

```
/*  
 * GccApplication1.c  
 *  
 * Created: 2019-05-14 13:08:04  
 * Author : mte15dis  
 */
```

```
#define F_CPU 8000000UL  
#include <stdio.h>  
#include <stdlib.h>  
#include <avr/interrupt.h>  
#include <time.h>  
#include <avr/io.h>  
#include <util/delay.h>  
#include <stdbool.h>  
#define BUTTON1 0  
#define BUTTON2 1  
#define BUTTON3 2  
#define BUTTON4 3  
#define LED1 4  
#define rs 7  
#define rw 6  
#define en 1  
#define csDA 4  
#define wrDA 5
```

```
void init_buttons(void);  
void init_dAConverter(void);  
void dAConvertercmd(unsigned char);  
void init_lcd(void);  
void init_led(void);  
void lcdcmd(unsigned char);  
void lcddisplay(unsigned char);  
uint8_t buttons;  
uint8_t counter = 64;
```

```
uint8_t sine_wave[256] = {  
    0x80, 0x83, 0x86, 0x89, 0x8C, 0x90, 0x93, 0x96,  
    0x99, 0x9C, 0x9F, 0xA2, 0xA5, 0xA8, 0xAB, 0xAE,  
    0xB1, 0xB3, 0xB6, 0xB9, 0xBC, 0xBF, 0xC1, 0xC4,  
    0xC7, 0xC9, 0xCC, 0xCE, 0xD1, 0xD3, 0xD5, 0xD8,  
    0xDA, 0xDC, 0xDE, 0xE0, 0xE2, 0xE4, 0xE6, 0xE8,  
    0xEA, 0xEB, 0xED, 0xEF, 0xF0, 0xF1, 0xF3, 0xF4,
```

```

0xF5, 0xF6, 0xF8, 0xF9, 0xFA, 0xFA, 0xFB, 0xFC,
0xFD, 0xFD, 0xFE, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFE, 0xFE, 0xFD,
0xFD, 0xFC, 0xFB, 0xFA, 0xFA, 0xF9, 0xF8, 0xF6,
0xF5, 0xF4, 0xF3, 0xF1, 0xF0, 0xEF, 0xED, 0xEB,
0xEA, 0xE8, 0xE6, 0xE4, 0xE2, 0xE0, 0xDE, 0xDC,
0xDA, 0xD8, 0xD5, 0xD3, 0xD1, 0xCE, 0xCC, 0xC9,
0xC7, 0xC4, 0xC1, 0xBF, 0xBC, 0xB9, 0xB6, 0xB3,
0xB1, 0xAE, 0xAB, 0xA8, 0xA5, 0xA2, 0x9F, 0x9C,
0x99, 0x96, 0x93, 0x90, 0x8C, 0x89, 0x86, 0x83,
0x80, 0x7D, 0x7A, 0x77, 0x74, 0x70, 0x6D, 0x6A,
0x67, 0x64, 0x61, 0x5E, 0x5B, 0x58, 0x55, 0x52,
0x4F, 0x4D, 0x4A, 0x47, 0x44, 0x41, 0x3F, 0x3C,
0x39, 0x37, 0x34, 0x32, 0x2F, 0x2D, 0x2B, 0x28,
0x26, 0x24, 0x22, 0x20, 0x1E, 0x1C, 0x1A, 0x18,
0x16, 0x15, 0x13, 0x11, 0x10, 0x0F, 0x0D, 0x0C,
0x0B, 0x0A, 0x08, 0x07, 0x06, 0x06, 0x05, 0x04,
0x03, 0x03, 0x02, 0x02, 0x02, 0x01, 0x01, 0x01,
0x01, 0x01, 0x01, 0x01, 0x02, 0x02, 0x02, 0x03,
0x03, 0x04, 0x05, 0x06, 0x06, 0x07, 0x08, 0x0A,
0x0B, 0x0C, 0x0D, 0x0F, 0x10, 0x11, 0x13, 0x15,
0x16, 0x18, 0x1A, 0x1C, 0x1E, 0x20, 0x22, 0x24,
0x26, 0x28, 0x2B, 0x2D, 0x2F, 0x32, 0x34, 0x37,
0x39, 0x3C, 0x3F, 0x41, 0x44, 0x47, 0x4A, 0x4D,
0x4F, 0x52, 0x55, 0x58, 0x5B, 0x5E, 0x61, 0x64,
0x67, 0x6A, 0x6D, 0x70, 0x74, 0x77, 0x7A, 0x7D
};

```

```

int main(void)
{
    init_buttons();
    init_led();
    init_lcd();
    init_dAConverter();

    bool flag1 = false;
    bool flag2 = false;
    bool flag3 = false;
    bool flag4 = false;
    lcdcmd(0x01); //clear display
    lcdcmd(0x80); //move cursor to beginning of first line
    lcdcmd(0x3C); //enable 5x7 mode for chars
    lcdcmd(0x0E); //display ON, cursor ON

    while (1)
    {

```

```

uint8_t x = 0;
dAConvertercmd(128);
buttons = PINA;

if(buttons & ((1<<BUTTON1))){

    if(!flag1){

        lcdcmd(0xc1); //move cursor to beginning of second line
        lcddisplay('C');
        flag1 = true;

        while(counter > 0){
            for(x =1; x<255; x++)
            {
                dAConvertercmd(sine_wave[x]);
                _delay_us(12.1); //c1
            }
            counter--;
        }
        counter = 64;

        lcdcmd(0x01); //clear display

    }
    } else {
        flag1 = false;
    }
}

if((buttons & (1<<BUTTON2))){

    if(!flag2){
        lcdcmd(0xc7); //move cursor to beginning of second line
        lcddisplay('D');

        flag2 = true;

        while(counter > 0){
            for(x =1; x<255; x++)
            {
                dAConvertercmd(sine_wave[x]);
                _delay_us(10.3); //d1
            }
            counter--;
        }
    }
}

```

```

        }
        counter = 64;
        lcdcmd(0x01); //clear display

    }
    } else {
        flag2 = false;
}

if((buttons & (1<<BUTTON3))){
    if(!flag3){
        lcdcmd(0xcC); //move cursor to beginning of fourth line
        lcddisplay('E');

        flag3 = true;

        while(counter > 0){
            for(x =1; x<255; x++)
            {
                dAConvertercmd(sine_wave[x]);
                _delay_us(9);//e1
            }
            counter--;
        }
        counter = 64;
        lcdcmd(0x01); //clear display
    }
    } else {
        flag3 = false;
}

if((buttons & (1<<BUTTON4))){
    if(!flag4){
        //lcdcmd(0xD4); //move cursor to beginning of fourth line
        lcdcmd(0xd2);
        lcddisplay('F');

        flag4 = true;

        while(counter > 0){
            for(x =1; x<255; x++)
            {
                dAConvertercmd(sine_wave[x]);
                _delay_us(8.2);//f1
            }
            counter--;

```

```

        }
        counter = 64;
        lcdcmd(0x01); //clear display
    }
    } else {
        flag4 = false;
    }
}

}

void init_buttons(){
    DDRA &= ~((1<<BUTTON1)|(1<<BUTTON2)|(1<<BUTTON3)|(1<<BUTTON4)); /*
Makes button pins of PORTA as Input, ~ inverterar biten.*/
}
void init_led() {
    DDRA |= (1<<LED1); /* Gör LEDpinen till Output*/
}
void init_lcd() {
    DDRB |= ((1<<7)|(1<<6)|(1<<5)|(1<<4)|(1<<3)|(1<<2)|(1<<1)|(1<<0)); /* Sätter LCD
pinsen till Output */
    DDRC |= ((1<<rs)|(1<<rw)|(1<<en));
}
}
void lcdcmd(unsigned char val) {
    PORTB = val;
    PORTC &= ~(1<<rs);
    PORTC &= ~(1<<rw);
    PORTC |= (1<<en);
    _delay_us(10);
    PORTC &= ~(1<<en);
    _delay_us(1000);
}
void lcddisplay(unsigned char val) {
    PORTB = val;
    PORTC |= (1<<rs);
    PORTC &= ~(1<<rw);
    PORTC |= (1<<en);
    _delay_us(10);
    PORTC &= ~(1<<en);
    _delay_us(1000);
}
}
void init_dAConverter() {

```

```
        DDRD |= ((1<<7)|(1<<6)|(1<<5)|(1<<4)|(1<<3)|(1<<2)|(1<<1)|(1<<0)); /* Sätter DA
pinsen till Output */
        DDRA |= ((1<<csDA)|(1<<wrDA));
    }
    void dAConvertercmd(unsigned char val) {
        PORTA &= ~(1<<csDA);
        PORTA &= ~(1<<wrDA);
        PORTD = val;
        PORTA |= (1<<wrDA);
        PORTA |= (1<<csDA);
    }
}
```