readme for faster digitalWrite etc.

I have taken a stab at speeding up digitalWrite,digitalRead and pinMode in the Arduino core for cases where the pin number and HIGH/LOW/INPUT/OUTPUT state are know at compile time. 3 files need modification, one more new file is added to the core.

/Applications/Arduino.app/Contents/Resources/Java/hardware/arduino/cores/arduino/wiring_digital.c
This primarily gets altered names for the old versions of digitalWrite (__digitalWrite now), digitalRead, and pinMode. code to turn off analogWrite was commented out.

/Applications/Arduino.app/Contents/Resources/Java/hardware/arduino/cores/arduino/wiring.h
This gets the new macro definitions for digitalWrite, digitalRead pinMode and noAnalogWrite. If pin numbers and HIGH/LOW values are known at compile time, single instructions can be used. Otherwise the older, slower code is used, without turning off analogWrite/PWM.

/Applications/Arduino.app/Contents/Resources/Java/hardware/arduino/cores/arduino/
digital_write_macros.h contains complex macros referred to by wiring.h above.

/Applications/Arduino.app/Contents/Resources/Java/lib/keywords.txt
This gets the new keyword noAnalogWrite.

Compatibility
PWM is no longer turned off by digitalWrite. noAnalogWrite is a name proposed a few months ago for that functionality. (Honestly, I have serious doubts whether anyone commonly intermixed analogWrite and digitalWrite on a single pin. While it sort of makes sense in the context of an LED, it makes little sense for other devices and even with an LED why would you not analogWrite the maximum or minimum value??)

Because single instruction (often single byte) reads and writes are shorter than a subroutine call to the old digitalWrite etc., you should notice your sketches generally get shorter as well as faster. Faster may be a compatibility issue for some sketches.

Testing done
I have read and written each pin on a Mega and an Uno. I tested a Teensy sketch; that probably doesn’t actually use the new code in any way, but the sketch ran and so this didn’t break teensyduino. I ran a sketch that I use to test LCDs and one that uses OneWire and Wire.