

Arduino Tiny ATtiny 85

Chipje programmeren met behulp van Arduino:

Youtube: <https://www.youtube.com/watch?v=sycSdI49h1Y>



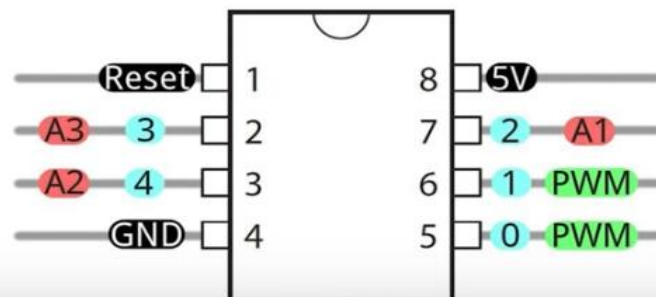
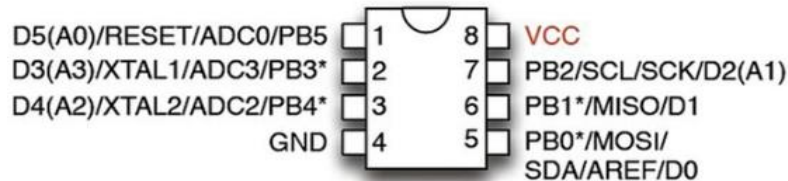
Dit is de Tiny die ik (Dré) heb gekocht: € 2,83

<https://nl.aliexpress.com/item/1005007456620827.html?gatewayAdapt=glo2nld>

De ATtiny werkt zonder externe componenten zoals oscillator etc.

ATtiny 85 8KByte program flash 500 Byte EEprom en 500 byte RAM.

ATtiny85 Pinout



5 digitale I/O pootjes
3 analoge ingangen pootjes
2 PWM pootjes

Naderhand gelezen, niet getest:
I²C: 5 = SDA ; 7 = SCL

Mogelijk verwarring: welke pootjes moet ik gebruiken?

Bij het AANSLUITEN voor programmeren van bootloader en sketch, kies je de **fysieke** pootnummers, de zwarte cijfers, die IN de rechthoek staan geschreven!

Voor de sketchen (blink etc.), kies je de **blauwe**, **rode** of **groene** pootjes. (= interne poort B.)

Kies de Uno: Hulpmiddelen → bord → Arduino AVR boards → Arduino UNO

Kies de Nano: Hulpmiddelen → bord → Arduino AVR boards → Arduino Nano

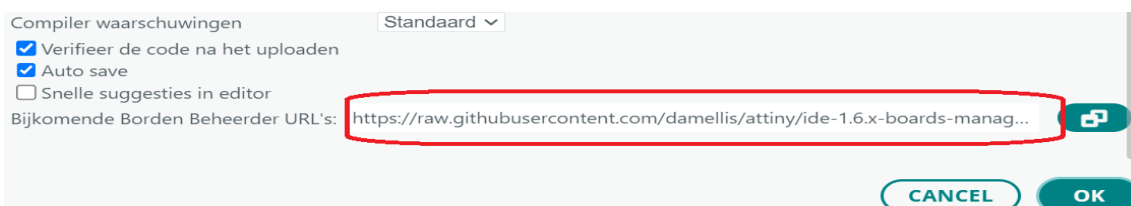
Hulpmiddelen → processor → ATmega328 p (Old Bootloader)

Voor het programmeren is ook een 10 µF condensator (elco) nodig.

-1- Arduino aanpassen:

Files → voorkeuren → voeg onderaan deze link toe:

https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json



Druk vervolgens op OK en start de Arduino IDE opnieuw.

Nu worden de ATtiny borden geladen.

Hulpmiddelen → bord → bordbeheer → en typ ATtiny
Selecteer: **attiny by David A Mellis** en klik op install.
Nu de Arduino IDE opnieuw starten.

-2- Arduino UNO of Nano setup:

Hulpmiddelen → selecteer de poort

File → voorbeelden → 11 Arduino ISP → Arduino ISP

Dit opent een nieuwe (zeer lange) sketch.

Hulpmiddelen → selecteer bord (UNO of Nano) en → poort → upload de ISP sketch.

→ Hier laad je dus de Arduino Uno of Nano op, kies dus een van deze twee bordjes. ←

Je kan de Arduino nu loskoppelen en naar de volgende stap gaan. **elco nog NIET aansluiten**

-3- ATtiny aansluiten:

Er wordt gesproken over 'pootje 1' maar is dat digitaal pootje 1 of juist fysiek pootje 1?

De pootjes waarover hier wordt gesproken zijn de **FYSIEKE POOTJES** (1=reset gemerkt).

Tiny	Arduino	
+5V = -8- ==	5V	Controleer of de bedrading goed zit, stevig aandrukkenetc.
GND -4- ==	GND	Gebruik bij voorkeur nieuwe draden, eea is erg gevoelig.
-7- ==	-13-	
-6- ==	-12-	
-5- ==	-11-	
-1- ==	-10-	

-4- Bootloader programmeren

Hulpmiddelen → bord → bord → ATtiny. → **attiny 25/45/85**.

Hulpmiddelen → programmeerapparaat → **Arduino as ISP**.

Hulpmiddelen → clock internal **1 MHz** (mag ook 8 MHz zijn maar begin met 1 MHz)

Hulpmiddelen → processor → selecteer **ATtiny 85**

Hulpmiddelen → selecteer de poort (dezelfde als die voor de Arduino)

Sluit NU pas de 10 µF elco aan op de reset en GND van de Arduino UNO of Nano, dus niet op de Tiny

-5- bootloader branden

Heb je een spiksplinternieuwe Tiny, dan is er nog geen bootloader aanwezig.

Ga wéér naar hulpmiddelen: → **bootloader branden** de bootloader wordt nu meteen geladen.

Krijg je een foutmelding, dan ligt dat meestal aan de bedrading.

Het branden van de bootloader is een éénmalig gebeuren voor elke ATtiny afzonderlijk.

-6- tiny programmeren

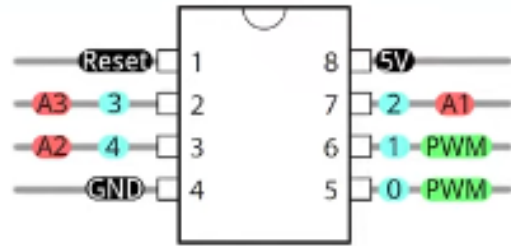
Dat gaat **NIET** met de standaard 'upload' knop.

Ga naar: sketch → **uploaden met behulp van een programmeerapparaat**.

Dan wordt het programma in de ATtiny 85 geladen.

Voorbeeld programma **Blink.ino**

```
void setup() {  
  pinMode(4, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(4, HIGH);  
  delay(1000);  
  digitalWrite(4, LOW);  
  delay(1000);  
}
```



Nu gebruik je de **digitale (blauwe) nummers**, dus digitaal pootje -4- zit op fysiek adres -3-.

Er is géén led buildin, dat veranderen we dus in pootje -4-

Dus nu naar sketch → **uploaden met behulp van een programmeerapparaat.**

Nu de blink is geladen, kan je alle bedrading verwijderen, behalve de voeding.

Een LED via een weerstand aan digitaal pootje -4- (= fysiek pootje -3-) naar GND.

De LED knippert zoals in blink is geprogrammeerd.

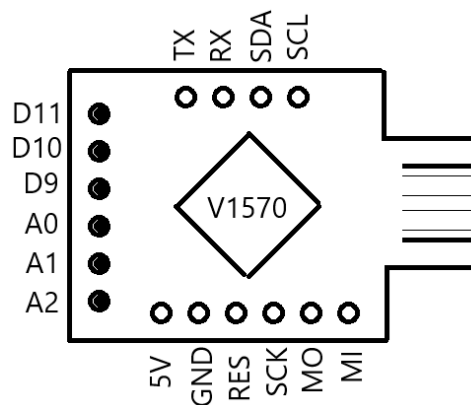
Digistump

Dit is ook een ATtiny 85, maar dan in een schakeling, kost € 5,89



https://nl.aliexpress.com/item/1005006048714346.html?search_direct=true&spm=a2g0o.productlist.0.0&gatewayAdapt=glo2nld

De aansluitingen:



Er is een heel lang verhaal op internet te vinden, maar uiteindelijk bleek dat je deze digistump als **Arduino Leonardo** kan programmeren. Houdt alleen rekening met de afwijkende pootjes.

Nu een programma laden, dat gaat gewoon zoals gebruikelijk.

```
void setup() {  
  pinMode(11, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(11, HIGH);  
  delay(100);  
  digitalWrite(11, LOW);  
  delay(100);  
}
```

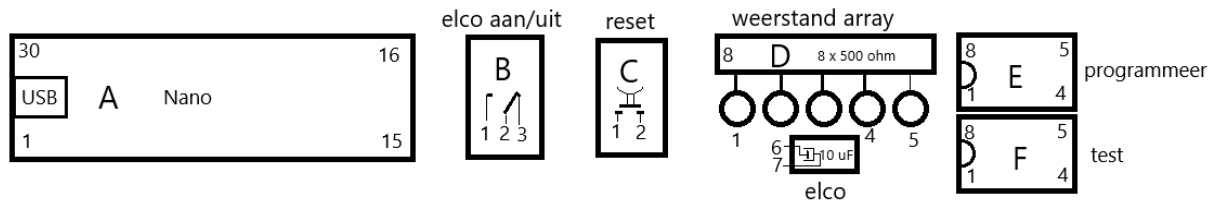
Tiny netlist

Hieronder **mijn** opstelling, de aansluitingen staan hierboven. In de netlist staat hoe ik het uiteindelijk heb uitgevoerd. Positie **E** is om de chip te programmeren.

Positie **F** is aangesloten op het resetknopje en de LEDjes via de weerstandarray

Bij mij moet voor het programmeren van de chip de ELCO persé zijn ingeschakeld.

Om het programmeer programma in de Nano te zetten, moet de ELCO uit!



De nummers van de aansluitpunten komen niet overeen met de poortnummers.

Het zijn de cijfers zoals je normaal telt bij chips, bovenzijde linksom, onderzijde rechtsom.

Dit doe ik altijd zo om vergissingen te voorkomen. Deze methode is voor mij al jaren failsafe!

A12-D8-E8-F8	Voeding + 5V
A14-C1-D7-E4-F4	Ground -
A1-E7	D 13 SCK Serial Clock
A13-B1	Nano reset naar schakelaar voor ELCO
A28-E1	D10 Slave Select
A29-E5	D11 Master Out Slave In
A30-E6	D12 Master In Slave Out
B2-D6	Schakelaar naar ELCO +
C2-F1	Resetknopje naar Tiny testpositie
D1-F5	LED 1
D2-F6	LED 2
D3-F7	LED 3
D4-F2	LED 4
D5-F3	LED 5

Test programma voor de ATtiny

De tiny heeft 5 uitgangen, waarvan twee PWM pootjes (0 en 1)

Allen op poort B. B6 is de reset, maar daar kan ik niet zo veel mee, alleen resetten.

Tiny 85 kan op 1 of op 8 MHz draaien. In de bootloader te kiezen.

Aansluiting LED's Anode via weerstandarray op + lampje aan door -0- uit te sturen.

// declaratie

```
int nightrider[] = { 30, 29, 27, 23, 15 }; // heen en weer gaande lamp (-0- = lamp aan)
```

```
int kamers[] = { 3, 5, 25, 15, 30, 31 }; // diverse lamp combinaties (-0- = lamp aan)
```

```
// Zo heeft elk huisje zijn eigen ritme. Het hele dorp gaat immers niet gelijktijdig naar bed.
```

```
void setup() {
```

```
  DDRB = 31; // alle 5 pootjes uitgang
```

```
  // DDR is Data Direction Register van poort B -1- = uitgang.
```

```
  PORTB = 31; // alle 5 lampjes uit (hoog = uit)
```

```
}
```

```
void loop() {
```

```
  for (int n = 0; n < 31; n++) { // binair tellen 0.. 31
```

```
    PORTB = (31 - n); // telkens -1- er bij hier tellen de nullen!
```

```
    delay(300); // korte pauze tussen twee getallen
```

```
  }
```

```

PORTB = 31; // lampen uit, alle pootjes hoog
delay(1000); // seconde wachten

for (int n = 0; n < 4; n++) { // looplamp wordt viermaal uitgevoerd
  for (int n = 0; n < 5; n++) {
    digitalWrite(n, !(digitalRead(n))); // zet alle lampjes aan en daarna weer uit
    delay(500); // halve seconde wachten
  }
  delay(1000); // seconde wachten voor volgende opdracht
}

for (int n = 0; n < 5; n++) { // opdracht 5 maal uitvoeren
  for (int n = 0; n < 5; n++) {
    PORTB = nightrider[n]; // nightrider heen en weer gaand lampje
    delay(100); // lees de array uit en schakel de lampjes
  }
  for (int n = 4; n > 0; n--) { // de andere richting op zonder 1e lampje
    PORTB = nightrider[n];
    delay(100);
  }
  PORTB = 30; // 1e lampje apart, voorkom dubbele tijd
  delay(100);
  PORTB = 31; // alle lampjes uit voor volgende opdracht
}
delay(1000); // seconde wachten

for (int n = 255; n >= 0; n--) { // dimmer
  analogWrite(0, n); // eerste lampje langzaam feller
  delay(10);
}
delay(1000);
for (int n = 0; n <= 255; n++) { // eerste lampje langzaam zachter
  analogWrite(0, n);
  delay(10);
}
delay(1000);

for (int n = 255; n >= 0; n--) { // zelfde verhaal voor tweede lampje
  analogWrite(1, n);
  delay(10);
}
delay(1000);
for (int n = 0; n <= 255; n++) {
  analogWrite(1, n);
  delay(10);
}
delay(1000);

for (int n = 0; n < 6; n++) { // kamers
  PORTB = kamers[n]; // ook hier een array voor het simuleren van lopen naar toilet,
  delay(2000); // woonkamer, slaapkamer, etc.
} // Uiteraard moet je de tussenliggende tijden langer maken.
} // array naar eigen inzicht invullen -0- = lampje aan.
// De dimmers werken onafhankelijk hiervan.

```

Dré Jansen

Met de pijl < keer je weer terug naar de Nieuwsbrief.