

Лабораторна робота №3

Тема: Makeblock Music Robot Kit V2.0

Мета: Розібратися з будовою, програмним забезпеченням та навчитися програмувати робота Makeblock Music Robot Kit V2.0.

Завдання: Використовуючи стандартний набір нот доступних в коді, змінити мелодію на довільну.



MAKEBLOCK MUSIC ROBOT KIT V2.0

Makeblock Music Robot Kit V2.0 - це оновлений комплект, заснований на Makeblock Music Kit V1.0. Побудований на базі Me Orion (Base on Arduino UNO). Makeblock Orion - легка у використанні плата на базі Arduino Uno з поліпшенням освіти. Вона забезпечує вісім портів RJ25 для підключення до всіх інших модулів серії Me з кольоровими етикетками, може врятувати вас від жахливих дротів і допоможе вам зосередитися на дивовижних проектах винахідника. Makeblock Orion підтримує більшу частину програмного забезпечення (Arduino / Scratch /

AduBlock), і ми пропонуємо спеціальне програмне забезпечення для графічного програмування (Scratch for Robot / Makeblock HD App).

Специфікація:

- Робоча напруга: 6-12В постійного струму;
- Мікроконтролер: ATmega238;
- Кут виявлення: віддайте перевагу на 30 градусів;
- Розмір: 80 x 60 x 18 мм (Довжина x Ширина x Висота);

В якості крокового двигуна для руху каретки з механізмом використовується двигун NEMA17. Кроковий двигун типорозміру NEMA17, ток на обмотку 1.7А. Як правило використовується в 3D принтерах, екструдерах і невеликих ЧПУ верстатах. Розпіновка буває двох варіантів, в залежності від партії. Варіант 1: червоний А +, зелений А-, жовтий В +, синій В-. Варіант 2: зелений А +, чорний А-, синій В +, червоний В-.

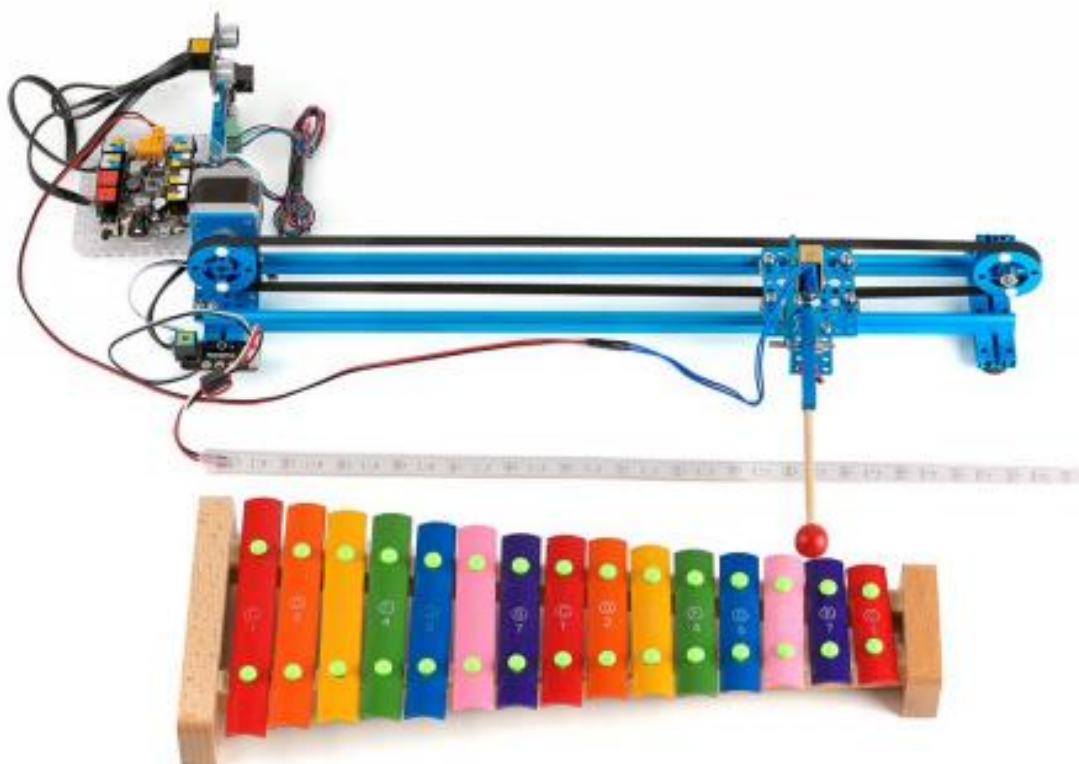
Характеристики:

- модель: JK42HS40-1704-13A (HS4401)
- кут повороту за один крок: 1.8 °
- діаметр вала: 5 мм
- довжина вала: 24мм
- довжина мотора: 40 мм
- ток на обмотку: 1.7А
- напруга: 5 - 24В
- опір обмотки: 1.65 Ом
- індуктивність обмотки: 3.2 мГн
- крутний момент утримання: 3.6 кг / см
- кількість контактів на роз'ємі: 4
- маса: 280 г

Він керує кроковим двигуном для переміщення за допомогою ультразвукового датчика, який визначає напрямок жестів. Коли дерев'яна паличка рухається до точного положення, вона потрапляє в ксилофон, керуючи

електромагнітом. Ви також можете керувати музичним роботом верхнім комп'ютером або DIY музичними партитурами, програмуючи будь-яку музику, щоб грати суміш стилів.

Розмістіть Music Robot Kit V2.0 так як показано нижче.



Підключіть Me Orion до ПК через USB-кабель, запустіть програмне забезпечення Music Bot, клацніть випадające меню Serial і виберіть правильний порт (очікуйте COM1), як показано нижче.



З випадючого списку оберіть необхідний тип керуючої плати. В даному випадку необхідно обрати Arduino UNO. Та натисніть кнопку Connect.



Після оновлення, Music Robot Kit V2.0 продовжуватиме рухатися, поки не досягне сенсора Me Limit Switch V2.0. Коли він торкнеться кінцевого вимикача, він повернеться на невелику відстань і потрапить у першу ноту. Одночасно включається перша світлодіодна смуга RGB. Залишайтеся на 3 секунди, музичний робот буде грати «Брат Джон», і він зупиниться, коли музика закінчиться. Ви можете керувати музичним роботом двома способами: за допомогою жестів або комп'ютера.

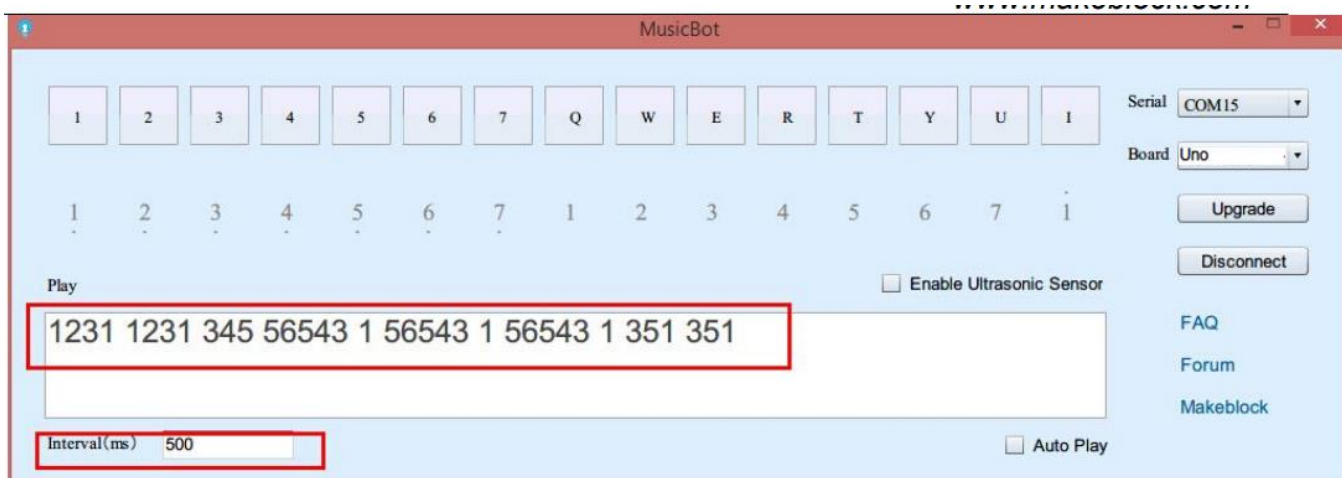
Коли ваша рука перед певною нотою рухатиметься, музичний робот потрапить відповідно до клавіші ноти. Переконайтеся, що ваша рука знаходиться

в зоні вимірювання щільності ультразвукового датчика. Ви також можете використовувати MusicBot, щоб увімкнути / вимкнути ультразвукову функцію.

Коли ви натискаєте на певну ноту програмне забезпечення Music Bot, музичний робот потрапить у відповідну клавішу, як показано нижче.



Також ви можете вводити ноти через відповідне поле для вводу з клавіатури. Введіть формулу в області відтворення (пробіл відповідає паузі) і клацніть на вікні автоматичного відтворення, музичний робот автоматично потрапить у ксилофон. Ви можете DIY власну музику, весело провести час з DIY. Дані інтервалу (одиниці: мілісекунди) - час паузи під час кожної ноти, переконайтеся, що він не менше 300.



Також можна програмувати MusicBot за допомогою Arduino IDE.

Зразок коду:

```

#include "MePort.h"
#include "MeDCMotor.h"
#include "MeUltrasonic.h"
#include "MeRGBLed.h"
#include "MePort.h"
#include "Stepper.h"

char music_score[]="89:8089:80;<0;<000<=<;:0800<=<;:08009050800090508";
int dirPin = mePort[PORT_1].s1;
int stpPin = mePort[PORT_1].s2;
int sw    = mePort[PORT_6].s2;
Stepper stepper(Stepper::DRIVER,stpPin,dirPin);
MeUltrasonic ultraSensor(PORT_3);
MeRGBLed led(PORT_6,SLOT_1);
MeDCMotor kicker(M1);

char mode=0;
int targetIndex = 0;
int currentIndex = 0;
int onestep = 80;

void setup()
{
  Serial.begin(115200);
  led.setNumber(15);
  indicators(15,0,0,0);
  led.show();
  initStepper();
  indicators(1,100,0,0);
  led.show();
  kickoff();
  delay(5000);
  music();
}

void loop()
{
  if(mode) ultra_control();
}

```

```
upper_computer();  
delay(50);  
}
```

```
void upper_computer()  
{  
  if(Serial.available())  
  {  
    char temp = Serial.read();  
    if(temp=='M')  
    {  
      mode=1;  
      return;  
    }  
    if(temp=='N')  
    {  
      mode=0;  
      return;  
    }  
    if(temp>0 && temp<16)  
    {  
      targetIndex = 16- temp;  
      moveStepper();  
    }  
  }  
}
```

```
void initStepper()  
{  
  stepper.setMaxSpeed(500);  
  stepper.setAcceleration(15000);  
  stepper.setCurrentPosition(0);  
  stepper.run();  
  pinMode (sw,INPUT_PULLUP);  
  delay(500);  
  stepper.moveTo(-10000);  
  while(1)  
  {
```

```

if(!stepper.run() || !digitalRead(sw))
{
    delay(100);
    if(!stepper.run() || !digitalRead(sw))
        break;
}
stepper.run();
}
delay(2000);
stepper.setCurrentPosition(0);
stepper.run();
stepper.moveTo(40);
while(stepper.currentPosition() != 40)
{
    stepper.run();
}
stepper.stop();
stepper.setMaxSpeed(10000);
stepper.setAcceleration(10000);
stepper.setCurrentPosition(80);
stepper.run();
}

```

```

void music()
{
    char i=0;
    while(music_score[i]!='\0')
    {
        targetIndex=(music_score[i]-48);
        targetIndex=16-targetIndex;
        if (targetIndex==16) delay(200);
        else moveStepper();
        i++;
    }
}

```

```

void kickoff()
{

```



```
kicker.run(115);  
delay(55);  
kicker.stop();  
}
```

```
void moveStepper()  
{  
  if(targetIndex>0 && targetIndex<16)  
  {  
    int stepPos = targetIndex*onestep;  
    stepper.moveTo(stepPos);  
    while(stepper.run());  
    int r=random(1,50);  
    int b=random(1,50);  
    int g=random(1,50);  
    indicators(targetIndex,r,b,g);  
    currentIndex = targetIndex;  
    kickoff();  
  }  
}
```

```
void ultra_control()  
{  
  int value=0;  
  value = ultraSensor.distanceCm();  
  if(value==0) return;  
  if(value <70)  
  {  
    targetIndex=value/10+9;  
    if(targetIndex!=currentIndex)  
    {  
      moveStepper();  
    }  
  }  
}
```

```
void indicators(int count,byte r,byte g,byte b)  
{  
  for(int x=count;x<15;x++)
```

```
{  
    led.setColorAt(x,0,0,0);  
}  
led.show();  
for(int x=0;x<count;x++)  
{  
    led.setColorAt(x,r,g,b);  
}  
led.show();  
}
```

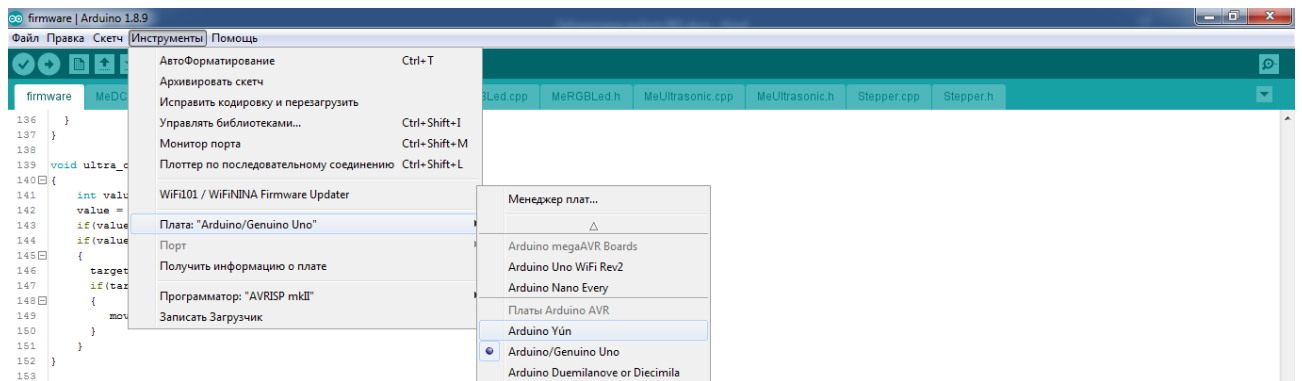
Інтерфейс програми Arduino для програмування контролера

Arduino (Ардуіно) — апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері (наприклад: Processing, Adobe Flash, Max/MSP, Pure Data, SuperCollider).

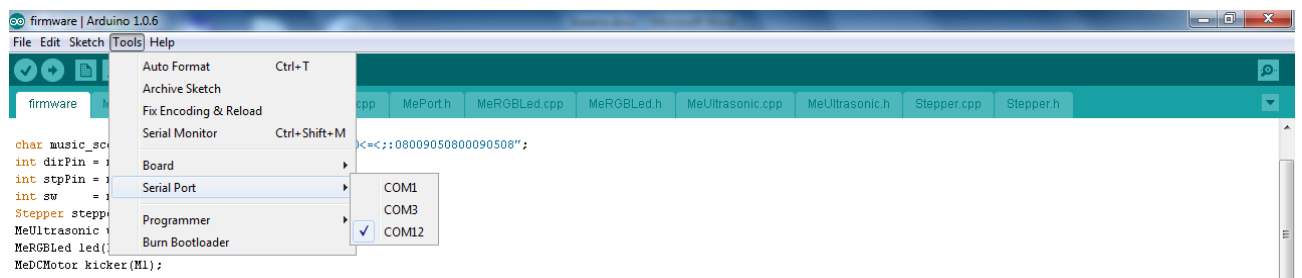
```
1 /*  
2   Fast pin writing example, for Maple.  
3  
4 */  
5  
6 #define PIN_PA6  
7 #define PIN_PORT_GPIOB  
8 #define PIN_BIT 4  
9  
10 HardwareTimer pwmtimer(1); //PWM2  
11  
12 void setup() {  
13   //pinMode(PIN, OUTPUT);  
14   //pinMode(PIN, PWM);  
15   //pinMode(PA8, PWM); //PWM1  
16  
17   pinMode(PIN, PWM);  
18   pwmtimer.setPrescaleFactor(1);  
19   pwmtimer.setPeriod(50);  
20 }  
21  
22 void loop() {  
23   for (int i = 0; i < 65535; i+=50)  
24   {  
25     pwmWrite(PIN, i);  
26     delay(20);  
27   }  
28   //delay(2000);  
29 }
```

7 Arduino/Genuino Uno на COM7

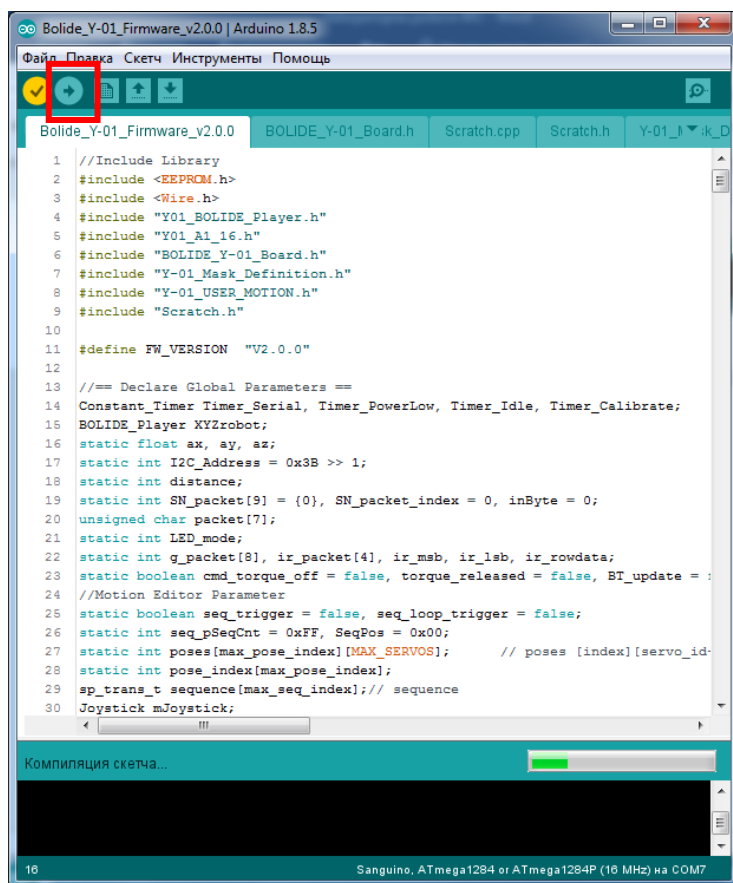
Обираємо необхідну плату зі списку



Підключаємо та обираємо Serial Port



Та натискаємо Upload



Оформлення звіту

Звіт повинен містити:

- назву та мету роботи;
- програма написана в Arduino IDE;
- висновки по роботі.