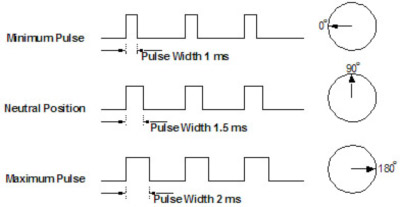
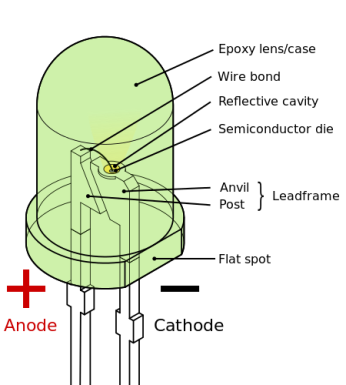
[](https://en.wikipedia.org/wiki/File:RGB-SMD-LED.jpg)dasdasdasdasd

### בגיליון זה

ארדוינו אונו 1

רכיבים 1

רכיבים 2

תוכנה 3

תוכנה 4

תוכנה 5

Schematic 6

רכיבים

Servos are controlled by sending an electrical pulse of variable width, or **pulse width modulation** (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90 degrees in either direction for a total of 180 degree movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the [motor](http://www.jameco.com/webapp/wcs/stores/servlet/JamecoSearch?langId=-1&storeId=10001&catalogId=10001&freeText=motor&search_type=jamecoall) determines position of the shaft, and based on the duration of the pulse sent via the control wire; the [rotor](http://www.jameco.com/webapp/wcs/stores/servlet/JamecoSearch?langId=-1&storeId=10001&catalogId=10001&categoryName=cat_3540&subCategoryName=Electromechanical%20%2F%20Switches%20%2F%20Rotary&category=354055&refine=1&position=1&history=kv7hqebe%7CfreeText~rotor%5Esearch_type~jamecoall%5EprodPage~50%5Epage~SEARCH%252BNAV%405hha4bcd%7Ccategory~35%5EcategoryName~category_root%5Eposition~1%5Erefine~1%5EsubCategoryName~Electromechanical%5EprodPage~50%5Epage~SEARCH%252BNAV) will turn to the desired position. The servo motor expects to see a pulse every 20

A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback. It also requires a [servo drive](http://www.kollmorgen.com/en-us/products/drives/servo/servo-drives/) to complete the system. The drive uses the feedback sensor to precisely control the rotary position of the motor. This is called closed-loop operation. By running the system closed-loop, servo motors provide a high performance alternative to stepper and AC induction motors.

**Servo**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/).

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various expansion boards (*shields*) and other circuits. The boards feature serial communications interfaces, including [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus" \o "Universal Serial Bus) (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). In addition to using traditional compiler toolchains, the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on the [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) language project. Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode()](https://www.arduino.cc/en/Reference/PinMode), [digitalWrite()](https://www.arduino.cc/en/Reference/DigitalWrite), and [digitalRead()](https://www.arduino.cc/en/Reference/DigitalRead) functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.



# ארדוינו אונו (Arduino uno)

תאריך הידיעון

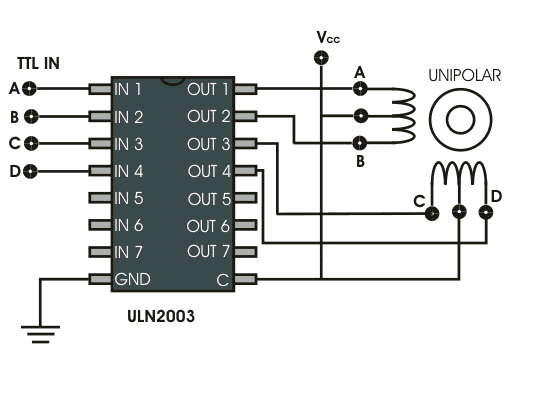
כרך 1, גיליון 1

Internet Of Things

[RGB](https://en.wikipedia.org/wiki/RGB_color_model) LEDs consist of one red, one green, and one blue LED. By independently [adjusting](https://en.wikipedia.org/wiki/Pulse-width_modulation) each of the three, RGB LEDs are capable of producing a wide [color gamut](https://en.wikipedia.org/wiki/Color_gamut). Unlike dedicated-color LEDs, however, these obviously do not produce pure wavelengths. Moreover, such modules as commercially available are often not optimized for smooth color mixing.

A light-emitting diode (LED) is a two-[lead](https://en.wikipedia.org/wiki/Lead_(electronics)) [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) [light source](https://en.wikipedia.org/wiki/Light_source). It is a [p–n junction](https://en.wikipedia.org/wiki/P%E2%80%93n_junction) [diode](https://en.wikipedia.org/wiki/Diode), which emits light when activated. When a suitable [voltage](https://en.wikipedia.org/wiki/Voltage) is applied to the leads, [electrons](https://en.wikipedia.org/wiki/Electron) are able to recombine with [electron holes](https://en.wikipedia.org/wiki/Electron_hole) within the device, releasing energy in the form of [photons](https://en.wikipedia.org/wiki/Photon). This effect is called [electroluminescence](https://en.wikipedia.org/wiki/Electroluminescence), and the color of the light (corresponding to the energy of the photon) is determined by the energy [band gap](https://en.wikipedia.org/wiki/Band_gap) of the semiconductor. LEDs are typically small (less than 1 mm2 ) and integrated optical components may be used to shape the [radiation pattern](https://en.wikipedia.org/wiki/Radiation_pattern).

**Led & RGB**



Arduino

A **stepper motor drive** is a circuit which is used to drive or run a stepper motor. It is often called a stepper motor driver. A **stepper motor drive** usually consists of a controller, a driver and the connections to the motor.

Essential Components of Stepper Motor Drive

1. Controller(Essentially a micontroller or a microprocessor)
2. A driver IC to handle the motor current
3. A power supply unit

Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the motor will rotate, one step at a time.  
  
With a computer controlled stepping you can achieve very precise positioning and/or speed control. For this reason, stepper motors are the motor of choice for many precision motion control applications.   
  
Stepper motors come in many different sizes and styles and electrical characteristics. This guide details what you need to know to pick the right motor for the job.

**Step Motor**

milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90-degree position. Shorter than 1.5ms moves it to 0 degrees, and any longer than 1.5ms will turn the servo to 180 degrees.

רכיבים

עמוד 2

תוכנה

עמוד 3

#include <Servo.h>

Servo myservo; // create servo object to control a servo

const int PWM=9;

#include <Stepper.h>

const int stepsPerRevolution = 400; // change this to fit the number of steps per revolution

Stepper myStepper(stepsPerRevolution, 2, 3, 4, 5);

const int Red=11;

const int Green=12;

const int Blue=13;

const int Led = 10;

char serialdata[5]={48,48,48,48,48};// initialize with all 0

int motor,servo,pos = 20;

void setup() {

Serial.begin(9600);

Serial.setTimeout(50);

myservo.attach(PWM); // attaches the servo on pin 9 to the servo object

myStepper.setSpeed(50);

pinMode(Red,OUTPUT);

pinMode(Green,OUTPUT);

pinMode(Blue,OUTPUT);

pinMode(Led,OUTPUT);

digitalWrite(Red,HIGH);

digitalWrite(Green,HIGH);

digitalWrite(Blue,HIGH);

}

תוכנה

עמוד 4

Void loop() {

int val1,val2,val3,val4,val5,lf = 10,i=0;

if(Serial){ //if doesnt found usb open, do not read!

Serial.readBytesUntil(lf, serialdata, 5);

}

//Download Buffer

val1=(int)serialdata[0];

val2=(int)serialdata[1];

val3=(int)serialdata[2];

val4=(int)serialdata[3];

val5=(int)serialdata[4];

//LED

if (val1 == 'A' && val2 == 'L'){ //AL1/AL0

if(val3=='1'){

digitalWrite(Led,HIGH);

Serial.println("AL1");

}

if(val3=='0'){

digitalWrite(Led,LOW);

Serial.println("AL0");

}

}

//SERVO

if (val1 == 65 && val2 == 83){//AS1/AS0

if(val3==49){

Serial.println("AS1"); //send eco

servo = 1;

}

if(val3==48){

Serial.println("AS0");

servo = 0;

}

}

if(servo == 1){

for (pos = pos; pos <= 170; pos += 2) {

myservo.write(pos);

delay(20);

}

for (pos = 170; pos >= 20; pos -= 2) {

myservo.write(pos);

delay(20);

}

}

//STEP

if (val1 == 65 && val2 == 77){//AM1/AM0

if (val3 == 48 ){

Serial.println("AM0");

motor = 0;

}

if (val3 == 49){

Serial.println("AM1");

motor = 1;

}

}

if (motor == 1){

myStepper.step(400);

delay(200);

}

//LED-RGB

if(val1==65 && val2==82){ //AR-0,1,2,3,4,5,6,7.

if(val3==55){

digitalWrite(Red,LOW);

digitalWrite(Green,LOW);

digitalWrite(Blue,LOW);

תוכנה

עמוד 5

Serial.println("AR7"); //send eco

}

if(val3==54){

digitalWrite(Red,HIGH);

digitalWrite(Green,LOW);

digitalWrite(Blue,LOW);

Serial.println("AR6"); //send eco

}

if(val3==53){

digitalWrite(Red,LOW);

digitalWrite(Green,HIGH);

digitalWrite(Blue,LOW);

Serial.println("AR5"); //send eco

}

if(val3==52){

digitalWrite(Red,HIGH);

digitalWrite(Green,HIGH);

digitalWrite(Blue,LOW);

Serial.println("AR4"); //send eco

}

if(val3==51){

digitalWrite(Red,LOW);

digitalWrite(Green,LOW);

digitalWrite(Blue,HIGH);

Serial.println("AR3"); //send eco

}

if(val3==50){

digitalWrite(Red,HIGH);

digitalWrite(Green,LOW);

digitalWrite(Blue,HIGH);

Serial.println("AR2"); //send eco

}

if(val3==49){

digitalWrite(Red,LOW);

digitalWrite(Green,HIGH);

digitalWrite(Blue,HIGH);

Serial.println("AR1"); //send eco

}

if(val3==48){

digitalWrite(Red,HIGH);

digitalWrite(Green,HIGH);

digitalWrite(Blue,HIGH);

Serial.println("AR0"); //send eco

}

}

//RESET BUFFER

serialdata[0]=0;

serialdata[1]=0;

serialdata[2]=0;

serialdata[3]=0;

serialdata[4]=0;

serialdata[5]=0;

serialdata[6]=0;

serialdata[7]=0;

serialdata[8]=0;

serialdata[9]=0;

serialdata[10]=0;

serialdata[11]=0;

serialdata[12]=0;

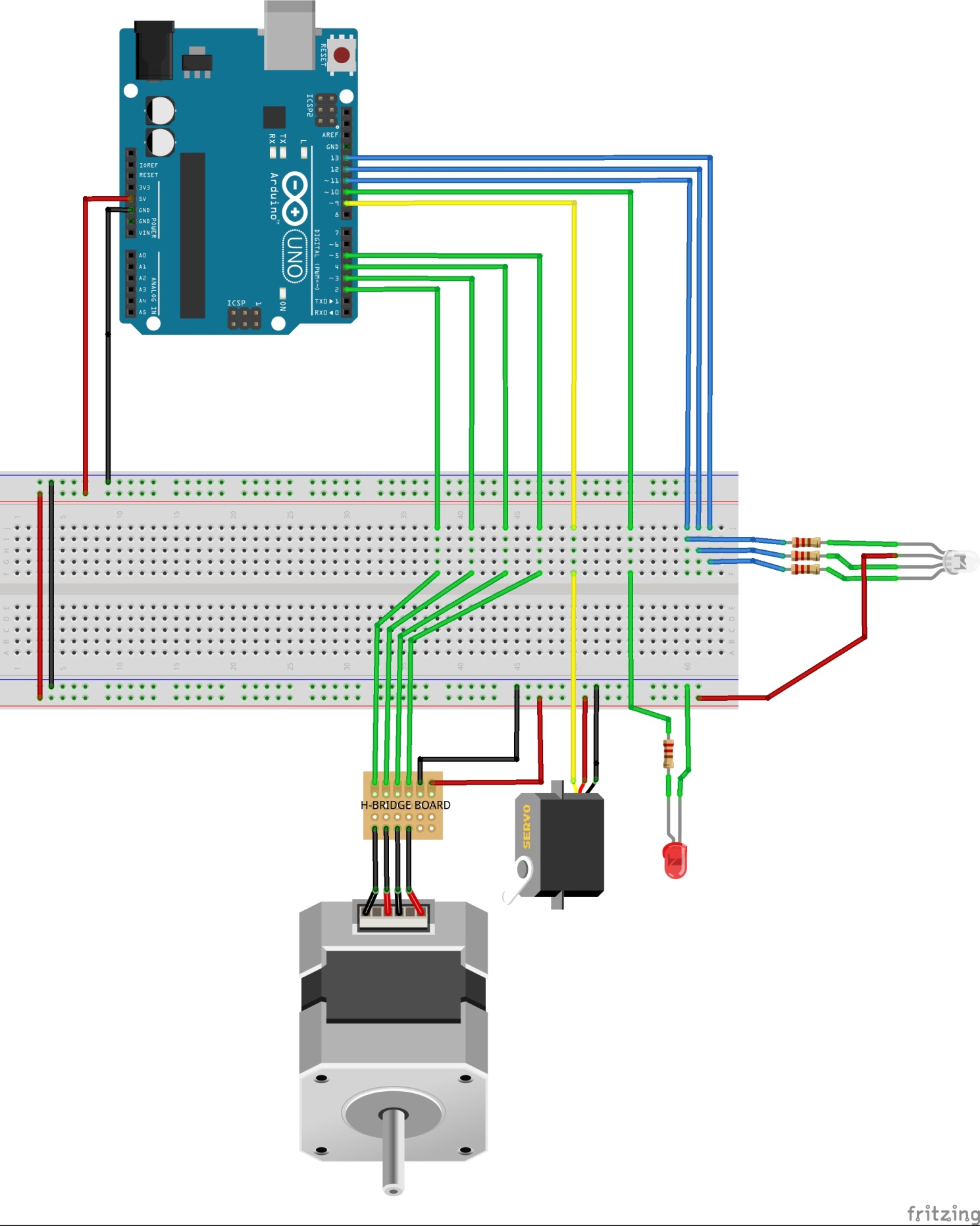
serialdata[13]=0;

serialdata[14]=0;

}

עמוד 6

Schematic

****