

Raspberry Pi och Tellstick, ett program i C.

Ett program skrivet i C för att med Tellstick tända och släcka en eller flera lampor.

Programmet startas med kommandot:

```
/home/pi/Development/Cprog/Tellstick/tellstick-13 -f </dev/null &>/dev/null &
```

Programmet kommer att skifta loggfil kl 00:01, så kallad loggroll.

För att avbryta programmet måste man använda shellkommandot:

```
kill -9 PID
```

där PID, process ID, är det id som sätts på programmet. PID kan hittas med hjälp av shellkommandot

```
ps -ef
```

Programmet tellstick-13.

```
/* tellstick-13.c
 * 2014-11-07 Jan Pihlgren
 * 2014-11-16
 *
 * Loggroll körs varje natt klockan 00:01.
 *
 * Programmet utnyttjar en indatafil där tänd och släcktider
anges.
 * PATH/tellstick-13 -f
 *
 * Indatafilens organisation:
 * löpnr tändtid släcktid brytarnr/enhetsnr
 * ex:
 * 1 15:00 17:00 2 ordningsnr, tändtid, släcktid, brytarenr
 *
 * Programmet stoppas med kommando:
 * kill -9 PID eller sudo kill -9 PID
 * där PID hittas med kommando ps -ef och leta efter
startkommandot ovan.
 *
 */
#include <tellus-core.h>
#include <time.h>
#include <stdio.h>
```

```

#include <unistd.h>
#include <string.h>
#include <stdlib.h>

struct strombr {
    char nr[2];                // Ordningsnummer
    char ontid[10];
    char offtid[10];
    char switchnr[2];        // Brytarens/enhetens nr
};
struct strombr brytare[25];    // Plats för 25 brytare/enheter

char indatafile[] = "/home/pi/Script/tidschema_ny.data";
char logfile[] = "/home/pi/temp/tellstick.log";
int antpost =0;
int postnr;

void loggroll();
void readdata_ny();

int main(int argc, char *argv[]) {

    time_t tid;
    struct tm *info;
    char buffer[80];
    char sec[80];
    char klockan[10];
    int i;
    int go = 1; // fortsatt programmet
    char version[] = "Version: tellstick-13";
    char rolltid[] = "00:01";

    strcpy(brytare[0].ontid,"18:00");
    strcpy(brytare[0].offtid,"06:00");

    loggroll();                /* Rulla loggfilerna */
    readdata_ny();            /* läs in data för täntider och släcktider i arra[]

```

```

*/

FILE *fp;
if (argc > 1) {
    if (strcmp(argv[1], "-f") == 0) {                /* Läs från datafil om tänd- och
släcktider ? */
        /* Hämtas data från fil */
        postnr = 0;                                /* Första
tändtiden, post nr 1 */
//        strcpy(starttid,brytare[postnr].ontid);
//        strcpy(stopptid,brytare[postnr].offtid);
    }
}
// else{
//     // strcmp ?
//}

fp=fopen(logfile, "a");
fprintf(fp,"%s\n",version);
fprintf(fp,"brytare %s\n ",brytare[postnr].switchnr);
fprintf(fp,"Tändning sker klockan %s\n",brytare[postnr].ontid);
fprintf(fp,"Släckning sker klockan %s\n",brytare[postnr].offtid);
fclose(fp);

while(go != 0){
    time(&tid);
    info = localtime(&tid);
    strftime(buffer,80,"%H:%M",info);
    strcpy(klockan,buffer);
    strftime(sec,80,"%H:%M:%S",info);
    fp=fopen(logfile, "a");
    for(i = 0; i < antpost; i++){
        if(strcmp(klockan,brytare[i].ontid) == 0){
            tdTurnOn(atoi(brytare[i].switchnr));
            fprintf(fp,"Tänder %s Kl %s\n",brytare[i].switchnr,klockan);
        }
        if(strcmp(klockan,brytare[i].offtid) == 0){
            tdTurnOff(atoi(brytare[i].switchnr));

```

```

        fprintf(fp, "Släcker %s Kl %s\n", brytare[i].switchnr, klockan);
    }
}
fclose(fp);

if(strcmp(rolltid, klockan) == 0){
    loggroll(); // Rulla
    readdata_ny(); /* läs in
data igen */

    fp=fopen(logfile, "a");
    fprintf(fp, "%s\n", version);
    if(strcmp(argv[1], "-f") == 0){
        fprintf(fp, "Ant poster = %d\n", antpost);
        fprintf(fp, "Tändning kl %s. Släckning kl
%s\n", brytare[0].ontid, brytare[0].offtid);
    }
    fclose(fp);
}

sleep(60);
}
tdClose();

return 0;
}

void loggroll(){

    char oldfile1[] = "/home/pi/temp/tellstick.old1";
    char oldfile2[] = "/home/pi/temp/tellstick.old2";
    char oldfile3[] = "/home/pi/temp/tellstick.old3";

    /* Radera oldfile3 */
    remove(oldfile3);

    /* Flytta filer */

```

```

        rename(oldfile2, oldfile3); // flytta oldfile2 till oldfile3
        rename(oldfile1, oldfile2); // flytta oldfile1 till oldfile2
        rename(logfile, oldfile1); // flytta logfile till oldfile1
    return;
}

void readdata_ny(){
    FILE *data;

    data = fopen(indatafile,"r");
    antpost=0;
    while(1){
        fscanf(data,"%s",brytare[antpost].nr);
        fscanf(data,"%s",brytare[antpost].ontid);
        fscanf(data,"%s",brytare[antpost].offtid);
        fscanf(data,"%s",brytare[antpost].switchnr);
        antpost++;
        if(feof(data)){
            antpost--;
            break;
        }
    }
    fclose(data);
    return;
}

```

Datafilen tidschema_ny.data

```

1 04:30 07:50 1
2 05:00 06:00 2
3 16:10 19:15 1
4 16:15 17:05 2

```

Makefile

```

CC=gcc
CFLAGS=-Wall -ltelldus-core

OBJS = tellstick-13.o

```

```
all: ${OBJS}
    ${CC} -o tellstick-13 ${CFLAGS} ${OBJS}
```

```
tellstick-13.c:
    ${CC} ${CFLAGS} -c tellstick-13.c
```

```
clean:
    rm -f tellstick-13.o
```