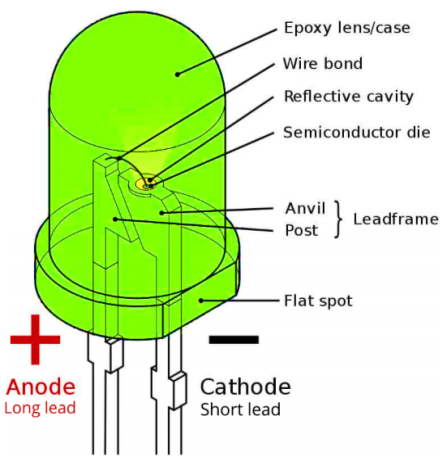
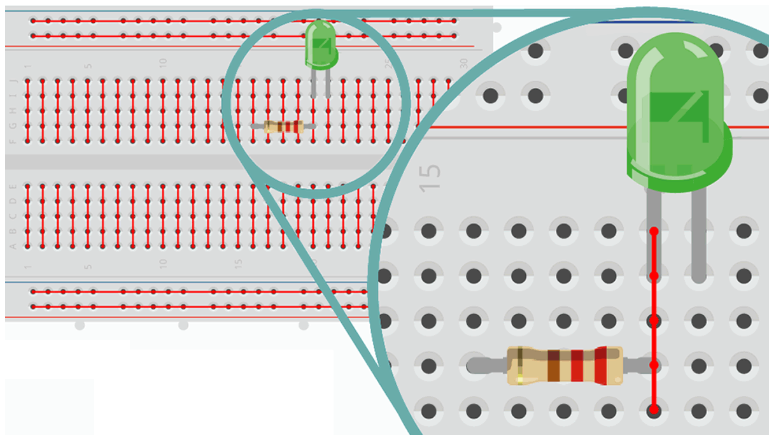
|  |  |  |
| --- | --- | --- |
| **Datové typy** | | |
| boolean | logická hodnota (true/false) | |
| byte | 8-bitová hodnota (0÷255) | |
| int | celé číslo (-32768÷32767) | |
| unsigned int | kladné celé číslo (0÷65535) | |
| long | Dlouhé celé číslo (-231 ÷ 231 ) | |
| float | reálné číslo (-3,4\*1038 ÷ 3,4\*1038 ) | |
| char | Znak | |
| String | řetězec znaků | |
|  |  | |
| **Proměnné** | | |
| int age = 25; | | |
| float temp = 36.5; | | |
| char znak = 'A'; | | |
|  |  | |
| **Pole** | | |
| int myNumbers[] = {25, 50, 75, 100}; myNumbers[0] = 12; for (int i=0;i<4;i++){  Serial.println(myNumbers[i]); } | | |
|  |  | |
| **Další syntaxe** | | |
| // jednořádkový komentář | | |
| /\* víceřádkový komentář \*/ | | |
| #define LED\_PIN 13 | | |
|  |  | |
| **Aritmetické operátory** | | |
| + | Součet | |
| - | Rozdíl | |
| \* | Součin | |
| / | Podíl | |
| % | modulo - zbytek po dělení | |
| = | Přiřazení | |
| ++ | inkrementace | |
| -- | Dekrementace | |
| += | součet a přiřazení (a+=b je jako a = a+b) | |
| -+ | rozdíl a přiřazení (a-=b je jako a = a-b) | |
| \*= | součin a přiřazení (a\*=b je jako a = a\*b) | |
| /= | rozdíl a přiřazení (a-=b je jako a = a-b) | |
|  |  | |
| **Porovnání** | | |
| > | Větší | |
| < | Menší | |
| >= | větší nebo rovno | |
| <= | menší nebo rovno | |
| == | rovná se | |
|  |  | |
| **Logické operátory** | | |
| && | a (AND) | |
| || | nebo (OR) | |
|  |  | |
| **Netisknutelné znaky** | | |
| \n | nový řádek | |
| \r | návrat na začátek řádku | |
| \t | odsazení (tabulátor) | |
| **Řídící struktury (podmínky, cykly)** | | | |
| **if** | | Neúplná podmínka if (podmínka) { ... }  if(a==b) { Serial.println("Obě čísla jsou stejná"); } | |
| **if-else** | | Úplná podmínka if (podmínka) { ... } else { ... }  if(a==b) { Serial.println("Obě čísla jsou stejná"); } else { Serial.println("Čísla se navzájem liší."); } | |
| **for** | | Cyklus se známým počtem opakovaní for(start; podmínka; krok) { … }  for(i=1; i<=10; i++) { Serial.println(i); } | |
| **switch** | | Přepínač podle hodnoty proměnné switch (proměnná) { case 1: ...; break; case 2: ...; break; ... default: ...; break; }  switch (i) { case 1: Serial.println("Stiskli jste jedničku."); break; case 2: Serial.println("Stiskli jste dvojku."); break; case 8: Serial.println("Stiskli jste osmičku."); break; default: Serial.println("Zadali jste neplatné číslo."); } | |
| **while** | | Cyklus s neznámym počtem opakovaní a podmínkou na začátku while(podmínka) { ... }  i=1; while(i<=10) { Serial.println(i); i++; } | |
| **do-while** | | Cyklus s neznámým počtem opakovaní  a podmínkou na konci do { ... } while(podmínka);  i=1; do { Serial.println(i); i++; } while(i<=10); | |
| **continue** | | V těle cyklu for, while a do-while umožňuje přeskočit zbytek těla cyklu a pokračovat znovu od začátku cyklu | |
| **break** | | V těle cyklu for, while a do-while umožňuje přeskočit zbytek těla cyklu a celý cyklus ukončit. Program pak pokračuje dalšími příkazy za ukončeným cyklem. | |
| **vlastní funkce** | | int soucet(int a, int b) {  return (a + b);  } | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Digital I/O** | pinMode | Nastavení pinu jako INPUT, OUTPUT nebo INPUT\_PULLUP | pinMode(button\_pin, INPUT); |
| digitalWrite | Zápis 1 (HIGH) nebo 0 (LOW) na pin | digitalWrite(led\_pin, HIGH); |
| digitalRead | Čtení stavu digitálního pinu | stav=digitalRead(button\_pin); |
|  |  |  |  |
| **Analog I/O** | analogWrite | Nastavení PWM signálu na pinu | analogWrite (PWMpin, hod); 0 ≤ hod ≤ 255 |
| analogRead | Čtení stavu analogového pinu (ADC) Vrací hodnotu 0-1023 | stav=analogRead(AINpin); |
|  |  |  |  |
| **Čas** | millis | Počet milisekund od startu programu (přeteče po 50 dnech) | cas=millis(); |
| micros | Počet mikrosekund od startu programu (přeteče po 70min) | cas=micros(); |
| delay | Pozastavení na daný počet milisekund | delay(1000); |
| delayMicroseconds | Pozastavení na daný počet mikrosekund | delayMicroseconds(500); |
| pulseIn | Měření délky pulzu na pinu | duration = pulseIn(pin, HIGH); |
|  |  |  |  |
| **Sériová linka** | Serial.begin | Inicializace a nastavení rychlosti sériového portu | Serial.begin(9600); |
| Serial.available | Vrátí počet přijatých bajtů | pocet=Serial.available(); |
| Serial.print | Odeslání textu nebo dat přes sériovou linku | Serial.print("Ahoj"); nebo Serial.print(cislo); |
| Serial.println | Podobně jako Serial.print a navíc se pošle i znak nového řádku | Serial.println("Ahoj"); nebo Serial.println(cislo); |
|  |  |  |  |
| **Servo** | #include <Servo.h> | Přidání knihovny pro servo | #include <Servo.h> |
| Servo servo1 | Vytvoření objektu z knihovny Servo | Servo servo1; |
| servo.attach | Inicializace serva, nastavení ovládacího pinu | servo1.attach(servo\_pin); |
| servo.write | Nastavení polohy serva ve stupních | servo1.write(90); //úhel 90 stupnu |
|  |  |  |  |
| **LCD displej** | #include <Wire.h> #include <LiquidCrystal\_I2C.h> | Přidání knihovny pro I2C komunikaci a knihovny pro LCD displej | #include <Wire.h> #include <LiquidCrystal\_I2C.h> |
| LiquidCrystal\_I2C lcd | Vytvoření objektu z knihovny LiquidCrystal\_I2C | LiquidCrystal\_I2C lcd(0x27, 20, 4); |
| lcd.begin | Inicializace displeje | lcd.begin(); |
| lcd.clear | Vymazání displeje | lcd.clear() |
| lcd.setCursor | Nastavení pozice kurzoru  (znak, řádek) | lcd.setCursor(0,1); |
| lcd.print | Vypsání textu nebo čísel na displeji | lcd.print(„Ahoj!“); |

**Nepájivé pole (breadboard) Polarita LED diody**