Unit Test All The Things

22 September 2017

Charles Haynes

Unit tests are

Small
Fast
Specific
Independent

A test is not a unit test if:

- It talks to the database
- It communicates across the network
- It touches the file system
- It can't run at the same time as any of your other unit tests
- You have to do special things to your environment (such as editing config files) to run it.

Tests that do these things aren't bad. Often they are worth writing, and they can be written in a unit test harness. However, it is important to be able to separate them from true unit tests so that we can keep a set of tests that we can run fast whenever we make our changes.

-- Michael Feathers

So what?

Debugging is really hard
Unit tests are easy
Making code testable improves it

How can you test things?

Hardware

Software emulation

Native: On device

Cross environment: Mocking APIs

Unit tests

Concretely: temperature data logger

Start with Adafruit DHT11 example code

Receive readings from an external sensor

```
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht->readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht->readTemperature();
```

Write them to Serial

```
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
```

Decompose

```
template <class DHTImpl>
class DHTtester {
private:
  DHTImpl* dht;
public:
  DHTtester(DHTImpl* d) : dht(d) {}
  void setup();
  void loop();
};

template <class DHTImpl>
void DHTtester<DHTImpl>::setup() {
  {
    Serial.begin(9600);
    Serial.println("DHTxx test!");

    dht->begin();
  }
}
```

Isolate

Mock your external dependencies

```
class DHT {
  public:
   DHT(uint8_t pin, uint8_t type, uint8_t count=6);
   void begin(void);
   float readTemperature(bool S=false, bool force=false);
   float convertCtoF(float);
```

```
class DHTMock : public DHT_ {
public:
  MOCK_METHOD0(begin, void());
  MOCK_METHOD2(readTemperatureImpl, float(bool, bool));
  virtual float readTemperature(bool S=false, bool force=false) { return readTemperatureImpl(S, force);
  MOCK_METHOD1(convertCtoF, float(float));
```

Test

Framework

Google Test: C++ testing and mocking
arduino-mock: Google Test stub library for Arduino

Test: setup

```
TEST(DHTtester, setup) {
  SerialMock* serialMock = serialMockInstance();
  DHTMock* dhtMock = DHTMockInstance();
  DHTtester<DHTMock> dhttester(dhtMock);

  EXPECT_CALL(*serialMock, begin(_));
  EXPECT_CALL(*dhtMock, begin());

  dhttester.setup();

  releaseDHTMock();
  releaseSerialMock();
}
```

Test: loop

```
TEST(DHTtester, loop) {
```

```
  SerialMock* serialMock = serialMockInstance();
  DHTMock* dhtMock = DHTMockInstance();
  DHTtester<DHTMock> dhttester(dhtMock);

  float humidity = 55.5;
  float tempC = 22.2;
```

```
  EXPECT_CALL(*dhtMock, readHumidityImpl(false)) // not forced
    .WillOnce(Return(humidity));
  EXPECT_CALL(*dhtMock, readTemperatureImpl(false, false)). // celsius, not forced
    WillOnce(Return(tempC));
```

```
  EXPECT_CALL(*serialMock, print(Matcher<double>(humidity), _));
  EXPECT_CALL(*serialMock, print(Matcher<double>(tempC), _));
```

```
  dhttester.loop();

  releaseDHTMock();
  releaseSerialMock();
```

Summary

Small, specific, and independent

Isolate your dependencies

Automate

Questions?

References

github.com/charles-haynes/unit-test-all-the-things (https://github.com/charles-haynes/unit-test-all-the-things)

Google Test (https://github.com/google/googletest)

Arduino Mock (https://github.com/wjszlachta/arduino-mock)

ESP8266 Geolocation library using Google Test for unit testing (https://github.com/CanTireInnovations/esp8266-geolocation-publisher)

A Set of Unit Testing Rules (http://www.artima.com/weblogs/viewpost.jsp?thread=126923)

ArduinoUnit: an on-device unit testing framework (https://github.com/mmurdoch/arduinounit)

Fake Function Framework (fff): one header file, but only subclassing (https://github.com/usr42/fff)

CMock: autogenerated mocks, C only, needs Ceedling (https://github.com/ThrowTheSwitch/CMock)

PlatformIO: alternative IDE, built in unit testing framework (http://docs.platformio.org/en/latest/what-is-platformio.html)

QEMU support for Xtensa (ESP8266) (http://wiki.osll.ru/doku.php/etc:users:jcmvbkbc:qemu-target-xtensa)

QEMU support for AVRs (Arduino) (https://lists.gnu.org/archive/html/qemu-devel/2016-07/msg00098.html)

ESP8266 Simulator: Allows Native Code to Link/Run on Linux (https://github.com/afnid/espsim)

Thank you

Charles Haynes
ceh@ceh.bz (mailto:ceh@ceh.bz)