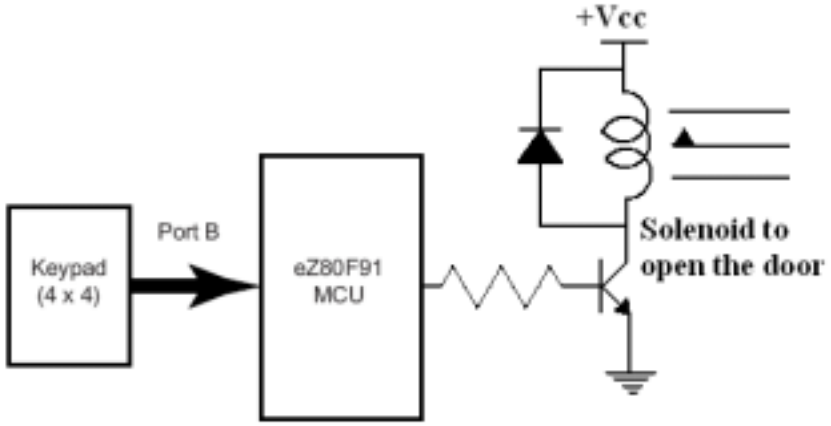
# *Ingress control* (Project Number eZ2997)

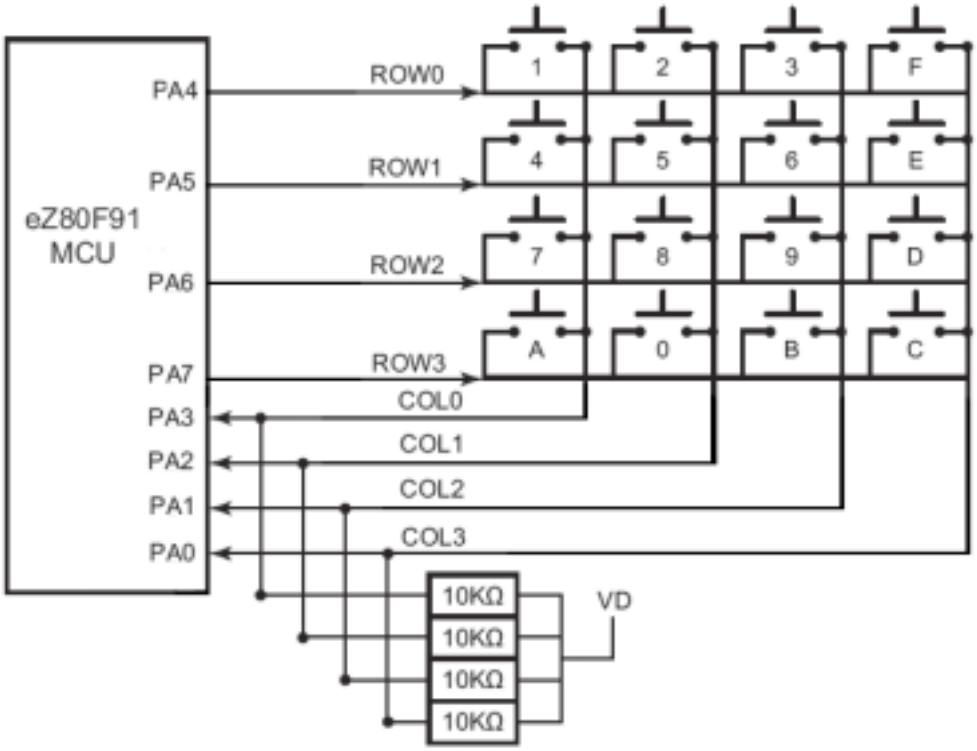
***Abstract:***
Ingress control is a device that permits the entry of an individual to a place when this provides his/ her ID number and password. A database with information about all the users is stored in the embedded circuit, eZ80 Acclaim Contest Circuit via TCP/ IP. This database is generated from any computer connected to TCP/ IP, where the embedded circuit is connected to and an IP address is previously assigned. Connection to the Ingress Control Circuit shows a main page with links to add a new user, delete a user, show users information and set its real time clock.

Information that is stored in Ingress Control Circuit is user's name, ID number (4 digits), Password (4 digits), flag to indicate everyday access at certain time, flag to indicate date from the time the user has permission, and this date; flag to indicate date until the time the user has permission to enter and this date.

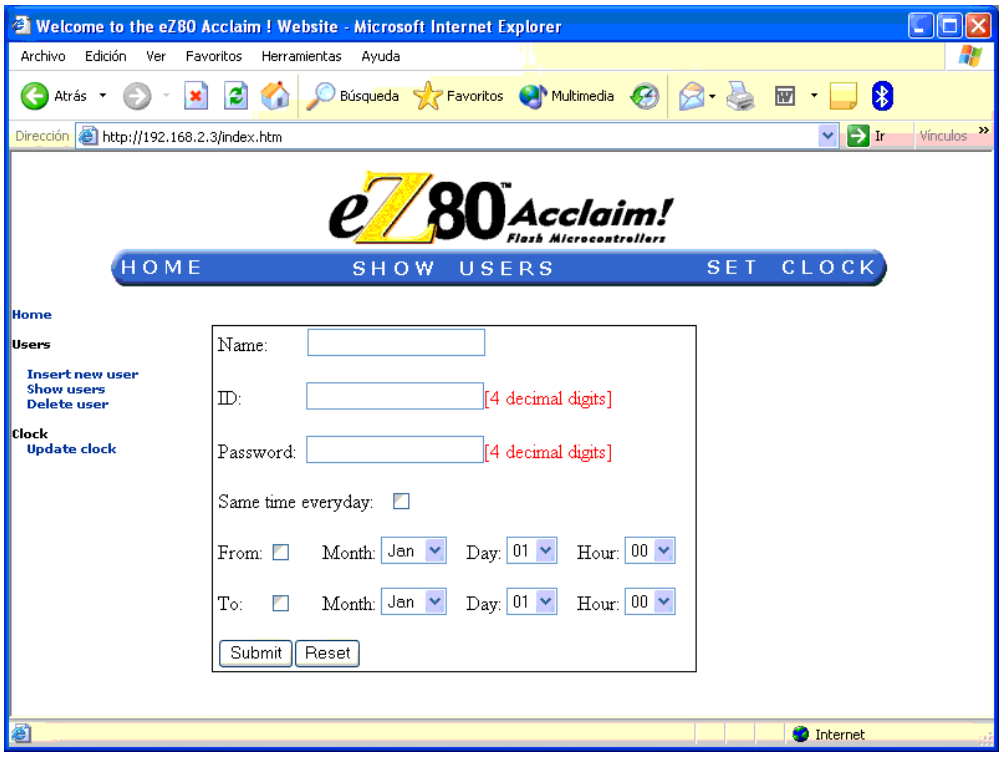
With this information the system administrator can allow entering a user, for example, every day, but restricted only from 8:00 am to 6:00 pm, or can restrict the entrance of a user for some days. When flags are not selected, the current user does not have any restriction to enter. Here there is a picture of the Ingress Control Circuit:

The hardware consists of the ZiLOG's eZ80F91 MCU as the main unit and a keyboard connected to it. Also, there is a digital output line that is activated for five seconds when the user successfully enters his/ her ID number and password.


+Vcc
Keypad
(4 x 4)
Port B
eZ80F91
MCU
Solenoid to
open the door


eZ80F91
MCU
PA4
PA5
PA6
PA7
PA3
PA2
PA1
PA0
ROW0
ROW1
ROW2
ROW3
COL0
COL1
COL2
COL3
1
2
3
F
4
5
6
E
7
8
9
D
A
0
B
C
10KΩ
10KΩ
10KΩ
10KΩ
VD

Software implemented is based on the ZiLOG TCP/IP stack (ZTP) and it also involves integrating user web pages into the web server using ZDSII. Page forms are used to send information to Ingress Circuit and dynamic pages where also generated from Ingress Circuit to the client computer. The entire program was written in C and pages were integrated to the program very easy using the software mentioned above. An example of a web page retrieve from Ingress Control Circuit is shown next.




A listing part of the program, where it is shown how to insert a new user, once the client computer sends the form page with all information about the user, is presented here.

Listing part of the C code for inserting a new user.

```
/* *********************************************************
* File : access_cgi.c
* Description :this file contane the  function, which communicate
with ZTP
*
********************************************************************* */


/*****************************************************************
```

```
**    includes
*****************************************************************/


#include <stdlib.h>
#include <httpd.h>
#include <access_form.h>

extern struct access_perm access_member[20];
extern int member_ptr;


/*****************************************************************
**    CGI function
*****************************************************************/


int access_cgi(struct http_request *request)
{
      char  *str1;
      int input_sw;
      int temp;
      char temp_sj[30];
      char Form_reply[] = {"Form has been submitted. For next
form, press back button of browser."};

      http_output_reply(request,HTTP_200_OK);

      str1 = http_find_argument(request,(unsigned char*)"name");
      strcpy(access_member[member_ptr].access_name,str1);
      str1 = http_find_argument(request,(unsigned char*)"ID");
      strcpy(access_member[member_ptr].access_id,str1);
      str1 = http_find_argument(request,(unsigned char*)"psw");
      strcpy(access_member[member_ptr].access_psw,str1);
      str1 = http_find_argument(request,(unsigned
char*)"sametime");
      if(str1[0]=='O')
            access_member[member_ptr].access_everyday= 0x1;
      else
            access_member[member_ptr].access_everyday= 0x0;
      str1 = http_find_argument(request,(unsigned char*)"from");
      if(str1[0]=='O')
            access_member[member_ptr].access_from= 0x1;
      else
            access_member[member_ptr].access_from= 0x0;
      str1 = http_find_argument(request,(unsigned char*)"to");
      if(str1[0]=='O')
            access_member[member_ptr].access_to= 0x1;
      else
            access_member[member_ptr].access_to= 0x0;
      str1 = http_find_argument(request,(unsigned char*)"month1");
      input_sw = atoi(str1);
      access_member[member_ptr].access_fromM=input_sw;
      str1 = http_find_argument(request,(unsigned char*)"day1");
      input_sw = atoi(str1);
      access_member[member_ptr].access_fromD=input_sw;
      str1 = http_find_argument(request,(unsigned char*)"hour1");
      input_sw = atoi(str1);
      access_member[member_ptr].access_fromH=input_sw;
      str1 = http_find_argument(request,(unsigned char*)"month2");
      input_sw = atoi(str1);
```

```
      access_member[member_ptr].access_toM=input_sw;
      str1 = http_find_argument(request,(unsigned char*)"day2");
      input_sw = atoi(str1);
      access_member[member_ptr].access_toD=input_sw;
      str1 = http_find_argument(request,(unsigned char*)"hour2");
      input_sw = atoi(str1);
      access_member[member_ptr].access_toH=input_sw;


__http_write(request,Form_reply,strlen(Form_reply));
__http_write(request,"<h5> Software Version: ",23);
__http_write(request,(char*)VERSION,strlen((char*)VERSION));

      if(member_ptr<20)
            member_ptr++;// initially, no more than 20 users
}


/*************************************************************
**    end of file
*************************************************************
```