 **Faculty of Automatic Control and Computer Engineering**

**IASI**

SMART GARAGE

**Studen *ț*i :**

**Nicu Iulian-Andrei**

**Zapodeanu Adrian-Vlad**

**Grupa: 1406A**

2019 – 2020

**Description**

A smart garage is a building, that is equipped with special structured wiring to enable occupants to remotely control or program an array of automated home electronic devices by entering a single command. For example, a homeowner can use a web page to, control lighting, program a system for open or close the garage door , and perform many other tasks.

**Introduction**

* Smart Garage project was inspired by a simple rule : control of electronic devices from web page. So, in this project we succeed to implement:
* A Bluetooth communication from which we can control:
* An 12 V Led which simulate the lights from garage.
* A servo motor with which we open and close the door.

**Used Materials**

* An XMC 1100 Development Board.
* Raspberry Pi 3 Model B+
* Electrical Resistances (An electrical quantity that measures how the device or material reduces the [electric current](http://www.rapidtables.com/electric/Current.htm) flow through it.
* Red 5V Diffused LED with a viewing angle of 60 degrees.
* Servomotor MG996R

**Source Cod**

**#include** <DAVE.h>

uint8\_t data[] = "Infineon Technologies";

**char** c[2];

**int** **main**(**void**)

{

DAVE\_STATUS\_t status;

status = DAVE\_Init();

UART\_SetRXFIFOTriggerLimit (&seriala, 1);

**if**(status != *DAVE\_STATUS\_SUCCESS*)

{

XMC\_DEBUG("DAVE APPs initialization faidir1\n");

}

**else**{

**while**(1){

**if**(UART\_GetRXFIFOStatus(&seriala)) // caracater disponibil

{

UART\_ClearRXFIFOStatus(&seriala, 2); // anuleaza stare

c[0]= UART\_GetReceivedWord(&seriala);

c[1]= UART\_GetReceivedWord(&seriala);

**if**(c[0]=='a')

{

DIGITAL\_IO\_SetOutputHigh(&DIGITAL\_IO\_0);

}

**if**(c[0]=='b')

{

DIGITAL\_IO\_SetOutputLow(&DIGITAL\_IO\_0);

}

**if**(c[0]=='c')

{

PWM\_SetDutyCycle(&PWM\_0,810);

}

**if**(c[0]=='d')

{

PWM\_SetDutyCycle(&PWM\_0,480);

}

}

}

}

}

[**Web page source activates**](http://embedac.ro/SI/Lab/Lab6/L6.txt)

<html>

<head>

<?php

if (isset($\_POST['ron'])) // Red on

{

exec('sudo python /var/www/pin7l.py');

}

if (isset($\_POST['roff'])) // Red Off

{

exec('sudo python /var/www/pin7h.py');

}

if (isset($\_POST['gon'])) // Green On

{

exec('sudo python /var/www/pin12l.py');

}

if (isset($\_POST['goff'])) // Green Off

{

exec('sudo python /var/www/pin12h.py');

}

if (isset($\_POST['bon'])) // Blue On

{

exec('sudo python /var/www/pin3l.py');

}

if (isset($\_POST['boff'])) // Blue Off

{

exec('sudo python /var/www/pin3h.py');

}

?>

<title>L6\_web server generalizat pt. Web of Things </title>

</head>

<body background="FX3.png">

<center>

<table witdh="400" border="1" bgcolor="silver">

<td>

<Font color='blue'><b>

<center>

Universitatea Tehnica 'Gh.Asachi' Iasi - Fac. Automatica si Calculatoare Zapodeanu Adrian Vlad, Nicu Iulian Andrei, Grupa 1406A<br>

<img src="Mbed\_mar.png" width="200" height="130"> </center>

</Font><b>

<center> <h1> <Font color='navy'>

L6 - Server generalizat ptentru<br> Web of Things <br>

</h1></font>

<center>

<form method="post">

<table

style="width: 50%; text-align: left; margin-left: auto; margin-right: auto;"

border="2" cellpadding="2" cellspacing="2">

<tr>

<td style="text-align: center;"><button name="ron">RED ON</button></td>

<td style="text-align: center;"><button name="roff">RED OFF </button></td>

</tr>

<tr>

<td style="text-align: center;"><button name="gon">GREEN ON</button></td>

<td style="text-align: center;"><button name="goff">GREEN OFF</button></td>

</tr>

<tr>

<td style="text-align: center;"><button name="bon">BLUE ON</button></td>

<td style="text-align: center;"><button name="boff">BLUE OFF</button></td>

</tr>

</tbody>

</table>

</form>

<br>

&copy 2020 Fac. Automatica si Calculatoare

</td>

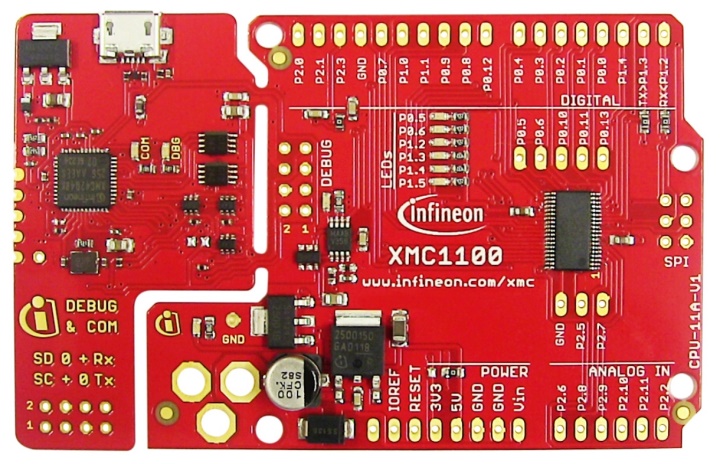
</table>

</body>

</html>

|  |  |
| --- | --- |
| import RPi.GPIO as GPIO  import time  GPIO.setmode(GPIO.BOARD)  GPIO.setup(7,GPIO.OUT)  *GPIO.output(7,0)*  *# time.sleep(0.2)*  *# GPIO.output(7,0)*  *# time.sleep(0.2)* | *import RPi.GPIO as GPIO*  *import time*  *GPIO.setmode(GPIO.BOARD)*  *GPIO.setup(7,GPIO.OUT)*  GPIO.output(7,1)  # time.sleep(0.2)  # GPIO.output(7,0)  # time.sleep(0.2) |
| import RPi.GPIO as GPIO  import time  GPIO.setmode(GPIO.BOARD)  GPIO.setup(12,GPIO.OUT)  #while True:  GPIO.output(12,0)  # time.sleep(0.2)  # GPIO.output(24,GPIO.LOW)  # time.sleep(0.2) | import RPi.GPIO as GPIO  import time  GPIO.setmode(GPIO.BOARD)  GPIO.setup(12,GPIO.OUT)  #while True:  GPIO.output(12,GPIO.HIGH)  # time.sleep(0.2)  # GPIO.output(24,GPIO.LOW)  # time.sleep(0.2) |
|  |  |

**Used Materials**

.