

LedUpKidz via Arduino 1.8.9 - 1.8.10

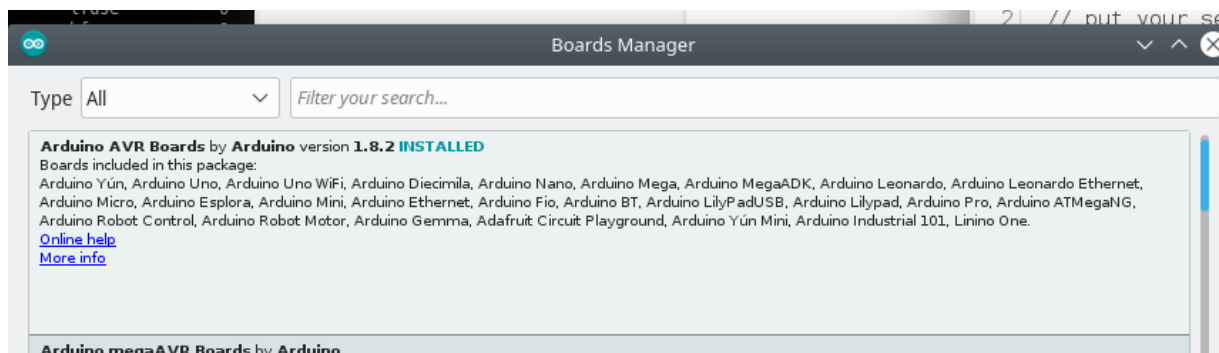
We geven hier de stappen om de LedUpKidz te programmeren via Arduino IDE 1.8.x

(opgelet! na stap 4 kan je de code niet meer veranderen. Dus led u p kidz niet herprogrammeerbaar. Dit is momenteel nog enkel mogelijk met de high voltage serial programming en oudere versie van ARDUINO, zie na deze sectie)

1. Installeer Arduino 1.8.x

Installeer de Arduino IDE van arduino.cc. Volgende is getest met Arduino IDE 1.8.9 - 10 - 12

In boards manager, getest met

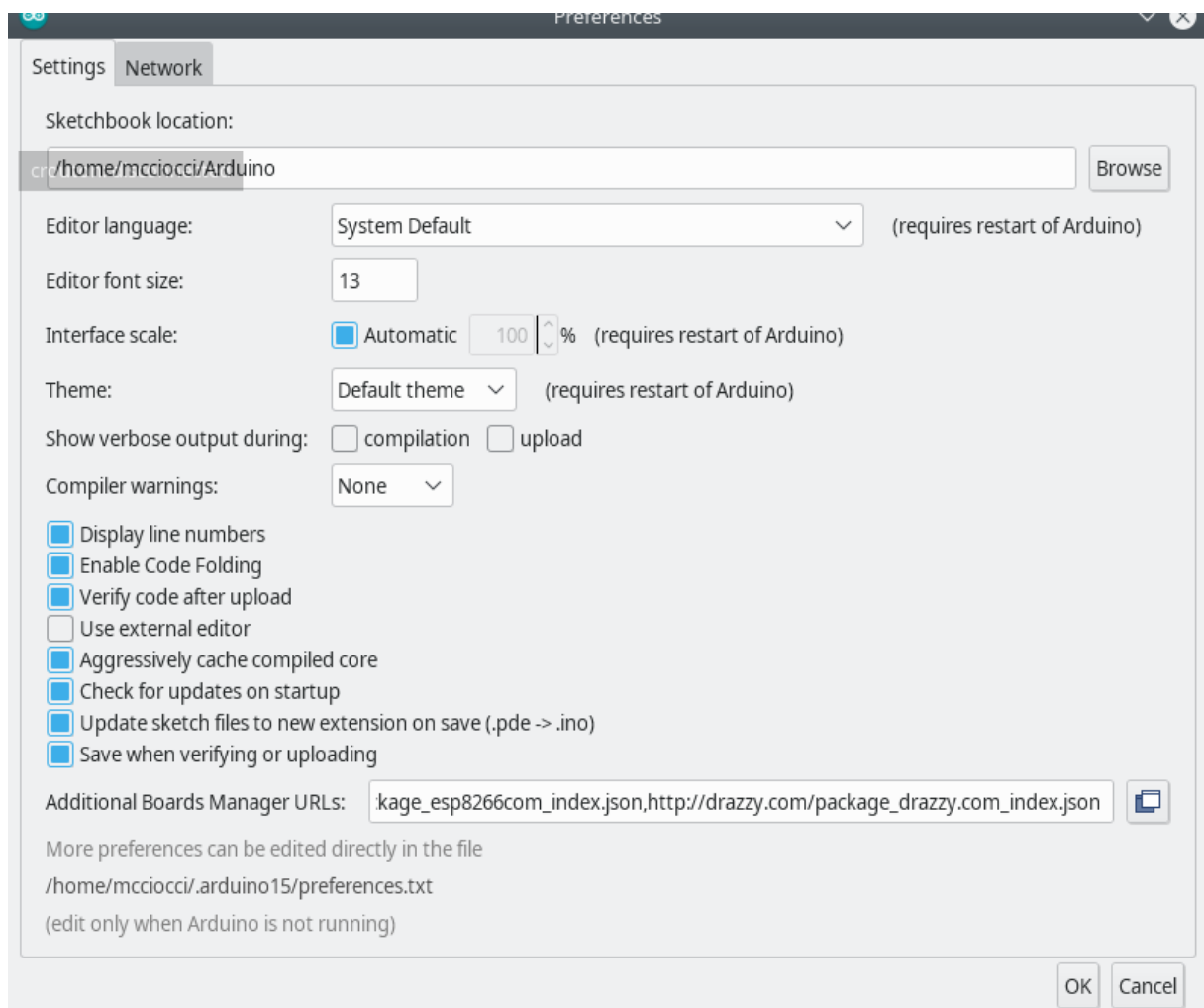
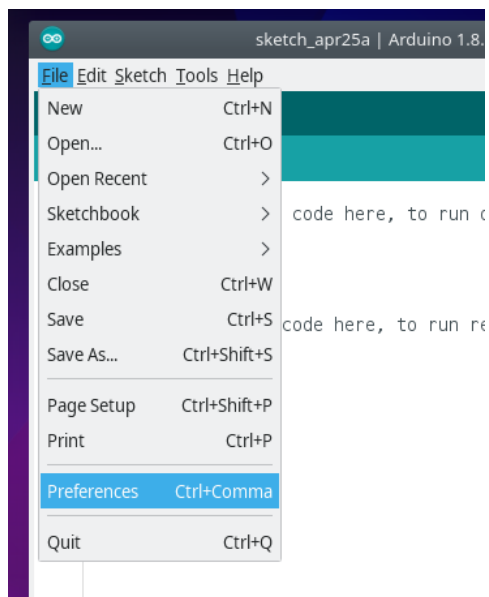


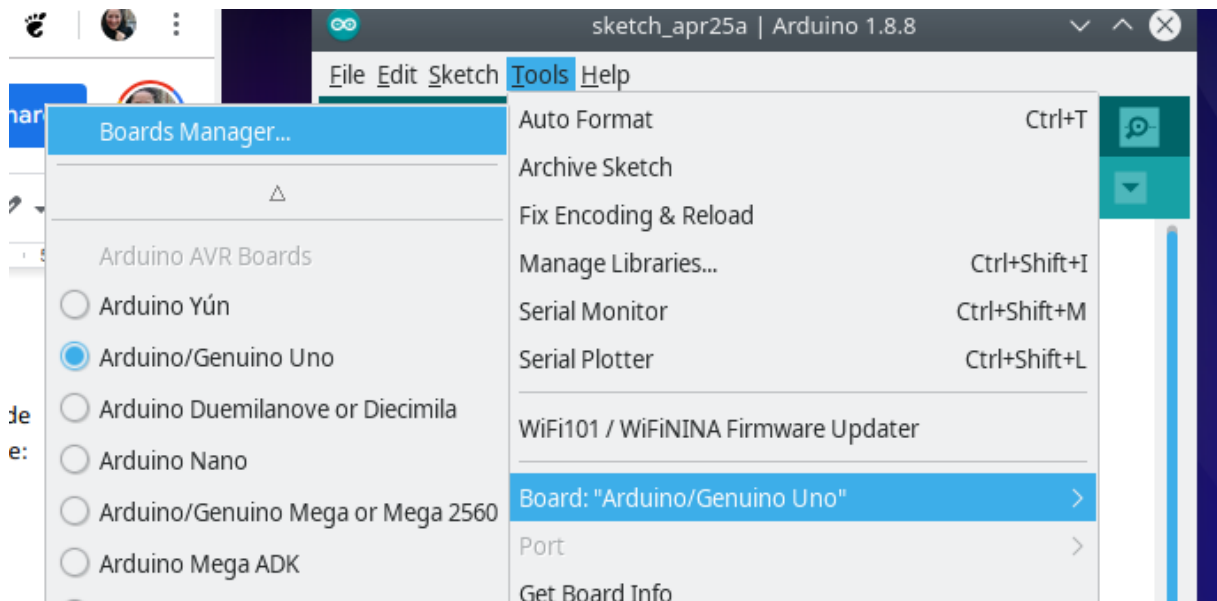
2. Installeer ATTinyCore

Installeer [ATTinyCore](https://github.com/SpenceKonde/ATTinyCore). Je doet dat via de bord manager in de IDE. Volg de volgende stappen:

1. Open eerst de Instellingen/Preferences. Voeg bij extra borden toe de volgende bord defenitie: http://drazzy.com/package_drazzy.com_index.json

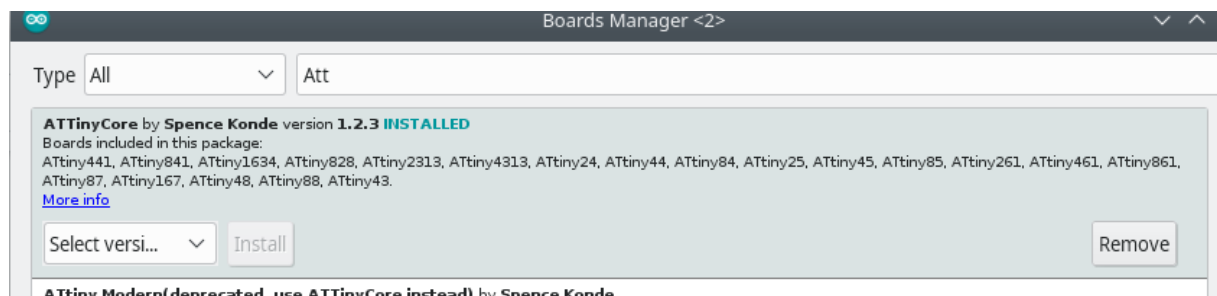
Let op, Heb je al een bord eerder toegevoegd, meerdere bord definities dienen met een komma gescheiden te zijn.





<screenshot>

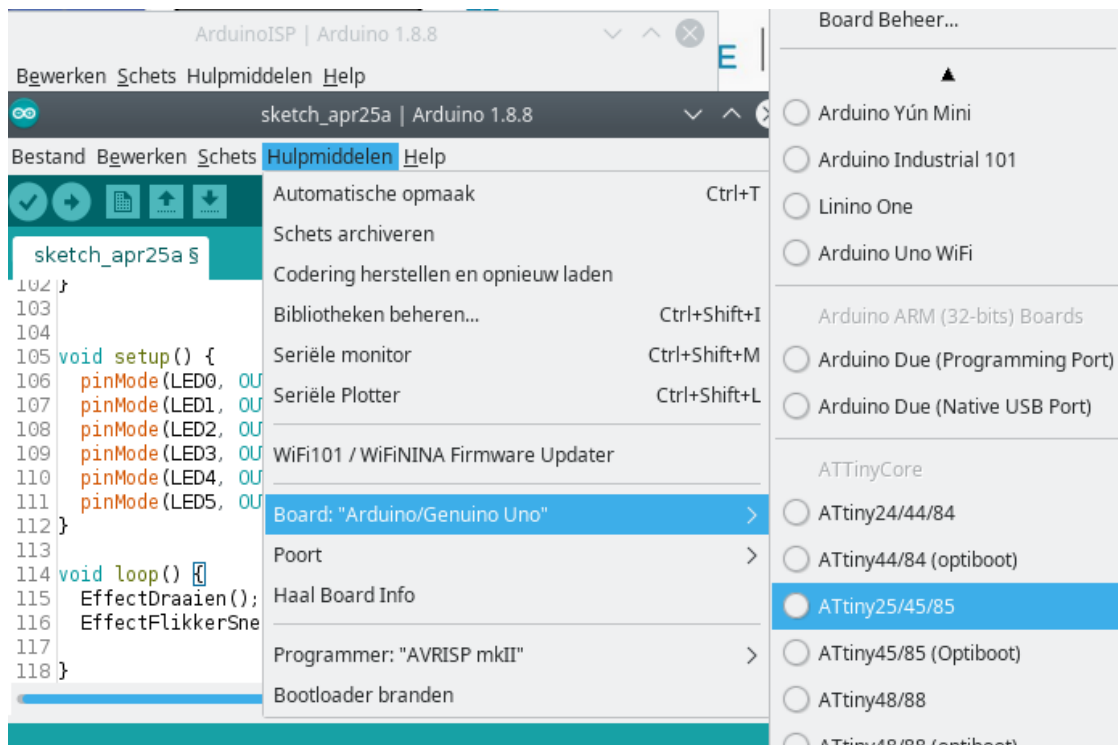
2. Ga nu naar de bordenmanager, zoek ATTinyCore, en druk op installeer om het toe te voegen. We hebben getest met versie 1.2.3 en 1.3.2 en 1.3.3



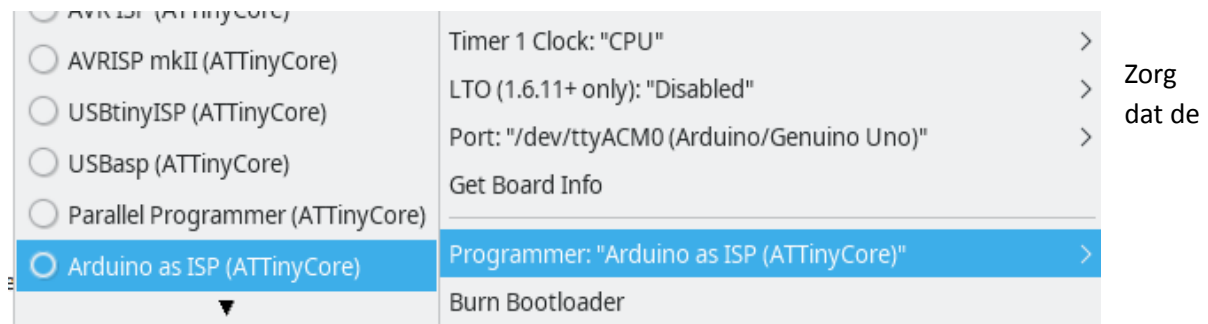
3. Code opladen

Connecteer onze Arduino as ISP Programmer (Om een extra Arduino as ISP te maken volg stappen op <https://www.arduino.cc/en/tutorial/arduinoISP#toc5>, dus met de default programmer AVRISP MKII sketch ArduinoISP opladen) met je PC via de USB kabel, en plaats een chip op de chiphouder. Het stipje op de chip moet linksboven, waar op de houder ook een stip staat.

In de Arduino IDE, selecteer als bord ATtiny25/45/85. De LedUpKidz gebruikt de ATtiny85.

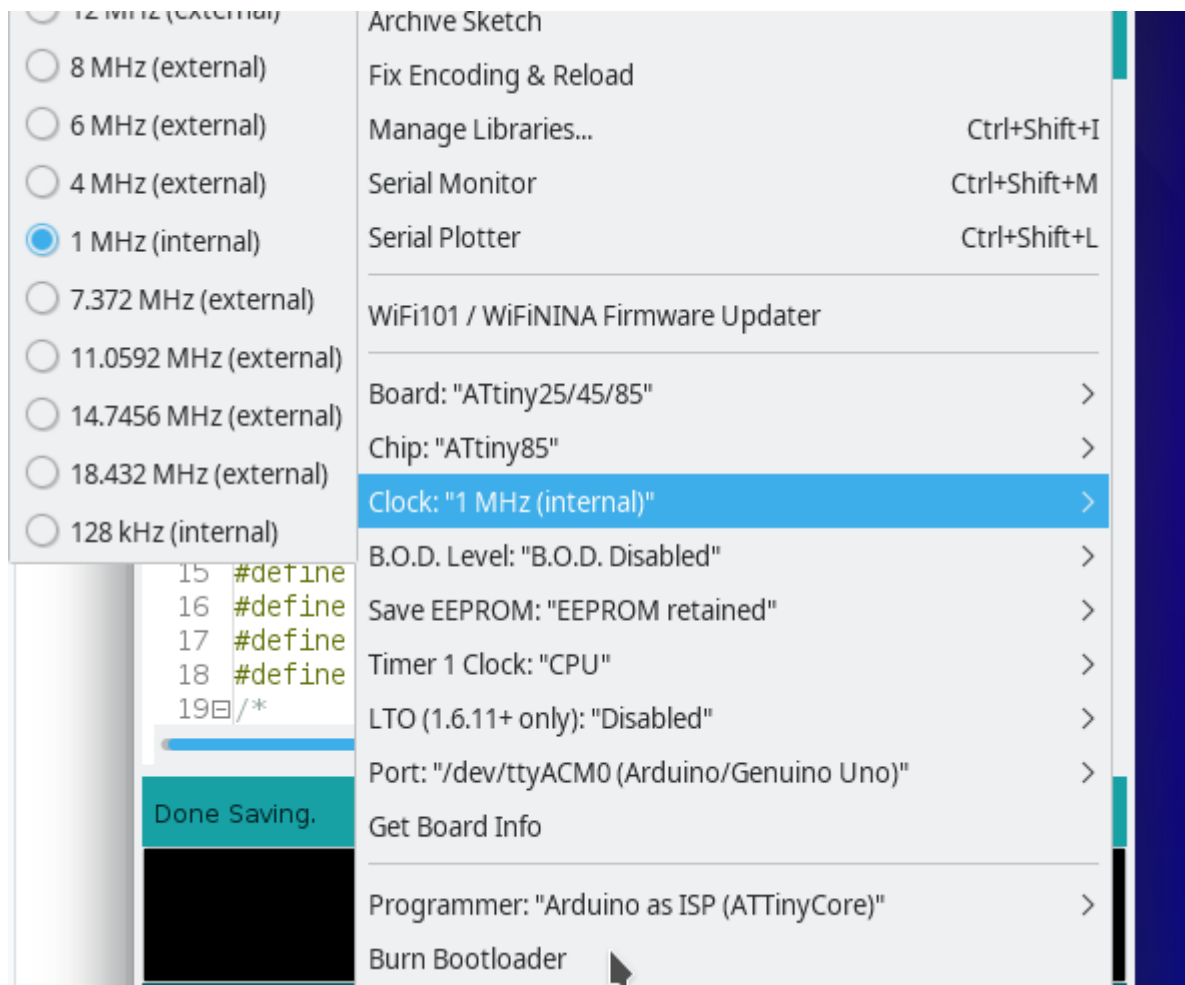


Wijzig de programmer van AVRISP mkII naar Arduino as ISP(ATTiny core),



waarden zijn zoals hieronder, dus:

2. board: ATtiny25/45/85, chip ATtiny85
3. Clock: 1 MHz (internal)
4. Andere waarden op hun default laten



Druk nu op de knop: **burn bootloader** (**DIT HOEFT NIET !!! MAG JE OVERSLAAN**).

Je kan nu de code opladen naar de chip zoals je doet voor een gewone Arduino

OPGELET: Gebruik je later de IDE voor een gewone Arduino UNO of Nano, vergeet niet alles terug te wijzigen naar de waarden voor de Uno of Nano.

4. RESET disable via COMMAND LINE

doe in konsole (command venster)

Bovenstaande is genoeg voor gewoon gebruik van de ATtiny85, maar niet voor de LedUpKidz, gezien deze laatste ook de RESET als een gewone pin gebruikt.

1e bit van de hfuse parameter op 0 zetten doet een RESET Disable. Je kan enkel daarna programma opnieuw flashen door een reset te doen via 12V!

Kort:

1. Ga naar juiste directory

```
cd ~/.arduino15/packages/arduino/tools/avrdude/6.3.0-arduino17/bin
```

- 2.

3. Dan per chip na burn bootloader en program uploaden via Arduino as ISP aan 1 MHz (vervang /dev/ttyACM0 door de juiste poort, versie (IDE 1.8.10 is 1.3.2) en mcciocci door juiste user!):

Zilver Dell laptop Cristina:

```
./avrdude  
-C/home/cristina/.arduino15/packages/ATTinyCore/hardware/avr/1.3.2/avrdude.conf -v -pattiny85 -cstk500v1 -P/dev/ttyACM0 -b19200 -Ufuse:w:0b01010111:m
```

gilian / lennard dell labo3

```
./avrdude  
-C/home/ingegnadmin/.arduino15/packages/ATTinyCore/hardware/avr/1.3.3/avrdude.conf -v -pattiny85 -cstk500v1 -P/dev/ttyACM0 -b19200 -Ufuse:w:0b01010111:m
```

Oude dell laptop

```
./avrdude  
-C/home/mcciocci/.arduino15/packages/ATTinyCore/hardware/avr/1.3.2/avrdude.conf -v -pattiny85 -carduino -P/dev/ttyACM0 -b19200 -Ufuse:w:0b01010111:m
```

4. ./avrdude

```
-C/home/ingegno/.arduino15/packages/ATTinyCore/hardware/avr/1.3.3/avrdude.conf -v -pattiny85 -cstk500v1 -P/dev/ttyACM0 -b19200 -Ufuse:w:0b01010111:m
```

Na de code op je bord gezet te hebben, kun je de reset disabelen met avrdude via de terminal. Open sketch_feb25a een terminal (Konsole op KDE):

OPGELET: Nieuwere versie, niet -carduino optie maar -cstk500v1, laatste is programmer Attinycore Arduino as ISP via arduino; -carduino is voor Mega

1. Ga naar de directory met avrdude van je Arduino install en vervang XX door de versie op je PC:

```
cd ~/.arduino15/packages/arduino/tools/avrdude/6.3.0-arduinoXX/bin
```

eg onze black dell

```
cd ~/.arduino15/packages/arduino/tools/avrdude/6.3.0-arduino17/bin
```

2. Doe dan volgend commando terwijl de Arduino as ISP nog steeds aan je PC hangt (vervang *benny* with jouw username hier):

```
./avrdude  
-C/home/benny/.arduino15/packages/ATTinyCore/hardware/avr/1.2.3/avrdude.conf  
-v -pattiny85 -carduino -P/dev/ttyACM0 -b19200 -Uhfuse:w:0b01010111:m
```

3.

```
./avrdude  
-C/home/ingegno@ingegnoadmin-HP-ProBook-650-G1/.arduino15/packages/ATTinyCore/hardware/avr/1.2.3/avrdude.conf -v -pattiny85 -carduino -P/dev/ttyACM0  
-b19200 -Uhfuse:w:0b01010111:m
```

eg onze black dell om **reset disable** fuses te branden (eerste bit 0) zonder programma te verwijderen (5e bit 0) en SPI enabled (derde bit 0):

```
./avrdude  
-C/home/mcciocci/.arduino15/packages/ATTinyCore/hardware/avr/1.2.3/avrdude.conf -v -pattiny85 -carduino -P/dev/ttyACM0 -b19200 -Uhfuse:w:0b01010111:m
```

low fuses for 1 MHz:

```
./avrdude  
-C/home/mcciocci/.arduino15/packages/ATTinyCore/hardware/avr/1.2.3/avrdude.conf -v -pattiny85 -carduino -P/dev/ttyACM0 -b19200 -Uhfuse:w:0x57:m  
-Ulfuse:w:0x62:m
```

low fuses for 8 MHz:

```
./avrdude  
-C/home/mcciocci/.arduino15/packages/ATTinyCore/hardware/avr/1.2.3/avrdude.conf -v -pattiny85 -carduino -P/dev/ttyACM0 -b19200 -Uhfuse:w:0x57:m  
-Ulfuse:w:0xE2:m
```

eg onze black dell om gewone fuses te branden aan 1 Mhz en enkele andere zaken zoals voor ledupkidz (uitgenome reset disable):

```
./avrdude  
-C/home/mcciocci/.arduino15/packages/ATTinyCore/hardware/avr/1.2.3/avrdude.conf -v -pattiny85 -carduino -P/dev/ttyACM0 -b19200 -e -Uefuse:w:0xFF:m  
-Uhfuse:w:0b11010111:m -Ulfuse:w:0x62:m
```

5. RESET disable via RESCUE SHIELD

1. Connect the HV rescue shield with USB cable to PC

2. Connect to port `/dev/ttyUSB0` for linux
Programmer does not matter, you will only talk via serial monitor to the board
3. Click on the icon for serial monitor, verify speed is **9600 baud**
4. You will see the info screen to select what chip you want to burn fuses for, for LEDUPKIDZ select 3: ATtiny 8 pin HVSP

```

/dev/ttyUSB0
3
Select mode:
1: ATmega (28-pin)
2: ATtiny2313
3: ATtiny (8-pin) / HVSP
Selected mode: ATtiny/HVSP
Insert target AVR and press button.
  
```

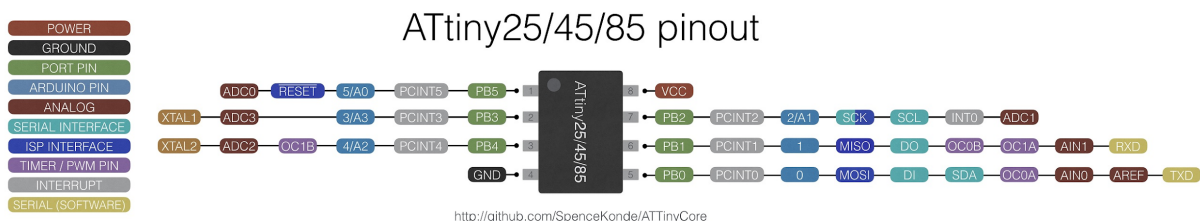
Usage to **burn the fuses for LedUpKidz AFTER setting a program on it** via Arduino as ISP:

1. Burn the default fuses to set an ATtiny back to default:
Send `0x62` for low fuses and **`0x57`** for high fuses
 - a. Enter desired LFUSE hex value: `0x62`
Enter desired HFUSE hex value: **`0x57`**

If you want to UNDO the reset disable: Burn the default fuses to set an ATtiny back to default:

1. Send `0x62` for low fuses and **`0xDF`** for high fuses
 - a. Enter desired LFUSE hex value (ie. `0x62`): `0x62`
Enter desired HFUSE hex value (ie. `0xDF`): `0xDF`

EXTRA - Bedrading Arduino as ISP to ATtiny

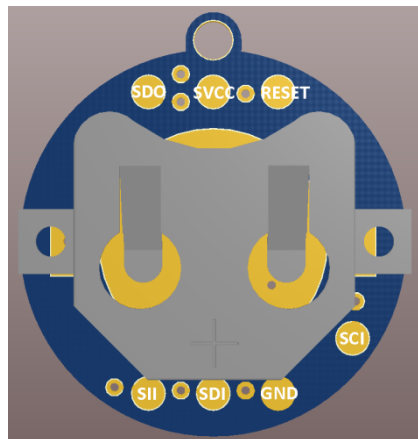
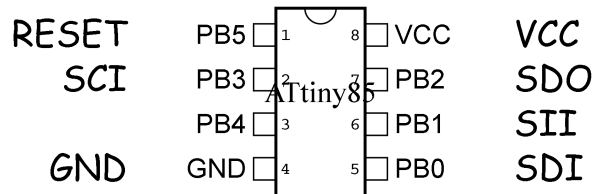


pins op attiny85 zijn

- 1 8
- 2 7
- 3 6
- 4 5

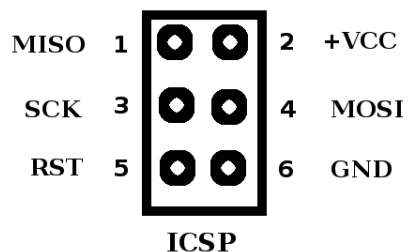
pin MISO - PB1 - 6 to D12 on arduino as ISP

pin VCC - 8 to 5V on arduino as ISP
 pin SCK - PB2 - 7 to D13 on arduino as ISP
 pin MOSI - PB0 - 5 to D11 on arduino as ISP
 pin RST - PB5 - 1 to D10 on arduino as ISP
 pin GND - 4 to GND on arduino as ISP



EXTRA - ICSP

Je kan met deze programmer alle devices programmeren die een ICSP header hebben, dus ook andere Arduino's. Die ICSP heeft volgende 6 pinnen:



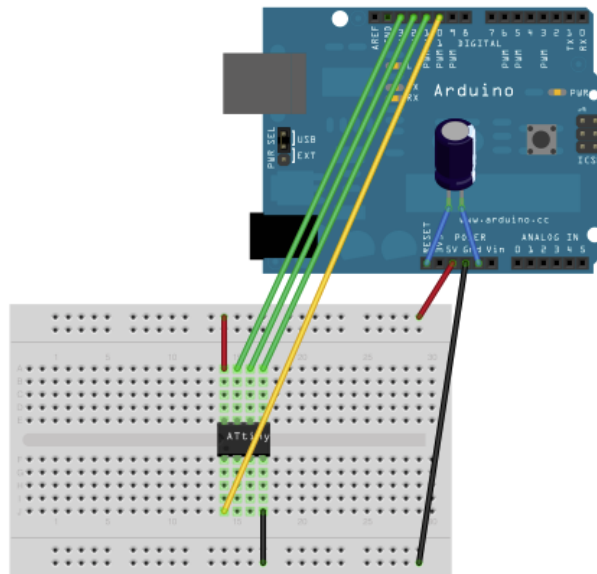
En deze pinnen van 1 tot 6 (normaal staat er een nummer 1 op de PCB zodat je oriëntatie juist hebt van de ICSP) moeten connecteren naar de Arduino ISP als:

pin 1 to D12 on arduino as ISP
 pin 2 to 5V on arduino as ISP
 pin 3 to D13 on arduino as ISP
 pin 4 to D11 on arduino as ISP

pin 5 to D10 on arduino as ISP
pin 6 to GND on arduino as ISP

Referenties

- see the fuses on a attiny here:
<http://eleccelerator.com/fusecalc/fusecalc.php?chip=attiny861&LOW=62&HIGH=FF&EXTENDED=FF&LOCKBIT=FF>
- <https://www.instructables.com/id/Program-an-ATtiny-with-Arduino/> De bekabeling is:



So the Arduino needs to be loaded with the ISP program.

Error: If you obtain with above guide on upload the error

```
avrdude: stk500_getparm(): (a) protocol error, expect=0x14, resp=0x14
avrdude: stk500_getparm(): (a) protocol error, expect=0x14, resp=0x01
avrdude: stk500_initialize(): (a) protocol error, expect=0x14, resp=0x10
avrdude: initialization failed, rc=-1
        Double check connections and try again, or use -F to override this check.
```

Then you have or

1. you did the wrong wiring between arduino and attiny
2. you used a wrong condensor between reset and GND. This should be eg 10 uF, 63V. Slightly different condensators might work, but eg I received above error with a 10 uF, 50V. So try different condensor!

Programmeren van de Ingegno ooring 1.0.6 IDE

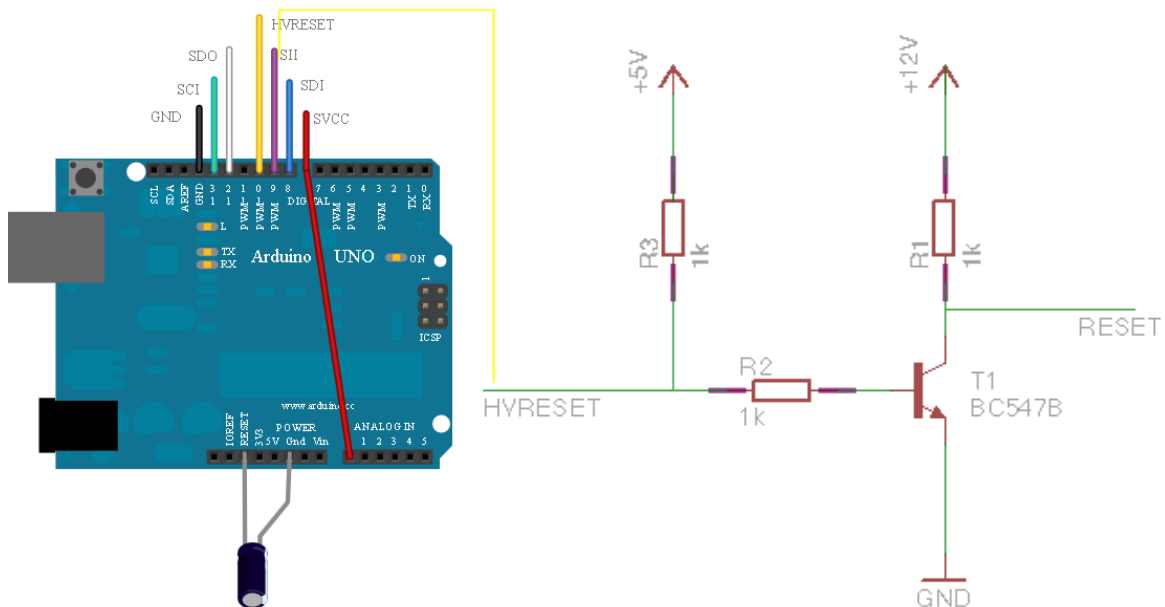
De ooring kan enkel via HVSP (High Voltage Serial Programming) geprogrammeerd worden, omdat de reset pin op de ooring een uitgang is en niet meer werkt als reset.

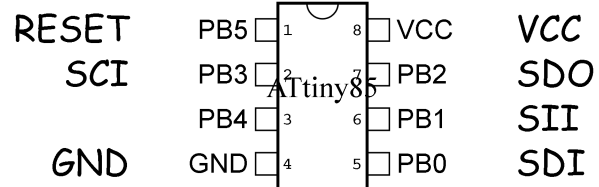
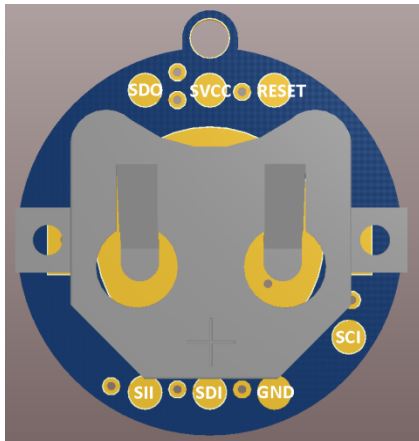
Alle informatie is te vinden op <http://microtherion.github.io/ScratchMonkey>

Benodigheden:

- Arduino (UNO)
- NPN transistor bv BC547 B
- 3 weerstanden 1K Ohm
- 12V voeding

Aansluiting Arduino/ooring:





Uitleg programmeren Arduino omgeving:

<http://microtherion.github.io/ScratchMonkey/GettingStarted.html>

Installation

ScratchMonkey is hosted on GitHub. You find a tar.gz on

<https://drive.google.com/drive/u/1/folders/0B70yEOOrXGk3remdyNFMxTEx2SW8>

To download a release, get it at <https://github.com/microtherion/ScratchMonkey/tags>.

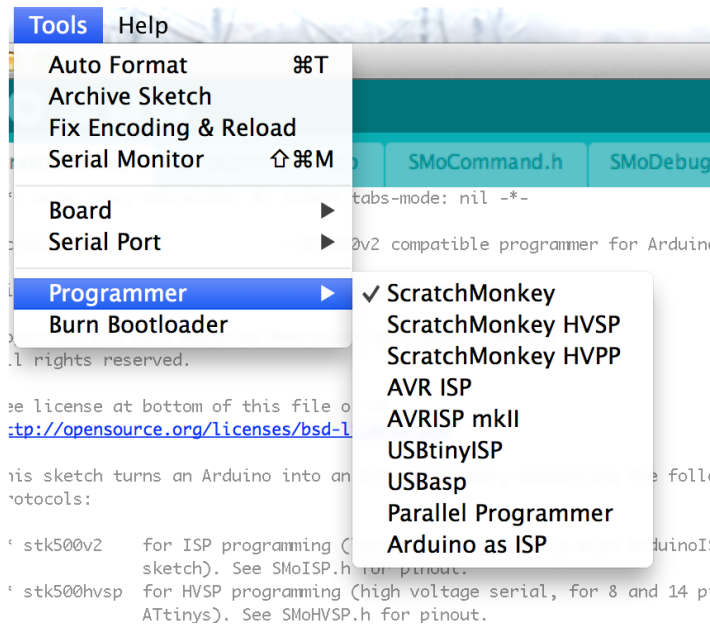
To get the latest version (which may or may not be in a tested, or even usable, state), clone or fork the master repository at <https://github.com/microtherion/ScratchMonkey>.

Once you have an unpacked copy of ScratchMonkey on your disk, **move the directory to your Arduino Sketchbook directory**. On a Mac, this is `~/Documents/Arduino` by default.

On linux: `~/sketchbook`

Finally, to make ScratchMonkey available in the Arduino IDE (Instructions given for Mac, but other platforms should work similarly):

If you want to go back to the default programmer later, this is normally **AVRISP mkII**



- Create a hardware directory in your Sketchbook directory, if it doesn't exist already:

```
mkdir ~/Documents/Arduino/hardware
```

Linux:

```
mkdir ~/sketchbook/hardware
```

- Copy or link the contents of the ScratchMonkey hardware directory to the Sketchbook hardware directory:

- `ln -s \`
`~/Documents/Arduino/ScratchMonkey/hardware/scratchmonkey \`
`~/Documents/Arduino/hardware/scratchmonkey`

Linux:

```
ln -s ~/sketchbook/ScratchMonkey/hardware/scratchmonkey  
~/sketchbook/hardware/scratchmonkey
```

If your installation was successful, the **Tools** → **Programmer** menu in your Arduino IDE should look like the picture on the right after your next restart.

Making ATTINY available in Arduino

Now we make attiny available as a target for the Arduino IDE. For this, you need to unzip the attiny.zip file in the hardware directory created above.

You find the zip in google drive (current master from <https://github.com/damellis/attiny/> does not work in Arduino 1.0.5).

Your sketchbook hardware directory should look like this now:

> Home > sketchbook > hardware

Name	Size	Modified
attiny 2 items 2/10/16 8:15 PM		
variants 2 items 5/18/15 8:41 PM		
boards.txt 4.4 KiB 5/19/15 5:24 PM		
scratchmonkey 1 item 9/27/15 6:27 PM		
programmers.txt 345 B 9/27/15 6:27 PM		

Connecting With The Target

Now it's time to connect your Arduino to your target MCU. The details for this vary between the different programming methods, and will be discussed in the individual sections.

Uploading the ScratchMonkey sketch

DO NOT DO THIS SECTION ON OUR EXISTING PROGRAMMER!! ONLY IF YOU MAKE A NEW PROGRAMMER, OTHERWISE SKIP THIS SECTION AND DO SECTION Burning a Bootloader

Before communicating with your target MCU, you first have to upload the ScratchMonkey software to your Arduino:

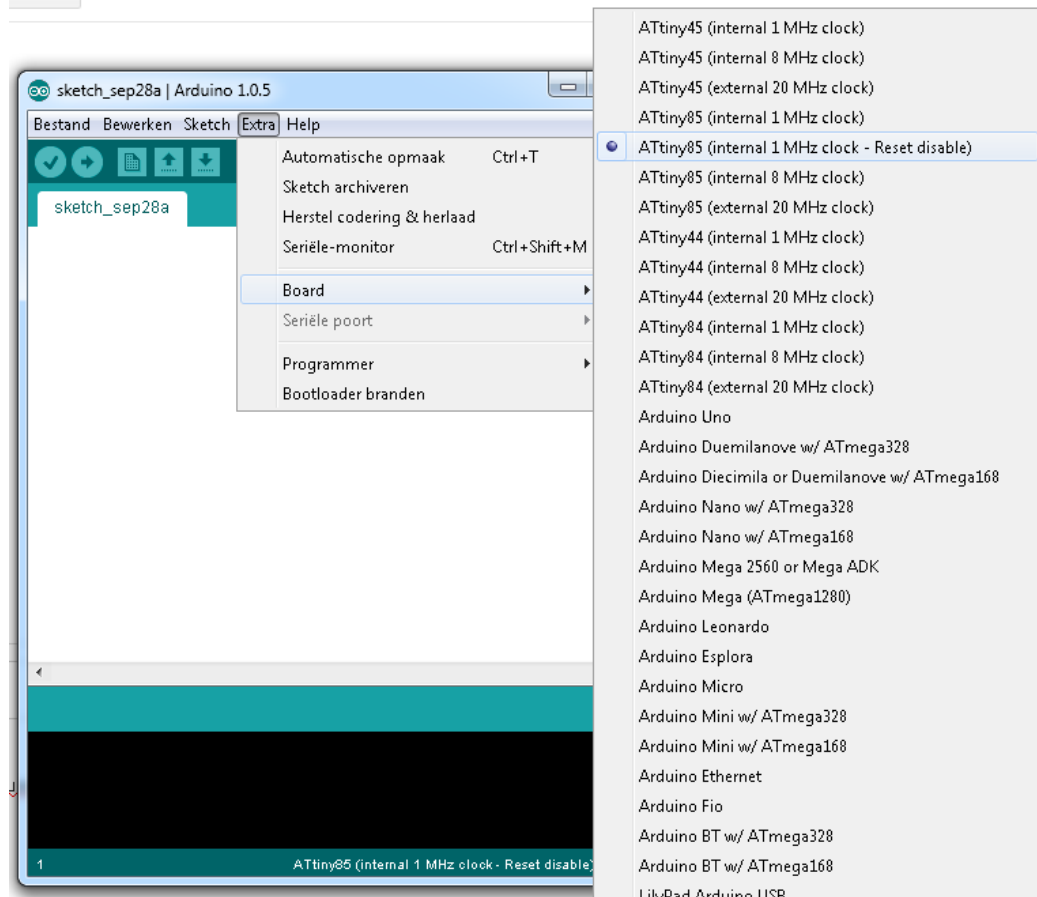
- Open the ScratchMonkey sketch in Arduino.
- From the `Tools` → `Board` menu, select your Arduino board, normally Arduino UNO, and make sure you have selected the normal programmer: **AVRISP mkII**.
- Upload the ScratchMonkey sketch.

Now that your Arduino is programmed, it's often helpful to temporarily disable its **RESET** pin (This is not necessary on an Arduino Micro or Leonardo). On an Arduino Uno, this can be done by connecting a 10μF capacitor between the **RESET** and **GND** pins (With the "-" side of the capacitor to **GND**). Doing this prevents the board from resetting when `avrdude` connects to it, which will at least speed up the target programming process.

Burning a Bootloader

Now it's time to communicate with your target. In a **first step**, you may need to upload a bootloader to your target and/or reprogram its fuses (Despite the name, `Tools` → `Burn Bootloader` will NOT necessarily **burn a bootloader**. With typical ATtiny cores, only the fuse settings are programmed with this item.)

- From the Tools → Board menu, select your target processor or board. For the earring ATtiny85 (internal 1MHz clock- reset disable)



- From the Tools → Programmer menu, select:
 - ScratchMonkey for In-System Programming
 - **ScratchMonkey HVSP** for High Voltage Serial Programming → **we do this HVSP !!**
 - ScratchMonkey HVPP for High Voltage Parallel Programming
- Select Tools → Burn Bootloader.

Uploading a Sketch

To upload your actual program:

- Open your sketch in the Arduino IDE.
- Select your board (ATtiny 85 (internal 1Mhz clock - reset disable)) and programmer (**ScratchMonkey HVSP**) as above.
- Upload your sketch.

Dus eenmaal ScratchMonkey.ino op de arduino staat, selecteer ik als board ATtiny 85 (internal 1Mhz clock - reset disable), kies juiste programmer, en dan doe ik burn loader.

Dan voor uploaden van een sketch., datzelfde ATtiny bord geselecteerd houden.
Gewoon zien dat ik bij gewone Arduino niet vergeet alles weer gewoon te zetten.

De 12V is niet zo nauwkeurig, 10V (8V) werkt ook. Je kan de stroom zo veel mogelijk dichtdraaien, om een eventuele kortsluiting in te perken. (maar dat zou niet mogen gebeuren).

Een testprogramma:

```
// Led Up Girlz Demo code -*-
//
// LedUpGirlz 0.1
//
// File: LedUpGirlz.ino          - Main program of sketch showing example use
//
// Copyright, Public Domain :
//      Ingegno.be              - 2015
//      Peter Vanveerdeghem - 2015,

#define LED0 5
#define LED1 2
#define LED2 1
#define LED3 0
#define LED4 4
#define LED5 3

void setup(){

    pinMode(LED0, OUTPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    pinMode(LED5, OUTPUT);

    digitalWrite(LED0, HIGH);
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, HIGH);
}

void loop(){

    for (int i=0; i <= 5; i++){
        digitalWrite(LED5, HIGH);
        digitalWrite(LED0, LOW);
        delay(100);
        digitalWrite(LED0, HIGH);
        digitalWrite(LED1, LOW);
        delay(100);
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, LOW);
        delay(100);
        digitalWrite(LED2, HIGH);
        digitalWrite(LED3, LOW);
        delay(100);
        digitalWrite(LED3, HIGH);
        digitalWrite(LED4, LOW);
        delay(100);
        digitalWrite(LED4, HIGH);
        digitalWrite(LED5, LOW);
        delay(100);
    }

    for (int i=0; i <= 20; i++){
        digitalWrite(LED0, HIGH);
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, HIGH);
        digitalWrite(LED3, HIGH);
        digitalWrite(LED4, HIGH);
        digitalWrite(LED5, HIGH);
        delay(50);
        digitalWrite(LED0, LOW);
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, LOW);
    }
}
```

```

digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);
    delay(50);
}

digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);

for (int i=0; i <= 5; i++){
    digitalWrite(LED5, LOW);
digitalWrite(LED0, HIGH);
delay(100);
digitalWrite(LED0, LOW);
digitalWrite(LED1, HIGH);
delay(100);
digitalWrite(LED1, LOW);
digitalWrite(LED2, HIGH);
delay(100);
digitalWrite(LED2, LOW);
digitalWrite(LED3, HIGH);
delay(100);
digitalWrite(LED3, LOW);
digitalWrite(LED4, HIGH);
delay(100);
digitalWrite(LED4, LOW);
digitalWrite(LED5, HIGH);
delay(100);
}

digitalWrite(LED0, HIGH);
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);
digitalWrite(LED4, HIGH);
digitalWrite(LED5, HIGH);
}

```

Bestukken Ooring

Neem de stukken en leg ze op een wit blad A4 papier:

6 led D0 tot D5

6 bijhorende weerstanden, R0 tot R5

- blauw: 330R
- groen: test met 820R Nee, teveel! Test 330R
- rood: test met 820R Nee, teveel! Test 330R
-