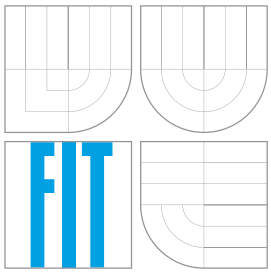
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NÁZEV PRÁCE

THESIS TITLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JORGE CUADRADO SAEZ

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Dr. Ing. DUŠAN KOLÁŘ, Ph.D.

BRNO 2016

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém jazyce.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

Sem budou zapsána jednotlivá klíčová slova v českém jazyce, oddělená čárkami.

Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

Citace

Jorge Cuadrado Saez: Název práce, bakalářská práce, Brno, FIT VUT v Brně, 2016

Název práce

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jorge Cuadrado Saez
9. května 2016

Poděkování

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant, apod.).

Obsah

Kapitola 1

Introduction

Kapitola 2

Aim of the project

Kapitola 3

Metodology

3.1 Tools

3.1.1 Git

3.1.2 Sublime Text

3.1.3 Deamons

3.1.4 File configs

3.2 TDD

Kapitola 4

The system: Hardware

4.1 The Raspberry Pi

4.1.1 History

4.1.2 Social Importance

4.1.3 Hardware

4.1.4 Power save works

4.2 The Module

4.2.1 Installation

4.2.2 PCF8563

Why a RTC?

How it works (Hardware)

How use it

Problems during the development

4.2.3 DS18B20

How it works (Hardware)

How use it

Kapitola 5

The system: Software

5.1 The Operative System

There are so many option available in the market. Every one has his owns pros and cons but at the end I decided bet for compatibility and i chosed de official debian flavour designed specially for Raspberry Pi.

5.1.1 Why raspbian lite?

There are so many reasons why choose Raspbian. But the most important is because is a distribution made by Raspberry.org specially for the Rasperry Pi and it means that all the comunity behind Raspberry Pi is more likely to use this distribution. That is so usefull in order to find solutions, tutorials, documentation and libraries specially designed to use in this distribution with a Raspberry Pi. Also i choosde the Lite version. This version is special because do not have the X server and any other graphical user interface support. This is a great fact because is lighter than the normal version.

5.1.2 Consecuences during the develop

It was a really great decision because all the oficial documentation concerning to Raspberry Pi is thought using Raspbian.

Wifi setup

For the wifi configuration i used a wpa_supplicant command line utility in order to connect the device to Eduroam Wifi network.

Components disable

One of the important points in the project is the possibility of move the device everywhere only with a battery supliying power. This means that we have a important necesity of save all the power possible and develop a efficient system. One of the most important actions was the disable of all the hardware components that are not needed in the device. The disabled components are:

- HDMI socket
- USB hub

- Audio socket
- Ethernet socke

5.1.3 Other alternatives

Between the alternatives to Raspbian there is the oficial Debian distribution, Slackware, Archlinux and other special designed distributions for Rasperry Pi.

5.2 The languaje

One of the most important decisions was about what languaje is most desirable and fits better this project to finish successfully. There was an important mather to choose one languaje which permit a fast development and works eficiently in the Raspberry Pi.

5.2.1 Why Python?

Finally one of the best alternatives was Python. The main reason is that it is already included in the operative system choosed, Raspbian. Thanks to this we avoid occupy more space in memmory with another languaje libraries. Furthermore Python provides a great support and a great amount of libraries to interact with the operative system and the Raspberry Pi hardware. For this reasons Python grants a fast development.

5.2.2 Consecuences during the develop

Because my inexperiency using Python it was difficult to get use with the use of Python. Despite of that Python solved so many problems simple and efficiently using the default python libraries like the cron library, the daemon library and the cryptographic library.

5.2.3 Other alternatives

Other great alternatives were C and JavaScript. The first one provides a low level and so efficient aproximation to the problem and it is also by default in the operative system, despite of this the develop using C requires more time and i am not familiarizated with it. In the other side JavaScript would be a great option using also NodeJS. Also JavaScript provides great libraries and it is so comfortable for the developer to develop backend and frontend in the same languaje but it is not focus in interact with the operative system, there is not information about performance and power consumption in Raspberry Pi and it is not included in the operative system by default.

5.3 The web server

This decision was conditioned by the code languaje choosen. The problem was to decide between all the posibilities based in python asuming also that we need a so eficient and light weight server. The web server have to provide static web pages and the web application needs to have a public part with the data and some graphics and also a private part where it is possible to change de device configuration like power saving mode, scp address where store the data and the frequency with the sensor take the samples.

5.3.1 Why http.server

At the end Http.server was the choosen option. It is a really simple web server included in the standart Python library. It has only the really necesary to works and do not provide anything outside the basic functionalities of a web server. The main feature of this web server that match perfectly this project is that is really light weight, other light weight servers are based on Http.server also. In this project we provide a static and so simple web page so we do not need anything more.

5.3.2 Consecuences during the develop

The http.server demostrated to be so symple to use. It did not give any problems and the develop of the basic functionallity was fast. One of the challenges using this server was develop a session for ensure the post and get requests are only from logged persons kepping save the system for anonymous users.

5.3.3 Other alternatives

Python offer a great ammount of alternatives for make a simple and light weight server. So many of this are also based on Http.server. This is the main reason because i prefered to choose http.server.

5.3.4 Web server parts

The web server is composed by multiple subsystems. In one hand it has a cookie storege subsystem to avoid non autoriced request. In the other a subsystem which remake the html files using templates. This subsystem regenerate the html updating the data shown to the user and making a fake effect of dynamic web page. Also i made a subsystem which change the web static links in prevision of multiple installations permiting also a fast deployment and allowing web server to update the html files if detect an ip change.

Kapitola 6

The system: All together

6.1 Subsystem: The temperature

The main objetive of the system is measure the temperature. This was implemented as a subsystem using a deamon. For this purpose i used the YapDi library, developed by Kasun Herath. As explained in XXXX, the used sensor is the digital termometer DS18B20.

The Subsystem store the samples in plain text data files stored in the data folder of the application. Every sample is in a different line and is composed by the measure, in celsius degrees, and the time stamp.

This subsystem use also a frequency configuration file which allows the daemon to update the frequency while it is running and a file size configuration that determinated the number of samples in each new file created.

6.1.1 How it works

Due to the importance of this system it is always running. Every time the system is boot this daemon is launch after checking if the configuration file exist, else the configuration file is created with a default frequency value of 10 minutes.

When the daemon starts running it get the frequency from the configuration file. Then gets the last file created, if it is completed, that means the file has the maximun number of samples permited by the user and write in a configuration file, the daemon will create a new data file using the actual timestamp as part of the name. Else the daemon will continue using this not full data file for store the samples.

6.2 Subsystem: The scp

One of the requierements of the system is the necessity of send periodically the data to a machine. The address of the machine must be configurable by the user in the web application. For this purpose i implemented a daemon which runs in background sending the data periodically. The daemon was implemented en Python and use the YapDi library, developed by Kasun Herath, which simplifies the configuration of the daemons and permit to program it in python and also de Paramiko library wich simplifies the SCP conexion development and permit to inicialiced the SCP conection without write directly the password of the user in the shell.

6.2.1 How it works

This subsystem only starts work when the configuration is set up in the web page application. For security reason all the SCP conexion data is not stored in the device and also is encrypted in memory. Despite this fact if the system is reboot this configurations will get lost. This could be undesirable by the user if he wants to use the power saving mode 3 which hibernate the device in a schedule set up by the user. For this reason the web application permit the user to allow the system to store the scp conexion data. In this case the device will check if the scp configuration is stored in the configurations folder and in this case the system will start the daemon after reboot.

The subsystem wrap an SSH conexion to create an SCP conexion. If the conexion can not be created because of the configuration data is not valid the daemon will kill his own proccess.

Once the conection is created the daemon will check which files was not send to the actual target machine using a configuration file where the daemon write all the sended files. The daemon will send also the not completed data files but it will update the data file in the target machine once it's completed.

When the file is send the daemon will mark the file as sended in a configuration file and then it will create a backup of the data file in another directory.

The daemon also has a reset function which allows the server or and external script to erase the configuration files which store the sended files permiting send another time all the data files or erase the file when the target machine is changed.

6.3 Subsystem: The web server

The web server was designed with the intention to show the data in a graphic and more understandable way. And furthermore allows the user to set up the configurations for the rest of the subsystems and also permit change the power mode of the device and set up the schedules for the power saving modes.

The web server is made using the standart python library. I used the Http.server library which provides a light weight web server with the basic functionallity for a static web page.

6.3.1 How it works

The web server is wake up every time the device is boot in power saving mode 0 or 1. If it is in power saving mode 2 it will be wake up only in the right moments allows for the user.

The web server allow to HTTPS conexións wrapping the conexion with SSL encriptation. Also I implemented a session system which permit keep the security in the request made to the server only allowing logged users to change the configuration.

This is a static web server which means that only can serve static web pages. This is desirable in our project because we do not need a complex web page and also the resources consumed by the server are smaller than in a more complex web server.

The problem of a static web server it the static content and this is not a problem for the configuration section but it was problematic in the public part of the web application because we need to show the last samples and update the content periodically.

For this reason the web server use a template where it adds the new samples on every get request of the homepage. At first look that seems not so efficient but in a web page

like that with a few rate of visits allow as to grant dynamic content without use a dynamic server which definitelly is heavier and spend more resources than our static server.

Moreover the server will detect if the Ip has changed since the last boot and in this case it will remake the webpages links and the device host name.

6.3.2 Endpoints

The web server only allow GET and POST request. The GET endpoints are:

- /
- /configuration
- /login

And the POST endpoints are:

- /login
- /configuration
- /scp
- /pmnormal
- /pm1
- /pm2_one
- /pm2_eachone
- /pm2_multiple
- /pm3_one
- /pm3_eachone
- /pm3_multiple

The configuration POST endpoint allow the user to setup the server number of samples shown in the home page, the frequency which the samples are taken and the frequency which the scp send the new data files.

The SCP post endpoint permit the user to set up the scp configuration.

The other post endpoints are designed for change between the power saving modes, the 2 and 3 power saving mode are dividen in three endpoints, one for each input type to set up the this power mode schedules.

6.3.3 Data validation

All the data recived from the user is validated before change the configurations. For this purpose i used regular expresion which only will match with the correct type of data. After check the data type is correct there a second validation which checks if the data is between the ranges permited and if the configurations are consistent and not contradictory.

6.3.4 Session implementation

For allow secure get and post request it is needed to implement a session. The problem of use a so simple static web server like http.server it that there is not also a default library that allows to keep sessions. For this reason I implemented a session using cookies.

Every time a user is logged the server will generate a cookie which will be sent in the asnwer headers of the http request.

The value of the cookie is a hash of the current timestamp.

This cookie is stored in the session subsystem and when a request to the server is made the server will check if the cookie was created in the server and if the cookie still alive.

6.4 Web application

The objetive of the web application is to show the last temperature samples and allow the configuration of the device parameters from any location at any time.

The web page is composed by two main sections. There is a public section made for show visitants the last samples and the graphics which simplify the compenssion of the data. And there is also a private section where the users can change the device configurations and control the device behaviour.

One of the most important things once the device has publish its address is protect them from intruders and any other atack like man in the midle, service denegation etc.

For this reason was necesary to add a login page in order to protect the private section. This section request users to introduce their credentials before grants them access to the private section and also permit the server to create a session with a cookie for also prevent request to server which are not send from the web application.

The web application has been designed to be simple and light with the aim of be the most lightweight possible in order to save baterry sending the less amount of files possible for answer the requests to the server.

6.4.1 Pages

Home

In the home page the server shows the last samples in a table and a graphich with the samples shown.

Login

The login page allows the user to get access to the configuration page.

Configuration

The configuration page allows to change the device behaviour. The configurations that user can change are the scp target address, the frequency of the samples, the number of samples the web page show, the size of the data files stored in the device and the power saving mode. In this last configuration the user can change between 4 power saving modes, in the mode 2 and mode 3 the user also have 3 ways of input the schedule of the power saving modes, one where the interval of time is the same for all days, other where the user can input a diferent interval for every day of the week and also a last one where the user can input multiple intervals for each day.

The configuration page shows always what power saving mode is actualy set up in the device configuration. Because the web server only serve static content there is 4 html files for the configuration stored in the device. Each one of these files change the actual selected mode. This is not a clever solution for big and complex web pages but in our case there is only four options which only change the class of one element whitouth any other collateral consecuence and thanks to this solution we can save battery of the device avoiding add programs which precompile the html files or another complex solutions.

Error Page

Another important matter was give feedback about errors to user when the data input was not correct or when a internal problem happends. This is a main topic in a user interface and a static web server is so limited for give feedback to user.

For this problem is not possible to use the solution implemented in the configuration secction for show the current power saving mode because the number of possible states is too big and during implementation the number of errors handled will grow for sure.

For this problem the web application redirect the user to a generic web page whith a text explaining the problem.

6.4.2 Scripts

With the aim of save battery sending so many data the web application was dessigned having in mind make a simple web page trying to use all the power of html5 functions and decreasing the number of scripts needed.

For these reason there is only two scripts used in the web application.

configuration.js

The configuration script is only used in the configuration section. This script has two main functions. In one hand the script create a pop up asking confirmation to the user if the values introduced are between the allowed ranges but not so desirable in order to save battery or if the values are in conflict with another configurations. In the other hand the script create the accordions with the three ways for input the power saving mode 2 and 3 schedules.

chart.js

This script get the values of the samples directly from the html table of the html file and using that data the script generate the graphic in the home page.

Also the script control the size of the window resizing the graphic for keep the responsive dessing.

6.4.3 Styles

Althought the web application is concerned in the functionallity the dessign was made keeping in mind the user xperience. Because there is strict requirement of energetic eficiency the dessign was made also avoiding send more data than necesary in every server request answer. For that reason there is no big dessign libraries stored in the device and the dessign strictly respect the KISS principle.

The style is provided only for a simple css style sheet for all the web application which control the position of the elements and provide some media queries which implement a responsive design.

Responsive

Nowadays the way to use internet is so different from four or five years ago. The smarth phones changed completly the way to use web pages and now is mandatory to dessign not only a desktop web page but also a mobile dessign prepared for small screens.

For this objetive there are media queries implemented in the style sheet which permit adapt the content to smaller screens changing what is show and where mantaining a understandable and clean dessign.

The style sheet

The stylesheet is stored in the css folder in the project folder. In the style sheet there is coded a generic dessign for all the web application and specific dessigns for each web page.

6.4.4 Libraries

The web application has two dependencies with external libraries. One of the libraries JQuery and the other one is Google charts tool.

The advantaje of this two libraries is that both are stored in remote servers and the server doesn't have to send it to clients.

6.5 Directories

The project folder is composed by the following directories.

- Logs
- Data
- Config
- CSS
- HTML
- JS
- Lib
- Scripts

6.5.1 Logs

Logs generated by the subsystems.

6.5.2 Data

Data files stored by the temperature subsystem and backups.

6.5.3 Config

Configuration files of the system. There are this files:

- crontable: file with the actual setted crontable.
- file_size: size of the current and next data file.
- frequency_scp: frequency of the scp transmisions, in hours.
- frequency_temp: time between samples, in minutes.
- login: hash of the login and password credentials.
- powermode: actual power mode, number between 0 and 4.
- samples_show: number of samples show by the web page.
- scp: address and credentials of the target machine.
- sendedFiles: files already sended to target scp machine.

6.5.4 CSS

Style sheets of the web application.

6.5.5 HTML

In this folder are stored all the html files. These files are generated by a python script using the html templates stored in the template folder inside this same directory.

6.5.6 js

Scripts used in the web application pages.

6.5.7 lib

Python auxiliar modules used in all the subsystems. In this folder are stored utility.py, web_maker.py, CookieStorage.py and transfer.py.

6.5.8 scripts

This folder store the power saving mode scripts needed for change the device configurations. The scripts should be moved to /bin for allow the server to use it.

6.6 Models

6.7 Power saving modes

The system implements four power saving modes which change the configuration of the device enabling or disabling some configuration, hardware and programs lowering the device functionalities but decreasing the power consumption in order to increase the battery life time of the device.

6.7.1 Why?

The project is designed for stay working using a battery as power supply during the most time possible. The big problem of this kind of system is the battery life time because are thought for be working during long periods. Normally this kind of devices works in devices with few computational power but with big battery life time like Arduino or another chips similar to Arduino. But with the arrival of the Internet of the things (IoT) is comming the necesity an also the posibillity of more powerfull devices like Raspberry Pi.

Is truth that Raspberry Pi is a device which consume so few power, and a big part of the Raspberry Pi community think that try to decrease the power consumption is a non sense. Contrary to this point of view this thesis try to optimice the power consumption of the device as much as possible in order to increase the battery life time of the device trying in the same proccess to test the raspberry pi as an embedded system.

This is not a easy task because even if Raspberry Pi has a lower consumption for a computer continue consuming so mucho if we compare this device with other like the before mentioned Arduino.

Make Raspberry Pi consume the same or less than systems like Arduino is of course impossible and its not the aim of this thesis try to achieve that. The two type of systems are so diferent and the objetive of both systems is not the same. Raspberry pi is for sure more powerfull and is a multipurpose device which can work as a multimedia station or like a embedded system like this but the chips like Arduino are dessigned for be only embedded systems and for sure are better option thinking only in power consumption. The problem of this chips is the lack of computational power which does not permit them to create a web server or work above a complex operative system like Raspbian. This fact is not trivial since a complete operative system grants lots of services, security systems and also make the develop of applications significatively faster.

With this facts in mind the next power saving modes were developed.

6.7.2 Description

All the power modes were implemented using bash scripts which can be use from any directory of the device if they are placed in the /bin folder of the device user.

This scripts also call another python scripts if it is necesary and principally enable or dissable the power of the hardware components which are not needed in the device.

The power mode scripts not only desactivate the ethernet interface saving computational resources, also disable the bus which gives power to the component even when the interface is down saving in this way the power destinated to that interface.

There are hardware components which are disabled always because they are not usefull for this system like the HDMI port or the ethernet port. But for security reason there is a power saving mode which enable all the disabled components if necessary.

6.7.3 When

Every power mode is dessigned to be work in some specific part of the life cycle of the device. The default power saving mode which will be active if there is not a configuration about that is the first power saving mode.

The zero power saving mode is only thought for special moments where the hdmi port or the ethernet wire could be necesary but is not thought for the normal life cycle of the device.

The other two power saving modes are dessigned as hard saving power modes which decreasy the device functionallity significantly but increasy in the same weay the battery life time. These modes will disable the internet conection disabling also the web server in order to save the most battery possible. These modes are designed to work following a schedule set up by the user wich as a rule can not be full time in order to let the user change to other modes and recover the control of the device at least a couple of hours every day.

6.7.4 PM0

The zero power mode is thought as a special mode which will enable another time all the hardware functionalities. This mode will increase the battery consumption to the maximum and for these reason is not dessigned to be working in the device.

This mode is thought for maintenance of the device when it is necesary to conect the device to a screen and the HDMI port is needed to be enable.

It is not recomendable to use this mode in the normal life cycle of the device.

6.7.5 PM1

The first power consumption mode is the default power saving mode in the device, it will be set if there is not another configuration and this mode will be automatically enable when the device goes out of the second or third power saving mode.

This power saving mode decrease the frequency of the CPU in order to decrease the power consumption. The truth is that the power consumption only decrease a bit but the collateral fact is that the heat generated for the CPU decrease also.

Even if the power consumption decrease only a few with the lower CPU frequency is better than nothing and the heat decrease is useful in order to increase the life of the device and in order to avoid affect the temperature sensor samples. Of course this power saving mode also disable the HDMI and the ethernet port. It also would be so useful disable the USB hub letting active only the USB port used by the Wifi module but the way is dessign the hardware does not allow to make this because the USB ports are not independent ones of others, they are all conected to the same master usb root port. This dessign was made thinking in mobile systems and it is simpler than a normal desktop computer USB hub and it does not allow to only let one of the usb ports enabled. It is possible to dissable the data transmision of the ports but the device will continue giving them energy so it is not useful to disable them. How i will explain in possible improvements dissable the data transfer would be useful as a security functionallity.

6.7.6 PM2

This power saving mode was dessigned thinking in extend the battery life lowering the device functionallities even if the conexion with the device is lost in order of make possible get a device which can be working for longer periods.

The main fact in this power saving mode is that the wifi module is disabled and the user won't be capable of connect with the device for change the configurations. If this power saving mode would be working withouth stop it wouldn't be useful because the user would need in some moment to be physically in front of the device, connect it to a screen and change the power mode manually.

In order to avoid the necesity of g find the device for change the configuration this mode is not allow by the system to be working 24 hours every day. The user must set up a schedule for this power saving mode in the configuration page of the web application which using a crontab will manage the power saving mode automatically. The shedule will restrict the maximum time which the device can be in the second power saving mode, 23 hours is the maximum time for a day, permiting user to change the configuration at least one hour every day.

This power saving mode will disable the same things disabled in the first power saving mode and also will disable the USB hub, the Wifi interface and the web server.

As an exception, if the scp subsystem is running, it will detect if the second power saving mode is enabled, in this case the subsystem will enable the wifi interface in order to send the data to the target machine address and then it will disable the wifi interface another time.

6.7.7 PM3

In a harware system like raspberry pi is not possible to let running only the temperature subsystem in order to extend the battery life time if we don't use a special operative system which wouldn't allow us to run a web server and would be difficult and hard to develop or if we don't use special hardware like an Arduino connected to the Raspberry pi for example.

With the objetive to increse the battery life time extreamly the third power saving mode will shutdown the device setting an alarm in an RTC module which will wake up the device just before take a temperature sample.

When the device wake up first take the sample and then check if in the schedule set up by the user is time to change to another power saving mode or if the system have to shut down itself another time until next sample time. If it have to shutdown another time, before this, the system set up a new alarm in the RTC which only can store one alarm at the same time.

Like the second power saving mode, the third power saving mode follows a schedule set up by the user with the same rules and reasons as the second power saving mode.

In this power saving mode the SCP subsystem will work only if the user mark the checkbox of the configuration web application acepting store the scp configuration data in the device, else the data will be lost on every shutdown the device will do. Also if the scp configuration is stored in the device the scp data transfer frequency will not be followed and the subsystem will send the new data every time the device will wake up in order to do not lose data.

6.7.8 Another not implemented actions for decrease power consumption

6.8 Life cicle

6.8.1 Boot script

6.8.2 PM change

6.9 Complementary work

6.9.1 Setup script

6.9.2 SCP remote control

6.9.3 User manual

6.9.4 Installation manual

Kapitola 7

Real world uses

Kapitola 8

Conclusions

Přílohy

Seznam příloh