

ORduino

by Open Robotics team

Structure

```
void setup(); void loop()
```

Constants

```
HIGH | LOW //3-5V | 0-2V
INPUT | OUTPUT
true | false
143 //Decimal
0173 //Octal
B10111111 //Bin (8 bits only)
0x7B //Hex
7U //Force unsigned
10L //Force long
15UL //Force long unsigned
10.0 //Forced floating point
2.4e5 //240'000
```

Arrays

```
int myints[6];
int mynums[]={2,4,8,3,6};
int myvals[6]={2,4,-8,3,2};
```

Random

```
randomSeed(x); //int,long
long random(max);
long random(min,max);
```

Bits & Bytes

```
lowByte(x);
highByte(x);
bitRead(val,bitn);
bitWrite(val,bitn,bit);
bitSet(val,bit);
bitClear(val,bit);
bit(bitn); //0-LSB,7-MSB
```

Data types

```
void
boolean //0,1,false,true
char //-128..127
unsigned char //0..255
byte //0..255
int //-32'768..32'767
unsigned int //0..65'535
word //0..65'535
long //-2'147'483'648..2'147'483'647
unsigned long //0..4'294'967'295
float //-3.4028235E+38 to 3.4028235E+38
double //same as float
```

Strings

```
char S1[15];
char S2[8]='o','r','d','u','i','n','o';
char S3[8]='o','r','d','u','i','n','o','\0';
//included \0 null termination
char S4[]="ORduino";
char S5[8]="ORduino";
char S6[15]="ORduino";
```

Time

```
unsigned long millis(); //50 days overflow
unsigned long micros(); //70 min overflow
delay(ms);
delayMicroseconds(us);
```

Math

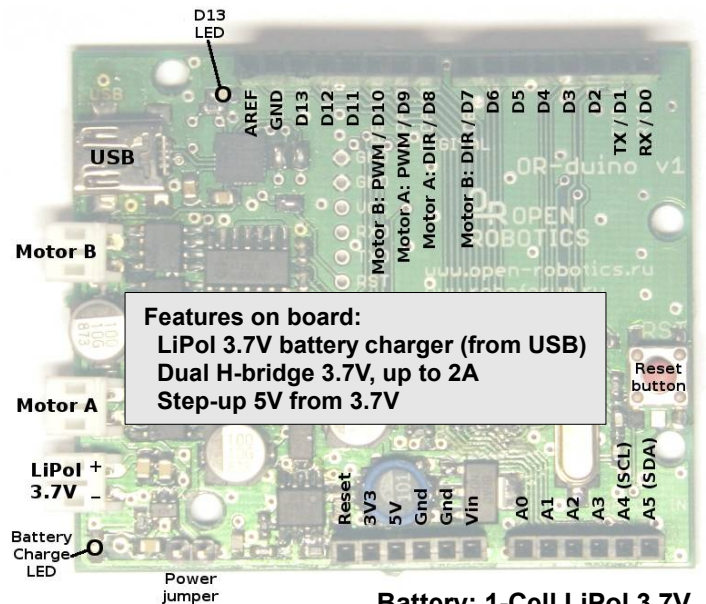
```
min(x,y); max(x,y); abs(x);
constrain(x,minval,maxval);
map(x,fromL,fromH,toL,toH)
pow(base,exponent); sqrt(x);
sin(rad); cos(rad); tan(rad);
```

MOTORS TEST Example

```
int pwmA=9, dirA=8; //Motor A PWM, DIR connected to digital pins 9, 8
int pwmB=10, dirB=7; //Motor B PWM, DIR connected to digital pins 10, 7
```

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

```
void loop()
{
digitalWrite(dirA, HIGH); digitalWrite(pwmA, HIGH); //Full forward A
digitalWrite(dirB, HIGH); digitalWrite(pwmB, HIGH); //Full forward B
delay(1000);
digitalWrite(dirA, LOW); digitalWrite(pwmA, HIGH); //Full backward A
digitalWrite(dirB, LOW); digitalWrite(pwmB, HIGH); //Full backward A
delay(1000);
}
```



Features on board:
LiPo 3.7V battery charger (from USB)
Dual H-bridge 3.7V, up to 2A
Step-up 5V from 3.7V

Battery: 1-Cell LiPo 3.7V
Motors: DC, 3.7V, up to 2A

Digital pins

```
pinMode(pin, [INPUT|OUTPUT]);
digitalWrite(pin,value);
int digitalRead(pin);
//write HIGH to inputs
//to use pull-up res
```

Advanced I/O

```
//PWM (OC*) pins only
analogWrite(pin,value);

tone (pin, freqhz); //kill PWM on 3,11
tone (pin, freqhz, duration_ms);
noTone(pin);

shiftOut(dataPin,clockPin,[MSBFIRST|LSBFIRST],value);

unsigned long pulseIn(pin, [HIGH|LOW]);
```

Servo

```
#include <Servo.h> //up to 12 servo
Servo. //kill PWM on 9,10 pins
attach(pin,min_uS,max_uS);
write(angle); //0..180
writeMicroseconds(uS); //1000..2000
read(); attached(); detach();
```

External Interrupts

```
attachInterrupt(interrupt,function, [LOW,CHANGE,RISING,FALLING]);
detachInterrupt(interrupt);
interrupts();
noInterrupts();
//attachInterrupt(0, blink, CHANGE);
```

SPI

```
SPI. //include <SPI.h>
begin(); end();
setBitOrder(); //LSBFIRST|MSBFIRST
SetClockDivider(divider);
//divider=SPI_CLOCK_DIV[2..128]
setDataMode(); //SPI_MODE[0..3]
byte transfer(byte);
```

Analog pins

```
analogReference([DEFAULT,INTERNAL,EXTERNAL]);
int analogRead(pin); //call twice if
//switching pins from high Z source
```

EEPROM

```
EEPROM. //include <EEPROM.h>
byte read(intAddr);
write(intAddr,myByte);
```

Hardware Serial

```
Serial. //no #include needed
begin([300,1200,2400,4800,9600,14400,28800,38400,57600,115200]);
end();
int available();
int read();
flush();
print(val [,format]);
//BYTE|BIN|OCT|DEC|HEX
//or digits after '.' for floats
//Serial.print(78, HEX) gives "4E"
//Serial.print(1.2345, 3) gives "1.234"
Println(); //same, but <cr> added
write();
```

Hardware i2c

```
Wire. //include <Wire.h>
begin(); //join as master
begin(addr); //join as slave #addr
requestFrom(addr,cnt);
beginTransmission(addr); //step 1
send(myByte); //step 2
send(char* myString);
send(byte* data, size);
endTransmission(); //step 3
byte available(); //Num of bytes
byte receive(); //Get next byte
onReceive(handler);
onRequest(handler);
```

Software Serial

```
#include <SoftwareSerial.h>
SoftwareSerial.
begin(longSpeed); //up to 9600
char read(); //blocks till data
print(myData); println(myData);
```