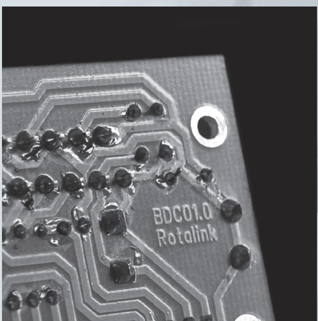
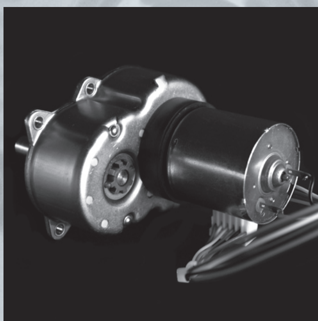
RED DRIVE

Programmable DC Motor Controller

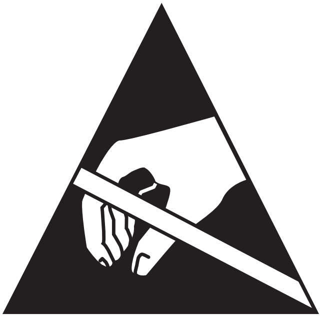
User Manual





Rotalink
Miniature Motors,Transmission and Control

www.rotalink.com



Protection Against Electrostatic Discharge

Electrostatic discharges (ESD) can harm the electronic components used within the Red Drive controller. Conditions can occur where an electrostatic charge will build up on the user or other object and then discharge into another object, such as the Red Drive controller. To prevent damage from these ESD events, you should discharge electricity from your body prior to contact with the Red Drive unit. You can protect against ESD by using a conductive wrist strap or other method of grounding. Please contact Rotalink for further information and advice.

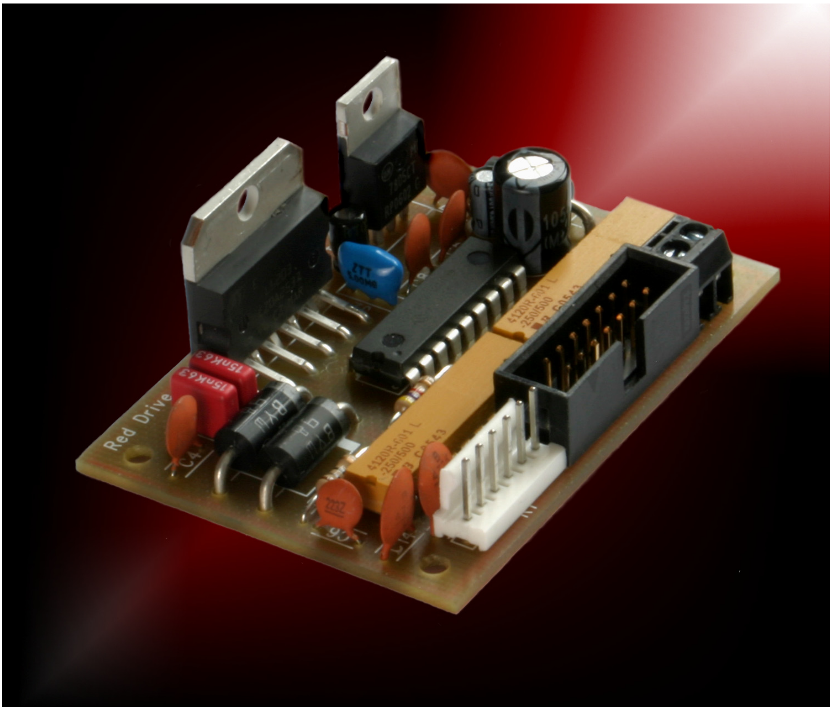
CE Marking

The Red Drive Controller is designed to be a component incorporated within equipment manufactured by our customers and is not sutiable for use by an ‘end user’. As such, it is not CE marked.
The method of installation for the Red Drive Controller and its associated motor will have a large impact on any CE test results, as will the selection of the motor. We will be pleased to advise our customers on ways to ensure their equipment complies with any relevant directives.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Introduction

The Red Drive Controller is a compact yet powerful motor driver and controller. In combination with the supplied software and Serial Adapter unit, it allows the user to specify how, and when their motor will move. Full control is available over speeds, distances (0 - 44 million output revolutions) and direction The unit is able to interact with switches, LEDs, buzzers and other user defined inputs and outputs. It is even possible to interact with other Red Drive controllers. The Red Drive software provides the user with an incredibly easy to use interface for defining all aspects of the controller. No complex programming is required, just a simple drag and drop flowchart style interface which uses real world values such as rpm and seconds.

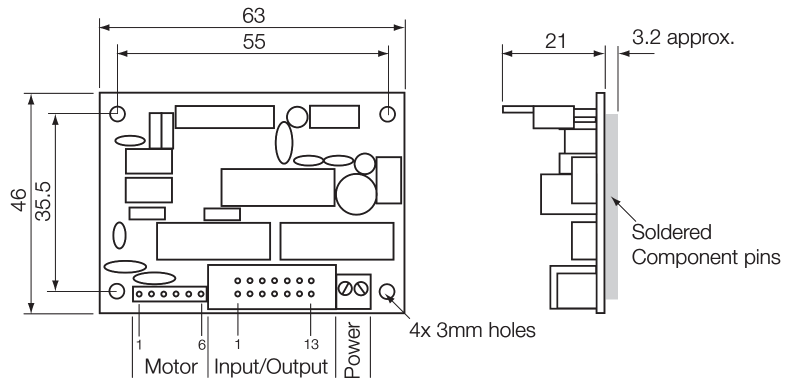


Contents

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Hardware

DC1A Controller

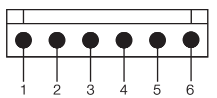


Continuous Current: 1 A **Voltage Range:** 9-30 V

Motors may draw larger currents for short periods of time, however the unit may cut the power supplied to the motor at higher currents to protect from overheating.

CAUTION: BOTH VERTICALLY MOUNTED COMPONENTS GENERATE HEAT IN OPERATION. THESE COMPONENTS MAY BECOME HOT.

Connecting Motors

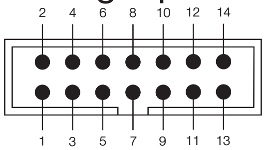


(Connector shown from above)

Pin 1: Connection to Motor
Pin 2: Connection to Motor
Pin 3: Encoder Ground (0 V)
Pin 4: Encoder A
Pin 5: Encoder B (unconnected for use with Single Channel encoder)
Pin 6: Encoder Supply (+5 V)

CAUTION: IF THESE CONNECTIONS ARE INCORRECT, IRREPARABLE DAMAGE MAY OCCUR TO THE ENCODER AND RED DRIVE CONTROLLER.

Connecting Inputs and Outputs



(Connector shown from above)

Red Drive features a standard 14 Way (7 x 2) IDC header. Two 5 Volt and two 0 Volt pins are provided alongside 9 user definable input or output pins with one input only pin. Pin 14 is also used as the 'home' input (see page 17). Pin 13 can be set as the global reset pin (see page 32). All Pins operate at 5 volt TTL levels. 5 Pins also support analogue inputs.

Pin Number	Input/Output	Analogue /Digital input	Pull-up Resistor?	Special feature
1	+5 Volts			
2	+5 Volts			
3	Both	Analogue + Digital		
4	Both	Digital		
5	Both	Digital	Yes	
6	Input Only	Digital		
7	0V GND			
8	0V GND			
9	Both	Digital	Yes	
10	Both	Analogue + Digital		
11	Both	Digital	Yes	
12	Both	Analogue + Digital		
13	Both	Analogue + Digital	Yes	Global Reset
14	Both	Analogue + Digital	Yes	Home Input

Note: Any pins not used should be set as outputs (via the Program Parameters window). Where this is not possible (i.e. Pin 6), connect to 0 V.

The 14 Way header is also used as the programming connector. For programming to take place the Red Drive SA1 serial adapter must be connected when the power supply is switched on. This instructs the Red Drive controller to enter Programming, rather than run mode.

Note: Pins 1-11 MUST NOT all be high (+5 V) when the unit is switched on. This will place the Red Drive controller into programming mode, during which downloaded flowcharts are NOT run.

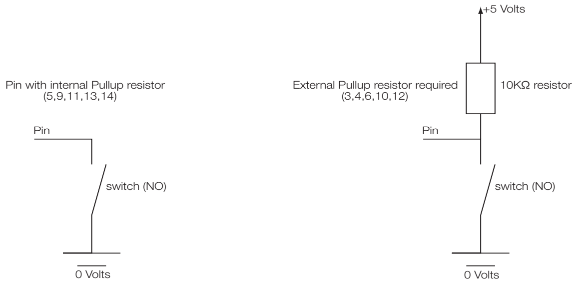
Outputs:

Maximum current sunk by all I/O pins: 200 mA
Maximum output current sunk by any one I/O pin: 25 mA
All outputs operate at +5 V, other voltages or current requirements will need to be managed with additional external circuitry such as an optocoupler.

CAUTION: FAILURE TO LIMIT THE MAXIMUM CURRENT OR VOLTAGE TO THE ABOVE LEVELS MAY RESULT IN IRREPARABLE DAMAGE.

Inputs:

Internal pull-up resistors make pins 5, 9, 11, 13 and 14, ideal for use with simple switches (identified in the Red Drive software with a ‘ * ’). In this configuration the switch is simply fitted between the pin and ground (0 V). Other inputs will require an external pull-up resistor (10K Ohm) to ensure the pin is held at either 0 V or +5 V. Both examples are shown below.



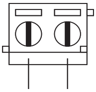
In the above configurations, a normally open switch (NO) will produce a LOW (0 V) when pressed. A normally closed switch (NC) will produce a HIGH (+5 V) when pressed.

CAUTION: ATTEMPTING TO USE A SWITCH WITHOUT EITHER AN INTERNAL OR EXTERNAL PULL-UP RESISTOR, MAY RESULT IN UNDESIRED OPERATION.

All inputs operate at 0-5 V, other input voltages will need to be managed with additional external circuitry, such as an optocoupler.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Connecting Power

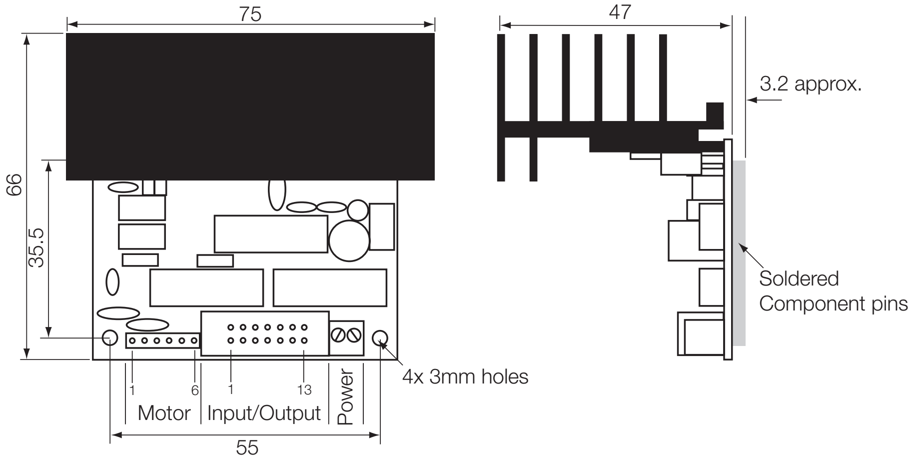


V_{SS}: Supply Ground (0 V DC)
V_{DD}: Supply Positive voltage (+9 to +30 V DC)

Timing

The DC1A PCB features a ceramic resonator to provide timing. This component has an accuracy of ±0.5%. Due to the nature of the microprocessor design further delays could effect the timing accuracy provided by ‘wait’ blocks (especially if the motor is moving). Taking the worst case scenario, if you were to set a wait for 24 hours (86400 seconds) you should only rely on an accuracy of ±14.4 minutes (864 seconds), that is ±1%. If more accurate timings are required, a external clock could be attached to the Input pins and used to trigger an event.

DC2A Controller



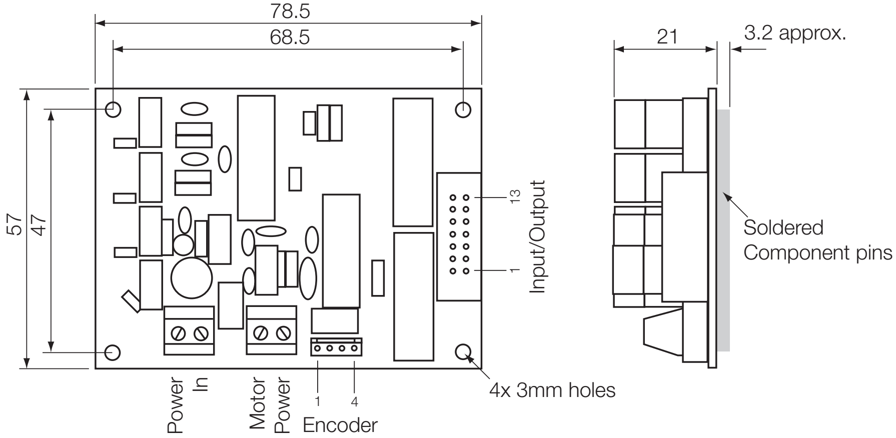
Continuous Current: 2 A

A heat sink is attached to the side of the DC2A. In all other ways this unit is functionally identical to the DC1A.

CAUTION: IN OPERATION THE HEAT SINK FITTED TO THE RED DRIVE DC2A MAY BECOME HOT.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

DC10A Controller

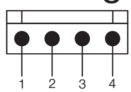


Continuous Current: 10 A Voltage Range: 15-30 V

Motors may draw larger currents for short periods of time, however extended use above 10A may result in damage to the Red Drive controller.

CAUTION: IN OPERATION THE COMPONENTS AND SURFACE OF THE DC10A MAY BECOME HOT.

Connecting Encoders

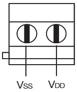


(Connector shown from above)

Pin 1: Encoder Ground (0 V)
Pin 2: Encoder A
Pin 3: Encoder B (unconnected for use with Single Channel encoder)
Pin 4: Encoder Supply (+5 V)

CAUTION: IF THESE CONNECTIONS ARE INCORRECT, IRREPARABLE DAMAGE MAY OCCUR TO THE ENCODER AND RED DRIVE CONTROLLER.

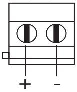
Connecting Power



V_{SS}: Supply Ground (0 V DC)
V_{DD}: Supply Positive voltage (+15 to +30 V DC)

CAUTION: DAMAGE WILL OCCUR TO THE RED DRIVE CONTROLLER IF THE POWER IS CONNECTED INCORRECTLY.

Connecting Motors



+ : Positive motor connection wire (usually red)
- : Negative motor connection wire (usually black)

Connecting Inputs and Outputs

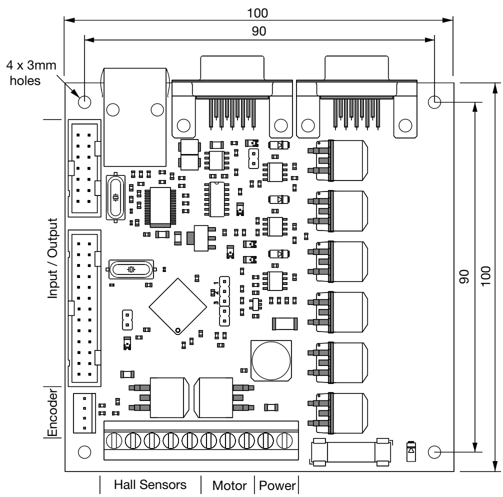
Please see DC1A information above (page 7).

Timing

Please see DC1A information above (page 9).

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

BLDC10A Controller

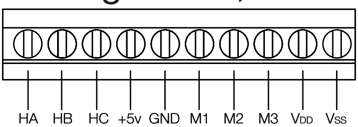


Continuous Current: 10 A Voltage Range: 12-30 V

Motors may draw larger currents for short periods of time, however extended use above 10A may result in damage to the Red Drive controller.

CAUTION: IN OPERATION THE COMPONENTS AND SURFACE OF THE BOARD MAY BECOME HOT.

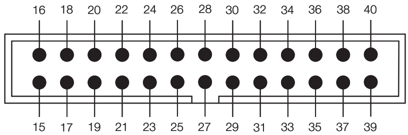
Connecting Power, Hall effect sensors and Motor



VSS:	Supply Ground (0V DC)
VDD:	Supply Positive voltage (+12 to +30 V DC)
M1:	Motor Phase 1
M2:	Motor Phase 2
M3:	Motor Phase 3
HA:	Hall effect A
HB:	Hall effect B
HC:	Hall effect C
+5v:	5 volt supply for Hall effects
GND:	Ground (0V DC) for Hall effects

Connecting Inputs and Outputs

The I/O function in the same way as on the DC1A (see page 7). The BLDC10A offers an extra 26 way (13x2) IDC header and pins 15-40 can be found here. Functionally of the full 40 pins are listed below:



(Connector show

Pin Number	Input/Output	Analogue /Digital input	Pull-up Resistor?	Special feature
1,2,15,16	+5 Volts			
7,8,23,24,28, 31,32,39,40	0V GND			
3	Both	Analogue + Digital		
4	Both	Analogue + Digital		
5	Both	Analogue + Digital		
6	Both	Analogue + Digital		
9	Both	Analogue + Digital		
10	Both	Analogue + Digital		
11	Both	Analogue + Digital		
12	Both	Analogue + Digital	Yes	
13	Both	Digital	Yes	Global Reset
14	Both	Digital	Yes	Home Input
17	Both	Digital		
18	Both	Digital		
19	Both	Digital		
20	Both	Digital		
21	Both	Digital		
22	Both	Digital		
25	Both	Digital		
26	Both	Digital	Yes	
27	Both	Digital	Yes	
29	Both	Digital		
30	Both	Digital		
33	Both	Digital		
34	Both	Digital	Yes	
35	Both	Digital	Yes	
36	Both	Digital	Yes	
37	Both	Digital		
38	Both	Digital		

Connecting Encoders

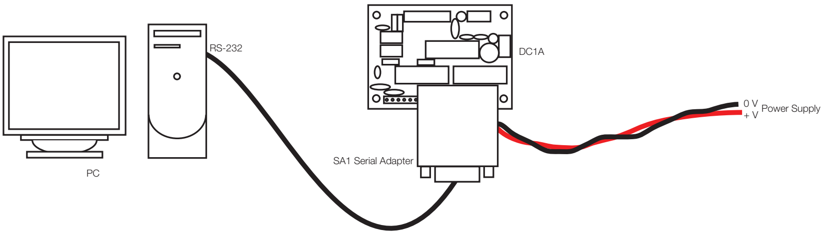
Please see DC10A information above (page 11).

Timing

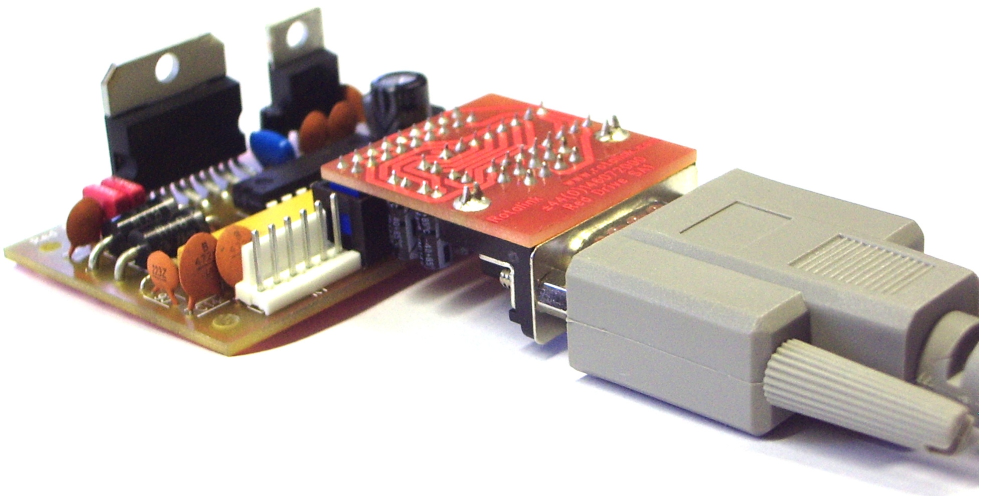
The BLDC10A makes use of an accurate crystal to control it's timing rather than the resonator featured in the DC1A controller. The crystal has an accuracy greater than 0.00005%. As explained in the DC1A Timing section (page 9) there are several other factors which will affect the accuracy of Red Drive, particularly when the motor is running and tests should be taken to confirm suitability for a given project.

Connection to PC

Flowchart programs can be written without the Red Drive controller being connected to a PC. This is useful for early design layout and for adjusting sequences. Care should be taken when working this way as it becomes easy to make errors which could then cause damage to the users product. For example an incorrect distance calculation could cause an overshoot, resulting in a linear system crashing into an end stop. It is always best to build up sequences slowly, testing every change that is made. This does require the unit to be connected to a PC.



The serial adapter must be connected to the Red Drive controller before switching the power on. Otherwise the unit will not enter programming mode.



Software

Installation

The Red Drive software can be found on the supplied CD. It does not require installation (first run may require administrator rights) and can be run direct from this disk. If required, the Red Drive program can be installed onto a PC by copying the file RedDrive.exe to the desired location.
This software can also be downloaded from the Rotalink website at www.rotalink.com.

Concept

The Red Drive software has been designed to be as easy to use as possible. The system is designed around a flowchart concept with blocks for moves, waits and other commands. Blocks can be easily moved, added and deleted with just clicks and drags. Any values needed are in easy to understand revolutions per minute (of the gearbox output shaft) and number of revolutions. There's no need to continually calculate gear ratios, and times are all in seconds. Programs can be saved or printed at any stage for later completion or review.

Calibration

It is recommended that all systems feature a calibration point (home position) on their final movement output. For a lead screw this would often be a micro-switch at one extreme of travel, for a rotary system we recommend an optical sensor and slotted disk or pin on the last rotary device (this may be the output shaft of the gearbox, or perhaps a secondary pulley). It is necessary on all systems where absolute positioning (E.g. 100 mm from the gear case, 5 output revolutions from horizontal) is required, for the unit to move to a known point on start up. This ensures that any interrupted programs (power cuts etc.) will always start from the correct position. Rotary systems which only require relative positioning (e.g. rotating 1 more revolution) may not require this calibration point. If there is any doubt, Rotalink will be happy to help. If a calibration point is present it is good practice to calibrate the 'home' position as often as possible. This ensures the most accurate positioning is always available. The 'Move Home' block requires that this calibration point is fitted to Pin 14 in the same way as a standard switch (see page 8).

Positioning System

While Red Drive can be used in either a relative or absolute positioning system, all distances used by Red Drive are in the form of absolute positions. That is, every distance to move to is quoted as a distance from ‘home’. All clockwise movements are in a positive direction, and counter clockwise movements are in a negative direction. For example, moving 2 revs CW, a further 8 revs CW and then 10 revs CCW would be written:

Move CW at x rpm to distance 2 revs
Move CW at x rpm to distance 10 revs
Move CCW at x rpm to distance 0 revs

At power on, the home position will default to the customer set value in the Home section of the Program Parameters window (see page 22). Negative numbers are not allowed, to move counter clockwise beyond the home position requires the home position to be set as a positive number in the Program Parameters window (see Program Parameters, page 22). If the first move is to be CCW, the home position should be set equal to the largest total movement, for example:

Home position set to 10 revs (in Program Parameters window)
Move CCW at x rpm to distance 8 revs
Move CCW at x rpm to distance 0 revs
Move CW at x rpm to distance 10 revs

Relative positioning (move another 2 revolutions) is possible by using the Reset Home block (described on page 27).

CAUTION: FAILURE TO ENSURE ALL DISTANCES GIVEN ARE RELATIVE TO THE HOME POSITION CAN RESULT IN UNFORESEEN AND POTENTIALLY DAMAGING MOTION.

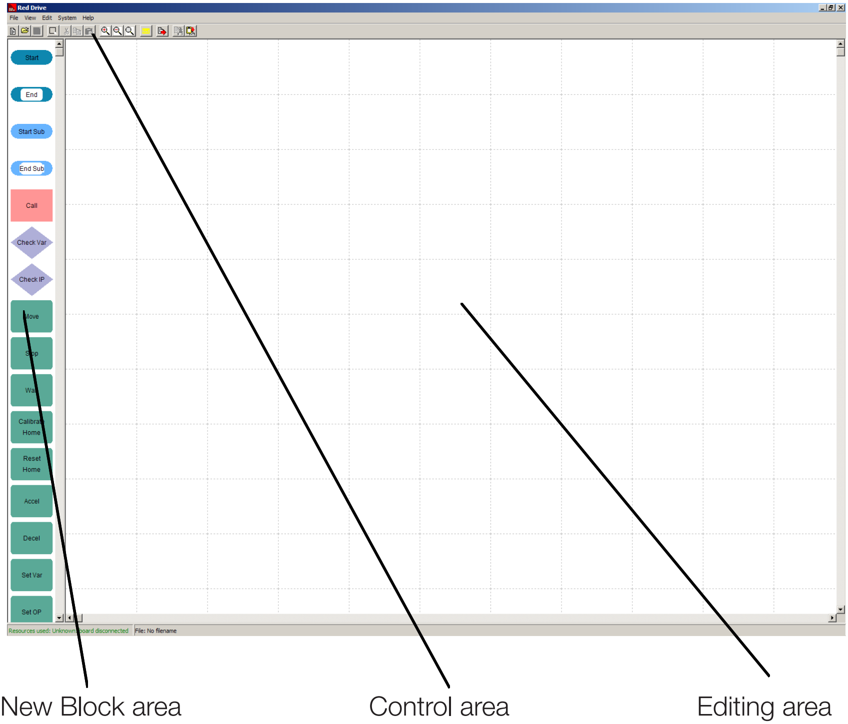
Selecting an Encoder

When using Red Drive with Brushed motors, two encoder are available, Single and Dual Channel. Dual channel encoders provide information on direction, distance and speed; single channel encoders only provide distance and speed information. The lack of direction information results in a small loss of position for Single Channel systems. Brushless motors can use a Dual Channel encoder, or if a low accuracy is acceptable, their inbuilt hall effect sensors. Red Drive has been designed for encoders with an output of 48 pulses per rev. Fewer pulses will result in a lower accuracy, both in speed and position.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Running the Software

To run the software, double click on the Red Drive icon . The following screen will appear:



Note: On some machines a COM port error may occur. This can be ignored until programming, and is dealt with in the 'Programming the Hardware' section of this manual.

There are three areas to the Red Drive window. The 'new block' area, 'control' area and the 'editing' area.

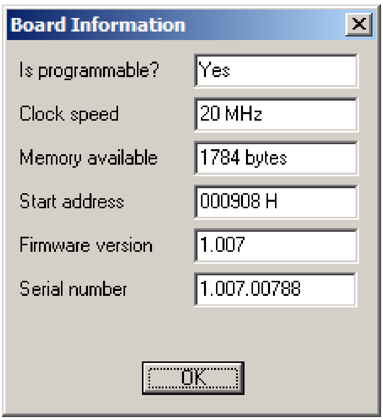
New Block Area:

Down the left-hand side a pallet holds all the various blocks from 'Start' to 'Set OP'. These are placed in the editing area by clicking and dragging as required.

Control Area:
This area controls the entire flowchart, files can be saved and loaded, the flowchart zoom level can be changed, blocks can be copied and pasted, Red Drive parameters are set and the flowchart downloaded to the controller.
Editing Area:
This is where the program flowchart is built.

Programming the Hardware

The first step to downloading a flowchart program to the Red Drive controller is to connect it to a PC. This is done via the Serial Port (also known as an RS-232 port). A suitable cable must be plugged into this port and into the Serial Adapter board (see page 16). This board must then be connected to the Red Drive unit via its 14 Pin I/O connector (see the Hardware section of this manual for the location of this connector). If the PC features more than one serial port or is using a USB serial adapter it may be necessary to configure the correct serial port. Under the System heading in the Control area the menu 'Set Com Port' can be selected. This will produce a window with a drop down menu indicating all available Com Ports. After selecting the relevant Com port (it is safe to use trial and error), clicking OK confirms the change.
Only after this has been completed can power be applied to the Red Drive Unit via its power connector.
Clicking the Connect button  will cause the Red Drive software to connect to the unit and report its status.



Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

If the Board Information window does not appear, please check the power to the controller and the connection between the controller and PC. An answer of ‘No’ to the board being programmable may indicate that you require a later version of either the Red Drive software or controller firmware. Upgrades will be made available from the rotalink website (www.rotalink.com)

The Compile button  should now be usable. This will ready the flowchart for download to the Red Drive controller. If there are any errors (see ‘Compiler Errors’ page 59), a window will appear in the lower section of the screen. If this window is open (a cross in the top left can be used to close the window) successful compiles will be indicated here rather than in a pop-up message.

If the compile was successful, the Download  button can be used to send the program to the Red Drive Controller.

A window will appear indicating if the download was successful. If so, switching the power off, removing the serial adapter and then connecting the motor leaves the system ready to start (for further programming the motor may remain connected). Switching power on to the Red Drive controller will begin the downloaded flowchart program.

Note: The power to the Red Drive controller must be switched off, and then on again before any flowchart programs will run. Without cycling the power supply, the unit will remain in programming mode.

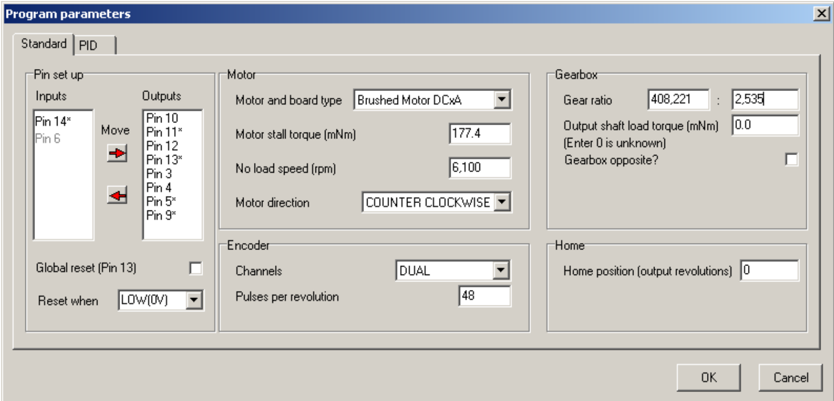
CAUTION: SWITCHING ON THE POWER TO THE RED DRIVE CONTROLLER WITHOUT A SA1 SERIAL ADAPTER WILL RESULT IN RED DRIVE STARTING THE FLOWCHART PROGRAM. IF THE FIRST BLOCK IS A MOVE, THE UNIT WILL IMMEDIATELY START MOVING.

Fast Communications

Under the same menu as ‘Set Com Port’ (see ‘Programming the Hardware’ above) is a ‘Fast Communications’ option. Selecting this causes the PC to download flowcharts to the Red Drive controller at a faster speed. Unfortunately this option does not function successfully with all PCs, if you experience communication problems, please deselect this option.

Program Parameters

Selecting the 'System' menu in the Control area and then 'Program parameters' results in the following window being displayed.



This is used to define any parameters that have an effect on the entire program. These only need to be setup once for each program and are saved along with the rest of the flowchart information.

Pin set up:

This section sets the I/O Pins as either inputs or outputs. Unused pins should be set to outputs. Select the required pin and use the arrows to move their definition. An '*' next to the pin number indicates an internal pull up.

Note: **Pin 14 must be set to input if the 'Calibrate Home' block is to be used.**

Global reset (Pin 13):

Should be set to use the global reset function. 'Reset when' indicates if there should be a reset on a HIGH(+5V), or LOW(0V) (see page 32).

Motor:

Motor and board type:

Select between Brushless motor (BLDCxA) and Brushed motor (DCxA).

Motor stall torque (mNm):

This is available from the Rotalink motor data sheet and has a value in mNm. Take care to enter the value for the correct Red Drive supply voltage.

No load speed (rpm):

Also available from the Rotalink motor data sheet.

Motor direction:
The normal direction of motor rotation, shown on the Rotalink data sheet. See also the 'Working out Motor/Gearbox direction' section (page 58).

Encoder:

Channels:
Single, Dual or Hall effect depending upon encoder and board type.

Number of pulses:
Available from the Rotalink encoder/motor data sheet, 48 pulses is normal for our dual channel encoder.

Gearbox:

Gear Ratio:
The ratio is entered in its 150:3 form.

Note: Gear ratios must be entered in their full form. A gear ratio of 472:3 could be described as 160:1. However entering this approximate ratio would result in incorrect output distance and speed.

Output shaft load torque (mNm):
This value is not critical, however a value closer to the actual load will produce a more accurate and smoother running system.

Note: If Speed feedback is switched off then this feature is important to ensure actual speed is as close to the required speed as possible.

Gearbox opposite?:
Tick if the gearbox shaft rotates in the opposite direction to that of the motor. This information is available on the gearbox label ('SA' = same, 'OP' = opposite). See 'Working out Motor/Gearbox direction' (page 58).

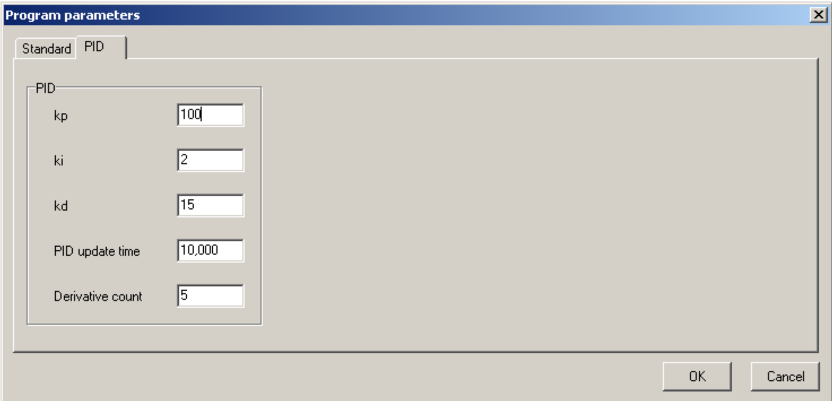
Home:

Home position (output revolutions):
Normally left at 0 output revolutions, alternative values may be required depending on position of the 'home' (calibration) input. Further information can be found in the 'Positioning System' section (page 18).

CAUTION: PROGRAM PARAMETERS MUST BE SETUP BEFORE PROGRAMMING. ANY INCORRECT PARAMETERS MAY LEAD TO UNFORESEEN AND POTENTIALLY DAMAGING MOTION.

PID

PID (proportional, integral, derivative) is a mathematical system used by Red Drive to accurately maintain the desired speed. The proportional term looks at the current error, the integral term at past errors and the derivative term forecasts future errors. These 3 terms combine to produce a quick reacting, accurate system. In most cases the default settings will be adequate, however as PID is a compromise between response time and smoothness, it can be tuned to better suit particular applications. The Program parameters 'PID' tab provides control over this function.



kp:
Sets up how much of an effect the proportional term has in the PID calculation.

ki:
Sets up the integral term in the PID calculation.

kd:
Sets up the derivative term in the PID calculation.

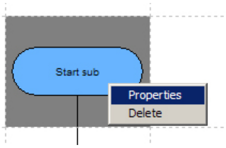
PID update time:
How often (10,000 = every 1mS ; 1,000 = every 100µS) the PID system is run. Decreasing this value will improve the response time of the system, at the expense of smoothness.

Derivative count (DCxA):
How often the derivative element is calculated. In the above window the derivative element is calculated once every ten PI calculations.

Pulses for speed timing (BLDCxA):
How many pulses the speed measurement is taken over (more pulses gives smoother but slower reacting output).

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Flowchart Blocks

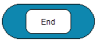


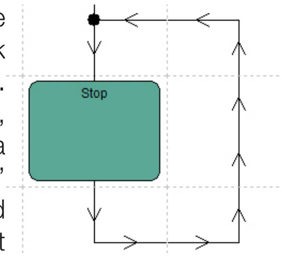
Left clicking (and holding) over the block in the new block area allows it to be dragged into the editing area (release the left button to place). When the block is placed in the editing area, right clicking opens a short menu from where the block can be deleted or the properties accessed. (Double clicking on the block also displays the properties window).

Start:

This is required to indicate the starting position of the program, the block connected beneath this is the first command to be run.

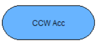
End:





This indicates the end position of the program. Upon reaching the End block the unit will return to run from the Start. An end block is not always required, as the flowchart can also end with a loop. For example ending with a 'Stop' can be achieved by looping around a 'Stop' block as shown. A flowchart may feature more than one end block

Start Sub:

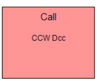


Indicates the start of a subroutine. Each subroutine must be given a name, this can be set by right clicking on the block and clicking on Properties. Enter the name for the subroutine into the lower box. Subroutines are useful in reducing the amount of memory used by the program, and also to reduce design time by allowing blocks to be reused (see page 31).

End Sub:

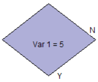
From here the Red Drive will return to run the block following the Call which referenced the subroutine.

Call:



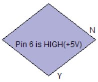
If a subroutine has been created this block is used to run the blocks within the 'Start Sub' 'End Sub'. In the Properties window is a drop down menu from which the relevant subroutine can be selected. Calls can be used within subroutines, but this is limited to a chain of 6 calls (see page 62).

Check Var:



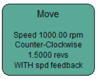
With this block a variable can be compared to either a set value, or to a second variable. Either a 'Yes' or a 'No' path can be taken. The variable can be checked for being either equal (=), less than(<) or greater than(>) the given value. As with all blocks these options are defined in the Properties window accessed by right clicking on the selected block. Red Drive supports the use of up to 128 individual variables, each one of which can hold a value from 0 to 255. Variables are identified by their 'variable number'. Variable 1 is set aside for error logging (see page 32).

Check IP:



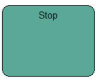
As with the 'Check Var' block, this provides two paths that can be taken. Any of the 10 I/O pins that have been configured as inputs can be checked, and the user is free to define either following 'Y' if the pin is 'LOW' (0 Volts) or if the pin is 'HIGH' (+5 V)

Move:



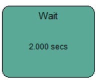
Using the properties window this block gives the user the ability to move at a set speed for a set distance. Stopping the motor will also require the use of a 'stop' block (see also 'Positional Accuracy', page 30). If 'use distance' is unchecked the unit will set the speed and then move directly onto the next block. This is useful when followed by a 'wait' block to provide motion for a set time.

Stop:



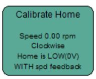
This causes the motor to stop moving.

Wait:



In Properties a time to wait is set in seconds. If the block previous to this left the motor rotating, then the motor will continue rotating during this wait. A Stop block is required prior to a wait to provide a timed pause in motion.

Calibrate Home:

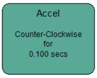


Motor will follow the user defined speed and direction until the Red Drive controller receives an input on the 'home pin' (Pin 14). At this point the next block will be run, often a 'stop'. This command is used to ensure the correct positioning of the Red Drive output after start up and during operation. Good practice would dictate that a 'Calibrate Home' block should be the first movement block following the 'Start'.

Reset Home:

This block resets the position counter. This is particularly useful when the Red Drive controller is being used in a relative positioning system. Issuing this command will reset the position counter allowing a basic 'move 1 rev' block to be repeated, causing the output to move in 1 rev steps.

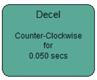
Accel:



Red Drive employs a fairly basic straight line approximation acceleration and deceleration. Using the Properties window the start speed, end speed, time and final position can be defined. Final position (selected via the drop down menu) only ensures the final position accuracy, it does not control the position during the Acceleration. Using this function, after the acceleration time has been run, the unit will continue to move at the final speed until the final position is met, only then will Red Drive move on to the next block. See page 30 for further description.

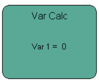
Note: **Acceleration blocks control speed not position. If an acceleration is attempted which the motor cannot match, position control will be reduced.**

Decel:



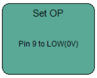
As per acceleration, but for deceleration.

Var Calc:



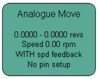
Variable calculations provide for a greater program complexity where required. Generally used for program loops (carrying out a sequence x number of times) they can also be used to process an analogue input or for a range of other uses. Addition, subtraction, multiplication and setting (either to an integer or to a second variable) are all catered for. Variables 'roll over' when they reach 256 (65536 for BLDCxA). If variable 5 is already equal to 250 and you add 100, variable 5 will become equal to 94. Variable 1 is used for error logging (see page 32) therefore the value of this variable on start-up is unknown. Variables 2-64 can be saved into EEPROM on each calculation. If the power is interrupted these variables will be reloaded. This is useful to resume sequences after a power failure. Variables 65-128 can not be saved, at power on these variables are set to 0.

Set OP:



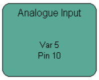
Allows the user to set an output pin to either 'HIGH' (+5 V) or 'LOW' (0 V).

Analogue Move:



Provides a simple follower system. Required speed and distance limits are entered. Analogue voltage is taken from selected pin and motor moves between the distance limits. For example voltage set to 2.5V (50% on a standard potentiometer), motor will move in either direction to a position halfway between the min and max positions. A standard setup would see the two outside leads of a potentiometer connected to 0V and 5V, with the 'wiper' (usually the middle lead) connected to a suitable Analogue Input pin (see page 7 or 14).

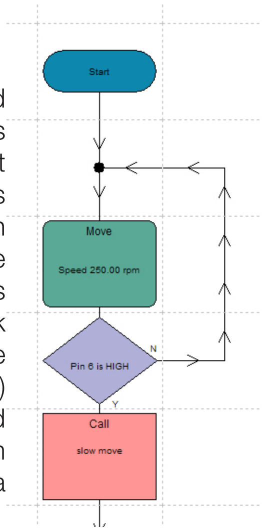
Analogue Input:



Looks at the voltage on the selected pin, loads a value in the range 0-255 into the selected variable. i.e. 3V = 153. (BLDCxA uses an extended range 0-1023). Comparisons can then be made on this value using the Var Calc block (described above). The result can be saved to EEPROM by using the Var Calc Block.

Analogue inputs are very subseptable to noise, and care should to taken when using analogue inputs, it may be nessasary to program averaging or unexpected value disregard fuctions.

Linking Blocks



Directional links between the blocks are required to control which blocks activate when. The blocks can only be linked from top and bottom (except decisions). As is common with other flowcharts these are represented by drawing lines between the relevant blocks. To do this, move the mouse cursor over the block to link from and when a cross hair (+) appears, click and drag into the next block and release. Loops can be formed with the use of Check Var/Check IP blocks. A node (black dot) is drawn when required, indicating the connected loop. Paths can also cross without the program flow joining. This is shown by the absence of a node.

Speed Feedback

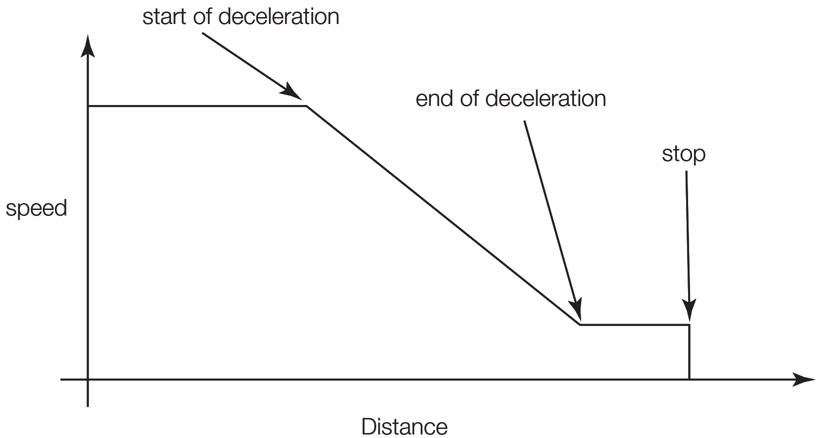
By ticking ‘Speed Feedback’, the Red Drive controller will constantly monitor the speed of the motor and (using the PID system) automatically adjust the power to ensure the correct operating speed. To reduce any initial acceleration (caused by the actual speed being less than the required speed) it is important that the ‘output shaft load torque’ is entered in the ‘program parameters’ window. Entering 0 will still produce a working system, however this value is used to calculate the initial starting speed. A value closer to the actual load will reduce any initial acceleration, resulting in a smoother system. It should also be noted that the faster the motor is moving, the steadier the output speed. This is the result of more encoder pulses per second giving rise to more opportunities to measure and to correct the speed.
Switching ‘Speed Feedback’ on also has the benefit of providing an extra fault detection mode (See fail safe page 32). If the unit is blocked for any reason, the Controller will detect the lack of movement and restart. This helps protects the motor from overheating and the possibility of damage to the users machine. This feature will not protect the gearbox from breaking under shock loads.

EEPROM

The Red Drive controller is able to save 64 Variables into EEPROM memory. This special area of memory is retained when the power is switched off and is useful for monitoring product usage over time, and for keeping track of variables which need to be monitored despite power cycling.
The EEPROM doesn’t last forever, a limited number of writes can be performed, Red Drive employs a sharing technique across the EEPROM memory to improve this life.
Red Drive makes use of 128 EEPROM ‘cells’, each cell has a life of a minimum 100K writes, due to the cell sharing this results in a minimum life of up to 12 million writes. The worst case is 64 variables being written to EEPROM equally resulting in a minimum write life of 200K writes. By looking at how many variables are being saved to EEPROM it should be possible to predict the minimum EEPROM life expected in a given application.

Positional accuracy

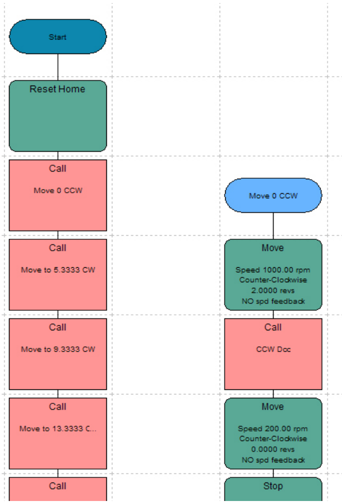
Whilst the Red Drive controller has a high level of accuracy, it does require use of good programming techniques to maintain this. As an example, driving the motor at full speed and expecting it to stop exactly where required is simply unobtainable, the systems inertia will act to cause the unit to overshoot. This can be improved upon by stopping the motor before the final rest position, and as long as each system has the same inertia, a consistent stop will be achieved (however it should be noted that as a DC motor heats up its breaking potential changes). System inertia is however unlikely to be consistent, and where possible this should be overcome by driving the motor to its rest position. To do so accurately requires the motor to be driven slowly, and deceleration blocks should be used to achieve this slower speed before requesting the motor to stop. The diagram below shows this in more detail.



The distance between the start of the deceleration and the end of the deceleration (in time) should be selected to complement the system being controlled. Positional overshoot will occur if this time is set too short. The distance between the end of deceleration and the stop is again selected to suit the system being controlled. This gap is to allow for inertial differences in manufactured systems. The lower the speed at the end of the deceleration, the more accurate the final rest position of the motor will be. The deceleration block has a ‘Type - Final Position’ feature that provides this method of control in a single block.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Making use of Calls



Calls are best used to save memory with often repeated sequences of blocks. In order to use a call, the subroutine to be called must first be setup.
The blocks to be called should be placed between a ‘Start Sub’ and ‘End Sub’ block. The block Properties menu should then be used to give the subroutine a name. The ‘Call’ block will be placed in the main flowchart, and using the block Properties of the Call, the subroutine name can be chosen from the drop down menu. The illustration shows the subroutine ‘Move 0 CCW’ being called by the first ‘call’ block in the main flowchart.
It is worth noting that due to the memory used to process a ‘call’ you should not package any single blocks (except acceleration blocks). Doing so will result in more memory being used.

Interacting with other Red Drive units

While sharing data with other Red Drive units is not currently available, using the I/O system it is possible to have units communicating with each other. For example, one Red Drive unit cannot tell another where it needs to move (data cannot be sent), but it can tell it to move (a pulse can be sent which must then be interpreted by the second Red Drive controller). At any given time there are likely to be only a small subset of possible movements. Clever use of the various I/O pins and design of the flowchart programs can allow a large amount of interaction to take place.
Care needs to be taken with electrical noise when connecting 2 units together, for distances greater than 30cms differential signals may reduce ‘false signals’. At greater distances in noisy environiments an isolated signal may be required to stop damage to the Red Drive unit.

CAUTION: LINKING RED DRIVES OVER LONG DISTANCES WITHOUT ADITIONAL PROTECTION COULD LEAD TO UNEXPECTED AND PERMAMENTLY DAMAGING EVENTS.

Fail Safe

The Red Drive controller features two types of fail safe. Internal computation (the controller decides an error has occurred) and also a external global fail safe (An external system, alerts Red Drive to an error). Both of these cause the Controller to reset, with operation commencing from the programmed flow charts ‘Start’ block. An error code will be logged in Variable 1 which can then be used to provide different start-up options (i.e. could just restart, wait for the user to press a switch before restarting or light a maintenance LED).

Error codes logged in Variable 1

Var1 value	Description of error
1	Global (external) Reset (See Below)
2	Position more than 44 million revs (Position counter overrun)
3	Position less than -20 revs (Position counter under run)
4	The controller cannot turn the motor (if speed feedback on)
5	Flowchart call stack overflow (see page 62)
6	Power failed (switched off?) / High current draw
7	EEPROM failure
249-255	Red Drive internal error - Please contact Rotalink

CAUTION: THESE RESETS WILL ONLY PROTECT THE MOTOR AND GEARBOX TO A LIMITED EXTENT. IF THE GEARBOX DRIVES AT HIGH SPEED INTO A BLOCKAGE, THE RED DRIVE CONTROLLER OFFERS NO PROTECTION.

Global (external) reset

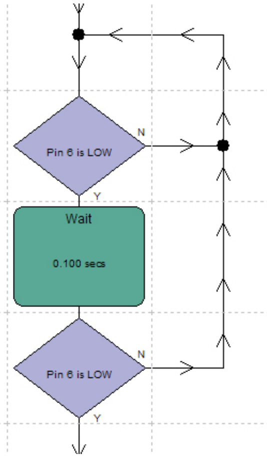
The Red Drive controller features a user definable global fail safe input on Pin 13. This is set up in the Program Parameters window (see page 22) and can be disabled if not required (Pin 13 can then be used as a standard I/O pin). The reset condition can be set to either HIGH (+5V) or LOW (0V). When the required reset voltage is placed on pin 13 the system will freeze. This reset condition will remain until the voltage is removed from Pin 13 at which point operation commences from the ‘Start’ block

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Ensuring Inputs only switch when required

‘Bouncing’ switches

Switches are rarely pressed cleanly. It’s common for a switch to make a connection several times when the user only wanted it pressed once. Sometimes this is due to the user not pressing cleanly and often the contacts of the switch actually bounce. To avoid counting switch inputs more than is required, it is good practice to check a switch several times before accepting the result of a Check IP. As shown, the switch is checked, if it’s been pressed there’s a short wait before it’s checked again. Only if the switch is still pressed then the result accepted. 0.1 seconds has been proven to work well for this, however the user may wish to experiment with alternative values.



Electrical Noise

There will always be electrical noise in our environment, from mobile phones to office equipment. Usually the level of this noise is small, however it’s always best to reduce the effect any noise may have with good design.
The most important rule is to keep any wiring short. The shorter any wiring looms, the less area an input signal has to be effected by noise. The next step is to design the flowchart program to ignore noise. When bouncing switches were mentioned above, it was suggested that a wait block should be placed between two Check IP blocks. This is important on other inputs as well. The length of the wait in this case should be less than the shortest expected input pulse. For example if a piece of equipment was providing a 5ms pulse to instruct Red Drive to start, the wait could be 4ms long. Using this method most short noise spikes will be ignored by the Red Drive controller.

Saving & Loading

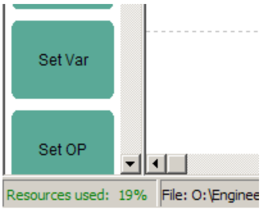
Programs can be saved and loaded at any time using either the ‘File’ menu in the Control area, or the save/load icons also in the control area. The current file name and location is always displayed at the bottom of the main window under the editing area.

Show block hints

A useful feature found under the ‘view’ menu (control area) is ‘show block hints’. If you hover over a block, the block information will be displayed in a pop up at the system font size. This is particularly useful when using the software with a zoom setting which makes the text within a block difficult to read.

Program Size

There is a limit to the size of program the Red Drive hardware can contain. After a Controller has been connected to the PC and its setup information downloaded (see ‘Programming the hardware’), the software will keep a running total of the resources used as the flowchart is created. This information is displayed in the bottom left hand corner of the main window, just below the New block area.



This text remains green when the flowchart is well within the memory of the Red Drive unit, yellow when approaching the limit and red if the memory resources of the Red Drive controller are exceeded. Reviewing the flowchart program for repeated block combinations and replacing them with Call blocks and subroutines can often dramatically reduce the resources used. It would take a very large and complex program to fill the memory of the Red Drive, however Red Drive variants with expanded memory capabilities may be available. Contact Rotalink if this is an issue.

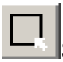
Zoom and scroll functions

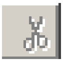
As well as using the zoom buttons in the Control Area and the standard scroll bars, if the PC is fitted with a scroll wheel mouse, this can also be used to control zooming and scrolling of the Editing area.

- Moving the scroll wheel backwards and forwards zooms in and out.
- Pressing the ‘Alt’ key and scrolling moves the Editing area left/right.
- Pressing the ‘Ctrl’ or ‘Shift’ keys and scrolling, moves the Editing area up/down.

Cut / Copy & Paste

The Red Drive software allows the user to copy (or cut) large sections of flowchart. The Control area of the main window features the following four icons:

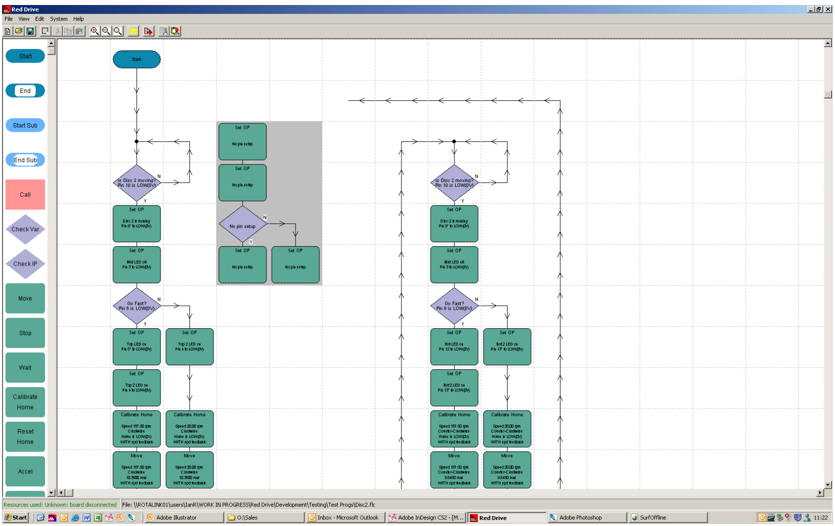

Select


Cut


Copy

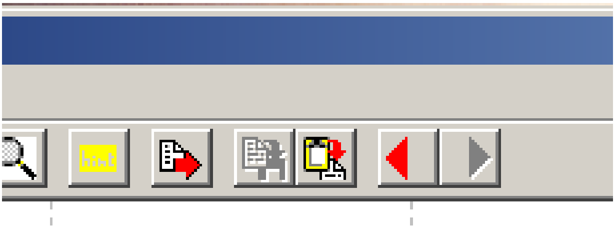

Paste

Clicking on the select icon allows the user to click and drag a grey area over a group of blocks in the editing area. This selection can then be Cut or Copied (using the above icons) as required. Clicking on the Paste icon then allows the user to move the new block(s) around the editing area to the required space, clicking with the left button then places the selection. A small no entry sign for the cursor indicates that the blocks can not be placed, this usually indicates a block or line beneath the selection which will need to be moved before the paste can take place.
Cut /Copy & Paste can be done between flowcharts. Simply make the selection and Cut/Copy, then open the new flowchart and use the Paste icon here.



Undo / Redo

The Red Drive software features a simple undo/redo function. The undo/redo buffer is constantly updated and holds a record of the last 100 actions. To undo simply press the left had arrow, to redo, press the right hand arrow. These arrows appear red when an undo/redo action is possible.



Printing

Printing is available under the File menu in the control area. Using the Single page / Multiple pages options and the Page size selector bar, flowcharts can be printed at almost any scale.
The Settings tab allow the page borders to be modified as well as providing dialogue boxes for Header and Footer information. The following control codes can be added into the Header and Footer dialogue:

- '~n' Prints the page number
- '~N' Prints the total number of pages to be printed
- '~f' Prints the file name without path
- '~F' Prints the file name with full path
- '~d' Prints the current date
- '~D' Prints the date of the last flowchart modification
- '~t' Prints the current time
- '~T' Prints the time of the last flowchart modification
- '~' followed by any other character causes the word 'Error' to be inserted

Upgrading the Controller

If any future upgrades are made to the Red Drive system, these may require changes to the firmware stored within the Controller. These upgrades will be distributed via our website (www.rotalink.com) and can be installed by using the 'System' menu (found in the Control area) and then clicking on 'Firmware Download'.

Tutorials

All of the following tutorials will be based around our standard SP2902 unit running on a DC1A controller. The SP2902 comprises a D3857-2490 motor, dual channel encoder and 230 gearbox. This unit is designed to be run from 24 Volts. These tutorials are designed to be followed sequentially, as each builds upon the program created in the last. The flowcharts are saved on the Red Drive CD, Tutorial1.flc for example is the flowchart produced when Tutorial 1 is completed.

Note: Care must be taken when following these tutorials with a different motor, gearbox or supply voltage. It is important to ensure that the correct Program Parameters data is used. Output speeds may also need to be altered for the tutorials to work as intended.

Tutorial 1 - Start, End, Move, Wait, Stop

This tutorial will introduce the initial setup of a motor and gearbox system using the program parameters window. The Start, End, Move, Wait and Stop blocks will be used and the distance concept will be explored. The aim of this tutorial is to have the gearbox turning clockwise one revolution, stopping and waiting for 1 second, and then moving counter clockwise back to the start position, stopping and waiting for 1 second. This sequence will then be repeated.

Program Parameters

The first step to producing any flowchart program is always to setup the constraints of the system, the speeds and torques available from the motor the gear ratios and normal directions of rotation. Without these correctly set up, at best any flowchart programs won't work as expected, at worst, the user may break either or both their machine and motor/gearbox.

The Program Parameters Window can be found by clicking on System in the Control area and then selecting Program Parameters. This window is split into 5 sections. Pin setup, Motor, Gearbox, Encoder and Home. The Pin setup section should be ignored for now, this will become relevant in Tutorial 5 + 6 when inputs and outputs are introduced.

On the next page part of our D3857-2490 Data sheet is shown. Running from a supply voltage of 24V, the motor setup information can be copied into the Program Parameters window. Stall Torque is 177.4 mNm and the

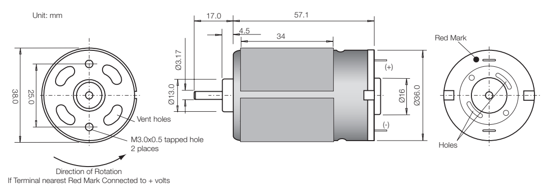
No load speed is 6100 rpm. The Motor Direction can be read from the drawing. Assuming that the terminal next to the red dot is connected to the positive wire (red wire) then the direction of the arrow can be used, showing that the motor rotates counter clockwise (CCW).

DC Motors

Miniature Motors, Transmission and Control

D3857-2490

Issue 02

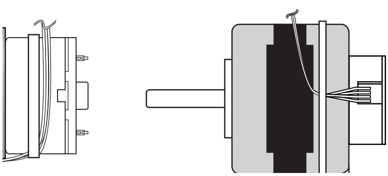


Model Number	Volts V	Performance: No Load Speed rpm	No Load Current A	Maximum Efficiency Speed rpm	Maximum Efficiency Current A	Maximum Efficiency Torque mNm	Maximum Efficiency Output W	Stall Torque mNm	Stall Current A	Max Power W
D3857-2490	24	6100	0.21	5250	0.90	23.8	13.1	177.4	5.83	27.5

Performance characteristics and general specification are measured from sample motor.

From the Label attached to the side of the gearbox the Gear ratio can be read. For this unit the Gear ratio is 408221:2535. An OP on the label indicates that the Gearbox has an opposite direction of rotation to the motor, an SA would indicate that the rotation is the same as that of the motor.

These tutorials will be undertaken with a unloaded gearbox, so the Output shaft load torque is 0 mNm. Information regarding the encoder can be taken from our encoder data sheet. As shown below, the encoder is dual channel, featuring 48 pulses per revolution.



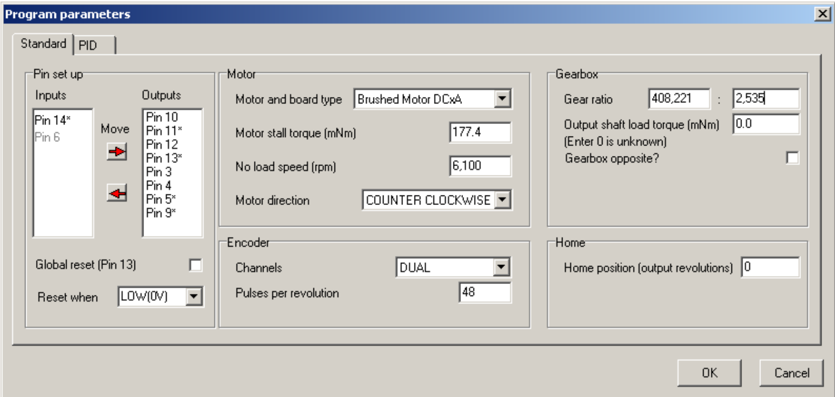
General Specification

Number of Channels	2
Pulses per Motor rev.	48
Output	CMOS/TTL
Supply Voltage	5 Vdc
Supply Current	<30mA

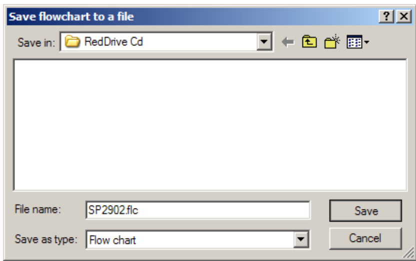
Wiring Colour Code

+5V	Brown
Ground	Blue
Channel A	Yellow

The home position indicates the position value to be taken by the Red Drive controller when the unit calibrates. This is done both on start up and if the user issues a reset home block, 0 should be entered here.
After completing these stages, the Program Parameters window should look as below.



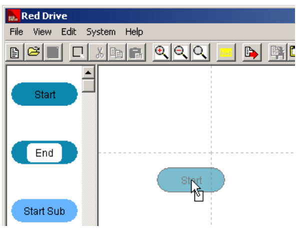
Since the Program Parameters will generally be the same for any system using this motor and gearbox combination (the output load may change), it may be useful to save the empty program as a template.



Once the Program Parameters have been set, work can begin on drawing a flowchart program.

Start

The first command for any flowchart program is 'Start', this tells the Red Drive controller where to begin. Click and drag this block from the new block area into the editing area as shown below.



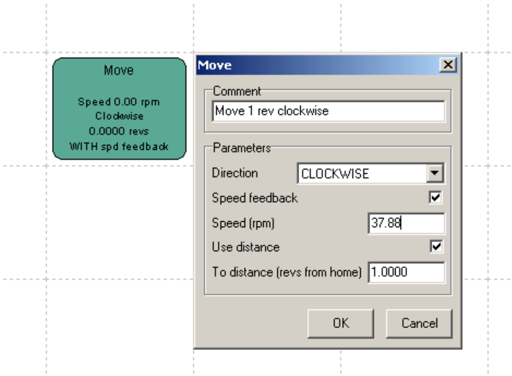
Move

As stated at the beginning of this tutorial, the first task of this flowchart is to move the gearbox shaft 1 revolution clockwise. To do this a Move block should be placed underneath the Start block. This is done in the same way as with the Start block.

Double clicking on this Move block displays the Properties window for the block.

The comment box can be used to enter a brief description of the block. 'Move 1 rev clockwise' is appropriate in this case. The direction is correct at Clockwise. Speed Feedback should be left ticked to allow the Red Drive controller to monitor and ensure the correct running speed. Resting the mouse pointer over the Speed (rpm) text entry box causes a pop up to be displayed. This shows the minimum and maximum speed at which the gearbox can move. To move as fast as possible, the maximum speed should be entered, for the SP2902 unit this will be 37.88 rpm. The output shaft needs to rotate by 1 revolution. Use distance needs to be ticked to allow a distance of 1 to be entered into the To distance (revs from home) box.

Note: When a system is switched on, the home position is set to the value contained in the Program Parameters. To move 1 rev clockwise the number entered into the To distance (revs from home) field should be 1 greater than the home position. If the home position is 0, to move one revolution the To distance value should be 1. If the home position is 5, the To distance should be set to 6.



Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Click on OK to confirm the values entered.
This move block will now instruct the motor to move the gearbox output shaft 1 revolution clockwise.

Stop
After ending the move block the motor will continue to turn unless told otherwise. The next block required is a Stop. The only parameters available for a Stop are a comment. This can be left blank.

Wait
Because a stop has no parameters a method is required to produce a stop which lasts for 1 second. Placing a wait block beneath the stop will do this. Double click on the wait, and enter 1 into the Time to wait (seconds) box.

Move
A counter clockwise move back to the start position needs to take place next. Press and hold the ‘control’ key (often labelled Ctrl on a PC Keyboard) click on the first move and drag to beneath the wait block. This will copy the first move, reducing the amount of data needing to be entered. The comment box should be changed to ‘Move back to home’, the direction is counter clockwise and the To distance (revs from home) should be set to 0.

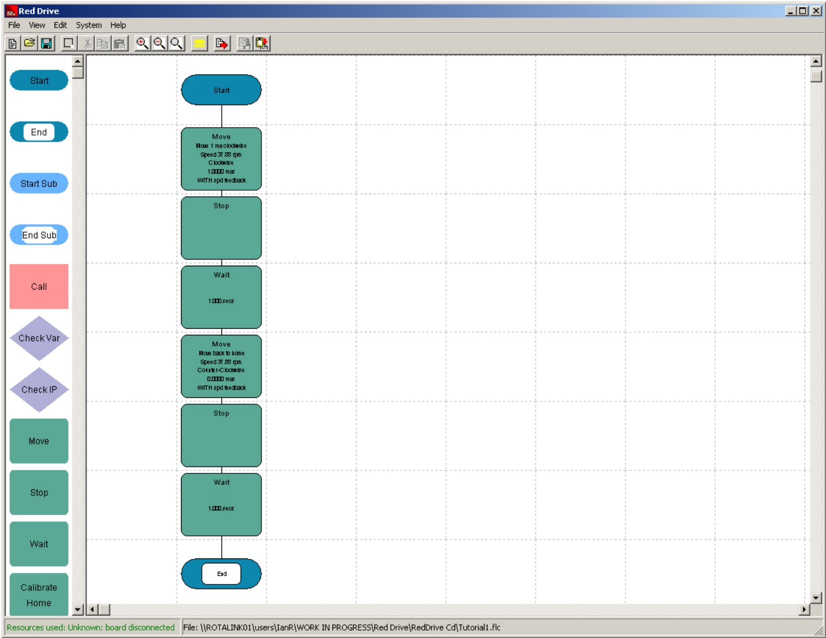
Note: Red Drive operates an absolute positioning system. All distances are given as positions, so a move of 1 rev counter clockwise (CCW) from position 5 revs would be To distance 4 revs. A move of 1 rev CCW from position 1 rev would be To distance 0 revs.

Stop, Wait
Following this move a stop and wait should be added (these can be copied using the Ctrl key as with the move)

End
The final block to be added should be an End block. This tells the Red Drive Controller that the sequence of blocks has finished and the unit should return to run from the start block. This return function provides the repeat loop required for this flowchart program.

Linking

Finally all the blocks should be joined together. Place the mouse pointer just below the Start block so a cross hair appears (+) click and then drag the pointer into the End block. The flowchart should look as below:



Downloading and Running

All that remains now is for the flowchart program to be downloaded onto the Red Drive controller. The section 'Connecting to PC' (page 16) details the physical process of connecting, 'programming the hardware' (page 20) runs through the downloading stage.

Once the 'Programming successful' window has been displayed, the Red Drive controller can be switched off, the Serial Adapter removed and the motor plugged in. Switching the power supply back on should start the motor in its sequence of clockwise and counter clockwise movements.

Note: If our SP2902 unit is being used with this program the user may notice that the Red Drive controller is moving the output shaft a little over one revolution before stopping - an overshoot has occurred. Insuring positional accuracy will be looked at in tutorial 2.

Tutorial 2 - Accelerate, Decelerate, Position control

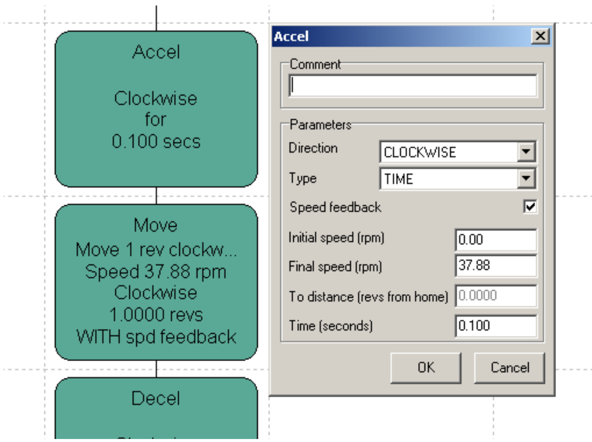
This tutorial will look at the use of the Accelerate (Accel) block and how it can be used to reduce the large start up currents common with DC motors, and also Decelerate (Decel) blocks, and how they can be used to improve the position control attained by the Red Drive Controller.

Accel

DC motors generally have a stall current far in excess of their running current. The D3657-2490 unit has a stall current of 6.25 Amps, but at maximum efficiency, the point at which a design should aim to run the motor, only 0.84 A is consumed. When the motor is switched on, for a very short period of time the motor is not turning, so the stall current is drawn. If this period is too long, or for too great a current it may cause the Red Drive controller to reset. In these situations it is best to accelerate the motor up to speed. Applying a lower voltage to the motor for the short period where the motor is stalled will produce a lower stall current, ensuring that the motor starts without causing a reset.

The flowchart created in Tutorial 1 can be modified to include a 'soft start'. To do so an Accel block should be added between the start block, and the first move block. This can be achieved by moving the mouse pointer over the move block until a 'hand' appears. Click and hold the left mouse button and drag downwards into the next space. A gap will have opened up, and an Accel block can then be taken from the new block area and directly placed over the linking line. This will automatically connect it into the main flowchart.

Double clicking on the Accel block will open up the properties window for this block.



The direction of the first move is clockwise, and it follows that the output shaft should also be turning clockwise through this acceleration. There are two types of acceleration. Those which are purely based on TIME (accelerate from 500 rpm to 100 rpm in 1 second), and also those which are to a FINAL POSITION (accelerate from 500 rpm to 100 rpm in 1 second and then move to 2 revs from home). In this example an acceleration based on TIME is sufficient. Speed feedback is left on, and the initial speed should be set to 0 rpm. The final speed should be set to the same speed as the move (37.88 rpm). Time in seconds should be set to produce a smooth start to the motor, 0.1 seconds has been shown to be a good value to try. The user may wish to vary this value to suit their impression of a 'good' and 'clean' start.

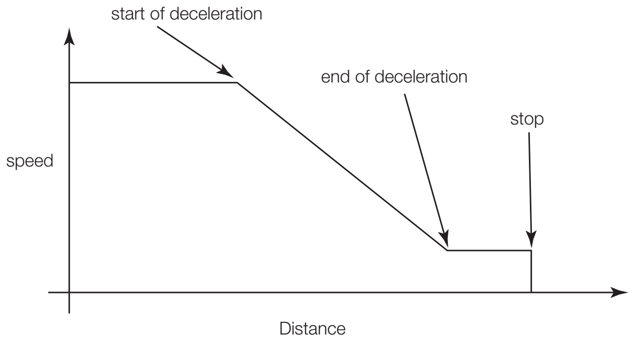
Downloading this program and then running should demonstrate a noticeable difference in smoothness of start comparing the clockwise rotation with the counter clockwise movement.

This Accel block should be copied and placed before the CCW move. (Ctrl-click drag can be used to copy the block). The direction must be changed to CCW.

Note: It is important to make sure that the appropriate direction is inserted into an acceleration (or deceleration) block, unexpected motion can result if this is not correct.

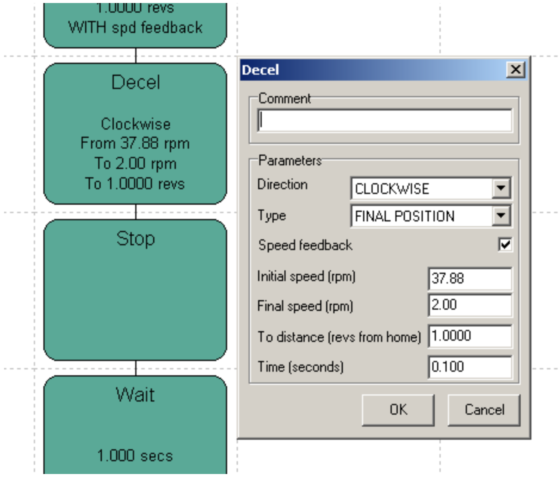
Position Control

As described in the section 'Positional Accuracy' (page 30), running a motor at full speed and then expecting it to stop exactly where required is impossible. By using Decel blocks the rest position can be more accurately controlled. A 'soft stop' as shown below will now be added to the flowchart program.



Decel

A Decel block can be added between the first move block, and the stop block. This can be done in a similar manner as with the Accel block. Open the properties window for this block.



As with the acceleration block, the direction of rotation should match that of the move it is associated with, first move is clockwise, so this deceleration will also be in a clockwise direction. As with the accel block you can choose from two types, TIME and FINAL POSITION. This example is decelerating to a position, so the type should be set to FINAL POSITION. Speed feedback is left on, and the initial speed should be the same as the previous move (37.88 rpm). To achieve the best accuracy on the final stop, the final speed should be set as slow as possible i.e. as slow as the motor can move without stalling. For our SP2902 unit, 2 rpm is a good speed to use. We wish to end this decel block after 1 rev, so To distance (revs from home) should be set to 1. Time can be set to 0.1 seconds.

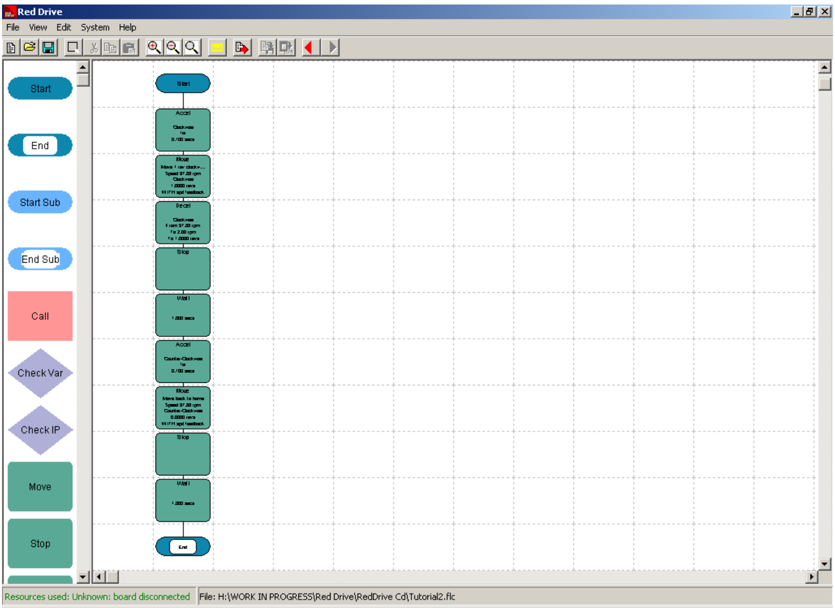
Revising move

The initial clockwise move ends at a position of 1 rev from home, if we left the flowchart as it is, this move would run to 1 rev, and then the deceleration block would be looked at. The deceleration would be run, but because the position counter had already moved beyond 1 rev the unit would quickly skip onto the next block - the unit would continue to overshoot as at the end of Tutorial 1. The decel block needs to be run before a position of 1 rev from home is met. This revised position should be final position (1 rev) minus the distance taken to decelerate in 0.1 seconds.

This deceleration distance can be calculated fairly easily. Because the deceleration curve is a straight line we can assume that the distance will be the same as if the unit was moving for the same length of time at the average speed.
(Initial speed - final speed)/2 + final speed = average speed
(37.88 - 2) / 2 + 2 = 19.94
19.94 rpm = 0.33 revs per second
Multiply by 0.1 seconds (decelerate time) = 0.033 revs.
The inertial of the system will have an impact on this deceleration distance, allowances should be made for this by increasing the value. In this example a deceleration distance of 0.06 revolutions will be assumed. Distance in the move block should therefore be changed to 0.94 revs.

CCW deceleration and move

A similar process should be carried out in the CCW direction. The move distance should be changed to 0.06 revs and then a copy of the CW deceleration block should be made changing the direction to CCW and the Final position to 0 revs. The flowchart should look as below:



Compiling, downloading and running this flowchart will result in the output shaft completing exactly one revolution before waiting and then reversing. The CCW deceleration will see the unit returning to its starting point

Tutorial 3 - Start Sub, End Sub, Call

Start Sub, End Sub and Call blocks are the building blocks of subroutines, this tutorial will introduce what they are for, and how to use them. Subroutines are mini flowcharts used to contain often used blocks. A pointer (Call) to this mini flowchart can be used in place of the blocks. When the flowchart reaches this Call, the mini flowchart will run and then the main chart will continue. This is useful in two ways:

- Writing programs is quicker and involves less typing.
- Programs featuring subroutines can be more efficient, allowing more complex flowchart programs to fit within the Red Drive memory.

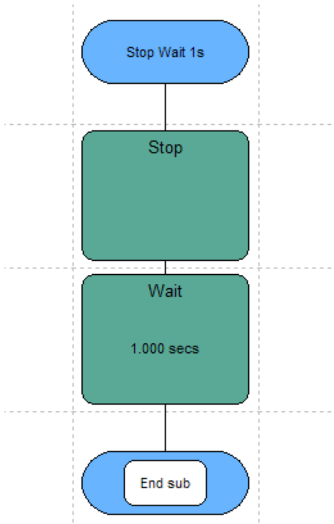
Tutorial 2 used Stop and Wait blocks to end each movement. In a more complex program, these blocks could be used many times. To make programming easier, and to save memory, a subroutine should be used.

Start Sub

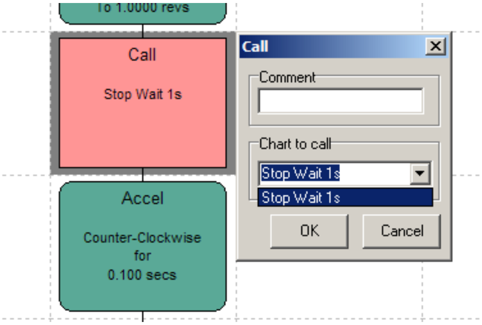
Each subroutine must begin with a Start Sub block. Click and drag to place a Start Sub to the right of the Start block. The Properties window (double click the block) shows a Name dialogue beneath the familiar Comment dialogue. A name must be given to the subroutine here, 'Stop Wait 1s' seems appropriate. The Stop and Wait blocks can then be moved beneath the start sub by holding the Shift key and dragging into place.

End Sub

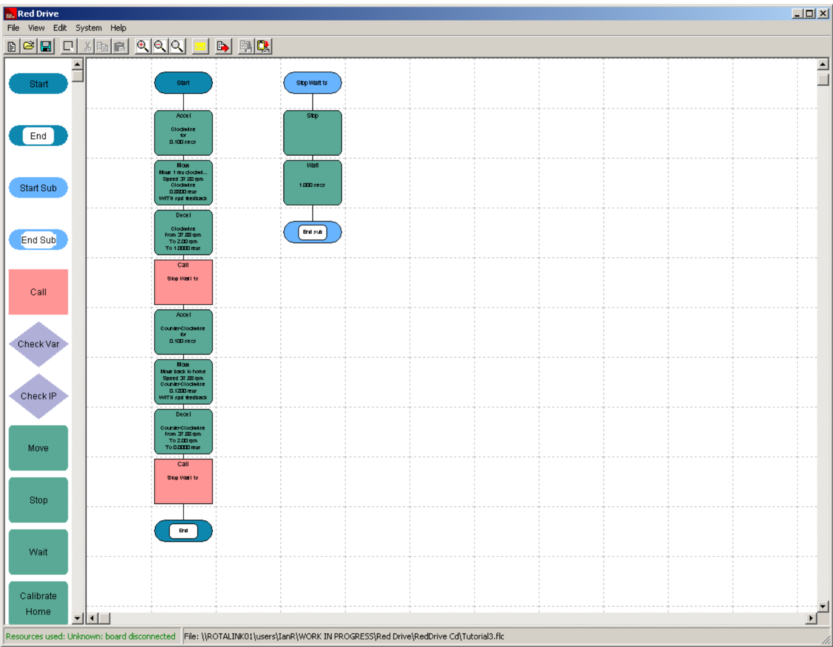
Finally a End Sub block should be added and the three blocks joined.



The subroutine has now been created, all that remains for this subroutine to be used are some Call blocks. The First Call block should be located in place of the moved Stop and Wait blocks. The Properties window is similar to that of Start Sub. Clicking on the down arrow in the Chart to Call section produces a drop down menu of all the subroutines created so far. ‘Stop Wait 1s’ should be selected.



The second Stop and Wait blocks should also be deleted (right click), and the Call block copied in place. Compiling, downloading and running should see the motor move as before. By using subroutines, the program is now neater and easier to follow and expand in future.



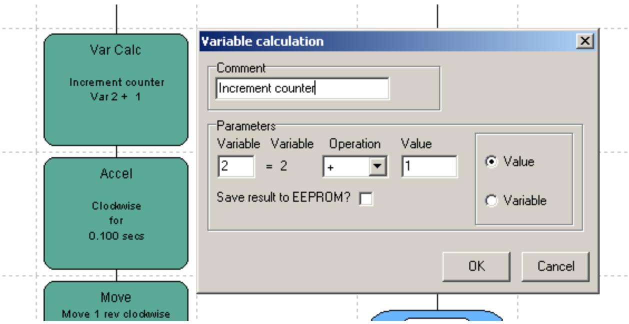
Tutorial 4 - Check Var, Var Calc

Variables can be very useful in reducing the amount of flowchart that has to be written to complete a given task. As with subroutines they have a dual purpose of reducing the time spent writing a flowchart, and also in saving memory. The flowchart program written so far in these tutorials is going to be modified again. Rather than simply moving clockwise and then counter clockwise endlessly, every 5 cycles (CW & CCW motion), a delay of 5 seconds will be added.

This could easily be achieved by copying all the blocks 5 times, and then adding a delay of 5 seconds before the End block. Writing in this way takes a lot of time, and wastes memory. If the flowchart was written in this way 14% of the Red Drive DC1A controllers memory would have been used. A better way is to use Check Var and Var Calc blocks.

Var Calc

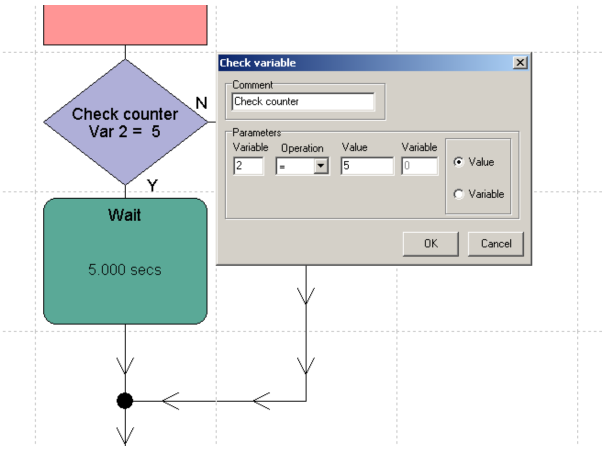
In order to count the number of times the system runs through a CW/CCW motion a counting block is required, Var Calc is used for this purpose. Insert a Var calc block just below the Start block. Open its Properties window.



In the Parameters section are several options. The first option to set is whether the calculation is against a value or variable (right hand box), we want to create a counter, so value should be selected. Looking at the main calculation line, Variable is just a marker indicating where the data will be stored. Variable 1 is set aside for error logging (see page 32) so we will use variable 2. This block is being used as a counter, so the Operation needs to be '+' (this is changed using the drop down menu). Value should be set to 1, to ensure that the Variable increments by only 1 each time (2 would see an increment of 2). We don't need this data on power on, so leave 'Save result to EEPROM?' unticked.

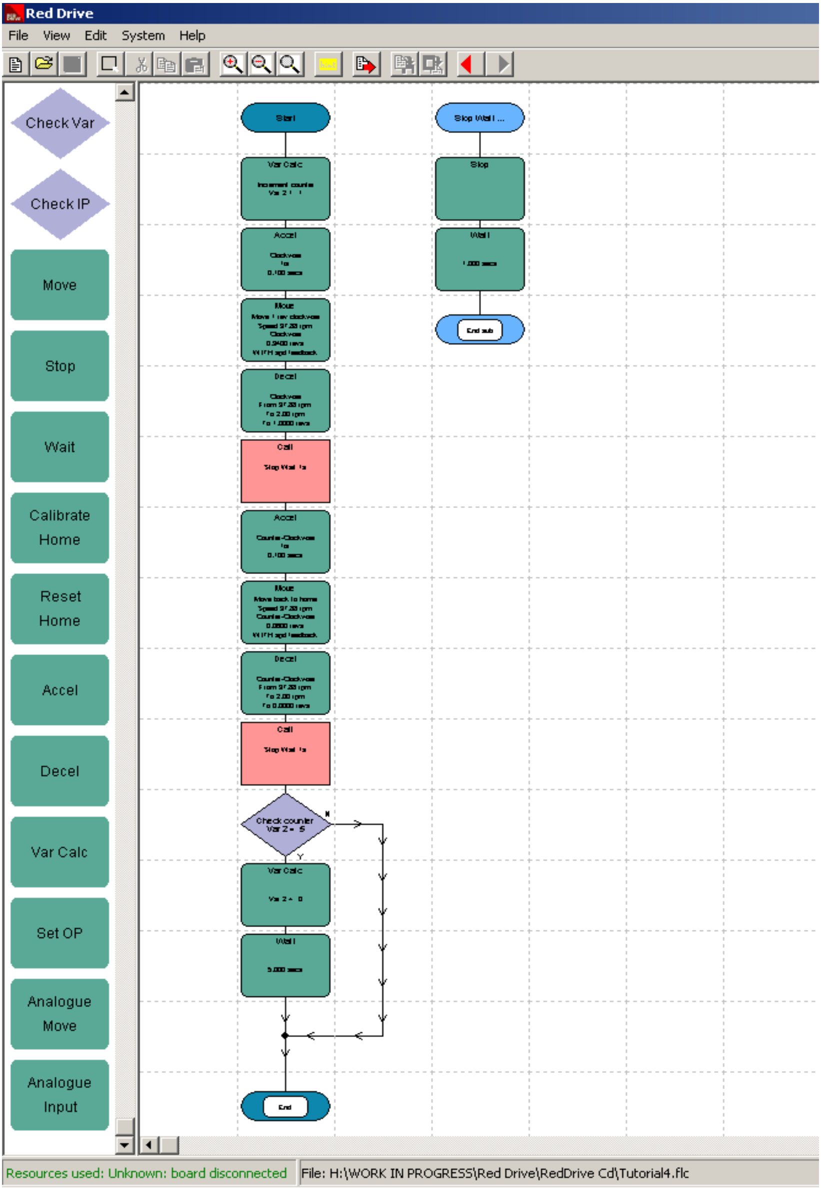
Check Var

A delay of 5 seconds is required after 5 counts have occurred, to know if this has happened, use of the Check Var decision block is required. This block should be placed directly above the End block, with the 'Y' being connected to the End. In Properties there are again several options which take the same format as the Set Var block. The data to be checked was set up in Variable 2, so this must be checked with the Check Var block. The value should be set to 5 (ensuring the selection to the right hand side is 'value'), to check if the CW/CCW motion has been run 5 times. The Operation is either equal (=) or greater than (>) or less than (<). In this case we want to wait for 5 seconds when Variable 2 = 5. Therefore the Operation is equal (=). Is Variable 2=5? If the result is yes, then Variable 2 is equal to 5 and the delay must be run. A Wait block (set to 5 seconds) should be inserted between the Check Var block and the End block. Following from the 'N' a link should be made skipping the wait block.



Compiling and running this program will see the CW/CCW motion run 5 times, then there will be a delay of 5 seconds before the system runs the CW/CCW motion again. However, the unit will not run this delay for another 250 motions (Variable counters reset at 255). The system will keep checking Variable 2, however it becomes equal to 6 then 7,8,9,10. The delay is only run when the Variable is equal to 5. To provide the required operation Variable 2 must be reset to 0. This is done using a further Var Calc block. Using the Properties window the Parameters are set to Variable 2, Operation =, Value 0. When this block is run, Variable 2 will become = to 0. With the flowchart program as below the system will carry out 5 CW/CCW motions, delay for an extra 5 seconds, and then carry out a further 5 CW/CCW motions repeating this sequence.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000



Using this method of repeating sequences the program memory has been reduced from 14% to a much smaller 4%. This technique saved memory, and has resulted in a simpler, easier to follow program.

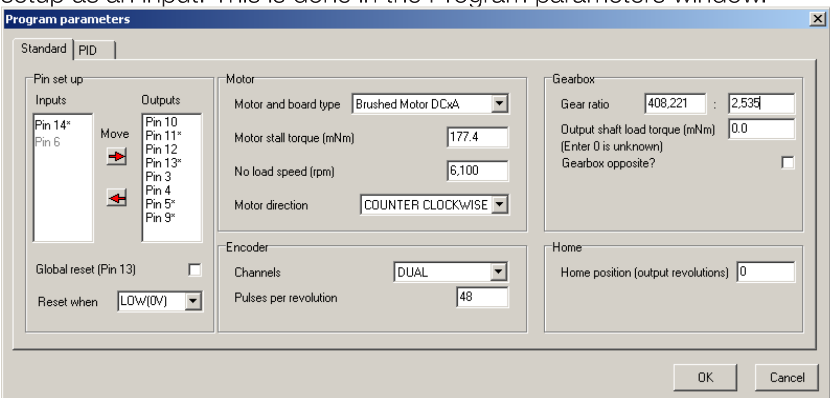
Tutorial 5 - Using Home, Calibrate Home, Reset Home

Home is an important concept when making use of a Red Drive controller. Systems requiring fixed positional accuracy make most use of the 'Calibrate Home' block. Relative positional systems make use of the 'Reset Home' block. This tutorial will first look at methods for setting up the hardware of a Calibration point, and then the basics of using the Calibrate Home block to ensure accuracy. Use of Reset Home will be discussed highlighting its particular usage for rotary systems.

Using Home

Home is an important and useful tool when used with Red Drive. Whilst the encoder keeps track of the position of the system, this information is lost if there is a power cut, and can be disrupted by high levels of electrical noise in the surrounding area. These problems can be managed with the use of a home position.

At its simplest, the home position is set up with a switch. When the unit reaches and activates this switch, it is deemed to have reached 'home'. The SP2902 unit features a Optical switch. This should be connected to Pin 14 as shown in the 'Connecting Inputs and Outputs' section of the manual (page 7 or 14). Each time the pin on the output shaft passes though the optical switch, a signal is sent to the Red Drive controller, which can then be used to calibrate the home position. Pin 14 also needs to be setup as an input. This is done in the Program parameters window.

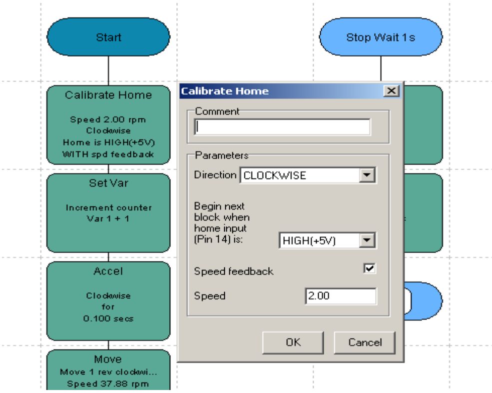


All available pins are listed either under Inputs or Outputs. The home calibration switch is on Pin 14. This must be defined as an Input in this window. If Pin 14 is currently defined as an output, it can be moved by clicking on 'Pin 14' and then clicking on the bottom arrow. Pin 14 should then be listed under Inputs.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Calibrate Home

The main use of calibrating the home position is to ensure the absolute position of the system when the power is switched on. Therefore, the Calibrate Home block should be the first block in the flowchart. Opening the Calibrate Home Properties displays the following window



The direction is very important in a linear system, if this is wrong, the system will never find the home switch, and home will never be calibrated. For a rotary system, movement in either direction will cause the home switch to be activated. In this example the direction can be left as clockwise. 'Begin next block when home input (pin 14) is:' should be set to HIGH(5V). This relates to how the switch is set up, and how it is activated. Our switch outputs 0 volts to Pin 14 when the detector is clear, and 5 volts when the pin passes though. After receiving this input, the flowchart program will immediately move onto the next block. To ensure accuracy, the motor should pass through the optical unit slowly. Setting the speed to 2 rpm (as used for final position accuracy in Tutorial 2) should be sufficient.

Downloading and running this program should see the system stopping and starting around the home position, regardless of the start position.

Reset Home

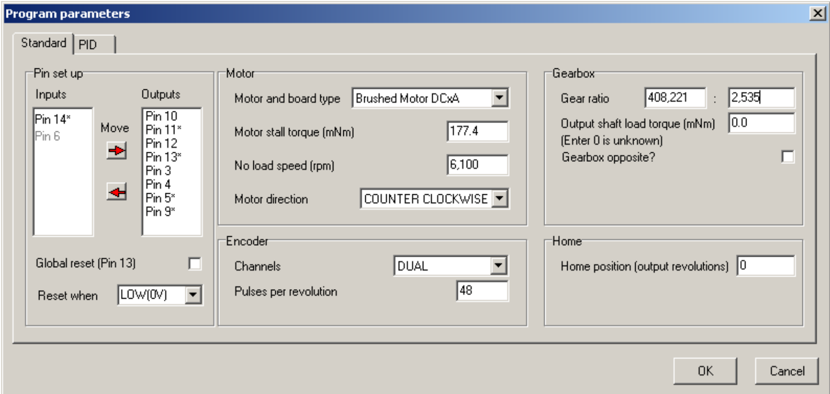
Reset Home is useful for applications where relative positioning is required. If a rotation of 1 rev is needed for driving a pump, driving to 1 rev (first cycle), and then to 2 revs from home (second cycle) may not be suitable. If there was an over pump with the first movement, an under pump (to move to 2 revs from home) might not be wanted, rather another attempt to pump for 1 revolution. In this instance use of a Reset Home block at the beginning of each movement would cause the unit to 'forget' any previous error and simply move a further 1 revolution. The motion would be Start, Reset Home, Move 1 rev, End.

Tutorial 6 - Check IP, Set OP

This tutorial will introduce the use of inputs and outputs within the Red Drive system, how they are set up, and how they are used. The easiest way to introduce inputs (IP) is to use the home calibration switch as an input.

Setting up Inputs and Outputs

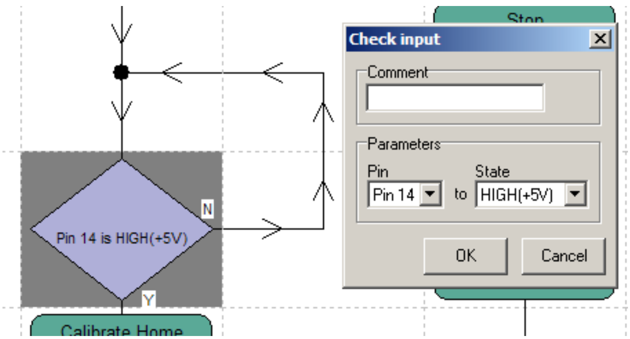
All inputs and Outputs are setup in the Program parameters window.



The available pins are listed either under Inputs or Outputs. Pin 14 should have been set as an input in Tutorial 5 (in order to use Calibrate Home). If Pin 14 is listed as an output, move to the input column as per tutorial 5.

Check IP

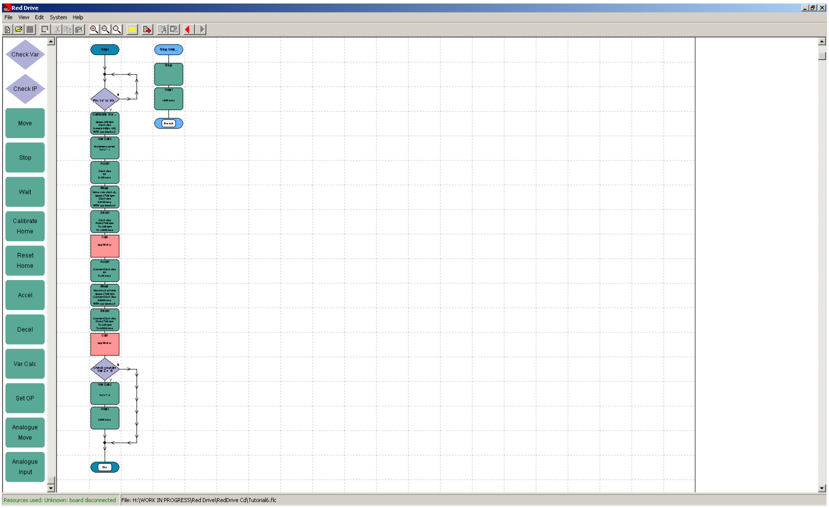
Often it's not appropriate to have the motor running as soon as the power is switched on, rather the system should wait for a user input before beginning the flowchart program. This is particularly useful for responding to any system errors in a fail safe manner (see page 32). To function as a 'user start' switch, a Check IP block will be required at the very beginning of the flowchart, with the 'N' looping back into the Check IP block.



Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

The properties window lists 2 Parameters, Pin and State. Pin 14 is the only input with a switch connected, This should be selected from the drop down menu. State follows the same principle as used for Calibrate Home discussed in Tutorial 5, this should be set to HIGH(+5V).

The End block will need to be deleted, and a loop run from this point up to the Calibrate Home block. This ensures Check IP is only run once - each time the power is switched on. Without this loop the system would wait for a user input before beginning each cycle. The same switch is being used twice, so a wait is required after the Check IP command to ensure that setting this switch does not also cause the unit to 'calibrate home'.



Compiling and then downloading this flowchart program should result in the system waiting until the optical switch is blocked (a piece of card or paper can be used). The unit will then wait for a further 2 seconds before calibrating the home position and starting its CW and CCW sequences.

Set OP

When an output is attached to Red Drive and its relevant I/O pin set up in the Program Parameters window, it can be controlled using the Set OP block. In the block properties window the Pin can be selected, and then set as either HIGH(+5V) or LOW(0V). This can then be used to switch LEDs on/off, set relays, or send signals to other Red Drive Controllers.

Troubleshooting

Unit is drawing much more current than expected

Some motors are unsuitable to be driven by Red Drive or any other PWM controller. This is due to the type of suppression fitted to the motor. The only solution in this instance is to use a different motor. Rotalink will be happy to assist with a new selection.

Irregular speed using speed feedback

There are 2 causes for this. Either the required speed is simply too low and the motor is constantly starting and stalling, or the PID system is set too aggressive for the speed required. Setting the kp ki and kd values to 0 will demonstrate if the required speed is physically possible with this motor. If this test is successful it should be possible to tune the PID system to improve smoothness (see page 24).

Unit restarts part way through a cycle

Restarts are caused by the Red Drive controller experiencing a reset condition. See the Fail safe section (page 32).
The two main causes of this failure are discussed below:
1) With a DC1A / DC2A if the motor has a stall current of 4 A or more, it is possible to overload the system (causing a reset) by trying to stop or start too quickly. Use of acceleration and deceleration blocks to soft start and soft stop the system should remove this overload condition.
2) Matching the power supply to the selected motor and motion is critical. If the power supply cannot deliver enough current quickly enough, the voltage to the Red Drive controller can drop below that at which it can operate. The key indication of this fault is the program restarting when starting or stopping a motor. DC motors draw large amounts of current when switched on (for a short period of time the stall current is required), and also when stopping (due to acting as a generator). If these events are drawing too much current there are two options. Firstly a larger power supply can be selected (ideally the power supply should match the motor stall current), secondly the flowchart can be modified to include ‘soft starts’. By accelerating (and conversely decelerating) the motor to speed, this initial current surge is reduced. How long this soft start needs to take is entirely dependent upon the motor, load being driven and the current rating of the power supply being used.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Unit accelerates to maximum speed and then restarts

If speed feedback is switched on (box ticked), then this is indicative of a failed encoder. The Red Drive controller is receiving no position feedback from the encoder, and so increases power in an attempt to get the unit moving. The motor will be accelerated to full power without the encoder sending a pulse to the Red Drive Controller, causing an error within Red Drive and therefore a system reset. This error will also be shown by a value of 4 being placed in Variable 1 on reset (see page 32).

Skipping of blocks (flowchart block fails to run)

The Red Drive controller considers position to be important above all other parameters. If the controller is asked to complete a move which is not possible, for example moving CW to position 5 when the system is already at position 6, it will assume that the command is no longer required (as the system has already moved passed position 5), and skip onto the next block. If the distance is correct and the block still does not appear to be running, then the direction should be checked and the system checked for any over shoot from a previous block.

Unit fails to start

Firstly check the Program Parameters. If the motor normal direction is incorrect Red Drive will detect that the direction is wrong and reset.
Is the global reset function is being used (see page 32)? Check that this is set the correctly (is a LOW (0V) signal being checked for when a HIGH(+5) is required)? Otherwise, has the controller been configured to use another I/O pin to supply the ground or 5 volt signal for the global fail safe pin? The global reset check operates before these pins are set up. Supply for a global fail safe switch must come from the +5V pins.
If possible, also check the motor for movement. If the gear ratio is large the error may be ‘Unit accelerates to maximum speed and then stops’.

Unit fails to carry out programmed function

This maybe an overshoot, random change in direction or other ‘strange’ event.
Usually caused by excessive electronic noise effecting the Red Drive controller. Please ensure that all unused I/O pins are set as outputs in the Program Parameters window (see page 22).

Working out Motor/Gearbox direction

If this information is not available, it can be calculated. Set up a program to run the motor for 1 revolution CW (it is best to switch off speed feedback for this test) and then stop (see M+GBtest.flc). Set all known 'Program Parameters', leaving motor direction as CCW and Gearbox reverse blank. Run the motor:

- Output is CW, unit runs for just 1 rev; the system is set up correctly.
- Output ran CCW for 1 rev; Gearbox is reversing, tick in 'Program Parameters'.
- Output CW, did not stop after one rev; motor direction is wrong, set to CW and tick the gearbox reverse.
- Output CCW and did not stop; set motor direction to CW.

Failed to connect to board

This error indicated a communication problem between the Red Drive controller and the PC. Things to check are:

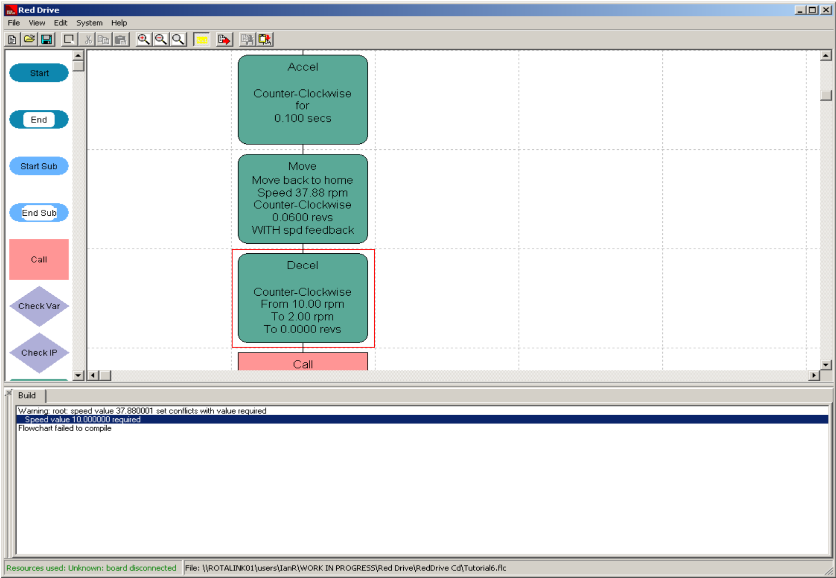
1. Red Drive board is powered
2. Serial Adapter was connected to Red Drive before it was powered? (Cycle the power just to check)
3. Correct Com port selected, System -> Set Com Port (should be safe to use trial and error)
4. If using a USB to Serial adapter has the correct driver been installed

USB

Whilst the Red Drive controller is designed to be programmed via an RS-232 serial port, it is possible to operate from a USB port via a 'USB to Serial Adapter'. For obvious reasons we cannot confirm that every adapter will work successfully with Red Drive, however Rotalink can supply tested adapters if required.

Compiler Errors

If a flowchart program fails to compile a new window will appear in the lower half of the screen.



A textual list of the failures will be listed under the Build tab. A list of common errors and their fixes are described below. This window can be closed using the x in the top left corner.

Warning: xxxxxxxxx Speed value set xx conflicts with value required

Speed value xxxx required

The initial speed set in an Accel or Decel block does not match the speed set in a preceding block. Clicking on these lines highlights (with a red box) the offending block. Either the Accel/Decel or preceding block needs to have a modified speed value entered.

Error: No start element found

A start block is required to indicate the first block to be run by the flowchart program.

Error: Block not connected at bottom

All blocks except the End block require a line indicating the direction of program flow after the block has run. This line should be drawn from the bottom of the block.

Error: Choice not connected at bottom

A choice block (purple diamond) does not have a line indicating direction of program flow from its ‘Y’ option. This line should be drawn from the bottom of the block.

Error: Choice not connected at right

A choice block (purple diamond) does not have a line indicating direction of program flow from its ‘N’ option. This line should be drawn from the right hand side of the block.

Error: Unconnected line

A direction flow line has been drawn from a block, but has not been connected. Without this connect the flowchart program does not know which block should be run next.

Error: End can’t be used to end a sub-chart

An End block has been used to end a subroutine, this should be replaced by an End-sub block.

Error: xxx bytes used is > xxx available

The completed flowchart is too large to be fitted within the Red Drive controller memory. Either reduce the memory usage (using sub routines or Check Var commands to repeat blocks) or contact Rotalink for availability of a Red Drive controller with a greater memory capacity.

Warning: Wait time limited to xxx seconds

A wait has been entered which is longer than Red Drive can manage. If a very long wait is required successive waits should be used.

Warning: Speed xxx > no load speed xxx / gear ratio xxx

A speed has been entered which is greater than the maximum speed available with the system as defined in the Program Parameters window.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Warning: Initial speed xxx > no load speed xxx / gear ratio xxx

The Initial speed set up in an accelerate or decelerate block has been entered as a value greater than the maximum speed available with the system as defined in the Program Parameters window.

Warning: Final speed xxx > no load speed xxx / gear ratio xxx

The Final speed set up in an accelerate or decelerate block has been entered as a value greater than the maximum speed available with the system as defined in the Program Parameters window.

Warning: Speed and torque formula limited to 4294967295

Acceleration / deceleration time is too long for Red Drive to control. If a very long acceleration or deceleration is required, successive accel/decel blocks should be used.

Warning: Speed and torque formula (restricted DP) limited to 4294967295

Acceleration / deceleration time is too long for Red Drive to control. If a very long acceleration or deceleration is required, successive accel/decel blocks should be used.

Warning: Speed and torque formula (restricted DP) limited to 0

Acceleration / deceleration time is too short for Red Drive to control. A larger time value should be used. A very quick acceleration or deceleration (as quick as motor can manage) is achieved by removing the accel/decel block and simply starting with a Move or stopping with a stop block.

Error: Accel/decel: Processed initial and final speeds both xxx

Both Initial and Final speed in an Accel or Decel block are the same, there is no acceleration or deceleration taking place.

Warning: Speed, gear ratio and encoder pulse formula limited to 65535

Speed is set too slow for Red Drive to control.

Warning: Distance and encoder formula limited to 4294967295

A distance has been set which is beyond the range Red Drive can manage.

Error: Calibrate Home requires pin 14 to be input; it is currently an output.

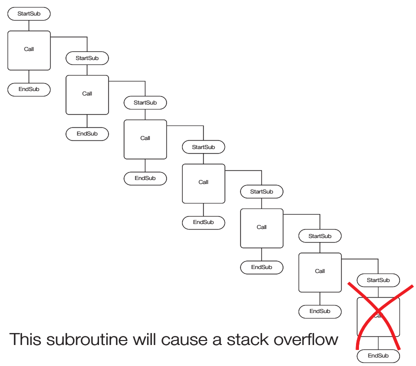
Pin 14 should be selected as output in program parameters

Error: Torques: Output shaft xxx > motor stall xxx * gear ratio xxx

The torque being moved is greater than the torque the motor can produce

Error: Stack Overflow

There is a limit of 6 nested subroutines within the Red Drive system. Calling a 7th subroutine will result in a Stack Overflow error. The Flowchart will need to be re-ordered to ensure only 6 subroutines are nested.



Warning: Disassembly problem

Please confirm the latest version of the Red Drive software is in use. If error continues, please contact Rotalink.

Error: default in CompileFlowChart

Please confirm the latest version of the Red Drive software is in use. If error continues, please contact Rotalink.

Rotalink Limited, Cropmead, Crewkerne, Somerset TA18 7HQ
Tel: +44 (0)1460 72000

Conversion Formula

Torque Conversion factors

Known Units	mNm Conversion Factor
Nm	0.001
Ncm	0.1
lb ft	0.000738
lb in	0.00885
oz in	0.142
Kg m	0.000102
gr m	10.2
dyne cm	10000

If you need to convert 24 oz in into mNm for use with the Red Drive software, take the known unit (24 oz in) and multiply by the mNm conversion factor (0.142).

e.g. 24 oz in x 0.142 = 3.408 mNm

Lead screw Conversion formula

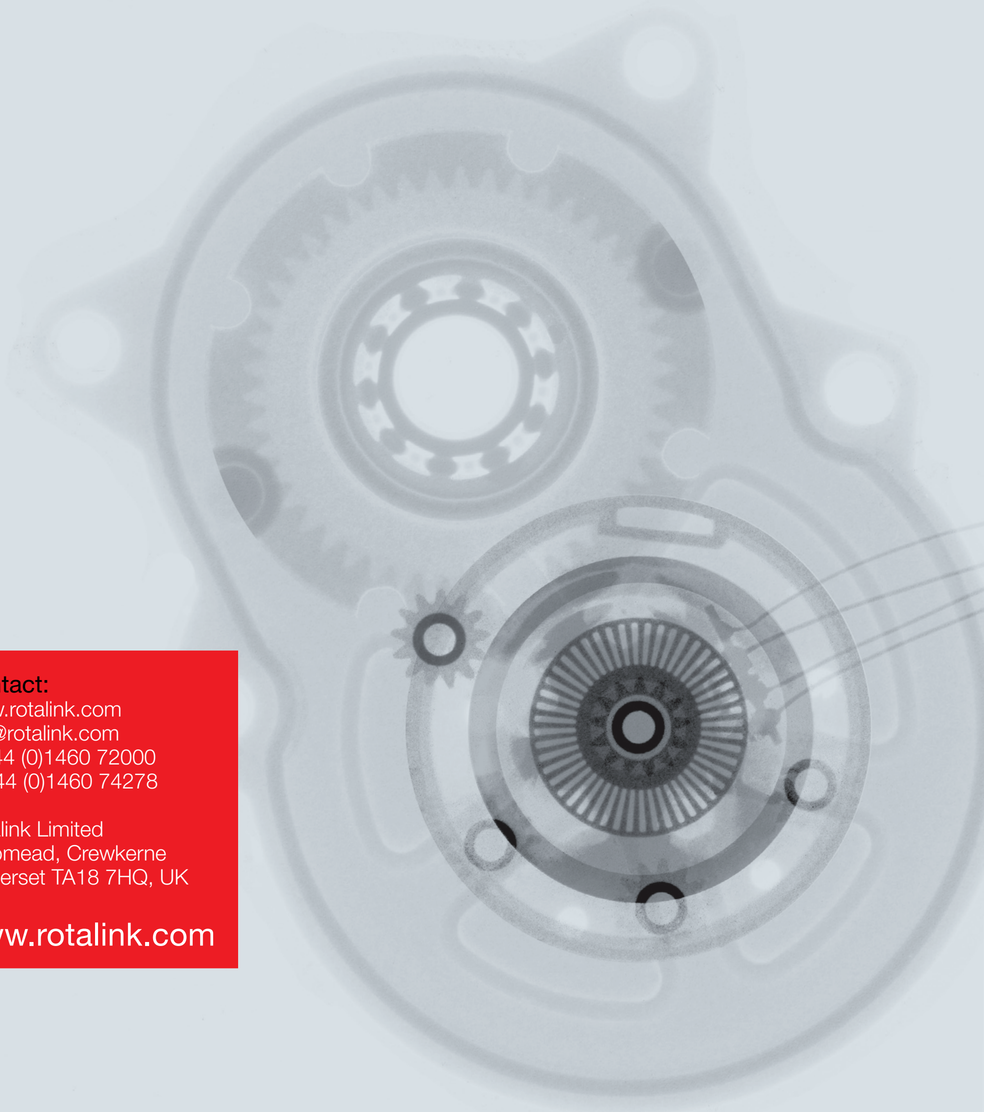
If you are using a lead screw the following formula will be useful:

Distance:
Output shaft revolutions x lead screw pitch = linear distance

e.g. 3.5 revolutions x 80 mm pitch = 280 mm

Speed:
Output shaft speed x lead screw pitch = linear speed

e.g. 500 rpm x 80 mm pitch = 40000 mm per minute
(≈666 mm per second)



Contact:
www.rotalink.com
info@rotalink.com
T: +44 (0)1460 72000
F: +44 (0)1460 74278

Rotalink Limited
Cropmead, Crewkerne
Somerset TA18 7HQ, UK

www.rotalink.com