

```
#include "main.h"
```

```
struct eeeprom_giess *gpEEPROM = (struct eeeprom_giess*)0;
#define EEPROM_ADDR_TICKCOUNT (uint16_t*)(sizeof(struct eeeprom_giess)*6)

struct time_st zuletzt_gegossen[2][6];

#define AUTOMATIK_OFF 0xFF
#define AUTOMATIK_TIME 'T'
#define AUTOMATIK_MEAS 'M'

#define PUMP1_PORT PORTD
#define PUMP1_PIN 0x08
#define PUMP2_PORT PORTD
#define PUMP2_PIN 0x10
#define PUMP3_PORT PORTD
#define PUMP3_PIN 0x04
#define PUMP4_PORT PORTD
#define PUMP4_PIN 0x40
#define PUMP5_PORT PORTD
#define PUMP5_PIN 0x80
#define PUMP6_PORT PORTB
#define PUMP6_PIN 0x01

void get_flower_name(unsigned char index, char* name);
void print_time(struct time_st *pTime);
void print_time2();

void switch_pump(unsigned char index, char on_off)
{
    // char name[16];
    if(index == 0)
    {
        if(on_off) PUMP1_PORT |= PUMP1_PIN;
        else PUMP1_PORT &= ~PUMP1_PIN;
    }
    if(index == 1)
    {
        if(on_off) PUMP2_PORT |= PUMP2_PIN;
        else PUMP2_PORT &= ~PUMP2_PIN;
    }
    if(index == 2)
    {
        if(on_off) PUMP3_PORT |= PUMP3_PIN;
        else PUMP3_PORT &= ~PUMP3_PIN;
    }
    if(index == 3)
    {
        if(on_off) PUMP4_PORT |= PUMP4_PIN;
        else PUMP4_PORT &= ~PUMP4_PIN;
    }
    if(index == 4)
    {
        if(on_off) PUMP5_PORT |= PUMP5_PIN;
        else PUMP5_PORT &= ~PUMP5_PIN;
    }
    if(index == 5)
    {
        if(on_off) PUMP6_PORT |= PUMP6_PIN;
        else PUMP6_PORT &= ~PUMP6_PIN;
    }
}

// get_flower_name(index, name);
if(on_off)
{
    // printf("* giesse %s !\r\n", name);
    memcpy(&zuletzt_gegossen[1][index], &zuletzt_gegossen[0][index], sizeof(struct time_st));
    memcpy(&zuletzt_gegossen[0][index], &zeit, sizeof(struct time_st));
}
else
// printf("* %s fertig gegossen. *\r\n", name);
}

char pump_status(unsigned char index)
```

```

{
    if(index == 0)
    {
        if(PUMP1_PORT & PUMP1_PIN) return 1;
        else return 0;
    }
    if(index == 1)
    {
        if(PUMP2_PORT & PUMP2_PIN) return 1;
        else return 0;
    }
    if(index == 2)
    {
        if(PUMP3_PORT & PUMP3_PIN) return 1;
        else return 0;
    }
    if(index == 3)
    {
        if(PUMP4_PORT & PUMP4_PIN) return 1;
        else return 0;
    }
    if(index == 4)
    {
        if(PUMP5_PORT & PUMP5_PIN) return 1;
        else return 0;
    }
    if(index == 5)
    {
        if(PUMP6_PORT & PUMP6_PIN) return 1;
        else return 0;
    }
    return -1;
}

void get_flower_name(unsigned char index, char* name)
{
    eeprom_read_block(name, gpEEPROM[index].name, 16);
    name[15] = 0;
    if(name[0] < 0x20)
    {
        strcpy(name, "Blume");
        name[5] = index+'1';
        name[6] = 0;
    }
}

void print_menu()
{
    unsigned char i;
    char name[16];
    uart_puts_P(PSTR("\r\n\nGiess-o-mat Hauptmenue\r\n"));
    for(i=1; i<=6; i++)
    {
        get_flower_name(i-1, name);
        printf(" %d - %s\r\n", i, name);
    }
    uart_puts_P(PSTR(" s - Sensorwerte anzeigen\r\n"));
    uart_puts_P(PSTR(" c - Sensorwerte -> CSV\r\n"));
    uart_puts_P(PSTR(" k - aktuelle Konfiguration anzeigen\r\n"));
    uart_puts_P(PSTR(" u - Uhr stellen\r\n"));
}

void print_adcs()
{
    char i;
    unsigned char sec = 0;

    while(!uart_is_rx())
    {
        if(zeit.sec != sec)
        {
            autogiess();
            uart_puts_P(PSTR("\r\n"));
            print_time(&zeit);
            for(i=0; i<6; i++)

```

```

    {
        uint16_t freq = get_freq(i);
        printf("%5u0", freq);
    }
    sec = zeit.sec;
}
}
uart_getchar();
}

```

```

void print_csv()

```

```

{
    int i;
    unsigned char sec = 0;
    uint32_t wsum[6];
    char pump_on[6];
    int mincount = 1;
    int print_header = 1;

    uart_puts_P(PSTR("\r\nCSV output. Press ESC to end\r\n"));

    for(i=0; i<6; i++)
    {
        wsum[i] = 0;
        pump_on[i] = 0;
    }

    while(1)
    {
        if(zeit.sec != sec)
        {
            autogiess();
            sec = zeit.sec;
            for(i=0; i<6; i++)
            {
                uint16_t freq = get_freq(i);
                wsum[i] += (uint32_t)freq;

                if(pump_status(i) != 0)
                    pump_on[i] = 1;
            }

            if(sec == 0)
            {
                if(print_header)
                {
                    int i;
                    char name[16];
                    uart_puts_P(PSTR(";Zeit"));
                    for(i=0; i<6; i++)
                    {
                        get_flower_name(i, name);
                        printf(";%s", name);
                    }
                    uart_puts_P(PSTR("\r\n#P;"));
                    print_time2();
                    for(i=0; i<6; i++)
                    {
                        uint16_t schwelle = 0;
                        eeprom_read_block(&schwelle, &gpEEPROM[i].pump_on, sizeof(schwelle));
                        printf("%u", schwelle);
                    }
                    print_header = 0;
                }

                if(mincount == 10)
                {
                    int pumped = 0;
                    uart_puts_P(PSTR("\r\n#S;"));
                    print_time2();

                    for(i=0; i<6; i++)
                    {
                        uint32_t mw = wsum[i] / 60;
                        mw = mw / mincount;
                    }
                }
            }
        }
    }
}

```