Platformio has official documentation on tdd / unit testing also . PlatformIO has its own unit testing framework

<http://docs.platformio.org/en/stable/plus/unit-testing.html>

PlatformIO offers 2 types of test

1. Desktop
2. Embedded

We are going to do the embedded type test “ PlatformIO wraps test and main firmware (from [src\_dir](http://docs.platformio.org/en/stable/projectconf.html#projectconf-pio-src-dir)) with own PlatformIO Unit Testing Framework, builds special firmware for a target device and upload it. After uploading, PlatformIO connects to embedded device (board) using [test\_port](http://docs.platformio.org/en/stable/projectconf.html#projectconf-test-port) , starts test, collects results and shows test results on the host machine.”

I guess test port is Serial for uno as platformIO said “Please note that PlatformIO Unit Testing Framework uses Serial/UART as communication interface between PlatformIO Unit Test Engine and target device. If you use Serial in your project, please wrap/hide Serial-based blocks with #ifndef UNIT\_TEST macro.”

1. At first write a test program like I did in “tddOnUno” [see better arduino ide doc ]
2. Then create a “test” folder under the project folder
3. Then wrap your main.cpp code by

#ifndef UNIT\_TEST

#endif

This means if unit testing is ongoing then this part wont compile

1. The create test\_main.cpp under test folder
2. Now the test\_main.cpp should have the unity.h header file and the setup() and loop() function like the main.cpp .

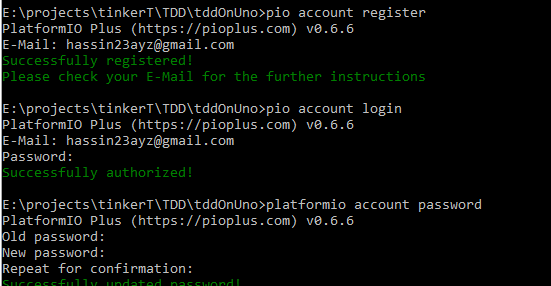
There should be UNITY\_BEGIN() and UNITY\_END()

Use RUN\_TEST(functionName) to test the function which name is in the ()

You can also do other type of tests also see this <https://github.com/ThrowTheSwitch/Unity#unity-test-api>

Make sure you have a platformio account and you are registered to do the following:

* At first create an platformio account using github
* The open command prompt
* then register from the command line
* then login with the mail and emailed password
* then change password
* see the following snap for the steps



Now go back to coding in eclipse . you will make a corresponding test\_main.cpp by which you will test things of main.cpp

Please see the following comparison table:

|  |  |
| --- | --- |
| **#include** <Arduino.h>  #ifndef UNIT\_TEST  **#define** BLINKPIN 13  **void** **doBlink**(**int** i)  {  **if**(i == 1)  {  **digitalWrite**(BLINKPIN, HIGH);  **delay**(1000);  **digitalWrite**(BLINKPIN, LOW);  **delay**(1000);  }  }  **void** **setup**()  {  **pinMode**(BLINKPIN,OUTPUT);  }  **void** **loop**()  {  doBlink(1);  }  #endif | **#include** <Arduino.h>  **#include** <unity.h>  **#ifdef UNIT\_TEST**  **void** **test\_led\_builtin\_pin\_number**(**void**) {  TEST\_ASSERT\_EQUAL(LED\_BUILTIN, 13);  }  **void** **setup**() {  UNITY\_BEGIN(); *// IMPORTANT LINE!*  RUN\_TEST(test\_led\_builtin\_pin\_number); //unit testing the Pin number  pinMode(LED\_BUILTIN, OUTPUT);  UNITY\_END();  }  **void** **loop**()  {}  **#endif** |
| **Main.cpp** | **Test\_main.cpp** |

* After writing the code run platformio:test command
* Then from the project folder open command line and run

platformio test - e uno --verbose

* or click platformio:test in eclipse
* you will see test passed
* but if you modify BLINKPIN value you will see error