ARDUINO

Published : 2011-08-31

Table of Contents

JOHDANTO

Johdanto

Arduino on työkalu sulautettujen järjestelmien suunnitteluun ja rakenteluun. Se koostuu avoimiin suunnitelmiin perustuvasta mikrokontrollerista, johon liittyy sisään- ja ulostulon tuki ja standardoitu ohjelmointikieli.

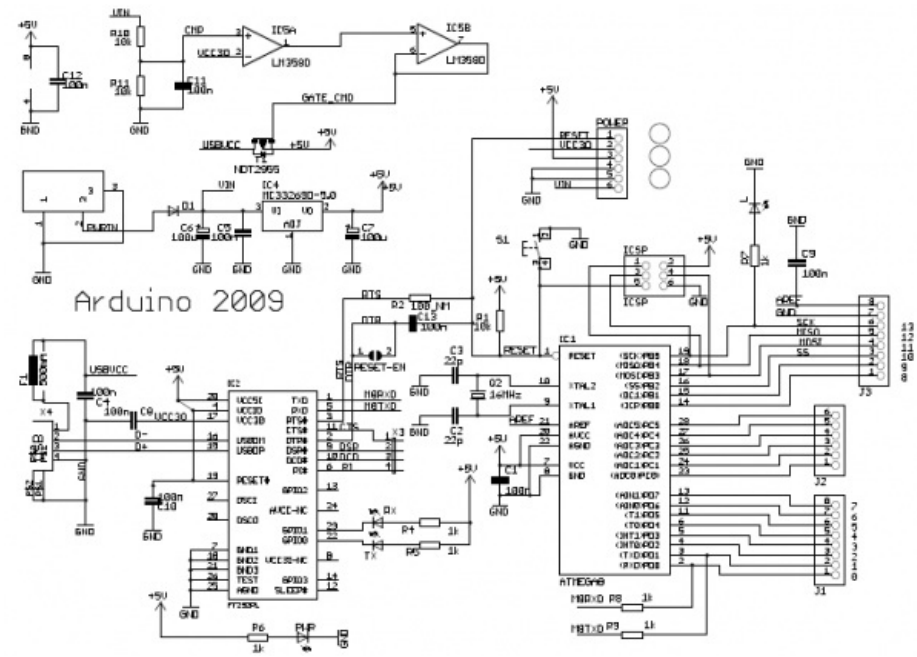
Arduinoa ohjelmoidaan Wiring-ohjelmointikielellä, joka on käytännössä hieman muunneltua C++-ohjelmointikieltä.

Avoimen lähdekoodin mikrokontrolleri

Kuinka laite voi perustua avoimeen lähdekoodiin?

Arduinon piirrustukset ovat saatavilla internetistä, voit käyttää niitä samaan tapaan kuin avoimen lähdekoodin ohjelmia. Voit tutkia niitä, muuttaa niitä tai rakentaa Arduinon niiden avulla.

Voit ladata eri Arduino-mallien piirrokset Arduinon sivuilta ja muokata niitä tahtoessasi.



Mahdollisesti helpompaa on kuitenkin tilata Arduino verkkokaupasta. Tällainen on Suomessa esimerkiksi Paeae.com.

OHJELMOINTIYMPÄRISTÖ

Asennus Ubuntuun

Somebody Should Set The Title For This Chapter!

Arduinon kehitysympäristö

Arduinon kehitysympäristö koostuu tekstieditorista, jolla kirjoitetaan koodia, viestialueesta, tekstipäätteestä, yleisimmät toiminnot sijaitsevasta työkalupalkista, ja sarjasta valikkoja. Se ottaa yhteyden Arduino-mikrokontrolleriin ladatakseen ohjelmia ja viestiäkseen niiden kanssa.

Arduinolla kirjoitettuja ohjelmia kutsutaan luonnoksiksi ("sketch"). Nämä luonnokset kirjoitetaan tekstiedistorissa. Siinä on ominaisuudet tekstin leikkaamiseen ja liittämiseen, sekä tekstin etsimiseen/korvaamiseen. Konsoli näyttää tekstiulostulon Arduinon ympäristöstä, siihen sisältyy täyelliset virheviestit ja muuta tietoa. Työkalupalkin nappulat antavat sinun tarkistaa ja ladata ohjelmia, luoda, avata ja tallentaa luonnoksia, sekä avata sarjamonitorin:


Varmista/Käännä
Tarkastaa koodisi virheet.


Pysäytä
Pysäyttää sarjamonitorin, tai ottaa muut napit pois käytöstä.


Uusi
Luo uuden luonnoksen.


Avaa
Näyttää valikon kaikista luonnoskirjassasi olevista luonnoksista. Sen napsauttaminen avaa sen nykyisessä ikkunassa.


Tallenna
Tallentaa luonnoksesi.


Lataa Arduinon sisään/ulostuloon
Kääntää koodisi ja lataa sen Arduinon sisään/ulostuloon.


Sarjamonitori
Avaa sarjamonitorin.

Lisäkomennot löytyvät viidestä valikosta: File (tiedosto), Edit (muokkaa), Sketch (luonnos), Tools (työkalut) ja Help (apua). Nämä valikot ovat tilannesidonnaisia, joten ainostaan juuri nyt tehtävään työhön liittyvät vaihtoehdot ovat tarjolla.

Edit

- *Copy for Discourse*
  Copies the code of your sketch to the clipboard in a forum suitable for posting to the forum, complete with syntax coloring.
- *Copy as HTML*
  Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

Sketch

- *Verify/Compile*
  Checks your sketch for errors.
- *Import Library*
  Adds a library to your sketch by inserting #include statements at the code of your code. For more details, see libraries below.
- *Show Sketch Folder*
  Opens the sketch folder on the desktop.
- *Add File...*
  Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu.

Tools

- *Auto Format*
  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements instead curly braces are indented more.
- *Board*
  Select the board that you're using. See below for descriptions of the various boards.
- *Serial Port*
  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- *Burn Bootloader*
  The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega (which normally come without a bootloader). Ensure that you've selected the correct board from the **Boards** menu before burning the bootloader. When using an AVR ISP, you'll need to select the item corresponding to your programmer from the **Serial Port** menu.

Sketchbook

The Arduino environment includes the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the **File > Sketchbook** menu or from the **Open** button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the **Preferences** dialog.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the **Tools > Board** and **Tools > Serial Port** menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbserial-1B1** (for a USB board), or **/dev/tty.USA19QW1b1P1.1** (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably **COM1** or **COM2** (for a serial board) or **COM4**, **COM5**, **COM7**, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be **/dev/ttyUSB0**, **/dev/ttyUSB1** or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload to I/O Board** item from the **File** menu. Current Arduino boards will reset automatically and begin the upload. With older boards that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino environment will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch > Import Library** menu. This will insert one or more **#include** statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its **#include** statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources. To install these third-party libraries, create a directory called **libraries** within your sketchbook directory. Then unzip the library there. For example, to install the DateTime library, its files should be in the **/libraries/DateTime** sub-folder of your sketchbook folder.

Third-Party Hardware

Support for third-party hardware can be added to the **hardware** directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the **hardware** directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the platforms page on the Google Code developers site.

Serial Monitor

Displays serial data being sent from the Arduino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to **Serial.begin** in your sketch. Note that on Mac or Linux, the Arduino board will reset (rerun your sketch from the beginning) when you connect with the serial monitor.

You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

Preferences

Some preferences can be set in the preferences dialog (found under the **Arduino** menu on the Mac, or **File** on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

Boards

The board selection has two effects: the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader.

- *Arduino Duemilanove or Nano w/ ATmega328*
  An ATmega328 running at 16 MHz with auto-reset. Also used for the 16 MHz (5V) versions of the Arduino Pro or Pro Mini with an ATmega328.
- *Arduino Diecimila, Duemilanove, or Nano w/ ATmega168*
  An ATmega168 running at 16 MHz with auto-reset. Compilation and upload is equivalent to Arduino NG or older w/ ATmega168, but the bootloader burned has a faster timeout (and blinks the pin 13 LED only once on reset). Also used for the 16 MHz (5V) versions of the Arduino Pro and Pro Mini with an ATmega168.
- *Arduino Mega*
  An ATmega1280 running at 16 MHz with auto-reset.
- *Arduino Mini*
  Equivalent to Arduino NG or older w/ ATmega168 (i.e. an ATmega168 running at 16 MHz without auto-reset).
- *Arduino BT*
  ATmega168 running at 16 MHz. The bootloader burned includes codes to initialize the on-board bluetooth module.
- *LilyPad Arduino w/ ATmega328*
  An ATmega328 running at 8 MHz (3.3V) with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328.

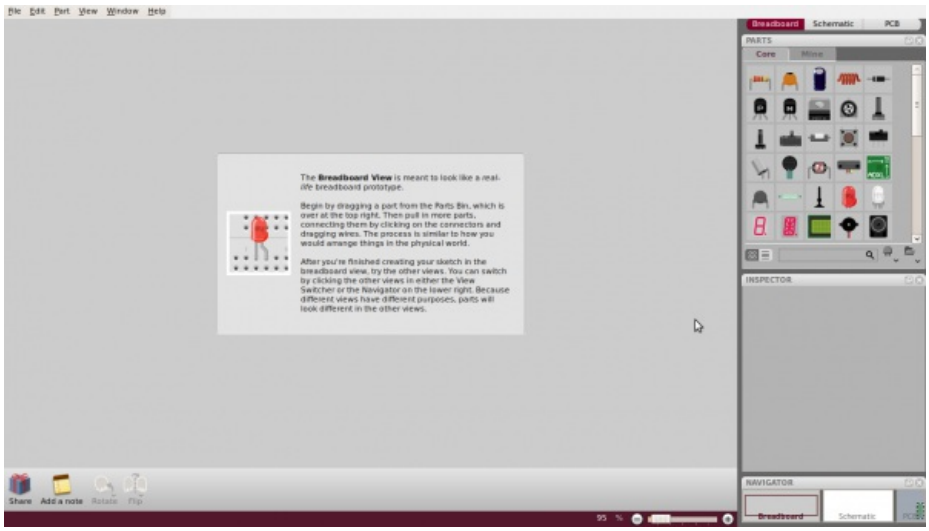
- *LilyPad Arduino w/ ATmega168*
  An ATmega168 running at 8 MHz. Compilation and upload is equivalent to the Arduino Pro or Pro Mini (8 MHz) w/ ATmega168. The bootloader burned, however, has a slower timeout (and blinks the pin 13 LED three times on reset) because the original versions of the LilyPad didn't support auto-reset. They also didn't include an external clock, so the burn bootloader command configures the fuses of ATmega168 for an internal 8 MHz clock.

  If you have a recent version of the LilyPad, (w/ a 6-pin programming header), you'll want to select Arduino Pro or Pro Mini (8 MHz) w/ ATmega168 before burning the bootloader.
- *Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328*
  An ATmega328 running at 8 MHz (3.3V) with auto-reset. Equivalent to LilyPad Arduino w/ ATmega328.
- *Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168*
  An ATmega168 running at 8 MHz (3.3V) with auto-reset.
- *Arduino NG or older w/ ATmega168*
  An ATmega168 running at 16 MHz *without* auto-reset. Compilation and upload is equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset).
- *Arduino NG or older w/ ATmega8*
  An ATmega8 running at 16 MHz *without* auto-reset.

FRITZING

Fritzing

Fritzing luotiin helpottamaan Arduino-projektien dokumentaatiota.



Alueet:

- Parts

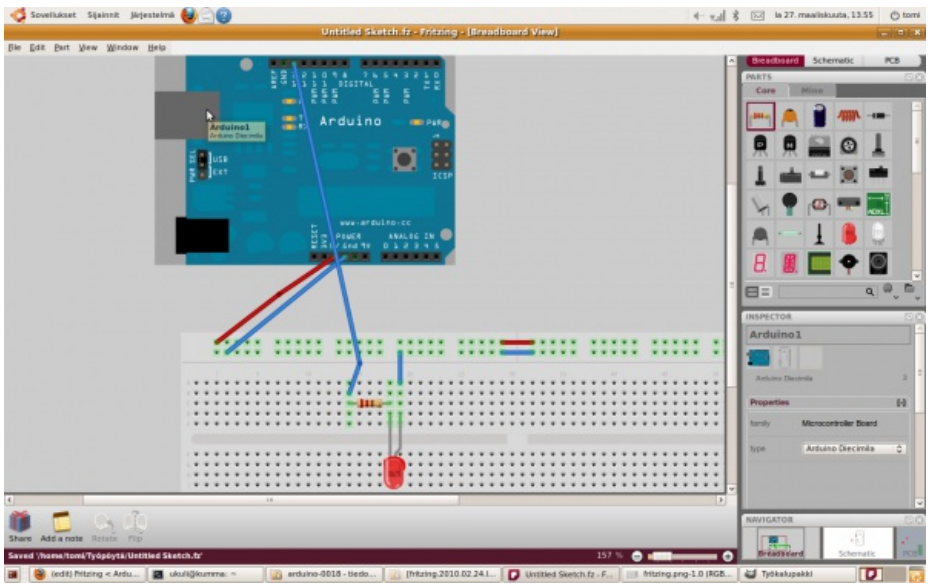
- Project Area

- Inspector

- Navigator

1) Lediesimerkki (BLINK)

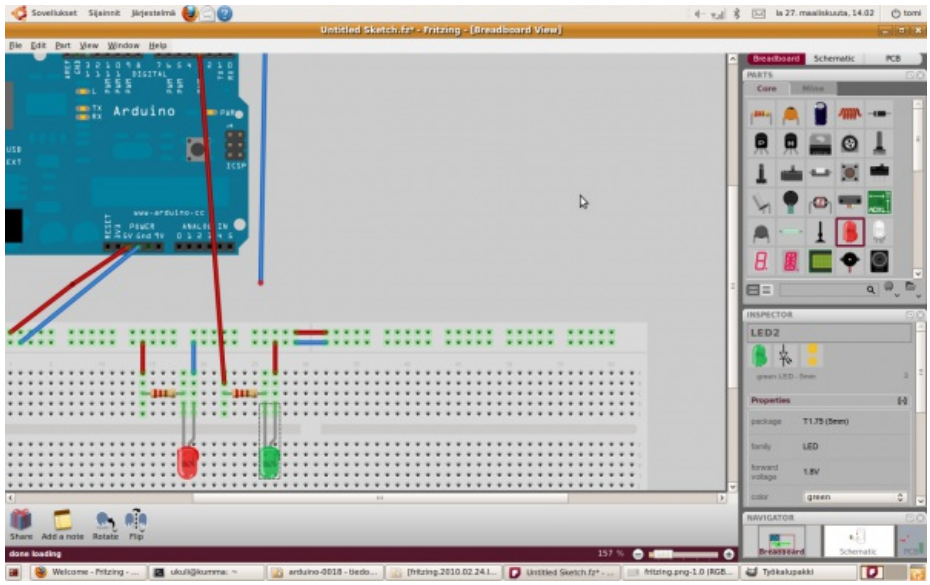
2) Pidä hiiri pohjassa - vedä breadboardilla - voit vetää johtoa ja näyttää toisiinsa kytketyt kolot.



ANALOG - SELITÄ ANALOGINEN

DIGITAL - SELITÄ DIGITAALINEN

Käytä koodilla Fading:



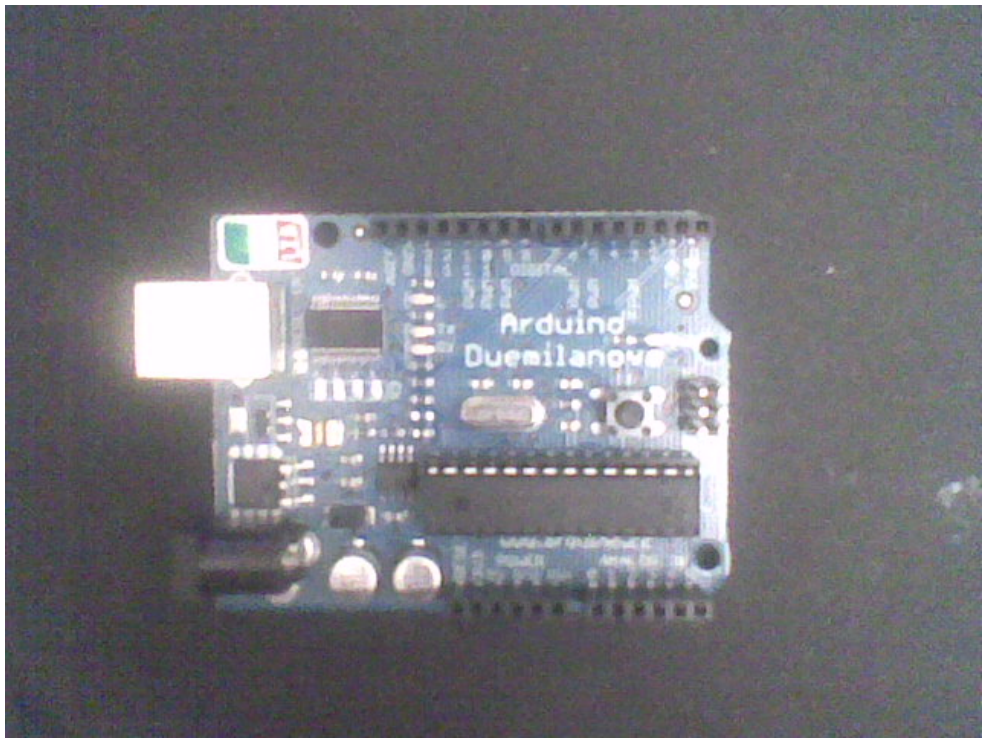
ARDUINO

Arduino

Tämä yksinkertainen esimerkki Arduinon ohjelmoinnista käytöstä yhden komponentin kanssa on otettu Arduinon kotisivulta.

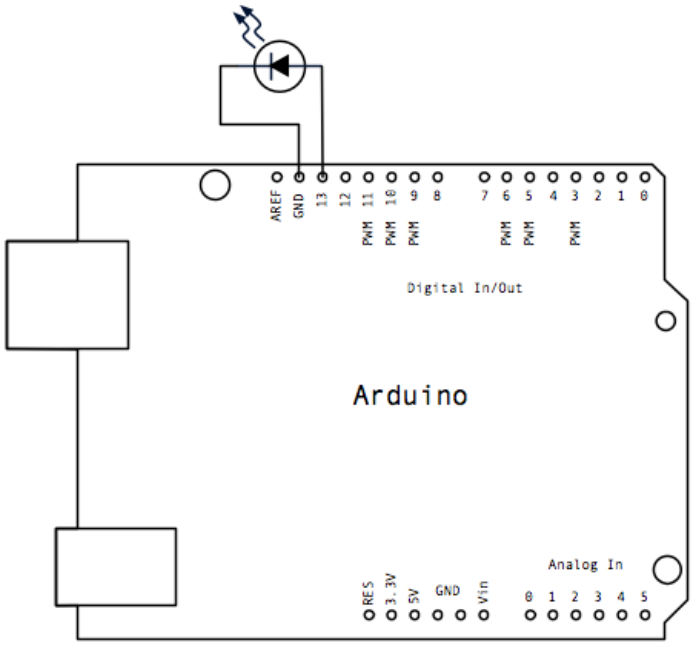
Kuvassa näemme Arduino Duomilanove -mikrokontrollerin.

(Parempi kuva!)

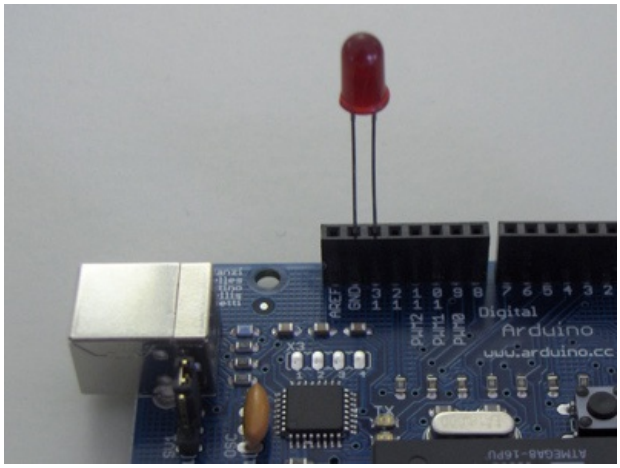


Ohjelmoidaksesi Arduinoa joudut kytkemään sen tietokoneeseesi USB-kaapelilla. USB-kaapeli toimii ohjelmointikaapelina, mutta samalla myös sähköjohtona, joten Arduino saa virran USB-johdosta.

Tässä esimerkissä tarvitset yhden ledin, jonka liität Arduinoon kaavakuvan osoittamalla tavalla.



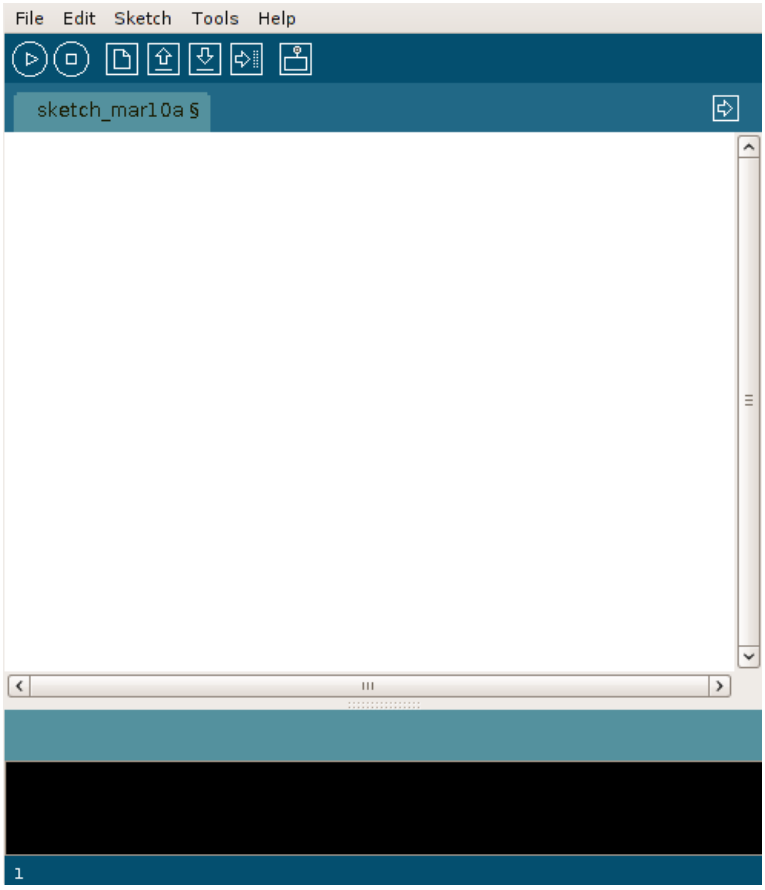
Alla ledi liitettynä Arduinoon:



Ohjelmoidaksesi Arduinoa joudut kytkemään sen tietokoneeseesi USB-kaapelilla. USB-kaapeli toimii ohjelmointikaapelina, mutta samalla myös sähköjohtona, joten Arduino saa virran USB-johdosta.

(Parempi kuva!)

Nyt voit käynnistää Arduino IDE:n tietokoneessasi. Tämä ohjelma toimii Arduinon ohjelmointiympäristönä.



Seuraavaksi voit ohjelmoida Arduinon tällä yksinkertaisella koodilla.

```
/*
  Blink

 Turns on an LED on for one second, then off for one second, repeatedly.

 The circuit:
 * LED connected from digital pin 13 to ground.

 * Note: On most Arduino boards, there is already an LED on the board
 connected to pin 13, so you don't need any extra components for this example.


 Created 1 June 2005
 By David Cuartielles

 http://arduino.cc/en/Tutorial/Blink

 based on an orginal by H. Barragan for the Wiring i/o board

This example code is in the public domain.


 */

int ledPin =  13;    // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts
```

```
void setup()   {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);   // set the LED on
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // set the LED off
  delay(1000);                  // wait for a second
}
```

OHJELMOINTIKIELI

Ohjelmointikieli

Rakenne

setup()

Asetusfunktio setup() kutsutaan ohjelmaluonnoksen alkaessa. Käytä sitä alustaaksesi muuttujat, pinnien tilat, kirjastojen käytön aloituksen, jne. Käynnistysfunktio suoritetaaan vain kerran, jokaisen käynnistyksen tai alkutilaan asettamisen yhteydessä.

Esimerkki

```
int buttonPin = 3;

void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  // ...
}
```

loop()

Asetusfunktio setup() alustaa ja asettaa alkuarvot, jonka jälkeen loop()-silmukkafunktio aloittaa silmukan, jolloin ohjelmasi voi muuttua ja reagoida tilanteen kehitykseen. Käytä sitä hallitaksesi aktiivisesti Arduino-mikrokontrolleria.

Esimerkki

```
int buttonPin = 3;

// setup alustaa sarja- ja nappipinnin
void setup()
{
  beginSerial(9600);
  pinMode(buttonPin, INPUT);
}

// silmukka tarkastaa nappipinnin joka kerralla, ja lähettää sarjalliseen, mikäli
// sitä painetaan
void loop()
{
  if (digitalRead(buttonPin) == HIGH)
    serialWrite('H');
  else
    serialWrite('L');

  delay(1000);
}
```

if (ehdollinen operaattori) ja == < > (vertailuoperaattorit)

Ehdollinen operaattori if, jota käytetään vertailuoperaattorin yhteydessä, testaa onko tietty ehto saavutettu, kuten ylittääkö sisääntulo tietyn arvon. Tällaisen testin muoto on:

```
if (jokuMuuttuja > 50)
{
  // Tee jotain tässä.
}
```

Ohjelma testaa onko jokuMuuttuja suurempi kuin 50. Jos se on, ohjelma tekee tietyn toiminnon. Toisella tavalla ilmaistuna, mikäli lausunto sulkujen sisällä on totta, aaltosulkujen sisällä olevat lausunnot ajetaan. Jos ei, ohjelma hyppää koodin yli.

Aaltosulut voidaan jättää pois if-lausunnon jälkeen. Jos näin tehdään, seuraavalla rivillä (puolipisteen määrittämästä) tulee ainoa ehdollinen lausunto.

```
if (x > 120) digitalWrite(LEDpin, HIGH);

if (x > 120)
digitalWrite(LEDpin, HIGH);

if (x > 120){ digitalWrite(LEDpin, HIGH); }

if (x > 120){
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}                                 // kaikki ovat oikein
```

Sulkujen sisällä arvioitavat lausunnot vaativat yhden tai useamman operaattorin käyttöä:

Vertailuoperaattorit:

```
x == y (x on sama kuin y)
x != y (x ei ole sama kuin y)
x <  y (x on pienempi kuin y)
x >  y (x on suurempi kuin y)
x <= y (x on pienempi tai yhtäsuuri kuin y)
x >= y (x on suurempi tai yhtäsuuri kuin y)
```

ELEKTRONIIKKA

Elektroniikka

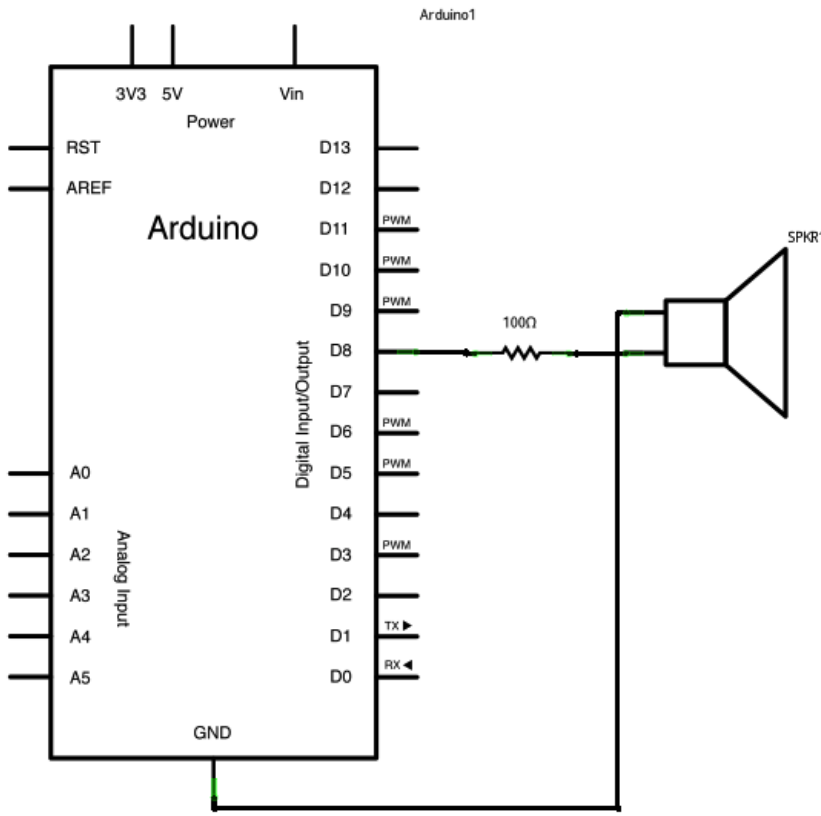
Piezo-summeri

Melodian soittaminen tone() -funktiolla

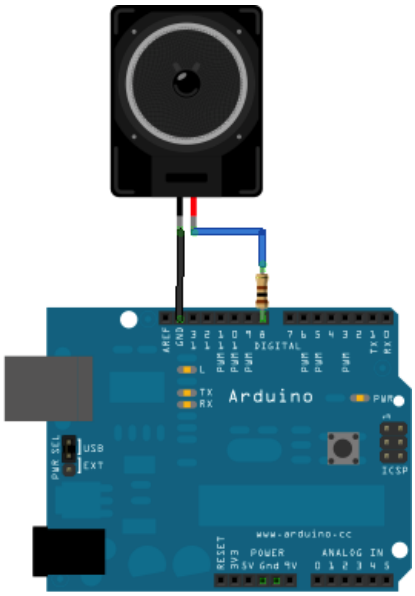
Tätä testiä varten tarvitset Arduinon, piezo-summerin, 100 ohmin resistorin, hyppylankoja ja testipöydän.

Sadan ohmin resistorissa on ruskea, musta, ruskea ja kultainen viiva.

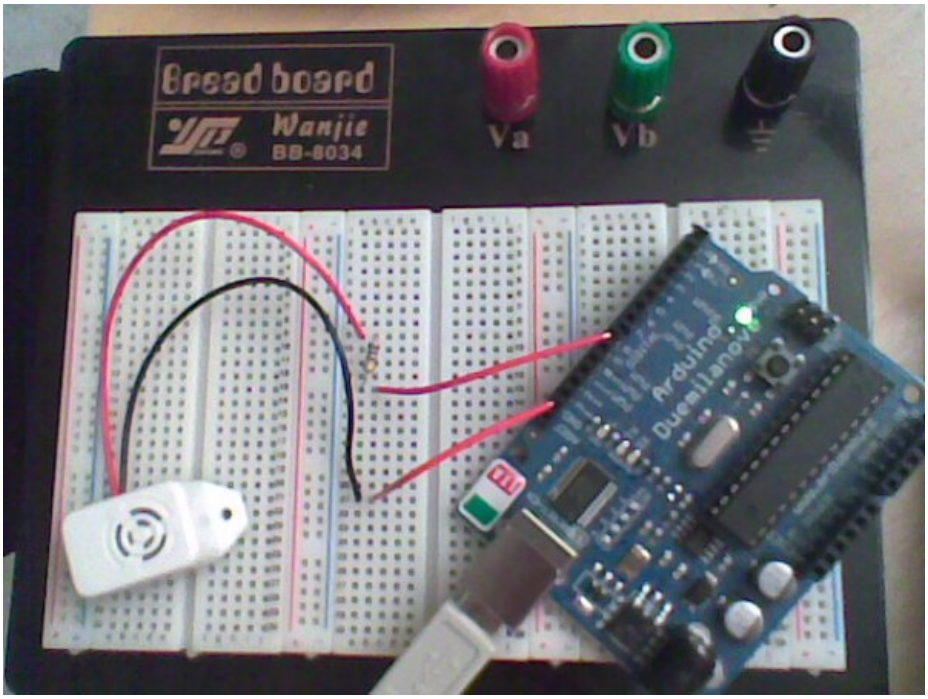
Alla on projektin kytkentäkaavio.



Fritzing-ohjelmalla tehty kuva kytkennöistä.



(Parempi kuva...)



Ensin joudut laatimaan pitching.h -ohjelman. Tässä määritellään nuotit.
Paina ensin uusi välilehti -nappia:

Nimeä uusi tiedosto nimellä pitching.h ja paina OK-nappia.

Kirjoita uuteen välilehteen seuraava koodi:

```
/*************************************************
 * Public Constants
```

```
 ************************************************/

#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
```

```
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978
```

Kirjoita sen jälkeen tämä koodi päävälilehteen.

```
/*
  Melody

 Plays a melody

 circuit:
 * 8-ohm speaker on digital pin 8

 created 21 Jan 2010
 by Tom Igoe

This example code is in the public domain.

 http://arduino.cc/en/Tutorial/Tone

 */
 #include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3,NOTE_G3, NOTE_A3, NOTE_G3,0, NOTE_B3, NOTE_C4};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4,4,4,4,4 };

void setup() {
  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {

    // to calculate the note duration, take one second
    // divided by the note type.
```

```
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000/noteDurations[thisNote];
    tone(8, melody[thisNote],noteDuration);

    // to distinguish the notes, set a minimum time between them.
    // the note's duration + 30% seems to work well:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
  }
}

void loop() {
  // no need to repeat the melody.
}
```

Paina upload-nappia ja Arduino soitaa piezo-summerilla melodian: 

Servomoottori

Esimerkki servomoottorin liittämisestä Arduinoon ja ohjelmoinnista.

Sensori

Esimerkki sensorin liittämisestä Arduinoon ja sen ohjelmoinnista.

ROBOTTI

Robotti

Kokonaisen robotin rakentaminen!

ROBOTIN OHJELMOINTI

Ohjelmointi

TEKIJÄT

Lisenssi

Arduino.cc

Tämän oppaan pohjana on käytetty paljon Arduino.cc -sivuilla Creative Commons -lisenssillä julkaistua dokumentaatiota ja sivun koodiesimerkkejä, jotka on julkaistu Public Domainissa.

Kirjoittajat

OHJELMOINTIESIMERKKI
© Tomi Toivio 2010

ASENNUS UBUNTUUN
© Tomi Toivio 2010

AVOIN LÄHDEKOODI
© Tomi Toivio 2010

TEKIJÄT
© adam hyde 2006, 2007
Modifications:
Tomi Toivio 2010

ELEKTRONIIKKA
© Tomi Toivio 2010

FRITZING
© Tomi Toivio 2010

JOHDANTO
© adam hyde 2006, 2007
Modifications:
Tomi Toivio 2010

SERVOMOOTTORI
© Tomi Toivio 2010

ROBOTIN OHJELMOINTI
© Tomi Toivio 2010

OHJELMOINTIKIELI

OHJELMOINTIYMPÄRISTÖ

PIETSOSUMMERI

ROBOTTI

SENSORI

Free manuals for free software

General Public License

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating

that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Made with Booki

Visit http://software.booki.cc