**ATTINY PROGRAMMER**

**En toegepast op een ultrasone eindeloop schakelaar**

Frank Van Aelst

Inhoud

[1. Inleiding 4](#_Toc437120309)

[2. Mogelijkheden van de ATtiny 5](#_Toc437120310)

[3. Programmeren 6](#_Toc437120311)

[3.1 De Hardware 6](#_Toc437120312)

[3.2 De Software 7](#_Toc437120313)

[3.3 De bootloader 9](#_Toc437120314)

[3.4 De Finale 9](#_Toc437120315)

[Stap 1: Arduino voorbereiden 9](#_Toc437120316)

[Stap 2: De Arduino programmeren als ISP 10](#_Toc437120317)

[Stap 3: De bootloader van je ATtiny aanpassen 11](#_Toc437120318)

[Stap 4: De sketch in de ATtiny85 schuiven 13](#_Toc437120319)

[3.5 WAARSCHUWING: 14](#_Toc437120320)

[4. De Ultrasone Eindstop 16](#_Toc437120321)

[4.1 Test op de Arduino 16](#_Toc437120322)

[4.2 Vertalen naar de ATtiny85 19](#_Toc437120323)

[5. Manufacturing 21](#_Toc437120324)

[6. Bronnen 23](#_Toc437120325)

[6.1 ATtiny-tips 23](#_Toc437120326)

[6.2 Onderdelen 23](#_Toc437120327)

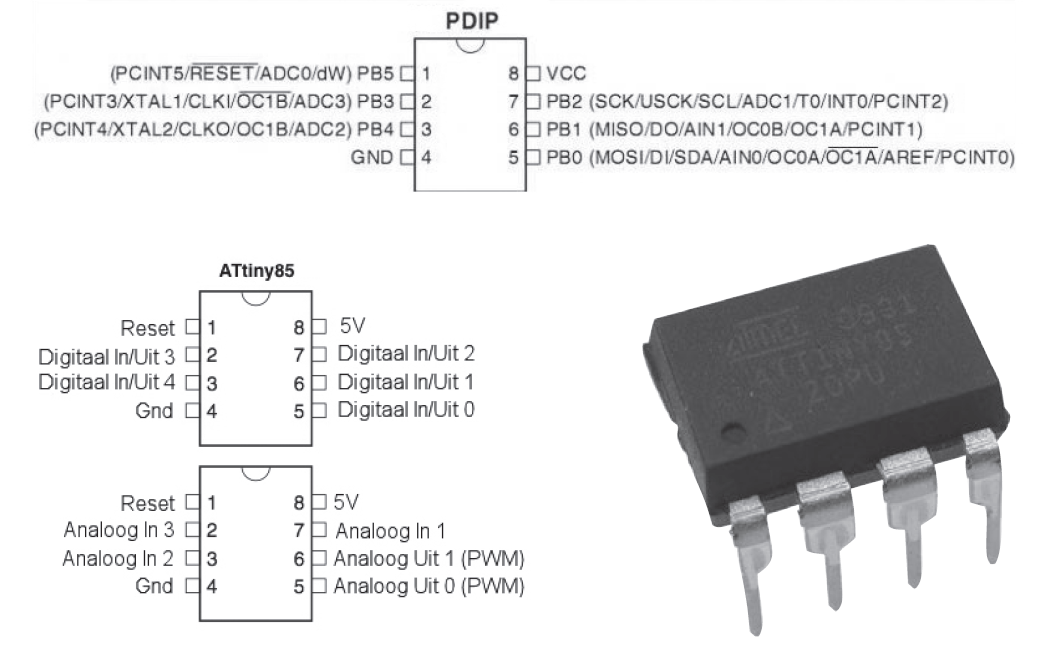
[6.3 Extras 23](#_Toc437120328)

# Inleiding

Een Arduino is een tof speeltje, en al is hij niet groot, en via onze Chinese leverancier niet duur, toch is hij soms wat overkill. De gebruikte chip die op het board zit is de Atmega328. Deze heeft in zijn uitgebreide familie nog wat broertjes die minder kunnen, nog kleiner zijn en zelfs nog goedkoper (via hetzelfde kanaal).

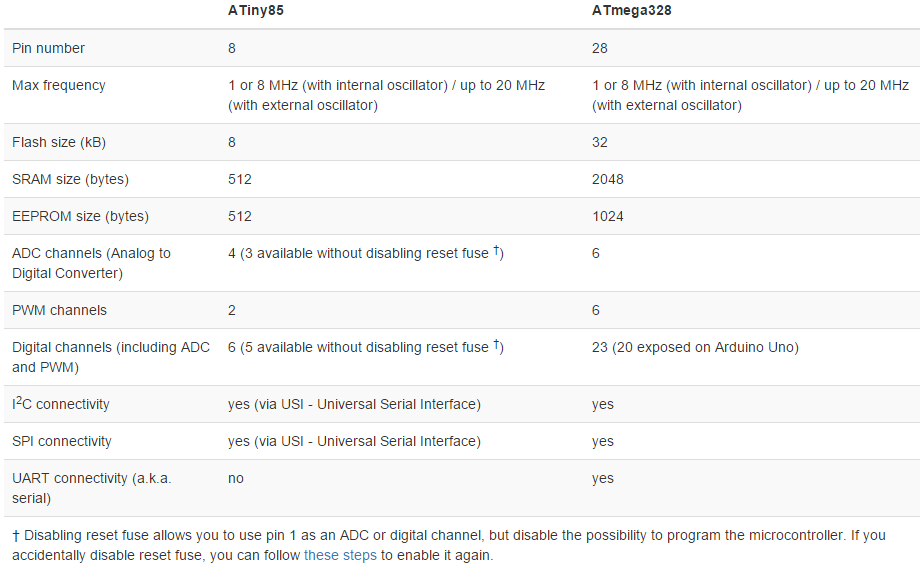
Als oldtimer fanaten en auto-prutsers hebben we een autolift op de kop weten te tikken. Op die lift staat wel een eindstop, zodat de auto niet door het dak gaat, maar voor grotere (hogere) wagens is het toch beter om vroeger de pompmotor uit te schakelen. Simpel projectje, dacht ik: een lichtje en een sensor, wat elektronica, een relais en wat klein spul. Klein probleem: overdag kan het zijn dat de sensor de juiste “verlichting” niet ziet en de lift vrolijk omhoog laat gaan met als gevolg: een sportmodel van een busje gemaakt. Dan maar met ultrasone techniek, dit is al beter controleer baar, maar vraagt weer een pak materiaal meer… tenzij we met een Arduino gaan werken.

Al snel zien we dat het scriptje hiervoor twee keer niks is en de benodigde I/O pinnen ook al niet veel voorstellen. Dus waarom niet een ATtiny85 gebruiken? Dit is een 8-pins controllertje met 5 beschikbare I/O pinnen. Uit de datasheets blijkt ook dat iedere pin 40mA mag geven, maar wel met een totaal van 60mA maximum aan alle poorten tegelijk. Het IC lust maximum 6V en 200mA.



# Mogelijkheden van de ATtiny

We weten al dat het programmeren van een Arduino zeer eenvoudig is. Een lange lijst van commando’s ligt voor ons open. Lange programma’s kunnen vlot in het geheugen van de Atmel Atmega 328. Er is een ruimte voorzien van 32kB. Voor de ATtiny is dit iets beperkter, afhankelijk van het model, 2kB tot 8kB, zie de grote verschillen in onderstaande tabel.



Wanneer we beiden vergelijken dan kunnen we stellen dat een ATtiny een zeer eenvoudige versie is van de grotere ATmega328P, of Arduino. Dikwijls hebben we genoeg aan kleine toepassingen waarbij we een enkele actie ondernemen, waar dan nog weinig hardware aansturing nodig is en geen complexe beslissingen moeten genomen worden. Voldoet je project hieraan, dan is een ATtiny-tje genoeg. Trouwens ik noem hier de ATtiny 85, maar ook de andere zijn gelijkaardig, ATtiny 24, 25, 44, 45, 84, enz. ze verschillen alleen in geheugengrootte of pinaantal .

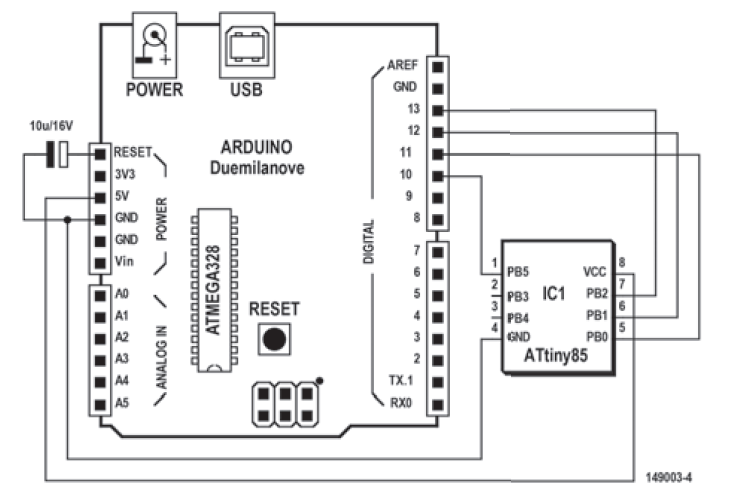
Om wat ruimte te sparen moet je dus efficiënt programmeren. De commando-lijst helpt je hierbij een beetje. Zo zijn er minder instructies beschikbaar. Zo kent hij de INPUT definitie niet, maar gaat ervan uit dat, als je niets definieert, het automatisch een input is. Zo zijn sommige Arduino libraries niet compatible met de ATtiny. De ATtiny kent ook geen I2C communicatie, wel de SPI versie (hiermee programmeer je hem trouwens). Doch er zijn trouwens voor vele dingen wel een workaround te vinden.

# Programmeren

Je vindt bij de Oosterse leveranciers voldoende programmer-boards voor de ATtiny’s voor een schappelijke of “andere” prijs. Maar het kan gratis als je al een Arduino en een breadboard bezit.

## De Hardware

Het grote schema om een programmeer toestel te bouwen is hieronder getekend: 6 draden, een Arduino Uno (de voorgestelde Duemilanove is kan ook) een capaciteit van 10uF/16V (microfarad) en een breadbordje. Meer heb je niet nodig. Het kan zelfs zijn dat de capaciteit ook niet nodig is, maar deze zorgt voor een filtering van eventuele storingen op de lijn ;-) Hij voorkomt ook dat de UNO ongewenst Reset als tijdens het programmeren de ATtiny wordt gereset, waardoor de bootloader wordt geactiveerd.

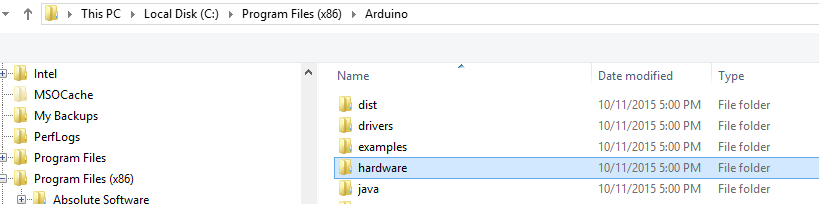


Dat is al, ben je van plan om meerder ATtiny’s te programmeren in de toekomst zou je dit op een stukje gaatjesprint kunnen zetten, bestudeer dan even of je het niet kan uitbreiden met een socket voor de ATtiny84 of een ander model. De Capaciteit kan je met een jumpertje eventueel in of uitschakelen. Mogelijkheden te over. Let wel, de Arduino werkt in gewoon gebruik niet met de capaciteit!

## De Software

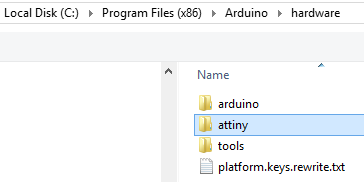
Zorg dat je een recente versie van de Arduino software hebt. <http://www.arduino.cc/en/Main/Software>

Check of je in de Arduino map, een map “hardware” hebt staan. Indien niet, dan maken we die aan, geen hoofdletters gebruiken!



Download “attiny-ide-1.6.x.zip” van de website <https://github.com/damellis/attiny/tree/ide-1.6.x>

Ga in de zipfile tot je de directory “attiny” vindt. Kopieer dit naar de hardware-map van daarnet.

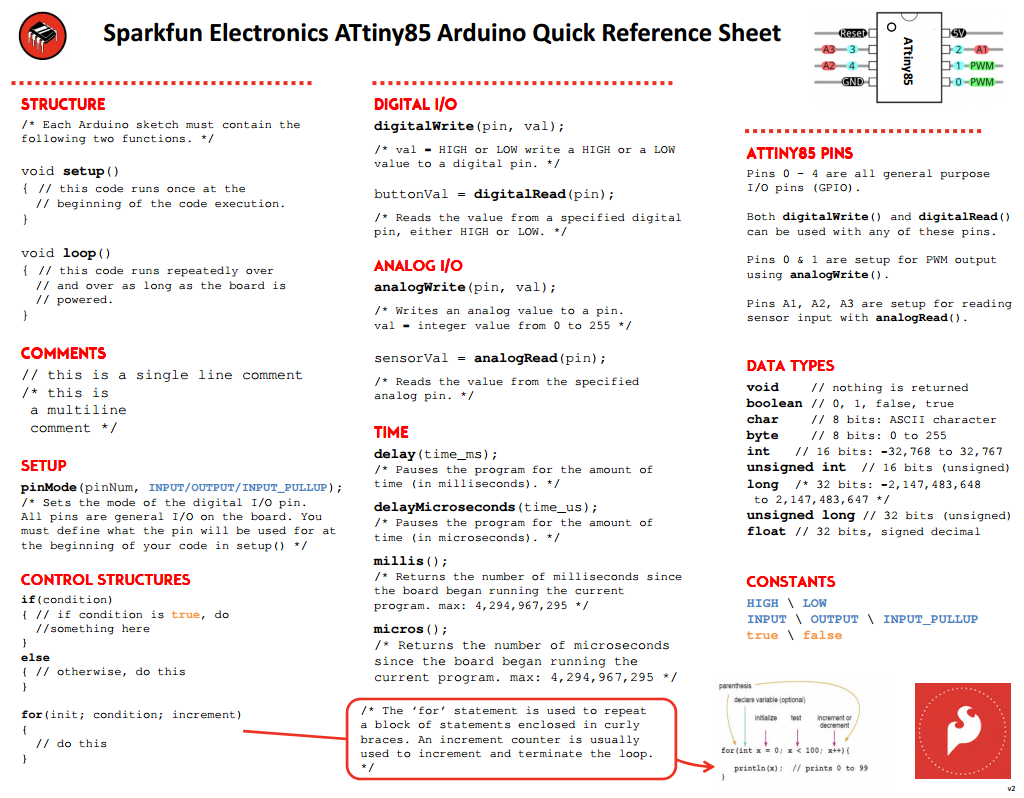


Tot zover de installatie, maar voor de zekerheid even testen. Start je Arduino programma op en kijk bij “Extra” “Boards” of de lijst is uitgebreid met de ATtiny familie.

Voor we programmeren toch nog even herinneren dat “tiny” het Engelse woord voor klein is, zoals hoger aangehaald geldt dit ook voor de instructieset, een overzicht:

<https://cdn.sparkfun.com/assets/e/9/1/f/5/Tiny_QuickRef_v2.pdf>

kopie hiernaast

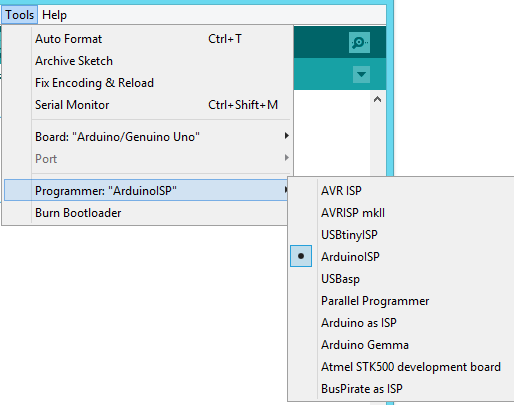


Kort samengevat, de volgende Arduino commandos worden gesupporteerd:

* pinMode()
* digitalWrite()
* digitalRead()
* analogWrite()
* analogRead()
* shiftOut()
* pulseIn()
* millis()
* micros()
* delay()
* delayMicroseconds()
* SoftwareSerial

## De bootloader

Hiermee neem je een beslissing met welke snlheid de interne klok van de ATtiny zal draaien 1MHz, 8MHz of 20MHz. Zal je er een externe clockgenerator aanhangen of niet? Dit laatst is van groot belang, bij de keuze “Extern” zal de ATtiny niet werken zonder die externe clock-generator! In principe moet je maar eenmaal de bootloader laden, hij blijft tot je hem overschrijft. Een nieuwe shetch raakt niet aan de bootloader.



## De Finale

We gaan testen of alles wel werkt. Het eenvoudigste programma dat we kennen is het “Blink” programmaatje. Dit sturen we naar een ATtiny.

Het programmeren gebeurt in meerdere stappen, eerst laden we het programma in de Arduino, daarna schuiven we het in de ATtiny.

### **Stap 1:** Arduino voorbereiden

* Zet nog geen ATtiny in je nieuwe programmeertoestel.
* Zorg dat je bij Extra (Tools) de Programmer op “ArduinoISP” staan hebt. Zou normaal zo zijn als je de Arduino nog niet als programmer hebt gebruikt.
* Laad het “Blink” programma.
* Pas de OUTPUT poort aan, I/O 13 wordt I/O 0, dus verander “13” door “0” hierdoor gaat de I/O pin 0 van de ATtiny als output dienen (we hebben maar 5 poorten, remember)

**Het aangepaste programma:**

void setup() {

// initialize digital pin 0 as an output.

pinMode(0, OUTPUT);

}

// the loop function runs over and over again forever

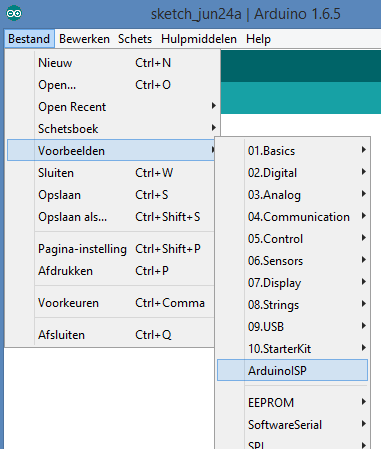
void loop() {

digitalWrite(0, HIGH); // turn the LED on

delay(1000); // wait for a second

digitalWrite(0, LOW); // turn the LED off

delay(1000); // wait for a second

}

Je zou het kunnen testen op je Arduino, maar dan moet je een LED met weerstand aansluiten op I/O 0

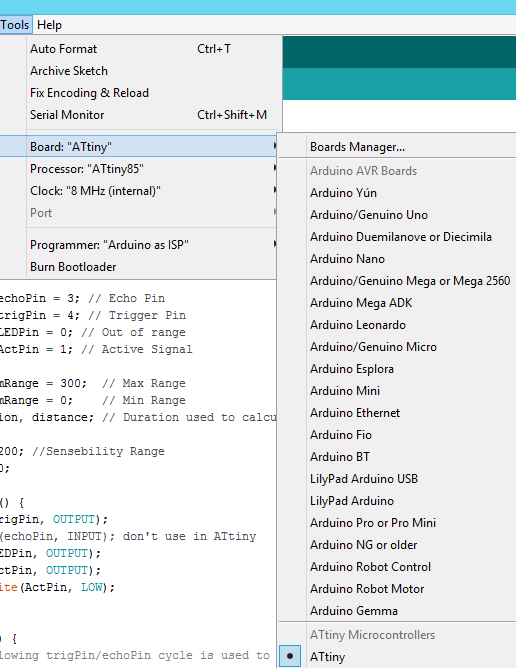
### **Stap 2:** De Arduino programmeren als ISP

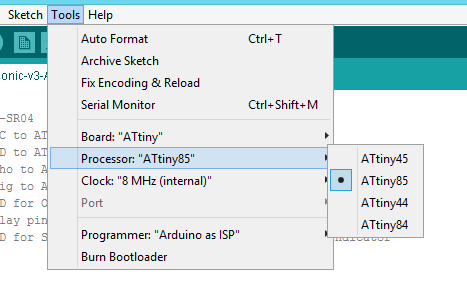
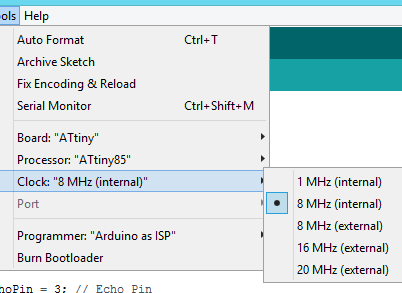
* Kies “ArduinoISP” uit bestand > voorbeelden
* Upload dit naar je Arduino

### Stap 3: De bootloader van je ATtiny aanpassen

Je zou dit kunnen overslaan als je tevreden bent met de instelling, maar indien je wilt wijzigen:

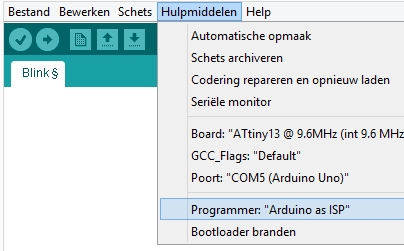
* Kies het juist “Board” : ATtiny



* Kies de juiste “Processor” ATTiny 85
* Selecteer de gewenste Clock frequentie

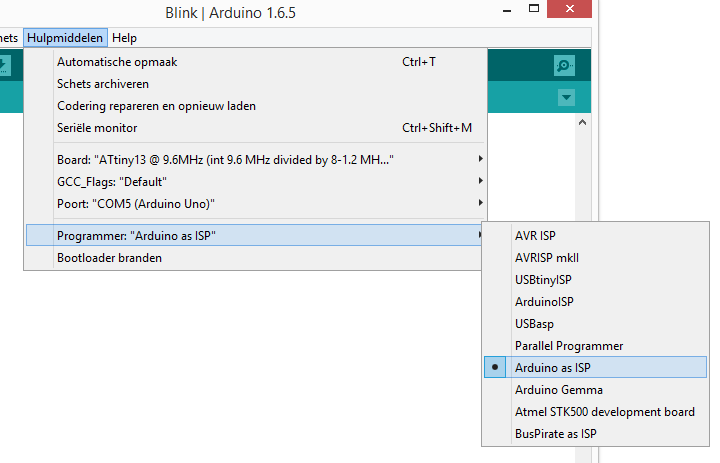
Zoals gezegd hier moet je uiterst voorzichtig te werk gaan, kies INTERNAL als je geen externe clock gaat gebruiken, anders moet je, om het te kunnen terugzetten, een externe clock gaan zoeken.

* Brand nu de bootloader
* Zet nu pas de ATtiny op zijn plaats (voetje of breadboard)

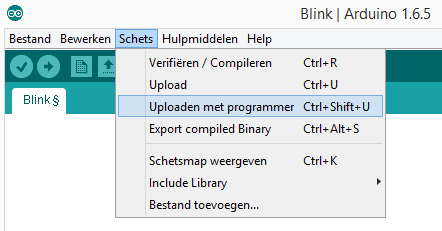


### Stap 4: De sketch in de ATtiny85 schuiven

* Zet nu de Arduino als programmeer toestel: Extra (Tools) “Arduino as ISP”



* Laad de sketch in de ATtiny85



* Haal nu je IC uit de programmer
* Als alles is goed gegaan dan kan je op pin 5 van het IC een LED hangen ,
* 5V aan pin 8 en op pin4 GND

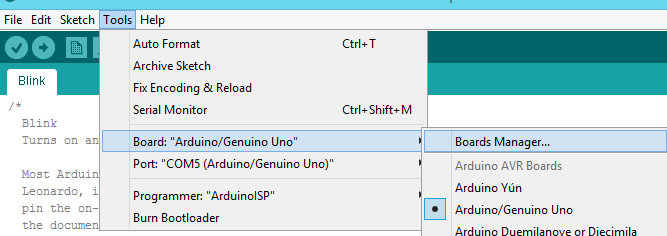
3.5 WAARSCHUWING:

Bij mij liep het mis de eerste keer, ineens waren alle Arduino’s niet meer compatible en kreeg ik een foutmelding : “**avr-gcc: error: C:\Users\a2\AppData\Local\Temp\build859542859958490258.tmp/core.a: No such file or directory  
Error compiling.”**

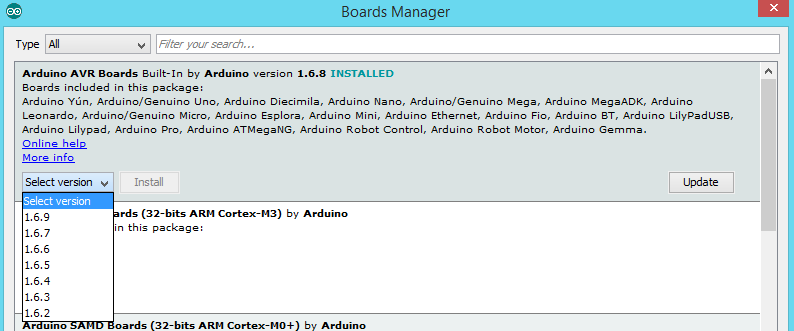
Oplossing : <https://forum.arduino.cc/index.php?topic=358313.0>

Kort samengevat moet je het volgende doen:

* Open de Boards Manager



* Wacht tot hij volledig is geladen.
* Klik op Arduino AVR Boards



* Selecteer “Select Version” en kies “1.6.8”, versie 1.6.9 bevat een bug volgens het forum
* Klik “Install”
* Close en start opnieuw op. Nu zou alles moeten werken.

# De Ultrasone Eindstop

In de inleiding wees ik er op dat het de bedoeling is om te voorkomen dat een voertuig op een autolift niet hoger komt dan 20cm onder het dak. Hiervoor ga ik de afstand op die hoogte meten tussen de sensor en een voorwerp (auto) dat passeerd.

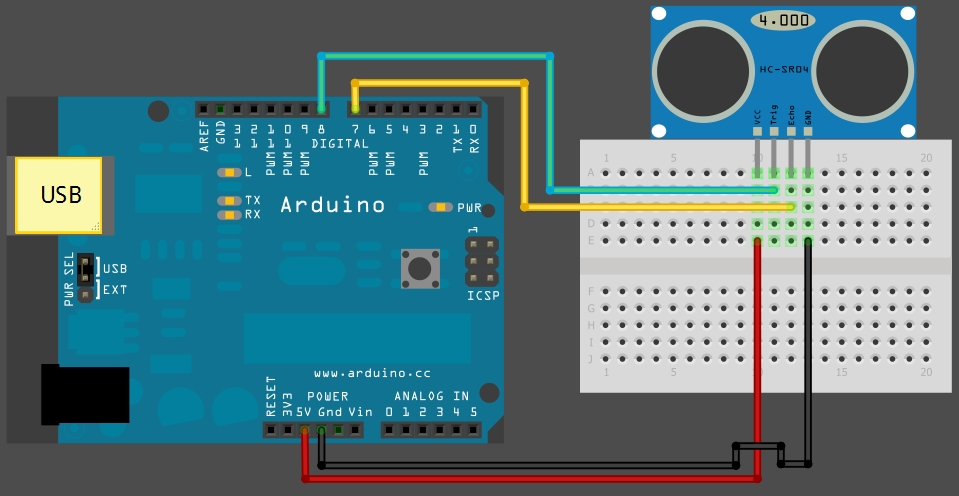
Enkele praktische zaken:

* De liftpalen staan 4m uit elkaar.
* Het dak is schuin, het laagste punt waar de auto zou kunnen komen is 3,4m
* De elektrisch schema van de lift kan mij voorzien van spanning 25V ongefilterde DC spanning
* Door een relaiscontact in serie te zetten met de huidige eindeloop schakelaar kan ik de pomp uitschakelen.

Met andere woorden: als ik een afstand meet op 3.2m hoogte en zie dat er binnen een range tot 2m een voorwerp is dan moet de lift stoppen. Bij het zakken gaat de lift eerst nog 10cm omhoog om uit de mechanische vergrendeling te geraken, vandaar die 20cm.

## Test op de Arduino

Wat hebben we nodig? Een ultrasoon module, type HC-SR04. Een Arduino, wat draadjes, weerstanden en LEDs.



Met de LEDs simuleren we het relais en “out of range”

**Het programma:**

/\*

HC-SR04

VCC to arduino 5v

GND to arduino GND

Echo to Arduino pin 7

Trig to Arduino pin 8

LED for Out of range pin 13

LED simulating the Relay pin 11

LED for Signal Active pin 9 // shall be the same function as pin 11 in the final design

\*/

const int echoPin = 7; // Echo Pin

const int trigPin = 8; // Trigger Pin

const int LEDPin = 13; // Out of range

const int ActPin = 9; // Active Signal

const int RelPin = 11; // Relay Pin

int maximumRange = 300; // Max Range

int minimumRange = 0; // Min Range

long duration, distance; // Duration used to calculate distance

int Max = 20; //Sensebility Range, here a small range just to do the test on my desk

int Min = 10;

void setup() {

Serial.begin (9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(LEDPin, OUTPUT);

pinMode(ActPin, OUTPUT);

pinMode(RelPin, OUTPUT);

digitalWrite(ActPin, LOW);

digitalWrite(RelPin, LOW);

}

void loop() {

/\* The following trigPin/echoPin cycle is used to determine the

distance of the nearest object by bouncing soundwaves off of it.

\*/

digitalWrite(trigPin, LOW); // the pulse generation

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); // check when the pulse is coming back

//Calculate the distance (in cm) based on the speed of sound.

distance = duration/58.2;

if (distance >= maximumRange || distance <= minimumRange){

/\* Send a negative number to computer and Turn LED ON

to indicate "out of range" \*/

Serial.println("-1");

digitalWrite(LEDPin, HIGH);

}

else {

/\* Send the distance to the computer using Serial protocol, and

turn LED OFF to indicate successful reading. \*/

Serial.println(distance);

digitalWrite(LEDPin, LOW);

if (distance > Min && distance < Max){

digitalWrite(ActPin, HIGH);

digitalWrite(RelPin, HIGH);

delay(1000); //put relays actieve for 1~~0~~ seconds, whithin this time the operator should stop pushing the UP button

digitalWrite(RelPin, LOW);

digitalWrite(ActPin, LOW);

}

}

}

**Uitleg:**

Het programma is deels overgenomen van de Arduino website.

Het relais wordt tijdens de proefopstelling vervangen door een LED (pin11). Op pin9 staat nog een LED die eigenlijk dezelfde functie heeft als de voorgaande, ik had dit voorzien voor het geval ik nog ergens een controle wou inbouwen. De LED op pin 13 gaat branden wanneer er geen signaal terugkomt binnen een aanvaarbare termijn. Deze wordt eigenlijk bepaald door het meetbereik (Max 300cm). Dit meetbereik loopt van 0 tot 300cm, maar het gebied dat mij interesseert is het bereik van 10 tot 20cm (de afstand op mijn bureau). In eerste instantie gaat er een puls worden gegeneerd en uitgestuurd (2mms hoog, dan 10mms laag *–mms = microseconden*). Met pulseIn wachten we tot wanneer het signaal terugkomt. De tijd die dit duurde wordt gedeeld door een constante om de afstand in cm te hebben. Afhankelijk van deze uitkomst wordt er beslist of dit Out of Range was (>300cm) of binnen de ingestelde range (10 – 20cm). Het relais onderbreekt gedurende 10 seconden de pomp, in die tijd moet de operator de UP-knop losgelaten hebben. (in principe gebeurt er dan niets omdat het systeem dan terug actief wordt en terug uitschakeld)

Ter informatie: de snelheid van het geluid in de lucht varieert van 307m/s (-40C) tot 354m/s (+40C). Bij 20C is dit 343m/s, en dit bij een gemiddelde vochtigheidsgraad. M.a.w. willen we een nauwkeurig meettoestel bouwen moet hiermee rekening gehouden worden, dit is hier zeker niet nodig.

## Vertalen naar de ATtiny85

In het programma zijn er wat meer LEDs gebruikt dan noodzakelijk. Die verdwijnen uit het programma, ook de seriële controle naar de serial monitor mag weg uit het ATtiny programma.

**Het programma:**

/\*

HC-SR04

VCC to ATtiny85 pin8 5v

GND to ATtiny85 pin4 GND

Echo to ATtiny pin2 Dig3

Trig to ATtiny pin3 Dig4

LED for Out of range pin 5 Dig0

Relay pin 6 Dig1

LED for Signal Active pin 6 Dig1 = relais active indicator

\*/

const int echoPin = 3; // Echo Pin

const int trigPin = 4; // Trigger Pin

const int LEDPin = 0; // Out of range

const int ActPin = 1; // Active Signal

int maximumRange = 300; // Max Range

int minimumRange = 0; // Min Range

long duration, distance; // Duration used to calculate distance

int Max = 200; //Sensebility Range

int Min = 0;

void setup() {

pinMode(trigPin, OUTPUT);

// pinMode(echoPin, INPUT); not applicable in ATtiny

pinMode(LEDPin, OUTPUT);

pinMode(ActPin, OUTPUT);

digitalWrite(ActPin, LOW);

}

void loop() {

/\* The following trigPin/echoPin cycle is used to determine the

distance of the nearest object by bouncing soundwaves off of it.

\*/

digitalWrite(trigPin, LOW); // the pulse generation

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); // check when the pulse is coming back

//Calculate the distance (in cm) based on the speed of sound.

distance = duration/58.2;

if (distance >= maximumRange || distance <= minimumRange){

/\* Turn LED ON to indicate "out of range" \*/

digitalWrite(LEDPin, HIGH);

}

else {

// turn LED OFF to indicate distance whithin range

digitalWrite(LEDPin, LOW);

if (distance > Min && distance < Max){

digitalWrite(ActPin, HIGH);

delay(10000); //put relays active for 10 seconds, whithin this time the operator should stop pushing the UP button

digitalWrite(ActPin, LOW);

} } }

De sketch is vereenvoudigd, de serial monitor is weg, de INPUT is gedeactiveerd. De meetafstanden zijn nu aangepast:

* de juiste pinnummers staan in de inleiding,
* meetbereik tot 3m, alles erboven geeft een “Out of range” signaal, > LED,
* actief bereik tussen 0 en 2m, het relais trekt aan en een LED brand.

# Manufacturing

Na het idee te hebben ge-prototyped moeten we er een bruikbaar geheel van maken. We weten al dat de lift mij een 24 Vac spanning geeft die enkel wordt gelijkgericht en me dus een ongefilterde 24Vdc geeft, dit geeft mij een 36V zuivere DC-spanning

De 36V moet naar 12V er moet 24V weg wat enorm vermogensverlies betekend. 24V overbruggen doe je tegenwoordig beter met een geschakelde voeding. Als we dan toch bezig zijn houdt er ons niets tegen om ineens vanaf de netspanning te vertrekken. Bij Aliexpress hebben ze voor een prijsje zulke dingen (<6euro voor een 12V/25W voeding) Die 12V is bruikbaar voor het relais en voor de ATtiny en de rest van de schakeling hangen we er voor minder dan een euro een switcheker aan die 5V maakt uit de 12V.

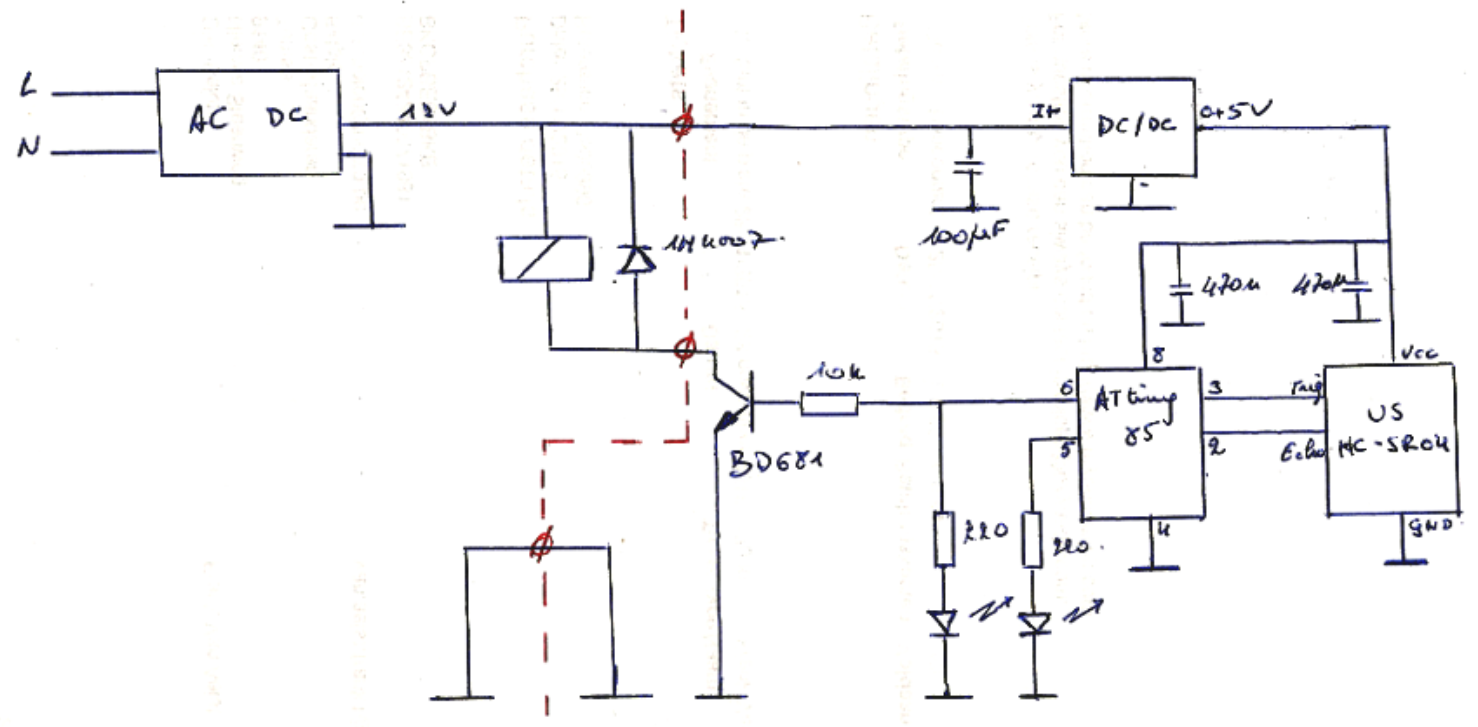
Het geheel verbruikt alles samen maximum 50mA op 5V (gemeten X veiligheidscoëfficiënt). Het relais heb ik nog niet, maar een 12V type zal worden gebruikt, met 6A contacten. Een gemiddeld relais verbruik al gauw een 1 a 1.5W om aan te trekken terwijl het houd vermogen al snel een 0.75W bedraagt, wat een spoelstroom geeft op 12V van 0.75W/12V = 62.5mA. Als piekstroom geeft dat 125mA. De ganse schakeling verbruikt worst case 175mA of 9W. de voeding is dan ruim voldoende.

Het ontwerp zal in twee delen worden gebouwd (stippellijn in het schema), enerzijds het deel dat de “zware stromen en spanningen voert, een ander deel dat bovenaan de lift is gemonteerd waar de meting wordt gedaan.

**Uitleg bij het schema:**

Via de AC/DC convertor wordt de netspanning omgezet naar 12V DC spanning. Via een andere omvormer wordt 5V gemaakt om de ATtiny85 en de ultrasone transducer te voeden. De ATtiny85 stuurt de beide LEDs en de transistor aan. De transistor zorgt ervoor dat het relais bediend wordt. Het relais wordt verder gevoed uit de 12V. De diode die over het relais staat dient om de zelfinductie stroom bij het schakelen op te vangen.

**Schema:**



# Bronnen

## ATtiny-tips

<https://en.wikipedia.org/wiki/Atmel_AVR_ATtiny_comparison_chart>

<https://goddess-gate.com/projects/en/arduino/attiny85vsatmega328>

<http://highlowtech.org/?p=1695>

<http://www.parkansky.com/arduino-error.htm>

<http://domoticx.com/attiny-programmeren-met-arduino-als-isp/>

<http://www.creativeapplications.net/tutorials/small-programmable-wearable-and-cheap-enjoying-the-attiny85-tutorial/>

## Onderdelen

<http://nl.aliexpress.com/item/5-sets-Lot-Coil-Power-Relay-LY2NJ-12V-DC-Miniature-Relay-DPDT-8-Pins-10A-240VAC/32346762802.html?detailNewVersion=2>

<http://nl.aliexpress.com/item/secure-quality-simple-operation-AC-220V-50-60HZ-to-DC-12V-25W-2A-Lighting-Transformer-SMD/32499983630.html>

<http://nl.aliexpress.com/item/free-shipping-2-pcs-lot-DC-DC-voltage-power-supply-module-3-a-adjustable-step-down/1767408831.html>

<http://nl.aliexpress.com/item/5pcs-lot-HC-SR04-Ultrasonic-module-SR04-Ultrasonic-distance-measuring-module-Ultrasonic-smart-car-sensors/32476575236.html>

<http://www.tme.eu/at/Document/c163f86796c830f0aff6cac97f19a590/BD675AS.pdf>

## Extras

<https://code.google.com/p/arduino-tiny/>

<http://playground.arduino.cc/Code/USIi2c>

<http://melgior.plazilla.com/page/4295007315/een-attiny-programmer-bouwen-voor-de-arduino-supermakkelijk>

<http://www.ebay.com/itm/New-1pc-USB-ISP-Programmer-For-ATMEL-AVR-ATMega-ATTiny-51-Development-Board-/221452134543?pt=LH_DefaultDomain_0&hash=item338f93688f>