ŚCIĄGAWKA ARDUINO

SZKIC

Podstawowa struktura szkicu

```
void setup() {
// uruchamiana na starcie (podłączenie napięcia lub reset)
}

void loop() {
// powtarzana przez cały cykl działania programu
}
```

Definicje funkcji

```
<ret. type> <func. name>(<param. type> <param. name>){ … } i.e.:
float circleCircumference(int radius) { return 3.14 * 2 * radius; }
void printGreeting(string name) { Serial.println(name); }
```

ZMIENNE, TABLICE, TYPY DANYCH

Typy Danych

```
bool/boolean true false
char -128 - 127
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 - tak jak float (poza Due)
void - brak zwracanej wartości
```

Tablice

```
int even[] = {2, 4, 6, 8};
int pins[6];
pins[0] = 10; //indeksowanie od 0
pins[6] = 7; //Częsty błąd -
indeksowanie od 0 do rozmiar-1 !!
```

Typ string

```
char ex1[3] = {'H','i','\0'};
char ex2[3] = {H','i'};
char ex3[] = „Hi";
char ex4[3] = „Hi";
```

Kwalifikatory dostępu

```
static //zachowane pomiędzy
wywołaniami funkcji
volatile //w RAM (dobre do przerwań)
const //tylko do odczytu
PROGMEM //przechowywane we flash
```

Stałe numeryczne

123	dziesiętny
0b01111110	binarny
0123	oktalny -podstawa 8
0xA2	hexadecymalny
123U	wymuś bez znaku
123L	wymuś long
123UL	wymuś unsigned long
123.0	wymuś float
1.23e6	1.23*10^6

OPERATORY

Arytmetyczne

```
= przypisanie
+ dodawanie
- odejmowanie
* mnożenie / dzielenie
% modulo - reszta z dzielenia
```

Porównania

```
== równe
!= nierówne
<  mniejsze niż
>  większe niż
<= mniejsze lub równe
>= większe lub równe
```

Boolowskie

```
&& koniunkcja (and)
|| suma (or)
! negacja (not)
```

Operatory skrócone

```
++ inkrementacja -- dekrement.
+= dodawanie     -= odejmowanie
*= mnożenie      /= dzielenie
```

Operatory bitowe

```
& koniunkcja (and)
| suma (or)
^ różnica symetryczna (xor)
~ negacja (not)
<< przesunięcie bitowe w lewo
>> przesunięcie bitowe w prawo
```

Skrócone operatory bitowe

```
&= skrócony bitowy iloczyn
|= skrócona bitowa suma
```

Wskaźniki / Referencja

```
& referencja: pobierz wskaźnik
* dereferencja: pobierz wartość
```

FUNKCJE WBUDOWANE

PORTY WEJŚCIA/WYJŚCIA (I/O)

Cyfrowe I/O

```
pinMode(pin, mode);
mode - INPUT, OUTPUT, INPUT_PULLUP
int digitalRead(pin);
digitalWrite(pin, state);
state - HIGH, LOW
```

Analogowe I/O

```
int analogRead(pin);
analogReference(source);
source - DEFAULT, INTERNAL, EXTERNAL
analogWrite(pin, value); //PWM
```

Zaawansowane I/O

```
tone(pin, freq_hz);  noTone(pin);
tone(pin, freq_hz, duration_ms);
byte shiftIn(dataPin, clkPin, order);
shiftOut(dataPin, clkPin, order, val);
bitOrder - MSBFIRST, LSBFIRST
unsigned long pulseIn(pin, state,
timeout); //param. timeout opcjonalny
pulseInLong //analogicznie do pulseIn
```

Bity i bajty

```
byte lowByte(x); byte highByte(x);
byte bitRead(x, bitnumber);
bitWrite(x, bitnumber, bit);
bitSet(x, bitnumber);
bitClear(x, bitnumber);
bit(bitnumber);
```

Funkcje matematyczne

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

Przerwania zewnętrzne

```
attachInterrupt(interrupt, ISR,
mode);
mode - LOW, CHANGE, RISING,
FALLING
detachInterrupt(interrupt);
interrupts();
noInterrupts();
```

Konwersje typów

```
char(val);      byte(val);
int(val);       word(val);
long(val);      float(val);
```

Liczby pseudolosowe

```
randomSeed(seed);
long random(max); //min = 0
long random(min, max);
```

Czas

```
unsigned long millis(); //<50 days
unsigned long micros(); //<70 mins
delay(miliseconds);
delayMicroseconds(useconds);
```

BIBLIOTEKI ARDUINO

Serial - komunikacja przez UART

```
begin(long speed);
end();
int available() //il. dostęp. bajtów
int read(); //-1 jeśli brak
int peek(); //odczyt bez usuwania
flush();
print(data); println(data);
write(byte); write(char* str);
write(byte* data, size);
serialEvent();
```

EEPROM.h - pamięć nieulotna

```
byte read(address);
write(address, byte);
put(addr, data); get(addr, data);
EEPROM[index]; //dostęp jako tablica
```

SoftwareSerial.h - UART na dow. pin

```
SoftwareSerial(rxPin, txPin);
bool listen(); //tylko 1 słucha
bool isListening();
begin, read, peek, print, println,
write, available //jak w Serial
```

Wire.h - komunikacja I2C

```
begin(); //dołącz jako master
begin(addr); //doł. jako slave
requestFrom(address, count);
setClock(clkFreq);
beginTransmission(addr);
write(byte)
write(char* str);
write(byte* data, length);
byte endTransmission();
int available(); //il. bitów
byte read(); //nast. bajt
onReceive(handler);
onRequest(handler);
```

Servo.h - obsługa serwomech.

```
attach(pin, min_us, max_us);
write(angle); //0 to 180
writeMicroseconds(useconds);
//1000 - 2000; 1500 środek
int read(); //kąt od 0 do 180
bool attached();
detach();
```

INSTRUKCJE STERUJĄCE

```
if (x > 0) { … } else { … }
switch (x) {
  case 1:
        …
        break;
  case 2:
        …
        break;
  default:
        …
        break;
}
while (x < 10) { … }
for (int i = 0; i < 10; i++) { … }
do { … } while (x < 10);
break; //Opuść pętlę/switch natychmiast
continue; //do początku kol. iteracji
```


Autor: Michał Zięba

Na podstawie pracy Mark Liffiton

www.przygodyzkodem.pl
GITHUB: przygodyzkodem
IG: przygodyzkodem
FB: przygodyzkodem