

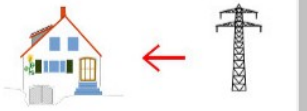
- house
- [e3dc Values](#)
 - [Webcam Keller H](#)
 - [Webcam Keller S](#)
 - [Webcam Garten](#)
 - [Akkustatus](#)
 - [Solar-0](#)
 - [Solar-1](#)
 - [Akku AZ](#)
 - [Carport](#)
 - [sdm630-energy](#)
 - [sdm230-energy](#)
 - [air Values](#)
 - [Twingo ZE](#)

sunrise : 08:07 Uhr
 sunset : 16:18 Uhr
 date : 2024-01-08
 sunhours : 4

- charge 0
- charge 1
- charge 2
- charge 3


08.01.2024; 11:00

Total system power: 79.85 Watts
 in 79.85 Watts




Zähler 440703-7282801
 1.8.1(Bezug) 0020019.709*kWh
 2.8.1(Einsp) 0016918.982*kWh

E3DC S10 Energy




solar: 320 W




Batterie +

battery: 0 W 100%


date: 2024/01/08
time: 10:57



SB1600: 94 W



house : 80 W



grid : -1007 W

Reload

notify events via mail/sms

send a sms from Computer

remote controll radio power sockets

- RF 1 Licht Flur oben
- RF 2 LötKolben
- RF 3 Ladegerät Fernseh
- RF 4 Stern & Schwibbnonen

OUT 7	Schuppen	<input type="checkbox"/>	0	IN 0	<input type="checkbox"/>
OUT 6	Carport	<input type="checkbox"/>	0	IN 1	<input type="checkbox"/>
OUT 5	Adventsstern	<input type="checkbox"/>	0	IN 2	<input type="checkbox"/>
OUT 4	Aussen S	<input type="checkbox"/>	0	IN 3	<input type="checkbox"/>
OUT 3	Kellerlicht	<input type="checkbox"/>	0	IN 4	<input type="checkbox"/>
OUT 2	-----	<input type="checkbox"/>	0	IN 5	<input type="checkbox"/>
OUT 1	Heizung Jac	<input type="checkbox"/>	0	IN 6	<input type="checkbox"/>
OUT 0	Heizung WZ	<input type="checkbox"/>	0	IN 7	<input type="checkbox"/>

OUT 0.0	Keller Zisterne N	<input type="checkbox"/>	0	IN 0	<input type="checkbox"/>
OUT 0.1	---	<input type="checkbox"/>	0	IN 1	<input type="checkbox"/>
OUT 1.0	Pumpe Gewächshaus	<input type="checkbox"/>	0	IN 2	<input type="checkbox"/>
OUT 1.1	---	<input type="checkbox"/>	0	IN 3	<input type="checkbox"/>
OUT 2.0	Garage Zisterne G	<input type="checkbox"/>	0	IN 4	<input checked="" type="checkbox"/>
OUT 2.1	Garage Zisterne V	<input type="checkbox"/>	0	IN 5	<input checked="" type="checkbox"/>
OUT 3.0	Waschk. Kühlbox	<input type="checkbox"/>	0	IN 6	<input checked="" type="checkbox"/>
OUT 3.1	Waschküche	<input type="checkbox"/>	0	IN 7	<input checked="" type="checkbox"/>
OUT 4.0	Schlafzimmer Heizung	<input type="checkbox"/>	0		
OUT 4.1	Schlafzimmer Licht	<input type="checkbox"/>	0		
OUT 5.0	GaragenTor	<input type="checkbox"/>	0		
OUT 5.1	---	<input type="checkbox"/>	0		
OUT 6.0	---	<input type="checkbox"/>	0		

0:000.0	911 :08.01. 10:59	Veranda	Pin0: Heizung	27.5
1:005.0	611 :08.01. 11:00	Luftrohr Keller	Pin1: Keller (Holz)	8.0
2:-07.0	011 :08.01. 10:59	Gewaechshaus	Pin1: Aussen	-8.0
3:009.5	311 :08.01. 10:59	Kellergang	Pin2: Arbeitszimmer	22.5
4:-09.5	311 :08.01. 10:59	Schuppen	Pin2: Boden	0.5
			Pin3: Esszimmer	19.0
			Pin3: Server	16.0

Time :08.01. 11:00
 Bus:1 Dev:44
 Dist[0]: 019,8: 05.12. 13:09
 Dist[1]: 019,8: 28.12. 09:36
 Dist[2]: 019,0: 06.01. 12:08

- house
- [e3dc Values](#)
 - [Webcam Keller H](#)
 - [Webcam Keller S](#)
 - [Webcam Garten](#)
 - [Akkustatus](#)
 - [Solar-0](#)
 - [Solar-1](#)
 - [Akku AZ](#)
 - [Carport](#)
 - [sdm630-energy](#)
 - [air Values](#)

sunrise : 07:55 Uhr
 sunset : 16:41 Uhr
 date : 2023-01-23
 sunhours : 0

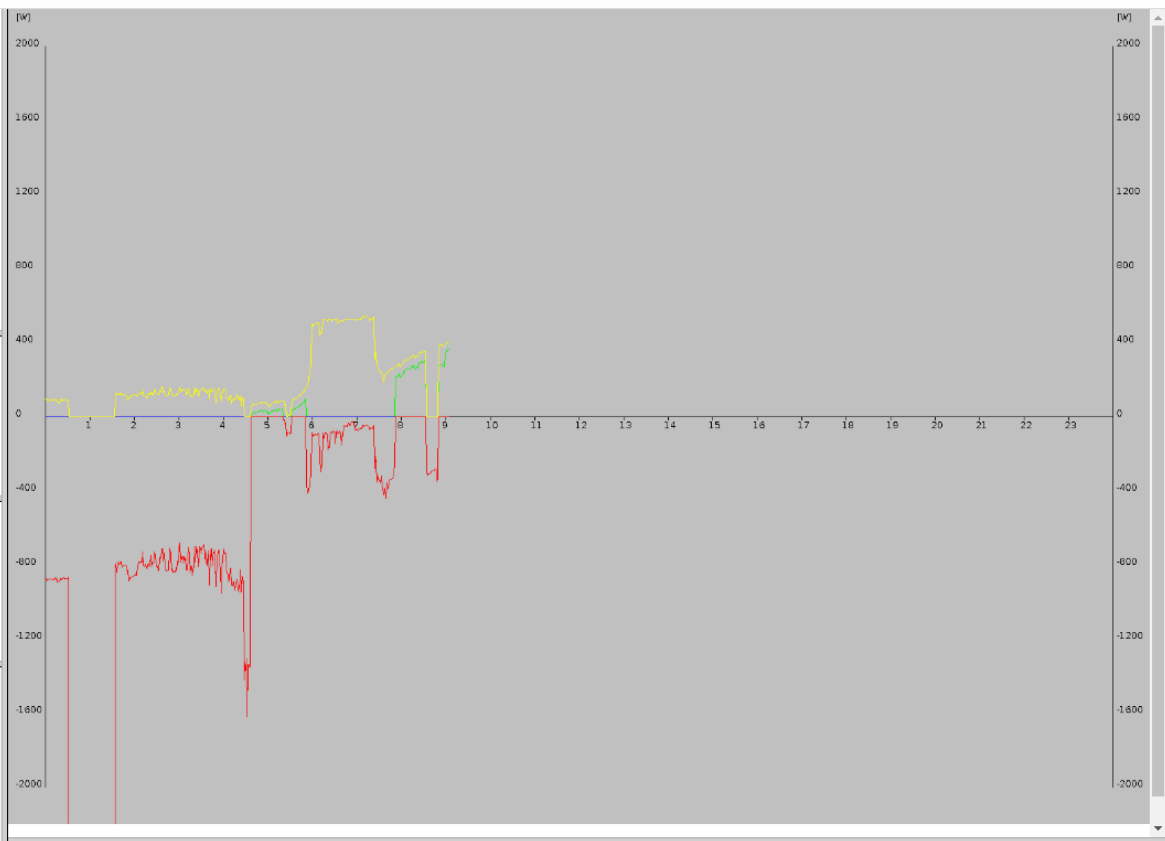
- charge 0
- charge 1
- charge 2
- charge 3

23.01.2023; 09:09

Total system power: -368.64 Watts
 out -368.64 Watts

→

Zähler 440703-7282801
 1.8.1(Bezug) 0019961.708*kWh
 2.8.1(Einsp) 0013454.822*kWh



notify events via mail/sms

send a sms from Computer

- remote controll radio power sockets
- RF 1 Licht Flur oben
 - RF 2 LötKolben
 - RF 3 Ladegerät Fernseh
 - RF 4 Stern & Schallbrunn

OUT 7	Schuppen	<input type="checkbox"/>	<input type="text" value="0"/>	IN 0	<input type="checkbox"/>
OUT 6	Carport	<input type="checkbox"/>	<input type="text" value="0"/>	IN 1	<input type="checkbox"/>
OUT 5	Adventsstern	<input type="checkbox"/>	<input type="text" value="0"/>	IN 2	<input type="checkbox"/>
OUT 4	Aussen S	<input type="checkbox"/>	<input type="text" value="0"/>	IN 3	<input type="checkbox"/>
OUT 3	Kellerlicht	<input type="checkbox"/>	<input type="text" value="0"/>	IN 4	<input type="checkbox"/>
OUT 2	-----	<input checked="" type="checkbox"/>	<input type="text" value="0"/>	IN 5	<input type="checkbox"/>
OUT 1	Heizung Jac	<input type="checkbox"/>	<input type="text" value="0"/>	IN 6	<input type="checkbox"/>
OUT 0	Heizung WZ	<input type="checkbox"/>	<input type="text" value="0"/>	IN 7	<input type="checkbox"/>

OUT 0.0	Keller Zisterne N	<input type="checkbox"/>	<input type="text" value="0"/>	IN 0	<input type="checkbox"/>
OUT 0.1	---	<input type="checkbox"/>	<input type="text" value="0"/>	IN 1	<input type="checkbox"/>
OUT 1.0	Pumpe Gewächshaus	<input type="checkbox"/>	<input type="text" value="0"/>	IN 2	<input type="checkbox"/>
OUT 1.1	---	<input type="checkbox"/>	<input type="text" value="0"/>	IN 3	<input type="checkbox"/>
OUT 2.0	Garage Zisterne G	<input type="checkbox"/>	<input type="text" value="0"/>	IN 4	<input checked="" type="checkbox"/>
OUT 2.1	Garage Zisterne V	<input type="checkbox"/>	<input type="text" value="0"/>	IN 5	<input checked="" type="checkbox"/>
OUT 3.0	Waschk. Kühlbox	<input type="checkbox"/>	<input type="text" value="0"/>	IN 6	<input checked="" type="checkbox"/>
OUT 3.1	Waschküche	<input type="checkbox"/>	<input type="text" value="0"/>	IN 7	<input checked="" type="checkbox"/>
OUT 4.0	Schlafzimmer Licht	<input type="checkbox"/>	<input type="text" value="0"/>		
OUT 4.1	Schlafzimmer Heizung	<input type="checkbox"/>	<input type="text" value="0"/>		
OUT 5.0	GaragenTor	<input type="checkbox"/>	<input type="text" value="0"/>		
OUT 5.1	---	<input type="checkbox"/>	<input type="text" value="0"/>		
OUT 6.0	---	<input type="checkbox"/>	<input type="text" value="0"/>		
OUT 6.1	Bad	<input type="checkbox"/>	<input type="text" value="0"/>		

Time :23.01. 09:10
 Bus:1 Dev:44
 0:003.5 911 :23.01. 09:09
 1:006.0 311 :23.01. 09:08
 2:000.5 711 :23.01. 09:08
 3:009.5 211 :23.01. 09:09
 4:-01.0 311 :23.01. 09:09
 5:013.5 711 :15.11. 13:15
 6:010.0 811 :23.01. 09:09
 7:-01.0 311 :23.01. 09:09
 8:033.5 001 :23.01. 09:09

Time :23.01. 09:09
 Bus:1 Dev:44
 Dist[0]: 020,4: 16.12. 11:27
 Dist[1]: 072,7: 20.01. 10:54
 Dist[2]: 019,7: 19.01. 19:27

Pin0:	Heizung	26.5
Pin1:	Keller (Holz)	8.0
Pin1:	Aussen	0.0
Pin2:	Arbeitszimmer	23.0
Pin2:	Boden	4.5
Pin3:	Esszimmer	21.0
Pin3:	Server	15.0

CGI mit C

Common Gateway Interface

- CGI Scripte dienen der dynamischen Erzeugung von Webseiten, der html-Code wird von einem Programm dynamisch erzeugt. Der Aufruf dieses Programms erfolgt durch den http-Server, wenn er einen entsprechenden Link bekommt.
- CGI Scripte werden serverseitig ausgeführt, im Gegensatz zu Applets oder java script
- CGI Scripte können mit den verschiedensten Werkzeugen hergestellt werden, häufige Werkzeuge sind Perl, C und Shellscriptsprachen

Apache2 konfigurieren

kann bei unterschiedlichen Versionen variieren

- https://www.server-world.info/en/note?os=Ubuntu_20.04&p=httpd

- **Konfigurierung in apache2**

```
a2enmod cgid
```

```
a2enmod userdir
```

```
systemctl restart apache2
```

- **In /etc/apache2/mods-enabled/userdir.conf ergänzen:**

```
<Directory /home/*/public_html/cgi-bin>  
    Options ExecCGI  
    SetHandler cgi-script  
</Directory>
```

Verzeichnisse

- Dokumentenwurzel: `/var/www` oder `/srv/www`
- CGI-Verzeichnis: `/usr/lib/cgi-bin` oder unter der Dokumentenwurzel
- Dokumente im User-Dir: `public_html`
- CGI im User-Dir: `public_html/cgi-bin`

- Rechte setzen

Ein erstes CGI-Programm

- Ausgabe einer ASCII-Code-Tabelle
- C-Code auf der nächsten Seite
- Arbeitsschritte
 - C-Code compilieren
 - `gcc -o ascii.cgi ascii.c`
 - Binary ablegen unter
 - `$HOME/public_html/cgi-bin`
 - `/usr/lib/cgi-bin`
- Aufruf über den Browser
 - `http://www.informatik.htw-dresden.de/~sxxxx/cgi-bin/ascii.cgi`

```
#include <stdio.h>
#include <ctype.h>
```

```
int main()
```

```
{
  int i,j;
  printf("Content-Type: text/html");
  printf("\n\n") ;
  puts("<html><head><title>CGI-Script - ASCII Codetabelle</title></head>\n");
  puts("<body><h2><hr>\n");
  puts("<pre>\n");
```

```
for (i=0; i<4; i++)printf("|dec hex Char ");
printf("\n");
```

```
for (i=0; i<32; i ++)
```

```
{
  printf("\n| ");
  for (j=0; j < 128; j += 128/4)
  {
    printf("%3d %2X ",i+j,i+j);  if (isgraph(j+i))printf(" %c |",j+i); else printf(" . |");
  }
}
```

```
puts("</pre><hr>");
```

```
puts("<a href='\"http://www.informatik.htw-dresden.de/~beck/a.beck.html\"'>A. Beck</A> <p>");
```

```
puts("</body>\n");
```

```
puts("</html>\n");
```

```
return 0;
```

```
}
```

Arbeitsschritte:

- Quelltext erfassen
- Compilieren gcc ascii.c -o ascii.cgi
- nach cgi-bin kopieren
- Lese-/Ausführungsrechte für alle für ascii.cgi
- Aufruf über Browser

<http://www.htw-dresden.de/~s12355/cgi-bin/ascii.cgi>

html-Quelltext wird
Über die Standardausgabe
An den Browser ausgegeben

http://ipc145.informatik.htw-dresden.de/~beck/cgi-bin/ascii.cgi

dec	hex	Char	dec	hex	Char	dec	hex	Char	dec	hex	Char
0	0	.	32	20	.	64	40	@	96	60	`
1	1	.	33	21	!	65	41	A	97	61	a
2	2	.	34	22	"	66	42	B	98	62	b
3	3	.	35	23	#	67	43	C	99	63	c
4	4	.	36	24	\$	68	44	D	100	64	d
5	5	.	37	25	%	69	45	E	101	65	e
6	6	.	38	26	&	70	46	F	102	66	f
7	7	.	39	27	'	71	47	G	103	67	g
8	8	.	40	28	(72	48	H	104	68	h
9	9	.	41	29)	73	49	I	105	69	i
10	A	.	42	2A	*	74	4A	J	106	6A	j
11	B	.	43	2B	+	75	4B	K	107	6B	k
12	C	.	44	2C	,	76	4C	L	108	6C	l
13	D	.	45	2D	-	77	4D	M	109	6D	m
14	E	.	46	2E	.	78	4E	N	110	6E	n
15	F	.	47	2F	/	79	4F	O	111	6F	o
16	10	.	48	30	0	80	50	P	112	70	p
17	11	.	49	31	1	81	51	Q	113	71	q
18	12	.	50	32	2	82	52	R	114	72	r
19	13	.	51	33	3	83	53	S	115	73	s
20	14	.	52	34	4	84	54	T	116	74	t
21	15	.	53	35	5	85	55	U	117	75	u
22	16	.	54	36	6	86	56	V	118	76	v
23	17	.	55	37	7	87	57	W	119	77	w
24	18	.	56	38	8	88	58	X	120	78	x
25	19	.	57	39	9	89	59	Y	121	79	y
26	1A	.	58	3A	:	90	5A	Z	122	7A	z
27	1B	.	59	3B	;	91	5B	[123	7B	{
28	1C	.	60	3C	<	92	5C	\	124	7C	
29	1D	.	61	3D	=	93	5D]	125	7D	}
30	1E	.	62	3E	>	94	5E	^	126	7E	~
31	1F	.	63	3F	?	95	5F	_	127	7F	.

A. Beck


```
<html><head><title>CGI-Script - ASCII Codetabelle</title></head>
```

```
<body><h2><hr>
```

```
<pre>
```

dec	hex	Char	dec	hex	Char	dec	hex	Char	dec	hex	Char
0	0	.	32	20	.	64	40	@	96	60	`
1	1	.	33	21	!	65	41	A	97	61	a
2	2	.	34	22	"	66	42	B	98	62	b
.	.	.									
23	17	.	55	37	7	87	57	W	119	77	w
24	18	.	56	38	8	88	58	X	120	78	x
25	19	.	57	39	9	89	59	Y	121	79	y
26	1A	.	58	3A	:	90	5A	Z	122	7A	z
27	1B	.	59	3B	;	91	5B	[123	7B	{
28	1C	.	60	3C	<	92	5C	\	124	7C	
29	1D	.	61	3D	=	93	5D]	125	7D	}
30	1E	.	62	3E	>	94	5E	^	126	7E	~
31	1F	.	63	3F	?	95	5F	_	127	7F	.

```
</pre><hr>
```

```
<a href="http://www.informatik.htw-dresden.de/~beck/">A. Beck</A>
```

```
<p>
```

```
</body>
```

```
</html>
```

Formulare

Doku unter

<https://wiki.selfhtml.org/wiki/HTML/Elemente/form>

<https://wiki.selfhtml.org/wiki/HTML/Tutorials/Formulare>

```
<form action="http://myhost/cgi-bin/myprog.cgi" method="post">  
<!-- hier folgen die Formularelemente ->
```

```
</form>
```



Oder get

Parameterübergaben

GET	POST
<code>char* query</code> <code>query = getenv("QUERY_STRING");</code>	<code>char query[128];</code> <code>gets(query);</code>

Querystring:

QUERY_STRING=P1=Max,P2=Moritz

& trennt die Controlls

= Wert des Controlls, falls sinnvoll

Mit Funktionen aus `string.h` kann der Querystring untersucht werden

```
if(strstr(Param, "P1")) ...
```

Mit `char* GetPost=getenv("REQUEST_METHOD");`

kann man Get bzw. POST ermitteln

Environment

```
#include <stdio.h>

#include <ctype.h>

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

    char **pEnv=env;

    puts("Content-Type: text/html\n");

    puts("<html><head><title>CGI-Script CGI-Environment</title>");
    puts("</head><body><h2><hr>\n");
    puts("<pre>\n");
    while(*pEnv)
    {
        puts(*pEnv);pEnv++;
    }
    puts("</body></html>");
}

//http://www.htw-dresden.de/~beck/cgi-bin/env.cgi?P1:Max,P2:Moritz
```

TZ=NFT-1DFT,M3.5.0/02:00:00,M10.5.0/03:00:00
DOCUMENT_ROOT=/u/web/doc
HTTP_ACCEPT=image/png, image/*, */*
HTTP_ACCEPT_ENCODING=gzip
HTTP_HOST=www.htw-dresden.de
HTTP_USER_AGENT=Mozilla/3.0 (compatible; StarOffice/5.2;Linux)
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin
REMOTE_ADDR=212.185.244.91
REMOTE_HOST=pd4b9f45b.dip.t-dialin.net
REMOTE_PORT=61197
SCRIPT_FILENAME=/home/rex/fil/beck/public_html/cgi-bin/env.cgi
SERVER_ADDR=141.56.15.3
SERVER_ADMIN=schreiber@htw-dresden.de
SERVER_NAME=www.htw-dresden.de
SERVER_PORT=80
SERVER_SIGNATURE=Apache/1.3.9 Server at www.htw-dresden.de Port 80
SERVER_SOFTWARE=Apache/1.3.9 (Unix)
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.0
REQUEST_METHOD=GET
QUERY_STRING=P1=Max,P2=Moritz
REQUEST_URI=/~beck/cgi-bin/env.cgi
SCRIPT_NAME=/~beck/cgi-bin/env.cgi

Ausgabe von html

- Start mit

```
printf("Content-Type: text/html");  
printf("\n\n") ;
```

- Ausgabe von html-Text aus html-Datei

```
// F=fopen("/var/www/demo.html", "rt");  
F=fopen("demo.html", "rt");  
while (fgets(buf, 128, F))  
{  
    . . .  
    puts(buf);  
}  
fclose(F);
```

Hier variable Inhalte einpflegen
und html-Text programmtechnisch
erzeugen

Beispiel - Termine

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// http://www.htw-dresden.de/~beck/cgi-bin/env.cgi?Tag=31,Mon=12

int main()
{
    FILE * FT;
    int t,m,T,M;
    char *p;
    char *Param;
    char vBuf[128];
    char Text[128];

    FT=fopen("termine.txt","r");
    if (FT==NULL) puts("Dateifehler<p>");
```

Erstes Beispiel - Termine

```
else
{
    puts("Content-Type: text/html\n\n");
    puts("<html><head><title>CGI-Script Termine");
    puts("</title></head><body><h2><hr>\n");
    puts("<pre>");

    Param=malloc(1024); //(len+1);
    Param=getenv("QUERY_STRING");

    puts(Param);
    puts("Folgende Eintragungen wurden gefunden:");

    p=strstr(Param, "Tag"); p+=4;
    T=strtol(p, (char**)NULL, 10);

    p=strstr(Param, "Mon"); p+=4;
    M=strtol(p, (char**)NULL, 10);
    printf("Tag: %d, Mon: %d<br>", T, M);
}
```


Erstes Beispiel - Termine

```
while (fgets (vBuf, 128, FT) )
{
    sscanf (vBuf, "%d.%d. %[^\n]", &t, &m, Text);
    if (T==t && M==m)
        printf ("%d.%d. :%s<p>", t, m, Text);
}

puts ("<p>");

puts ("</pre><hr>");
puts ("<a href=\"http:www.htw-dresden.de/~beck/a.beck.html\">");
puts ("A. Beck</A> <p>");
puts ("</body>\n");
puts ("</html>\n");
}
return 0;
}
```

Ohne Formulareingabe, GET-Request

Http-link mit Parametern (GET-Request)

CGI-Script Termine - Mozilla Firefox

CGI-Script Termine x Gedenk- und Geburtstage x +

ipc145.informatik.htw-dresden.de/~beck/cgi-bin/ter1.cgi?Tag=7,Mon=12

Tag=7,Mon=12
Folgende Eintragungen wurden gefunden:
Tag: 7, Mon: 12
7.12. :Geburtstag Hans Huckebein
7.12. :Witwe Bolte verstorben
7.12. :Das ist der 07. Dezember

puts(Param);

printf("%d.%d. :%s<p>",t,m,Text);

printf("Tag: %d, Mon: %d
",T,M);

[A. Beck](#)

GET oder POST

- Es scheint sinnvoll, den GET-Request zu verwenden, um ein Formular (erstmalig) auszugeben.
- POST ist gut geeignet, um auf eine Formularausgabe zu reagieren.

Ein Formular



The screenshot shows a Mozilla Firefox browser window titled "Gedenk- und Geburtstage". The address bar contains the URL "ipc145.informatik.htw-dresden.de/~beck/cgi-bin/ter2.cgi". Below the address bar, there is a form with two input fields: "Tag" and "Monat". A "Senden" button is located below the "Tag" field.

- Ein Formular wird ausgefüllt und via Submitbutton (hier senden) abgeschickt.
- Es kann ein je nach Form-tag ein GET oder POST-Request generiert werden.

Die html-Datei mit Formular

```
<!doctype html>
<html>
<head>
<title>Gedenk- und Geburtstage</title>
</head>
<body>
  <form name="Termine" action="ter2.cgi" METHOD="POST" id="terr">
    <table cellpadding="4" cellspacing="0">
      <tr>
        <td style="width:50px"> <label>Tag</label> </td>
        <td style="width:50px"> <input type="text" name="Tag"
        > </td>
        <td style="width:50px"> <label>Monat</label> </td>
        <td style="width:50px"> <input type="text" name="Mon"
        > </td>
      </tr>
      <tr>
        <td> <input type="submit" value="Senden"></td>
      </tr>
    </table>
  </form>
  <br><hr><br>
</body>
</html>
```

Hier wird der
Tag eingefügt

Hier wird der
Monat eingefügt

Hier werden die
Daten eingefügt

Beispiel mit Formular

```
int main()
{
    FILE * ft, *fhtml;
    char *p, text[128];
    char *Method;
    int dayr, monr; // Tag/Monat aus Request

    ft=fopen("termine.txt", "r");
    fhtml=fopen("termine.html", "r");
    if (ft==NULL || fhtml==NULL) puts("Dateifehler<p>");
    else
    {
        puts("Content-Type: text/html\n\n\r");
        Method=getenv("REQUEST_METHOD");
//        puts(Method);
        if (strcmp(Method, "GET")==0)
        // Ausgabe des leeren Formulars
        {
            while(fgets(text, 128, fhtml)) puts(text);
        }
    }
}
```

```

else
if (strcmp(Method, "POST")==0)
// Ausgabe ausgefülltes Formular und Daten
{
    // Lesen des Querystrings
    fgets(text, 128, stdin);
    printf("QueryString: %s<br>\n", text);
    p=strstr(text, "Tag=");
    dayr=atoi(p+strlen("Tag="));
    p=strstr(text, "Mon=");
    monr=atoi(p+strlen("Mon="));
    while(fgets(text, 128, fhtml))
    {
        if (strstr(text, "name=\"Tag\"") !=NULL)
            {puts(text); printf("value=%02d", dayr);}      else
        if (strstr(text, "name=\"Mon\"") !=NULL)
            {puts(text); printf("value=%02d", monr);}     else
        if (strstr(text, "</form>") !=NULL)
            {puts(text); putData(ft, dayr, monr);}        else
        puts(text);
    }
}
}
return 0;
}

```

```

int putData(FILE* ft,int dayr, int monr)
{
    int dayt, mont, n=0;
    char buf[128],text[128], *p;
    puts("<br><hr><br>");
    while (fgets(buf,128,ft))
    {
        strcpy(text,buf);
        p=strtok(buf," .\t");
        if (p)
        {
            dayt=atoi(p);
            if (dayt)
            {
                p=strtok(NULL," .\t");
                if (p)
                {
                    mont=atoi(p);
                    while(strtok(NULL," .\t"));
                    if (dayt==dayr &&mont==monr)
                    { n++; printf("%s<br>",text); }
                }
            }
        }
    }
    return n;
}

```

Tag und Monat
Aus dem Request

Tag aus Datei

Monat aus Datei

Ausgabe des
gefundenen
Textes

Die Ergebnis-html-Datei mit Formular

```
<!doctype html>
<html>
<head>
<title>Gedenk- und Geburtstage</title>
</head>
<body>
  <form name="Termine" action="ter2.cgi" METHOD="POST" id="termine">
    <table cellpadding="4" cellspacing="0">
      <tr>
        <td style="width:50px"> <label>Tag</label> </td>
        <td style="width:50px"> <input type="text" name="Tag"
value=07      > </td>
        <td style="width:50px"> <label>Monat</label> </td>
        <td style="width:50px"> <input type="text" name="Mon"
value=12      > </td>
      </tr>
      <tr>
        <td> <input type="submit" value="Senden"></td>
      </tr>
    </table>
  </form>
  <br><hr><br>
  7.12. Geburtstag Hans Huckebein
  <br>07.12. Witwe Bolte verstorben
  <br>07.12. Das ist der 07. Dezember
  <br><br><hr><br>
</body>
</html>
```

Hier wurde der
Tag eingefügt

Hier wurde der
Monat eingefügt

Hier wurden die
Daten eingefügt

Ergebnis

Gedenk- und Geburtstage - Mozilla Firefox

CGI-Script Termine × Gedenk- und Geburtstage × +

← → ↻ 🏠 ⓘ ipc145.informatik.htw-dresden.de/~beck/cgi-bin/ter2.cgi

Tag Monat

7.12. Geburtstag Hans Huckebein
07.12. Witwe Bolte verstorben
07.12. Das ist der 07. Dezember
