Digital Control

LABORATORY REPORT

Raspberri pi and Nodemcu

Amirkabir University of Technology
(Tehran Polytechnic)

Amirhossein Sheikholeslami
Mahboobeh Shakeri
Ahmadreza Nazari
Niloofar Raeesolsadati
Amin Razzaghi

Contents

1. Introduction 1

2. Arduino 1

   - Nodemcu 2

   - Turning an led on by using NodeMCU 3

3. Raspberri pi 4

   - Installation.. 4

   - Connecting to raspberri pi and turning on LED 6

Introduction

At this course are aim is to control different plant's by using microcontroller's;

A microcontroller (MCU for microcontroller unit, or UC for -controller) is a small computer on a single integrated circuit. In modern terminology, it is similar to, but less sophisticated than, a system on a chip (SoC); an SoC may include a microcontroller as one of its components. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

1.Arduino

One of these microcontroller is Arduino that we used in this course.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

NODEMCU

then we got familiar with NodeMCU: NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS.

Turning an led on or off by using NodeMCU

At this point of examination we should connect the NodeMCU and all devices that are going to be used get connected to the same network.so we got connected all devices to one modem.after that we got one of the pin's as an output then after setting a password and SSID one of the wifi's we creat a local server from NodeMCU.the we got in a loop of the code and by using server.available command we got clear that the server is available and if it is not then the cod is closed.

For clearance of the cod we define a request and print it, we got this from client command.in order that by using code(...)all the string that user's sent till the end of controlling command that is defined by åre assumed as request, but if , after ẘas any other request and the data was in the client by using client.flash command we delet that data.after that with using indexof command of code(...)we searched for request that if it is on or off. So if it was on we let the output pin to be on and if it was off the output pin is going to be off.at this point we were done with ARDUINO code.then we had to creat a HTML code that creat a control window for the request that client is going to send.
For implementation of HTML code we should put all the code in a string and append them.the core of the desingning should be to creat to button for turning the LED off and on that any client with any device's can get accesses to this page.but the point is that the address of this page is the IP of the NodeMCU that we can find it from the wiffi control panel or by using the wifi.localIP command in ARDUINO.
The aim of this examination is that with using any device or different client from NodeMCU that here is local server turning on or off a lamp from the web.

2.Raspberri Pi

Another microcontroller is raspberry pi that we used is:

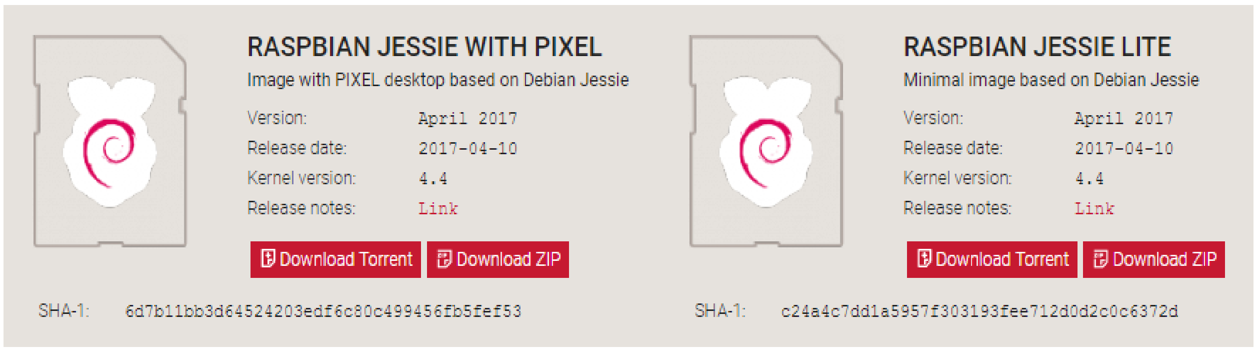
The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) and cases. However, some accessories have been included in several official and unofficial bundles. Raspberry pi is one of the microcontroller that it's operating system is linux. Linux is a family of open source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991 by Linus Torvalds. Linux is typically packaged in a Linux distribution (or distro for short). Distributions include the Linux kernel and supporting system software and libraries, many of which are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but the Free Software Foundation uses the name GNU/Linux to emphasize the importance of GNU software, causing some controversy. Popular Linux distributions include Debian, Fedora, and Ubuntu. raspian is personalized distribution of linux for raspberry pi .

Installation

As we said for controlling plant's we use microcontroller so at this time we used raspberry pi.
for using raspberry pi at first we must install raspian on it.to install raspian we should follow these step's:

Step 1: Download Raspbian

I promised to show you how to install Raspbian on the Raspberry Pi, so it's about time that we got started! First things first: hop onto your computer (Mac and PC are both fine) and download the Raspbian disc image. You can find the latest version of Raspbian on the Raspberry Pi Foundation's website here. Give yourself some time for this, especially if you plan to use the traditional download option rather than the torrent. It can easily take a half hour or more to download.



RASPBIAN JESSIE WITH PIXEL
Image with PIXEL desktop based on Debian Jessie

Version:	April 2017
Release date:	2017-04-10
Kernel version:	4.4
Release notes:	Link

SHA-1: 6d7b11bb3d64524203edf6c80c499456fb5fef53

RASPBIAN JESSIE LITE
Minimal image based on Debian Jessie

Version:	April 2017
Release date:	2017-04-10
Kernel version:	4.4
Release notes:	Link

SHA-1: c24a4c7dd1a5957f303193fee712d0d2c0c6372d

Figure 1: downloading raspbian

Step 2: Unzip the file

The Raspbian disc image is compressed, so you'll need to unzip it. The file uses the ZIP64 format, so depending on how current your built-in utilities are, you need to use certain programs to unzip it. If you have any trouble, try these programs recommended by the Raspberry Pi Foundation:

- Windows users, you'll want 7-Zip.
- Mac users, The Unarchiver is your best bet.
- users will use the appropriately named Unzip.

Step 3: Write the disc image to your microSD card

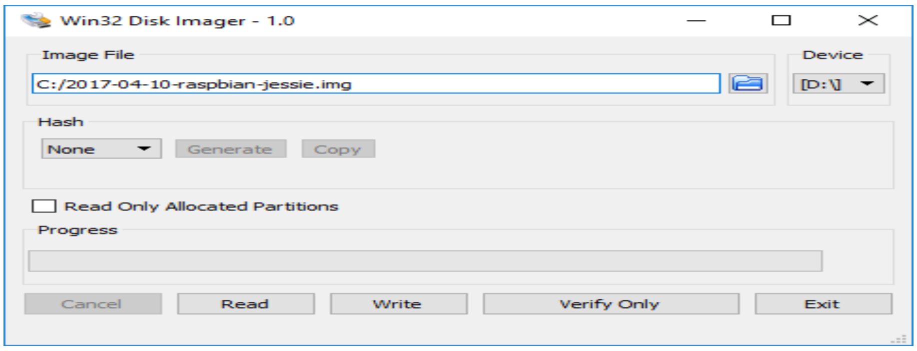


Image File: C:/2017-04-10-raspbian-jessie.img
Device: [D:\]
Hash: None

Figure 2: boot imagefile on SD-card

Next, pop your microSD card into your computer and write the disc image to it.

You'll need a specific program to do this:

- Windows users, your answer is Win32 Disk Imager.
- Mac users, you can use the disk utility that's already on your machine.
- Linux people, Etcher – which also works on Mac and Windows – is what the Raspberry Pi Foundation recommends.

The process of actually writing the image will be slightly different across these programs, but it's pretty self-explanatory no matter what you're using. Each of these programs will have you select the destination (make sure you've picked your microSD card!) and the disc image (the unzipped Raspbian file). Choose, double-check, and then hit the button to write.

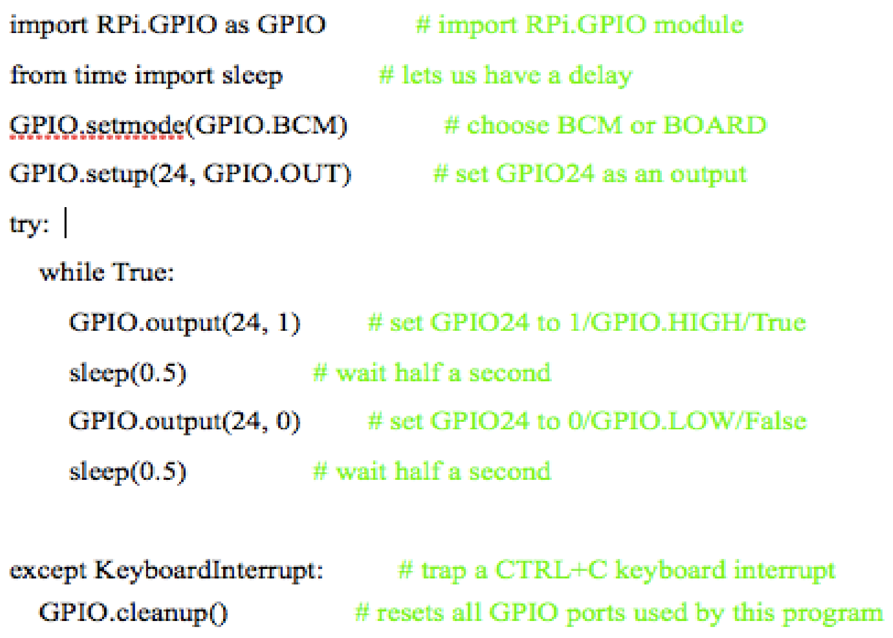
Connecting to raspberri pi

Then by using SSH we get connected to raspberry pi, for doing this we made a folder named WPAsupplicant that we put the SSID and the password of the modem that both our system and raspberry pi are connected to.

Turning On an LED

After that we put the SD card in and we connected the charger to turn the raspberry pi on.then we found the IP of raspberry pi from the modem panel and by using putty in windows or terminal in linux we connected to the raspberry pi.

By using VNC we can see graphical of raspberry pi viritually.
After wanted to turn an LED on and off so we creat a python code:



```
import RPi.GPIO as GPIO          # import RPi.GPIO module
from time import sleep           # lets us have a delay
GPIO.setmode(GPIO.BCM)           # choose BCM or BOARD
GPIO.setup(24, GPIO.OUT)         # set GPIO24 as an output
try:
    while True:
        GPIO.output(24, 1)       # set GPIO24 to 1/GPIO.HIGH/True
        sleep(0.5)               # wait half a second
        GPIO.output(24, 0)       # set GPIO24 to 0/GPIO.LOW/False
        sleep(0.5)               # wait half a second

except KeyboardInterrupt:        # trap a CTRL+C keyboard interrupt
    GPIO.cleanup()               # resets all GPIO ports used by this program
```

Figure 3: Python code