

دانشکده شهید شمس‌پور

گزارشات مرتبط با درس مباحث ویژه

پیش‌نویس جزئیات پروژه درس مباحث ویژه - راه‌اندازی خانه هوشمند

محمدیاسین داوده و مهدی صفریان

استاد ساسان برهلیا

۲۵ دی ۱۳۹۹

فهرست مطالب

| | |
|---|----------------------------|
| ۲ | ۱ پیش نیازها |
| ۳ | ۲ راه اندازی اولیه برد پای |
| ۳ | ۱.۲ مختصری بر SSH |
| ۴ | ۲.۲ راه اندازی Tor |
| ۵ | ۳ کار با ماژول‌ها |
| ۵ | ۱.۳ اصطلاحات |
| ۵ | ۲.۳ برقراری اتصال |
| ۵ | ۳.۳ برنامه نویسی |
| ۶ | ۴.۳ معرفی فایل‌ها |
| ۶ | ۱.۴.۳ settings.yaml |
| ۸ | ۲.۴.۳ اضافه کردن زبان جدید |

۱ پیش نیازها

برای هر پروژه IoT که در آن وضعیت سنسورها نیاز به تحلیل یا منطق داشته باشد نیاز به پردازنده‌ای قابل برنامه نویسی وجود دارد. برای پشتیبانی از چنین سنسورهایی این پردازنده‌ها باید مجهز به پین‌های ورودی و خروجی عمومی یا GPIO^۱ باشند تا بتوانند سیگنال‌های سنسورها را دریافت، اکتویتورها را فعال و انرژی قطعات را تأمین کنند. میکروکنترلرها پردازنده‌های کوچکی هستند که برای چنین پروژه‌هایی ساخته شده‌اند. از این جهت می‌توان از چنین میکروکنترلرهایی مانند آردوینو برای این پروژه استفاده کرد. ما در این پروژه از برد Raspberry Pi 3B استفاده می‌کنیم که یک برد با پردازنده آرم و GPIO است. برای اتصال سنسورها به GPIO احتیاج است. سنسورها معمولاً ورودی‌های 3.3V و 5.0V استفاده می‌کنند. در نتیجه مهم است که GPIO بتواند 5.0V را پشتیبانی کند در غیر این‌ها به یک PSU^۲ جداگانه وجود دارد. برای این پروژه ما سه ماژول گلدان هوشمند، تشخیص گاز و الکتریز هوشمند نیازمند سنسورهای متفاوتی هستیم. برای راحتی کار از ماژول‌های این سنسورها استفاده می‌کنیم که آماده به کار هستند روی یک برد هستند. یک سنسور رطوبت خاک^۳، سنسور مجاورت مادون قرمز^۴ و سنسور گاز^۵ مورد استفاده است. برای انجام فعالیت‌ها به دو عدد رله، یک موتور چرخشی و یک پمپ آب نیاز است. به طور کل لیست وسایل مورد استفاده در این پروژه به شرح زیر است:

• سنسورها

- ماژول رطوبت خاک
- ماژول مجاورت مادون قرمز
- ماژول تشخیص گاز

• عملگرها

- موتور چرخش
- پمپ آب

• سایر

- Raspberry Pi 3B
- سیم
- رله (دو عدد یا یک رله دو کاناله)

^۱General Input Output Pins
^۲Power Supply Unit
^۳YL-69
^۴IRFC51
^۵MQ-5

- بردبرد (Breadboard) (اختیاری)

- بردبرد PSU (اختیاری)

- سیستمی ثانوی (اختیاری در صورت استفاده از Raspberry)

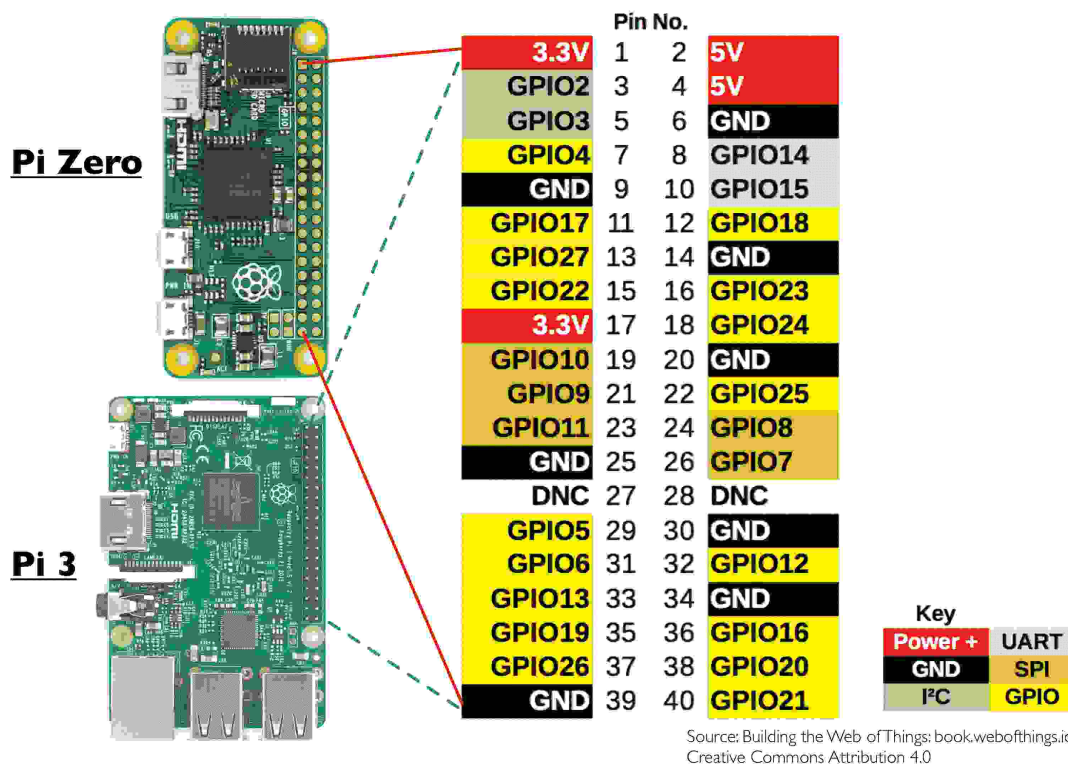
در این برگه فرض بر این است که کاربر از برد پای، یک سیستم ثانوی و سیستم عامل های Raspbian/RaspberryOS و لینوکسی دیگر استفاده می کنند که همه بر پایه SystemD است.

۲ راه اندازی اولیه برد پای

پیش از استفاده از برد پای روی آن سیستم عامل لینوکس را ریخته و نصب می کنیم. پس از این، برای برنامه نویسی در پایتون نیاز به کتابخانه RPi.GPIO برای مدیریت GPIO داریم که به طور پیش فرض روی توزیع Raspberry Pi OS و Raspbian موجود است.

برای کار با برد پای از طریق سیستمی ثانوی می توان با SSH یا کابل TTL که به پین های GND، ۸ (کابل سفید) و ۱۰ (کابل سبز) ارتباط برقرار می کند متصل شد. پیش از برقراری این ارتباطات باید با raspberry-config قابلیت SSH یا/و سریال را فعال کرد (بیشتر در ۱.۲).

پس از این با ادیتور مورد علاقه با پشتیبانی از SSH (مثلاً Emacs Tramp) یا ابزاری مناسب کار با TTL مانند tio به شکل `/dev/ttyUSB` به کدزدن پرداخت.



شکل ۱: پین های GPIO برد پای

۱.۲ مختصری بر SSH

برای ارتباط آسان و ویرایش فایل ها با دستگاهی ثانوی - به جز برد پای - نیازمند ارتباطی هستیم. یکی از بهترین پروتکل های ارتباطی برای این کار SSH است که به کاربر یک شل کامل نه تنها فقط برای ارسال اطلاعات متنی - به عکس TTL - بلکه می توان فایل و سایر داده های پیچیده تر را روی آن انتقال داد.

در صورت غفلت در تنظیم صحیح، SSH می‌تواند پروتکلی خطرناک باشد. امنیت SSH را تأمین کنید. ارتباطات را به شبکه محلی محدود کنید و حتی امکان پس از انجام این پروژه قابلیت‌های SSH را غیرفعال کنید.

برای راه‌اندازی یک سرور ساده SSH کافی است که دیمن آنرا روی سرور نصب کرده و آنرا اجرا کنید. برای این کار ابتدا با کابل TTL با سرور ارتباط برقرار می‌کنیم. روی آن OpenSSH را نصب کرده (از پیش با Raspbian/RaspberryOS همراه است) و آنرا اجرا می‌کنیم (برای سیستم‌دی: `systemctl enable sshd` و `systemctl start sshd`). پس از راه‌اندازی و بررسی SSH می‌توان به آن متصل شد. برای اتصال راحت‌تر بهتر است به سرور خود آی‌پی استاتیک از DHCP داده یا از سمت خود سرور درخواست آی‌پی ترجیحی کنید. اگر سرور با موفقیت راه‌اندازی شده باشد می‌توانید با `ssh $username@$ip -p $port` به آن متصل شد. پورت پیش‌فرض SSH پورت ۲۲ است. پس از این باید با پرامپت رمز خود مواجه شوید. برای اتصال ساده‌تر و امکان مدیریت راحت می‌توانید به فایل کانفیگ خود (پیش‌فرض `~/.ssh/config`) خطوط زیر را به این صورت اضافه کنید (`$alias` نام مستعار سرور باشد):

```
1 cat >> ~/.ssh/config <<EOF
2 Host $alias
3     HostName $ip
4     User $user
5     Port $port
6 EOF
```

پس از این می‌توانید با `ssh $alias` به سرور خود متصل شوید.

• اگر این نام مستعار را در کانفیگ فایل خود وارد نکرده‌اید از اینجا به بعد به جای `$alias` از همان قالب `username@ip:port` استفاده کنید.

در نهایت برای گذر از مشکل درخواست رمز می‌توانید از کلید هویت استفاده کنید. برای این کاری کافیت با `ssh-keygen -t rsa` یک جفت کلید عمومی با نام دلخواه بسازید و با `ssh-copy-ip $alias` کلید عمومی خود را به لیست کلیدهای مجاز (پیش‌فرض `~/.ssh/authorized_keys`) بیافزایید. پس از این اگر `ssh-agent` `eval $(ssh-agent)` کرده و کلید خصوصی را به هویت خود با `ssh-add ~/.ssh/key` بیافزاید، سرور از شما رمز عبوری نمی‌خواهد. برای اضافه کردن کلید به طور اتوماتیک می‌توانید از پارامتر `IdentityFile` در فایل کانفیگ استفاده کنید. به این صورت کانفیگ نهایی شما شبیه فایل زیر خواهد بود:

کد ۱: کانفیگ نمونه نام مستعار در سیستم ثانوی

```
1 Host rpi
2     HostName 192.168.1.111
3     User pi
4     IdentityFile ~/.ssh/id_rpi
```

۲.۲ راه‌اندازی Tor

برای ارتباط با ثبات‌تر یا از روترهایی که دسترسی به شبکه تلگرام ندارند می‌توانید به سادگی تور را نصب کرده (`apt install tor`) و مشابه SSH سرویس آنرا اجرا (`systemctl start tor`) و فعال (`systemctl enable tor`) کنید. در صورتی که با خطا مواجه شد باید آنرا (یا پلی برای ارتباط) تنظیم کنید. اگر تور فعال باشد به طور پیش‌فرض روتر/پروکسی روی پورت ۹۰۵۰ (مرورگر تور ۹۱۵۰) خواهید داشت که به اتصال ما کمک می‌کند.

۶ علامت متغیر در شل است. به طور مثال جای `$ip` آی‌پی آدرس خود را بنویسید یا با `ip="192.168.1.111"` آنرا تنظیم کنید.

ممکن است استفاده از تور یا پروکسی توسط مدیر شبکه، ISP یا دولت قانونی تلقی نشود. پیش از راه‌اندازی بررسی کنید.

۳ کار با ماژول‌ها

۱.۳ اصطلاحات

ماژول‌های پین‌هایی دارند که با عنوانی مانند VCC، AO و... نشانه‌گذاری شده‌اند. پین VCC پین برق است. ابتدا باید مشخص کرد که برق مورد نیاز هر ماژول 3.3V یا 5.0V است. ولتاژ کم رساندن به ماژول خطر ساز نیست از بیش از ولتاژ مورد قبول به آن تحمیل کردن ممکن است باعث خرابی شود. ماژول‌هایی که اینجا استفاده می‌کنیم همه از 5.0V استفاده می‌کنند. پین GND پین منفی، پین DO پین خروجی دیجیتالی (دودویی) است و در نهایت پین AO پین خروجی آنالوگ است. معمولاً مقاومتی به نام پتانسیومتر (شکل ۲) روی این ماژول‌ها وجود دارد که آستانه یک (یا صفر) شدن DO را مشخص می‌کند.



شکل ۲: پتانسیومترهای مختلف

۲.۳ برقراری اتصال

برای کار با هر ماژولی باید برق آنرا تأمین کرده و سپس پین DO آنرا به یک GPIO (زرد روی ۱) وصل می‌کنیم. بعداً برای دسترسی به آن می‌توان از طریق شماره‌های فیزیکی پایه (۱-۴۰) یا شماره GPIO به آن دسترسی پیدا کرد. در نهایت GND را متصل می‌کنیم.

هرگز رله یک کاناله را به 5.0V متصل نکنید.

۳.۳ برنامه‌نویسی

پس از اتصال ماژول‌ها به برد می‌توان آنها را برنامه‌نویسی کرد.

یکی از آسان‌ترین زبان‌ها برای شروع Hy یا Python می‌باشد. در این پروژه از های استفاده می‌کنیم که لیستی از پایتون است. اگر لیست، پایتون یا های را بلدید می‌توانید هر کدام از دیگر سه را در چند ساعت بیاموزید. بعلاوه کدهای پایتون و های با hy2py و py2hy به یکدیگر تبدیل می‌شوند. برای جزئیات بیشتر درباره های به سایت آن <https://hylang.org> و مستندات آن^۷ مراجعه کنید. علاوه بر این نیازمند درک پایه‌ای از YAML دارید.

۴.۳ معرفی فایل‌ها

- **settings-example.yaml** همانطور که از نام آن مشخص است این فایل صرفاً یک مثال است و باید یک فایل با نام **settings.yaml** بسازید و اطلاعات درون مثال را به فایل جدید منتقل کنید و با توجه به شرایط خودتان آن را تکمیل کنید.

تمامی تنظیمات مربوط به پروژه اعم از افرادی که می‌توانند به بات تلگرامی دستور دهند، توکن (token) بات، حسگرها و ماژول‌هایی که قرار است به برد متصل شوند و دستوراتی که بات باید نسبت به آن‌ها واکنشی نشان دهد در این فایل نگه‌داری می‌شوند. (برای اطلاعات درباره نحوه تکمیل فایل تنظیمات مراجعه شود به بخش ۱)

- **settings.hy**

عناصر و تنظیماتی که در فایل **settings.yaml** نوشته می‌شوند توسط این اسکریپت فراخوانی می‌شوند. اگر تنظیمات جدیدی اعمال شود این اسکریپت به راحتی تغییرات جدید را به برنامه می‌دهد و از آن به بعد بات تلگرامی با تنظیمات اعمال شده جدید کار می‌کند. هدف از ساخت این فایل ایجاد متغیرهایی پویا است که بتوان از طریق دیگر اسکریپت‌های hy از این متغیرها استفاده کرد و از فراخوانی تنظیمات در هر اسکریپت به صورت جداگانه جلوگیری شود.

- **board.hy**

کنترل رله، دریافت و اجرای دستورات از طریق بات تلگرامی و تعیین زمان‌بندی سیستمی یا همان کرون جاب توسط این اسکریپت مدیریت می‌شوند که زیر یک کلاس به نام **Module** قرار دارند. کلاسی دیگری به نام **GPIO** وجود دارد که مدیریت ماژول‌های متصل به برد که در فایل تنظیمات اضافه شده‌اند را برعهده دارد و آن را به لیست ماژول‌های موجود اضافه می‌کند.

- **bot.hy**

تمامی توابع عملگر بات تلگرامی در این اسکریپت وجود دارد. ابتدا با یک کلاس به نام **brain** مواجه هستیم. توابع که در این کلاس استفاده شدن صرفاً برای تعامل ربات با کاربر در تلگرام است تا بتواند لیست ماژول‌های موجود یا وضعیت آن را نمایش دهد.

۱.۴.۳ settings.yaml

۱. دسترسی‌ها

برای تعیین دسترسی افرادی که می‌توانند به بات تلگرام فرمانی دهند باید نام کاربری تلگرام افراد را در بخش **admins** تعریف کنید.

```
admins:
- "@person1"
- "@person2"
- "@person3"
```

۲. اتصال بات تلگرام

برای اتصال برنامه به یک بات تلگرامی ابتدا باید با استفاده از **@botfather** یک بات تلگرامی بسازید و سپس آدرس توکن (token) بات را در قسمت **token** بنویسید.

^۷<https://docs.hylang.org/en/stable/language/api.html>

token: 1234567890123456789012345678901234567890

۳. مشخص کردن نوع برد

نوع بردها یا BOARD هستند یا BCM که استاندارد خواندن پین‌های موجود بر روی بردها است. در شکل ۱

gpio: BOARD # or BCM

۴. انتخاب زبان بات

متغیر locale زبان بات را مشخص می‌کند که مثلاً به چه زبانی پاسخ شما را بدهد.

پیش فرض فایل تنظیمات فارسی (fa-IR) است.

locale: "fa-IR" # or "en-GB"

۵. کلید پروکسی

برای راه اندازی پروکسی که قبلاً گفته شده بود به شکل زیر از آن استفاده می‌کنیم.

یادآوری: تنها از پروکسی نوع socks5 پشتیبانی می‌شود.

proxy: socks5://[ip]:[port]

۶. تعریف سنسورها

برای تعریف سنسورها باید در بخش modules به ازای هر پینی که فعال است حسگر خود را تعریف کنیم. مدیریت سنسورهای می‌تواند به صورت دستی یا حتی خودکار باشد. برای تعریف حالت خودکار سنسورها در بخش ۷ درباره آن توضیح خواهیم داد. در حالت دستی شما قادر هستید در هر لحظه به بات فرمان دهید که چه حسگری را فعال یا غیر فعال کند.

برای تعریف یک حسگر به شکل زیر عمل کنید:

modules:

- **pin:** 11 # pin number

io: out # اگر دستگاهی مانند رله باشد و ورودی بگیرد مقدار in را قرار می‌دهیم

اگر دستگاه تنها یک حسگر باشد که فقط خروجی تولید می‌کند out را قرار می‌دهیم

اگر در بخش profiles حالت خودکاری را ثبت کنید نام آن را می‌نویسید. **auto:** pot #

مشخص می‌کند حسگر دستی (man) است یا خودکار (auto) **profile:** man #

اسامی کلیدی که کمک می‌کند تا بات آن حسگر را بشاند **aliases:** #

نکته مهم: اولین نام، نام اصلی است که در کیبورد شیشه‌ای تلگرام نمایش داده می‌شود. **en-GB:** #

- pot

- plant

- goldan

همین طور هم برای دیگر زبان‌ها

۷. حالت خودکار سنسورها

پروفایل‌ها رفتارهایی هستند که انتظار داریم یک حسگر در حالت خودکار از خود نشان دهد، برای مثال می‌خواهیم حسگر رطوبت خاک هر ۵ ساعت چک شود و اگر رطوبت خاک به اندازه کافی نبود رله پمپ آب را روشن کند و گلدان را آبیاری کند.

در بخش پروفایل در حال حاضر مقادیر duration که مدت زمان فعال ماندن را مشخص می‌کند، cron که الگوی کرون جاب سیستمی را می‌گیرد، gate شماره پین گیت است؛ وجود گیت به ما کمک می‌کند تا اگر در کرون لحظه فعال سازی یک راه‌انداز (رله) فرا رسیده بود؛ گیت وضعیت حسگرها را بررسی می‌کند که آیا لازم است اجازه فعال‌سازی راه‌انداز داده شود یا نه، command شامل دستوراتی است که به بات داده می‌شود و پیامی را برمی‌گرداند (فایل strings.yaml در پروژه را مطالعه کنید، کلید و دستورات مورد قبول کلاس brain در آنجا تعریف شده است).

```
۱ profiles:
۲   pot:
۳     duration: 5 # per second
۴     cron: '* * * * *' # check every minute
۵     gate: 38 # pin number 38
۶     command: select %s; on; # read strings.yaml in command section
```

۲.۴.۳ اضافه کردن زبان جدید

برای اضافه کردن زبان جدید ابتدا رشته‌های مرتبط با نام هر ماژول را برای صدا کردن آن بروزرسانی کنید. پس از این با بروزرسانی فایل تنظیمات و تغییر locale به آن زبان می‌توانید از باتی با آن زبان استفاده کنید. رشته خاص obj_pos در فایل strings.yaml مشخص‌کننده مکان مفعول هر جمله در هر زبانی است. به طور مثال وقتی برای فارسی این رشته عدد صفر را دارد یعنی بات باید اولین کلمه هر جمله دستوری را به عنوان شی/ماژولی که باید روی آن دستور را اجرا کنید انتخاب می‌کند.