MOTORplate Quick Reference Guide

Revision 1.00

Stepper Motor Functions

stepperCONFIG(addr,motor,dir,resolution,rate,acceleration) - this command has to be executed to initialize a stepper motor. See the definitions below for a detailed explanation of each argument.

stepperMOVE(addr,motor,steps) - the command will move a stepper number a fixed number of steps using the acceleration, step rate, direction, and resolution values specified in the stepperCONFIG call.

stepperJOG(addr,motor) - this command will instruct the specified stepper to rotate at a rate using the acceleration, step rate, direction, and resolution values specified in the stepperCONFIG call.

stepperSTOP(addr,motor) - this will instruct the specified jogging stepper to come to a stop using the acceleration value specified in the stepperCONFIG call. If the acceleration value is set to zero, the motor will stop instantly. Once stopped, current will continue to flow through the windings to maintain the hold torque.

stepperRATE(addr,motor,rate) - this will change the step rate of a jogging or stationary stepper motor. If the motor is spinning at a steady step rate will change a using the acceleration value specified in the stepperCONFIG call. If the motor is in the process of accelerating, decelerating, or stopping, this command will be ignored.

stepperOFF(addr,motor) - remove power from a stopped motor. If not used, the current will continue to flow through the stepper windings maintaining the hold torque.

enablestepSTOPint(addr,motor) - generate interrupt when stepper motor come to a complete stop. Useful for knowing when a move command has completed or when a motor has come to a stop under acceleration and deceleration conditions.

disablestepSTOPYint(addr,motor) - turn off the STOP interrupt

enablestepSTEADYint(addr,motor) - generate interrupt when stepper motor speed has reached a steady rate. Useful for knowing when a rate change can be applied under acceleration and deceleration conditions.

disablestepSTEADYint(addr,motor) - turn off the STEADY interrupt

DC Motor Functions

dcCONFIG(addr,motor,dir,speed,acceleration) - this command has to be executed before starting up one of the DC motor outputs. See the definitions below for a detailed explanation of each argument.

dcSPEED(addr,motor,speed) - use this function to change the speed of stopped or running motor. If this command is issued while a motor is accelerating or decelerating it will be ignored.

dcSTART(addr,motor) - this command will instruct the specified DC motor to start running using the acceleration, speed, and direction values specified in the dcCONFIG call. This function can also be called to restart a stopped DC motor.

dcSTOP(addr,motor) - this command will instruct the specified DC motor to come to a stop using the acceleration value specified in dcCONFIG call. If the acceleration value is set to zero, the motor will stop instantly.

enabledcSTOPint(addr,motor) - generate interrupt when DC motor come to a complete stop. Useful for knowing when a move command has completed or when a motor has come to a stop under acceleration and deceleration conditions.

disabledcSTOPYint(addr,motor) - turn off the STOP interrupt

enabledcSTEADYint(addr,motor) - generate interrupt when DC motor speed has reached a steady rate. Useful for knowing when a speed change can be applied under acceleration and deceleration conditions.

disabledcSTEADYint(addr,motor) - turn off the STEADY interrupt

Sensor Functions

getSENSORS(addr) - read all four sensor inputs as a single byte. The order of data will be:

|0|0|0|0|Sensor 4|Sensor 3|Sensor 2|Sensor 1|

This instruction should be used when polling sensors that are being used as limit switch inputs.

getTACHcoarse(addr,tachnum) - each sensor input can also serve as a tachometer. The coarse value is a sixteen bit number that is updated eight times per second and will not be as accurate as the value obtained with the getTACHfine function.

getTACHfine(addr,tachnum) - this is a much more accurate 16-bit tachometer value that is updated once per second.

LED Control Functions

setLED(addr) - turn on the LED

clrLED(addr) - turn off the LED

toggleLED(addr) - if LED is on, turn off. If LED is off, turn on.

System Level Functions

getID(addr) - return Pi-Plate descriptor string

getFWrev(addr) - return FW revision in byte format

getHWrev(addr) - return HW revision in byte format

getADDR(addr) - return address of pi-plate. Used for polling available boards at power up.

intEnable(addr) - enable interrupts from the MOTORplate. GPIO22 will be pulled low if an enabled event occurs.

intDisable(addr) - disables and clears all interrupts on the MOTORplate.

getINTflag0(addr) - returns bit interrupt flag0 value then clears the register.

getINTflag1(addr) - returns bit interrupt flag1 value then clears the register.

Poll() - determine what Pi-Plates are on stack as well as their addresses

RESET(addr) - use this function to return the MOTORplate to a known state. Stops and removes power to the motors. Also clears out all configuration settings.

Definitions

addr: Address - MOTORplates have jumpers on the board that allow their address to be set to a value between 0 and 7.

motor: Motor Designator - used for selecting the motor to drive. For stepper motors this is either an 'a' or a 'b' - the case is insensitive but the characters have to be enclosed in quotation marks. For DC motors, use the numbers 1 through 4 to select the motor to drive.

dir: Direction - the two accepted arguments are clockwise 'cw' and counter clockwise 'ccw' - the case is insensitive. These are somewhat arbitrary and are dependent on how you have wired up your motor. So, some experimentation may be required.

resolution: Stepper Resolution - MOTORplates can drive stepper motors usingfour different step resolutions. These can be specified with either a number or characters. Again, for characters, the case is insensitive butthey have to be enclosed in quotation marks:

Full Steps: 0 or 'F'

Half Steps: 1 or 'H'

Four Microsteps: 2 or '4M'

Eight Microsteps: 3 or '8M'

rate: Stepper Rate - MOTORplates can drive stepper motors at step rates from1 to 2000 steps per second. Values outside of these limits will produce an error. Note that the maximum step rate you can achieve is dependent on the motor, the motor voltage, your step rate, your load, and your acceleration rate.

acceleration: Motor Acceleration Time - MOTORplates have the ability to gradually ramp up (and down) to the desired motor step rate or speed. This feature prevents the motor from stalling under high load conditions. Values for acceleration range from 0.0 for no acceleration to 5.0 seconds.

steps: Step Count - This value is used when moving a stepper motor a fixed number of steps this value can range from 1 to 65,535. The actual amount of travel will be dependent on the motor and the resolution.

speed: DC Motor Speed - This is a value that represents the speed of the DC motor in percent. It ranges from 0.0 to 100.0. The actual speed will be dependent on the motor, the motor voltage, and any gearing

tachnum: Tachometer Number - there are four sensor inputs / tachometers on the MOTORplate. These are referenced with the numbers 1 through 4.