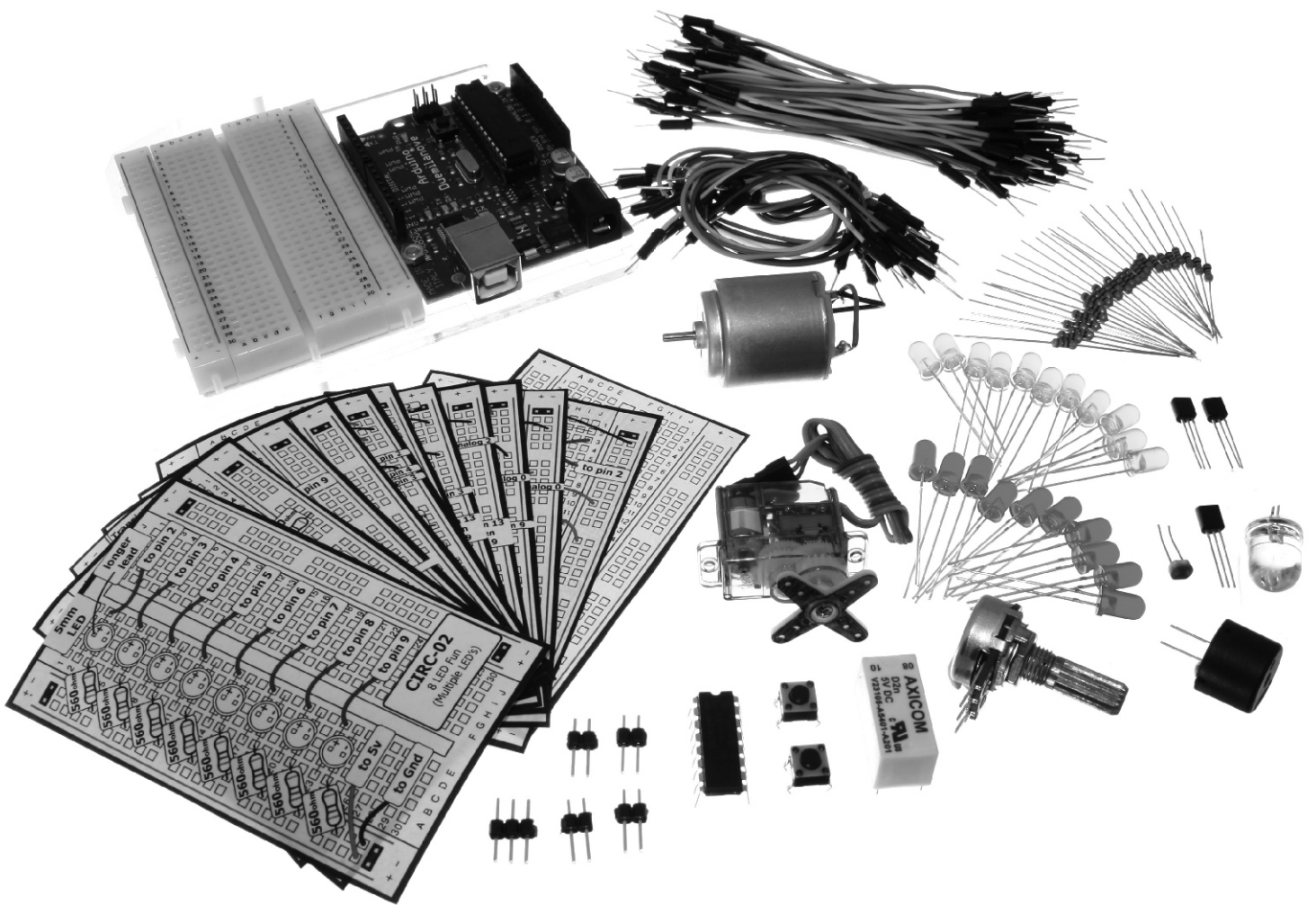


# Arduino Esperimentazio Gida



(ARDX)

## Hitz batzuk

### Kit honi buruz

Kit honen helburu nagusia ondo pasatzea da. Hortik haratago, zirkuitu errazak erabiliz elektronikako osagai Ezberdinekin eroso ibiltzea bilatzen da. Zirkuitua munta tu ondoren azalduko dizugu nola funtzionatzen duen. Arazoak badituzu, galderaren bat egin nahi baduzu edo gehiago ikasi nahi baduzu, email honetara idatz dezakezu: [help@oomlout.com](mailto:help@oomlout.com)



### Hardware libreari buruz

.:oomlout:.-eko proiektu guztiak irekiak dira. Zer esan nahi du horrek? Dena, hau da, Kit honen egikaritzapena, 3D ereduak edo kodea jaitsi daitezke modu irekian. Horrez gain banatu eta aldatzeko aukera daukazu. Eta gakoa non dago? Oso erraza Creative Commons (By - Share Alike ) lizentziarekin kaleratu dugu lana. Zure lanean .:oomlout:. Egile bezala agertu beharko da eta modu askean banatu beharko duzu. Zergatik? Gu software libreko kodearekin ikasi dugu eta esperientzia oso ona izan da Uste dugu antzeko esperientzia primerakoa izango litzatekeela mundu fisikoan.

Creative Commons CC (By - Share Alike) lizentziari buruzko informazio gehiagorako)  
( <http://tinyurl.com/2dkzmd> )

### .: oomlout :. buruz

Diseinu enpresa txiki xarmangarri bat gara

“kode irekiko proiektu lirainak egiten ditugu”

Ikusi nahi baduzu zeintzuk diren:

<http://www.oomlout.com>

### Arazoei buruz

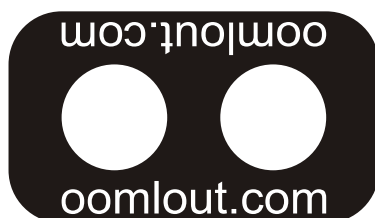
Sortzen dugun edozein gauza kalitaterik handienarekin egin nahi dugu. Azalpen txar Zatiaren bat faltan edo antzekoren bat ikusten baduzu idatziguzu mesedez.

Gure emaila hau da:

[help@oomlout.com](mailto:help@oomlout.com)

Arazoei buruz jakitea gustoko dugu, hurrengo bertsioak hobetzeko

Milesker .:oomlout:. hautatzeagatik



.: Non topatu dena :.

Hasi baino lehen

{ASEM }	Osagaiak muntatzen	02
{NST}	Programa instalatzen	03
{PR0G}	Programazioko tutoriala	04
{ELEK}	Elektronikako tutoriala	06

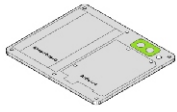
Zirkuituak

{CIRC01}	Hasiera - (LEDekin keinuak egin)	08
{CIRC02}	8 LEDekin jolasean - (LED asko)	10
{CIRC03}	Biratu Motore Biratu-(Transistoreak eta Motoreak)	12
{CIRC04}	Servo Bakar Bat- (Servoak)	14
{CIRC05}	8 LED gehiago - (74HC595 Shift erregistroa)	16
{CIRC06}	Musika - (Piezo Elementuak)	18
{CIRC07}	Botoiak sakatzen - (sakagailuak)	20
{CIRC08}	Tortsioak - (Potentziometroak)	22
{CIRC09}	Argia - (Fotoerresistoreak)	24
{CIRC10}	Tenperatura - (TMP36 Tenperatura Sentsorea)	26
{CIRC11}	Karga handiagoak - (erreleak)	28

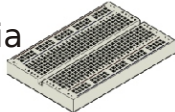
# 01 ASEM

Piezak elkartuz

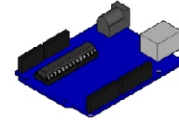
:: Dena elkartzen ::



Arduino Euskarria  
x1



Breadboard  
x1



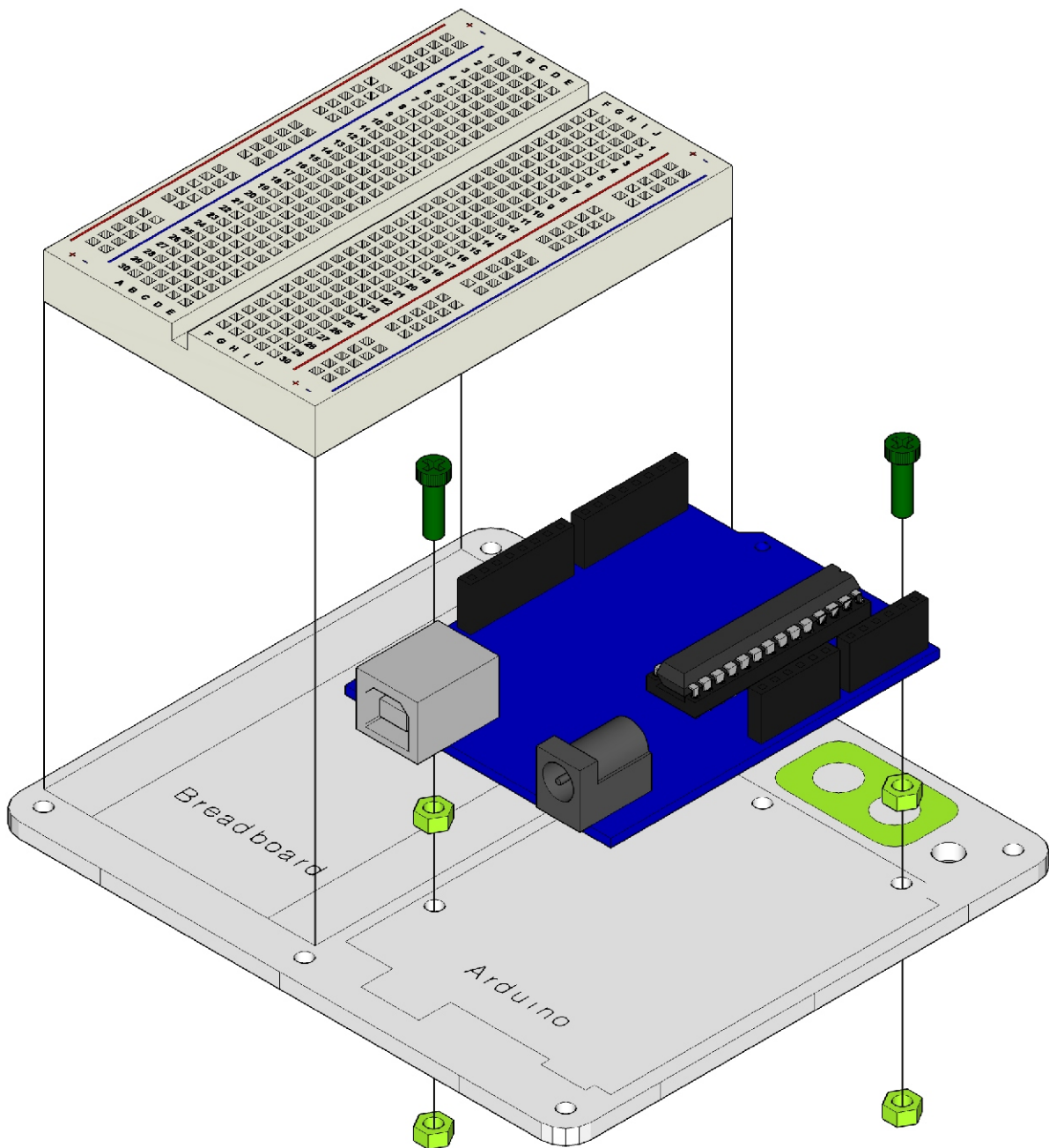
Arduino  
x1



3mm x 10mm  
Euskarria x2



3mm pilota  
x4



Arduinori buruzko sarrera bilatuz gero:  
:: <http://tinyurl.com/9txjmh> ::

# :: IDEa instalatzen ::

Arduinoko programak idazteko programa  
Hasieran pixkat arraroa badirudi ere, erabiltzen hastean  
Trikimailu guztiak argi edukiko dituzu

### 1go pausua: Programa jaitsi

Joan  
<http://arduino.cc/en/Main/Software>  
Zure SOrako programa jaitsi

#### Windows XP

### 2. pausua: Softwarea ireki

Deskonprimitu  
arduino-00XX-win.zip (xx- version #)  
Instalazio helbide gomendatua  
c:\Program Files\

### 3. pausua: Lasterbidea

Ireki  
c:\program files\arduino-00XX\ (xx- version #)  
Eskubiko klik  
Arduino.exe send to>Desktop (create shortcut)

### 4. pausua: Arduino konektatu

Arduinoa konektatu  
Arduinoa USB kablea erabiliz Pcaren USB portu batera  
konektatu  
Itxoin leiho bat agertu arte

### 5. pausua: hardware berria gehitu

Hautatu  
Install from a Specific destination  
Helbide hau hautatu  
c:\program files\arduino-00XX\drivers\FTDI USB Drivers  
Bukatuta

#### Mac OSX

### 2. pausua: Softwarea ireki

Deskonprimitu  
arduino-00XX-mac.zip (xx- version #)  
Karpeta mugitu  
//Applications karpetara/

### 3. pausua: Lasterbidea

Ireki  
//Applications/arduino-00XX/ (xx- version #)  
Komando klik  
Make Alias  
Idazmahaira arrastatu

### 4. pausua Driverrak instalatu

//Applications/Arduino-00XX/drivers/  
FTDIUSBSerialDriver\_V2\_2\_9\_Intel.dmg  
(V2\_1\_9.dmg if not an Intel Mac)  
Berrabiarazi

### 5. pausua Arduinoa konektatu

Arduinoa konektatu  
konektatu

:: OHARRA: ::

:: Arazorik bai? ::

:: Informazio gehiago? Linux erabiltzen duzu? ::

:: <http://tinyurl.com/r99d8u> ::

### Arduino programazio tutoriala

Arduino C programazio lenguaian programatzen da. Honako hau sarrera xume bat da Programazioan ibilia den norbaitentzat, Arduinoko trikimailuak argitzeko.

Kontzeptu hauek pixkat arraroak iruditzen bazaizkizu, ez arduratu, zirkuituekin hasi eta ulertzen joango zara. Sarrera luzeagoa nahi baduzu, Arduino.cc webgunea bisita ezazu.

### Egitura

Arduino programek (edo sketchek ) Bi funtzio dituzte (errutinak deiturikoak).

`void setup(){ }`  
Giltzen arteko kodea  
Behin bakarrik exekutatu da  
Arduinoa hastean

`void loop(){ }`  
Funtzio hau setup ondoren exekutaten da eta etengabe Exekutaten egongo da elikadura kendu arte

### Syntaxia

Cko elementu deseroso bat formatua da (oso bo teretsua baina zaila. Gogoratzen bazara ez duzu arazorik izango

`//` (lerro bateko iruzkina)  
Notak idazteko aproposa  
Kodeak zer egiten duen Jarraitu ahal izateko  
Lerro bukaeraraino idatzitakoa programak ez dio jaramorik egingo.

`/* */` (lerro anitzeko iruzkina)  
Gauza asko komentatu nahi Badituzu iruzkin hau erabili.  
Lerro bukaeraraino idatzitakoa programak ez dio jaramorik egingo.

`{ }`(giltzak)  
Kode blokeen hasiera eta bukaera markatzen da Bigiztetan eta errutinetan erabilia

`;` (puntu-koma)  
Kode lerro bakoitza ; bukatu behar du. Konpilazio errore asko puntu-komak ahazteagatik izaten dira

### Aldagaiak

Programa bat azken finean agindu multzo bat da digitu Batzuk mugitzen duena. Aldagaiak mugimenduak Egiteko erabiltzen dira.

`int` (integer)

2 byte(16 bit)-etatko digitua. gordetzen du. Ez du dezimalik eta -32,768tik 32768 tarteko balioak gorde ditzake

`long` (long)

Integerrak txiki geratzen Direnean erabiliak. 4byte (32 bit) okupatzen du eta -2,147,483,648tik 2,147,483,648ra gorde dezake

`boolean` (boolean)  
Egia/gezurra balioak gordetzeko Oso RAM gutxi erabiltzen du

`float` (float)

Dezimalak duten digituentzat 4 bytes (32 bit okupatzen du eta -3.4028235E+38 3.4028235E+38 tarteko balioak gorde ditzake

`char` (character)

Karaktere bat gordetzen du ASCII kodea (ie 'A' = 65). Byte bat (8 bit) erabiltzen du. Arduinoak string-ak arrayak bezala tratatzen ditu.

## Eragile matematikoak

Zenbakiak manipulatzeko eragiketak.

= (esleipena) (eg.  $x = 10 * 2$  (x orain 20 balio du))  
% (modulo) zatiketaren hondarra itzultzen du  
(ex.  $12 \% 10 = 2$ )  
+ (gehiketa)  
- (kenketa)  
\* (biderketa)  
/ (zatiketa)

## Konparazio eragileak

Eragiketa logikoak egiteko

== (bedina da) (ad.  $11 == 10$  FALSE edo  $12 == 11$  TRUE da)  
!= Ez da berdina (ad.  $12 != 10$  TRUE edo  $12 != 12$  FALSE da)  
< (txikiagoa) (ad.  $12 < 10$  FALSE edo  $12 < 12$  FALSE da)  
> (handiagoa) (ad.  $12 > 10$  TRUE edo  $12 > 12$  FALSE da)

## Kontrol egiturak

Programaren fluxua kontrolatzen dute, zer exekutatu da gero (Interneten gehiago Topatuko ditugu)

```
if(baldintza){ }  
else if(baldintza ){ }  
else { }  
Baldintza egia bada giltzen arteko kodea exekutatu da, bestela elseko kodea exekutatu da
```

```
for(int i= 0; i< # errep-kop; i++){ }  
Kode zati bat etengabe errepikatu nahi dugunean erabilia. (i++ edo i-- erabil dezakegu)
```

## Digitala

```
pinMode(pin, modua);  
Pinen erabilera modua definitzeko balio du. Pin aldagaian Pinzenbakia (0-19 (0-5 analo 14-19.ak dira). Modua INPUT edo OUTPUT izan daiteke.
```

```
digitalWrite(pin, balioa);  
Pin bat OUTPUT bada, HIGH (+5Vrekin elikatuta) edo LOW izan daiteke.
```

```
int digitalRead(pin);  
Pin bat INPUT bada, HIGH (+5V) edo LOW (lurrera Konektatuta irakur daiteke.
```

## Analogikoa

Arduinoak makina digitalak dira baina badakite mundu analogikoarekin ibiltzen. (trikimailuen bidez) asko Erabiltzen da munduarekin elkarreragiteko.

```
int analogWrite(pin, balioa);  
Arduinoaren pin batzuk PWM (pultsu bidezko modulazioa) onartzen dute. Pina oso azkar piztu eta itzaltzen du Irteera analogiko bat emulatuz. Irteeran 0 (0v) eta 255 (+5V) arteko balioak izango ditugu.
```

```
int analogRead(pin);  
Pin analogikoak sarrera bezala Definitzean boltaia irakur dezakegu. 0 (0V) eta 1024 (5V) arteko balioak itzuliko dizkigu
```



(ARDX) .:Arduino Esperimentazio Kita: (ARDX)

### Egingo duguna

LEDak (light emitting diodes edo argia igortzen duten diodoak euskaraz), edozein motatako eraikuntza edo muntai dotoretan erabiltzen direnez Kit honetan sartu izan ditugu. Gauza erraz batekin hasiko gara, LED bat piztu eta itzaliz, etengabe, keinada edo blink efektu atsegin bat lortuz. Lehenik, jakin ezazu behean zerrendatutako erabiliko diren elementuak, konexio eskema zure breadborrera eraman ezazu ondoren eta azkenik konekta ezazu den dena. Behin zirkuitua eraiki duzula, programa kargatu beharko duzu. Horretarako, konekta ezazu Arduino plaka zure ordenagailuko USB portu batera. Ondoren aukeratu portu egokia Herramientak > Serie Ateka > (Arduino plakaren komunikazio portua). Hau egin eta gero programa kargatu ahal izango duzu hurrengo esteka jarraituz Fitxategia > Kargatu I/O plakara (ctrl+U). Azkenik argiak kontrolatzeko aukeraz goza ezazu.

Programa kargatzerakoan arazoak izanez gero, hauek konpontzeko erabateko gida hemen aurki dezakezu: <http://ardx.org/TRBL>

### Zirkuitua

#### Osagaiak



CIRC-01  
Breadboard plaka  
x1



2 Pin buru  
x4



10mm LED  
x1

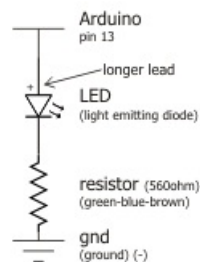


Kable



560 Ohm Erresistentzia  
Berde-Urdin-Marroi  
x1

#### Eskema



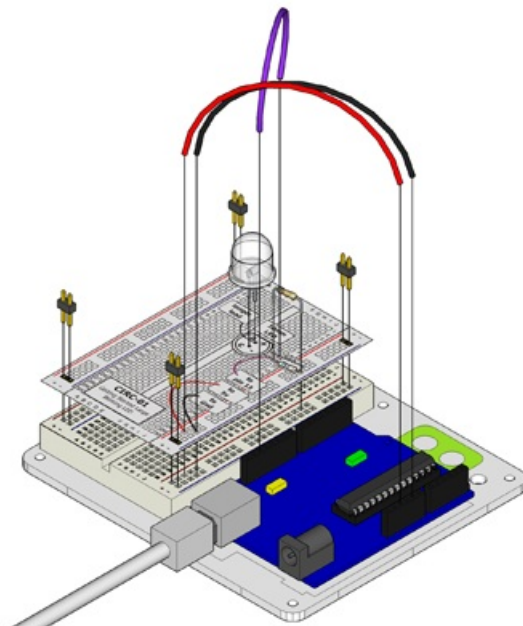
#### Baliabideak

.:jaitsi:.

breadboard-aren diseinu-orria  
<http://ardx.org/BBL01>

.:ikus:.

muntaiaren bideoa  
<http://ardx.org/VIDE01>



**Kodea** (ez duzu zertan dena idatzi behar)

**Fitxategia > Adibideak > 1.Oinarrizkoa > keinu egin**

(**arduino.cc** webgune ikusgarritik hartua, beste ideia onen bila bazabiltza hemen begiratu)

/\*

\*Keinu egin

\*LED bat pizten du du segundu batez, ondoren itzaltzen du segundu batez. etengabe.

\*Zirkuitua:

\* LEDa pin digital 13tik lurrara konektaturik.

\* Nota: Arduino plaka gehienek, badaukate jada LED bat plakan bertan 13. pinera konektaturik

\* beraz, ez da beste elementu extrarik behar adibide honetarako.

\*Sortze-Data: 2005eko Ekainaren 1

\*Egilea: David Cuartielles

\*<http://arduino.cc/en/Tutorial/Blink>

\*H. Barraganen kodean oinarrituta ;Wiring s/i plakarentzat <

\*/

```
int ledPin = 13; // pin digital 13ra konektatutako LEDa
```



```
// setup() funtzioa behin exekutatzen da, sketcha hastean

void setup() {
  // Pin digitala irteera bezala hasieratu:
  pinMode(ledPin, OUTPUT);
}

// loop() funtzioa etengabe exekutatzen da,
// Arduinoa elikatua dagoen heinean.

void loop()
{
  digitalWrite(ledPin, HIGH); // LEDa piztu
  delay(1000);                // segundu bat itxoin
  digitalWrite(ledPin, LOW);  // LEDa itzali
  delay(1000);                // segundu bat itxoin
}
```

## Ez dabil? (3 posible soluzio)

**LEDa ez da pizten?**  
LEDak soilik norabide egokian argitzen dute. Saiatu hau 180 gradu biratuz konektatzen. (ez kezkatu, alderantziz montatzeak ez dakar betirako ondorio txarrik).

**Programa ez du kargatzen**  
Batzutan gertatzen da hau, gehienetan serie ateka desegokia identifikatzeagatik, hau alda dezakezu **herramintak>serie ateka>**

**Oraindik ez duzula lortu?**  
Zirkuitu lan egokirik ez egitea ez da batere atsegina, mezu elektronikoko bat bidalguz eta ahalik eta laisterren zurekin harremanetan jarriko gara.

[help@oomlout.com](mailto:help@oomlout.com)

## Hobeto egiteko prest?

### Pina aldatu:

LEDa 13. pinera konektaturik dago, baina Arduinoaren edozein pin erabil dezakezu. Aldatzeko konekta ezazu 13. pinera lotutako kablea zure aukerako beste edozein batera (0tik 13ra) (nahi izanez gero 0-5 analogikoak erabil ditzakezu. 0 14 da...)

Ondoren kodean alda ezazu errenkada:

```
int ledPin = 13; -> int ledPin = pinberria;
```

Gero kargatu programa: (ctrl-u)

### Aldatu keinada denbora:

Ez dela nahikoa segundu bakarra piztu ta bakarra itzali?

Alda ezazu kodean hurrengo errenkadak:

```
digitalWrite(ledPin, HIGH);
delay(pizten deneko denbora); //(segunduak * 1000)
digitalWrite(ledPin, LOW);
delay(itzaltzen deneko denbora); //(segunduak * 1000)
```

### Disdiraren :

Digital (on/off) kontrolarekin Arduinoak zenbait pin kontrola ditzake analogikoki (disdira). (xehetasun gehio ondorengo zirkuituetan). Honekin jokatzeko.

Aldatu LEDa bai eta kablea 9. pinera

```
ledPin = 13; -> int ledPin = 9;
```

Ordezka ezazu kode hori kode nagusiaren { } class="bigCode">loop() beste honekin:

```
analogWrite(ledPin, zenbaki berria);
```

(zenbaki berria) = edozein zenbaki 0 eta 255 bitartean. 0 = off, 255 = on, erdiko tartetan = disdira ezberdina

**Desagertze efektua:**

Erantsitako beste programa erabiliko dugu. Irekitzeko joan zaitez.

**Fitxategi > Adibideak > Analogiko > Desagertzea (Fading)**

Ondoren kargatu programa eta ikusi nola desagertzen den eta berriz agertu ere.



(ARDX) .:Arduino Esperimentazio Kita:. (ARDX)

### Egingo duguna

Keinada efektua LED bakarrean sortzeko gai izan gara, baina orain maila igotzeko garaia iritsi da. Zortzi konektatuko ditugu. Arduinoa pixkat gehiago ustiatzeko aukera daukagu ere, argiztapen sekuentzia ezberdinak sortuz. Zirkuitu hau oso egokia da norberak egindako programekin saiakerak egiteko, baita Arduinoak berak nola lan egiten duen ulertzeko.

LEDekin trebatzen garen bitartean, zure programa motzak bezain kuttunak uzteko zenbait metodo ikusiko ditugu.

() loop-ak (bigiztak) - kode zati bat etengabe errepikatzea nahi denean erabiliak.  
array[]-ak - aldagaiak errazago maneiatzeko erabiliak (aldagai talde bat da).

### Zirkuitua

#### Osagaiak



CIRC-02  
Breadboard plaka  
x1



560 Ohm-eko  
erresistentzia  
Berdea-Urdina-Marroia  
x8



2 Pin  
buru  
x4

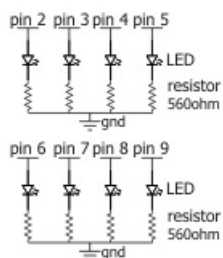


5mm LED Berdea  
x8



Kable

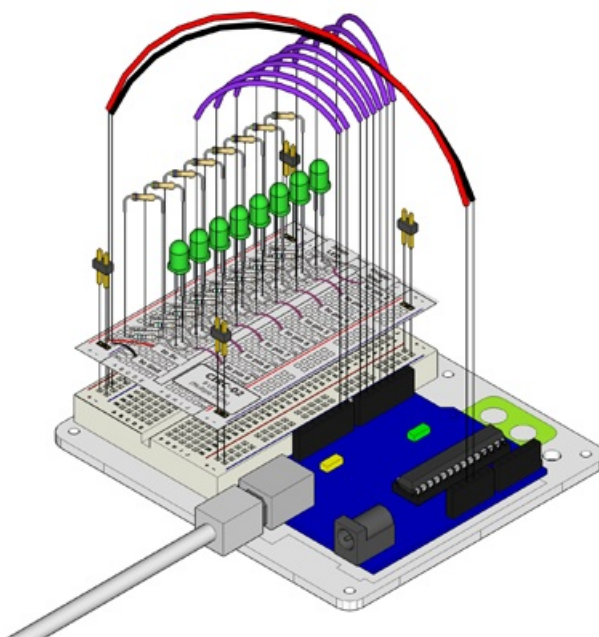
#### Eskema



#### Baliabideak

.:jaitsi.:  
breadboard-aren diseinu-orria  
<http://ardx.org/BBS02>

.:ikus.:  
muntaiaren bideoa  
<http://ardx.org/VIDE02>



### Kodea (ez duzu zertan dena idatzi behar)

Jaitsi ezazu kodea ( <http://ardx.org/CODE02> )

(eta ondoren kopia ezazu testua Arduinoaren sketchean)

```

/* -----
 * | Arduino Saiakera Kita Adibidezko Kodea |
 * | CIRC-02 .: 8 LEDekin Jolasa .: (aunitz LED) |
 * -----
 *
 * LEDen animazio erraz batzuk
 *
 * Zirkuitu honi buruzko informazio gehiago http://tinyurl.com/d2hrud
 *
 */

//LED Pin Aldagaiak
int ledPinak[] = {2,3,4,5,6,7,8,9}; //Array bat non LED bakoitza zein pinera konektaturik
//dagoen adierazten den
//adbz. LED #0 konektaturik dago 2. pinera LED
//#1, 3. eta horrela mantendu

```

```

//array bat helbideratzeko erabili ledPins[0] honen
//baliokidea izango 2
//and ledPins[7] honen baliokide 9

/*
 * setup() - funtzio honek soilik behin exekutatzen da Arduinoa pizterakoan
 * Guk kontroleko hiru pinak irteera bezala izendatuko ditugu
 *
 */
void setup()
{
    /* Ezar ezazu LEDetara konkektatutako pin bakoitza irteera (output) bezala ( maila
    * altua HIGH(on) edo maila baxua LOW(off))*/
    for(int i = 0; i < 8; i++){ //hau bigizta bat da eta zortzitan errepikatuko da
        pinMode(ledPins[i],OUTPUT); //LED bakoitzaren pina irteera bezala ezarri
    } //kode horrek beheko hau ordezkatzten du

    /* (hemen idatzitako kode laguntzailea ez da exekutatuko)
    * hauek dira goiko bigiztak ordezkatzten dituen errenkadak.
    * Goikoaren berdin berdina exekutatuko du
    * baina testu gutxiago erabiliz
    pinMode(ledPins[0],OUTPUT);
    pinMode(ledPins[1],OUTPUT);
    pinMode(ledPins[2],OUTPUT);
    pinMode(ledPins[3],OUTPUT);
    pinMode(ledPins[4],OUTPUT);
    pinMode(ledPins[5],OUTPUT);
    pinMode(ledPins[6],OUTPUT);
    pinMode(ledPins[7],OUTPUT);
    (laguntzako kodearen amaiera)*/
}

/*
 * loop() - funtzio hau setup amaitu ostean hasiko da eta ondoren errepikatu
 * oneAfterAnother izeneko funtzioa deituko dugu). Beste jokaera ezberdin bat nahi izanez gero
 * komentatu barik (ezabatu bi barratxoak) beste errenkadetako bakoitza
 */
void loop() // behin eta berriz exekutatu
{
    oneAfterAnotherNoLoop(); //Honek LED bakoitza banan banan piztu eta ondoren itzaliko du
    //oneAfterAnotherLoop(); //berdina egiten du oneAfterAnotherNoLoop bezala baina
    //askoz testu gutxiagorekin
    //oneOnAtATime(); //honek LED bakarra piztuko du eta itzaltzean hurrengoa piztuko
    //da etengabe;
    //biratuz (Piztutako argia mugitzen dela ematen du, errenkada
    //higikorra irudituz
    //inAndOut(); //erdiko bi LEDak piztu, gero hauek mugitu erditik kanpoalderantz
    //eta ondoren berriz erdian kokatzen ditu
}

/*
 * oneAfterAnotherNoLoop() - LED bat piztuko du eta delayTime denboraz itxaron hurrengo
 * LEDa berriro pizteko
 * eta horrela denak piztu arte, ondoren denak itzali baina banan banan
 *
 * Bigiztarik gabe egin dezake hau baina testu luzeagoa idatzi behar da
 * oneOnAtATimeLoop() emaitza bera lortzen da askoz kode gutxiagorekin
 */
void oneAfterAnotherNoLoop(){
    int delayTime = 100; //itxarote denbora (milisegundutan) LED artean
    //egin ezazu azkarragoa txikiagotuz eta handituz, aldiz mantsoagoa
    digitalWrite(ledPins[0], HIGH); //LED #0 pizten du (2. pinari konektatua
    delay(delayTime); //itxaron delayTime milisegundutan
    digitalWrite(ledPins[1], HIGH); //LED #1 pizten du (3. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan
    digitalWrite(ledPins[2], HIGH); //LED #2 pizten du (4. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan
    digitalWrite(ledPins[3], HIGH); //LED #3 pizten du (5. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan
    digitalWrite(ledPins[4], HIGH); //LED #4 pizten du (6. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan
    digitalWrite(ledPins[5], HIGH); //LED #5 pizten du (7. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan
    digitalWrite(ledPins[6], HIGH); //LED #6 pizten du (8. pinari konektatua )
    delay(delayTime); //waits delayTime milliseconds
    digitalWrite(ledPins[7], HIGH); //LED #7 pizten du (9. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan

    //LED bakoitza itzaltzen du
    digitalWrite(ledPins[7], LOW); //LED #7 itzaltzen du (9. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan
    digitalWrite(ledPins[6], LOW); //LED #6 itzaltzen du (8. pinari konektatua )
    delay(delayTime); //itxaron delayTime milisegundutan
}

```

```

digitalWrite(ledPins[5], LOW); //LED #5 itzaltzen du (7. pinari konektatua )
delay(delayTime); //itxaron delayTime milisegundutan
digitalWrite(ledPins[4], LOW); //LED #4 itzaltzen du (6. pinari konektatua )
delay(delayTime); //itxaron delayTime milisegundutan
digitalWrite(ledPins[3], LOW); //LED #3 itzaltzen du (5. pinari konektatua )
delay(delayTime); //itxaron delayTime milisegundutan
digitalWrite(ledPins[2], LOW); //LED #2 itzaltzen du (4. pinari konektatua )
delay(delayTime); //itxaron delayTime milisegundutan
digitalWrite(ledPins[1], LOW); //LED #1 itzaltzen du (3. pinari konektatua )
delay(delayTime); //itxaron delayTime milisegundutan
digitalWrite(ledPins[0], LOW); //LED #0 itzaltzen du (2. pinari konektatua )
delay(delayTime); //itxaron delayTime milisegundutan
}

/*
 * oneAfterAnotherLoop() - LED bat piztuko du eta gero itxaron delayTime hurrengo
 * LEDa pizteko LED guztiak piztu arte, ondoren denak itzali banan banan ere
 *
 * hau bigizta bat erabiliz lortua izan da, kode askoz gutxiago erabiliz.
 * than oneOnAtATimeNoLoop() does exactly the same thing with less typing
 */
void oneAfterAnotherLoop(){
    int delayTime = 100; //itxarote denbora (milisegundutan) LED artean
    //egin ezazu azkarragoa txikiagotuz eta handituz, aldiz mantsagoa

//LED bakoitza bestearen ostean pizten du
for(int i = 0; i <= 7; i++){
    digitalWrite(ledPins[i], HIGH); //LED #i pizten du i aldia exekutatzen den bakoitzean
    delay(delayTime); //etengabe errepikatzen denez itxarote denbora
    //berdina izango da 7 alditan

//LED bakoitza bestearen ostean itzaltzen du
for(int i = 7; i >= 0; i--){ //goian bezala baina 0. pinatik gora hasi ordez 7.tik
    //hasten da eta beherantz gainera
    digitalWrite(ledPins[i], LOW); //LED #i pizten du i aldia exekutatzen den bakoitzean
    delay(delayTime); //aldi bakoitzean bat txikitzen denez 0 izan arte
    //errepikatuz itxarote denbora
    //berdina izango da

}

/*
 * oneOnAtATime() - LED bat pizten du eta ondoren hurrengo beste LED guztiak itzaliz
 */
void oneOnAtATime(){
    int delayTime = 100; //itxarote denbora (milisegundutan) LED artean
    //egin ezazu azkarragoa txikiagotuz eta handituz, aldiz mantsagoa

for(int i = 0; i <= 7; i++){
    int offLED = i - 1; //Zein LED piztuta zegoen kalkulatzeko du honen bitartez
    if(i == 0) { //i = 7 izan arte i minus 1 (Adbz. i = 2 kasurako
        //piztu 2. LEDa eta itzali 1. LED)
    } //hala ere i = 0 denean ez dugu nahi 0. LEDa itzaltzea -(ez dago)
    //7. LEDa itzali beharrean (inguruan biraka)
    digitalWrite(ledPins[i], HIGH); //i. LEDa pizten du
    digitalWrite(ledPins[offLED], LOW); //aurretik piztutako azken LEDa itzaltzen du
    delay(delayTime);
}
}

/*
 * inAndOut() - honek erdiko bi LEDak piztuko ditu, gero ondoko
 * bazterretako biak, ondoren beste kanpoko biak
 * Argia sartzen eta ateratzen deneko itxura
 */
void inAndOut(){
    int delayTime = 100; //itxarote denbora (milisegundutan) LED artean
    //egin ezazu azkarragoa txikiagotuz eta handituz, aldiz mantsagoa

//Argia erdiko lekuetatik kanpoalderantz ateratzen du
for(int i = 0; i <= 3; i++){
    int offLED = i - 1; //Zein LEDa piztuta zegoen kalkulatzeko du honen bitartez
    if(i == 0) { //i = 1, 7 izan arte i minus 1 (adbz. i = 2 izanik piztutako azkena
        3. izango da; //piztu 2. LEDa eta itzali 1. LEDa)
    } //hala ere i = 0 denean ez dugu nahi 1. LEDa itzaltzea -(ez dago)
    //7. LEDa guk itzali beharrean (inguruan biraka)
    int onLED1 = 3 - i; //hau da martxan jarritako lehenengo LEDa adbz. LED #3 i = 0
//izanik eta LED //
    // #0 i = 3 denean
    int onLED2 = 4 + i; //hau da martxan jarritako lehen LEDa adbz. LED #4 i = 0 denean
// eta LED
//
// #7 i = 3 izanda
    int offLED1 = 3 - offLED; //piztu dugun azken LEDa itzaltzen du

```

```

int offLED2 = 4 + offLED; //piztu dugun azken LEDa itzaltzen du

digitalWrite(ledPins[onLED1], HIGH);
digitalWrite(ledPins[onLED2], HIGH);
digitalWrite(ledPins[offLED1], LOW);
digitalWrite(ledPins[offLED2], LOW);
delay(delayTime);
}

//LEDak erdirantz bideratzeko
for(int i = 3; i >= 0; i--){
  int offLED = i + 1; // Zein LED piztuta zegoen kalkulatzeko du honen bitartez
  if(i == 0) { //i = 1, =7 izan arte i minus 1 (Adbz. i = 2 kasurako piztuta egon
    den azkenekoa 7.,; //piztu 2. LEDa eta itzali 1. LED)
  } //hala ere i = 0 denean ez dugu nahi 1. LEDa itzaltzea -(ez dago )
  //7. LEDa itzali beharrean (inguruan biraka)
  int onLED1 = 3 - i; //hau da martxan jarritako lehenengo LEDa adbz. LED #3 i = 0
//izanik eta LED
  //
  // #0 i = 3 denean
  int onLED2 = 4 + i; //hau da martxan jarritako lehen LEDa adbz. LED #4 i = 0 denean
// eta LED
  //
  // #7 i = 3 izanda
  int offLED1 = 3 - offLED; //piztu dugun azken LEDa itzaltzen du
  int offLED2 = 4 + offLED; //piztu dugun azken LEDa itzaltzen du

  digitalWrite(ledPins[onLED1], HIGH);
  digitalWrite(ledPins[onLED2], HIGH);
  digitalWrite(ledPins[offLED1], LOW);
  digitalWrite(ledPins[offLED2], LOW);
  delay(delayTime);
}
}

```

### Ez dabil? (3 posible soluzio)

#### LED batzuk ez dute argitzen

Oso erraza da LEDa alderantziz konektatzea. Konproba ezazu ez dabilenak norabide egokian ez egoteagatik ote diren.

#### Sekuentzia ez du errespetatzen

Zortzi kable izanda ohikoa da baten bat nahastea. Bitan ziurtatu konekzioa, lehenengo LEDa 2. pinera dago eta horrela jarraian.

#### Berriz saiatu

Kablen bat konekzioa egin gabe geratzea ez da arraroa. Kendu den dena, utzi dena garbi eta prest eta ekin berriz muntaiari, errazagoa izaten da hau akatsa bilatzea baino

### Hobeto egiteko prest?

#### Bigiztaak konmutatu:

loop() funtzioan 4 errenkada daude. Azkenengo hirurak '/' batekin hasten dira, beraz jakin bezala programa nagusian ez dira exekutatu, azalpen edo iruzkinak bezala idatziak izan direlako, egilearen komentarioentzat bezala. Bigiztaz aldatzen duen exekutatzeko soilik aldatu bi barra hori void loop() code to:

```

//oneAfterAnotherNoLoop();
oneAfterAnotherLoop();
//oneOnAtATime();
//inAndOut();

```

Kargatu programa eta ziurtatu ez dela ezer aldatzen. Begirada bat bota diezaiokezu bi funtzioetara, zeintzuk berdina egiten duten, baina ikuspuntu desberdinekin (iradokizuna, bigiztaduna)

#### Joku estrak:

Sekuentziak erabat asperturik? Orduan froga itzazu beste bi joku simple. Azalpenak deuseztatu eta karga ezazu programa zure zirkuituan. (borra itzazu 3. eta 4. zutabeetako marratxoak)

#### Zure sekuentziekin sakondu:

Atxikitutako kodean murgildu zaitez eta hasi gauzak aldatzen. Garrantzitsuena LED bat pizteko agindua

```
digitalWrite(pinNumber, HIGH);
```

eta itzaltzekoa

```
digitalWrite(pinNumber, LOW);
```

argi izatea da. Idaztean hutsune zabalak utzi, ez baidute eraginik programan.



### Egingo duguna

Arduinoaren pinak oso aproposak dira tentsio txikiago osagaiak kontrolatu ahal izateko, adibidez LEDak kontrolatzeko ikusi berri dugun moduan. Hala ere, tentsio handiagoko osagaiak kontrolatu behar direnean (hala nola jostailuzko motorren edo garbigailuaren kontrola), kanpoko transistore baten beharra dago. Transistoreek erabilgarritasun ikaragarria daukate. Korrante handi bat konmutatu dezakete korronte askoz txikiago batekin. Transistoreek 3 pin dituzte beti. Negatibo motako transistoreetan (NPN) edo negatiboentzat, karga kolektoreko pinera konektatzen da, eta emisorearena lurrara. Gero, korrante txiki batek basetik emisora jariatzen denean, transistoretik korrantea jarioko da baita zure motorra biratzen hasi ere (hau Arduinoko pina HIGH bezala ezartzean gertatzen da). Milaka transistore ezberdinak daude, literalki, egoera bakoitza era egokienean kontrolatua izateko. Guk P2N2222AGa aukeratu dugu asmo edo helburu orokorran oinarrituz. Gure kasurako faktore garrantzitsuenak diren tentsio maximoa (40V) eta korrante maximoa (200mA) nahikoak dira gure jostailuzko motorrentzat (xehetasun guztiak hurrengo datasheetean aurki daitezke <http://ardx.org/2222>).

(1N4001 diodoa flyback diodoa bezala jokatzen du, zergatia jakiteko bisita ezazu: <http://ardx.org/4001>))

### Zirkuitua

#### Osagaiak



CIRC-03  
Breadboard orria  
x1



2 Pin  
buru  
x4



Transistorea  
P2N2222AG (TO92)  
x1



Kable



Jostailuzko  
Motorea  
x1

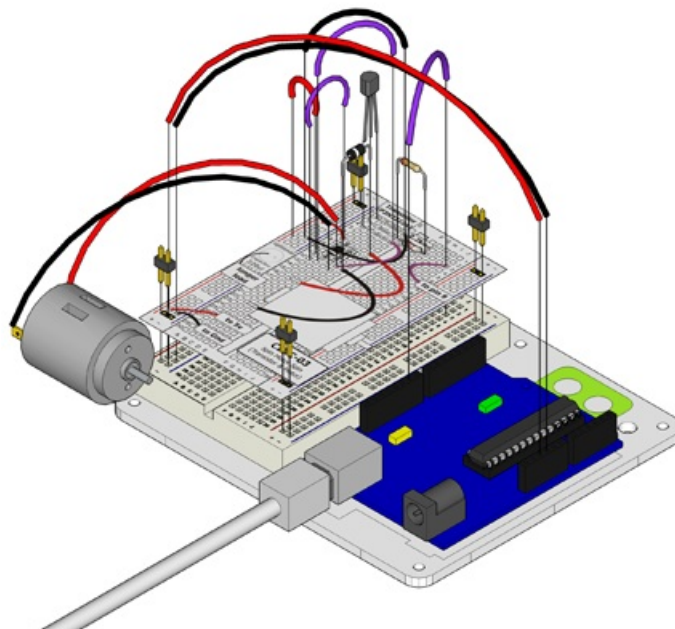
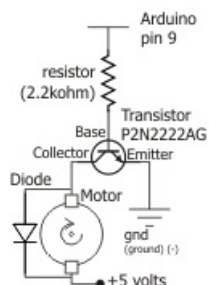


Diodoa  
(1N4001)  
x1



10k Ohm-eko  
Erresistentzia  
Marroi-Beltza-Laranja  
x1

#### Eskema



#### Baliabideak

.:jaitsi.:  
breadboard diseinu-orria

<http://ardx.org/BBS03>

.:ikusi.:  
muntaiaren bideoa

<http://ardx.org/VIDE03>

### Kodea (ez duzu zertan dena idatzi behar)

Jaitsi ezazu kodea ( <http://ardx.org/CODE02> )

(eta ondoren kopia ezazu textua Arduinoaren sketchean)

```

/* -----
 * | Arduino Saiakera Kita Adibidezko Kodea |
 * | CIRC-03 .: Biratu Motorea Biratu .: (Transistor and Motor) |
 * -----
 *
 * Arduinoaren pinak guztiz aproposak dira LEDen kontrolerako baina potentzi handiagoak behar
 * erabiltzen dituztenak kontrolatzeko
 * potentzia handitzen duen zerbait beharko duzu azkarregir apurtu nahi ez badituzu.
 * Elementu potenteagoak kontrolatu ahal izateko transistore baten laguntza beharko dugu.
 * Hemen transistore bat erabiliko da jostailuzko motore txiki bat kontrolatzeko

```



```

*
* http://tinyurl.com/d4wht7
*
*/

int motorPin = 9; // motoreari konektatutako pina definitzen du
                // (9.,10.,11. edota 3. pinak; erabiliz gero abiadura kontrola daiteke)

/*
* setup() - funtzio honek soilik behin exekutatzen da Arduinoa abiaraztean
* Motoreari esleitutako pina irteera bezala ezarri (pina egoera altuan jarriz (+5v)
* edo baxuan (lurra) (-))
* edo hobeto esanda sarrera bezala ez dela definitu (Pina egoera altuan
* edo baxuan dagoen konprobatuz
*/
void setup()
{
    pinMode(motorPin, OUTPUT);
}

/*
* loop() - funtzio hau setup amaitu ostean hasiko da eta ondoren errepikatu
* motorOnThenOff() izeneko funtzioa deitzen dugu
*/

void loop() // behin eta berriz exekutatu
{
    motorOnThenOff();
    //motorOnThenOffWithSpeed();
    //motorAcceleration();
}

/*
* motorOnThenOff() - motorea pizten du eta ondoren itzali
* (kode hau berbera LEDen keinada efektuan erabili genuenaren&nbsp;
berdina&nbsp;del&nbsp;nabarmendu zaitez
*/
void motorOnThenOff(){
    int onTime = 2500; //motorea pizturik egongo den denbora, milisekundutan
    int offTime = 1000; //motorea itzalirik egongo den denbora, milisekundutan

    digitalWrite(motorPin, HIGH); // Motorea abiarazten du
    delay(onTime); // itxaron onTime milisegundu bitartean
    digitalWrite(motorPin, LOW); // Motorea itzaltzen du
    delay(offTime); // itxaron offTime millisekunduz
}

/*
* motorOnThenOffWithSpeed() - motorea pizten du baina abiaduraren kontrolarekin ere
* (kode hau berbera LEDen keinada efektuan erabili genuenaren&nbsp;
berdina&nbsp;del&nbsp;nabarmendu zaitez
*/
void motorOnThenOffWithSpeed(){

    int onSpeed = 200; // 0 (geldirik) eta 255 (abiadura maximoan) bitarteko edozein zenbaki
    int onTime = 2500; ///motorea pizturik egongo den denbora, milisekundutan

    int offSpeed = 50; // 0 (geldi) eta 255 (abiadura maximoa) bitarteko zenbakia
    int offTime = 1000; // motorea geldirik iraungo duen milisegundu kopurua

    analogWrite(motorPin, onSpeed); // Motorea martxan jartzen du
    delay(onTime); // onTime millisegundu kopuruaz itxoin
    analogWrite(motorPin, offSpeed); // Motorea itzaltzen du
    delay(offTime); // onTime millisegundu kopuruaz itxoin
}

/*
* motorAcceleration() - motorea azelerazio maximoa izan arte azeleratu ondoren
* azelerazioa berriro 0 izan arte jaisten du
*/
void motorAcceleration(){
    int delayTime = 50; //abiadura urratsen arteko denbora

    //Motorea azeleratzen du
    for(int i = 0; i < 256; i++){ //abiadura baoitza 0tik 255ra arte
        analogWrite(motorPin, i); //Abiadura berria ezartzen du
        delay(delayTime); // delayTime millisegundu kopuruaz itxoin
    }

    //Motorearen abiadura txikitzen du
    for(int i = 255; i >= 0; i--){ ///abiadura baoitza 0tik 255ra arte
        analogWrite(motorPin, i); //Abiadura berria ezartzen du
        delay(delayTime); // delayTime millisegundu kopuruaz itxoin
    }
}

```

## Ez dabil? (3 posible soluzio)

### Motorea ez du biratzen?

Motorea ez du biratzen?  
Transistorea zuk elikatzen  
baduzu, konproba ezazu  
bitan datasheetan irteerako  
pina bateragarria dela  
P2N2222AG-kin  
(aunitz erreserbatuak  
baitaude).

### Zorterik ez?

Motorea zuk elikatzen  
baduzu konproba ezazu bitan  
ere 5V-kin ibiltzen dela eta  
potentzi geihegirik erakartzen  
ez duela.

### Oraindik ezta?

Batzutan Arduino plaka  
ordenagailutik  
deskonektatzen da. Froga  
ezazu deskonektatuz eta  
berriro konketatuz zure USB  
portuan.

## Hobeto egiteko prest?

### Abiadura kontrola:

LED baten argitasuna kontrolatzeko Arduinoak duen gaitasuna ezagutu genuen aintzinago, orain funtzio hori berrerabiliko dugu motorearen abiadura kontrolatzeko. Hau lortzeko, Arduinoak modulazio bidezko pultsazioa (PWM) erabiltzen du. Eta hau, Arduinoak ikaragarri azkar jarduteko duen ahalmenean oinarritzen da. Pinatik datorren tentsioa kontrolatu beharrean, Arduinoak pinaren egoera konmutatuko du oso azkar. Ordenagailuen munduan 0-tik 5V-era segundu bakar batean askotan egin daiteke, baina gizakion munduan guk tentsio erresultantea ikusten dugu bakarrik. Adibidez Arduinoak PMW 50%-ra ezartzen badu, guk argia 50% ilunduta ikusiko dugu, gure begiek ez baitira gai hainbeste aldizkakotasun nabaritzeko. Berezitasun berbera trantsistoreekin gertatzen da. Ez didazula sinisten? Ekin ba!

`loop()` bukleko sekzioan alda ezazu hurrengoa honen ordeez.

```
// motorOnThenOff();  
motorOnThenOffWithSpeed();  
//motorAcceleration();
```

Gero karga ezazu programa. Abiadura zehazten duzu `onSpeed` eta `offSpeed` aldagaien balioak aldatuz.

### Azeleratu eta gelditu:

Zergatik bi abiadura izanda gelditzen da?? Zergatik ez motorea azeleratu eta gelditu. Lortzeko soilik alda ezazu `loop()` irakurtzeko kodea

```
// motorOnThenOff();  
// motorOnThenOffWithSpeed();  
motorAcceleration();
```

Ondoren kargatu programa eta ikusi nola zure motorea era mantsoan azeleratzen den abiadura maximoa izan arte, gero berriro era mantsoan ere gelditzen dihoan. Azelerazio abiadura aldatzea nahi izanez gero, alda ezazu `delayTime` (balio handiagoa azelerazio denbora luzeagoa eragingo du).



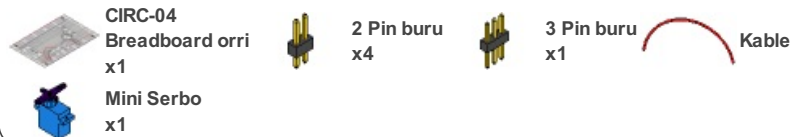
(ARDX) .:Arduino Esperimentazio Kita:. (ARDX)

## Egingo duguna

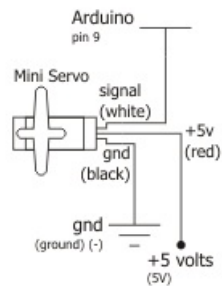
Motore bat biratzea primerako entretenimendua da baina mugimenduaren kontrola beharrezkoa duten proiektuak datozenean, honekin ez gara asetuko edo ez da nahikoa izango. Orduan? Hobby serboak. Masan ekoizten dira, sorta handia dago eskuragarri eta euro pare batetik ehundaka eurotara balio dezakete. diren mezatara dira ekoiztu, zabalduena eskuragarri eta ezer kostatuko dolar pare bat ehunka. Barruan aldaketa kaxa txiki bat dago (mugimendua askoz ahaltsuagoa egiteko) eta elektronikako zenbait osagai (kontrolatzea errazteko). Serbo estandarrek 0 eta 180 bitarteko graduetan finka daitezke. Non finkatzea, denbora pultsazio batek kontrolatzen du, 1.25 milisegundo (0 gradu) eta 1,75 milisegundo (180 gradu) bitartean (1,5 milisegundo 90 graduko). Denbora hauek egilearen arabera aldatzen dira. Pultsua 25-50 milisegundotan behin bidaltzen bada serboa leunki jokatuko du. Arduinoaren ezaugarri nagusietako bat eranstean duen software-liburutegia da, non bi serbo batera kontrolatzeko aukera (pin 9 edo 10 konektatuta), kode lerro bakar baten truke ematen den.

## Zirkuitua

### Osagaiak



### Eskema



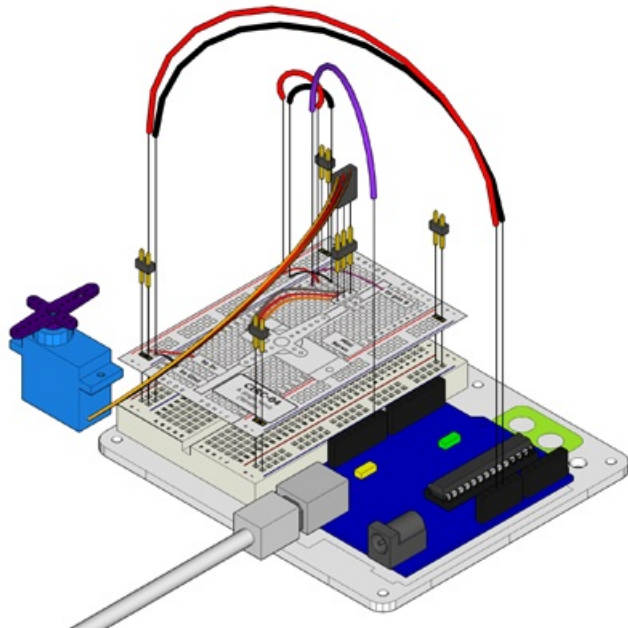
### Baliabideak

..Jaitsi:.

breadboard-aren diseinu-orria  
<http://ardx.org/BBS04>

..ikusi:.

muntaiaren bideoa  
<http://ardx.org/VIDE04>



## Kodea (Ez da zertan den dena idatzi behar)

### Fitxategia > Adibideak > Serbo > Sweep

( arduino.cc txoko miresgarriaren adibide soil bat, beste ideei harrigarriak nahi izanez gero begira ezazu)

```
// Sweep
// BARRAGANek egina

#include <Servo.h>

Servo myservo; // serbo objektu bat sortzen du serboa kontrolatzeko
                // gehienez zortzi serbo objektu sor daitezke

int pos = 0; // serboaren tokia gordetzeko

void setup()
{
  myservo.attach(9); // 9. pinan dagoen serboa serbo objektuarekin lotzen du;
}
```

```

void loop()
{
  for(pos = 0; pos < 180; pos += 1) // 0 gradutik 180 gradura higitzen da
  {
    myservo.write(pos);           // gradu bateko urratsekin
    delay(15);                     // serboari 'pos' aldagaiko posizioa joateko esaten dio
  }
  for(pos = 180; pos >= 1; pos -= 1) // 180 gradutik 0 gradura higitzen da
  {
    myservo.write(pos);           // serboari 'pos' aldagaiko posizioa joateko esaten dio
    delay(15);                     // 15ms itxaron serboak posizioa hel dadin
  }
}

```

### Ez dabil? (3 posible soluzio)

#### Serboa ez du biratzen?

Nahiz eta koloredun kableak erabili arren izugarri erraz da serbo alderantziz konektatzea. Hau izan daiteke arrazoia.

#### Oraindik ez?

Serboa elikatzeaz ahaztu izatea izan zen mugimendua ekiditzen izan zen behin edo bitan guri gertatutakoa, beraz konproba ezazu kable gorria 5V-era eta marroia lurrara konektaturik dauden.

#### Fits and Starts

If the servo begins moving then twitches, and there's a flashing light on your Arduino board, the power supply you are using is not quite up to the challenge. Using a fresh battery instead of USB should solve this problem.

### Hobeto egiteko prest?

#### Potentiometroarn bidezko kontrola:

Orain soilik sarrerekin jokatu dugu, baina gehiago nahi izanez gero, tira, hemen duzu adibidezko programa Fitxategia > Serbo-Liburutegia > Knob. Potentiometro bidezko kontrola erabiltzen du (CIRC08). Argibideak online aurki ditzakezu hemen: <http://ardx.org/KNOB>

#### Auto sinkronizazioa:

Serboaren kontrola Arduino dakarren liburutegia erabiltzea oso erakargarria izan arren, noizbait dibertigarria izan daiteke zuk zeuk programatzea, helburua lortzeko bidea zuk erabakiz. saia zaitetz. Pultsazioa zunenean kontrolatzen gabilta beraz 20 pin eskuragarri dituen edozein Arduino froga dezakezu (kodea txukundu beharko duzu probatu baino lehen).

```

int servoPin = 9;

void setup() {
  pinMode(servoPin, OUTPUT);
}

void loop() {
  int pulseTime = 2100; // (mikrosegundu kopurua
                        // geltokiak (1500 90 gradu
                        // 900 0 gradu 2100 180 gradu)

  digitalWrite(servoPin, HIGH);
  delayMicroseconds(pulseTime);
  digitalWrite(servoPin, LOW);
  delay(25);
}

```

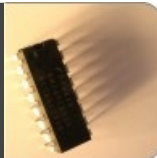
#### Idei bikainak:

Serboak era aunitzeko gauza ikaragarriak lortzeko erabil daitezke, hemen kutunetako batzuk.

Xmas Hit Counter  
<http://ardx.org/XMAS>

Open Source Robotic Arm (Arduino bezala serbo bidezko kontrola erabiltzen du)  
<http://ardx.org/RARM>

Servo Walker  
<http://ardx.org/SEWA>



(ARDX) .:Arduino Esperimentazio Kita:. (ARDX)

### Egingo duguna

Txipekin jolasteko garaia heldu da, zirkuitu integratuak (Zlak) bezala ezagutuak. Txiparen kanpoko itxura nahiko etsigarria izan daiteke. Arduino plakako txipa (mikrokontrolagailua) adibidez, edo guk zirkuitut honetan erabiliko duguna (shift erregistroa) oso antzekoak dirudite baina oso ezberdinak dira. Arduino plakako ATmega txipak dolar batzuk kostatzen du, 74HC595ak ordea, xentimo batzuk bakarrik balio du. Hasierako pausuak emateko txip egokia da hau, txipekin eta bere datu-orriekin gustora gaudenean (sarean eskuragarri <http://ardx.org/74HC595>) txipen mundua zure esku egongo da. Shift erregistroa (serie/paralelo bihurtzaile bezala ere ezagututa), 8 irteera gehiago eskainiko dizkigu (LEDak eta antzekoak kontrolatzeko) Arduinoko 3 pin bakarrik erabiliz. Elkarren artean konekta daitezke, lau pin bakarrik erabiliz ia mugarik gabeko irteerak izan ditzakegularik. Erabilpena honakoa da: lehenengo eta behin datuak "erlojuarekin gaitu" egiten dira, gero "itxi" egiten da (zarraila botatzen da). Hortarako datuen pina HIGH edo LOW bezala ezartzen dugu, erlojuaren pultsoa jaso, datuen pina gaitu berriz eta erlojuarekin pultsuekin batera datuak bidaltzen dira 8 bit bidali arte. Bukaeran zarraila aktibatzen da eta 8 bitak shift erregistroaren pinetara bidaltzen dira. Konplikatuak dirudi baina oso sinplea bihurtzen da frogak egin ondoren: <http://ardx.org/SHIF> )

### Zirkuitua

#### Osagaiak



CIRC-05  
Breadboard  
Orria  
x1



2 Pinetako Burukoak  
x4



Shift  
Erregistroa  
74HC595  
x1



Kablea

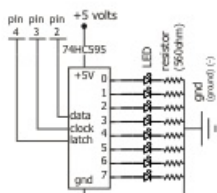


LED Gorriak  
x8



560 Ohmiotako  
Erresistentzia  
Berdea-Urdina-Marroia  
x8

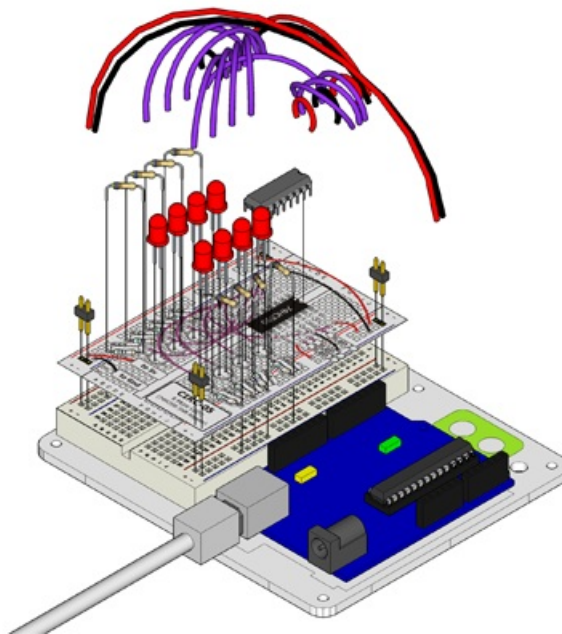
#### Eskema



#### Baliabideak

.:jaitsi:.  
breadboard diseinu-orria  
<http://ardx.org/BBLS05>

.:ikus:.  
muntaiaren bideoa  
<http://ardx.org/VIDE05>



### Kodea (ez duzu zertan dena idatzi behar)

Hemendik jaitsi ( <http://ardx.org/CODE05> )

(kopiatu eta itsatsi Arduinoko Sketch huts batean.)

```

/*
-----
*   | Arduino Esperimentazio Kita Adibideko Kodea   |
*   | CIRC-05 .: 8 LED gehiago .: (74HC595 Shift erregistroa) |
*   -----
*
* Iadanik 8 LED kontrolatu ditugu, orain modu eleganteago batean egingo dugu.
* 8 pin erabili ondez 3 bakarrik erabiliko ditugu, gehi txip bat
*
*
*
*/

```

```

*/

//Pinen Definizioak
//Pinen Definizioak
//74HC595-ak hiru pin erabiltzen dituen
//serie komunikazioko esteka erabiltzen du
int datuak = 2;
int erlojua = 3;
int zarraila = 4;

//LED bakarria kontrolatzeko
int egoeraLEDA = 0;
const int ON = HIGH;
const int OFF = LOW;

/*
 * setup() - funtzio hau behin exekutatzen da Arduinoa piztean.
 * Hiru kontrol pinak irteera bezala definitzen ditugu
 */
void setup()
{
  pinMode(datuak, OUTPUT);
  pinMode(erlojua, OUTPUT);
  pinMode(zarraila, OUTPUT);
}

/*
 * loop() - funtzio hau setup eta gero exekutatuko da eta etengabe errepikatzen egongo da.
 * Piztu nahi dugun LEDa definitu ondoren errutina baten bidez egoerak 74HC595ra bidaltzen
 dizkio
 */
void loop()                // etengabe exekutatzen
{
  int itxaronDenb = 100; //LEDen egoerak eguneratzeko milisegundu kopurua
  for(int i = 0; i < 256; i++){
    LEDakEguneratu(i);
    delay(itxaronDenb);
  }
}

/*
 * LEDakEguneratu() - LEDEgoerak funtzioak ezarritako LEDen egoerak 74HC595 bidaltzen dizkio
 *
 */
void LEDakEguneratu(int balioa){
  digitalWrite(zarraila, LOW);    //Txiparen zarraila aktibatzen du
  shiftOut(datuak, erlojua, MSBFIRST, balioa); // 8 bitak shift erregistroari bidali
  digitalWrite(zarraila, HIGH);  //zarraila aktibatzen du datuak erakutsiz.
}

/*
 * LEDakEguneratuLuzea() -LEDEgoerak funtzioak ezarritako LEDen egoerak 74HC595
 * bidaltzen dizkio
 * LEDakEguneratu bezalakoa, baina datu bidalketa software bidez egiten da prozesua
 * ikusi ahal izateko.
 */
void LEDakEguneratuLong(int balioa){
  digitalWrite(zarraila, LOW);    //Txiparen zarraila aktibatzen du
  for(int i = 0; i < 8; i++){      //8 aldiz errepikatuko du (behin bit bakoitzeko)
    int bit = balioa & B10000000; //Bit maskara erabiltzen dugu 8. bita
                                   //bakarrik hautatzeko(orain helbideratzen ari garena)
    balioa = balioa << 1;         //Bita posizio bat mugitzen dugu horrela hurrengo bueltan

                                   //7. bita 8.a izango da, gure matematikak egiteko behar
                                   //duguna.
    if(bit == 128){digitalWrite(datuak, HIGH);} //8. pina gaituta badago dure datuen
                                                //pinean HIGH idatzi
    else{digitalWrite(datuak, LOW);}           //8. pina desgaituta badago, LOW idatzi
    digitalWrite(erlojua, HIGH);              //Hiru lerro hauek erlojuaren pultsua
                                                //sortzen dute

    delay(1);
    digitalWrite(erlojua, LOW);
  }
  digitalWrite(zarraila, HIGH); //Zarraila ixten du datuak bidaliz
}

//LEDak banaka-banaka aldatzeko maskara hauek erabiltzen ditugu.
//Informazio gehiagorako http://en.wikipedia.org/wiki/Bitwise\_operation
int bitak[] = {B00000001, B00000010, B00000100, B00001000, B00010000, B00100000,
               B01000000, B10000000};

```

```
int maskarak[] = {B11111110, B11111101, B11111011, B11110111, B11101111, B11011111,
                  B10111111, B01111111};
/*
 * LEDaAldatu(int leda, int egoera) - LED bat aldatzen du
 * LEDak Otik 7ra doaz eta bere balioak 0 (OFF) edo 1 (ON) izan daiteke.
 */
void LEDaAldatu(int leda, int egoera){
    ledEgoera = ledEgoera & maskarak[leda]; //ledEgoeraren bita garbitzen du.
    if(egoera == ON){ledEgoera = ledEgoera | bitak[leda];} //Bita aktibatuta dagoenean idatzi
    LEDakEguneratu(ledState); //LEDen egoera berria shift erregistrora bidali
}
```

## Ez dabil? (3 gauza probatzeko)

**Arduinoaren elikadura LEDa itzaltzen da.**  
Batizutan gertatzen da, txipa aldrebes sartzen badugu. Azkar konponduz gero ez da ezer apurtuko.

**Ez dabil oso ongi**  
Sentitzen dut etengabe berdina errepikatzea, baina ziur aski kable bat gaizki dago.

**Etsita?**  
Eposta bat bidalguzu, zirkuitua erraza eta konplexua da aldi berean. Zuen arazoan berri jakin nahi dugu, hurrengo bertsioetan gehitzeko.  
help@oomlout.com

## Hobeto egiteko prest ?

### Dena eskuz eginda:

Arduinoak ekintza konplexuak errazteko aukera dauka, datuen korritze automatikoa adibidez. Dena den, Arduinoarekin gauzak erraztu ala konplikatu ditzakegu. Loop funtzioaren lerro hau aldatu:

```
updateLEDs(i) -> updateLEDsLong(i);
```

Programa igo eta ikusi ez dela ezer aldatu. Kodean begiratu gero txiparekin bitez-bit komunikatzen ari garela ikusiko duzu. (Informazio gehiagorako: <http://ardx.org/SPI>).

### LEDak banaka-banaka kontrolatzen:

LEDak CIRC-02n kontrolatu genituen bezala erabiltzen hasiko gara. 8 LEDen egoerak byte batean gordetzen dira (8 biteko balio bat) ikusi hemen informazio gehiagorako <http://ardx.org/BINA>. Arduinoa oso ondo daki bitak manipulatzeko eta eragiketa multzo bat dauka gure lana errazteko. Biten matematikari buruzko informazioa nahi izanez gero: (<http://ardx.org/BITW>).

### Gure inplementazioa.

loop() funtzioko kodea honekin ordeztu:

```
int itxaronDenb = 100; //Zenbat milisegundu itxoin
                        //LEDen aldaketan artean.
for(int i = 0; i < 8; i++){
    LEDaAldatu(i, ON);
    delay(itxaronDenb);
}
for(int i = 0; i < 8; i++){
    LEDaAladatu(i, OFF);
    delay(itxaronDenb);
}
```

Kodea igotzean argiak banaka-banaka piztu eta itzaltzen joango dira. Kodea eta wikipedia begira dezakezu ikus dezazun nola funtzionatzen duen, eta eposta bat bidalguzu galdera gehiago baldin badituzu.

### Animazio gehiago:

Gauzak interesgarriagoak jarriko dira. CIRC-02ko kodea begiratu gero (ondo pasatzen 8LEDekin) ikusiko duzu LEDen balioak aldatzen ditugula `digitalWrite(led, egoera)`, `LEDaAldatu` funtzioan erabiltzen dugun formatu bera. CIRC-02n erabilitako animazioak berrerabili ditzakezu, kodea sketch honetan kopiatuz eta `digitalWrite()` guztiak `LEDaAldatu()` -gatik aldatuz. Boteretsua? bai, asko gainera. (gauza pare bat gehiago konpondu beharko duzu baina konpilazio erroreak jarraitu eta dena martxan jarriko da.).



(ARDX) .:Arduino Espermentazio Kita: (ARDX)

**Egingo duguna**

Orain arte argia, mugimendua eta elektroiak kontrolatu ditugu. Goazen musika pixkat egitera. Baina soinua fenomeno analogikoa da, nola egingo dugu gure Arduino digitalarekin? Berriz ere bere izugarritzko abiadura erabiliko dugu portaera analogikoa simulatzeko. Hortarako piezoa Arduinoaren pin digital batera konektatuko dugu. Piezoek click antzeko zarata egiten dute korrontea pasatzean. Frekuentzia konkretu bateko pultsua bidalten badiogu (440 aldiz segundu batean, La nota lortzeko) klik horiek elkartuko dira notak sortzeko. Goazen espermentatzera honekin, Arduinoak "Twinkle Little Star" abestia joko du..

**Zirkuitua****Atalak**

CIRC-06  
Breadboard Orria  
x1



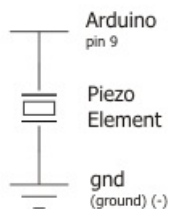
2 Pinetako Burukoak  
x4



Piezo Elementua  
x1



Kablea

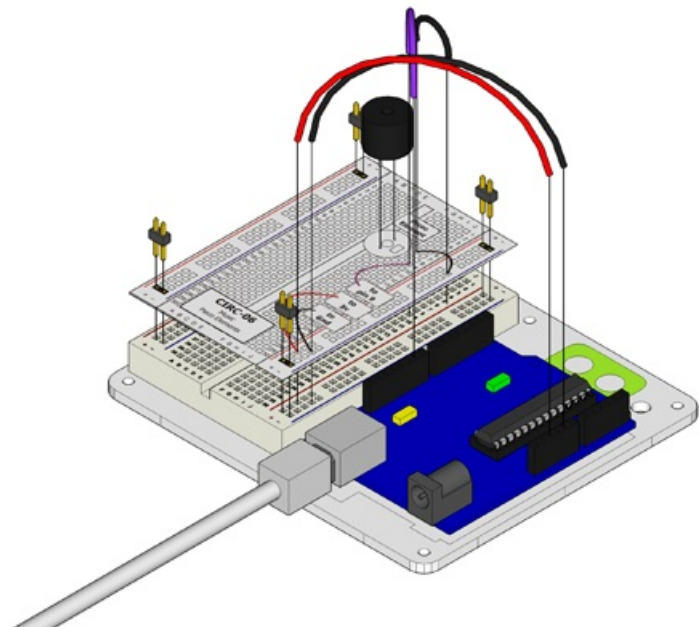
**Eskema****Baliabideak**

.:jaitsi:.

breadboard diseinu-orria  
<http://ardx.org/BBS06>

.:ikusi:.

muntaiaren bideoa  
<http://ardx.org/VIDE06>

**Kodea** (ez duzu zertan dena idatzi behar)Hemendik jaitsi kodea ( <http://ardx.org/CODE06> )

(Kopiatu eta itsatsi kodea Arduinoko Sketch huts batean)

```

/* Musika
 * (copyleft) 2005 D. Cuartielles K3rentzat
 *
 * Adibide honek piezo bat erabiltzen du musika egiteko. Dagokion frekuentziako
 * laukizuzen formako seinale bat bidaltzen dio piezoari, dagokion tonoa sortuz.
 *
 *
 * Tonoak kalkulatzeko ondoko formula matematikoa erabili dugu:
 *
 *
 *      tonoLuz = periodoa / 2 = 1 / (2 * toneMaiztasuna)
 *
 * tono ezberdinak ondoko taulan definituta daude:
 *
 * nota  maiztasuna  periodoa  tonoLuz
 * do      261 Hz      3830    1915
 * re      294 Hz      3400    1700
 * mi      329 Hz      3038    1519
 * fa      349 Hz      2864    1432
 * sol     392 Hz      2550    1275
 * la      440 Hz      2272    1136
 * si      493 Hz      2028    1014

```



```

* do          523 Hz          1912  956
*
* http://www.arduino.cc/en/Tutorial/Melody
*/

int speakerPina = 9;

int luzera = 15; // nota kopurua
char notak[] = "ccggaagffeeddc "; // hutsuneak isilunea adierazten du.
int erritmoa[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
int tempoa = 300;

void TonoaJo(int tonoa, int notaLuzera) {
  for (long i = 0; i < notaLuzera * 1000L; i += tonoa * 2) {
    digitalWrite(speakerPina, HIGH);
    delayMicroseconds(tonoa);
    digitalWrite(speakerPina, LOW);
    delayMicroseconds(tonoa);
  }
}

void NotaJo(char nota, int notaLuzera) {
  char izenak[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
  int tonoak[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

  // notaren izenari dagokion tonoa jotzen du.
  for (int i = 0; i < 8; i++) {
    if (izenak[i] == nota) {
      TonoaJo(tonoak[i], notaLuzera);
    }
  }
}

void setup() {
  pinMode(speakerPina, OUTPUT);
}

void loop() {
  for (int i = 0; i < luzera; i++) {
    if (notak[i] == ' ') {
      delay(erritmoa[i] * tempoa); // itxoin
    } else {
      NotaJo(notak[i], erritmoa[i] * tempoa);
    }

    // noten arteko geldiduna
    delay(tempoa / 2);
  }
}

```

## Ez dabil? (3 gauza probatzeko)

### Ez da ezer entzuten

Piezoaren tamaina eta forma ikusita erraza da breadboard-aren zulo txo egokiak ez topatzea. Errebisa ezazu ondo konektatuta dagoela.

### Musika etengabe jotzen dagoen bitartean ezin duzu pentsatu?

Piezoa deskonektatu, pentsatu, programa igo eta piezoa konektatu berri.

### Twinkle Twinkle Little Star entzuteaz nazkatuta?

Kodea idatzita dagoen moduan oso erraza da zure abesti propioak idaztea, ondoko kodea begiratu nondik hasi ikusteko.

## Hobeto egiteko prest?

### Abiadurarekin jolasten:

Nota bakoitzaren abiadura aldagaien kalkuluetan oinarrituta dago, beraz nota bakoitzaren soinua edo abiadura alda dezakegu. Melodiaren abiadura aldatzeko lerro bakarria aldatu behar da.

```
int tempo = 300; ---> int tempo = (berria #)
```

Melodia astiroago joateko zenbaki handiagoa idatzi, edo txikiagoa azkarrago joateko.

### Notak finduz:

Notak pixkat tonoz kanpo egoteagatik arduratuta bazaude lasai, konpondu daiteke. Notak komentatutako blokean dagoen formula jarraituz lortu dira. Nota konkretu baten balioa aldatzeko tonoa[] arrayko dagokion balioa aldatu ondo entzun arte. (nota bakoitzaren izena izenak[] arrayarekin bat dator (adb. do = 1915 )

```
char izenak[] = { 'do', 're', 'mi', 'fa', 'sol', 'la', 'si', 'Do' };
```

```
int tonoak[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
```

#### **Zure melodioa propioak idatziz:**

Programa 'Twinkle Twinkle Little Star' jotzeko diseinatuta dago, baina programatuta dagoen moduan abestia aldatzea erraza da. Abesti bakoitza zenbaki (int) bat eta bi array-ek osatzen dute. Luzera zenbakiak nota kopurua definitzen du, notak[] array-ak nota bakoitza definitzen du eta erritmoa[] array-ak nota bakoitzaren luzera definitzen du. Adibide Batzuk:

#### **Twinkle Twinkle Little Star**

```
int luzera = 15;
char notak[] = "dodosolsollalasalafamireredo ";
int erritmoa[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,
                  1, 1, 2, 4 };

Zorionak Zuri (lehengo lerroa)
int luzera = 13;
char notak[] = "dodoredofamidodoredosolfa ";
int erritmoa[] = {1,1,1,1,1,2,1,1,1,1,2,4};
```



(ARDX) .:Arduino Esperimentazio Kita:. (ARDX)

**Egingo duguna**

Orain arte irteerikin soilik ibili gara, bada garaia Arduinoak entzun, ikusi eta sentitu dezan. Sakagailua konektatzea oso erraza da. Osagai bat dago, erresistentzia (pull up moduan), itxuraz tokiz kanpo dagoena. Arduinoak erresistentzia honela egotea eskatzen du ez duelako gu bezala nozitzen (hau da, botoia sakatuta edo sakatu gabea). Bere ordeztu boltaia begiratu eta HIGH edo LOW den erabakitzen du. Botoia sakatzen dugunean Arduinoak LOW seinalea jasoko du, baina botoia sakatuta ez dagoenean boltaiaren balioa aldakorra izango da (batzutan erroreak sortuz). Ziur egon nahi baldin badugu Arduinoak HIGH irakurriko duela botoia sakatuta ez dagoenean, erresistentzia gehitzen dugu. (oharra: lehenengo adibide programak bi botoietako bat bakarrik erabiltzen du)

**Zirkuitua****Osagaiak**

CIRC-07  
Breadboard Orria  
x1



10k Ohm-erresistentzia  
Marroia-Beltza-Laranja  
x2



2 Pinetako Burukoak  
x4



560 Ohm-erresistentzia  
Berdea-Urdina-Marroia  
x1



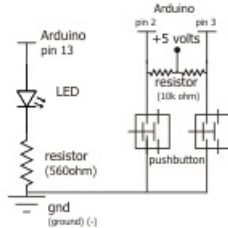
Sakagailuak  
x2



Kablea



LED Gorria  
x1

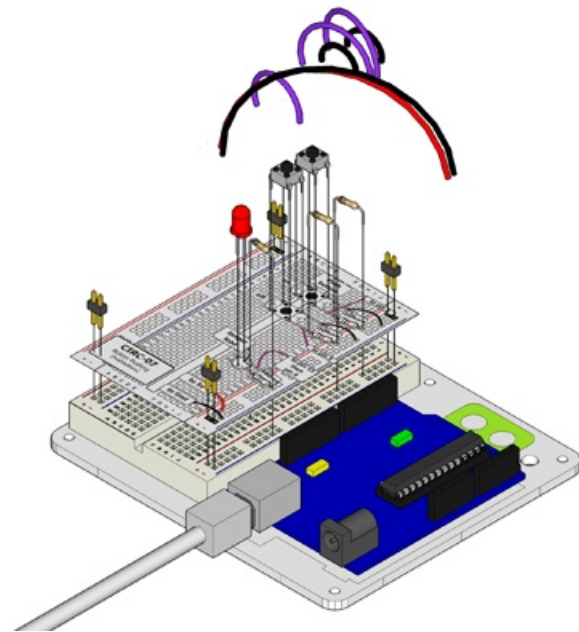
**Eskema****Baliabideak**

..Jaitsi:.

breadboard diseinu-orria  
<http://ardx.org/BBLS07>

..Ikusi:.

muntaiaren bideoa  
<http://ardx.org/VIDE07>

**Kodea** (ez duzu zertan dena idatzi behar)**Fitxategia > Adibideak > 2.Digital > Botoia**

(arduino.cc webgune ikusgarritik hartua, ideia onen bila bazabiltza begiratu hor)

/\*

Botoia

Piztu eta itzali 13. pin digitalean konektatuta dagoen argia ematen duen diodoa (LED) 7. pin digitalean konektatuta dagoen sakagailua sakatzean

Zirkuitua:

- \* LEDa, 13. pin digitaletik lurrera konektatuta
- \* sakagailua, 2. pin digitalera eta +5V-etara konektatuta
- \* 10K erresistentzia, lurrera eta 2. pin digitalera konektatuta

\* Oharra: Arduino gehienetan LED bat dago 13. pinera konektatuta.

```

http://www.arduino.cc/en/Tutorial/Button
*/

// Konstanteak ez dira aldatzen. Pin zenbakiak esleitzeko
// erabiltzen dira hemen
const int botoiPina = 2;    // sakagailuaren pin zenbakia
const int ledPina = 13;    // LEDaren pin zenbakia

// aldagaiak aldatuko dira:
int botoiEgoera = 0;    // sakagailuaren egoera irakurtzeko aldagaia

void setup() {
  // LEDaren pina irteera bezala hasieratzen dugu:
  pinMode(ledPina, OUTPUT);
  // sakagailua sarrera bezala hasieratzen dugu:
  pinMode(botoiPina, INPUT);
}

void loop(){
  // sakagailuaren egoera irakurri:
  botoiEgoera = digitalRead(botoiPina);

  // ikusi ea sakagailua sakatuta dagoen.
  // hala bada, botoiEgoeraren balioa HIGH izango da:
  if (botoiEgoera == HIGH) {
    // LEDa piztu on:
    digitalWrite(ledPina, HIGH);
  }
  else {
    // LEDa itzali:
    digitalWrite(ledPina, LOW);
  }
}

```

### Ez dabil? (3 gauza probatzeko)

#### Argia ez da pizten

Sakagailua lauki bat da eta horregatik erraza da gaizki jartzea. 90 gradutako buelta bat eman eta ikusi ea horrela funtzionatzen duen.

#### Argia ez da itzaltzen pixkanaka-pixkanaka

Etengabe egiten dugun errorea, LEDa pixkanaka-pixkanaka itzal dadin 9. pinean konektatu behar dugu, ez 13.ean.

#### Sinpeegia?

Ez arduratu. Zirkuitu hau oso sinplea da osagaiekin erraz jolastu ahal izateko, baina behin elkartuta zerua da muga bakarra.

### Hobeto egiteko prest?

#### Piztu botoia, itzali botoia:

Hasierako adibidea nahiko sinplea da (hortarako ez dut Arduino bat behar), goazen pixkat konplikatuzera. Botoi batek LEDa piztuko du eta besteak itzali. Kodean aldaketa hauek egin behar dira:

```

int ledPina = 13; // LEDarentzako pina
int botoiPin1 = 3; // sakagailu 1
int botoiPin2 = 2; // 2.a

void setup() {
  pinMode(ledPina, OUTPUT); // LEDa irteera bezala hasieratu
  pinMode(botoiPin1, INPUT); // sakagailua sarrera bezala hasieratu
  pinMode(botoiPin2, INPUT); // 2.a ere bai
}

void loop(){
  if (digitalRead(botoiPin1) == LOW) {
    digitalWrite(ledPina, LOW); // LEDa itzali
  } else if (digitalRead(botoiPin2) == LOW) {
    digitalWrite(ledPina, HIGH); // LEDa itzali
  }
}

```

Programa zure plakara igo, eta hasi LEDa pizten eta itzaltzen.

#### LEDa piztu eta itzali pixkanaka-pixkanaka:

Botoiak seinale analogiko bat kontrolatzeko erabiliko ditugu. LEDa 13. pinetik 9.era mugituko dugu,

eta kodean ondoko aldaketak egingo ditugu:

```
int ledPina = 13; ----> int ledPina = 9;
```

Ondoren loop() funtzioan kodea aldatu irakur dezan.

```
int balioa = 0;
void loop(){
  if (digitalRead(botoiPin1) == LOW) { balioa--; }
  else if (digitalRead(botoiPin2) == LOW) { balioa++; }
  balioa = constrain(balioa, 0, 255);
  analogWrite(ledPina, balioa);
  delay(10);
}
```

**Itzaltze abiadura aldatzen:**

LEDa azkarrago edo astiroago itzaltzea nahi baduzu, kode lerro bakarra aldatu behar da;

```
delay(10); ----> delay(new #);
```

Azkarrago itzaltzeko zenbakia txikitu, astiroago joateko zenbakia handitu.




(ARDX) .:Arduino Esperimentazio Kita:. (ARDX)


**Egingo duguna**


Pin digitaletaz aparte, Arduinoak sarrera analogiko bezala erabil daitezkeen 6 pin dauzka. Sarrera hauek boltai bat (0-5 boltio tartekoa) hartu eta zenbaki digital batera bihurtzen du, Otik (0 volt) 1023ra (5 volt), hau da, 10 bitetako bereizmena. Sarrera hauei etekina ateratzeko potentziometroak (erresistentzia aldagarri izenaren ezagutuak baita) oso erabilgarriak izan daitezke. 5 boltioetara konektatzen badugu, erdiko pinak 0-5 boltio arteko balio aldagarri bat emango digu biratuta dagoen angeluaren arabera (erdian 2,5 volt, adibidez). Itzulitako balioa aldagai bezala erabil ditzakegu gure programetan.


**Zirkuitua**


**Osagaiak**


- 

**CIRC-08  
Breadboard  
Orria  
x1**
- 

**2 Pinetako Burukoak  
x4**
- 

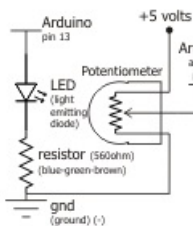
**Potentziometroa  
10k ohm  
x1**
- 

**Kablea  
x1**
- 

**LED Berdea  
x1**
- 

**560 Ohmioetako  
erresistentzia  
Berdea-Urdina-Marroia  
x1**

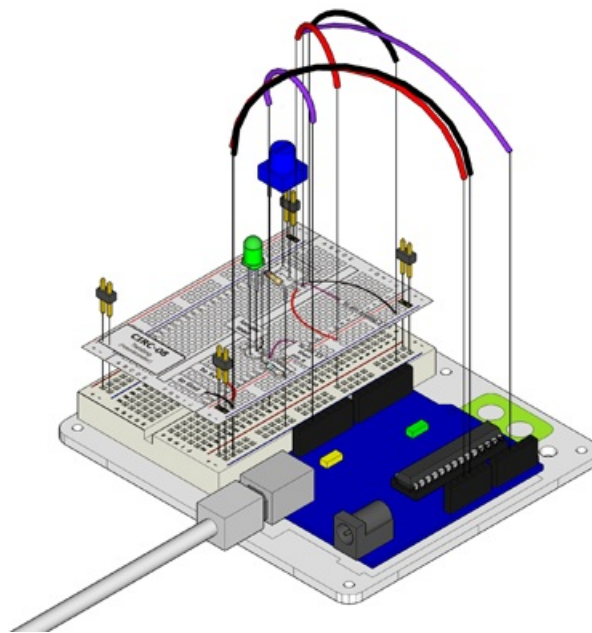
**Eskema**



**Baliabideak**

.:download:.  
breadboard-aren diseinu-orria  
<http://ardx.org/BBLS08>

.:ikus:.  
muntaiaren bideoa  
<http://ardx.org/VIDE08>



**Kodea** (ez duzu zertan dena idatzi behar)

**Fitxategia > Adibideak > 3.Analogikoa > AnalogInput**

(**arduino.cc** webgunetik hartuta, **beste ideia onen bila bazabiltza hemen begiratu**)

/\*

Sarrera Analogikoa

Sarrera analogikoaren demostrazioa: 0. pinean konektatuta dagoen sentsoretik irakurriko dugu eta 13. pinean dagoen LEDa piztu / itzaliko dugu. LEDa piztuta egongo den denbora kopurua analogRead() funtziotik irakurriko dugun balioaren berdina izango da

Zirkuitua:

- \* 0. sarrera analogikora konektatuta dagoen potentziometroa
- \* Potentziometroaren erdiko hankatxo pin analogikora
- \* Pin bat lurrera (berdin dio zein)
- \* Bestea +5V-tara
- \* LEDaren anodoa (hanka luzea) 13. irteera digitalera
- \* LEDaren katodoa (hanka motza) lurrera

\* Oharra:Arduino gehienek LED bat integratuta dakartzate 13. pinean, beraz LEDa hautazkoa da.

Aldatua: 2009ko Ekainak 16

<http://arduino.cc/en/Tutorial/AnalogInput>

```
*/

int sentsorePina = 0; // potentziometroarentzako sarrera pina hautatu
int ledPina = 13; // LEDaren pina hautatu
int sentsoreBalioa = 0; // sentsoretik datorren balioa gordetzeko aldagaia

void setup() {
  // ledPina irteera bezala adierazi:
  pinMode(ledPina, OUTPUT);
}

void loop() {
  // sentsoretik balioa irakurri:
  sentsoreBalioa = analogRead(sentsorePina);
  // ledPina piztu
  digitalWrite(ledPina, HIGH);
  // programa milisegundu gelditu:
  delay(sentsoreBalioa);
  // ledPina itzali
  digitalWrite(ledPina, LOW);
  // programa milisegundu gelditu:
  delay(sentsoreBalioa);
}
```

### Ez dabil? (3 gauza probatzeko)

#### Batzutan bakarrik dabil

Normalean potentziometroaren hankatxoengatik izaten da, erraz askatzen dira eta. Normalean potentziometroa barrura bultzatuta konpontzen da.

#### Ez dabil

Konproba ezazu potentziometroa ez dagoela pin digital batera konektatuta (elikadura pinen azpikoak)

#### Still Backward

Desmuntatu eta berriro muntatu. Batzutan laguntzen du.

### Hobeto egiteko prest?

#### Atalasea aldatzen:

Batzutan balio batek atalase zehatz bat gainditzean irteera bat aldatzea nahi dugu. Potentziometroarekin egiteko `loop()` funtzioko kode hau aldatu:

```
void loop() {
  int atalasea = 512;
  if(analogRead(potPina) > atalasea){ digitalWrite(ledPina, HIGH);}
  else{ digitalWrite(ledPina, LOW);}
}
```

Kode honek LEDa piztuko du balioa 512 baino handiagoa denean (erdia gutxi gora behera), atalasearen balioa aldatuta bereizmena alda daiteke.

#### Itzaltzea:

LEDaren argitasuna potentziometroarekin kontrolatuko dugu. Egin beharreko lehenengo gauza LEDaren pina aldatzea izango da. 13. pinetik 9.-era aldatu eta kodean lerro bat aldatu:

```
int ledPina = 13; ----> int ledPina = 9;
```

Loop funtzioan aldaketa hauek egin:

```
void loop() {
  int balioa = analogRead(potPina) / 4;
  analogWrite(ledPina, balioa);
}
```

Kodea igo eta ikusi nola zure LEDa itzaltzen doan potentziometroaren arabera. (Oharra: balioa zati lau egitearen arrazoia zera da: `analogRead()` funtzioak 0-1023 bitarteko balioa emango digula (10 bit), `analogWrite()` funtzioak aldiz 0-255 tartekoak erabiltzen dituela (8 bit) )

#### Servo bat kontrolatzen:

Adibide polita da hau, bi zirkuitu elkartzea ahalbidetuko diguna. CIRC-04 adibidean egin genuen moduan, servoa konektatu eta Knob programa ireki (Fitxategia > Adibideak > Servo-Liburutegia >

Knob ), eta kode lerro bat aldatu:

```
int potpin = 0; ----> int potpin = 2;
```

Kodea zure Arduinora igo eta ikusi nola mugitzen den servoa potentziometroa giratzen duzun heinean.





(ARDX) .:Arduino Espermentazio Kita: (ARDX)

**Egingo duguna**

Sarrerak potentziometro batetaik hartzea erabilgarria da gizakiok kontrolatzen ditugun esperimentuetarako, baina nola egin inguruneak kontrolatzen duen esperimentua egin nahi badugu? Oinarri berdinak erabiltzen ditugu baina potentziometro (tortsoan oinarritutako erresistentzia) baten ordez fotoerresistore (argian oinarritutako erresistentzia) bat erabiliko dugu. Arduinoak ez du zuzenean erresistentziaren aldaketak nozitzen, boltaia baizik, horregatik boltaia banatzaile bat (<http://ardx.org/VODI>) erabiliko dugu. Sentsorearen pinean dagoen boltaia zehatza neurtu daiteke, baina gure helburuetarako (argiaren aldaketa erlatiboak neurtu) balioekin esperimentatu dezakegu eta ikusi nola funtzionatzen duen. Argi asko dagoenean balio baxuak irakurriko ditugu, eta ilunpetan balio altuak.

**Zirkuitua**

**Atalak**



**CIRC-09 Breadboard Orria**  
x1  
**10k Ohm-etako erresistentzia Marroia-Beltza-Laranja**  
x1



**2 Pinetako burukoak**  
x4  
**560 Ohm-etako erresistentzia Berdea-Urdina-Marroia**  
x1



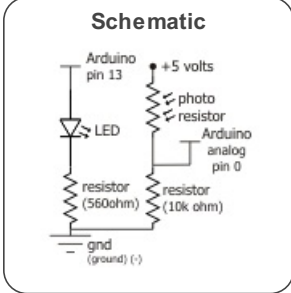
**Fotoerresistorea**  
x1



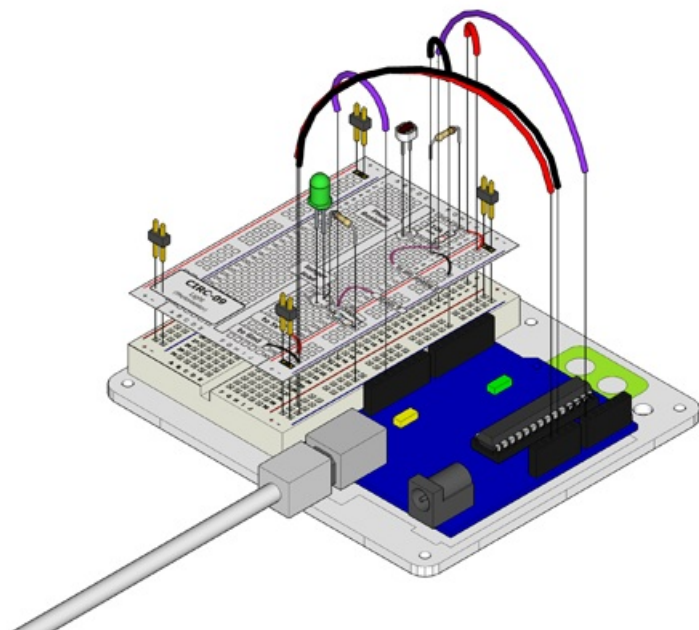
**Kablea**



**LED Berdea**  
x1



**Baliabideak**  
.:jaitsi:.  
breadboard-aren diseinu-orria  
<http://ardx.org/BBLS09>  
.:ikusi: muntaiaren bideoa  
<http://ardx.org/VIDE09>



**Kodea** (ez dago zertan dena idatzi behar)

**Kodea hemendik jaitsi daiteke:** <http://ardx.org/CODE09>

**(kopiatu eta pegatu Arduinoko Sketch batean)**

```

/*
 * Fotoerresistorean eragindako argitasunaren arabera LEDaren intentsitatea
 * aldatuko duen programa sinplea
 *
 */

//Fotoerresistorea Pin
int argiPin = 0; //Fotoerresistorea konektatuta dagoen pin analogikoa
//connected to
//Fotoerresistorea ez dugu kalibratuko beraz
//balio hau sentsorearen irakurketa gordina da (argi erlatiboak)

//LED Pina
int ledPina = 9; //LEDa konektatuko dugun pina
//LEDaren intentsitatea kontrolatzen ari gara
//beraz PWMdun pin bat erabiliko dugu (pultsuen zabalera
// bidezko modulazioko pinak)

void setup()

```

```

{
  pinMode(ledPina, OUTPUT); //LEDaren pina irteera bezala definituko dugu
}
/*
 * loop() - Funtzio hau setup bukatu ondoren hasiko da
 * eta etengabe errepikatzen da
 */
void loop()
{
  int argiMaila = analogRead(argiPin); //Argi maila
                                     // irakurri
  argiMaila = map(argiMaila, 0, 900, 0, 255);
               //Balioa moldatu 0tik 900era izan ordez
               //0tik 255era bihurtu

  argiMaila = constrain(argiMaila, 0, 255); //Balioa 0-255 tartean
                                             //dagoela ziurtatu
  analogWrite(ledPina, argiMaila); //Balioa idatzi
}

```

### Ez dabil? (3 gauza probatzeko)

#### LEDA beltza geratzen da

Aldiro-aldiri errepikatzen dugun errorea, bi modutan funtzionatuko lukeen LEDak asmatuko balituzkete zenbat denbora aurreztuko genuke. Buelta eman eta saiatu berriz.

#### LEDak ez dio argiaren aldaketei erantzuten

Fotoerresistorearen hankatxoaren tamaina estandarra ez denez, erraza da gaizki jartzea. Konprobatu ondo dagoela.

#### Oraindik ez dabil?

Agian argi gehiegi edo gutxiegi duen gela batean zaude. Argiak piztu eta itzali ea horrek laguntzen duen, edo linterna bat baduzu, saia zaitetz berarekin.

### Hobeto egiteko prest?

#### Erantzuna alderantzizkatu:

Agian kontrako erantzuna nahi duzu. Ez arduratu, oso erraza da eta, lerro hau aldatu:

```

analogWrite(ledPina, argiLevel);
----> analogWrite(ledPin, 255 - argiLevel);

```

Kodea igo eta ikusi nola aldatzen den erantzuna.

#### Gaueko argia:

LEDaren intentsitatea kontrolatu ordez, LEDa piztu edo itzaliko dugu ingurunean dagoen argiaren arabera. `loop()` funtzioko kodea aldatu.

```

void loop(){
  int argiMailaMinimoa = 300;
  if(analogRead(argiPin) > argiMailaMinimoa){
    digitalWrite(ledPina, HIGH);
  }else{
    digitalWrite(ledPina, LOW);
  }
}

```

#### Argiaren bidez kontrolatzen den servoa:

Goazen erabiltzera argia noitzeko ikasi berri duguna servo bat kontrolatzeko. (eta bide batez Arduinoko kodea moldatzen praktikatuko dugu). Servo bat pin 9ra konektatu (CIRC-04n egin genuen bezala). Ondoren Knob adibide programa ireki (CIRC-08n egin genuen bezala) Fitxategia > Adibideak > Servo-Liburutegia > Knob. Kodea zure plakara igo eta ezer aldatu gabe funtzionatzen duela ikus ezazu.

#### Servoaren barruti osoa erabili:

Ikusiko zenuen bezala, servoak ez du bere barruti oso erabiltzen ari. 0. pinean konektatuta dagoen boltai banatzailea honen erruduna da, 0-5 volt tarteko balioak erabili ordez tarte txikiagoan mugitzen ari da (balio hauek setup-aren arabera alda daitezke). Hau konpontzeko balio = `map(balioa, 0, 1023, 0, 179)`; lerroarekin jolas ezazu. Nondik jarraitu argi ez baduzu, bisitatu webgune hau: <http://arduino.cc/en/Reference/Map> .



### Egingo duguna

Ze magnitute neurtuko dugu orain gure Arduinoarekin? Tenperatura. Hortarako pixkat konplikatu den ZI (zirkuitu integratua) erabiliko dugu, gure P2N2222AG transistorearen antzekoa. Hiru pin dauzka, lurra, +5voltiotako elikadura eta seinalea. Irteeran 10 milivoltio emango ditu zelsius gradu bakoitzeko. (0 azpiko tenperaturak neurtzeko 500mVetako desplazamendua du: 25 gradu C = 750 mV, 0 gradu C = 500mV). Balio digitaletatik graduetara bihurtzeko Arduinoko matematika tresnak erabiliko ditugu. Balioak erakusteko IDEaren ezaugarri bat erabiliko dugu, debug leihoa. Balioa pantailan erakutsiko dugu serie bidezko konexio baten bidez. Goazen lanera.

Azken oharra, zirkuitu honek IDEko serie monitorea erabiltzen du. Irekitzeko, programa igo ondoren antena eta lauki itxura duen botoian klik egin.

TMP36 datu-orria:

<http://ardx.org/TMP36>

### Zirkuitua

#### Osagaiak



CIRC-10  
Breadboard  
Orria  
x1



2 pinetako  
burukoak  
x4

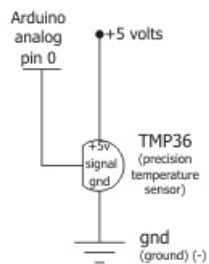


TMP36  
Tenperatura  
sentsorea  
x1



Kablea

#### Eskema



#### Baliabideak

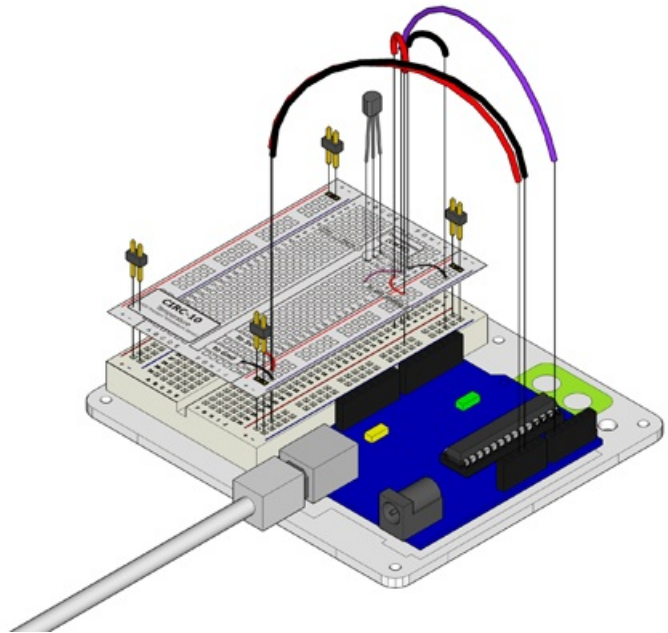
.:download:.

breadboard-aren diseinu-orria  
<http://ardx.org/BBLS10>

.:ikusi:.

muntaiaren bideoa

<http://ardx.org/VIDE10>



**Kodea** (ez duzu zertan dena idatzi behar)

**Kodea hemendik jaitsi: ( <http://ardx.org/CODE10> )**

**(Arduinoko sketch huts batean kopiatu eta itsatsi)**

```

/* -----
 * | Arduino Esperimentazio Kita, Adibideko Kodea |
 * | CIRC-10 .: Tenperatura .: (TMP36 Tenperatura Sentsorea) |
 * -----
 *
 * Oraingo tenperatura IDEaren debug leihoan erakusten duen programa sinplea
 *
 * Zirkuitu honi buruzko informazio gehiagorako: http://tinyurl.com/c89tvd
 */

//TMP36aren Pin Aldagaiak
int tenperaturaPina = 0; //TMP36aren Vout pina konektatuko dugun pin analogikoa
//Bereizmena 10 mV / Celsius gradu

```

```

// (500 mV-
etako desplazamendua) temperatura negatiboak eduki ahal izateko

/*
 * setup() funtzioa behin exekutatu da, sketcha hastean
 * Serie konexioa hasieratzen dugu, ordenagailuarekin
 */
void setup()
{
  Serial.begin(9600); //Serie konexioa ordenagailuarekin hasieratur
                    //emaitza ikusteko seriethe monitorea ireki
                    //Fitxategia barraren azpiko botoia (kutxa baten itxura dauka, antena batekin)
}

loop() funtzioa etengabe exekutatu da,
{
  float temperatura = getVoltage(temperaturaPina); //Boltoa irakurtzen dugu temperatura sentsoretik.
  temperatura = (temperatura - .5) * 100; //Bihurketa: 10mV gradu bakoitzeko
500mVtioko desplazamenduarekin 10mV gradu bakoitzeko 500mVtioko desplazamenduarekin
                    //Graduetara: ((boltoa - 500mV) times 100)
  Serial.println(temperatura); //emaitza pantailaratu
  delay(1000); //segundu bat itxaron
}

/*
 * BoltaiaIrak(pina) - pin aldagaiak definitzen duen pin analogikoak duen boltaia itzultzen da.
 * pin
 */
float BoltaiaIrak(int pina){
  return (analogRead(pina) * .004882814); //0-1023 tarteko balio digitaletara bihurtzen
                    // 0tik 5 voltioetara (irakurritako balio bakoitza 5
milivoltioen balioak da balio bakoitza ~ 5 milivoltioen5 balioak da
                    //olts
}

```

### Ez dabil? (3 kontu probatzeko)

#### Ez da ezer gertatzen

Programa honek ez du inolako seinalerik ematen, martxan dagoela ikusteko Arduino IDEko serie monitorea ireki beharko duzu.

#### Zaborra erakusten du

Hau gertatzen da serie monitorean abiadura eta programarena berdina ez direnean. Konpontzeko, serie monitorea azpikaldean dagoen botoia sakatu eta "9600 baud" ezarri.

#### Temperaturaren balioa ez da aldatzen

Sentsorea eskuetan eduki edo izotz pixkat erabili hozteko.

### Hobeto egiteko prest?

#### Boltoa erakutsi:

Lerro bat aldatzearekin nahiko da. Gure sentsoreak 10mVko irteera du gradu bakoitzeko, beraz boltoa erakusteko `BoltaiaJaso()` funtzioaren emaitza erakutsi besterik ez dugu egin behar.

```
temperature = (temperature - .5) * 100; lerroa borratu
```

#### Fahrenheit graduak erakutsi:

Oso erraza berriz, matematiketan bakarrik egin behar dira aldaketak. Celsius graduetatik Fahrenheitera pasatzeko formula hau erabiltzen dugu.

$$( F = C * 1.8 ) + 32 )$$

lerro hau gehitu

```
temperatura = (((temperatura - .5) * 100)*1.8) + 32;
Serial.println(temperatura); baino lehen
```

#### Irteera esanguratsuagoa:

Irteerari mezu bat gehituko diogu serie monitorean agertzen dena esanguratsuagoa izan dadin. Hau lortzeko hasierako kodea hartu eta

```
Serial.println(temperature); ----> ordez Serial.print(temperature); eta
Serial.println(" degrees centigrade"); idatzi.
```

Lehenengo lerroko aldaketak hurrengo irteera lerro berean agertuko dela esan nahi du, ondoren testu esanguratsua eta lerro berria idatziko ditugu.

#### Serie abiadura aldatzen:

Noizbait informazio asko atera nahiko dugu serieko portutik, orduan abiadura gakoa izango da. Orain

9600baudetara transmititzen ari gara, baina abiadura handiagoak posible dira, lerro hau aldatuz:

```
Serial.begin(9600); ----> Serial.begin(115200);
```

Sketch-a igo, serie monitoreo aktibatu, abiadura 9600etik 115200era aldatu. Orain 12 aldiz azkarrago transmititzen ari zara.



(ARDX) .:Arduino Esperimentazio Kita:. (ARDX)

**Egingo duguna**

Gure hurrengo zirkuitua test bat da. CIRC03an ikasi duguna erabiliko dugu rele bat kontrolatzeko. Rele bat elektrikoki kontrolatzen den interruptore mekaniko bat da. Plastikozko kutxa txikiaren barruan elektromagnetoa dago, energia jasotzean interruptorea mugitzen duena (oso polita den zarata bat ateraz). Tamaina ezberdinetako releak eros ditzakezu, kit honetan erabiltzen denaren laurdenetik hasita hozkailu baten tamainakoetara, bakoitzak korrante kantitate bat kontrolatzeko gai dena. Oso erabilgarriak dira elementu fisikoak direlako. Orain arte erabili dugun silizio guztia jostagarria bada ere, batzutan polita da ehundaka interruptore aktibatzea zerbait handia egiteko. Releak eta Arduinoekin ibilitezak amesteko aukera ematen dute. Gaurko teknologiarekin iraganeko teknologia kontrolatzen. (1N4001 diodoa kontrako diodo moduan erabiltzen da, informazio gehiagorako webgune honetara jo: <http://ardx.org/4001> )

**The Circuit****The Parts**

CIRC-11  
Breadboard orria  
x1



Diodoa  
(1N4001)  
x1



Transistorea:  
P2N2222AG  
(TO92)  
x1



Relea:  
(SPDT)  
x1



10k Ohm  
erresistentzia:  
Marroia-Beltza-  
Laranja  
x1



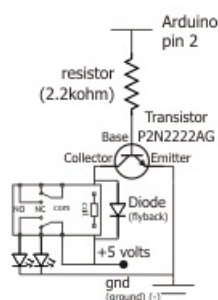
560 Ohm  
Erresistentzia:  
Berdea-Urdina-  
Marroia  
x2



LED berdea:  
x1

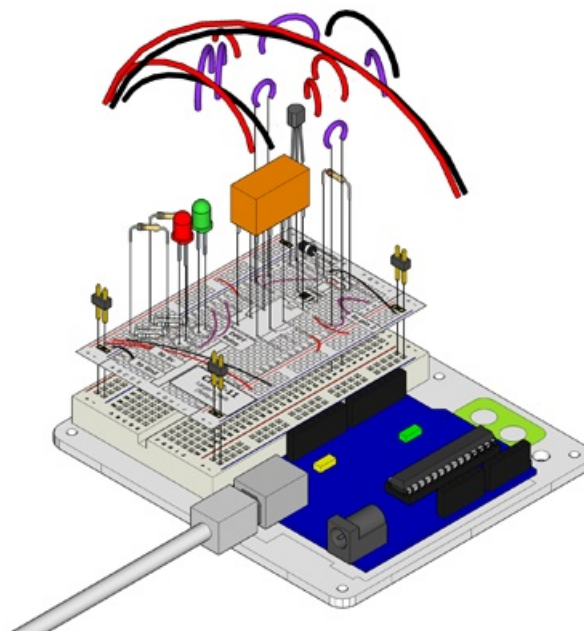


LED  
gorria:  
x1

**Schematic****center>Resources**

.:jaitsi:.  
breadboard-aren diseinu-orria  
<http://ardx.org/BBS11>

.:ikusi:.  
muntaiaren bideoa  
<http://ardx.org/VIDE11>

**Code** (ez duzu zertan dena idatzi behar)**Fitxategia > Adibideak > 1.Oinarrizkoa > keinu egin**

(arduino.cc webgune ikusgarritik hartua, beste ideia onen bila bazabiltza hemen begiratu)

```

/*
  Keinu egin
  Etengabe LED bat pizten du segundu batez, ondoren itzaltzen du segundu batez.
  The zirkuitua:
  * LEDa 13. pinetik lurrera konektatuta.
  * Abisua: Arduino plaka gehienetan iadanik LED bat dago plakan
  13. pinera konektatuta, beraz ez duzu zertan gailu gehiagorik instalatu behar adibide honetarako.

  Sortze-data: 2005eko Ekainak 1
  Egilea: David Cuartielles
  http://arduino.cc/en/Tutorial/Blink
  H. Barraganen kodean oinarrituta, Wiring s/i plakarentzat
  */

```

```

int ledPin = 2;    // Relea 2. pin digitalera

// setup() funtzioa behin exekutatzen da, sketcha hastean

void setup()  {
  // Pin digitala irteera bezala hasieratu:
  pinMode(ledPin, OUTPUT);
}

// loop() funtzioa etengabe exekutatzen da,
// Arduinoa elikatua dagoen heinean.

void loop()
{
  digitalWrite(ledPin, HIGH);  // LEDa piztu
  delay(1000);                 // segundu bat itxoin
  digitalWrite(ledPin, LOW);   // LEDa itzali
  delay(1000);                 // segundu bat itxoin
}

```

### Ez dabil? (3 kontu probatzeko)

#### Ez da ezer gertatzen

Adibideko kodeak 13. pina erabiltzen du eta relea 2.enera konektatuta dago. Konproba ezazu kodean aldaketa egin duzula.

#### Ez dago Klik zaratarik

Transistorea edo harila ez dabilzate ongi. Konproba ezazu transistorea ondo dagoela.

#### Ez dabil ondo

Erabiltzen diren releak ez daude pentsatuak **\*\*breadboard\*\*** batean erabiltzeko, soldatzeko baizik. Agian indar asko egin beharko duzu zirkuitua funtzionatzeko (eta batzutan kanporaka aterako da).

### Hobeto egiteko prest?

#### Back-EMF Pultsua ikusi

Diodoaren orde LED bat jarri. Keinu egiten ikusiko duzu harilak itzaltzean askatzen duen boltaiarengatik.

#### Motore bat kontrolatzen

CIRC-03 ikasgaien transistore bat erabiliz motore bat kontrolatzen ikasi genuen. Baina motore indartsuagoak kontrolatzeko hobe da releak erabiltzea. Horretarako, kendu LED gorria eta motorea konektatu bere orde (560 Omhetako erresistentziarik gabe)

#### Motorearen norabidea kontrolatzen

Bukatzeko, hobekuntza pixkat konplikatua. DC motore baten norabidea kontrolatzeko korrontearen norantza aldatzeko gai izan behar gara. Eskuz egingo bagenu, kableak elkartrukatuko genituzke. Elektrikoki egiteko h-zubi deituriko muntaia egin behar dugu DPDT rele bat erabiliz motorearen norantza kontrola daiteke, ondoko zirkuitua muntatuta. Pixkat konplikatua dirudi baina kable gutxi batzuk gehituta egin daiteke. Saia zaitetz!



"><http://www.ardx.org/src/circ/ARDX-EX-11-01.png>>".