Arduino CLI

A work in progress

by Dave Hylands
dhylands@gmail.com
https://github.com/dhylands/duino_cli

Disclaimer

I was hoping to have this all polished and installable using `pip install` but I've run into some glitches and need to rename all of my git repositories.

So today will mostly be a demo, and I will prepare detailed installation instructions once I get everything setup (should be done by the next meeting).

Arduino CLI

My goal was to create an extendable CLI that allow interacting with an Arduino like microcontroller. I also despise the Arduino IDE, and personally prefer using CLI based tools. To that end I’ve created a number of Arduino libraries along with a build system that can build using the official arduini-cli

duino_cli

This is the CLI, written in Python which interacts with the Arduino. It currently knows how to copy files to/from the LittleFS file system on the Arduino.

My intention is to make it so that you don’t modify these files directly, but rather that package plugins will be created to extend the functionality. The duino_littlefs package would be one of those. Users should be able to create their own packages for adding functionality.

Some of the duino_cli commands

Use the help command to find out what commands are available

```
CLI> help

Documented commands (type help <topic>):
========================================
args      echo  format  history  info  mkdir  quit  remove  upload
download  exit  help    hls      ls    ping   read  rmdir   write
```

To get more details, type help followed by the command:

```
CLI> help upload
usage: upload FILE DIR

Uploads FILE from the host to the Arduino. The file will be placed in the directory DIR.

positional arguments:
  FILE        Name of file to upload.
  DIR         Directory on Arduino to place the file in.

optional arguments:
  -h, --help  show this help message and exit
```

DuinoEsp32LittleFs

This is the example program which runs on the Arduino. It uses the duino_littlefs library to most of the work.

duino_util

This is a collection of low level common C++ Utility functions and classes.

AsciiHex - Produces ASCII hex dumps. Very useful for debugging packets.

Crc8 - Calculates CRC-8 over some data.

millis - Introduces a notion of time - useful for timeouts

strncpy_s - Version of strncpy which always creates a null terminated string

TimedActionSequence - Perform actions with a time delay between them. Useful for creating LED patterns

duino_bus

Defines the packet used to exchange data between the host and the Arduino. Also has code for sending packets over a serial link or socket link.

This particular package has both python and C++ implementations.

duino_log

Has logging functions.

Can output using the Arduino Serial object, or using linux files.

Under linux can generate colorized output.

Has a generic StrPrintf function which is fully re-entrant and can output using a function pointer. This allows writing directly to character based LCDs without having to create an intermediary character buffer.

duino_led

Has support for driving an LED using Arduino GPIO lines. Can also drive NeoPixel type LEDs.

When you create an LED you can specify whether it’s active low or active high. Then you turn your LED on or off without needing to know whether that particular LED needs GND or VCC to turn it on.

duino_littlefs

Contains a packet parser for parsing LittleFS related packets. There is also a packet handler which performs the appropriate actions on the littlefs file system.

Currently this package only contains C++ code, but it will eventually also have the littlefs python code that is currently in duino_cli.

duino_makefile

This library contains the common makefile snippets used by all of the other libraries. It isn't needed when using the Arduino IDE.

make compile - Compiles a library/program using arduino-cli (same as clicking the checkbox in the Arduino IDE)

make upload - Compiles a program (*.ino) using the arduino-cli and uploads it to the board

make style - Reformats C++ source code to have a consistent appearance.

make lint - Runs the C++ source code through cpplint

Duino_makefile - continued

make docs - Generates HTML documentation extracted from the source code.

make clean - Removes all object files

make uinttest - Runs the test suite

make coverage - Does coverage analysis to see how much of code is executed by the test suite.

make install-cli - Installs the arduino-cli tool and sets it up

make program - compiles a test program