AccelStepper library for Arduino

This is the Arduino **AccelStepper** library. It provides an object-oriented interface for 2, 3 or 4 pin stepper motors and motor drivers.

The standard Arduino IDE includes the Stepper library (http://arduino.cc/en/Reference/Stepper) for stepper motors. It is perfectly adequate for simple, single motor applications.

AccelStepper significantly improves on the standard Arduino Stepper library in several ways:

- Supports acceleration and deceleration
- Supports multiple simultaneous steppers, with independent concurrent stepping on each stepper
- API functions never delay() or block
- Supports 2, 3 and 4 wire steppers, plus 3 and 4 wire half steppers.
- Supports alternate stepping functions to enable support of AFMotor (https://github.com/adafruit/Adafruit-Motor-Shield-library)
- Supports stepper drivers such as the Sparkfun EasyDriver (based on 3967 driver chip)
- Very slow speeds are supported
- Extensive API
- Subclass support

The latest version of this documentation can be downloaded from http://www.airspayce.com/mikem/arduino/AccelStepper The version of the package that this documentation refers to can be downloaded from http://www.airspayce.com/mikem/arduino/AccelStepper/AccelStepper-1.57.zip

Example Arduino programs are included to show the main modes of use.

You can also find online help and discussion at http://groups.google.com/group/accelstepper Please use that group for all questions and discussions on this topic. Do not contact the author directly, unless it is to discuss commercial licensing. Before asking a question or reporting a bug, please read

- http://en.wikipedia.org/wiki/Wikipedia:Reference_desk/How_to_ask_a_software_question
- http://www.catb.org/esr/faqs/smart-questions.html
- http://www.chiark.greenend.org.uk/~shgtatham/bugs.html

Tested on Arduino Diecimila and Mega with arduino-0018 & arduino-0021 on OpenSuSE 11.1 and avr-libc-1.6.1-1.15, cross-avr-binutils-2.19-9.1, cross-avr-gcc-4.1.3_20080612-26.5. Tested on Teensy http://www.pjrc.com/teensy including Teensy 3.1 built using Arduino IDE 1.0.5 with teensyduino addon 1.18 and later.

Installation

Install in the usual way: unzip the distribution zip file to the libraries sub-folder of your sketchbook.

Theory

This code uses speed calculations as described in "Generate stepper-motor speed profiles in real time" by David Austin http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf or http://www.embedded.com/design/mcus-processors-and-socs/4006438/Generate-stepper-motor-speed-profiles-in-real-time or http://web.archive.org/web/20140705143928/http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf with the exception that **AccelStepper** uses steps per second rather than radians per second (because we dont know the step angle of the motor) An initial step interval is calculated for the first step, based on the desired acceleration On subsequent steps, shorter step intervals are calculated based on the previous step until max speed is achieved.

Adafruit Motor Shield V2

The included examples AFMotor_* are for Adafruit Motor Shield V1 and do not work with Adafruit Motor Shield V2. See https://github.com/adafruit/Adafruit_Motor_Shield_V2_Library for examples that work with Adafruit Motor Shield V2.

Donations

This library is offered under a free GPL license for those who want to use it that way. We try hard to keep it up to date, fix bugs and to provide free support. If this library has helped you save time or money, please consider donating at http://www.airspayce.com or here:

Donate

Trademarks

AccelStepper is a trademark of AirSpayce Pty Ltd. The **AccelStepper** mark was first used on April 26 2010 for international trade, and is used only in relation to motor control hardware and software. It is not to be confused with any other similar marks covering other goods and services.

Copyright

This software is Copyright (C) 2010 Mike McCauley. Use is subject to license conditions. The main licensing options available are GPL V2 or Commercial:

Open Source Licensing GPL V2

This is the appropriate option if you want to share the source code of your application with everyone you distribute it to, and you also want to give them the right to share who uses it. If you wish to use this software under Open Source Licensing, you must contribute all your source code to the open source community in accordance with the GPL Version 2 when your application is distributed. See https://www.gnu.org/licenses/gpl-2.0.html

Commercial Licensing

This is the appropriate option if you are creating proprietary applications and you are not prepared to distribute and share the source code of your application. Purchase commercial licenses at http://airspayce.binpress.com/

Revision History

Version

1.0 Initial release

1.1 Added speed() function to get the current speed.

1.2 Added runSpeedToPosition() submitted by Gunnar Arndt.

1.3 Added support for stepper drivers (ie with Step and Direction inputs) with _pins == 1

1.4 Added functional contructor to support AFMotor, contributed by Limor, with example sketches.

1.5 Improvements contributed by Peter Mousley: Use of microsecond steps and other speed improvements to increase max stepping speed to about 4kHz. New option for user to set the min allowed pulse width. Added checks for already running at max speed and skip further calcs if so.

1.6 Fixed a problem with wrapping of microsecond stepping that could cause stepping to hang. Reported by Sandy Noble. Removed redundant _lastRunTime member.

1.7 Fixed a bug where setCurrentPosition() did not always work as expected. Reported by Peter Linhart.

1.8 Added support for 4 pin half-steppers, requested by Harvey Moon

1.9 setCurrentPosition() now also sets motor speed to 0.

1.10 Builds on Arduino 1.0

1.11 Improvments from Michael Ellison: Added optional enable line support for stepper drivers Added inversion for step/direction/enable lines for stepper drivers

1.12 Announce Google Group

1.13 Improvements to speed calculation. Cost of calculation is now less in the worst case, and more or less constant in all cases. This should result in slightly beter high speed performance, and reduce anomalous speed glitches when other steppers are accelerating. However, its hard to see how to replace the sqrt() required at the very first step from 0 speed.

1.14 Fixed a problem with compiling under arduino 0021 reported by EmbeddedMan

1.15 Fixed a problem with runSpeedToPosition which did not correctly handle running backwards to a smaller target position. Added examples

1.16 Fixed some cases in the code where abs() was used instead of fabs().

1.17 Added example ProportionalControl

1.18 Fixed a problem: If one calls the funcion runSpeed() when Speed is zero, it makes steps without counting. reported by Friedrich, Klappenbach.

1.19 Added MotorInterfaceType and symbolic names for the number of pins to use for the motor interface. Updated examples to suit. Replaced individual pin assignment variables _pin1, _pin2 etc with array _pin[4]. _pins member changed to _interface. Added _pinInverted array to simplify pin inversion operations. Added new function setOutputPins() which sets the motor output pins. It can be overridden in order to provide, say, serial output instead of parallel output Some refactoring and code size reduction.

1.20 Improved documentation and examples to show need for correctly specifying **AccelStepper::FULL4WIRE** and friends.

1.21 Fixed a problem where desiredSpeed could compute the wrong step acceleration when _speed was small but non-zero. Reported by Brian Schmalz. Precompute sqrt_twoa to improve performance and max possible stepping speed

1.22 Added Bounce.pde example Fixed a problem where calling moveTo(), setMaxSpeed(), setAcceleration() more frequently than the step time, even with the same values, would interfere with speed calcs. Now a new speed is computed only if there was a change in the set value. Reported by Brian Schmalz.

1.23 Rewrite of the speed algorithms in line with http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf Now expect smoother and more linear accelerations and decelerations. The desiredSpeed() function was removed.

1.24 Fixed a problem introduced in 1.23: with runToPosition, which did never returned

1.25 Now ignore attempts to set acceleration to 0.0

1.26 Fixed a problem where certina combinations of speed and accelration could cause oscillation about the target position.

1.27 Added stop() function to stop as fast as possible with current acceleration parameters. Also added new Quickstop example showing its use.

1.28 Fixed another problem where certain combinations of speed and accelration could cause oscillation about the target position. Added support for 3 wire full and half steppers such as Hard Disk Drive spindle. Contributed by Yuri Ivatchkovitch.

1.29 Fixed a problem that could cause a DRIVER stepper to continually step with some sketches. Reported by Vadim.

1.30 Fixed a problem that could cause stepper to back up a few steps at the end of accelerated travel with certain speeds. Reported and patched by jolo.

1.31 Updated author and distribution location details to airspayce.com

1.32 Fixed a problem with enableOutputs() and setEnablePin on Arduino Due that prevented the enable pin changing stae correctly. Reported by Duane Bishop.

1.33 Fixed an error in example AFMotor_ConstantSpeed.pde did not setMaxSpeed(); Fixed a problem that caused incorrect pin sequencing of FULL3WIRE and HALF3WIRE. Unfortunately this meant changing the signature for all step*() functions. Added example MotorShield, showing how to use AdaFruit Motor Shield to control a 3 phase motor such as a HDD spindle motor (and without using the AFMotor library.

1.34 Added setPinsInverted(bool pin1Invert, bool pin2Invert, bool pin3Invert, bool pin4Invert, bool enableInvert) to allow inversion of 2, 3 and 4 wire stepper pins. Requested by Oleg.

1.35 Removed default args from setPinsInverted(bool, bool, bool, bool, bool) to prevent ambiguity with setPinsInverted(bool, bool, bool). Reported by Mac Mac.

1.36 Changed enableOutputs() and disableOutputs() to be virtual so can be overridden. Added new optional argument 'enable' to constructor, which allows you toi disable the automatic enabling of outputs at construction time. Suggested by Guido.

1.37 Fixed a problem with step1 that could cause a rogue step in the wrong direction (or not, depending on the setup-time requirements of the connected hardware). Reported by Mark Tillotson.

1.38 run() function incorrectly always returned true. Updated function and doc so it returns true if the motor is still running to the target position.

1.39 Updated typos in keywords.txt, courtesey Jon Magill.

1.40 Updated documentation, including testing on Teensy 3.1

1.41 Fixed an error in the acceleration calculations, resulting in acceleration of haldf the intended value

1.42 Improved support for FULL3WIRE and HALF3WIRE output pins. These changes were in Yuri's original contribution but did not make it into production.

1.43 Added DualMotorShield example. Shows how to use **AccelStepper** to control 2 x 2 phase steppers using the Itead Studio Arduino Dual Stepper Motor Driver Shield model IM120417015.

1.44 examples/DualMotorShield/DualMotorShield.ino examples/DualMotorShield/DualMotorShield.pde was missing from the distribution.

1.45 Fixed a problem where if setAcceleration was not called, there was no default acceleration. Reported by Michael Newman.

1.45 Fixed inaccuracy in acceleration rate by using Equation 15, suggested by Sebastian Gracki.
Performance improvements in runSpeed suggested by Jaakko Fagerlund.

1.46 Fixed error in documentation for runToPosition(). Reinstated time calculations in runSpeed() since new version is reported not to work correctly under some circumstances. Reported by Oleg V Gavva.

1.48 2015-08-25 Added new class **MultiStepper** that can manage multiple AccelSteppers, and cause them all to move to selected positions at such a (constant) speed that they all arrive at their target position at the same time. Suitable for X-Y flatbeds etc.
Added new method maxSpeed() to **AccelStepper** to return the currently configured maxSpeed.

1.49 2016-01-02 Testing with VID28 series instrument stepper motors and EasyDriver. OK, although with light pointers and slow speeds like 180 full steps per second the motor movement can be erratic, probably due to some mechanical resonance. Best to accelerate through this speed.
Added isRunning().

1.50 2016-02-25 **AccelStepper::disableOutputs** now sets the enable pion to OUTPUT mode if the enable pin is defined. Patch from Piet De Jong.
Added notes about the fact that AFMotor_* examples do not work with Adafruit Motor Shield V2.

1.51 2016-03-24 Fixed a problem reported by gregor: when resetting the stepper motor position using setCurrentPosition() the stepper speed is reset by setting _stepInterval to 0, but _speed is not reset. this results in the stepper motor not starting again when calling setSpeed() with the same speed the stepper was set to before.

1.52 2016-08-09 Added **MultiStepper** to keywords.txt. Improvements to efficiency of **AccelStepper::runSpeed()** as suggested by David Grayson. Improvements to speed accuracy as suggested by David Grayson.

1.53 2016-08-14 Backed out Improvements to speed accuracy from 1.52 as it did not work correctly.

1.54 2017-01-24 Fixed some warnings about unused arguments.

1.55 2017-01-25 Fixed another warning in MultiStepper.cpp

1.56 2017-02-03 Fixed minor documentation error with DIRECTION_CCW and DIRECTION_CW. Reported by David Mutterer. Added link to Binpress commercial license purchasing.

1.57 2017-03-28 _direction moved to protected at the request of Rudy Ercek. setMaxSpeed() and setAcceleration() now correct negative values to be positive.

Author

Mike McCauley (mikem@airspayce.com) DO NOT CONTACT THE AUTHOR DIRECTLY: USE THE LISTS