/\*  
 \* Opdracht 3.2 - ADC Baby!  
 \*   
 \* Eigenaar:  
 \* Maarten Paauw <s1094220@student.hsleiden.nl>  
 \* s1094220  
 \* INF3C  
 \*   
 \* Versie: 1  
 \* Aangemaakt: 23 februari 2018  
 \* Gewijzigd: 27 maart 2018  
 \*/  
  
#include <avr/io.h>  
#include <util/delay.h>  
#include <math.h>  
  
// Initialiseer de potential meter.  
void initADC()  
{  
 // Zet op 5 volt.  
 ADMUX |= (1 << REFS0);  
  
 // Divider op 128 voor 10 bit precisie.  
 ADCSRA |= ((1 << ADPS0) | (1 << ADPS1) | (1 << ADPS2));  
  
 // AD enable.  
 ADCSRA |= (1 << ADEN);  
}  
  
// Lees de potential meter.  
uint16\_t readADC()  
{  
 // Starten met lezen.  
 ADCSRA |= (1 << ADSC);  
  
 // Loop totdat er een waarde is.  
 loop\_until\_bit\_is\_clear(ADCSRA, ADSC);  
  
 // Geef de waarde terug.  
 return ADC;  
}  
  
// Dynamische timeout.  
void delay(uint16\_t time)  
{  
 // For loop.  
 for (uint16\_t i = 0; i < time; i++)  
 {  
 // Wacht voor 1 ms.  
 \_delay\_us(1);  
 }  
}  
  
// Preciezere afronding.  
// https://stackoverflow.com/questions/5731863/mapping-a-numeric-range-onto-another  
double round(double d)  
{  
 // Tel er 0.5 bij op en rond het getal af.  
 return floor(d + 0.5);  
}  
  
// Genereer de slope.  
// https://stackoverflow.com/questions/5731863/mapping-a-numeric-range-onto-another  
double slope (int output\_start, int output\_end, int input\_start, int input\_end)  
{  
 // Geef de slope terug.  
 return 1.0 \* (output\_end - output\_start) / (input\_end - input\_start);  
}  
  
// Map het getal van een range in een nieuwe range.  
// https://stackoverflow.com/questions/5731863/mapping-a-numeric-range-onto-another  
double map(int input, int input\_start, int output\_start, double static\_slope)  
{  
 // Geef het nieuwe getal terug.  
 return output\_start + round(static\_slope \* (input - input\_start));  
}  
  
// De main functie.  
int main(void)  
{  
 // B Bank initialiseren.  
 DDRB = (1 << PB5);  
  
 // Initialiseer de potential meter.  
 initADC();  
  
 // Defineer de slope.  
 double static\_slope = slope(10000, 100, 1, 1024);  
  
 // Loop voor altijd.  
 while (1)  
 {  
 // Lees de waarde van de potential meter uit.  
 uint16\_t pwm = readADC();  
  
 // Zet de B pin aan.  
 PORTB = (1 << PB5);  
  
 // Bereken de delay.  
 double mapped = map(pwm, 1, 10000, static\_slope) / 2;  
  
 // Wacht voor een X aantal µs.  
 delay(mapped);  
  
 // Zet de B pin uit.  
 PORTB = 0;  
  
 // Wacht voor een X aantal µs.  
 delay(mapped);  
 }  
  
 // Geef een 0 terug.  
 return 0;  
}