

Bevezetés az Arduino és az elektronika világába II.

Agócs Norbert és Nagy Dániel

2020. szeptember 8.

Tartalom

- 1 Ismétlés
- 2 Fogalomak: Áramkör és Rövidzár
- 3 Fizika: A gomb
- 4 Programozási alapok: If
- 5 Feladatok
- 6 Arduino alapok: digitalRead
- 7 Űrhajó műszerfal

Tartalom

- 1 Ismétlés
- 2 Fogalmak: Áramkör és Rövidzár
- 3 Fizika: A gomb
- 4 Programozási alapok: If
- 5 Feladatok
- 6 Arduino alapok: digitalRead
- 7 Űrhajó műszerfal

Mit tanultunk eddig?

Hardware:

- Ardunio alaplapp
- Breadboard
- LED
- Ellenállás

Software

- Változó létrehozása
- Alapműveletek ($=$, $+$, $-$, $/$, $*$)
- Alapvető parancsok
 - `Serial.begin(9600)`
 - `Serial.println(" ")`
 - `pinMode()`
 - `digitalWrite()`
 - `delay()`

Ismétlő feladat: Villógó LED

Készítsünk egy LED-et, ami egyre gyorsabban kezd el villogni.

Ismétlő feladat: Villógó LED

Készítsünk egy LED-et, ami egyre gyorsabban kezd el villogni.

- LED

Ismétlő feladat: Villógó LED

Készítsünk egy LED-et, ami egyre gyorsabban kezd el villogni.

- LED
- Ellenállás (100 - 500 Ω)

Ismétlő feladat: Villógó LED

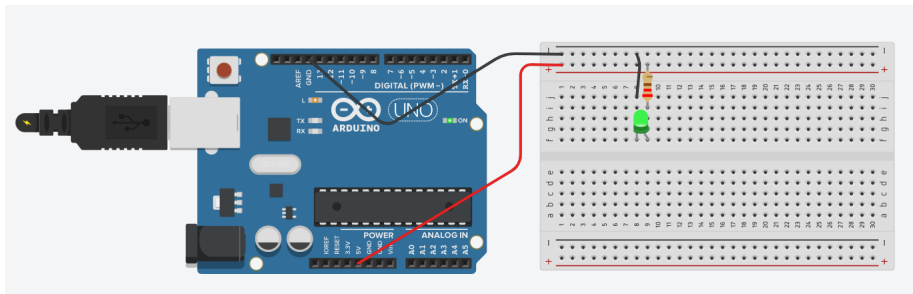
Készítsünk egy LED-et, ami egyre gyorsabban kezd el villogni.

- LED
- Ellenállás ($100 - 500 \Omega$)
- Kábelek

Ismétlő feladat: Villógó LED

Készítsünk egy LED-et, ami egyre gyorsabban kezd el villogni.

- LED
- Ellenállás (100 - 500 Ω)
- Kábelek



Megoldás

- Segítség: Sima villogtatás, csak a várakozási idővel kell játszani.

Megoldás

- Segítség: Sima villogtatás, csak a várakozási idővel kell játszani.
- Újdonság: Minden ciklus végén csökkentjük a vározkodási időt

Megoldás

- Segítség: Sima villogtatás, csak a várakozási idővel kell játszani.
- Újdonság: Minden ciklus végén csökkentjük a várakozási időt

```
int led = 7;
float ido = 1000;

void setup()
{
    pinMode(led, OUTPUT);
}

void loop()
{
    digitalWrite(led, HIGH);
    delay(250);
    digitalWrite(led, LOW);
    delay(ido);
    ido = ido/1.2; //varakozasi ido csokkentese
}
```

Tartalom

- 1 Ismétlés
- 2 Fogalomak: Áramkör és Rövidzár
- 3 Fizika: A gomb
- 4 Programozási alapok: If
- 5 Feladatok
- 6 Arduino alapok: digitalRead
- 7 Űrhajó műszerfal

Mi az az áramkör?

Áramkör

Mi az az áramkör?

Áramkör

Egy áramforrás vezetékekkel összekapcsolva egy fogyasztóval.

Mi az az áramkör?

Áramkör

Egy áramforrás vezetékekkel összekapcsolva egy fogyasztóval.

Fogyasztó

Mi az az áramkör?

Áramkör

Egy áramforrás vezetékekkel összekapcsolva egy fogyasztóval.

Fogyasztó

Minden, ami áram hatására csinál valamit (világít, hangot ad ki, ...).

Mi az az áramkör?

Áramkör

Egy áramforrás vezetékekkel összekapcsolva egy fogyasztóval.

Fogyasztó

Minden, ami áram hatására csinál valamit (világít, hangot ad ki, ...).

Rövidzárlat

Mi az az áramkör?

Áramkör

Egy áramforrás vezetékekkel összekapcsolva egy fogyasztóval.

Fogyasztó

Minden, ami áram hatására csinál valamit (világít, hangot ad ki, ...).

Rövidzárlat

Nagyon kis ellenállású összekötetés két áramköri elem között.

- Az áram mindig a legkisebb ellenállású úton halad
- A rövidzárlat nagyon veszélyes!

Mi az az áramkör?

Áramkör

Egy áramforrás vezetékekkel összekapcsolva egy fogyasztóval.

Fogyasztó

Minden, ami áram hatására csinál valamit (világít, hangot ad ki, ...).

Rövidzárlat

Nagyon kis ellenállású összekötetés két áramköri elem között.

- Az áram mindig a legkisebb ellenállású úton halad
- A rövidzárlat nagyon veszélyes!

Tartalom

- 1 Ismétlés
- 2 Fogalmak: Áramkör és Rövidzár
- 3 Fizika: A gomb**
- 4 Programozási alapok: If
- 5 Feladatok
- 6 Arduino alapok: digitalRead
- 7 Űrhajó műszerfal

A nyomógomb

Áramköri szerepe: Fizikailag megszakítja az áramkört, tehát nem tud folyni az áram az adott áramkörben, amíg nincs lenyomva.



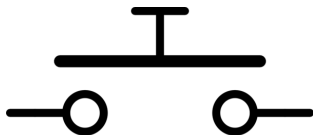
(a) A gomb, amit találhattok az iskolai arduino készletekben

A nyomógomb

Áramköri szerepe: Fizikailag megszakítja az áramkört, tehát nem tud folyni az áram az adott áramkörben, amíg nincs lenyomva.



(a) A gomb, amit találhattok az iskolai arduino készletekben



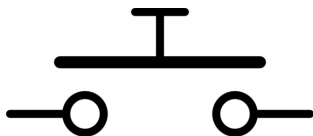
(b) A gomb sematikus ábrája

A nyomógomb

Áramköri szerepe: Fizikailag megszakítja az áramkört, tehát nem tud folyni az áram az adott áramkörben, amíg nincs lenyomva.



(a) A gomb, amit találhattok az iskolai arduino készletekben



(b) A gomb sematikus ábrája

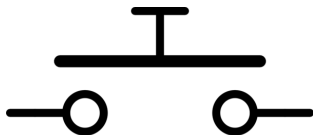
Mire tudjuk mi felhasználni?

A nyomógomb

Áramköri szerepe: Fizikailag megszakítja az áramkört, tehát nem tud folyni az áram az adott áramkörben, amíg nincs lenyomva.



(a) A gomb, amit találhattok az iskolai arduino készletekben



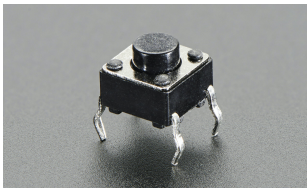
(b) A gomb sematikus ábrája

Mire tudjuk mi felhasználni?

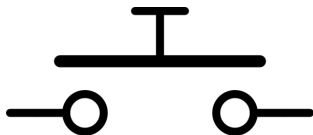
- Folyamatok elindítása gombnyomással

A nyomógomb

Áramköri szerepe: Fizikailag megszakítja az áramkört, tehát nem tud folyni az áram az adott áramkörben, amíg nincs lenyomva.



(a) A gomb, amit találhattok az iskolai arduino készletekben



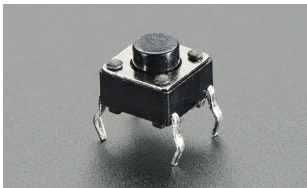
(b) A gomb sematikus ábrája

Mire tudjuk mi felhasználni?

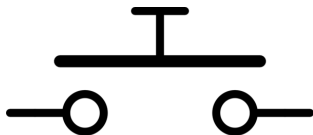
- Folyamatok elindítása gombnyomással
- Állapotok változtatása gombnyomással

A nyomógomb

Áramköri szerepe: Fizikailag megszakítja az áramkört, tehát nem tud folyni az áram az adott áramkörben, amíg nincs lenyomva.



(a) A gomb, amit találhattok az iskolai arduino készletekben



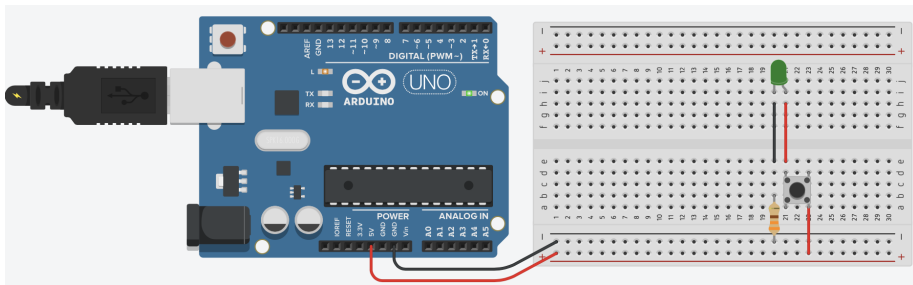
(b) A gomb sematikus ábrája

Mire tudjuk mi felhasználni?

- Folyamatok elindítása gombnyomással
- Állapotok változtatása gombnyomással

Ezek azok a lehetőségei, amikhez kell a programozás ...

Csináljunk gombnyomásra felvillanó LED-et!



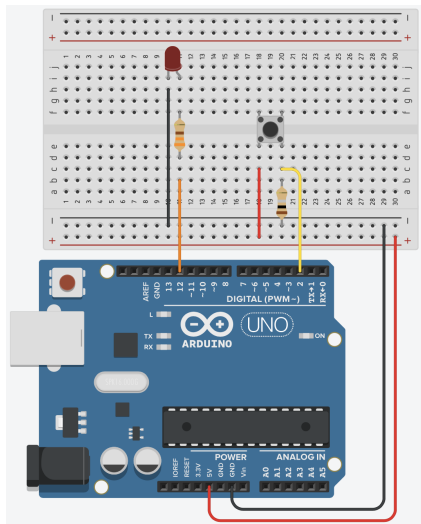
Fontos:

- Az ellenállás legyen 100Ω és 500Ω között.
- Ha nem világít a LED akkor fordítva kötöttük be.

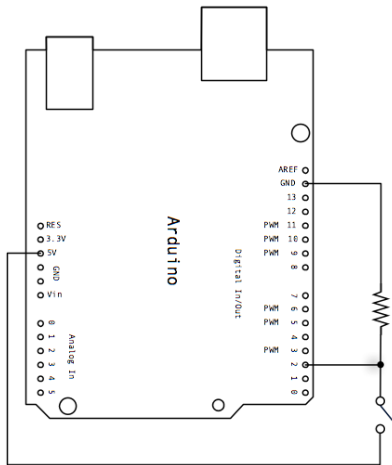
Csináljunk gombnyomásra villogó LED-et!

Mi a célunk?

- Ne csak akkor világítson a LED, ha nyomom a gombot.
- Gombnyomásra különféle akciókat tudjunk végrehajtani bármilyen eszközzel.
- Mi kell ehhez?
 - A kapcsolás megértése
 - `If ()`
 - `digitalRead()`



A gomb bekötése az áramkörbe



Miért így kötjük be?

- Az arduino pinjeivel csak feszültséget tudunk mérni
 - A feszültség méréséhez, kell egy "segéd" áramkör
 - Ezt az áramkört fogjuk a gombbal megszakítani
- Az áramkör a következő módon áll össze:
 - 0V - Ellenállás - Gomb - 5V
 - Az ellenállás a rövidzár elkerülése miatt kell
- Ha a gomb be van kapcsolva, akkor folyik az áram, és tudunk mérni a pinen.

Tartalom

- 1 Ismétlés
- 2 Fogalmak: Áramkör és Rövidzár
- 3 Fizika: A gomb
- 4 Programozási alapok: If
- 5 Feladatok
- 6 Arduino alapok: digitalRead
- 7 Űrhajó műszerfal

Bevezetés: If-else parancs

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float osztando = 10; float oszto = 5; //változók  
    if(oszto == 0) {  
        Serial.println("Nullával_nem_lehet_osztani!");  
    }  
    else {  
        float hanyados = osztando/oszto;  
        Serial.println("A_hányados:_", hanyados);  
    }  
}
```


Bevezetés: If-else parancs

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float osztando = 10; float oszto = 5; //változók  
    if(oszto == 0) {  
        Serial.println("Nullával_nem_lehet_osztani!");  
    }  
    else {  
        float hanyados = osztando/oszto;  
        Serial.println("A_hányados:_", hanyados);  
    }  
}
```

if(feltétel) {...} else {...}

Bevezetés: If-else parancs

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float osztando = 10; float oszto = 5; //változók  
    if(oszto == 0) {  
        Serial.println("Nullával_nem_lehet_osztani!");  
    }  
    else {  
        float hanyados = osztando/oszto;  
        Serial.println("A_hányados: ", hanyados);  
    }  
}
```

if(feltétel) {...} else {...}

- Ha **igaz** a feltétel, akkor le fut az if parancs {...} közötti része.

Bevezetés: If-else parancs

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float osztando = 10; float osztó = 5; //változók  
    if(osztó == 0) {  
        Serial.println("Nullával_nem_lehet_osztani!");  
    }  
    else {  
        float hanyados = osztando/osztó;  
        Serial.println("A_hányados: ", hanyados);  
    }  
}
```

if(feltétel) {...} else {...}

- Ha **igaz** a feltétel, akkor le fut az if parancs {...} közötti része.
- Ha **hamis** a feltétel nem fut le az if parancs {...} közötti része, csak else része, ha van.

Összehasonlító operátorok

Művelet	Operátor	Példa	Eredmény
Egyenlő	==	5==8	false
Nem egyenlő	!=	5!=8	true
Kisebb	<	5<8	true
Nagyobb	>	5>8	false
Kisebb egyenlő	<=	5<=8	true
Nagyobb egyenlő	>=	5>=8	false
És	&&	5<=8 && 8>10	false
Vagy		5<=8 8>10	true
Nem	!	!(5 < 10)	false

Összehasonlító operátorok

Művelet	Operátor	Példa	Eredmény
Egyenlő	==	5==8	false
Nem egyenlő	!=	5!=8	true
Kisebb	<	5<8	true
Nagyobb	>	5>8	false
Kisebb egyenlő	<=	5<=8	true
Nagyobb egyenlő	>=	5>=8	false
És	&&	5<=8 && 8>10	false
Vagy		5<=8 8>10	true
Nem	!	!(5 < 10)	false

Logikai operátorok:

Összehasonlító operátorok

Művelet	Operátor	Példa	Eredmény
Egyenlő	<code>==</code>	<code>5==8</code>	false
Nem egyenlő	<code>!=</code>	<code>5!=8</code>	true
Kisebb	<code><</code>	<code>5<8</code>	true
Nagyobb	<code>></code>	<code>5>8</code>	false
Kisebb egyenlő	<code><=</code>	<code>5<=8</code>	true
Nagyobb egyenlő	<code>>=</code>	<code>5>=8</code>	false
És	<code>&&</code>	<code>5<=8 && 8>10</code>	false
Vagy	<code> </code>	<code>5<=8 8>10</code>	true
Nem	<code>!</code>	<code>!(5 < 10)</code>	false

Logikai operátorok:

- **ÉS:** Mindkét feltételnek igaznak kell lennie

Összehasonlító operátorok

Művelet	Operátor	Példa	Eredmény
Egyenlő	<code>==</code>	<code>5==8</code>	false
Nem egyenlő	<code>!=</code>	<code>5!=8</code>	true
Kisebb	<code><</code>	<code>5<8</code>	true
Nagyobb	<code>></code>	<code>5>8</code>	false
Kisebb egyenlő	<code><=</code>	<code>5<=8</code>	true
Nagyobb egyenlő	<code>>=</code>	<code>5>=8</code>	false
És	<code>&&</code>	<code>5<=8 && 8>10</code>	false
Vagy	<code> </code>	<code>5<=8 8>10</code>	true
Nem	<code>!</code>	<code>!(5 < 10)</code>	false

Logikai operátorok:

- **ÉS:** Mindkét feltételnek igaznak kell lennie
- **VAGY:** Az egyik feltételnek kell igaznak lennie

Összehasonlító operátorok

Művelet	Operátor	Példa	Eredmény
Egyenlő	==	5==8	false
Nem egyenlő	!=	5!=8	true
Kisebb	<	5<8	true
Nagyobb	>	5>8	false
Kisebb egyenlő	<=	5<=8	true
Nagyobb egyenlő	>=	5>=8	false
És	&&	5<=8 && 8>10	false
Vagy		5<=8 8>10	true
Nem	!	!(5 < 10)	false

Logikai operátorok:

- ÉS: Mindkét feltételnek igaznak kell lennie
- VAGY: Az egyik feltételnek kell igaznak lennie
- NEM: Az ellentetjére változtatja az eredményt

Tartalom

- 1 Ismétlés
- 2 Fogalmak: Áramkör és Rövidzár
- 3 Fizika: A gomb
- 4 Programozási alapok: If
- 5 Feladatok**
- 6 Arduino alapok: digitalRead
- 7 Űrhajó műszerfal

Mit ír ki a kód?

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float a = 3;  
    int b = 5;  
    if(a != b); {  
        Serial.println("Ok");  
    }  
    else {  
        Serial.println("Nem_ok!");  
    }  
}
```

Mit ír ki a kód?

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float a = 3;  
    int b = 5;  
    if(a != b); {  
        Serial.println("Ok");  
    }  
    else {  
        Serial.println("Nem_ok!");  
    }  
}
```

Megoldás: Nem ír ki semmit.

Mit ír ki a kód?

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float Galambos = 1;  
    int dolgozat = 5;  
    if(Galambos == dolgozat) {  
        Serial.println("Hha_itt_bajok_vannak!");  
    }  
    else {  
        Serial.println("Huh,_ezt_megúsztuk!");  
    }  
}
```

Mit ír ki a kód?

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    float Galambos = 1;  
    int dolgozat = 5;  
    if(Galambos == dolgozat) {  
        Serial.println("Hha_itt_bajok_vannak!");  
    }  
    else {  
        Serial.println("Huh,_ezt_megusztuk!");  
    }  
}
```

Megoldás: Huh, ezt megúsztuk!

Mit ír ki a kód?

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    int alma = 3;  
    int korte = 5;  
    int szilva = 6;  
    if(alma * szilva - korte <= alma) {  
        Serial.println("Ok");  
    }  
    else {  
        Serial.println("Nem_ok!");  
    }  
}
```

Mit ír ki a kód?

```
void setup() {  
    Serial.begin(9600); //Kommunikáció  
}  
  
void loop() {  
    int alma = 3;  
    int korte = 5;  
    int szilva = 6;  
    if(alma * szilva - korte <= alma) {  
        Serial.println("Ok");  
    }  
    else {  
        Serial.println("Nem_ok!");  
    }  
}
```

Megoldás: Nem ok!

Tartalom

- 1 Ismétlés
- 2 Fogalomak: Áramkör és Rövidzár
- 3 Fizika: A gomb
- 4 Programozási alapok: If
- 5 Feladatok
- 6 Arduino alapok: digitalRead**
- 7 Űrhajó műszerfal

A beolvasás függvénye

A függvény: `digitalRead(pin)`

- Beolvassa egy pin állapotát
- Lehetséges értékek: HIGH vagy LOW

Mire kell ez nekünk?

- Állapot (HIGH, LOW) beolvasása
- Felhasználói bemenetek érzékelése (pl.: gomb, kapcsoló)
- Visszajelzés (pl.: Van-e feszültség a LED-en?)

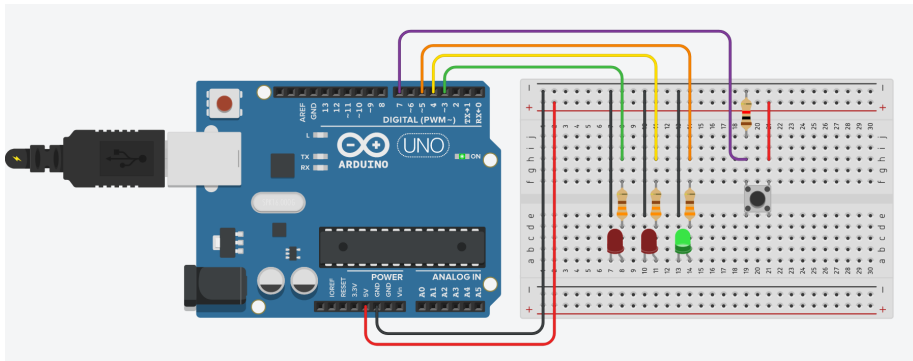
Tartalom

- 1 Ismétlés
- 2 Fogalomak: Áramkör és Rövidzár
- 3 Fizika: A gomb
- 4 Programozási alapok: If
- 5 Feladatok
- 6 Arduino alapok: digitalRead
- 7 Űrhajó műszerfal

Készítsük el egy úrhajó műszerfalának egy részét!

Alkatrészek

- 3db LED
- 3db 220 Ω -os ellenállás
- 1db Gomb
- 1db 1k Ω -os ellenállás



A kód I.

Használjuk az újonnan tanult parancsot!

- If() és else
- digitalRead()

```
int led_piros1 = 3;
int led_piros2 = 4;
int led_zold = 5;
int gomb = 7;

int gomb_allapot = 0;

void setup()
{
  pinMode(led_piros1, OUTPUT);
  pinMode(led_piros2, OUTPUT);
  pinMode(led_zold, OUTPUT);
  pinMode(gomb, INPUT);
}
```

A kód II.

```
void loop(){
  gomb_allapot = digitalRead(gomb);

  if(gomb_allapot == LOW) {
    digitalWrite(led_piros1, LOW);
    digitalWrite(led_piros2, LOW);
    digitalWrite(led_zold, HIGH);
  }
  else {
    digitalWrite(led_piros1, HIGH);
    digitalWrite(led_piros2, LOW);
    digitalWrite(led_zold, LOW);

    delay(250);

    digitalWrite(led_piros1, LOW);
    digitalWrite(led_piros2, HIGH);

    delay(250);
  }
}
```