

Léptetőmotor rezgetése



```
// Include the Arduino Stepper Library
#include <Stepper.h>

// Number of steps per output rotation
const int stepsPerRevolution = 200;

// Create Instance of Stepper library
Stepper myStepper(stepsPerRevolution, 4, 3, 2, 1);

void setup()
{
    // set the speed at 60 rpm:
    myStepper.setSpeed(60);
    // initialize the serial port:
    Serial.begin(9600);
}

void loop()
{
    // step one revolution in one direction:
    Serial.println("clockwise");
    myStepper.step(stepsPerRevolution);
    delay(500);

    // step one revolution in the other direction:
    Serial.println("counterclockwise");
    myStepper.step(-stepsPerRevolution);
    delay(500);
} csak razag a léptetőmotor
```

A léptetőmotorod jelenlegi programja egy teljes fordulatot tesz előre és vissza, de ha csak „rázzuk” a léptetőmotort (azaz apró mozgásokkal, előre-hátra lépegetéssel vibrációs hatást akarunk elérni), akkor néhány kisebb lépés szükséges oda-vissza, nem pedig egy teljes kör.

Itt egy módosított példa arra, hogy a léptetőmotor „rezegjen” (pl. kis lépéseket tegyen gyors egymásutánban, irányváltásokkal):

```
cpp

#include <Stepper.h>

const int stepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 4, 3, 2, 1);

void setup() {
```

```
myStepper.setSpeed(100); // gyorsabb sebesség, hogy rezegjen
Serial.begin(9600);
}

void loop() {
  Serial.println("Shaking...");
  for (int i = 0; i < 10; i++) {
    myStepper.step(5); // kis lépés előre
    delay(50); // rövid szünet
    myStepper.step(-5); // kis lépés vissza
    delay(50); // rövid szünet
  }

  delay(1000); // 1 másodperc szünet két rázás között
}
```

Mit csinál ez a program:

- A motor 5 lépést megy előre, majd 5 lépést vissza, 10-szer egymás után.
- Így rezgő mozgás jön létre, mintha „ráznád” a motort.
- A sebesség (100 RPM) és a lépések száma (5) finomítható, hogy erősebb vagy lágyabb legyen a hatás.

Ha más típusú mozgást szeretnél (például növekvő amplitúdójú rezgést vagy véletlenszerű irányváltást), szívesen segítek abban is.



de az előző programmal rezeg az a baj

Köszí a pontosítást! Ha a **korábbi programoddal a motor csak „rezeg”** (tehát **nem forog rendesen**, pedig kellene), akkor néhány dolog lehet a probléma oka. Itt van egy lépésenkénti hibakeresési útmutató:

1. Helytelen bekötés / vezetékek sorrendje

A `Stepper` könyvtárban megadott **lábak sorrendje nagyon fontos**:

```
cpp

Stepper myStepper(stepsPerRevolution, 4, 3, 2, 1);
```

Ez azt jelenti, hogy az IN1, IN2, IN3, IN4 lábak sorrendben az Arduino **4, 3, 2, 1** lábaira vannak kötve. Ha nem ebben a sorrendben kötötted be a motor vezérlőmodulját, akkor csak „rezegni” fog, mert nem kapja a megfelelő sorrendű impulzusokat.

 **Ellenőrizd**, hogy a motorvezérlő (pl. ULN2003 vagy hasonló) bemenetei **pontosan ebben a sorrendben** csatlakoznak.

2. Túl alacsony tápfeszültség / gyenge táp

A léptetőmotor gyakran **külső tápellátást** igényel (pl. 5V vagy 12V-os adapter), mert az Arduino önmagában nem képes elég áramot adni.

Ha csak USB-ről megy a motor, **nem lesz elég erős**, csak rezegni fog vagy ki se mozdul.

🔧 **Próbáld külső tápról (pl. 12V adapter) hajtani**, és ellenőrizd, hogy a GND közös az Arduino és a motorvezérlő között.

⚡ 3. Sebesség túl alacsony vagy túl magas

A `` .setSpeed(60);`` beállítás megfelelő lehet, de:

- Ha túl alacsony (pl. 5 RPM alatt), a motor nehezen indul.
- Ha túl magas (pl. 200+ RPM), a motor nem bírja követni a lépéseket → szintén rezgés.

🔧 **Próbáld módosítani az RPM értéket**, pl.:

```
cpp
myStepper.setSpeed(100);
```

🔄 4. Lépések száma nem megfelelő

Ha a motorod nem 200 lépéses (pl. 28BYJ-48 esetén ez **4096 lépés** egy teljes körhöz), akkor ez is okozhat furcsa viselkedést.

🔧 **Próbáld megnézni a motor pontos típusát**, és ennek megfelelően állítsd be:

```
cpp
const int stepsPerRevolution = 4096; // ha pl. 28BYJ-48
```

✅ Összefoglalva:

A „csak rezeg” probléma valószínű oka(i):

1. Hibás vezeték sorrend (lábak rosszul bekötve).
2. Kevés tápellátás – külső tápegység kell.
3. Túl alacsony vagy túl magas sebesség.
4. Rossz ``stepsPerRevolution`` érték.

Ha küldesz egy fotót a bekötésről vagy megírod a motor típusát (pl. 28BYJ-48 vagy NEMA 17), még pontosabban tudok segíteni.



az értékek változtatása nem segít egy wemos d1 minin vagyok

Köszö a visszajelzést, és az, hogy **Wemos D1 Mini**-t használ, nagyon fontos információ! Ez **magában hordozza a legvalószínűbb okát** annak, hogy a léptetőmotorod csak „rezeg” és nem forog:

⚠ A probléma: A `Stepper` könyvtár nem működik jól ESP8266-tal (Wemos D1 Mini)

A `Stepper.h` könyvtár az **ATmega328-alapú** Arduino-kra lett optimalizálva (pl. UNO, Nano), és **nem kompatibilis jól** az **ESP8266** (Wemos D1 Mini) architektúrájával. Ezért jön elő a jelenség, hogy a motor csak **rezeg vagy furcsán viselkedik**.

✅ Megoldás: Használj másik könyvtárat, amely működik ESP8266-on is

A javasolt megoldás egy **kompatibilis motorvezérlő könyvtár** használata, pl.:

🔧 AccelStepper könyvtár

Az `AccelStepper` könyvtár **ESP8266-kompatibilis** és sokkal megbízhatóbb:

Példa kód Wemos D1 Mini-hez:

```
cpp

#include <AccelStepper.h>

// Define pins for motor driver IN1-IN4
#define IN1 D1
#define IN2 D2
#define IN3 D3
#define IN4 D4

// Create stepper motor object (4 wire mode)
AccelStepper stepper(AccelStepper::FULL4WIRE, IN1, IN3, IN2, IN4);

void setup() {
  stepper.setMaxSpeed(1000); // beállított maximális sebesség
  stepper.setSpeed(200);     // aktuális mozgási sebesség
}

void loop() {
  stepper.runSpeed(); // folyamatos mozgás
}
```

Ez a kód **folyamatos forgást** indít el a motoron, és **biztosan működik Wemos D1 Mini-vel**.

🔧 Fontos bekötési megjegyzés:

- D1 = GPIO5
- D2 = GPIO4
- D3 = GPIO0
- D4 = GPIO2

Győződj meg róla, hogy a vezérlőmodul (pl. ULN2003) IN1–IN4 bemenetei ehhez passzolnak.

Telepítés:

Menj az **Arduino IDE-ben a Könyvtárkezelőbe**, és:

- Keress rá: ``AccelStepper``
- Telepítsd a megfelelő könyvtárat

Ha szeretnéd, küldhetek egy komplett példát rázáshoz vagy pontos forgatáshoz. Csak írd meg, milyen típusú motorod van (pl. 28BYJ-48?), és hogyan kötötted be.



nincs típusa forog de kicsi a nyomaték

Köszí az infót! Ha **a motor forog, de gyenge a nyomaték**, az azt jelenti, hogy **működik az alapvezérlés**, de valami miatt **nem tud elég erőt kifejteni**. Ez **nem szoftveres**, hanem inkább **elektromos vagy mechanikai** probléma.

Nézzük végig a **lehetséges okokat és megoldásokat**:



1. Nem megfelelő vagy gyenge tápellátás

A **Wemos D1 Mini** önmagában **nem tud elég áramot adni** egy léptetőmotor meghajtásához. Ha a motorod csak az USB-ről kap áramot, akkor **szinte biztos**, hogy ez a fő probléma.



Megoldás:

- **Adj külső tápot** a motorvezérlőnek (pl. 5V vagy 12V adapter).
- Csatlakoztasd a **motorvezérlő tápbemenetére** (pl. ULN2003: VCC/GND).
- A **GND legyen közös** az Arduino és a motorvezérlő között!



Pl. ha ULN2003 modult használsz:

- VCC: külső 5V (nem Wemos 5V láb!)
- GND: közös a Wemos és a táp
- IN1-IN4: D1-D4, vagy más GPIO



2. Nem megfelelő stepper sebesség / gyorsítás

Ha túl gyorsan próbálsz indítani a motort, az nem tudja követni a jeleket, és „elgyengül” → nyomatékvesztés.



Megoldás:

- Lassabban indítsd el (pl. ``setSpeed(100)`` helyett ``setSpeed(50)``).

- Használd **gyorsulásvezérlést** ``AccelStepper``-rel:

```
cpp

stepper.setAcceleration(200);
stepper.setMaxSpeed(300);
stepper.setSpeed(100);
```

Ez megakadályozza, hogy a motor „megszökjön” induláskor.

3. Nem megfelelő lépésmód

Ha 28BYJ-48 vagy hasonló motorod van, akkor lehet **féllépés (half-step) módban** megy, ami **gyengébb nyomatékot** ad.

✓ Megoldás:

- Próbáld ``AccelStepper::FULL4WIRE`` vagy ``AccelStepper::HALF4WIRE`` módokkal, és nézd meg, melyik ad jobb nyomatékot.

4. Mechanikai ellenállás / túl nagy terhelés

Ha valamit mozgat a motor (pl. egy tengelyt, kart, fogaskereket), és az **túl nehéz vagy szorul**, akkor úgy tűnhet, hogy a motor gyenge.

✓ Megoldás:

- Próbáld **motor terhelés nélkül**.
- Ha így erősebb, akkor **mechanikán kell lazítani** vagy **áttételt alkalmazni**.

Összefoglalva:

Probléma	Megoldás
Motor forog, de gyenge	Külső táp 5–12V
Túl gyors indulás	Lassabb sebesség, gyorsulásvezérlés
Féllépéses mód	Próbáld FULL4WIRE móddal
Terhelés túl nagy	Teszteld üresjáratban

Ha küldesz egy fényképet a bekötésedről vagy leírod pontosan, mit vezérel a motor (fogaskerék, szíj, tengely), még konkrétabban tudok segíteni.



12v ról megy half wire 4 modeban jobban

Nagyon jó hír, hogy már **12V-ról megy**, és hogy **HALF4WIRE** módban **jobban működik** – ezek alapján a motorod valószínűleg egy **28BYJ-48**, vagy legalábbis valami **unipoláris, belső áttételes léptetőmotor**.

Amit most tudunk:

- Motor működik, forog.
- 12V táp használata segített – ez **megfelelő áramerősséget** ad.
- HALF4WIRE módban erősebb, azaz jól megy **féllépés módban**, ez **finomabb, de gyengébb** működés, viszont néha mechanikusan hatékonyabb, mert simábban indul.

Tippek a nyomaték további növeléséhez

1. Használj FULL4WIRE módot próbára

- Ez általában **nagyobb nyomatékot** ad, cserébe kevésbé sima a mozgás.

```
cpp
```

```
AccelStepper stepper(AccelStepper::FULL4WIRE, IN1, IN3, IN2, IN4);
```

- Nézd meg, így is elég simán indul-e. Ha igen, **maradj FULL4WIRE módban** a nyomaték miatt.

2. Gyorsulásvezérlés beállítása

- Ez megakadályozza, hogy a motor lemaradjon induláskor, és így **több erőt tud kifejteni**:

```
cpp
```

```
stepper.setMaxSpeed(400);  
stepper.setAcceleration(200);
```

3. Áttétel vagy mechanikai segítség

- Ha nagy terhet kell mozgatnia, akkor gondolkozz egy **fogaskerekes áttételben** vagy **csigahajtásban**.
- A 28BYJ-48 már **belső áttételes**, de ha más motorod van, ezt lehet külsőleg pótolni.

4. Áramkorlát ellenőrzése (ha van driver pl. A4988)

- Ha nem ULN2003-at használasz, hanem pl. A4988, DRV8825 stb., akkor a **potméterrel állítható az áramerősség**.
- Több áram = több nyomaték, **de vigyázat, ne melegedjen túl**.

Példa: erős és biztonságos forgatás

```

cpp

#include <AccelStepper.h>

#define IN1 D1
#define IN2 D2
#define IN3 D3
#define IN4 D4

AccelStepper stepper(AccelStepper::FULL4WIRE, IN1, IN3, IN2, IN4);

void setup() {
  stepper.setMaxSpeed(400); // max sebesség
  stepper.setAcceleration(200); // gyorsulás
  stepper.setSpeed(200); // aktuális sebesség
}

void loop() {
  stepper.runSpeed(); // folyamatos forgás
}

```

Ha szeretnél **vezérelhető mozgást** (pl. menjen 180 fokot és álljon meg), abban is tudok segíteni – csak szólj.

Illetve ha a motor típusa mégis más, mint a 28BYJ-48, vagy gyanús valami, küldj képet, és megnézzük pontosabban.



egy nema17 nek mennyi a nyomatéka?

A **NEMA 17** nem egy konkrét motor, hanem egy **méretosztály** (43.2 × 43.2 mm-es négyzetes homlokfelület). Ezért a **nyomaték** típustól és gyártótól függően nagyon eltérő lehet.



Általános NEMA 17 nyomatékértékek:

Típus	Tápfeszültség	Áram	Tartónyomaték (Holding torque)
Gyenge típus	12V, 1A	~0.3 Nm (30 N·cm)	
Átlagos	12–24V, 1.5–2A	~0.4–0.5 Nm (40–50 N·cm)	
Erősebb NEMA 17	24V, 2A+	akár 0.6–0.8 Nm (60–80 N·cm)	



Fontos tudni:

- A **tartónyomaték (holding torque)** azt jelzi, mekkora erővel tud a motor ellenállni a mozgásnak áram alatt, de álló helyzetben.
- Forgó nyomaték** (running torque) kisebb lehet, főleg magas fordulaton.
- Az **áram és a feszültség növelésével** nő a nyomaték – ehhez **stepper driver** kell (pl. A4988, TMC2208, DRV8825).
- Általában a **24V tápellátás** hatékonyabb, mint a 12V – főleg magasabb fordulatszám mellett.

Példa konkrét típusra:

Egy gyakori típus:

- **Model:** 17HS4401
- **Tartónyomaték:** 0.40 Nm (40 N·cm)
- **Áram:** 1.7 A
- **Lépésszög:** 1.8°
- **Táp:** 12–24V

Vezérléshez ajánlott:

- **Driver:** A4988, DRV8825, TMC2209
- **Táp:** 24V/2A (optimális)
- **Hűtés:** szükséges, ha nagy áramon használsz

Ha megírod, hogy milyen típusú NEMA 17-ed van (felirat a motoron), akkor pontosabb adatot tudok adni. Vagy küldhetsz képet a címkéről.



```
Linking .pio\build\d1_mini\firmware.elf
c:/users/bence/.platformio/packages/toolchain-xtensa/bin/../lib/gcc/xtensa-lx106-elf/10.3.0/../../../../xtensa-lx106-elf/bin/ld.exe: .pio\build\d1_mini\src\main.cpp.o:(.text.setup+0xc):
undefined reference to `__ZN12AccelStepper11setMaxSpeedEf'
c:/users/bence/.platformio/packages/toolchain-xtensa/bin/../lib/gcc/xtensa-lx106-elf/10.3.0/../../../../xtensa-lx106-elf/bin/ld.exe: .pio\build\d1_mini\src\main.cpp.o:(.text.setup+0x10):
undefined reference to `__ZN12AccelStepper8setSpeedEf'
c:/users/bence/.platformio/packages/toolchain-xtensa/bin/../lib/gcc/xtensa-lx106-elf/10.3.0/../../../../xtensa-lx106-elf/bin/ld.exe: .pio\build\d1_mini\src\main.cpp.o: in
function `setup':
main.cpp:(.text.setup+0x23): undefined reference to `__ZN12AccelStepper11setMaxSpeedEf'
c:/users/bence/.platformio/packages/toolchain-xtensa/bin/../lib/gcc/xtensa-lx106-elf/10.3.0/../../../../xtensa-lx106-elf/bin/ld.exe: .pio\build\d1_mini\src\main.cpp.o:(.text.loop+0x0):
undefined reference to `__ZN12AccelStepper8runSpeedEv'
c:/users/bence/.platformio/packages/toolchain-xtensa/bin/../lib/gcc/xtensa-lx106-elf/10.3.0/../../../../xtensa-lx106-elf/bin/ld.exe: .pio\build\d1_mini\src\main.cpp.o:
(.text.startup._GLOBAL__sub_I_stepper+0x0): undefined reference to `__ZN12AccelStepperC1Ehhhhhb'
c:/users/bence/.platformio/packages/toolchain-xtensa/bin/../lib/gcc/xtensa-lx106-elf/10.3.0/../../../../xtensa-lx106-elf/bin/ld.exe: .pio\build\d1_mini\src\main.cpp.o: in
function `__GLOBAL__sub_I_stepper':
main.cpp:(.text.startup._GLOBAL__sub_I_stepper+0x1a): undefined reference to
`__ZN12AccelStepperC1Ehhhhhb'
collect2.exe: error: ld returned 1 exit status
*** [.pio\build\d1_mini\firmware.elf] Error 1
===== [FAILED]
```

Took 5.38 seconds

```
=====

* The terminal process "C:\Users\Bence\.platformio\penv\Scripts\platformio.exe 'run'" terminated
with exit code: 1.
* Terminal will be reused by tasks, press any key to close it.
```

Ez a hibaüzenet azt jelenti, hogy a ``AccelStepper`` **könyvtár nincs megfelelően beépítve** a projektedbe, így a linker nem találja a könyvtár függvényeit.

🔥 Hibák magyarázata:

- ``undefined reference to 'AccelStepper::setMaxSpeed'`` stb. → ezek mind azt jelzik, hogy a függvénydefiníciók hiányoznak → azaz **nincs belinkelve a könyvtár forrása**.

✅ Megoldás lépésről lépésre PlatformIO-ban

1. 📦 Győződj meg arról, hogy a ``AccelStepper`` könyvtár telepítve van

Menj a ``platformio.ini`` fájlodhoz, és adj hozzá egy sort a ``lib_deps`` alá:

```
ini

[env:dl_mini]
platform = espressif8266
board = dl_mini
framework = arduino
lib_deps =
  mike-matera/AccelStepper@^1.61
```

Vagy ha a ``lib_deps`` már létezik:

```
ini

lib_deps =
  ...
  mike-matera/AccelStepper
```

Alternatív megoldásként a könyvtár telepítése terminálból:

```
bash

pio lib install "mike-matera/AccelStepper"
```

2. 🧹 Tisztítsd le a buildet és építsd újra

A PlatformIO-ban futtasd:

```
bash

pio run --target clean
pio run
```

3. 📁 Győződj meg arról, hogy a forráskódodban helyesen van include-olva:

```
cpp

#include <AccelStepper.h>
```

🔄 Ellenőrzés után próbáld újrafordítani

Ha minden rendben van, a `undefined reference` hibák megszűnnek, és a firmware szépen lefordul.

Ha még mindig hibát kapsz, írd be ide a `platformio.ini` fájlod teljes tartalmát és a `main.cpp` elejét, hogy ellenőrizni tudjam pontosan.



a ./include ben van

Köszí, hogy írtad: ha a `AccelStepper.h` a **saját** `./include` mappában van, akkor PlatformIO **nem tudja automatikusan belinkelni** a hozzá tartozó `.cpp` fájlt, **kivéve ha azt is odatetted és helyesen konfiguráltad.**

A `header` (`.h`) fájl **önmagában nem elég** – a linkerhiba azért van, mert a hozzá tartozó forrás (`.cpp`) nincs beépítve.

✅ Megoldási lehetőségek

🔧 1. Helyes módszer: Használd a PlatformIO könyvtárkezelőt (`lib_deps`)

Ez a legmegbízhatóbb módszer:

`platformio.ini`:

```
ini

[env:d1_mini]
platform = espressif8266
board = d1_mini
framework = arduino
lib_deps =
    mike-matera/AccelStepper
```

Ez letölti a teljes könyvtárat, `.cpp`-estül, és automatikusan beépíti. Ez a **javasolt módszer**.

🔧 2. Ha ragaszkodsz a saját `./include` könyvtárhoz:

- A `AccelStepper.cpp` fájlt is **tedd be a `src/` könyvtárba**, vagy
- Külön hozz létre egy saját könyvtárat a `lib/` mappában:

Fájlstruktúra például:

markdown

```
lib/
└─ AccelStepper/
    └─ AccelStepper.h
    └─ AccelStepper.cpp
```

PlatformIO automatikusan felismeri a `lib/AccelStepper/`-t mint könyvtárat.

A `include/` mappa PlatformIO-ban **csak fejlécekre (header files)** való, nem fordul le benne `.cpp`.

📌 Összefoglalás:

Módszer	Előny	Hátrány
`lib_deps` a `platformio.ini`-ben	Automatikus, működik	Internet kell
`lib/AccelStepper/` saját könyvtárként	Offline működik, testreszabható	Neked kell karbantartani
`include/` használata	Nem működik <code>.cpp</code> nélkül	Nem javasolt forrásokhoz

Írd meg, melyik megoldás szimpatikus, és segítsek beállítani. Ha szeretnéd, küldheted a mappa/fájlszerkezetet, és ránézek!