mchr3k / **arduino**

Arduino projects http://mchr3k-arduino.blogspot.com/

75 commits | **1** branch | **0** releases | **1** contributor

Branch: **master** ▾ | New pull request | Find file | Clone or download ▾

Martin Add errorcount details, move temp calcs into the arduino — Latest commit `62feca7` on 23 Mar 2012

LEDs	Add LED code	6 years ago
TinyISP	Switch to use an Arduino 1.0 compatible core	6 years ago
attiny_int	Update interrupt example for use with ATmega328	6 years ago
hardware	Comment out boards which I'm not using, fix typo in wsn_arduino, add …	6 years ago
libraries	Cleanup some files, add support for deleting single files and cating …	6 years ago
serialdownload	Speed up serial connection	7 years ago
tempsensor	More wsn_arduino work	6 years ago
timelapseservo	Add time lapse code	7 years ago
wsn_arduino	Add errorcount details, move temp calcs into the arduino	6 years ago
wsn_attiny	Add errorcount details, move temp calcs into the arduino	6 years ago
wsn_server	Speed up data download, add local temp sensor, minor cleanup of temp …	6 years ago
.gitignore	Move c# code	7 years ago
.gitmodules	Add submodule link to manchester lib	6 years ago
README.markdown	Commit details about making changes to libs	6 years ago

README.markdown

Read this first

This project uses git submodules to include the Manchester library from (arduino-libs-manchester). The correct way to interact with this is as follows.

Getting started

Use one of the following:

```
git clone --recursive <git url> .
```

Or

```
git clone <git url> .
git submodule update --init
```

Picking up a change to the Submodule

A git submodule refers to an external repository at an explicit revision. There are two ways to update the content of a submodule.

```
git submodule update libraries/Manchester
```

This will discard any changes in libraries/Manchester and fetch the current fixed revision associated with the submodule.

```
cd libraries/Manchester
git pull origin master
cd ..
git add Manchester
git commit -m 'pickup latest changes from external repo'
```

This is the correct approach to fetch the latest changes from the external repository. The git pull within the submodule folder does the fetching from the external repository. The subsequent git commit in the parent folder publishes into the enclosing repository that the revision of the submodule has been changed.

Making changes to a submodule in place

Before you make any changes you have to modify the submodule checkout to actually refer to the master branch rather than a specific checkout.

```
cd libraries/Manchester
git checkout master
```

Once this is complete you can make changes and check them in as normal, treating libraries/Manchester as its own git repository. Any time you change the revision of the files within libraries/Manchester you should also choose whether to checkin the corresponding update to libraries/Manchester within the enclosing respository which still refers to an explicit revision.