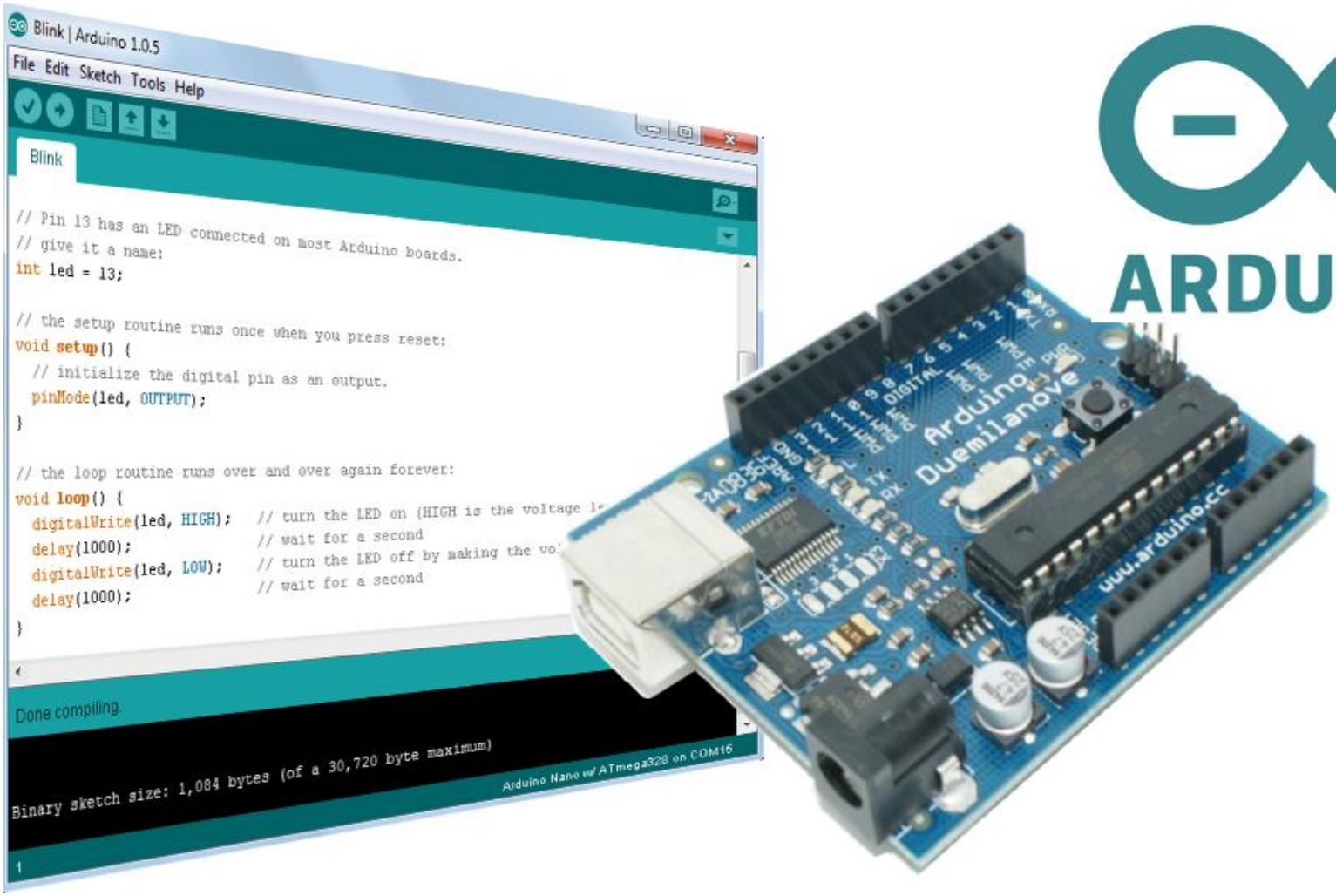


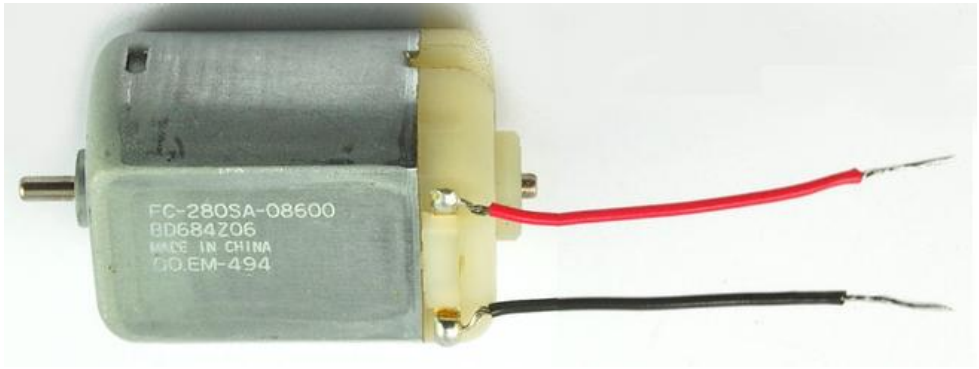
Bevezetés a mikrovezérlők programozásába:

DC motorok vezérlése

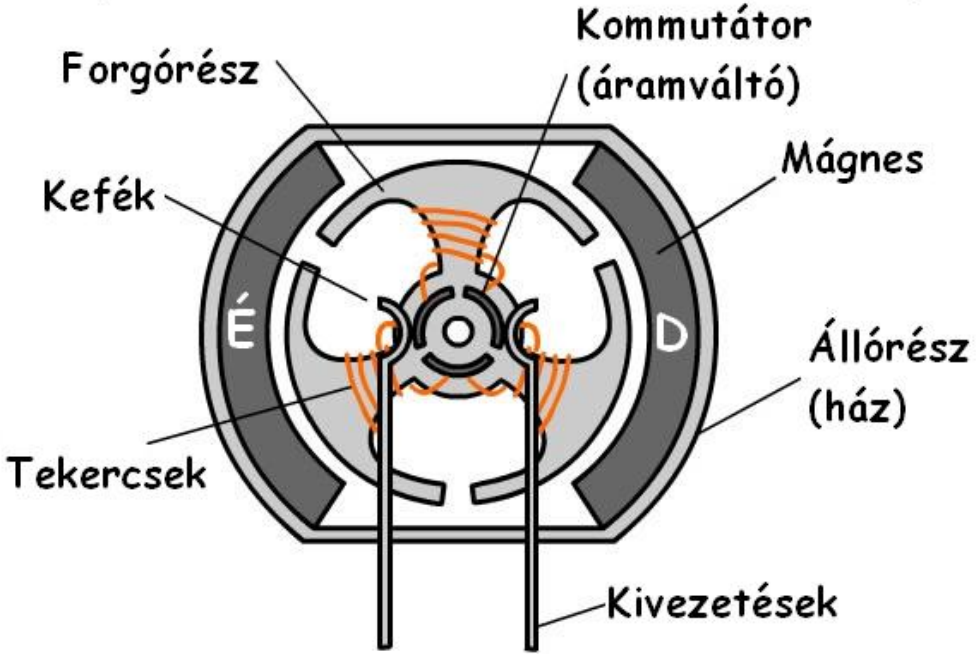
Lab 18 projektek

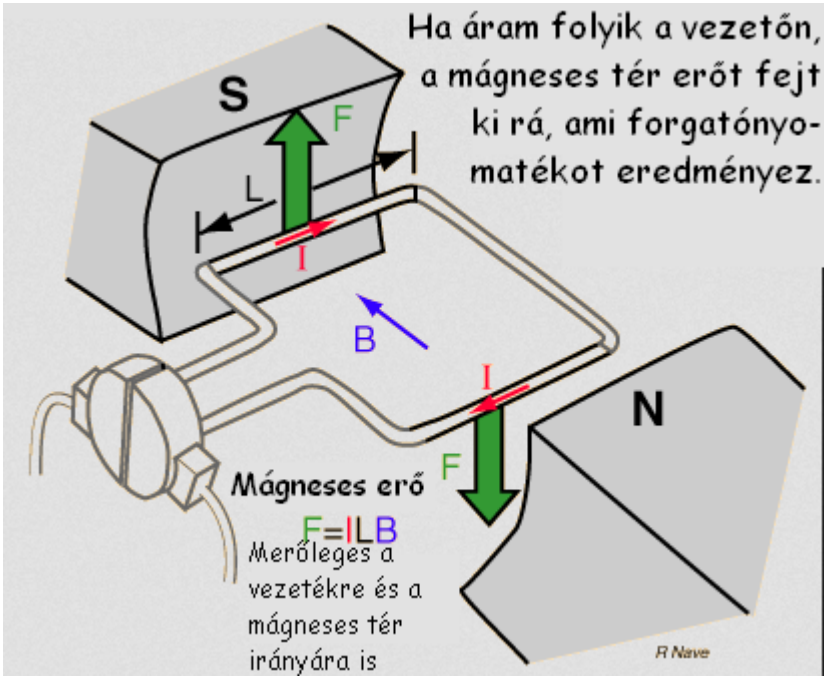
- **L293D_test_1M.ino** – tesztprogram egy motor vezérléséhez
- **L293D_test_2M.ino** – tesztprogram két motor vezérléséhez
- **L293D_test2_2M.ino** – tesztprogram két motor egyszerűsített vezérléséhez

Kisteljesítményű DC motorok



Tipikus kefés motor metszeti képe

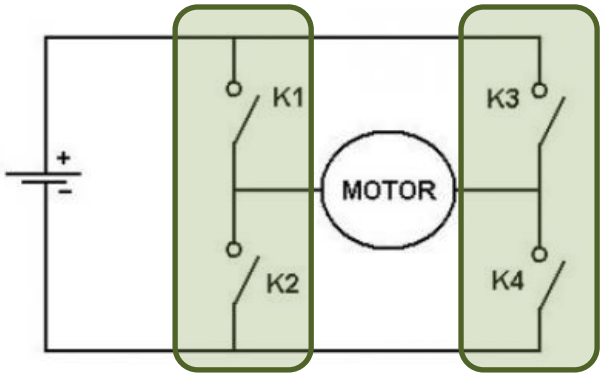




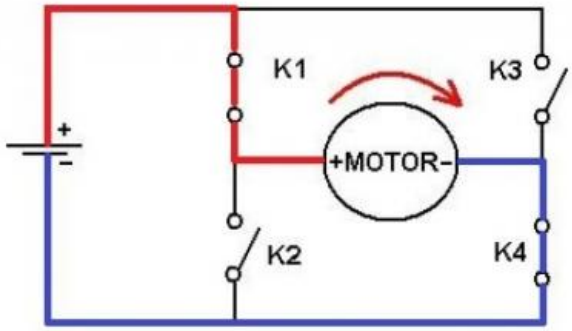
Feszültség: 3 V – 6V
Áram: 0.4 A – 0.6 A
Forgásirány váltás: polaritásváltással

Motorvezérlés: H-híd

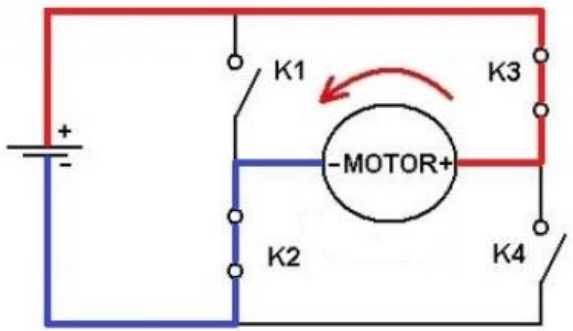
Ha a motor forgásirányát meg akarjuk változtatni, fel kell cserélni a polaritást. Ezt négy kapcsolóval tudjuk megoldani. Az elrendezést H-hídnak nevezzük.



Fél híd **A** Fél híd



B

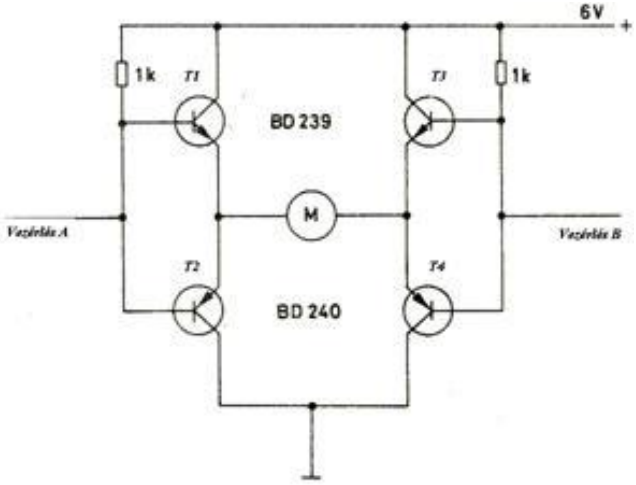


C

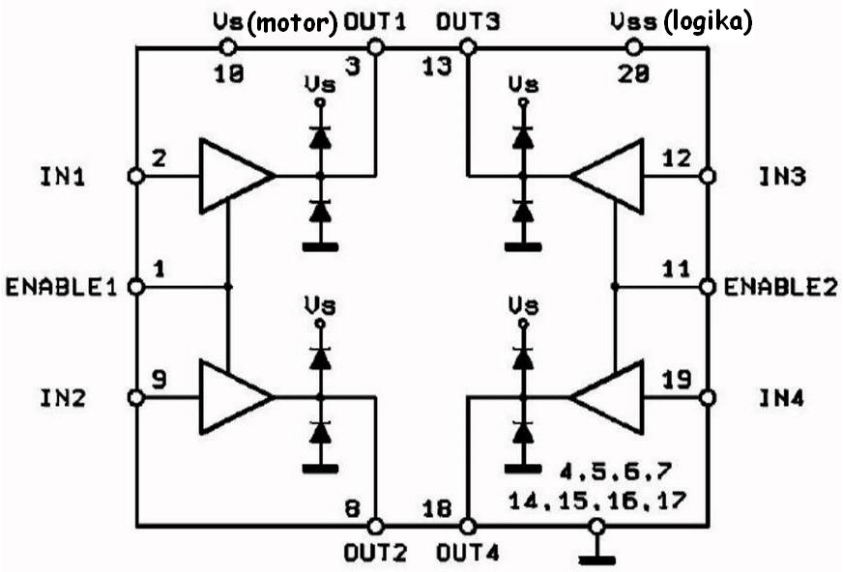
Forrás és leírás: hobbirobot.hu/content/vonalkoveto-i-robotika-kezdoknek

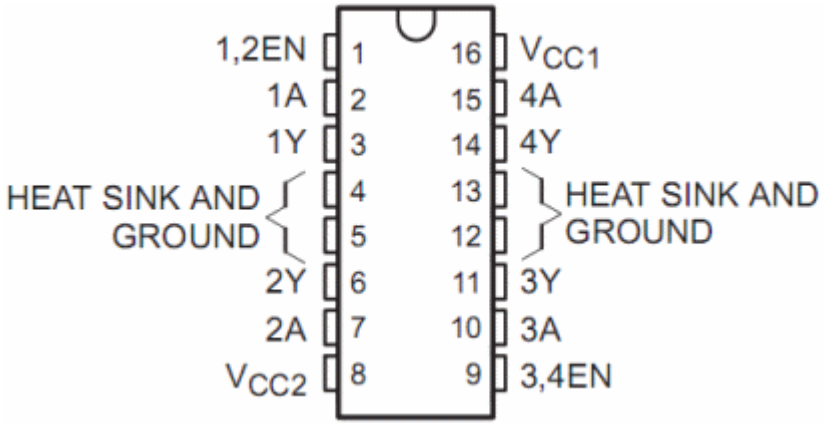
K1 és **K4** zárásakor a motor az egyik irányba forog.
K2 és **K3** zárásakor az ellenkező irányba forog.

Elektronikus vezérlés: tranzisztorokkal
Mivel **K1** és **K2,** s hasonlóan **K3** és **K4** sohasem lehet egyidejűleg zárva, elegendő félhidanként egy-egy vezérlőjel, mely a kapcsolókat ellenütemben zárja.



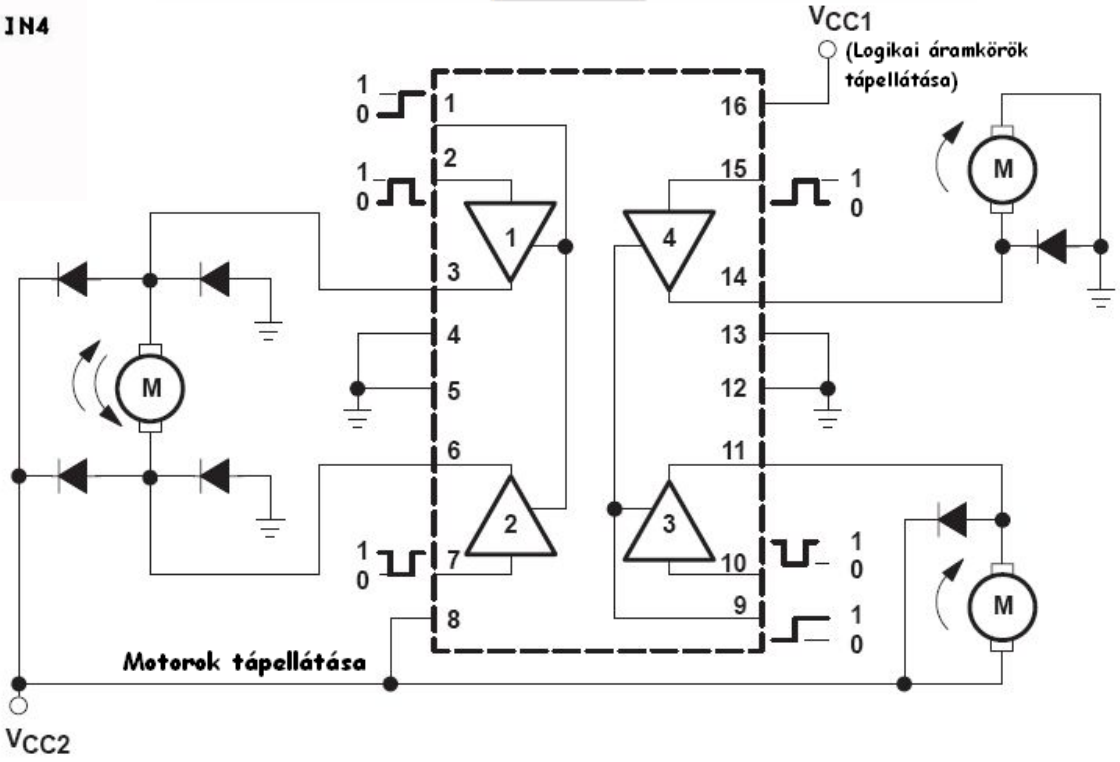
L293D 4db félhíd, vagy 2db H-híd





Félhíd: egyirányú forgás vezérlésére (pl. ventillátorok)
H-híd: polaritásváltást igénylő vezérlésekhez

V_{CC} = 4,5 – 36 V
V_A, V_{EN} = 2.3 – 7 V
I_{max} = 0,6 A
I_{peak} = 1,2 A (100 µs)

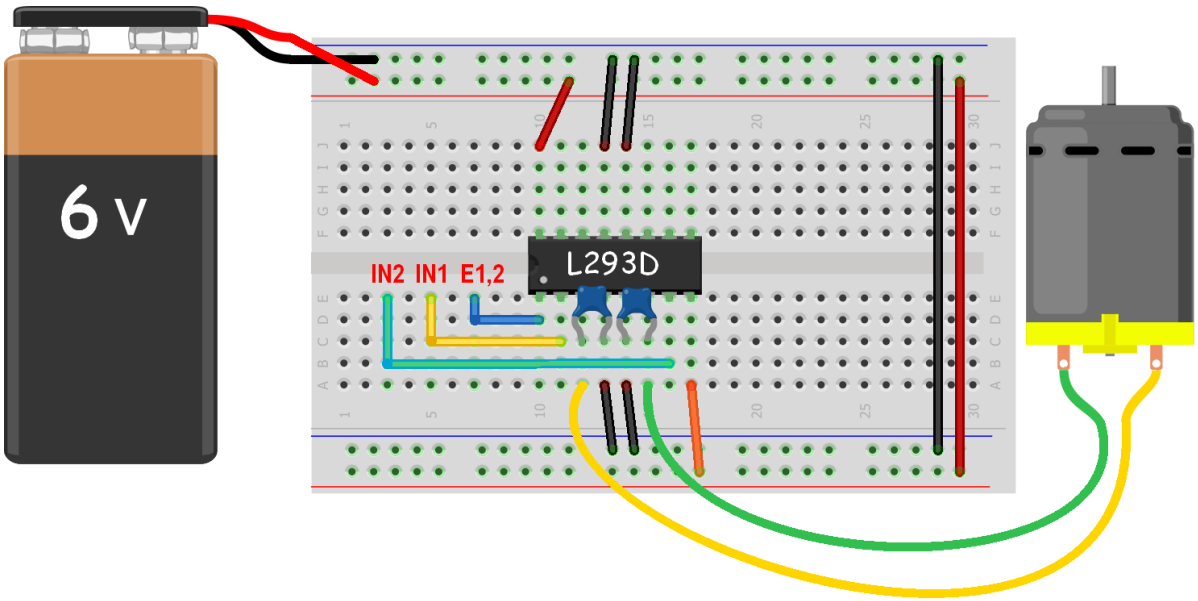


Egy motor vezérlése

Enable1,2	IN1	IN2	Motor
H	0	0	Stop
H	1	0	Előremenet
H	0	1	Hátramenet
H	1	1	Stop
L	X	X	Stop

E1,2: Letiltja vagy engedélyezi az 1. H-hidat

In1 és **In2** a forgás irányát adja meg.
Csak az **1,0** illetve **0,1** kombináció esetén van forgás.



Egyszerűsítési lehetőségek:
(ha sokalljuk a kivezetéseket)

1. Ha **E1,2** állandóan magas szintet kap, **In1** és **In2** önmagában elég a vezérléshez.
2. Ha **In2** = $\overline{\textbf{In1}}$ (inverter, IC-vel vagy tranzisztorral) akkor **E1,2** és **In1** elegendő a vezérléshez.

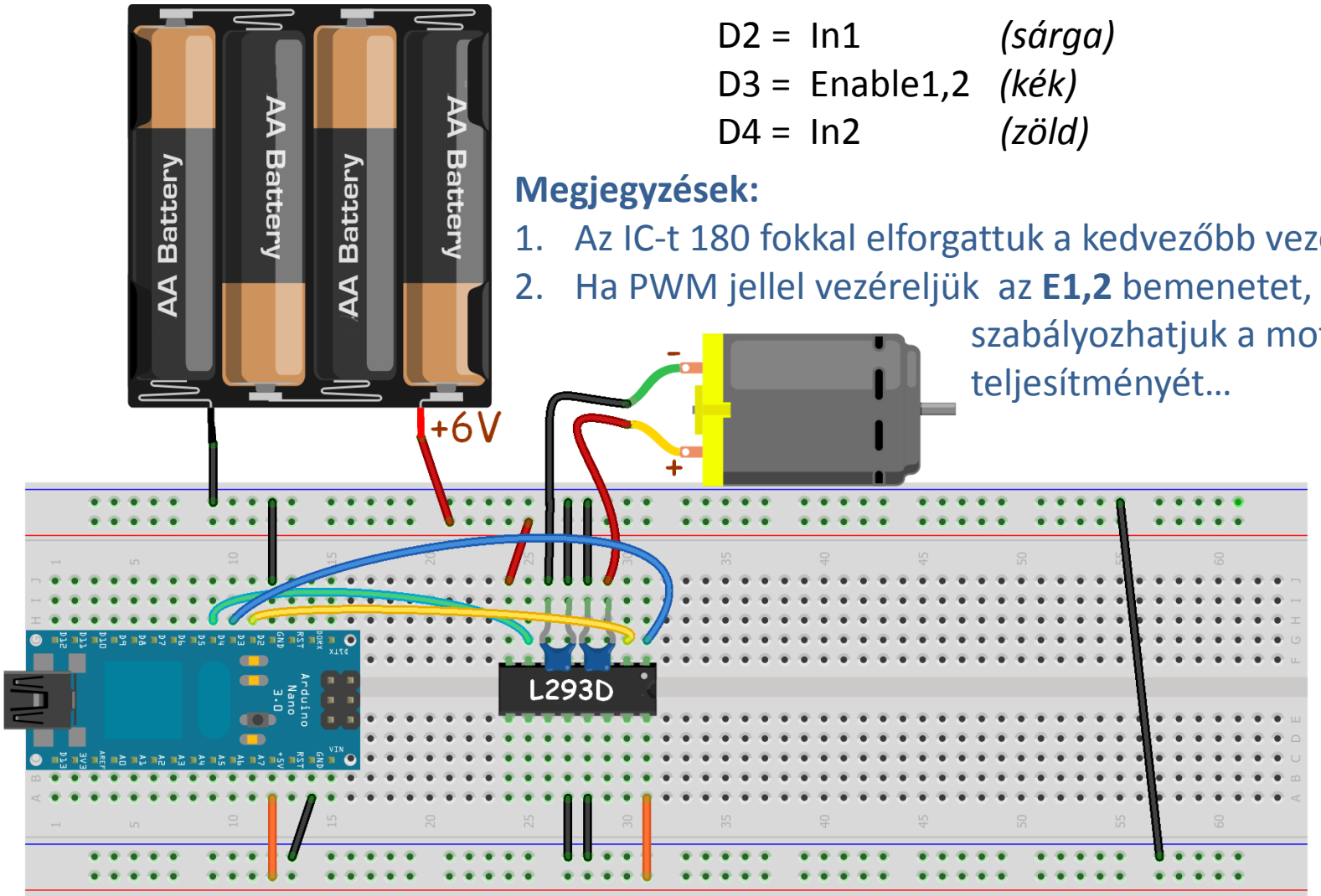
Egy motor vezérlése Arduinoval

Bekötési vázlat:

D2 = In1 *(sárga)*
D3 = Enable1,2 *(kék)*
D4 = In2 *(zöld)*

Megjegyzések:

1. Az IC-t 180 fokkal elforgattuk a kedvezőbb vezetékezéshez!
2. Ha PWM jellel vezéreljük az **E1,2** bemenetet, akkor szabályozhatjuk a motor teljesítményét…



Az **Arduino** itt az USB csatlakozón keresztül kap tápfeszültséget!

Made with Fritzing.org

L293D_test_1M.ino

Egy motor vezérlése. A végtelen ciklusban 5s várakozás után 2s előremenet, 1s várakozás, majd 2s hátramenet ismétlődik. Az előre- és hátramenet 75 %-os teljesítménnyel történik.

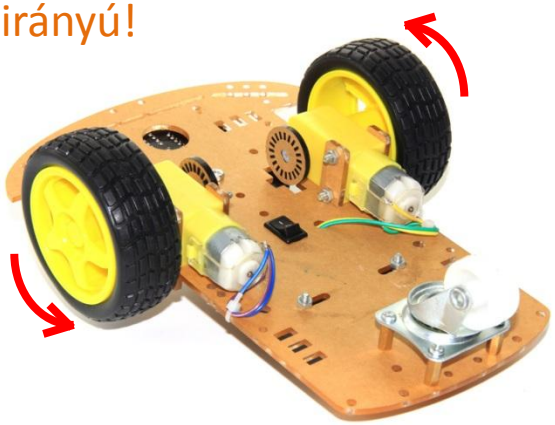
```
#define PWMA    3     //L293D pin1
#define D1A     2     //L293D pin2
#define D2A     4     //L293D pin7


void setMotor(int speed, boolean reverse)
{
  analogWrite(PWMA, speed);
  digitalWrite(D1A, !reverse);
  digitalWrite(D2A, reverse);
}


void setup()
{
  Serial.begin(9600);
  pinMode(PWMA, OUTPUT);
  pinMode(D1A, OUTPUT);
  pinMode(D2A, OUTPUT);
  Serial.println("L293D test program");
}
```

```
void loop()
{
  delay(5000);
  Serial.println("move forward 75%");
  setMotor(196, 0); // Forward 75%
  delay(2000);
  setMotor(0, 0);   // Stop motor
  delay(1000);
  Serial.println("move reverse 75%");
  setMotor(196, 1); // Reverse 75%
  delay(2000);
  setMotor(0, 0);   // Stop motor
}
```

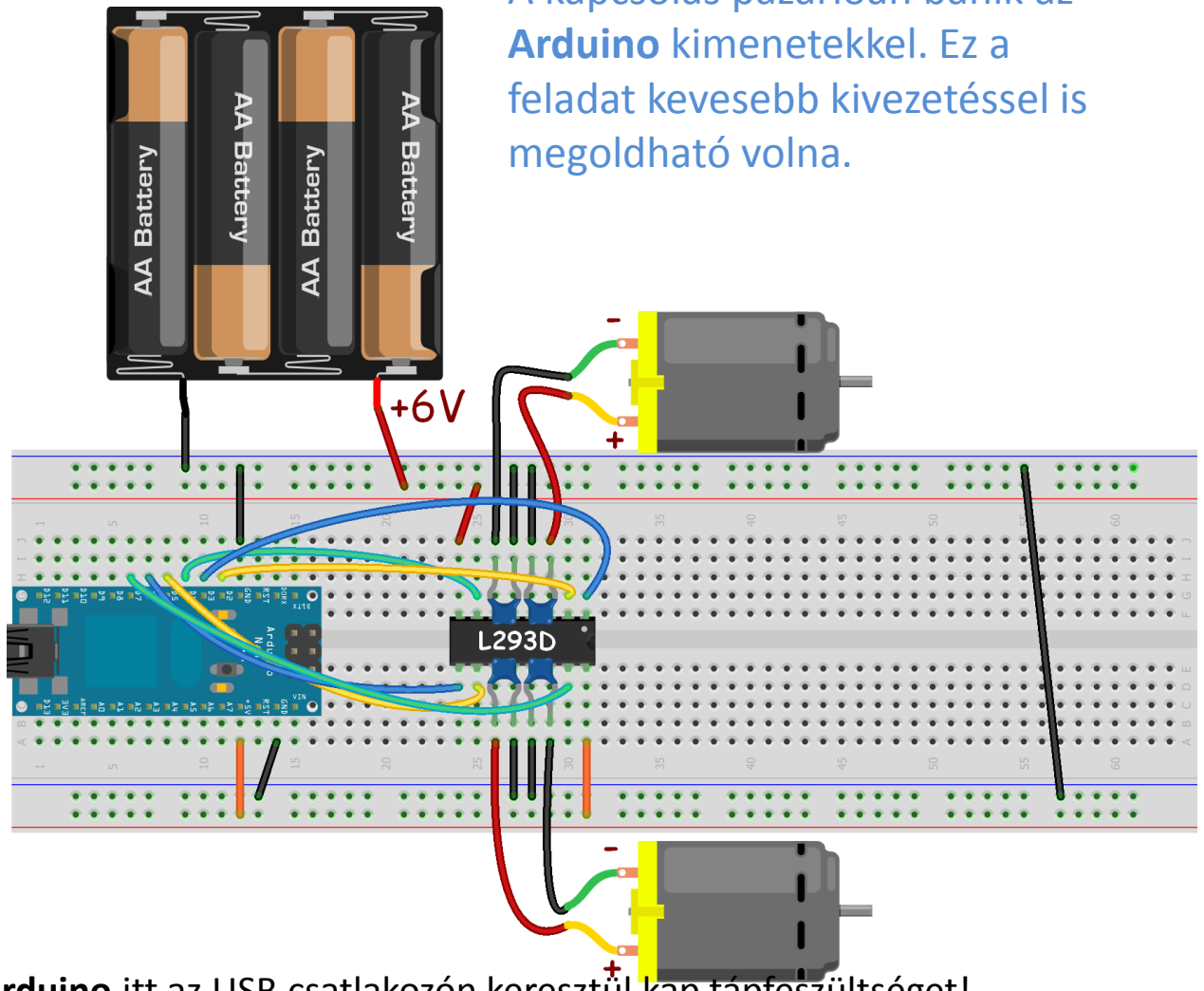
Két motor vezérlése Arduinoval

Megjegyzés:
A rajzon látható kapcsolásban azonos vezérlés mellett a két motor azonos irányba forog.

A fényképen látható független jobb és bal kerék meghajtású kocsihoz azonban az egyik motor polaritását fel kell cserélni!
Egyenesvonalú mozgásnál a két motor forgása ellentétes irányú!



A kapcsolás pazarlóan bánik az **Arduino** kimenetekkel. Ez a feladat kevesebb kivezetéssel is megoldható volna.



Az **Arduino** itt az USB csatlakozón keresztül kap tápfeszültséget!

Made with Fritzing.org

L293D_test_2M.ino

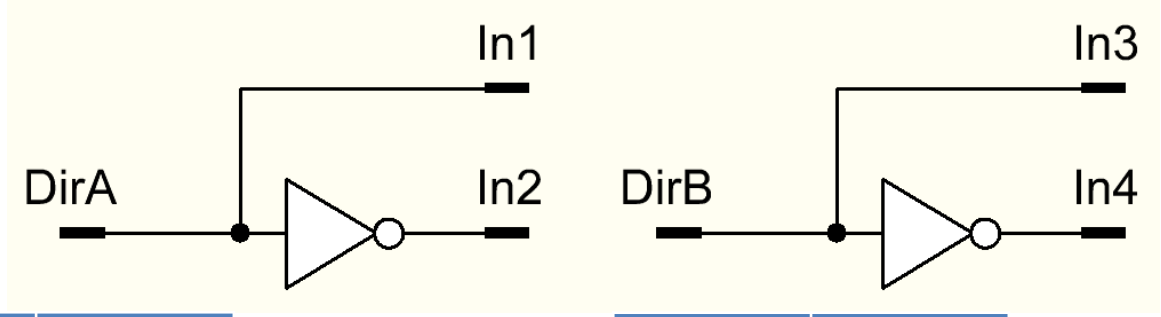
Két motor vezérlése. A végtelen ciklusban 5s várakozás után 2s előremenet, 1s várakozás, majd 2s hátramenet ismétlődik. Az előre- és hátramenet 75 %-os teljesítménnyel történik.

```
#define PWMA    3     //L293D pin1
#define D1A     2     //L293D pin2
#define D2A     4     //L293D pin7
#define PWMB    6     //L293D pin9
#define D3B     5     //L293D pin10
#define D4B     7     //L293D pin15

void setup() {
  Serial.begin(9600);
  pinMode(PWMA, OUTPUT);
  pinMode(D1A, OUTPUT);
  pinMode(D2A, OUTPUT);
  pinMode(PWMB, OUTPUT);
  pinMode(D3B, OUTPUT);
  pinMode(D4B, OUTPUT);
  Serial.println("L293D test program");
}
```

```
void setMotor(int speed, boolean
reverse) {
  analogWrite(PWMA, speed);
  analogWrite(PWMB, speed);
  digitalWrite(D1A, !reverse);
  digitalWrite(D2A, reverse);
  digitalWrite(D3B, !reverse);
  digitalWrite(D4B, reverse);
}

void loop() {
  delay(5000);
  Serial.println("move forward 75%");
  setMotor(196, 0); // Forward 75%
  delay(2000);
  setMotor(0, 0);   // Stop motor
  delay(1000);
  Serial.println("move reverse 75%");
  setMotor(196, 1); // Reverse 75%
  delay(2000);
  setMotor(0, 0);   // Stop motor
}
```

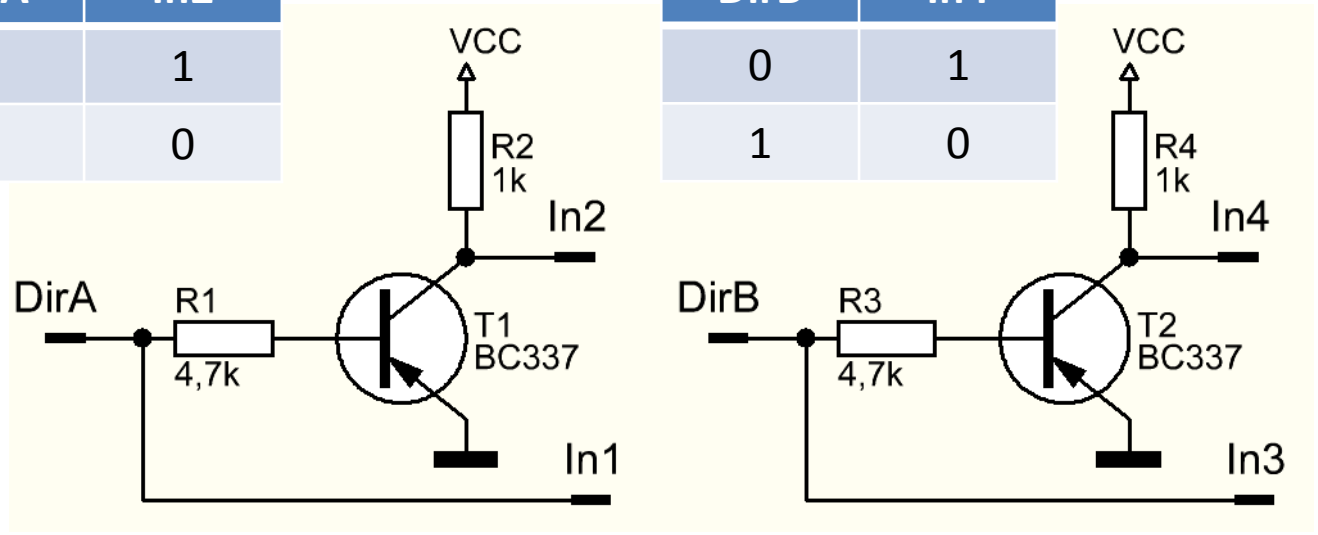
Egyszerűbb forgásirány-váltás

Vegyük észre, hogy **In2** mindig **In1** inverze (az előző programban `D2A = !D1A`)!
Hasonlóan **In4** is mindig **In3** inverze (az előző programban `D4B = !D3B`).
Ezeket az összefüggéseket elektronikusan is megvalósíthatjuk (logikai kapukkal, vagy tranzisztorokkal), s akkor kevesebb **Arduino** kivezetésre lesz szükségünk.



DirA	In2
0	1
1	0

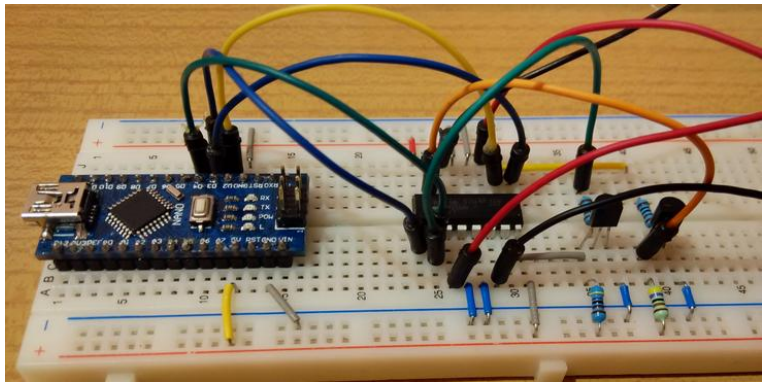
DirB	In4
0	1
1	0



Egyszerűbb forgásirány-váltás

Megjegyzés: Ebben a példában a motorok teljesítményét nem lehet függetlenül szabályozni (**PWMA = PWMB**), de ez nem mindig jó döntés!

Általában motoronként egy irányváltó és egy teljesítmény-szabályozó bemenetet szokás használni.



Az **Arduino** itt még az USB csatlakozón keresztül kap tápfeszültséget, de kipróbáláskor a **Vin** bemenetén kell táplálni (vagy a motor +6V-járól, vagy egy 9 V-os elemről)

Made with Fritzing.org

L293D_test2_2M.ino

Két motor egyszerűsített vezérlése. A végtelen ciklusban 5s várakozás után 2s előremenet, 1s várakozás, majd 2s hátramenet ismétlődik. Az előre- és hátramenet 75 %-os teljesítmény-nyel történik.

```
#define PWMA    3     //L293D pin1,pin9
#define DIRA    2     //L293D pin2
#define DIRB    4     //L293D pin10


void setMotor(int speed, boolean reverse)
{
  analogWrite(PWMA, speed);
  digitalWrite(DIRA, !reverse);
  digitalWrite(DIRB, reverse);
}


void setup()
{
  Serial.begin(9600);
  pinMode(PWMA, OUTPUT);
  pinMode(DIRA, OUTPUT);
  pinMode(DIRB, OUTPUT);
  Serial.println("L293D test program");
}
```

Itt szoftveresen fordítottuk meg az egyik motor forgásirányát!

```
void loop()
{
  delay(5000);
  Serial.println("move forward 75%");
  setMotor(196, 0); // Forward 75%
  delay(2000);
  setMotor(0, 0);   // Stop motor
  delay(1000);
  Serial.println("move reverse 75%");
  setMotor(196, 1); // Reverse 75%
  delay(2000);
  setMotor(0, 0);   // Stop motor
}
```

Megjegyzés:
Vegyük észre, hogy a kivezetések önkényes átnevezésétől eltekintve ez a program teljes mértékben megegyezik a **L293D_test_1M.ino** programmal!

Tervek, tapasztalatok

- Azonos vezérlés esetén a motorok nem egyformán „húznak”
  - Ez a motoronkénti független teljesítmény-szabályozást indokolja
- A jármű nem tartja az irányt
  - Szenzorok (giroszkóp, iránytű, forgásérzékelők) beépítése szükséges
  - Algoritmust kell kidolgozni a szenzorok által ellenőrzött mozgáskoordinációhoz
- Mozgásvezérlés lehetőségei
  - Távvezérlés (infra, Bluetoth vagy WiFi)
  - Vonalkövető robot (optoszenzorok)
  - Akadálykikerülő robot (ultrahangos vagy optikai távolságérzékelő)

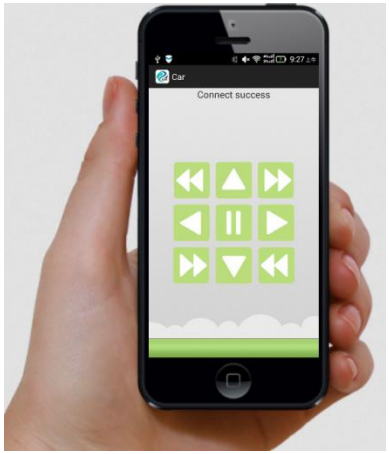
Egy távvezérlésre kidolgozott példa:

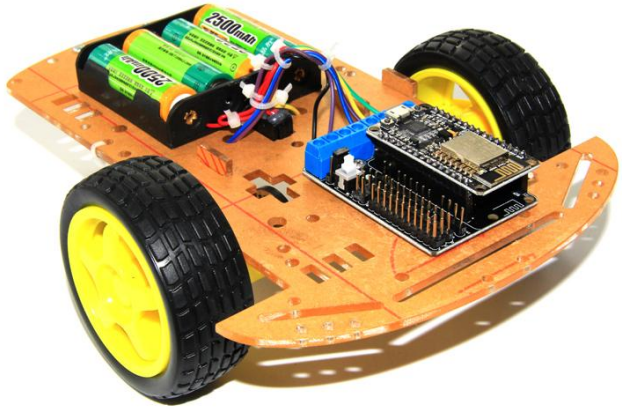
Do it Yourself WiFi car

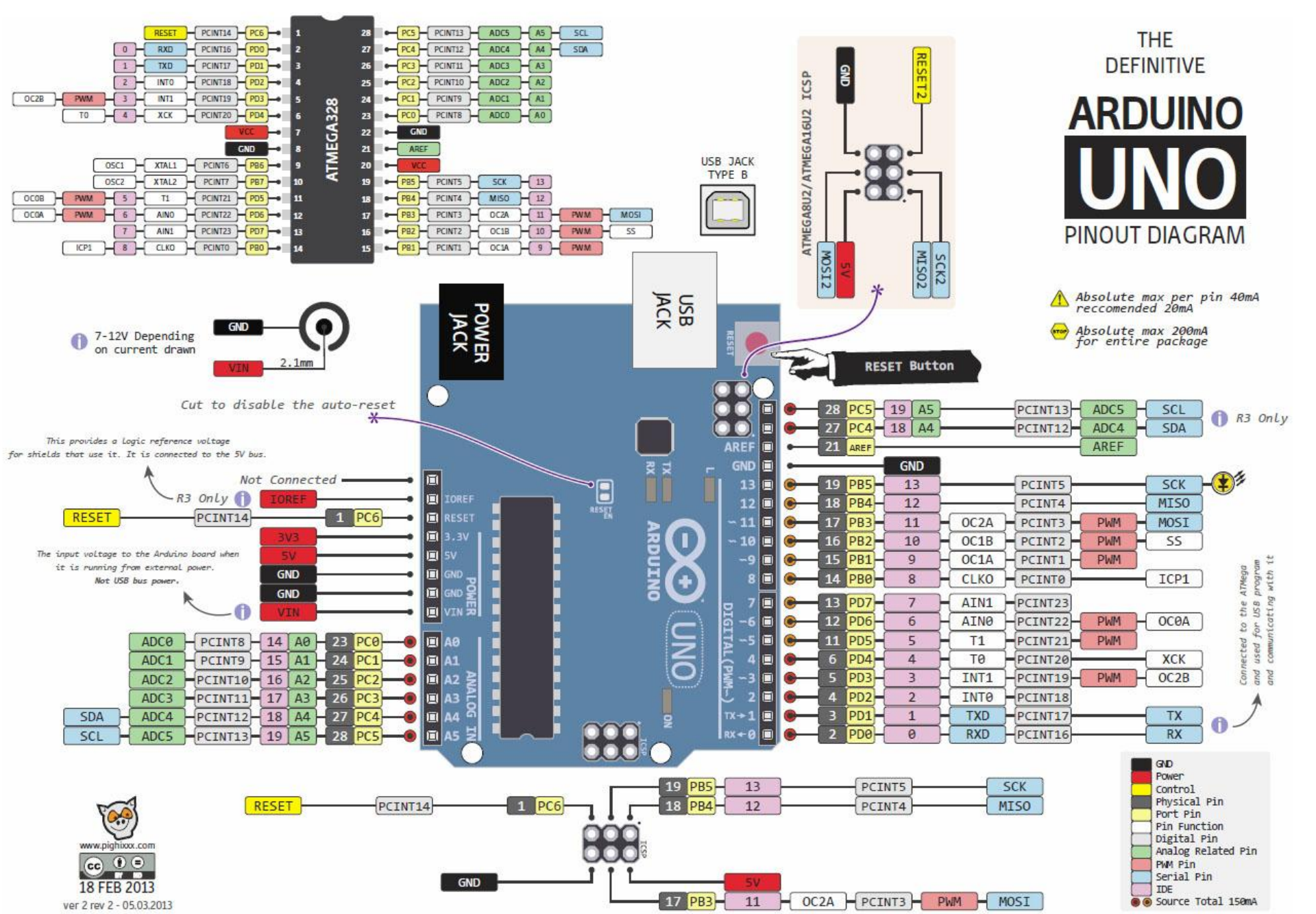
Instructables

User manual

Termék honlapja






THE DEFINITIVE
ARDUINO UNO
PINOUT DIAGRAM
ATMEGA328
ATMEGA8U2/ATMEGA16U2 ICSP
USB JACK TYPE B
POWER JACK
USB JACK
RESET Button
7-12V Depending on current drawn
2.1mm
Cut to disable the auto-reset
This provides a Logic reference voltage for shields that use it. It is connected to the 5V bus.
Not Connected
R3 Only
The input voltage to the Arduino board when it is running from external power. Not USB bus power.
Absolute max per pin 40mA reccomended 20mA
Absolute max 200mA for entire package
Connected to the ATMega and used for USB program and communicating with it
www.pighixxx.com
18 FEB 2013
ver 2 rev 2 - 05.03.2013
GND
Power
Control
Physical Pin
Port Pin
Pin Function
Digital Pin
Analog Related Pin
PWM Pin
Serial Pin
IDE
Source Total 150mA

Emlékeztető: Arduino nano v3.0