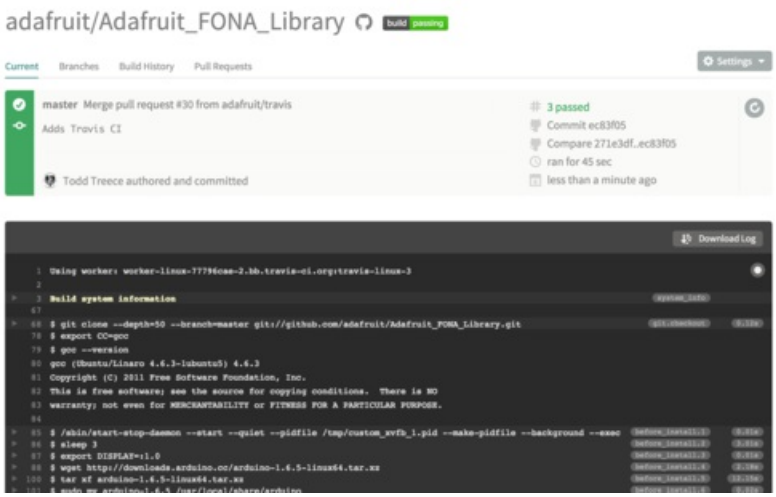
adafruit learning system

Continuous Integration, Arduino, and You

Created by Todd Treece

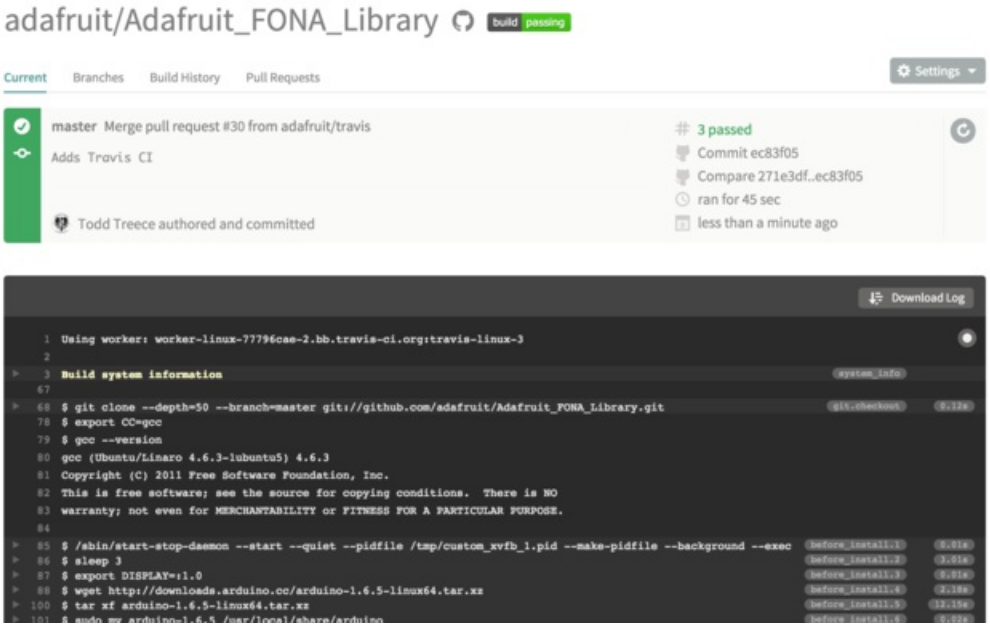


Last updated on 2018-08-22 03:48:59 PM UTC

Guide Contents

Guide Contents 2
Overview 3
Travis CI Configuration 4
Enable Travis CI for a Repository 4
Add the Travis CI Configuration File 4
Build Status Icon 6

Overview



Bugs can be introduced into software very easily, and it takes quite a bit of effort to ensure that the 100+ Adafruit Arduino Libraries compile cleanly. Luckily, there is an easy way to verify that your Arduino Libraries compile after every push to GitHub by using Travis CI (https://adafru.it/fuN).

What is Travis CI? Travis CI is an open source continuous integration (https://adafru.it/fuO) service that automatically tests and builds your GitHub projects. The service is free for open source projects, and the setup is relatively painless.

Travis CI supports a number of languages and platforms, but there currently isn't native support for building Arduino projects. Luckily it's easy to customize builds using a configuration file, so building Arduino projects only requires a few simple modifications to the config. Let's get started.

Travis CI Configuration

The first step you will need to complete is to sign into Travis CI (https://adafru.it/fuN) using your GitHub account. This process requires you to authorize access to your GitHub account so Travis can get a list of your repositories. The Travis CI documentation has a great getting started page (https://adafru.it/fuP) that will help you with this process if you require more assistance.

Enable Travis CI for a Repository

You will need to visit your Travis CI profile (https://adafru.it/fuQ), and enable the service for the repository you wish to build. We will be using the Adafruit FONA Library (https://adafru.it/dDC) in this example, but this process should work for verifying Arduino sketches in any GitHub repository.

Add the Travis CI Configuration File

Now that we have enabled the service, we will need to add a **.travis.yml** to the root of the repository. Make sure you don't miss the first period at the beginning of the **.travis.yml** filename.

Let's take a quick look at the full config file for the FONA library.

```
language: c
before_install:
  - "/sbin/start-stop-daemon --start --quiet --pidfile /tmp/custom_xvfb_1.pid --make-pidfile --background
  - sleep 3
  - export DISPLAY=:1.0
  - wget http://downloads.arduino.cc/arduino-1.6.5-linux64.tar.xz
  - tar xf arduino-1.6.5-linux64.tar.xz
  - sudo mv arduino-1.6.5 /usr/local/share/arduino
  - sudo ln -s /usr/local/share/arduino/arduino /usr/local/bin/arduino
install:
  - ln -s $PWD /usr/local/share/arduino/libraries/Adafruit_FONA
  - arduino --install-library "Adafruit SleepyDog Library,Adafruit MQTT Library"
script:
  - arduino --verify --board arduino:avr:uno $PWD/examples/FONAtest/FONAtest.ino
  - arduino --verify --board arduino:avr:uno $PWD/examples/IncomingCall/IncomingCall.ino
  - arduino --verify --board arduino:avr:uno $PWD/examples/AdafruitIO_GPS/AdafruitIO_GPS.ino
notifications:
  email:
    on_success: change
    on_failure: change
```

The **language** section tells Travis that the project is primarily written in **C**. Since we are building using the Arduino IDE, this setting is not critical. The **before_install** section configures the build environment to run with a virtual display using **xvfb,** and also installs the Arduino IDE. The Arduino IDE CLI currently requires a graphical user interface to be present (https://adafru.it/fuR), so this configuration step is necessary for the Arduino IDE to run in the Travis environment. If you would like to read more about **xvfb** configuration for Travis CI, there is a handy page in the Travis CI documentation (https://adafru.it/fuS) with configuration examples.

The **language** and **before_install** sections should not need to be modified for your Arduino project, so you should be able to copy and paste the text into your **.travis.yml** config file.

```
language: c
before_install:
  - "/sbin/start-stop-daemon --start --quiet --pidfile /tmp/custom_xvfb_1.pid --make-pidfile --background
  - sleep 3
  - export DISPLAY=:1.0
  - wget http://downloads.arduino.cc/arduino-1.6.5-linux64.tar.xz
  - tar xf arduino-1.6.5-linux64.tar.xz
  - sudo mv arduino-1.6.5 /usr/local/share/arduino
  - sudo ln -s /usr/local/share/arduino/arduino /usr/local/bin/arduino
```

The **install** section symlinks the FONA library to the Arduino library path so the IDE has access to it when building examples. It also uses the Arduino CLI to install other library dependencies needed when running the included example sketches. If you would like to read more about the Arduino CLI **--install-library** command, you can refer to the CLI documentation on GitHub (https://adafru.it/fuR).

You will need to modify these commands to point to your library, and to install any other library dependecies needed to compile your example sketches. If you do not need to install any dependencies, you can omit that line from your config.

```
install:
  - ln -s $PWD /usr/local/share/arduino/libraries/Adafruit_FONA
  - arduino --install-library "Adafruit SleepyDog Library,Adafruit MQTT Library"
```

The **script** section is where the library is built and checked by Travis. We use the Arduino sketches in the **examples** directory to confirm that the FONA library compiles successfully on an Arduino Uno. This is a very simple sanity check to confirm that no regressions have been added by a commit or pull request. *This doesn't check if the code functions as expected, it merely confirms that the code builds successfully.*

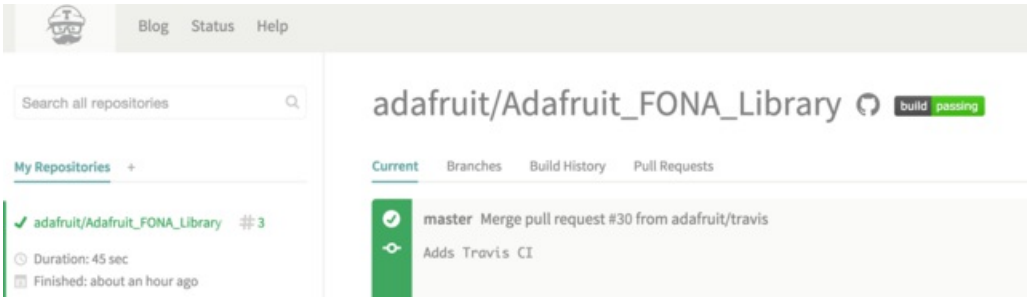
You will need to modify the **script** section to point to your example sketches. Verifying the examples in a library is the easiest way to check to make sure a library builds properly, but you can use this section to test any Arduino sketch.

```
script:
  - arduino --verify --board arduino:avr:uno $PWD/examples/FONAtest/FONAtest.ino
  - arduino --verify --board arduino:avr:uno $PWD/examples/IncomingCall/IncomingCall.ino
  - arduino --verify --board arduino:avr:uno $PWD/examples/AdafruitIO_GPS/AdafruitIO_GPS.ino
```

The **notifications** section of the configuration file tells Travis to only email the author and commiter if the build status changes. The Travis CI documentation (https://adafru.it/fuT) has more details about the notification options available.

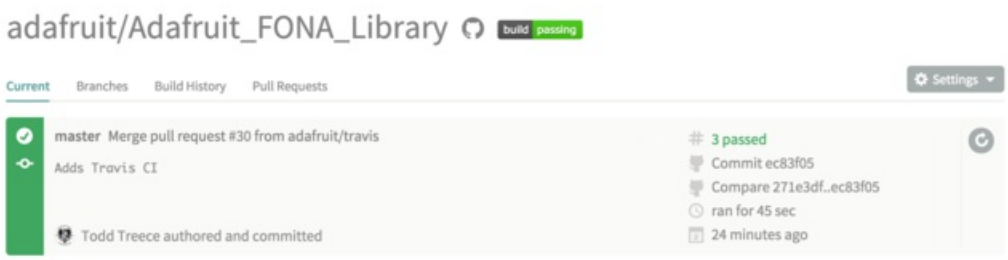
```
notifications:
  email:
    on_success: change
    on_failure: change
```

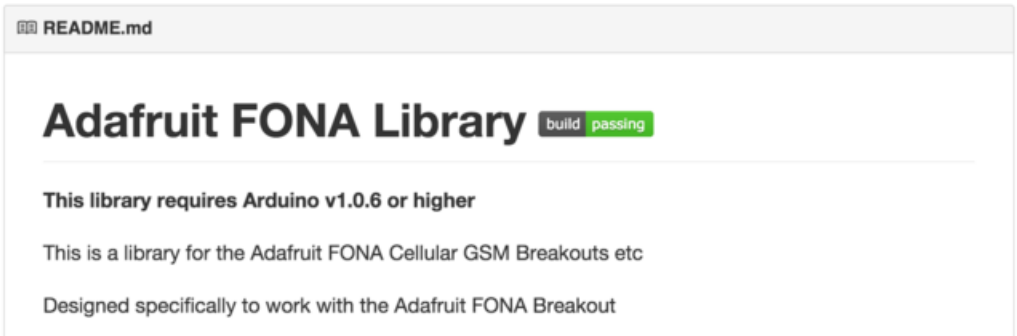
Once you are finished making changes, commit the file to your repo, and push it to GitHub. If everything is working, you will then see your build start on travis-ci.org (https://adafru.it/fuN).



Build Status Icon

A quick way to see the current build state of your library is to add the Travis CI build icon for your project to the README. You can do this by clicking on the build status icon next to your project's name on travis-ci.org (https://adafru.it/fuN).



You will be presented with a modal window that has a few format options for the icon. Select markdown from the dropdown, copy & paste the markdown text into your project's README, and push it up to GitHub.

You will then be able to easily check the build status of your project without logging into Travis CI.