# Digital-Analog Conversion

Kasper Rijksen

Joery van den Hoff

## Circuit online

Wij hebben de circuit van de caterpillar nagemaakt op de simulator ([www.falstad.com/circuit](http://www.falstad.com/circuit)).

Onze link is:

http://www.falstad.com/circuit/#%24+1+5.0E-6+23.47059216675035+63+5.0+43%0Ar+176+0+224+0+0+10000.0%0Ar+128+32+128+80+0+20000.0%0AS+128+80+128+128+0+0+false+0%0Aw+112+368+112+128+0%0Aw+144+304+144+128+0%0Aw+128+32+128+0+0%0Aw+176+0+128+0+0%0Aw+272+0+224+0+0%0Aw+224+32+224+0+0%0Aw+240+304+240+128+0%0Aw+208+368+208+128+0%0AS+224+80+224+128+0+1+false+0%0Ar+224+32+224+80+0+20000.0%0Ar+272+0+320+0+0+10000.0%0Ar+464+0+512+0+0+10000.0%0Ar+416+32+416+80+0+20000.0%0AS+416+80+416+128+0+1+false+0%0Aw+400+368+400+128+0%0Aw+432+304+432+128+0%0Aw+416+32+416+0+0%0Aw+464+0+416+0+0%0Aw+368+0+320+0+0%0Aw+320+32+320+0+0%0Aw+336+304+336+144+0%0Aw+304+368+304+144+0%0AS+320+80+320+144+0+1+false+0%0Ar+320+32+320+80+0+20000.0%0Ar+368+0+416+0+0+10000.0%0Ar+656+0+704+0+0+10000.0%0Ar+608+32+608+80+0+20000.0%0AS+608+80+608+128+0+1+false+0%0Aw+592+368+592+128+0%0Aw+624+304+624+128+0%0Aw+608+32+608+0+0%0Aw+656+0+608+0+0%0Aw+560+0+512+0+0%0Aw+512+32+512+0+0%0Aw+528+304+528+128+0%0Aw+496+368+496+128+0%0AS+512+80+512+128+0+1+false+0%0Ar+512+32+512+80+0+20000.0%0Ar+560+0+608+0+0+10000.0%0Ar+848+0+896+0+0+10000.0%0Ar+800+32+800+80+0+20000.0%0AS+800+80+800+128+0+1+false+0%0Aw+784+368+784+128+0%0Aw+816+304+816+128+0%0Aw+800+32+800+0+0%0Aw+848+0+800+0+0%0Aw+752+0+704+0+0%0Aw+704+32+704+0+0%0Aw+720+304+720+128+0%0Aw+688+368+688+128+0%0AS+704+80+704+128+0+1+false+0%0Ar+704+32+704+80+0+20000.0%0Ar+752+0+800+0+0+10000.0%0Aw+112+368+208+368+0%0Aw+896+0+896+304+0%0Aw+896+304+896+432+0%0Ag+784+432+784+448+0%0Av+784+432+784+368+0+0+40.0+5.0+0.0+0.0+0.5%0Aw+784+432+896+432+0%0Aw+128+0+0+0+0%0Aw+144+304+240+304+0%0Aw+240+304+336+304+0%0Aw+336+304+432+304+0%0Aw+432+304+528+304+0%0Aw+528+304+624+304+0%0Aw+624+304+720+304+0%0Aw+720+304+816+304+0%0Aw+816+304+896+304+0%0Aw+208+368+304+368+0%0Aw+304+368+400+368+0%0Aw+400+368+496+368+0%0Aw+496+368+592+368+0%0Aw+592+368+688+368+0%0Aw+688+368+784+368+0%0Aw+144+304+0+304+0%0Ap+0+0+0+304+1%0Ao+78+64+0+34+5.0+9.765625E-5+0+-1%0A

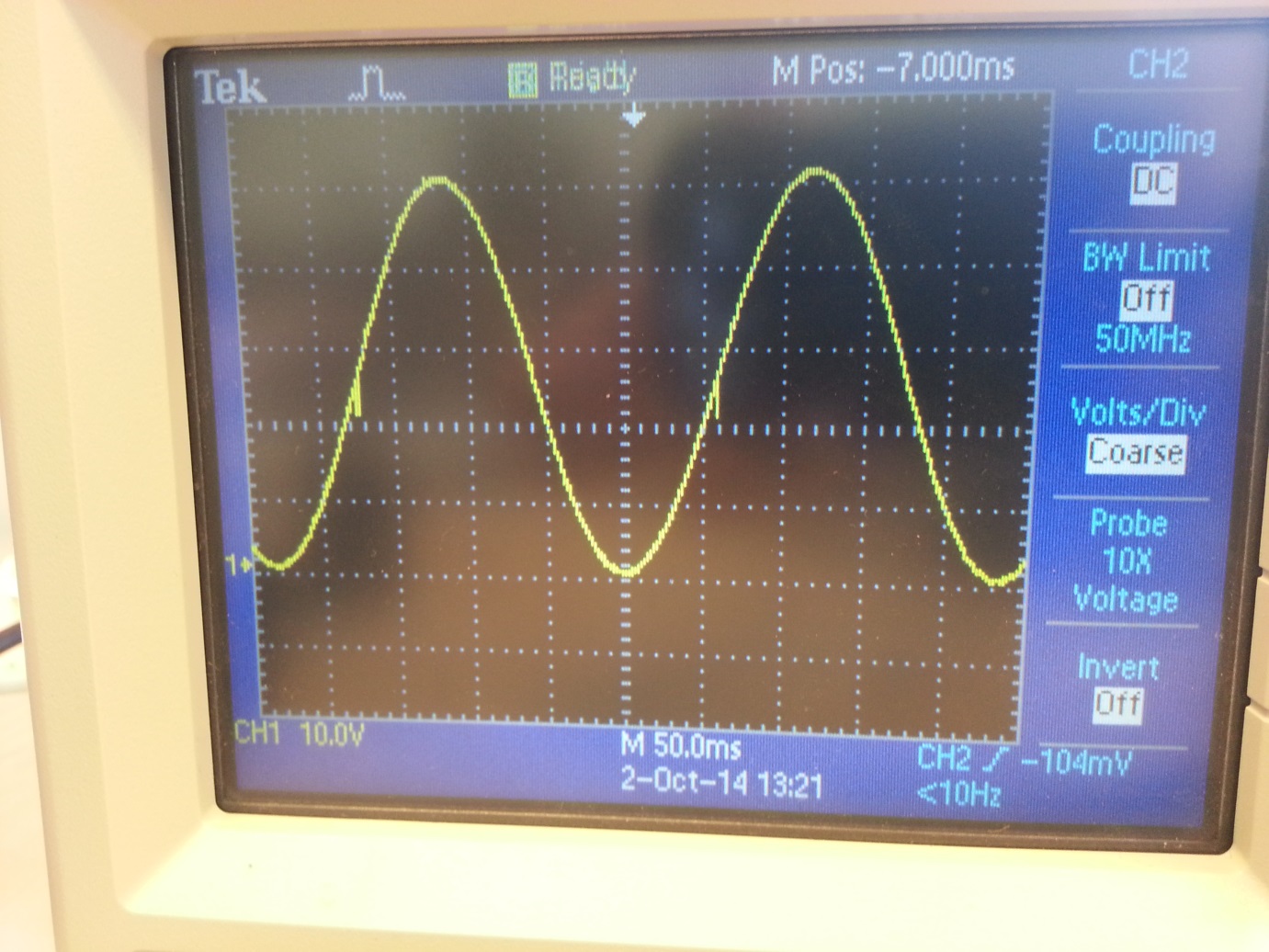
In het bestand ‘voltage.xlsx’, hebben we een aantal metingen opgeschreven. Op de x-ax staan de bit values van 0 tot 255 en op de y-as staat het bijbehorende voltage.

## Real hardware

Hierna hebben we het nagemaakt op onze breadboard met de caterpillar. We hebben een filmpje gemaakt waarin we laten zien dat het werkt op de osciliscoop:

<http://youtu.be/pMT4fqlmNEQ>

De sinus waaden die worden gebruikt door de arduino hebben wij gestopt in ‘sinusValues.xlsx’. Hier kan je zien dat de waardes overeenkomen met een sinus.



Hierboven is een hoge resolutie foto van de osciliscoop. Onze instelling was op dit moment 148/1023.

Dat betekent dat de OCR1A op ongeveer 10336 stond. Met prescale 1 betekent het dat er 10336 clock cycles voor 1 graden nodig is. Dus voor 360 graden zijn er 3721119 clock cycles nodig. Dat delen door de 16Mhz van onze arduino geeft ongeveer 233 ms. Op het plaatje hierboven kan je zien dat een sinus iets minder dan 5 hokjes nodig heeft. Een hokje is 50ms op het plaatje hierboven, dus dat komt overeen met wat wij verwacht hadden.

## Code toelichting

In setup() doen wij een aantal dingen:

* Wij genereren de 360 waardes van de sinus in een 8bit range.
* Wij zetten de analoge input klaar voor handmatige ADC conversie. (Wij schrijven naar 5 verschillende registers)
  + Wij zetten de ACME bit, dit is zodat we zelf een analoog signaal kunnen selecteren
  + Wij zetten ADC aan met de ADEN bit
  + Wij zetten de REFS0 bit om aan tegeven dat we de interne reference voltage willen gebruiken
  + Dan zetten wij de MUX bitjes, die aangeven welke analoge poort wij willen gebruiken (in dit geval is dat A0, dus 000)
  + En we zetten de power saving uit (PRADC) zodat we de ADC kunnen gebruiken.
* Wij zetten 8 digitale poorten als output, zodat we de volledige 8 bits conversie kunnen doen.
* We zetten de timer klaar met prescaler 1 en CTC mode. (overgenomen van vorige opdracht)

In loop() wordt er gekeken wat de waarde is van de potentiometer, en indien deze is aangepast wordt de OCR1A register geupdate (voor de timer CTC). Wij doen dit handmatig met registers in plaats van de analogRead() te gebruiken.

In de timer interrupt zorgen wij dat we de volgende waarde schrijven naar poort d en c. De 2 LSB worden geschreven op poort c en de rest wordt geschreven op poort d.