How To Extend A Library Using Hierarchy <under construction>

Navigation

What is hierarchy?

A hierarchy is a pyramidal arrangement of items (objects, names, values, categories, etc.), in which the items are represented as being "above," "below," or "at the same lev as" one another. Wik

Hierarchy is just a technical way of describing the relationship between classes in c++. You would probably not hesitate if said *Both a Circle and a Triangle is a Shape* but it might seem wierd if I say that *Both class Circle and class Triangle inheri from class Shape*

Hierarchy is also a way of making a project modular.
If you have a set of classes that you can choose from, building a new class becomes much easier.
Say you'll want to make a new class called TemperatureSensor, this class will need to provide the user with:
void begin();
int read();
void debug(char debugString);*

Let's say you have access to these classes:

```
#ifndef DEBUGGABLE_H#define DEBUGGABLE_Hclass Debuggable {    public:        inline void debug(char* debugString){            Serial.println(debugString);        }};#endif
#ifndef SENSOR_H#define SENSOR_H//abstract class Sensorclass Sensor {    public:        //initialize the sensor        inline virtual void begin(){ /*nothing*/ };        //read function must be
```

Now, you can create a TemperatureSensor like this:

```
#ifndef TEMPERATURESENSOR_H#define TEMPERATURESENSOR_H#include <WProgram.h>//include classes Sensor and Debuggable#include "Sensor.h"#include "Debuggable.h
```

And later, you could add a LightSensor like this:

```
#ifndef LIGHTSENSOR_H#define LIGHTSENSOR_H#include <WProgram.h>//include classes Sensor and Debuggable#include "Sensor.h"#include "Debuggable.h"class LightSensor : pu
```

The

Why do we use hierarchy?

How to make an inherited class?

This section will walk you through all the steps of making an inherited class from idea to result.

Define wanted functionality

The first thing you will need to do is to define what you want to add to the base class. This could be done in a series of way
I suggest starting with a list of desired functions and a short description of what they should do.

boolean begin();return true if device has been initialized successfully initializes the target deviceboolean available();return true if device is avalable, false if not.it should not try to poll device if no connection is already established

Familiarize with the superclass

You will need to know what data and functions you'll inherit, and thus being able to use. It will be a timesaver to either memorize the functions, or to have the headerfile easily accessible.

Learn c++ hierarchy syntax

Header Example

```
. #ifndef MYINHERITEDCLASS_H
. #define MYINHERITEDCLASS_H
.
. #include "MyClass.h"
.
. class MyInheritedClass : public MyClass {
. public:
.     void myClassFunction();
. };
.
. #endif
```

[Get

'Code Analysis:'

Lines 1 and 2 are referred to as an include guard
It basically prevents the code to get included into the program binary multiple times.
Line 4 includes MyClass
Line 6 is the beginning of the class itself. The ' : public MyClass' indicates inheritance.MyInheritedClass extends MyClass.
Line 7 uses the public keyword
Line 8 declares the function 'void myClassFunction();'
Line 9 is the end of the class.
Line 11 ends the #ifndef preprocessor directive

Implementation

First have a look at this: Basic Library Implementation

Debugging

Example implementation

I will extend the Button library and make a Switch library. The switch library will need both a press and a release in order change state. It's task is to transform a momentary switch (button) to a on-off pushbutton using software logic.

Define wanted functionality

add update()add isOn()add isOff()

Base class study

Public datas and functions

- #define PULLUP HIGH
- #define PULLDOWN LOW
- Button(uint8_t switchPin, uint8_t switchMode=PULLDOWN);
- void pullup();
- void pulldown();
- bool isPressed();

[Get

Create inherited class

//Under Construction

Usage of inherited class

//Under Construction

Links

Information about this page

Part of AlphaBeta Tutorials.
Last Modified: April 16, 2011, at 08:18 AM
By: wyojustin