دانشکده برق، کامپیوتر و فناوری های پیشرفته

گروه مهندسی برق الکترونیک

دستورکار آزمایشگاه سیستم‌های دیجیتال ۲

تهیه کنندگان:

مهندس سیدکیان موسوی کیا

تاریخ تنظیم:

مهرماه ۱۴۰۳

مقدمه..... ۶

۰-۱- چند نمونه از مجموعه برد های SBC ٦

۰-۲- آشنایی با سری برد های Raspberry Pi ٦

۰-۳- انواع برد های Raspberry Pi ۷

۰-٤- معرفی برد Pi ۳ Model B ۸

۰-۵- تمرینات وتکالیف ۱۰

آزمایش ۱..... ۱۱

-۱-۱ مراحل راه اندازی Raspberry Pi ۳B ۱۱

۱-۱-۱- نصب سیستم عامل بر روی sd card ۱۱

۱-۱-۲- اتصالات Raspberry Pi ۳B ۱۳

۱-۱-۳- تنظیمات اولیه سیستم عامل بر روی Raspberry Pi ۳B ۱۳

۲-۲- کنترل Raspberry Pi از راه دور ۱۸

آزمایش ۲..... ۲۰

۲-۱- زبان برنامه نویسی ومعرفی نرم افزارکد نویسی ۲۰

۲-۱-۱- معرفی برنامه Thonny IDE ۲۰

-۲-۱-۲معرفی ترمینال رزبری پای ۲۲

۲-۱-۲-۱- برخی از کد های مهم ترمینال رزبری پای ۲٤

۲-۲-تمرینات وتکالیف ۲٦

آزمایش۳..... ۲۶

۳-۱- آشنایی با زبان برنامه نویسی پایتون جلسه اول ۲٦

۳-۱-۱- انواع متغییر ها ۲٦

۳-۱-۲-نمونه توابع پایتون ۲۷

۳-۲-تمرینات وتکالیف ۳۱

آزمایش ۴..... ۳۲

٤-۱- آشنایی با زبان برنامه نویسی پایتون جلسه دوم ۳۲

٤-۱-۱-نمونه توابع و دستورات در پایتون ۳۲

٤-۲-تمرینات وتکالیف ۳٦

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 وتکالیف ٤۲

آزمایش ۶..... ۴۳

٦-۱- راه اندازی سنسور DHT ۱۱ ..... ٤۳

٦-۱-۱- اتصالات مدار ٤۳

٦-۱-۲- برنامه نویسی ٤٤

٦-٢-راه اندازی سنسور اولتراسونیک ٤٦
٦-٢-١- نحوه کار سنسورهای التراسونیک ٤٦
٦-٢-٢- اتصالات مدار ٤٧
٦-٢-٣- برنامه نویسی ٤٨
٦-٣- تمرینات و تکالیف ٥٢
آزمایش ۷..... ۵۳
۷-۱- راه اندازی LCD ۲*۱٦ کاراکتری ۵۳
۷-۱-۱- اتصالات ۵٤
۷-۱-۲- برنامه نویسی ۵٦
۷-۲- تمرین کلاسی ۵۸
۷-۳- تمرینات و تکالیف ۵۸
آزمایش ۸..... ۵۹
۸-۱- راه اندازی دوربین با رزبری پای ۵۹
۸-۱-۱- اتصالات ٦۰
۸-۱-۲- برنامه نویسی ٦۱
۸-۲- عکس برداری با استفاده از ترمینال رزبین : ٦٤
۸-۳- عکس برداری با استفاده از اسکریپت پایتون: ٦٥
۸-٤- تمرین کلاسی ٦٦

۸-۵- فیلم برداری با استفاده از اسکریپت پا یتون ٦٦
۸-٦- تمرینات و تکالیف ٦٧
آزمایش۹..... ۶۸
۹-۱- طرز نصب و راه اندازی matlab support package for raspberry pi ٦٨
۹-۲- تمرین کلاسی ۷۳
۹-۳- چارچوب تشخیص اشیاء ویولا-جونز ۷۳
۹-۳-۱- مقدمه ۷۳
۹-۳-۲- شرح مسئله ۷٤
۹-۳-۳- اجزاء چارچوب ۷۵
۹-۳-٤- ویژگی های شبه Haar ۷۷
۹-۳-۵- تصویر انتگرالی ۷۹
۹-۳-٦- الگوریتم یادگیری ۸۰
۹-۳-۷- نمونه کد نویسی تشخیص چهره ۸۲

مقدمه

SBC مخفف عبارت Single Board Computer به معنای کامپیوتر تک برد است. همانطور که از اسم آن بر می‌آید کامپیوتری است که در یک برد کوچک خلاصه می‌شود. 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 GPIOو SPI و I۲C و LCDو Camera و ... نیز در این سیستم‌ها وجود دارند. همین ویژگی ها باعث می‌شود این سیستم‌ها فراتر از یک کامپیوتر عادی بوده و کاربری‌های جالبی برای آنها پیدا شود.

۰-۱- 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://all۳dp.com/۱/single-board-computer-raspberry-pi-alternative

۰-۲- 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.

۰-۳- انواع برد های RASPBERRY PI

- Pi ۱ Model B (۲۰۱۲)
- Pi ۱ Model A (۲۰۱۳)
- Pi ۱ Model B+ (۲۰۱٤)
- Pi ۱ Model A+ (۲۰۱٤)
- Pi ۲ Model B (۲۰۱٥)
- Pi Zero (۲۰۱٥)
- Pi ۳ Model B (۲۰۱٦)
- Pi Zero W (۲۰۱۷)
- Pi ٤ Model B (۲۰۱۹)
- Pi ٤ (۲۰۲۰)
- Pi Zero ۲W (۲۰۲۱)
- Pi Pico (۲۰۲۱)
- Pi ٥ (۲۰۲۳)

۰-٤- معرفی برد PI ۳ MODEL B

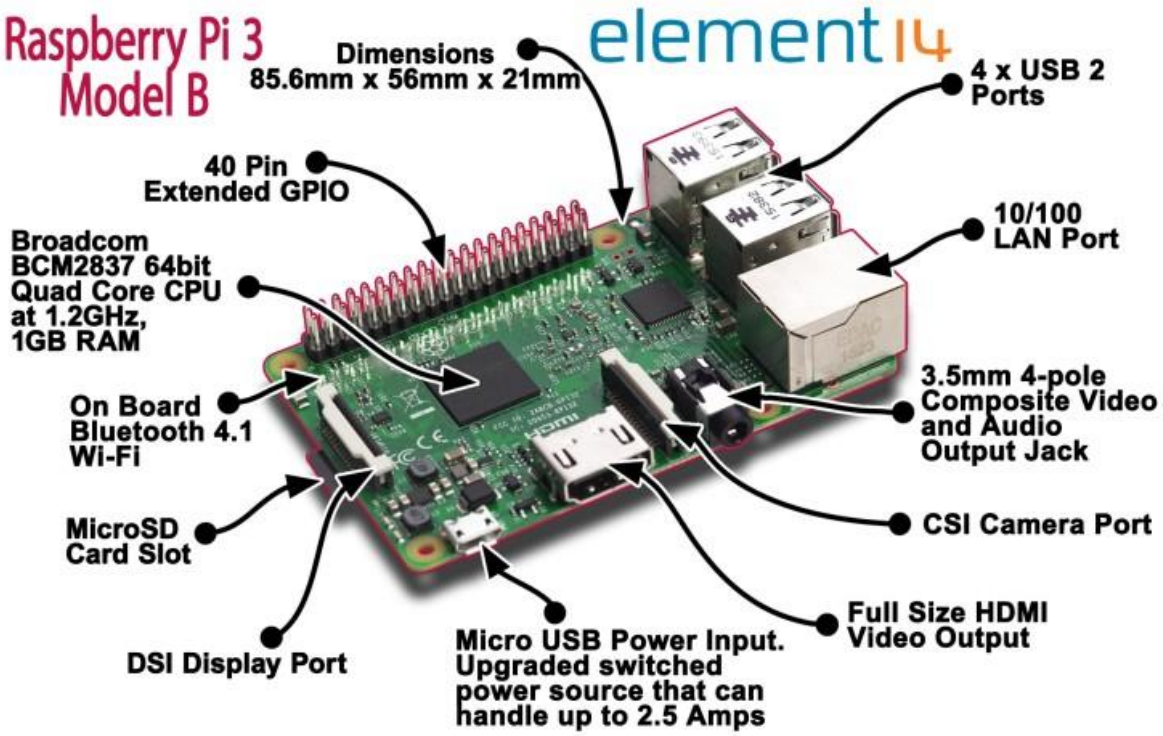
شکل ۰-۱: برد Pi ۳ Model B

جدول ۰-۱ : مشخصات فنی Pi ۳ Model B

پردازنده	Broadcom BCM۲۳۸۷ chipset ۱,۲GHz Quad-Core ARM Cortex-A۵۳
پردازنده گرافیکی	دو هسته ای VideoCore IV® Multimedia Co-Processor
بلوتوث	٤.۱ LE
وای فای	دارد
حافظه	۱ گیگابایت LPDDR۲
سیستم عامل	لینوکس – Windows ۱۰ IoT (سیستم عامل از کارت Micro SD بوت می شود)
ابعاد	۸۵ × ٥٦ × ۱۷ میلی متر
تغذیه	سوکت میکرو یو اس بی ۵V – ۲,۵A

جدول ۰-۲ : اتصالات Pi ۳ Model B

سوکت اترنت	BaseT ۱۰۰/۱۰
خروجی ویدئو	HDMI (rev ۱،۳) & ۱.٤
کامپوزیت	RCA (PAL و NTSC)
خروجی صدا	جک ۳.۵ میلی متری، HDMI
کانکتورUSB	USB ٤ x USB ۲،۰
پین های ورودی و خروجی	هدر برد ٤۰ پین با رابط GPIO ۲،٥٤ ، تعداد ۲۷ پین GPIO و همچنین پین های ۳.۳+، ۵+ ولت و GND
ورودی دوربین	۱۵ پین MIPI رابط سریال دوربین (CSI-۲)
ورودی صفحه نمایش	رابط سریال (DSI) کانکتور کابل فلکس تخت ۱۵ جهته با دو دیتا
خطوط کلاک	یک خط کلاک
ورودی کارت حافظه	Micro SDIO Push/pull اسلات



شکل ۰-۲ : برد Pi ۳ Model B در یک نگاه

۰-۵- تمرینات وتکالیف

۱- برنامه Win۳۲ Disk و برنامه SD Card.Formatter را برای جلسه بعد دانلود کنید.

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

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

آزمایش ۱

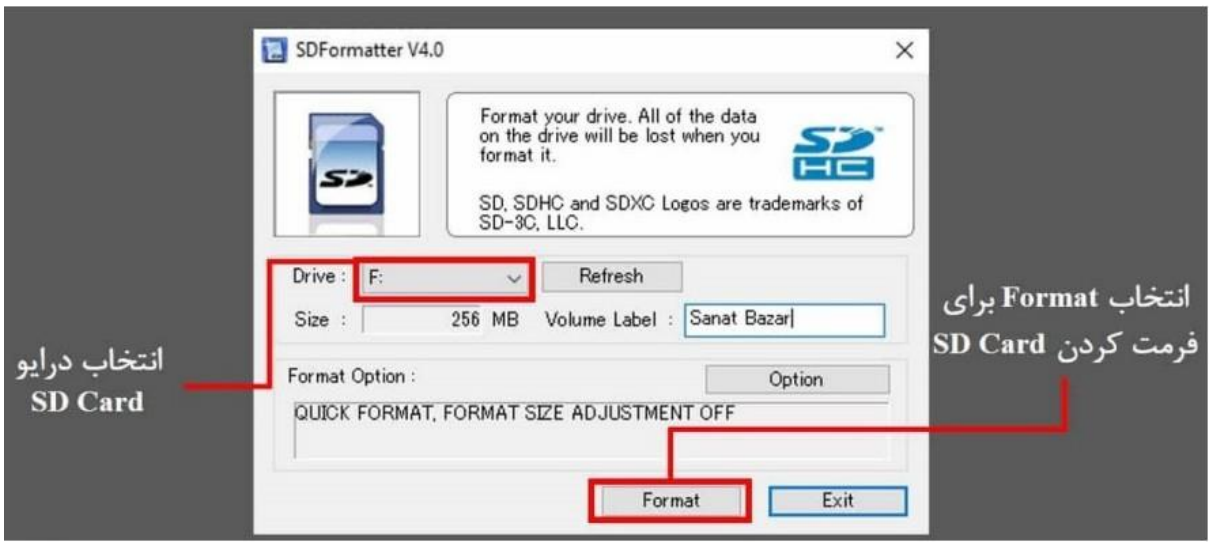
۱-۱- مراحل راه اندازی RASPBERRY PI ۳B

۱-۱-۱- نصب سیستم عامل بر روی SD CARD

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

https://www.yasdl.com/۱٤٤۹۰۲/%D۸٪AF%D۸٪A۷٪D۹٪۸٦٪D۹٪۸٤٪D۹٪۸۸٪D۸٪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 GB ٤ 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.

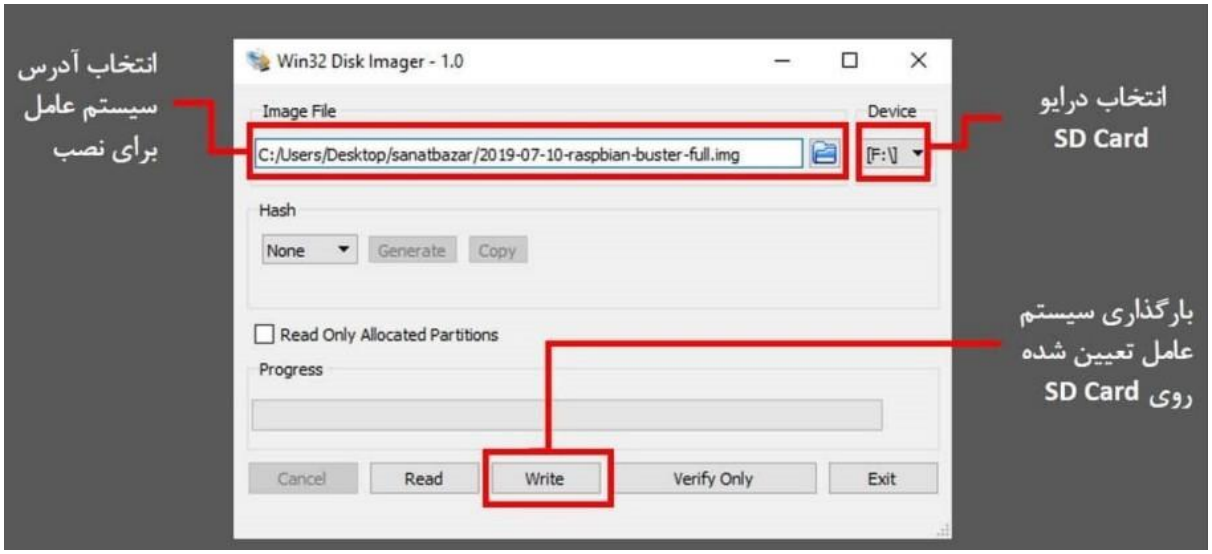
برنامه Win۳۲.Disk را از لینک زیر دانلود کنید.

https://www.yasdl.com/۲۰۵۰۵۲/%D۸٪AF%D۸٪A۷٪D۹٪۸٦٪D۹٪۸٤٪D۹٪۸۸٪D۸٪AF-win۳۲-disk-imager.html

- فایل را از حالت زیپ استخراج کرده و نصب کنید.
- سیستم عامل مورد نیاز برای Raspberry Pi رااز لینک زیر دانلود کنید.

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

- برنامه Win۳۲.Disk را باز کنید.



شکل ۱-۲ : محیط برنامه Win۳۲.Disk

- فایل سیستم عاملی که دانلود کرده اید را در قسمت Image File
- روی گزینه write کلیک کنید.
- در صورت نیازگزینه read برای گرفتن پشتیبانی از sd card می باشد.

۱-۱-۲- اتصالات RASPBERRY PI ۳B

- مانیتور

اکثر مانیتور ها به عنوان نمایشگر برای Pi Raspberry کار می کنند، اما باید از صفحه نمایش با ورود ی 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 ۵V – ۲،۵A نیاز داریم.

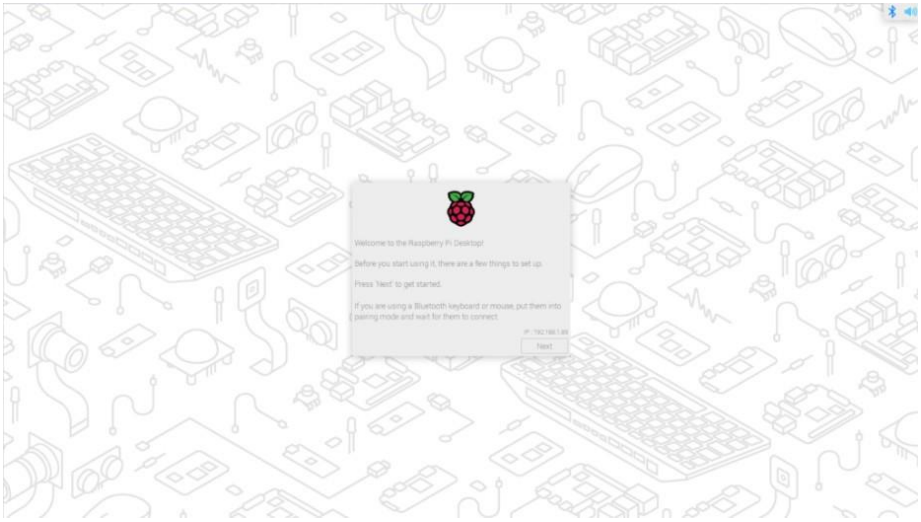
- کارت SD

به یک SD card Micro نیاز داریم که حداقل ٤ گیگابایت ظرفیت داشته باشد.

۱-۱-۳- تنظیمات اولیه سیستم عامل بر روی RASPBERRY PI ۳B

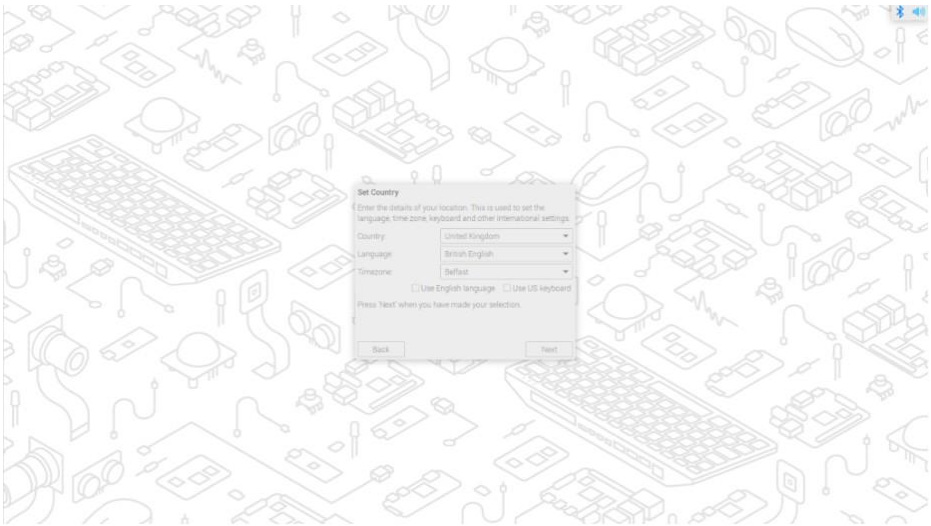
ادوات مورد اشاره در قسمت بالا را متصل کنید.

Raspberry Pi ۳B را روشن نمایید.



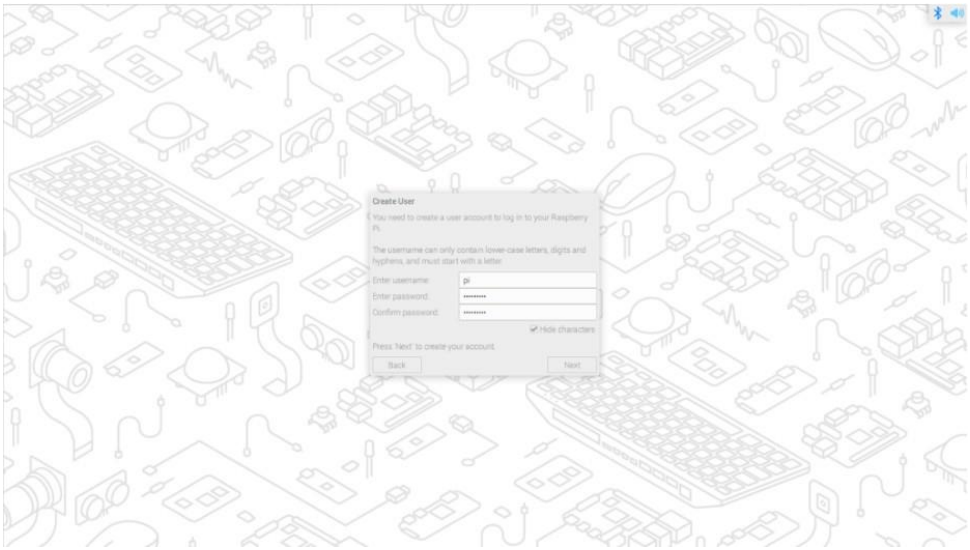
شکل ۱-۳ : پنجره نصبی سیستم عامل Raspberry Pi

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



شکل ۱-٤ : 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.

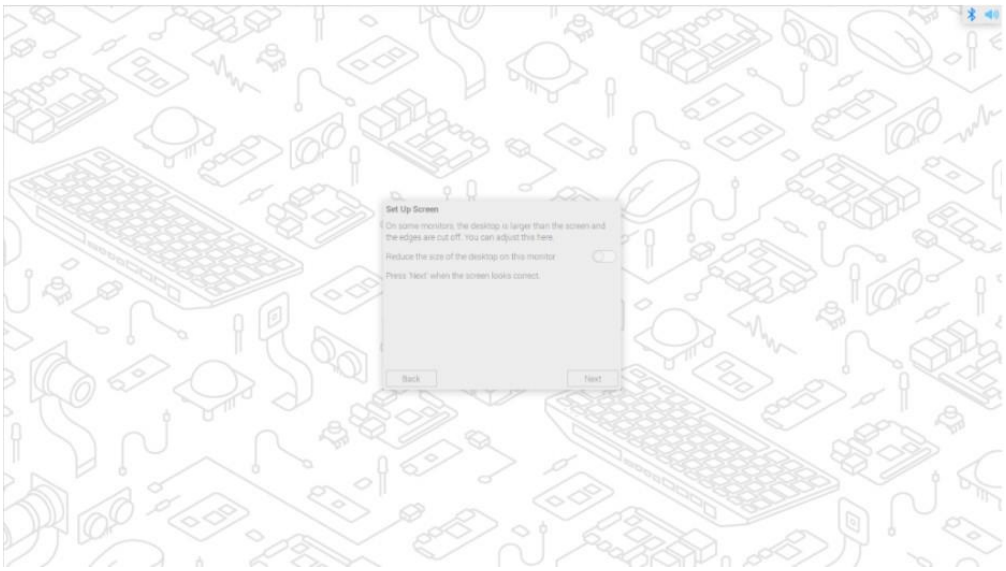


شکل ۱-۵ : تنظیمات حساب کاربری

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 ۱-٦ : 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.

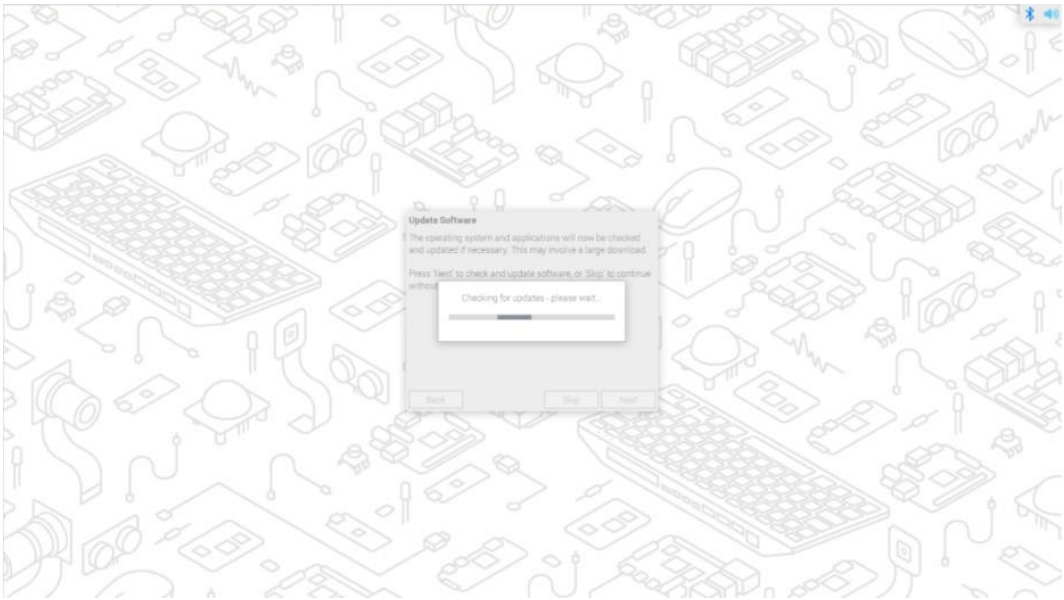


شکل ۱-۷ : پیکربندی صفحه نمایش

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

شکل ۱-۸ : تنظیمات شبکه بی سیم

شبکه بی سیم خود را اضافه کنید



شکل ۱-۹ : 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 مجددا راه اندازی کنید.

۲-۲- کنترل RASPBERRY PI از راه دور

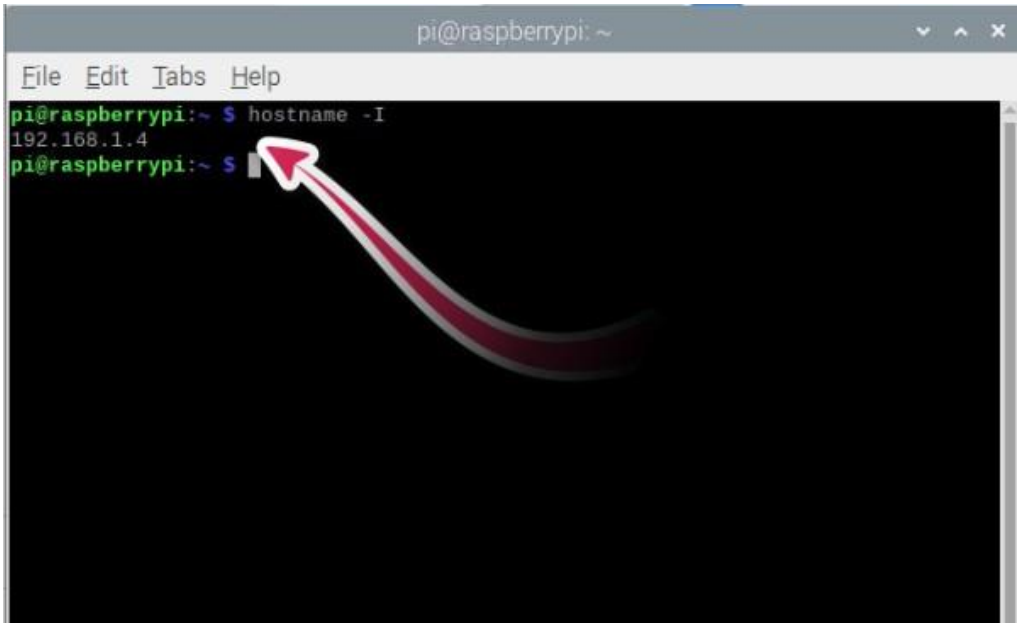
برنامه vnc viewer را از لینک زیر دانلود کنید.

https://soft۹۸.ir/internet/remote-control/۲۳۳٤-realvnc.html

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

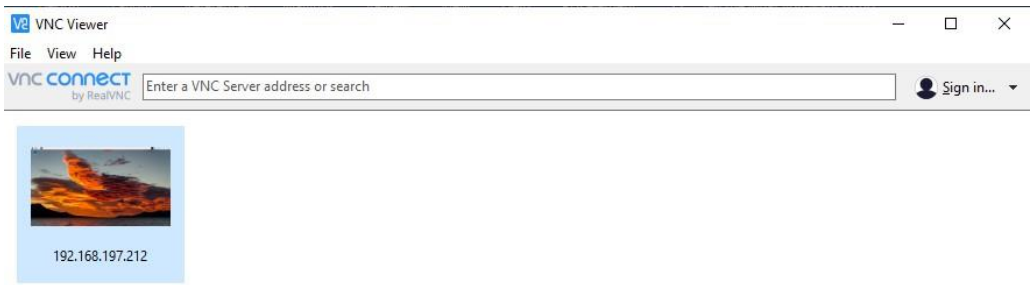
Raspberry Pi و کامپیوتر خود را به یک وای فای مشترک متصل نمایید.

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.

۲-۱- زبان برنامه نویسی ومعرفی نرم افزارکد نویسی

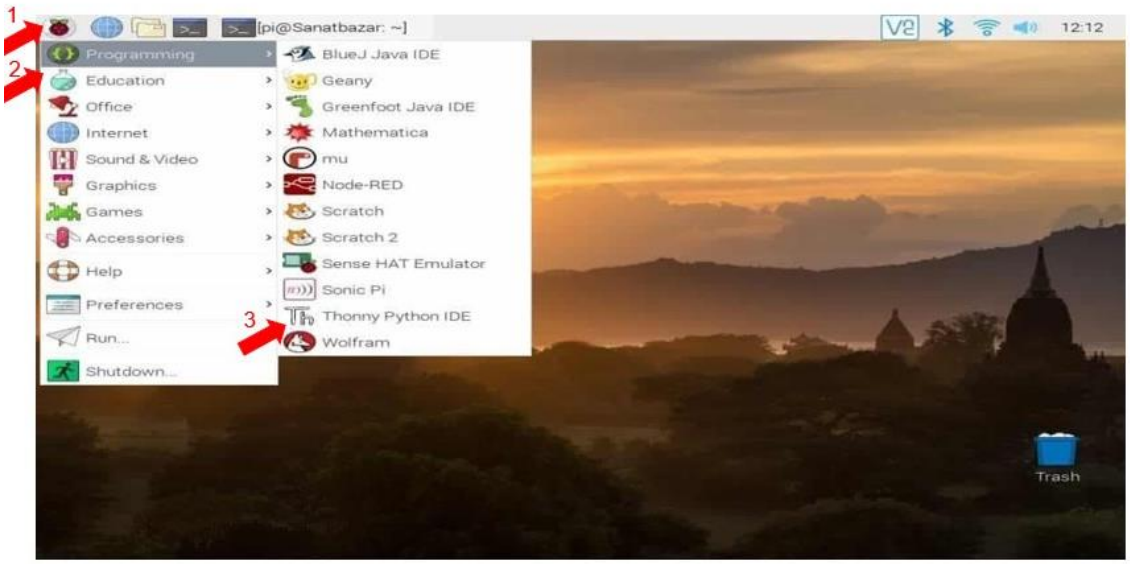
۲-۱-۱- معرفی برنامه THONNY IDE

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

۲- روی گزینه programing نگه دارید.

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

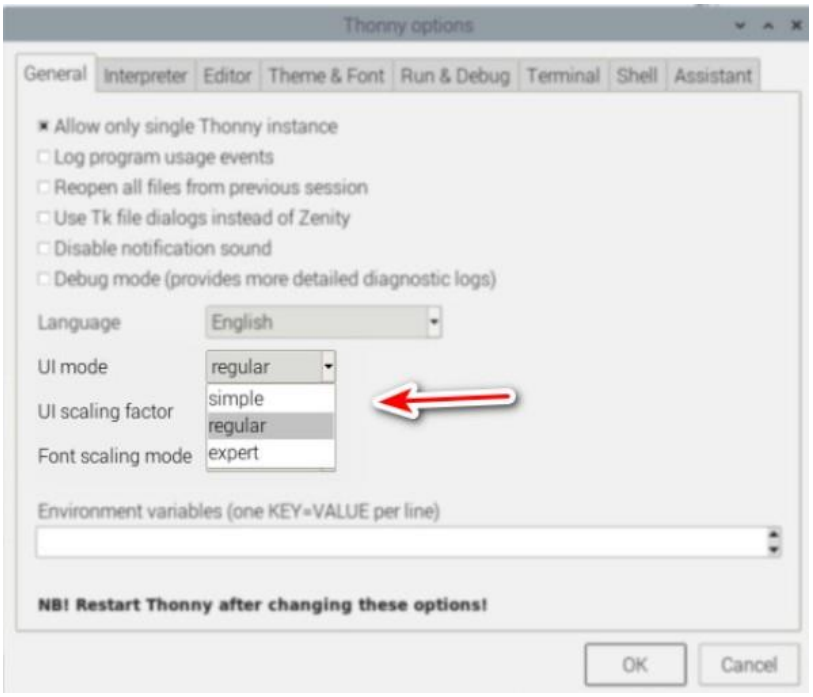
نکته در بعضی از نسخه های سیستم عامل رزبرین ممکن است نرم افزار Thonny IDE به طور پیش فرض نباشد که در این صورت باید آن را نصب کنید.



شکل ۲-۱ : باز کردن نرم افزار THONNY IDE

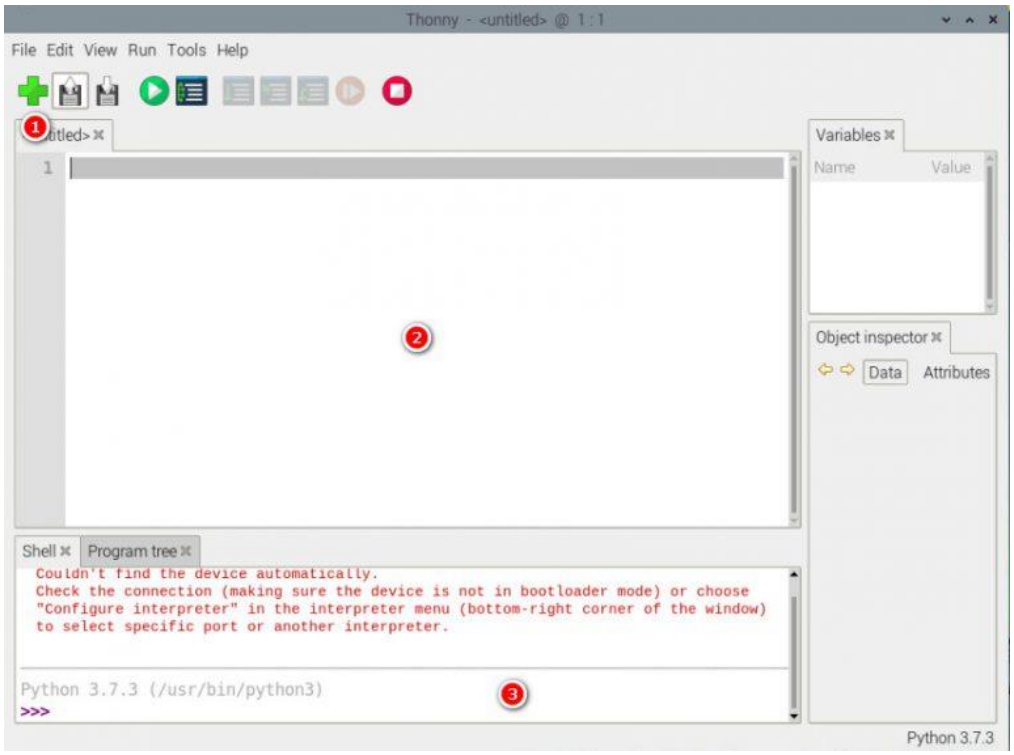
نرم افزار Thonny IDE از دو بخش تشکیل شده است:

Regular Mode و Simple Mode که پیشنهاد می‌شود 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.

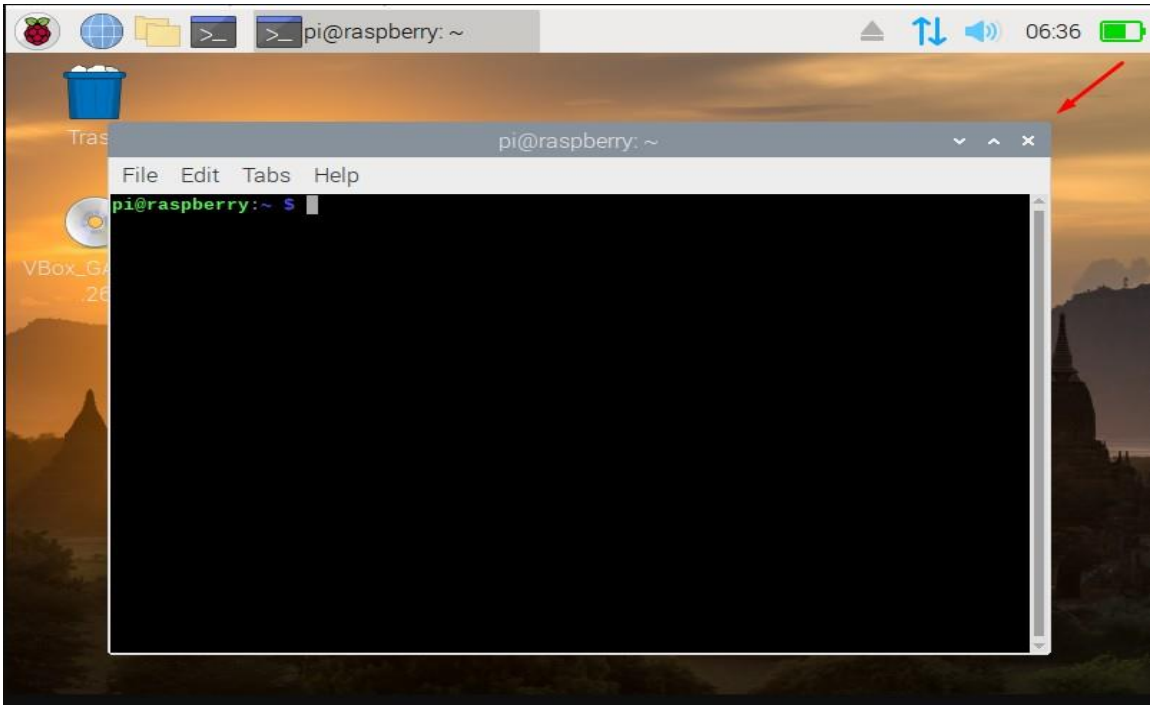
۲: **Script Area** در این محیط برنامه شما نوشته می‌شود. هر خط از برنامه با عدد نمایش داده می‌شود و در صورت اشتباه در کدنویسی در بخش Shell خطی که ایراد دارد را نمایش میدهد.

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 رزبری‌پای، روی آیکون چهارم از سمت چپ در نوار بالا کلیک کنید.

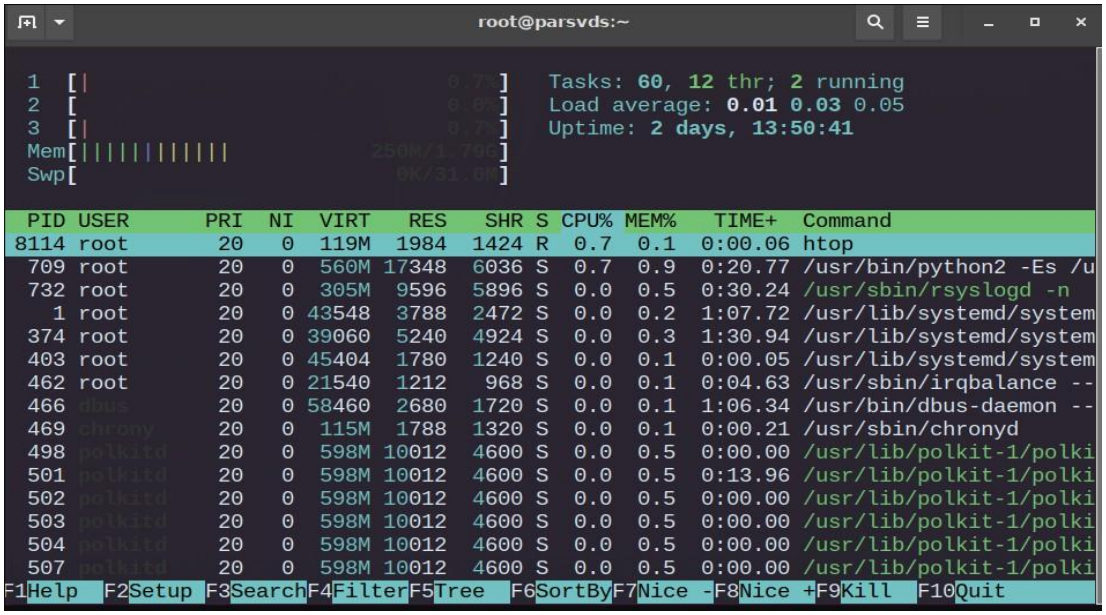
شکل ۲-٤ : آیکون ترمینال رزبری پای



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 اینتراکتیو و یا نیمه گرافیکی است،این دستور پس از اجرا تمامی



شکل ۲-٦ : 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 جرییات 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)

دستورcd desktob : به دسکتاب می رود.

دستور .. cd به دایرکتوری قبلی برمی گردد.

دستور ../.. cd دو مرحله به دایرکتوری قبلی برمی گردد.

- دستور mkdir : این دستور دایرکتوری ایجاد می کند.

نحوه استفاده از این دستور : ابتدا وارد دایرکتوری که می خواهید در آن دایرکتوری جدید ایجاد کنید شوید سپس دستور مقابل را اجرا کنید. (نام دایرکتوری که می خواهید ایجاد کنید – فاصله - mkdir)

- دستور remdir : این دستور دایرکتوری را حذف می کند.

نحوه استفاده از این دستور : ابتدا وارد دایرکتوری که می خواهید در آن دایرکتوری را حذف کنید شوید سپس دستور مقابل را اجرا کنید. (نام دایرکتوری که می خواهید حذف کنید– فاصله - remdir)

برای حذف برخی از فایل ها که دایرکتوری نیستند از دستور rm استفاده میکنیم.

- دستور python –version : این دستور ورژن پایتون سیستم را نشان می دهد.
- دستور nano pi.py برنامه نانو را برای اجرا در پایتون فرا می خواند.

با دکمه control و x و سپس y و در نهایت enter می توان از این برنامه خارج شد.

- دستور python : برنامه پایتون را اجرا می کند.

با دستورquit() می توان از پایتون خارج شد.

با دستور python (file name) .py فایل (file name).py که دارای کد های پایتونی است اجرا می شود.

۲-۲- تمرینات وتکالیف

۱- درباره مفهوم شی گرایی در پایتون تحقیق کنید.

آزمایش۳

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

۳-۱-۱- انواع متغییر ها

- متغییر 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.
- متغییر int : مخفف کلمه integer هست. این نوع برای متغیرهای عددی حاوی عدد صحیح استفاده می‌شود. فقط اعداد منفی و مثبت کامل (بدون نقطه اعشار) را می‌توان در متغیر نوع int ذخیره کرد.
- متغییر float : این نوع متغییر اعداد اعشاری و کسری را نیز علاوه بر اعداد صحیح شامل می شوند.
- متغییر char یا string : در نوع داده‌ای کاراکترهای یونی‌کد(Unicode) ، حروف و کلمات قرار می‌گیرند. نوع داده‌هایی که به صورت کلمه و جمله هستند، «string»و داده‌هایی که از حروف تشکیل شده‌اند نیز «char»گفته می‌شود .char یک حرف و String مجموعه‌ای از حروف یا کلمات است. در نوع char تنها امکان ذخیره‌سازی یک کاراکتر واحد وجود دارد.
- متغییر 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.

- متغییر 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.
- متغییر tuple : تاپل Tuple یک مجموعه (Collection) از داده ها است که مرتب و غیر قابل تغییر است.که با پرانتز () ساخته می شود.

```
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 ۳-٤ :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 ۳-٦ : 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 = ۰. 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.

۳-۲- تمرینات وتکالیف

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 ٤

٤-١- آشنایی با زبان برنامه نویسی پایتون جلسه دوم

٤-١-١- 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 ٤-١ : مثال دسترسی به اعضای TUPLE()

- دسترسی به اعضای دیکشنری

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

```
my_dict = {'name':'python', 'age': 26}

print(my_dict['name'])
```

```
python
```

شکل ٤-٢ : 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 ٤-٣ : 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")
```

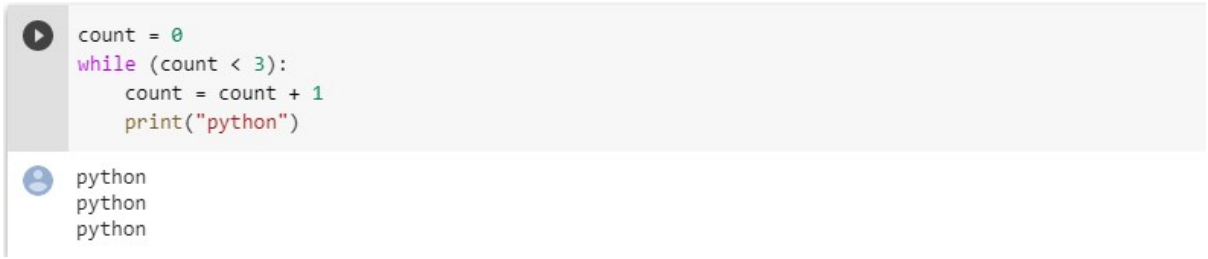
```
i is greater than 15
i'm in else Block
i'm not in if and not in else Block
```

شکل ٤-٤ : 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.



شکل ٤-٥ : مثال دستور WHILE

- دستور while-else

مانند حلقه‌هایfor ، در حالی که حلقه‌ها می‌توانند یک بلوک else اختیاری نیز داشته باشند. قسمت else در صورتی اجرا می شود که شرط در حلقه while به False ارزیابی شود.

حلقه while را می توان با یک دستور break خاتمه داد. در چنین مواردی، قسمت دیگری نادیده گرفته می شود. 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 ٤-٦ : مثال تابع WHILE-ELSE

٤-٢- تمرینات وتکالیف

١- انواع نام گذاری پین های رسپبری پای BCM وBoard را بررسی کنید.

٢- تفاوت gpiozero باrpi.gpio را بیان کنید.

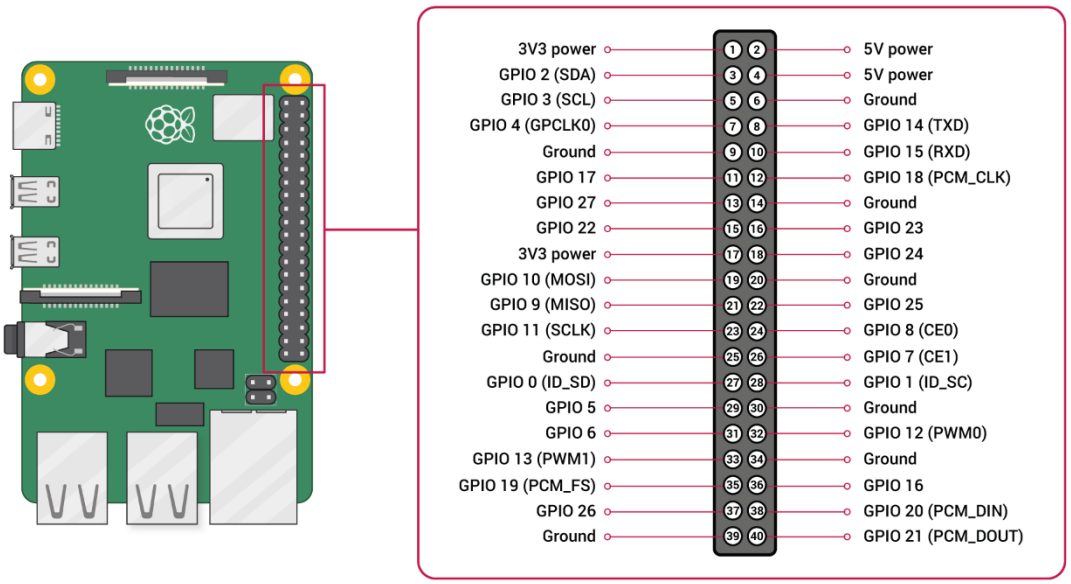
٣- مفهوم PWM توضیح دهید.

۵-۱- 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.

۵-۱-۱- اتصالات مدار



شکل ۵-۱ : 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 محتلف

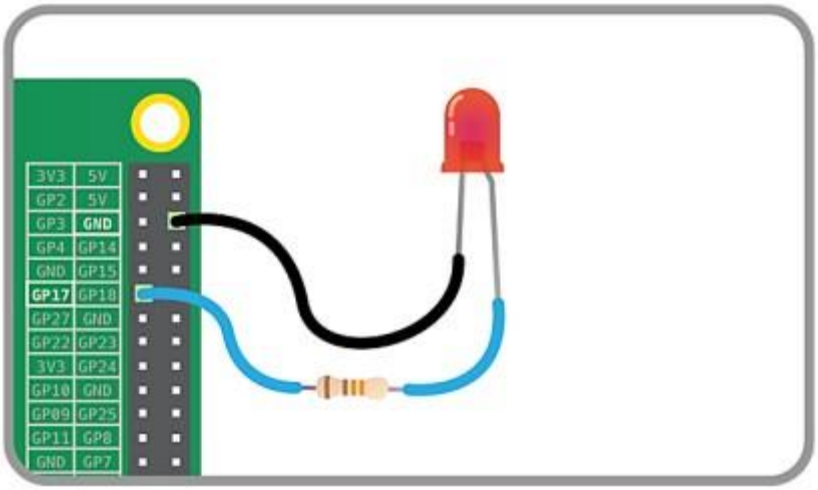
هر یک از این ۱۷ پین GPIO می توانند حداکثر جریان ۱۵ میلی آمپر را ارائه دهند. و مقدار جریان از تمام GPIO نمی تواند از ۵۰ میلی آمپر فراتر رود. بنابراین می توان از هر یک از این پین های GPIO حداکثر ۳ میلی آمپر را بکشیم. به منظور انجام پروژه یکی از این پین ها را به دلخواه انتخاب کنید.

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 GPIO۱۷ 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 باشد.

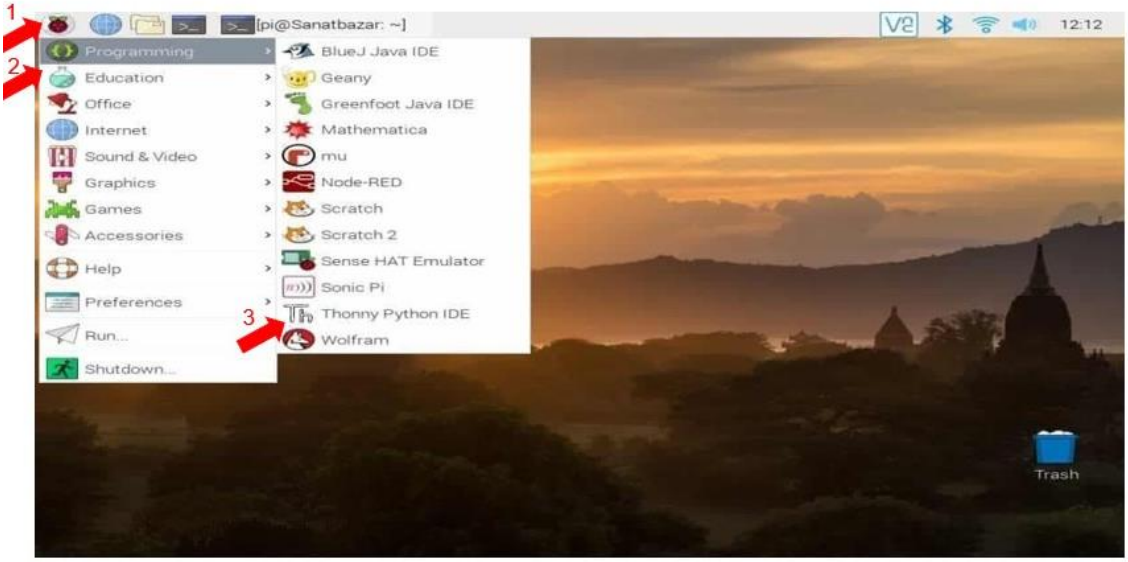


شکل ۵-۳ : سیم بندی LED

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 ۵-٤ : 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 [۱]: from gpiozero import LED

Line [۲]: led = LED(۱۷)
```

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

```
Line [۳]: led.on()
```

برای خاموش کردن LED، فرمان مقابل را تایپ کنید:

```
Line [٤]: led.off()
```

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

٤- 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 کد های 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 [١]: from gpiozero import LED
Line [٢]:   from time import sleep
Line [٣]:   led = LED(١٧)
Line [٤]:   while True:
Line [٥]:         led.on()
Line [٦]:         sleep(١)
Line [٧]:         led.off()
Line [٨]:         sleep(١)
```

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 کد های 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 بخواهیم 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 GPIO۱۷به 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 [۱]: import RP۱.GPIO as GPIO
Line [۲]: #GPIO.setmode(GPIO.BCM)
Line [۳]:  GPIO.setmode(GPIO.BOARD)
Line [٤]: GPIO.setwarnings(False)
Line [٥]: GPIO.setup(۱۱,GPIO.OUT)#۱۷
Line [٦]:  my_pwm-GPIO.PWM(۱۱,۱۰۰)
Line [۷]: my_pwm.start(۱۰۰)
```

۵-۲- تمرینات وتکالیف

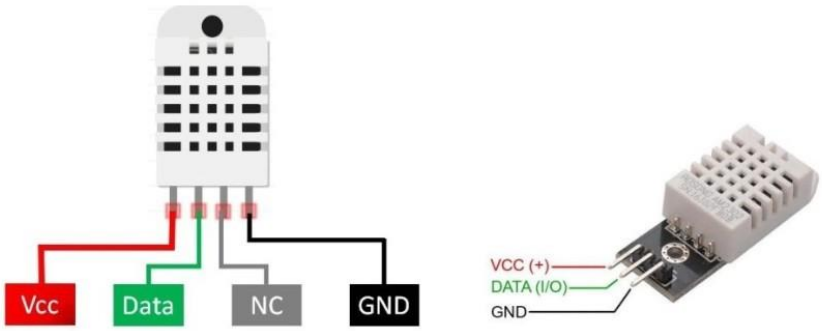
۱- در باره طرز کارکرد سنسور التراسونیک مدل HC SR۰٤ را با جزیات کافی جهت پیاده سازی با رزبری پای تحقیق کنید.

آزمایش ٦

٦-١- راه اندازی سنسور DHT ١١

- سنسور DHT١١ مقادیر رطوبت و دما را اندازه‌گیری کرده و به صورت سریال روی یک خط مهیا می‌سازد.
- این سنسور می‌تواند رطوبت را به صورت درصدی(٢٠ تا ٩٠ درصد RH)اندازه‌گیری کرده و دما را در محدوده ٠ تا ٥٠ درجه سلسیوس اندازه‌گیری کند.
- سنسور دارای ٤ پین است؛ که یکی از این پین‌ها برای ارسال داده به شکل سریال به کار می‌رود.
- پالس‌های TON و TOFF به عنوان منطق ١ یا منطق ٠ یا پالس شروع و پایان فریم دیکود می‌شوند.

٦-١-١- اتصالات مدار



شکل ٦-١ : پایه های سنسورDHT٢٢

١- تعریف پایه های سنسورDHT١١

٢- نحوه اتصال بین سنسور DHT٢٢ و رزبری پای

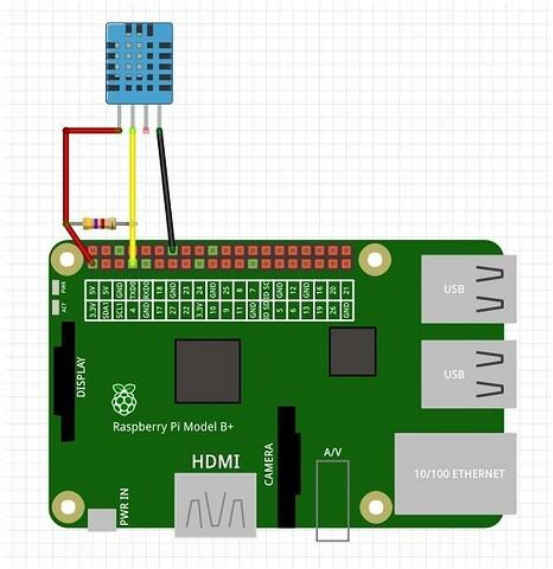
همانند تصویر زیر برای اتصال سنسور DHT۲۲ و رزبری پای مراحل زیر را دنبال می کنیم:

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)

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



شکل ٦-۲ : سیم بندی سنسور DHT۱۱

٦-۱-۲- برنامه نویسی

در این مثال از کتابخانه‌ی پایتون سنسورDHT ، نوشته Adafruit در گیت‌هاب استفاده می‌کنیم. کتابخانه‌ی Adafruit سنسورDHT ، برای خواندن رطوبت و دما روی رزبری‌پای یا Beaglebone Black نوشته شده است. این کتابخانه برای سنسورهایی مانند DHT۲۲ ،DHT۱۱ یا AM۲۳۰۲ قابل استفاده می باشد.

کتابخانه ی Adafruit سنسور DHT را ازلینک 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.

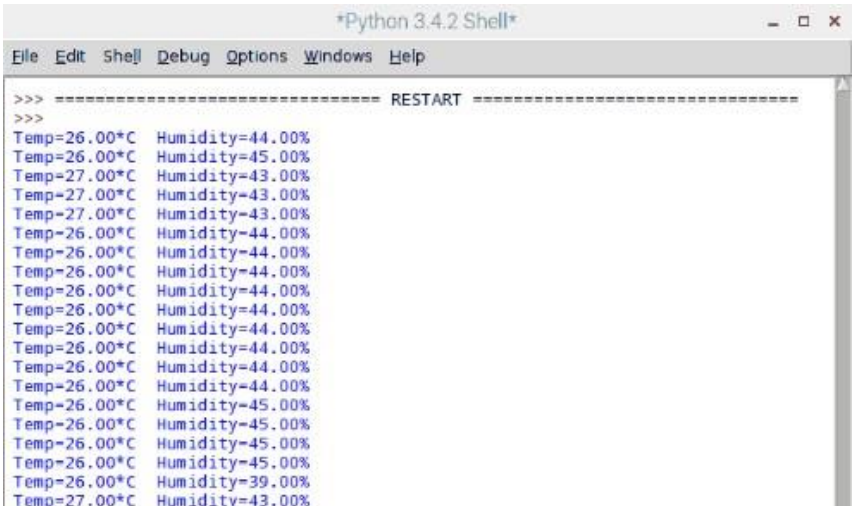
sudo python setup.py install

برنامه Thonny IDE را باز کرده وکد 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 ٦-٣ : کد راه اندازی سنسور دما ورطوبت

خروجی به صورت زیر خواهد بود.



شکل ٦-٤ : پنجره خروجی کد سنسور دما ورطوبت

٦-٢- راه اندازی سنسور اولتراسونیک

سنسورهای اولتراسونیک موجی در فرکانس معمولاً بین ۲۳ کیلوهرتز و ٤٠ 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.

سنسور های 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 برا ی یافتن طعمه خود استفاده میکنند . از آنجایی که سرعت صوت در هوا در دمای اتاق ۳٤۳ متر در ثانیه است، این زمان را میتوان به راحتی به مسافت تبد یل کرد.

٦-٢-١- 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 .

٦-٢-٢- اتصالات مدار

- معرفی پایه های سنسور التراسونیک

سنسوری که ما قصد راه اندازی اون رو داریم ، سنسور ٠٤- SR هست که این سنسور دارا ی ٤ پایه می باشد.

شکل ٦-٥ :پایه های سنسور التراسونیک

- سیم بندی مدار

اتصال VCC به پین شماره ٢(٥ V)

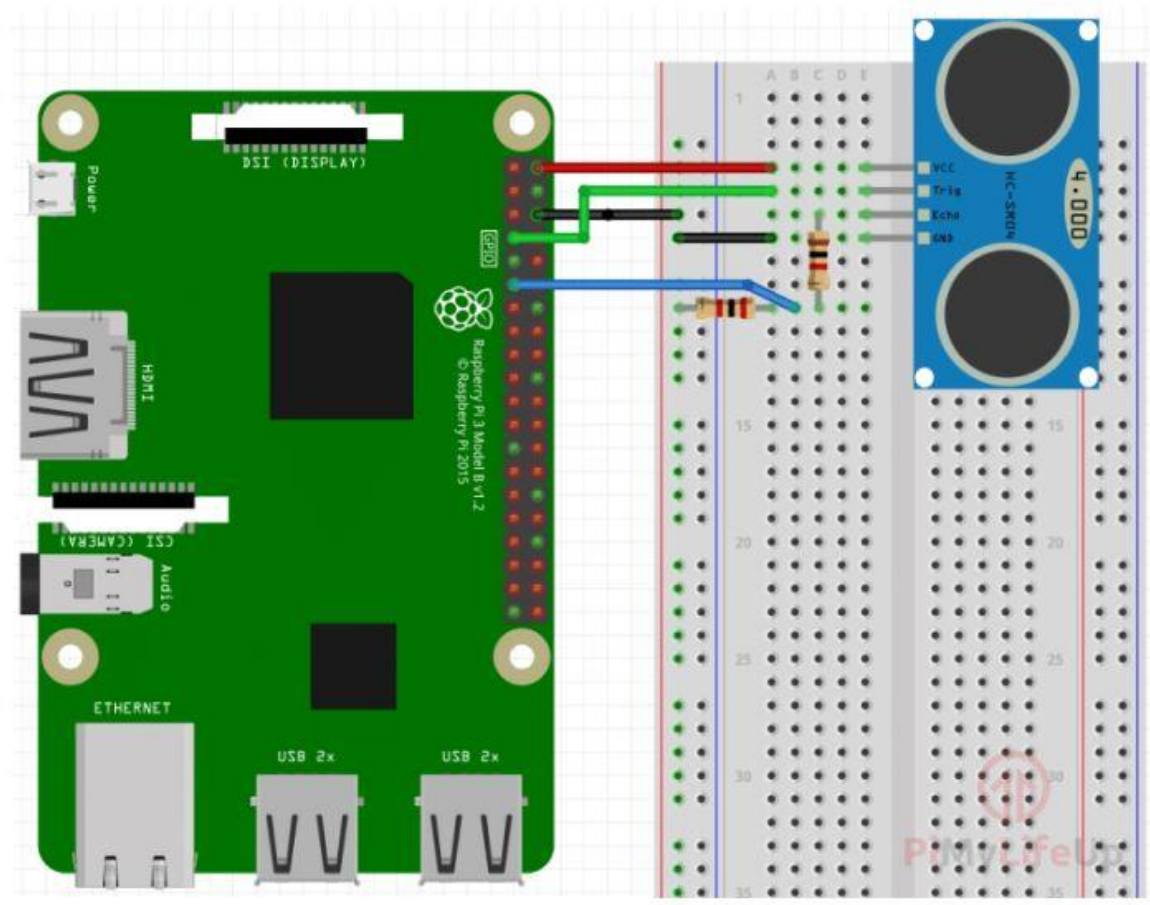
اتصال Trig به پین شماره ٧(GPIO ٤)

اتصال Echo به مقاومت(١K Ω R١)

اتصال مقاومت (٢K Ω R٢) بین R١ و زمین

اتصال نقطه‌ی میانی R١ و R٢ به پین شماره ١١

اتصال GND به پین شماره ٦(Ground)



شکل ٦-٦ : سیم بندی سنسور التراسونیک

٦-٢-٣- 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 = ۷
ECHO = ۱۱
```

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(۲)
```

سنسور SRF۰٤ 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 ٤۰ کیلوهرتز(برای دریافت پاسخ ازECHO) ارسال کند. بنابراین ابتدا پایه TRIGرا 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(۰,۰۰۰۰۱)
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)==۰:
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 دقیقا قبل از لحظه ای است که پایه به حالت HIGH برود .

حال وقتی که سیگنالی دریافت شد ، پایه به حالت HGIH 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 whileرا 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)==۱:
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 ۱۷۱۵۰
```

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

```
distance = round(distance, ۲)
```

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 وکد زیر را وارد نمایید.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO. BOARD)
TRIG = ۷
ECHO = ۱۱
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(۲)
GPIO.output(TRIG, True)
time.sleep(۰,۰۰۰۰۱)
GPIO.output(TRIG, False)
while GPIO.input(ECHO)==۰ :
pulse_start = time.time ( )
while GPIO.input(ECHO)==۱ :
```

```
pulse_end = time.time ( )

pulse_duration = pulse_end - pulse_start

distance = pulse_duration * ۱۷۱۵۰

distance = round(distance, ۲)

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

GPIO.cleanup( )
```

خروجی به صورت زیر خواهد بود.

```
Distance Measurement In Progress

Waiting For Sensor To Settle

Distance: ۲۹,٦۹ cm
```

٦-۳- تمرینات و تکالیف

۱- طرز کارکرد و اتصال LCD ۱٦*۲ با استفاده از لینک زیر مطالعه نمایید.

https://pimylifeup.com/raspberry-pi-lcd-۱٦x۲/

۷-۱- راه اندازی LCD ۱٦*۲ کاراکتری

- LCD ۱٦*۲:

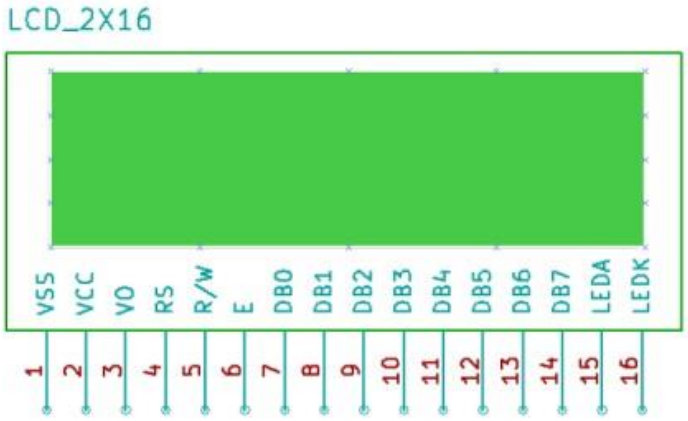
ال سی دی مورد استفاده در این پروژه LCD ۱٦*۲ می باشد و برای راه اندازی آن از مد ٤ بیت استفاده شده و آی سی کنترلر این ال سی دی HD Hitachi ٤٤۷۸۰ می باشد.

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

جدول ۷-۱ : پایه های ال سی دی

PIN NO	Symbol	Fuction
۱	VSS	GND
۲	VDD	+٥V
۳	V۰	Contrast adjustment
٤	RS	H/L Register select signal
٥	R/W	H/L Read/Write signal
٦	E	H/L Enable signal
۷	DB۰	H/L Data bus line
۸	DB۱	H/L Data bus line
۹	DB۲	H/L Data bus line
۱۰	DB۳	H/L Data bus line
۱۱	DB٤	H/L Data bus line
۱۲	DB٥	H/L Data bus line
۱۳	DB٦	H/L Data bus line
۱٤	DB۷	H/L Data bus line
۱٥	A	+٤.۲v for LED

١٦	K	Power supply for BKL(٠v)



شکل ٧-١ : ٢*١٦ LCD

٧-١-١- اتصالات

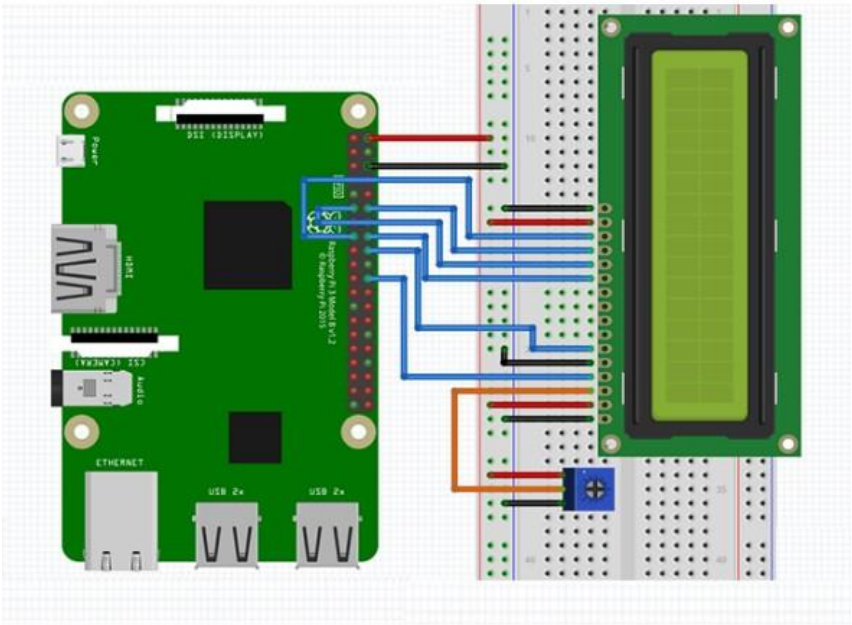
همانند تصویر زیر برا ی اتصال ٢*١٦ LCD و رزبری پای مراحل زی ر را دنبال می کنیم :

اتصال پین مثبت ٢*١٦ LCD به تغذ یه ٥ ولت برد رزبر ی پای (سیم قرمز)

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

اتصال پین های دیتا ٢*١٦ LCD

قرار دادن پتانسیومتر ۱ کیلو اهم جهت تنظیم میزان روشنایی



شکل ۷-۲ : سیم بندی LCD ۱۶*۲

۱-Pin ۱ (Ground) goes to the ground rail.

۲-Pin ۲ (VCC/۵v) goes to the positive rail.

۳-Pin ۳ (V۰) goes to the middle wire of the potentiometer.

٤-Pin ٤ (RS) goes to GPIO۲٥ (Pin ۲۲)

٥-Pin ٥ (RW) goes to the ground rail.

٦-Pin ٦ (EN) goes to GPIO۲٤ (Pin ۱۸)

۷-Pin ۱۱ (D٤) goes to GPIO۲۳ (Pin ۱٦)

۸-Pin ۱۲ (D٥) goes to GPIO۱۷ (Pin ۱۱)

۹-Pin ۱۳ (D٦) goes to GPIO۱۸ (Pin ۱۲)

۱۰-Pin ۱٤ (D۷) goes to GPIO۲۲ (Pin ۱٥)

۱۱-Pin ۱٥ (LED +) goes to the positive rail.

۱۲-Pin ۱٦ (LED -) goes to the ground rail.

۷-۱-۲- برنامه نویسی

قبل از اسکریپت نویسی پایتون برای LCD ۱٦X۲ 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 می‌کنیم:

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

حال قبل از دانلود کتابخانه LCD ۱٦X۲ با وارد کردن دستور زیر در ترمینال از نصب بودن آخرین نسخه پکیج‌های setuptools, wheel و pip مطمئن می‌شویم:

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

اکنون با استفاده از دستور زیر کتابخانه پایتون برای ارتباط LCD ۱٦X۲ 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 ۱٦X۲ را نصب می‌کنیم:

```
cd ./Adafruit_Python_CharLCD
sudo python setup.py install
```

حال با استفاده از کد زیر از کتابخانه پایتون LCD ۱٦X۲ می‌توانیم در تمامی اسکریپت‌های پایتون استفاده کنیم:

```
import Adafruit_CharLCD as LCD
```

ارتباط با LCD۱٦X۲ در پایتون بسیار راحت می‌باشد. فقط کافی است ابتدا پین‌های متصل‌شده مربوطه به LCD ۱٦□۲ با رزبری را در ابتدای اسکریپت پایتون تعریف کنیم:

lcd_rs = ٢٥

lcd_en = ٢٤

lcd_d٤ = ٢٣

lcd_d٥ = ١٧

lcd_d٦ = ١٨

lcd_d٧ = ٢٢

lcd_backlight = ٤

lcd_columns = ١٦

lcd_rows = ٢

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

lcd = LCD.Adafruit_CharLCD(lcd_rs, lcd_en, lcd_d٤, lcd_d٥, lcd_d٦, lcd_d٧, 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 می‌دهیم و پس از ٥ 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 = ٢٥

lcd_en = ٢٤

lcd_d٤ = ٢٣

lcd_d٥ = ١٧

lcd_d٦ = ١٨

```
lcd_d٧ = ٢٢

lcd_backlight = ٢

lcd_columns = ١٦

lcd_rows = ٢

lcd = LCD.Adafruit_CharLCD(lcd_rs, lcd_en, lcd_d٤, lcd_d٥, lcd_d٦, lcd_d٧, lcd_columns, lcd_rows, lcd_backlight)

lcd.message(Redronic')

time.sleep (٥)

lcd.clear()

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

time.sleep(٥)

lcd.clear()

lcd.message('GoodbyenWorld!')

time.sleep(٥)

lcd.clear()
```

۷-۲- تمرین کلاسی

حال برنامه ای بنویسید که دما ورطوبت را در ال سی دی کاراکتری نمایش دهد.

۷-۳- تمرینات و تکالیف

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

آزمایش ۸

۸-۱- راه اندازی دوربین با رزبری پای

دوربین های رزبر ی پای به صورت رسمی در دو نسخه ی ۵ (در سال ۲۰۱۳) و ۸ مگاپیکسلی (در سال ۲۰۱٦)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 و یژگی های این ماژول می توان به ابعاد کوچک(۹*۲٤*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 ار 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 V۱	Camera Module V۲
اندازه	۲۰mm * ۲۵ mm * ۹ mm	۲۰mm * ۲۵ mm * ۹ mm
وزن	۳g	۳g
کیفیت	۵ Megapixels	۸ Megapixels
رزولوشن	۲۵۹۲ * ۱۹٤٤ Pixels	۳۲۸۰ * ۲٤٦٤ Pixels
فیلم برداری	۱۰۸۰p @ ۳۰fps, ۷۲۰p @ ٦۰fps and ٦٤۰x٤۸۰p ٦۰/۹۰fps	۱۰۸۰p @ ۳۰fps, ۷۲۰p @ ٦۰fps and ٦٤۰x٤۸۰p ٦۰/۹۰fps

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 ز یست 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

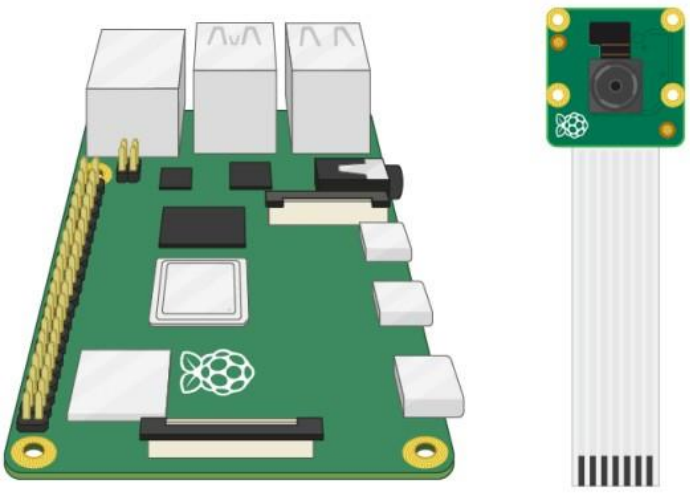
در ابتدا اطمینان حاصل میکنیم که رزبری پای خاموش باشد. سپس مراحل زیر را اجرا میکنیم :

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 .

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

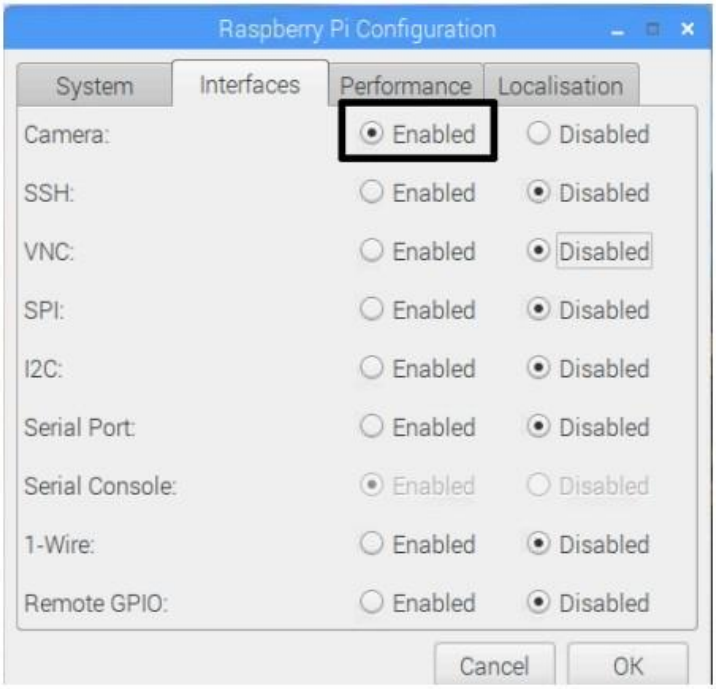


شکل ۸-۱ : اتصال دوربین به رزبری پای

۸-۱-۲- برنامه نویسی

۱- رزبری پای را روشن میکنید.

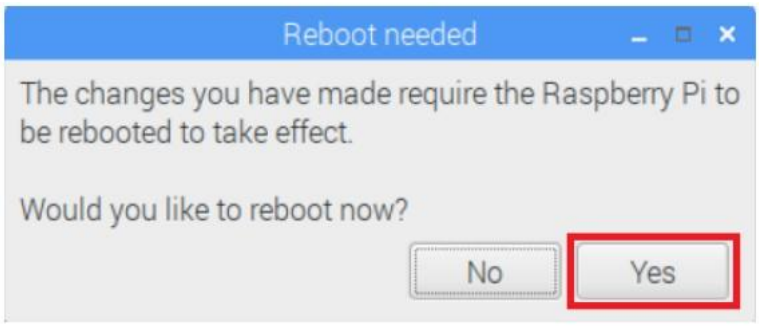
۲- به منوی اصلی رزبری پای رفته و پس از آن بر زیر منوی perferences کلیک میکنیم سپس Configuration Pi Raspberry را باز میکنیم.



شکل ۸–۲ : PI-CONFIGURATION-MENU

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

٤- سیستم عامل را Reboot کرده و مجددا رزبری پای را راه اندازی میکنیم.

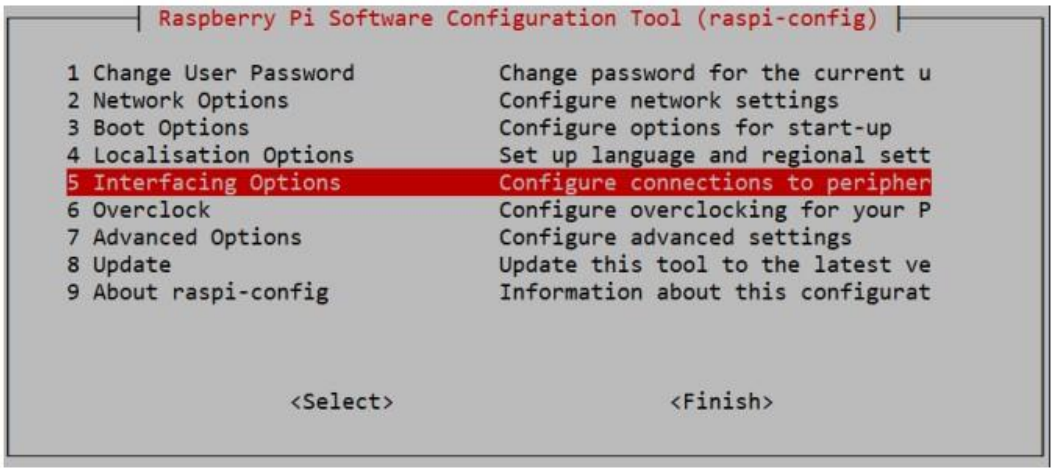


شکل ۸–٤ : REBOOT سیستم عامل

شکل ۸–۳ : PI-CONFIGURATION-INTERFACES-ANNOTATED

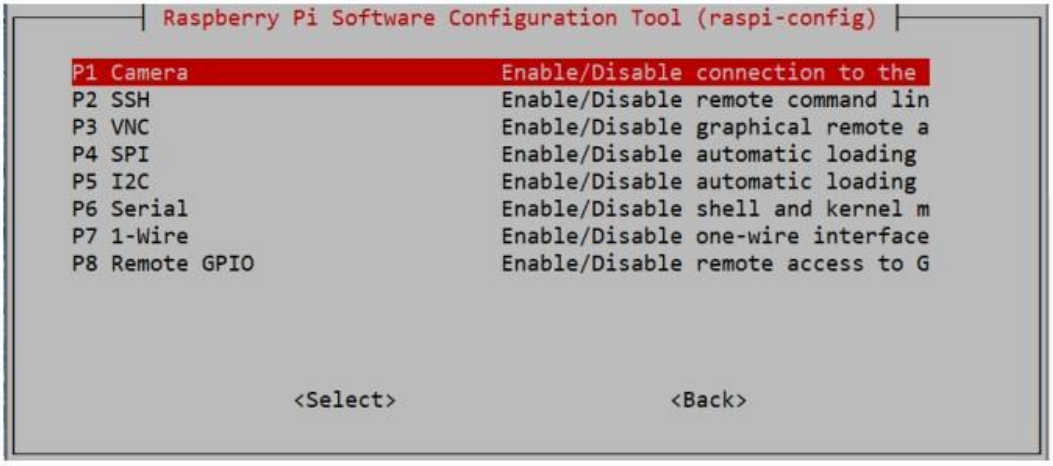
٥- دوربین را به رزبری پای متصل میکنیم.

٦- 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

٧- به بخش Options Interfacing میرویم. دوربین را فعال کرده و رزبر ی پای خود را Reboot میکنیم.



شکل ۸-٦ : 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 راهنمایrasistill ، برا ی گرفتن یک عکس ساده و ذخیره ی آن می توان از دستور زیر استفاده کرد:

raspistill -o نام و محل ذخیره عکس

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 سوییچ t- کنترل کرد:

raspistill -t ۱۰۰۰ -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 سوییچ w- و h- برای تغییر سایز و اندازه ی تصاویر گرفته شده استفاده می شود(w- برای width و h- برا یheight) برای مثال می خواهیم تصویری با ابعاد ٦٤٠ * ٤٨٠ ایجاد کنیم:

raspistill -w ٦٤٠ -h ٤٨٠ -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 دستورpreview_start ، دوربین شروع به تصویربرداری می کند. اگر رزبری پای ما مستقیما 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(٥)

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.

۸-٤- تمرین کلاسی

برنامه ای بنویسید که ۱۰ عکس متوالی بگیرد.

۸-٥- فیلم برداری با استفاده از اسکریپت پا یتون

برا ی یک فیلم برداری ساده از دستورات زیر استفاده می کنیم:

```
from camera import PiCamera

time import

camera = PiCamera()

camera.start_recording('test.h٢٦٤')

time.sleep(١٠)

camera.stop_ recording()
```

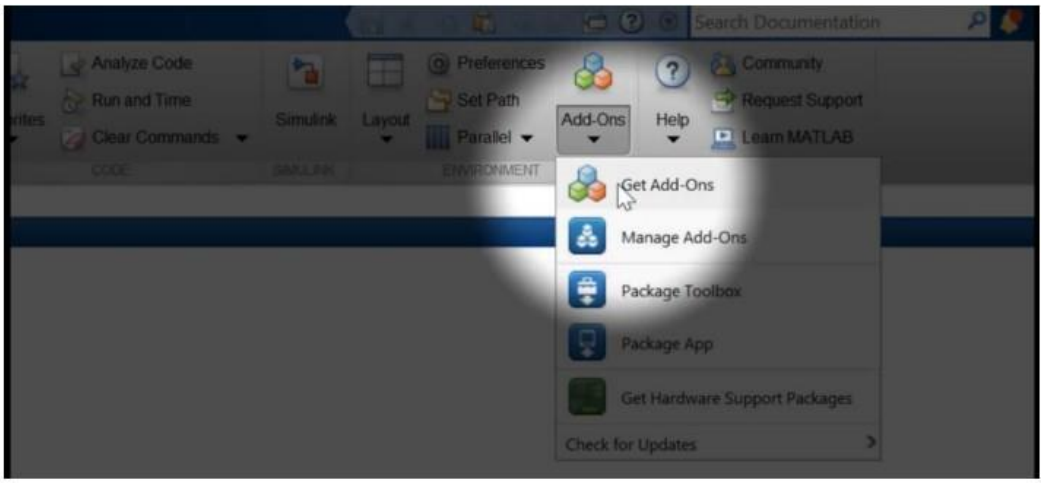
با این دستور، برد رزبری پای به مدت ۱۰ ثانیه فیلم برداری می کند.

۸-٦- تمرینات و تکالیف

۱- درباره طرز نصب و راه اندازی matlab support package for raspberry pi با توجه به لینک زیر برای ایجاد ارتباط بین رسپبری پای و متلب تحقیق کنید.

https://www.youtube.com/watch?v=۳۲ByiUdOwsw

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 Ons-Add 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.

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

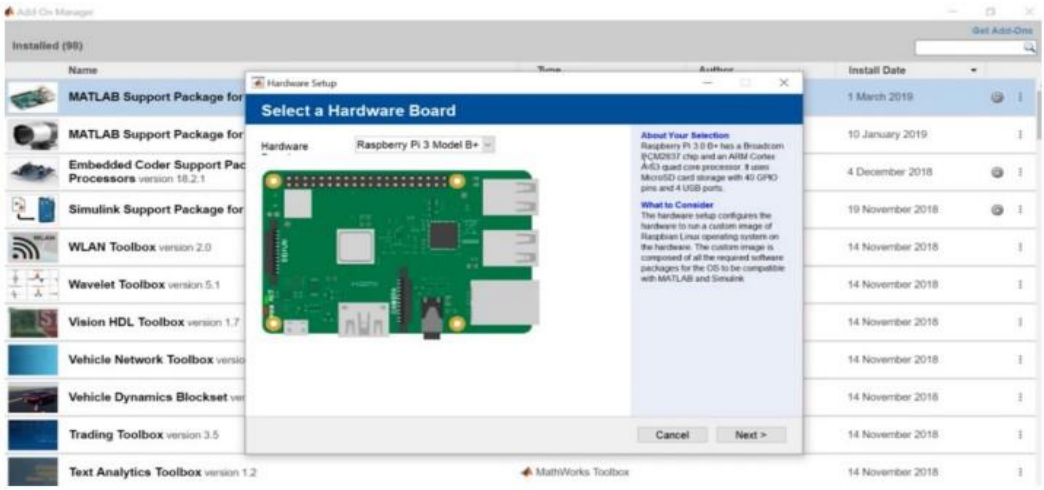
شکل ۹-٤ : ورود به حساب کاربری MATHWORK

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

٦- دوباره در صفحه اصلی رو آیکون Ons-Add کلیک کرده و گزینه Ons-Add Manage کلیک میکنیم تا پنجره زیر باز شود.

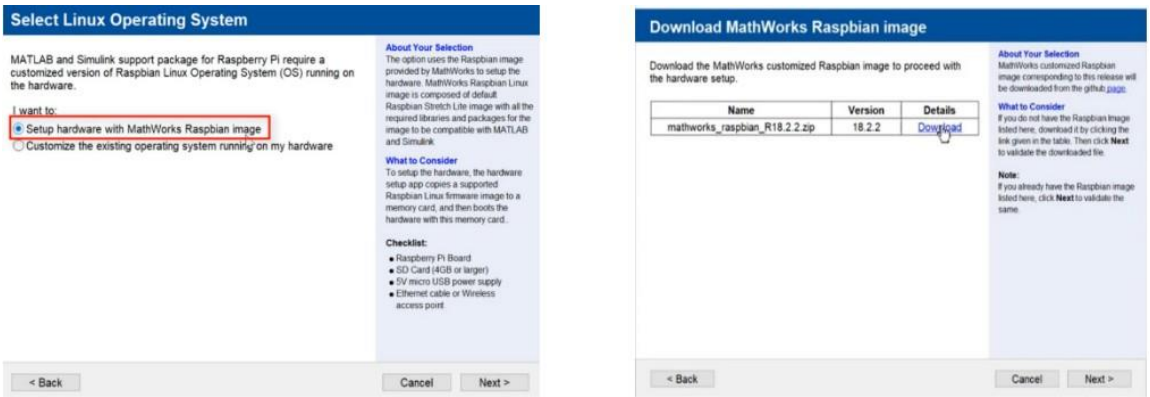
٧- در پنجره فوق در گزینه اول روی علامت setting یا چرخدنده کلیک کرده تا پنجره Select Bord a Hardware باز شود. از قسمت Hardware نوع و مدل بورد رزبری پای خود را انتخاب میکنیم. و در ادامه روی Next کلیک مینماییم.

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



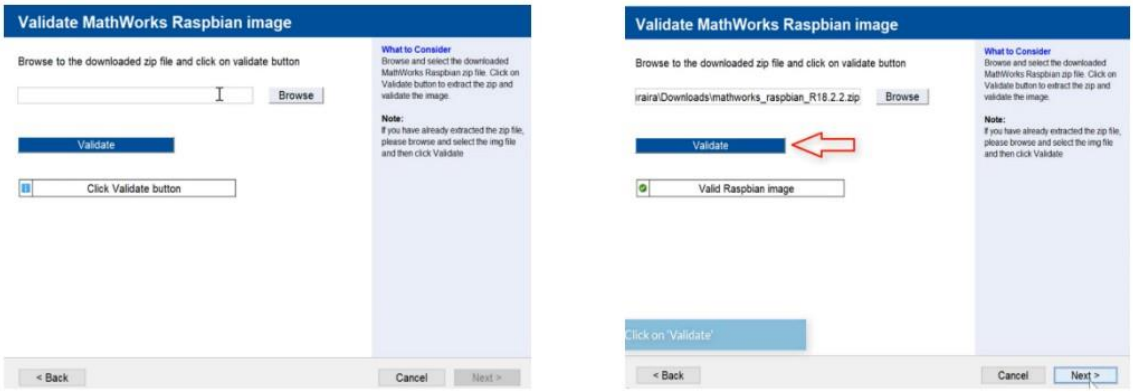
شکل ٩-٥ : مراحل نصب

انتخاب میکنیم. در پنجره جدید روی Download کلیک میکنیم.



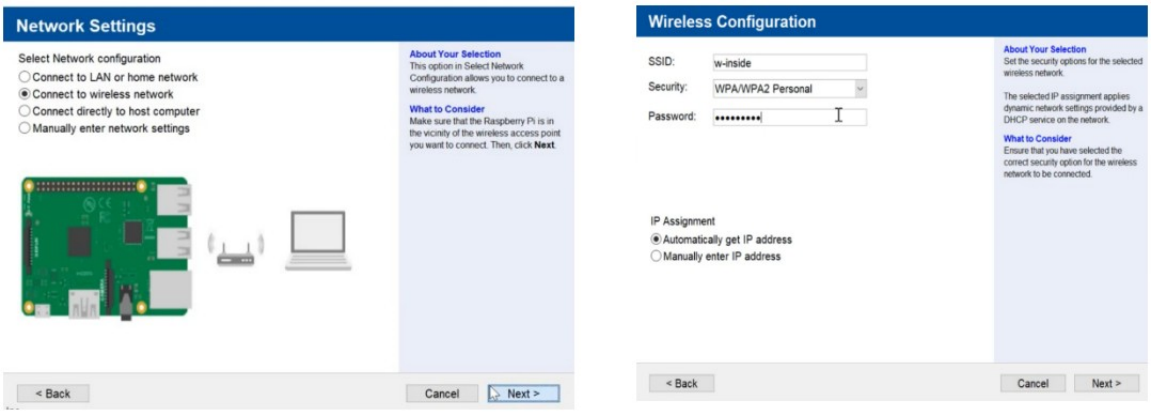
شکل ٩-٦ : مراحل نصب

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



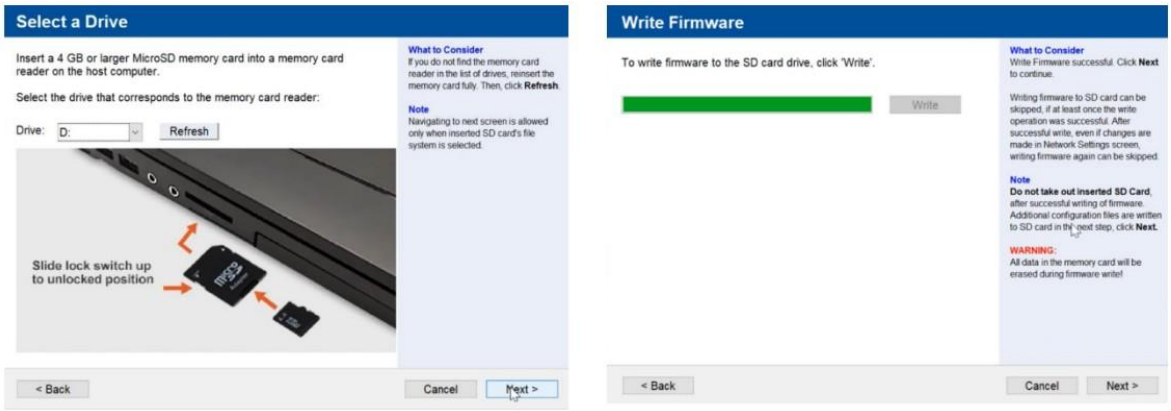
شکل ۹-۷ : مراحل نصب

۱۰- در ادامه در پنجره زیر نحوه اتصال متلب به رزبری پای مشخص میگردد، برای اتصال بصورت Wirelessمشخصات شبکه رابط یا وایفای متصل به رایانه و رزبری را وارد میکنیم.



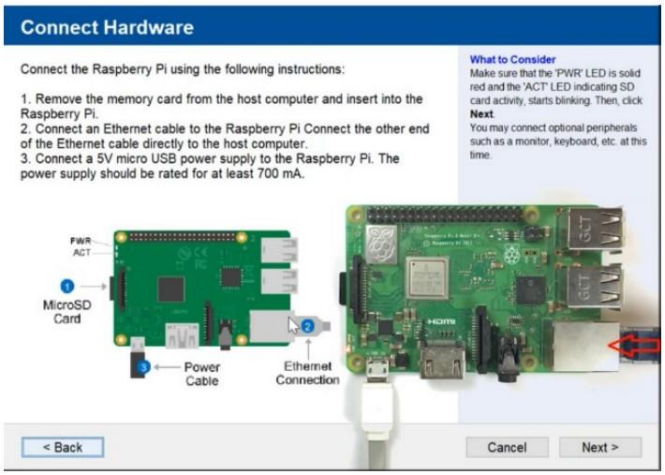
شکل ۹-۸ : مراحل نصب

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 .

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

۹-۲- تمرین کلاسی

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

۹-۳- چارچوب تشخیص اشیاء ویولا-جونز

۹-۳-۱- 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 .

مقاله ی اصلی سه نوآوری کلیدی ارائه می کرد. اولین نوآوری استفاده از نمایشی برای تصویر به نام تصویر انتگرالی بود که به کمک آن امکان محاسبه بسیار سریع و یژگی های مورد نیاز برای تشخیص در زمان ثابت فراهم می گردید

نوآوری دوم توسعه یک الگوریتم یادگیری ماشین مبتنی بر آدابوست به منظور انتخاب مجموعه ای کوچک اما کارا از میان مجموعه بزرگی از ویژگی های تصویری بود و نوآور ی سوم ارائه یک معماری آبشاری به منظور ترکیب بهینه مجموعه ای از الگوریتم های طبقه بند آماری بود که به موجب آن قسمت های پس زمینه تصویر با الگوریتم های طبقه بندی سریع دور ریخته می شوند و الگوریتم های پیچیده تر طبقه بندی روی قسمت های مهم تر تصویر تمرکز می کنند . با وجود اینکه تا کنون الگوریتم های بسیار زیادی به منظور تشخیص چهره توسعه داده شده اند می توان گفت از این میان الگوریتم ویولا جونز بیشترین تاثیر را بر دهه های پیش روی خود داشت و قادر بود با سرعت بسیار زیاد دقت مطلوبی در مکان یابی چهره ارائه دهد.

۹-۳-۲- شرح مسئله

مسئله تشخیص چهره یکی از مسائل شناخته شده بینایی رایانه ای است. مسئله مورد نظر چهره انسان را در تصاویر دیجیتال مکانیابی می کند . فرآیند تشخیص چهره به دو سوال پاسخ می دهد : اول اینکه آیا چهره ای در یک تصویر و یا دنباله ای از تصاویر ویدئویی وجود دارد؟ دوم اینکه این چهره در کجای تصویر واقع شده است؟

اغلب انسان ها میتوانند این کار را به آسانی انجام دهند، اما یک کامپیوتر نیاز به دستورالعمل های دقیق و پیچیده ای دارد. پیچیدگی مسئله با تفاوت مقیاس تصویر، زاویه تصویربرداری، شرایط نوری متفاوت و پوشیده بودن قسمتی از چهره چند برابر می شود.

با وجود اینکه تا کنون الگوریتم های بسیار زیادی به منظور تشخیص چهره توسعه داده شده اند می توان گفت از این میان الگوریتم ویولا- جونز بیشترین تاثیر را بر دهه های پیش روی خود داشت و قادر بود و اولین الگوریتم بلادرنگ مکان یابی چهره با دقتی مطلوب به شمار می رود.

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.

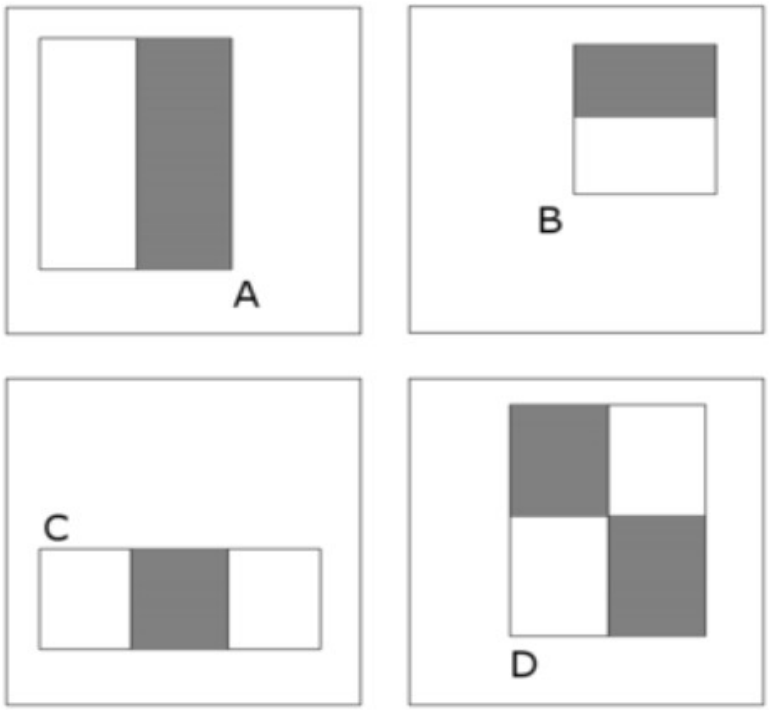
از نقاط ضعف این الگوریتم میتوان به حساسیت به شرایط نوری و اندازه چهره در تصویر اشاره کرد. این الگوریتم همچنین در تشخیص چهره های چرخیده دچار مشکل می شود و ممکن است یک چهره را به عنوان چند چهره در همپوشانی زیرپنجره های مورد جستجو پیدا کند. علیرغم همه این موارد و توسعه روش های جد ید تشخیص چهره با استفاده 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 وافقی و سایر ساختارهای ساده تصویری حساس هستند، با این وجود این حساسیت به هیچ عنوان انعطاف پذ یر نیست. بر خالف 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 و یژگیهای مستطیلی محصور شده در پنجره تشخیص

ارزیابی سریع آن ها با استفاده از تصویر انتگرالی این محدودیت انعطاف را به میزان قابل قبول ی 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 و یژگی 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 آبشاری برا ی پیداکردن سریع مناطق جالب توجه تصویر بهره می گیرد که امکان تشخیص چهره بلادرنگ 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 آبشاری

۹-۳-٤- ویژگی های شبه HAAR

ویژگی های شبه Haar عبارتند از نواحی روشن و تیره مستطیل شکلی که بر روی زیر پنجره ٢٤ در ٢٤ پیکسل مورد بررسی (نامزد احتمالی در برگیرنده 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 پیکسلهای داخل مستطیل های تیره کاسته می شوند . سایر نقاط عکس که خارج از این مستطیل ها قرار دارند در محاسبه ویژگی در نظر گرفته نمیشوند . به این ترتیب برای هر پنجره ٢٤ در ٢٤ 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 و یژگی های کمابیش مشترک چهره های انسانی است.

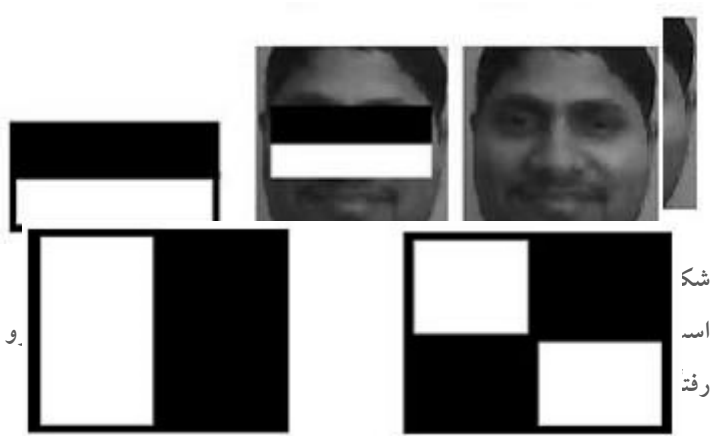
تمامی چهره های انسانی دارای ویژگی های کمابیش مشابهی هستند. این قواعد می تواند با استفاده از ویژگی های شبه Haar تطبیق داده شوند. به عنوان مثال چند ویژگی مشترک برای چهره های انسانی عبارتند از:

- 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 ویژگ ی ها بر سه دسته هستند (تصاویر زیر):



شکل ۹-۱٤ : 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 یکد یگر به صورت افقی یا عمودی هستند.
- ویژگی های سه - مستطیلی که مقاد یر دو مستطیل کناری را از مستطیل وسط کم می کند.
- ویژگی های چهار- مستطیلی که به صورت قطری در کنار یکد یگر قرار گرفته اند.

این سه نوع ویژگی در مکان های مختلف زیرپنجره ۲٤ در ۲٤ پیکسلی تعریف می شوند و تعداد آنها از ۱۸۰ هزار عدد بیشتر است ، بنابراین نه تنها نیاز به روش ی سریع برا ی 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.

۹-۳-۵- تصویر انتگرالی

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 سر یع 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 پیسکل 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 $s(x,-۱)=۰$ و $ii(-۱,y)=۰$ است.

$$s(x,y) = s(x,y-۱) + i(x,y)$$

$$ii(x,y) = ii(x-۱,y) + s(x,y)$$

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 درمجاورت یکد یگر قرار دارند، مثال برا ی محاسبه هر ویژگی دو – مستطیلی تنها به مقادیر شش پیکسل، برای محاسبه هر ویژگی سه –مستطیلی تنها به مقادیر هشت پیکسل و برا ی محاسبه هر ویژگی چهار– مستطیلی تنها به مقادیر و نه به پیکسل از تصویر انتگرالی نیاز است.

۹-۳-٦- 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 زیرپنجره ۲٤*۲٤ پیکسل استاندارد، مجموعا M ۱٦۲,۳۳٦ 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 بند ی کننده "ضعیف" ساده است:

$$h(x)=sgn(\Sigma\alpha_j\, h_j\,(x))$$

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

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

مقدار آستانه θ_j و قطبیت $s_j\epsilon\pm 1$ در آموزش و ضرایب αj 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 $j=1,...,M$

- 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.

$$\Theta_j, s_j = \arg\min \sum \omega^i_j \varepsilon^i_j \text{ where } \varepsilon^i_j = \begin{cases} 0, & if\ y^i = h_j(x^i, \Theta_j, s_j) \\ 1, & otjer \end{cases}$$

- 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.

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

۹-۳-۷- نمونه کد نویسی تشخیص چهره

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

```
function FaceDetect()
%#codegen
%Create raspi & webcam obj
raspiObj = raspi('192.168.233.85','pi','pi');
cam = cameraboard(raspiObj,'Resolution','320x240','FrameRate',30);
while true
    img = snapshot(cam);
    faceDetector = vision.CascadeObjectDetector;
    bbox = step(faceDetector, img);
    Ifaces = insertObjectAnnotation(img,'rectangle',bbox,'Face');
    imagesc(Ifaces), title('Detected Faces'),
    drawnow;
end
end
```

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