**Controlul pinilor GPIO ai unei plăci de dezvoltare Raspberry PI dintr-o interfață web**

**Morariu Bogdan**

**AN 3 CTI**

**Grupa 4.2**

**Requirements**

•Interfața web va conține o opțiune de selecție a ieșirilor / intrărilor (GPIO) active de pe placa de dezvoltare. Pinii GPIO nefolosiți vor fi powered-down.

• Interfața web va permite selecția modului de utilizare a fiecărui pin activ al plăcii de dezvoltare: on / off (0 sau 1 logic) sau posibilitate de generare semnal PWM;

• Se va implementa o metodă securizată pentru logarea în interfața web, de exemplu cu Google VPN Authenticator;

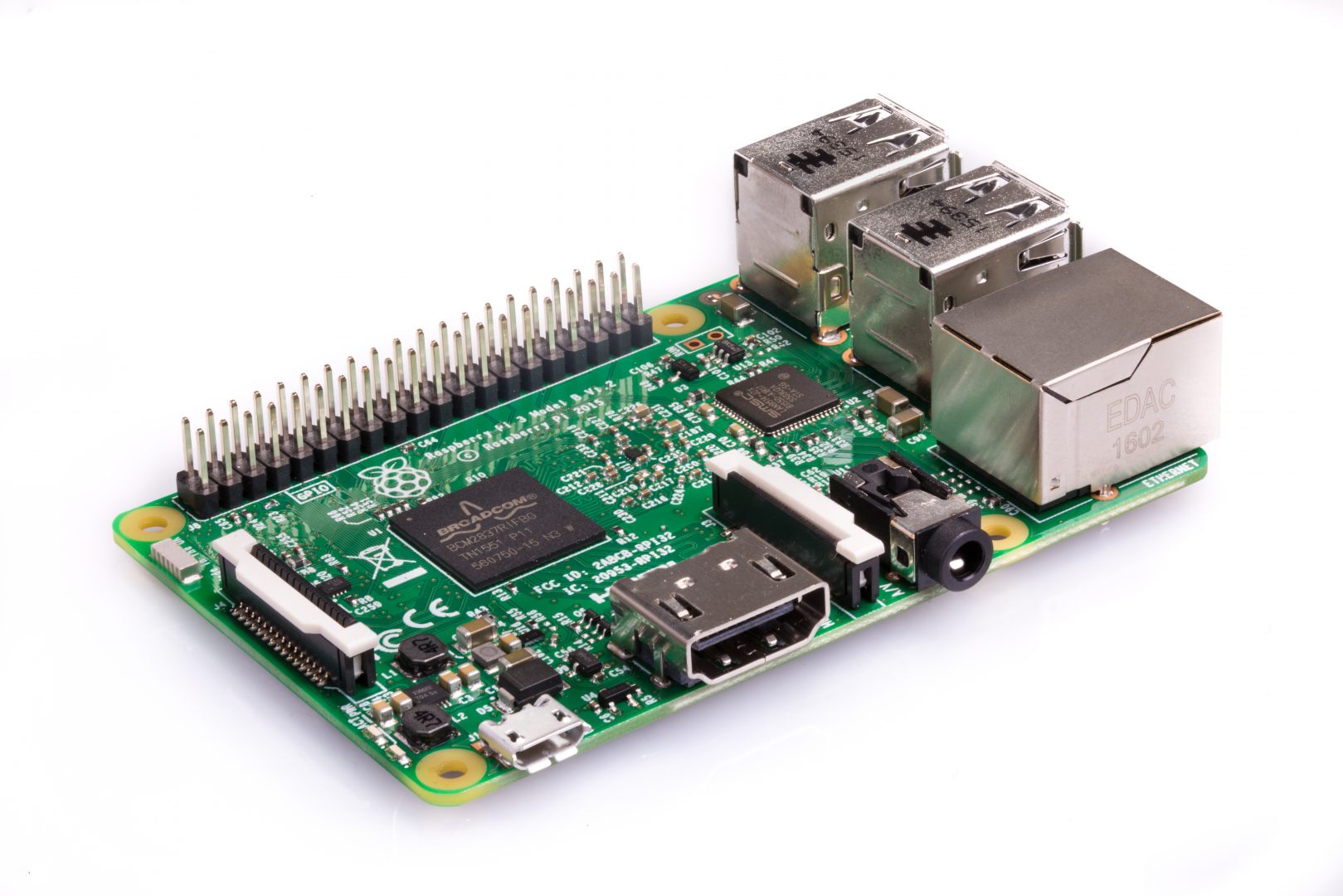
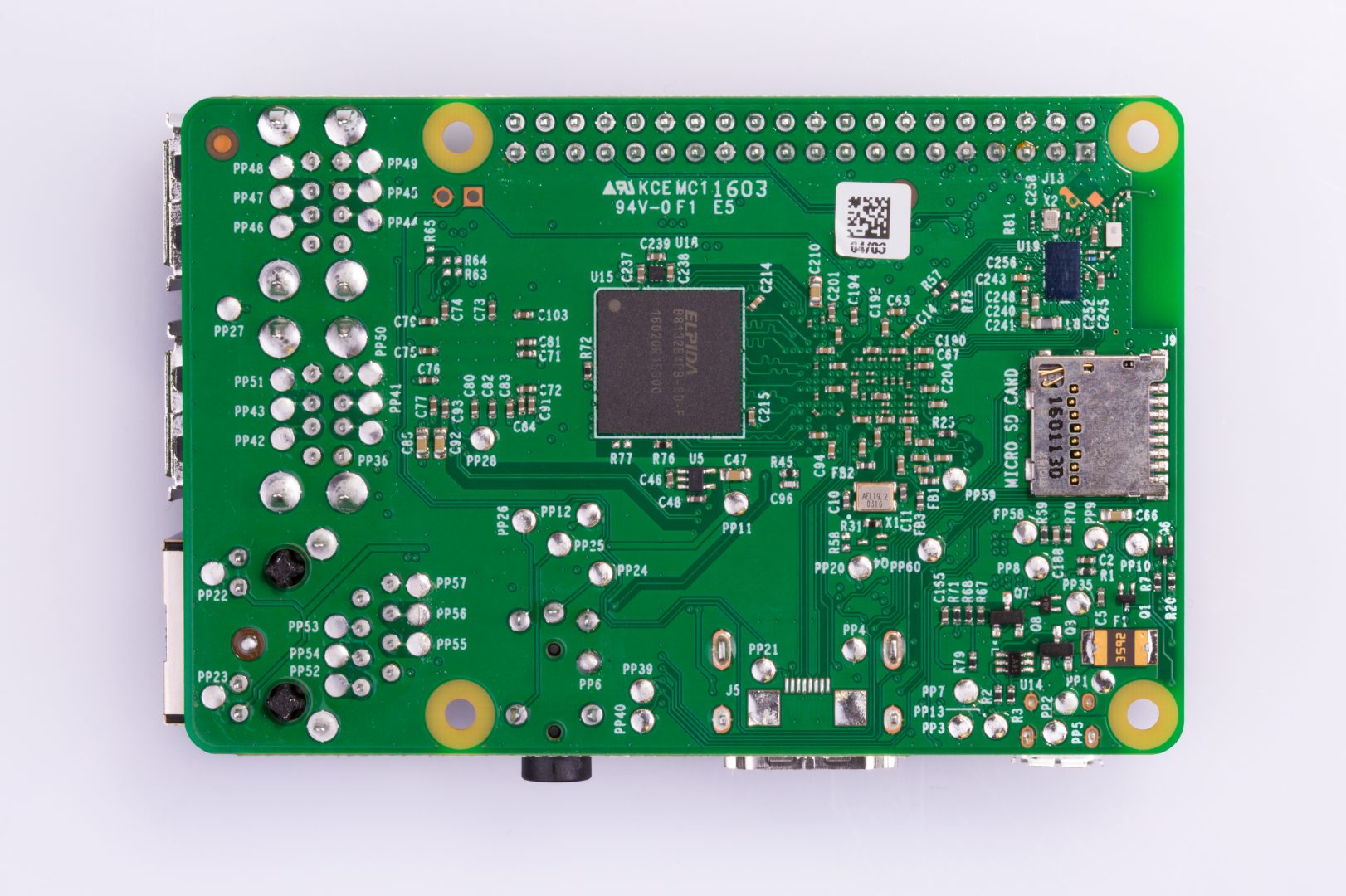
• Starea setărilor curente dorite de utilizator va fi salvată fie în memoria EEPROM a microcontrolerului, fie pe un card SD, astfel încât, la întreruperea alimentării, sistemul să repornească din ultima stare stabilă cunoscută.

# Raspberry Pi 3 Model B

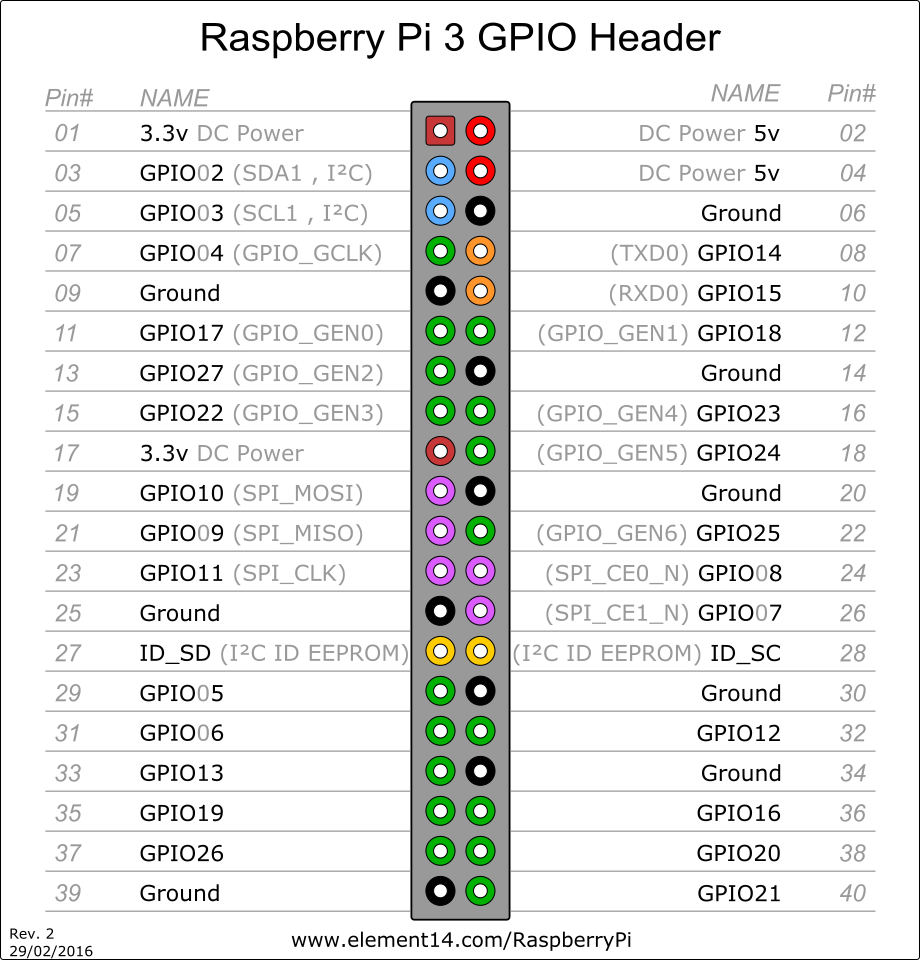
The Raspberry Pi 3 Model B is the earliest model of the third-generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016.

Specifications:

* Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
* 1GB RAM
* BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
* 100 Base Ethernet
* 40-pin extended GPIO
* 4 USB 2 ports
* 4 Pole stereo output and composite video port
* Full size HDMI
* CSI camera port for connecting a Raspberry Pi camera
* DSI display port for connecting a Raspberry Pi touchscreen display
* Micro SD port for loading your operating system and storing data
* Upgraded switched Micro USB power source up to 2.5A



The Raspberry Pi Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third-party Ubuntu, Windows 10 IoT Core, RISC OS, and specialised media centre distributions. It promotes Python and Scratch as the main programming languages, with support for many other languages. The default firmware is closed source, while an unofficial open source is available. Many other operating systems can also run on the Raspberry Pi, including the formally verified microkernel, seL4. Other third-party operating systems available via the official website include Ubuntu MATE, Windows 10 IoT Core, RISC OS and specialised distributions for the Kodi media centre and classroom management.



RaspberryPi is based on an ARM architecture controller(ARMv8-A) : Broadcom BCM2837, which has 4× Cortex-A53 at 1.2 GHz.

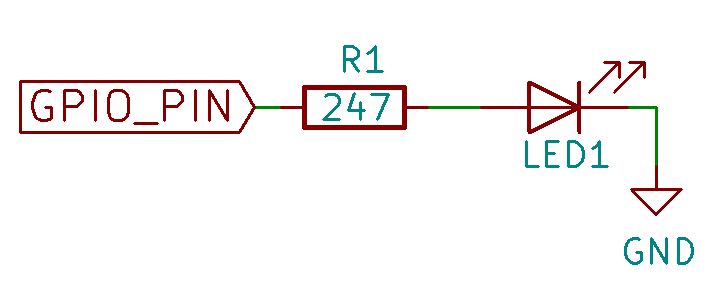
From a peripheral POV, Raspberry Pi contains 17 GPIO + other functionalities specific to applications with microcontrollers: UART, I2C, SPI, I2S audio.

All the GPIO pins can be set as input and output. When they are configured as output, we can set the logic level of the pin "1" logic (~3,3V), respectively "0" logic (~0V).

Using the timeing units, the pins can be configured as PWM(Pulse width modulation). In this mode, for every pin we can set a frequency, respectively a duty cycle.

The LEDs are connected between the GPIO pin and Ground alongside a resistance(rezistenta de limitare a curentului), which assures that approximatively 10mA pass through the LEDs, to have a good luminosity without excessively using the pin. In these conditions, level "1" logic on the pin means the LED is on, and "0" means the LED is off.

To be able to control the luminosity of a LED using a PWM signal, we set the frequency at 100Hz(so that the human eye can perceive the blink), while the duty-cycle can be selected on numeric steps on 10 bits(from 0 to 1023, where 0 means static OFF , and 1023 means static ON).For example, 512 is the state where the luminosity is set on half.

****

For this project I have used PHP and WiringPi.

***WiringPi*** is a ***PIN*** based GPIO access library written in C for the BCM2835, BCM2836 and BCM2837 SoC devices used in all **Raspberry Pi.** versions. It’s released under the GNU LGPLv3 license and is usable from C, C++ and RTB (BASIC) as well as many other languages with suitable wrappers It’s designed to be familiar to people who have used the Arduino “wiring” system and is intended for use by experienced C/C++ programmers. It is not a newbie learning tool.

***WiringPi*** is developed directly on a Raspberry Pi running 32-bit Raspbian. I do not support any other platform, cross compiling or operating systems. It has been ported to other platforms, other operating systems and some are cross compiling, however this author does not maintain those systems. If you are trying to use ***wiringPi*** on a platform other than the Raspberry Pi with Raspbian then you must contact the person who did the port and not me.

***WiringPi*** includes a command-line utility **gpio** which can be used to program and setup the GPIO pins. You can use this to read and write the pins and even use it to control them from shell scripts.

***WiringPi*** is extendable and modules are provided to extend ***wiringPi*** to use analog interface devices on the Gertboard, and to use the popular **MCP23x17/MCP23x08** (I2C 7 SPI) GPIO expansion chips, as well as  module that will allow blocks of up to 4 **74×595**shift registers to be daisy-chained together for an additional 32-bits worth of output as a single unit. (You can have several blocks of 4 74x595s if needed) One of the extension modules allows you to use an ATmega (e.g. Arduino, or the Gertboard) as more GPIO expansion too – via the Pi’s serial port.

**Functions** **used**:

* **void pwmWrite (int pin, int value) ;**

Writes the value to the PWM register for the given pin. The Raspberry Pi has one on-board PWM pin, pin 1 (BMC\_GPIO 18, Phys 12) and the range is 0-1024. Other PWM devices may have other PWM ranges.This function is not able to control the Pi’s on-board PWM when in *Sys* mode.

* **void pinMode (int pin, int mode) ;**

This sets the mode of a pin to

either **INPUT**, **OUTPUT**, **PWM\_OUTPUT**or**GPIO\_CLOCK**. Note that only ***wiringPi*** pin 1 (BCM\_GPIO 18) supports PWM output and only ***wiringPi*** pin 7 (BCM\_GPIO 4) supports CLOCK output modes.

This function has no effect when in *Sys* mode. If you need to change the pin mode, then you can do it with the **gpio** program in a script before you start your program.

* **gpio -g …**

The optional **-g** flag causes pin numbers to be interpreted as BCM\_GPIO pin numbers rather than standard ***wiringPi*** pin numbers.

* **gpio mode <pin> in/out/pwm/clock/up/down/tri**

This sets the mode of a pin to be input, output, pwm or clock mode, and additionally can set the internal pull-up/down resistors to pull-up, pull-down or none.

* **gpio write <pin> 0/1**

This sets an output pin to high (1) or low (0)

* **gpio pwm <pin> <value>**

Set the pin to a PWM value (0-1023 is supported)

* **gpio read <pin>**

Reads and prints the logic value of the given pin. It will print 0 (low) or 1 (high).

**Function examples:**

system("gpio -g mode x out"); -> sets the mode of the pin “x” to be output

system("gpio -g write x 1"); -> sets the output of the pin “x” to high(led is on)

system("gpio -g write x 0"); -> sets the output of the pin “x” to low(led is off)

**Installing the software:**

* 1. sudo apt–get update
  2. sudo apt–get upgrade
  3. Installing wiringPI:
* git clone git://git.drogon.net/wiringPi
* cd wiringPi
* ./build
  1. Installing nginx and PHP:
     + sudo apt-get install nginx
     + sudo apt-get install php5-fpm php-apc
     + sudo nano /etc/nginx/sites-available/default
     + Then you want to add these lines below the  # listen [::]::80 line.
     + After that, use sudo nginx –t
     + Sudo systemctl reload nginx

listen 80;

server\_name $domain\_name;

root /var/www;

index index.html index.htm;

access\_log /var/log/nginx/access.log;

error\_log /var/log/nginx/error.log;

location ~\.php$ {

fastcgi\_pass unix:/var/run/php5-fpm.sock;

fastcgi\_split\_path\_info ^(.+\.php)(/.\*)$;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_param HTTPS off;

try\_files $uri =404;

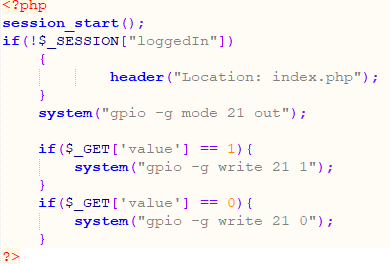
include fastcgi\_params;

}

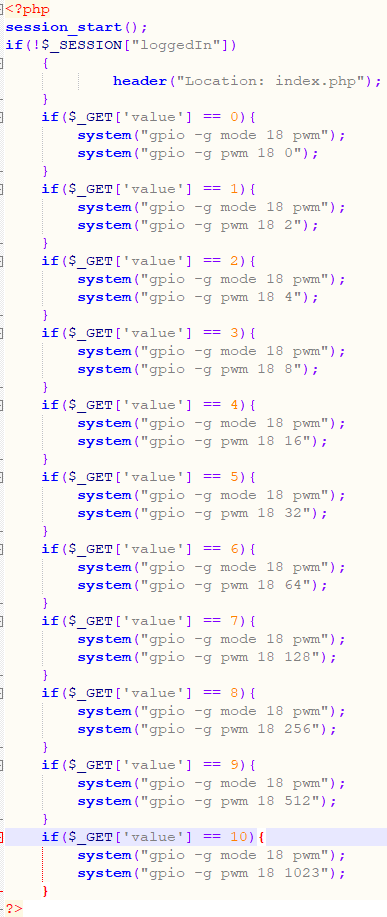
Afterwards, you run the project on the ip address of the pi, found in the ifconfig.

**CODE EXAMPLES**

Example with input (1 LED ON / 0 LED OFF)



Example with PWM (Value goes from 0 to 10 -> values from 0->1024)



**BIBLIOGRAFIE**

<http://wiringpi.com/examples/>

<http://www.tinkernut.com/>

**GITHUB**

Github link to the project: <https://github.com/bmorariu/SI>