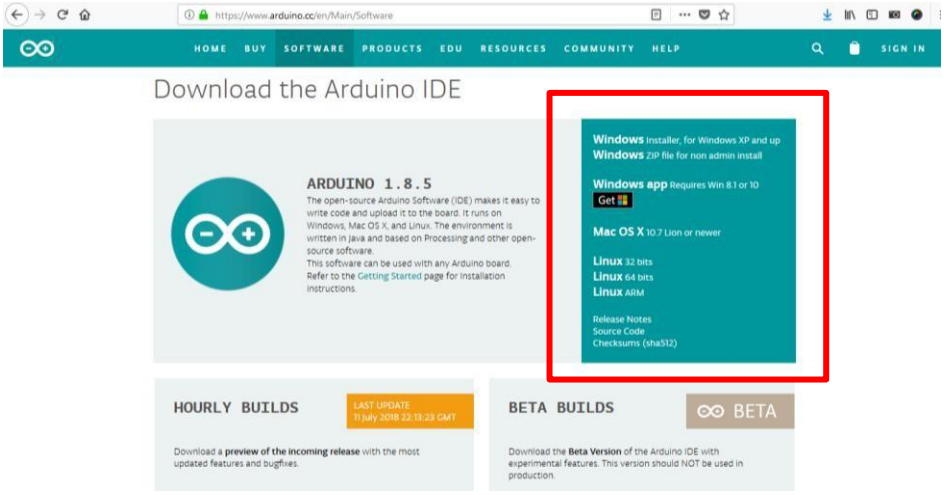
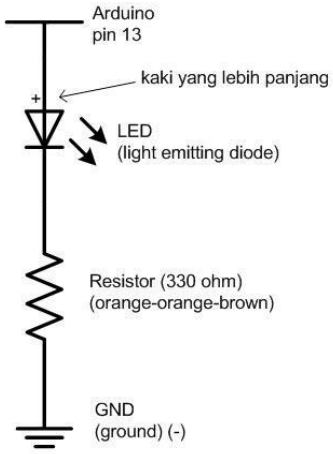
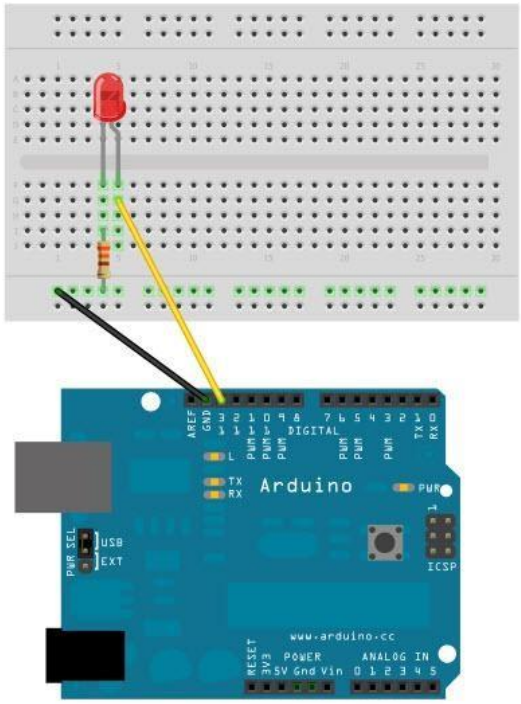
Petunjuk Programming Arduino :

1. Unduh Arduino IDE
   Pilih versi windows / mac os / linux.



2. Rangkaian di arduino dan project board (sumber gambar: google.com).

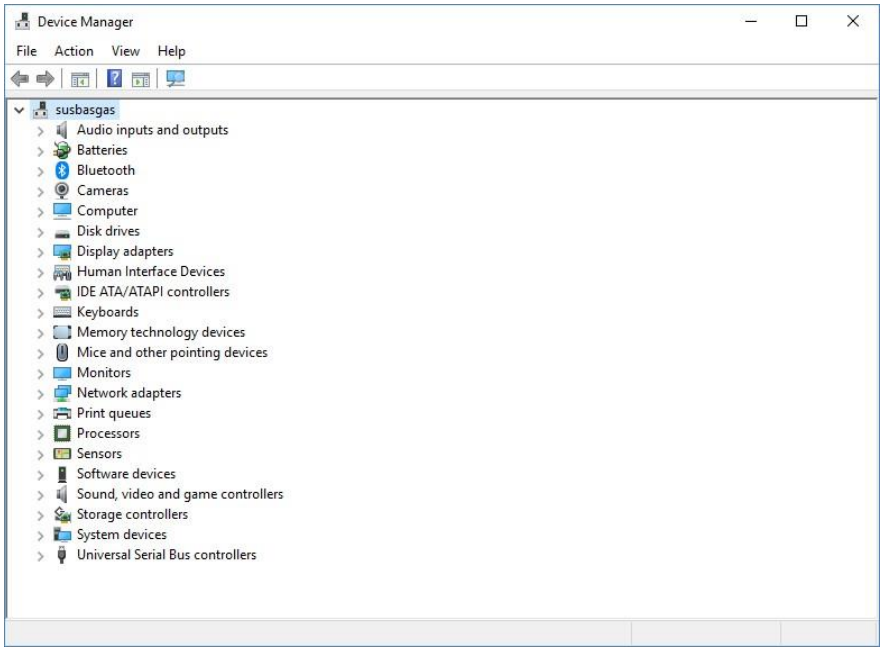




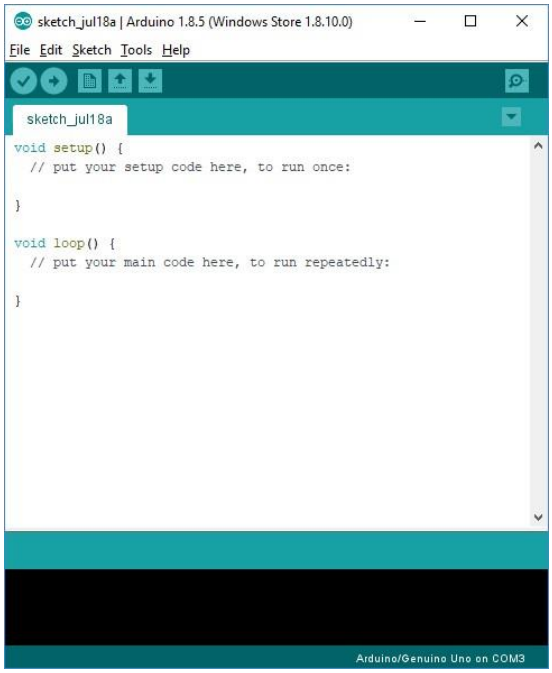
Keterangan:

1. Pin di arduino yang digunakan nomor 13 dan GND
2. Besaran resistor 330 ohm
3. LED

3. Melakukan pemrograman pertama di Arduino IDE
   a. Colokkan kabel dari arduino ke USB di komputer, lalu buka **computer > device manager** (**Control Panel\System and Security\System**)untuk melihat port yang digunakan.



   b. Buka Arduino IDE



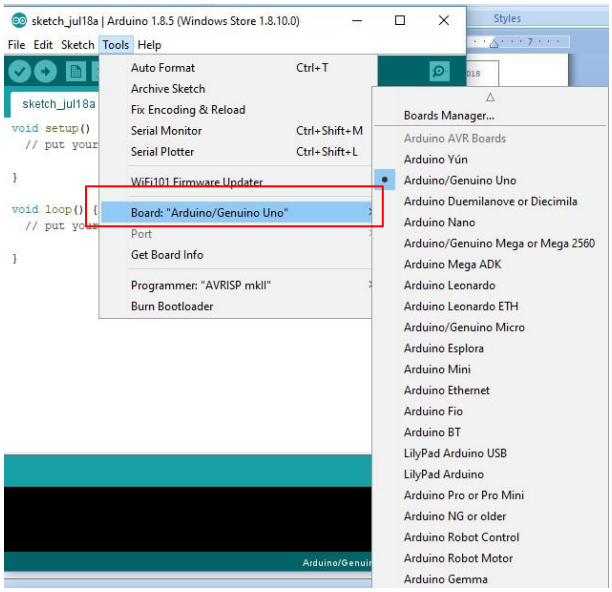
c. Tuliskan program untuk membuat LED menjadi berkedip.

```
const int pinn=12;

void setup() {
  // put your setup code here, to run once:
  pinMode(pinn,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(pinn,HIGH);
  delay(1000);

  digitalWrite(pinn,LOW);
  delay(1000);
}
```

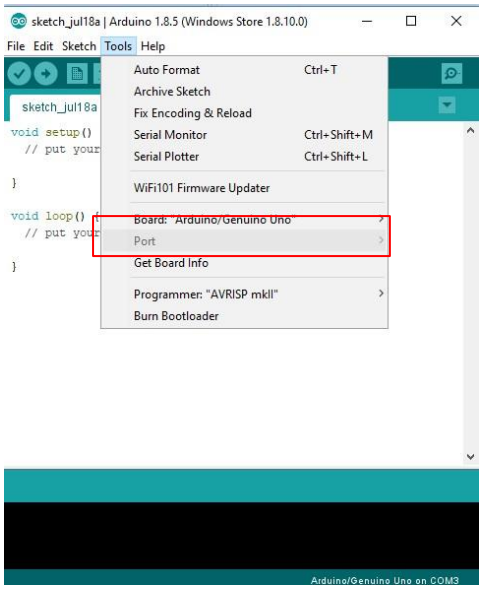
Keterangan:

1. **Case sensitive** artinya huruf kecil dan huruf besar sangat mempengaruhi dari keberhasilan atau kegagalan di dalam menulis program.
2. **const int pinn=12;** pinn adalah variabel dengan tipe data integer dan digunakan untuk menyimpan angka. Angka 12 menunjukkan pin yang digunakan di arduino. Jika melihat rangkaian di atas, maka pin yang digunakan adalah 13. Jadi angka 12 sebaiknya diubah ke angka 13 atau ke port yang sedang digunakan saat ini.
3. **digitalWrite(pinn,HIGH);** HIGH digunakan untuk menyalakan LED
4. **digitalWrite(pinn,LOW);** LOW digunakan untuk mematikan LED

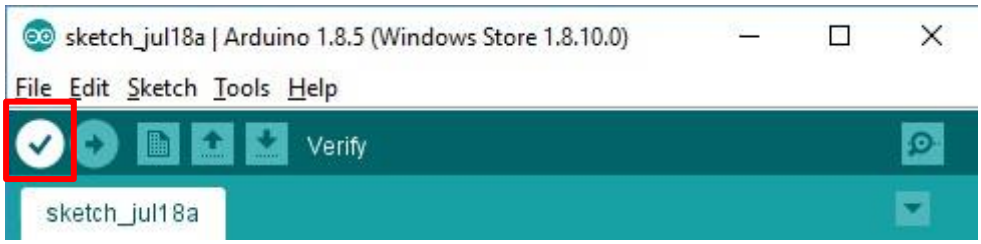
4. Melakukan *running* program arduino LED berkedip.
   a. Setting board arduino (tools > board > arduino Genuino Uno)



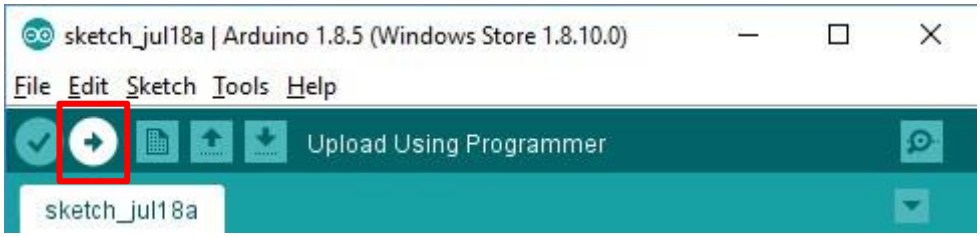
b. Setting port



c. Compile program. Memeriksa dan memastikan baris program tidak terjadi kesalahan. Tunggu sebentar untuk melihat hasil compile yang menyatakan program tidak terdapat error atau program masih terdapat error.

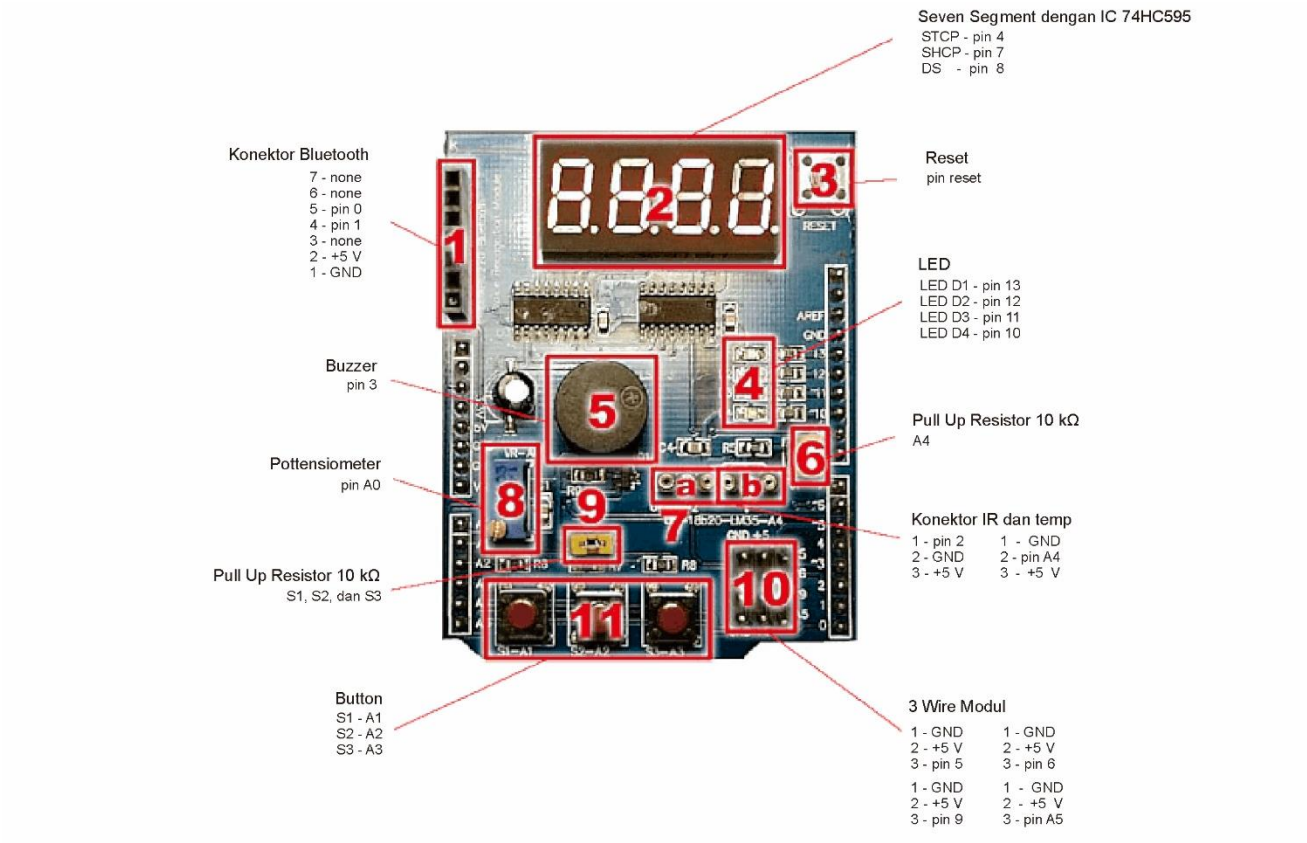


d. Upload ke Arduino. Digunakan untuk mengirimkan program ke Arduino yang digunakan.

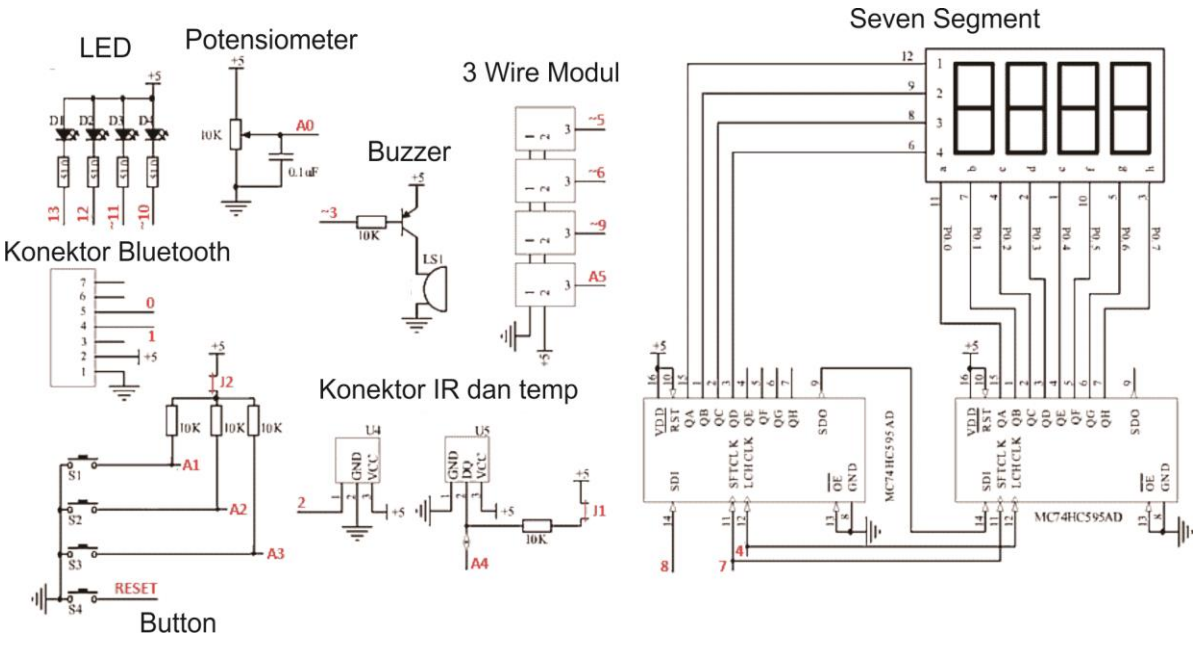


e. Perhatikan LED menyala secara berkedip.

Pinout Arduino Multifunction Board



Schematic Arduino Multifunction Board



Arduino Programming Cheat Sheet

Primary source: Arduino Language Reference
http://arduino.cc/en/Reference/

Structure & Flow

Basic Program Structure
```
void setup() {
  // Runs once when sketch starts
}
void loop() {
  // Runs repeatedly
}
```

Control Structures
```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
for (int i = 0; i < 10; i++) { ... }
break;     // Exit a loop immediately
continue;  // Go to next iteration
switch (var) {
  case 1:
    ...
    break;
  case 2:
    ...
    break;
  default:
    ...
}
return x;  // x must match return type
return;    // For void return type
```

Function Definitions
```
<ret. type> <name>(<params>) { ... }
e.g. int double(int x) {return x*2;}
```

Operators

General Operators

=	assignment		
+	add	-	subtract
*	multiply	/	divide
%	modulo		
==	equal to	!=	not equal to
<	less than	>	greater than
<=	less than or equal to		
>=	greater than or equal to		
&&	and	||	or
!	not		

Compound Operators

++	increment
--	decrement
+=	compound addition
-=	compound subtraction
*=	compound multiplication
/=	compound division
&=	compound bitwise and
|=	compound bitwise or

Bitwise Operators

&	bitwise and	|	bitwise or
^	bitwise xor	~	bitwise not
<<	shift left	>>	shift right

Pointer Access

& reference: get a pointer
* dereference: follow a pointer

Built-in Functions

Pin Input/Output
```
Digital I/O - pins 0-13 A0-A5
  pinMode(pin,
    [INPUT, OUTPUT, INPUT_PULLUP])
  int digitalread(pin)
  digitalWrite(pin, [HIGH, LOW])

Analog In - pins A0-A5
  int analogRead(pin)
  analogReference(
    [DEFAULT, INTERNAL, EXTERNAL])

PWM Out - pins 3 5 6 9 10 11
  analogWrite(pin, value)
```

Advanced I/O
```
tone(pin, freq_Hz)
tone(pin, freq_Hz, duration_ms)
noTone(pin)
shiftOut(dataPin, clockPin,
  [MSBFIRST, LSBFIRST], value)
unsigned long pulseIn(pin,
  [HIGH, LOW])
```

Time
```
unsigned long millis()
  // Overflows at 50 days
unsigned long micros()
  // Overflows at 70 minutes
delay(msec)
delayMicroseconds(usec)
```

Math
```
min(x, y)   max(x, y)   abs(x)
sin(rad)    cos(rad)    tan(rad)
sqrt(x)     pow(base, exponent)
constrain(x, minval, maxval)
map(val, fromL, fromH, toL, toH)
```

Random Numbers
```
randomSeed(seed) // long or int
long random(max) // 0 to max-1
long random(min, max)
```

Bits and Bytes
```
lowByte(x)    highByte(x)
bitRead(x, bitn)
bitWrite(x, bitn, bit)
bitSet(x, bitn)
bitClear(x, bitn)
bit(bitn)  // bitn: 0=LSB 7=MSB
```

Type Conversions
```
char(val)     byte(val)
int(val)      word(val)
long(val)     float(val)
```

External Interrupts
```
attachInterrupt(interrupt, func,
 [LOW, CHANGE, RISING, FALLING])
detachInterrupt(interrupt)
interrupts()
noInterrupts()
```

Libraries

Serial - comm. with PC or via RX/TX
```
begin(long speed)  // Up to 115200
end()
int available()  // #bytes available
int read()   // -1 if none available
int peek()   // Read w/o removing
flush()
print(data)    println(data)
write(byte)    write(char * string)
write(byte * data, size)
SerialEvent()  // Called if data rdy
```

SoftwareSerial.h - comm. on any pin
```
SoftwareSerial(rxPin, txPin)
begin(long speed)  // Up to 115200
listen()       // Only 1 can listen
isListening()  // at a time.
read, peek, print, println, write
  // Equivalent to Serial library
```

EEPROM.h - access non-volatile memory
```
byte read(addr)
write(addr, byte)
EEPROM[index]  // Access as array
```

Servo.h - control servo motors
```
attach(pin, [min_uS, max_uS])
write(angle)  // 0 to 180
writeMicroseconds(uS)
   // 1000-2000; 1500 is midpoint
int read()    // 0 to 180
bool attached()
detach()
```

Wire.h - I²C communication
```
begin()      // Join a master
begin(addr) // Join a slave @ addr
requestFrom(address, count)
beginTransmission(addr) // Step 1
send(byte)              // Step 2
send(char * string)
send(byte * data, size)
endTransmission()       // Step 3
int available() // #bytes available
byte receive()  // Get next byte
onReceive(handler)
onRequest(handler)
```

Variables, Arrays, and Data

Data Types

boolean	true | false
char	-128 - 127, 'a' '$' etc.
unsigned char	0 - 255
byte	0 - 255
int	-32768 - 32767
unsigned int	0 - 65535
word	0 - 65535
long	-2147483648 - 2147483647
unsigned long	0 - 4294967295
float	-3.4028e+38 - 3.4028e+38
double	currently same as float
void	i.e., no return value

Strings
```
char str1[8] =
  {'A','r','d','u','i','n','o','\0'};
  // Includes \0 null termination
char str2[8] =
  {'A','r','d','u','i','n','o'};
  // Compiler adds null termination
char str3[] = "Arduino";
char str4[8] = "Arduino";
```

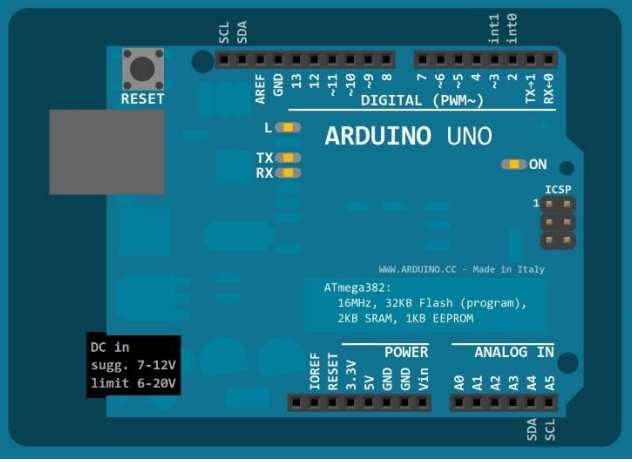
Numeric Constants

123	decimal
0b01111011	binary
0173	octal - base 8
0x7B	hexadecimal - base 16
123U	force unsigned
123L	force long
123UL	force unsigned long
123.0	force floating point
1.23e6	1.23*10^6 = 1230000

Qualifiers

static	persists between calls
volatile	in RAM (nice for ISR)
const	read-only
PROGMEM	in flash

Arrays
```
int myPins[] = {2, 4, 8, 3, 6};
int myInts[6];   // Array of 6 ints
myInts[0] = 42;  // Assigning first
                 // index of myInts
myInts[6] = 12;  // ERROR! Indexes
                 // are 0 though 5
```



Adapted from:
- Original: Gavin Smith
- SVG version: Frederic Dufourg
- Arduino board drawing: Fritzing.org

Contoh Arduino Multifunction

1. LED Blinking

```
#define LED_D4 10

void setup() {
  pinMode(LED_D4,OUTPUT);
}

void loop() {
  digitalWrite(LED_D4,HIGH);
  delay(1000);
  digitalWrite(LED_D4,LOW);
  delay(1000);
}
```

2. Buzzer Delay

```
#define Buzzer 3

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

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

3. LED Combination

```
int LED[4]={13,12,11,10};
int i;

void setup() {
  for(i=0;i<=3;i++)
  pinMode(LED[i],OUTPUT);
}

void loop() {
  for(i=0;i<=3;i++){
    digitalWrite(LED[i],HIGH);
    delay(1000);
  }
   for(i=0;i<=3;i++){
    digitalWrite(LED[i],LOW);
    delay(1000);
  }
}
```

4. Switch Combination

```
#define Buzzer 3
#define LED_D4 10
#define Sw_1 A1
#define Sw_2 A2
#define Sw_3 A3

void setup() {
  pinMode(Buzzer,OUTPUT);
  pinMode(LED_D4,OUTPUT);
  pinMode(Sw_1,INPUT);
  pinMode(Sw_2,INPUT);
  pinMode(Sw_3,INPUT);
  digitalWrite(LED_D4,HIGH);
  digitalWrite(Buzzer,HIGH);
}

void loop() {
  if(digitalRead(Sw_1)==0){
    digitalWrite(LED_D4,LOW);
  }else if(digitalRead(Sw_2)==0){
    digitalWrite(LED_D4,HIGH);
  }else if(digitalRead(Sw_3)==0){
    digitalWrite(Buzzer,LOW);
  }else if(digitalRead(Sw_3)==1){
    digitalWrite(Buzzer,HIGH);
  }
}
```

5. Membaca Potensiometer dengan Serial Monitor

```
#define Potensio A0

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

void loop() {
  Serial.println(analogRead(Potensio));
  delay(100);
}
```

6. PWM

```
#define PinPWM 5

void setup() {

}

void loop() {
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    analogWrite(PinPWM, fadeValue);
    delay(30);
  }
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    analogWrite(PinPWM, fadeValue);
    delay(30);
  }
}
```

7. Servo Motor

```
#include <Servo.h>

#define PinServo 5
Servo myservo;
int pos = 0;

void setup() {
  myservo.attach(PinServo);
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
    myservo.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(15);
  }
}
```

8. Library Timer dengan Serial Monitor

```
#include <TimerOne.h>

void setup() {
  Serial.begin(9600);
  Timer1.initialize(1000000);
  Timer1.attachInterrupt(timerInt);
}

int i;
void loop() {
  Serial.println(i);
  delay(2000);
}

void timerInt(){
 i++;
}
```

9. Seven Segment

```
#define Potensio A0
#define S_STCP 4
#define S_SHCP 7
#define S_DS 8

byte
Segment_Map[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
byte Segment_Digit[4]={0xF8,0xF4,0xF2,0xF1};

void setup() {
  Serial.begin(9600);
  pinMode(Potensio,INPUT);
  pinMode(S_STCP,OUTPUT);
  pinMode(S_SHCP,OUTPUT);
  pinMode(S_DS,OUTPUT);
}

int baca;
int digit_baca;
int digit_hasil;
long TS_Pot;

void loop() {
    digitalWrite(S_STCP,LOW);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Map[2]);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Digit[3]);
    digitalWrite(S_STCP,HIGH);
    delay(1);
    digitalWrite(S_STCP,LOW);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Map[0]);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Digit[2]);
    digitalWrite(S_STCP,HIGH);
    delay(1);
    digitalWrite(S_STCP,LOW);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Map[1]);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Digit[1]);
    digitalWrite(S_STCP,HIGH);
    delay(1);
    digitalWrite(S_STCP,LOW);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Map[8]);
    shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Digit[0]);
    digitalWrite(S_STCP,HIGH);
    delay(1);
}
```

10. Potensiometer dengan Tampilan Seven Segment

```
#define Potensio A0
#define S_STCP 4
#define S_SHCP 7
#define S_DS 8

byte Segment_Map[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
byte Segment_Digit[4]={0xF8,0xF4,0xF2,0xF1};

void setup() {
  Serial.begin(9600);
  pinMode(Potensio,INPUT);
  pinMode(S_STCP,OUTPUT);
  pinMode(S_SHCP,OUTPUT);
  pinMode(S_DS,OUTPUT);
}

int baca;
int digit_baca;
int digit_hasil;
long TS_Pot;

void loop() {
    if(millis()-TS_Pot>=100){
      baca=analogRead(Potensio);
      TS_Pot=millis();
    }
      digit_baca=baca;
    for(int i=0;i<=3;i++){
      digit_hasil=digit_baca%10;
      digitalWrite(S_STCP,LOW);
      shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Map[digit_hasil]);
      shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Digit[i]);
      digitalWrite(S_STCP,HIGH);
      digit_baca/=10;
    }
    delay(1);
}
```

11. Seven Segmen Timer

```
#include <TimerOne.h>
#define S_STCP 4
#define S_SHCP 7
#define S_DS 8

byte Segment_Map[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
byte Segment_Digit[4]={0xF8,0xF4,0xF2,0xF1};

void setup() {
  pinMode(S_STCP,OUTPUT);
  pinMode(S_SHCP,OUTPUT);
  pinMode(S_DS,OUTPUT);
  Timer3.initialize(1000000);
  Timer3.attachInterrupt(timerInt);

}

int waktu[4];
void loop() {

    for(int i=0;i<=3;i++){
      digitalWrite(S_STCP,LOW);
      shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Map[waktu[i]]);
      shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Digit[i]);
      digitalWrite(S_STCP,HIGH);
    }
    delay(1);
}

void timerInt(){
  waktu[0]++;
  if(waktu[0]>=10){
    waktu[0]=0;
    waktu[1]++;
  }
  if(waktu[1]>=10){
    waktu[1]=0;
    waktu[2]++;
  }
  if(waktu[2]>=10){
    waktu[2]=0;
    waktu[3]++;
  }
  if(waktu[3]>=10){
    waktu[0]=0;
    waktu[1]=0;
    waktu[2]=0;
    waktu[3]=0;
  }
}
```

12. Potensiometer dengan Servo Motor

```
#include <Servo.h>

#define Potensio A0
#define S_STCP 4
#define S_SHCP 7
#define S_DS 8
#define PinServo 5

byte Segment_Map[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
byte Segment_Digit[4]={0xF8,0xF4,0xF2,0xF1};

Servo Svm;

void setup() {
  Serial.begin(9600);
  pinMode(Potensio,INPUT);
  pinMode(S_STCP,OUTPUT);
  pinMode(S_SHCP,OUTPUT);
  pinMode(S_DS,OUTPUT);
  Svm.attach(PinServo);
}

int baca;
int digit_baca;
int digit_hasil;
long TS_Pot;
void loop() {
    if(millis()-TS_Pot>=100){
      baca=map(analogRead(Potensio),0,1023,0,180);
      TS_Pot=millis();
      Svm.write(baca);
    }
      digit_baca=baca;
    for(int i=0;i<=3;i++){
      digit_hasil=digit_baca%10;
      digitalWrite(S_STCP,LOW);
      shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Map[digit_hasil]);
      shiftOut(S_DS,S_SHCP,MSBFIRST,Segment_Digit[i]);
      digitalWrite(S_STCP,HIGH);
      digit_baca/=10;
    }
    delay(1);
}
```