

Raspberry Pi jako wyłącznik ethernetowy



Ostatnio triumf popularności święci komputerki linuksowy Raspberry Pi oparty na układzie Broadcom BCM2835. Zyskał on taką sławę, że pisano o nim w prasie ogólnopolskiej oraz „ogólno-komputerowej” co wynika najprawdopodobniej z bardzo atrakcyjnej ceny. Najwyższy czas, aby obok całego tego medialnego szumu opisać jakieś zastosowania praktyczne. W artykule pokażemy jak przygotować prostą aplikację „przełącznika ethernetowego” umożliwiającą włączenie i wyłączenie urządzeń dołączonych do portów GPIO, za pomocą przeglądarki internetowej.

Komputer Raspberry Pi ma stosunkowo wydajną jednostkę centralną zbudowaną w oparciu o rdzeń ARM1176 taktowany zegarem 700 Mz. Wyposażony jest w układ graficzny GPU VideoCore4 i ma wbudowane 256 lub 512 MB pamięci RAM współdzielonej z GPU. Urządzenie nie ma interfejsu do dołączenia dysku twardego ani wbudowanej pamięci NAND, więc jedyną możliwością aby zainstalować własne oprogramowanie jest zastosowanie zewnętrznej karty SD. Komputer jest wyposażony także w podwójny interfejs USB umożliwiający dołączenie dowolnych zewnętrznych urządzeń USB.

Ze względu na brak interfejsu dla dysku zewnętrznego, urządzenie dość kiepsko nadaje się jako domowy mini serwer plików. W założeniu projektantów jest to komputer edukacyjny umożliwiający naukę programowania, przeglądanie Internetu, dostęp do poczty itp., bez konieczności zakupu komputera PC. W praktyce jednak mała moc obliczeniowa procesora oraz niewielka ilość pamięci (jak na potrzeby aplikacji typu desktop) powodują, że wykorzystanie go do tego celu jest możliwe jedynie przez osoby mające wyjątkowo dużo cierpliwości. Dużo lepiej w takim zastosowaniu sprawdzi jakikolwiek tablet, który można nabyć za podobne pieniądze.

Pomimo że użycie komputerki jako zamiennika komputera PC jest bardzo wątpliwe, doskonale sprawdzi się on w naszych zastosowaniach np. w prostych układach sterowania, automatyki, wizualizacji itp., które

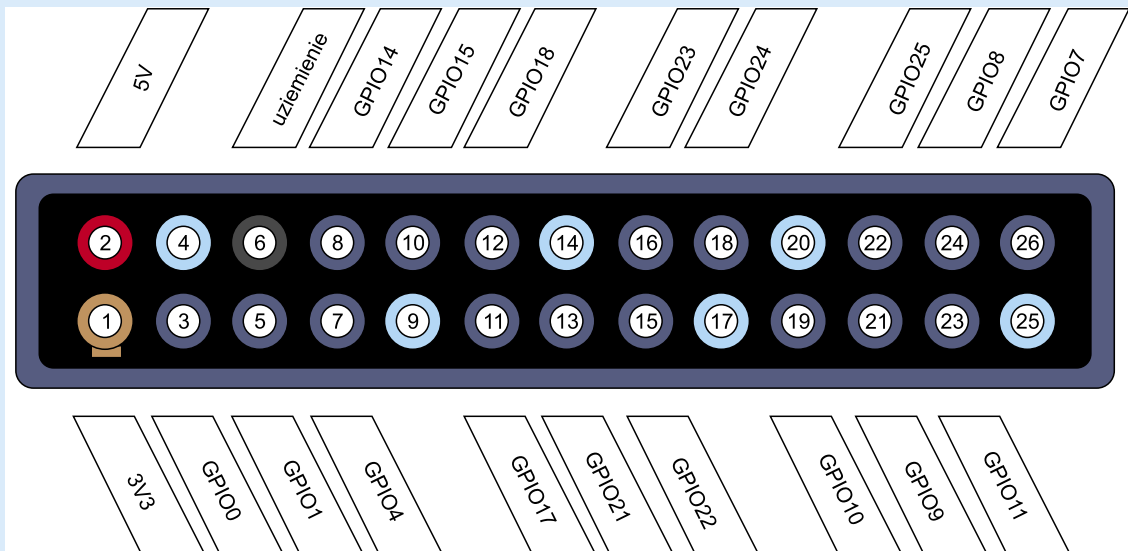
Dodatkowe materiały na CD/FTP:
ftp://ep.com.pl, user: 63048, pass: 632vmey5

nie wymagają dużej ilości pamięci czy wydajności CPU. Raspberry Pi na jednym ze złączy ma wyprowadzone kilkanaście portów GPIO umożliwiających sterowanie różnymi dołączonymi do nich układami czy urządzeniami. Dostępność kompletnego systemu Linux, umożliwia użycie open-sourcowych serwerów WWW czy języków skryptowych, dzięki czemu prostą aplikacją sterującą można przygotować w jeden wieczór.

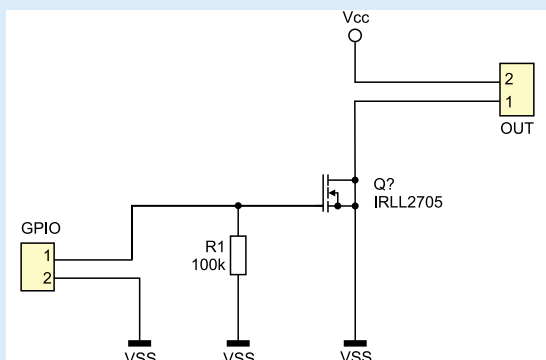
Aplikacja wyłącznika ethernetowego – schemat, opis aplikacji

Oprogramowanie „wylacznik ethernetowy” umożliwia sterowanie urządzeniami dołączonymi do portów GPIO Raspberry Pi za pośrednictwem przeglądarki WWW, umożliwiając zdalne sterowanie typu załącz/wyłącz do 17 dowolnych urządzeń. Maksymalna liczba sterowanych urządzeń jest ograniczona liczbą dostępnych portów GPIO komputerki. Wyprowadzenia GPIO zostały dołączone do złącza o rastrze 2,54 mm oznaczonego na płytce symbolem P1 (**rysunek 1**).

Oprócz portów wejścia-wyjścia na złączu dostępne są również napięcia 3.3 V (z wewnętrznego stabilizatora) i 5 V (z zasilacza zewnętrznego). Wydajność prądowa



Rysunek 1. Rozmieszczenie wyprowadzeń GPIO



Rysunek 2. Schemat pojedynczego wyprowadzenia GPIO

portów układu BCM2835 pozwala na wysterowanie co najwyżej diody LED, dlatego w wypadku konieczności sterowania układów pobierających większy prąd konieczne będzie wykonanie odpowiednich driverów. Jeśli chcemy sterować układami zasilanymi niskim napięciem stałym, wystarczający może okazać układ z tranzystorami MOSFET IRL2705 umożliwiający sterowanie odbiornikiem o napięciu zasilania max 40 V i prądzie do około 2 A. Schemat takiego układu dla pojedynczego kanału (urządzenia) pokazano na **rysunku 2**.

Tranzystor IRL2705 ma bardzo niskie napięcie załączenia bramki (Rdson) umożliwiając jego sterowanie bezpośrednio z portu mikrokontrolera. W wypadku, gdy jest konieczne dołączenie układów zasilanych np. bezpośrednio z sieci energetycznej, układ możemy uzupełnić o dodatkowy przekaźnik oraz diodę zabezpieczającą.

Od strony programowej, porty GPIO układu BCM2835 dostępne są poprzez standardowy mechanizm za pomocą interfejsu *sysfs*. Sterowanie liniami odbywa się za pomocą plików znajdujących się w katalogu */sys/class/gpio*. Szczegółowy opis tego interfejsu był podany przy okazji kursu programowania komputera BF210 w [1]. Sterowanie portami za pomocą interfejsu */sys/class/gpio* jest stosunkowo proste, jednak wymaga kilku operacji do wykonania, a także domyślnie wymagane są uprawnienia administratora. Aby ułatwić zadanie, postanowiono wykorzystać aplikację *WiringPI*, która umożli-

wia sterowanie portami bezpośrednio z linii poleceń, poprzez wywołanie prostej aplikacji o nazwie *gpio*. Plik wykonywalny *gpio* ma ustawiony bit uprawnień *SETUID*, dzięki czemu jest możliwe sterowanie portami GPIO z konta użytkownika nieuprzywilejowanego, który ma ustawione prawa do wykonania pliku *gpio*.

Dysponując gotową komendą *gpio* wystarczy wywołać ją za pośrednictwem polecenia *system()* (lub odpowiednika w innym języku), aby móc sterować portami wejścia-wyjścia. Główną wadą sterowania portami za pomocą dodatkowego programu jest jego powolne działanie wynikające z ciągłego tworzenia i zakańczania nowego procesu, jednak w wypadku sterowania za pomocą skryptu PHP, nie ma to istotnego znaczenia. Aplikacja działa jako serwis WWW pod kontrolą serwera http i w całości napisano ją w języku skryptowym PHP. Jako serwer pracuje popularny *lighttpd*, charakteryzujący się bardzo dobrą wydajnością oraz niewielkimi wymaganiami systemowymi, co czyni go idealnym rozwiązaniem w systemach embedded. W przeciwieństwie do serwera *apache* wykorzystuje on asynchroniczne operacje wejścia-wyjścia (rodzina wywołań systemowych *epoll()*) zamiast tworzenia rozwidleń procesów serwera (wywołanie systemowe *fork()*), dla każdego połączenia oddzielnie.

Aplikacja „wylącznika ethernetowego” zapewnia prostą wizualizację aktualnego stanu włączenia lub wyłączenia danej linii GPIO oraz umożliwia sterowanie portami, za pomocą dowolnej przeglądarki internetowej. Wygląd interfejsu użytkownika aplikacji w oknie przeglądarki internetowej pokazano na **rysunku 3**.

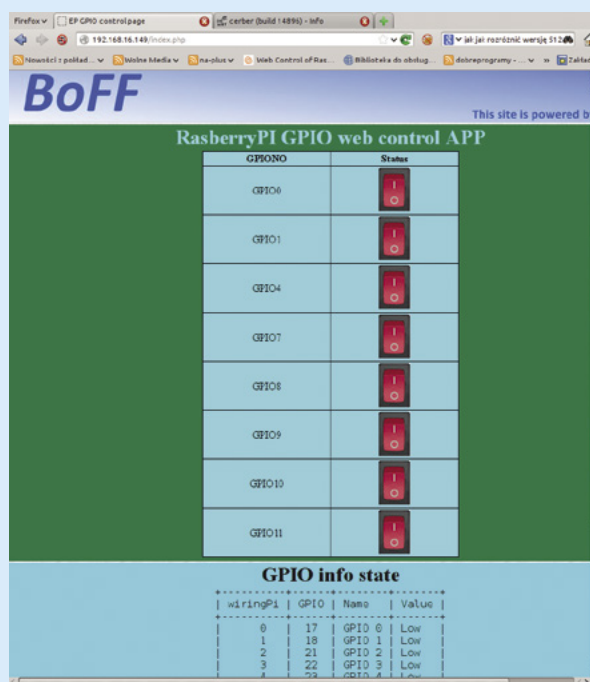
Po wpisaniu w przeglądarce nazwy hosta lub bezpośrednio adresu IP na komputerze docelowym zostanie wyświetlona strona www zawierająca opis numerów portów wejścia-wyjścia oraz przełączniki zapewniające wizualną kontrolę stanu linii portów. Sterowanie urządzeniem odbywa się poprzez kliknięcie odpowiedniego przełącznika, w wyniku czego następuje fizyczna zmiana stanu linii portu – a zatem załączenie bądź wyłączenie sterowanego przez nią urządzenia. Poniżej poziomu prze-

```

Listing 1. Zawartość pliku index.php
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">
<head>
<title>EP GPIO control page</title>
<link rel="stylesheet" type="text/css" href="style.
css" />
</head>
<body>
<div id="header">
<div id="header">

</div>
<div id="content">
<h1> RaspberryPI GPIO web control APP </h1>
<?php
include(,gpio_form.php');
$gpio_table =_array(0, 1, 4, 7, 8, 9, 10,
11);
generate_gpio_table($gpio_table); ?>
</div>
<div id="pininfo">
<h1> GPIO info state </h1>
<pre><?php print_gpio_info(); ?></pre>
</div>
<div id="footer">
&copy; <a href="http://www.boff.pl"> BoFF</a>
</div>
</body>
</html>

```



Rysunek 3. Wygląd aplikacji sterującej

łączników jest wyświetlana informacja tekstowa o stanie wszystkich dostępnych portów GPIO pochodząca wprost z wyjścia *stdout* aplikacji *gpio*. Wyświetlana informacja służy w zasadzie wyłącznie do celów diagnostycznych i może być w prosty sposób wyłączona bezpośrednio w kodzie PHP. Po wyłączeniu informacji diagnostycznej na stronie www będzie wyświetlany tylko i wyłącznie panel wyłączników, bez dolnego paska statusu.

Działanie programu

Aplikacja „wyłącznika ethernetowego” umieszczona jest w głównym katalogu serwera */srv/http* i składa się z kilkunastu plików:

- **index.php** – główny plik projektu opisujący wygląd strony internetowej.
- **style.css** – plik zawierający kaskadowy arkusz stylów który jest odpowiedzialny za definicje wyglądu strony.
- **gpio_form.php** – jest to główny plik aplikacji odpowiedzialny za renderowanie wizualizacji stanu wyłączników oraz ich obsługę.
- ***.png, *.gif** – pliki graficzne projektu.

Po wpisaniu do przeglądarki nazwy hosta lub adresu komputerka serwer *lighttpd* rozpoczyna obsługę żądania od przeglądarki, w wyniku zostanie uruchomiony plik *index.php*. (listing 1).

Plik ten w głównej części zawiera definicję strony internetowej, oraz niewielki fragment kodu PHP, rozpoczynający się od znacznika `<?php`. Główny układ strony został podzielony na 4 części za pomocą sekcji `div`: `<div id="header">` - zawierający banner `<div id="content">` zawierający tabelkę z wizualizacją stanu portów, `<div id="pinfo">` zawierający informację o stanie linii wszystkich portów, oraz `<div id="footer">` - zawierający definicję stopki. W zmiennej `$gpio_table` zdefiniowano listę konfiguracyjną portów, do których dołączone są urządzenia zewnętrzne, którą należy dostosować do własnej konfiguracji dołączonych urządzeń. Numery na liście są zgodne z numeracją portów przedstawioną na rys. 1. Za renderowanie fragmentu strony związanej ze stanem portów oraz ich sterowaniem odpowiada funkcja `generate_gpio_table()`, do której jako pierwszy argument przekazywana jest lista konfiguracyjna wykorzystanych portów. Kod funkcji umieszczony został w pliku `gpio_form.php`. W kolejnej sekcji `<div id="pinfo">` jest wywoływana funkcja `print_gpio_info()`, której wynik stanowi bezpośredni rezultat wykonania polecenia `gpio readall`. Jak już wcześniej wspomniano sekcja ta pełni

Listing 2. Kod funkcji odpowiedzialnej za obsługę portów

```
GPIO
function generate_gpio_table($gpio_array)
{
    $gvar = new shm mem();
    if( !isset($gvar[,'gpio_initialized']) )
    {
        $gvar[,'gpio_initialized'] = true;
        foreach( $gpio_array as &$g )
        {
            gpio_hw_init( $g );
        }
    }
    if(count($_POST)>1)
    {
        $val = key($_POST);
        $gpiov = explode(" ", $val)[1];
        gpio_set($gpiov, !gpio_get($gpiov) );
    }
    $method = $_SERVER[,'PHP_SELF'];
    echo "
    <form method=\"post\" action=\"\$method\">
    <table class=\"gpio\">\n
    <thead>\n
    <td> GPIO\n </td>\n
    <td> Status </td>\n
    </thead>\n";
    foreach( $gpio_array as &$id )
    {
        echo "<tr>\n";
        echo "<td> GPIO$id </td>\n";
        echo "<td>\n";
        gpio_image($id,
        gpio_get($id));
        echo "</td>\n";
        echo "<tr>\n";
    }
    echo "</table> </form>\n";
}
```

jedynie rolę informacyjną i jeśli nie chcemy wyświetlać tych informacji wystarczy, usunąć z pliku całą sekcję `<div id="pinfo"></div>`. Kod funkcji odpowiedzialnej za obsługę portów `generate_gpio_table()` przedstawiono na listingu 2.

Funkcja składa się z dwóch części: fragmentu odpowiedzialnego za sterowania portami oraz fragmentu odpowiedzialnego za wizualizowanie stanu przełączników. Skrypt rozpoczyna się od stworzenia obiektu klasy przechowującej zmienne w pamięci dzielonej przez cały okres

Listing 3. Funkcje `gpio_get` i `gpio_set`

```
//GPIO set state
function gpio_set($pin, $value)
{
    $value = (int)$value;
    $ret = shell_exec("gpio -g write $pin $value");
    if( !empty($ret) )
        throw new Exception("Unable to set GPIO
mode");
    //echo "SET $pin = $value ";
}

//GPIO get state
function gpio_get($pin)
{
    $ret = substr(shell_exec("gpio -g read $pin"),
0, 1);
    return intval($ret);
}
```

REKLAMA



Softstart do żarówek samochodowych AVT 1599

Urządzenie, które w momencie włączania oświetlenia dołącza do żarówek dodatkową, szeregową rezystancję. Ogranicza to prąd włókna do bezpiecznej wartości. Dopiero po upływie pewnego czasu, podczas którego żarnik jest wstępnie rozgrzany, następuje jego pełne zasilanie.

Wybrane parametry:

- opóźnione, pełne zasilanie żarówek samochodowych
- prąd wstępnie rozgrzewający żarniki ograniczony do 5A
- czas rozgrzewania (opóźnienia pełnego zasilania) ok. 5sek
- zasilanie: 12Vdc

www.sklep.avt.pl

uruchomienia komputera. Na początku sprawdzamy czy została zdefiniowana zmienna `gpio_initialized`. Jeśli jest ona niezdefiniowana to oznacza, że skrypt jest wywołany pierwszy raz od uruchomienia urządzenia i należy zainicjalizować wszystkie używane porty w kierunku wyjścia. Kolejną czynnością jest sprawdzenie, czy skrypt został wywołany w wyniku kliknięcia w obszarze jednego z wyłączników, co możemy zidentyfikować poprzez sprawdzenie czy skrypt został wywołany w wyniku żądania post. Jeśli otrzymaliśmy żądanie POST w super zmiennej `$_POST` otrzymujemy tablicę zawierającą informację o danych wpisanych do formularza. W naszym wypadku formularz składa się z pól typu image, które zwracają wartości w postaci tablicy asocjacyjnej zawierającej pozycję x, y kliknięcia w dany obrazek w postaci dwóch elementów: string: nazwa_x => int: pozycja_x, oraz string: nazwa_y => int: pozycja_y. Parsując klucz zawierający nazwę elementu, możemy zidentyfikować obrazek który został kliknięty. Każdy obrazek posiada w formularzu nazwę w postaci `gpio_NR`, gdzie NR określa identyfikator portu. Rezultat wykonania funkcji `explode()` zawiera pole NR które zapisane zostaje w postaci zmiennej `$gpiov`. Zmienna ta jest wykorzystywana przez funkcję `gpio_get()` i `gpio_set()` do ustawienia odpowiedniego stanu linii portu, w zależności od klikniętego obrazka. Funkcje `gpio_get` i `gpio_set` przedstawiono na **listingu 3**.

Zasada działania tych funkcji jest bardzo prosta i sprowadza się do wywołania programu `gpio` z odpowiednimi argumentami, za pomocą funkcji `shell_exec()`. Funkcja ta jako argument przyjmuje polecenie do wykonania przez powłokę, a jako wartość zwraca łańcuch tekstowy stanowiący wynik wykonania komendy. Wpisując zatem odpowiednie argumenty dla programu `gpio`, możemy kontrolować poszczególne linie wejścia-wyjścia komputera.

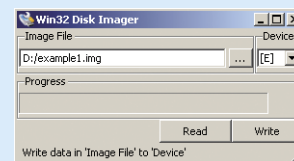
Wygląd aplikacji kolorystyka oraz formatowanie została zrealizowana za pomocą arkusza stylów CSS, a zatem edytując plik `style.css` możemy dostosować aplikację do własnych preferencji kolorystycznych, formatowania itp.

Uruchomienie przykładu – najprostszy sposób

Najprostszą metodą na uruchomienie aplikacji jest bezpośrednie wgranie na kartę SD przygotowanego obrazu dysku, zawierającego system Linux wraz przygotowanym przykładem aplikacji „przełącznika ethernetowego”. Obraz został przygotowany dla karty o pojemności minimum 4 GB i jest udostępniony w postaci pliku o nazwie `Raspberry-gpio.7z` (spakowany za pomocą programu 7ZIP). Pierwszą czynnością jest rozpakowanie z archiwum pliku obrazu `Raspberry-gpio.img`, który następnie należy wgrać na kartę SD. W systemie Windows obraz można nagrać za pomocą programu `Win32diskImager`. Aplikację można pobrać ze strony <https://wiki.kubuntu.org/Win32DiskImager>. Po zakończeniu pobierania do czytnika kart SD należy włożyć kartę, czytnik podłączyć do komputera i uruchomić program. Jako opcje `image file` należy wybrać plik `Raspberry-gpio.img`, natomiast jako `device` należy wybrać identyfikator dysku, któremu został przypisany czytnik kart SD (**rysunek 4**).

Po wybraniu odpowiednich opcji należy wcisnąć przycisk `Write`, co spowoduje rozpoczęcie zapisywania obrazu karty. Z poziomu systemu Linux, obraz można

nagrać na kartę za pomocą standardowego polecenia `dd` (`dd if=example.img of=/dev/identyfikator_urz_czytnika`). Identyfikator przypisany dla czytnika kart najłatwiej możemy



Rysunek 4. Okno programu Win32DiskImager

sprawdzić za pomocą polecenia `dmesg`. Po nagraniu karty należy umieścić ją w złączu komputera. Komputer należy podłączyć do sieci lokalnej ethernet a następnie włączyć zasilanie i odczekać chwilę, tak aby Linux uruchomił się oraz pobrał adres IP z serwera DHCP. Aby dowiedzieć się jaki adres otrzymała nasza płytka możemy sprawdzić w logach serwera DHCP. Jeśli wszystko przebiegło poprawnie po wpisaniu w przeglądarkę przydzielonego adresu powinna ukazać się strona z aplikacją „przełącznika ethernetowego”.

Uruchomienie przykładu na bazie systemu podstawowego

W poprzednim punkcie pokazaliśmy najprostszy sposób gwarantujący szybką drogę do uruchomienia przykładu. Jednak być może niektórzy czytelnicy zamiast gotowego przykładu będą woleli przejść drogę samodzielnej konfiguracji systemu oraz uruchomienia serwera www, co będzie dużo cenniejszą lekcją niż wgranie gotowego obrazu.

Standardowo dla Raspberry Pi jest dostępnych kilka gotowych dystrybucji przy czym do najważniejszych należą dystrybucje: `Debian` oraz `ArchLinux`. Obraz zaprezentowany w niniejszym artykule oparty jest o system `ArchLinux` z uwagi na to, że zdaniem autora dużo lepiej nadaje się on do systemów wbudowanych niż `Debian`, który ma dużo „ciężkich” skryptów opartych o interpreter języka Perl. Proces uruchomienia rozpoczynamy od pobrania dystrybucji `ArchLinux` z oficjalnej strony fundacji: <http://www.raspberrypi.org/downloads>. Dystrybucja jest dostępna w postaci obrazu, który należy nagrać na kartę SD w sposób opisany w poprzednim podpunkcie. Po nagraniu karty należy włożyć ją do złącza SD komputera, dołączyć kabel Ethernet oraz uruchomić komputer. Należy sprawdzić na serwerze DHCP adres, który został przydzielony, a następnie połączyć się z komputerem wykorzystując SSH. W systemach Linux wystarczy w terminalu wydać polecenie `ssh root@ip_komputera`, natomiast użytkownicy Windows mogą posłużyć się programem `PuTTY`. Po połączeniu zostaniemy zapytani o hasło – wówczas należy wpisać „root”. Po prawidłowym wpisaniu hasła powinniśmy zostać prawidłowo zalogowani do powłoki komputerka. Pierwszą czynnością jaką należy wykonać jest aktualizacja systemu za pomocą polecenia `pacman -Syu`. Gdy proces aktualizacji zakończy się należy doinstalować brakujące pakiety, które będą potrzebne do uruchomienia serwera www oraz kompilacji programu `gpio`, czego dokonujemy wydając w powłoce polecenie: `pacman -S php php-cgi lighttpd base-devel mc wget`. Podobnie jak aktualizacja, proces instalacji może trwać chwilę, w zależności od szybkości użytej karty SD.

Po zainstalowaniu oprogramowania z repozytorium należy skonfigurować serwer www oraz interpreter php. W tym celu w pliku `/etc/php/php.ini` trzeba znaleźć linię: `cgi.fix_pathinfo` i zmienić jej wartość na `cgi.fix_pa`

thinfo=1. W podobny sposób należy edytować plik: /etc/lighttpd/lighttpd.ini i po frazie „png” => „image/png” w linii mimetype.assign dopisać: „css”=>„text/css”. Następnie należy stworzyć plik /etc/lighttpd/conf.d/fastcgi.conf, włączający obsługę PHP w serwerze lighttpd za pomocą mechanizmu FAST-CGI o następującej zawartości:

```
fastcgi.server = (
    „.php” => (
        „localhost” => (
            „bin-path” => „/usr/bin/php-cgi”,
            „socket” => „/run/lighttpd/php-
fastcgi.sock”,
            „max-procs” => 4, # default value
            „bin-environment” => (
                „PHP_FCGI_CHILDREN” => „1”, #
default value
            ),
            „broken-scriptfilename” => „enable”
        )
    )
)
```

Ostatnią czynnością konfiguracyjną jest dodanie użytkownika, na uprawnieniach którego będzie pracował serwer www, oraz uruchomienie serwera co możemy dokonać wydając następującą sekwencję poleceń:

```
groupadd http
useradd http
chown -R http /var/log/lighttpd
systemctl enable lighttpd
systemctl start lighttpd
```

W wyniku wykonania ostatniego polecenia serwer www powinien zostać uruchomiony. Aby upewnić się że serwer pracuje poprawnie należy włączyć przeglądarkę

internetową a w pasku adresu wpisać adres przydzielony Raspberry, w wyniku czego w przeglądarce powinniśmy zobaczyć stronę zawierającą pustą listę plików serwera www.

Kolejną czynnością jest skompilowanie i zainstalowanie oprogramowania *WiringPI* umożliwiającego sterownie liniami I/O z wiersza polecenia za pomocą komendy *gpio*. Aby tego dokonać należy pobrać oprogramowanie bezpośrednio na komputer za pomocą polecenia: *wget http://project-downloads.drogon.net/files/wiringPi.tgz* a następnie rozpakować je za pomocą polecenia *tar xvzf wiringPi.tgz*. Po rozpakowaniu należy przejść do katalogu aplikacji (*cd wiringPI*), a następnie wydać polecenie *./build*, co spowoduje rozpoczęcie procesu kompilacji i instalacji pakietu. Po skończonej instalacji, aby upewnić się, że wszystko przebiegło poprawnie, należy wydać polecenie *gpio readall*, co spowoduje wyświetlenie w terminalu stanu wszystkich portów GPIO. Ostatnią czynnością jest instalowanie plików aplikacji „*przełącznika ethernetowego*”, w głównym katalogu serwera www. W tym celu należy pobrać ze strony *www.elektroniki.praktycznej* przykład bezpośrednio na kartę Raspberry (plik jest dostępny na serwerze ftp – login i hasło podano w ramce na początku artykułu), a następnie rozpakować go do katalogu głównym serwera za pomocą polecenia *tar xvzf Raspberry-gpio-php.tar.gz -C /srv/http*. Po rozpakowaniu plików i wpisaniu w pasku adresu przeglądarki adresu komputerka powinniśmy ujrzeć stronę „*wyłącznika ethernetowego*”, co świadczy o poprawnym działaniu programu.

Lucjan Bryndza, EP
lucck@boff.pl

REKLAMA

Nowy konkurs Microchip

Firma Microchip organizuje konkurs dla czytelników Elektroniki Praktycznej, w której mogą oni wygrać zestaw składający się z płytki z czujnikiem „Thermal/Linear Intelligent Sensor PICtail Plus Daughter Board” oraz płytki deweloperskiej Explorer 16 Development Board.

Oferowany zestaw zawiera obwody umożliwiające kondycjonowanie sygnałów pochodzących z czujników temperatury i innych liniowych, tak by przetwarzać je za pomocą mikrokontrolera sygnałowego serii dsPIC lub mikrokontrolera PIC24H. Alternatywnie, płytka z obwodami kondycjonującymi może współpracować z 16-bitowym 28-pinowym zestawem startowym. Umożliwia testowanie projektów, w których implementowane jest przetwarzanie sygnałów analogowych na cyfrowe oraz zaawansowana obsługa sensorów.

Ponadto płytka deweloperska Explorer 16 pozwala na testowanie rozwiązań z użyciem nowych mikrokontrolerów Microchip serii PIC24, mikrokontrolerów sygnałowych dsPIC33 i najnowszych, 32bitowych układów z serii PIC32MX. W zestawie znajdują się dwa wymienne moduły: jeden z układem PIC24FJ128GA010, a drugi z dsPIC33FJ256GP710DSC.

Aby wziąć udział w konkursie wystarczy zarejestrować się na stronie internetowej pod adresem: <http://www.microchip-comps.com/ep-pictail>

