

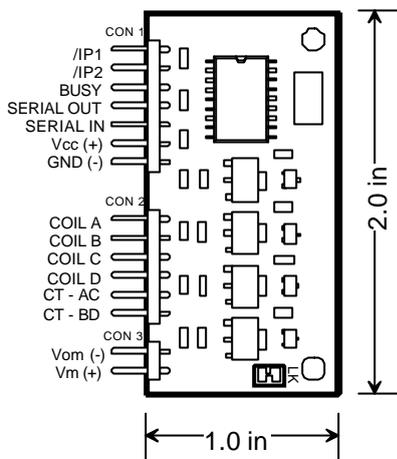
### Description

The **Little Step-U** is a complete, serially controlled, drive system for unipolar stepper motors. Using an intelligent module allows the host system to concentrate on the task at hand while the **Little Step-U** performs all calculations and operation of the motor.

The desired operating speed, ramp time and drive mode can be configured once and then a single command used as required, to move to fixed or relative positions. While the motor is in motion, a BUSY output is active and the movement can be optionally interrupted by one of the two external inputs. The position of the motor is maintained in a readable register which can be set to zero (Home) by a command or when an input is activated. A similar register is available to save a second position (Mark) either by command or from an input. These points can then be referenced directly with a 'Move to home' or 'Move to mark' command.

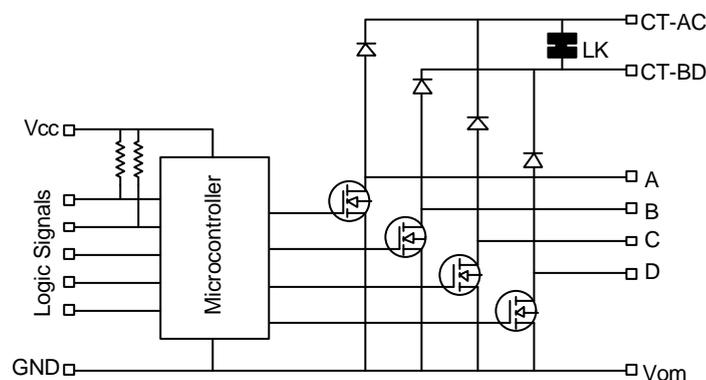
A built-in ramping function accelerates the motor from a standstill to the desired speed over a specified time period, allowing faster final speeds to be achieved, and also to decelerate the motor to give maximum load braking. When the motor is stopped, a command specifies whether the windings are left energized (for maximum holding torque) or turned off (for lower power consumption / dissipation).

A remote function is included for use in more complex systems. Once activated, the two external inputs become STEP and DIR, with the motor stepping in the indicated direction with every pulse applied to the step input.



### Features

- Unipolar motors to 35V, 3.0A
- Full step, half step and wave drive modes
- 1-5,000 steps per second
- Inbuilt speed ramp (0.1-20.0 seconds)
- Simple serial commands (TTL interface)
- Separate power input for motor and logic
- Suitable for direct and L/R operation in all modes
- On board clamping diodes
- Current position register (+/-100,000,000)
- Zero position and user marker registers
- Absolute & Relative movement commands
- Busy output
- Two inputs suitable for optical stops
- Slave mode using inputs for STEP & DIR
- Stop with motor coils on or off



## DC Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Logic Supply Voltage	$V_{CC}$	4.5	5.0	5.5	V	
Vcc risetime	$t_{VCC}$	0.05			V/ms	1
Logic supply current	$I_{CC}$		1.5	3.0	mA	
Input Low Voltage	$V_{IL}$	0.0		1.0	V	
Input High Voltage	$V_{IH}$	4.3		$V_{CC}$	V	
Output Low Voltage	$V_{OL}$			0.6	V	
Output High Voltage	$V_{OH}$	4.3			V	
IP1/2 Source Current	$I_{OL}$		1.0		mA	
Motor Voltage	$V_M$			50	V	
Motor current	$I_M$			3.0	A	
Motor resistance	$R_M$	2.8			$\Omega$	
Clamp diode current	$I_{CL}$			250	mA	
Step rate	$F_{STEP}$	1		5000	steps/sec	
Speed Error	$F_{ERR}$			+/- 0.5	%	2
Comms bit rate	$F_{BIT}$		2400		bps	
BUSY rise/fall time	$T_{RF}$		10	25	ns	
Response time to IP1/2	$T_{INP}$			1	ms	3
DIR hold time	$T_{DIR}$	8			us	
Max Slave step rate	$F_{EXT}$			10000	steps/sec	4
Operating temp	$t_{OP}$	0		70	$^{\circ}C$	5
Storage temp	$t_{ST}$	-40		100	$^{\circ}C$	

- [1] Required to ensure reliable startup.  
 [2] As a percentage of requested speed  
 [3] Response to IP is lesser of 1ms or 1 step  
 [4] If no comms while stepping. Max = 6000 steps/sec with comms  
 [5] Refer loading derating graph

## Pin functions

Connector 1

Pin	Function
/IP1	Optional input 1. Set home. Emergency stop.
/IP2	Optional input 2. Set mark. Emergency stop.
BUSY	Output - High when motor moving
SERIAL OUT	Data from <b>Little step-U</b> to controller. TTL
SERIAL IN	Data from controller to <b>Little Step-U</b> . TTL
Vcc	Supply to logic circuits
GND	Supply ground. Common with motor circuits.

Connector 2

Pin	Function
A	Winding A output
B	Winding B output
C	Winding C output
D	Winding D output
CT-AC	Common for windings A & C
CT-BD	Common for windings B & D

Connector 3

Pin	Function
$V_{OM}$	Motor supply ground. Common with logic circuits
$V_M$	Motor Supply +

## Hardware Connection

### Power Supply

The **Little Step-U** divides the circuitry into two distinct sections with separate power supplies required for the motor drive and the logic. The logic/control section must be powered by a regulated 5V supply applied to the VCC and GND pins. The motor power supply voltage is dictated by the motor characteristics and is connected to the VM(+) and VOM(-) pins. The GND and VOM pins are connected on the circuit board as a common reference. The motor power supply must have sufficient capacity for the motor and if the power supply is not physically close to the motor/controller, a capacitor (eg. 4700uF) may be required at the controller to ensure smooth operation of the motor.

### Motor Connection

The motor connector has 4 connections (A, B, C & D) that are switched to ground in the sequence required to cause the motor to step. Note that stepper motor manufacturers have several methods of labelling the connections to the motor. The **Little Step-U** labels the pins in the sequence they are switched when operating the motor with one coil on at a time and driving in a clockwise direction. ie. A, then B, then C, then D and back to A. The centre taps of the two coils are connected to CT-AC and CT-BD, corresponding to the common connection of coils A & C and coils B & D respectively. These two connections provide the freewheel diodes for each of the motor coils and are connected on the circuit board.

**IMPORTANT:** A poor or intermittent connection to any of the stepper motor terminals can result in destructive voltages being generated and causing permanent damage to the Little Step-U. Similarly, do not connect or disconnect any stepper motor connections while power is applied.

The connection point between CT-AC and CT-BD is indicated on the circuit board by the designator "LK". This track can be cut and rejoined by soldering a wire between two pads if needed. In most cases this isn't required but is provided for applications using and L/xR configuration in half step mode.

### Interface connections

The Little Step-U acts on commands received serially from a host controller via the SERIAL IN pin. This is an asynchronous interface operating at 2400 bits per second, 8 data bits (LSB first), 1 stop bit and no parity. The signal levels are TTL. Some commands illicit a response and these are sent via the SERIAL OUT pin using the same protocol and signal levels as the SERIAL IN pin.

When a command has been sent to the Little Step-U that causes the motor to move, the BUSY pin will be high while the motor is in motion.

Two input pins are provided for use with stepper operation in the form of limit or home switches or can be used as general inputs which can be read by software. These pins accept logic level inputs and will source 1mA when pulled low, making them suitable for direct connection to the output of opto-couplers.

## Software commands

Communication with the Little Step-U is by a TTL level serial interface at 2400bps. All commands must begin with a "{" character and end with a "}" Characters outside of the braces (including carriage return and line feeds) are ignored. Responses are enclosed by "[" and "]" characters.

### A - SET SPEED

Syntax {Axxxx} Default value 100 steps/sec

Sets the maximum speed in steps per second for any commands that follow. The numeric value can be any number from 1 to 5000.

### B - SET RAMP TIME

Syntax {Bxxx} Default value 0 seconds

The Little Step-U has a built in function to accelerate and decelerate the stepper motor to/from the speed specified by the command above. This command sets the amount of time in tenths of a second that the motor will accelerate to that speed. Valid numbers are 0 to 200 being 0.0 to 20.0 seconds in 0.1 second increments.

Any move function that utilises the ramping facility will accelerate and decelerate at the rate defined by this command. The distance moved includes the steps used to ramp up and ramp down so a move of 1000 steps will move exactly that amount. If the number of steps requested is less than the number required to ramp to full speed and ramp down again, the Little Step-U will accelerate at the same rate until it is half way to its destination and then ramp down again to stop at the commanded position.

See the discussion on ramping after the command descriptions.

### C - SET STEP MODE

Syntax {Cx} Default value 0 (Full step drive)

Stepper motors can be driven by powering the windings sequentially using three different techniques. The method used is specified by this command with x being 0, 1 or 2.

{C0} Full step  
For every step, two windings are energised.

Step	Coil A	Coil B	Coil C	Coil D
1	ON	ON	Off	Off
2	Off	ON	ON	Off
3	Off	Off	ON	ON
4	ON	Off	Off	ON

{C1} Wave  
Each step has only one winding energised.

Step	Coil A	Coil B	Coil C	Coil D
1	ON	Off	Off	Off
2	Off	ON	Off	Off
3	Off	Off	ON	Off
4	Off	Off	Off	ON

{C2} Half step  
 Each alternate step has one or two windings energised giving the motor twice the number of steps per revolution.

Step	Coil A	Coil B	Coil C	Coil D
1	ON	Off	Off	Off
2	ON	ON	Off	Off
3	Off	ON	Off	Off
4	Off	ON	ON	Off
5	Off	Off	ON	Off
6	Off	Off	ON	ON
7	Off	Off	Off	ON
8	ON	Off	Off	ON

When the mode is changed to or from Half step, the position, mark and speed values are adjusted accordingly so that the motor will turn at the same rotational speed and the mechanical positions remain unchanged.

Example: A 200 step/rev motor is being used. The current position is 3600, speed is set at 200 steps per second and there is a mark at 1200. When the mode is changed from full-step to half step, the speed value will be 400 (giving the same number of revolutions of the motor per second). The current position register and the mark register will contain the values 7200 and 2400 respectively.

## D - GO TO ABSOLUTE POSITION

Syntax {Dxxxxxxx}

At anytime after powerup, the Little Step-U keeps track of its current position in steps. This command instructs the Little Step-U to calculate the direction and number of steps required to move to another position and then it performs the move. At the end of the move, the position will be the number specified in this command. The value can be any number from -10000000 to +10000000, including 0. The number may be preceded by a "+" or "-" as required, but must not include a comma or decimal point.

Example: If the current position is 2500, a command of {D-1000} will cause the motor to run counter-clockwise for 1500 steps, stopping at position 1000.

## E - GO TO RELATIVE POSITION

Syntax {Exxxxxx}

This command is similar to the "D" command in respect to the numeric parameter, but commands the motor to move to a position relative to its current position. Positive numbers are clockwise and negative numbers are counter-clockwise.

Example: If the current position is 2500, a command of {E1000} will cause the motor to run 1000 steps clockwise to end at position 3500.

## F - GO TO ABSOLUTE POSITION WITH INPUTS 1 & 2 AS STOPS

Syntax {Fxxxxxx}

This command is the same as the "D" command except that the two input pins are monitored. If either of them is pulled low during the move, the motor will be decelerated to a stop at the rate defined in the RAMP command (B). The position register will reflect the current position when the motor has stopped. These inputs are level sensitive so if a "F" command is used when one of the inputs is low, the motor will stop immediately.

Note: Allowing the motor to continue to running past a stop indicator is done for two reasons; This method allows the motor to provide maximum braking torque during deceleration and the position register integrity is maintained.

Example: If the current position is 2500 and a limit switch attached to I/P2 is at position 5000, the command {F8000} will result in the motor stopping at position 5087 (depending on the SPEED and RAMP settings).

## **G - GO TO RELATIVE POSITION WITH INPUTS 1 & 2 AS STOPS**

Syntax {Gxxxxx}

This command is the same as the "E" command except that the two input pins are monitored as in the command above.

Example: If the current position is 2500 and a limit switch attached to I/P1 is at the home position (0), the command {G-5000} will result in the motor stopping at position -87 (depending on the SPEED and RAMP settings).

## **H - GO CLOCKWISE UNTIL INPUT 2**

Syntax {H}

This command is used to find a limit stop connected to I/P2 and has no numeric parameter. The motor will accelerate at the rate determined by the RAMP setting, up to the SPEED setting and not stop (ramp down) until I/P2 is pulled low.

Example: The motor is at position 28 and a limit switch is at position 1000. The SPEED has been set to a slow value and RAMP set to 0. After an {H} command the position is 1000.

## **J - GO COUNTER-CLOCKWISE UNTIL INPUT 1 AND ZERO POSITION**

Syntax {J}

The J command is similar to the H command above except that the direction is reversed and I/P1 is used to halt the movement. When the movement is stopped, the position register is cleared to 0 to mark this position as "Home".

Example: The motor is at an unknown position and a {J} command is used at a slow speed and 0 RAMP rate. At the end of the movement (dictated by I/P1) the position register is 0.

## **K - GO CLOCKWISE UNTIL INPUT 2 AND MARK POSITION**

Syntax {K}

The motor runs continuously in a clockwise direction until stopped by I/P2 being pulled low. When this occurs, the position is recorded in the MARK register.

Example: The motor has been moved to the home position using the {J} command. Issuing a {K} command will run the motor and when it has stopped, the position of the other limit switch has been recorded.

## **M - GO TO THE MARK POSITION**

Syntax {M}

This command instructs the Little Step-U to move to the absolute position held in the "Mark" register. The contents of this register is set to 0 when the device is powered on and can be set using the "K" or "R" commands. As with the "D" (Go to absolute position) command, the controller will determine the direction and will use the Ramp and Speed settings to move to that position.

Example: The Little Step-U is being used in a component placement machine and the position to pick up the next component is known to always be 123456. After each placement, a {M} command will place the pickup head over the next component to be placed.

## **N** - GO TO THE HOME POSITION

Syntax            {N}

Sending an "N" command will cause the Little Step-U to move the motor to the 0, or home position. The position register defaults to 0 when the device is powered on and can be set to 0 by the "J" or "Q" commands.

Example: A Little Step-U is being used to position a print head in a dot matrix printer. At each Carriage Return in the data being printed, an {N} command is issued to return the head to the left margin, ready for the next line.

## **P** - SET OFF STATE POWER

Syntax            {Px}

When the motor is not being moved, the coils can be left energised for maximum holding torque or turned off to reduce power consumption and motor heating. This command sets the state as off {P0} or on {P1} and will remain that way until changed with another "P" command. Power on default is off.

Example: A motor is being used in a high vibration environment and retaining position is more important than power consumption. The {P1} command ensures maximum holding torque.

## **Q** - SET HOME POSITION

Syntax            {Q}

This command unconditionally sets the position register to 0. All position information and absolute moves become relative to this new home position.

Example: A drilling sequence needs to be repeated several times over a panel. The total size of the program can be reduced by moving the offset amount, resetting the home position with the {Q} command and repeating the same sequence.

## **R** - SET MARK

Syntax            {R}

This command instructs the Little Step-U to remember the current position. It is used in conjunction with the {M} command which will return the motor to this same position regardless of any intervening moves.

Example: A stepper motor is being used in a piece of hastily constructed test equipment to rub an abrasive over a surface. The distance is found by trial and error and a mark is set with the {R} command. The experiment can then be run using the {Q} (Go to home) and {R} (Go to mark) commands in a loop.

## **S** - SWITCH TO REMOTE MODE

Syntax            {S}

The Little Step-U can be operated as simple translator/drive for use in systems with enhanced motion control capability or for the purposes of experimentation. This mode of operation is entered by sending the {S} command. Input pin 1 will then act as the STEP input and Input pin 2 will act as a DIRection input.

On each high to low transition (falling edge) of the STEP input, the motor will move one step in the direction indicated by the DIR input. A high on the DIR input will cause the motor to step clockwise and a low, counter-clockwise. This input must be stable when STEP is taken low and for tDIR afterwards. When in REMOTE mode, the mode settings that were set with the {C} command (wave, full, half step) remain in force and the position

register is updated with each step. The {S} command will turn the motor windings on, regardless of the value set by the {P} command.

Example: During the commissioning phase of a machine, the operator wishes to move to a target position one step at a time using a pushbutton. The {S} command is sent and the microcontroller is used to debounce each button press and apply a single pulse. When the desired destination is reached, the position is read using the {U} command.

## **T** - RETURN FROM REMOTE MODE

Syntax            {T}

When the Little Step-U is operating in the REMOTE mode, sending a {T} will return to the standard operating mode. The position register contents are valid and the motor windings will be powered on or off according to the value previously set by the {P} command.

Example: Once the setup phase described above has been completed, a {T} command readies the controller for full speed operation.

## **U** - SHOW CURRENT STATUS

Syntax            {U}  
Returns          [aaaa,bbbb,ccc]

The {U} command instructs the Little Step-U to send the current values of the Position, Speed and Ramp registers back to the controller. The data consists of enclosing square brackets "[" and "]" containing the three (decimal) values separated by commas.

Example: A stepper is being used to move a print head mechanism. In order to determine the paper width the user has selected, the motor is moved to one stop and zeroed and then moved slowly to the other stop with RAMP set to 0. When the motor stops (BUSY goes low), the position is retrieved with the {U} command. The controller can then use this value to scale the print head movement commands and ensure that an image will fit on the page.

## **V** - SHOW STATE OF INPUT PINS

Syntax            {V}  
Returns          [a,b]

The {V} command returns the current state of the two input pins in the form of "0" for a low input and "1" for a high.

Example: An {G-1000} command is issued which instructs the Little Step-U to move 1000 steps counter-clockwise from the current position and to stop if it encounters a low input on either IP1 or IP2. Checking that the complete move was made successfully is made easier by only having to check two single digits with the {V} command, than subtracting the current position from the earlier position and checking the result.

### ***Multi-parameter commands***

Movement related commands have the option of including speed, ramp time and/or step mode values in the form of {X,speed,ramp,mode} where X is the command listed below and speed, ramp and mode are valid values as described for the A, B and C commands above. Each parameter is separated by a comma and may either be a value or no value (without spaces) if that parameter is to remain unchanged. Any setting changed by using a command in this form will remain at the new value for any subsequent commands unless explicitly changed.

**D**            GO TO ABSOLUTE POSITION  
**E**            GO TO RELATIVE POSITION

<b>F</b>	GO TO ABSOLUTE POSITION WITH INPUTS 1 & 2 AS STOPS
<b>G</b>	GO TO RELATIVE POSITION WITH INPUTS 1 & 2 AS STOPS
<b>H</b>	GO CLOCKWISE UNTIL INPUT 2
<b>J</b>	GO COUNTER-CLOCKWISE UNTIL INPUT 1 AND ZERO POSITION
<b>K</b>	GO CLOCKWISE UNTIL INPUT 2 AND MARK POSITION
<b>M</b>	GO TO THE MARK POSITION
<b>N</b>	GO TO THE HOME POSITION

Examples:

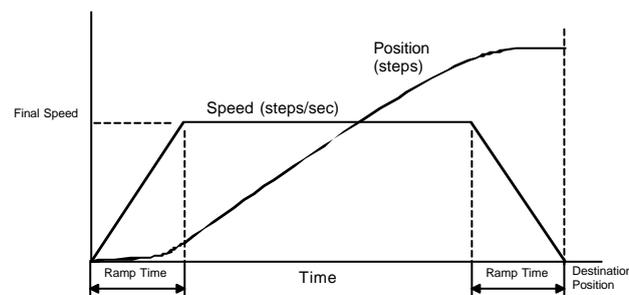
{D-2000,200,5}            Go to absolute position -2000 with a maximum speed of 200 steps per second, accelerating to that speed over a period of 0.5 seconds and use the currently set drive mode.

{N,,,0}            Go to the HOME position using the current speed and ramp settings but change the drive mode to full step.

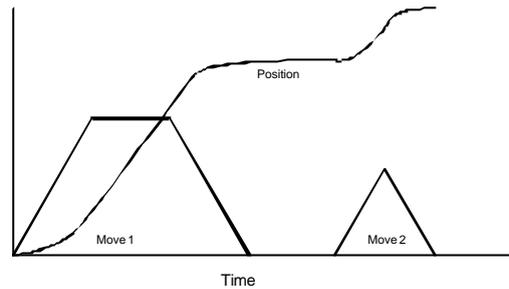
### ***Ramping or Trapezoidal speed profile.***

The ability to accelerate a stepper motor over a period of time, greatly improves the useful range of operations the motor can perform. Not only does this allow the motor to reach higher speeds, but also reduces wear on mechanical components. While acceleration is defined as a change in speed over a period of time (steps/sec/sec), it is easier to think of the motor accelerating to a desired speed over a defined time period, so the B command is used in this manner.

The graph below illustrates a typical move where the final speed, ramp time and destination position have been specified. Both the position (in steps) and the speed (in steps per second) show how the motor moves slowly at the beginning, at the maximum rate for the majority of the move and then transitions smoothly into the deceleration phase, before coming to a stop at the desired position. Because of the shape of the curve, the speed plot is known as a trapezoidal speed profile.



When the distance to be moved is less than that required to ramp up to full speed and then ramp down again, the acceleration rate is kept the same and half way through the move, the Little Step-U will begin the deceleration. An example is illustrated below. The speed and ramp values are unchanged for two discrete move operations. The first is long (far) enough for the motor to reach the specified operating speed but because Move 2 is shorter, the motor transitions directly from accelerating in speed to the deceleration profile.



The ramp function used in the Little Step-U is a piecewise, linear ramp of frequency in 64 steps. The motor is run at each of these speeds for a period that is 1/64 of the ramp time requested with the {B} command.

If long ramp periods are used with a high final speed, the steps are discernible to the ear but still achieve the desired effect of not accelerating the motor at excessive rates.

When using relatively slow final speed settings with short ramp times, the situation arises where the first few steps take longer than the allocated time.

For example, a motor accelerating to a final speed of 640 steps/sec will run at 10 steps/sec, 20, 30, etc. At 10 steps/sec, each step takes 100ms. If the ramp time setting is 2 (for 200ms), clearly a problem has arisen as the first step has used half of the ramp time for one step.

The Little Step-U addresses this situation by dividing the ramp time into 64ths. If the length of one step at the required speed is less than 1/64 of the ramptime, that step rate is skipped.

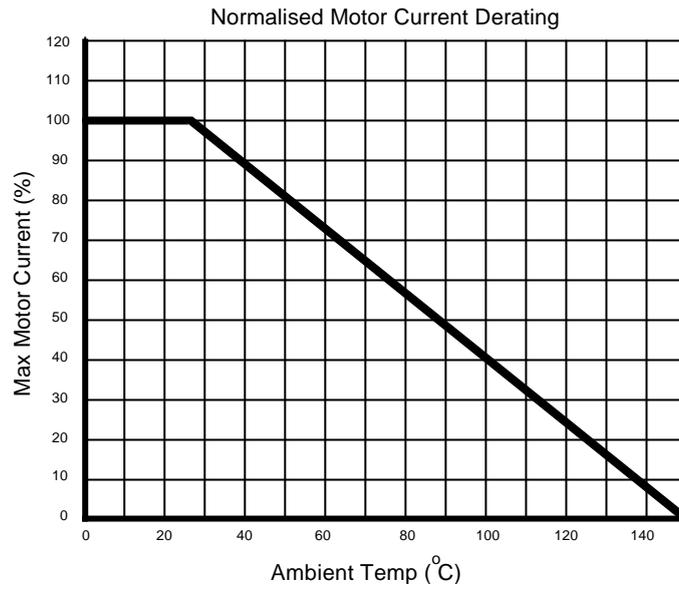
In the example above, 1/64 of the total ramp time (200ms) is 3.1ms which is the period of one step at 320 steps/sec, so the first step will be at the 320 steps/sec rate, the next at 330 steps/sec, etc. This means that the motor doesn't actually turn until halfway through the ramping period (320 = half of the final speed 640). Rather than have the motor doing nothing for the duration of this dead period, the delay is bypassed and the motor will appear to start immediately at 320 steps/sec and take only 100ms to get to the final speed, even though the B command specified 200ms.

The net result is that the motor acceleration in steps per second per second is still the same and this is the important parameter. Having it specified as a time to reach final speed is an easier concept to grasp and is also a simple figure to calculate from a required acceleration rate.

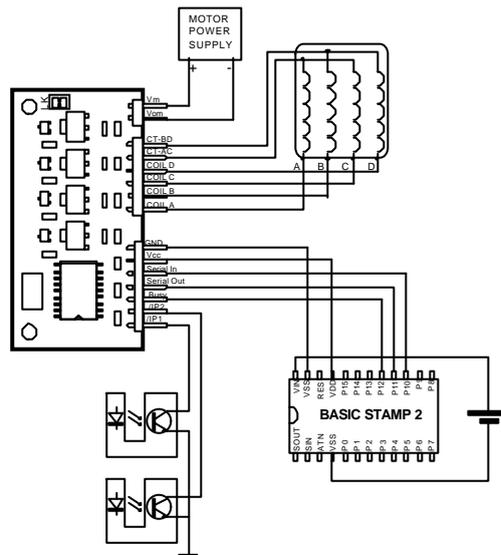
### ***Ramping and forced stops***

When the motor is forced to a stop from an external input (IP1 or IP2) and a ramp value other than zero has been specified, the motor will begin decelerating at that point. Obviously the final resting position will be different from that when the input occurred. Where the input is used to specify a position, the register is updated when the input occurs, not when the motor completes the ramp down.

For example, the {J} command runs the motor counter-clockwise until IP1 is pulled low and marks that position as zero. If the ramp value is 0, the motor will stop there and the position register will hold 0. If a ramp period has been specified, the position register will hold a negative value, indicating the number of steps used to decelerate to a stop. The {N} command (Go to the Home position) can then be used to move the motor to where the input occurred.



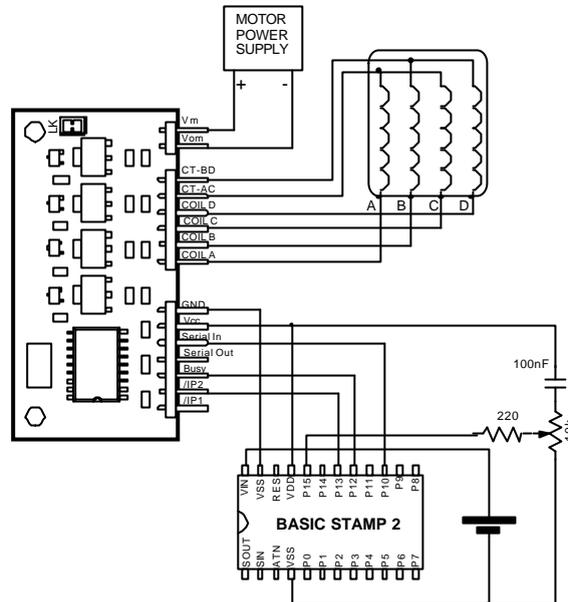
## Application example



The information in this document is copyright TLA Microsystems Ltd, 1999 and may not be reproduced in part or in full without permission. All information is believed to be correct and accurate however, TLA Microsystems Ltd assume no responsibility for inaccuracies or omissions. Specifications and ratings may be superseded without notice.

## AP-NOTE 1

Using a BASIC STAMP 2 to find the best speed setting for a motor



```

' Experiment to find best speed setting for a motor.
' The RCTIME function is used to read the setting of a variable resistor
' on P15. This value is used to set a new speed by multiplying it with
' FACTOR and the result sent to the PC screen so the user can see the
' value being tested. This SPEED value is sent to the Little Step-U
' and it is instructed to move 100 steps clockwise.
' The program pauses for 500ms and the sequence is repeated.
' The {G} command tells the Little Step-U to move 100 steps clockwise
' from the current position, monitoring both inputs and if one of them
' is pulled low, stop. The program begins by pulling IP2 low (P13) so
' if the motor is running at a slow speed, resetting the BASIC STAMP will
' also stop the current move.

' Little Step-U connected as follows
'   serial in p10
'   busy          p12
'   IP2           p13

factor   con      2
pot      var      word
speed    var      word

input    12
output   13

low 13
pause 1
high 13

loop:
if IN12 = 1 then loop

high 15
pause 1
rctime 15,1,pot
speed = pot * factor
range
debug ? speed

serout 10,396,["{A",dec speed, "}]
serout 10,396,["{G100}"]

pause 500
goto loop

```