/\*

http://www.airspayce.com/mikem/arduino/AccelStepper/

Basic Usage

You can create multiple AccelStepper objects, giving a unique name to each motor. AccelStepper can manage any number of motors, as long as you repetitively call their "run" functions.

////Defining & Configuring Motors

**AccelStepper mystepper(DRIVER, step, direction);** // Шаговый двигатель, управляемый выделенной платой.

**AccelStepper mystepper(FULL2WIRE, pinA, pinB);** // Биполярный шаговый двигатель, управляемый Н-мостом.

**AccelStepper mystepper(FULL4WIRE, pinA1, pinA2, pinB1, pinB2);**// Униполярный двигатель, управляемый четырмя транзисторами.

**mystepper.setMaxSpeed(stepsPerSecond);** // Установка максимальной скорости (оборотов в минуту). Скорость по умолчанию очень низкая, так что её требуется переопределить. При движении шаговый двигатель будет ускоряться до этой максимальной скорости и замедляться при подходе к концу движения.

**mystepper.setAcceleration(stepsPerSecondSquared);** // Установка ускорения, в шагах в секунду за секунду.

////Position Based Control

**mystepper.isRunning();** // Возвращает true, если ШД в режиме вращения.

**mystepper.currentPosition();** // Возвращает текущее абсолютное положение (в шагах).

**mystepper.setCurrentPosition (long position);** // Установить счетчик как текущую позицию. Полезно как задание нулевой координаты. Обнуляет текущую скорость до нуля.

**mystepper.targetPosition();** // Конечное положение (в шагах).

**mystepper.distanceToGo();** // Вернуть расстояние до указанного положения. Может использоваться для проверки: достиг ли двигатель указанной конечной точки.

**mystepper.moveTo(long absolute);**  // Переместиться в абсолютно указанное положение. Само движение запускается функцией run().

**mystepper.move(long relative);** // Переместиться в относительно указанное положение. Само движение запускается функцией run(). Значение distance может быть больше или меньше нуля.

**mystepper.run();** // Начать движение с ускорением. Один шаг. Для продолжения движения следует вызывать функцию повторно как можно чаще - в loop() или иной функции.

**mystepper.runToPosition();** // Начать движение и подождать когда двигатель достигнет указанной точки. БЛОКИРУЮЩАЯ ФУНКЦИЯ - функция не прерывается, пока двигатель не остановится!

**mystepper.runToNewPosition();** // Начать движение с ускорением, до заданной позиции. БЛОКИРУЮЩАЯ ФУНКЦИЯ!

**mystepper.stop();** // Максимально быстрая остановка (без замедления), используя текущие параметры скорости и ускорения. Также задает новое конечное положение.

////Speed Based Control

**mystepper.setSpeed(stepsPerSecond);** // Установить скорость (в шагах за секунду). Само движение запускается функцией runSpeed().

**mystepper.runSpeed();** // Начать движение с текущей заданной скоростью (без плавного ускорения). Для продолжения движения двигателя следует вызывать функцию повторно как можно чаще.

**mystepper.runSpeedToPosition();** // Начать движение с текущей заданной скоростью (без плавного ускорения), до заданной позиции. БЛОКИРУЮЩАЯ ФУНКЦИЯ!

////Power Control

**mystepper.disableOutputs();** // Деактивировать занятые пины и установить их в LOW. Снимает напряжение с обмоток двигателя, экономя энергию (вал при это может проворачиваться под внешними нагрузками)

**mystepper.enableOutputs();**  // Активировать зарезервированные пины и устанавивает их в режим OUTPUT. Вызывается автоматически при запуске движения.

\*/