

### Introduction

Motorised accessories, such as blinds and curtains, can be incorporated onto iLight system to allow control from iLight input devices, alongside lighting. The common motor types and their means of control are:

#### Power or Input Switching

- Direct switching of power to mains AC motors
- Direct switching of power to low-voltage DC motors (reversing polarity)
- Volt-free pulse or latching input to a control unit supplied with the blind system.

There are two products in the iLight range that are designed for this purpose:

#### **RI-2 Relay Interface**



- Wall mounted (metal enclosure)
- 8 x 3A change-over relays

#### **SCMC0410 Curtain Controller**



- Din rail module (MPM2400 Master Processor required)
- 4 x 10A change-over relays

**NOTE:** Full product specifications can be found at [www.ilight.co.uk/products.html](http://www.ilight.co.uk/products.html)

#### Data Only Connection

Some blind setups exist that manage the power switching to the motors themselves, and would only require a data link, such as serial RS232 or RS485 to trigger control functions. In this scenario, an **SI-2 System Integrator** (shown, right) would be used to transmit serial data to a decoder associated with the blind control. (See pg X)



### Contents

#### Wiring Configurations

|   |    |
|---|----|
| Mains AC Motor – Wiring example.....                            | p2 |
| Low-voltage DC Motor – Wiring example (Polarity Switching)..... | p3 |
| Volt-free Input Control – Wiring example.....                   | p4 |

#### Programming Examples

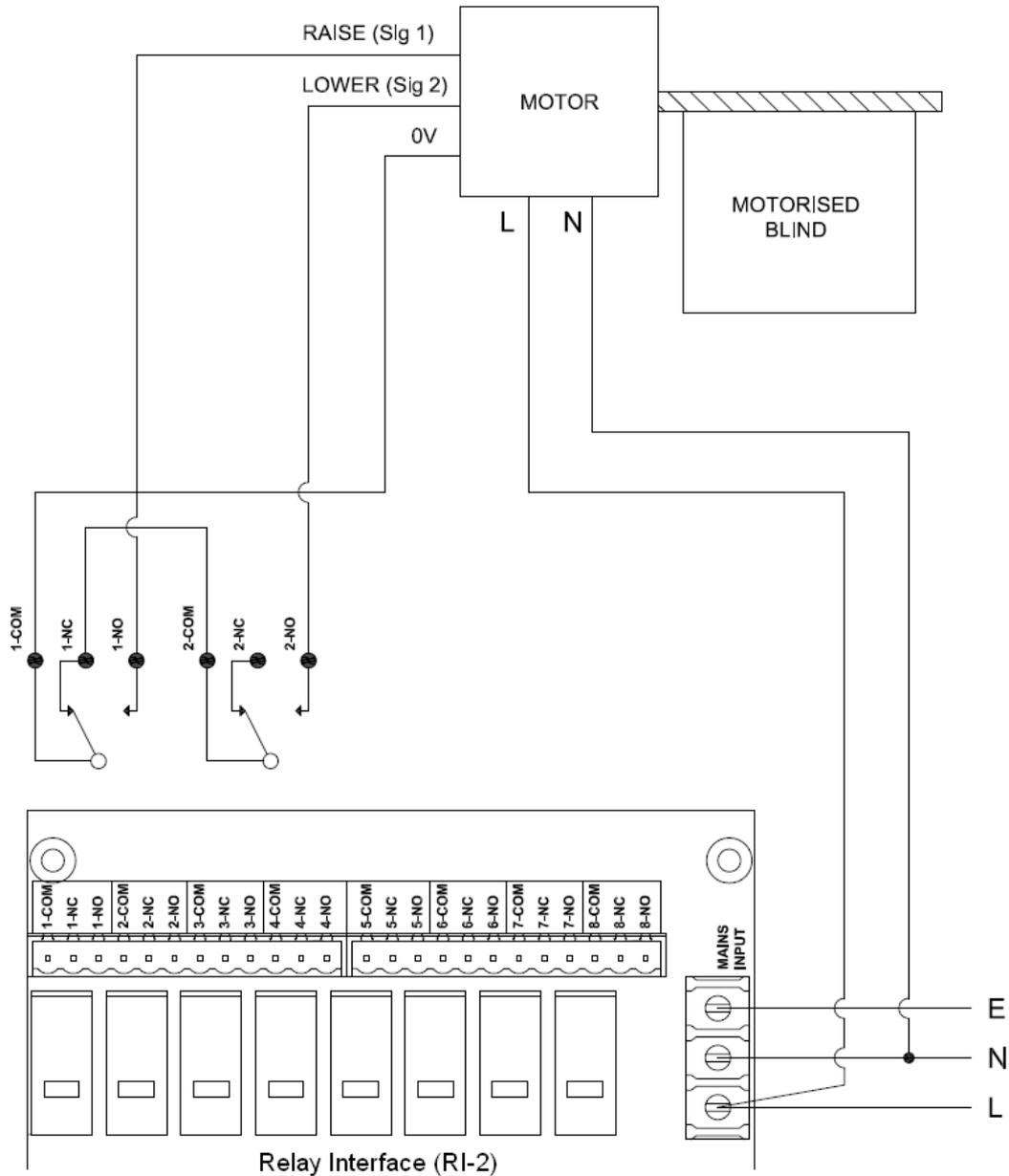
|  |    |
|--|----|
| General Configuration of Output Device and Scenes.....               | p5 |
| 3 button programming example.....                                    | p6 |
| 2 button programming example.....                                    | p6 |
| Single button programming example.....                               | p8 |
| Volt-free input programming example (momentary contact closure)..... | p9 |

|                                 |     |
|---------------------------------|-----|
| Serial Integration Example..... | p10 |
| Master Control Advice.....      | p11 |

|                             |     |
|-----------------------------|-----|
| Troubleshooting & FAQs..... | p12 |
|-----------------------------|-----|

# Mains AC Motor

Wiring Example (iLight RI-2 Relay Interface depicted)



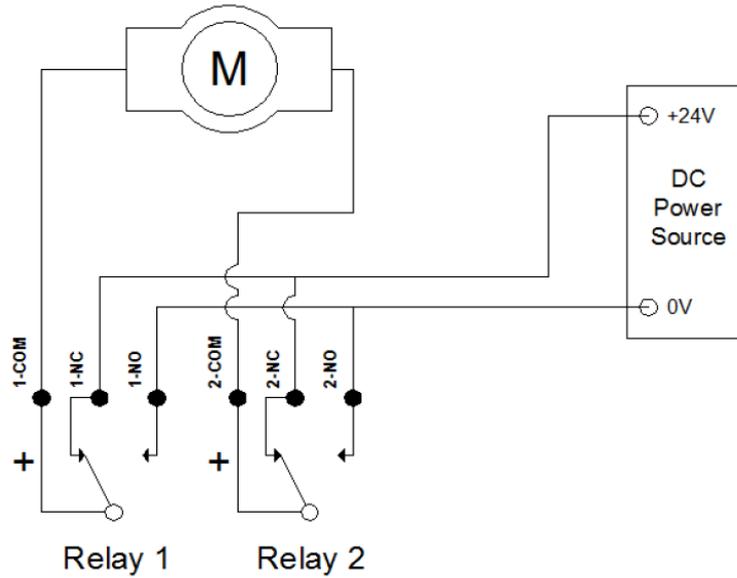
In this example, a 230V supply (max 3A) is wired to the Common of Relay 1. A pair of interlocking relays are required per motor unit and when energised these switch between either feeds (L1 or L2), or break both connections to stop the motor if both are switched off.

Cabling must be mains rated. Neutral is connected directly to the motor.

AC motors cannot normally be group wired, so a pair of relay outputs must be allocated per motor.

# Low Voltage DC Motor (Polarity Switching)

## Wiring Example

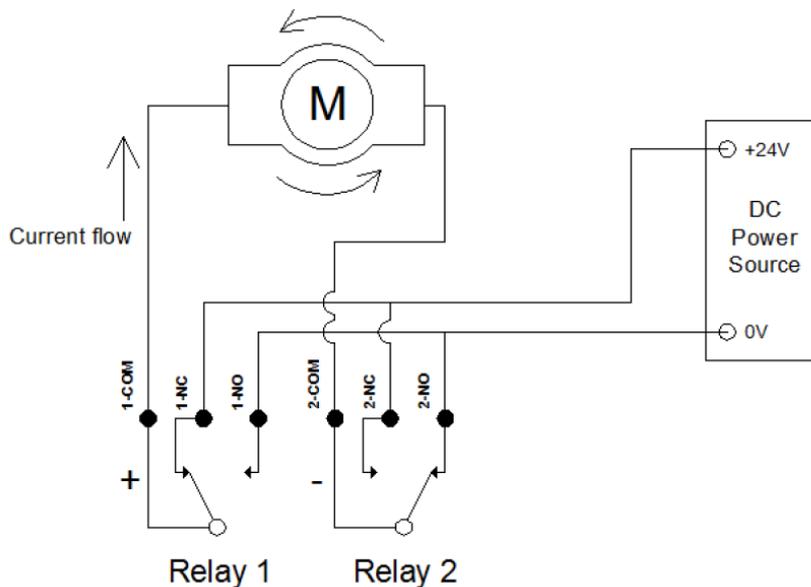


Low-voltage DC motors have their direction of movement dictated by the direction that current is flowing through the motor. This is achieved by being able to reverse the polarity of the DC supply to the motor, and the wiring configuration above shows the method of achieving this using two relay outputs.

With both relay outputs off (NC) there is no current flow through the motor.

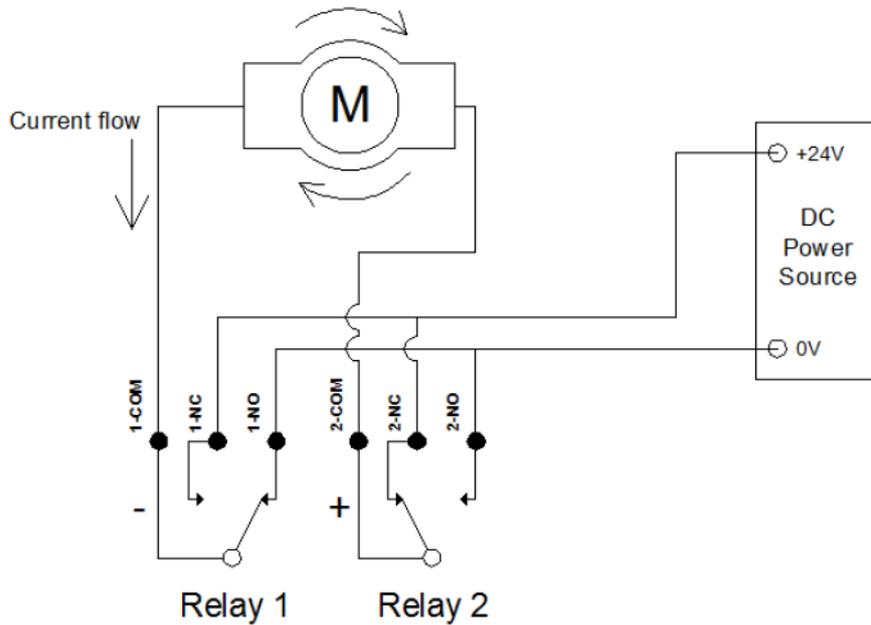
### Movement Direction A:

Relay 1 output off, Relay 2 output on results in current flowing from the positive to negative terminals as shown and the motor turns a in a particular direction.



### Movement Direction B:

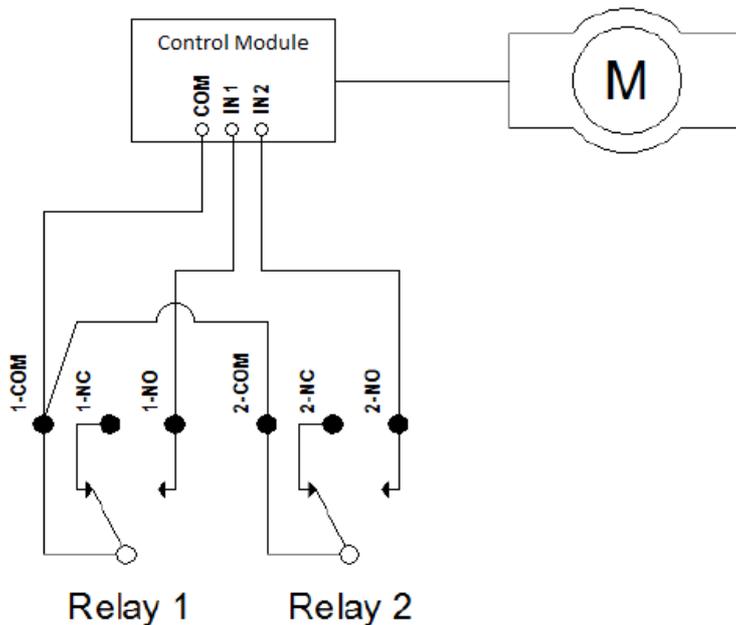
Relay 1 output on, Relay 2 output off results in current flowing in the opposite direction, because the positive to negative terminals are now reversed, and the motor turns in the opposite direction from before.



### Volt-free Input Control

Wiring example:

**NOTE:** Illustrative example only - Consult installation material of specific system being used for wiring method.



In this example, the iLight outputs carry no power but are used to make volt-free contact closure to a control module that forms part of the blind system. The motor is then connected in some way to this module. Some systems may have an individual input for each direction and to stop the motor, both inputs are triggered. The iLight output module can take the place of a remote switch which may be supplied as part of this blind system.

Turning Relay 1 on closes input 1 on the Control Module. Turning Relay 2 on closes Input 2 on the Control Module. Turning both relays on simultaneously would close Inputs 1 and 2 which may be the means of stopping the motor. This may only require momentary (pulse) action of the relay rather than latching.

Alternatively, the Control Module may simple require each input to be held closed for the duration of operation, and when the relay is then turned off, the motor stops. Programming will need to be adapted to ensure the correct method of operating the inputs is used.

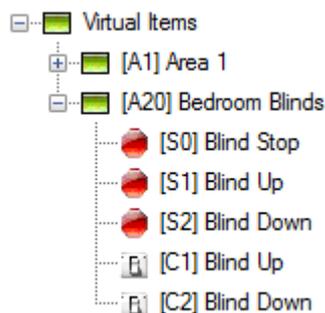
For a programming example based on the illustrated configuration, see page

## Programming Examples

### General Configuration of Output Device and Scenes

The configuration of outputs and scenes for blind control is common to all the control examples below.

You should first assign the blind(s) to their own virtual area, distinct from any lighting in the room. Two channels must be created, with each named according to the direction of travel. These are then assigned to the appropriate output. Naming the channels and scenes makes programming much easier to follow:



There are 3 scenes required. The Stop scene (both outputs off) is Scene 0 for this area and will be available by default. Two more scenes are required, one for each direction, with the channel levels configured accordingly:

#### Scene 1 (Up)

|                          |   |                          |            |  |     |                                 |                                       |                                |         |
|--------------------------|---|--------------------------|------------|--|-----|---------------------------------|---------------------------------------|--------------------------------|---------|
| <input type="checkbox"/> | 1 | <input type="checkbox"/> | Blind Up   |  | 100 | <input type="checkbox"/> Ignore | <input type="checkbox"/> Scene Modify | <input type="checkbox"/> Flash | Outputs |
| <input type="checkbox"/> | 2 | <input type="checkbox"/> | Blind Down |  | 0   | <input type="checkbox"/> Ignore | <input type="checkbox"/> Scene Modify | <input type="checkbox"/> Flash | Outputs |

#### Scene 2 (Down)

|                          |   |                          |            |  |     |                                 |                                       |                                |         |
|--------------------------|---|--------------------------|------------|--|-----|---------------------------------|---------------------------------------|--------------------------------|---------|
| <input type="checkbox"/> | 1 | <input type="checkbox"/> | Blind Up   |  | 0   | <input type="checkbox"/> Ignore | <input type="checkbox"/> Scene Modify | <input type="checkbox"/> Flash | Outputs |
| <input type="checkbox"/> | 2 | <input type="checkbox"/> | Blind Down |  | 100 | <input type="checkbox"/> Ignore | <input type="checkbox"/> Scene Modify | <input type="checkbox"/> Flash | Outputs |

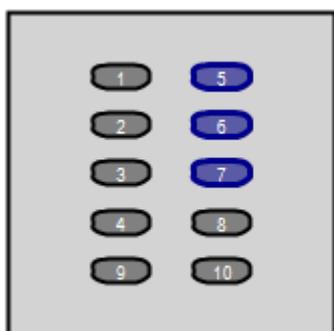
Outputs of the relay controller must be configured to switch on at 1% or higher, and off at 0%. This is achieved by setting the Minimum Level to 0, and the Maximum level to 1 for all outputs.

| General  |           | Outputs                        | Heartbeat |     |        |          |                    |
|----------|-----------|--------------------------------|-----------|-----|--------|----------|--------------------|
| Settings |           | Levels                         |           |     |        |          |                    |
|          | Name      | Channel                        | Min       | Max | Option | Curve    | Startup            |
| 1        | Channel 1 | [A20] Bedroom Blinds: [C1] ... | 0         | 1   | -      | Switched | Restore Last Scene |
| 2        | Channel 2 | [A20] Bedroom Blinds: [C2] ... | 0         | 1   | -      | Switched | Restore Last Scene |

With this configured, you can then proceed to program the input device to any of the control scheme examples below according to power / wiring method.

### 3 button Control for AC / DC Motors

This straightforward configuration uses 3 separate iLight inputs (e.g. Control Panel buttons), one for each direction, and a third input to stop the motor.

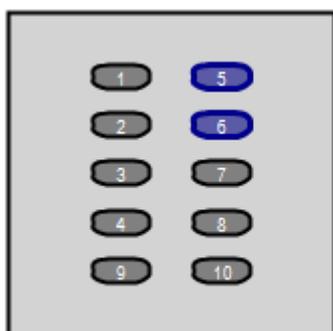


- 3)  No Action
- 4)  No Action
- 5)  [A20] Bedroom Blinds: [S1] Blind Up Fade 0
- 6)  [A20] Bedroom Blinds: [S0] Blind Stop Fade 0
- 7)  [A20] Bedroom Blinds: [S2] Blind Down Fade 0
- 8)  No Action

**NOTE:** All scene fade times must be set to 0s to allow instantaneous switching.

### 2 button Control for AC / DC Motors

This configuration reduces control to 2 separate iLight inputs (e.g. Control Panel buttons). In this example there is a button for each direction, and a second press of each button will either stop the motor or reverse the direction of travel.



- 3)  No Action
- 4)  No Action
- 5)  Sequence 1, Step 0
- 6)  Sequence 2, Step 0
- 7)  No Action
- 8)  No Action

Two sequences are required for this control scheme.

**NOTE:** All scene fade times must be set to 0s to allow instantaneous switching.

### Sequence 1

|                      |     |       |  |
|----------------------|-----|-------|--|
| 01 Blind Up / Stop   | 001 | 50ms  | Stop Sequence 2                              |
| 02 Blind Down / Stop | 002 | 50ms  | [A20] Bedroom Blinds: [S0] Blind Stop Fade 0 |
| 03 [Empty]           | 003 | 500ms | [A20] Bedroom Blinds: [S1] Blind Up Fade 0   |
| 04 [Empty]           | 004 | 50ms  | Pause Sequence 1                             |
| 05 [Empty]           | 005 | 50ms  | [A20] Bedroom Blinds: [S0] Blind Stop Fade 0 |
| 06 [Empty]           |     |       |  |

### Sequence 2

|                      |     |       |  |
|----------------------|-----|-------|--|
| 01 Blind Up / Stop   | 001 | 50ms  | Stop Sequence 1                              |
| 02 Blind Down / Stop | 002 | 50ms  | [A20] Bedroom Blinds: [S0] Blind Stop Fade 0 |
| 03 [Empty]           | 003 | 500ms | [A20] Bedroom Blinds: [S2] Blind Down Fade 0 |
| 04 [Empty]           | 004 | 50ms  | Pause Sequence 2                             |
| 05 [Empty]           | 005 | 50ms  | [A20] Bedroom Blinds: [S0] Blind Stop Fade 0 |
| 06 [Empty]           |     |       |  |

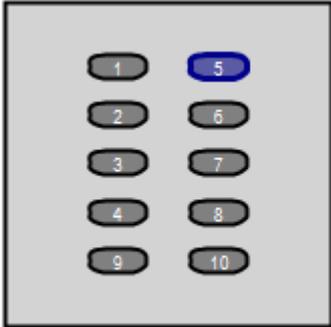
#### How it works:

- A. With the blind fully open (up), the user presses (and releases) button 6 to lower the blind. Sequence 2 is started from line 001. This stops sequence 1, to ensure it starts from line 001 when next triggered (and not from the line 004 pause position. The blind is stopped for 500ms and then starts going down. The sequence is paused at line 004.
- B. If the user wishes to stop the blind before it reaches the fully down position, they press button 6 again (the same direction of current travel). This resumes sequence 2 from line 004 which then stops the blind.
- C. If the user wishes to for the blind to continue in the same direction they press button 6 again which starts sequence 2 from line 001 (the sequence, having reached the end of the list of actions, is considered stopped).
- D. If the user does step A and B but then wants the blind to go back up again, they press button 5. This will start sequence 1 at line 001 which results in the Up scene being triggered.
- E. If the user does step A but wants the blind to reverse direction rather than stop, they press button 5 (the opposite direction of current travel). This starts sequence 1 at line 001 which results in the Up scene being triggered. This is where the Stop scene and 500ms delay comes into play. It ensures that the motor comes to a rest before reversing direction. Immediate direction reversal without stopping first can cause wear to motors and shorten their lifespan.

The sequence to trigger the opposite direction will always start from line 001 after the blinds start moving because it is stopped whenever the current direction is triggered. This prevents any double pressing being required to reverse the direction of travel.

### Single button Control for AC / DC Motors

This configuration reduces control to just a single iLight inputs (e.g. Control Panel button). In this example the button will toggle up / stop / down / stop etc. While this is useful in requiring only a single button for control, it necessarily follows that it is less intuitive to an end user as they will have to be in line site of the blind(s) to see what direction they are travelling. There can be a scenario where if the blind is left to reach its limiter, the next press will be to stop the blind, so a double press is required to get the blind to operate in the opposite direction.



- 3) No Action

---

- 4) No Action

---

- 5) Sequence 1, Step 0

---

- 6) No Action

---

- 7) No Action

---

- 8) No Action

The button resumes a single sequence from wherever it is paused, triggering the next action in the sequence.

|                            |  |
|----------------------------|--|
| 01 Blind Up/Stop/Down/Stop | 001 50ms [A20] Bedroom Blinds: [S0] Blind Stop Fade 0  |
| 02 [Empty]                 | 002 500ms [A20] Bedroom Blinds: [S1] Blind Up Fade 0   |
| 03 [Empty]                 | 003 50ms Pause Sequence 1                              |
| 04 [Empty]                 | 004 50ms [A20] Bedroom Blinds: [S0] Blind Stop Fade 0  |
| 05 [Empty]                 | 005 50ms Pause Sequence 1                              |
| 06 [Empty]                 | 006 50ms [A20] Bedroom Blinds: [S0] Blind Stop Fade 0  |
| 07 [Empty]                 | 007 500ms [A20] Bedroom Blinds: [S2] Blind Down Fade 0 |
| 08 [Empty]                 | 008 50ms Pause Sequence 1                              |
| 09 [Empty]                 | 009 50ms [A20] Bedroom Blinds: [S0] Blind Stop Fade 0  |

#### How it works:

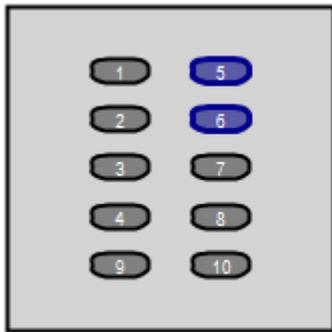
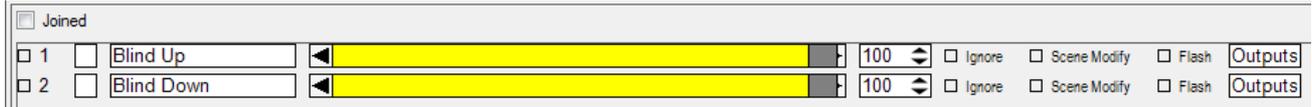
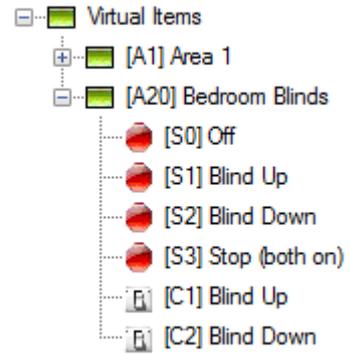
The user presses button 5 and scene 1 is triggered (the blind movement is delayed by 500s ensuring it is not moving already). The sequence is then paused at line 003. To stop the blind mid-travel, the user must press the button again, which resumes the sequence from line 003 and stops the blind. A third press will trigger the opposite direction. A fourth press stops the blind again.

To reverse the direction of the blind mid-travel, the user must double-tap the button, in doing so skipping the stop option. The stop scene with 500ms delay is in place prior to each direction to ensure the motor rests for sufficient time before reversing direction. Without this, the user could press the button fast enough to more-or-less instantaneously reverse direction. Immediate direction reversal without stopping first can cause wear to motors and shorten their lifespan.

## Momentary (Pulse) Switching to Volt-free Input

This configuration is based upon the previous 2 button example, but modified to generate a pulse (quick on/off) action rather than a latching relay action.

Assuming that there a requirement to have both inputs simultaneously closed to stop the blind, there would be the need to create another scene which has both outputs set to 100%.



- 3) No Action
- 4) No Action
- 5) Sequence 1, Step 0
- 6) Sequence 2, Step 0
- 7) No Action
- 8) No Action

As per the 2 button control example, two sequences are required. The sequences are modified to change the stop command from Scene 0 to Scene 3 and after each start or stop command, Scene 0 is activated (after 200ms) to set the relays to the off position. The 200ms delay can be considered as de-bounce setting, and adjusted accordingly if there is any requirement for the input to be held closed for a minimum amount of time.

### Sequence 1

|                      |     |       |  |
|----------------------|-----|-------|--|
| 01 Blind Up / Stop   | 001 | 50ms  | Stop Sequence 2                                  |
| 02 Blind Down / Stop | 002 | 50ms  | [A20] Bedroom Blinds: [S3] Stop (both on) Fade 0 |
| 03 [Empty]           | 003 | 500ms | [A20] Bedroom Blinds: [S1] Blind Up Fade 0       |
| 04 [Empty]           | 004 | 200ms | [A20] Bedroom Blinds: [S0] Off Fade 0            |
| 05 [Empty]           | 005 | 50ms  | Pause Sequence 1                                 |
| 06 [Empty]           | 006 | 50ms  | [A20] Bedroom Blinds: [S3] Stop (both on) Fade 0 |
| 07 [Empty]           | 007 | 200ms | [A20] Bedroom Blinds: [S0] Off Fade 0            |

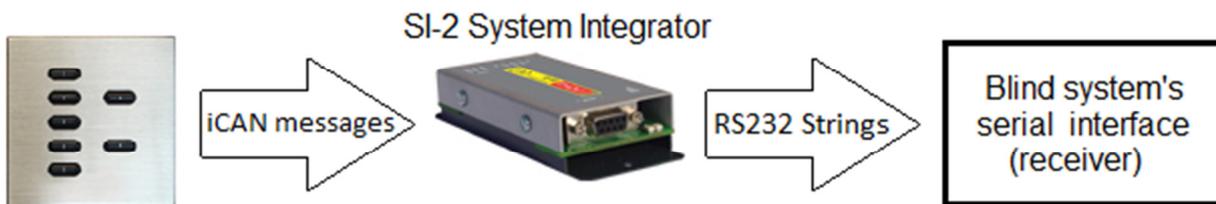
### Sequence 2

|                      |     |       |  |
|----------------------|-----|-------|--|
| 01 Blind Up / Stop   | 001 | 50ms  | Stop Sequence 1                                  |
| 02 Blind Down / Stop | 002 | 50ms  | [A20] Bedroom Blinds: [S3] Stop (both on) Fade 0 |
| 03 [Empty]           | 003 | 500ms | [A20] Bedroom Blinds: [S2] Blind Down Fade 0     |
| 04 [Empty]           | 004 | 200ms | [A20] Bedroom Blinds: [S0] Off Fade 0            |
| 05 [Empty]           | 005 | 50ms  | Pause Sequence 2                                 |
| 06 [Empty]           | 006 | 50ms  | [A20] Bedroom Blinds: [S3] Stop (both on) Fade 0 |
| 07 [Empty]           | 007 | 200ms | [A20] Bedroom Blinds: [S0] Off Fade 0            |

## Serial Integration Example

A blind system may require serial strings to be sent to a control module to trigger the motors to move. The syntax for the messages that need to be received should be provided in the blind manufacturer's literature. It may be possible to send commands to control one or multiple groups of blinds according to the capabilities of the blind system.

Either way, the iLight SI-2 System Integrator is able to be programmed to output ASCII, Hex or Decimal format serial data. Up to 100 custom output strings can be created and associated with a corresponding output action, which is a message broadcast from an input device on the iLight system. Whenever that particular unique message is heard by the SI-2, the output serial string is forwarded from the SI-2 to the 3<sup>rd</sup> party device.



The same control schemes used in the previous examples could be used for configuring the input devices, but instead of the scene messages controlling relay outputs, these scene messages could be programmed in the SI-2 to trigger Output Strings of a syntax recognized by the blind system's serial interface.

In this example, the scene messages are added to the SI-2 Output Actions list:

| General  |  | Strings       | Sequences      | Diagnostics    |
|--|--|---------------|----------------|----------------|
| ASCII Options: <input type="checkbox"/> Rx <input type="checkbox"/> Tx |  |               |                |                |
| Input Actions  |  | Input Strings | Output Actions | Output Strings |
|  | Action                                       | Notes         |                |                |
| 1  | [A20] Bedroom Blinds: [S0] Stop Fade 0       |               |                |                |
| 2  | [A20] Bedroom Blinds: [S1] Blind Up Fade 0   |               |                |                |
| 3  | [A20] Bedroom Blinds: [S2] Blind Down Fade 0 |               |                |                |
| 4  | No Action                                    |               |                |                |

The corresponding Output Strings are then configured with the message that needs forwarding to the blind system's serial interface:

| General  |                 | Strings       | Sequences      | Diagnostics    |
|--|-----------------|---------------|----------------|----------------|
| ASCII Options: <input type="checkbox"/> Rx <input type="checkbox"/> Tx |                 |               |                |                |
| Input Actions  |                 | Input Strings | Output Actions | Output Strings |
| <input type="button" value="Clear Strings"/>                           |                 |               |                |                |
|  | Name            | Data          | Format         |                |
| 1  | Blind Stop      | blind1@00<CR> | ASCII          |                |
| 2  | Blind Up        | blind1@01<CR> | ASCII          |                |
| 3  | Blind Down      | blind1@02<CR> | ASCII          |                |
| 4  | Output String 4 |               | ASCII          |                |

## Master Control Advice

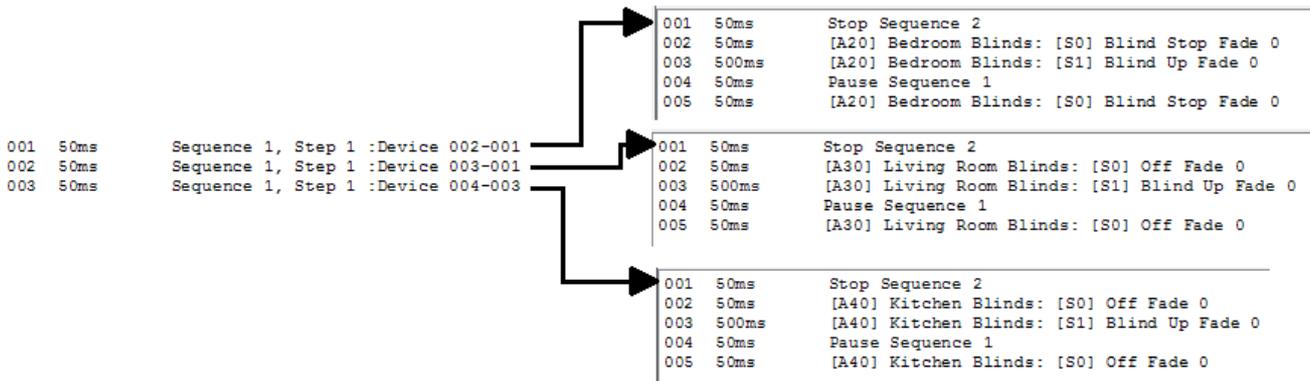
For master control multiple blinds operated by a 3 button Control example (page ) you would simply have a sequence that sends multiple scene messages to the different blind areas, as there are no sequences involved.

```

001 50ms [A20] Bedroom Blinds: [S1] Blind Up Fade 0
002 50ms [A30] Living Room Blinds: [S1] Blind Up Fade 0
003 50ms [A40] Kitchen Blinds: [S1] Blind Up Fade 0
  
```

For master control multiple blinds operated by a 2 button Control example (page ) you would write two sequences that contain the multiple areas to be operated, but also need to make sure that you send sequence control messages to set the individual local sequences to the appropriate line for synchronising these local controls, or they will not perform the expected action when subsequently used.

The reason you cannot remotely trigger the multiple sequences simply using a resume command ('Start Sequence at Line 0) for each local sequence, is because the blinds may be in different starting positions. Some could be fully up, their fully down, and some partially closed, which means it is unclear what will happen by resuming their local sequences from where currently paused. You must make it such that the Master control sequence definitely performs the up or down action expected by the user.



It is not recommended to adopt a single button configuration for a Master blind control switch as it is not clear what the blinds will do when this button is pressed and you must make it such that the operator can decide if the blinds are to all go up or down.

The best way to implement this would be with a 2-button Master control scheme as shown above, which starts the local sequences at the appropriate line to move the blinds in the desired direction. Not only does this provide a clear action when either button is pressed, by manipulating the existing local sequences, the toggle action of the local switches is kept up-to-date with the last action of the master sequence.

## Troubleshooting & FAQs

### What is the maximum load that can be connected to the relay outputs?

The RI-2 relay outputs are rated at 3A. The SCMC0410 din rail module outputs are rated at 10A.

### Is it possible to connect AC motors in parallel?

Normally it is not possible to connect AC mains powered motors in parallel as a group. Advice must be sought from the motor system's manufacturer about specific installation requirements.

### Can you control the blinds via the iLight Remote Smartphone app?

Yes, this is just a case of programming custom buttons in the app to operate the scenes or sequences related to the blinds.

### In the event of a fire alarm, can the blinds be made to open?

Yes, you would need to check the Alarm Scene setup for the blind control in the Scene Editor and make sure that the channels are set so that the blinds are opened when the Alarm Scene is activated.