

ДОДАТОК А

Скетч програми для Arduino

```

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

#include <Wire.h>
#include <Servo.h>
Servo up_down, left_right, front_back, clamp; // Створіть сервоприводний
об'єкт
int clmp = 45;
int frnt_bck = 45;
int lft_right = 90;
int up_dwn = 45;

#include <SoftwareSerial.h> // для роботи з Bluetooth на довільних пінах
SoftwareSerial bluetooth(1, 0); // RX, TX
char command = ' '; // Команда з Bluetooth

#include <DHT11.h>
DHT11 dht11(12);
int temperature = 0;
int humidity = 0;

#include <FastLED.h>
#define LED_PIN 13 // пін, до якого підключена стрічка
#define LED_COUNT 12
CRGB leds[LED_COUNT];
int currentLED = 0; // Змінна для поточного світлодіода

class Motor { // Створюємо клас для роботи з моторами
private:
    int pwmPin; // Пін для ШІМ
    int dirPin; // Пін для напрямку
public:
    // Конструктор
    Motor(int pwm, int dir) {
        pwmPin = pwm;
        dirPin = dir;
        pinMode(pwmPin, OUTPUT);
        pinMode(dirPin, OUTPUT);
    }
}

```

```

    stop(); //зупинити двигун при ініціалізації
}
// встановити швидкість (-255 до 255)
void setSpeed(int speed) {
    speed = constrain(speed, -255, 255); // Обмеження швидкості в діапазоні від -
255 до 255
    if (speed > 0) {
        digitalWrite(dirPin, LOW); // Напрямок вперед
        analogWrite(pwmPin, speed); // Установка швидкості
    } else if (speed < 0) {
        digitalWrite(dirPin, HIGH); // Напрямок назад
        analogWrite(pwmPin, -speed); // Установка швидкості
    } else {
        stop(); // Якщо швидкість 0, зупиніть двигун
    }
}

// Повна зупинка двигуна
void stop() {
    analogWrite(pwmPin, 0); // Вимкнути ШЫМ
    digitalWrite(dirPin, LOW);
}
};

```

```

Motor leftMotor(3, 4); // 3 - PWM, 4 - напрямок обертання двигуна
Motor rightMotor(11, 5);

```

```

// Переменные для таймера
unsigned long previousMillis_rgb = 0; // останній час оновлення
const long interval_rgb = 30; // Інтервал у мілісекундах (100 мс)
unsigned long previousMillis_lcd = 0;
const long interval_lcd = 200;

```

```

const int leftSideSensor = A3; // Лівий бічний датчик
const int leftFrontSensor = A2; // Передній лівий датчик
const int rightFrontSensor = A1; // Передній правий датчик
const int rightSideSensor = A0; // Правий бічний датчик
const int threshold = 500; // Порогове значення для датчиків

```

```
int cmode = 1;
```

```

void setup() {
    clamp.attach(10);
}

```

```

up_down.attach(9);
left_right.attach(8);
front_back.attach(7);

lcd.init();
lcd.backlight(); // Увімкніть підсвічування дисплея
lcd.print("LCD працює");

FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, LED_COUNT); //
ініціалізація стрічки
FastLED.clear(); // Гасимо всі світлодіоди
FastLED.show();

bluetooth.begin(9600); // ініціалізація Bluetooth
}

void loop() {
unsigned long currentMillis = millis(); // таймер реберної стрічки

if (currentMillis - previousMillis_rgb >= interval_rgb) { //Перевірте, чи ми
передаємо час для перемикання
previousMillis_rgb = currentMillis; // оновлення часу
leds[currentLED] = CRGB::Black; // гасити попередній світлодіод
leds[6 + currentLED] = CRGB::Black;
currentLED++; // перехід до наступного світлодіода
if (currentLED >= LED_COUNT) {
currentLED = 0; // Якщо ми дійшли до кінця, ми повертаємось до початку
}
leds[currentLED] = CRGB(255, 40, 0); // Запалюємо світлодіод
помаранчевого кольору
leds[6 + currentLED] = CRGB(255, 50, 50); // малиновий колір (RGB)
FastLED.show();
}

if (currentMillis - previousMillis_lcd >= interval_lcd) { //таймер для обробки
температури датчика вологи
previousMillis_lcd = currentMillis;

int result = dht11.readTemperatureHumidity(temperature, humidity);
if (result == 0) {
lcd.clear();
lcd.setCursor(1, 0);
String s = String("Тепло : ") + temperature + String(" °C");
lcd.print(s);
}
}
}

```

```

    lcd.setCursor(1, 1);

    s = String("Волога: ") + humidity + String(" % ");
    lcd.print(s);
}

}
// Значення читання з ІЧ -датчиками
int leftFrontValue = analogRead(leftFrontSensor);
int rightFrontValue = analogRead(rightFrontSensor);
int leftSideValue = analogRead(leftSideSensor);
int rightSideValue = analogRead(rightSideSensor);

// Перевірка наявності команди за допомогою Bluetooth
if (bluetooth.available() > 0) {
    command = bluetooth.read();
}
// Обробка команд Bluetooth
if (command != ' ') {
    if (command == '2' && cmode == 2) {cmode = 1; command = ' '; }
    if (command == '2' && cmode != 2) {cmode = 2; command = ' '; }
    if (command == '3' && cmode == 3) {cmode = 1; command = ' '; }
    if (command == '3' && cmode != 3) {cmode = 3; command = ' '; }

    if (cmode == 1) { TruckCommand(command); }
    if (cmode == 2) { ManipulatorCommand(command); }

    if (command == '0') { command = ' '; }
    return;
}

// Автономний рух (обхід перешкод)
if (cmode == 3) {
    if (leftFrontValue < threshold && rightFrontValue < threshold) {
        turnAround();
    } else if (leftFrontValue < threshold) {
        turnRight();
    } else if (rightFrontValue < threshold) {
        turnLeft();
    } else if (leftSideValue < threshold) {
        moveRight();
    } else if (rightSideValue < threshold) {
        moveLeft();
    } else {

```

```

    moveForward();
}
}
}

// Функції керування моторами
void moveForward() {
    leftMotor.setSpeed(150);
    rightMotor.setSpeed(150);
}
void moveBackward() {
    leftMotor.setSpeed(-150);
    rightMotor.setSpeed(-150);
}
void turnLeft() {
    leftMotor.setSpeed(-100);
    rightMotor.setSpeed(100);
}
void turnRight() {
    leftMotor.setSpeed(100);
    rightMotor.setSpeed(-100);
}
void turnAround() {
    leftMotor.setSpeed(-150);
    rightMotor.setSpeed(-150);
    delay(1000);
    leftMotor.setSpeed(-100);
    rightMotor.setSpeed(100);
    delay(1000); // Час повороту
}
void moveLeft() {
    leftMotor.setSpeed(50);
    rightMotor.setSpeed(150);
}
void moveRight() {
    leftMotor.setSpeed(150);
    rightMotor.setSpeed(50);
}
void stopMotors() {
    leftMotor.setSpeed(0);
    rightMotor.setSpeed(0);
}
//Обробка команд Bluetooth для візка
void TruckCommand(char cmd) {

```

```

switch (cmd) {
  case 'F':moveForward(); break; // Вперед
  case 'B': moveBackward();break; // Назад
  case 'L':turnLeft();break; // повернути праворуч
  case 'R':turnRight();break; // повернути праворуч
  default:stopMotors();break;
}
}
// Обробка команд Bluetooth для маніпулятора
void ManipulatorCommand(char cmd) {
  switch (cmd) {
    case 'F': frnt_bck += 3; break; // Вперед
    case 'B': frnt_bck -= 3; break; // Назад
    case 'L': lft_rght += 3; break; // повернути ліворуч
    case 'R': lft_rght -= 3; break; // повернути праворуч
    case 'U': up_dwn += 3; break; // вгору
    case 'D': up_dwn -= 3; break; // вниз
    case 'O': clmp++; break; // Відкрити
    case 'C': clmp--; break; // Закрити
  }
  clamp.write(clmp);
  up_down.write(up_dwn);
  left_right.write(lft_rght);
  front_back.write(frnt_bck);
  delay(50);
}

```

ДОДАТОК Б

Публікація. Розробка реконфігурованого мобільного робота

УДК 621.321

РОЗРОБКА РЕКОНФІГУРОВАНОГО МОБІЛЬНОГО РОБОТА

Дмитрієв Д.В.

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки, 14

E-mail: danylo.dmytriiev@nure.ua

Анотація: Метою цієї статті є дати огляд технологій, який можна використовувати для впровадження робототехніки до освітнього закладу. У цій статті розглядаються як комплексні робототехнічні системи так і проекти, які реалізують лише певні елементи робототехнічної системи, такі як електроніка, апаратне забезпечення або програмне забезпечення. Зроблено висновки про актуальність приладу сьогодні.

Ключові слова: "Зроби сам"; Освіта; Відкрите джерело; Апаратне забезпечення; Проектне навчання; Швидке прототипування; Робототехніка.

DEVELOPMENT OF A RECONFIGURED MOBILE ROBOT

D.Dmitriiev

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, pr. Nauki, 14

E-mail: danylo.dmytriiev@nure.ua

Abstract: The purpose of this article is to provide an overview of technologies that can be used to implement robotics in an educational setting. This article considers both complex robotic systems and projects that implement only certain elements of a robotic system, such as electronics, hardware, or software.

Keywords: DIY; Education; Open Source; Hardware; Project-Based Learning; Rapid Prototyping; Robotics.

Сучасне виробництво та індустріальні процеси відзначаються постійними змінами та швидкими технологічними розвитками. Для ефективного використання робочих ресурсів та підвищення продуктивності, підприємства повинні шукати інноваційні рішення, зокрема в області автоматизації. Однак існує проблема несталість вимог до роботизованих систем. Несталість вимог може виникати через різноманітні фактори, такі як зміни у виробничому процесі, введення нових завдань чи технологічні інновації. Існує потреба у гнучких підходах до автоматизації, які дозволяють системам швидко адаптуватися до змінних умов. Одним з ключових напрямків розв'язання цієї проблеми є розробка реконфігурованих мобільних роботів.

Створення роботів – популярний вибір проекту для впровадження проблемно-орієнтованого навчання (PBL) у класи. Причиною, чому це такий популярний вибір, пояснюється багатопрофільністю теми: робототехніка вимагає безлічі різних наукових, технічних і технологічних навичок, як фізика, електроніка, математика та програмування. Створення роботів є популярним вибором проекту для реалізації проблемно-орієнтованого навчання (PBL) у класах. математика та програмування. Це ідеальний предмет, тому що з ним можна пов'язати так багато різних. курсів. Крім того, самі роботи захоплюють увагу дітей та підлітків, забезпечуючи натхнення та мотивацію.

Підхід PBL загалом і використання робототехніки в освіті зокрема мають низку інших відмінностей від більш традиційних способів навчання. PBL учні навчаються справлятися з цими реальними проблемами, використовуючи творче вирішення проблем, важливі навичка

УДК 621.321

РОЗРОБКА РЕКОНФІГУРОВАНОГО МОБІЛЬНОГО РОБОТА**Дмитрієв Д.В.**

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки, 14

E-mail: danylo.dmytriiev@nure.ua

Анотація: Метою цієї статті є дати огляд технологій, який можна використовувати для впровадження робототехніки до освітнього закладу. У цій статті розглядаються як комплексні робототехнічні системи так і проекти, які реалізують лише певні елементи робототехнічної системи, такі як електроніка, апаратне забезпечення або програмне забезпечення. Зроблено висновки про актуальність приладу сьогодні.

Ключові слова: "Зроби сам"; Освіта; Відкрите джерело; Апаратне забезпечення; Проектне навчання; Швидке прототипування; Робототехніка.

DEVELOPMENT OF A RECONFIGURED MOBILE ROBOT**D.Dmitriiev**

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, pr. Nauki, 14

E-mail: danylo.dmytriiev@nure.ua

Abstract: The purpose of this article is to provide an overview of technologies that can be used to implement robotics in an educational setting. This article considers both complex robotic systems and projects that implement only certain elements of a robotic system, such as electronics, hardware, or software.

Keywords: DIY; Education; Open Source; Hardware; Project-Based Learning; Rapid Prototyping; Robotics.

Сучасне виробництво та індустріальні процеси відзначаються постійними змінами та швидкими технологічними розвитками. Для ефективного використання робочих ресурсів та підвищення продуктивності, підприємства повинні шукати інноваційні рішення, зокрема в області автоматизації. Однак існує проблема несталість вимог до роботизованих систем. Несталість вимог може виникати через різноманітні фактори, такі як зміни у виробничому процесі, введення нових завдань чи технологічні інновації. Існує потреба у гнучких підходах до автоматизації, які дозволяють системам швидко адаптуватися до змінних умов. Одним з ключових напрямків розв'язання цієї проблеми є розробка реконфігурованих мобільних роботів.

Створення роботів – популярний вибір проекту для впровадження проблемно-орієнтованого навчання (PBL) у класи. Причиною, чому це такий популярний вибір, пояснюється багатопрофільністю теми: робототехніка вимагає безлічі різних наукових, технічних і технологічних навичок, як фізика, електроніка, математика та програмування. Створення роботів є популярним вибором проекту для реалізації проблемно-орієнтованого навчання (PBL) у класах. математика та програмування. Це ідеальний предмет, тому що з ним можна пов'язати так багато різних курсів. Крім того, самі роботи захоплюють увагу дітей та підлітків, забезпечуючи натхнення та мотивацію.

Підхід PBL загалом і використання робототехніки в освіті зокрема мають низку інших відмінностей від більш традиційних способів навчання. PBL учні навчаються справлятися з цими реальними проблемами, використовуючи творче вирішення проблем, важливий навичка

реального життя. Технічним навичкам підхід PBL також дозволяє учням вивчати важливі соціальні навички, такі як комунікація, лідерство, планування та співробітництво.

Хоча PBL з використанням робототехніки пропонує перспективну альтернативу традиційним методам навчання, його широкомасштабне впровадження в освіту дійсно створює кілька проблем. Всього описують 5 великих проблем:

- «Нестача часу у вчителів»
- «Брак підготовки вчителів»
- «Нестача відповідних за віком навчальних матеріалів»
- «Брак готових до використання навчальних матеріалів»
- «Брак низки доступних платформ робототехніки».

Історично склалося так, що робототехніка в освіті зазвичай впроваджується, використовуючи готові комплекти робототехніки, що включають все необхідне побудови функціонального робота. Хоча це відмінний спосіб стати і працює швидко, у нього є кілька недоліків.

- Комплекти «все в одному» зазвичай дорожчі.
- Ці комплекти складно з'єднати з іншими компонентами, як-от стандартизовані компоненти, компоненти, виготовлені третіми особами, або стандартними датчиками та виконавчими механізмами.
- Знайти або купити запасні частини не завжди легко.
- Не всі компоненти використовуються або необхідні для роботи, тому ви платите за компоненти, які вам не потрібні.

При цьому ці системи є чудовою відправною точкою, оскільки вони надають набір сумісних будівельних блоків та електроніка, програмне забезпечення для простого (графічного) програмування, інструкції та мережу спільноти.

Mindstorms (рис.1) - лінійка продуктів компанії Lego, що надає необхідні інструменти для створення простих роботів за допомогою кубиків Lego.

Mindstorms побудований на основі програмованого мікрокомп'ютерного блоку, який може керувати трьома двигунами і зчитувати з чотирьох датчиків. Двигуни та датчики можна підключити до блоку управління за допомогою з'єднувальних роз'ємів, тому не потрібні спеціальні навички для складання функціонального набору електроніки робота. Сам програмований блок можна запрограмувати за допомогою графічної мови програмування NXT-G, яка йде в комплекті з наборами. Альтернативно, блок може бути запрограмований з використанням однієї з багатьох доступних сторонніх програм, які забезпечують підтримку таких мов, як C++ чи Java. Для складання конструктивних частин робота використовують кубики в стилі Lego Technic. Через це можна легко вставити блоки з інших наборів Lego для розширення потенційного рівня складності роботів.

Простота використання, розмір спільноти Mindstorms та знайомство з кубиками Lego робить його популярним вибором як платформи для робототехніки в освіті, особливо при роботі з дітьми молодшого віку. Популярність Mindstorms виражається безліччю ресурсів, доступних для викладачів, таких як безліч книг, інтернет-спільноти та семінари, побудовані на основі екосистеми Mindstorms.

Одним із головних недоліків Lego Mindstorms є вартість. Інша проблема полягає в тому, що програмований блок обмежений максимум 3 двигунами та 4 датчиками. При додаванні ще одного двигуна або датчика до роботи стикаєшся з фізичною межею того, що можуть зробити роботи Mindstorms. Також складніше інтегрувати компоненти робототехніки сторонніх виробників у роботі Mindstorms.



Рисунок 1 - Lego Mindstorms NXT.

Система Vex Robotics схожа на Lego Mindstorms у тому, що це платформа, яка надає всі необхідні елементи, необхідні створення функціонального робота: структурні компоненти, електроніка, програмне забезпечення. Основна відмінність полягає в тому, що, хоча Mindstorms має тенденцію бути схожою на іграшку, підходить для дітей молодшого віку. Натомість Vex вибирає серйозніший підхід до робототехніки, орієнтований на учнів старших класів та дорослих.

В якості елементів конструкції використовуються пластикові та перфоровані металеві балки. З'єднання виконуються за допомогою гайок і гвинтів замість замикань і фрикційних з'єднань. Vex пропонує два різні варіанти мікроконтролерів для використання з її продуктами: мікроконтролер PIC і мікроконтролер Cortex, обидва з яких можна запрограмувати з використанням письмового коду, на відміну від графічної мови програмування, що використовується в Mindstorms.

Платформа Vex більше відповідає «справжній» інженерній практиці, ніж Lego Mindstorms, і пропонує більшу ступінь гнучкості. Така гнучкість робить його більш складним у використанні, це означає, що ця платформа найкраще підходить для студентів старшого віку та дорослих. Головним недоліком цієї платформи є висока вартість продуктів і ступінь складності, що робить його менш підходящим для дітей.

Однією з основних вимог до функціонального робота є електронна система, яка може зчитувати датчики, обробляти інформацію та керуючі виходи.

Розроблена в 2005 році платформа Arduino являла собою комбінацію плати мікроконтролера, набір бібліотек C++ та інтегроване середовище розробки (IDE), спрямовану на створення мікроконтролерів, доступних художникам та дизайнерам.

Плати Arduino (рис.2) не призначені спеціально для використання у робототехніці, але цю функціональність можна додати за рахунок використання додаткових модулів. Ці модулі можна прикріпити поверх плати Arduino і забезпечте її додатковими функціональними можливостями, такими як дисплей або схема прямого керування постійним струмом двигуна.

Наявність великої кількості модулів та плат, простота використання, особливо в порівнянні з іншими мікроконтролерами, та низька вартість сприяли створенню Екосистеми Ардуїно.



Рисунок 2 - Arduino Leonardo.

Часто роботи програмується за допомогою низькорівневої текстової програми - мови програмування, наприклад, С. Ці текстові мови можуть бути досить складним завданням для людей без попереднього досвіду. Важко перевести концепції людської мови в алгоритми. Потрібно подбати про те, щоб синтаксис програми був правильним. Графічна мова програмування може полегшити проблему, одночасно забезпечуючи інтерфейс, який привабливіший для дітей. Scratch (рис.3), розроблений у MIT Media Lab у 2006 році, є графічною мовою програмування, призначеною для навчання дітей принципам програмування у вигляді створення простих ігор та інтерактивного кіно. Язык програмування Google Blockly (рис.4) відрізняється з інших графічних мов програмування. Blockly можна розглядати як набір бібліотек, які значно полегшують розробку Scratch-подібної мови програмування. Blockly написано на Javascript і призначено для роботи в середовищі веб-браузера. Використовуючи блочний інтерфейс прикладного програмування (API), можна легко визначити свій власний набір блоків, щоб створити повністю індивідуальну графічну мову програмування. Відмінною особливістю Blockly є те, що він може автоматично перевести програму Blockly в письмовий код, що читається. Мовні генератори для Javascript та Python вже існують. Проте з допомогою API можна створювати власні генератори.

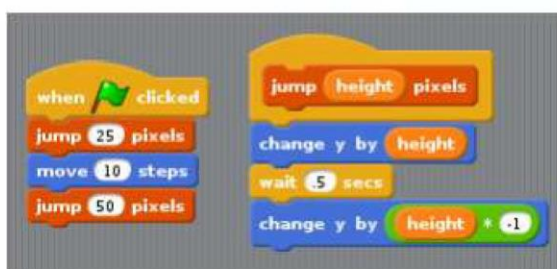


Рисунок 3 – Scratch.

ArduBlock - це плагін для Arduino. Він є інтегрованим інструментом, який дозволяє писати програми для Arduino, використовуючи той самий стиль графічного оформлення блоків як Scratch та Blockly. Крім блоків, які мають дослівний переклад функцій бібліотеки

Arduino, це також надає деякі зумовлені блоки для роботи зі сторонніми компонентами Arduino, таких як реле або джойстик. Це чудовий інструмент для використання в освіті, особливо коли студенти вже познайомилися з аналогічною мовою, такими як Scratch.

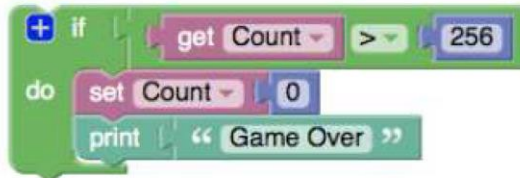


Рисунок 4 – Blockly.

Виробники роботів-аматорів, як правило, покладаються на широкий спектр методів, коли справа доходить до побудови фізичного втілення своїх роботів.

MakeBlock – (рис.5) це система комерційного виробництва, спеціально націленого на створення роботів. Система MakeBlock побудована на основі декількох різних типів алюмінієвих балок, які з'єднуються разом за допомогою стандартних гвинтів.

Система MakeBlock – чудовий спосіб зробити дуже жорсткого робота. Різьбові отвори дозволяють легко підключати сторонні компоненти до рами. Система використовує один і той же отвір. Відстань між отворами, як у кубиків Lego Technic для покращення сумісності. Основні недоліки MakeBlock – це відносно висока вартість та обмеженість доступності.

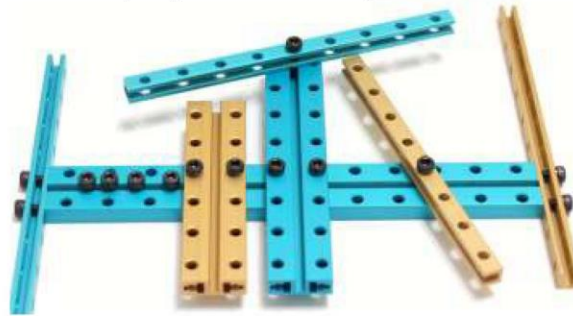


Рисунок 5 – MakeBlock.

Методологія DIY обіцяє перетворити освіту зі спостереження та слухання студентів на активну участь за допомогою оригінальних інтерактивних практичних уроків під керівництвом інструкторів, доповнених прикладами та обладнанням, які можна виготовити лише за необхідності, що дозволяє проводити персоналізовані та настроювані заняття та індивідуальне навчання в рамках загальних рамок. Ситуація полягає в тому, що педагоги обирають платформу робототехніки «все в одному», тому що вони не знають жодних альтернатив або тому, що вони не можуть знайти чітку відправну точку для альтернативних платформ. І хоча роботи, створені за допомогою набору «все в одному», можуть працювати краще, учні, які створюють свого власного робота або повністю з нуля, або комбінуючи елементи з різних систем, як описано вище, зазвичай отримують набагато глибше розуміння технологій, інженерії та творчого вирішення проблем. Групи класів, що використовують повну платформу робототехніки, також зазвичай залишаються в рамках цієї платформи, додавання стороннього або саморобного компонента відбувається нечасто. Роботи, створені з

використанням повної платформи, як правило, мають більшу функціональність і зазвичай їх легше побудувати, у той час як створення роботів з нуля, як правило, потребує більшого досвіду, але також дає учням більше розуміння.

Новий спосіб навчання - методика DIY обіцяє перетворити освіту зі спостереження за учнями в їх активну участь за допомогою оригінальних інтерактивних практичних уроків під керівництвом інструкторів та доповнений прикладами та обладнанням, які можна виготовляти лише за необхідності, що дозволяє персоналізувати та налаштовувати класи та індивідуальне навчання у загальних рамках.

Висновок. В результаті дослідження визначені продукти, які можна використовувати як інструменти для реалізації проектів робототехніки в освіті. Категорії, які були розглянуті, є комплексні one-робототехнічні платформи, електроніка, апаратне та програмне забезпечення. Взагалі, є два способи побудувати робота: або використовуючи повну робототехнічну платформу або створивши робота з нуля. Повні системи простіше у використанні, дозволяють більш швидкі результати та краще підходять для молодих студентів. Недоліком є те, що вони, як правило, дорожчі. Навпаки, створення робота з нуля це набагато складніше і більше підходить для старшокласників, але дає набагато краще розуміння технології, є більш гнучким і може бути дешевшим. В останні роки створення робота з нуля стало набагато простіше завдяки численним проектам та продуктам, які реалізують певні аспекти робота, такі як апаратне забезпечення, програмне забезпечення або електроніка. Ці продукти та проекти часто можуть бути пов'язані з недавніми тенденціями DIY та Maker Movement.

Ці тенденції характеризуються використанням верстатів із ЧПУ та співробітництво через Інтернет для створення фізичних апаратних проектів. Можна зробити висновок, що DIY і Maker субкультура може вплинути на освіту, оскільки вона не тільки заохочує інтерес молодих людей до STEM-пов'язаних областей, це також сприяє творчості та технологічному володінню. Всі ці навички, безсумнівно, матимуть завтра життєво важливе значення у суспільстві.

Модель навчання, яка практикується досі, розглядає студентів як пасивних учасників навчально-педагогічного процесу. Робототехніка в освіті є стратегічною альтернативою, яка може залучити студентів і зробити їх провідними в цьому процесі. Важливим стимулом, виховним та навчальним фактором є участь в олімпіадах, конкурсах, фестивалях, змаганнях з робототехніки.

ЛІТЕРАТУРА

1. J. Johnson, "Children, robotics, and education," *Artificial Life and Robotics*, vol. 7, pp. 16-21, 2003. W. Sunthonkanokpong, "Future Global Visions of Engineering Education," *Procedia Engineering*, vol. 8, pp. 160-164, 2011.
2. S. Bell, "Project-based learning for the 21st century: Skills for the future," *The Clearing House*, vol. 83, pp. 39-43, 2010. Матарик и др.
3. M. J. Mataric, N. Koenig, and D. Feil-Seifer, "Materials for enabling hands-on robotics and STEM education," 2007.
4. D. Li, "Getting Started with Arduino in ArduBlock: Example 01," <http://blog.ardublock.com/wp-content/uploads/2012/02/untitled1.jpg>, Ed., ed, 2012. Scratch, "Scratch 2.0: Create your own block,"
5. D. Li. (2012). ArduBlock. Available: ardublock.com
6. Arduino. (2013). Arduino. Available: <http://arduino.cc>
7. N. Fraser. (2013). blockly - A visual programming editor. Available:

<http://code.google.com/p/blockly/>

8. ShenzhenHuluRoboticTechnology. (2012). Makeblock - Aluminum Robot Kit. Available: <http://makeblock.cc/>

9. MIT. (2012, September 10). Scratch | Home | imagine, program, share. Available: <http://scratch.mit.edu/>.

10. Abu-Jassar, A. T., Attar, H., Yevsieiev, V., Amer, A., Demska, N., Luhach, A. K., & Lyashenko, V. (2022). Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks. Computational Intelligence and Neuroscience, 2022, Article ID 5866922. <https://doi.org/10.1155/2022/5866922>.

11. Lyashenko, V., Abu-Jassar, A.T., Yevsieiev, V., Maksymova, S. Automated Monitoring and Visualization System in Production, Int. Res. J. Multidiscip. Technovation, 5(6) 2023 09-18. <https://doi.org/10.54392/irjmt2362>

12. Amer Abu-Jassar, Vladyslav Yevsieiev, & Svitlana Maksymova. (2024). The Optical Flow Method and Graham's Algorithm Implementation Features for Searching for the Object Contour in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(3), 64–75.

13. Vladyslav Yevsieiev, Samariddin, S. M., Nikolay Starodubtsev, & Amer Abu-Jassar. (2024). ACTIVE CONTOURS METHOD IMPLEMENTATION FOR OBJECTS SELECTION IN THE MOBILE ROBOT'S WORKSPACE. Journal of Universal Science Research, 2(2), 135–145.

14. Svitlana Maksymova, Vladyslav Yevsieiev, & Ahmad Alkhalailah. (2024). The Monitoring System Architecture Development. Journal of Universal Science Research, 2(1), 69–79.

15. Andrii Bondariiev, Svitlana Maksymova, & Vladyslav Yevsieiev. (2023). Automated Monitoring System Development for Equipment Modernization. Journal of Universal Science Research, 1(11), 6–16. Retrieved from <https://universalpublishings.com/index.php/jusr/article/view/2484>

16. Yevsieiev, V. Comparative Analysis of the Characteristics of Mobile Robots and Collaboration Robots Within INDUSTRY 5.0. / V. Yevsieiev, D. Gurin // In the VI International Scientific and Theoretical Conference, September 8, 2023. Chicago, USA. P.92-94

17. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. Technical science research in Uzbekistan, 2(2), 115- 124.

18. Gurin, D., & et al. (2024). Effect of Frame Processing Frequency on Object Identification Using MobileNetV2 Neural Network for a Mobile Robot. Multidisciplinary Journal of Science and Technology, 4(8), 36-44.

19. Yevsieiev, V., & et al. (2024). Building a traffic route taking into account obstacles based on the A-star algorithm using the python language. Technical Science Research In Uzbekistan, 2(3), 103-112

20. Gurin, D., & et al. (2024). MobileNetv2 Neural Network Model for Human Recognition and Identification in the Working Area of a Collaborative Robot. Multidisciplinary Journal of Science and Technology, 4(8), 5-12.

21. Yevsieiev, V., & Uskov, S. (2024). *Development of the Layout Concept of a Small-Dimensioned Mobile Robot With Increased Accessibility* (Doctoral dissertation, International Scientific Unity).

22. Yevsieiev, V., & Gurin, D. (2024). *New Concepts of Human Interactions and Collaborative Robot-Manipulators in the Concepts of Industry 5.0* (Doctoral dissertation, Collection of scientific papers «SCIENTIA»).

Науковий керівник: Євсєєв В.В., професор кафедри КІТАР, доктор технічних наук, професор

ДОДАТОК В

Демонстраційний матеріал у вигляді презентації

