Arduino a podmienky, If, Else, Else if, operátory, relačné, logické, switch :)

Dalo by sa povedať, že podmienky sú vlastne možnosť reakcie na vzniknuté situácie.

Zjednodušene – keď sa niečo stane, budeme na to musieť nejako reagovať. Napríklad: Zazvoní zvonček pri dverách. Naša reakcia môže byť, že ideme otvoriť alebo ho ignorujeme.

Takéto rozhodnutia majú veľa spoločného s vetvením programu. Normálne program začne prvým príkazom a potom vykonáva jeden za druhým. Podmienky však umožnia preskočiť na iné miesto programu, alebo vykonať iný kód, v závislosti od aktuálnej situácie. Veľmi často túto situáciu povedzme stav programu, ovplyvňuje používateľ. Avšak môže na to mať vplyv aj vrátená hodnota z nejakéj funkcie (výpočtu) alebo z pripojeného zariadenia. Napríklad senzor pohybu zaznamená aktivitu a poskytne vstup základnej doske. Podľa skutočného stavu sa môžeme rozhodnúť, či spustiť poplach a volať políciu, alebo sa rozhodnúť, že je všade pokoj a pohoda.

Je veľa situácií, v ktorých sa hodí podmienku použiť. Dokonca by sme povedali, že sa bez tejto zákadnej znalosti nezaobíde ani ten najlepší programátor. To je dôvod, prečo si vysvetlíme, ako s podmienkami pracovať a hlavne poznať ich funkčnosť. Pre lepšiu predstavu si všetko ukážeme na jednotlivých príkladoch.

Zápis podmienok

Možno povedať, že všetky podmienky majú tvar: Pokiaľ niečo, tak vykonáme toto. Inak urobíme tamto. Teraz si to však musíme rozobrať detailnejšie.

Kľúčové slovo if

Ak chceme zapísať podmienku, tak nám na to poslúži kľúčové slovo if (písané malými písmenami). Tento výraz znamená v preklade „Keď, ak...“, čo nám napovedá, ako tento typ podmienky funguje. Slovo if nám hovorí, že pokiaľ je podmienka splnená, prebehne kus kódu, ktorý sa nachádza v bloku programu pod konkrétnou podmienkou.

Mali by ste vedieť:

Blok programu v C, C++ a teda aj Wiring je všetko, čo obsahujú zložené zátvorky {...}.

Skúsme si to pre lepšiu predstavu ukázať na príklade:

```
1  void setup()  //Prebehne len raz
2                //Nastavíme komunikačnú rýchlosť na 9600 baud za sekundu
3  {
4    Serial.begin(9600) ;
5  }
6  //Hlavný cyklus programu
7
8  void loop()
9  {
10 if(1 < 2)   // Pýtame sa, či je 1 menšia ako 2, vždy to je pravda takže prebehne blok nižšie
11           {
12             Serial.println("podmienka_platí") ; //Výpis do sériového monitoru
13           }
14 }
15 //Ak podmienku obrátime na (1 > 2), celý blok programu v zátvorke sa preskočí
```

Ukážka programu s podmienkou if

Pokiaľ by sme podmienku obrátili:

if (1 > 2) {...}, celý blok programu v zložených zátvorkách sa preskočí a pokračuje až za ním.

Teda vieme, že if môžeme použiť v prípade, keď treba o niečom rozhodnúť - niečo sa stalo a reagujeme. Môžeme ho tiež použiť, keď sa niečo nestane.

Dobrým príkladom môže byť, keď program vyzve používateľa, aby zadal kladné číslo, ktoré je potom porovnávané s číslom iným. Keď podmienku užívateľ nesplní a zadá napríklad číslo záporné, môžeme podmienku if použiť na to, aby užívateľovi „vynadala“ za jeho chybu a nám sa tak nezrútil celý program.

Kľúčové slovo else

Keď sa zamyslíme nad spracovaním podmienky s príkazom if, zistíme, že niečo chýba. Taká odpoveď na otázku: Čo keď to prvé je nesplnené?

Copyright © 2013 - 2026 Encyklopédia poznania

Jazyk Arduina ponúka riešenie aj tejto otázky a tým je kľúčové slovo else (môžeme ho preložiť ako „inak“). Prevedené do programátorskej reči:

```
if(niečo)
{
//ak je "niečo" pravda, vykoná sa tento kód
}

else  //inak
{
//"niečo" je zle, vykoná sa tento kód
}
```

Programová konštrukcia podmienky else

Pre lepšiu predstavu fungovania si ukážeme praktický príklad a trošku upravíme program, ktorý sme si ukazovali vyššie:

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

void loop()
{
  if(1 > 2) //jednoduchá podmienka, kde sa pýtame či 1 je väčšie ako 2
  {
    Serial.println("podmienka platí") ; //podmienka platí ---> vypíše sa text
  }

  else //prvá podmienka neplatí
  {
    Serial.println("podmienka neplatí") ; //vypíše sa toto
  }
}
```

Vylepšená programová konštrukcia podmienky else

Viac volieb else if

Aby sme mohli vytvoriť viac možností jednotlivých podmienok nám poslúži výraz else if. Takto môžeme nastaviť neobmedzene možností a tým sa pripraviť na rôzne scenáre, ktoré by mohli v priebehu programu nastať. Trochu upravíme náš predošlý program a ukážeme si, ako by sme mohli implementovať else if.

Copyright © 2013 - 2026 Encyklopédia poznania

```
int a = 1 ; //premenná, ktorú neskôr budeme porovnávať (celé číslo)

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

void loop()
{
  if(a > 3) //pýtame sa, či je "a" väčšie ako 3
  {
    Serial.println("premenná je väčšia ako 3") ; //vypíšeme text
  }
  else if(a == 1) //pýtame sa, či sa premenná rovná 1
  {
    Serial.println("premenná je rovná 1") ; //áno, je rovná jednej
  }
  else if(a == 2) //pýtame sa, či je premenná rovná 2
  {
    Serial.println("premenná je rovná 2") ;
  }
  else
  {
    Serial.println("premenná je asi 0 alebo záporná") ; //viac možností nemáme
  }
}
```

Programová konštrukcia podmienky else if

Operátory využívané pri podmienkach

Tu by som mohol rozobrať mnoho matematických teórií, ale nám postačí, že jednoduchá znalosť. Na každej množine čísel je definovaná relácia.

Teda porovnanie. Programátori tohto faktu často využívajú a v podmienkach najčastejšie.

Každý (bežný) programovací jazyk má k dispozícii takzvané **relačné operátory**, ktoré sú neoddeliteľnou súčasťou if výrazu.

Jeden relačný operátor (porovnanie: ==) sme využili náš kód vyššie. Predkladáme tu tabuľku s ďalšími príkladmi relačných operátorov, ktoré sa pri podmienkach tiež často používajú:

Operátor	Kompletný zápis	Význam
==	x==y	ak sa x rovná y
!=	x!=y	ak sa x nerovná y
<	x<y	ak je x menší ako y
>	x>y	ak je x väčšia ako y
<=	x<=y	ak je x menší alebo rovný y
>=	x>=y	ak je x väčší alebo rovný

Mali by ste vedieť:

Je potrebné odlišovať zápis x==y, kde porovnávame dve hodnoty a zápis x = y, kde hodnotu priraďujeme!

Podmienky a logické operátory

Vieme ako na to, keď je podmienka jedna, ale čo robiť v prípade, keď chceme, aby platilo viac podmienok naraz?

Aj pre túto situáciu tu existuje riešenie. Poslúži nám na to operátor &&, ktorý v slovenčine znamená „súčasne“ a ||, jeho význam je „alebo“. V anglickom jazyku AND a OR.

Mali by ste vedieť:

Klávesové skratky na napísanie logických operátorov: && = Ctrl + Alt + C, || = Ctrl + Alt + W.

Operátor	Kompletný zápis	Význam
&&	x and y, prípadne x && y	ak je x a súčasne y
||	x or y, prípadne x || y	ak je x alebo y

Copyright © 2013 - 2026 Encyklopédia poznania

Kľúčové slovo - switch

Ak máme podmienok viac, chceme začať nejakou voľbou a pri tom si udržať kód prehľadný, môžeme použiť sekvenciu switch. Ide o istú alternatívu k else if, ktorá je viac prehľadná. Funkcionalitou je switch dosť podobný else if.

Ukážeme si jednoduché využitie a celkový zápis switch:

```
int voľba = 1; //premenná s hodnotou 1

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

void loop()
{
  switch(voľba) //switch s dosadenou premennou "voľba" ---> hodnota premennej je 1, čo znamená, že prebehne case 1
  {
    case 1: // voľba číslo 1 - prebehne, ak premenná "voľba" bude rovná 1
      Serial.println("zvolená možnosť 1"); //výpis textu do sériového monitoru
      break ;                              //zabránime vyhodnoteniu ďalších volieb
    case 2:
      Serial.println("zvolená možnosť 2");
      break ;
    case 3: //voľba číslo 3
      Serial.println("zvolená možnosť 3");
      break ;
    default: // predvolená voľba - prebehne, ak nebude premenná rovná žiadnej z možností
      Serial.println("Žiadna zo zvolených možností"); //výpis textu do sériového monitoru
  }
}
```

Programová konštrukcia switch

Mali by ste vedieť:

Každý blok switch ide nahradiť blokom else if, záleží na vašej preferencii a na situačnej prehľadnosti programu.

Zdroje

Prevzaté a upravené z:

- **https://www.itnetwork.sk/hardver-pc/arduino/programovaci-jazyk/podmienky-a-ich-pouzitie.**

Copyright © 2013 - 2026 Encyklopédia poznania