As of June 16, 2024

A Step-By-Step Hands-on Guide to Designing IoT Devices using Arduino Nano with AI

http://neuro.musashino-u.ac.jp/~takefuji/arduinoNanoE.pdf

By Yoshiyasu Takefuji

Contents

Summary

Since the advent of open-source software, it is inexpensive and easy to build an IoT (internet of things) device by using Arduino Nano. This book illustrates how to design and implement IoT devices using Arduino Nano. Arduino Nano has a USB interface for flashing programs and bi-directional serial communications. Arduino Nano is composed of an ATmega328 device with 32kB flash memory, 2kB RAM, and 1kB EEPROM. ATmega328 has 8-channel 10-bit ADC and 23 digital I/O pins. ATmega328 device is able to control i2c and SPI devices with UART serial communications. This book shows design examples including multi-tasking, IoT with SIGFOX (Low Power Wide Area Network: LPWAN), thermal sensors for tracking a heat source, and USB-host for distance measurement. In this book, Windows 10 is mainly used for developing with Arduino Nano. MacOS information is little discussed.

You should buy the following parts at least:

1. Soldered Arduino Nano,
https://akizukidenshi.com/catalog/g/gM-16282/
2. Two breadboards,
https://akizukidenshi.com/catalog/g/gP-00315/
3. 0.65mm single wires at least more than 2m,
https://akizukidenshi.com/catalog/g/gP-08996/
4. 0.8mm wire stripper (use 0.8mm hole for 0.65mm wire),
https://www.amazon.co.jp/%E3%83%99%E3%83%83%E3%82%BB%E3%83%AB-VESSEL-3500E-1-%E3%83%AF%E3%82%A4%E3%83%A4%E3%83%BC%E3%82%B9%E3%83%88%E3%83%AA%E3%83%83%E3%83%91%E3%83%BC-No-3500E-1/dp/B000CED24U/ref=sr_1_6?
or
https://www.amazon.co.jp/%E3%83%99%E3%83%83%E3%82%BB%E3%83%AB-VESSEL-3500E-2-%E3%83%AF%E3%82%A4%E3%83%A4%E3%83%BC%E3%82%B9%E3%83%88%E3%83%AA%E3%83%83%E3%83%91%E3%83%BC-No-3500E-2/dp/B000CEAXM4/ref=sr_1_7?

5. Several Green LEDs (with internal resistor),
https://akizukidenshi.com/catalog/g/gI-06246/

Options:
ADXL345 (I2C),
OLED128x32 (I2C with 4 pins),
BME280 (I2C),
servo motor (SG-90 or SG-90-HV).

Chapter 1 Where can I buy Arduino Nano?

Arduino Nano is composed of ATmega328 and USB serial for flashing programs. Signals including TXD, RXD can be used through USB port. One piece is around 1500 Yen. You should buy soldered Arduino Nano if your soldering skill is poor.

Remember that you must make sure the delivery schedule when ordering because of the covid-19 pandemic. Buy stocked components in Japan. DO NOT ORDER ANY COMPONENTS ABROAD.

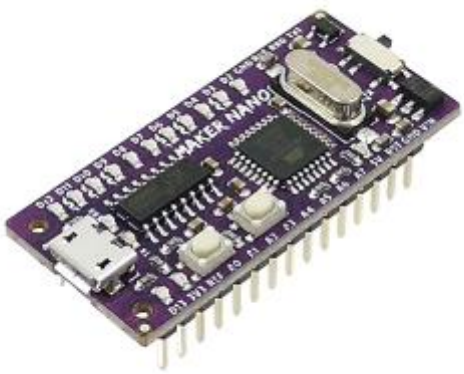


Fig. 1-1 Arduino Nano from Akizuki

You can buy Arduino Nano from Akizuki:
https://akizukidenshi.com/catalog/g/gM-16282/

Fig.1-2 MicroB USB connector
https://akizukidenshi.com/catalog/g/gC-07607/

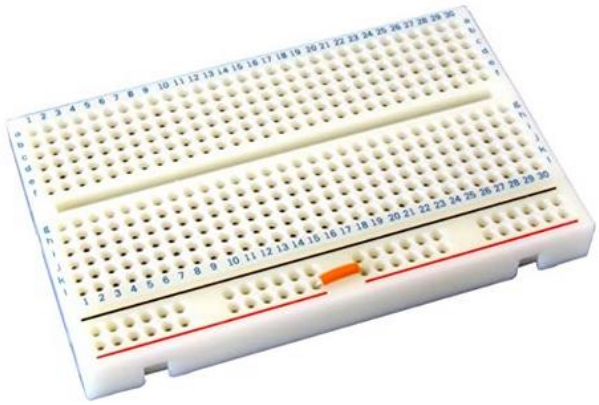


Fig. 1-3 Breadboard

You need to buy two breadboards from akizuki:

https://akizukidenshi.com/catalog/g/gP-00315/

Breadboard connections are described where holes in a green circle are connected.

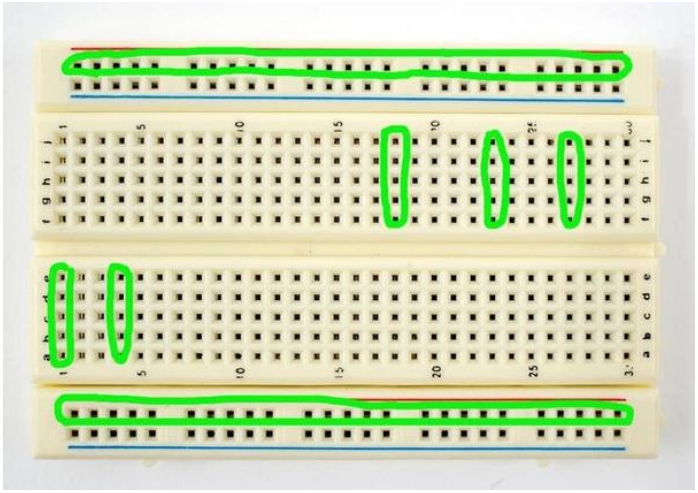


Fig. 1-4 breadboard connections

The following example shows how components are connected by 0.65mm single wires:

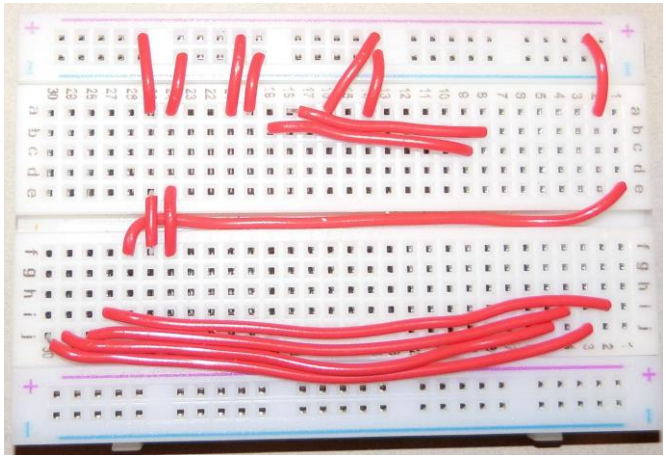


Fig. 1-5 Breadboard with 0.65mm single wires

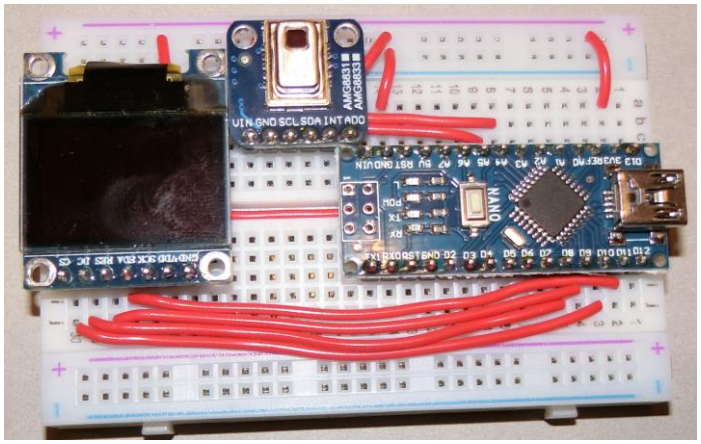


Fig. 1-6 Breadboard with components and 0.65mm single wires

You can stick the necessary components (Arduino Nano, LED,…) where 0.65mm single wires are used for establishing connections.

DO NOT BUY JUMPER CABLES!!!

The reliability of jumper cables is questionable due to their inconsistent connections with breadboards.

Consider purchasing a single wire that is 0.65mm in diameter and at least 2 meters in length.

https://akizukidenshi.com/catalog/g/gP-08996/
You only need 2m so that this will be for 10 students

Wire stripper tool:

https://www.amazon.co.jp/%E3%83%99%E3%83%83%E3%82%BB%E3%83%AB-VESSEL-3500E-1-%E3%83%AF%E3%82%A4%E3%83%A4%E3%83%BC%E3%82%B9%E3%83%88%E3%83%AA%E3%83%83%E3%83%91%E3%83%BC-No-3500E-1/dp/B000CED24U/ref=sr_1_6?

You should use **0.8mm hole** for stripping 0.65mm wires. DO NOT USE 0.65mm hole of the wire stripper since you may cut 0.65mm wires.

Chapter 2 How to setup the development environment for Arduino Nano

You can download the latest Arduino library for your operating system from:
https://www.arduino.cc/en/Main/Software

You can download the following example: led0.ino
http://neuro.musashino-u.ac.jp/~takefuji/led0.ino
https://github.com/y-takefuji/IoT/raw/main/led0.ino

http://neuro.musashino-u.ac.jp/~takefuji/list.html

In your .bashrc, .zshrc or .profile file, add the following line:
take= 'http://neuro.musashino-u.ac.jp/~takefuji'

Run the following commands on your terminal.
$ source .bashrc
$ echo $take
The previous command show the proper text. To download the led0.ino, run the following command.
$ wget $take/led0.ino

If your Arduino software was successfully installed on your PC, Arduino software will show the popped screen by clicking the downloaded file led0.ino as shown in Fig. 2-1. In Fig. 2-1 “int ledPin=2;” in led0.ino program indicates the location of D2 of LED’s short pin. Note that a LED has a long pin (GND) and short pin (Positive DC).

You must firmly inset Arduino NANO on the breadboard. Make sure that you have to insert a LED between long pin at GND and short pin at D2 as shown in Fig. 2-2.

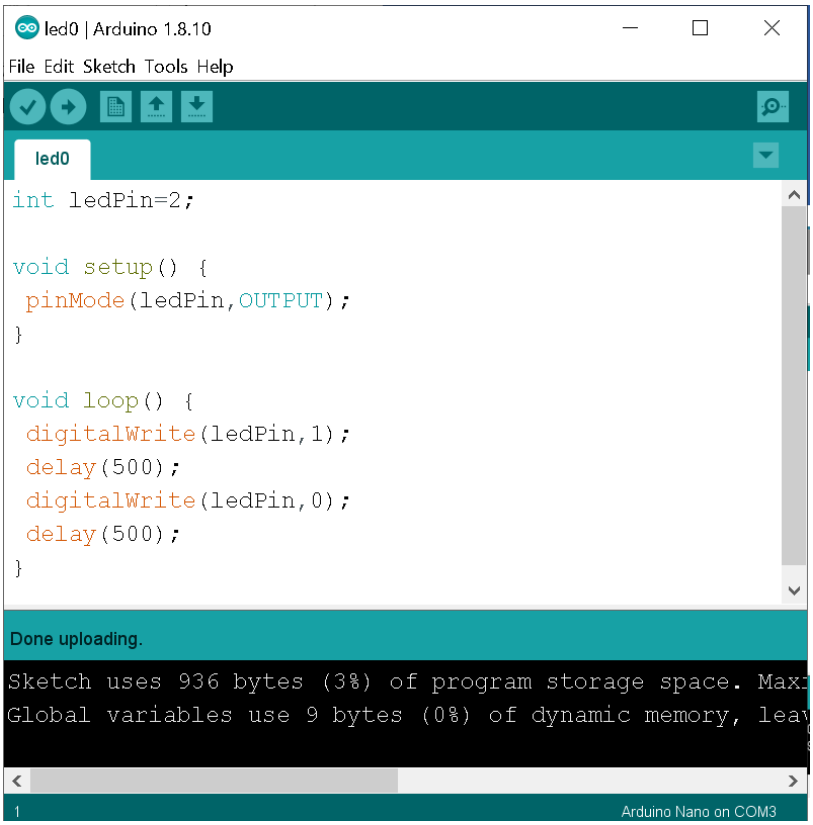


Fig. 2-1 Arduino screen of led0.ino

Ensure that the Arduino Nano is securely connected to the breadboard.

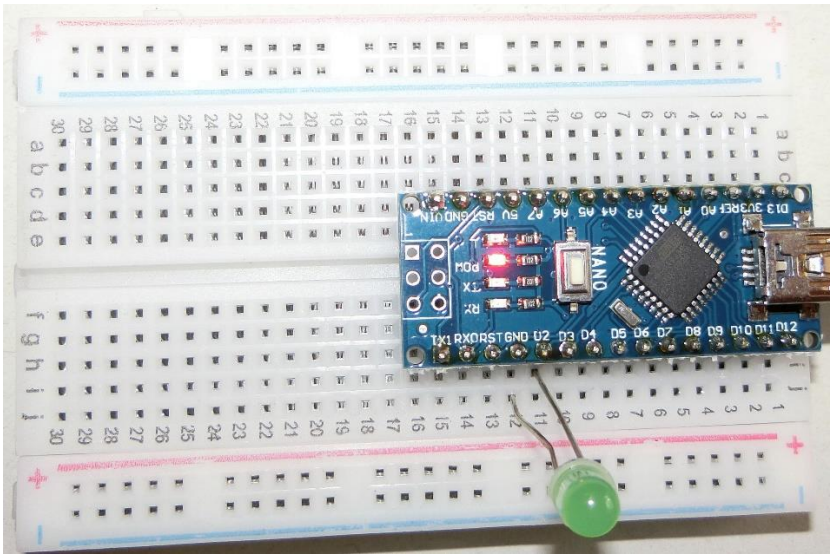
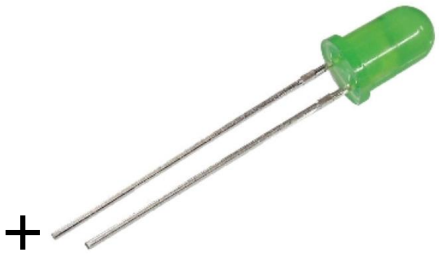


Fig. 2-2 A LED flashing by led0.ino (**long pin: D2, short pin: GND**)



LED**: long pin: D2, short pin: GND**

Before running the following program, you have to setup the connection between your PC and Arduino Nano. Fig. 2-3 shows how you can set Port "COM3" in this example. Depending on your PC, this COM number or COM name may be different. On Mac, you may have the longer name.

Fig. 2-3 indicates Arduino Nano port number is COM3. The followings are needed for establishing connection between PC and Arduino Nano.

1. From Menu Tools -> Port, select proper COM number.
2. Select Board from Menu Tools, select the followings respectively:
   Board: "Arduino Nano"
   Processor: "ATmega328"
   Port:"COM3"
   Programmer: "Arduino as ISP”

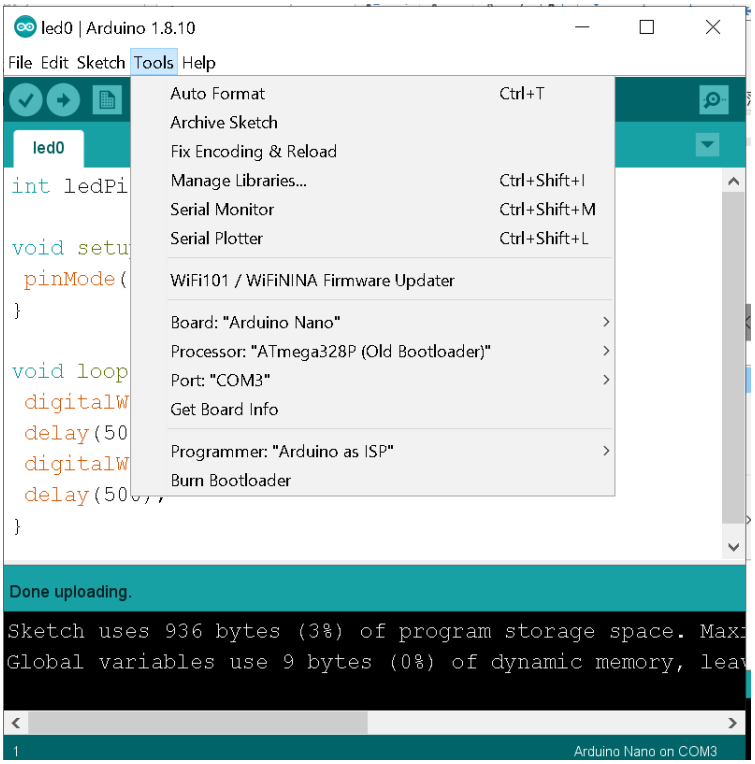


Fig. 2-3 Arduino Nano setting screen

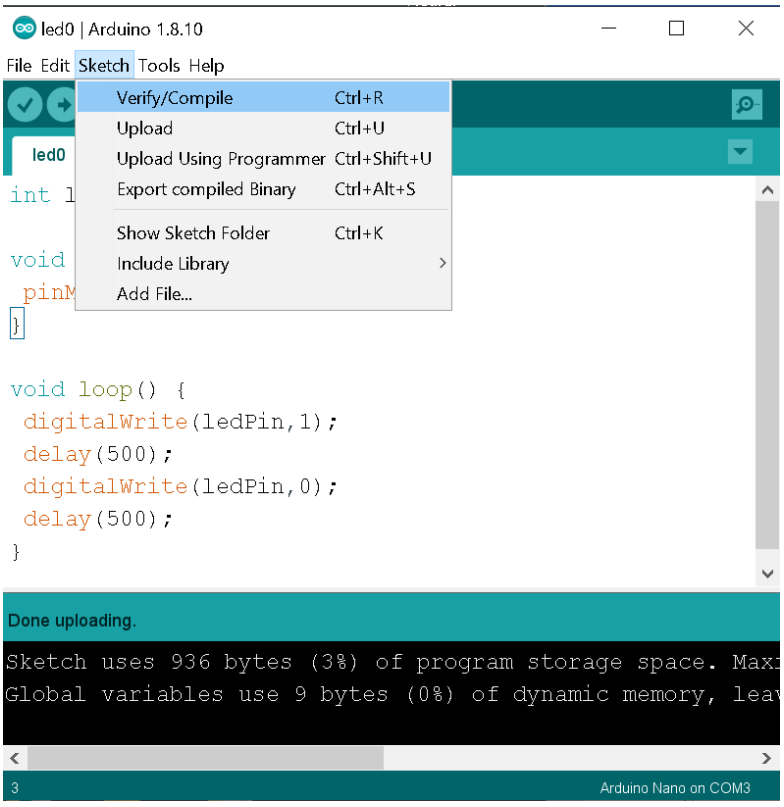


Fig. 2-4 Compile and Upload

In order to compile your program, select Sketch menu→ Verifi/Compile menu. Run Sketch menu → Upload menu, then hex file will be flashed into ATmega328.

If you select Sketch menu→ Upload Using Programmer, Bootloader will be flashed together. If you run Sketch menu → Export compiled Binary, then hex file will be generated in led0 folder.

Tools menu → Serial Monitor can change baud rate of Arduino Nano USB. Default baud rate is 9600. In this example, 115200Baud is set.

After selecting Upload command, led0.ino will blink a LED every second. It can be seen at:
https://www.youtube.com/watch?v=CITnEkizLRI

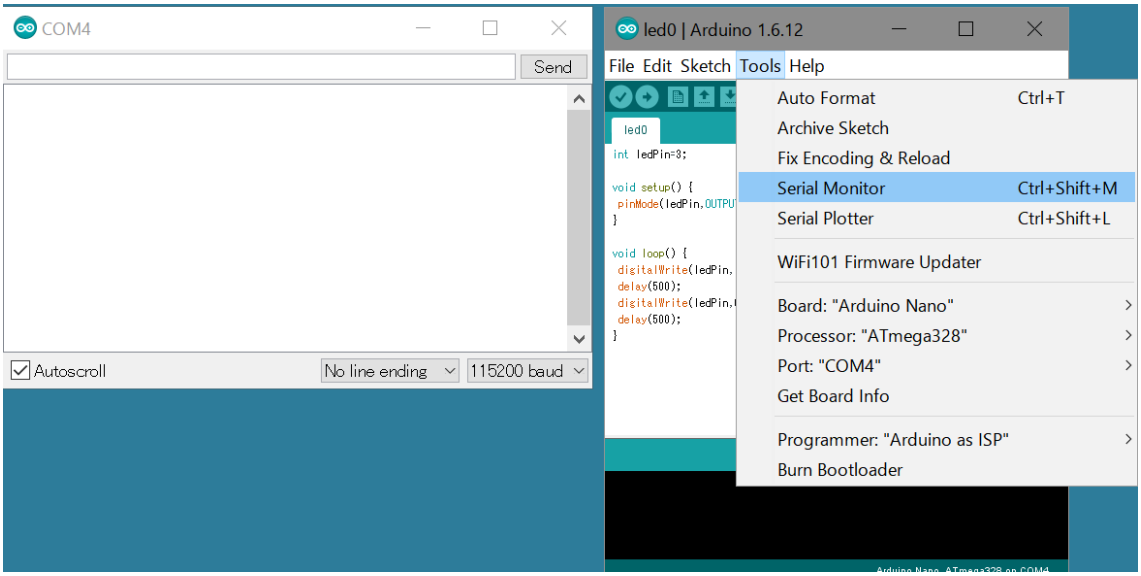


Fig. 2-5 Monitor screen

*How to install python on your PC,
Read https://github.com/y-takefuji/python-novice

Assignments:

1. You should practice 6 projects using led0.ino (digital output), led1.ino (digital input), led2.ino (analog input), led3.ino (serial from Arduino to PC), and led4.ino with blink.py (serial from PC to Arduino) respectively. blink.py sends ‘1’ or ‘0’ to

Arduino to control a single LED.

2. Make string.ino. When you can send “123” from PC for lighting up LED and “000” for lighting off LED.
3. Build blinkN.py and blinkN.ino where PC can send the number and Arduino’s LED blinks the received number of times.
4. Download Arduino-Scheduler-master.zip from my site and import Scheduler.cpp and Scheduler.h. Download led_m.ino and practice it.
5. Make a firefly.ino program as shown at:
   https://youtu.be/WpZSlk13-vg
6. Create led_pgen.ino for Arduino Nano and python program plotserial.py on PC where a single LED generates electric power by feeding light to LED where the generated electric power values will be sent to PC through USB. On PC, generated electric power values are plotted on screen.
   https://youtu.be/Lj34cn2UcCo
7. Make an led_switch.ino demonstrating the duality function of a single LED.
   https://youtu.be/4Hk6RGi1zLQ
8. Download and expand adxl345_pitch_roll.tar and practice adxl345 I2C device.
   http://neuro.musashino-u.ac.jp/~takefuji/adxl345_pitch_roll.tar
   https://youtu.be/7ekkDMEk4o0
9. Practice BME280 I2C sensor:
   Download bme280.ino
   http://neuro.musashino-u.ac.jp/~takefuji/bme280.ino
10. Download OLED128x64.ino and practice OLED128x64 I2C device:
    http://neuro.musashino-u.ac.jp/~takefuji/OLED128x64.ino
11. Practice combination of OLED128x64, ADXL345, and BME280 I2C devices.
    Download oled128x64_adxl345_bme280.tar file:

http://neuro.musashino-u.ac.jp/~takefuji/oled128x64_adxl345_bme280.tar

https://www.youtube.com/watch?v=0GyXGndFAKk

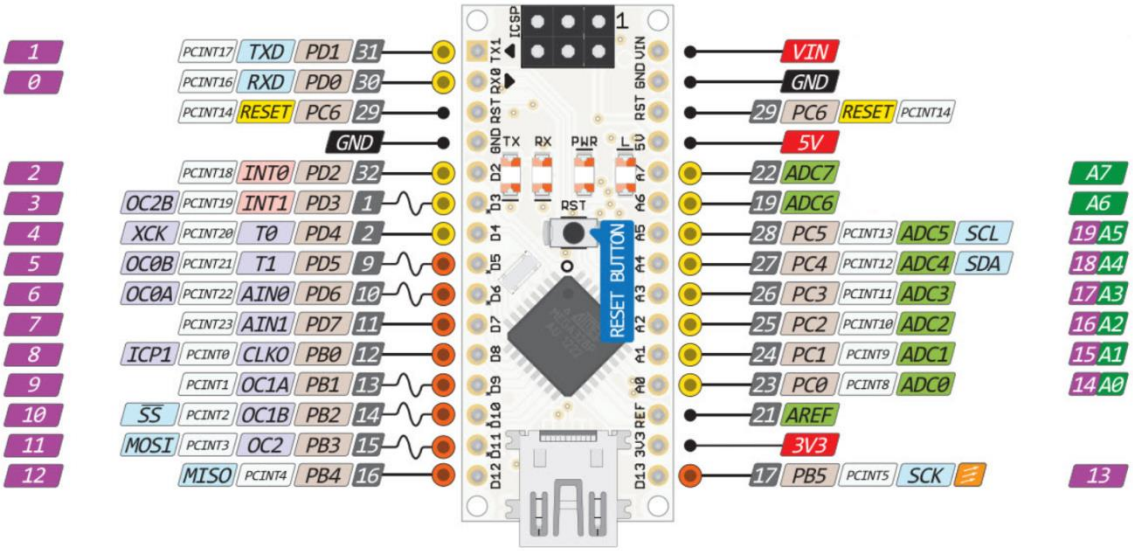


Fig. 2-6 Arduino Nano pin-configuration

2.1 TROUBLE SHOOTING

For Windows users, download the file, WCH-Nano-Drivers.zip and install it.
https://arduino-info.wikispaces.com/file/view/WCH-Nano-Drivers.zip

For Mac users, download the file, CH341SER_MAC.ZIP and install it.
http://neuro.musashino-u.ac.jp/~takefuji/CH341SER_MAC.ZIP

2.2 How to add libraries.

Multi-tasking with Arduino
Download the following file:
https://github.com/mikaelpatel/Arduino-Scheduler/archive/master.zip
Portable Scheduler by Mikael Patel

Click Sketch->Include Library->Add ZIP Library and open downloaded the master.zip file for installing arduino-scheduler.

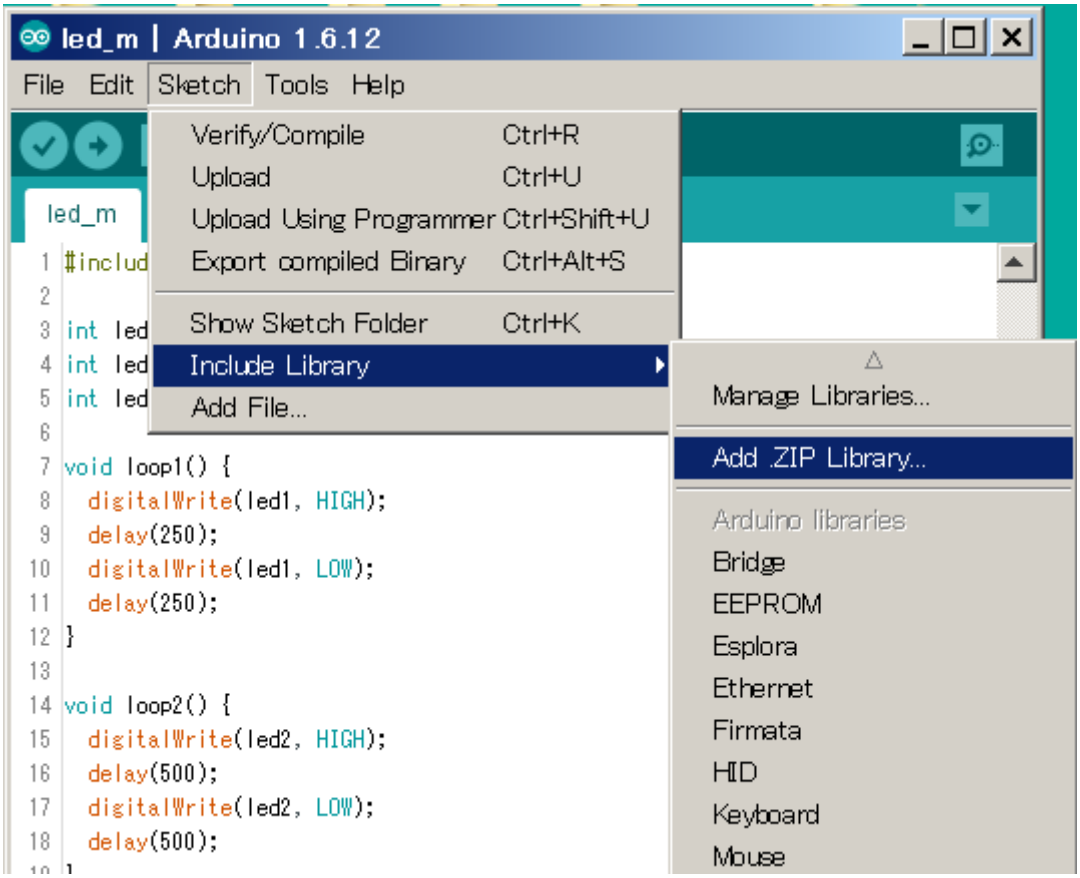


Fig. 2-7 how to add library in your system

Or click Sketch->Include Library->Manage Libraries.

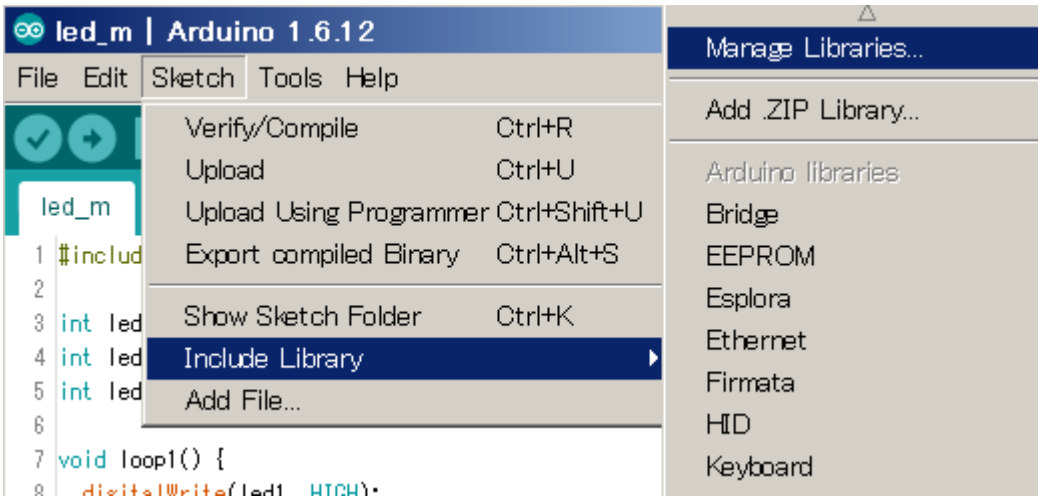


Fig. 2-8 how to find library from the existing libraries

And type "Arduino-Scheduler" in Library Manager and install the Portable Scheduler.

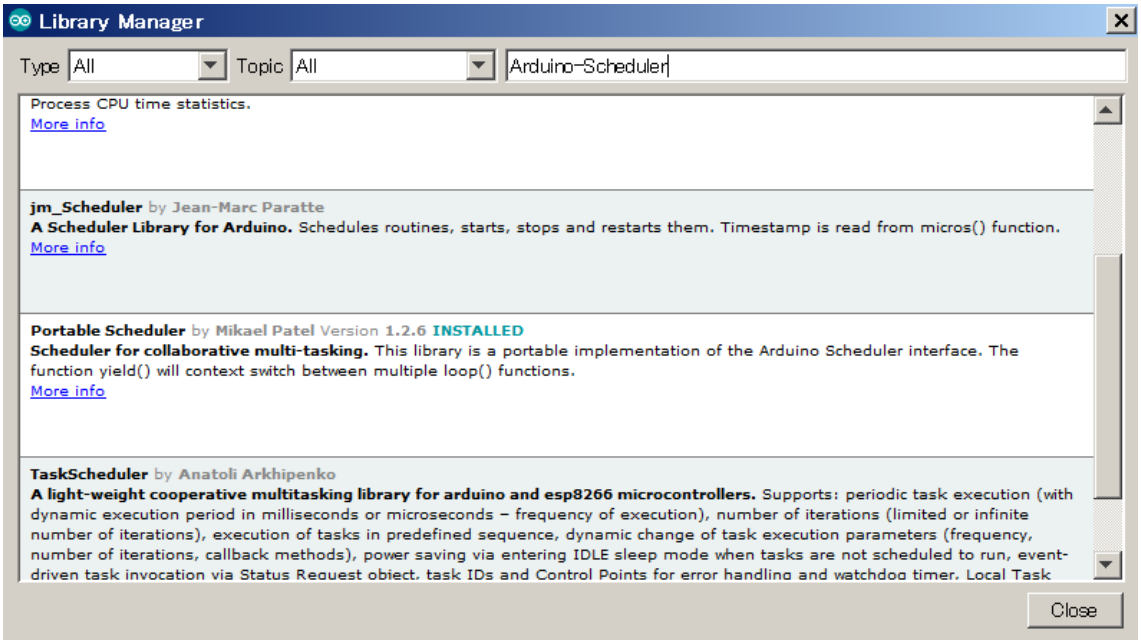


Fig. 2-9 Searching Arduino-scheduler library

Download the multi-task program: led_m.ino
http://neuro.musashino-u.ac.jp/~takefuji/led_m.ino
https://github.com/y-takefuji/IoT/raw/main/led_m.ino

This example shows three LED blinking: 500ms, 1 second, 2 second blinking simultaneously.

led_m | Arduino 1.6.12

File Edit Sketch Tools Help

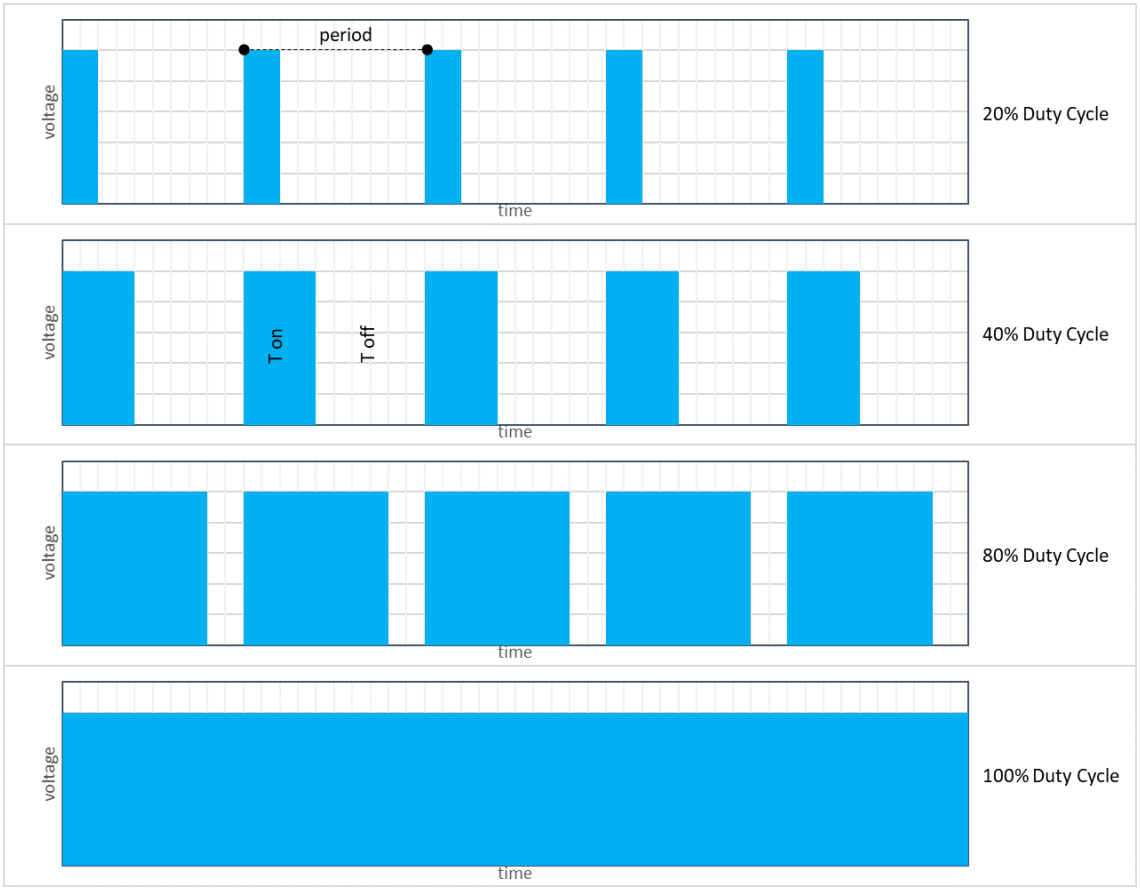
led_m

```
#include <Scheduler.h>

int led1 = 8;
int led2 = 9;
int led3 = 10;

void loop1() {
  digitalWrite(led1, HIGH);
  delay(250);
  digitalWrite(led1, LOW);
  delay(250);
}

void loop2() {
  digitalWrite(led2, HIGH);
  delay(500);
  digitalWrite(led2, LOW);
  delay(500);
}

void setup() {
 pinMode(led1,OUTPUT);
 pinMode(led2,OUTPUT);
 pinMode(led3,OUTPUT);
 Scheduler.startLoop(loop1);
 Scheduler.startLoop(loop2);
}

void loop() {
  digitalWrite(led3, HIGH);
  delay(1000);
  digitalWrite(led3, LOW);
  delay(1000);
}

```

Fig. 2-10 Multi-tasking program

Duty cycle

In order to change the light intensity of LED, pwm (pulse width modulation) is used:



Chapter 3 Major components (OLED, BMP280, ADXL345, MOSFET)

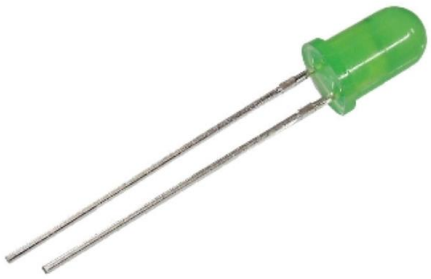


Fig. 3-1 green LEDs with internal resistor (long leg: +, short leg: GND)
http://akizukidenshi.com/catalog/g/gI-06246/

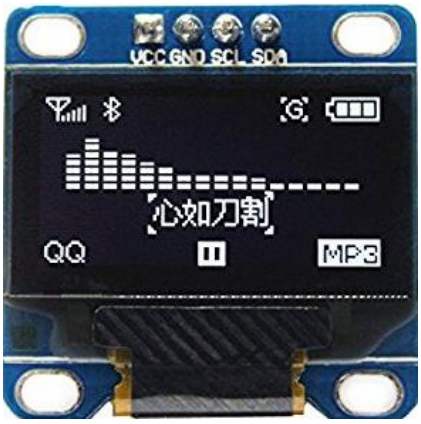


Fig. 3-2 128x64 OLED I2C display
Around 800 Yen: **BUY I2C device**

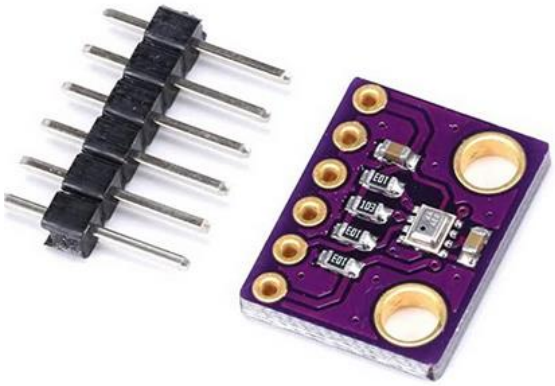


Fig. 3-3 BME280 I2C device (air pressure humidity temperature)
Around 700 Yen

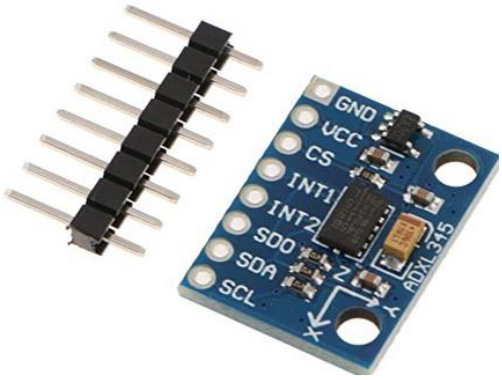


Fig. 3-4 ADXL345 I2C device (3D accelerometer)

Fig. 3-5 Wire stripper

Around 2000 Yen, you should strip 0.65mm single wire by **0.8mm stripper hole**.

Fig. 3-6 GY-MAX4466 (Electret Microphone Amplifier)

Around 400 Yen

http://neuro.musashino-u.ac.jp/~takefuji/mic.ino

https://github.com/y-takefuji/IoT/raw/main/mic.ino

Demo is at:

https://www.youtube.com/watch?v=yCwpIfL0nUM

Fig. 3-7 HC-SR04 (Ultrasonic Wave Detector Ranging Module)

Around 100 Yen

http://neuro.musashino-u.ac.jp/~takefuji/ultrasonic.ino

https://github.com/y-takefuji/IoT/raw/main/ultrasonic.ino

https://www.youtube.com/watch?v=pa6nUzGx_k0

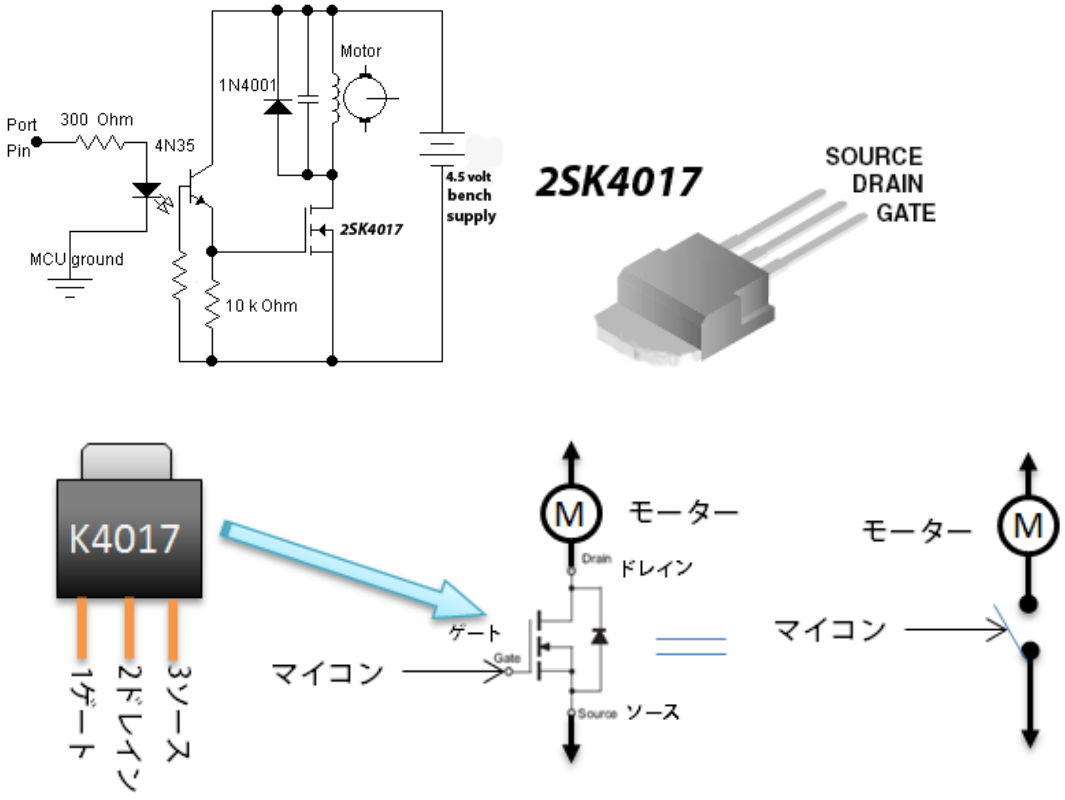


Fig. 3-6 N-channel MOSFET

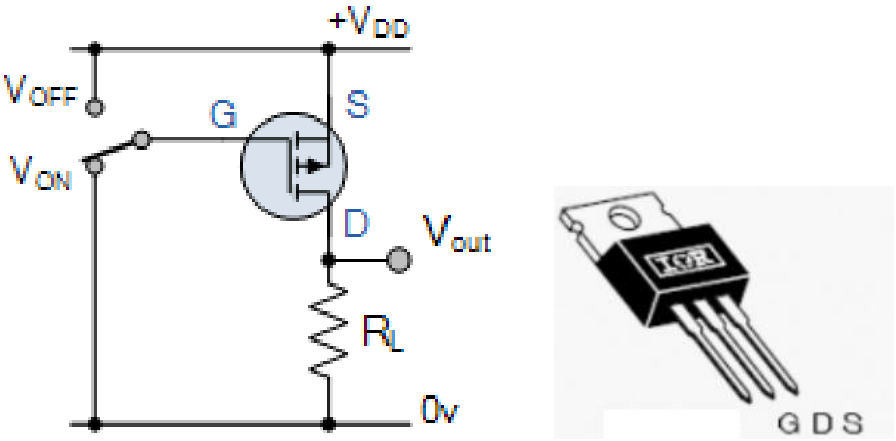


Fig. 3-7 P-channel MOSFET(IRFU5505)

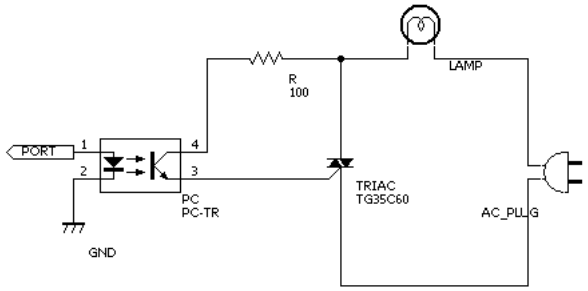


Fig.3-8 Triac (TG35C60) to control AC lamp
https://akizukidenshi.com/catalog/g/gK-00210/

ArduinoNano → PC (communications)
mic.ino oscillo.py

--------mic.ino---------

```
int mic=A7;
unsigned int val=0;

void setup() {
Serial.begin(115200);
```

```
}

void loop() {
for(int i=0;i<178;i++){
   val +=analogRead(mic);
}
val=val/178;
Serial.println(val,DEC);
val=0;
}
```

---------oscillo.py----------

```
import pyrealtime as prt

if __name__ == '__main__':
  s= prt.SerialReadLayer(device_name='COM3', baud_rate=115200)
  prt.TimePlotLayer(s, window_size=50, ylim=(-10, 500))
  prt.LayerManager.session().run()
```

PC → ArduinoNano (PC sends a command to Arduino)

--------blinkN.ino---------

See the fun projects:

https://github.com/y-takefuji/IoT

Chapter 4 Schematic Editor (Bsch)

For Windows users:

Run VC_redist.x86.exe in Runtime folder before executing bsch3v.exe.

https://www.suigyodo.com/online/bs3vp160504rtl.zip

https://www.suigyodo.com/online/bs3vp240108rtl.zip

or

https://www.suigyodo.com/online/e/bsc092e3.zip

For Mac users:

https://neuro.musashino-u.ac.jp/kenkyukai/bsch3vm-0.47.007.tar.bz2

or

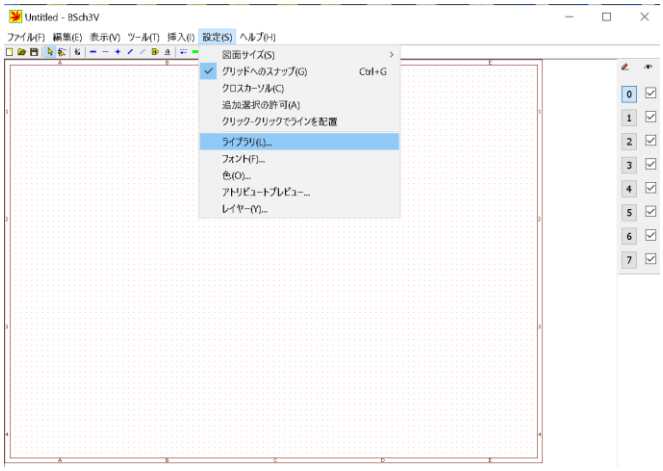
https://ftp.vector.co.jp/58/26/767/bsch3vm-0.47.008.tar.bz2

You must download components for Bsch editor:

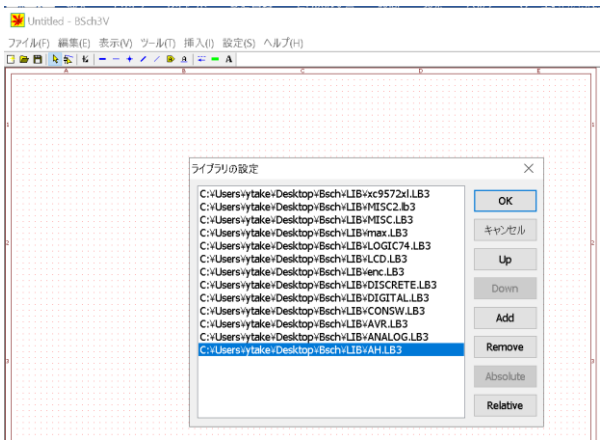
http://neuro.musashino-u.ac.jp/~takefuji/bsch_lib.tar

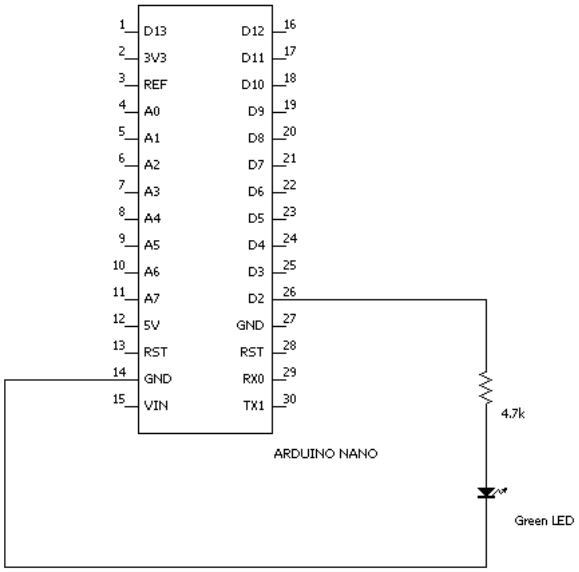
How to install components for Bsch.

1. Start Bsch by clicking Bsch execution file.



2. Open 設定→ライブラリ
3. Copy and Paste LB3 files





Circuit diagram of led0.ino

Chapter 5 Firmata Library

Without programming C++, Firmata on Arduino allows users to control Arduino from Python programs on PC.

1. You should install StandardFirmata.
   File→Examples→Firmata→FirmataStandard
2. Download Python examples:
   firmataled.py (digital write)

https://github.com/y-takefuji/IoT/blob/master/firmataled.py

   firmatadigitalread.py (digital read),

https://github.com/y-takefuji/IoT/blob/master/firmatadigitalread.py

   and firmataled2.py (analog read)

https://github.com/y-takefuji/IoT/blob/master/firmataled2.py

Chapter 6 How to use MediaPipe for hands and poses

Hands recognition

airpiano.py can be downloaded from:
https://neuro.musashino-u.ac.jp/~takefuji/airpiano.py
https://github.com/y-takefuji/airpiano

aircalc.py can be downloaded from:
https://neuro.musashino-u.ac.jp/~takefuji/aircalc.py
https://github.com/y-takefuji/air_calculator

Poses recognition

https://github.com/y-takefuji/mediapipe_pose