



 免費電子書

學習

arduino

Free unaffiliated eBook created from
Stack Overflow contributors.

#arduino

.....	1
1: arduino	2
.....	2
Arduino	2
Arduino	2
.....	2
Examples.....	2
.....	2
.....	2
.....	3
.....	4
.....	4
.....	4
LED -	4
2: Arduino IDE	5
Examples.....	5
Windows.....	5
Windows.....	5
Fedora.....	5
Ubuntu.....	5
macOS.....	5
3: I2C	6
.....	6
Examples.....	6
.....	6
4: MIDI	9
.....	9
Examples.....	9
MIDI THRU.....	9
MIDI.....	9
MIDI.....	10

MIDI.....	12
5: PWM -	15
Examples.....	15
PWM.....	15
.....	15
.....	15
.....	15
.....	16
PWMTLC5940.....	16
6: PythonArduino Uno	18
.....	18
.....	18
.....	18
Examples.....	18
ArduinoPython.....	18
7: SPI	20
.....	20
.....	20
.....	20
SPI.....	20
Examples.....	21
SPI1.....	21
8:	22
.....	22
.....	22
.....	22
Examples.....	22
.....	22
9:	24
.....	24
.....	24

.....	24
Examples.....	24
.....	24
Base64.....	25
.....	25
Python.....	26
Arduino.....	26
.....	26
10:	27
.....	27
.....	27
Examples.....	27
.....	27
11:	28
.....	28
Examples.....	28
.....	28
.....	28
12:	30
.....	30
Examples.....	30
.....	30
.....	31
13: EEPROM.....	33
.....	33
.....	33
.....	33
Examples.....	33
EEPROM.....	33
14: ArduinoAtmel Studio 7.....	34
.....	34
.....	

.....	34
.....	36
.....	38
.....	38
.....	39
.....	39
Examples.....	39
Atmel Studio 7.....	39
15:	40
.....	40
.....	40
Examples.....	40
.....	40
.....	40
.....	41
.....	41
16:	43
.....	43
.....	43
.....	43
Examples.....	43
.....	43
17:	45
.....	45
Examples.....	45
.....	45
18:	46
Examples.....	46
cardInfo.....	46
SD.....	48

SD	49
SD	50
Listfiles	51
SD/	53
19:	55
.....	55
.....	55
.....	55
.....	55
Examples	55
.....	55
elapsedMillis	56
millis	56
elapsedMilliselapsedMicros	57
1	57
20:	60
.....	60
.....	60
Examples	60
.....	60
.....	60
21:	62
.....	62
.....	62
.....	62
Examples	62
.....	62
22:	64
Examples	64
Arduino Uno R3	64
23:	67
.....	67

.....	67
Examples.....	67
.....	67
24:	69
Examples.....	69
.....	69
.....	69
.....	69
25:	70
.....	70
.....	70
.....	70
Examples.....	70
.....	70
.....	70
26:	72
.....	72
Examples.....	72
.....	72
.....	73

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [arduino](#)

It is an unofficial and free arduino ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official arduino.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: arduino

Arduino

Arduino◦

Arduino

- ◦ ◦
-
-
-

1.0.0 201658

Examples

"Arduino◦ File > Examples > 01. Basics > Bare MinimumArduino IDEFile > Examples > 01. Basics > Bare Minimum◦

```
void setup() {  
  // put your setup code here, to run once  
}  
  
void loop() {  
  // put your main code here, to run repeatedly  
}
```

setup()◦ I/O◦ loop() Arduino◦ Arduino

```
void setup() {  
  // put your setup code here, to run once  
}  
  
void loop() {  
  // put your main code here, to run repeatedly  
}
```

Arduino◦ ◦ ◦

setup()loop()◦ File > Examples > 01. Basics > BlinkArduino IDE◦ ArduinoLED13LED◦

```
// the setup function runs once when you press reset or power the board  
void setup() {
```

```

// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}

```

1. `setup()` ◦ **Arduino** `setup()` ◦

1. 13◦

LED;◦

2. `loop()` ◦ `loop()` ◦

1. `digitalWrite(13, HIGH);` LED◦

2. `delay(1000);` 1000◦

3. `digitalWrite(13, LOW);` LED◦

4. `delay(1000);` 1000◦

`loop()` LED211◦ **Arduino Uno** LED13◦ LED◦

[Arduino IDE](#)



smartbox

DataPacket.cpp

DataPacket.h

EnCoPacket.cpp

EnCoPacket.h

InstrumentationPacket.cpp

Instrumen

```
1 #include "keys.h"
2 #include "device.h"
3 #include "LowPower.h"
4 #include "instrumentationParamEnum.h"
5 #include "sensor.h"
6 // #include "Sensor.h"
7
8 #include <SoftwareSerial.h>
9 #include <avr/wdt.h>
10 #include <avr/sleep.h>
11
12 // Console
13 #define SERIAL_BAUD 9600
14 #define debugSerial Serial
15
16 // Button to send msg on which pin ?? 2 OR 3
17 // Pin change interrupt possible on other pins if needed ....
18 #define BTN_SEND_PIN 2
19 // PIN 2 => IRQ0 // 3 => IRQ1
20 #define IRQ 0
21
22 #define PIN_TX_RN2483 8
23 #define PIN_RX_RN2483 9
24
25 // Arduino's
26 #if defined (__AVR_ATmega328P__)
27     // Serial setup to connect Modem
28     #define PIN_PWR_RN2483 12
```

"DataPacket.h" contains unrecognized characters. If this code was created with an older version of the IDE, it may not be able to parse the code.

"LoRaModem.h" contains unrecognized characters. If this code was created with an older version of the IDE, it may not be able to parse the code.

2: Arduino IDE

Examples

Windows

1. <https://www.arduino.cc/en/Main/Software>
2. "Windows Installer"
- 3.

Windows

WindowsArduino IDE

1. <https://www.arduino.cc/en/Main/Software>
2. "Windows ZIP"
- 3.
4. `Arduino.exe`

Fedora

1. `sudo dnf install arduino`
2. `Arduinoarduino`

Ubuntu

1. `sudo apt-get install arduino`
2. `Arduinoarduino`

macOS

1. <https://www.arduino.cc/en/Main/Software>
2. Mac OS X°
3. `.zip°`
4. `ArduinoApplications°`

Arduino IDE <https://riptutorial.com/zh-TW/arduino/topic/3790/arduino-ide>

3: I2C

I2CArduino。 - SDASCL。 Arduino。 I2CArduino。 。 VCC

Examples

。 。 。 24.3。 10。

```
#include <Wire.h>

#define BUFFER_SIZE 4
#define MAX_NUMBER_OF_SLAVES 24
#define FIRST_SLAVE_ADDRESS 1
#define READ_CYCLE_DELAY 1000

byte buffer[BUFFER_SIZE];

void setup()
{
    Serial.begin(9600);
    Serial.println("MASTER READER");
    Serial.println("*****");

    Wire.begin();          // Activate I2C link
}

void loop()
{
    for (int slaveAddress = FIRST_SLAVE_ADDRESS;
         slaveAddress <= MAX_NUMBER_OF_SLAVES;
         slaveAddress++)
    {
        Wire.requestFrom(slaveAddress, BUFFER_SIZE);    // request data from the slave
        if(Wire.available() == BUFFER_SIZE)
        { // if the available data size is same as I'm expecting
            // Reads the buffer the slave sent
            for (int i = 0; i < BUFFER_SIZE; i++)
            {
                buffer[i] = Wire.read(); // gets the data
            }

            // Parse the buffer
            // In order to convert the incoming bytes info short, I use union
            union short_tag {
                byte b[2];
                short val;
            } short_cast;

            // Parse the temperature
            short_cast.b[0] = buffer[0];
            short_cast.b[1] = buffer[1];
            float temperature = ((float)(short_cast.val)) / 10;

            // Parse the moisture
            short_cast.b[0] = buffer[2];
            short_cast.b[1] = buffer[3];
```

```

        short moisture = short_cast.val;

        // Prints the income data
        Serial.print("Slave address ");
        Serial.print(slaveAddress);
        Serial.print(": Temprature = ");
        Serial.print(temprature);
        Serial.print("; Moisture = ");
        Serial.println(moisture);
    }
}
Serial.println("*****");

delay(READ_CYCLE_DELAY);
}
}

```

```

#include <Wire.h>

#define BUFFER_SIZE 4
#define MAX_NUMBER_OF_SLAVES 24
#define FIRST_SLAVE_ADDRESS 1
#define READ_CYCLE_DELAY 1000

byte buffer[BUFFER_SIZE];

void setup()
{
    Serial.begin(9600);
    Serial.println("MASTER READER");
    Serial.println("*****");

    Wire.begin();          // Activate I2C link
}

void loop()
{
    for (int slaveAddress = FIRST_SLAVE_ADDRESS;
        slaveAddress <= MAX_NUMBER_OF_SLAVES;
        slaveAddress++)
    {
        Wire.requestFrom(slaveAddress, BUFFER_SIZE);    // request data from the slave
        if(Wire.available() == BUFFER_SIZE)
        {
            // if the available data size is same as I'm expecting
            // Reads the buffer the slave sent
            for (int i = 0; i < BUFFER_SIZE; i++)
            {
                buffer[i] = Wire.read();    // gets the data
            }

            // Parse the buffer
            // In order to convert the incoming bytes info short, I use union
            union short_tag {
                byte b[2];
                short val;
            } short_cast;

            // Parse the temperature
            short_cast.b[0] = buffer[0];
            short_cast.b[1] = buffer[1];

```

```

float temperature = ((float)(short_cast.val)) / 10;

// Parse the moisture
short_cast.b[0] = buffer[2];
short_cast.b[1] = buffer[3];
short moisture = short_cast.val;

// Prints the income data
Serial.print("Slave address ");
Serial.print(slaveAddress);
Serial.print(": Temprature = ");
Serial.print(temprature);
Serial.print("; Moisture = ");
Serial.println(moisture);
}
}
Serial.println("*****");

delay(READ_CYCLE_DELAY);
}
}

```

I2C <https://riptutorial.com/zh-TW/arduino/topic/9092/i2c>

4: MIDI

MIDI。

Examples

MIDI THRU

MIDI Thru。 MIDIArduinoMIDI INMIDI OUT。 。

```
// This is a simple MIDI THRU.  Everything in, goes right out.
// This has been validate on an Arduino UNO and a Olimex MIDI Shield

boolean byteReady;
unsigned char midiByte;

void setup() {
    // put your setup code here, to run once:
    // Set MIDI baud rate:
    Serial.begin(31250);
    byteReady = false;
    midiByte = 0;
}

// The Loop that always gets called...
void loop() {
    if (byteReady) {
        byteReady = false;
        Serial.write(midiByte);
    }
}

// The little function that gets called each time loop is called.
// This is automated somewhere in the Arduino code.
void serialEvent() {
    if (Serial.available()) {
        // get the new byte:
        midiByte = (unsigned char)Serial.read();
        byteReady = true;
    }
}
```

MIDI

```
// This is a more complex MIDI THRU.  This version uses a queue.  Queues are important because
some
// MIDI messages can be interrupted for real time events.  If you are generating your own
messages,
// you may need to stop your message to let a "real time" message through and then resume your
message.

#define QUEUE_DEPTH 128
```



```

// Queue Logic for storing messages
int headQ = 0;
int tailQ = 0;
unsigned char tx_queue[QUEUE_DEPTH];

void setup() {
    // put your setup code here, to run once:
    // Set MIDI baud rate:
    Serial.begin(31250);
}

// getQDepth checks for roll over.  Folks have told me this
// is not required.  Feel free to experiment.
int getQDepth() {
int depth = 0;
    if (headQ < tailQ) {
        depth = QUEUE_DEPTH - (tailQ - headQ);
    } else {
        depth = headQ - tailQ;
    }
    return depth;
}

void addQueue (unsigned char myByte) {
    int depth = 0;
    depth = getQDepth();

    if (depth < (QUEUE_DEPTH-2)) {
        tx_queue[headQ] = myByte;
        headQ++;
        headQ = headQ % QUEUE_DEPTH; // Always keep the headQ limited between 0 and 127
    }
}

unsigned char deQueue() {
    unsigned char myByte;
    myByte = tx_queue[tailQ];
    tailQ++;
    tailQ = tailQ % QUEUE_DEPTH; // Keep this tailQ contained within a limit
    // Now that we dequeued the byte, it must be sent.
    return myByte;
}

void loop() {
    if (getQDepth>0) {
        Serial.write(deQueue());
    }
}

// The little function that gets called each time loop is called.
// This is automated somewhere in the Arduino code.
void serialEvent() {
    if (Serial.available()) {
        // get the new byte:
        addQueue((unsigned char)Serial.read());
    }
}

```

MIDI

```

// This is a MiDI clk generator.  This takes a #defined BPM and
// makes the appropriate clk rate.  The queue is used to let other messages
// through, but allows a clock to go immediately to reduce clock jitter

#define QUEUE_DEPTH 128
#define BPM 121
#define MIDI_SYSRT_CLK 0xF8

// clock tracking and calculation
unsigned long lastClock;
unsigned long captClock;
unsigned long clk_period_us;

// Queue Logic for storing messages
int headQ = 0;
int tailQ = 0;
unsigned char tx_queue[QUEUE_DEPTH];

void setup() {
    // Set MIDI baud rate:
    Serial.begin(31250);
    clk_period_us = 60000000 / (24 * BPM);
    lastClock = micros();
}

// getQDepth checks for roll over.  Folks have told me this
// is not required.  Feel free to experiment.
int getQDepth() {
    int depth = 0;
    if (headQ < tailQ) {
        depth = QUEUE_DEPTH - (tailQ - headQ);
    } else {
        depth = headQ - tailQ;
    }
    return depth;
}

void addQueue (unsigned char myByte) {
    int depth = 0;
    depth = getQDepth();

    if (depth < (QUEUE_DEPTH-2)) {
        tx_queue[headQ] = myByte;
        headQ++;
        headQ = headQ % QUEUE_DEPTH; // Always keep the headQ limited between 0 and 127
    }
}

unsigned char deQueue() {
    unsigned char myByte;
    myByte = tx_queue[tailQ];
    tailQ++;
    tailQ = tailQ % QUEUE_DEPTH; // Keep this tailQ contained within a limit
    // Now that we dequeued the byte, it must be sent.
    return myByte;
}

void loop() {
    captClock = micros();

    if (lastClock > captClock) {

```

```

// we have a roll over condition - Again, maybe we don't need to do this.
if (clk_period_us <= (4294967295 - (lastClock - captClock))) {
    // Add a the ideal clock period for this BPM to the last measurement value
    lastClock = lastClock + clk_period_us;
    // Send a clock, bypassing the transmit queue
    Serial.write(MIDI_SYSRT_CLK);
}
} else if (clk_period_us <= captClock-lastClock) {
    // Basically the same two commands above, but not within a roll over check
    lastClock = lastClock + clk_period_us;
    // Send a clock, bypassing the transmit queue
    Serial.write(MIDI_SYSRT_CLK);
}

if (getQDepth>0) {
    Serial.write(deQueue());
}
}

// The little function that gets called each time loop is called.
// This is automated somewhere in the Arduino code.
void serialEvent() {
    if (Serial.available()) {
        // get the new byte:
        addQueue((unsigned char)Serial.read());;
    }
}
}

```

MIDI

MIDI[™]. 4

-
-
-
-

0x80. 0x7F. 127MIDI. MIDI.

... EXCEPT ...

D7..D0		
1000nnnn	0kkkkkkk 0vvvvvvv	◦ ◦ kkkkkkkk. vvvvvvv.
1001nnnn	0kkkkkkk 0vvvvvvv	◦ ◦ kkkkkkkk. vvvvvvv.
1010nnnn	0kkkkkkk 0vvvvvvv	◦ “” kkkkkkkk. vvvvvvv.
1011nnnn	0ccccccc 0vvvvvvv	◦ ◦ ◦ 120-127“”. ccccccc0-119. vvvvvvv0-127.
1100nnnn	0pppppppp	◦ ◦ ppppppp.
1101nnnn	0vvvvvvv	◦ “” ◦ ◦ vvvvvvv.

D7..D0		
1110nnnn	0lllllll 0mmmmmmm	◦ ◦ ◦ 2000H◦ RPN 0◦ llllll7◦ mmmmmmm7◦

D7..D0		
1011nnnn	0ccccccc 0vvvvvvv	◦ Control Change120-127◦
		◦ ◦ c = 120v = 0
		◦ “”◦ ◦
		c = 121v = x◦
		◦ MIDI◦ ◦ Local Control On◦
		c = 122v = 0
		c = 122v = 127
		◦ All Notes Off◦
		c = 123v = 0◦
		c = 124v = 0Omni Mode Off
		c = 125v = 0Omni Mode On
		c = 126v = MPoly OffMOmni Off0Omni On
		c = 127v = 0Poly

D7..D0		
11110000	0iiiiiii [0iiiiiii 0iiiiiii] 0ddddd --- - -- 0ddddd 11110111	◦ MIDI◦ IDMMMAAMEI10iiiiiii30iiiiiii 0iiiiiii 0iiiiiii ◦ 1ID◦ ID0ddddd◦ ◦ System Exclusive◦
11110001	0nnndddd	MIDI◦ nnn =ddd =
11110010	0lllllll 0mmmmmmm	◦ 14MIDI1= 6MIDI◦ lLSBMSB◦
11110011	0sssssss	◦ ◦
11110100		◦
11110101		◦
11110110		◦ ◦

D7..D0		
11110111		◦ System Exclusive◦

D7..D0		
11111000	◦ 24◦	
11111001	◦	
11111010	◦ ◦ ◦	
11111011	◦ ◦	
11111100	◦ ◦	
11111101	◦	
11111110	◦ ◦ ◦ 300msActive Sensing◦ ◦	
11111111	◦ ◦ ◦ ◦	

MIDI <https://riptutorial.com/zh-TW/arduino/topic/9406/midi>

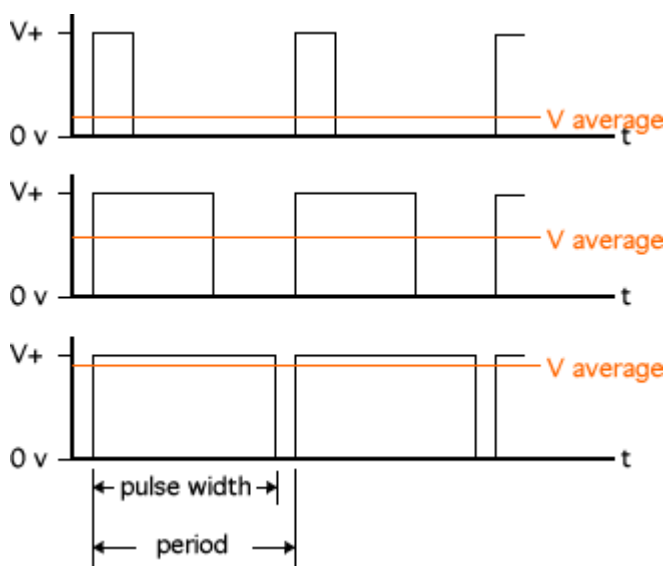
5: PWM -

Examples

PWM

Arduino `analogWrite()` Serial PWM

PWM 15V 0V



Arduino `analogWrite(pin, value)` pinvalue 00V 255V

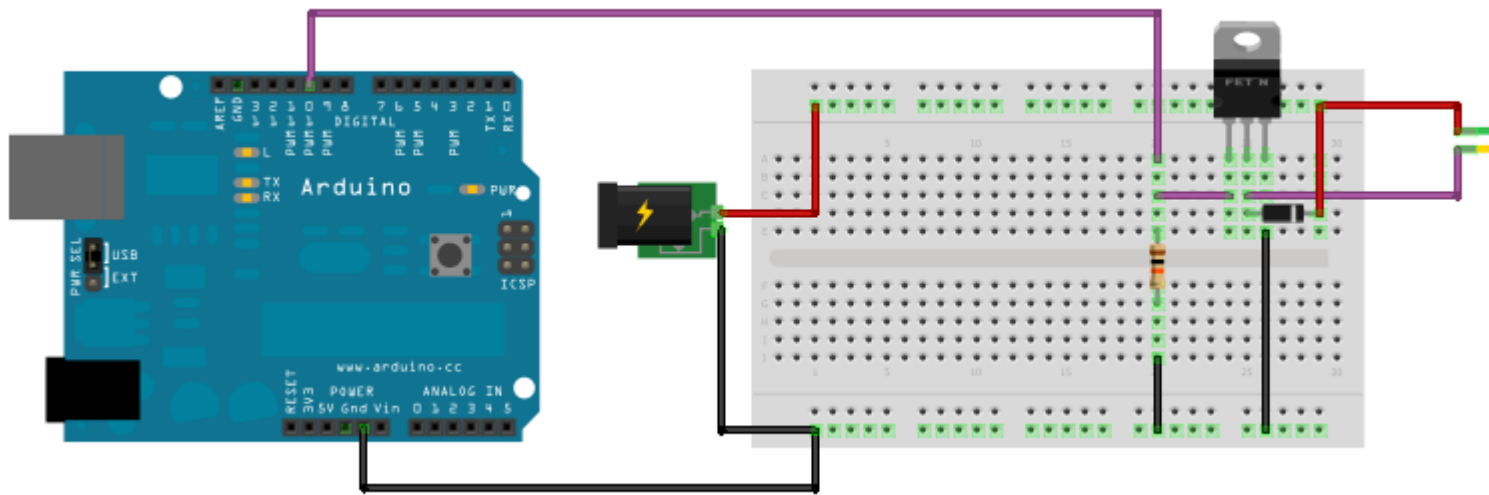
Arduino `ArduinoPWM`

MOSFET $<1\text{mA}$ Arduino PWM PWM MOSFET Arduino

- MOSFET BUZ11
- SB320
- 10K~1M
- 12V
-
-
-
- Arduino

MOSFET MOSFET MOSFET MOSFET PWM10 MOSFET

◦ ◦



Arduino。 0255。 。。

```
int in = 0; // Variable to store the desired value
byte pinOut = 10; // PWM output pin

void setup() { // This executes once
  Serial.begin(9600); // Initialize serial port
  pinMode(pinOut, OUTPUT); // Prepare output pin
}

void loop() { // This loops continuously
  if(Serial.available()){ // Check if there's data
    in = Serial.read(); // Read said data into the variable "in"
    analogWrite(pinOut, in); // Pass the value of "in" to the pin
  }
}
```

ArduinoPWM。

PWMTLC5940

ArduinoPWM [TLC5940](http://playground.arduino.cc/Learning/TLC5940)。 16120-4095。 <http://playground.arduino.cc/Learning/TLC5940>。 RGB LED。 LED。 PWM。

```
// Include the library
#include <Tlc5940.h>

void setup() {
  // Initialize
  Tlc.init();
  Tlc.clear();
}

unsigned int level = 0;
void loop() {
  // Set all 16 outputs to same value
  for (int i = 0; i < 16; i++) {
```

```
    Tlc.set(i, level);  
  }  
  level = (level + 1) % 4096;  
  // Tell the library to send the values to the chip  
  Tlc.update();  
  delay(10);  
}
```

PWM - <https://riptutorial.com/zh-TW/arduino/topic/1658/pwm---->

6: PythonArduino Uno

- `Serial.begin(baudrate)` // Set baud rate (bits per second) for serial data transmission
- `Serial.println(value)` // Print data to serial port followed by Carriage Return `\r` and Newline character `\n`
- `serial.Serial((port=None, baudrate=9600, bytesize=EIGHTBITS, parity=PARITY_NONE, stopbits=STOPBITS_ONE, timeout=None, xonxoff=False, rtscts=False, write_timeout=None, dsrdtr=False, inter_byte_timeout=None))` // Initialize serial port with all parameters
- `serial.readline()` // Read serial data which contains Carriage Return `\r` and Newline character `\n`



Arduino UnoArduino IDE 1.6.9Windows 10Python 2.7.12。

Examples

ArduinoPython

Arduino。

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println("Hello World!");
  delay(100);
}
```

`setup() Serial.begin(9600)。` 9600。 [Arduino Serial.begin](#)

`loop() “Hello World”。` `Serial.println(“Hello World!”)`ASCII。 `(\n)` (CR, `\r`) `(\n)`。 100。

COMArduinoCOMPython。

ArduinoPython

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println("Hello World!");
}
```

```
delay(100);  
}
```

pyserial。 Windowspyserial [Pythonpyserial](#) 。 COM。 Arduino。

`readline()` `while`。 100Arduino。 pyserial `readline()` pyserial documentation [PySerial ReadLine](#) 。

[PythonArduino Uno](#) <https://riptutorial.com/zh-TW/arduino/topic/6722/pythonarduino-uno>

7: SPI

◦

```
#define CSPIN 1 // or whatever else your CS pin is
// init:
pinMode(CSPIN, OUTPUT);
digitalWrite(CSPIN, 1); // deselect

// use:
digitalWrite(CSPIN, 0); // select
... perform data transfer ...
digitalWrite(CSPIN, 1); // deselect
```

MISO◦ MISO◦ MISO◦

- SPI
- SPI

SPI◦ SPI◦ SPI◦ SPI.beginTransactionSPISettingsSPI

```
#define CSPIN 1 // or whatever else your CS pin is
// init:
pinMode(CSPIN, OUTPUT);
digitalWrite(CSPIN, 1); // deselect

// use:
digitalWrite(CSPIN, 0); // select
... perform data transfer ...
digitalWrite(CSPIN, 1); // deselect
```

SPISettings

```
#define CSPIN 1 // or whatever else your CS pin is
// init:
pinMode(CSPIN, OUTPUT);
digitalWrite(CSPIN, 1); // deselect

// use:
digitalWrite(CSPIN, 0); // select
... perform data transfer ...
digitalWrite(CSPIN, 1); // deselect
```

beginTransaction()endTransaction()SPI - ◦

[ArduinoSPISettings](#)

SPI

```
ISR SPI_0_SPI using Interrupt (interrupt_number) {
    beginTransaction();
    beginTransaction();
    endTransaction();
}
```

Arduino SPI using Interrupt

Examples

SPI1

```
#include <SPI.h>
#define CSPIN 1

void setup() {
    pinMode(CSPIN, OUTPUT); // init chip select pin as an output
    digitalWrite(CSPIN, 1); // most slaves interpret a high level on CS as "deasserted"

    SPI.begin();

    SPI.beginTransaction(SPI_0_Settings(1000000, MSBFIRST, SPI_MODE0));
    digitalWrite(CSPIN, 0);

    unsigned char sent = 0x01;
    unsigned char received = SPI.transfer(sent);
    // more data could be transferred here

    digitalWrite(CSPIN, 1);
    SPI.endTransaction();

    SPI.end();
}

void loop() {
    // we don't need loop code in this example.
}
```

-
- SPI
- SPI_0_0

SPI <https://riptide.com/zh-TW/arduino/topic/4919/spi>

8:

- `digitalPinToInterrupt; //IDIDattachInterrupt()detachInterrupt() °`
- `attachInterruptdigitalPinToInterruptpinISRmode; //`
- `attachInterruptISR; //`
- `detachInterruptdigitalPinToInterrupt;`
- `detachInterrupt;`
- `noInterrupts; //`
- `; //noInterrupts() °`

	ID ° °
ISR	° °
	° LOWCHANGERISINGFALLING ° HIGH °

ISR ° ISR °

ISR`delay()millis() °`

Examples

2GND2 °

```
const int LED_PIN = 13;
const int INTERRUPT_PIN = 2;
volatile bool ledState = LOW;

void setup() {
    pinMode(LED_PIN, OUTPUT);
    pinMode(INTERRUPT_PIN, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), myISR, FALLING); // trigger when
    button pressed, but not when released.
}

void loop() {
    digitalWrite(LED_PIN, ledState);
}

void myISR() {
    ledState = !ledState;
    // note: LOW == false == 0, HIGH == true == 1, so inverting the boolean is the same as
    switching between LOW and HIGH.
```

```
}
```

◦ ◦ LED◦

<https://riptutorial.com/zh-TW/arduino/topic/2913/>

9:

- `Serial.begin(speed) //`
- `Serial.begin(speed, config)`
- `Serial[1-3].begin(speed) // Arduino Mega1-313。`
- `Serial[1-3].begin(speed, config) // Arduino Mega1-313。`
- `Serial.peek() //`
- `Serial.available() //`
- `Serial.print(text) //`
- `Serial.println(text) //Serial.print()`

9600	
5 - 88	

Arduino Mega。

```
Serial.begin(9600);  
Serial1.begin(38400);  
Serial2.begin(19200);  
Serial3.begin(4800);
```

Arduino。 config。

81 - SERIAL_8E1

62 - SERIAL_6O2

71 - SERIAL_7N1

Examples

。

```
byte incomingBytes;  
  
void setup() {  
  Serial.begin(9600); // Opens serial port, sets data rate to 9600 bps.  
}  
  
void loop() {  
  // Send data only when you receive data.  
  if (Serial.available() > 0) {  
    // Read the incoming bytes.  
    incomingBytes = Serial.read();  
  }  
}
```

```

    // Echo the data.
    Serial.println(incomingBytes);
  }
}

```

Base64

```

String base64="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";

void setup() {

    Serial.begin(9600); // Turn the serial protocol ON
    Serial.println("Start Typing");
}

void loop() {

    if (Serial.available() > 0) { // Check if data has been sent from the user
        char c = Serial.read();    // Gets one byte/Character from serial buffer
        int result = base64.indexOf(c); // Base64 filtering
        if (result >= 0)
            Serial.print(c); // Only print Base64 string
    }
}

```

```

byte incoming;
String inBuffer;

void setup() {
    Serial.begin(9600); // or whatever baud rate you would like
}

void loop(){
    // setup as non-blocking code
    if(Serial.available() > 0) {
        incoming = Serial.read();

        if(incoming == '\n') { // newline, carriage return, both, or custom character

            // handle the incoming command
            handle_command();

            // Clear the string for the next command
            inBuffer = "";
        } else{
            // add the character to the buffer
            inBuffer += incoming;
        }
    }

    // since code is non-blocking, execute something else . . . .
}

void handle_command() {
    // expect something like 'pin 3 high'
    String command = inBuffer.substring(0, inBuffer.indexOf(' '));
    String parameters = inBuffer.substring(inBuffer.indexOf(' ') + 1);
}

```



```

if(command.equalsIgnoreCase('pin')){
    // parse the rest of the information
    int pin = parameters.substring("0, parameters.indexOf(' ')).toInt();
    String state = parameters.substring(parameters.indexOf(' ') + 1);

    if(state.equalsIgnoreCase('high')){
        digitalWrite(pin, HIGH);
    }else if(state.equalsIgnoreCase('low')){
        digitalWrite(pin, LOW);
    }else{
        Serial.println("did not compute");
    }
} // add code for more commands
}

```

Python

ArduinoRaspberry PiArduinoPC

Arduino

```

void setup() {
    // Opens serial port, sets data rate to 9600 bps:
    Serial.begin(9600);
}

void loop() {
    // Sends a line over serial:
    Serial.println("Hello, Python!");
    delay(1000);
}

```

```

void setup() {
    // Opens serial port, sets data rate to 9600 bps:
    Serial.begin(9600);
}

void loop() {
    // Sends a line over serial:
    Serial.println("Hello, Python!");
    delay(1000);
}

```

<https://riptutorial.com/zh-TW/arduino/topic/1674/>

10:

◦ ◦ ◦ Arduino Servo.h Servo.h ◦

- #include <Servo.h> //
- Servo.attach//◦ Servo
- Servo.write//0 - 180
- Servo.read//

Examples

```
#include <Servo.h>

Servo srv;

void setup() {
  srv.attach(9); // Attach to the servo on pin 9
}
```

attach()◦ PWM◦ Arduino Mega Servo 910 analogWrite PWM◦

```
#include <Servo.h>

Servo srv;

void setup() {
  srv.attach(9); // Attach to the servo on pin 9
}
```

◦

<https://riptutorial.com/zh-TW/arduino/topic/4920/>

11:

C / C ++Arduino IDE。

.cpp。

.inoArduino IDE。

Arduino - -

Examples

```
int squareNum (int a) {  
    return a*a;  
}
```

int

squareNum

int a

return a*a

Anatomy of a C function

Datatype of data returned,
any C datatype.

"void" if nothing is returned.

Parameters passed to
function, any C datatype.

Function name

int myMultiplyFunction(int x, int y){

int result;

result = x * y;

return result;

}

Return statement,
datatype matches
declaration.

Curly braces required.

。

```
void setup(){  
    Serial.begin(9600);  
}  
  
void loop() {  
    int i = 2;  
  
    int k = squareNum(i); // k now contains 4  
    Serial.println(k);  
}
```

```
    delay(500);  
}  
  
int squareNum(int a) {  
    return a*a;  
}
```

<https://riptutorial.com/zh-TW/arduino/topic/2380/>

12:

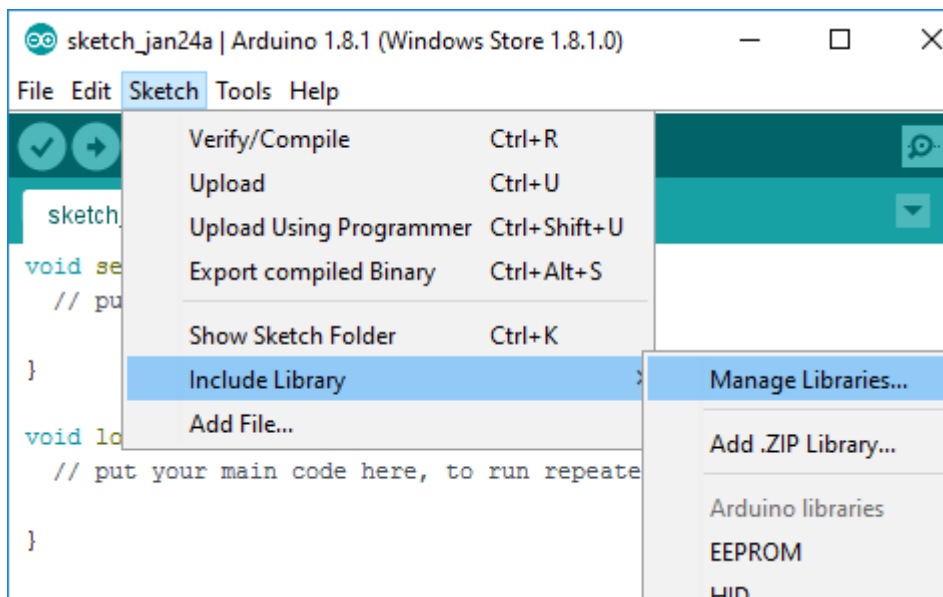
- Arduino IDE

-

Examples

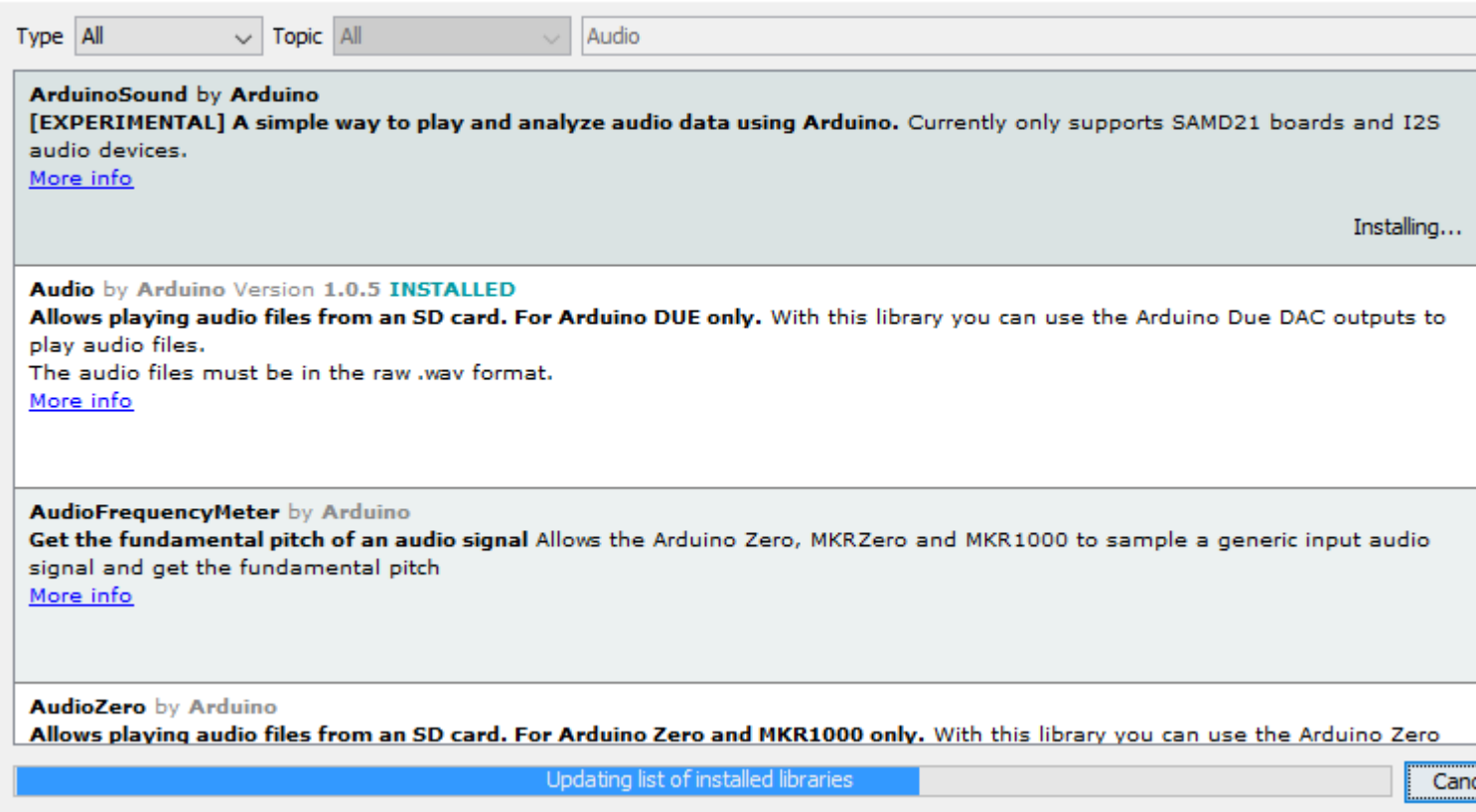
Arduino IDE

- “”>“”>“”。

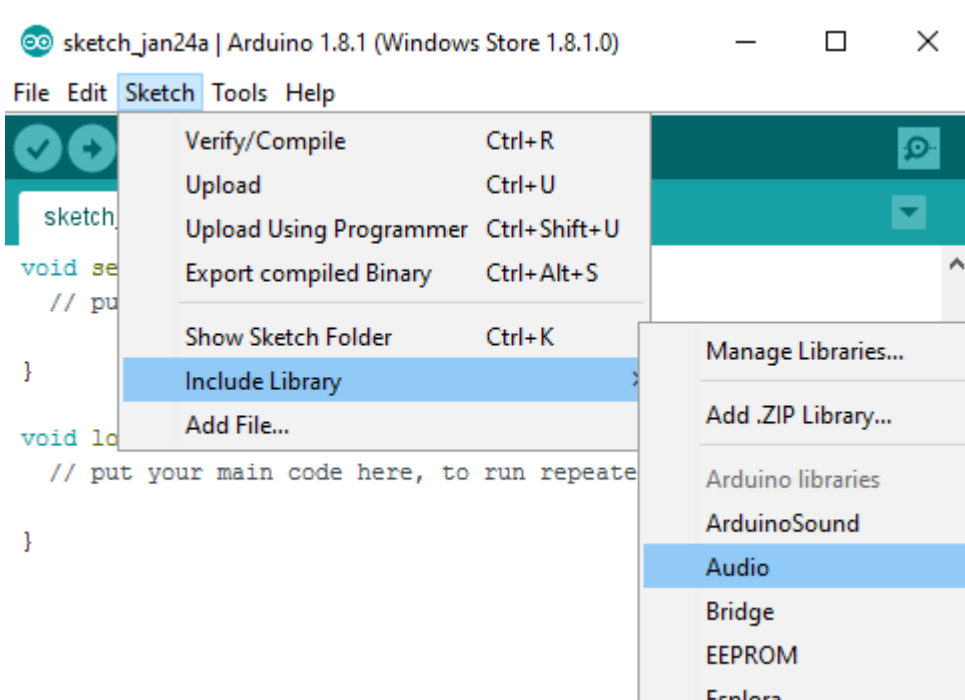


。

- “”。



-



-

IDE。



。

<https://riptutorial.com/zh-TW/arduino/topic/8896/>

13: EEPROM

- EEPROM.write; //EEPROM
- EEPROM.read; //EEPROM

EEPROM.write	
EEPROM	
	EEPROM.put(0, 0);
EEPROM	

- ATmega328P
- ATmega168P
- ATmega1280P
- ATmega2560P

Examples

EEPROM

<EEPROM.h>

```
#include <EEPROM.h>
```

```
#include <EEPROM.h>
```

EEPROM <https://riptutorial.com/zh-TW/arduino/topic/5987/EEPROM>

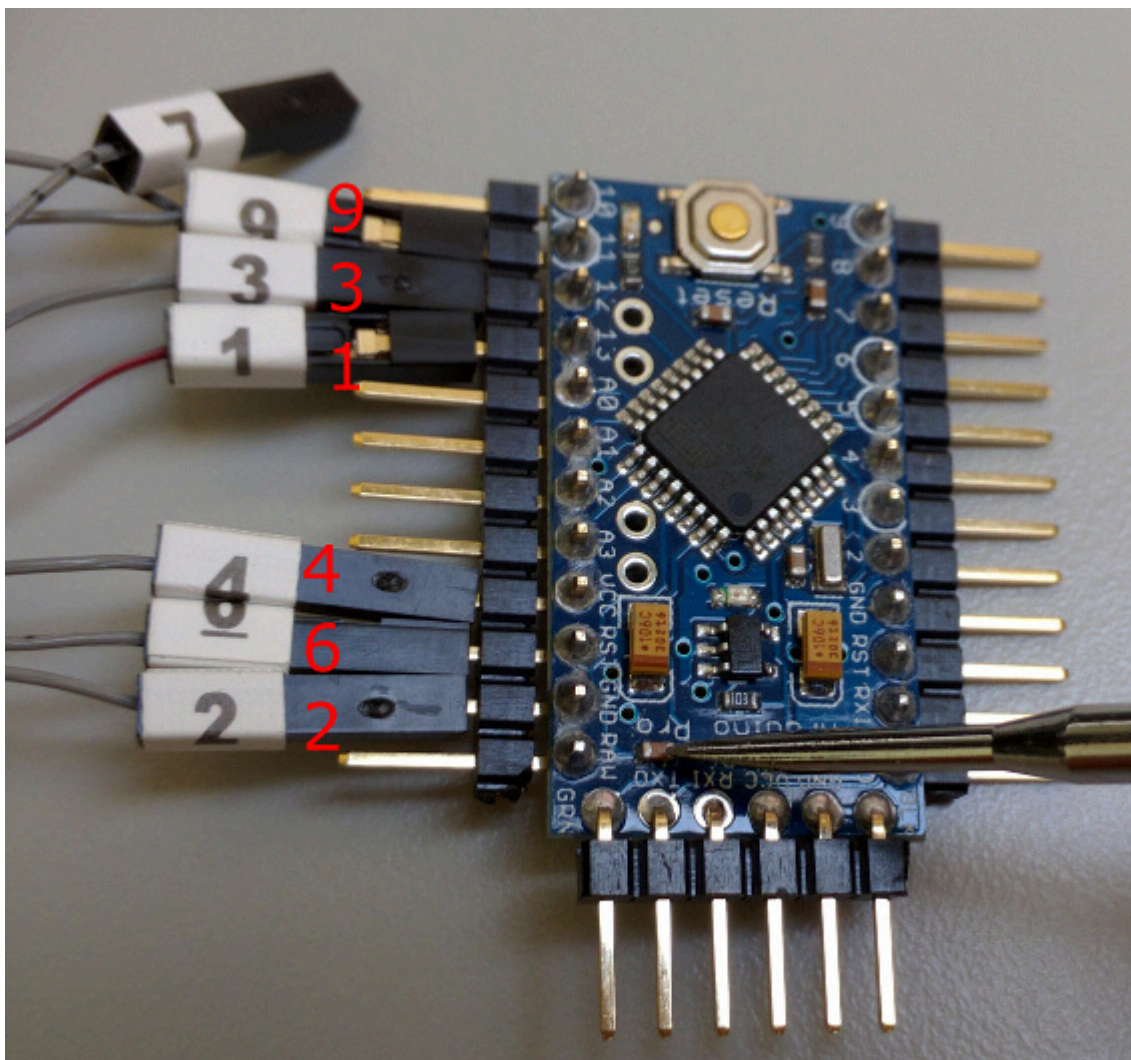
14: ArduinoAtmel Studio 7

- Atmel Studio 7。
- 。
- ISPAtmel Studio。
- [Atmel ICE](#) AVRarduinopro miniARMArduinosZeroDue。
- 。

-
- Uno
 - [6ICSP](#)。
 - Uno。
 - AVR。



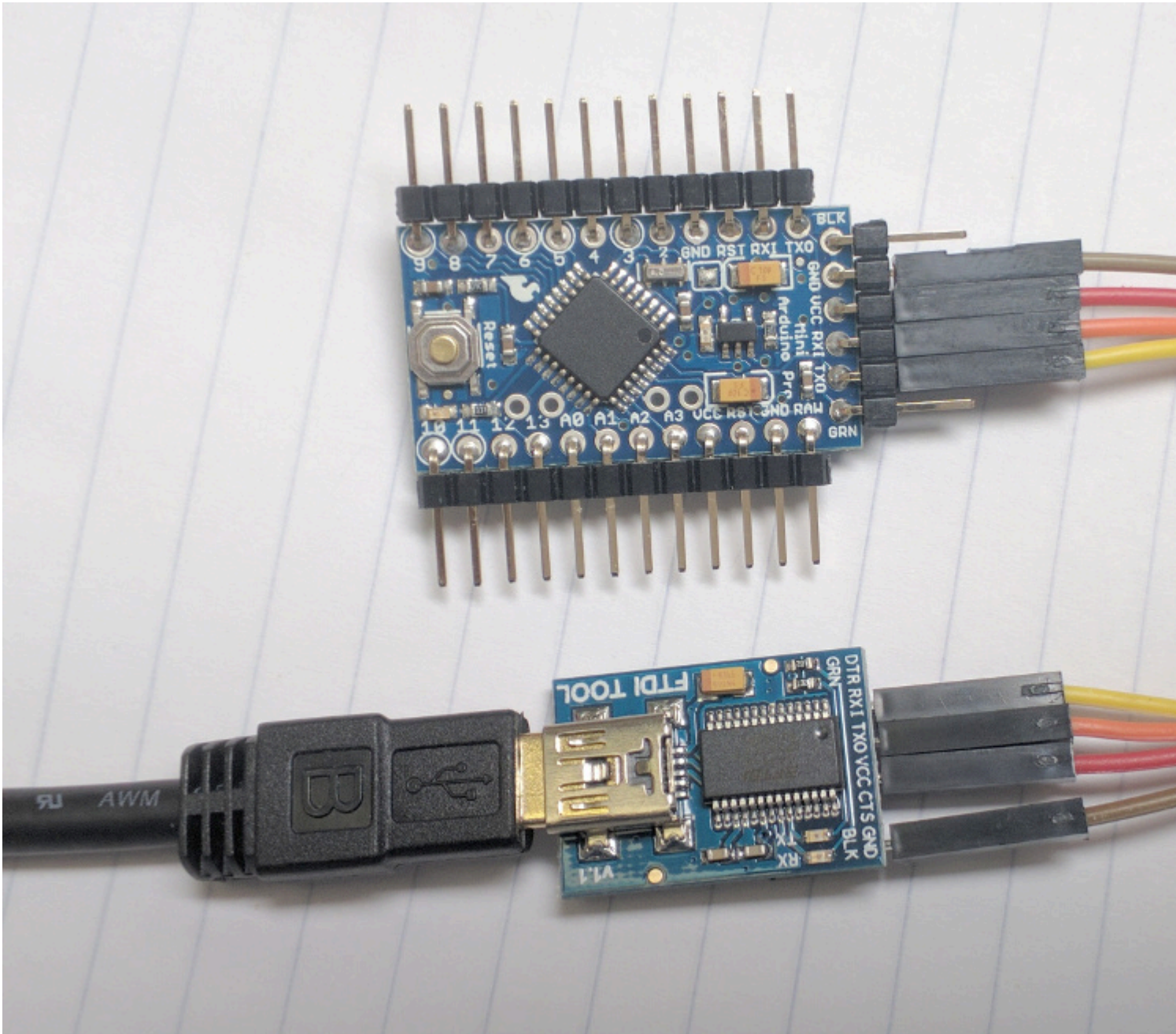
Arduino Pro Mini AVR。



UnoArduino IDE



Pro MiniFTDIDTRAtmelSWD。 TxRx。 ArduinoRxTxFTDITxRx。 FTDIRxTx。



Arduino。FTDIPro MiniUnoUSBAC。

Atmel ICEAtmel StudioArduino。

Atmel Studio - > - > “Arduino sketch”。

Project - > yourProjectName PropertiesTooldebugger / programmerAtmel ICEdebugWire。Debug
- > Start debugging and break。DWEN。Arduino。ArduinoArduino IDEDebug - > disable
debugWIRE。

。Atmel StudioAtmel Studio。

Arduino 1.6.6C ++ 11。C ++Arduino。Atmel Studio 7C ++ 11ToolChainAVR / GNU C ++-
std=c++11。

.cppC:\Users\YourUserName\Documents\Atmel

Studio\7.0\YourSolutionName\YourProjectName\ArduinoCore\src\core Atmel StudioSolution Explorer
Arduino Core / src / - > .hYourProjectName / Dependencies

Android IDEAtmel Studio - >Atmel Studio. - >.

Atmel Studio 7IDEArduinoArduino IDE/.

- > - > - >.

- Edit.CommentSelectionEdit.UncommentSelection
- View.NavigateForwardView.NavigateBackward
- Edit.MoveSelectedLinesUpEdit.MoveSelectedLinesDown
- Edit.GoToDefinition

Examples

Atmel Studio 7

ArduinoAtmel Studio. Atmel Studio. Arduino. ArduinoCoresrc - > coremain.cpp. Arduinofor
Arduino.

```
/* Begining of Auto generated code by Atmel studio */
#include <Arduino.h>
/* End of auto generated code by Atmel studio */

// Beginning of Auto generated function prototypes by Atmel Studio
void printA();
// End of Auto generated function prototypes by Atmel Studio

void setup() {
    Serial.begin(9600);
}

void loop() {
    printA();
}

void printA() {
    Serial.println("A");
}
```

ArduinoAtmel Studio 7 <https://riptutorial.com/zh-TW/arduino/topic/2567/arduinoatmel-studio-7>

15:

- for{}
- {}
-

General Remark◦ **setuploop**◦ ◦

do { } while(condition)◦ ◦

Examples

whiletrue◦ true while◦

100num1

```
int num = 0;
while (num < 100) {
    // do something
    num++;
}
```

for

```
int num = 0;
while (num < 100) {
    // do something
    num++;
}
```

```
int num = 0;
while (num < 100) {
    // do something
    num++;
}
```

for

```
int num = 0;
while (num < 100) {
    // do something
    num++;
}
```

forwhile◦

for100◦

```
for (int i = 0; i < 100; i++) {
    // do something
}
```

while

```
for (int i = 0; i < 100; i++) {
    // do something
}
```

◦

```
for (int i = 0; i < 100; i++) {
    // do something
}
```

while

```
for (int i = 0; i < 100; i++) {
    // do something
}
```

do whilewhile◦

100◦

```
int i = 0;
do {
    i++;
} while (i < 100);
```

1◦

```
int i = 0;
do {
    i++;
} while (i < 100);
```

while0false◦ do while◦

◦

break;◦

continue;◦

break101i = 0,1...1001000


```
for (int i = 0; i < 1000; i++) {  
    // execute this repeatedly with i = 0, 1, 2, ...  
    if (i >= 100) {  
        break;  
    }  
}
```

continuej50100

```
for (int i = 0; i < 1000; i++) {  
    // execute this repeatedly with i = 0, 1, 2, ...  
    if (i >= 100) {  
        break;  
    }  
}
```

<https://riptutorial.com/zh-TW/arduino/topic/2802/>

16:

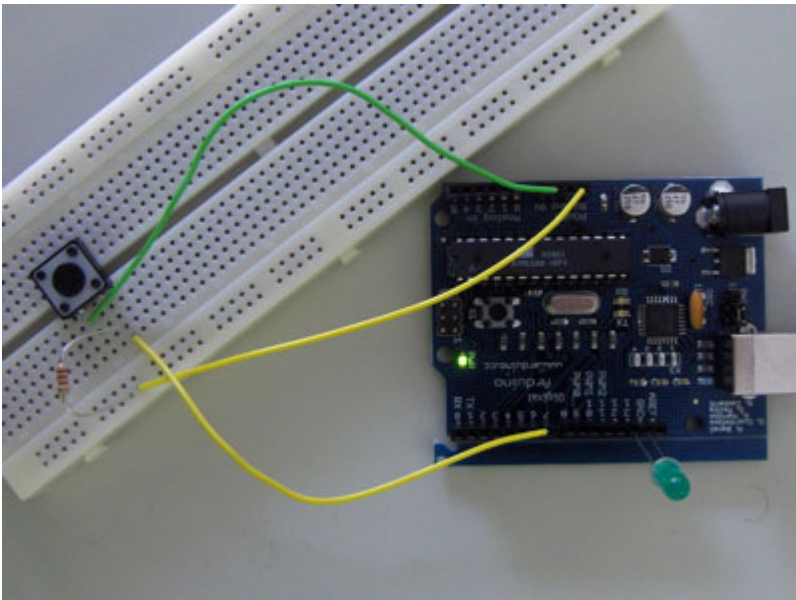
- `pinMode(pin, pinMode) //`
- `digitalRead(pin); //`

pinmode	INPUTINPUT_PULLUP
---------	-------------------

◦ 10◦ ◦

Examples

/LED◦



```
/* Basic Digital Read
 * -----
 *
 * turns on and off a light emitting diode(LED) connected to digital
 * pin 13, when pressing a pushbutton attached to pin 7. It illustrates the
 * concept of Active-Low, which consists in connecting buttons using a
 * 1K to 10K pull-up resistor.
 *
 * Created 1 December 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 */

int ledPin = 13; // choose the pin for the LED
int inPin = 7;   // choose the input pin (for a pushbutton)
int val = 0;     // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
```

```
pinMode(inPin, INPUT);    // declare pushbutton as input
}

void loop(){
  val = digitalRead(inPin); // read input value
  if (val == HIGH) {        // check if the input is HIGH (button released)
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

[Arduino.cc](https://www.arduino.cc)。

<https://riptutorial.com/zh-TW/arduino/topic/1662/>

17:

- `digitalWrite(pin, value)`

Examples

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

void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

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

[Arduino.cc](https://www.arduino.cc)

<https://riptutorial.com/zh-TW/arduino/topic/2477/>

Examples

cardInfo

```
/*
  SD card test

  This example shows how use the utility libraries on which the '
  SD library is based in order to get info about your SD card.
  Very useful for testing a card when you're not sure whether its working or not.

  The circuit:
  * SD card attached to SPI bus as follows:
  ** MOSI - pin 11 on Arduino Uno/Duemilanove/Diecimila
  ** MISO - pin 12 on Arduino Uno/Duemilanove/Diecimila
  ** CLK - pin 13 on Arduino Uno/Duemilanove/Diecimila
  ** CS - depends on your SD card shield or module.
            Pin 4 used here for consistency with other Arduino examples

  created  28 Mar 2011
  by Limor Fried
  modified 9 Apr 2012
  by Tom Igoe
  */

// include the SD library:
#include <SPI.h>
#include <SD.h>

// set up variables using the SD utility library functions:
Sd2Card card;
SdVolume volume;
SdFile root;

// change this to match your SD shield or module;
// Arduino Ethernet shield: pin 4
// Adafruit SD shields and modules: pin 10
// Sparkfun SD shield: pin 8
const int chipSelect = 4;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.print("\nInitializing SD card...");

  // we'll use the initialization code from the utility libraries
  // since we're just testing if the card is working!
  if (!card.init(SPI_HALF_SPEED, chipSelect)) {
    Serial.println("initialization failed. Things to check:");
    Serial.println("* is a card inserted?");
  }
}
```

```

    Serial.println("* is your wiring correct?");
    Serial.println("* did you change the chipSelect pin to match your shield or module?");
    return;
} else {
    Serial.println("Wiring is correct and a card is present.");
}

// print the type of card
Serial.print("\nCard type: ");
switch (card.type()) {
    case SD_CARD_TYPE_SD1:
        Serial.println("SD1");
        break;
    case SD_CARD_TYPE_SD2:
        Serial.println("SD2");
        break;
    case SD_CARD_TYPE_SDHC:
        Serial.println("SDHC");
        break;
    default:
        Serial.println("Unknown");
}

// Now we will try to open the 'volume'/'partition' - it should be FAT16 or FAT32
if (!volume.init(card)) {
    Serial.println("Could not find FAT16/FAT32 partition.\nMake sure you've formatted the
card");
    return;
}

// print the type and size of the first FAT-type volume
uint32_t volumesize;
Serial.print("\nVolume type is FAT");
Serial.println(volume.fatType(), DEC);
Serial.println();

volumesize = volume.blocksPerCluster();    // clusters are collections of blocks
volumesize *= volume.clusterCount();       // we'll have a lot of clusters
volumesize *= 512;                         // SD card blocks are always 512 bytes
Serial.print("Volume size (bytes): ");
Serial.println(volumesize);
Serial.print("Volume size (Kbytes): ");
volumesize /= 1024;
Serial.println(volumesize);
Serial.print("Volume size (Mbytes): ");
volumesize /= 1024;
Serial.println(volumesize);

Serial.println("\nFiles found on the card (name, date and size in bytes): ");
root.openRoot(volume);

// list all files in the card with date and size
root.ls(LS_R | LS_DATE | LS_SIZE);
}

void loop(void) {
}

```

SD

```
/*
  SD card datalogger

  This example shows how to log data from three analog sensors
  to an SD card using the SD library.

  The circuit:
  * analog sensors on analog ins 0, 1, and 2
  * SD card attached to SPI bus as follows:
  ** MOSI - pin 11
  ** MISO - pin 12
  ** CLK - pin 13
  ** CS - pin 4

  created  24 Nov 2010
  modified 9 Apr 2012
  by Tom Igoe

  This example code is in the public domain.

  */

#include <SPI.h>
#include <SD.h>

const int chipSelect = 4;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.print("Initializing SD card...");

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("card initialized.");
}

void loop()
{
  // make a string for assembling the data to log:
  String dataString = "";

  // read three sensors and append to the string:
  for (int analogPin = 0; analogPin < 3; analogPin++) {
    int sensor = analogRead(analogPin);
    dataString += String(sensor);
    if (analogPin < 2) {
      dataString += ",";
    }
  }
}
```

```

}

// open the file. note that only one file can be open at a time,
// so you have to close this one before opening another.
File dataFile = SD.open("datalog.txt", FILE_WRITE);

// if the file is available, write to it:
if (dataFile) {
  dataFile.println(dataString);
  dataFile.close();
  // print to the serial port too:
  Serial.println(dataString);
}
// if the file isn't open, pop up an error:
else {
  Serial.println("error opening datalog.txt");
}
}

```

SD

```

/*
  SD card file dump

  This example shows how to read a file from the SD card using the
  SD library and send it over the serial port.

  The circuit:
  * SD card attached to SPI bus as follows:
  ** MOSI - pin 11
  ** MISO - pin 12
  ** CLK - pin 13
  ** CS - pin 4

  created 22 December 2010
  by Limor Fried
  modified 9 Apr 2012
  by Tom Igoe

  This example code is in the public domain.

  */

#include <SPI.h>
#include <SD.h>

const int chipSelect = 4;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.print("Initializing SD card...");

```



```

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
}
Serial.println("card initialized.");

// open the file. note that only one file can be open at a time,
// so you have to close this one before opening another.
File dataFile = SD.open("datalog.txt");

// if the file is available, write to it:
if (dataFile) {
    while (dataFile.available()) {
        Serial.write(dataFile.read());
    }
    dataFile.close();
}
// if the file isn't open, pop up an error:
else {
    Serial.println("error opening datalog.txt");
}
}

void loop()
{
}

```

SD

```

/*
  SD card basic file example

  This example shows how to create and destroy an SD card file
  The circuit:
  * SD card attached to SPI bus as follows:
  ** MOSI - pin 11
  ** MISO - pin 12
  ** CLK - pin 13
  ** CS - pin 4

  created   Nov 2010
  by David A. Mellis
  modified 9 Apr 2012
  by Tom Igoe

  This example code is in the public domain.

  */
#include <SPI.h>
#include <SD.h>

File myFile;

void setup()
{
    // Open serial communications and wait for port to open:

```

```

Serial.begin(9600);
while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
}

Serial.print("Initializing SD card...");

if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    return;
}
Serial.println("initialization done.");

if (SD.exists("example.txt")) {
    Serial.println("example.txt exists.");
}
else {
    Serial.println("example.txt doesn't exist.");
}

// open a new file and immediately close it:
Serial.println("Creating example.txt...");
myFile = SD.open("example.txt", FILE_WRITE);
myFile.close();

// Check to see if the file exists:
if (SD.exists("example.txt")) {
    Serial.println("example.txt exists.");
}
else {
    Serial.println("example.txt doesn't exist.");
}

// delete the file:
Serial.println("Removing example.txt...");
SD.remove("example.txt");

if (SD.exists("example.txt")) {
    Serial.println("example.txt exists.");
}
else {
    Serial.println("example.txt doesn't exist.");
}
}

void loop()
{
    // nothing happens after setup finishes.
}

```

Listfiles

```

/*
Listfiles

This example shows how print out the files in a
directory on a SD card

The circuit:

```

```

* SD card attached to SPI bus as follows:
** MOSI - pin 11
** MISO - pin 12
** CLK - pin 13
** CS - pin 4

created    Nov 2010
by David A. Mellis
modified 9 Apr 2012
by Tom Igoe
modified 2 Feb 2014
by Scott Fitzgerald

This example code is in the public domain.
*/

#include <SPI.h>
#include <SD.h>

File root;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.print("Initializing SD card...");

  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  root = SD.open("/");

  printDirectory(root, 0);

  Serial.println("done!");
}

void loop()
{
  // nothing happens after setup finishes.
}

void printDirectory(File dir, int numTabs) {
  while(true) {

    File entry = dir.openNextFile();
    if (! entry) {
      // no more files
      break;
    }
    for (uint8_t i=0; i<numTabs; i++) {
      Serial.print('\t');
    }
    Serial.print(entry.name());
  }
}

```

```

    if (entry.isDirectory()) {
        Serial.println("/");
        printDirectory(entry, numTabs+1);
    } else {
        // files have sizes, directories do not
        Serial.print("\t\t");
        Serial.println(entry.size(), DEC);
    }
    entry.close();
}
}

```

SD/

```

/*
  SD card read/write

  This example shows how to read and write data to and from an SD card file
  The circuit:
  * SD card attached to SPI bus as follows:
  ** MOSI - pin 11
  ** MISO - pin 12
  ** CLK - pin 13
  ** CS - pin 4

  created   Nov 2010
  by David A. Mellis
  modified 9 Apr 2012
  by Tom Igoe

  This example code is in the public domain.

  */

#include <SPI.h>
#include <SD.h>

File myFile;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.print("Initializing SD card...");

  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  myFile = SD.open("test.txt", FILE_WRITE);

```

```

// if the file opened okay, write to it:
if (myFile) {
  Serial.print("Writing to test.txt...");
  myFile.println("testing 1, 2, 3.");
  // close the file:
  myFile.close();
  Serial.println("done.");
} else {
  // if the file didn't open, print an error:
  Serial.println("error opening test.txt");
}

// re-open the file for reading:
myFile = SD.open("test.txt");
if (myFile) {
  Serial.println("test.txt:");

  // read from the file until there's nothing else in it:
  while (myFile.available()) {
    Serial.write(myFile.read());
  }
  // close the file:
  myFile.close();
} else {
  // if the file didn't open, print an error:
  Serial.println("error opening test.txt");
}
}

void loop()
{
  // nothing happens after setup
}

```

<https://riptutorial.com/zh-TW/arduino/topic/6584/>

19:

- unsigned long millis
- unsigned long micros
- void delay(unsigned long milliseconds)
- void delayMicroseconds(unsigned long microseconds)
- `elapsedMillis` ◦
 - `elapsedMillis elapsedMillisObject;`
 - `elapsedMillisObject = 0; ""`
 - `unsigned long deltaT = elapsedMillisObject;`
 - `elapsedMillisObject += - =`

`delay()` `delayMicroseconds()` ◦ ◦

- `CPULEDCPU` ◦
- `delay()` ◦
- `100delay(500)` ◦

`millis()` `1 kHz` ◦ `millis()` ◦ `""` ◦ `millis()` ◦ `micros()` ◦

`elapsedMillismillis()` ◦ `elapsedMicroselapsedMicros` ◦

Examples

LED

```
// set constants for blinking the built-in LED at 1 Hz
#define OUTPIN LED_BUILTIN
#define PERIOD 500

void setup()
{
    pinMode(OUTPIN, OUTPUT);    // sets the digital pin as output
}

void loop()
{
    digitalWrite(OUTPIN, HIGH); // sets the pin on
    delayMicroseconds(PERIOD);  // pauses for 500 milliseconds
    digitalWrite(OUTPIN, LOW);  // sets the pin off
    delayMicroseconds(PERIOD);  // pauses for 500 milliseconds

    // doing other time-consuming stuff here will skew the blinking
}
```

CPU。 millis()elapsedMillis - 。

elapsedMillis

elapsedMillis

```
#include <elapsedMillis.h>

#define OUTPIN LED_BUILTIN
#define PERIOD 500

elapsedMillis ledTime;

bool ledState = false;

void setup()
{
    // initialize the digital pin as an output.
    pinMode(OUTPIN, OUTPUT);
}

void loop()
{
    if (ledTime >= PERIOD)
    {
        ledState = !ledState;
        digitalWrite(OUTPIN, ledState);
        ledTime = 0;
    }
    // do other stuff here
}
```

LED ledTime。

ledTimePERIOD750。 ledTime""250。 ledTimePERIOD LED。

millis

arduino

```
// set constants for blinking the built-in LED at 1 Hz
#define OUTPIN LED_BUILTIN
#define PERIOD 500 // this is in milliseconds

int ledState = LOW;

// millis() returns an unsigned long so we'll use that to keep track of time
unsigned long lastTime = 0;

void setup() {
    // set the digital pin as output:
    pinMode(OUTPIN, OUTPUT);
}

void loop() {
    unsigned long now = millis();
```

```

if (now - lastTime >= PERIOD) // this will be true every PERIOD milliseconds
{
    lastTime = now;
    if (ledState == LOW)
    {
        ledState = HIGH;
    }
    else
    {
        ledState = LOW;
    }
    digitalWrite(OUTPIN, ledState);
}

// now there's lots of time to do other stuff here
}

```

millis() - - elapsedMillis°

elapsedMilliselapsedMicros

```

#include <elapsedMillis.h>

void setup() {
    Serial.begin(115200);
    elapsedMillis msTimer;
    elapsedMicros usTimer;

    long int dt = 500;
    delay(dt);

    long int us = usTimer;
    long int ms = msTimer;

    Serial.print("delay(");Serial.print(dt);Serial.println(") took");
    Serial.print(us);Serial.println(" us, or");
    Serial.print(ms);Serial.println(" ms");
}

void loop() {
}

```

elapsedMilliselapsedMicros° °

1

```

unsigned long intervals[] = {250,2000}; //this defines the interval for each task in
milliseconds
unsigned long last[] = {0,0};           //this records the last executed time for each task

void setup() {
    pinMode(LED_BUILTIN, OUTPUT); //set the built-in led pin as output
    Serial.begin(115200);         //initialize serial
}

void loop() {
    unsigned long now = millis();
}

```



```

    if(now-last[0]>=intervals[0]){ last[0]=now; firstTask(); }
    if(now-last[1]>=intervals[1]){ last[1]=now; secondTask(); }

    //do other things here
}

void firstTask(){
    //let's toggle the built-in led
    digitalWrite(LED_BUILTIN, digitalRead(LED_BUILTIN)?0:1);
}

void secondTask(){
    //say hello
    Serial.println("hello from secondTask()");
}

```

15 intervalslast

```

unsigned long intervals[] = {250,2000}; //this defines the interval for each task in
milliseconds
unsigned long last[] = {0,0};           //this records the last executed time for each task

void setup() {
    pinMode(LED_BUILTIN, OUTPUT); //set the built-it led pin as output
    Serial.begin(115200);         //initialize serial
}

void loop() {
    unsigned long now = millis();
    if(now-last[0]>=intervals[0]){ last[0]=now; firstTask(); }
    if(now-last[1]>=intervals[1]){ last[1]=now; secondTask(); }

    //do other things here
}

void firstTask(){
    //let's toggle the built-in led
    digitalWrite(LED_BUILTIN, digitalRead(LED_BUILTIN)?0:1);
}

void secondTask(){
    //say hello
    Serial.println("hello from secondTask()");
}

```

if◦ thirdTask ◦

```

unsigned long intervals[] = {250,2000}; //this defines the interval for each task in
milliseconds
unsigned long last[] = {0,0};           //this records the last executed time for each task

void setup() {
    pinMode(LED_BUILTIN, OUTPUT); //set the built-it led pin as output
    Serial.begin(115200);         //initialize serial
}

void loop() {
    unsigned long now = millis();
    if(now-last[0]>=intervals[0]){ last[0]=now; firstTask(); }

```

```

    if(now-last[1]>=intervals[1]){ last[1]=now; secondTask(); }

    //do other things here
}

void firstTask(){
    //let's toggle the built-in led
    digitalWrite(LED_BUILTIN, digitalRead(LED_BUILTIN)?0:1);
}

void secondTask(){
    //say hello
    Serial.println("hello from secondTask()");
}

```

```

unsigned long intervals[] = {250,2000}; //this defines the interval for each task in
milliseconds
unsigned long last[] = {0,0};           //this records the last executed time for each task

void setup() {
    pinMode(LED_BUILTIN, OUTPUT); //set the built-in led pin as output
    Serial.begin(115200);         //initialize serial
}

void loop() {
    unsigned long now = millis();
    if(now-last[0]>=intervals[0]){ last[0]=now; firstTask(); }
    if(now-last[1]>=intervals[1]){ last[1]=now; secondTask(); }

    //do other things here
}

void firstTask(){
    //let's toggle the built-in led
    digitalWrite(LED_BUILTIN, digitalRead(LED_BUILTIN)?0:1);
}

void secondTask(){
    //say hello
    Serial.println("hello from secondTask()");
}

```

<https://riptutorial.com/zh-TW/arduino/topic/4852/>

20:

- `analogRead(pin) //`

```
Serial.println(val)
```

Examples

```
int val = 0;    // variable used to store the value
                // coming from the sensor

void setup() {
  Serial.begin(9600); //Begin serializer to print out value

  // Note: Analogue pins are
  // automatically set as inputs
}

void loop() {

  val = analogRead(0); // read the value from
                      // the sensor connected to A0.

  Serial.println(val); //Prints the value coming in from the analog sensor

  delay(10); // stop the program for
             // some time
}
```

◦ AREF Arduino ◦

```
#define analogPin 14    //A0 (uno)
#define AREFValue 5     //Standard for 5V Arduinos
#define ADCResolution 1023 //Standard for a 10bit ADC

int ADCValue = 0;
float voltage = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  readADC();
  Serial.print(voltage); Serial.println("V");
}

void readADC()
{
  ADCValue = analogRead(analogPin);
  float = ( (float)ADCValue/ADCRange ) * AREFValue ); //Convert the ADC value to a
float, divide by the ADC resolution and multiply by the AREF voltage
```

```
}
```

<https://riptutorial.com/zh-TW/arduino/topic/2382/>

21:

ArduinoLiquid Crystal LibraryHitachi HD44780LCD16。 16I2C。 5x7。 16216x2160220420x4 2004。

- #include <LiquidCrystal.h> //
- LiquidCrystalrsenabled4d5d6d7//
- LiquidCrystalrsrwenabled4d5d6d7
- LiquidCrystalrsenabled0d1d2d3d4d5d6d7
- LiquidCrystalrsrwenabled0d1d2d3d4d5d6d7

RS	LCDRSArduino
RW	LCDRWArduino
	LCDArduino
d0 - d7	LCDArduino。 d0d1d2d3;d4d5d6d7LCD。

Examples

```
/*
Wiring:
LCD pin 1 (VSS) -> Arduino Ground
LCD pin 2 (VDD) -> Arduino 5V
LCD pin 3 (VO) -> Arduino Ground
LCD pin 4 (RS) -> Arduino digital pin 12
LCD pin 5 (RW) -> Arduino Ground
LCD pin 6 (E) -> Arduino digital pin 11
LCD pin 11 (D4) -> Arduino digital pin 5
LCD pin 12 (D5) -> Arduino digital pin 4
LCD pin 13 (D6) -> Arduino digital pin 3
LCD pin 14 (D7) -> Arduino digital pin 2
*/

#include <LiquidCrystal.h> // include the library

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // start writing on the first row and first column.
  lcd.setCursor(0, 0);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
```

```
// No need to do anything to keep the text on the display  
}
```

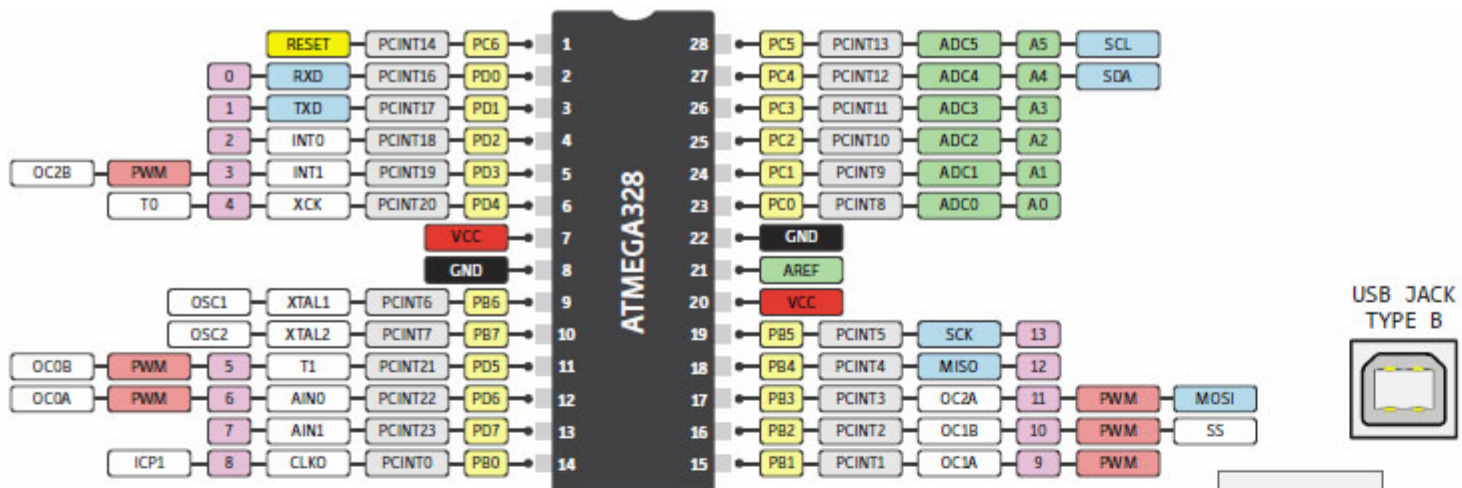
<https://riptutorial.com/zh-TW/arduino/topic/9395/>

22:

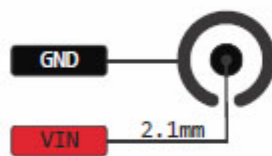
Examples

Arduino Uno R3

- /vin◦ I / OI / O◦ Arduino R3 Uno◦

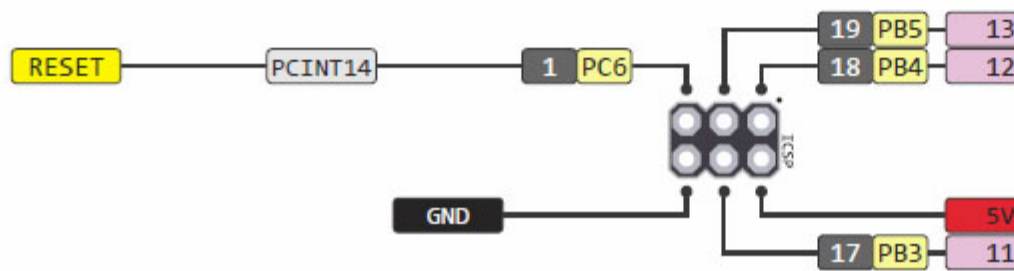
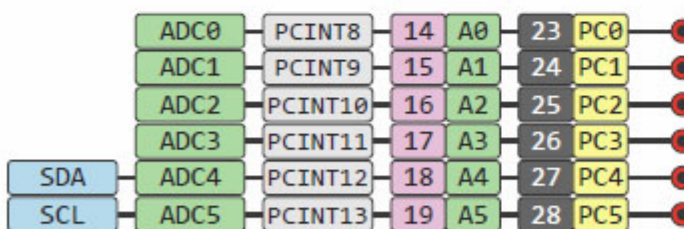
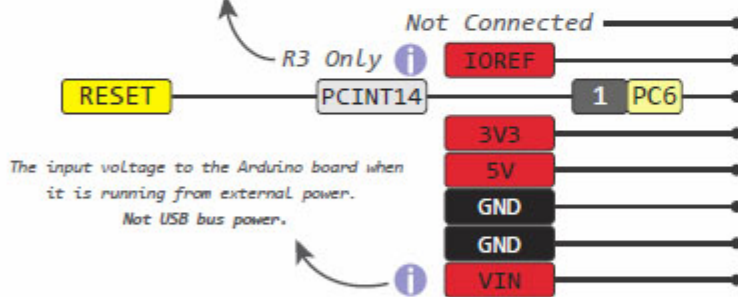


i 7-12V Depending on current drawn



Cut to disable the auto-reset *

This provides a Logic reference voltage for shields that use it. It is connected to the 5V bus.



PWM

PWM。 “”。

PWM3,5,6,9,10,11

PWMArduino Uno R3。 。 analogWrite - PWM。

ADCA0A1A2A3A4A5

SPII2C

Arduino Uno R3USBUSB。 SerialTx on 0Rx on 1

SPII2CArduino.....

SPI MOSI 11 MISO 12 SCLK 13 SS 10

I2CA5SCLA4SDA

LED

Arduino Uno R3LED13.LED13LED。 “”LED。

[Arduino](#)

13LED。 20k1.7V5VLEDLOW。 13pinModeINPUT。

LED13

<https://riptutorial.com/zh-TW/arduino/topic/4386/>

23:

SoftwareSerial.h	
SoftwareSerial rxPin txPin inverse_logic	<ul style="list-style-type: none"> ◦ rxPin 0. txPin 1. inverse_logic true LOW HIGH HIGH LOW. false.
	<ul style="list-style-type: none"> ◦ 300,600,1200,2400,4800,9600,14400,19200,28800,31250,38400,57600 115200.
isListening	<ul style="list-style-type: none"> ◦ false. 64.
	RX. read. SoftwareSerial listen().
	<ul style="list-style-type: none"> ◦ Serial.print().
println	<ul style="list-style-type: none"> ◦ Serial.println().
	<ul style="list-style-type: none"> ◦ ;. listen().
	<ul style="list-style-type: none"> ◦ Serial.write().

rx tx 0 1 SoftwareSerial tx rx.

Examples

```
#include <SoftwareSerial.h>
// its always better to change the default tx and rx as the may interfere with other process
// in future.

// configure tx , rx by default they will be 0 and 1 in arduino UNO
SoftwareSerial blue(3,2);
void setup() {
  // preferred baud rate/data transfer rate in general is 38400
  blue.begin(38400);
  // do initialization or put one time executing code here
}

void loop() {

  // put code that you want it to run every time no matter what
  if(blue.available()){
```

```
// put only that code which needsd to run when there is some data
// This means that the their is some data sent over the bluetooth
// You can do something with the data

int n;
// consider that the data received to be integer, read it by using blue.parseInt();

n = blue.parseInt();

}

}
```

<https://riptutorial.com/zh-TW/arduino/topic/2543/>

24:

Examples

```
variableType variableName;
```

```
variableType variableName;
```

```
variableType variableName;
```

```
variableType variableName;
```

```
int a; // declared previously  
a = 2;
```

```
int a; // declared previously  
a = 2;
```

- char 1
- byte 8
- int 16ATMEGA32Arduino Due
- unsigned int 16ATMEGA32Arduino Due
- long 32
- unsigned long 32
- float 4
- double 4ATMEGA8Arduino Due

```
char a = 'A';  
char a = 65;  
  
byte b = B10010;  
  
int c = 2;  
  
unsigned int d = 3;  
  
long e = 186000L;  
  
unsigned long f = millis(); // as an example  
  
float g = 1.117;  
  
double h = 1.117;
```

<https://riptutorial.com/zh-TW/arduino/topic/2565/>

25:

- randommax//0max
- randomminmax//minmax
- randomSeedseed//de◦



randomSeed() randomSeed(5) ◦ ◦

Examples

random()

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  long randomNumber = random(500); // Generate a random number between 0 and 499
  Serial.println(randomNumber);

  randomNumber = random(100, 1000); // Generate a random number between 100 and 999
  Serial.println(randomNumber);

  delay(100);
}
```

random() randomSeed()

```
void setup() {
  Serial.begin(9600);

  // If analog pin 0 is left unconnected, analogRead(0) will produce a
  // different random number each time the sketch is run.
  randomSeed(analogRead(0));
}

void loop() {
  long randomNumber = random(500); // Generate a random number between 0 and 499
  Serial.println(randomNumber);

  delay(100);
}
```

<https://riptutorial.com/zh-TW/arduino/topic/2238/>



Examples

```
#define NOTE_C4 262 //From pitches.h file defined in [Arduino Tone Tutorial][1]

int Key = 2;
int KeyVal = 0;

byte speaker = 12;

void setup()
{
  pinMode(Key, INPUT); //Declare our key (button) as input
  pinMode(speaker, OUTPUT);
}

void loop()
{
  KeyVal = digitalRead(Key);
  if (KeyVal == HIGH) {
    tone(speaker, NOTE_C4); //Sends middle C tone out through analog speaker
  } else {
    noTone(speaker); //Ceases tone emitting from analog speaker
  }

  delay(100);
}
```

[1] <https://www.arduino.cc/en/Tutorial/toneMelody>

<https://riptutorial.com/zh-TW/arduino/topic/2384/>

S. No		Contributors
1	arduino	Abhishek Jain , Christoph , Community , Danny_ds , Doruk , geek1011 , gmuraleekrishna , H. Pauwelyn , jleung513 , Martin Carney , Mizole Ni , Shef , uruloke , Wolfgang
2	Arduino IDE	geek1011 , Jeremy , jleung513 , sohnryang , uruloke
3	I2C	Asaf
4	MIDI	Rich Maes
5	PWM -	Danny_ds , Johnny Mopp , JorgeGT , Martin Carney
6	PythonArduino Uno	Danny_ds , Peter Mortensen , Stark Nguyen
7	SPI	Christoph
8		DavidJ , Martin Carney
9		blainedwards8 , Danny_ds , geek1011 , Leah , Martin Carney , MikeS159 , Morgoth , Nufail Achath , Peter Mortensen , uruloke
10		geek1011 , macro , Morgoth
11		datafiddler , Leah , MikeCAT
12		Oscar Lundberg
13	EEPROM	AZ Vcience , Chris Combs , Danny_ds , Jeremy , Peter Mortensen , RamenChef
14	ArduinoAtmel Studio 7	Danny_ds , Nate
15		datafiddler , Martin Carney , MikeCAT
16		Martin Carney , uruloke
17		uruloke
18		Danny_ds , robert
19		Christoph , Rei
20		Jake Lites , MikeS159 , Ouss4 , uruloke
21		Morgoth

22	Jeremy, Martin Carney	
23		Girish, Martin Carney
24	Leah, MikeCAT	
25		Danny_ds, Javier Rizzo Aguirre, MikeCAT
26	Jake Lites, MikeCAT	