

```

#define F_CPU 1000000UL // 1 MHz
#include <util/delay.h>
#include <avr/io.h>

/*Function declaration*/
void InitADC();
uint16_t ReadADC(uint8_t);
void lcd_init();
void RSHigh();
void RSLow();
void RWHigh();
void RWLow();
void EHigh();
void ELow();
void lcd_reset();
void write_command(char val);
void write_data(char val);
void LCD_print (char *str);
void VentHigh();
void VentLow();
void knappNer();
void knappUpp();
void print_status();
void vattning();

/*Variable declaration*/
char btn1;
char btn2;
uint16_t fukt;
int lcd_delay = 10;
int global_delay = 0 ;
int fuktapp = 0;

int main(void)
{
    InitADC();

    DDRA = 0b00000000;
    DDRB = 0b11111111;
    DDRD = 0b00100111;

    VentLow();

    lcd_reset();
    lcd_init();
    LCD_print("My name is");
    new_line();
    LCD_print("Feeding Daisy");
    // _delay_ms(2000);

    while(1)
    {
        btn1 = (0b00001000 & PIND);
        btn2 = (0b00010000 & PIND);

        fukt=ReadADC(1);
    }
}

```

```

        global_delay++;
        if(btn1 == 0b00001000){ // pin 3
            knappUpp();
        }
        if (btn2 != 0){
            knappNer();
        }

        if(global_delay > 15000){
            if(fukt<fuktapp) {
                vattning();
            }
            global_delay = 0;
            print_status();
        }
        _delay_ms(1);
    }
}

void InitADC(){
    ADMUX=(1<<REFS0);
    ADCSRA=(1<<ADEN) | (1<<ADPS2) | (1<<ADPS1);
}

uint16_t ReadADC(uint8_t ch){
    ch &= 0b0000111;
    ADMUX |= ch;

    //Start single conversion
    ADCSRA |= (1<<ADSC);

    //Wait for conversion to complete
    while(!(ADCSRA & (1<<ADIF)));

    ADCSRA |= (1<<ADIF);
    return(ADC);
}

void knappUpp(){
    if (fuktapp == 600) {
        return;
    }
    fuktapp = fuktapp + 100;
    print_status();
}

void print_status(){
    lcd_clear();
    char str[15];
    sprintf(str, "%d", (int) fuktapp);
    char stv[15];
    sprintf(stv, "%d", (int) fukt);
    LCD_print("Nuv fukt: ");
    LCD_print(stv);
    new_line();
    LCD_print("Önskad fukt: ");
}

```

```

        LCD_print(str);
        _delay_ms(200);
    }

void vattning(){
    VentHigh();
    _delay_ms(10000);
    VentLow();
}

void knappNer(){
    if (fuktapp == 0){
        return;
    }
    fuktapp = fuktapp - 100;
    print_status();
}

void lcd_init(){
    write_command(0b00111000); //Function set
    write_command(0b00001110); //Display on
    write_command(0b00000110); //Entry mode set
    lcd_clear();
}

void write_command(char val)
{
    PORTB = val; // DATA in
    _delay_ms(lcd_delay);
    RSLow();
    _delay_ms(lcd_delay);
    RWLow();
    _delay_ms(lcd_delay);
    ELow();
    lcd_reset();
}

void write_data(char val){
    if (val == '0') {
        PORTB = 0xEF;
    } else {
        PORTB = val;
    }
    _delay_ms(lcd_delay);
    RSHigh();
    _delay_ms(lcd_delay);
    RWLow();
    _delay_ms(lcd_delay);
    ELow();
    lcd_reset();
}

void lcd_reset(){
    _delay_ms(lcd_delay);
    EHigh();
    _delay_ms(lcd_delay);
    RWHigh();
    _delay_ms(lcd_delay);
}

```

```

        RSLow();
        _delay_ms(lcd_delay);
    }

void lcd_clear(){
    write_command(0b00000001);
}

void VentHigh(){
    PORTD |= (1<<5);
}

void VentLow(){
    PORTD &= ~(1<<5);
}

void ELow()
{
    // Clear E to 0
    PORTD &= ~(1<<0);
}
void EHigh()
{
    // Set E to 1
    PORTD |= (1<<0);
}
void RSLow()
{
    // RS (D-Port 2) to 0
    PORTD &= ~(1<<2);
}
void RSHigh()
{
    // RS to 1
    PORTD |= (1<<2);
}
void RWLow()
{
    // RW to 0
    PORTD &= ~(1<<1);
}
void RWHigh()
{
    // RW to 1
    PORTD |= (1<<1);
}

void LCD_print (char *str)
{
    int i;
    for(i=0; str[i]!='\0'; i++)
    {
        if (i == 16)
        {
            new_line();
        }
        if (i > 31)
        {

```

```
        return;
    }
    write_data(str[i]);
}
return;
}
```

```
void new_line(){
    PORTB = 0xC0;
    _delay_ms(lcd_delay);
    RWHigh();
    _delay_ms(lcd_delay);
    EHigh();
    _delay_ms(lcd_delay);
    RWLow();
    _delay_ms(lcd_delay);
    ELow();
    _delay_ms(lcd_delay);
    EHigh();
    lcd_reset();
}
```