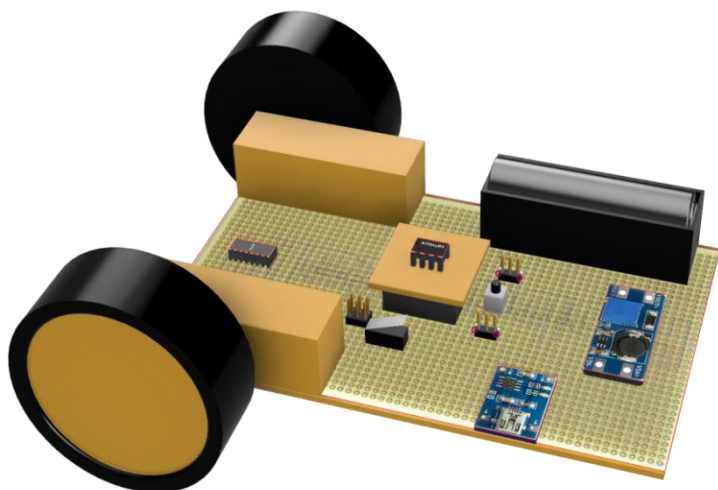


# Cing

Robot pre deti i dospelých

## Príručka 2.časť

Procesor Attiny84

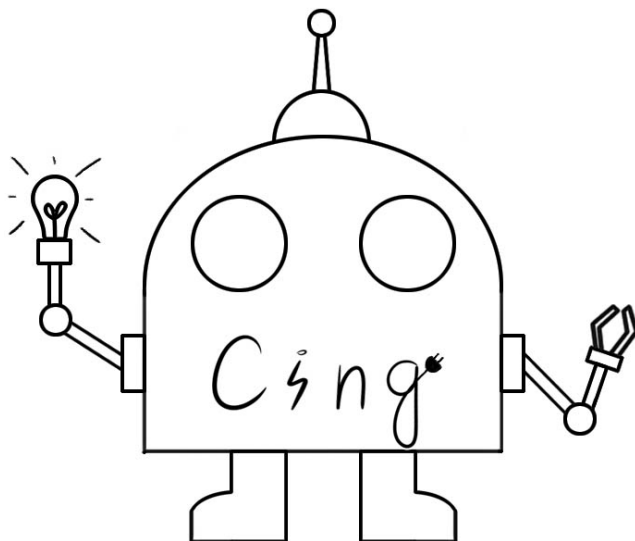


## ÚVOD

V prvej časti zo série robota Cing sme si ukázali najzákladnejšie pohyby robota, pohyby vpred a do strán a tiež sme si vyskúšali prácu z rôznymi senzormi a parametrami.

V tejto príručke si ukážeme, ako vylepšiť pohyby, vytvárať zložitejšie príkazy, vyskúšame si prácu s novými senzormi či komponentmi a naučíme sa, ako správne používať príkazy. Navyše ku kapitolám pribudnú úlohy, takže sa nemusíte obávať, že by toho bolo málo. Opäť sa budeme vzdelávať hrovou a zábavnou formou. Nemusíte sa báť, že to bude ťažké – práve naopak!

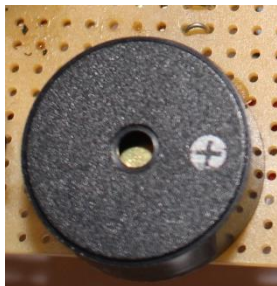
Vďaka poznatkom a vedomostiam z predchádzajúcej časti to bude ešte jednoduchšie!



## OBSAH

1.	ZOZNÁMME SA S NOVINKAMI .....	4
2.	POKRAČUJEME V POHYBE.....	5
❖	CHODENIE ROVNO A DOZADU .....	5
❖	SPOMAĽOVANIE A ZRÝCHĽOVANIE .....	6
3.	PREKÁŽKY .....	9
❖	OBCHÁDZKA PREKÁŽKY .....	9
❖	OBCHÁDZKA SO SLEDOVANÍM ČIARY.....	11
4.	OSVETLENIE.....	14
5.	TLAČIDLO.....	16
6.	BZUČIAK .....	18

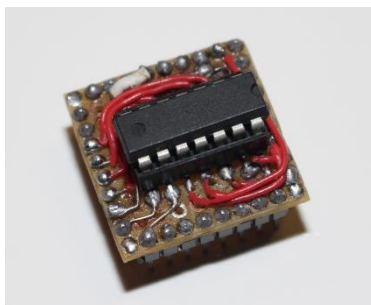
# 1. ZOZNÁMME SA S NOVINKAMI



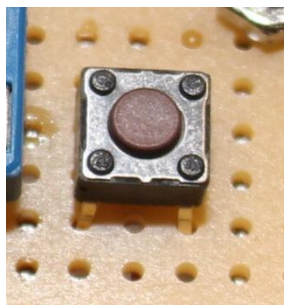
Bzučiak



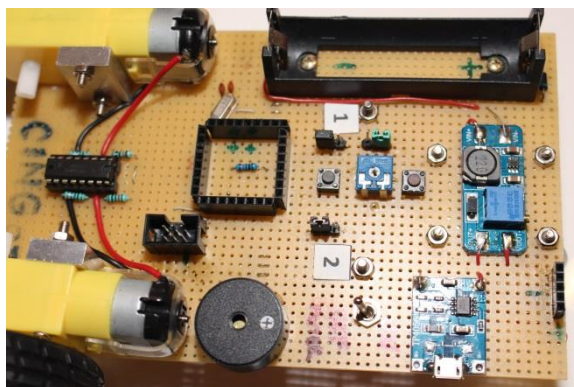
Model Cing Attiny84



Attiny84



Tlačidlo



Model Cing – Pohľad zhora

## 2. POKRAČUJEME V POHYBE

V predchádzajúcej časti sme sa zoznámili s programovacím prostredím robota Cing a naučili robota Cing pohnúť sa z miesta. S týmto procesorom však pribudla aj funkcia cúvanie. V tejto kapitole si ukážeme, ako naprogramovať Cinga tak, aby sa hýbal nielen dopredu, ale aj dozadu.

**Budeme potrebovať:**

- Robota Cing
- ICSP ASP programátor

### ❖ CHODENIE ROVNO A DOZADU

**Stratégia:**

Robot Cing sa bude pohybovať na oboch motoroch dopredu nejakú dobu, a potom cúvne naspäť dozadu.

**Program:**

```
1  #include "Attiny84_IO_basic.h"
2  Attiny attiny;
3  void setup() { }
4  void loop()
5  {
6    attiny.motor("AB", 1, "digital");
7    delay(1000);
8    attiny.motor("AB", -1, "digital");
9    delay(1000);
10 }
```

### **Vysvetlenie programu:**

V prvom a druhom riadku programu sú príkazy určené pre programovacie prostredie, ktoré implementujú knižnicu `<Attiny84_IO_basic.h>`.

Príkazy vo `void loop(){} v` svorkových zátvorkách:{} bežia v nekonečnom procese.

V šiestom riadku programu zapíname motory v portoch "AB" na 1000 milisekúnd (1 sekundu). Potom tieto motory zapneme na parameter -1, čo znamená, že motory v portoch "AB" pôjdu aj naďalej, avšak nie dopredu, ale dozadu. Teda na rozlíšenie smeru pohybu kolies používame znamienko -.

Ak ku parametru nedáme znamienko, znamená to, že sa robot bude hýbať dopredu.

## **❖ SPOMAĽOVANIE A ZRÝCHĽOVANIE**

### **Stratégia:**

Pri tomto programe využijeme znalosti z predchádzajúcej časti robota Cing. Robot Cing bude postupne zrýchľovať dopredu a potom to isté zopakuje aj smerom dozadu.

## Program:

---

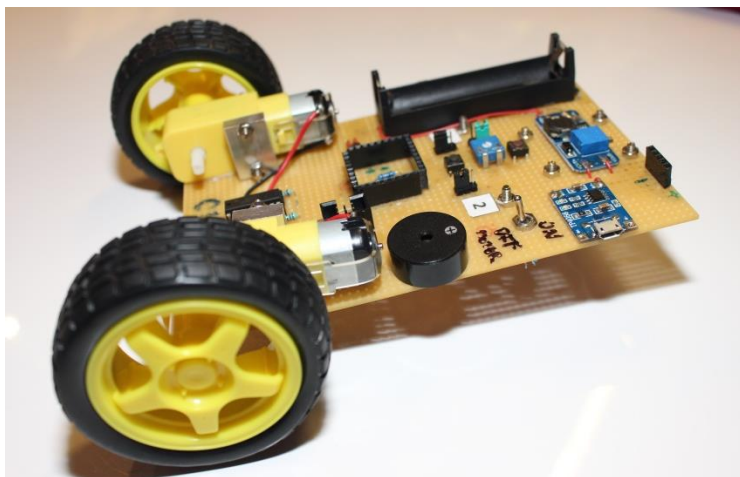
```
1  #include "Attiny84_IO_basic.h"
2  Attiny attiny;
3  void setup() {
4
5
6  }
7  void loop()
8  {
9      for (int x=0;x<100;x++)
10     {
11         attiny.motor("AB",x,"analog");
12         delay(10);
13     }
14     delay (1000);
15
16     for (int x=100;x>0;x--)
17     {
18         attiny.motor("AB",x,"analog");
19         delay(10);
20     }
21     for (int x=0;x<100;x++)
22     {
23         attiny.motor("AB",-x,"analog");
24         delay(10);
25     }
26     delay (1000);
27
28     for (int x=100;x>0;x--)
29     {
30         attiny.motor("AB",-x,"analog");
31         delay(10);
32     }
33 }
```

### Vysvetlenie programu:

Začiatok programu je rovnaký ako v minulej časti.

V deviatom riadku programu vytvárame počítadlo, pre ktoré platí, že hodnota  $x=0$ , a kým je  $x<100$ , tak po každom pohnutí sa robota sa ku  $x$  pripočíta 1. Znamená to teda, že robot bude postupne zrýchľovať z 1% na 100% na parametri "analog" a následne spomaľovať zo 100% na 0% na parametri "analog."

Program pokračuje s rovnakými príkazmi, avšak namiesto parametra  $x$  sme nastavili parameter  $-x$ . Znamená to teda, že robot Cing bude najprv zrýchľovať a spomaľovať smerom dopredu, a potom zrýchľovať a spomaľovať popri cúvaní.





### 3. PREKÁŽKY

V tejto kapitole si ukážeme, ako obísť prekážku dookola alebo po čiare.

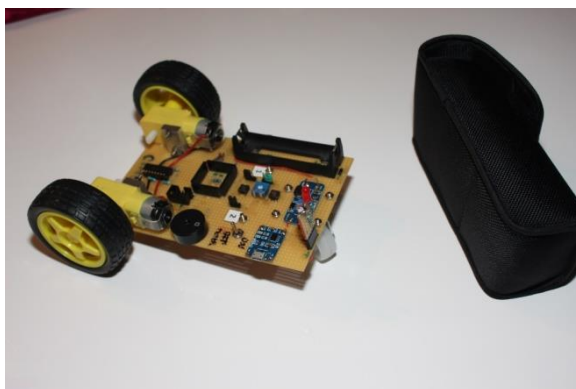
**Budeme potrebovať:**

- Robota Cing
- ICSP ASP programátor
- Svetelný senzor
- Ultrasonický senzor

#### ❖ OBCHÁDZKA PREKÁŽKY

**Stratégia:**

Robot Cing pôjde stále rovno na motoroch “AB”. Ak ultrasonickým senzorom zaznamená prekážku vo vzdialenosti menšej ako 20 cm, obíde ju podľa toho, ako mu to zvolíme v programe.



## Program:

---

```
1  #include "Attiny85_IO.h"
2  Attiny attiny;
3  void setup() {
4
5  }
6
7  void loop()
8  {
9      if (attiny.UltrasonicSensor()<20)
10     {
11         for (int x=0;x<4;x++)
12         {
13             attiny.motor("A", 1, "digital");
14             delay (500);
15             attiny.motor("A", 0, "digital");
16             attiny.motor("B", 1, "digital");
17             delay (500);
18             attiny.motor("AB", 1, "digital");
19             delay (500);
20         }
21     }
22     else
23     {
24         attiny.motor("AB", 1, "digital");
25     }
26 }
```

---

### **Vysvetlenie programu:**

V prvom a druhom riadku programu sú príkazy, ktoré implementujú knižnicu `<Attiny84_IO.h>`.

Príkazy vo `void loop(){}`  v svorkových zátvorkách: `{}` bežia v nekonečnom procese a opakujú sa.

V deviatom riadku programu overujeme hodnotu ultrasonického senzora.

Ak je prekážka vzdialená od robota na viac ako 20 cm, robot Cing pokračuje ďalej dopredu na motoroch "AB".

Ak je prekážka bližšie ako 20 cm, robot začne prekážku obchádzať tak, že sa najprv otočí doľava a potom doprava a následne pôjde na oboch motoroch dopredu. Tento proces zopakuje štyrikrát, pretože sme mu v jedenástom riadku programu zadali počítadlo, ktoré hovorí, že kým  $x < 4$  tak robot Cing bude tento proces opakovať. Tento proces zopakuje, kým bude mať  $x$  hodnoty:

$x = 0; x = 1; x = 2; x = 3$

Za vykonaním každého procesu pripočíta ku  $x$  1.

## **❖ OBCHÁDZKA SO SLEDOVANÍM ČIARY**

### **Stratégia:**

Robot Cing pôjde stále rovno na motoroch "AB". Ak ultrasonickým senzorom zaznamená prekážku vo vzdialenosti menšej ako 20 cm, obíde prekážku po čiare, ktorá je vedľa prekážky.

## Program:

---

```
1  #include "Attiny85_IO_basic.h"
2  Attiny attiny;
3  void setup() {
4
5
6  }
7
8  void loop()
9  {
10     if (attiny.UltrasonicSensor()<20)
11     {
12         attiny.motor("A", 1, "digital");
13         delay (500);
14         attiny.motor("A", 0, "digital");
15         attiny.motor("B", 1, "digital");
16         delay (500);
17         attiny.motor("AB", 1, "digital");
18         delay (500);
19     }
20     else
21     {
22         if (attiny.LightSensor(1, "digital")==0)
23         {
24             attiny.motor("B",1, "digital");
25         }
26         else
27         {
28             attiny.motor("A", 1, "digital");
29         }
30     }
31
32 }
~~
```

### Vysvetlenie programu:

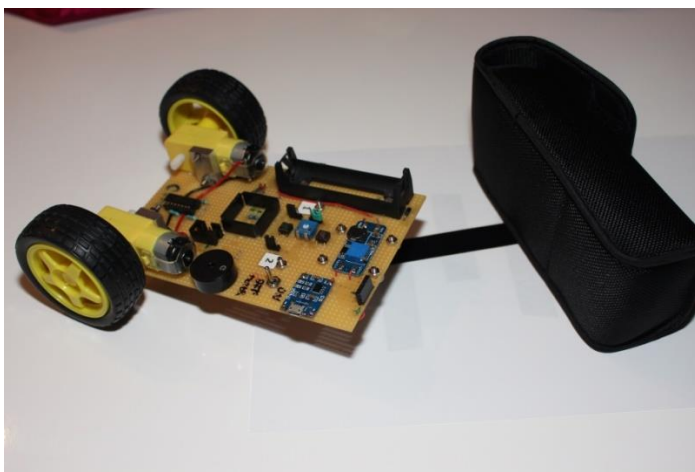
V prvom a druhom riadku programu sú príkazy, ktoré implementujú knižnicu `<Attiny85_IO.h>`.

Príkazy vo `void loop(){}`  v svorkových zátvorkách: `{}` bežia v nekonečnom procese a opakujú sa.

V deviatom overujeme hodnotu ultrasonického senzora.

Ak je jeho hodnota menšia ako `20`, robot sa zastaví.

Ak je hodnota väčšia, robot pokračuje, pričom sleduje aj čiaru.



## 4. OSVETLENIE

V tejto kapitole si ukážeme, ako obísť prekážku dookola alebo po čiare.

### Budeme potrebovať:

- Robota Cing
- ICSP ASP programátor
- Senzor osvetlenia
- Svetelný senzor

### Stratégia:

Ak senzor osvetlenia ovetlíme na viac ako 20%, robot zastane. Ak ho neosvetlíme, robot bude nasledovať čiaru.

### Program:

```
1  #include "Attiny85_IO_basic.h"
2  Attiny attiny;
3  #define ciara 0
4  void setup() {}
5  void loop()
6  {
7    if (attiny.ShineSensor()<20)
8    {
9      attiny.motor("AB", 0, "digital");
10   }
11   else
12   {
13     if (attiny.LightSensor(2, "digital")==ciara)
14     {
15       attiny.motor("B", 1, "digital");
16     }
17     else
18     {
19       attiny.motor("A", 1, "digital");
20     }
21   }
22 }
```

### Vysvetlenie programu:

V prvom a druhom riadku programu sú príkazy, ktoré implementujú knižnicu `<Attiny84_IO.h>`.

Príkazy vo `void loop(){}` v svorkových zátvorkách:{} bežia v nekonečnom procese a opakujú sa.

V treťom riadku riadku zadávame, že ak v programe zadáme slovo “`ciara`”, priradí mu hodnotu 0.

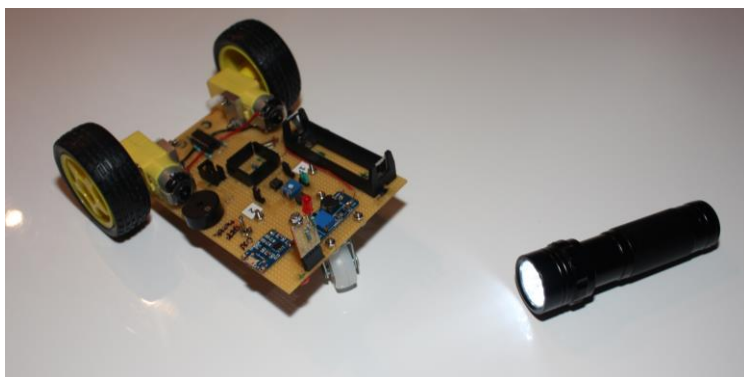
V siedmom riadku overujeme podmienku senzoru osvetlenia.

Ak je jeho hodnota viac ako 20, robot Cing sa zastaví na oboch motoroch.

Ak je jeho hodnota menšia ako 20, robot Cing pôjde ďalej po čiare.

V trinástom riadku overujeme podmienku pre svetelný senzor, avšak neporovnávame ju priamo s hodnotou 0, ale porovnávame ju so slovom `ciara`, ktoré ma hodnotu 0.

Takto si v programoch vieme definovať pojmy a ich hodnoty.



## 5. TLAČIDLO

V tejto kapitole si ukážeme, ako obísť prekážku dookola alebo po čiare.

### Budeme potrebovať:

- Robota Cing
- ICSP ASP programátor

### Stratégia:

Robot Cing pôjde pôjde dopredu a doprava, pokiaľ bude tlačidlo stisnuté. Ak tlačidlo stlačené nebude, robot bude stáť.

### Program:

```
1  #include "Attiny84_IO_basic.h"
2  Attiny attiny;
3  void setup() {}
4  void loop()
5  {
6      if (attiny.Button()==1)
7      {
8          while (true)
9          {
10             attiny.motor("AB", 1, "digital");
11             delay (1000);
12             attiny.motor("AB", 0, "digital");
13             attiny.motor("A", 1, "digital");
14             delay (1000);
15         }
16     }
17     else
18     {
19         attiny.motor("AB", 0, "digital");
20     }
21 }
```



### Vysvetlenie programu:

V prvom a druhom riadku programu sú príkazy, ktoré implementujú knižnicu `<Attiny84_IO.h>`.

Príkazy vo `void loop(){}`  v svorkových zátvorkách: `{}` bežia v nekonečnom procese a opakujú sa.

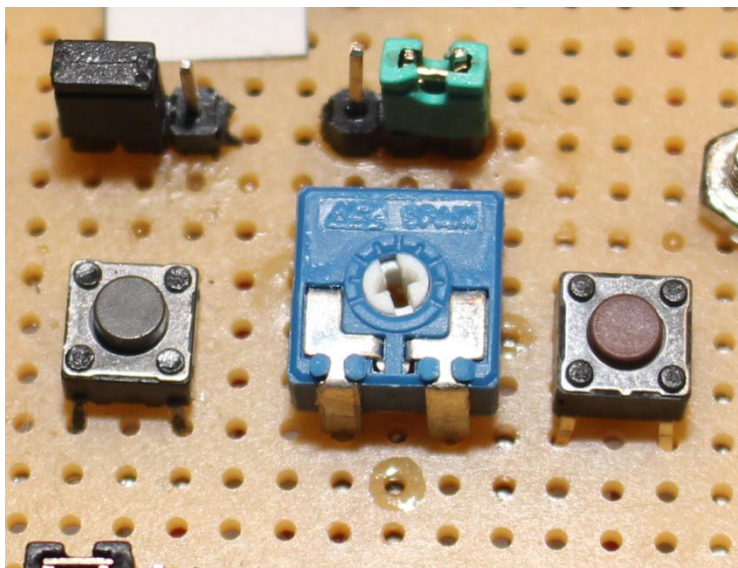
V šiestom riadku programu overujeme, či je tlačidlo zapnuté alebo vypnuté.

Vo ôsmom riadku máme podmienku `while(true)`, ktorá hovorí, že kým bude podmienka v `if` platiť, bude vykonávať súbor príkazov v zátvorke.

Ak podmienka platiť nebude, vykoná sa príkaz v zátvorke `else`.

Program teda hovorí, že pokiaľ bude tlačidlo stlačené, robot Cing pôjde dopredu a doprava.

Ak tlačidlo stlačené nebude, Cing bude stáť.



## 6. BZUČIAK

V tejto kapitole si ukážeme, ako obísť prekážku dookola alebo po čiare.

### Budeme potrebovať:

- Robota Cing
- ICSP ASP programátor
- Bzučiak

### Stratégia:

Bzučika bude sekundu zapnutý, a potom sekundu vypnutý.

### Program:

```
1  #include <Attiny84_IO_basic.h>
2  Attiny attiny;
3  void setup() {
4  }
5
6  void loop() {
7      attiny.Buzzer(1);
8      delay(1000);
9      attiny.Buzzer(0);
10     delay(1000);
11 }
```

### Vysvetlenie programu:

V prvom a druhom riadku programu sú príkazy, ktoré implementujú knižnicu `<Attiny84_IO.h>`.

Príkazy vo `void loop(){}`  v svorkových zátvorkách:{} bežia v nekonečnom procese a opakujú sa.

V siedmom riadku hovoríme, že zapneme bzučiak na sekundu. Potom sa bzučiak na ekundu vypne.

