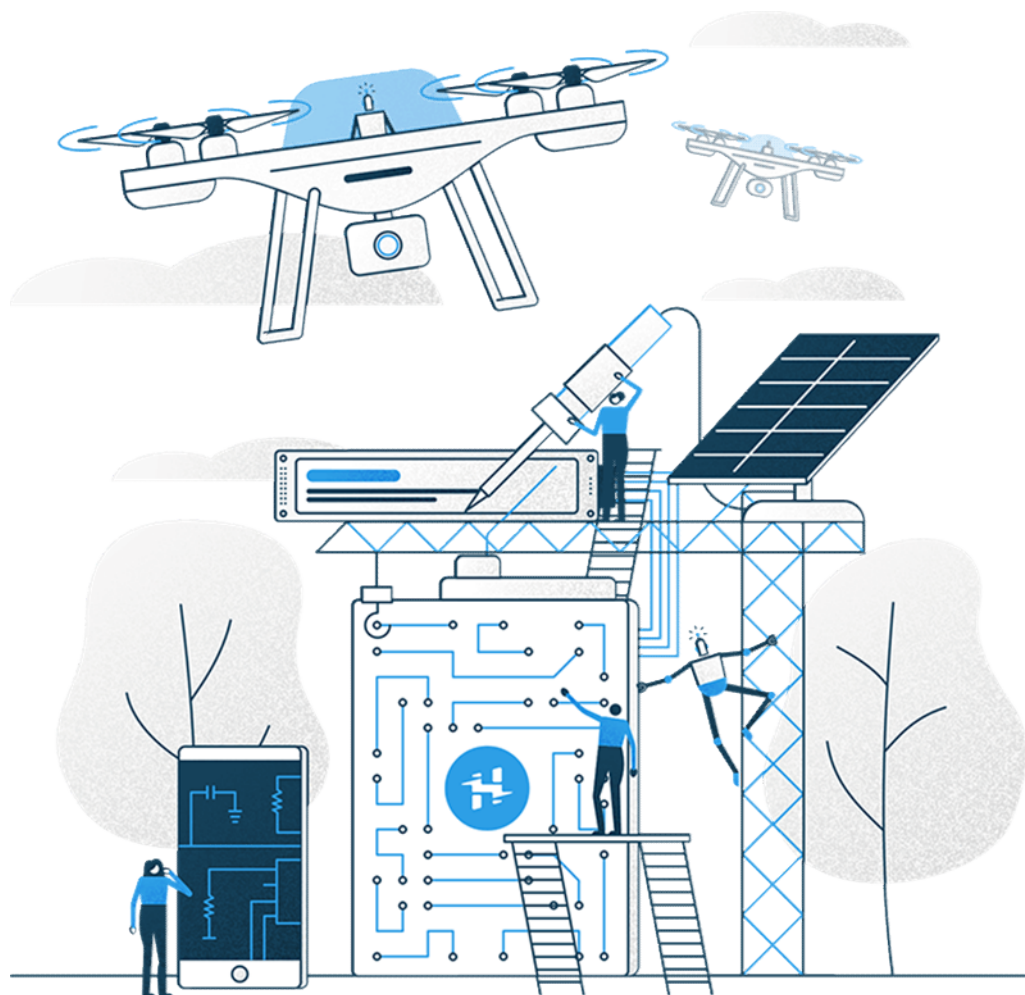


МІКРОПРОЦЕСОРНІ СИСТЕМИ УПРАВЛІННЯ



Lesson 8

Периферійний послідовний інтерфейс UART

У Ардуіно реалізована апаратна підтримка інтерфейсу послідовної передачі даних через порти 0, 1. Периферійний послідовний порта UART мікроконтролера ATmega 328 використовується для обміну інформацією з BT, WiFi модулем, ультразвуковим датчиком HC-SR04, ємнісним сенсором відбитків пальців Waveshare, датчиком CO2 K-33.

Його основні можливості: широкий діапазон швидкостей обміну даними; висока швидкість передачі при низькій частоті XTAL; 8 або 9-розрядний формат даних; виявлення помилок утрати даних при прийомі; виявлення помилок формату кадрів; виявлення помилкового стартового біта; три окремих переривання: по завершенню передачі, по порожньому регістрі передавача і по завершенню прийому.

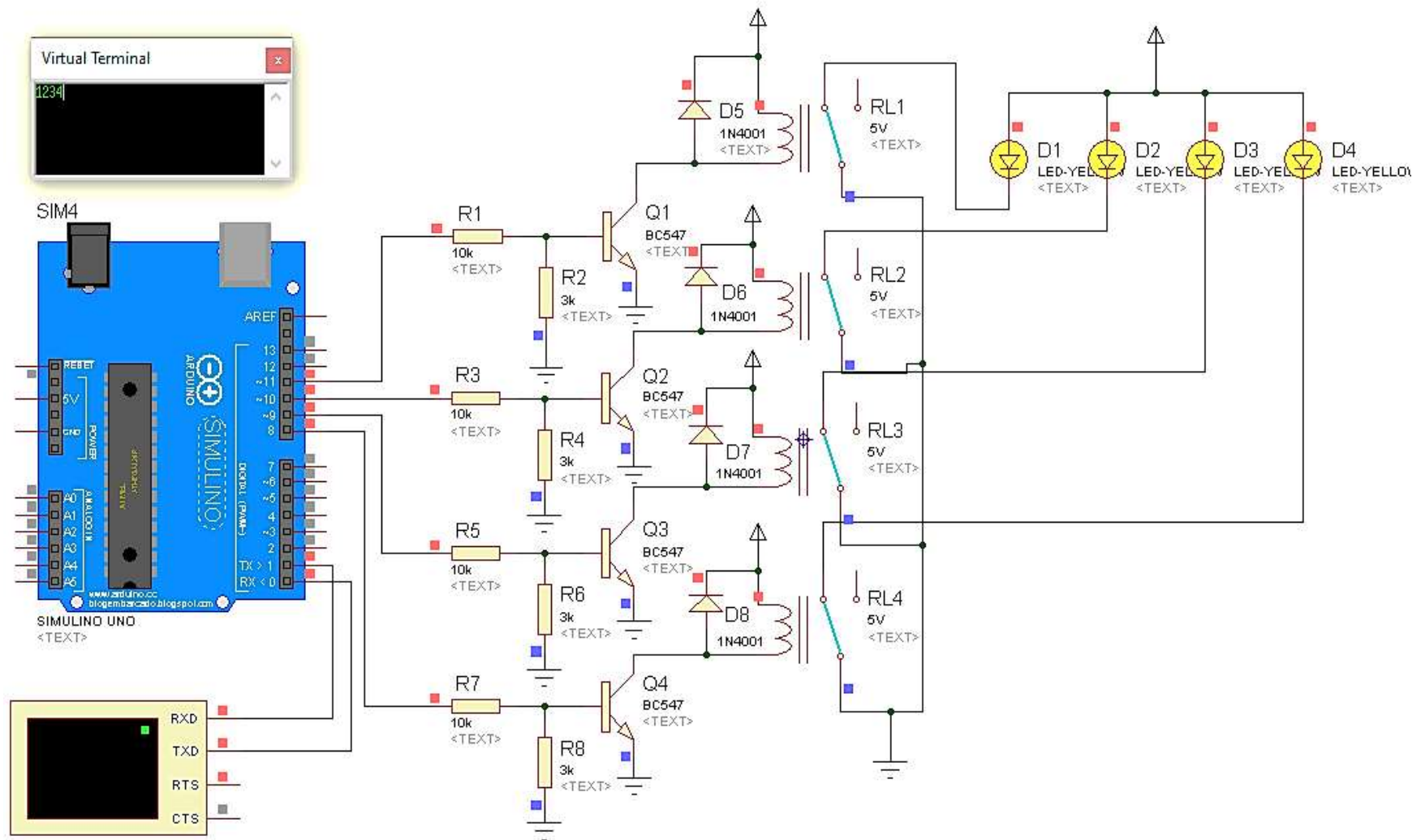
При передачі модуль UART додає до вхідного символу (8 або 9 біт) на початку — старт-біт (нуль), а в кінці — стоп-біт (одиниця), формуючи таким чином 10- або 11-бітову послідовність. Отримані значення передаються до регістра зсуву, який по черзі передає біти на вихід передавача TXD (вивід PD1). Швидкість видачі біт на вихід передавача визначається параметром baud rate (швидкість передачі інформації; вимірюється в бодах), яким можна керувати.

Периферійний послідовний інтерфейс UART

Приймач модуля UART безперервно перевіряє стан входу RXD, на якому за відсутності даних встановлюється рівень «1». Приймач зчитує інформацію з входу в 16 разів швидше. При виявленні на виводі RXD рівня «0» (тобто можливого старт-біта) мікроконтролер пропускає шість відліків, а потім робить три вибірки. Ці вибірки доводяться на відлік 8, 9 і 10 для кожного біта, що приймається, і, таким чином, зчитування значення біта відбувається в середині інтервалу його передачі, що дозволяє працювати з сигналами, що мають фронти великої тривалості.

Якщо мікроконтролер виявляє, що на виводі RXD все ще присутній рівень «0», тобто прийшов стар-біт, модуль UART переходить в робочий режим і починає зчитувати байт. Якщо ж на виводі RXD вже присутній рівень «1», вважається, що перший відлік був просто шумом, і модуль переходить до очікування коректного символу. Якщо приймач визначив, що прийшов дійсний символ, він починає брати по три відліки кожного біта в середині інтервалу його передачі. Якщо значення всіх трьох відліків біта не збігаються, то значення біта набуває рівним значенню двох однакових відліків. На завершення модуль зчитує вибірки, що відносяться до стоп-біту. Для того, щоб було вирішено про коректний прийом символу, принаймні, дві з цих вибірок мають дорівнювати одиниці. Інакше модуль вважає символ за невірно кадрований і реєструє помилку кадркування (framing error)

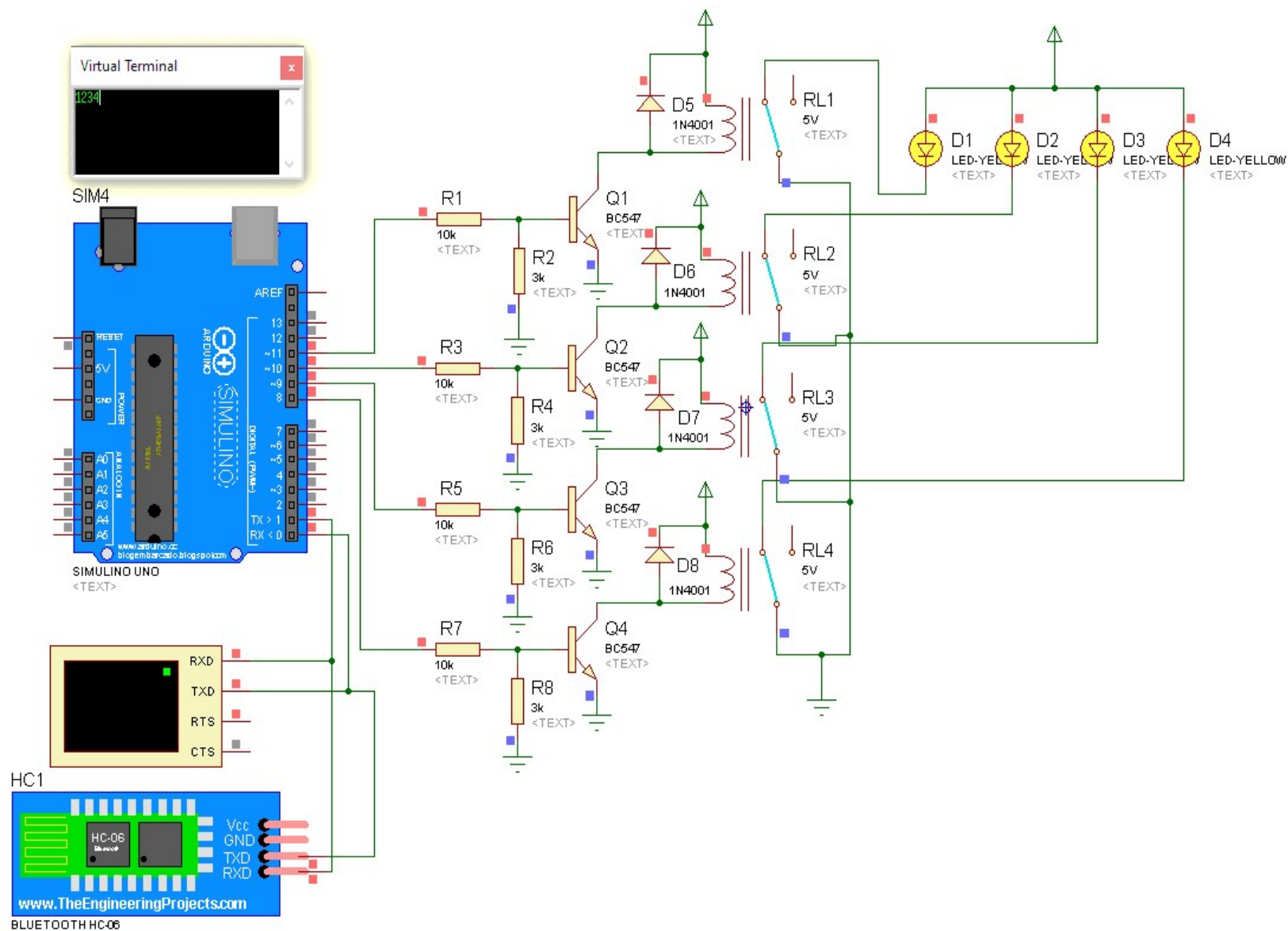
Керування 4 виконавчими пристроями по UART



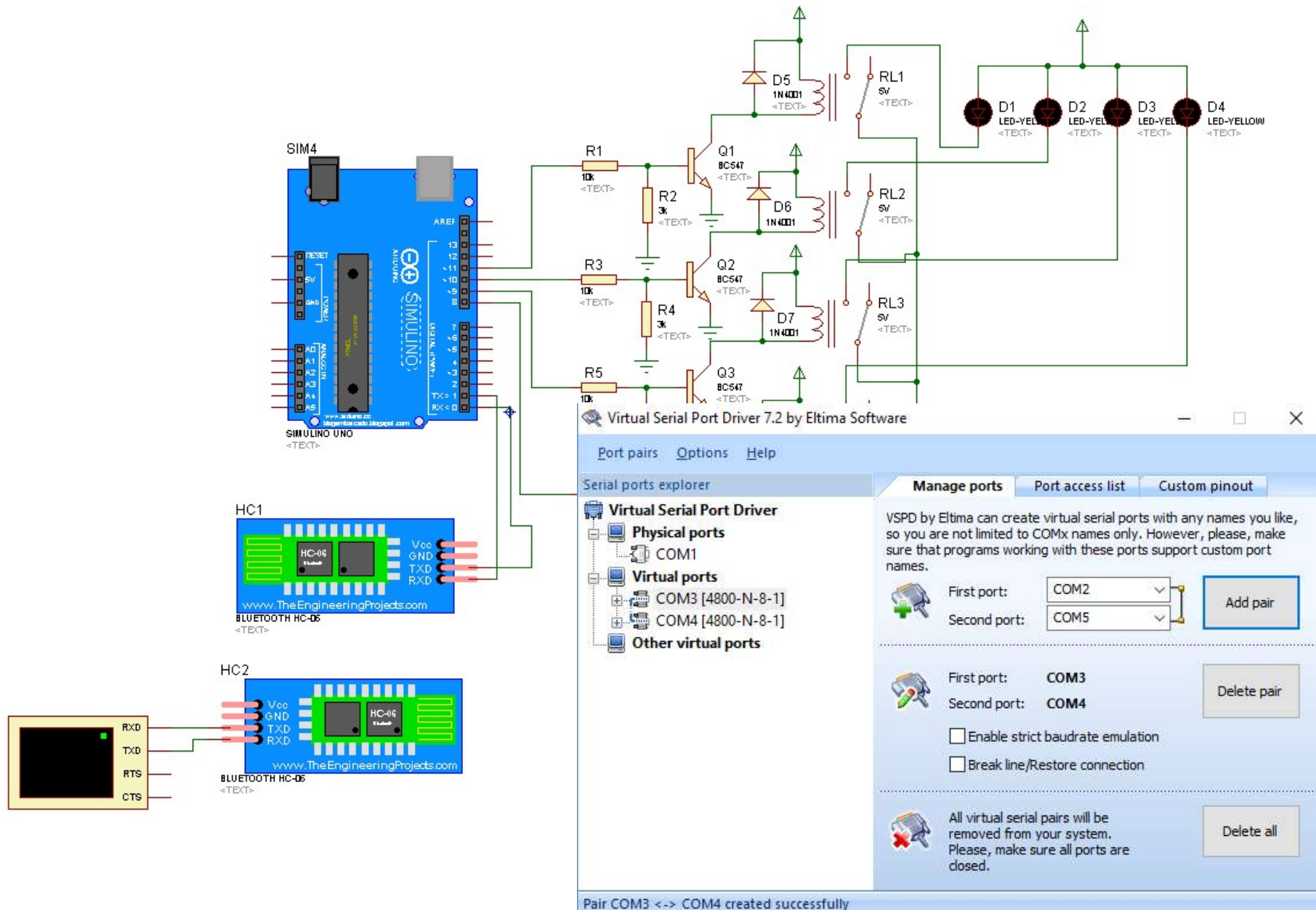
Керування 4 виконавчими пристроями по UART

```
1 char incomingByte;
2 int b1=0;
3 int load1 = 8;
4 int load2 = 9;
5 int load3 = 10;
6 int load4 = 11;
7 void setup() {
8     Serial.begin(9600);
9     pinMode(load1, OUTPUT);
10    pinMode(load2, OUTPUT);
11    pinMode(load3, OUTPUT);
12    pinMode(load4, OUTPUT);
13 }
14 void loop() {
15     if (Serial.available() > 0) { //якщо є дані
16         incomingByte = Serial.read(); // читаємо байт
17         if ( incomingByte == '1') {PORTB^=0b00000001;}
18         //переключити D8, решта залишити без змін
19         if ( incomingByte == '2') {PORTB ^=0b00000010;}
20         if ( incomingByte == '3') {PORTB ^=0b00000100;}
21         if ( incomingByte == '4') {PORTB ^=0b00001000;}
22     }
23 }
```

Керування 4 виконавчими пристроями по BT/UART



Керування 4 виконавчими пристроями по ВТ



Онлайн сервіс RemoteXY

РЕДАКТОР ПРИКЛАДНА ПРОГРАМА ДОКУМЕНТАЦІЯ Приклади СООБЩЕСТВО БЛОГ УВІЙТИ



Створіть графічний інтерфейс для вашого пристрою

ПРОБУВАТИ!

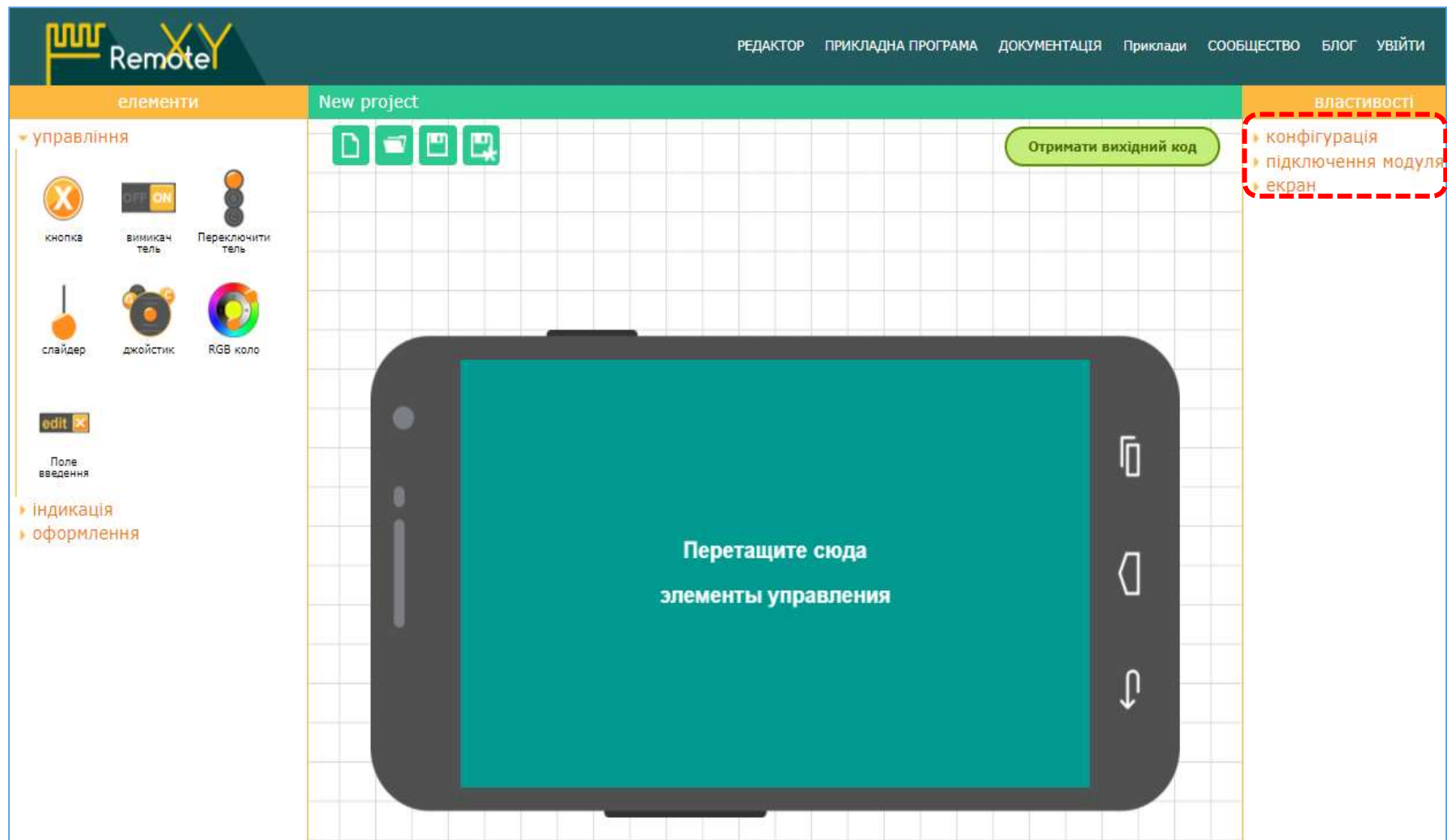
Підтримка:
Arduino, ESP8266, ESP32
Android, iOS
Bluetooth, Wi-Fi, Ethernet, USB
Internet з будь-якого місця

Чотири кроки для досягнення мети

1. Розробіть інтерфейс.
2. Завантажте вихідний код.
3. Підключіть модуль зв'язку.
4. Встановіть мобільний додаток для управління.



Онлайн сервіс RemoteXY




Онлайн сервіс RemoteXY

з'єднання



Bluetooth

Пристрій



Arduino UNO

модуль



HC-05 (06) Bluetooth module

середовище



Arduino IDE

сумісні модулі


HC-05 (06) Bluetooth module



HM-10 Bluetooth BLE module

доступні модулі


ESP8266 Wi-Fi module


Ethernet shield W5100


USB to UART converter


WiFi on chip


Bluetooth on chip

застосувати

скасування

властивості

конфігурація



Bluetooth



Arduino UNO



HC-05 (06) Bluetooth module



Arduino IDE

підключення модуля

екран

Онлайн сервіс RemoteXY

The screenshot displays the RemoteXY online service interface, which is a web-based tool for creating and managing mobile applications. The interface is divided into three main sections: a left sidebar with a component palette, a central workspace, and a right sidebar with configuration options.

Left Sidebar (елементи): Contains a list of UI components for the mobile app, including buttons, switches, sliders, joysticks, and RGB wheels. There is also an 'edit' button and a 'Поле введення' (input field) option.

Central Workspace (New project *): A large grid area where the mobile app is built. It features a central smartphone icon with the text 'Перетящите сюди елементи управління' (Drag elements here). Above the phone are icons for file operations (new, open, save, export) and a button labeled 'Отримати вихідний код' (Get source code).

Right Sidebar (властивості): Contains configuration settings for the app. The 'конфігурація' (configuration) section includes options for Bluetooth, Arduino UNO, HC-05 (06) Bluetooth module, and Arduino IDE. The 'підключення модуля' (module connection) section is highlighted with a red dashed box and includes settings for the connection interface (Hardware Serial), serial port (Serial, pins 0 (RX) and 1 (TX)), and baud rate (9600). The 'екран' (screen) section includes options for background color, orientation (вертикально), and access password.


Онлайн сервіс RemoteXY

The screenshot displays the RemoteXY online service interface, which is used for creating and managing remote control applications for Arduino IDE. The interface is divided into several sections:

- управління (Control):** This section on the left contains various interactive elements that can be added to the app. These include a button (кнопка), a toggle switch (вимикач тель), a rotary switch (Переключити тель), a slider (слайдер), a joystick (джойстик), and an RGB wheel (RGB коло). A red dashed box highlights the toggle switch element.
- індикація (Indicator):** This section includes an "edit" button and a text input field (Поле введення).
- оформлення (Appearance):** This section includes a "Label" element (мітка), a panel (панель), and a page (сторінка). A red dashed box highlights the "Label" element.
- Arduino IDE:** The top right corner shows the Arduino IDE logo and the text "Arduino IDE".
- підключення модуля (Module Connection):** This section on the right allows for configuring the module connection. It includes a dropdown for the connection interface (Hardware Serial), a dropdown for the serial port (Serial, pins 0 (RX) and 1 (TX)), and a dropdown for the baud rate (9600).
- екран (Screen):** This section allows for configuring the screen appearance. It includes a color picker for the background color (Колір фону), a dropdown for the orientation (вертикально), and a text input field for the access password (Пароль доступу).
- елемент (Element):** This section on the right provides detailed configuration for the selected element. It includes a color picker for the element color (колір), a color picker for the background color (Колір фону), a text input field for the variable name (Ім'я змінної), a text input field for the on-state label (Напис включення), a text input field for the off-state label (Напис виключення), a dropdown for the shape (отрісовка), and a dropdown for the number of digits (Прив'язати до висновку).

The central part of the interface shows a mobile app design on a grid. The app displays four channels (CH1, CH2, CH3, CH4) with toggle switches. A blue arrow points from the "Отримати вихідний код" (Get source code) button to the "елемент" (Element) configuration section.

Онлайн сервіс RemoteXY



Сергій ***[Мои проекты](#)[Мої токени](#)

[РЕДАКТОР](#)[ПРИЛОЖЕНИЕ](#)[ДОКУМЕНТАЦИЯ](#)[ПРИМЕРЫ](#)[СООБЩЕСТВО](#)[БЛОГ](#)

Исходный код для проекта: Новый проект

1. **Завантажити вихідний код** програми, відкрийте його в Arduino IDE.
2. Установити **бібліотеку RemoteXY** для Arduino IDE.
3. Скомпілюйте вихідний код та завантажте плату Arduino за допомогою Arduino IDE.
4. Правильно підключіть **модуль Bluetooth** к плате Arduino.
5. Установити мобільне додаток **RemoteXY, вер. 4.7.12** для смартфона / планшета.
6. Підключіть к Arduino з мобільного додатка.

project.ino

[Завантажити код](#)[Загрузить библиотеку](#)

```
/*
-- New project --

This source code of graphical user interface
has been generated automatically by RemoteXY editor.
To compile this code using RemoteXY library 2.4.3 or later version
download by link http://remotexy.com/en/library/
To connect using RemoteXY mobile app by link http://remotexy.com/en/download/
- for ANDROID 4.7.12 or later version;
- for iOS 1.4.7 or later version;
```


Онлайн сервіс RemoteXY

```
1 ///////////////////////////////////////////////////////////////////
2 //          RemoteXY include library          //
3 ///////////////////////////////////////////////////////////////////
4 // определение режима соединения и подключение библиотеки RemoteXY
5 #define REMOTEXY_MODE__HARDSERIAL
6 #include <RemoteXY.h>
7 #define REMOTEXY_SERIAL Serial
8 #define REMOTEXY_SERIAL_SPEED 9600
9 #pragma pack(push, 1)
10 uint8_t RemoteXY_CONF[] =
11 { 255,4,0,0,0,115,0,11,24,1,
12   2,0,25,8,22,11,2,26,31,31,
13   79,78,0,79,70,70,0,2,0,25,
14   25,22,11,2,26,31,31,79,78,0,
15   79,70,70,0,2,0,25,42,22,11,
16   2,26,31,31,79,78,0,79,70,70,
17   0,2,0,25,59,22,11,2,26,31,
18   31,79,78,0,79,70,70,0,129,0,
19   8,10,12,6,17,67,72,49,0,129,
20   0,8,28,12,6,17,67,72,50,0,
21   129,0,8,45,12,6,17,67,72,51,
22   0,129,0,8,61,12,6,17,67,72,
23   52,0 };
24 struct {
25     // input variables
26     uint8_t switch_1; // =1 если переключатель включен и =0 если отключен
27     uint8_t switch_2; // =1 если переключатель включен и =0 если отключен
28     uint8_t switch_3; // =1 если переключатель включен и =0 если отключен
29     uint8_t switch_4; // =1 если переключатель включен и =0 если отключен
30     // other variable
31     uint8_t connect_flag; // =1 if wire connected, else =0
32 } RemoteXY;
33 #pragma pack(pop)
```

Онлайн сервіс RemoteXY

```
38 #define PIN_SWITCH_1 8
39 #define PIN_SWITCH_2 9
40 #define PIN_SWITCH_3 10
41 #define PIN_SWITCH_4 11
42
43 void setup() {
44     RemoteXY_Init ();
45     pinMode (PIN_SWITCH_1, OUTPUT);
46     pinMode (PIN_SWITCH_2, OUTPUT);
47     pinMode (PIN_SWITCH_3, OUTPUT);
48     pinMode (PIN_SWITCH_4, OUTPUT);
49     // TODO you setup code
50 }
51
52 void loop()
53 {
54     RemoteXY_Handler ();
55     digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==0)?LOW:HIG);
56     digitalWrite(PIN_SWITCH_2, (RemoteXY.switch_2==0)?LOW:HIG);
57     digitalWrite(PIN_SWITCH_3, (RemoteXY.switch_3==0)?LOW:HIG);
58     digitalWrite(PIN_SWITCH_4, (RemoteXY.switch_4==0)?LOW:HIG);
59     // TODO you loop code
60     // используйте структуру RemoteXY для передачи данных
61     // не используйте функцию delay()
62 }
```

SoftwareSerial

Бібліотека SoftwareSerial дозволяє реалізувати послідовний інтерфейс на будь-яких цифрових портах Arduino за допомогою програмних засобів, які дублюють функціональність UART. Бібліотека дозволяє програмно створювати кілька послідовних портів, які працюють на швидкості до 115200 бод. Для пристроїв, що працюють з інвертованим сигналом, в бібліотеці передбачено відповідний параметр, що включає інвертування

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial mySerial(10, 11); // RX, TX
3
4 void setup() {
5     // Ініціалізація послідовного інтерфейсу
6     Serial.begin(57600);
7     Serial.println("Good");
8     // встановлюємо швидкість передачі даних для SoftwareSerial
9     mySerial.begin(9600);
10    mySerial.println("Hello, world!");
11 }
12
13 void loop() {
14     if (mySerial.available()){
15         Serial.write(mySerial.read());
16         // TODO you setup code
17     }
18 }
```