

Если вы видите что-то необычное, просто сообщите мне.

Go — Raspberry Pi GPIO

"Привет мир" руководство

Для относительно нового языка, к которым относится Go. Я использовать его с тех годов, по нескольким причинам, так как он полон сюрпризов.

Поэтому недавно, я купил новую pi3 и в основном использовал C для прошлой pi2. Однако, отметим, множество руководств использует питон, а я не большой фанат этого языка. Поэтому я случайным образом поискал библиотеке Go и различные руководства для Pi GPIO, и поигрался чуток с проводульками у меня дома.

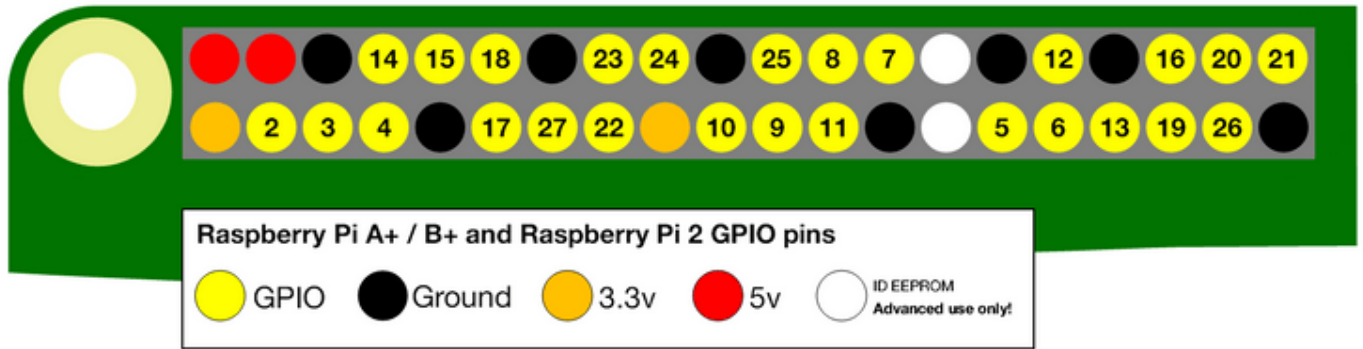
В этой статье мы собираемся построить очень простую LED переключалку используя Raspberry Pi и язык Golang.

Мы можем установить целое Go окружение внутри Raspberry Pi, однако, это займет много ресурсов внутри маленькой машинки. Поэтому, я предпочитаю разрабатывать и строить программу удаленно с моего рабочего места.

1. Используем GPIO

Мы собираемся создать очень простую мерцающую программу, которую установим и запустим на Raspberry Pi. Поэтому, перед этим, нужно понять базу, как работать с GPIO.

Самый простой способ считать GPIO интерфейсом, который лежит между системой Pi и внешней системой. IO модуль будет использовать GPIO пины, таким образом, чтобы raspberry могла читать ввод или показывать вывод от девайса или сенсора.



Как показано на рисунке, устройство имеет несколько типов пинов, такие как GPIO, земля, 3.3 вольтовые и 5 вольтовые пины. Для нашей инструкции мы собираемся играть с GPIO 18 пином и землей.

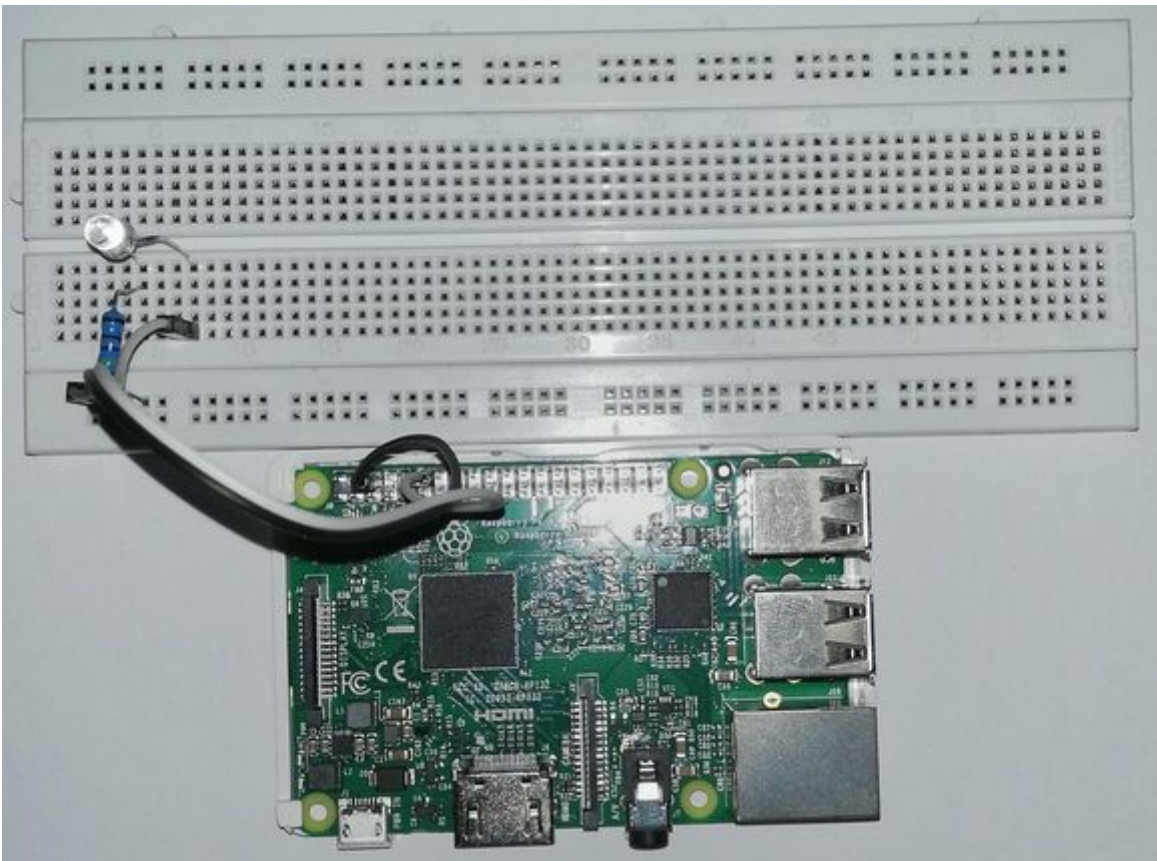
Мы так же можем получить доступ к GPIO информации через термина используя команду `pinout`.

2. Собираем LED устройство

Для этой инструкции, мы собираемся подготовить несколько вещей:

1. USB кабель и зарядник для питания Raspberry Pi,
2. 2 коротких переключателя проводов,
3. Макет,
4. LED, and
5. 330Ω резистор

Я не буду рассказывать глубокого процесс установки, так как это легко можно найти в интернете.



3. Разработка Go кода

Наконец, мы собираемся разработать код используя Go. Перво-наперво, мы будем использовать доступный такой пакет в интернете, как:

<https://github.com/stianeikeland/go-rpio>, and <http://gobot.io/documentation/platforms/raspi/>

Go-rpio поддерживает операции для GPIO используя Golang, поэтому я верю, что он достаточно подходит для наших целей. В то время как Gobot предоставляет целую SDK библиотеку для доступа к GPIO пинам и специфические драйвера как: LED, мотор, реле, сервопривод и т.д.

В этом случае, я собираюсь использовать Go-rpio для базового LED примера, так как мы не собираемся использовать эти драйвера на данный момент.

Как я упоминал в самом начале, мы собираемся разрабатывать удаленно. Поэтому сперва, скачаем пакет который будем использовать.

```
go get github.com/stianeikeland/go-rpio
```

Это пример кода для создания мерцающего приложения используя Go на raspberry pi:

```
package main

import (
    "fmt"
    "time"

    "github.com/stianeikeland/go-rpio"
)

func main() {
    fmt.Println("opening gpio")
    err := rpio.Open()
    if err != nil {
        panic(fmt.Sprintf("unable to open gpio", err.Error()))
    }

    defer rpio.Close()

    pin := rpio.Pin(18)
    pin.Output()

    for x := 0; x < 20; x++ {
        pin.Toggle()
        time.Sleep(time.Second / 5)
    }
}
```

Объяснение: Этот код открывает GPIO и устанавливает режим приема на пине 18. Пин будет настроен для вывода(Ввода для чтения, типа сенсора и т.д.). 20 переключит режим пина(Low > High > Low) и установит задержку 200ms. Как только операция выполнится, программа будет завершена.

4. Сборка и загрузка программы.

Для того, чтобы безопасно собрать программу мы будем использовать команду:

```
env GOOS=linux GOARCH=arm GOARM=5 go build
```

Команда выше говорит компилятору, что мы собираем программу для Linux и ARM архитектуру. На данный момент, есть поддержка нескольких версий ARM компилятором Go, но мы ищем версию 5. Затем мы собираемся загрузить \ скопировать файл на подключенную raspberry pi. Для этой цели, мы собираемся использовать scp команду.

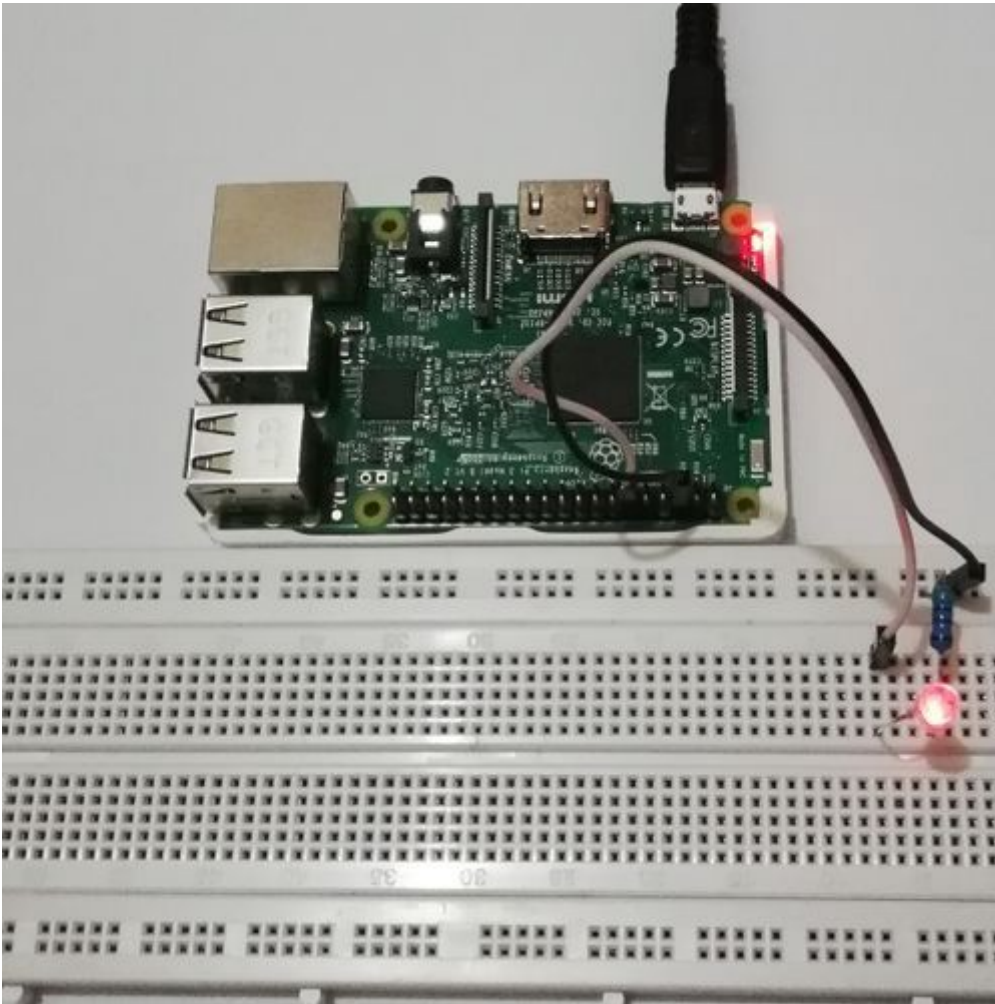
```
scp [go binary] pi@192.168.43.208:[remote dir]
```

5. Запуск программы

Как только бинарник был загружен(для примера, мы загрузили его по пути /home/pi/go/gorio), то чтобы его запустить нужно выполнить команду:

```
./gorio
```

Если программа верно собрана, и нет ошибок в коде, LED будет мерцать и остановится после 20 переключений.



6. Выводы

Можно использовать Raspberry-Pi для разработки программ внутри Raspberry-Pi. В этой статье, я показал вам простой пример, как создать мерцающий LED используя Raspberry Pi, на языке Golang.

7. Image Link and Resources

- [turning-on-an-led-with-your-raspberry](#)
 - [cross-compiling-golang-applications-raspberry-pi](#)
 - [go-rpio](#)
 - [gobot](#)
 - [Raspberry Pi icon in Color Style](#)
-

Revision #2

Created 2021-10-22 10:22:02 UTC by gasick

Updated 2023-04-16 19:30:04 UTC by gasick