**Listing 1. Główna część programu serwera Web**

#define HEADER "\

<html>\

<head>\

<meta charset=UTF-8>\

</head>\

<body bgcolor=green>\

<h1>Welcome to Led HTTP Server</h1>\

<h2>Touch one of the circles to toggle the led</h2>\

"

#define FOOTER "\

<svg width='0'\

height='0'>\

<defs>\

<g id='r'>\

<circle cx='32' cy='32' r='30' fill='red'/>\

</g>\

</defs>\

<defs>\

<g id='w'>\

<circle cx='32' cy='32' r='30' fill='white'/>\

</g>\

</defs>\

</svg>\

</body>\

</html>\

"

int fdleds**[**8**];**

int ledState**[**8**];**

int a **=** 0**;**

static int ahc\_ledhttpserver**(**void **\*** cls**,** struct MHD\_Connection **\*** connection**,**

const char **\*** url**,** const char **\*** method**,** const char **\*** version**,**

const char **\*** upload\_data**,** size\_t **\*** upload\_data\_size**,** void **\*\*** ptr**);**

static void initLeds**();**

static int toggleLed**(**int led**);**

int main**(**int argc**,** char **\*\*** argv**)**

**{**

printf**(**"Starting...\n"**);**

struct MHD\_Daemon **\*** d**;**

**if** **(**argc **!=** 2**)**

**{**

printf**(**"Usage: %s PORT\n"**,** argv**[**0**]);**

**return** 1**;**

**}**

initLeds**();**

d **=** MHD\_start\_daemon**(**MHD\_USE\_THREAD\_PER\_CONNECTION**,** atoi**(**argv**[**1**]),**

**NULL,**

**NULL,** **&**ahc\_ledhttpserver**,**

**NULL,** MHD\_OPTION\_END**);**

**if** **(**d **==** **NULL)** **return** 1**;**

**(**void**)** getc**(**stdin**);**

MHD\_stop\_daemon**(**d**);**

**return** 0**;**

**}**

static int ahc\_ledhttpserver**(**void **\*** cls**,** struct MHD\_Connection **\*** connection**,**

const char **\*** url**,** const char **\*** method**,** const char **\*** version**,**

const char **\*** upload\_data**,** size\_t **\*** upload\_data\_size**,** void **\*\*** ptr**)**

**{**

char **\***page**;**

char header**[]** **=** HEADER**;**

char footer**[]** **=** FOOTER**;**

char **\***lines**[**8**];**

int size **=** 0**;**

static int dummy**;**

struct MHD\_Response **\*** response**;**

int ret**,** i**;**

//Checking for various things

**if** **(**0 **!=** strcmp**(**method**,** "GET"**))** **return** MHD\_NO**;** /\* unexpected method \*/

**if** **(&**dummy **!=** **\***ptr**)**

**{**

/\* The first time only the headers are valid,

do not respond in the first round... \*/

**\***ptr **=** **&**dummy**;**

**return** MHD\_YES**;**

**}**

**if** **(**0 **!=** **\***upload\_data\_size**)** **return** MHD\_NO**;** /\* upload data in a GET!? \*/

**\***ptr **=** **NULL;** /\* clear context pointer \*/

//End of checking

int toggledLed **=** **-**1**;**

**if(**strlen**(**url**)** **>** 1**)**

**{**

**if((**int**)**url**[**1**]** **>=** '0' **&&** **(**int**)**url**[**1**]** **<=** '7'**);**

toggledLed **=** url**[**1**]** **-** '0'**;**

int res**;**

res **=** toggleLed**(**toggledLed**);**

ledState**[**toggledLed**]** **=** res**;**

**}**

char **\***imgStr **=** "<a href=/%i><svg width=\"70\" height=\"70\"><use xlink:href=\"#%s\" /></svg></a>\n"**;**

**for** **(**i **=** 0**;** i **<** 8**;** i**++)**

**{**

lines**[**i**]** **=** **(**char**\*)** malloc**(**strlen**(**imgStr**)** **+** 50**);**

sprintf**(**lines**[**i**],** imgStr**,** i**,** ledState**[**i**]** **==** 1 **?** "r" **:** "w"**);**

size **+=** strlen**(**lines**[**i**]);**

**}**

size **+=** strlen**(**header**);**

size **+=** strlen**(**footer**);**

page **=** **(**char**\*)** malloc**(**size **+** 1**);**

strcpy**(**page**,** header**);**

**for(**i **=** 7 **;** i **>=** 0 **;** i**--)**//Inverting, because 7th led is on the left

**{**

strcat**(**page**,** lines**[**i**]);**

free**(**lines**[**i**]);**

**}**

strcat**(**page**,** footer**);**

response **=** MHD\_create\_response\_from\_buffer**(**strlen**(**page**),** **(**void**\*)** page**,**

MHD\_RESPMEM\_PERSISTENT**);**

ret **=** MHD\_queue\_response**(**connection**,**

MHD\_HTTP\_OK**,** response**);**

MHD\_destroy\_response**(**response**);**

**return** ret**;**

**}**