидкості (Speeds adjustment)
- Загальна швидкість друку (General printing speed)
- Швидкість друку зовнішньої частини моделі (Model outer part printing speed)
- Швидкість друку підтримувачів (Support structure printing speed)
- Максимальна швидкість за всіма X та Y (Maximum speed for all X and Y)
- Максимальна швидкість за віссю Z (Maximum speed for Z axis)
- Буде-який-лише-елемент-моделі-має-розташовуватися-проміжком-між-ліній (Any element of the model must be located between lines)
- Множник-екструзії-друку-мостів (Bridging extrusion multiplier)
- Множник-швидкості-друку-мостів (Bridging speed multiplier)

Buttons: Hide Advanced, Select Models, OK, Cancel

ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

ПРОСКОП

1. Створіть програму, що дозволяє повертати робота в початкову позицію при зовнішньому впливі на нього.
Встановлюємо робота на початкову позицію, запускаємо програму. Потім вручну повертаємо робота на будь-який кут, натискаємо датчик дотику, і робот повинен повернутися в початкову позицію.
2. Написати програму, згідно якої робот обертається зі швидкістю, яка постійно збільшується, і виводить на блок EV3 даний кут і швидкість повороту робота.
3. Написати програму руху робота по квадрату з довжиною сторони квадрата, яка дорівнює довжині окружності колеса робота.

Датчик дотику

1. Встановіть на роботі два датчика дотику – один «дивиться» вперед, інший – назад.
Напишіть програму, щоб робот змінював напрямок руху на протилежний при зіткненні з перешкодою, при цьому:
 - При русі вперед опитується передній датчик.
 - При русі назад опитує задній датчик.
 - Не забудьте вказати різні порти для різних датчиків.
2. Напишіть програму, щоб робот вибрався з лабіринту, у якого фініш заздалегідь не визначений.
3. Напишіть програму, щоб робот вибрався з лабіринту.

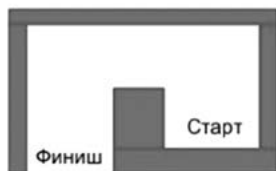


Рис. 176. Лабіринт
для робота

Дайте у звіті відповіді на питання:

- Що потрібно зробити робот після дотику до стінки?
- В який бік повинен крутитися мотор, щоб робот міг виконати розворот безперешкодно?
- Скільки разів робот повинен зробити однакові дії?

4. Написати програму, яка змушує робота рухатися вперед, при наїзді на перешкоду - від'їжджати назад, повертати вправо на 90 градусів і продовжувати рух вперед до наступної перешкоди.

5. «Робот-воїн».

Роботу знадобляться два датчика : відстані та дотику.

- «Воїн» крутиться швидко на одному місці навколо себе.
- Як тільки в його полі зору з'являється предмет ближче ніж 50 см, робот починає рухатися до цього предмету і зупиняється коли стукнеться об нього.
- Після удару, робот шукає нового супротивника.

Датчик кольору

1. «Робот-прибиральник»

Роботу знадобиться датчик відстані і датчик кольору. Завдання робота: виявити всередині рингу все сміття (сенсор відстані) і виштовхнути його за чорною лінію, що обмежує ринг. Сам робот не повинні виїжджати за межі рингу.

2. Написати програму, при якій пристрій видає один звук, коли до нього підносять предмет червоного кольору, і інший звук, коли підносять не червоний предмет.

3. Написати програму, яка змінює швидкість руху нашого робота в залежності від інтенсивності зовнішнього освітлення.

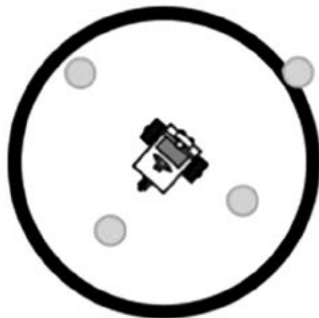


Рис. 177. Ринг
для робота

4. Написати програму для робота, що пересувається всередині кола, з чорним контуром за таким правилом:

- робот рухається вперед прямолінійно;
- досягнувши чорної лінії, робот зупиняється;
- робот від'їжджає назад на два оберти моторів;
- робот повертає вправо на 90 градусів;
- рух робота повторюється.

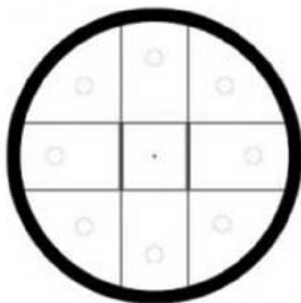


Рис. 178. Поле
для робота

5. Написати програму прямолінійного руху робота, який називає колір смуги, над якою він проїжджає. При досягненні чорної смуги робот промовляє «Стоп» і зупиняється.

Змагання роботів

Приклад завдання для організації змагань роботів.

Змагання «Траєкторія-пазл»

У цьому змаганні учасникам необхідно підготувати автономний робот, здатний проїхати від зони старту до зони фінішу по траєкторії, складеній з типових елементів, долаючи перешкоди.

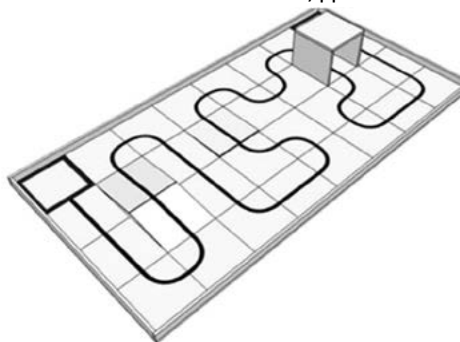


Рис. 179. Поле для змагання

Умови змагання

1. Робот повинен набрати максимальну кількість балів, рухаючись по чорній лінії, траєкторії від зони старту до зони фінішу.
2. Під час проведення спроби, учасники команд не повинні торкатися роботів.
3. Якщо під час заїзду робот з'їде з чорної лінії, тобто за нею опиняться всі колеса або інші деталі, то заїзд зупиниться і робот отримає бали, отримані до цього моменту.
4. Якщо під час заїзду робот стане рухатися неконтрольовано або не зможе продовжити рух протягом 20 секунд, то отримає бали, отримані до цього моменту.

Поле

1. Поле складається з секцій 300 x 300 мм на яких відзначена траєкторія по якій повинен слідувати робот.
2. Траєкторію може визначати чорна лінія на білому тлі або біла лінія на чорному тлі. Ширина лінії 25 мм.

3. Траєкторія може складатися з наступних секцій:

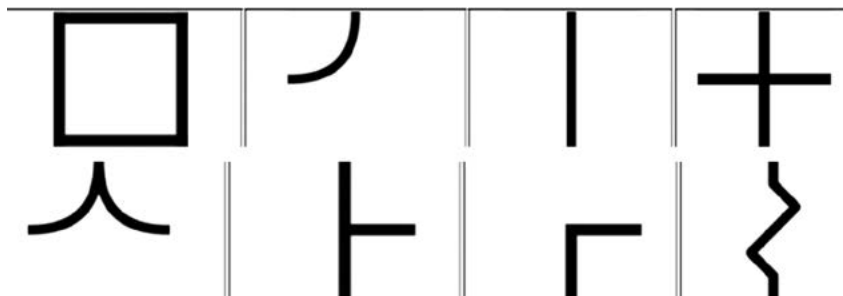


Рис. 180. Секції для поля

Можливі варіанти траєкторії руху:

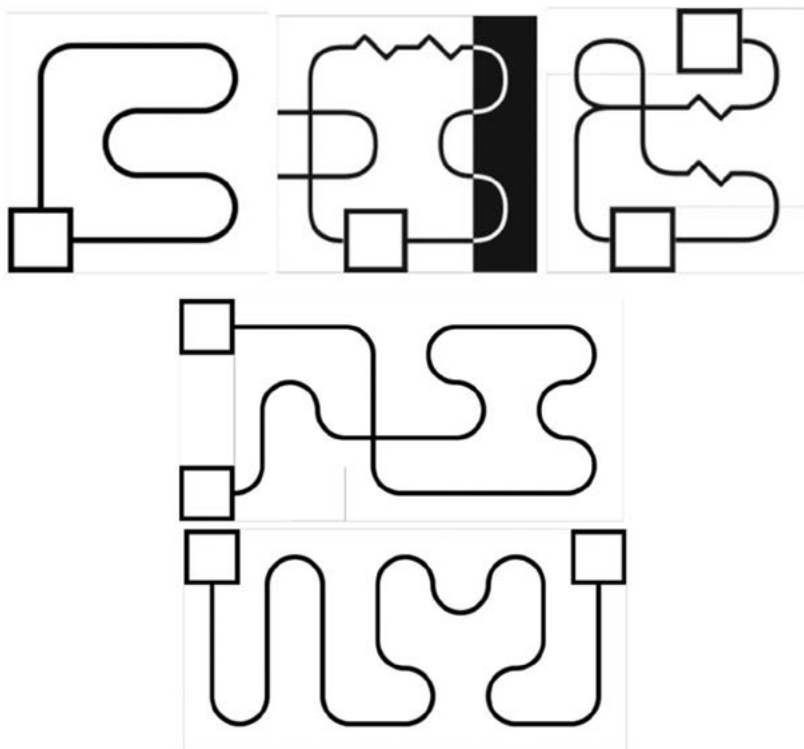


Рис. 181. Можливі траєкторії руху

4. На траєкторії можливе використання додаткових елементів: гірки, трамплінів, перешкод, тунелів, банки і т.д.

Робот

1. На роботів технічно не накладається обмеження на використання будь-яких комплектуючих, крім тих, які можуть якимось чином пошкодити поверхню поля.
2. Максимальні розміри роботи 250x250x250 мм.
3. Під час спроби робот може змінювати свої розміри, але без втручання людини.
4. Робот повинен бути автономним.
5. Робот, який на думку суддів може пошкодити покриття поля, буде дискваліфікований на весь час змагань.
6. Перед заїздом роботи перевіряються на габарити.

