



دانشکده برق، کامپیوتر و فناوری های پیشرفته

گروه مهندسی برق الکترونیک

دستور کار آزمایشگاه معماری کامپیوتر

تهیه کنندگان:

مهندس سید کیان موسوی کیا

امین پاشائی سلطان احمدی

تاریخ تنظیم:

اسفندماه ۱۴۰۱

مقدمه..... ۶

- ۱-۰-۱- چند نمونه از مجموعه برد های SBC ۶
- ۲-۰-۲- آشنایی با سری برد های Raspberry Pi ۶
- ۳-۰-۳- انواع برد های Raspberry Pi ۷
- ۴-۰-۴- معرفی برد Pi 3 Model B ۸
- ۵-۰-۵- تمرینات و تکالیف ۱۰

آزمایش ۱..... ۱۱

- ۱-۱-۱- مراحل راه اندازی Raspberry Pi 3B ۱۱
- ۱-۱-۱-۱- نصب سیستم عامل بر روی sd card ۱۱
- ۲-۱-۱-۲- اتصالات Raspberry Pi 3B ۱۳
- ۳-۱-۱-۳- تنظیمات اولیه سیستم عامل بر روی Raspberry Pi 3B ۱۳
- ۲-۲-۲- کنترل Raspberry Pi از راه دور ۱۸

آزمایش ۲..... ۲۰

- ۱-۲-۱-۲- زبان برنامه نویسی و معرفی نرم افزار کد نویسی ۲۰
- ۱-۱-۲-۱- معرفی برنامه Thonny IDE ۲۰
- ۲-۱-۲-۲- معرفی ترمینال رزبری پای ۲۲
- ۱-۲-۱-۲-۱- برخی از کد های مهم ترمینال رزبری پای ۲۴
- ۲-۲-۲-۲- تمرینات و تکالیف ۲۶

آزمایش ۳..... ۲۶

۳-۱- آشنایی با زبان برنامه نویسی پایتون جلسه اول ۲۶

۳-۱-۱- انواع متغیرها ۲۶

۳-۱-۲- نمونه توابع پایتون ۲۷

۳-۲- تمرینات و تکالیف ۳۱

آزمایش ۴..... ۳۲

۴-۱- آشنایی با زبان برنامه نویسی پایتون جلسه دوم ۳۲

۴-۱-۱- نمونه توابع و دستورات در پایتون ۳۲

۴-۲- تمرینات و تکالیف ۳۶

آزمایش ۵..... ۳۷

۵-۱- راه اندازی led با برد رزبری پای ۳۷

۵-۱-۱- اتصالات مدار ۳۷

۵-۱-۲- برنامه نویسی ۳۹

۵-۲- تمرینات و تکالیف ۴۲

آزمایش ۶..... ۴۳

۶-۱- راه اندازی سنسور DHT 11 ۴۳

۶-۱-۱- اتصالات مدار ۴۳

۶-۱-۲- برنامه نویسی ۴۴

۴۶-۲-۶-راه اندازی سنسور اولتراسونیک ۴۶

۴۶-۱-۲-۶- نحوه کار سنسورهای التراسونیک ۴۶

۴۷-۲-۲-۶- اتصالات مدار ۴۷

۴۸-۳-۲-۶- برنامه نویسی ۴۸

۵۲-۳-۶- تمرینات و تکالیف ۵۲

آزمایش ۷ ۵۳

۵۳-۱-۷- راه اندازی LCD 16*2 کاراکتری ۵۳

۵۴-۱-۱-۷- اتصالات ۵۴

۵۵-۲-۱-۷- برنامه نویسی ۵۵

۵۸-۲-۷- تمرین کلاسی ۵۸

۵۸-۳-۷- تمرینات و تکالیف ۵۸

آزمایش ۸ ۵۹

۵۹-۱-۸- راه اندازی دوربین با رزبری پای ۵۹

۶۰-۱-۱-۸- اتصالات ۶۰

۶۰-۲-۱-۸- برنامه نویسی ۶۰

۶۳-۲-۸- عکس برداری با استفاده از ترمینال رزبین : ۶۳

۶۴-۳-۸- عکس برداری با استفاده از اسکریپت پایتون: ۶۴

۶۵-۴-۸- تمرین کلاسی ۶۵

۵-۸ - فیلم برداری با استفاده از اسکریپت پایتون ۶۵

۶-۸ - تمرینات و تکالیف ۶۶

۶۷ آزمایش ۹

۱-۹ - طرز نصب و راه اندازی matlab support package for raspberry pi ۶۷

۲-۹ - تمرین کلاسی ۷۲

۳-۹ - چارچوب تشخیص اشیاء ویولا-جونز ۷۲

۱-۳-۹ - مقدمه ۷۲

۲-۳-۹ - شرح مسئله ۷۳

۳-۳-۹ - اجزاء چارچوب ۷۴

۴-۳-۹ - ویژگی های شبه Haar ۷۶

۵-۳-۹ - تصویر انتگرالی ۷۸

۶-۳-۹ - الگوریتم یادگیری ۷۹

۷-۳-۹ - نمونه کد نویسی تشخیص چهره ۸۱

SBC مخفف عبارت Single Board Computer به معنای کامپیوتر تک برد است. همانطور که از اسم آن بر می آید کامپیوتری است که در یک برد کوچک خلاصه می شود. دارای پردازنده، حافظه و ورودی خروجی های مورد نیاز برای یک کامپیوتر عادی است. اغلب علاوه بر آن امکان دسترسی مستقیم به یک سری پروتکل هایی همچون GPIO و SPI و I2C و LCD و Camera و ... نیز در این سیستم ها وجود دارند. همین ویژگی ها باعث می شود این سیستم ها فراتر از یک کامپیوتر عادی بوده و کاربری های جالبی برای آنها پیدا شود.

۱-۰- چند نمونه از مجموعه برد های SBC

- Raspberry Pi
- Odroid
- ASUS Tinker Board
- Libre Computer Board
- LattePanda
- Rock PI Rockchip
- Banana Pi
- LattePanda Alpha

بهترین برد های SBC سال ۲۰۲۲ را می توانید در لینک زیر مشاهده کنید.

<https://all3dp.com/1/single-board-computer-raspberry-pi-alternative>

۲-۰- آشنایی با سری برد های RASPBERRY PI

Raspberry Pi رایانه کوچکی (SBC) است که از سال ۲۰۰۶ در حال توسعه بوده و قطعات آن روی یک مادربرد به اندازه کارت بانکی سوار شده است همچنین سیستم عامل Raspbian را می تواند اجرا کند که یک نسخه اختصاصی از سیستم عامل لینوکس می باشد که اختصاصاً برای این رایانه طراحی شده است.

Raspberry Pi کاربردهای محاسباتی ابتدایی اداری، بازی‌های سطح پایین، دسترسی به اینترنت، ایمیل، بازپخش ویدئو و بسیاری از قابلیت‌های دیگر را دارد که به طور معمول از یک رایانه در قرن بیست و یکم انتظار می‌رود. Raspberry Pi همه این امکانات را با تعداد بسیار کمی از قطعات از جمله یک پردازنده ARM و قیمت بسیار پایین عرضه می‌کند.

۳-۰- انواع برد های RASPBERRY PI

- Pi 1 Model B (2012)
- Pi 1 Model A (2013)
- Pi 1 Model B+ (2014)
- Pi 1 Model A+ (2014)
- Pi 2 Model B (2015)
- Pi Zero (2015)
- **Pi 3 Model B (2016)**
- Pi Zero W (2017)
- Pi 4 Model B (2019)
- Pi 4 (2020)
- Pi Zero 2W (2021)
- Pi Pico (2021)
- Pi 5 (2023)

PI 3 MODEL B - معرفی برد



شکل ۱-۰: برد PI 3 MODEL B

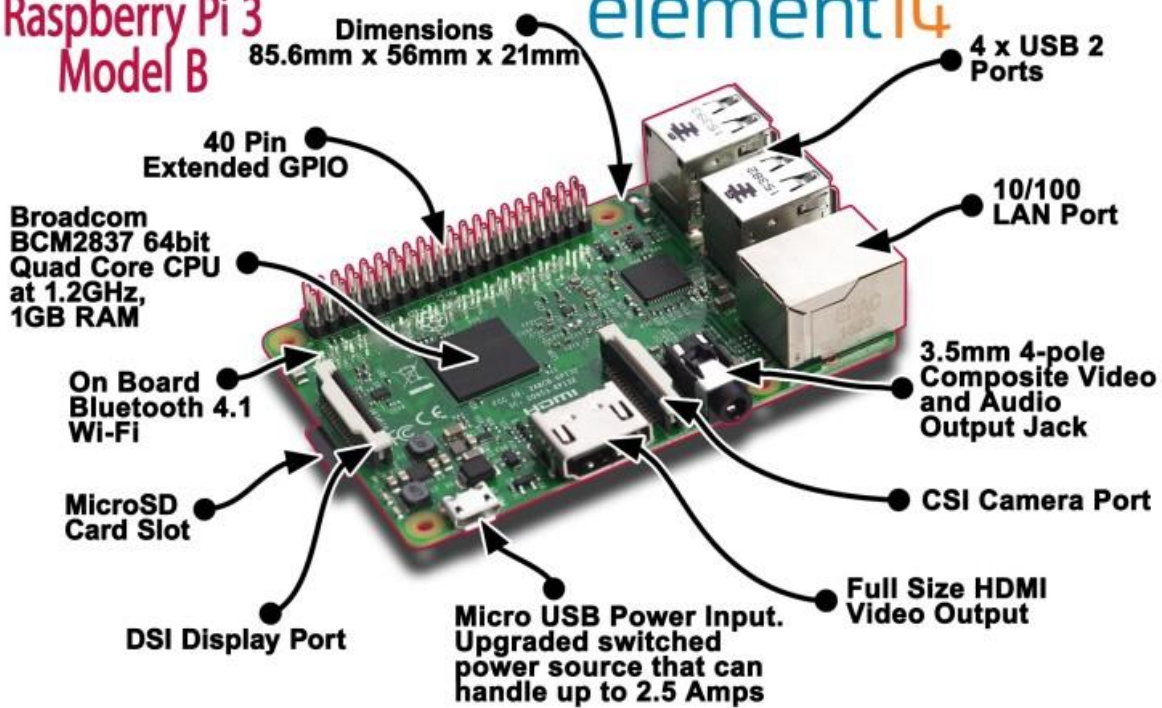
جدول ۱-۰: مشخصات فنی PI 3 MODEL B

Broadcom BCM2387 chipset 1.2GHz Quad-Core ARM Cortex-A53	پردازنده
VideoCore IV® Multimedia Co-Processor دو هسته ای	پردازنده گرافیکی
4.1 LE	بلوتوث
دارد	وای فای
۱ گیگابایت LPDDR2	حافظه
لینوکس - Windows 10 IoT (سیستم عامل از کارت Micro SD بوت می شود)	سیستم عامل
۸۵ × ۵۶ × ۱۷ میلی متر	ابعاد
سوکت میکرو یو اس بی 5V - 2.5A	تغذیه

BaseT ۱۰۰/۱۰	سوکت اترنت
۱.۴ & HDMI (rev 1.3)	خروجی ویدئو
(NTSC و PAL) RCA	کامپوزیت
HDMI جک ۳.۵ میلی متری،	خروجی صدا
USB 4 x USB 2.0	کانکتور USB
هدر برد ۴۰ پین با رابط GPIO 2.54، تعداد ۲۷ پین GPIO و همچنین پین های +۳.۳، +۵ ولت و GND	پین های ورودی و خروجی
۱۵ پین MIPI رابط سریال دوربین (CSI-2)	ورودی دوربین
رابط سریال (DSI) کانکتور کابل فلکس تخت ۱۵ جهته با دو دیتا	ورودی صفحه نمایش
یک خط کلاک	خطوط کلاک
اسلات Micro SDIO Push/pull	ورودی کارت حافظه

Raspberry Pi 3 Model B

element14



شکل ۲-۰: برد Pi 3 MODEL B در یک نگاه

۵-۰- تمرینات و تکالیف

۱- برنامه Win32 Disk و برنامه SD Card.Formatter را برای جلسه بعد دانلود کنید.

۲- سیستم عامل مورد نیاز برای Raspberry Pi را از لینک زیر دانلود کنید.

<https://www.raspberrypi.com/software/operating-systems/>

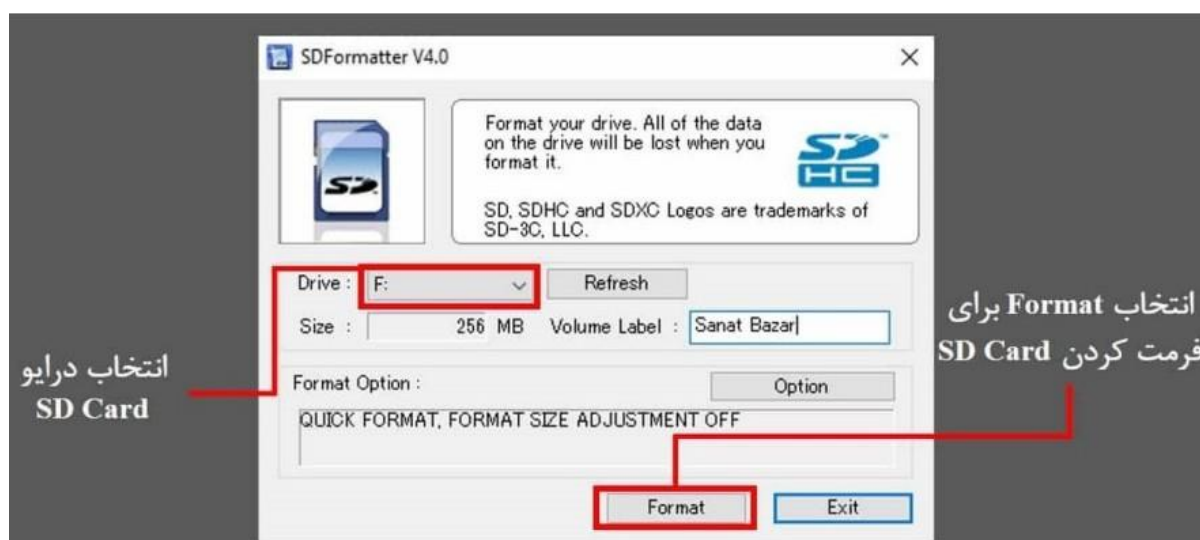
۱-۱- مراحل راه اندازی RASPBERRY PI 3B

۱-۱-۱- نصب سیستم عامل بر روی SD CARD

- برنامه SD Card.Formatter را از لینک زیر دانلود کنید.

<https://www.yasdl.com/144902/%D8%AF%D8%A7%D9%86%D9%84%D9%88%D8%AF-sd-card-formatter.html>

- فایل را از حالت زیپ استخراج کرده و نصب کنید.
- با استفاده از رم ریدر sd card (کارت حافظه باید حداقل 4 GB یا بیشتر ظرفیت داشته باشد) خود را وارد کامپیوتر نمایید.
- برنامه را باز کنید.



شکل ۱-۱: محیط برنامه SD CARD FORMATTER

- در قسمت select card ، sd card را انتخاب کنید.

- دکمه format را بزنید.

برنامه Win32.Disk را از لینک زیر دانلود کنید.

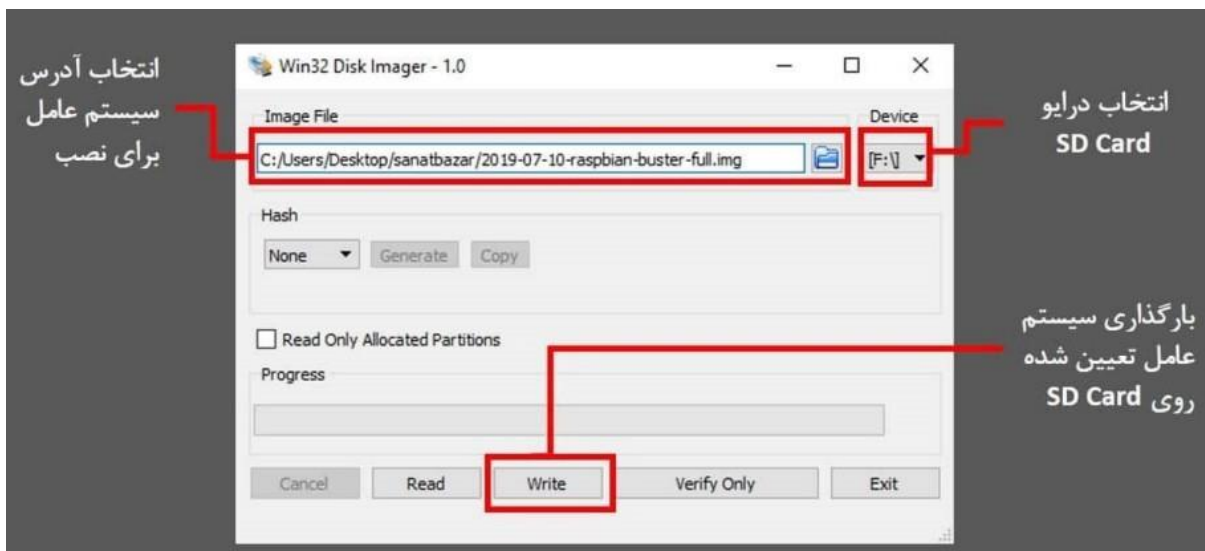
<https://www.yasdl.com/205052/%D8%AF%D8%A7%D9%86%D9%84%D9%88%D8%AF-win32-disk-imager.html>

- فایل را از حالت زیپ استخراج کرده و نصب کنید.

- سیستم عامل مورد نیاز برای Raspberry Pi را از لینک زیر دانلود کنید.

<https://www.raspberrypi.com/software/operating-systems/>

- برنامه Win32.Disk را باز کنید.



شکل ۱-۲: محیط برنامه WIN32.DISK

- فایل سیستم عاملی که دانلود کرده اید را در قسمت Image File

- روی گزینه write کلیک کنید.

- در صورت نیاز گزینه read برای گرفتن پشتیبانی از sd card می باشد.

- مانیتور

اکثر مانیتور ها به عنوان نمایشگر برای Pi Raspberry کار می کنند، اما باید از صفحه نمایش با ورودی HDMI پشتیبانی کند.

- کابل اتصال

برای اتصال مانیتور خود به Pi Raspberry به یک کابل HDMI نیاز داریم.

- صفحه کلید و ماوس

هر صفحه کلید و ماوس USB استاندارد با Pi Raspberry شما کار می کند، صفحه کلیدهای بیسیم و موس ها در صورت Pair شدن نیز قابل استفاده هستند

- منبع تغذیه

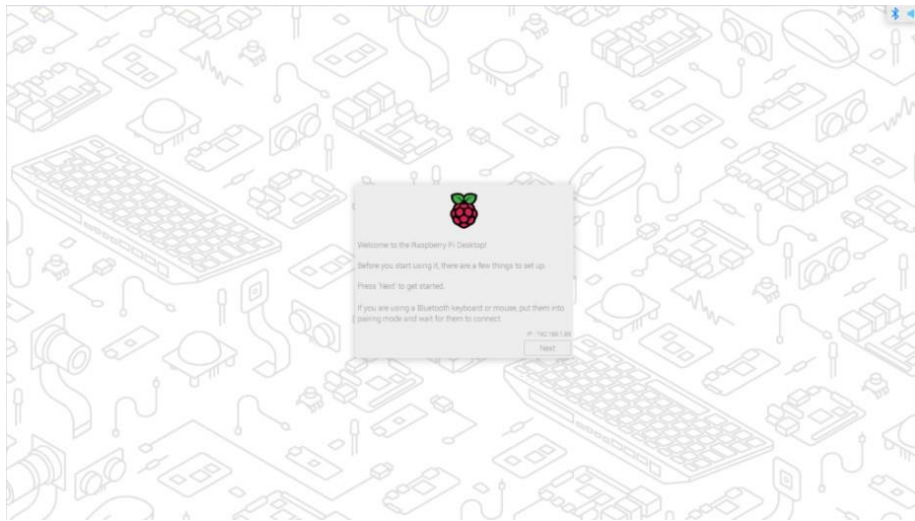
برای راه اندازی Pi Raspberry به یک منبع 5V - 2.5A نیاز داریم.

- کارت SD

به یک SD card Micro نیاز داریم که حداقل ۴ گیگابایت ظرفیت داشته باشد.

ادوات مورد اشاره در قسمت بالا را متصل کنید.

Raspberry Pi 3B را روشن نمایید.



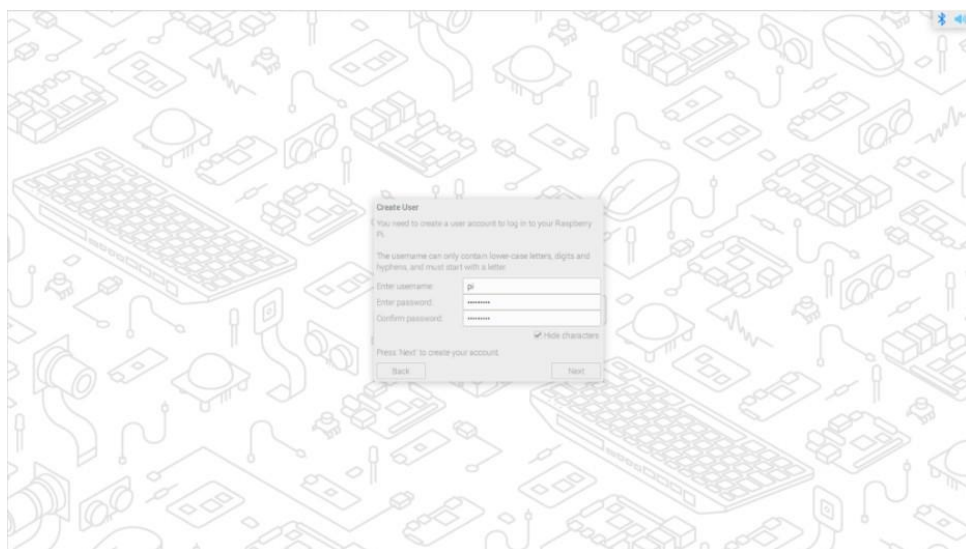
شکل ۱-۳: پنجره نصبی سیستم عامل RASPBERRY PI

پنجره نصبی سیستم عامل Raspberry Pi در اولین بوت اجرا می شود.



شکل ۱-۴: تنظیمات بین المللی و اطلاعات منطقه زمانی

پنجره نصبی سیستم عامل Raspberry Pi در اولین بوت اجرا می شود. پنجره نصبی به شما قابلیت هایی برای انجام تنظیمات بین المللی و اطلاعات منطقه زمانی خود شروع می دهد.



شکل ۱-۵: تنظیمات حساب کاربری

در این صفحه از شما خواسته می شود که یک حساب کاربری ایجاد کنید. در اینجا می توانید نام کاربری و رمز عبور خود را انتخاب کنید.



شکل ۱-۶: پیام اخطار استفاده از رمز عبور پیش فرض

با این حال، اگر بخواهید این حساب را ایجاد کنید، یک پیام اخطار فعال می‌شود و به شما توصیه می‌کند که از استفاده از رمز عبور پیش فرض قدیمی raspberry اجتناب کنید.



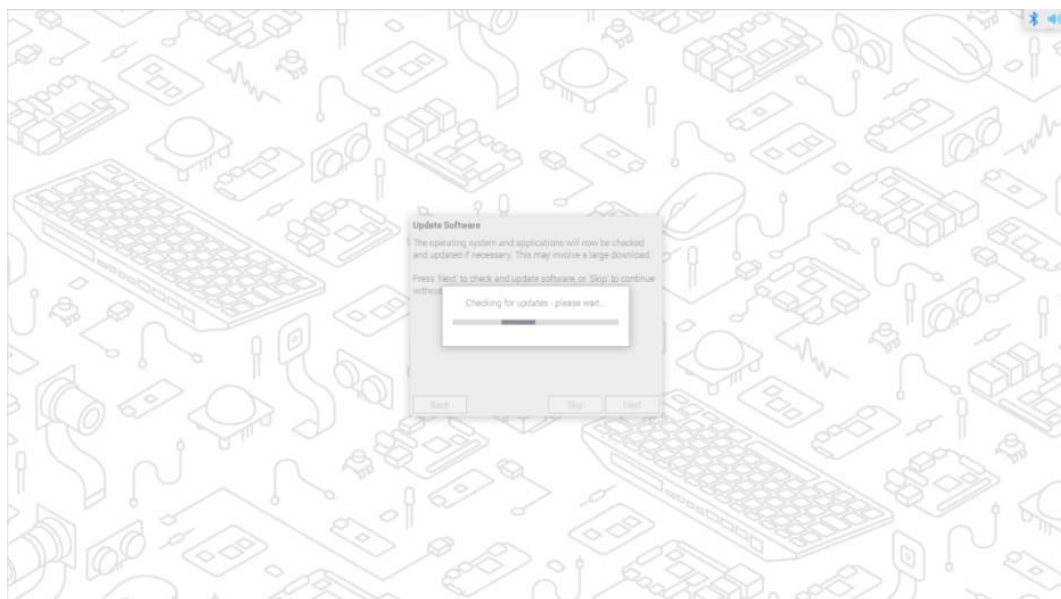
شکل ۱-۷: پیکربندی صفحه نمایش

پس از ایجاد یک حساب کاربری، می‌توانید صفحه نمایش خود را پیکربندی کنید.



شکل ۱-۸: تنظیمات شبکه بی سیم

شبکه بی سیم خود را اضافه کنید



شکل ۹-۱: بروز رسانی سیستم عامل

هنگامی که شبکه بی سیم شما پیکربندی شد و Raspberry Pi شما به اینترنت دسترسی پیدا کرد، از شما خواسته می شود که سیستم عامل را به آخرین نسخه خود به روز کنید. این به طور خودکار هر اصلاحیه و به روز رسانی مورد نیاز برای به روز رسانی سیستم عامل جدید شما را دانلود می کند.



شکل ۱-۱۰: پایان تنظیمات

پس از به روز رسانی سیستم عامل، از شما خواسته می شود تا Raspberry Pi خود را مجدداً راه اندازی کنید.

۲-۲- کنترل RASPBERRY PI از راه دور

برنامه vnc viewer را از لینک زیر دانلود کنید.

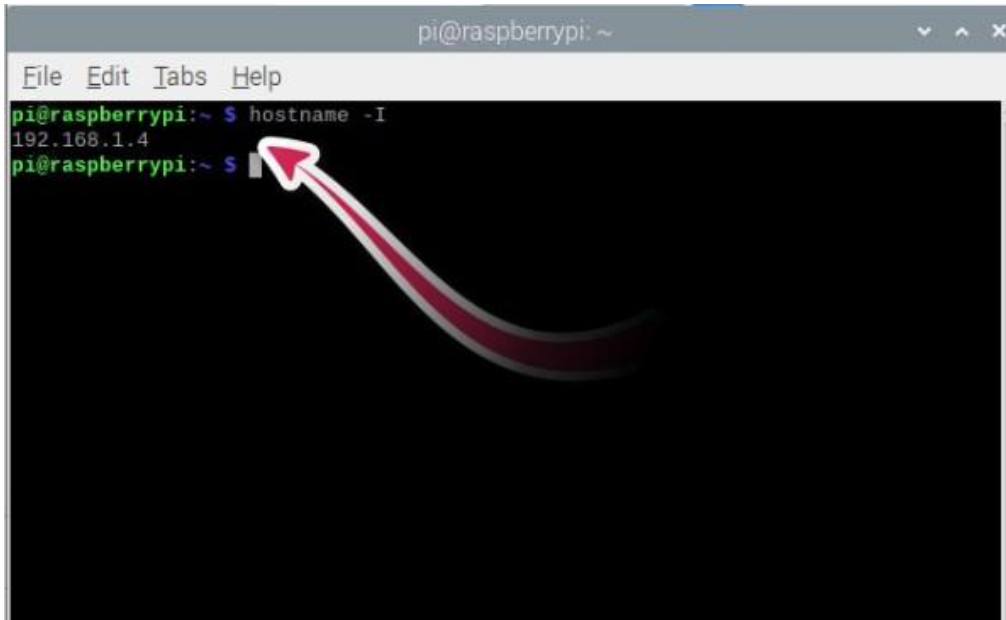
<https://soft98.ir/internet/remote-control/2334-realvnc.html>

فایل را از حالت زیپ درآورده و روی کامپیوتر خود نصب نمایید.

Raspberry Pi و کامپیوتر خود را به یک وای فای مشترک متصل نمایید.

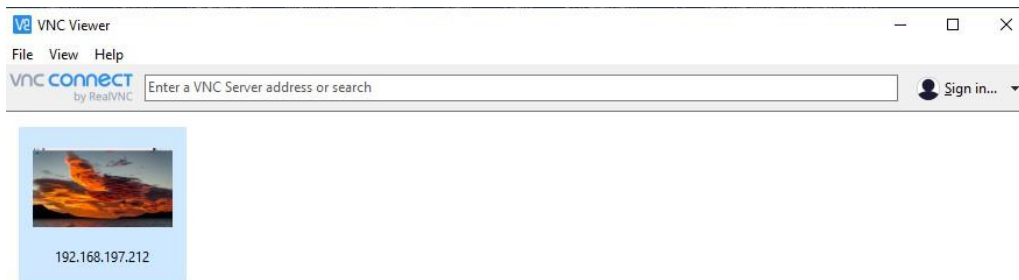
در قسمت ترمینال Raspberry Pi دستور hostname -I را اجرا کنید این دستور IP ، Raspberry Pi شما را نمایش

میدهد.



شکل ۱-۱۱: قسمت ترمینال RASPBERRY PI

برنامه vnc viewer را در کامپیوتر خود باز کنید.



شکل ۱-۱۲: محیط برنامه VNC VIEWER

IP به دست آمده از Raspberry Pi را در قسمت بالای صفحه تایپ کنید.

اکنون می توانید با کامپیوتر خود Raspberry Pi را کنترل کنید.

۱-۲- زبان برنامه نویسی و معرفی نرم افزار کد نویسی

۱-۱-۲- معرفی برنامه THONNY IDE

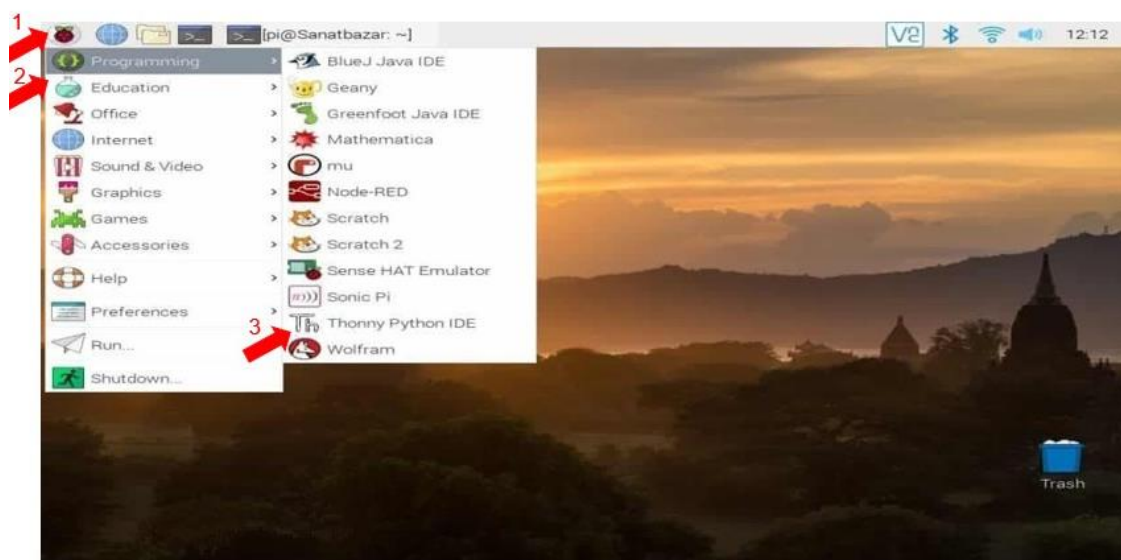
۱- در صفحه ی آغازین باز شده در نوار بالا روی لوگوی رزبری پای کلیک کنید.

۲- روی گزینه programming نگره دارید.

۳- روی لوگوی نرم افزار Thonny IDE کلیک کنید تا باز شود.

نکته در بعضی از نسخه های سیستم عامل رزبرین ممکن است نرم افزار Thonny IDE به طور پیش فرض نباشد

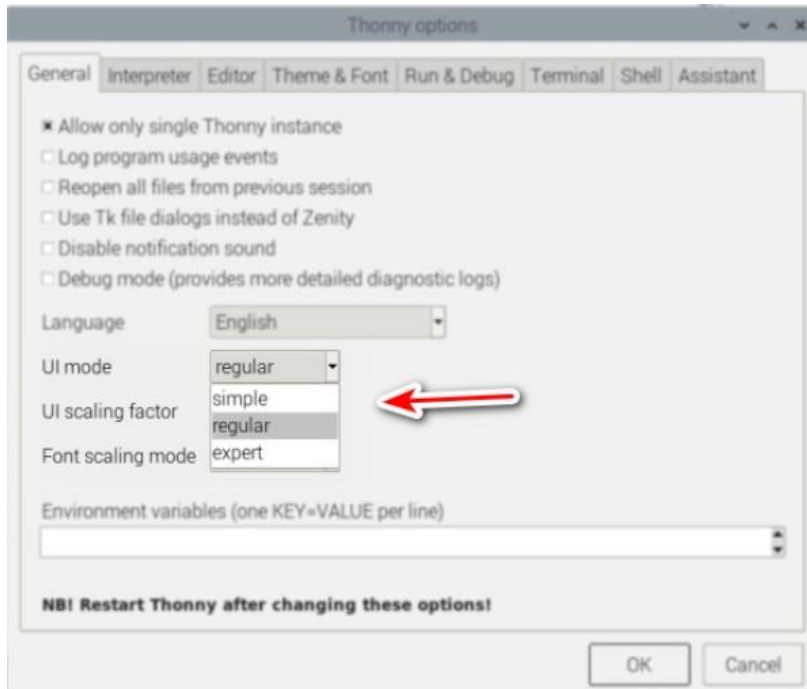
که در این صورت باید آن را نصب کنید.



شکل ۱-۲: باز کردن نرم افزار THONNY IDE

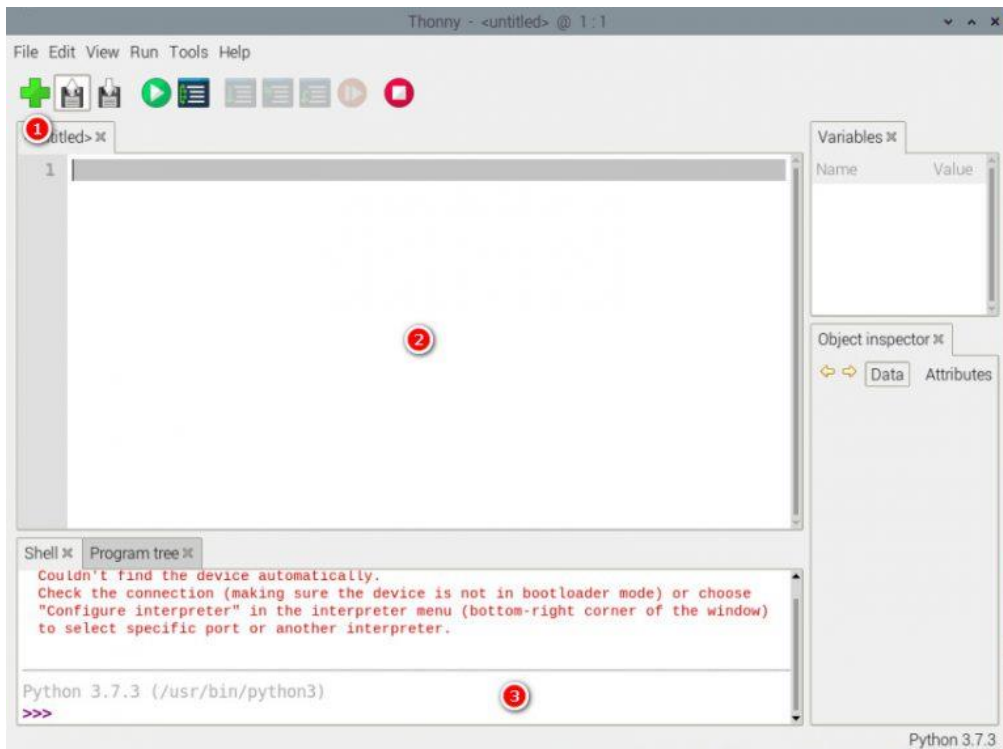
نرم افزار Thonny IDE از دو بخش تشکیل شده است:

Regular Mode و Simple Mode که پیشنهاد می شود در ابتدا از Simple Mode استفاده شود.



شکل ۲-۲: تنظیمات نرم افزار THONNY IDE

محیط نرم افزار تونی Thonny IDE همانند زیر است:



شکل ۲-۳: محیط نرم افزار THONNY IDE

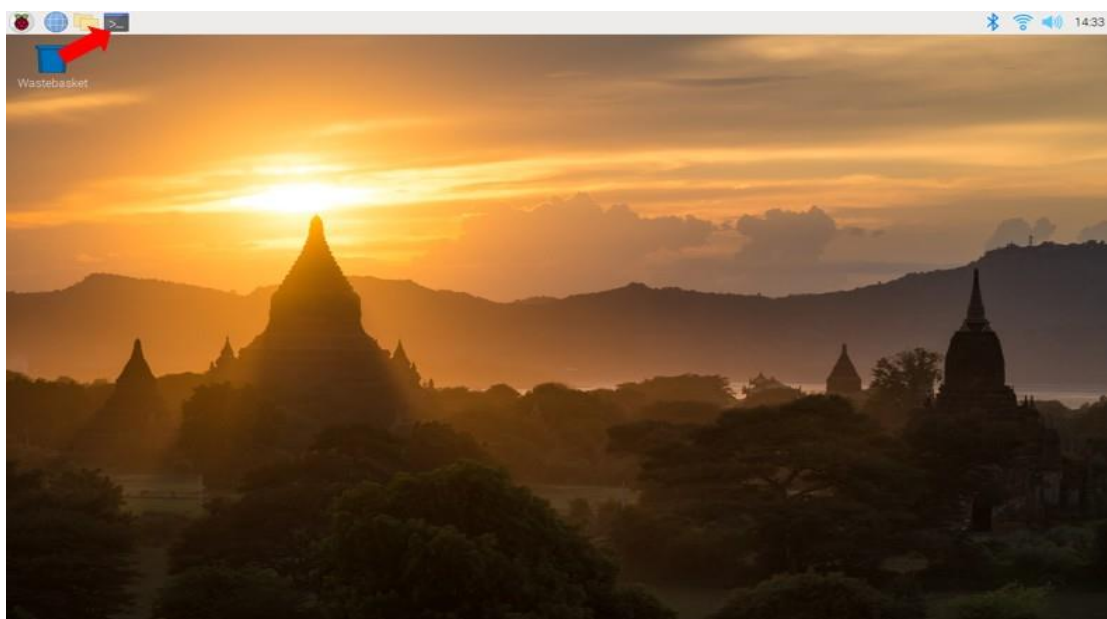
۱: **Toolbar** با انتخاب نوار ابزار میتوانید یک تب جدید برای برنامه نویسی باز کرده و سپس شروع به کدنویسی کنید.

۲: **Script Area** در این محیط برنامه شما نوشته می شود. هر خط از برنامه با عدد نمایش داده می شود و در صورت اشتباه در کدنویسی در بخش **Shell** خطی که ایراد دارد را نمایش میدهد.

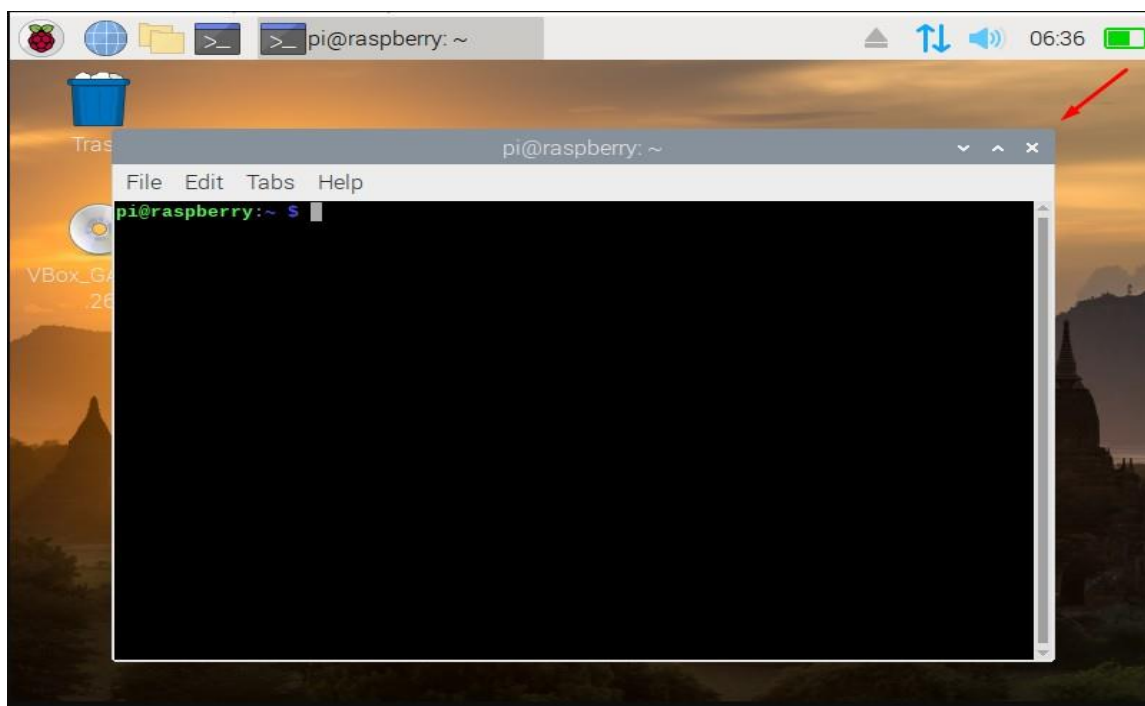
۳: **Python Shell** در این بخش میتوانید دستورات را به صورت مستقل تایپ و اجرا کنید. همچنین میتوانید دیتای برنامه نوشته شده را در این قسمت مشاهده کنید.

۲-۱-۲- معرفی ترمینال رزبری پای

۱- برای باز کردن ترمینال در رزبری پای، روی آیکون چهارم از سمت چپ در نوار بالا کلیک کنید.



شکل ۲-۴: آیکون ترمینال رزبری پای



شکل ۲-۵: ترمینال رزبری پای

- نکته: مزیت استفاده از ترمینال رزبری پای سرعت و دسترسی بیشتر نسبت به استفاده از محیط گرافیکی برای اجرای دستورات است.

- دستور `htop`: در اصل یک محیط ایتراکتیو و یا نیمه گرافیکی است، این دستور پس از اجرا تمامی

```

root@parsvds:~
Tasks: 60, 12 thr; 2 running
Load average: 0.01 0.03 0.05
Uptime: 2 days, 13:50:41
Mem[|||||] 250M/1.79G
Swp[ ] 0K/31.0M

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
8114 root 20 0 119M 1984 1424 R 0.7 0.1 0:00.06 htop
709 root 20 0 560M 17348 6036 S 0.7 0.9 0:20.77 /usr/bin/python2 -Es /u
732 root 20 0 305M 9596 5896 S 0.0 0.5 0:30.24 /usr/sbin/rsyslogd -n
1 root 20 0 43548 3788 2472 S 0.0 0.2 1:07.72 /usr/lib/systemd/system
374 root 20 0 39060 5240 4924 S 0.0 0.3 1:30.94 /usr/lib/systemd/system
403 root 20 0 45404 1780 1240 S 0.0 0.1 0:00.05 /usr/lib/systemd/system
462 root 20 0 21540 1212 968 S 0.0 0.1 0:04.63 /usr/sbin/irqbalance --
466 dbus 20 0 58460 2680 1720 S 0.0 0.1 1:06.34 /usr/bin/dbus-daemon --
469 chrony 20 0 115M 1788 1320 S 0.0 0.1 0:00.21 /usr/sbin/chronyd
498 polkitd 20 0 598M 10012 4600 S 0.0 0.5 0:00.00 /usr/lib/polkit-1/polki
501 polkitd 20 0 598M 10012 4600 S 0.0 0.5 0:13.96 /usr/lib/polkit-1/polki
502 polkitd 20 0 598M 10012 4600 S 0.0 0.5 0:00.00 /usr/lib/polkit-1/polki
503 polkitd 20 0 598M 10012 4600 S 0.0 0.5 0:00.00 /usr/lib/polkit-1/polki
504 polkitd 20 0 598M 10012 4600 S 0.0 0.5 0:00.00 /usr/lib/polkit-1/polki
507 polkitd 20 0 598M 10012 4600 S 0.0 0.5 0:00.00 /usr/lib/polkit-1/polki
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
    
```

شکل ۲-۶: نمایی از صفحه گرافیکی HTOP

پردازش های سیستم شما را نمایش می دهد.

- دستور `ls`: به ما لیستی از فایل ها را نشان داد. اما این لیست آن قدرها هم کامل نیست. در واقع با این دستور ما فایل های `hidden` را مشاهده نمیکنیم. برای این که فایل های مخفی شده همراه با برخی از جزییات آنها را مشاهده کنیم باید از دستور `ls -la` استفاده کنیم. این دستور به ما لیستی از تمام فایل ها همراه با برخی از جزییات آنها را به ما نشان می دهد.
- دستور `ls..help`: راهنمایی های مربوط به دستور `ls` را نمایش می دهد.
- دستور `pwd`: دستور `pwd` مخفف عبارت `print working directory` است `pwd` یکی از پایه ای ترین و پرکاربردترین دستورات در لینوکس است. هنگامی که دستور `pwd` اجرا می شود، مسیر کامل دایرکتوری که در آن هستیم را نشان می دهد.

- دستور `cd` : دستور `cd` مخفف `change directory` هست و برای تغییر مسیر یا جابه‌جایی از دایرکتوری فعلی به دایرکتوری دیگر استفاده می‌شود.
 نحوه استفاده از این دستور : (دایرکتوری مورد نظر -فاصله - `cd`)
 دستور `cd desktop` : به دسکتاپ می‌رود.
 دستور `cd ..` به دایرکتوری قبلی برمی‌گردد.
 دستور `cd ../../` دو مرحله به دایرکتوری قبلی برمی‌گردد.
- دستور `mkdir` : این دستور دایرکتوری ایجاد می‌کند.
 نحوه استفاده از این دستور : ابتدا وارد دایرکتوری که می‌خواهید در آن دایرکتوری جدید ایجاد کنید شوید سپس دستور مقابل را اجرا کنید. (نام دایرکتوری که می‌خواهید ایجاد کنید - فاصله - `mkdir`)
- دستور `rmkdir` : این دستور دایرکتوری را حذف می‌کند.
 نحوه استفاده از این دستور : ابتدا وارد دایرکتوری که می‌خواهید در آن دایرکتوری را حذف کنید شوید سپس دستور مقابل را اجرا کنید. (نام دایرکتوری که می‌خواهید حذف کنید - فاصله - `rmkdir`)
 برای حذف برخی از فایل‌ها که دایرکتوری نیستند از دستور `rm` استفاده می‌کنیم.
- دستور `python -version` : این دستور ورژن پایتون سیستم را نشان می‌دهد.
- دستور `nano pi.py` برنامه نانو را برای اجرا در پایتون فرا می‌خواند.
 با دکمه `control` و `x` و سپس `y` و در نهایت `enter` می‌توان از این برنامه خارج شد.
- دستور `python` : برنامه پایتون را اجرا می‌کند.
 با دستور `quit()` می‌توان از پایتون خارج شد.
- با دستور `python (file name).py` فایل `(file name).py` که دارای کدهای پایتونی است اجرا می‌شود.

۲-۲- تمرینات و تکالیف

۱- درباره مفهوم شی گرایی در پایتون تحقیق کنید.

آزمایش ۳

۳-۱- آشنایی با زبان برنامه نویسی پایتون جلسه اول

۳-۱-۱- انواع متغیرها

- متغیر bool : متغیر Boolean در پایتون مانند سایر زبان های برنامه نویسی دارای مقادیر true و false می باشد و ارزش عبارات را بر همین اساس بررسی می کنند. این متغیرها در بسیاری از عبارات شرطی مورد استفاده قرار گرفته و تصمیم گیری را بر اساس درست یا غلط بودن شرط انجام می دهند.
- متغیر int : مخفف کلمه integer هست. این نوع برای متغیرهای عددی حاوی عدد صحیح استفاده می شود. فقط اعداد منفی و مثبت کامل (بدون نقطه اعشار) را می توان در متغیر نوع int ذخیره کرد.
- متغیر float : این نوع متغیر اعداد اعشاری و کسری را نیز علاوه بر اعداد صحیح شامل می شوند.
- متغیر char یا string : در نوع داده ای کاراکترهای یونی کد (Unicode) ، حروف و کلمات قرار می گیرند. نوع داده هایی که به صورت کلمه و جمله هستند ، «string» و داده هایی که از حروف تشکیل شده اند نیز «char» گفته می شود. char ، یک حرف و String مجموعه ای از حروف یا کلمات است. در نوع char تنها امکان ذخیره سازی یک کاراکتر واحد وجود دارد.
- متغیر list : لیست در پایتون نوعی از داده است که میتواند همزمان چند آبجکت را در خود ذخیره کند. لیست پایتون مشابه آرایه در زبان های برنامه نویسی دیگر است اما با انعطاف پذیری بیشتر لیست ها در پایتون با استفاده از یک جفت براکت ایجاد میشوند. آبجکت هایی که در لیست ها ذخیره میشوند با نام آیتم (item) شناخته میشوند list. های پایتون برای کار با آبجکت های مشابه عالی هستند.

- متغیر `dictionary`: دیکشنری در پایتون، مجموعه‌ای بدون ترتیب از داده‌ها است که برای ذخیره‌ی مقادیر به شکل دو به دو استفاده می‌شود. ساختار کلی دیکشنری در پایتون، مانند دیکشنری‌های معمولی است؛ به این شکل که دیکشنری در پایتون به صورت کلید : مقدار (`key : value`) تعریف می‌شود، که در واقع کلید مشابه کلمات، و مقدار مشابه توضیحات آن کلمه در دیکشنری است.
- متغیر `tuple`: تاپل `Tuple` یک مجموعه (`Collection`) از داده‌ها است که مرتب و غیر قابل تغییر است. که با پرانتز `()` ساخته می‌شود.

```

x1=True
print(type(x1))
x2=10
print(type(x2))
x3=10.5
print(type(x3))
x4="amin"
print(type(x4))
x5= ["apple", 5, "cherry"]
print(type(x5))
x6={'name': 'John', 1: [2, 4, 3]}
print(type(x6))
x7=('foo', 'bar', 'baz', 3, 'quux', 'corge')
print(type(x7))

```

```

<class 'bool'>
<class 'int'>
<class 'float'>
<class 'str'>
<class 'list'>
<class 'dict'>
<class 'tuple'>

```

شکل ۳-۱: چند نوع متغیر در زبان پایتون

۳-۱-۲- نمونه توابع پایتون

- توابع در پایتون:

توابع در پایتون برای استفاده از کد در بیش از یک مکان از برنامه بکار می‌روند، و گاهی متد (`method`) یا رویه (`procedure`) نیز نامیده می‌شوند. پایتون توابع داخلی بسیاری از جمله `print()` را برای گردآوری کرده، اما همچنین این امکان را فراهم ساخته تا توابع دلخواه خود را ایجاد کنیم.

- تابع dir() در پایتون:

به دست آوردن محتوای یک شیء، شامل لیست نام همه ی ویژگی ها و متد های آن شیء می باشد.

```
class person:
    name = "python"
    age = 20
    country = "iran"
    print(dir(person))
```

```
['_class_', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
```

شکل ۳-۲: مثال تابع DIR()

- تابع upper():

تمام حروف متن را به حرف بزرگ تبدیل می کند.

```
txt = "python"
x = txt.upper()
print(x)
```

```
PYTHON
```

شکل ۳-۳: مثال تابع UPPER()

- تابع lower() در پایتون:

```
txt = "PYTHON"
x = txt.lower()
print(x)
```

```
python
```

شکل ۳-۴: مثال تابع LOWER()

تمام حروف متن را به حرف کوچک تبدیل می کند.

- تابع count()

تعداد باری که یک آیتم مشخص در لیست آمده است را برمیگرداند.

```
▶ points = [1, 4, 2, 9, 7, 8, 9, 3, 1]
x = points.count(9)
print(x)
```

2

شکل ۳-۵: مثال تابع COUNT()

- دستور for در پایتون

حلقه for بخشی از کد را برای مجموعه ای از مقادیر تکرار می کند. براساس آنچه در توضیحات پایتون آمده است، حلقه for در پایتون کمی متفاوت تر از عملکرد آن در زبان هایی مانند جاوا اسکریپت یا C کار می کند. حلقه for یک متغیر تکرار کننده دارد که مقادیر لیست (یا آرایه یا رشته) ارائه شده را پیمایش کند و در هر تکرار کد بدنه حلقه برای مقدار متغیر اجرا می شود.

- تابع range() در پایتون

```
range(stop)
range(start, stop[, step])
```

شکل ۳-۶: معرفی تابع RANGE()

آرگومان start اولین مقدار در محدوده است، اگر تابع range فقط با یک آرگومان فراخوانی شود مثل range(stop) پایتون فرض می کند مقدار شروع محدوده برابر با ۰ است یعنی آرگومان start = 0. مقدار این آرگومان جز محدوده حساب می شود.

آرگومان stop حد بالای محدوده را تعیین می کند. نکته مهم این است که حد بالا جز محدوده حساب نمی شود.

آرگومان step اختیاری است.

```
for i in range(5):  
    print(i, end=" ",)
```

خروجی: 0, 1, 2, 3, 4,

شکل ۳-۷: مثال تابع RANGE()

- تابع time.time()

برای استفاده از این تابع ابتدا باید کتابخانه آن را وارد برنامه کنیم.

```
import time  
st = time.time()  
for i in range(10):  
    print(i)  
print('all')
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
all
```

شکل ۳-۸: مثال تابع TIME()

نکته : به منظور اجرای کد ها به صورت آنلاین می توانید از سایت google clab استفاده نمایید.

۲-۳- تمرینات و تکالیف

۱- برنامه ای بنویسید که یک عدد را از کاربر گرفته و توان ۲ آنرا نمایش دهد. در صورتیکه کاربر جای عدد

ورودی دیگری وارد کند، به جای دادن خطا پیام مناسبی به کاربر نشان داده و دوباره از کاربر ورودی

درخواست کند.

آزمایش ۴

۴-۱- آشنایی با زبان برنامه نویسی پایتون جلسه دوم

۴-۱-۱- نمونه توابع و دستورات در پایتون

- دسترسی به اعضای tuple()

در مثال زیر عنصر دوم تاپل را چاپ کرده ایم:

```
▶ tuple1 = ('a', 'b', 'c', 'd', 'f', 'g')  
print(tuple1[2])
```

⊖ c

شکل ۴-۱: مثال دسترسی به اعضای TUPLE()

- دسترسی به اعضای دیکشنری

می توان با فراخوانی نام اعضای دیکشنری درون براکت []، به آنها دسترسی پیدا کرد.

```
▶ my_dict = {'name': 'python', 'age': 26}  
print(my_dict['name'])
```

⊖ python

شکل ۴-۲: مثال دسترسی به اعضای دیکشنری

- دستور if

ساده‌ترین دستور تصمیم‌گیری در پایتون است. از این دستور برای اجرا شدن یک بخش از کد، در صورتی که یک شرط خاص یا مجموعه‌ای از شروط در آن برقرار باشد، استفاده می‌شود.

دستور if مقادیر بولین (boolean) را می‌پذیرد، لذا شرط پس از ارزیابی، درست یا غلط خواهد بود. در صورت درست بودن شرط، دستور یا دستورات وابسته به آن اجرا می‌شوند و در غیر این صورت برنامه از آن بلوک کد خارج خواهد شد. در پایتون، شرط را می‌توان هم داخل پرانتز () و هم بدون پرانتز نوشت.

همان‌گونه که می‌دانید، پایتون از تورفتگی برای شناسایی یک بلوک کد استفاده می‌کند.

```
i = 10
if (i > 15):
    print ("10 is less than 15")
print ("I am Not in if")
```

I am Not in if

شکل ۴-۳: مثال دستور IF

- دستور if - else

با دستور if در صورتی که شرط برقرار باشد، دستورات داده شده اجرا می‌شوند و در صورت نادرستی شرط، کاری انجام نمی‌شود. اما اگر بخواهیم در صورت نادرست بودن شرط یا شروط، دستورات دیگری اجرا شوند، از دستور else در کنار if استفاده می‌کنیم.

```
i = 20;
if (i < 15):
    print ("i is smaller than 15")
    print ("i'm in if Block")
else:
    print ("i is greater than 15")
    print ("i'm in else Block")
print ("i'm not in if and not in else Block")
```

i is greater than 15
i'm in else Block
i'm not in if and not in else Block

شکل ۴-۴: مثال دستور IF-ELSE

- دستور while

کاربرد حلقه ها در برنامه نویسی، تکرار یک بلوک خاص از کد های نوشته شده می باشد. حلقه while در پایتون برای تکرار روی یک بلوک کد تا زمانی که عبارت شرطی درست باشد استفاده می شود، ما معمولاً زمانی از این حلقه استفاده می کنیم که تعداد دفعات تکرار را از قبل ندانیم.

در پایتون، بدنه حلقه while از طریق تورفتگی تعیین می شود. بدنه با تورفتگی شروع می شود و اولین خط بدون فرورفتگی پایان را نشان می دهد. پایتون هر مقدار غیر صفر را True تفسیر می کند و صفر به عنوان False تفسیر می شوند.

```
▶ count = 0
while (count < 3):
    count = count + 1
    print("python")
```

```
python
python
python
```

شکل ۴-۵: مثال دستور WHILE

- دستور while-else

مانند حلقه‌های for، در حالی که حلقه‌ها می‌توانند یک بلوک else اختیاری نیز داشته باشند. قسمت

else در صورتی اجرا می‌شود که شرط در حلقه while به False ارزیابی شود.

حلقه while را می‌توان با یک دستور break خاتمه داد. در چنین مواردی، قسمت دیگری نادیده گرفته

می‌شود. بنابراین، اگر شکستی رخ ندهد و شرط نادرست باشد، قسمت else یک حلقه while اجرا

می‌شود.

```

▶ i = 0
  while i < 4:
    i += 1
    print(i)
  else: # Executed because no break in for
    print("No Break\n")
i = 0
while i < 4:
  i += 1
  print(i)
  break
else: # Not executed as there is a break
  print("No Break")

```

```

1
2
3
4
No Break

1

```

شکل ۴-۶: مثال تابع WHILE-ELSE

۴-۲- تمرینات و تکالیف

۱- انواع نام گذاری پین های رسیبری پای BCM و Board را بررسی کنید.

۲- تفاوت gpiozero با rpi.gpio را بیان کنید.

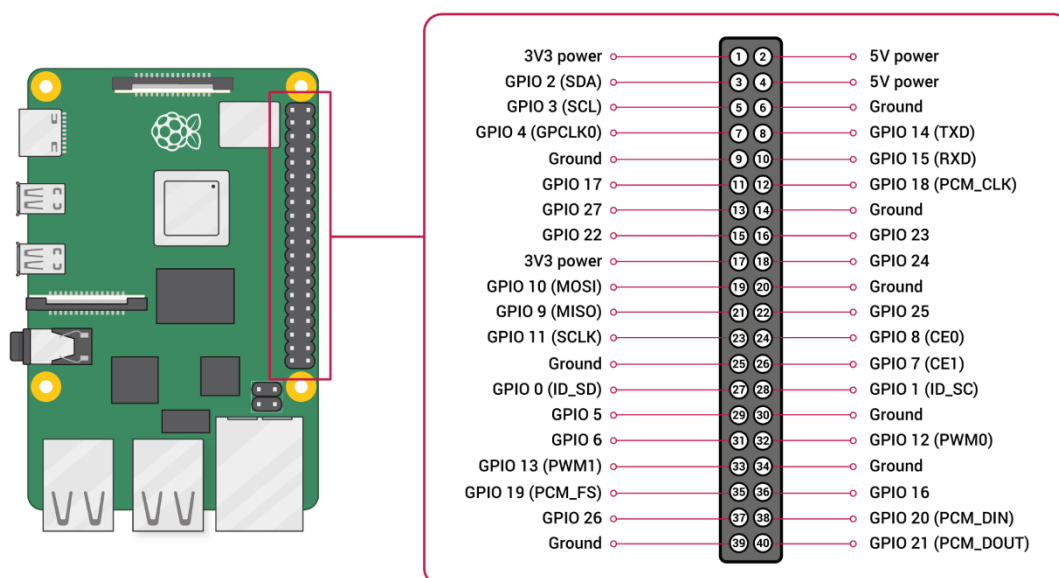
۳- مفهوم PWM توضیح دهید.

۱-۵- راه اندازی LED با برد رزبری پای

با استفاده از برد رزبری پای یک پروژه LED چشمک زن با شدت نور متغییر راه اندازی کنید.

چشمک زدن LED با اتصال یک LED به یکی از پین های GPIO برد رزبری پای انجام می شود.

۱-۱-۵- اتصالات مدار



شکل ۱-۵: پین های خروجی و ورودی رزبری پای

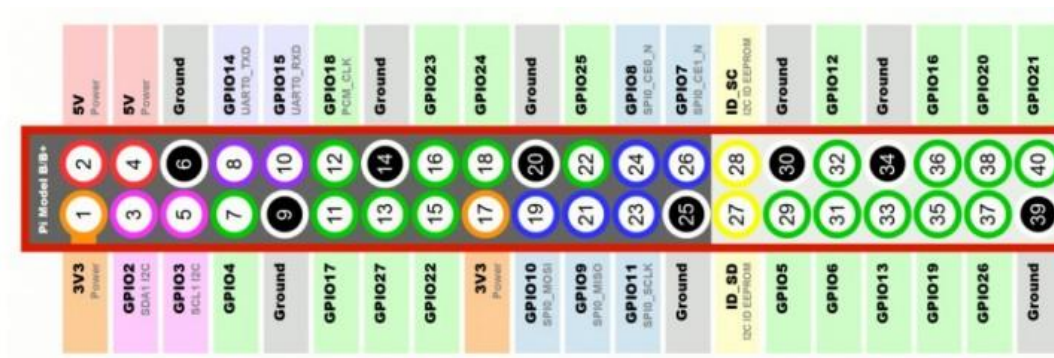
۱- پایه های GPIO برد رزبری پای

همانطور که در تصویر بالا نشان داده شده است ۴۰ پین خروجی برای رزبری پای وجود دارد در صورتی که از این

۴۰ پین، تنها ۲۶ پین GPIO هستند که میتوانند برنامه ریزی شوند. این پین ها از GPIO ۲ تا GPIO ۲۷ را شامل

می شوند .

این ۲۶ پین GPIO را می توان طبق نیاز برنامه ریزی کرد. برخی از این پین ها کارکردهای ویژه ای را انجام می دهند. بدون در نظر گرفتن GPIO های و یژه، ما ۱۷ عدد پین GPIO باقی مانده داریم. (پین های دایره سبز تصویر پایین)



شکل ۵-۲: تفکیک پین ها با رنگ های مختلف

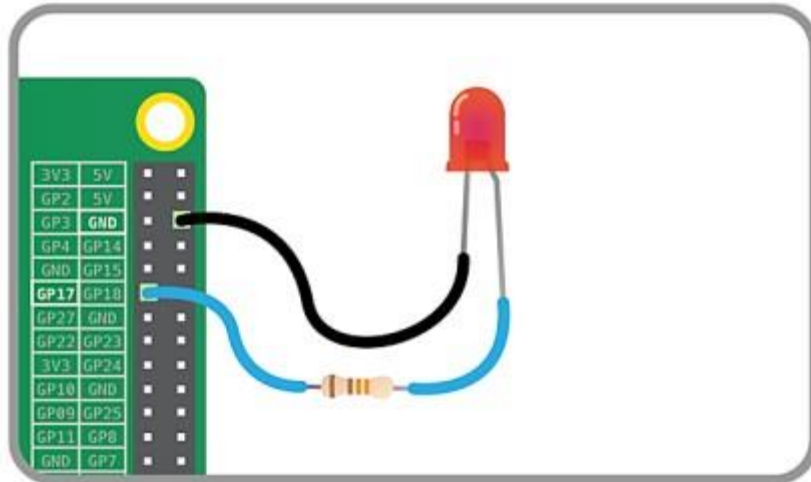
هر یک از این ۱۷ پین GPIO می توانند حداکثر جریان ۱۵ میلی آمپر را ارائه دهند. و مقدار جریان از تمام GPIO نمی تواند از ۵۰ میلی آمپر فراتر رود. بنابراین می توان از هر یک از این پین های GPIO حداکثر ۳ میلی آمپر را بکشیم. به منظور انجام پروژه یکی از این پین ها را به دلخواه انتخاب کنید.

۲- اتصال LED به رزبری پای

ابتدا به یک مقاومت ۱ کیلو اهم یا ۲۲۰ اهم و یک ال ای دی نیاز داریم.

LED ها دیودهای کوچک و ظریفی هستند و اگر جریان برق وارده به آن ها زیاد باشد می سوزند، برای محدود کردن عبور جریان الکتریکی از LED ها همیشه باید از یک قطعه مقاومت الکتریکی استفاده کنیم. پایه طولانی LED را به پین GPIO دلخواه (در اینجا به GPIO17 وصل کردیم) و پایه کوتاه را به پین GND رزبری پای وصل میکنیم.

نکته : مقاومت مورد نیاز می تواند بسته به ولتاژ LED متغیر باشد.

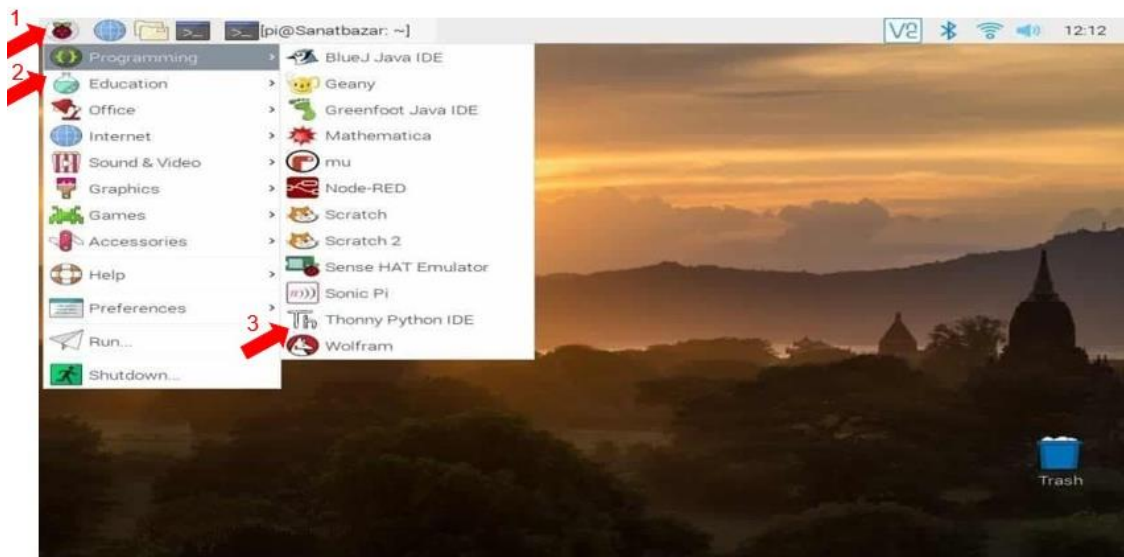


شکل ۳-۵: سیم بندی LED

۵-۱-۲- برنامه نویسی

۱- رزبری پای را راه اندازی کنید.

۲- منوی Programming را باز کنید و بر روی Thonny IDE کلیک نمایید .



شکل ۴-۵: نرم افزار THONNY IDE

۳- بخش اول : روشن و خاموش کردن LED

GPIO ZERO یک کتابخانه پایتون جدید است که بسیاری از قطعات GPIO از قبل در آن تعریف شده‌اند. این کتابخانه به صورت پیشفرض بر روی سیستم عامل PI نصب است.

ابتدا کتابخانه GPIO ZERO را ایمپورت کنید و برای رزبری پای مشخص کنید از کدام یک از پین‌های GPIO استفاده می‌کنید، در اینجا ما از پین ۱۷ استفاده کرده‌ایم.

```
Line [1]: from gpiozero import LED
```

```
Line [2]: led = LED(17)
```

برای روشن کردن LED، فرمان مقابل را تایپ و کلید Enter را فشار دهید:

```
Line [3]: led.on()
```

برای خاموش کردن LED، فرمان مقابل را تایپ کنید:

```
Line [4]: led.off()
```

با اجرای این فرمان‌ها، LED شما باید روشن و سپس خاموش شود.

۴- بخش دوم : چشمک زدن LED

به کمک کتابخانه `time` و یک حلقه کوچک می‌توانید کاری کنید که LED چشمک بزند.

با کلیک کردن بر روی `New` یک فایل جدید بسازید یا کدهای قبلی را کامنت کنید.

با کلیک کردن بر روی گزینه `Save` فایل جدید را ذخیره کنید. فایل را با نام دلخواه ذخیره کنید.

کد مقابل را وارد کنید:

```
Line [1]: from gpiozero import LED
Line [2]: from time import sleep
Line [3]: led = LED(17)
Line [4]: while True:
Line [5]:     led.on()
Line [6]:     sleep(1)
Line [7]:     led.off()
Line [8]:     sleep(1)
```

فایل را ذخیره کنید و با کلیک کردن بر روی گزینه Run کد را اجرا کنید.

در این حالت LED خاموش و روشن می‌شود. برای خارج شدن از برنامه بر روی گزینه Stop کلیک کنید.

۵- بخش سوم : کنترل روشنایی LED

با کلیک کردن بر روی New یک فایل جدید بسازید یا کدهای قبلی را کامنت کنید.

ما میخواهیم پرونده GPIO را از کتابخانه وارد کنیم. کد زیر باعث میشود بتوانیم پین های GPIO برد رزبری پای را برنامه ریزی کنیم. همچنین مانند قسمت قبلی آموزش، نام GPIO را به IO تغییر میدهیم. و هر بار خواهیم به پین های GPIO مراجعه کنیم از کلمه IO استفاده میکنیم.

```
import RPi.GPIO as GPIO
```

بعضی اوقات، پین های GPIO که سعی در استفاده از آنها داریم، ممکن است عملکردهای دیگری را انجام دهند. در

این حالت، هنگام اجرای برنامه، هشدارهایی دریافت خواهیم کرد. دستور زیر به PI می گوید هشدارها را نادیده

گرفته و به برنامه ادامه دهد.

```
GPIO.setwarnings(False)
```

کد زیر برای تنظیم کانال و فرکانس سیگنال PWM است. در اینجا متغیر p است که میتواند هر چیزی باشد. ما از GPIO17 به عنوان کانال خروجی PWM استفاده می کنیم. فرکانس سیگنال پی وی ام 100 انتخاب شده است. زیرا نمی خواهیم شاهد چشمک زدن LED باشیم.

```
my_pwm-GPIO.PWM (output channel , frequency of PWM signal)
```

از دستور زیر برای شروع تولید سیگنال PWM استفاده میشود 'DUTYCYCLE'. برای تنظیم نسبت روشن به خاموش است. ۰ بدین معنی است که LED برای ۰ درصد از زمان روشن می شود، ۳۰ به معنی LED برای ۳۰٪ زمان و ۱۰۰ یعنی کاملاً روشن خواهد شد.

```
my_pwm.start (DUTYCYCLE)
```

کد مقابل را وارد کنید:

```
Line [1]: import RP1.GPIO as GPIO
Line [2]: #GPIO.setmode(GPIO.BCM)
Line [3]: GPIO.setmode(GPIO.BOARD)
Line [4]: GPIO.setwarnings(False)
Line [5]: GPIO.setup(11,GPIO.OUT)#17
Line [6]: my_pwm-GPIO.PWM(11,100)
Line [7]: my_pwm.start(100)
```

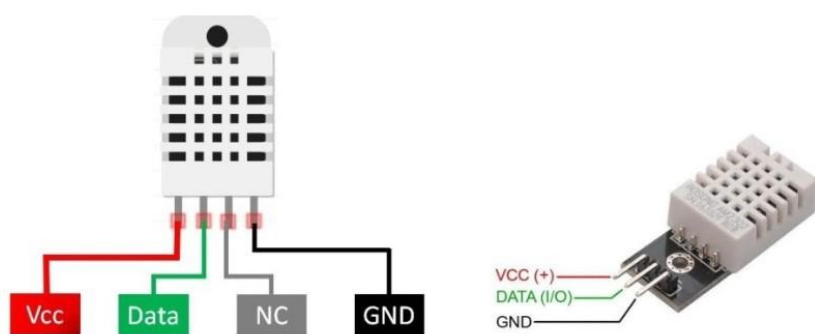
۵-۲- تمرینات و تکالیف

۱- در باره طرز کارکرد سنسور التراسونیک مدل HC SR04 را با جزئیات کافی جهت پیاده سازی با رزبری پای تحقیق کنید.

۶-۱- راه اندازی سنسور DHT 11

- سنسور DHT11 مقادیر رطوبت و دما را اندازه‌گیری کرده و به صورت سریال روی یک خط مهیا می‌سازد.
- این سنسور می‌تواند رطوبت را به صورت درصدی (۲۰ تا ۹۰ درصد RH) اندازه‌گیری کرده و دما را در محدوده ۰ تا ۵۰ درجه سلسیوس اندازه‌گیری کند.
- سنسور دارای ۴ پین است؛ که یکی از این پین‌ها برای ارسال داده به شکل سریال به کار می‌رود.
- پالس‌های TON و TOFF به عنوان منطق ۱ یا منطق ۰ یا پالس شروع و پایان فریم دیکود می‌شوند.

۶-۱-۱- اتصالات مدار



شکل ۶-۱: پایه‌های سنسور DHT22

۱- تعریف پایه‌های سنسور DHT11

۲- نحوه اتصال بین سنسور DHT22 و رزبری پای

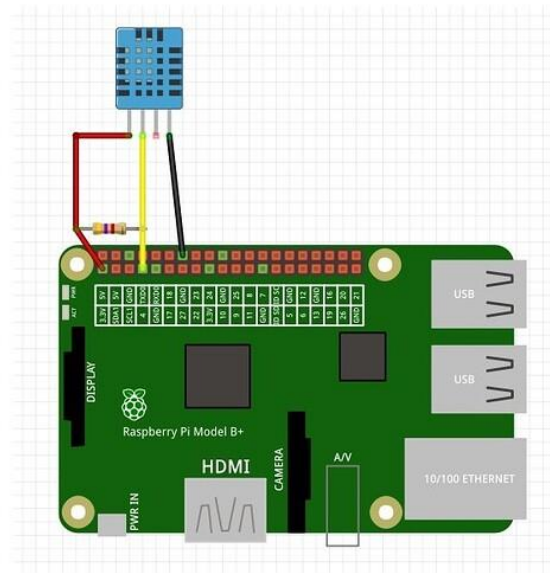
همانند تصویر زیر برای اتصال سنسور DHT22 و رزبری پای مراحل زیر را دنبال می کنیم:

۱- اتصال پین مثبت سنسور به پین ۳.۳ ولت برد رزبری پای

۲- اتصال پین منفی سنسور به پین GND برد رزبری پای

۳- اتصال پین دیتا سنسور به یکی از پین های I/O رزبری پای (که به GPIO4 متصل است)

۴- قرار دادن مقاومت ۱۰ کیلو بین پین دیتا و پین مثبت



شکل ۶-۲: سیم بندی سنسور DHT11

۶-۱-۲- برنامه نویسی

در این مثال از کتابخانه‌ی پایتون سنسور DHT، نوشته Adafruit در گیت هاب استفاده می کنیم. کتابخانه‌ی Adafruit

سنسور DHT، برای خواندن رطوبت و دما روی رزبری پای یا Beaglebone Black نوشته شده است. این کتابخانه

برای سنسورهایی مانند DHT22، DHT11 یا AM2302 قابل استفاده می باشد.

کتابخانه ی Adafruit سنسور DHT را از لینک زیر دانلود کنید.

https://github.com/adafruit/Adafruit_Python_DHT/archive/master.zip

کتابخانه را از حالت فشرده خارج کرده و با وارد کردن کد زیر در ترمینال رزبری پای، آن را در همان دایرکتوری که دانلود شده نصب کنید.

```
sudo python setup.py install
```

برنامه Thonny IDE را باز کرده و کد زیر را وارد نمایید.

```
import Adafruit_DHT

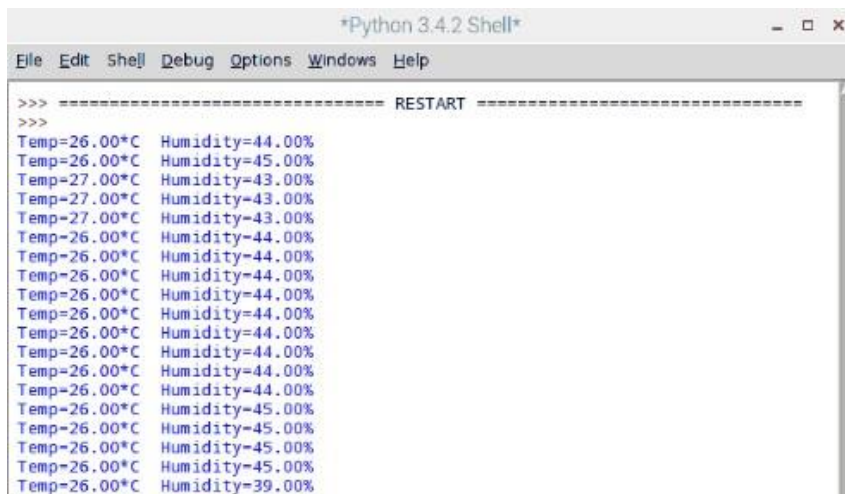
sensor = Adafruit_DHT.DHT11

pin = 4

while True:
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature, humidity))
```

شکل ۶-۳: کد راه اندازی سنسور دما و رطوبت

خروجی به صورت زیر خواهد بود.



```
*Python 3.4.2 Shell*
File Edit Shell Debug Options Windows Help
>>> ===== RESTART =====
>>>
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=45.00%
Temp=27.00*C Humidity=43.00%
Temp=27.00*C Humidity=43.00%
Temp=27.00*C Humidity=43.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=44.00%
Temp=26.00*C Humidity=45.00%
Temp=26.00*C Humidity=45.00%
Temp=26.00*C Humidity=45.00%
Temp=26.00*C Humidity=45.00%
Temp=26.00*C Humidity=39.00%
```

شکل ۶-۴: پنجره خروجی کد سنسور دما و رطوبت

۶-۲- راه اندازی سنسور اولتراسونیک

سنسورهای اولتراسونیک موجی در فرکانس معمولاً بین ۲۳ کیلوهرتز و ۴۰ کیلوهرتز را منتشر می کنند که بسیار بیشتر از محدوده شنیداری معمولی انسان در فرکانس ۲۰ کیلوهرتز است، از این رو اصطلاح اولتراسونیک برای این موج در نظر گرفته شده است.

سنسورهای التراسونیک با استفاده از این موج، مدت زمانی را که طول می کشد تا موج از یک شیء خارج شود، اندازه می گیرند ، نحوه کار این سنسور ها بر اساس اصول اولیه اکو شدن صدا است. این سیستم را خفاش ها دارند و از آن برای یافتن طعمه خود استفاده میکنند . از آنجایی که سرعت صوت در هوا در دمای اتاق ۳۴۳ متر در ثانیه است، این زمان را میتوان به راحتی به مسافت تبدیل کرد.

۶-۲-۱- نحوه کار سنسورهای التراسونیک

این سنسور جزء سنسورهای مجاورتی است که به وسیله امواج صوتی فرکانس بالا، یک پالس کوتاه ارسال می کند. این پالس با برخورد با جسم مورد نظر به سمت سنسور باز می گردد و سنسور با استفاده از زمان رفت و برگشت موج می تواند فاصله جسم مورد نظر را به صورت کامل و دقیق محاسبه کند.

سنسورهای التراسونیک به بو، نور و رنگ حساس نیستند و این پارامترها نمی توانند خللی در کارکرد آن ها ایجاد نمایند .

- معرفی پایه های سنسور التراسونیک

سنسوری که ما قصد راه اندازی اون رو داریم ، سنسور SR-۰۴ هست که این سنسور دارای ۴ پایه می باشد.



شکل ۶-۵: پایه های سنسور التراسونیک

- سیم بندی مدار

اتصال VCC به پین شماره ۲ (5 V)

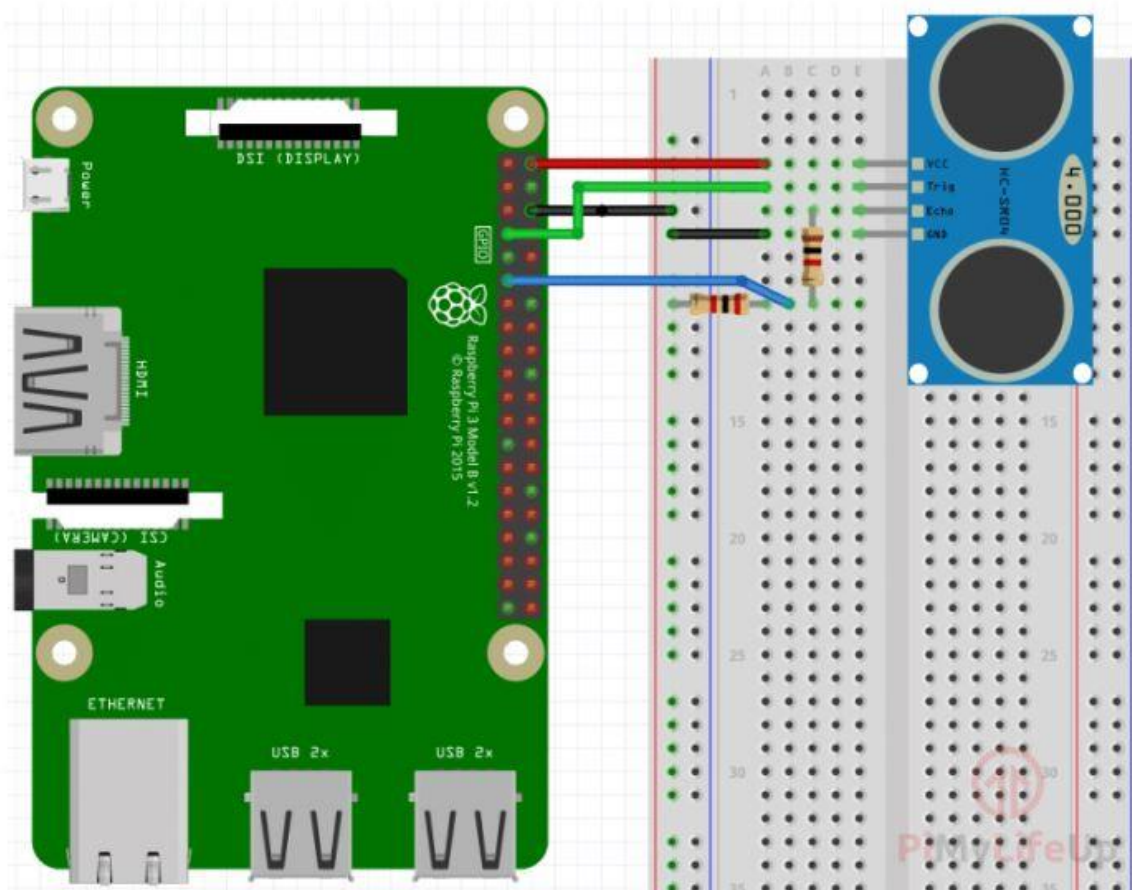
اتصال Trig به پین شماره ۷ (GPIO 4)

اتصال Echo به مقاومت $1K \Omega$ (R1)

اتصال مقاومت $2K \Omega$ (R2) بین R1 و زمین

اتصال نقطه‌ی میانی R1 و R2 به پین شماره ۱۱

اتصال GND به پین شماره ۶ (Ground)



شکل ۶-۶: سیم بندی سنسور التراسونیک

۶-۲-۳- برنامه نویسی

در ابتدا باید ماژول های time و RPi.GPIO را اضافه کنیم تا بتوانیم از GPIO استفاده کنیم و طول پالس را اندازه گیری کنیم. علاوه بر آن نوع شماره بندی پایه ها را نیز در خط سوم مشخص کرده ایم.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
```


در گام بعد برای پایه های مربوط به Trig و ECHO یک اسم خاص مشخص میکنیم تا در ادامه کد به راحتی بتوانیم شماره پایه ها را عوض کنیم. این نام ها TRIG و ECHO نام گرفته اند .

```
TRIG = 7
```

```
ECHO = 11
```

سپس یک پیغام برای مشخص تر شدن کار در خروجی نمایش می دهیم :

```
print ("Distance Measurement In Progress")
```

حال ورودی یا خروجی بودن پایه ها را مشخص میکنیم. میدانیم پایه مربوط به TRIG باید به عنوان خروجی تعریف شود و پایه مربوط به ECHO نیز باید به صورت ورودی تنظیم شود. این کار به کمک دو خط زیر انجام شده است .

```
GPIO.setup(TRIG,GPIO.OUT)
```

```
GPIO.setup(ECHO,GPIO.IN)
```

در گام بعد برای اطمینان ، پایه مربوط به ECHO را در حالت LOW قرار می دهیم و برای دو ثانیه صبر میکنیم .

```
GPIO.output(TRIG, False)
```

```
print ("Waiting For Sensor To Settle")
```

```
time.sleep(2)
```

سنسور SRF04 نیاز به یک پالس به طول ۱۰ uS دارد تا Trigger شود. پالس ارسالی از برد باعث می شود که سنسور ۸ پالس پشت سر با فرکانس ۴۰ کیلوهرتز (برای دریافت پاسخ از ECHO) ارسال کند. بنابراین ابتدا پایه TRIG را به مدت ۱۰ میکروثانیه در حالت HIGH قرار میدهیم و سپس دوباره آن را به حالت LOW میبریم.

کدهای زیر این کار را برای ما انجام می دهند :

```
GPIO.output(TRIG, True)
```

```
time.sleep(0.00001)
```

```
GPIO.output(TRIG, False)
```

پس از ارسال سیگنال ما نیاز داریم تا به پایه ورودی خود که به پایه ECHO سنسور وصل شده است ، گوش دهیم. سنسور پایه ECHO را برای مدتی در حالت HIGH قرار میدهد و دوباره آن را به حالت LOW می برد. بنابراین ما نیاز داریم تا مدت زمانی که پایه ECHO در حالت HIGH قرار دارد را اندازه گیری کنیم. برای این کار از حلقه while کمک میگیریم .

تابع time.time() آخرین زمان برای وقتی که شرط مربوط به while برقرار شده است را اندازه گیری می کند. به عنوان مثال اگر پایه از حالت LOW به حالت HIGH برود ، ما آخرین لحظه ای که پایه در حالت LOW هست رو به کمک خطوط زیر اندازه گیری میکنیم :

```
while GPIO.input(ECHO)==0:
```

```
    pulse_start = time.time ( )
```

توجه داشته باشید که هنوز پایه در حالت HIGH قرار نگرفته است و مقدار عبارت pulse_start به مرور زمان زیاد و زیادتر میشه و مقدار نهایی دقیقاً قبل از لحظه ای است که پایه به حالت HIGH برود .

حال وقتی که سیگنالی دریافت شد ، پایه به حالت HIGH میرود و ما نیاز داریم تا آخرین لحظه ای که این پایه در حالت HIGH است ، اندازه گیری شود. برای این حالت نیز مشابه حالت قبل عمل میکنیم ولی این بار شرط حلقه while در حالتی میگذاریم که پایه HIGH است. بنابراین به کمک این شرط نیز ما آخرین لحظه ای که این پایه در حالت HIGH است نیز اندازه گیری می شود .

```
GPIO.input(ECHO)==1:
```

```
    pulse_end = time.time ( )
```

حال میتوانیم مدت زمانی که پالس را در حالت HIGH بوده است را به کمک کد زیر محاسبه کنیم:

```
pulse_duration = pulse_end - pulse_start
```

فرمول محاسبه فاصله را مینویسیم.

```
distance = pulse_duration x 17150
```

برای زیبایی نیز فاصله بدست آمده را تا دو رقم اعشار گرد میکنیم .

```
distance = round(distance, 2)
```

سپس فاصله بدست آمده را در خروجی چاپ میکنیم .

```
print ("Distance:",distance,"cm")
```

در نهایت نیز نوع مدهای مربوط به پایه ها را پاک میکنیم. این کار به کمک دستور GPIO.Cleanup انجام می شود .

```
GPIO.cleanup( )
```

بنابراین کد نهایی به صورت زیر است ، برنامه Thonny IDE را باز کرده و کد زیر را وارد نمایید.

```
import RPi.GPIO as GPIO
```

```
import time
```

```
GPIO.setmode(GPIO. BOARD)
```

```
TRIG = 7
```

```
ECHO = 11
```

```
print ("Distance Measurement In Progress")
```

```
GPIO.setup(TRIG,GPIO.OUT)
```

```
GPIO.setup(ECHO,GPIO.IN)
```

```
GPIO.output(TRIG, False)
```

```
print ("Waiting For Sensor To Settle")
```

```
time.sleep(2)
```

```
GPIO.output(TRIG, True)
```

```
time.sleep(0.00001)
```

```
GPIO.output(TRIG, False)
```

```
while GPIO.input(ECHO)==0 :
```

```
pulse_start = time.time ( )
```

```
while GPIO.input(ECHO)==1 :
```

```
pulse_end = time.time ()  
pulse_duration = pulse_end - pulse_start  
distance = pulse_duration * 17150  
distance = round(distance, 2)  
print ("Distance:",distance,"cm")  
GPIO.cleanup()
```

خروجی به صورت زیر خواهد بود.

Distance Measurement In Progress

Waiting For Sensor To Settle

Distance: 29.69 cm

۶-۳- تمرینات و تکالیف

۱- طرز کارکرد و اتصال LCD 16*2 با استفاده از لینک زیر مطالعه نمایید.

<https://pimylifeup.com/raspberry-pi-lcd-16x2/>

۱-۷- راه اندازی LCD 16*2 کاراکتری

• LCD 16*2:

ال سی دی مورد استفاده در این پروژه LCD 16*2 می باشد و برای راه اندازی آن از مد ۴ بیت استفاده شده و آی

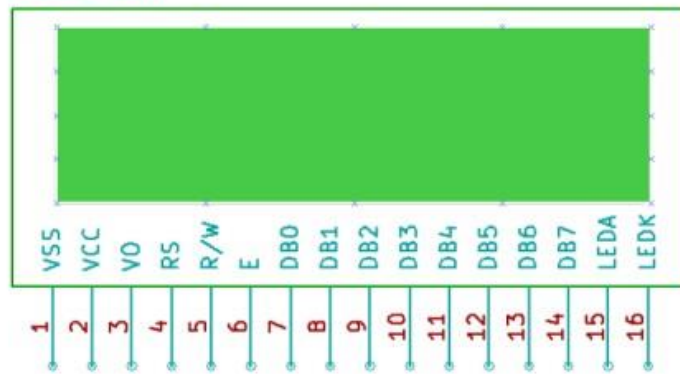
سی کنترلر این ال سی دی ۴۴۷۸۰ HD Hitachi می باشد.

در جدول زیر تمامی پایه های ال سی دی شرح داده شده است.

جدول ۱-۷ : پایه های ال سی دی

PIN NO	Symbol	Fuction
1	VSS	GND
2	VDD	+5V
3	V0	Contrast adjustment
4	RS	H/L Register select signal
5	R/W	H/L Read/Write signal
6	E	H/L Enable signal
7	DB0	H/L Data bus line
8	DB1	H/L Data bus line
9	DB2	H/L Data bus line
10	DB3	H/L Data bus line
11	DB4	H/L Data bus line
12	DB5	H/L Data bus line
13	DB6	H/L Data bus line
14	DB7	H/L Data bus line
15	A	+4.2v for LED
16	K	Power supply for BKL(0v)

LCD_2X16



شکل ۷-۱: LCD 16*2

۷-۱-۱- اتصالات

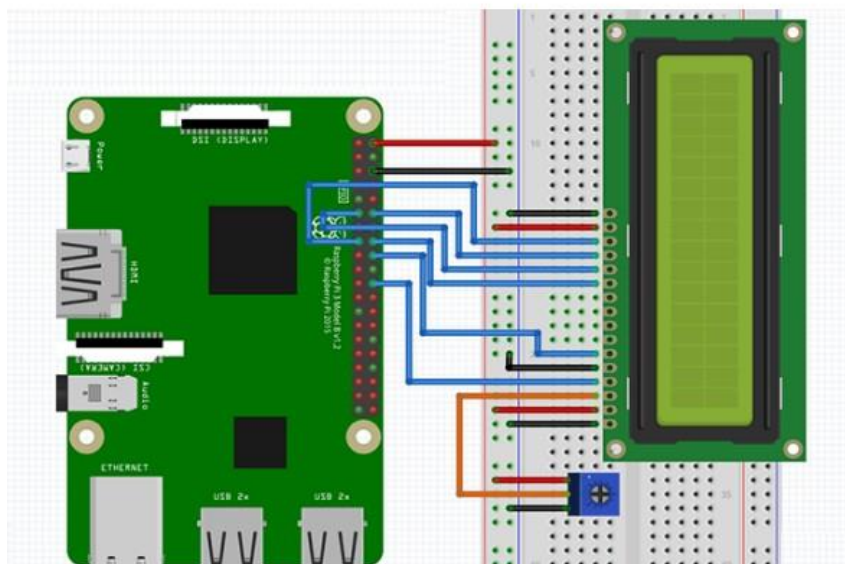
همانند تصویر زیر برای اتصال LCD 16*2 و رزبری پای مراحل زیر را دنبال می کنیم:

اتصال پین مثبت LCD 16*2 به تغذیه ۵ ولت برد رزبری پای (سیم قرمز)

اتصال پین منفی سنسور به پین GND برد رزبری پای (سیم مشکی)

اتصال پین های دیتا LCD 16*2

قرار دادن پتانسیومتر ۱ کیلو اهم جهت تنظیم میزان روشنایی



شکل ۷-۲: سیم بندی LCD 16*2

- 1-Pin 1 (Ground) goes to the ground rail.
- 2-Pin 2 (VCC/5v) goes to the positive rail.
- 3-Pin 3 (V0) goes to the middle wire of the potentiometer.
- 4-Pin 4 (RS) goes to GPIO25 (Pin 22)
- 5-Pin 5 (RW) goes to the ground rail.
- 6-Pin 6 (EN) goes to GPIO24 (Pin 18)
- 7-Pin 11 (D4) goes to GPIO23 (Pin 16)
- 8-Pin 12 (D5) goes to GPIO17 (Pin 11)
- 9-Pin 13 (D6) goes to GPIO18 (Pin 12)
- 10-Pin 14 (D7) goes to GPIO22 (Pin 15)
- 11-Pin 15 (LED +) goes to the positive rail.
- 12-Pin 16 (LED -) goes to the ground rail.

۷-۱-۲- برنامه نویسی

قبل از اسکریپت نویسی پایتون برای LCD 16X2 ابتدا باید از آپدیت بودن رزبری پای مورد استفاده مطمئن شویم.
بدین منظور پنجره ترمینال را باز کرده و دستورات زیر را جهت آپدیت سیستم عامل و نصب آن وارد می کنیم:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

در ادامه بعد از به روز رسانی و نصب آن باید python و (پکیج های کتابخانه استاندارد پایتون) pip را بر روی رزبری پای خود نصب کنیم. بدین منظور دستور زیر را در ترمینال وارد می کنیم:

```
sudo apt-get install python-dev python-pip
```

حال قبل از دانلود کتابخانه LCD 16X2 با وارد کردن دستور زیر در ترمینال از نصب بودن آخرین نسخه

پکیج های wheel, setuptools, و pip مطمئن می شویم:

```
sudo python -m pip install --upgrade pip setuptools wheel
```

اکنون با استفاده از دستور زیر کتابخانه پایتون برای ارتباط LCD 16X2 با رزبری پای را دانلود می‌کنیم:

```
git clone https://github.com/pimylifeup/Adafruit_Python_CharLCD.git
```

و با وارد کردن دستورات زیر کتابخانه پایتون مربوط به LCD 16X2 را نصب می‌کنیم:

```
cd ./Adafruit_Python_CharLCD
```

```
sudo python setup.py install
```

حال با استفاده از کد زیر از کتابخانه پایتون LCD 16X2 می‌توانیم در تمامی اسکریپت‌های پایتون استفاده کنیم:

```
import Adafruit_CharLCD as LCD
```

ارتباط با LCD16X2 در پایتون بسیار راحت می‌باشد. فقط کافی است ابتدا پین‌های متصل شده مربوطه به LCD

16X2 با رزبری را در ابتدای اسکریپت پایتون تعریف کنیم:

```
lcd_rs = 25
```

```
lcd_en = 24
```

```
lcd_d4 = 23
```

```
lcd_d5 = 17
```

```
lcd_d6 = 18
```

```
lcd_d7 = 22
```

```
lcd_backlight = 4
```

```
lcd_columns = 16
```

```
lcd_rows = 2
```

و با استفاده از کد پایتون زیر ال سی دی را پیکربندی کنیم:

```
lcd = LCD.Adafruit_CharLCD(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6, lcd_d7, lcd_columns,  
lcd_rows, lcd_backlight)
```


در ادامه کد پایتون با استفاده از تابع `message(text)` عبارت `Redronic` را بر روی ال سی دی نمایش می دهیم و پس از ۵ ثانیه با خواندن حروف وارد شده در `clw(command line window)` توسط کاربر، با استفاده از کد زیر حروف وارد شده را بر روی LCD نمایش می دهیم.:

```
text = raw_input("Type Something to be displayed: ")
```

برنامه کامل پایتون:

```
import time

import Adafruit_CharLCD as LCD

lcd_rs = 25

lcd_en = 24

lcd_d4 = 23

lcd_d5 = 17

lcd_d6 = 18

lcd_d7 = 22

lcd_backlight = 2

lcd_columns = 16

lcd_rows = 2

lcd = LCD.Adafruit_CharLCD(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6, lcd_d7, lcd_columns, lcd_rows, lcd_backlight)

lcd.message('Redronic')

time.sleep(5)

lcd.clear()

text = raw_input("Type Something to be displayed: ")

time.sleep(5)

lcd.clear()

lcd.message('GoodbyenWorld!')
```

time.sleep(5)

lcd.clear()

۲-۷- تمرین کلاسی

حال برنامه ای بنویسید که دما و رطوبت را در آل سی دی کاراکتری نمایش دهد.

۳-۷- تمرینات و تکالیف

۱- درباره نحوه اتصال و کار با دوربین رسیبری پای تحقیق کنید.

۸-۱- راه اندازی دوربین با رزبری پای

دوربین های رزبری پای به صورت رسمی در دو نسخه ی ۵ (در سال ۲۰۱۳) و ۸ مگاپیکسلی (در سال 2016) توسط شرکت رزبری پای به بازار معرفی شدند. این دوربین ها کاملاً با بردهای رزبری پای سازگار بوده و به سادگی با دستورات ی که در ادامه معرفی می کنیم، راه اندازی می شوند. از ویژگی های این ماژول می توان به ابعاد کوچک (۹*۲۴*۲۵ میلی متر) و وزن سبک آن (تقریباً ۳ گرم) اشاره کرد که این دوربین را برای انجام انواع پروژه ها، به خصوص ربات یک بسیار ایده آل می کند. اما جالب است بدانیم طبق اطلاعات ارائه شده توسط رزبری پای، از این محصولات بیشتر برای ساخت سیستم های امنیتی خانه ها همچنین تصویربرداری در محیط زیست استفاده شده است. خب این یعنی کاربرد دوربین ها فقط به پروژه های رباتیکی و دانشگاهی محدود نمی شود و هرکسی به سادگی می تواند از آن برای عملی کردن ایده های خلاقانه ی خود استفاده کند. در ادامه در جدول زیر مشخصات دوربین را بررسی میکنیم:

جدول ۸-۱: مشخصات دو نوع دوربین برای رزبری پای

مدل	Camera Module V1	Camera Module V2
اندازه	20mm * 25 mm * 9 mm	20mm * 25 mm * 9 mm
وزن	3g	3g
کیفیت	5 Megapixels	8 Megapixels
رزولوشن	2592 * 1944 Pixels	3280 * 2464 Pixels
فیلم برداری	1080p @ 30fps, 720p @ 60fps and 640x480p 60/90fps	1080p @ 30fps, 720p @ 60fps and 640x480p 60/90fps

علاوه بر مدل های بالا، نسخه ی Camera NoIR Pi هم وجود دارد که برای تصویربرداری در محیط های کم نور مناسب است. این دوربین بین طرفداران جانوران و محیط زیست بسیار محبوب بوده و از آن برای مانیتور کردن فعالیت حیوانات و گیاهان در محیط زیست و حتی منازل، استفاده می شود.

۸-۱-۱- اتصالات

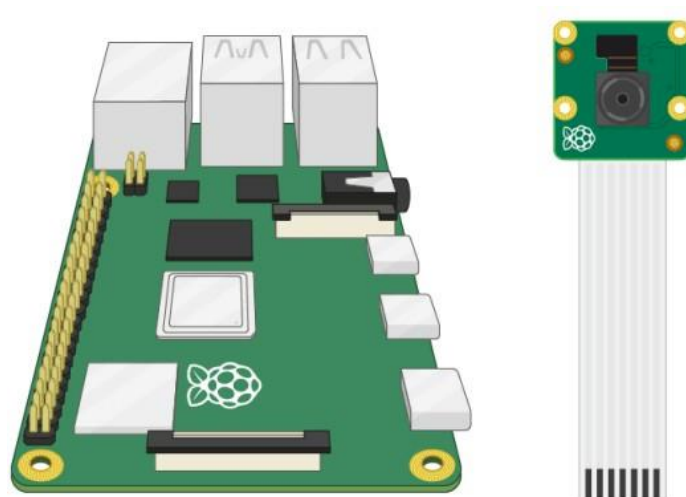
در ابتدا اطمینان حاصل میکنیم که رزبری پای خاموش باشد. سپس مراحل زیر را اجرا میکنیم :

۱- پورت ماژول دوربین را در رزبری پای پیدا میکنیم.

۲- لبه پورت را به آرامی بالا میکشیم.

۳- ماژول دوربین را از قسمت کابل آن وارد پورت میکنیم؛ دقت داریم که کابل فلت ماژول دوربین، از قسمت صحیح جا شده باشد .

۴- لبه پورت را که در مرحله ۲ بالا کشیده بودیم، به سمت پایین سر جای خود فشار میدهیم.

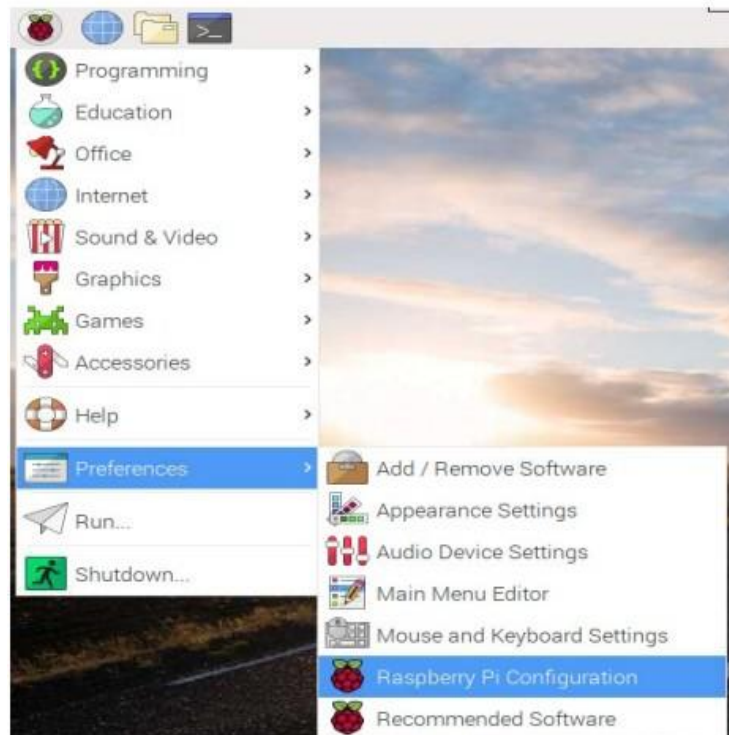


شکل ۸-۱: اتصال دوربین به رزبری پای

۸-۱-۲- برنامه نویسی

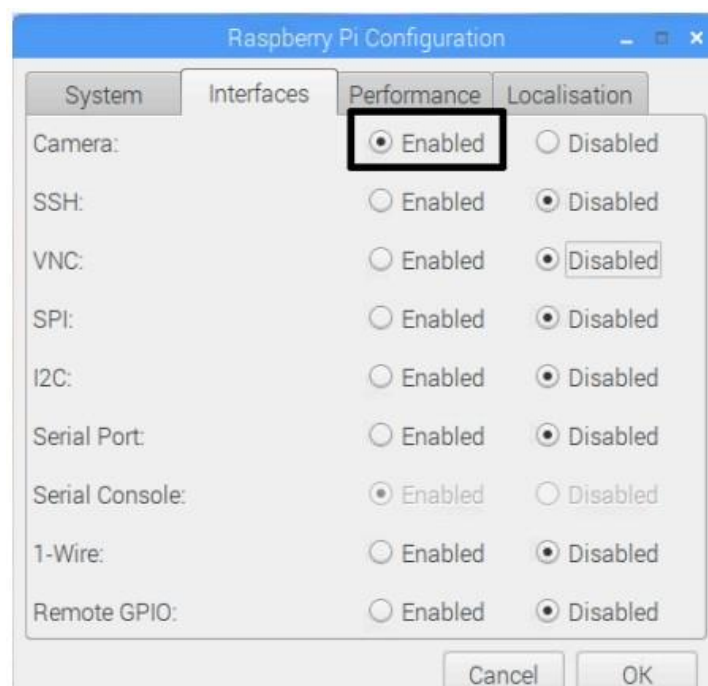
۱- رزبری پای را روشن میکنید.

۲- به منوی اصلی رزبری پای رفته و پس از آن بر زیر منوی preferences کلیک میکنیم سپس Configuration Pi Raspberry را باز میکنیم.



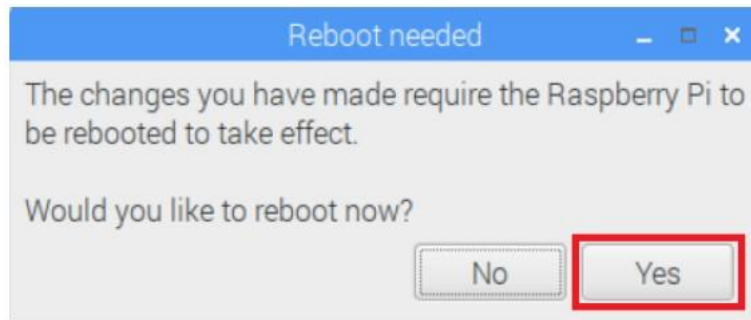
شکل ۸-۲ : PI-CONFIGURATION-MENU

۳- قسمت Interfaces را انتخاب کرده و اطمینان حاصل میکنیم که دوربین در حالت enabled باشد:



شکل ۸-۱ : PI-CONFIGURATION-INTERFACES-ANNOTATED

۴- سیستم عامل را Reboot کرده و مجدداً رزبری پای را راه اندازی میکنیم.



شکل ۸-۴: REBOOT سیستم عامل

۵- دوربین را به رزبری پای متصل میکنیم.

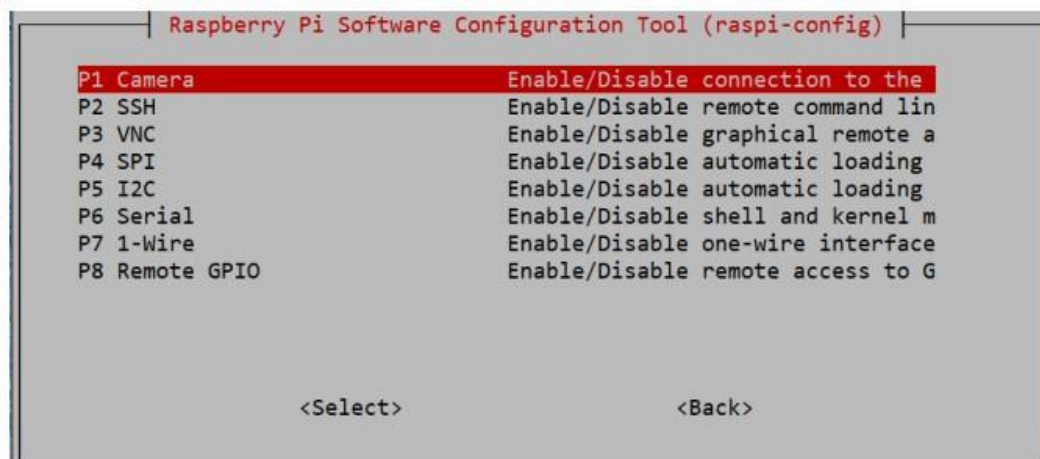
۶- با وارد کردن دستور "sudo raspi-config" در ترمینال سیستم عامل رزبری پای، وارد محیط تنظیمات

عمومی Raspberry Pi می شویم.



شکل ۸-۵: تنظیمات عمومی RASPBERRY PI

۷- به بخش Options Interfacing میرویم. دوربین را فعال کرده و رزبری پای خود را Reboot میکنیم.



شکل ۸-۶: بخش INTER FACEING OPTIONS

۸-۲- عکس برداری با استفاده از ترمینال رزبین :

کتابخانه های مورد نیاز دوربین، به صورت پیش فرض روی سیستم عامل رزبین نصب شده است. با دستور `raspistill` می توان یک عکس گرفت. برای آشنایی با نحوه ی استفاده از این دستور، آن را در ترمینال رزبین تایپ میکنیم .

`raspistill` : برای عکسبرداری با دوربین رزبری پای از دستور پرکاربرد و معروف `raspistill` استفاده می شود، این دستور دارای سوییچهای مختلفی است که در ادامه تعدادی از آن ها را بررسی می کنیم.

ذخیره ی عکس : با توجه به راهنمای `rasistill`، برای گرفتن یک عکس ساده و ذخیره ی آن می توان از دستور زیر استفاده کرد:

نام و محل ذخیره عکس `raspistill -o`

برای مثال کامند زیر یک عکس گرفته و در پوشه `pictures` با نام و فرمت `owl.jpg` ذخیره میکند.

`raspistill -o ~/ pictures/ owl. Jpg`

Command به صورت خودکار پس از ۵ ثانیه عکس برداری می کند.

برای باز کردن فایل تصویر در Command-Line می توانیم دستور نام و آدرس فایل تصویری `xdg open` را اجرا کنیم. البته برای مشاهده ی فایل های گرافیکی باید با VNC یا LCD به دسکتاپ سیستم عامل دسترسی داشته باشیم.

ایجاد تاخیر : در صورت نیاز می توان زمان تاخیر را با سویچ `-t` کنترل کرد:

```
raspistill -t 1000 -o ~/ pictures/ owl. Jpg
```

این زمان بر حسب میلی ثانیه محاسبه می شود. بنابراین در مثال بالا، پس از ۱۰ ثانیه تاخیر، عکسبرداری انجام می شود.

تغییر سایز تصویر : از دو سویچ `-w` و `-h` برای تغییر سایز و اندازه ی تصاویر گرفته شده استفاده می شود (`-w` برای `width` و `-h` برای `height`) برای مثال می خواهیم تصویری با ابعاد `۶۴۰ * ۴۸۰` ایجاد کنیم:

```
raspistill -w 640 -h 480 -o ~/ pictures/ owl. Jpg
```

۸-۳- عکس برداری با استفاده از اسکریپت پایتون:

برای کنترل دوربین رزبری پای با زبان پایتون، از کتابخانه ی `Picamera` استفاده می شود، این کتابخانه امکانات زیادی برای کار با دوربین در اختیار کاربر قرار میدهد.

نکته : با دستور `preview_start` ، دوربین شروع به تصویربرداری می کند. اگر رزبری پای ما مستقیماً به مانیتور متصل باشد، می توان خروجی دوربین را در حالت پیش نمایش (`preview`) دید .

در غیر این صورت اگر از VNC یا SSH استفاده کنیم، قادر به مشاهده ی پیش نمایش تصاویر نخواهیم بود. در کد زیر، ابتدا دوربین ۵ ثانیه به صورت پیش نمایش فعال می شود و سپس یک عکس گرفته و آن را بر روی دسکتاپ با نام `image` ذخیره می کند:


```
from camera import PiCamera

from time import sleep

camera = PiCamera()

camera.start_preview()

sleep(5)

camera.capture (' /home/pi/Desktop/image.jpg')

camera.stop_preview()
```

نکته حداقل ۲ ثانیه به دوربین باید اجازه دهیم تا نور خود را با توجه به محیط تنظیم کند، سپس دستور capture را استفاده میکنیم.

۸-۴- تمرین کلاسی

برنامه ای بنویسید که ۱۰ عکس متوالی بگیرد.

۸-۵- فیلم برداری با استفاده از اسکریپت پایتون

برای یک فیلم برداری ساده از دستورات زیر استفاده می کنیم:

```
from camera import PiCamera

time import

camera = PiCamera()

camera.start_recording('test.h264')

time.sleep(10)

camera.stop_recording()
```

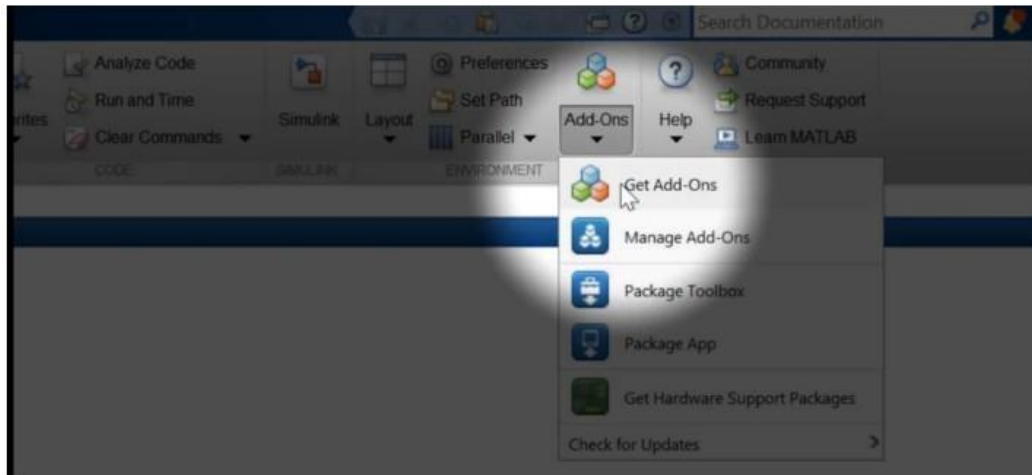
با این دستور، برد رزبری پای به مدت ۱۰ ثانیه فیلم برداری می کند.

۱- درباره طرز نصب و راه اندازی matlab support package for raspberry pi با توجه به لینک زیر برای

ایجاد ارتباط بین راسپبری پای و متلب تحقیق کنید.

<https://www.youtube.com/watch?v=32ByiUdOsw>

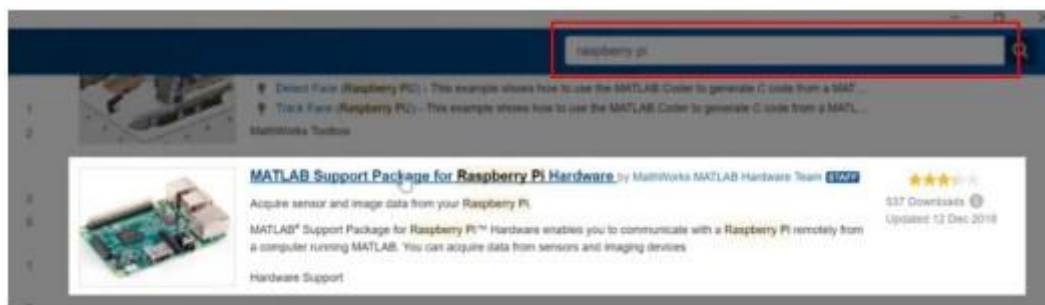
۹-۱- طرز نصب و راه اندازی MATLAB SUPPORT PACKAGE FOR RASPBERRY PI



شکل ۹-۱: آیکون GET ONS-ADD در متلب

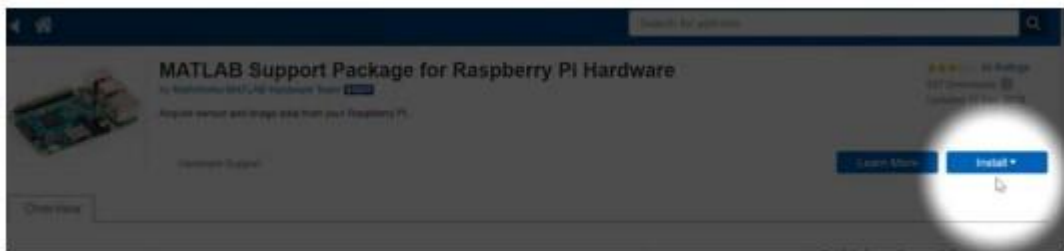
۱- متلب را باز کرده و روی آیکون Add-Ons کلیک کرده و Get Ons-Add را باز میکنیم.

۲- در پنجره باز شده در قسمت search عبارت pi raspberry را تایپ میکنیم، در ادامه طبق تصویر زیر گزینه دوم را انتخاب میکنیم.



شکل ۹-۲: انتخاب پک MATLAB SUPPORT PACKAGE FOR RASPBERRY PI

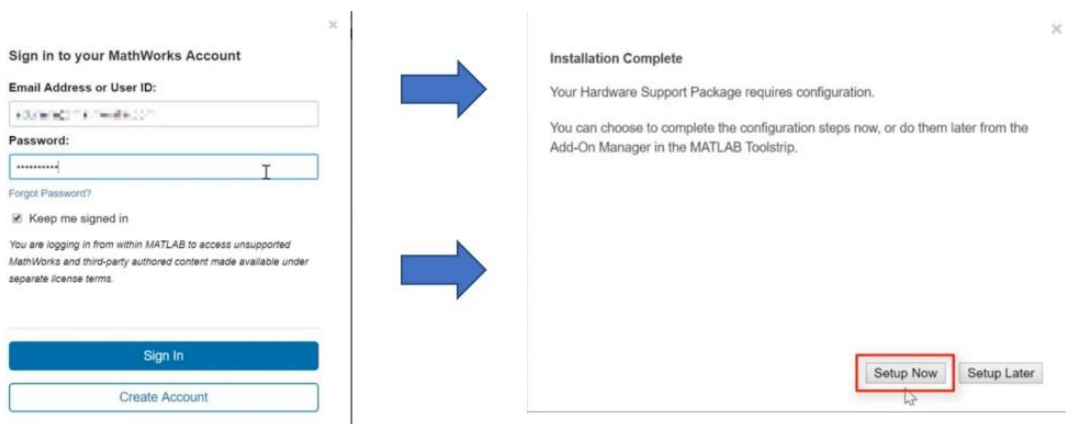
۳- بعد از کلیک روی گزینه دوم پنجره ای جدید باز شده که طبق تصویر زیر روی **install** کلیک



شکل ۹-۳: نصب پک MATLAB SUPPORT PACKAGE FOR RASPBERRY PI

میکنیم.

۴- برای نصب matlab package support باید وارد حساب کاربری mathwork خود شویم.



شکل ۹-۴: ورود به حساب کاربری MATHWORK

۵- بعد از کامل شدن نصب روی گزینه **now setup** کلیک میکنیم.

۶- دوباره در صفحه اصلی رو آیکن On-Add کلیک کرده و گزینه On-Add Manage کلیک میکنیم تا پنجره زیر باز شود.

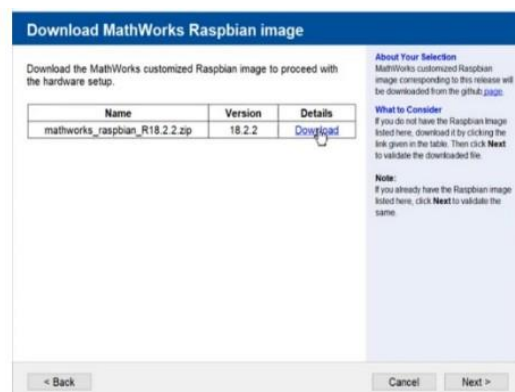
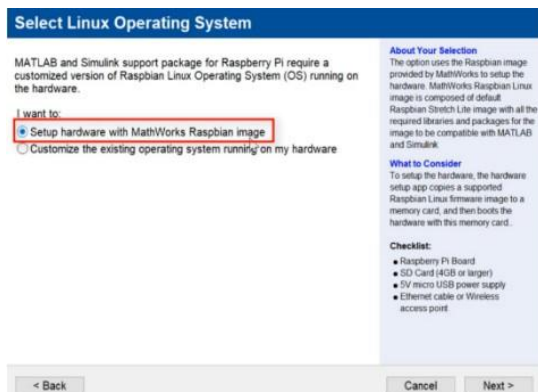
۷- در پنجره فوق در گزینه اول روی علامت setting یا چرخنده کلیک کرده تا پنجره Select Board Hardware باز شود. از قسمت Hardware نوع و مدل برد رزبری پای خود را انتخاب میکنیم. و در ادامه روی Next کلیک مینماییم.

۸- بعد از Next پنجره System Operating Linux Select باز شده و همانند تصویر زیر گزینه اول را



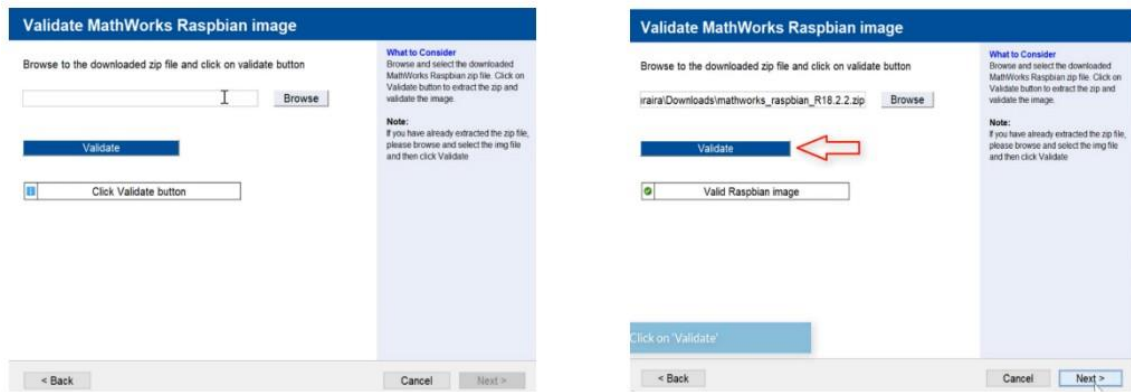
شکل ۹-۵: مراحل نصب

انتخاب میکنیم. در پنجره جدید روی Download کلیک میکنیم.



شکل ۹-۶: مراحل نصب

۹- بعد از دانلود فایل محل قرار گیری فایل را به متلب معرفی میکنیم.



شکل ۹-۷: مراحل نصب

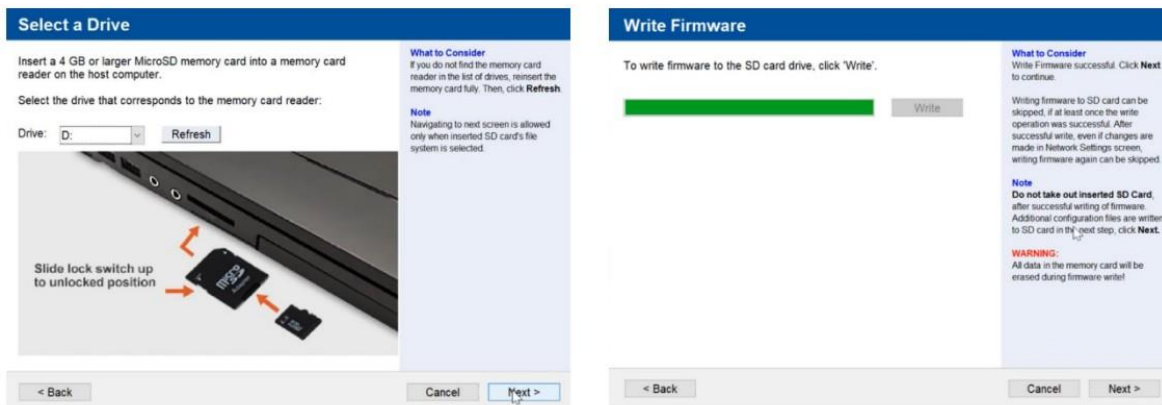
۱۰- در ادامه در پنجره زیر نحوه اتصال متلب به رزبری پای مشخص میگردد، برای اتصال بصورت

Wireless مشخصات شبکه رابط یا وایفای متصل به رایانه و رزبری را وارد میکنیم.



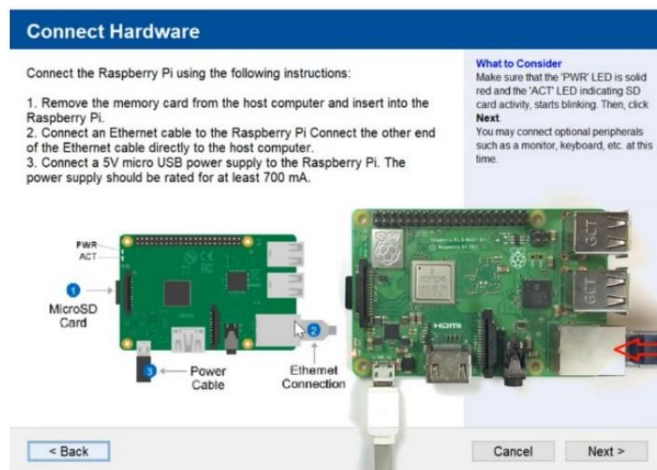
شکل ۹-۸: مراحل نصب

۱۱- در ادامه Micro SD Memory را در رایانه قرار می‌دهیم تا پکیج دانلود شده را روی آن بارگزاری نماییم.



شکل ۹-۹: مراحل نصب

۱۲- در ادامه با توجه به تصویر صفحه بعد رایانه و رزبری پای را با استفاده از کابل اترنت به یکدیگر متصل کرده و مموری کارت را داخل رزبری قرار می‌دهیم.



شکل ۹-۱۰: مراحل نصب

۱۳- در ادامه برای اطمینان از کانکت شدن روی گزینه Test Connection کلیک مینماییم .

۱۴- در ادامه با وارد شدن به Command window دستورات زیر را تایپ میکنیم.

۲-۹- تمرین کلاسی

۱- برنامه ی روشن و خاموش کردن LED با استفاده از پکیج رزبری پای متلب نوشته و اجرا کنید.

۳-۹- چارچوب تشخیص اشیاء ویولا-جونز

۱-۳-۹- مقدمه

```
Command Window
>> r=raspi

r =

raspi with properties:

    DeviceAddress: '169.254.0.2'
           Port: 18734
    BoardName: 'Raspberry Pi 3 Model B+'
    AvailableLEDs: {'led0'}
AvailableDigitalPins: [4, 5, 6, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]
AvailableSPIChannels: {'CE0', 'CE1'}
    AvailableI2CBuses: {'i2c-1'}
    AvailableWebcams: {}
           I2CBusSpeed: 100000

Supported peripherals
```

شکل ۹-۱۲: دستورات COMMAND WINDOW

چارچوب تشخیص اشیاء Jones-Viola یک چارچوب تشخیص اشیاء در تصویر مبتنی بر یادگیری ماشین با ارائه نرخ تشخیص بلادرنگ است که در سال ۲۰۰۱ توسط پاول ویولا و مایکل جونز ارائه شد، اگر چه این چارچوب می تواند برای تشخیص انواع اشیاء آموزش داده شود، در ابتدا به منظور تشخیص چهره توسعه داده شده بود و اولین الگوریتم تشخیص چهره بلادرنگ به شمار می رود .

مقاله‌ی اصلی سه نوآوری کلیدی ارائه می‌کرد. اولین نوآوری استفاده از نمایشی برای تصویر به نام تصویر انتگرالی بود که به کمک آن امکان محاسبه بسیار سریع و یژگی‌های مورد نیاز برای تشخیص در زمان ثابت فراهم می‌گردید. نوآوری دوم توسعه یک الگوریتم یادگیری ماشین مبتنی بر آدابوست به منظور انتخاب مجموعه‌ای کوچک اما کارا از میان مجموعه بزرگی از ویژگی‌های تصویری بود و نوآوری سوم ارائه یک معماری آبخاری به منظور ترکیب بهینه مجموعه‌ای از الگوریتم‌های طبقه‌بندی آماری بود که به موجب آن قسمت‌های پس‌زمینه تصویر با الگوریتم‌های طبقه‌بندی سریع دور ریخته می‌شوند و الگوریتم‌های پیچیده‌تر طبقه‌بندی روی قسمت‌های مهم‌تر تصویر تمرکز می‌کنند. با وجود اینکه تا کنون الگوریتم‌های بسیار زیادی به منظور تشخیص چهره توسعه داده شده‌اند می‌توان گفت از این میان الگوریتم ویولا-جونز بیشترین تاثیر را بر دهه‌های پیش‌روی خود داشت و قادر بود با سرعت بسیار زیاد دقت مطلوبی در مکان‌یابی چهره ارائه دهد.

۹-۳-۲- شرح مسئله

مسئله تشخیص چهره یکی از مسائل شناخته شده بینایی رایانه‌ای است. مسئله مورد نظر چهره انسان را در تصاویر دیجیتال مکانیابی می‌کند. فرآیند تشخیص چهره به دو سوال پاسخ می‌دهد: اول اینکه آیا چهره‌ای در یک تصویر و یا دنباله‌ای از تصاویر ویدئویی وجود دارد؟ دوم اینکه این چهره در کجای تصویر واقع شده است؟ اغلب انسان‌ها می‌توانند این کار را به آسانی انجام دهند، اما یک کامپیوتر نیاز به دستورالعمل‌های دقیق و پیچیده‌ای دارد. پیچیدگی مسئله با تفاوت مقیاس تصویر، زاویه تصویربرداری، شرایط نوری متفاوت و پوشیده بودن قسمتی از چهره چند برابر می‌شود.

با وجود اینکه تا کنون الگوریتم‌های بسیار زیادی به منظور تشخیص چهره توسعه داده شده‌اند می‌توان گفت از این میان الگوریتم ویولا-جونز بیشترین تاثیر را بر دهه‌های پیش‌روی خود داشت و قادر بود و اولین الگوریتم بلادرنگ مکان‌یابی چهره با دقتی مطلوب به شمار می‌رود.

الگوریتم ویولا-جونز نیاز به نمای کامل صورت از جلو دارد. بنابراین برای تشخیص، کل چهره باید به سمت دوربین باشد و نباید به اطراف منحرف شود. در حالی که به نظر می رسد این محدودیت ها تا حدودی باعث کاهش کارایی الگوریتم می شوند در بسیاری از کاربردها این میزان کارایی قابل قبول است.

از مزایای چارچوب ارائه شده این است که برای تشخیص چهره در مقیاس مختلف می توان به جای مقیاس تصویر، مقیاس ویژگی های استفاده شده را عوض کرد و همچنین می توان این الگوریتم را برای تشخیص اشیاء دیگری همچون خودرو آموزش داد.

این الگوریتم به میزان قابل قبولی Robust است به این معنا که حساسیت بالا (مثبت صحیح) و تشخیص بالا (منفی صحیح) دارد و همچنین پردازش آن به صورت بلادرنگ صورت می گیرد.

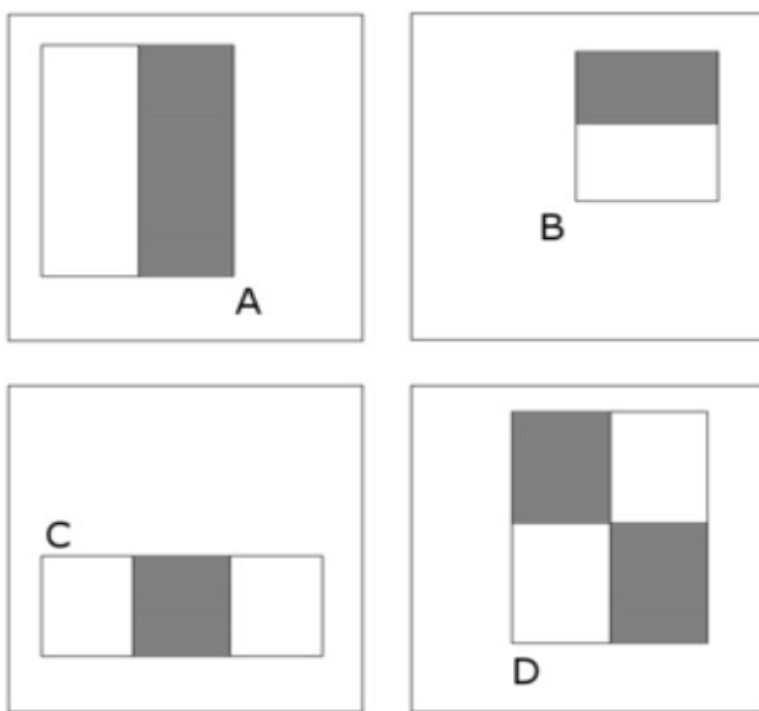
از نقاط ضعف این الگوریتم میتوان به حساسیت به شرایط نوری و اندازه چهره در تصویر اشاره کرد. این الگوریتم همچنین در تشخیص چهره های چرخیده دچار مشکل می شود و ممکن است یک چهره را به عنوان چند چهره در همپوشانی زیرپنجره های مورد جستجو پیدا کند. علیرغم همه این موارد و توسعه روش های جدید تشخیص چهره با استفاده از یادگیری عمیق که در تشخیص چهره از همه نژادها، در تمامی زوایا بدون حساسیت به مقیاس تصویر و شرایط نوری از کارایی بسیار بالایی برخوردار هستند، نوآوری هایی که در این چارچوب ارائه شد امروزه هنوز هم در سایر مسائل بینایی رایانه ای به وفور مورد استفاده قرار می گیرد.

۹-۳-۳- اجزاء چارچوب

الگوریتم ویولا - جونز با پیروی از کار دیگر محققان به جای استفاده از مقادیر تک تک پیکسل های تصویر اقدام به استخراج ویژگی های مستطیلی شبه Haar می کند که با الهام از توابع پایه موجک هار شکل گرفته اند و قبلاً در حیطه تشخیص اشیاء مبتنی بر تصویر مورد استفاده قرار گرفته بودند اما ویژگی های مورد استفاده ویولا - جونز معمولاً پیچیده تر هستند. شکل سمت چپ نشانگر چهار نوع مختلف از ویژگی های مورد استفاده در چارچوب است.

ارزش هر ویژگی مشخص شده برابر با مجموع تمام مقادیر پیکسل های داخل مستطیل های روشن است که از مجموع تمام پیکسل های داخل مستطیل های تیره کاسته می شوند.

ویژگی های مستطیلی از این نوع در مقایسه با گزینه هایی مانند فیلترهای قابل هدایت که در تحلیل بافت و فشرده سازی تصویر استفاده می شوند بسیار ابتدایی به شمار می روند. اگرچه این ویژگی ها به لبه های عمودی افقی و سایر ساختارهای ساده تصویری حساس هستند، با این وجود این حساسیت به هیچ عنوان انعطاف پذیر نیست. برخلاف فیلترهای قابل هدایت تنها جهتی که می توان برای ویژگی های شبه Haar در نظر گرفت افقی، عمودی و قطری است اما تعداد بسیار زیاد این ویژگی ها توصیف قابل قبولی از تصویر ارائه می کند و استفاده از آن ها و



شکل ۹-۱۳: نمونه ای از ویژگیهای مستطیلی محصور شده در پنجره تشخیص

ارزیابی سریع آن ها با استفاده از تصویر انتگرالی این محدودیت انعطاف را به میزان قابل قبولی جبران می کند. به منظور ارزیابی ویژگی های مستطیلی شبه Haar در زمان ثابت از تصویر انتگرالی استفاده می شود. پس از ساخته شدن تصویر انتگرالی مقدار هر کدام از این ویژگی ها فارغ از مکان و اندازه آنها در تصویر با جمع و تفریق حداکثر نه عدد قابل محاسبه است. پس از ارزیابی تمام ویژگی ها، مجموعه ای از کارا ترین آنها به کمک الگوریتم

یادگیری ماشین آدابوست انتخاب شده و مورد استفاده قرار می گیرند . این چارچوب همچنین از یک معماری آبخاری برای پیدا کردن سریع مناطق جالب توجه تصویر بهره می گیرد که امکان تشخیص چهره بلا درنگ را فراهم می کند . بنابراین اجزای این چارچوب عبارتند از:

- انتخاب ویژگی های شبه Haar
- ایجاد یک تصویر یکپارچه به منظور ارزیابی ویژگی ها
- آموزش آدابوست برای انتخاب بهترین ویژگی ها
- انتخاب مناطق جالب توجه تصویر با استفاده از معماری آبخاری

۹-۳-۴- ویژگی های شبه HAAR

ویژگی های شبه Haar عبارتند از نواحی روشن و تیره مستطیل شکلی که بر روی زیر پنجره ۲۴ در ۲۴ پیکسل مورد بررسی (نامزد احتمالی در برگزیده صورت) تعریف شده اند. ارزش هر ویژگی مشخص شده برابر با مجموع تمام مقادیر پیکسل های داخل مستطیل های روشن است که از مجموع تمام پیکسل های داخل مستطیل های تیره کاسته می شوند . سایر نقاط عکس که خارج از این مستطیل ها قرار دارند در محاسبه ویژگی در نظر گرفته نمی شوند . به این ترتیب برای هر پنجره ۲۴ در ۲۴ پیکسل متناظر با یک ویژگی یک عدد به عنوان ارزش این ویژگی محاسبه میشود . موضوعی که موفقیت این ویژگی های مستطیل شکل را تضمین می کند استفاده از ویژگی های کمابیش مشترک چهره های انسانی است .

تمامی چهره های انسانی دارای ویژگی های کمابیش مشابهی هستند . این قواعد می تواند با استفاده از ویژگی های شبه Haar تطبیق داده شوند . به عنوان مثال چند ویژگی مشترک برای چهره های انسانی عبارتند از:

- ناحیه پل بینی از چشم ها روشن تر است .
- ناحیه فرورفتگی چشم ها تیره تر از گونه ها است .

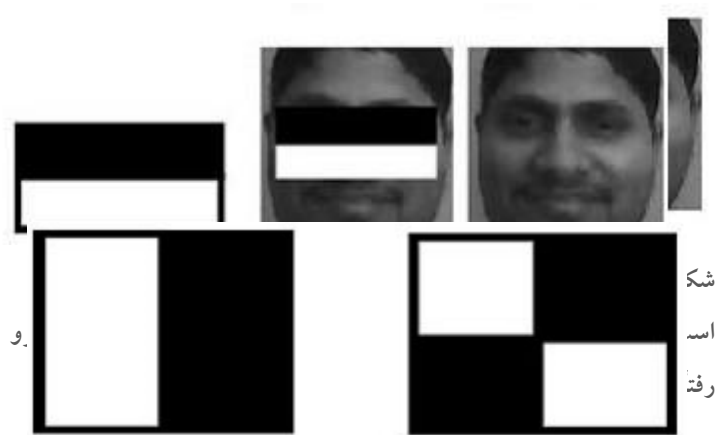
ترکیب خواص زیر ویژگی های مشترک صورت های انسانی را تشکیل می دهند:

- محل و اندازه : چشمها، دهان، پل بینی

- ارزش : جهت گرادیان مقادیر پیکسل ها

ویژگی هایی که ویوال و جونز استفاده می کنند دارای نواح ی مستطیل شکل تیره و روشنی هستند که با مساحت

برابر به صورت افق ی یا عمودی در کنار هم قرار دارند. این ویژگی ی ها بر سه دسته هستند (تصاویر زیر):



شکل ۹-۱۴ : سایر ویژگی های HAAR که شامل دو-مستطیلی عمودی و چهار-مستطیلی قطری است.

- ویژگی های دو - مستطیلی که شامل دو مستطیل با مساحت برابر در کنار یکدیگر به صورت افقی یا عمودی هستند.

- ویژگی های سه - مستطیلی که مقادیر دو مستطیل کناری را از مستطیل وسط کم می کند.

- ویژگی های چهار- مستطیلی که به صورت قطری در کنار یکدیگر قرار گرفته اند.

این سه نوع ویژگی در مکان های مختلف زیرپنجره ۲۴ در ۲۴ پیکسلی تعریف می شوند و تعداد آنها از ۱۸۰ هزار عدد بیشتر است ، بنابراین نه تنها نیاز به روش ی سریع برای ارزیابی مقادیر هر یک از این ویژگی ها احساس می شود، بلکه می بایست زیر مجموعه موثرتری از این ویژگی ها انتخاب شود که در ارزیابی تصویر نهایی مورد استفاده قرار گیرد.

همان طور که اشاره شد تعداد ویژگی های مورد استفاده در چارچوب ویولا-جونز متجاوز از ۱۸۰ هزار فیلتر است.

به همین دلیل بدون استفاده از روشی سریع به منظور ارزیابی این ویژگی ها امکان آموزش الگوریتم های طبقه

بندی به منظور انتخاب زیرمجموعه ای موثر از این ویژگی ها و همچنین امکان ارزیابی بلادرنگ آنها روی زیر

پنجره های نهایی تصویر وجود نخواهد داشت. همان طور که اشاره شد مقدار هر یک از این ویژگی ها از تفاضل

مجموع مقادیر پیکسل های داخل نواحی تیره از مجموع مقادیر پیکسل های داخل نواحی روشن بدست می آید.

ارزش ویژگی = (پیکسل های در ناحیه سفید) Σ - (پیکسل در ناحیه سیاه) Σ

ویولا - جونز نشان دادند که چگونه می توان با استفاده کارا از تصویر انتگرالی ویژگی های مستطیلی را در زمانی

ثابت ارزیابی کرد، این روش مزیت سرعت ارزیابی قابل توجهی را نسبت به سایر ویژگی های پیچیده جایگزین

فراهم آورد و استفاده از تصویر انتگرالی را در بینایی رایانه ای تثبیت کرد چرا که محاسبه تصویر انتگرالی تنها با یک

بار عبور از روی تصویر اصلی قابل انجام است.

در یک تصویر انتگرالی مقدار هر پیکسل برابر با مجموع تمام پیکسل های بالا و سمت راست این پیکسل روی تصویر

اصلی به علاوه مقدار خود پیکسل روی تصویر اصلی است:

$$ii(x,y) = \Sigma i(x',y')$$

که در آن $ii(y,x)$ تصویر انتگرالی و $i(y,x)$ تصویر اصلی است. با استفاده از دو رابطه بازگشتی زیر می توان تصویر

انتگرالی را تنها با یک بار عبور از روی تصویر اصلی ساخت.

که در آن $s(y,x)$ مقدار تجمعی سطری و $s(x,-1)=0$ و $ii(-1,y)=0$ است.

$$s(x,y) = s(x,y-1) + i(x,y)$$

$$ii(x,y) = ii(x-1,y) + s(x,y)$$

همان طور که در تصویر پایین نمایش داده شده است، با استفاده از تصویر انتگرالی می توان مجموع مقادیر داخل هر ناحیه مستطیلی روی تصویر اصلی را تنها با مقادیر چهار پیکسل روی تصویر انتگرالی محاسبه کرد. از آنجا که نواحی مستطیلی هر یک از ویژگی های شبه Haar همیشه در مجاورت یکدیگر قرار دارند، مثال برای محاسبه هر ویژگی دو - مستطیلی تنها به مقادیر شش پیکسل، برای محاسبه هر ویژگی سه - مستطیلی تنها به مقادیر هشت پیکسل و برای محاسبه هر ویژگی چهار - مستطیلی تنها به مقادیر و نه به پیکسل از تصویر انتگرالی نیاز است.

۹-۳-۶- الگوریتم یادگیری

سرعتی که با آن ویژگی ها ممکن است ارزیابی شوند باعث نمیشود از تعداد بالای ویژگی ها چشم پوشی کنیم. برای مثال، در یک زیر پنجره $24 * 24$ پیکسل استاندارد، مجموعاً $162,336 M$ ویژگی ممکن وجود دارد، و ارزیابی همه آن ها در زمان تست تصویر به شدت هزینه بر است. بنابراین، فریم ورک تشخیص شی از انواع الگوریتم یادگیری AdaBoost برای انتخاب بهترین ویژگی ها و آموزش دسته بندی کننده برای استفاده از آن ها استفاده کرده است. این الگوریتم یک دسته بندی کننده "قوی" به عنوان یک ترکیب خطی دسته بندی کننده "ضعیف" ساده است:

$$h(x) = \text{sgn}(\sum \alpha_j h_j(x))$$

هر دسته بندی کننده ضعیف یک تابع آستانه بر اساس ویژگی های زیر دارد.

$$h_j(x) = \begin{cases} -s_j, & f_j < \theta_j \\ s_j, & \text{other} \end{cases}$$

مقدار آستانه θ_j و قطبیت $s_j \in \pm 1$ در آموزش و ضرایب α_j تعیین شده است.

در زیر نسخه ساده شده الگوریتم یادگیری آورده شده است.

ورودی: مجموعه ای از N تصویر آموزشی مثبت و منفی با برچسب های (x^i, y^i) اگر تصویر یک چهره باشد برقرار است.

۱- مقدار دهی اولیه: انتساب وزن $\omega_1^i = \frac{1}{N}$ به هر تصویر i

۲- برای هر ویژگی f_j با $j=1, \dots, M$

- نرمال سازی وزن ها با جمع بستن با یک
- استفاده از ویژگی برای هر تصویر در مجموعه آموزشی، یافتن آستانه بهینه و قطبیت که خطای دسته بندی وزن شده را به حداقل می رساند، به طوری که رابطه زیر برقرار باشد.

$$\Theta_j, s_j = \arg \min \sum \omega_j^i \varepsilon_j^i \text{ where } \varepsilon_j^i = \begin{cases} 0, & \text{if } y^i = h_j(x^i, \Theta_j, s_j) \\ 1, & \text{otherwise} \end{cases}$$

- انتساب یک وزن به طوری که نسبت عکسی با نرخ خطا داشته باشد. در این روش بهترین دسته بندی کننده بیشتر در نظر گرفته می شود.
- وزن هایی برای تکرار بعدی، مثال برای تصاویر i که به درستی دسته بندی شده است کاهش می یابد.

$$h(x) = \text{sgn}(\sum \omega_j \alpha_j h_j(x))$$

همانطور که در شکل زیر مشاهده می کنید یک کد ساده به منظور تشخیص چهره به روش ویولا - جونز ارائه شده است.

```

1 function FaceDetect()
2   %#codegen
3   %Create raspi & webcam obj
4   raspiObj = raspi('192.168.233.85','pi','pi');
5   cam = cameraboard(raspiObj,'Resolution','320x240','FrameRate',30);
6   while true
7     img = snapshot(cam);
8     faceDetector = vision.CascadeObjectDetector;
9     bbox = step(faceDetector, img);
10    Ifaces = insertObjectAnnotation(img,'rectangle',bbox,'Face');
11    imagesc(Ifaces), title('Detected Faces'),
12    drawnow;
13  end
14 end
15
16

```

شکل ۹-۱۵: نمونه کد تشخیص چهره با روش ویولا - جونز