Test driven development for embedded C by James Grenning

* Test-Driven Development was the key to making meaningful progress on the code before hardware and throughout the development cycle
* DLP means debug later programming which is anti TDD
* TDD microcycle
* Refactoring : means cleaning up code
* There is Red Green Refactor from the java world
* A unit test harness is a software package that allows a programmer to express how production code should behave

My plan is to do tdd in eclipse so at first make the eclipse like env ready follow the better arduino env guide by me

<https://vimeo.com/131194135>

**Faking and Mocking Legacy Embedded C - James Grenning**

Kinds of test doubles

* Fakes
* Spy
* Null object
* Exploding fakes
* Mock object
* Others

When must you use test doubles ?

1. When the code under test cannot be conveniently tested with real collaborators
2. When manual verification is needed (printed output, user interaction)
3. User Interface ( use model view controller )
4. Database
5. When the results change (Time, random events)
6. When failures need to be simulated ( Network down )
7. Operating system calls, file system calls
8. Special Hardware Interaction

Let’s say our end code is a Light Scheduler. It uses Light Controller & Time Service also

Light Controller uses

/\*\* UML pic

\*\*/

Inheritance = Base class: child class = is A relationship = generalization = extends

Child class can override functions of the Base class

If a Function of a class is virtual

If a function of a base class is made pure virtual function then the class becomes Abstract Class then that function must be override in the child class. = implements

Class A {

B b;

}

If a class has other class type variables as members then this relationship is called Aggregation = has a relationship