



**Коммерциялық емес  
акционерлік  
қоғам**

**АЛМАТЫ  
ЭНЕРГЕТИКА ЖӘНЕ  
БАЙЛАНЫС  
УНИВЕРСИТЕТІ**

**«Электроника және  
робототехника»**

## **АСПАП ЖАСАУДАҒЫ КОМПЬЮТЕРЛІК ТЕХНОЛОГИЯЛАР**

**5B071600 – Аспап жасау мамандығының студенттеріне арналған  
зертханалық жұмыс бойынша әдістемелік нұсқаулар**

Алматы 2019

Құрастырушылар: Байкенов Б.С., Аязбай А.Е. Аспап жасаудағы компьютерлік технологиялар. 5B071600 – «Аспап жасау» мамандығының студенттеріне арналған зертханалық жұмыс бойынша әдістемелік нұсқаулар. – Алматы: АУЭС, 2019. – 42 бет.

Әдістемелік нұсқаулар IP-желілер бойынша қашықтан басқару және бақылаудың сандық жүйелерін құру үшін негізгі элемент болып табылатын Raspberry PI-3 микрокомпьютерін зерттеуге арналған.

Raspbian операциялық жүйесін, пакеттер мен қосымшаларды, SSH желілік хаттамасын, PHP веб-серверін, HTML серверін, MySQL серверін таңдау және орнату қарастырылды.

Сонымен қатар, смартфонның WEB-беті мен сенсорлық экраны арқылы токтың электр жетегінің IP-желілері бойынша қашықтықтан басқару және температураны бақылау жүйесін құрудың екі технологиясы келтірілген.

Әдістемелік нұсқаулар дәрістік материалдарды бекіту мақсатында құрастырылған және 5B071600 – «Аспап жасау» мамандығының студенттеріне арналған.

Сурет - 23, кітапнама.-7

Пікір беруші: доцент Курпенев Б.К.

«Алматы энергетика және байланыс университеті» коммерциялық емес акционерлік қоғамының 2019 жылға арналған басылым жоспары бойынша басылып шығарылады.

## **Кіріспе**

Raspberry Pi-бұл шағын, несие карта өлшемді, 900 МГц тактілік жиілігі және 1 Гбайт жедел жады модулі бар компьютер. Қанағаттанарлық қуатты, төмен энергия тұтынуды және аз өзіндік құнды ескере отырып, Raspberry Pi жеке шағын серверді жасау үшін пайдалануға болады.

Raspberry Pi басты ерекшеліктерінің бірі-платада GPIO (General Purpose Input/Output, жалпы мақсаттағы енгізу/шығару интерфейсі) енгізу/шығару аппараттық порттарының болуы, бұл оны қашықтықтан басқару және бақылау жүйелерінде, робототехникалық жобалар мен «ақылды үй» құрылғыларында пайдалану перспективаларын ашады.

«Аспап жасаудағы компьютерлік технологиялар» пәні бойынша зертханалық практикум 10 жұмыстан тұрады және шартты түрде екі бөлікке бөлінген – операциялық жүйені, қосымшаларды, серверлерді орнату және IP-желілер бойынша объектілерді басқару есептерін орындау үшін микрокомпьютерді бағдарламалау.

Әдістемелік нұсқаулардың бірінші бөлігін орындау көп уақытты алады және өте көп еңбекті қажет етеді, бірақ процестің өзі стандартты алгоритм бойынша жүреді.

Екінші бөлім PHP, Python, HTML, MySQL сияқты тілдерде микрокомпьютерді тікелей бағдарламалауға арналған және белгілі бір дағдылар мен құзыреттілікті талап етеді.

### **№1 зертханалық жұмыс. Raspberry PI операциялық жүйесін таңдау және орнату**

Мақсаты: Raspberry PI микрокомпьютерінің техникалық сипаттамасымен танысу, ОЖ-ны орнату және параметрлерді орнату бойынша практикалық дағдыларды алу.

#### **1.1 Теорияның қысқаша мазмұны**

Raspberry PI-ді іске қосу үшін оған операциялық жүйені орнату керек. Linux ресми үш бөлінуі бар:

- 1) Pidora - Fedora базасында.
- 2) Archlinux - бұл таратылымды орнату қолмен дерлік.
- 3) Raspbian - Debian негізделген.

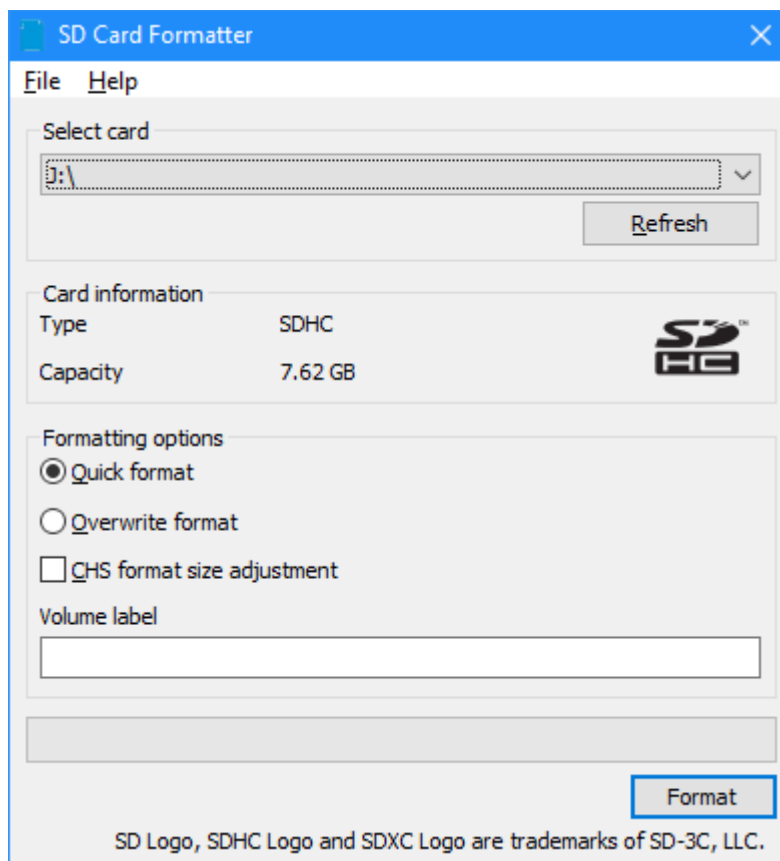
Осы үш операциялық жүйеден басқа Raspberry PI-да көбісі тасымалданған. Операциялық жүйе SD картасына орнатылғандықтан, басқа жүйені бастау үшін, осы жүйемен бірге құрылғыға картаны салу жеткілікті.

Raspberry PI-дегі үлестірімді орнатудың ең оңай жолы - алғашқы жүктеу үшін келесі операциялық жүйелерді таңдауға мүмкіндік беретін Raspberry PI құрушыларының NOOBS құралы: ArchLinux; OpenElec; Pidora; RaspBMC; Raspbian; RiscOs. NOOBS көмегімен Raspberry PI-дегі үлесті орнату үшін 4Гбайт немесе одан көп сыйымдылығы бар microSD картасы

қажет болады.

## 1.2 Орындау тәртібі

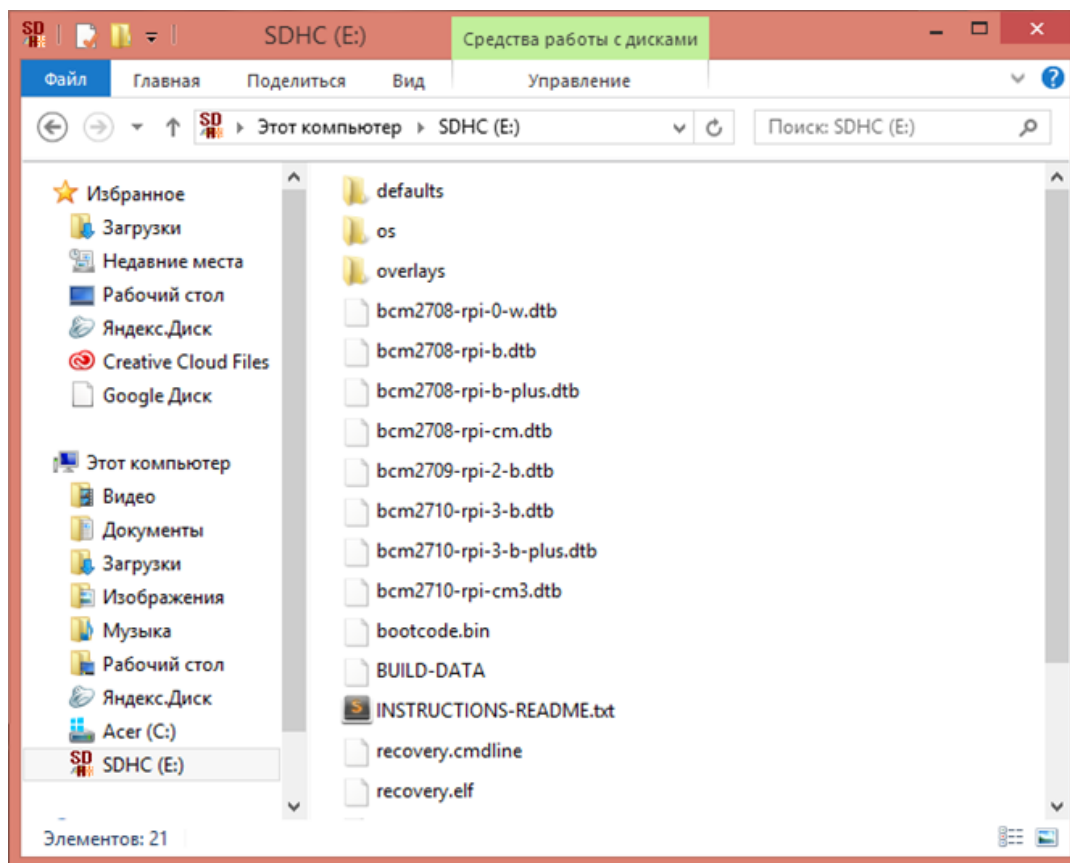
1.2.1 SD Card Formatter қосымшасын ашыңыз (1.1 сурет) және операциялық жүйені орнату үшін дайындық үшін жад картасын пішімдеңіз.



1.1 сурет - SD Card Formatter терезесі

1.2.2 Офистік суретті ресми сайттан жүктеу қажет (олардың көпшілігі бар, бірақ ең өзекті және функционалды болып табылады Linux негізіндегі Raspbian).

1.2.3 ОС-ды сәтті жүктеп алғаннан кейін NOOBS мұрағының мазмұнын бос жад картасына шығарып алуыңыз қажет, себебі біз картаны оқуға арналған адаптерлерді қолданып, оны ұяға салып, суретті жад картасына саламыз. Операцияның нәтижесі 1.2 суретте көрсетілген.



1.2 сурет - Тақтаға орнатуға дайын жад картасының мазмұны

1.2.4 Бұдан әрі, жад картасын Raspberry PI платформасына орнатыңыз, содан кейін құрылғыны ақпаратты және тиісті сымдарды көрсетуге дайындаңыз.

1.2.5 Мониторда Raspberry PI логотипінің қол жетімді бейнесі түрлі түсті гүлдермен көрінеді - бұл ОЖ-дың табысты жүктелуін дәлелдейді. Содан кейін, микрокомпьютерге арналған тиісті параметрлерді таңдап, қажетті параметрлерді бірте-бірте басыңыз.

1.2.6 Орнатуды күтудеміз. Жұмыс үстелімен және айналасындағы элементтермен танысу.

## 1.3 Есеп мазмұны

1.3.1 Жұмыс мақсаты.

1.3.2 Микрокомпьютердің скриншоттары мен фотосуреттерін алу арқылы ОЖ орнатылымының сипаттамасы.

1.3.3 Қорытындылар.

## 1.4 Сынақ сұрақтары

1. Raspberry PI дегеніміз не?
2. Операциялық жүйелер қандай және Raspbian арасындағы айырмашылықтарды сипаттайды?

3. Экранның шығыс құрылғысын қалай жақсартуға болады?
4. Raspberry PI моделіңіз туралы айтып беріңізші?
5. Жұмыс үстелін пәрмен жолынан барлық командаларға қол жеткізе аласыз ба?

## **№2 зертханалық жұмыс. Пакеттерді және қосымшаларды орнату**

Мақсаты: пәрмен жолымен танысу және қосымшаларды басқару, пакеттерді орнату, скриншотпен жұмыс жасау.

### **2.1 Теорияның қысқаша мазмұны**

Орнатылған Raspbian OS негізгі консоль - LXTerminal бар. Негізгі консольді графикалық интерфейсте терезе ретінде эмуляциялайтын алдын-ала орнатылған LXTerminal бағдарламасын пайдалану ыңғайлы. Бұл бағдарламаны ашқан кезде, келесі жолды көреміз:

```
pi@raspberrypi ~ $
```

Мұнда pi - ОС ішіндегі пайдаланушы аты, raspberrypi - хост атауы, ~ ағымдағы каталог, ~ ауыстырады / home / pi. Мұндай сызықты көргенде, біздің командаға жаза аламыз. Атап айтқанда, сіз файлдармен, қалталармен, мұрағаттармен әрекеттерді жасай аласыз (бірақ бұл әрекеттер графикалық режимде орындалады, ол әлдеқайда ыңғайлы). Кейбір әрекеттер әкімшілік құқықтарды талап етеді, оларды sudo командасының басында қосу арқылы ғана пайдалануға болады.

Бағдарламалық құралды орнатудың ең оңай жолы PiStore-ды пайдалану болып табылады. Бұл графикалық интерфейспен ыңғайлы және түсінікті әдіс, бірақ PiStore кітапханасында қалағандай көп бағдарлама жоқ. Raspberry Pi бағдарламаларына арналған бағдарламаларды іздеу, жүктеу және орнатудың баламалы жолы - apt-get консольдық утилитасы. Apt-get Raspberry Pi-да әр түрлі консольдық және графикалық бағдарламалар мен утилиталар туралы деректерді сақтайды.

### **2.2 Орындау тәртібі**

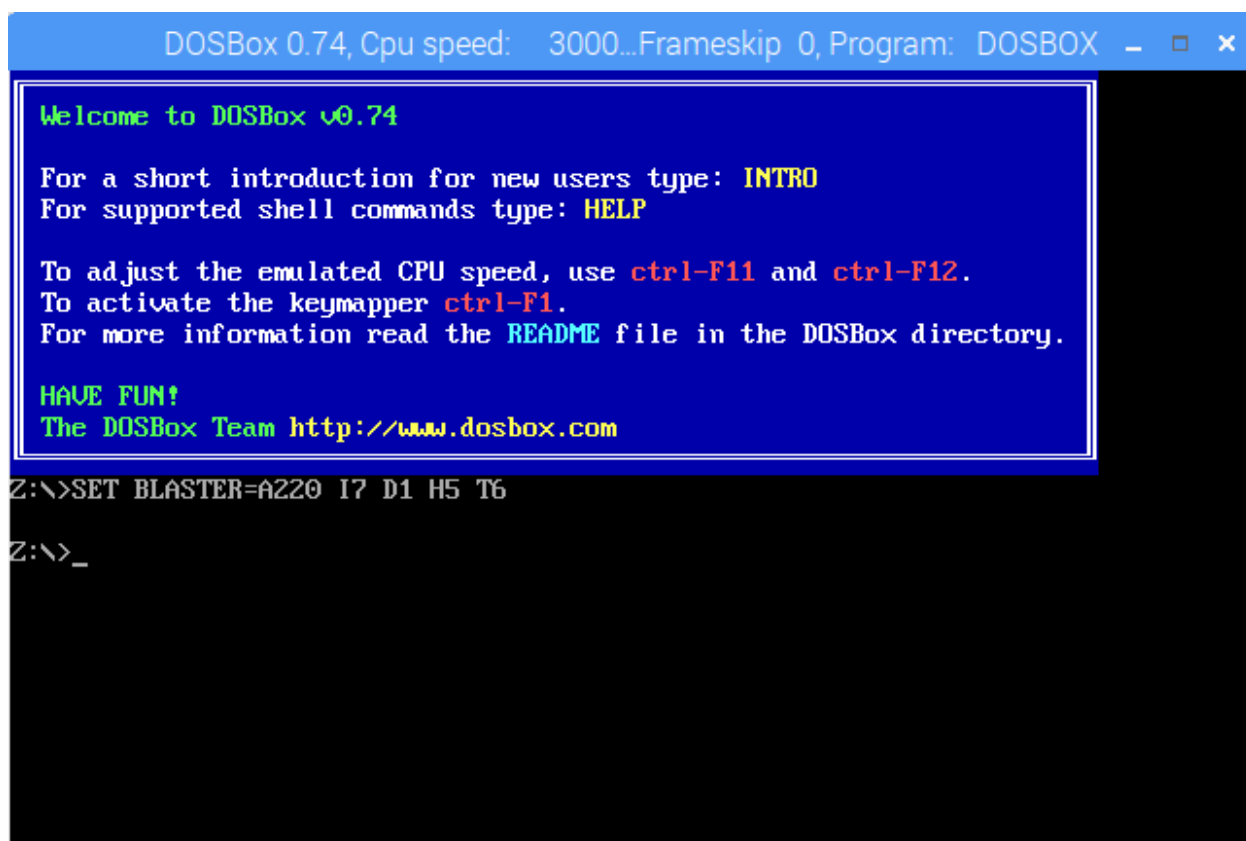
2.2.1 Осы утилитамен жұмыс істемес бұрын, бағдарлама дерекқорын келесі пәрменмен жаңарту қажет:

```
sudo apt-get update
```

2.2.2 Жаңартудан кейін бағдарламаны жүктеуге болады. Утилит деректер базасын іздеу үшін келесі сұранымды жасау қажет:

```
apt-cache search <запрос>
```

Мысалы: `sudo apt-get install dosbox`. Орнатуды Enter арқылы растау қажет.



2.1 сурет – DosBOX интерфейсі

2.2.3 Осы утилитамен мерзімді түрде мына әрекеттерді орындау қажет:

```
sudo apt-get update      (обновление базы программ)
sudo apt-get upgrade     (обновление установленных программ)
sudo apt-get autoremove  (удаления оставшихся после удаления программ библиотек,
```

Бағдарламаны жою үшін пәрменді пайдаланыңыз (жоюды растау қажет):

```
sudo apt-get remove <название>
```

2.2.4 Өкінішке орай, Raspbian OS жүйесінде скриншоттарды жасау үшін алдын ала орнатылған құралдар жоқ. Скриншот жасау үшін құралдарды іздеуді apt-get арқылы іздеуге болады:

```
apt-cache search screen capture
```

Тағыда ыңғайлы бағдарлама - scrot деректер жинағы, орнату:

```
sudo apt-get install scrot
```

Скриншот жасау үшін scrot пәрменін енгізу керек. Пәрменді орындауға кешіктіруді көрсете аласыз - бұл консольді азайту немесе жабу үшін қажет, әйтпесе ол скриншотқа түседі. 5 секундқа кешігу үшін, сіз:

```
scrot -d5
```

## 2.3 Есеп мазмұны

2.3.1 Жұмыс мақсаты.

2.3.2 Орнатылған бағдарламалардың сипаттамасы (міндетті емес), инженерлік коммуникациялардағы жұмыстардың скриншоттары.

2.3.3 Қорытындылар.

## 2.4 Сынақ сұрақтары

1. Dosbox басқа қандай қосымшаларды орнатуға болады? Ағымдағы бағдарламалар туралы айтып беріңіз.

2. Synaptic утилитасын орнату туралы бұйрық қандай болады?

3. Мұрағатта дұрыс буманы табу үшін қандай пәрменді қолдануға болады?

4. Администратордың құқықтарын растау туралы сұраудан бұрын қандай команда қосу керек?

## №3 зертханалық жұмыс. SSH протоколын орнату

Мақсаты: SSH протоколымен танысу, конфигурациялау және қосу.

### 3.1 Теорияның қысқаша мазмұны

Ең алдымен, Secure Shell желілік протоколы (қысқаша SSH деп аталады) туралы мәлімет алу керек.





3.1 сурет – SSH

SSH - екі қашықтағы құрылғылар арасында желі (жергілікті немесе Интернет) арқылы қосылуды қамтамасыз ететін протокол. Онда сіз кез-келген ақпаратпен алмаса аласыз:

- файлдар;
- видео және аудио;
- командалар.

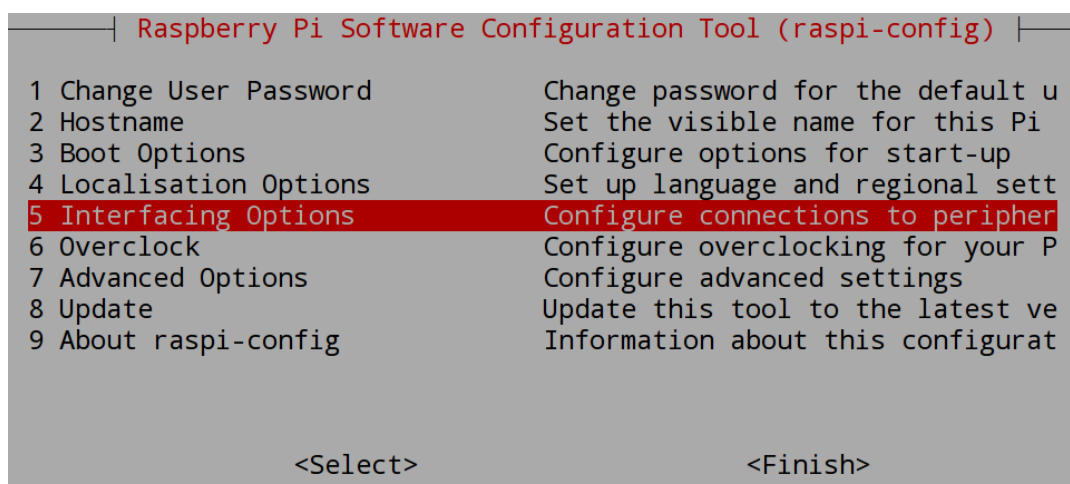
Артықшылықтары: алдымен деректерді қысқа уақытқа қысу мүмкін. Екіншіден, ол шифрланған туннельді жасайды.

Бір кабельдік құрылғыда SSH арқылы Raspberry Pi-ға қосылған кезде, басқа компьютерде жұмыс істейтін терминал іске қосылды. Secure Shell арқылы қосылуға болады, ол тек «таңқурай», сондай-ақ жер шарының басқа жағынан 1 метр болады.

## 3.2 Орындау тәртібі

3.2.1 Консольді ашыңыз және «raspi-config» командасын белгілеңіз және, әрине, құқықтары жоқ адамдар үшін «sudo» командасын қосыңыз.

3.2.2 Enter пернесін басқаннан кейінгі сәтте стандартты Raspberry конфигурациясының утилитасының интерфейсі пайда болады. Онда интерфрейстерді келтіру (бесінші) жолын табу керек.



### 3.2 сурет - Конфигурация интерфейстері

3.2.3 Енді сіз Raspberry Pi туралы SSH қосуыңыз керек. Мұны орындау үшін SSH элементін таңдаңыз (екінші секундта), Enter пернесін басыңыз да, Қосу пәрменін таңдап, қайтадан «Enter» пернесін басыңыз. Содан кейін негізгі экранды қайтару және Аяқтау таңдаңыз.

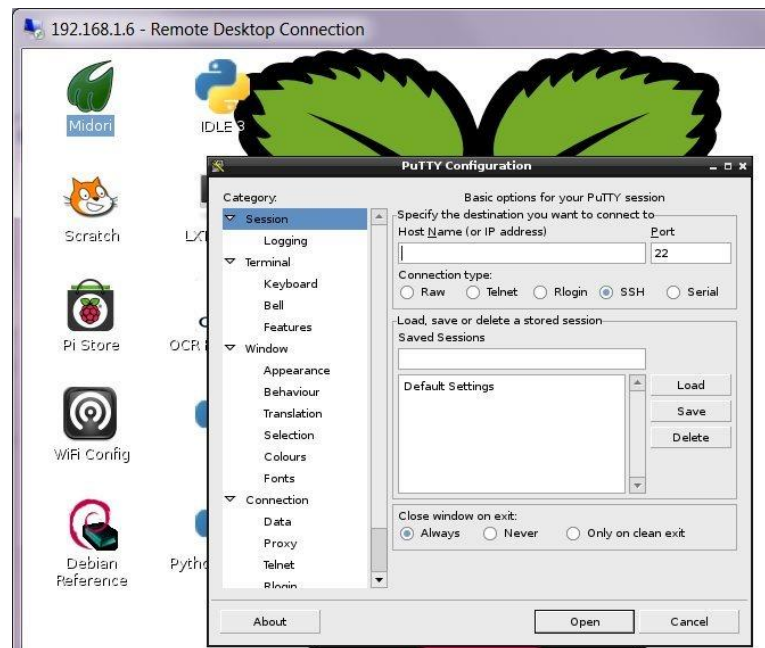
Ескерту – Осыдан кейін барлық өзгертулер қолданылады. Қазір Малинді қайта бастауға болады, бірақ бұлай болмаса да, SSH жұмыс істейді.

3.2.4 Бұдан әрі, біз хаттаманың жұмысын тексереміз және байланыс орнатуға тырысамыз. Мұны істеу үшін басқа басты компьютерден орнатылған «PuTTY» бағдарламасын орнатамыз.

3.2.5 «Сеанс» қойындысының «PuTTY параметрлері» терезесінде (терезені ашқаннан кейін белсенді) келесі әрекетті орындау қажет: «Хост атауы» өрісінде «Raspberries» мекенжайын енгізіңіз, порт нөмірін енгізіңіз (SSH үшін ең жиі 22-ші пайдаланылады, егер ол ештеңені өзгертпесе, SSH «Қосылу түрі» қосқыштарының тізімінде белсенді болуы керек.

3.2.6 «Қосылу» батырмасын басу қажет. Егер адрес, порт және қосылу түрі дұрыс көрсетілсе және екі құрылғыдағы желі дұрыс жұмыс істеп тұрса, онда Windows пәрмен жолы көрсетіледі және ол арқылы таңқураймен өзара әрекеттесуге болады және CMD қолдамайтын барлық пәрмендер қол жетімді болады.

Ескерту – Консоль пайда болған кезде, жүйеге кіру қажет. Мұны істеу үшін алдымен логинді көрсетіңіз («Malin» ішіндегі пайдаланушы аты) және екіншіден, тиісті пайдаланушыға тағайындалған құпия сөз. Егер олар дұрыс болса, онда компьютерге толық қолжетімділік беріледі.



3.3 сурет – PuTTY бағдарламасы

### 3.3 Есеп мазмұны

3.3.1 Жұмыс мақсаты.

3.3.2 Орнатылған бағдарламалардың сипаттамасы (міндетті емес), инженерлік коммуникациялардағы жұмыстардың скриншоттары.

3.3.3 Қорытындылар.

### 3.4 Сынақ сұрақтары

1. Secure Shell дегеніміз не?
2. SSH арқылы графикалық қосымшалармен жұмыс істеуге болады ма?
3. Егер жоқ болса, графикамен қашықтан қалай жұмыс істей аласыз?
4. SSH-пен қосылған кезіде қандай кемшіліктер бар?

### №4 зертханалық жұмыс. VNC қашықтан басқару құралын теңшеу

Мақсаты: желі протоколы арқылы ғана емес, сонымен қатар VNC қашықтан қатынасу арқылы басқаруды жүзеге асыру.

#### 4.1 Теорияның қысқаша мазмұны

Raspberry Pi-ді қашықтан басқаруға арналған ең кең таралған жолдардың бірі - Putty функциясын қолданып, SSH протоколын пайдаланып басқарманы басқару. Басқарудың басқа әдістерін ғана емес, сонымен қатар графикалық түрде, кез-келген компьютерден басқаруға болады. VNC (Virtual Network Computing) көмегімен Raspberry-ға қосылу арқылы оны толыққанды графикалық режимде қашықтан басқаруға болады.

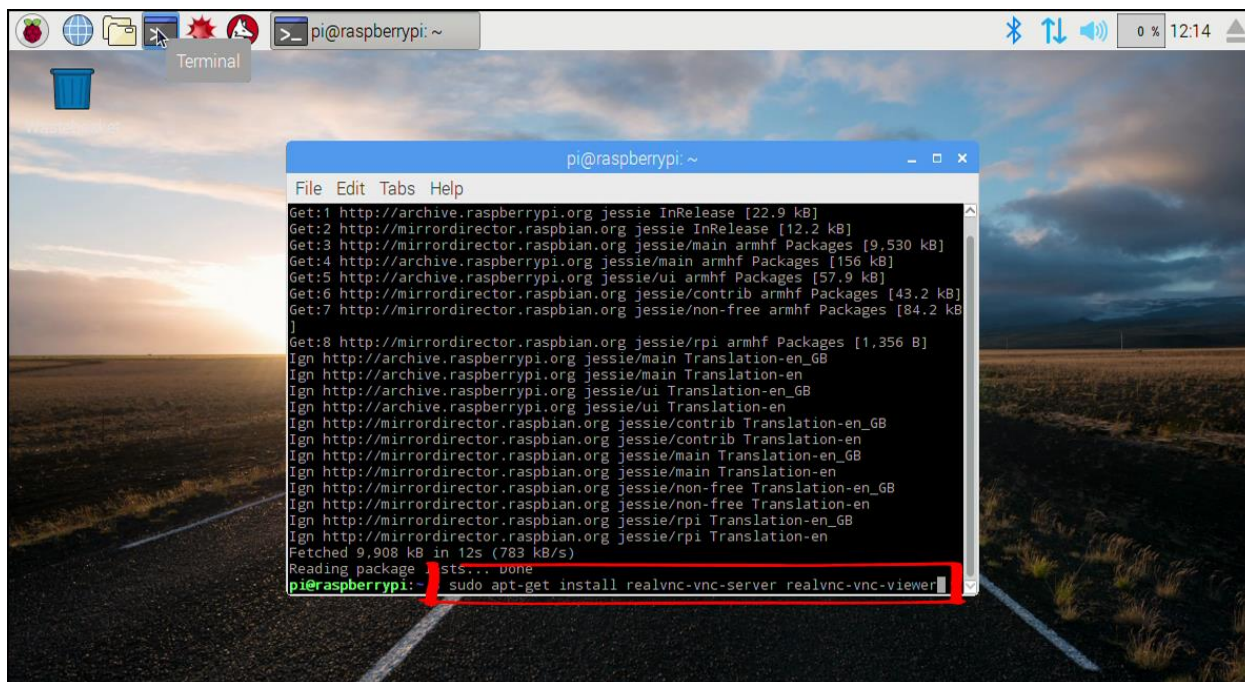
## 4.2 Орындау тәртібі

4.2.1 Терминалды ашыңыз және жүйеде пакеттер тізімін жаңартыңыз (жаңартылмағандар үшін):

```
sudo apt-get update
```

4.2.2 VNC серверін Raspberry PI-ға орнатамыз:

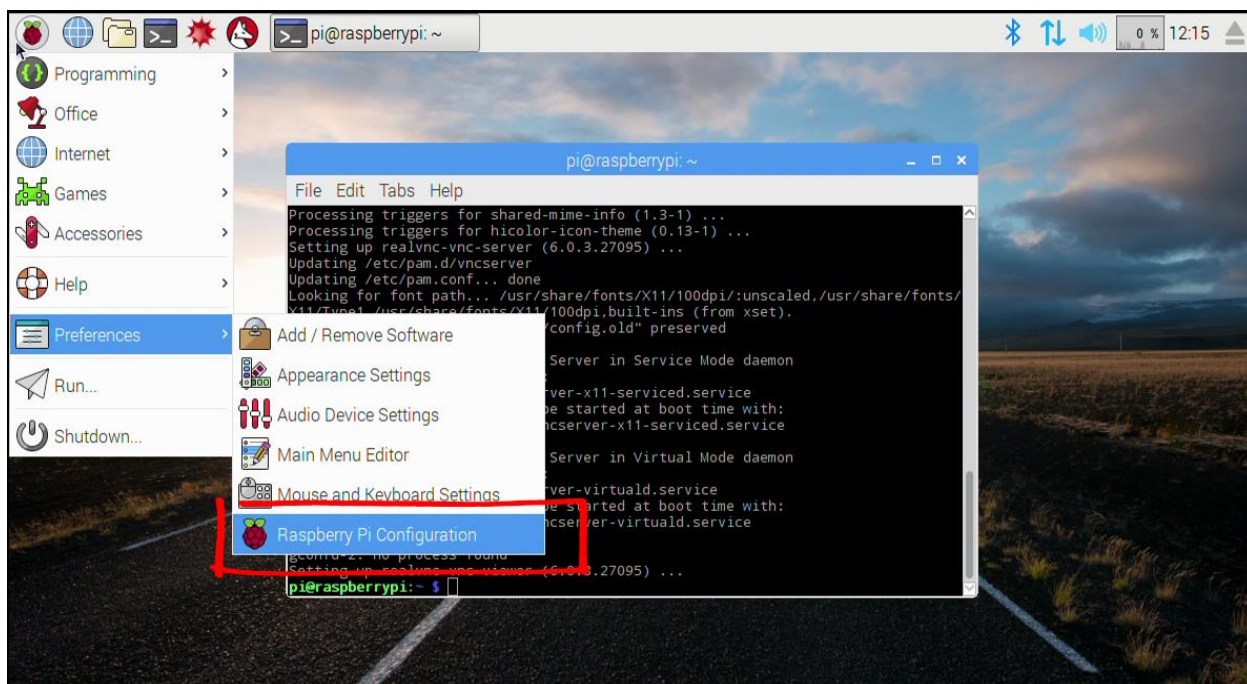
```
sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
```



4.1 сурет – VNC-ді орнату

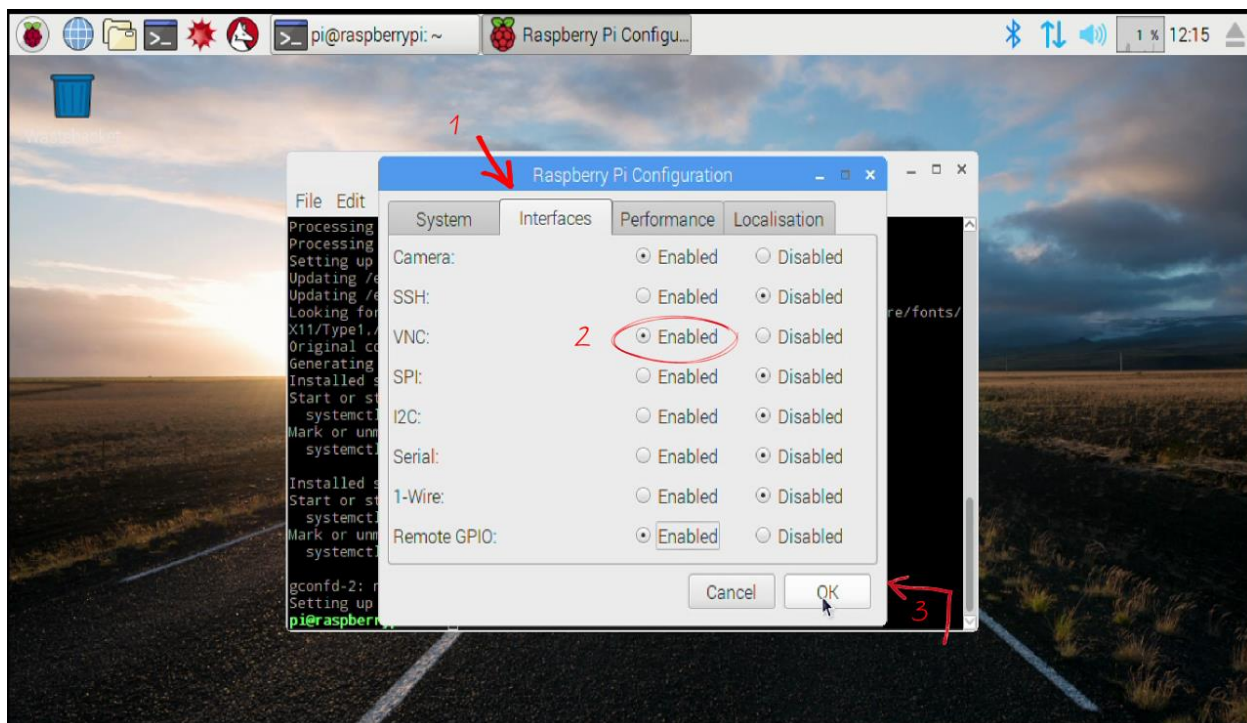
4.2.3 Мәзір элементіне өтіңіз: Menu > Preferences > Raspberry PI configuration.





4.2 сурет – Меню мәзірі

4.2.4 Raspberry Pi микрокомпьютерінің параметрлері терезесі ашылады. Интерфейстер қойындысына өтіп, VNC мәзір элементін тауып, Қосылған күйін таңдап, ОК түймешігін басыңыз.

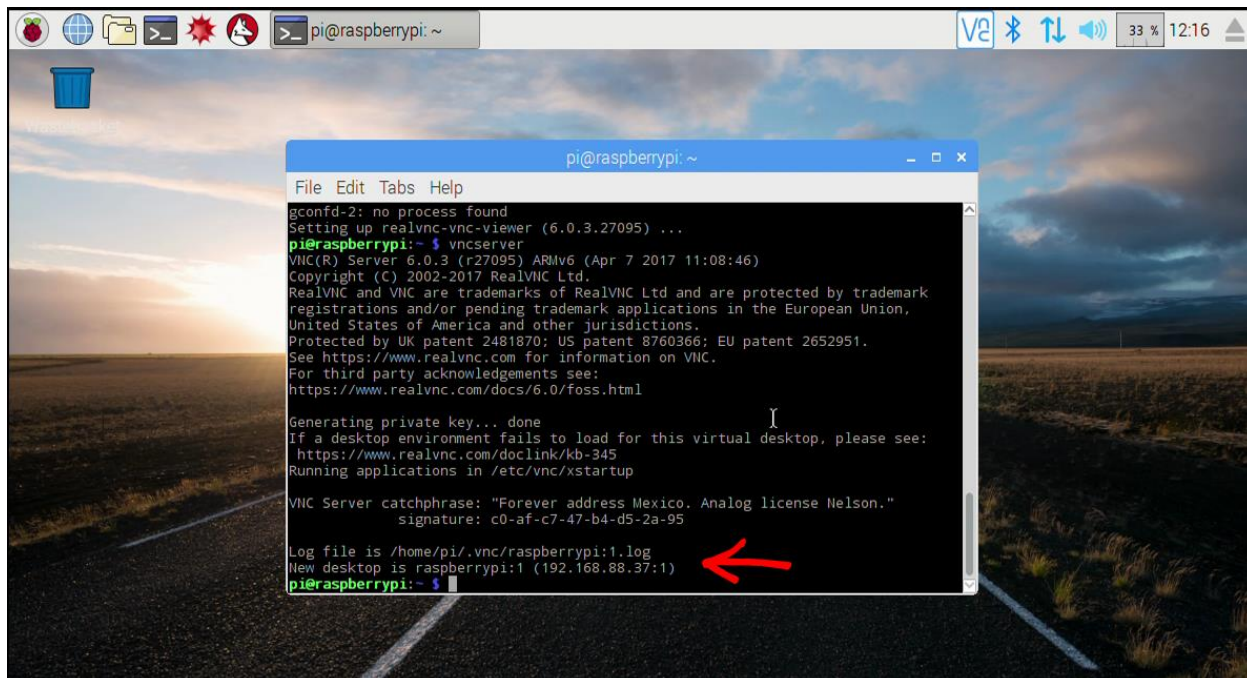


4.3 сурет – VNC-нің қосылуы

4.2.5 «Терминал» терезесіне оралыңыз және VNC серверін бастаңыз.

vncserver

4.2.6 Серверді IP адресімен және порт нөмірімен табысты іске қосу туралы білеміз.



The screenshot shows a Raspberry Pi desktop with a terminal window open. The terminal output is as follows:

```
gconfd-2: no process found
Setting up realvnc-vnc-viewer (6.0.3.27095) ...
pi@raspberrypi:~$ vncserver
VNC(R) Server 6.0.3 (r27095) ARMv6 (Apr 7 2017 11:08:46)
Copyright (C) 2002-2017 RealVNC Ltd.
RealVNC and VNC are trademarks of RealVNC Ltd and are protected by trademark
registrations and/or pending trademark applications in the European Union,
United States of America and other jurisdictions.
Protected by UK patent 2481870; US patent 8760366; EU patent 2652951.
See https://www.realvnc.com for information on VNC.
For third party acknowledgements see:
https://www.realvnc.com/docs/6.0/foss.html

Generating private key... done
If a desktop environment fails to load for this virtual desktop, please see:
https://www.realvnc.com/doclink/kb-345
Running applications in /etc/vnc/xstartup

VNC Server catchphrase: "Forever address Mexico. Analog license Nelson."
signature: c0-af-c7-47-b4-d5-2a-95

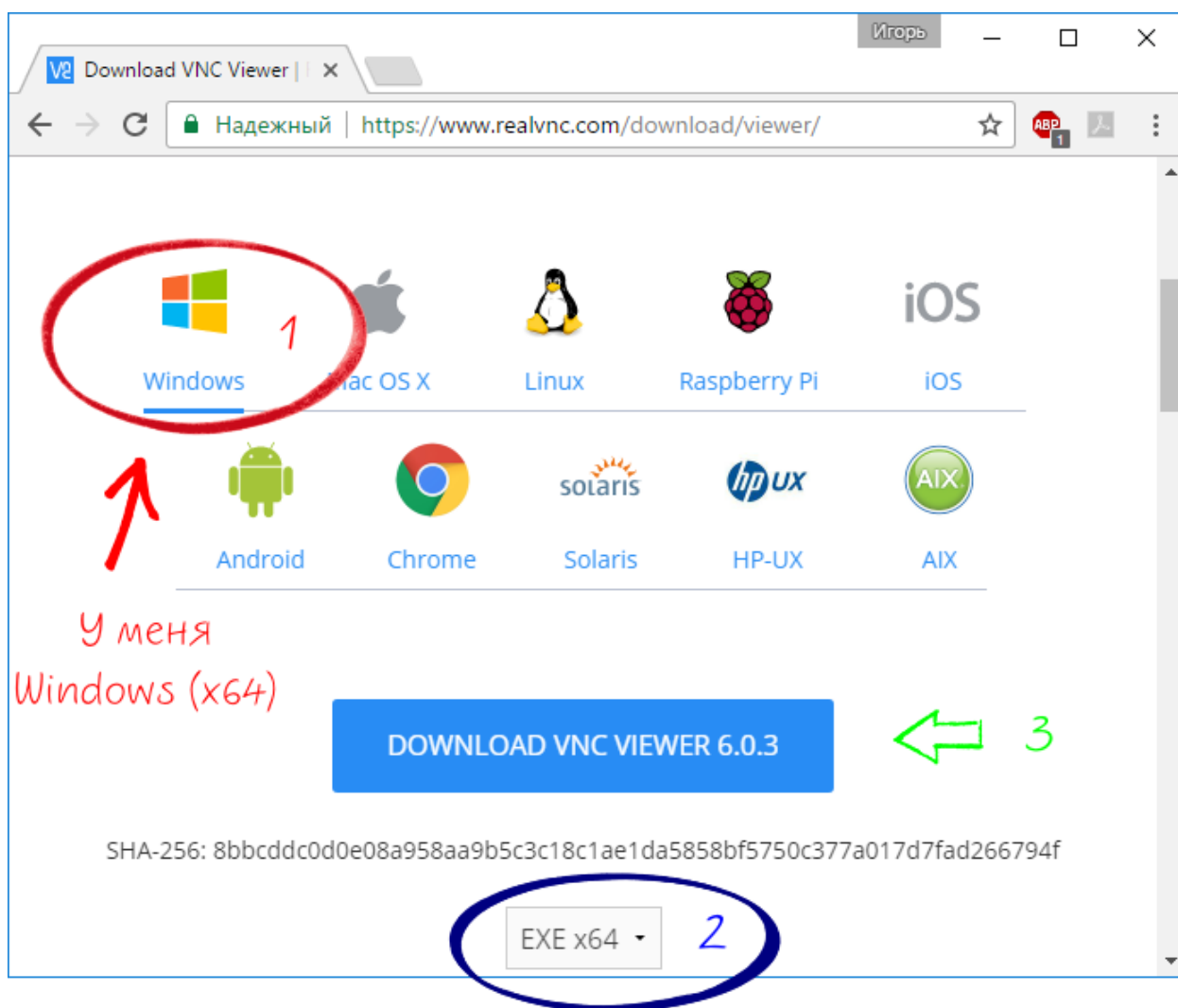
Log file is /home/pi/.vnc/raspberrypi:1.log
New desktop is raspberrypi:1 (192.168.88.37:1)
```

A red arrow points to the IP address 192.168.88.37 in the terminal output.

4.4 сурет – IP-адрес және порттың нөмірі

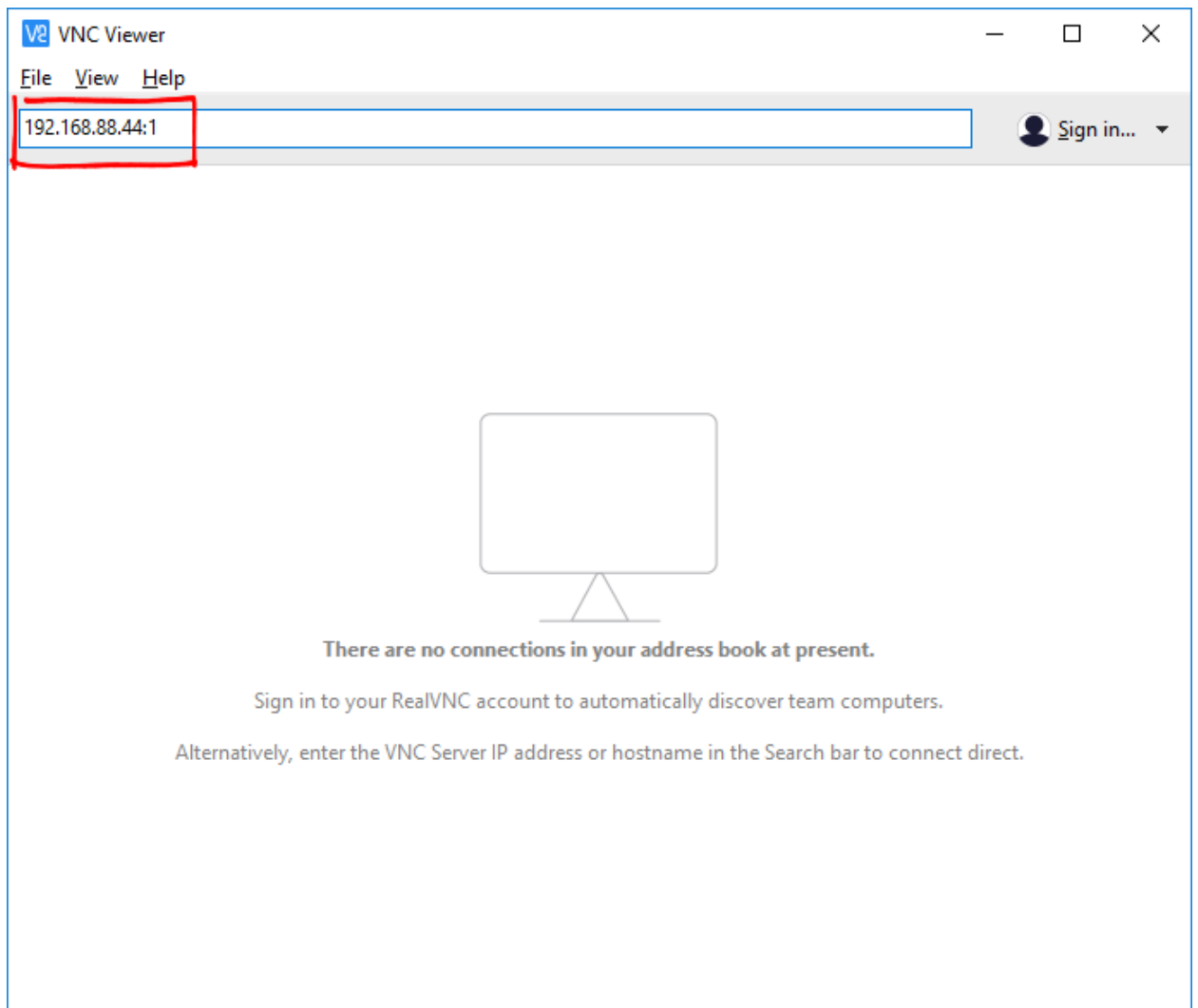
Ескерту – Біздің жағдайда IP-адресі 192.168.88.37, порты саны-1. Бұл барлық нәрсені білдіреді және клиенттік параметрлерге өтуге болады.

4.2.7 Raspberry Pi қашықтағы жұмыс үстеліне қосылу үшін VNC клиентін компьютерге орнатып, теңшеу керек. Ол үшін «REALVNC» ресми парағына өтіңіз. Амалдық жүйеге байланысты ортаны нұсқасын таңдаңыз:



4.5 сурет – Сайтты салыстыру

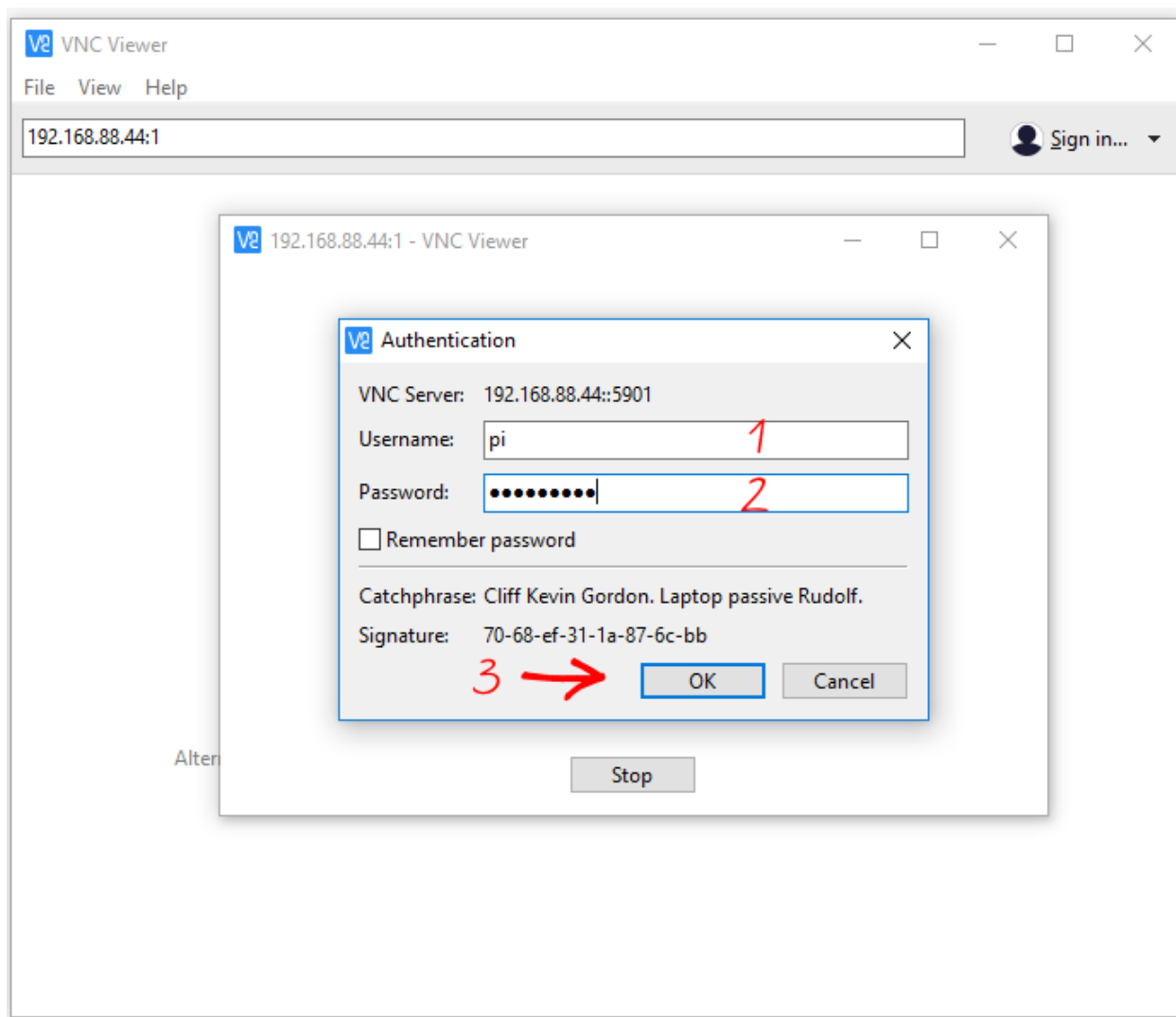
4.2.8 Мекенжай жолағында VNC серверін құру кезінде шығарылған IP мекенжайын енгізіңіз және Enter пернесін басыңыз.



4.6 сурет – Мекен-жайды енгізу

4.2.9 Қалқымалы терезеде Raspberry Pi-дің логинін және паролін енгізіңіз. Әдепкі пайдаланушы аты пі, құпия сөз - raspberry.





4.7 сурет – Деректерді енгізу

Содан кейін «OK» -ді таңдап, алдыңыздан Raspberry Pi-дің өшірілген жұмыс үстелі ашылады. Енді кез келген компьютерден графикалық түрде жұмыс істей аласыз, терминалды басқарудан бас тартасыз.

### 4.3 Есеп мазмұны

4.3.1 Жұмыс мақсаты.

4.3.2 Скриншоттармен қашықтан қол жеткізумен жасалған жұмыс туралы есеп.

4.3.3 Графикалық ортаны басқарудың қорытындылары мен артықшылықтары.

### 4.4 Сынақ сұрақтары

1. VNC дегеніміз не және ол SSH-ны басқарудан өзгеше?
2. SSH арқылы консольдық режимде VNC серверін орнатуды сипаттаңыз?

3. VNC серверін init.d каталогына жазу арқылы сценарий жазу керек пе?

## **№5 зертханалық жұмыс. Веб-серверді PHP, HTML және MySQL көмегімен орнату**

Мақсаты: Веб-серверді дұрыс қызмет көрсету және әрі қарай өндеу үшін тиісті сервистермен бірге орнатыңыз және баптаңыз.

### **5.1 Теорияның қысқаша мазмұны**

Өзіңіздің веб-серверіңіздің болуы өте ыңғайлы. Сынақ сценарийлері, жабдықтың жай-күйі туралы ақпаратты көрсету немесе кішкене жобаны орналастыру - бұл осындай сервермен мүмкін.



5.1 сурет – Веб-серверге арналған қызметтер

### **5.2 Орындау тәртібі**

5.2.1 Apache және PHP орнату үшін келесі пәрмендерді орындаңыз:

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

5.2.2 Енді қызметті қайта іске қосыңыз:

```
sudo service apache2 restart
```

немесе

```
sudo /etc/init.d/apache2 restart
```

5.2.3 Веб-браузерде Raspberry Pi-дің IP-мекен-жайын енгізіңіз. «It Works!» Деген қарапайым бетті көруге болады. IP-мекен-жайын білу үшін мына пәрменді жазыңыз:

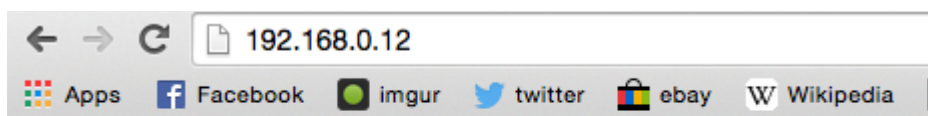
```
ifconfig
```

Бұдан әрі біз келесі параметрлерді көріп отырмыз:

```
Link encap:Ethernet HWaddr fe:fd:45:xx:xx:xx
inet addr:69.164.xxx.xxx Bcast:69.164.xxx.xxx Mask:255.255.255.0
inet6 addr: fe80::fcfd:xxx:xxx:xxx/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:35463998 errors:0 dropped:0 overruns:0 frame:0
TX packets:30563995 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:11300336376 (11.3 GB) TX bytes:33179556297 (33.1 GB)
Interrupt:76
```

Ескерту – Интернетке қосылғаныңызды тексеріңіз. Ethernet кабелі қосылған болса, eth0 белгісі пайда болады. Сымсыз желі wlan0 деп белгіленуі мүмкін, бірақ басқа нұсқау мүмкін.

Екінші жол «inet address: 69.164.xxx.xxx» Raspberry Pi-дің IP-мекен-жайын көрсетеді. Енді осы мекенжайды веб-браузерге енгізіңіз:



## It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

5.2 сурет – Сіз енгізгенде көрсету

Ескерту – Егер қате туралы хабар келесі мәтінмен көрсетілсе: «wget: command not found», «sudo apt-get install wget» іске қосыңыз.

### 5.3 Есеп мазмұны

5.3.1 Жұмыс мақсаты.

5.3.2 Тиісті скриншоттары бар веб-браузер жасаған жұмыс туралы есеп.

5.3.3 Өзіңіздің беттеріңіздегі өзгертілген мазмұн.

### 5.4 Сынақ сұрақтары

1. Келесі бумалар мен қызметтердің қайсысы веб-бетке және олардың мақсаттарына қосыла алады?

2. Веб-сервер файлдарының түбірі қандай?
3. «nano test.php» пәрменін пайдалану PHP қызметінің жұмысын тексеріп, веб-бетке тиісті командалармен барған кезде күн мен уақытты орнатыңыз.

## **№6 зертханалық жұмыс. MySQL серверін орнату**

Мақсаты: MySQL серверін орнату және теңшеу, деректер базасына кіру, өзгерту және сақтау.

### **6.1 Теориядан қысқаша мәліметтер**

MySQL – бұл реляциялық деректер қорын басқару жүйесі. MySQL сервері деректермен тиімді жұмыс істеуге мүмкіндік береді және бір уақытта бірнеше пайдаланушыларға жылдам қол жеткізуді қамтамасыз етеді.

Реляциялық деректер базасы-бір-бірімен байланысты екі өлшемді кестелер жиынтығы. Әрбір кестеде жазбалар жиынтығы бар. Өз кезегінде жазба-байланысты ақпаратты қамтитын өрістер жиынтығы. Деректер базасындағы кез келген өріс аты және белгілі бір түрі бар. Кестенің аты деректер қорының шегінде бірегей болуы тиіс. Өз кезегінде өріс аты кесте шегінде бірегей болуы керек.

SQL арнайы тілі деректер базасын және кестелерді жасауға, деректерді қосуға, өзгертуге және жоюға, сұрау бойынша деректерді алуға мүмкіндік береді.

### **6.2 Орындау тәртібі**

6.2.1 MySQL-ді орнату үшін келесі пәрменді пайдаланып бірнеше пакеттерді орнатыңыз.

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

6.2.2 Біз FTP файлын Raspberry Pi-ға / файлдарға көшіру үшін орнатамыз. Веб-қалта рұқсаттарының артықшылықтарын өзгерту.

```
sudo chown -R pi /var/www
```

6.2.3 Бұдан әрі vsftpd орнатамыз.

```
sudo apt-get install vsftpd
```

6.2.4 vsftpd.conf файлын өңдеу.

```
sudo nano /etc/vsftpd.conf
```

Келесі өзгерістерді жасаңыз.

- `anonymous_enable=YES` to `anonymous_enable=NO`
- Uncomment `local_enable=YES` and `write_enable=YES`
- then go to the bottom of the file and add `force_dot_files=YES`.

Файлды сақтап, CTRL-O, CTRL-X басып шығыңыз.

6.2.5 Содан кейін, vsftpd қайта іске қосыңыз.

```
sudo service vsftpd restart
```

6.2.6 Pi-ның бастапқы қалтасынан / var / www-ге жылдам сілтеме қосыңыз.

```
ln -s /var/www/ ~/www
```

Енді сіз / var / www қалтасына сіз FTP арқылы кіргенде пайда болатын жылдам сілтеме арқылы және Pi пайдаланушы аты арқылы қосыла аласыз.

## 6.3 Есептің мазмұны

6.3.1 Жұмыстың мақсаты.

6.3.2 Веб-браузердің тиісті скриншоттармен атқарылған жұмысы туралы есеп.

6.3.3 Өз беттеріндегі өзгертілген мазмұндар.

## 6.4 Бақылау сұрақтары

1. MySQL сервері не үшін арналған?
2. Деректер базасынан ақпаратты енгізу және шығару қалай жүзеге асырылады?
3. SQL мамандандырылған тіліне қандай функциялар жүктелген?

## №7 зертханалық жұмыс. Камерадан веб-беттегі бейнені тарату

Мақсаты: веб-бетте USB немесе CSI камерасынан хабар тарататын бейнелерді енгізу бойынша тәжірибелік дағдыларды иелену.

## 7.1 Теорияның қысқаша мазмұны

Сурет ағындарын іске асыру үшін Raspberry Pi арналған mjpg-streamer орнату керек.

## 7.2 Орындау тәртібі

7.2.1 Терминалды ашып, libjpeg8-dev кітапханасын теру арқылы орнатыңыз.

```
sudo apt-get install libjpeg8-dev
```

7.2.2 smake қолданбасын теру арқылы орнатыңыз.

sudo apt-get орнату cmake

7.2.3 mjpg-streamer орнату пакетін жүктеңіз.

wget <https://github.com/jacksonliam/michael-streamer/archive/master.zip>

7.2.4 Жүктелген мұрағатты ашыңыз.

unzip ./master

7.2.5 Қалтаға өтіңіз.

cd /home/pi/mjpg-streamer-master/mjpg-streamer-experimental

7.2.5 Бағдарламаны құрыңыз.

make clean all

7.2.6 Бағдарламаны қажетті атрибуттармен іске қосыңыз.

./mjpg-streamer -i «./input\_uvc.so -y» -o «./output\_http.so -w ./www»

7.2.7 Raspberry PI браузерінде (немесе сол желідегі кез келген басқа құрылғыда) бетті ашыңыз.

<http://127.0.0.1:8080/?action=stream> (127.0.0.1 орнына басқа құрылғыдан қатынау үшін Raspberry PI-дің IP-адресін енгізіңіз)

### 7.3 Есеп мазмұны

7.3.1 Жұмыс мақсаты.

7.3.2 Қабылданған шаралардың сипаттамасы.

7.3.3 Командаларды орындау нәтижелерінің скриншоттары.

7.3.4 Қорытындылар.

### 7.4 Сынақ сұрақтары

1. Raspberry PI дегеніміз не?
2. Операциялық жүйелер қандай және Raspbian арасындағы айырмашылықтарды сипаттайды?
3. Экранның шығыс құрылғысын қалай жақсартуға болады?
4. Raspberry PI моделіңіз туралы айтып беріңізші?
5. Жұмыс үстелін пәрмен жолынан барлық командаларға қол жеткізе аласыз ба?

### №8 зертханалық жұмыс. Flask көмегімен веб-серверді қосу

Мақсаты: веб-серверге қатынауды түзету және СИД процестерін басқару әдісін жасау.

#### 8.1 Теориядан қысқаша ақпарат

Бұл зертхана екі жарықдиодты қосуға / өшіруге болатын Raspberry Pi құрылғысын пайдалану арқылы жеке веб-серверді қалай жасау керектігін айтады, бірақ басқа деректерді шығару құрылғыларына ауыстыруға болады. Мысалы, реле немесе транзисторлар. Веб-серверді жасау үшін бізге Python



арқылы Flask микрофразаны қажет.

Осы жобаны аяқтау үшін сізге келесі қосымша жабдық қажет: тақтайша, 2 светодиода, 470 Ом резистор, сымдық сымдар.

## 8.2 Орындау тәртібі

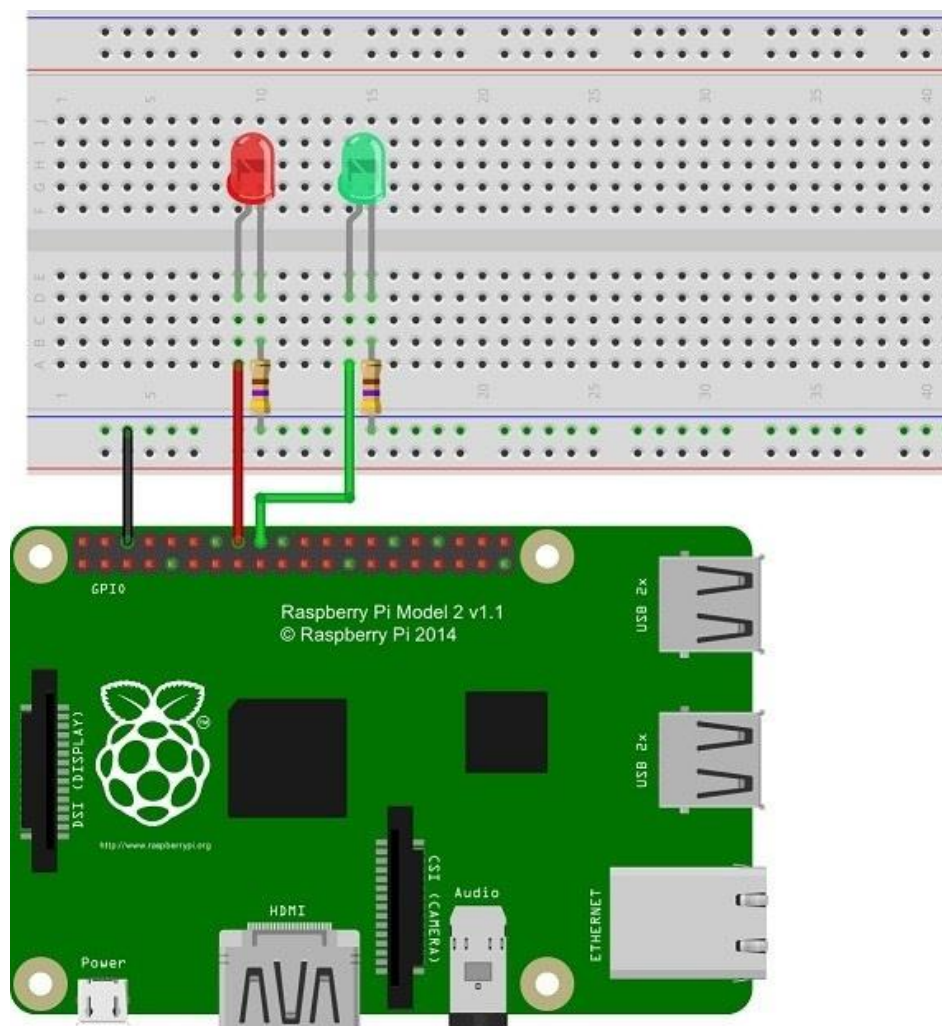
8.2.1 Raspberry Pi-ді веб-серверге айналдыру үшін бізге Flask микрофрейморгы қажет. Флэшке орнату үшін сізде pip орнатылған болуы керек. Raspberry Pi-ді жаңартып, портты орнатыңыз, келесі пәрмендерді іске қосыңыз.

```
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get upgrade
pi@raspberrypi ~ $ sudo apt-get install python-pip python-flask
```

8.2.2 Pip-тің көмегімен Flask-ті және оның тәуелділігін орнатамыз.

```
pi@raspberrypi ~ $ sudo pip install flask
```

8.2.3 Бұл жобаның схемасы өте қарапайым. Төмендегі суретте көрсетілгендей, 2 светодиотты GPIO-23 және 24-пиндерге қосыңыз.



8.1 сурет – Қосылу сұлбасы

8.2.4 Python-сценарийін жасаймыз. Бұл сценарий жобаның негізгі компоненті болып табылады. Ол веб-серверді өңдейді және Raspberry Pi GPIO контактілерімен өзара әрекеттеседі. Барлығын жақсы ұйымдастыру үшін, біз жаңа қалтаны құру арқылы бастаймыз.

```
pi@raspberrypi ~ $ mkdir web-server
pi@raspberrypi ~ $ cd web-server
pi@raspberrypi:~/web-server $
```

8.2.5 «app.py» атымен жаңа файл ашамыз.

```
pi@raspberrypi:~/web-server $ nano app.py
```

8.2.6 Төменде келтірілген сценарийді Raspberry Pi-ға көшіріп, қойыңыз.

```
import RPi.GPIO as GPIO
from flask import Flask, render_template, request
app = Flask(__name__)
GPIO.setmode(GPIO.BCM)
# Сандар туралы деректерді сақтау үшін «pins» деп аталатын сөздікті
жасаңыз,
# атаулар мен байланыс күйлері:
pins = {
    23 : {'name' : 'GPIO 23', 'state' : GPIO.LOW},
    24 : {'name' : 'GPIO 24', 'state' : GPIO.LOW}
}
# Әр контактіні шығу режиміне қойдық,
# және оларға LOW мәнін беріңіз:
for pin in pins:
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, GPIO.LOW)
@app.route("/")
def main():
    # Біз барлық байланыстардың мәртебесін оқып, «pins» сөздігінде
    сақтаймыз:
    for pin in pins:
        pins[pin]['state'] = GPIO.input(pin)
    # «pins» сөздігін «templateData» сөздігіне орналастырамыз:
    templateData = {
        'pins' : pins
    }
    # «templateData» сөздігін HTML-шаблонына жібереміз,
    # содан кейін оны пайдаланушыға қайтарыңыз:
    return render_template('main.html', **templateData)
# Бұл функция біреудің ақпараты бар URL мекенжайын сұрағанда
орындалады.
# байланыс нөмірі және орындалатын әрекет туралы:
@app.route("/<changePin>/<action>")
def action(changePin, action):
```



```

# Контактіні URL мекенжайынан бүтін санға түрлендіріңіз:
changePin = int (changePin)
# Құрылғының атауын шығарып алыңыз:
deviceName = pins [changePin] ['аты']
# URL мекенжайындағы әрекетті «қосулы» болса (яғни, «қосу»),
# одан кейін осы кодты орындаңыз:
егер әрекет == «қосулы» болса:
    # байланыс мәнін жоғары орнатыңыз:
    GPIO.output (өзгертуPin, GPIO.HIGH)
# HTML үлгісіне жіберілетін күй туралы хабарды жасаңыз:
message = «Қосылған» + deviceName + «қосулы.»
# URL мекенжайындағы әрекет «өшірілген» болса (яғни «өшірулі») болса,
# одан кейін осы кодты орындаңыз:
егер әрекет == «off»:
    # LOW байланыс мәнін орнатыңыз:
    GPIO.output (өзгертуPin, GPIO.LOW)
# HTML үлгісіне жіберілетін күй туралы хабарды жасаңыз:
message = «Turned» + deviceName + «өшірулі.»
# Біз барлық байланыстардың мәртебесін оқып шығамыз және сөздікке
«pins» сақтаймыз:
pins үшін:
    pins [pin] ['state'] = GPIO.input (pin)
# «Pins» сөздігін «templateData» сөздігіне енгізіңіз:
templateData = {
    «pins»: pins
}
қайтары render_template ('main.html', ** templateData)
__name__ == «__main__» болса:
    app.run (host = '0.0.0.0', port = 80, debug = True)

```

8.2.7 HTML файлын жасау. Жобаның жақсы ұйымдастырылуы үшін HTML тегтері Python сценарийінен бөлек болуы керек. Flask динамикалық деректерді Python сценарийінен HTML файлына беру үшін пайдаланылатын HTML үлгісін жасайтын Jinja2 пайдаланады. «Templates» деп аталатын жаңа қалтаны жасаңыз.

```

pi@raspberrypi:~/web-server $ mkdir templates
pi@raspberrypi:~/web-server $ cd templates
pi@raspberrypi:~/web-server/templates $

```

«Main.html» атымен жаңа файл жасаймыз.

```

pi@raspberrypi:~/web-server/templates $ nano main.html

```

8.2.8 Төмендегі суреттегі Raspberry Pi пішіміне көшіріп, қойыңыз.

```
<!DOCTYPE html>
```

```
<head>
```

```
<title> RPi веб-сервері </ title>
```

```
<! - Соңғы қысылған және жасалған CSS ->
```

<link rel = «stylesheet» href =  
«https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css» integrity  
= «talk2, басқалармен сөйлескісі келетін басқа адамдармен сөйлескісі келетін  
адамдармен сөйлескісі келетін инвесторлармен сөйлесу үшін.

<!-- Қосымша тақырып -->

<link rel = «stylesheet» href =  
«https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css»  
тұтастық = «sha384-fLW2N01MqjakBkx3l / M9EahuwpSfeNvVv анонимді <>

<!-- Соңғы сығылған және жасалынған javascript -->

<script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"  
integrity="sha384-  
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxs1yVqOtnepnHVP9aJ  
7xS" crossorigin="anonymous"></script>

</head>

<body>

<h1>RPi Web Server</h1>

{% for pin in pins %}

<h2>{{ pins[pin].name }}

{% if pins[pin].state == true %}

is currently <strong>on</strong></h2><div class="row"><div  
class="col-md-2">

<a href="/{{ pin }}/off" class="btn btn-block btn-lg btn-default"  
role="button">Turn off</a></div></div>

{% else %}

is currently <strong>off</strong></h2><div class="row"><div  
class="col-md-2">

<a href="/{{ pin }}/on" class="btn btn-block btn-lg btn-primary"  
role="button">Turn on</a></div></div>

{% endif %}

{% endfor %}

</body>

</html>

8.2.9 Веб-серверді бастаймыз. Raspberry Pi веб-серверін іске қосу үшін  
app.py файлын қамтитын қалтаға өтіңіз.

```
pi@raspberrypi:~/web-server/templates $ cd ..
```

Содан кейін келесі пәрменді іске қосыңыз.

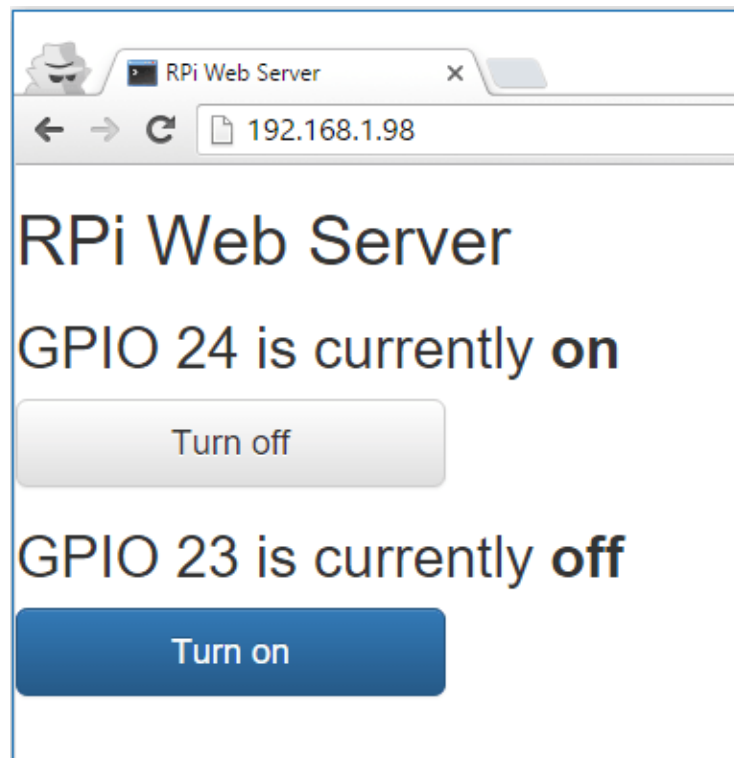
```
pi@raspberrypi:~/web-server $ sudo python app.py
```

Сол секундтан кейін веб-сервер басталуы тиіс, суретте нақты  
көрсетіледі.

```
pi@raspberrypi: ~/web-server
pi@raspberrypi:~/web-server $ sudo python app.py
app.py:24: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.s
  GPIO.setup(pin, GPIO.OUT)
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
app.py:24: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.s
  GPIO.setup(pin, GPIO.OUT)
* Debugger is active!
* Debugger pin code: 186-656-617
192.168.1.66 - - [05/Mar/2016 16:49:52] "GET /24/off HTTP/1.1" 200 -
192.168.1.66 - - [05/Mar/2016 16:49:53] "GET /23/on HTTP/1.1" 200 -
192.168.1.66 - - [05/Mar/2016 16:49:55] "GET /24/on HTTP/1.1" 200 -
192.168.1.66 - - [05/Mar/2016 16:49:55] "GET /23/off HTTP/1.1" 200 -
192.168.1.66 - - [05/Mar/2016 16:49:56] "GET /24/off HTTP/1.1" 200 -
192.168.1.66 - - [05/Mar/2016 16:49:57] "GET /23/on HTTP/1.1" 200 -
192.168.1.66 - - [05/Mar/2016 16:49:57] "GET /24/on HTTP/1.1" 200 -
```

8.2 сурет – Веб-сервердің қосылуы

Бұдан кейін браузерді ашып, Raspberry Pi-дің IP-мекен-жайын енгізіңіз.



8.3 сурет – Веб-сервердің жұмысы

Көрсеткішті көрсету және светодиодты қосу / өшіру параметрлерін сынап көріңіз.

### 8.3 Есеп мазмұны

8.3.1 Жұмыс мақсаты.

8.3.2 Веб-сайттың орындалуы туралы орындалған жұмыстар және

скриншоттар туралы есеп.

8.3.3 Әр оқушының бірегей GPIO портын таңдаудағы өзгерістер.

## 8.4 Сынақ сұрақтары

1. Қол жетімді элементтерді пайдаланып, диодтардың орнына басқа құрылғыларды қосыңыз.
2. Дағдыларға ие болу, сайттың дизайнын өзгерту және өз бетімен жеңілдету.
3. Arduino платформасын осы зертханаға қосуға болады ма? Бұл сұлбаға қандай өзгерістер енгізілетінін түсіндіріңіз.

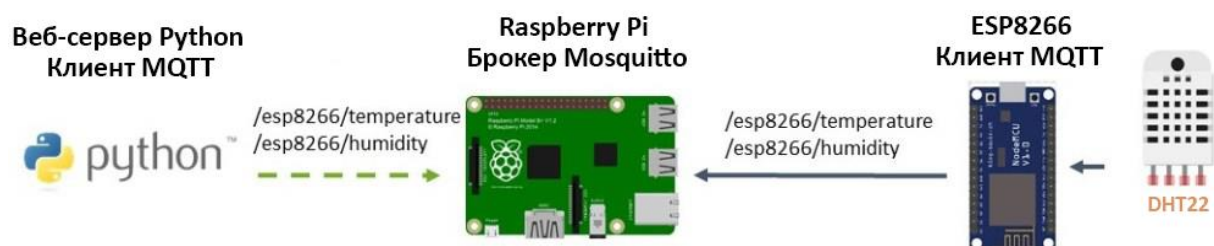
## №9 зертханалық жұмыс. Веб-серверді құру

Мақсаты: DHT22 датчигінен алынған температура мен ылғалдылық деректерін көрсететін автономды веб-серверді жасау.

### 9.1 Теориядан қысқаша мәліметтер

ESP8266 және MQTT протоколының көмегімен екі деректерді шығару құрылғысын (мысалы, жарықдиодты) басқаруға болады.

Веб-серверді құру үшін Python бағдарламалау тілін пайдаланатын Flask микрофреймворк қолданамыз. Жобаланатын қашықтан өлшеу жүйесінің сұлбасы төменде келтірілген.



9.1 сурет – Қашықтан өлшеу жүйесінің сұлбасы

### 9.2 Жұмысты орындау тәртібі

9.2.1 Төлем Тәтті Pi болады сұхбаттаса отырып, ESP8266 көмегімен хаттаманы MQTT. Сондықтан алдымен mosquito делдалын орнатамыз, содан кейін оны фонда іске қосыңыз.

```
pi@raspberrypi:~$ mosquitto -d
```

9.2.2 Бұдан әрі, Flask жұмыс істеуі үшін pip пакеттері қажет, командаларды жазыңыз.

```
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get upgrade
pi@raspberrypi ~ $ sudo apt-get install python-pip python-flask git-core
```

### 9.2.3 Енді кезінде көмек pip орнатамыз Flash және Paho MQTT.

```
pi@raspberrypi ~ $ sudo pip install flask
pi@raspberrypi ~ $ sudo pip install paho-mqtt
```

Ескерту – Бұл жоба Python Flask-та веб-бетті жасауға мүмкіндік беретін Socket IO пайдаланады. Бұл сізге ең жаңа деректерді көру үшін веб-бетті жаңартудың қажеті жоқ дегенді білдіреді, өйткені олар үнемі жаңартылады.

### 9.2.4 FlaskSocket IO бумасын орнату үшін келесіні жазыңыз.

```
pi@raspberrypi ~ $ sudo pip install flask-socketio
```

9.2.5 python-скриптті құру. Бұл скрипт-жобаның негізгі құрамдас бөлігі. Ол веб-серверді теңшейді, сондай-ақ батырмаларды басқанда ESP8266 туралы MQTT-хабарды жариялайды. Сонымен қатар, ол үнемі осы деректерді оқуға мүмкіндік беретін ылғалдылық және температура туралы ақпарат MQTT-топики қол қойылған. Бәрі жақсы ұйымдастыру үшін жаңа қалтаны құрудан бастаймыз.

```
pi@raspberrypi ~ $ mkdir web-server
pi@raspberrypi ~ $ cd web-server
pi@raspberrypi:~/web-server $
```

Жаңа файлды жасаймыз «app.py».

```
pi@raspberrypi:~/web-server $ nano app.py
```

Содан кейін, төменде көрсетілген Raspberry Pi сценарийін көшіріңіз және салыңыз.

```
import paho.mqtt.client as mqtt
from flask import Flask, render_template, request
from flask_socketio import SocketIO, emit
app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret!'
socketio = SocketIO(app)
# Шақыратын кері қоңырау функциясы,
# клиент серверден жауап алған кезде CONNACK:
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
# On_connect () - ке жазылсаңыз, байланыс жоғалғанда
# жазылымдарды қайта қосып, қалпына келтіруге болады.
client.subscribe("/esp8266/temperature")
client.subscribe("/esp8266/humidity")
# Сол жағдайға кері шақыру функциясы,
# егер ESP8266 хабарлама келсе PUBLISH:
def on_message(client, userdata, message):
    #socketio.emit('my variable')
```

```

print("Received message '" + str(message.payload) + "' on topic '"
      + message.topic + "' with QoS " + str(message.qos))
if message.topic == "/esp8266/temperature":
    print("temperature update")
    socketio.emit('dht_temperature', {'data': message.payload})
if message.topic == "/esp8266/humidity":
    print("humidity update")
    socketio.emit('dht_humidity', {'data': message.payload})
mqttc=mqtt.Client()
mqttc.on_connect = on_connect
mqttc.on_message = on_message
mqttc.connect("localhost",1883,60)
mqttc.loop_start()
# Сандар туралы деректерді сақтау үшін "pins" атты сөздік жасаңыз,
# контактілердің атаулары мен жағдайлары:
pins = {
    4 : {'name' : 'GPIO 4', 'board' : 'esp8266', 'topic' : 'esp8266/4', 'state' : 'False'},
    5 : {'name' : 'GPIO 5', 'board' : 'esp8266', 'topic' : 'esp8266/5', 'state' : 'False'}
}
# «pins» сөздігін «templateData» сөздігіне саламыз:
templateData = {
    'pins' : pins
}
@app.route("/")
def main():
    # "TemplateData" сөздігін HTML-үлгіге береміз,
    # содан кейін оны пайдаланушыға қайтарыңыз:
    return render_template('main.html',      async_mode=socketio.async_mode,
    **templateData)
# Бұл функция біреу ақпарат URL сұрағанда орындалады
# Контакт нөмірі ЖӘНЕ ОРЫНДАУ КЕРЕК әрекет туралы:
@app.route("/<board>/<changePin>/<action>")
def action(board, changePin, action):
    # URL контактісін бүтін санға айырбастаймыз:
    changePin = int(changePin)
    # Құрылғының атауы туралы деректерді алу:
    devicePin = pins[changePin]['name']
    # Егер әрекет URL - бұл " 1 " болса, осы кодты орындаңыз:
    if action == "1" and board == 'esp8266':
        mqttc.publish(pins[changePin]['topic'], "1")
        pins[changePin]['state'] = 'True'
    if action == "0" and board == 'esp8266':
        mqttc.publish(pins[changePin]['topic'], "0")
        pins[changePin]['state'] = 'False'
    # "TemplateData" сөздік "pins" және хабарлама:

```

```

templateData = {
    'pins' : pins
}
return render_template('main.html', **templateData)
@socketio.on('my event')
def handle_my_custom_event(json):
    print('received json data here: ' + str(json))
if __name__ == "__main__":
    socketio.run(app, host='0.0.0.0', port=8181, debug=True)

```

9.2.6 Өткен жұмыста сияқты HTML файлын құрайық. Жобаны жақсы ұйымдастыру үшін HTML тегтерін скрипттен бөлек Python-да ұстау керек. Flask Jinja2 - Бұл HTML-үлгілерге арналған қозғалтқыш, ол Python скриптынан HTML-файлға динамикалық деректерді беруге мүмкіндік береді. Templates " атты жаңа қалтаны жасаймыз».

```

pi@raspberrypi:~/web-server $ mkdir templates
pi@raspberrypi:~/web-server $ cd templates
pi@raspberrypi:~/web-server/templates $

```

Содан кейін «main.html» атты жаңа файл жасаңыз.

```

pi@raspberrypi:~/web-server/templates $ nano main.html

```

Төменде көрсетілген Raspberry Pi үлгісін көшіріңіз және салыңыз.

```

<!DOCTYPE html>
<head>
    <title>RPi Web Server</title>
    <!-- Соңғы сығылған және жинақталған CSS -->
    <link
                                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
integrity="sha384-
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGm
kzs7" crossorigin="anonymous">
    <!-- Опциональды тақырып -->
    <link
                                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-
theme.min.css"
                                                integrity="sha384-
fLW2N01lMqjakBkx3l/M9EahuwPsfENvV63J5ezn3uZzapT0u7EYsXMjQV+0En
5r" crossorigin="anonymous">
    <!-- Соңғы сығылған және жинақталған JavaScript -->
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"
integrity="sha384-
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxss1yVqOtnepnHVP9aJ
7xS" crossorigin="anonymous"></script>
    <script src="https://code.jquery.com/jquery-3.1.1.min.js" integrity="sha256-
hVVnYaiADRT02PzUGmuLJr8BLUSjGIZsDYGMjLv2b8="

```

```

crossorigin="anonymous"></script>
<meta name="viewport" content="width=device-width, initial-scale=1">
<script type="text/javascript"
src="//cdnjs.cloudflare.com/ajax/libs/socket.io/1.3.6/socket.io.min.js"></script>
<script type="text/javascript" charset="utf-8">
$(document).ready(function() {
    var socket = io.connect('http://' + document.domain + ':' + location.port);
    socket.on('connect', function() {
        socket.emit('my event', {data: 'I\'m connected!'});
    });
    socket.on('dht_temperature', function(msg) {
        var nDate = new Date();
        $('#readingsUpdated').text(nDate.getHours() + 'h:' + nDate.getMinutes() +
            'm:' + nDate.getSeconds() + 's').html();
        $('#temperature').text(msg.data).html();
    });
    socket.on('dht_humidity', function(msg) {
        $('#humidity').text(msg.data).html();
    });
});
</script>
</head>
<body>
<h1>RPi Web Server - ESP8266 MQTT</h1>
{% for pin in pins %}
<h2>{{ pins[pin].name }}
{% if pins[pin].state == 'True' %}
    is currently <strong>on</strong></h2><div class="row"><div class="col-md-
2">
    <a href="/esp8266/{{ pin }}/0" class="btn btn-block btn-lg btn-default"
role="button">Turn off</a></div></div>
{% else %}
    is currently <strong>off</strong></h2><div class="row"><div class="col-md-
2">
    <a href="/esp8266/{{ pin }}/1" class="btn btn-block btn-lg btn-primary"
role="button">Turn on</a></div></div>
{% endif %}
{% endfor %}
<h3>DHT Readings (updated <span id="readingsUpdated"></span></h3>
<h3>Temperature: <span id="temperature"></span>°C</h3>
<h3>Humidity: <span id="humidity"></span>%</h3>
</body>
</html>

```



### **9.3 Есептің мазмұны**

9.3.1 Жұмыстың мақсаты.

9.3.2 Өз үй-жайының температурасы мен ылғалдылығының өлшенген скриншоты.

9.3.3 Қорытындылар.

### **9.4 Бақылау сұрақтары**

1. DHT11 датчигінің DHT22-ден айырмашылығы неде?
2. Осы датчиктің артықшылықтары мен кемшіліктерін атаңыз.
2. Егер DHT11 сенсорын көрсетушінің орнына қоссаңыз, өлшеу нәтижесі өзгереді ме?

### **№10 зертханалық жұмыс. Температура мен ылғалдылықты телеөлшеу жүйесін әзірлеу**

Мақсаты: ESP8266 чипін бағдарламалау және IoT технологиясы бойынша телеөлшеу жүйесін құру.

#### **10.1 Жұмысты орындау тәртібі**

10.1.1 Дербес компьютерді немесе Raspberry Pi (Arduino IDE орнатылған жағдайда) қосамыз. ESP8266 чипін Raspberry Pi веб-серверімен байланыстыра алу үшін PubSubClient кітапханасын орнату қажет. Оның көмегімен MTTT (шын мәнінде, бұл Python веб-серверімен ESP8266 «қарым-қатынас» мүмкіндік береді) қолдайтын серверге жазылу/жариялау түрінде хабарламаларды жіберуге қабілетті Клиентті құруға болады.

10.1.2 pubsubclient кітапханасын жүктеу немесе оқытушыдан алу керек. Жүктелген ZIP-архивті тарқату. Нәтижесінде сізде «pubsub client-master» атты папка болуы керек. Бұл қалтаны «pubsubclient» деп өзгертіңіз. «Pubsubclient» қалтасын IDE Arduino «libraries» қалтасына жылжытыңыз.

Ескерту – Кітапханамен жиынтықта бірнеше скетч-мысалдар бар. Оларды көру үшін IDE Arduino файлына басыңыз > мысалдар > PubSubClient (File > Examples > PubSubClient).

10.1.3 DHT датчигіне арналған кітапхананы орнату. Zip-оқытушы мұрағатын аламыз. Бұл кітапхана ESP8266 немесе Arduino платасы арқылы температура мен ылғалдылық туралы деректерді оқу үшін кез келген DHT сенсорын пайдалануды жеңілдетеді.

10.1.4 жүктелген ZIP-мұрағатын ашыңыз. Нәтижесінде сізде «DHT-sensor-library-master» атты папка болуы керек. Оны «DHT» деп атаңыз. «DHT» қалтасын IDE Arduino «libraries» қалтасына жылжытыңыз. Қайта IDE Arduino ашыңыз.

10.1.5 Соңында, ESP8266 (SSID деректерін, құпия сөзді және Raspberry Pi IP мекенжайын ауыстырыңыз).

```

// ESP8266WiFi, PubSubClient және DHT кітапханаларын жүктеңіз:
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
// төмендегі жолдардың бірінде түсініктеме белгілерін алып тастаймыз,
// DHT датчигінің түріне сәйкес:
// #define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT21 // DHT 21 (AM2301)
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// ESP8266 роутерге қосылуы үшін осы константаларды өзгертеміз:
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";
// Raspberry Pi IP мекенжайына төменде айнымалыны өзгертеміз,
// ол брокерге қосылуы мүмкін MQTT:
const char* mqtt_server = "YOUR_RPi_IP_Address";
// инициализациялау espClient:
WiFiClient espClient;
PubSubClient client(espClient);
// Светодиоды ESP8266 сәйкес GPIO-түйректеріне жалғаңыз:
const int ledGPIO5 = 5;
const int ledGPIO4 = 4;
// DHT датчигі:
const int DHTPin = 14;
// DHT датчигін инициализациялау:
DHT dht(DHTPin, DHTTYPE);
// таймер үшін қосымша айнымалылар:
long now = millis();
long lastMeasure = 0;
// төменде функцияны өзгертіңіз; ол ESP8266 роутерге қосылады:
void setup_wifi() {
    delay(10);
// WiFi желісіне қосылудан бастаймыз:
    Serial.println();
    Serial.print("Connecting to "); // " Қосы "
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("WiFi connected - ESP IP address: ");
// "WiFi орындалды қосылу-ESP8266 IP мекенжайы: "
    Serial.println (Wi Fi.localIP());
}

```

```

// бұл функция қандай да бір девайс хабарлама жариялағанда орындалады
// ESP8266 қол қойылған топикте; төмендегі функцияны өзгерту
// сіздің бағдарламаның логикасына сәйкес – девайс кезде
// ESP8266 қол қойылған топик хабарлама жариялайды,
// сіздің бағдарламаңыз қалай жұмыс істеді:
void callback(String topic, byte* message, unsigned int length) {
    Serial.print("Message arrived on topic: ");
    // "Хабарлама топикке келді: "
    Serial.print(topic);
    Serial.print(". Message: "); // ". хабарламалар: "
    String messageTemp;
    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }
    Serial.println();
    // егер сіздің жобаңыз 2 LED емес, одан да көп,
    // төменде көп кодты қосыңыз:
    // / / егер топик home/office / sp1 / gpio2 хабарлама алса,
    // бұл хабарлама үшін - " 0 "немесе" 1»;
    // жіберілген мәнге сәйкес GPIO-контактін ауыстырып қосамыз:
    if(topic=="esp8266/4"){
        Serial.print("Changing GPIO 4 to ");
    // "GPIO-контактін 4 күйін өзгерту "
        if(messageTemp == "1"){
            digitalWrite(ledGPIO4, HIGH);
            Serial.print("On");
        }
        else if(messageTemp == "0"){
            digitalWrite(ledGPIO4, LOW);
            Serial.print("Off");
        }
    }
    if(topic=="esp8266/5"){
        Serial.print("Changing GPIO 5 to ");
    // "GPIO-Контакт күйін өзгерту 5 "
        if(messageTemp == "1"){
            digitalWrite(ledGPIO5, HIGH);
            Serial.print("On");
        }
        else if(messageTemp == "0"){
            digitalWrite(ledGPIO5, LOW);
            Serial.print("Off");
        }
    }
}

```

```

Serial.println();
}
// бұл функция переподключает ESP8266 - MQTT-брокерге;
// ESP8266 жазылуды қаласаңыз, осы функцияны өзгертіңіз
// көптеген топиктерге:
void reconnect() {
// қосылғанға дейін циклді қайта іске қосамыз:
while (!client.connected()) {
    Serial.println("Attempting MQTT connection...");
    // "Попытка қосылу MQTT-брокерге... "
    // Қосылуға тырысамыз:
    /*
егер сізде бірнеше құрылғыларды қосу мәселелері болса
- mqtt-брокерге өзгертіңіз орынға төмен.
    * ESP8266 чипін өзгерту үшін, оған беру керек
    бірегей атауы. Сондықтан ол қазір көрінеді:
    if (client.connect ("ESP8266Client")) {
    * Егер сіз қосқыңыз келсе-брокер
        қосымша құрылғылар, оны осылай атауға болады:
        if (client.connect ("ESP Office")) {
            * Басқа ESP үшін:
            if (client.connect ("ESPGarage")) {
            Бұл қосылу мәселесін шешу керек
            МТТТ-брокер:
            Сондай-ақ, бұл атау керек екенін ескеріңіз
            төменде көрсетілгенге сәйкес келу,
            loop блогында ().
            */
            if (client.connect ("ESP8266Client")) {
                Serial.println ("connected"); // " қосылған"
            // топикке жазамыз немесе қайта жазамыз;
            // тек бір ғана емес, бірнеше топиктерге жазылуға болады
            // (бұл мысалға келетін болсақ, бұл
            // көптеген жарықдиодты басқару):
            client.subscribe("esp8266/4");
            client.subscribe("esp8266/5");
        } else {
            Serial.print ("failed, rc="); // " қосылу мүмкін емес"
            Serial.print(client.state());
            Serial.println ("try again in 5 seconds");
            // "Келесі әрекетке дейін 5 секунд"
            // қайтадан көріңіз алдында 5 секунд күтеміз:
            delay (5000);
        }
    }
}
}

```

```

}
// бұл функция GPIO-ESP8266 контактілерін теңшейді
// деректерді шығару режиміне, дәйекті коммуникацияны іске қосады
// жылдамдығы 112500 бод, баптап, MQTT-брокер
// және кері қоңырау функциясын көрсетеді;
// кері қоңырау функциясы хабар алу үшін және,
// шын мәнінде, жарықдиодты басқару:
void setup() {
    dht.begin();
    pinMode(ledGPIO4, OUTPUT);
    pinMode(ledGPIO5, OUTPUT);
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}
// бұл жоба үшін loop () функциясына ештеңе өзгертудің қажеті жоқ;
// мәні, Бұл функция қосылымды орнатады
// ESP8266 және MQTT-брокер:
void loop() {
    if (!client.connected()) {
        reconnect();
    }
    if (!client.loop())
        /*

```

**ЕГЕР СІЗДЕ БІРНЕШЕ ҚҰРЫЛҒЫЛАРДЫ ҚОСУ МӘСЕЛЕЛЕРІ БОЛСА  
- MQTT-БРОКЕРГЕ ӨЗГЕРТІҢІЗ ОРЫНҒА ТӨМЕН.**

```

* ESP8266 чипін өзгерту үшін, оған беру керек
    бірегей атауы. Сондықтан ол қазір көрінеді:
    if (client.connect ("ESP8266Client")) {
* Егер сіз қосқыңыз келсе-брокер
    бірнеше құрылғы, оны осылай атауға болады:
    if (client.connect ("ESP Office")) {
* Басқа ESP үшін:
    if (client.connect ("ESPGarage")) {
    Бұл қосылу мәселесін шешу керек
    МТТТ-брокер:
    Сондай-ақ, бұл атау керек екенін ескеріңіз
    төменде көрсетілгенге сәйкес келу,
    loop блогында ().
    */
    client.connect("ESP8266Client");
    now = millis();
// температура мен ылғалдылық туралы жаңа деректерді жариялаймыз
// әрбір 10 секунд:

```

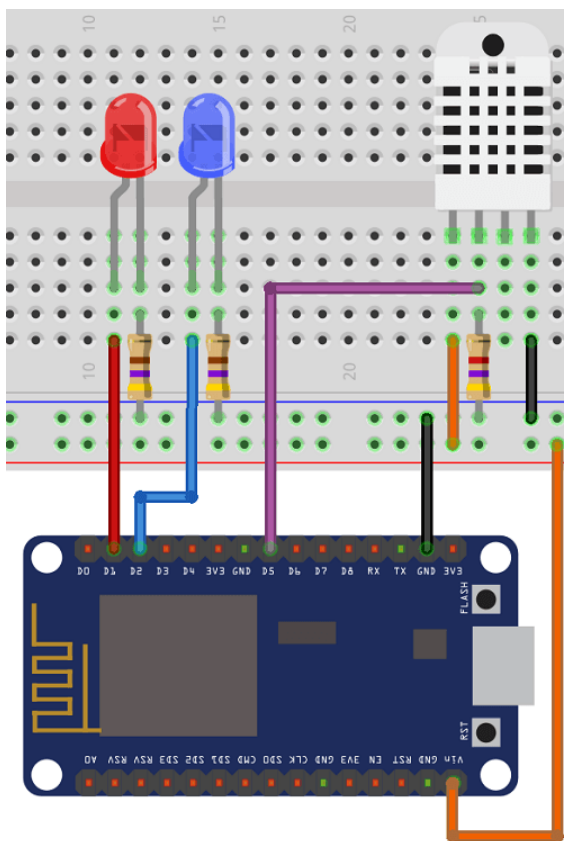
```

if (now - lastMeasure > 10000) {
    lastMeasure = now;
// датчиктен деректер 2 секундқа артта қалуы мүмкін
// (бұл өте баяу сенсор)
    float h = dht.readHumidity();
// температураны бүтін санаймыз (әдепкі бойынша):
    float t = dht.readTemperature();
// считываем температуру Фаренгейтах (сплеш fahrenheit = true):
    float f = dht.readTemperature(true);
// датчиктен деректерді оқып шыққанымызды тексереміз,
// егер жоқ болса, операцияны аяқтаймыз және қайтадан көріңіз:
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT sensor!");
// "Сәтсіз қолданылған деректер датчигінің!"
        return;
    }
// тұтас температураны есептейміз:
float hic = dht.computeHeatIndex(t, h, false);
    static char temperatureTemp[7];
    dtostrf(hic, 6, 2, temperatureTemp);
// егер Фаренгейттердегі температураны есептеу қажет болса,
// төмендегі жолдардан түсініктеме белгілерін алып тастаймыз:
// float hif = dht.computeHeatIndex(f, h);
// static char temperatureTemp[7];
// dtostrf(hic, 6, 2, temperatureTemp);
    static char humidityTemp[7];
    dtostrf(h, 6, 2, humidityTemp);
// температура мен ылғалдылық туралы деректерді жариялаймыз:
    client.publish("/esp8266/temperature", temperatureTemp);
    client.publish("/esp8266/humidity", humidityTemp);
    Serial.print("Humidity: "); // "Влажность: "
    Serial.print(h);
    Serial.print(" %\t Temperature: "); // " %\t Температура: "
    Serial.print(t);
    Serial.print(" *C ");
    Serial.print(f);
    Serial.print(" *F\t Heat index: "); // " *F\t Тепловой индекс: "
    Serial.print(hic);
    Serial.println(" *C ");
    // Serial.print(hif);
    // Serial.println(" *F");
}
}

```

10.1.6 Жобаны аяқтау үшін келесі компоненттер қажет: бір ESP8266 12E модулі, бір DHT22 сенсоры, бір 4700 Ом резисторы, екі 470 Ом резисторы, екі жарық диоды.

Ескерту – Бұл жоба үшін DHT датчигінің басқа да модельдері жұмыс істейді, бірақ ол үшін кодта аз өзгерістер қажет.



10.2 сурет – Қосылу схемасы

Ескерту – DHT датчигінің дұрыс жұмыс істеуі үшін 5 вольт қажет, сондықтан оны бес вольт беретін ESP8266 Vin контактісіне қосылғанына көз жеткізіңіз.

10.1.7 Raspberry Pi веб-серверін іске қосу үшін файл бар қалтаға өтіңіз «app.py»:

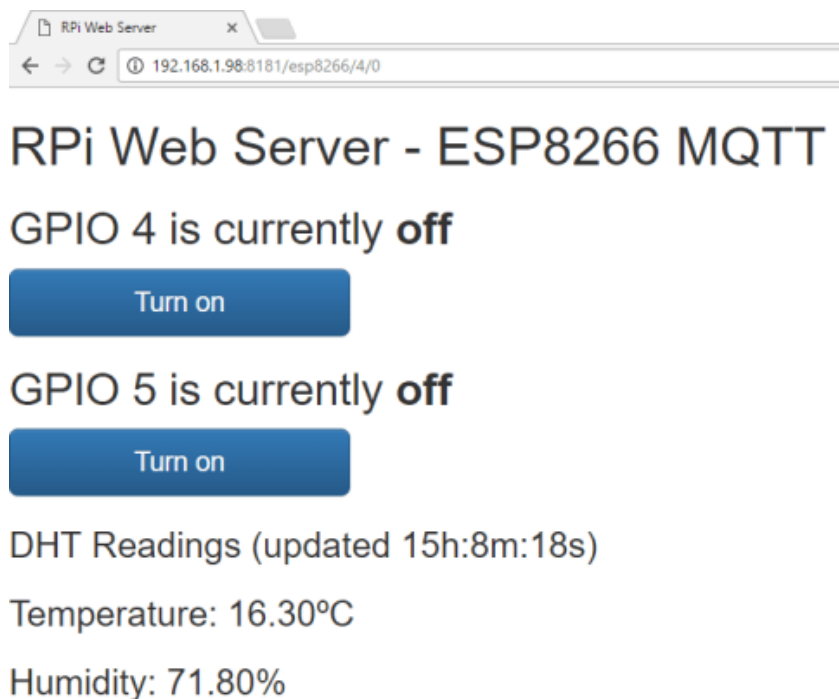
```
pi@raspberrypi:~/web-server/templates $ cd ..
```

Содан кейін келесі команданы іске қосыңыз:

```
pi@raspberrypi:~/web-server $ sudo python app.py
```

Содан кейін бірден веб-серверді іске қосу керек және бұл портта болуы тиіс ":81". Браузерде Raspberry Pi IP мекенжайын жазыңыз. Бұл жағдайда не істеуге болады? <http://192.168.1.98:8181>»

Ескерту – IP-мекен-жайдан кейін міндетті түрде ":8181" деген жазба болуы тиіс.



10.3 сурет – Веб-бетті көрсету

Көріп отырғаныңыздай, веб-сервер жұмыс күйінде, температура мен ылғалдылық туралы деректер Socket IO жүйесінде асинхронды жаңартылады, атап айтқанда жаңа деректер үнемі жаңартылады. Оқытушыға тиісті порттарға қосылған жарықдиодты түймелердің температурасы мен ылғалдылығы туралы деректерді және жұмыс істеу қабілетін көрсетіңіз.

## 10.2 Есептің мазмұны

10.2.1 Жұмыстың мақсаты.

10.2.2 Кодта жарықдиодты басқа порттарды ескере отырып өзгерістер енгізу.

10.2.3 Бөлменің температурасы мен ылғалдылығын өлшеу.

10.2.4 Қорытындылар.

## 10.3 Сынақ сұрақтары

1. ESP8266 чип Raspberry Pi веб-серверімен байланысу үшін кітапхананы қай кітапханаға орнату керек?
2. Қайта шақыру функциясы не істейді?
3. ESP8266-12E MQTT брокерінің қызметі қандай?
4. ESP8266-12E чипі мен Raspberry Pi микрокомпьютермен қандай арнаға қосылған?



## Қолданылған әдебиет

- 1 Васильев А.Н. Python на примерах. Практический курс по программированию. – СПб.: Наука и техника, 2017. – 432 с.
- 2 Войтов Н.М. Основы работы с Linux. Учебный курс. – М.: ДМК Пресс, 2010 – 216 с.
- 3 Иго Т. Arduino, датчики и сети для связи устройств: Пер. с англ. - СПб.: БХВ-Петербург, 2016. - 544 с.
- 4 Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. – СПб.: БХВ-Петербург, 2016. – 464 с.
- 5 Петин В.А. – Микрокомпьютеры Raspberry Pi. Практическое руководство. СПб.: БХВ-Петербург, 2015. – 240 с.
- 6 Петин В.А. Проекты с использованием контроллера Arduino – СПб.: БХВ-Петербург, 2016. – 464 с.
- 7 Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Дженльменский набор Web-мастера. – СПб.: БХВ-Петербург, 2019. – 912 с.

## Мазмұны

Кіріспе.....	3
№1 зертханалық жұмыс.....	3
№2 зертханалық жұмыс.....	6
№3 зертханалық жұмыс.....	8
№4 зертханалық жұмыс.....	11
№5 зертханалық жұмыс .....	18
№6 зертханалық жұмыс .....	20
№7 зертханалық жұмыс .....	21
№8 зертханалық жұмыс .....	22
№9 зертханалық жұмыс .....	28
№10 зертханалық жұмыс .....	33
Қолданылған әдебиет .....	41

Бас жоспар 2019 ж., ұстаным 30

Байкенов Бахытжан Сергеевич  
Аязбай Абу-Алим Ерикулы

## АСПАП ЖАСАУДАҒЫ КОМПЬЮТЕРЛІК ТЕХНОЛОГИЯЛАР

5B071600 – «Аспап жасау» мамандығының студенттуріне арналған  
зертханалық жұмыс бойынша әдістемелік нұсқаулар

Редактор Ж. Изтелеуова

Стандарттау жөніндегі маман Г. Мухаметсариева

Подписано в печать 28.09.19

Тираж 20 экз.

Объем 2,1 уч.-изд. л.

Формат 60x84 1/16

Бумага типографическая № 1

Заказ\_\_\_\_\_Цена 1100 тг.

Копировально-множительное бюро  
некоммерческого акционерного общества  
«Алматинский университет энергетики и связи»  
050013, Алматы, ул. Байтурсынова, 126/1