Bilaga C. C-kod

```
#include <avr/io.h>
#include <stdio.h>

#define F_CPU 8000000UL

#include <avr/delay.h>

#define BIT0 0b00000001;
#define BIT1 0b00000010;
#define BIT2 0b00000100;
#define BIT3 0b00001000;
#define BIT4 0b00010000;
#define BIT5 0b00100000;
#define BIT6 0b01000000;
#define BIT7 0b10000000;

#define E_High PORTC |= BIT1
#define E_Low PORTC &= ~BIT1
#define RS_High PORTC |= BIT7
#define RS_Low PORTC &= ~BIT7
#define RW_High PORTC |= BIT0
#define RW_Low PORTC &= ~BIT0


void _delay_s(int s)                    //Delay in seconds
{
        s *= 5;
        while(s > -1)
        {
                _delay_ms(200);
                s--;
        }
}

int string_length(char str[])           //Checks how long a string is
{
  int i = 0;
  while(str[i] != '\0')
  {
        i++;
  }
  return(i);
}

void enable()                           //Enables the display
{
        E_High;
        _delay_us(1);
        E_Low;
}

void write(int a)                       //Writes to display
{
        PORTD = a;
        RW_Low;
        RS_High;
        _delay_ms(3);
        E_High;
        _delay_us(3);
        E_Low;
        _delay_ms(3);
}
```

```
int get_val( char c)                                    //Gets the ascii value for a character
{
      int ascii_value;
      ascii_value = c;
      return(ascii_value);
}

void print_str_to_lcd(char str[])
{
   int i = 0;
   while(i < string_length(str))
   {
      write(get_val(str[i]));
            _delay_s(1);
     i++;
            if(i > 15)
            {
                  PORTD = 0x18;                         //Shift display to the left
                  RS_Low;
                  RW_Low;
                  enable();
            }
      }
      while(i > 15)
      {
            PORTD = 0x18;                               //Shift display to the left
            RS_Low;
            RW_Low;
            enable();
            _delay_s(1);
            i--;
      }
}

void print_arr_to_lcd(int number[],int length)
{
   int i = 0;
   while(i < length)
   {
      write((number[i]));
     i++;
      }
      while(i < 5)
      {
            write(0x20);
            i++;
      }
}



void wait_til_not_busy()                                //Checks the busy flag at the display
{

      DDRD = 0;                     //PORT A input
      RW_High;                      //Read
      int a = 0;
      while(a != 0x80)
      {
            a = PIND&BIT7;
            enable();

      }
      RW_Low;                                 //Write
      DDRD = 0xFF;  //PORT A output
}

void init_lcd()                     //Initalizes the display
{
      _delay_ms(20);
```

```
	DDRD = 0xFF;		//Port D output
	DDRC = 0x83;		//Port C, Pin 0,1 and 7, output

	PORTD = 0x30;

	RS_Low;
	RW_Low;				//Write
	enable();
	_delay_ms(5);		//Wait 4.1 ms or more
	enable();
	_delay_us(150);		//Wait 100 us or more
	enable();
	wait_til_not_busy();
	PORTD = 0x30;		//8 bit, 1 line disp, 7x5 dots
	enable();
	wait_til_not_busy();
	PORTD = 0x08;		//Disp. off
	enable();
	wait_til_not_busy();
	PORTD = 0x01;		//Clear disp. and return cursor to home position
	enable();
	wait_til_not_busy();
	PORTD = 0x06;		//Set the shift mode
	enable();

}

void display_on()
{
	wait_til_not_busy();

	RS_Low;
	RW_Low;			//Write

	PORTD = 0x0C;		// Dsp. on, curs off, blink off

	enable();

	wait_til_not_busy();
}

void hex_to_ascii(long int hex)
{
	int b[5];
	int i = 0;
	while (i < 5)
	{
		b[i] = 0;
		i++;
	}

	i = 0;

	while (hex != 0)
	{
		b[i] = hex%10;			//The rest from divided by 10
		hex = hex/10;
		i++;
	}

	int length = i;
	int number[i];
	int j = 0;
	while(j < i)
	{
		number[j] = 0;
		j++;
	}
	i = 0;
	while (i < length)
```

```
	{
		number[i] = 0x30|b[i];
		i++;
	}

	print_arr_to_lcd(number,length);
}


int measure(void)
{
	long int time = 0;
	int s;
	for(s = 0; s < 10; s++)			//Makes 10 measures
	{
		TCCR1B = 0x02;	//timer prescaler 8
		DDRA = 0;			//input
		int a = 0;

		int i=0;
		while(i<200)			//Wait until there's been a zero long enough
		{
			a = PINA&BIT0;
			if (a == 0)
			{
				i++;
			}
			else
			{
				i=0;
			}

		}
		while(a != 0x01)						//Wait for light beam
		{
			a = PINA&BIT0;
		}

		TCNT1 = 0;

		a=PINA&BIT1;
		while(a == 0x02)						//Wait for ultrasonic
		{
			a = PINA&BIT1;
		}

		TCCR1B = 0;
		time += TCNT1;					//Time difference in microseconds
	}
	time = time/s;					//The average time difference after 10 measures


	//**** Calibrates the measuring device to show the right value *****
	long int mm = ((time*343*0.9232/1000))+37-5.16;

	if( mm < 37)
	{
		mm = 50;
	}else if( mm >= 37 && mm < 50)
	{
		mm = 60;
	}else if( mm >=50 && mm < 66)
	{
		mm = 70;
	}else if( mm >= 66 && mm < 77)
	{
		mm = 80;
	}else if( mm >= 77 && mm < 121)
	{
		mm += 5;
```

```
	}else if(mm >= 135 && mm < 199)
	{
		mm -= 5;
	}


	if(mm >= 199 && mm < 1101)
	{
		mm = 0.00004098337154*mm*mm+0.94367692402316*mm+6.91459963312528;
		if(mm > 184 && mm < 328)
		{
			mm -= 4;
		}else if(mm > 389 && mm < 448)
		{
			mm += 5;
		}else if(mm >= 448 && mm < 560)
		{
			mm += 8;
		}else if(mm >= 645 && mm < 671)
		{
			mm -= 5;
		}else if(mm >=671 && mm < 850)
		{
			mm -= 15;
		}else if(mm >= 990 && mm <= 1100)
		{
			mm += 5;
		}

	}
	mm += 5;					//Round off

   return(mm/10);			//Return the value in centimeters
}

int main()
{
	init_lcd();
	display_on();


	/****Start message****/
	print_str_to_lcd("Welcome to Fredrik and Einar's measure device!!!");

	init_lcd();
	display_on();

	PORTD = 0x8F;			//Sets the print address
	enable();

	PORTD = 0x04;			//Sets the display to print backwards
	enable();

	print_str_to_lcd("mc"); //Prints "cm" at the end of the display

	while(1)
	{
		RS_Low;
		RW_Low;
		PORTD = 0x8C;			//Sets the print address
		enable();
		hex_to_ascii(measure());		//Lets start!!!

	}

	return 0;
}
```