

**Elevator Trainer
34-150**



Feedback



Electricity & Electronics



Control & Instrumentation



Process Control



Mechatronics



Telecommunications



Electrical Power & Machines



Technology Training for tomorrow's world

Elevator Trainer

34-150



Feedback Instruments Ltd, Park Road, Crowborough, E. Sussex, TN6 2QR, UK.
Telephone: +44 (0) 1892 653322, Fax: +44 (0) 1892 663719.
email: feedback@fdbk.co.uk website: <http://www.fbk.com>

Manual: 34-150 Ed03 012008

Printed in England by FI Ltd, Crowborough

Feedback Part No. 1160-34150

Notes



THE HEALTH AND SAFETY AT WORK ACT 1974

We are required under the Health and Safety at Work Act 1974, to make available to users of this equipment certain information regarding its safe use.+

The equipment, when used in normal or prescribed applications within the parameters set for its mechanical and electrical performance, should not cause any danger or hazard to health or safety if normal engineering practices are observed and they are used in accordance with the instructions supplied.

If, in specific cases, circumstances exist in which a potential hazard may be brought about by careless or improper use, these will be pointed out and the necessary precautions emphasised.

While we provide the fullest possible user information relating to the proper use of this equipment, if there is any doubt whatsoever about any aspect, the user should contact the Product Safety Officer at Feedback Instruments Limited, Crowborough.

This equipment should not be used by inexperienced users unless they are under supervision.

We are required by European Directives to indicate on our equipment panels certain areas and warnings that require attention by the user. These have been indicated in the specified way by yellow labels with black printing, the meaning of any labels that may be fixed to the instrument are shown below:



CAUTION -
RISK OF
DANGER



CAUTION -
RISK OF
ELECTRIC SHOCK



CAUTION -
ELECTROSTATIC
SENSITIVE DEVICE

Refer to accompanying documents

PRODUCT IMPROVEMENTS

We maintain a policy of continuous product improvement by incorporating the latest developments and components into our equipment, even up to the time of dispatch.

All major changes are incorporated into up-dated editions of our manuals and this manual was believed to be correct at the time of printing. However, some product changes which do not affect the instructional capability of the equipment, may not be included until it is necessary to incorporate other significant changes.

COMPONENT REPLACEMENT

Where components are of a 'Safety Critical' nature, i.e. all components involved with the supply or carrying of voltages at supply potential or higher, these must be replaced with components of equal international safety approval in order to maintain full equipment safety.

In order to maintain compliance with international directives, all replacement components should be identical to those originally supplied.

Any component may be ordered direct from Feedback or its agents by quoting the following information:

- | | |
|------------------------|----------------------------|
| 1. Equipment type | 2. Component value |
| 3. Component reference | 4. Equipment serial number |

Components can often be replaced by alternatives available locally, however we cannot therefore guarantee continued performance either to published specification or compliance with international standards.



OPERATING CONDITIONS

WARNING:

This equipment must not be used in conditions of condensing humidity.

This equipment is designed to operate under the following conditions:

Operating Temperature	10°C to 40°C (50°F to 104°F)
Humidity	10% to 90% (non-condensing)

CE DECLARATION CONCERNING ELECTROMAGNETIC COMPATIBILITY

Should this equipment be used outside the classroom, laboratory study area or similar such place for which it is designed and sold then Feedback Instruments Ltd hereby states that conformity with the protection requirements of the European Community Electromagnetic Compatibility Directive (89/336/EEC) may be invalidated and could lead to prosecution.

This equipment, when operated in accordance with the supplied documentation, does not cause electromagnetic disturbance outside its immediate electromagnetic environment.

COPYRIGHT NOTICE

© **Feedback Instruments Limited**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Feedback Instruments Limited.

ACKNOWLEDGEMENTS

Feedback Instruments Ltd acknowledge all trademarks.

IBM, IBM - PC are registered trademarks of International Business Machines.

MICROSOFT, WINDOWS XP, WINDOWS 2000, WINDOWS ME, WINDOWS NT, WINDOWS 98, WINDOWS 95 and Internet Explorer are registered trademarks of Microsoft Corporation.



WARNINGS:

1. This equipment is shipped with a safety bracket and a counter balance weight securing screw, to prevent damage in transit. The bracket and screw must be removed before the equipment is used.
2. This equipment must use an earthed mains connection.
3. Transparent covers are fitted to the top and front of the unit to prevent the user becoming caught in the moving mechanism. The top cover also prevents accidental contact with a high power resistor which may become very hot during normal use.

Do not operate the equipment with the covers removed.



ELEVATOR TRAINER

Warnings

Notes



TABLE OF CONTENTS

1	Introduction	1-1
1.1	Overview and basic features	1-2
1.2	Technical Detail	1-2
2	Initial Set-up of Equipment	2-1
2.1	Checking the interface with the PLC	2-1
2.2	Basic control sequences	2-1
3	Assignments	3-1
4	Assignment 1: Familiarisation	4-1
4.1	Objectives	4-1
4.2	Practical 1: Elevator Hardware	4-1
4.3	Practical 2: Connecting the SLC Controller to the Elevator	4-1
4.4	Practical 3: Using the PLC Software	4-2
5	Assignment 2: PLC Fundamentals	5-1
5.1	Objectives	5-1
5.2	Practical 1: A Simple Program	5-1
5.3	Practical 2: Latches	5-1
5.4	Practical 3: AND, OR and NOT Conditions	5-2
5.5	Practical 4: Using a Timer	5-3
5.6	Practical 5: Using a Timer to Control the Bell	5-4
5.7	Practical 6: Master Control Relays	5-5



6	Assignment 3: Elevator Initialisation	6-1
6.1	Objectives	6-1
6.2	Practical 1: Initialisation	6-1
7	Assignment 4: Basic Elevator Control	7-1
7.1	Objectives	7-1
7.2	Practical 1: Manual Car Control	7-1
7.3	Practical 2: Door and Bell Operation	7-2
7.4	Practical 3: Moving the Car to the Top	7-2
7.5	Practical 4: Move Car to Top then to the Bottom	7-4
7.6	Practical 5: Creating a Loop using a Timer	7-4
8	Assignment 5: Subroutines	8-1
8.1	Objectives	8-1
8.2	Practical 1: Creating Subroutines	8-1
8.3	Practical 2: Initialisation as a Subroutine	8-1
8.4	Practical 3: Showing the Current Level	8-2
9	Assignment 6: A Two Level Elevator	9-1
9.1	Objectives	9-1
9.2	Practical 1: A Two Level Elevator	9-1
9.3	Practical 2: Two Level Elevator with Door Control	9-1
10	Assignment 7: Visiting Floors	10-1
10.1	Objectives	10-1
10.2	Practical 1: Choosing the Correct Floor Sensor	10-1
10.3	Practical 2: Stopping at Every Floor - Moving Up	10-1



ELEVATOR TRAINER

Contents

10.4	Practical 3: Stopping at Every Floor - Moving Down	10-3
10.5	Practical 4: Stopping at Every Floor - Both Directions	10-3
10.6	Practical 5: Every Floor with Arrival Routine	10-3
11	Assignment 8: Processing Inputs	11-1
11.1	Objectives	11-1
11.2	Practical 1: Deciding on the Direction of Travel	11-1
12	Assignment 9: Visiting Floors using Pushbutton Control	12-1
12.1	Objectives	12-1
12.2	Practical 1: Visiting all Floors using Request Pushbuttons	12-1
12.3	Practical 2: A Constantly Moving Elevator	12-2
13	Assignment 10: A Working Elevator	13-1
13.1	Objectives	13-1
13.2	Practical 1: Determining the Highest and Lowest Levels	13-1
13.3	Practical 2: Full Control (Simple Floor Requests)	13-1
13.4	Practical 3: Full Control (Directional Floor Requests)	13-2
13.5	Practical 4: Full Control (All Floor Controls)	13-3
14	Assignment 11: Advanced Elevator Programming	14-1
14.1	Objectives	14-1
14.2	Practical 1: Priority for Top Level	14-1
14.3	Practical 2: Enhanced Elevator Control	14-1
14.4	Practical 3: Multi-Speed Elevator Control	14-1
15	Appendix A Specification	15-1



ELEVATOR TRAINER

Contents

15.1	Features	15-1
15.2	Dimensions	15-1
15.3	Internal Power Supply:	15-2
15.4	Environment	15-2
16	Appendix B External Controller Interface	16-1
17	Appendix C Interfacing with Allen-Bradley SLC-500 Series	17-1
18	Appendix D Interfacing with Allen-Bradley MicroLogix 1000 Series	18-1
19	Appendix E Interfacing with Mitsubishi FXo-30MR-ES	19-1
20	Appendix F Fault Switching Settings	Error! Bookmark not defined.
21	Appendix G Routine Maintenance	21-1
21.1	Refitting the chain	21-1
21.2	Load measurement offset adjustment	21-2
21.3	Position pot offset adjustment	21-2
21.4	Output protection fuse	21-3



1 Introduction

The 34-150 Elevator provides a sophisticated application with which students may study the use and programming of Programmable Logic Controllers (PLCs). The advanced features of the elevator also allow complex analogue and digital control techniques to be investigated.



Figure 1-1

Starting with simple program sequences to control the elevator car speed, direction and floor arrival/departure, the student can progress to advanced floor request handling and continuous (analogue) control with acceleration profiling and compensation for varying car loads. The elevator requires an external controller (supplied by the customer). The flexible electrical interface will easily connect to a range of PLCs.

A set of exercises are provided to familiarise the student with the equipment and take them step by step through the process of writing a control program. A number of sample programs are provided which cover most of the practicals presented in this manual.



1.1 Overview and basic features

The motorised elevator car can be controlled (using an external controller) to move between four floors in response to requests from call pushbuttons located at each floor and 'within' the elevator car (these are located on an external panel). The car door is motorised so on arrival at a floor it can be opened to allow passengers to enter/leave. Logic sensors are located on each floor to indicate the presence of the car. These sensors also provide advance warning of arrival as the car approaches each floor to allow the car to be slowed prior to arriving at the destination floor.

The internal motor controller unit provides closed-loop motor speed and current control and has both logic and analogue interfaces. This allows the elevator to be controlled by a basic PLC using only digital I/O (a minimum of 16 inputs and 16 outputs is recommended) as a simple digital output can be used to request a fixed car speed. More sophisticated control may be implemented if analogue I/O is available, as the elevator interface provides signals for motor speed and current demand, motor speed feedback and car position feedback. For example, the integral motor speed loop may be opened and an external controller may be used to implement the speed loop externally instead. A continuous position signal is provided to locate the position of the car throughout its travel.

A load-cell is incorporated into the elevator car which gives an analogue output proportional to force and hence the mass of the car under static conditions. By adding or removing weights (supplied), the effect of varying passenger loads can be investigated.

1.2 Technical Detail

1.2.1 Drive motor arrangement

The elevator car is raised and lowered by a dc motor (referred to as the 'winder motor') via a 200:1 gearbox which is coupled to a toothed drive sprocket (see Figure 1-2). One end of a toothed drive belt is connected to the car and the other end is connected to a counterbalance weight which balances the car at about 40% load.

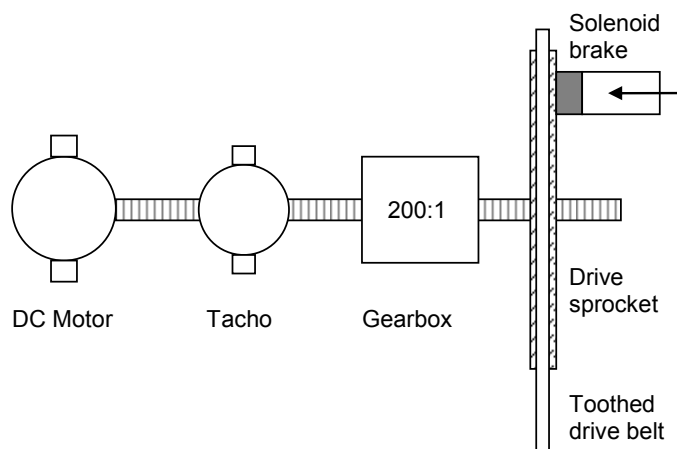


Figure 1-2: Drive motor arrangement



The winder motor is driven by a fully protected power amplifier with an integral current control so if the motor is stalled, current is limited to a safe maximum value.

A mechanical brake (driven by a solenoid) operates on the winder motor so that the car can be held in position. The brake is normally applied when the brake release input is held low or left floating and also when the unit is unpowered. Driving the brake release input high releases the brake.

1.2.2 Integral speed and current control

A tacho is coupled to the winder motor (before the gearbox) and used to drive an integral closed-loop speed controller (see Figure 1-3). The external speed demand can be in the form of a continuous analogue signal or a fixed speed demand (e.g. full speed, half speed).

The tacho output is made available externally either for monitoring or control purposes. A changeover switch is incorporated into the control circuit (located on the elevator rear panel) to select between speed control and current control mode. By selecting the 'current' (I) position of the changeover switch, the internal speed control loop is opened and a current demand may be input. In this case the speed control loop must be implemented using an external controller.

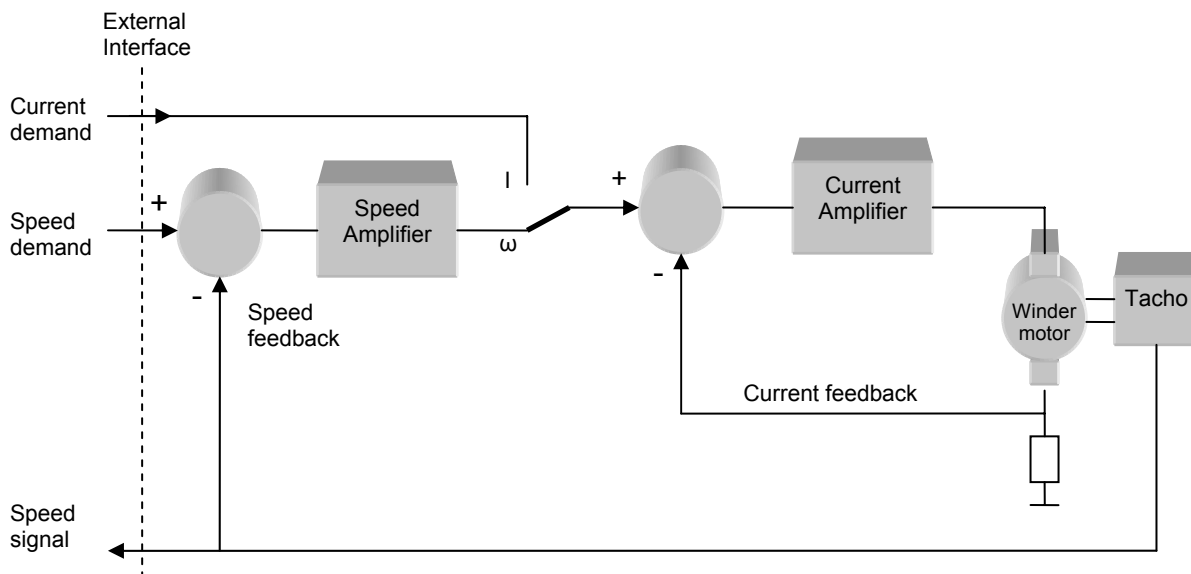


Figure 1-3

1.2.3 Elevator car

The elevator car door is motorised and can be operated using the 'door control' signal input. Taking this input high opens the door. Taking it low or leaving it floating closes the door. The door motor is controlled by an internal circuit which responds to the door control input by opening or closing the door as required and then removes power from the door motor automatically. Microswitches detect the door status and 'door open'/'door closed' signals are made available at the rear panel for use by the external controller.

Visual indication of the position of the elevator car as it arrives at each floor is provided by LEDs on the front of the unit. These LEDs are driven automatically by the internal controller. Front panel



LEDs in the shape of arrows are provided to indicate the direction of travel of the car. These LEDs must be driven by the external controller.

A set of weights is provided that can be placed inside the car to simulate varying passenger loads. A peg is located inside the car on which to place the weights to centralise them.

The car interior is illuminated by LEDs which are permanently on.

1.2.4 Call pushbuttons

Elevator call pushbuttons are provided for each floor with separate up/down pushbuttons for Floor 2 and Floor 3. This enables the external controller to differentiate between requests for upward or downward travel and allows more sophisticated algorithms to be implemented. A separate set of floor request pushbuttons is provided on the front panel to represent the request pushbuttons inside the car.

If the external controller has a limited number of digital input points, the floor request pushbuttons can be connected together. For example, Floor 2 up and down request pushbuttons could be commoned with Car Floor 2 request pushbutton so that only a single input point is used rather than three.

1.2.5 Logic control

The elevator is designed so that it can be controlled using only logic signals if required. In this case, the car movement can be controlled using a subset of five logic inputs: 100% speed, 60% speed, 30% speed, direction and brake release. Typically only two speeds will be used: 60% for 'full speed' and 30% when approaching a floor. At 100% speed the car travels at approximately 0.1 m/s and takes approximately 7 seconds to travel from Floor 1 to Floor 4.

The sequence of operations is as follows: the direction input should be set first to select the direction of travel (up or down), the brake is released then one of the speed inputs can be driven high to select the desired speed. The car will move with the commanded speed and direction. When it approaches a floor, the floor sensor signal will become active and this can be used by the external controller to slow the car by driving the 30% speed input. As the car continues towards the floor the floor sensor will go inactive. Once the car arrives at the floor, the floor sensor will become active again at which point the car should be stopped and the brake applied. Even using simple logic-only control, the internal high performance speed control ensures that the car changes speed precisely and requesting zero speed will stop the car rapidly and hold it stationary. When using logic-only control it is recommended that 60% speed is used as full speed and 30% speed when approaching a floor.

The car is prevented from travelling above Floor 4 or below Floor 1 by physical end-stops. Microswitches are triggered when the car reaches the end-stops which reduce the motor power amplifier output so the unit is protected. Front panel LEDs indicate when the travel limits have been reached.

A 'manual' control is provided on the front panel which allows the car to be manually driven up and down at 60% speed (the brake is released automatically). This is particularly useful during testing and debugging as it allows the car to activate all floor sensors in turn to check that they are being read correctly by the external controller.



1.2.6 Analogue control

If an external controller is available with analogue I/O capability, more sophisticated forms of control may be implemented. The external controller can drive the car with an analogue speed demand signal which will allow acceleration and deceleration profiling. A rear panel changeover switch allows a current demand to be input directly to the power amplifier, bypassing the internal speed loop. By using the motor speed feedback signal provided, the external controller can implement its own speed control loop.

The elevator interface provides an analogue position feedback signal which identifies the position of the car throughout its travel. This can be used by the external controller to implement a position control system for example, or to drive a mimic showing car movement. A load-cell is incorporated into the car assembly to provide continuous measurement of car loading. The conditioned load signal is available at the interface and a trim pot is provided to zero the offset. The external controller could use the load signal to implement a form of control based on passenger load. Statically the load output signal is proportional to force $F=mg$ where m is the mass of the car plus passenger load and g is the acceleration due to gravity. As the car accelerates and decelerates the output becomes $F=m(g+A)$ where A is the acceleration of the car, hence this signal may be used as a measure of car acceleration.

Two weights are provided which can be placed on a mounting peg inside the car to simulate varying passenger loads. The counterbalance has a mass of 1.35 kg and the empty car has a mass of 1.0 kg. With no weights added the car has a net mass of -0.35 kg, with one weight it has a net mass of approximately zero and with two weights added it has a net mass of +0.35 kg.

1.2.7 Other features

Top and front transparent covers are fitted to prevent objects becoming trapped in the moving mechanism. The front cover is easily removable to allow access to change the internal car weights. A microswitch is fitted to prevent the unit from being operated accidentally with the front cover removed.

Two units may be mounted side by side to form a dual elevator. Fixing holes are located in the side panels to allow the units to be bolted together. Dual elevator control is considerably more complex and opens up a wide range of control strategies, for example quickest response, minimum energy usage, floor priorities etc. Up to twice the number of controller I/O points will be required, depending on which floor call pushbuttons are commoned together.

A set of fault conditions can be selected using a DIP switch located on the rear panel. The switch is labelled only by switch number so it possible to set fault-finding exercises and students will not be able to work out the fault from the switch setting. The faults are listed Appendix F.

A chime is provided to announce arrival of the elevator car at a requested floor. The chime input is driven by the external controller and requires a pulse of a few hundred milliseconds to cause it to sound once. Longer pulses will cause the chime to sound repetitively and this can be used as an audible warning, for example on detection of a fault by the external controller.

Support is provided for both Allen-Bradley and Mitsubishi PLCs.



1.2.8 Interface with external controller

The elevator is connected to the external controller via a set of connectors located on the rear panel, see Figure 1-4. Stranded insulated cable 7/0.2 or similar is recommended.

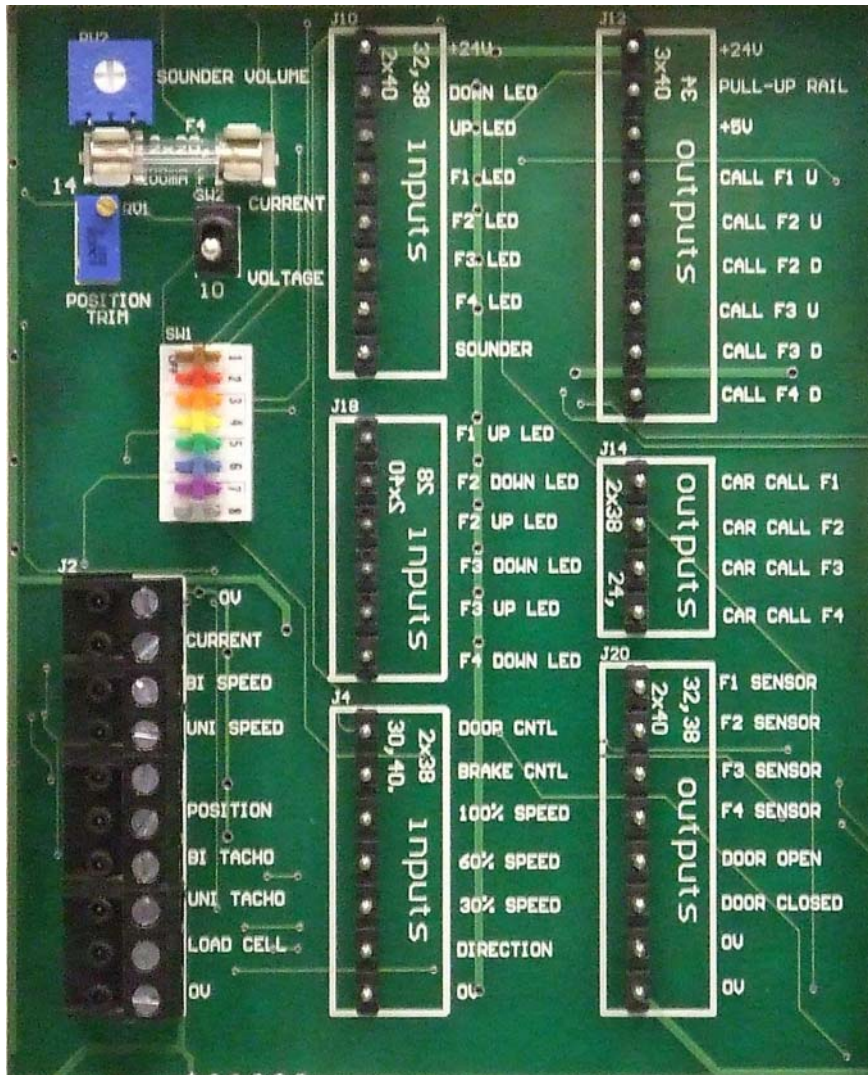


Figure 1-4

Although it is not mandatory to follow the connection instructions given below, they are recommended if you wish to use any of the example programmes supplied with the equipment. Using any other connection configuration will require the I/O in the example programmes to be changed.

You will need a small screwdriver to make the connections.

A summary of the input/output signals used to interface with the external controller are listed in Table 1-1. All signals are made available on pluggable terminal blocks which are located on the rear panel.



Table 1-1: External signal summary

PLC Digital Inputs	PLC Digital Outputs	PLC Analog Inputs	PLC Analog Outputs
Elevator Outputs	Elevator Inputs	Elevator Outputs	Elevator Inputs
Connector J12	Connector J10	Connector J2	Connector J2
Floor 1 up call pushbutton	Car going down indicator	Car position	Motor current demand
Floor 2 up call pushbutton	Car going up indicator	Car speed (Bi)	Car speed demand (BI)
Floor 2 down call pushbutton	Destination Floor 1 (car indicator)	Car speed (Uni)	Car speed demand (Uni)
Floor 3 up call pushbutton	Destination Floor 2 (car indicator)	Car passenger load	
Floor 3 down call pushbutton	Destination Floor 3 (car indicator)		
Floor 4 down call pushbutton	Destination Floor 4 (car indicator)		
	Chime		
Connector J14	Connector J18		
Car Floor 1 call pushbutton	Car 'on its way' Floor 1 up LED		
Car Floor 2 call pushbutton	Car 'on its way' Floor 2 down LED		
Car Floor 3 call pushbutton	Car 'on its way' Floor 2 up LED		
Car Floor 4 call pushbutton	Car 'on its way' Floor 3 down LED		
	Car 'on its way' Floor 3 up LED		
	Car 'on its way' Floor 4 down LED		
Connector J20	Connector J4		
Floor 1 sensor	Open/close car door		
Floor 2 sensor	Brake release		
Floor 3 sensor	Car 100% speed		
Floor 4 sensor	Car 60% speed		
Car door open	Car 30% speed		
Car door closed	Car direction up/down		

All digital inputs may be driven active using a dc input voltage in the range 3 V to 24 V. This provides a flexible interface which will function with a range of PLCs.



All digital outputs are implemented using open collector transistors connected internally to a common pull-up rail which is made available on an external connector. This pull-up rail may be connected to an external supply ranging from 3 V to 24 V dc to allow interface with a wide range of external controllers. 5 V and 24 V outputs are made available on the rear panel for use as pull-up supplies.

The analogue car speed demand input can use a bipolar ± 10 V signal or a unipolar 0+10 V (selected by DIP switch (SW1, switch No2 located on the rear panel). The motor current demand input uses a bipolar ± 10 V signal.

The analogue car speed output (tacho) is available in both unipolar 0+10 V and bipolar ± 10 V form on two separate outputs.

The analogue car passenger load signal output is a bipolar ± 10 V signal and its offset can be trimmed so that the output is zero with no passenger load. Refer to Appendix G.

For detailed information on the I/O including connections, logic sense and signal scaling information, refer to Appendix B.

For recommended connections with the Allen-Bradley SLC-500 series, refer to Appendix C. Connection details for the Allen-Bradley MicroLogix 1000 series are given in Appendix D and for the Mitsubishi FXo-30MR-ES (or equivalent) in Appendix E.

Minimum PLC specification

The 34-150 Elevator can be interfaced with a range of PLCs due to its flexible interface. The minimum I/O recommended is 12 digital inputs and 16 digital outputs.



2 Initial Set-up of Equipment

Ensure that the transit bracket used to safely ship the elevator has been removed.

Ensure that the toothed belt is fitted over the drive sprocket correctly. If not, refer to Appendix G for fitting instructions.

Ensure that the front cover is in place then plug in the unit and apply power. Note that this equipment must use an earthed mains connection. Switch on the front panel power switch and check that the power LED on the front panel illuminates.

Operate the 'Manual' switch and check that the car moves up and down freely. The unit is now ready for use.

2.1 Checking the interface with the PLC

Connect the elevator to the PLC as described in Appendix C, D or E as appropriate. To confirm that the connections have been made correctly, the easiest method is to use the diagnostic capabilities of the PLC. The Allen-Bradley PLCs have indicator LEDs which can be used to view the state of each input as it is changed. For example the floor request pushbuttons can be pressed and released; the floor indicators can be changed by causing the car to pass a floor using the Manual control. The same information can be obtained using Allen-Bradley RSLogix software. By going online and selecting the Input table, the state of each input can be viewed as it is changed. The outputs can be tested using the forcing function. By selecting the Output table and enabling Forces in the processor, each individual output can be forced into a desired state to confirm that it is connected correctly. This process can be used with the SLC-500 and MicroLogix family of PLCs.

2.2 Basic control sequences

There are four basic sequences required to control car movement: initialisation, depart from a floor, arrive at a floor, and handle floor requests. These are briefly described below.

Initialisation

The external controller must be able to identify the position of the car in order to determine in which direction to move. When using a logic-only controller which does not receive continuous position information from the elevator, an initialisation routine must be executed when the controller runs for the first time to identify the starting position.

Arrive at a floor

The sequence of operations is to sense arrival, slow down and stop. The brake is then applied and the door opened. The steps are as follows:

Poll the destination floor sensor to detect the approach signal

When the signal is received, reduce car speed to minimum

Poll the destination floor sensor to detect the arrival signal

When the signal is received, set the speed to zero and apply the brake

Sound the arrival chime



Turn off floor call LEDs

Turn off up/down arrow LEDs

Open the car door

Depart from floor

The departure sequence is as follows:

Close the car door

Set the motor direction

Turn on the appropriate up/down arrow LEDs

Release the brake

Set the motor to the desired speed

Floor request handling

Handling floor requests can be a complex task. Multiple requests may be made simultaneously and requests may also occur whilst the car is in transit, possibly requiring intermediate stops. A simple handler algorithm is described below.

Register the first floor request that is received and ignore any further requests. Go to the floor selected by the passengers (via the car floor request switch), then poll for further requests.

Poll for requests on floor call and car switches

Register the first request received and ignore any further requests

Turn on the floor acknowledge LED for the appropriate request switch

Determine the current position of the car (if not available, perform initial floor seek)

If not at correct floor, perform *depart from floor* routine

Perform *arrive at floor* routine

Allow passengers to leave/enter the car

Poll only the car floor request switches for the destination floor

If no request within a set time, close the door and return to start of sequence

If car request received, perform *depart from floor* routine

Perform *arrive at floor* routine

Poll for further requests on car switches only

If no requests within a set time, close the door and return to start of sequence

2.2.1.1 Advanced floor request handling

An advanced form of request handling would poll all request switches continuously and determine the floor calling sequence based on an algorithm such as order of convenience, floor priority or minimising energy usage.



3 Assignments

This section takes the user step by step through the process of developing a programme to control the elevator. It is assumed that the user has a reasonable knowledge of Allen-Bradley RSLogix software and Mitsubishi Melsec software as appropriate. Beginning with basic interaction with the elevator I/O, the assignments progress to develop routines for arrival, departure etc. as outlined in the previous section, finally building these into a complete elevator control programme.

The following software packages were used to develop the sample PLC programs:

RSLogix 500 version 6.00

Mitsubishi Melsec GP version 8.25b

Please note that there are PDF copies of the Micrologix software in a folder marked **Micrologix sample programs** on the CD copy of this manual. These will assist you in your PLC programming, particularly useful if you are not using the Micrologix PLC and wish look at the structure of the program.



Notes



4 Assignment 1: Familiarisation

4.1 Objectives

- To learn about the elevator hardware.
- To learn how to connect the PLC to the elevator.
- To learn how to load a sample program into the PLC and start the program.
- To learn how to run the elevator test routine.

Some of the instructions within this assignment will depend on the particular controller you are using.

4.2 Practical 1: Elevator Hardware

This practical is to learn about the elevator hardware.

- Identify the components:
 - Floor pushbuttons
 - Elevator car pushbuttons
 - Floor and level indicators
 - Jog (car control) and travel indicators

Turn on power to the elevator using the switch at the front. Move the elevator car to the top of the shaft using the jog switch. Notice how the floor indicators change. When the elevator car is approaching a level, the level light will flash on and then turn off again. When the car is at the level it will turn on.

If you move the car above a level, either using the jog pushbutton or under control of the PLC the upper travel light will be displayed. If the limit is reached you will not damage the elevator.

If the jog pushbutton is used, the brake is automatically released. When the jog pushbutton is released, the brake is applied. The brake and operation of the elevator motor will be studied in later practicals.

4.3 Practical 2: Connecting the SLC Controller to the Elevator

Use the wiring diagram found in the appendix. The available number of inputs and outputs may differ depending on the model of the PLC you are using.

Some PLCs can display whether an input signal is active. On the SLC-500 this is achieved using a set of LED displays. If other controllers are used, the inputs may be visible using software.

When all inputs have been connected, press one of the elevator car pushbuttons and observe whether a corresponding signal can be seen by the PLC.



Connect the PLC to the PC using the supplied serial cable.
Configuration that is used by the sample programs:

- Slot 0 – 1747-L531E SLC 5/03 Processor
- Slot 1 – 1746-IV16 16-bit input (sink)
- Slot 2 – 1746-OB16 16-bit output transistor output
- Slot 4 – Unused

4.4 Practical 3: Using the PLC Software

The software you use to load programs to the controller depends on the type of controller you are using.

4.4.1 Allen-Bradley SLC-500 and Micrologix 1000

Loading the software into the SLC-500 controller, turning the key and starting the controller.

The following instructions are specific to the Allen-Bradley/Rockwell Automation software.

Two packages are required to be installed on the PC: RsLinx and RsLogix. RsLinx is the interface between the PC and the PLC - it acts as a gateway between the two. It is used to tell the RsLogix software which controllers can be used. Two sets of instructions are described below and on the following pages. The first describes how to establish a connection between the PLC and the controller. The second describes how to download a new program to the PLC.

4.4.1.1 Checking the link to the controller

Start RsLinx by clicking on the RsLinx icon found in the system tray. It contains a program called RSWho which searches for a list of all the controllers that are connected to your computer.

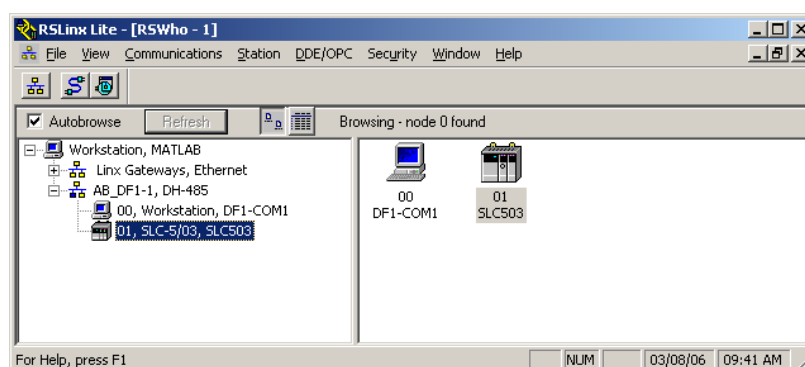


Figure 4-1



Right clicking on a menu option will allow a controller to be configured. A useful 'auto configure' option allows the correct baud and parity settings to be determined.

The following driver is used: AB-DF1-1, DH-485

This describes the protocol that the computer uses to communicate to the PC.

4.4.1.2 Downloading a program to the controller

Start the RSLogix software.

Using the File...Open menu option and select the program file (RSS file).

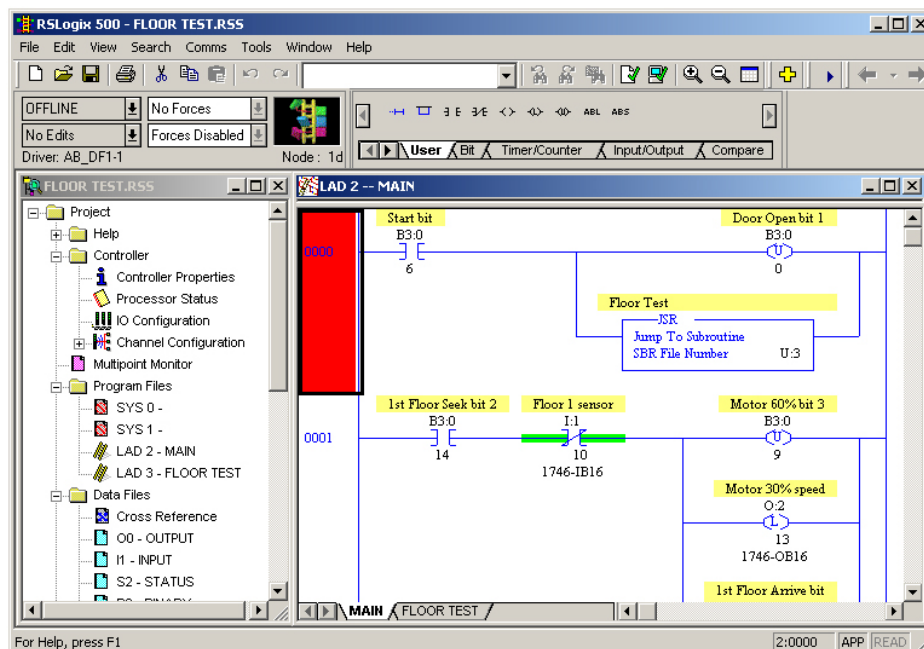


Figure 4-2: Example Screen Display

Turn on the controller.

Turn the key on the controller to PROG.

Click on the Comms menu and select Download

Depending on the software you will then be asked for revision information, click OK

A warning will appear asking whether the current PLC program is to be overwritten click OK.

You will then be asked if you wish to go online. Click OK.

When the computer is connected on-line to the controller the computer will show the same states as the controller. When ready turn the key on the front of the controller to RUN start the program. This will start the program.



NOTES:

1. To make changes to ladder logic using the RSLogix software you must take the controller offline.
2. When you download a program to the controller it will remain in memory until you download a different program. The program will remain in the controller's memory even if you turn the power off. This means once you have programmed the controller you can run the PLC programs without having to need a computer.

4.4.2 Mitsubishi FXGP/WIN Software

Controller type (used with the software): FX0 / FX0S

To download one of the sample ladder logic program carry out the following steps:

Start the MELSEC software

Select Project and Open from the menu and navigate to the location where the project files are held. Select the required project file.

From the Online menu, select Write to PLC.

Ensure the PLC is ready to receive programs by ensuring the pushbutton adjacent to the communication socket on the controller is set to STOP.

Click on the 'Param + Prog' pushbutton:

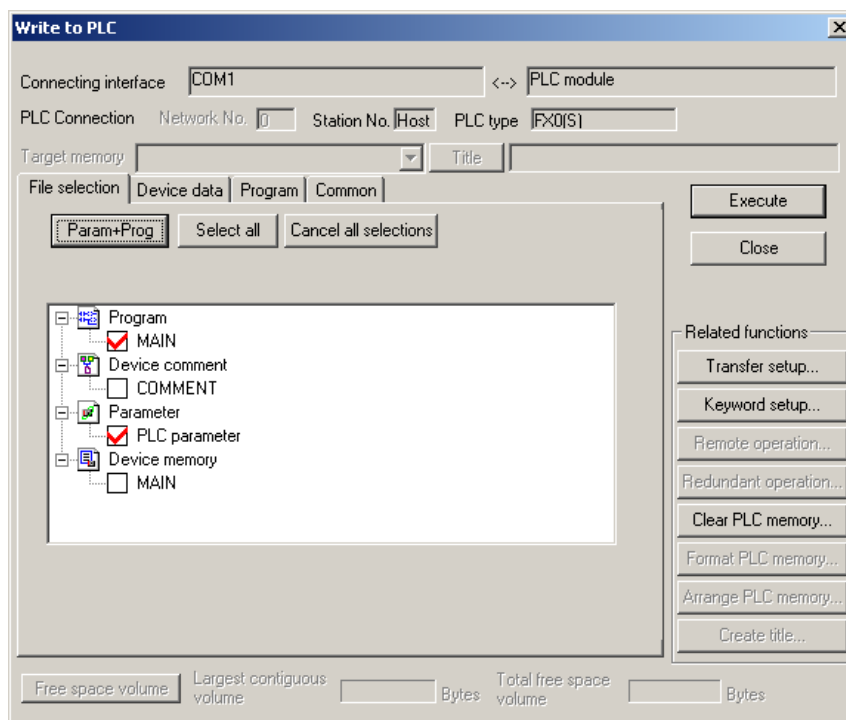


Figure 4-3



Click on Execute.

Set the RUN/STOP switch to RUN to start the program.

NOTES:

1. The exact instructions described within the sample programs may have to be different depending on the particular version of the controller and software you are using. Some interpretation of these instructions may be required.
2. Before attempting to download any programs to the controller ensure there are no other programs attempting to use the communication port. If another program is using the port the Melsec software will not be able to program the controller.

4.4.3 Practical 4: Running the Elevator Test Routine

The elevator is supplied with a test procedure. This can be used to exercise each of its features to:

- Check the controller is connected to the hardware properly
- Check that each of the components are functioning successfully

The following steps are performed:

Step number	Operation	Time (secs)
1	Door open	2
2	Down arrow	1
3	Up arrow	1
4	All the lights for Floor 1	1
5	All the lights for Floor 2	1
6	All the lights for Floor 3	1
7	All the lights for Floor 4	1
8	Brake (off when energized)	2
9	Direction (up when energized) Motor 30%	0.5
10	Motor 60%	0.5
11	Direction (down when de-energized) Motor 30%	0.5
12	Motor 60% Brake (on when de-energized)	.5 sec
13	Bell	.1 sec



Notes



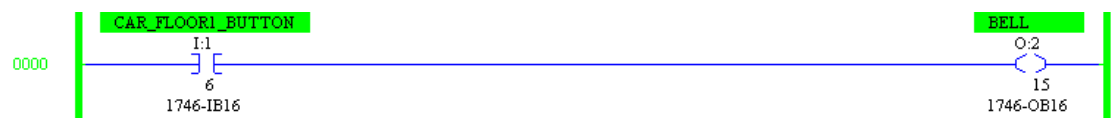
5 Assignment 2: PLC Fundamentals

5.1 Objectives

- To learn about the PLC inputs and outputs.
- To learn about the idea of a latch.
- To learn how to use time delays to show an indicator.
- To learn how to use the elevator bell.
- To learn about the Master Control Relay function.

5.2 Practical 1: A Simple Program

Make the bell sound when the Floor 1 car pushbutton is pressed.



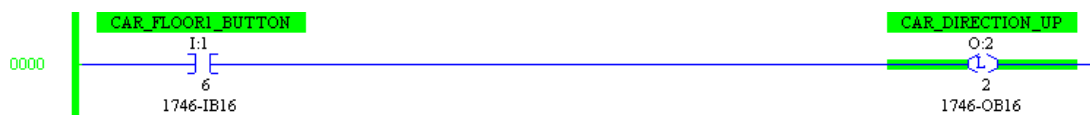
The bell will ring for as long as the alarm bit it is connected to is high.

5.3 Practical 2: Latches

This practical introduces the concept of a latch and shows how it can be used in ladder logic. A latch is a 'sticky switch'. When it is set to on, it remains on until you explicitly switch it off.

In this practical a program will be constructed which simulates a simple toggle switch. Pressing a switch once turns on a lamp. Pressing a switch for a second time turns it off.

The following program shows how the Floor 1 car pushbutton can be used to turn on the car direction indicator.

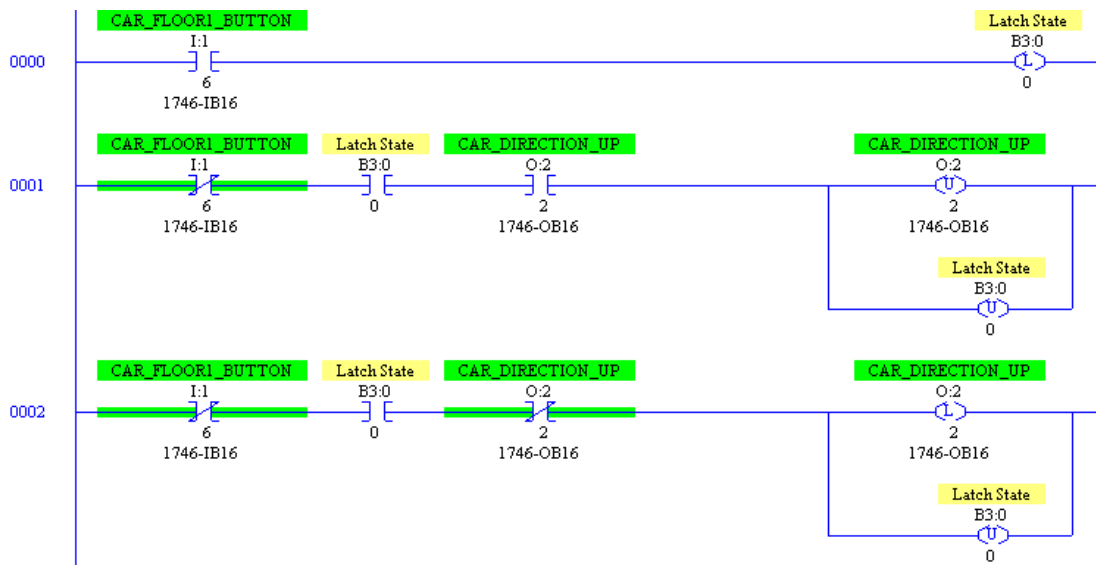


This will form the basis of the program.



The PLC has different types of memory. The amount of memory will depend on your particular controller. The SLC 5/03 CPU has 16 bits of binary memory which can be used by the logic programs. Many of the programs that are described will utilise latches, along with other techniques such as subroutines to control the flow of control.

It is also important to know that latched outputs can also serve as inputs to a ladder logic program.



It is necessary to consider the OFF state. If the following ladder logic conditions are written:

If the pushbutton is pressed and the lamp is off, turn the lamp on

If the pushbutton is pressed and the lamp is on, turn the lamp off

This program uses a 'latch bit' within memory to ensure that pushbutton press is remembered.

When the pushbutton is pressed, the lamp will be turned off and on so quickly it will not be possible to notice the changes. This example shows how important it is to consider every aspect of the system that you have to control.

5.4 Practical 3: AND, OR and NOT Conditions

This practical demonstrates how ladder logic can be used to write AND and OR conditions. This practical uses the input pushbuttons on the front of the elevator.

Write two different programs that perform the following:

Car Pushbutton 1 or Car Pushbutton 2 lights Car Pushbutton 3

Car Pushbutton 1 and Car Pushbutton 2 lights Car Pushbutton 3

Write a program that show a light if Car Pushbutton 1 is pressed and Car Pushbutton 2 is NOT pressed.

These forms of ladder logic are fundamental to many of the programs used to control the elevator.



5.5 Practical 4: Using a Timer

This practical introduces the timer.

There may be several different types of timer available on your controller. The TON and TOFF timer are the most important. The TON timer which means 'Timer On' waits until a particular input has been set to high for a specified length of time. The TOFF, meaning 'Timer Off' activates a particular output for a set length of time.

The action of pressing a pushbutton lights the Car Direction Up pushbutton for ten seconds.

Each timer has three possible outputs:

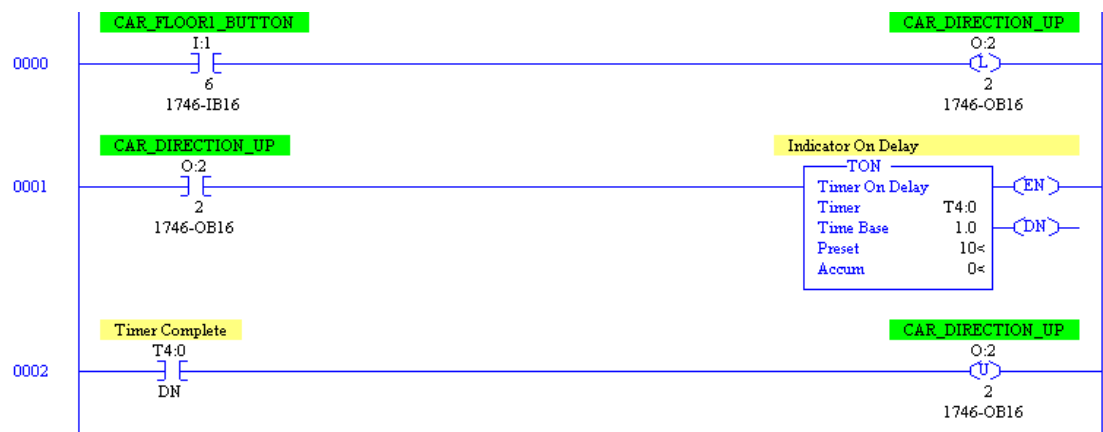
EN – Enabled bit

DN – Time done bit

TT – Timer timing bit

A flow chart illustrating this practical is shown below:

Corresponding ladder logic is show below:



NOTE:

The ladder logic instructions will be different depending on the controller you use. This example is taken from the Allen Bradley software.

Change the length of the timer. See how this effects the operation of the program.

In the elevator a timer can be used in a number of different ways:

To ensure that signals set by the PLC are received by the electronics of the equipment that is under control (in this case, the elevator)

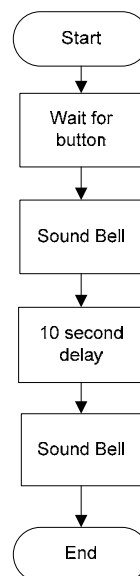
To ensure that the car has 'settled' before applying the brake and opening the doors.

To control the length of time the doors remain open before closing.



5.6 Practical 5: Using a Timer to Control the Bell

This practical builds on the previous practical, combining two timers together within the same program.





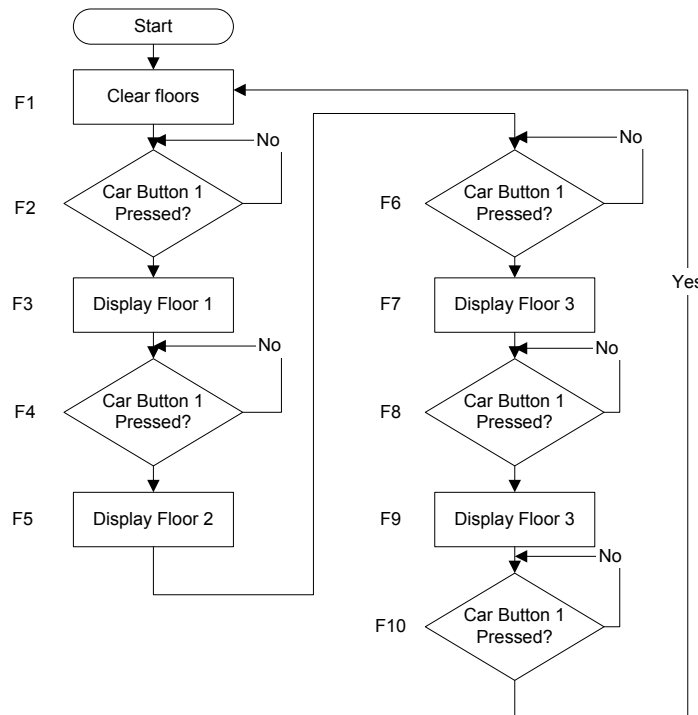
5.7 Practical 6: Master Control Relays

This practical introduces the concept of the Master Control Relay and how they can be used to control the flow of a program. The Master Control Relay allows parts of a ladder logic program to be turned on and off allowing the program to only perform the operations that are required. This can be particularly useful with a long program.

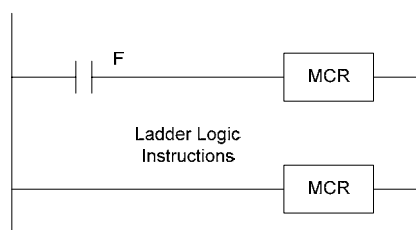
Consider the following scenario:

When the car pushbutton for Floor 1 is pressed, a lamp for the higher floor is displayed. For example, when the car Floor 1 pushbutton is pressed for the first time the lamp for Floor 1 will be highlighted. When it is pressed again the lamp for Floor 1 and 2 are lit. When all floor lamps are lit and the pushbutton is pressed again, all floors are extinguished.

This problem is represented using the following flowchart:



Each box has been labelled with a flowchart operation number, F1 through F10. Using the flowchart, a program can be derived where the Master Control Register is turned on and off using the state bits. This is shown in the following figure:





This type of design is called **Flowchart Based Design**. Another approach to designing PLC programs is **State Based Design**. Although it is possible to write successful ladder logic programs without the Master Control relay, limiting which instructions can be executed can be sometimes very useful.



6 Assignment 3: Elevator Initialisation

6.1 Objectives

- To learn about the elevator initialisation procedure.
- To learn about how to detect which floor the elevator is currently sitting at.
-

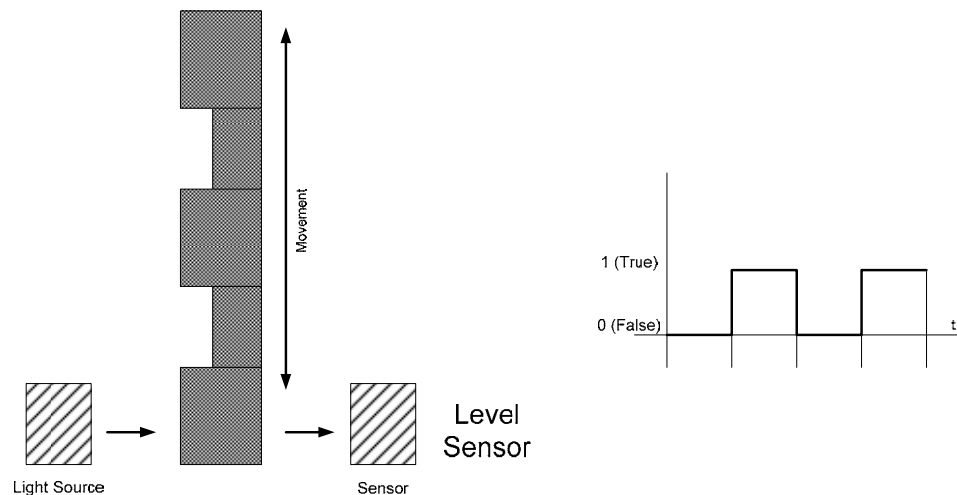
6.2 Practical 1: Initialisation

When you turn on the PLC controller for the first time it does not know what state the elevator hardware is set to. The elevator could be sitting between levels and the car door could be open. Before starting to control the elevator, the PLC must set the elevator to a known state. This is achieved through an initialisation routine.



In the real world, separate sensors would be used for above, below, and at each floor. In this trainer, however, to reduce the number of PLC inputs required, a single sensor for each floor is combined with the three-finger gate mounted to the car. This change is compensated for in the PLC programs.

It is important to understand how the level sensors operate. The following diagram shows the design of the level sensors:



The timing diagram on the right of the diagram shows how the state of the sensors change. If the car is at the bottom of the elevator (level 1), no signal is received since the light source is blocked. If the car is not at the level or is approaching a level, the sensor signal is high which means that the light signal is received.

The first time the initialisation procedure is run it is possible that the car is situated at a level or between levels (as if the car was approaching a level).

To summarise, a level sensor can have three states:

Car approaching floor state (sensor is off)

Car approaching floor level (sensor is on)

Car is at level (sensor is off)

If the level 1 sensor is detected the car will travel upwards until the level 2 sensor is found. At this point the main initialisation procedure will be carried out.

This procedure can be also represented as a **State Diagram**.

As suggested by the flow chart the initialisation routine can be divided into a number of stages:

Close the door and release the brake

See if car is at level 1

Move car to level 2

Return car to level 1

Turn motor off and apply brake

To disengage the brake, it must be energised. This is an important safety feature: if there is no power to the elevator, it will be stopped by default.



Some PLCs (such as the Allen-Bradley SLC and Micrologix controllers) contain a facility that allow ladder logic to be executed when a new controller program is started for the first time. This is performed by using some status bits. The act of changing the operating mode (from PROG mode to RUN, for example) will reset the 'start' status bit allowing the routine to begin again.

If your controller does not possess this capability the initialisation routine could begin when a pre-defined start pushbutton (or combination of pushbuttons) are pressed.

The algorithm behaves the way it does because if the car causes the Floor 1 sensor to be active, this could mean one of two possible states:

The car is at the level

The car is above the level

Since it is impossible for the controller know at which level the car is sitting at, if the level 1 is detected, it moves to the first level 2 sensor before moving down the elevator again. It is also possible for the elevator to sit between the two times when the car can be sensed at level 1. Because of this situation, it is necessary to move to car upwards to level 2 to begin the initialisation procedure.

Notice there is a difference in height between the floor entrance and the car entrance. Consider how a timer could be used to reduce the height of this gap. Are there any problems that might arise if a timer was used in this way?



7 Assignment 4: Basic Elevator Control

7.1 Objectives

- To learn about some of the elevator inputs and outputs.
- To learn how it is possible to control the car manually.
- To learn how to write a simple PLC program to move the elevator car.

7.2 Practical 1: Manual Car Control

In this practical you will simulate the operation of the jog pushbutton by programming the PLC to use the car level request pushbuttons. In writing the program you will gain an understanding of how to use the elevator inputs and outputs.

Use the level 1 car pushbutton to move the car down. Use the level 2 car pushbutton to move the car up. Rather than latching the output bits another approach is to set the outputs directly from the inputs. When either pushbutton is pressed you should release the elevator brake. When you release the pushbutton, the brake should be reapplied. You cannot damage the motor by driving the elevator car to the top of the shaft.

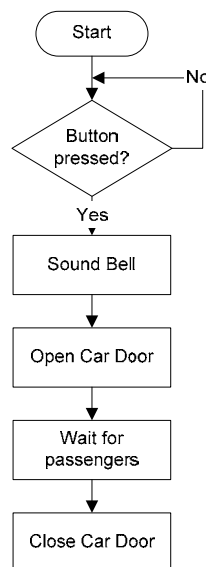
Consider the case if both pushbuttons were pressed at the same time. Add some logic to the program so that if two pushbuttons are pressed at the same time, the elevator car stops. Modify the program to experiment with different motor speeds.



7.3 Practical 2: Door and Bell Operation

This practical introduces how the bell and door could be used together to indicate the arrival of a car.

Use a pushbutton (such as the pushbutton to call the car from Floor 1) to start the program. This simulates the arrival of a car. Open the door, wait for five seconds, then close the door.



Consider how the program can be made more robust. What might happen if the pushbutton is pressed when the timer is running?

Is it possible to determine whether the door has been closed successfully?

7.4 Practical 3: Moving the Car to the Top

This practical introduces some more level sensors, and shows how the level sensor can be used as an approach sensor.

In this practical the car will be moved to the top level. When the car is approaching the top level it will slow (in a similar way that was shown in the initialisation procedure). At the end of the routine, the brake will be applied.

You do not have to include this initialisation routine. Use the jog control to move the elevator car to a level below level 4. What do you think might happen if the program begins when the car is at level 4?

Like the initialisation procedure it is important to consider the states the controller enters into.

Rather than using flowcharts, an alternative way to think about PLC program design is to use **State Diagrams**.

A state diagram is a way to represent a state machine. A state diagram represents a set of states within a system and a description of the transitions that can occur between them. When designing state diagrams it is useful to think about the following points:



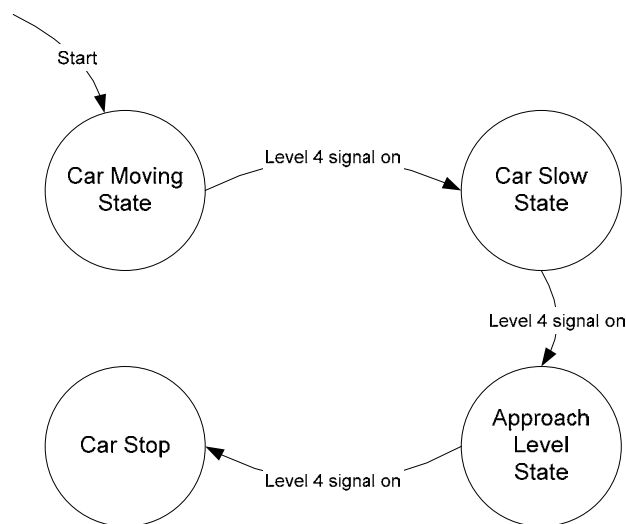
What is the system normally supposed to do?

Can anything change the behaviour of the system?

What might change the behaviour?

Is there a sequence of actions?

It is also important to remember that there might be a 'wait state'. The following diagram shows a simple state diagram for a program to move the elevator car to level 4:



In ladder logic, states can be represented by **binary bits** which allow certain sets of ladder logic conditions to be executed. Also, individual states within state diagrams can contain other state diagrams. This is related to the concept of the procedure which will be explored in the Subroutines assignment.

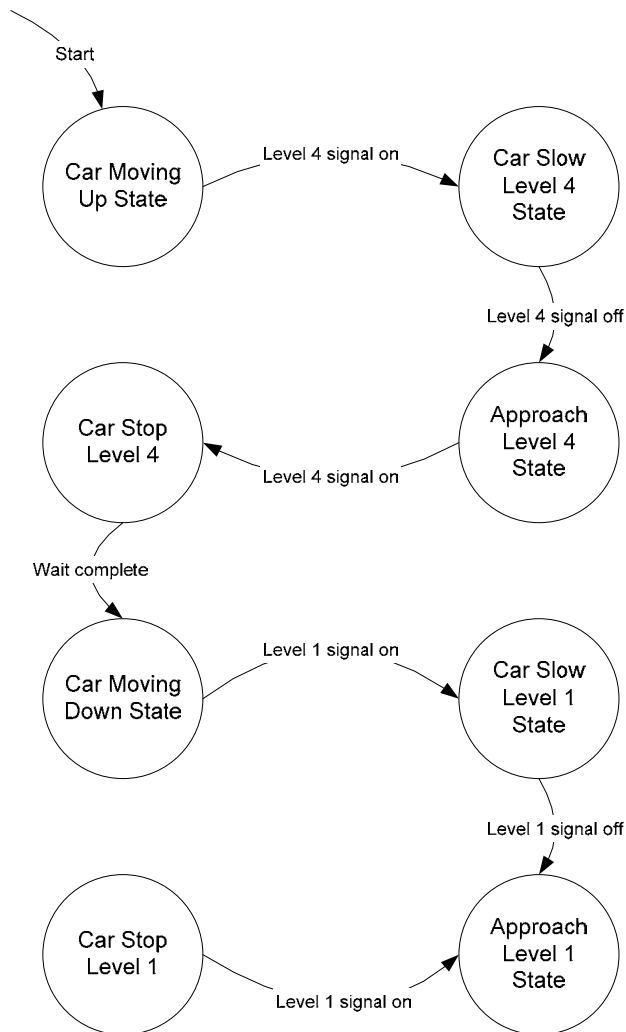
NOTE:

Some PLC software allows you to create programs using a combination of ladder logic and SFC. SFC, an abbreviation of Sequential Flow Chart allows you to divide your program into separate states, similar to how state diagrams are represented. The Mitsubishi controller programming software offers SFC support whilst the version of the Allen Bradley/Rockwell Automation software used to create the sample programs for this manual does not.



7.5 Practical 4: Move Car to Top then to the Bottom

This practical shows how sequences of actions can be combined together. State diagram is below:



This diagram was constructed by adding two state machines together. There is a lot of commonality between the two components of the diagram. This can be resolved by using a subroutine or a procedure. Subroutines are explored in a later assignment.

7.6 Practical 5: Creating a Loop using a Timer

Consider how the program could be extended to make the car continually move between the top and bottom of the elevator. A timer can be used to change between the 'up' and 'down' states.



8 Assignment 5: Subroutines

This assignment introduces the concept a subroutine.

8.1 Objectives

- To learn what a subroutine is and when a subroutine might be useful.
- To learn how to call subroutines from a main program.
- To learn how to write a general purpose level approach and depart subroutine.

By the end of this practical a framework to construct a two level elevator would have been constructed.

NOTE:

Different PLCs and their respective programming systems may adopt different approaches to create subroutines. Always investigate the features that your particular controller supports.

8.2 Practical 1: Creating Subroutines

This practical builds on the Basic Elevator Control assignment.

Many PLC systems offer the capability to divide programs into subroutines. Subroutines can be implemented using either the Allen-Bradley or the Mitsubishi PLCs.

A subroutine serves several purposes:

A subroutine allows a larger program to be divided up into smaller sections which can be understood more easily by programmers and maintainers.

A subroutine allows the same programming instructions to be shared between different elements of the same program, or even between different programs.

In this assignment, create a subroutine that:

- Moves the car to the top of the elevator shaft.
- Moves the car to the bottom of the elevator shaft.

8.3 Practical 2: Initialisation as a Subroutine

This program will be the basis for all later practicals and assignments.

Convert the elevator initialisation routine that was presented in an earlier assignment converting it so the initialisation routine is contained within a subroutine. Call this subroutine when the controller is started for the first time.



8.4 Practical 3: Showing the Current Level

In this practical you will write a program that moves the car to the top of the elevator, then to the bottom of the elevator. This movement operation will be repeated.

Unlike the previous practical you do not need to change the speed of the motor on floor arrival and departure. Instead the floor level lamp is to be illuminated once when the car passes a floor. The behaviour will be similar to the hard wired behaviour which can be observed by the floor indicators.

By understanding how to solve this problem by detecting state changes using the level sensors, you will be able to more easily write the programs for the later assignments.

To write the program, the following steps are recommended:

Create a program that contains an initialisation routine.

Add a 'move up' subroutine that moves upwards until the level 4 sensor is detected.

Add a 'move down' subroutine that moves to the bottom until level1 sensor is detected.

Add a general purpose function that indicates whether a level sensor has been detected.



9 Assignment 6: A Two Level Elevator

9.1 Objectives

- To understand how to implement a simple two level elevator system.
- To learn how a PLC can be extended by making calls to subroutines.

9.2 Practical 1: A Two Level Elevator

Two level (first and top), level pushbuttons.

The elevator goes to the bottom level and waits until either the level 4 car pushbutton is pressed or the level 4 elevator call pushbutton is pressed.

The elevator travels to the top floor and waits until either the level 1 car pushbutton is pressed or the level 1 elevator call pushbutton is pressed.

The program is repeated.

To begin to think about this problem consider how it can be decomposed. Thinking about the problem in terms of state diagrams may be useful. Do not add door control. This will be added within the next practical. Also consider whether your program is to allow requests whilst the car is moving.

9.3 Practical 2: Two Level Elevator with Door Control

As above, but adding the door opening and closing routine. Using the subroutine explored in the previous practical. Add bell control. Remember that since the controller may send signals faster than the electronics of the elevator can accept, it is important to create a delay to ensure that the bell is rung.

Implement the door open and close routine within a separate subroutine or state. Consider how the request function may change to cater for the case when the car is on the level that has been requested.

At this point, the program does not have any interlock. If additional floor buttons are pressed, the door will open and close while the car is travelling. Obviously this is not a desirable condition, and will be addressed later.



Chapter 9

ELEVATOR TRAINER

Assignment 6: A Two Level Elevator

Notes



10 Assignment 7: Visiting Floors

10.1 Objectives

- To learn how to write a sequence of actions using ladder logic.
- To understand that differences of sequences of actions need to be programmed even if the action may seem to be similar.

Use subroutines to do this.

10.2 Practical 1: Choosing the Correct Floor Sensor

In earlier assignments programs may have been written that use the inputs from specific level sensors. In this practical an approach to make the 'level sensing' part of the programs general purpose.

It is generally good practice to attempt to divide programs into general purpose subroutines. This has the following advantages:

It reduces the amount of memory used in a controller

It reduces the amount of effort required to carry out testing

It makes programs easier to understand and maintain

Write a program that does the following:

Allows the user to select the level sensor that they are interested in looking at by pressing the car pushbuttons.

Displays whether the level sensor is high or low by illuminating the 'car moving up' indicator.

When the program is running the level sensors can be studied by moving the car with the manual jog control. This programming can be used as a basis for the programs explored in the later practicals and assignments.

10.3 Practical 2: Stopping at Every Floor - Moving Up

Travel to the top of the elevator shaft and stop at every floor on the way up

Steps:

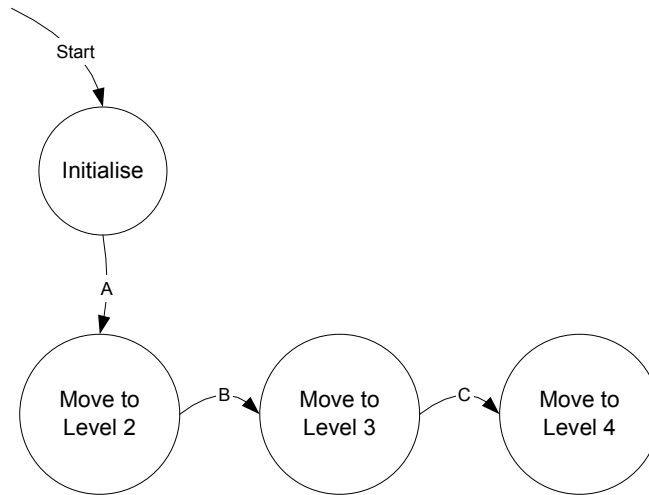
Execute initialisation routine.

Run one or more subroutines to move to each level until the top is reached.

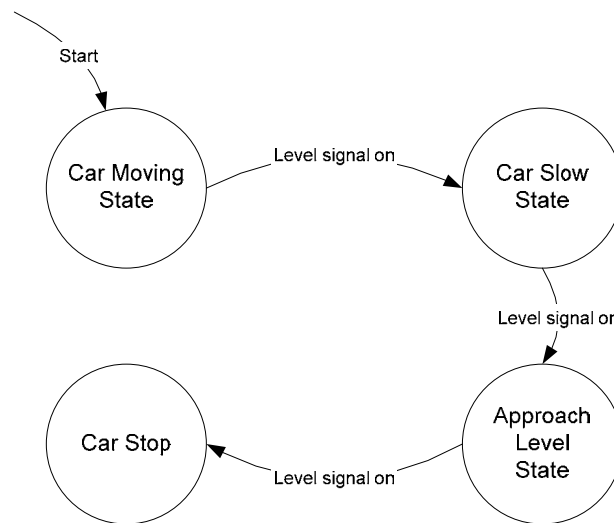
The action of slowing the car is the same no matter which floor the elevator is visiting.



A state diagram can be used to represent the procedure:



A second state diagram can be used to describe the process of moving between the floors:



The approach routine is as follows:

When the car is running at half speed, wait until the appropriate level sensor has been detected.
When detected, change to quarter speed

When sensor passed (the sensor is not active), indicate that it has been passed by setting a flag.

When the level sensor hits for a second time (when running at quarter speed), stop the car and exit the subroutine.

The final program will have a number of components

An initialisation procedure

A procedure to change which floor is being sensed



A procedure that controls the approach to the floor.

Instead of moving the car between levels you could write a program to simulate car movement using the floor lights. At the start of a program, Floor 1 could be lit. A short delay would follow where the Floor 1 light would be extinguished and the Floor 2 light would be lit.

10.4 Practical 3: Stopping at Every Floor - Moving Down

Steps:

- Execute initialisation routine.

- Run a subroutine that moves the car to the top of the elevator

- Run one or more subroutines to move to each level until the bottom is reached.

10.5 Practical 4: Stopping at Every Floor - Both Directions

Combine the programs for the last two practicals together (move the car up, stopping at every floor, and moving car down the elevator, stopping at every floor) to form a single program.

Make an enhancement to the program to run the program again after a short delay. This can be implemented using a timer.

10.6 Practical 5: Every Floor with Arrival Routine

Enhance the routine developed using the previous practical so that the car opens and closes on every floor. It may be useful to base the door opening routine on the two level elevator program.

The next practical will explore how to process the inputs from the pushbuttons.



ELEVATOR TRAINER

Chapter 10

Assignment 7: Visiting Floors

Notes



11 Assignment 8: Processing Inputs

11.1 Objectives

- To understand how programs can be written to make decisions about the direction of the car.

This assignment is a simulation assignment. This means that the inputs on the front of the elevator are used to explore how a suitable algorithm can be constructed. This algorithm will be used in later assignments.

11.2 Practical 1: Deciding on the Direction of Travel

Deciding which way to move the car is a fundamental part of a fully working algorithm. There are several ways to achieve this:

- Using combinations of logic.
- Using integer comparisons of the car location.
- Using integers can provide a more flexible approach since it can allow you to make changes to your algorithm more easily than if logic was used.

This demonstrates how you can decide in which direction the car should be moving. Unlike earlier practicals, this practical simulates the location of the car.

Set the location of the car using the floor pushbuttons

Set where the car should move to using the car pushbuttons

The movement decision is then shown by the going up and going down pushbuttons.

Depending on the controller you are using you may or may not be able to light all the pushbuttons.



ELEVATOR TRAINER

Chapter 11

Assignment 8: Processing Inputs

Notes



12 Assignment 9: Visiting Floors using Pushbutton Control

In this practical a simple elevator program will be created.

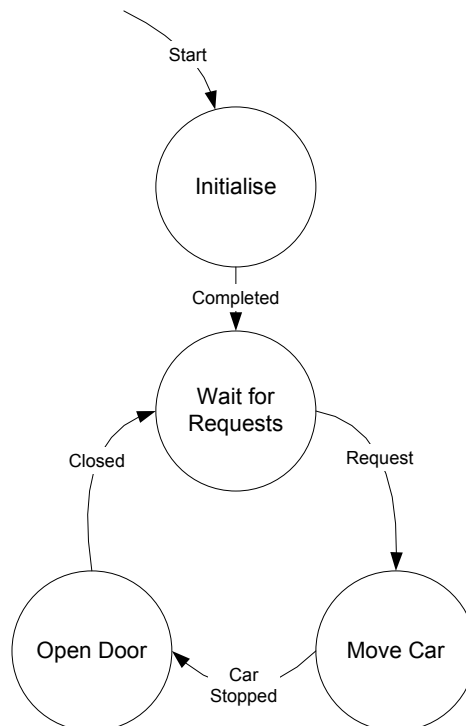
12.1 Objectives

- To create a program that implements a simple working elevator that accepts requests one at a time.
- To create a program that simple elevator program where the car can selectively service floors when an elevator is continually moving.

12.2 Practical 1: Visiting all Floors using Request Pushbuttons

This practical does not allow more than one input to be entered into the system at a time. The car gets to a floor and then waits for the next input. A more complicated algorithm is explored with the 'A Working Elevator' assignment.

When the car is at the required level, turn off the request lights (both in the car and on the level).



The car and the level pushbuttons have the same effect. On levels 2 and 3 the request up and down pushbuttons operate in the same way. A more complicated scenario is explored in a later assignment.



Use the ideas explored within the previous practical to determine the direction that the car must travel in to service a floor. Think about whether this program would work in practice, listing how it might be improved.

12.3 Practical 2: A Constantly Moving Elevator

Write a program which combines the ideas explored in the previous practical with the assignment where a program was written that visited the top of the elevator and then visited the bottom.

If a floor is requested (by a passenger waiting at a floor or a passenger within a car), the car will stop at the requested floor when it passes by.

Begin by writing a program that moves the car between the top of the elevator and the bottom.

Modify the program so it approaches the top and bottom floors using quarter speed before changing directions. Use quarter speed when leaving and arriving at floors.

Add a mechanism that allows you to latch floor lamps when either the car or the floor pushbuttons are pressed.

Add a general purpose level approach function which is connected to the level.

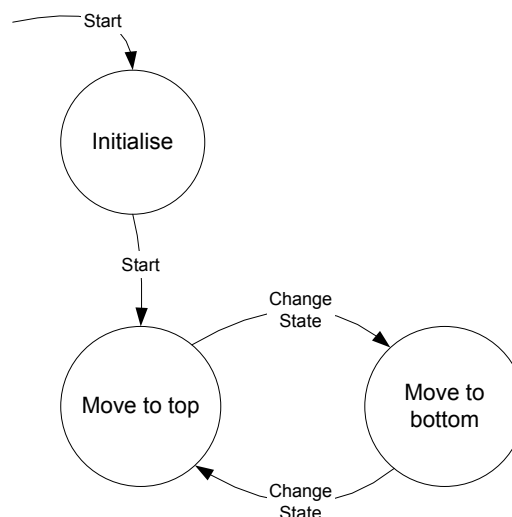
Add a door open and close routine.

To begin designing the program it is recommended that you try drawing a state diagram for:

The 'up' part of the program.

The 'down' part of the program.

The high level state diagram for this program will look similar to the following:



The elevator can accept floor requests at any time.

As a starting point, begin by writing a program that moves the car from the bottom to the stop. It is important to ensure that the car does not move until the door is closed. Also consider the case of how to handle the situation where a request pushbutton is pressed the moment the car is about to leave the floor.



ELEVATOR TRAINER

Chapter 12 Assignment 9: Visiting Floors using Pushbutton Control

There are two main problems with this elevator control strategy:

- It does not use energy efficiently.
- It does not serve passengers as well as it could do.

The next assignment explores how to implement a more efficient elevator control program.



ELEVATOR TRAINER

Chapter 12
Assignment 9: Visiting
Floors using Pushbutton Control

Notes



13 Assignment 10: A Working Elevator

13.1 Objectives

- To understand how to control an elevator using a more complicated algorithm.
- To understand how to determine the highest and lowest levels selected.
- To understand how to make use of directional floor controls.
- To understand how to make use of both floor controls and car pushbutton controls.

13.2 Practical 1: Determining the Highest and Lowest Levels

One of the most important parts of the full control program is the ability to determine the extent to which an elevator should travel. In this practical the elevator car pushbuttons will be used to simulate which floors are requested.

The floor lamp pushbuttons, found on the right of the elevator will be used to display which is the highest (or the lowest selected) level.

The suggested approach to design such as program is:

Design and implement a mechanism to latch and unlatch the car pushbutton lamps. This will be used to indicate a level request. More information about how to perform this is provided in a previous assignment.

Decide on a way to determine which is the highest pushbutton selected.

Connect the result of this operation to the floor output lamps.

When the highest level algorithm has been determined, modify the routine so the lowest requested floor is displayed by the floor lamps.

This practical is best performed with a controller that has more than 11 outputs (so the latching can be seen on the elevator hardware). If the Micrologix 1000 PLC is used, you can still write the program but you will not be able to see the latching of the inputs.

As well as writing logic to solve this problem, it is also possible to use the integer arithmetic capabilities of your controller.

13.3 Practical 2: Full Control (Simple Floor Requests)

Algorithm fundamental: If car is moving up, service those up before going back down again. The reason for adopting such an approach is to save energy.

Consider the algorithm used within the 'visiting floors using pushbutton control' assignment. Although this is a working program, the passenger who pushes the pushbutton fastest gets control of the elevator. This is clearly undesirable.

The elevator will know about the top and bottom levels, and whether the elevator needs to travel further to service a demand.



One of the biggest differences between this problem and the Visiting Floors using Request Pushbuttons lies with the ability to receive car requests at any time.

A further complexity lies with how to process the car pushbuttons. For this practical, the car pushbuttons are either ignored or they should have the same action as the floor pushbuttons. The final practical will complete the algorithm by adding the car pushbuttons.

Before beginning to design the program it is useful to think of the problem in terms of requirements:

The car must visit the requested floor as quickly as possible.

The elevator should attempt to minimise the amount of energy used.

A description of an algorithm that attempts to satisfy these requirements is as follows:

Move to the first level that has been requested. If there are requests that are higher than the current level of the car continue moving the car upwards until every requested floor has been visited. When there are no more requests higher than the current car level, change the direction of the car and service requests that are below the current level (ignoring requests that are higher than the current car level). When there are no further lower requests (or the bottom of the elevator has been reached), change direction of the car and service requests higher than the current floor level. The procedure should be repeated.

This algorithm can be divided into a number of steps.

Algorithm:

Perform an initialisation routine so the elevator is set to level 1. Set direction to up.

Check for elevator requests

Determine

Two status bits: low remaining and high remaining. This indicates that the elevator has other floors to service. This bit is determined by calculating the difference between the level where the car was last recorded, and the highest floor that is to be serviced.

To make it easier, it is recommended that initially that the car speed is not changed when the car departs and leaves floors.

13.4 Practical 3: Full Control (Directional Floor Requests)

An example:

Following initialisation (where the elevator is initialised to sit at level 1), assume the following requests: 3U, 4D, 2D. The following sequence will be executed:

All up requests are serviced, i.e. in this case only 3U will be serviced.

Next the elevator reaches the fourth floor to service 4

Finally it services the remaining down requests, which in this case is 2D.

Use this example as a way to test your routine.



13.5 Practical 4: Full Control (All Floor Controls)

In full size elevators remember that the passengers do not usually see which floors the elevator is about to call at (which level pushbuttons have been pressed by passengers). This practical expands the previous practical by adding logic to ensure that the passengers in the car do not receive more information than they need.



Chapter 13

ELEVATOR TRAINER

Assignment 10: A Working Elevator

Notes



14 Assignment 11: Advanced Elevator Programming

14.1 Objectives

Sample programs for these advanced assignments are not provided.

14.2 Practical 1: Priority for Top Level

If a request from Level 4 is received, providing that the passengers within the car have not made any requests, the requests from the other levels are ignored.

14.3 Practical 2: Enhanced Elevator Control

Modify the program so the length of time the doors open and close depend on whether there is a demand for the elevator either from within the car or from other levels.

14.4 Practical 3: Multi-Speed Elevator Control

All practicals so far have used only the 60% elevator speed.

To ensure that passengers have a smooth ride and the physical forces on the elevator equipment are reduced, some elevators vary their running speed. The change of speed can be performed when an elevator car departs and approaches a floor.

In this practical you will write an elevator control program that uses different speeds. If the car is departing the floor, then its speed should increase from 30% to 60%. If the car is approaching the destination floor, then its speed should reduce from 60% to 30%.

A simple state diagram illustrating how a multi-speed approach/depart routine may look is given in Figure 14-1.

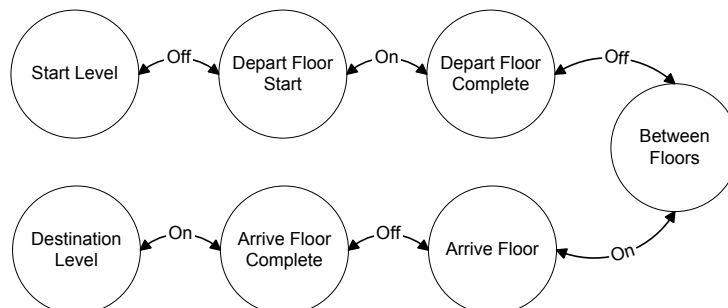


Figure 14-1

As a further enhancement to the program, consider using the sensors on intermediate levels to increase the speed of the elevator to 100%. This way it possible to increase the performance of the elevator. This approach can be used to simulate a sophisticated multiple demand elevator.



ELEVATOR TRAINER

Chapter 14
Assignment 11:
Advanced Elevator Programming

Notes



15 Appendix A Specification

15.1 Features

4 floors

Elevator car driven by belt with counterbalance

'Normally on' mechanical brake to hold car at desired floor

Winder motor driven by fully protected power amplifier (e.g. if elevator car jams or car is driven with the brake applied)

Built-in closed loop speed control with option to implement loop via an external controller

Elevator car with motorised door, internal illumination and sounder to announce arrival

Visual indication of position of elevator car and direction of travel

Up/down call pushbuttons on each floor

Floor request pushbuttons 'inside' elevator car (located on external panel)

Discrete motor interface for PLC logic-only control: direction up/down, 100% speed, 60% and 30% speed

Time to travel from ground to top floor at 100% speed: ~7 secs

Vertical travel limit switches

Manual drive switch on front panel to aid testing and debugging

Fault simulation selected via rear panel DIP switches

Example code provided for Allen-Bradley and Mitsubishi PLCs

Electrical interface: logic I/O will operate from 3 V to 24 V (using external pull-up voltage), analogue I/O operates from 0–10 V or ± 10 V

Multiple units may be mounted side by side to form complex multi-elevator systems

Load cell to indicate passenger load in elevator car

Selection of weights provided to simulate varying passenger loads

Analogue motor position feedback signal

15.2 Dimensions

H 1060 x W 340 x D 260 mm

Weight 14 kg



ELEVATOR TRAINER

Appendix A Specification

15.3 Internal Power Supply:

Universal input range 110–230 V ac 50/60 Hz

Power 50 VA

15.4 Environment

Typical laboratory conditions

Temperature: 10–40°C

Humidity: 10–90% non condensing



16 Appendix B External Controller Interface

All logic input levels conform to the following:

Low = $0\text{ V} \leq V_{in} \leq 0.5\text{ V}$ or input floating

High = $3.0\text{ V} \leq V_{in} \leq 24\text{ V}$

All inputs are active high.

All logic output levels conform to the following:

Low = $V_{out} < 0.3\text{ V}$

High = $V_{out} \approx$ pull-up voltage

All outputs are active low.

Table 16-1 shows the connections available on the rear panel connectors together with a description of their function and voltage levels.

Table 16-1: External interface connections

Signal Name	Connector	Type	Description	Scaling/Logic sense
0V	J2 P1 P10	Power rail	0V	
Current	J2 P2	Analogue input	Feeds a current demand directly into motor current controller	-10V = -0.25A +10V = +0.25A
Bi speed	J2 P3	Analogue input	Elevator car speed demand (bipolar)	-10V = - full speed +10V = + full speed
Uni Speed	J2 P4	Analogue input	Elevator car speed demand (unipolar)	0V = -full speed +5V = zero speed +10V = + full speed
Position	J2 P6	Analogue output	Car position signal	0→+10V = Floor 1 – Floor 4
Bi Tacho	J2 P7	Analogue output	Motor speed signal (bipolar)	-10V = -full speed 0V = zero speed +10V = +full speed
Uni Tacho	J2 P8	Analogue output	Motor speed signal (unipolar)	0V = -full speed +5V = zero speed +10V = +full speed
Load Cell	J2 P9	Analogue output	Elevator car load signal	



Appendix B

ELEVATOR TRAINER

External Controller Interface

Signal Name	Connector	Type	Description	Scaling/Logic sense
Door cntl	J4 P1	Digital input	Open/close elevator car door	Low = close High = open
Brake cntl	J4 P2	Digital input	Release the elevator car brake	Low = apply brake High = release brake
100% speed	J4 P3	Digital input	Run winder motor at full speed	Low = off High = run
60% speed	J4 P4	Digital input	Run winder motor at 60% speed	Low = off High = run
30% speed	J4 P5	Digital input	Run winder motor at 30% speed	Low = off High = run
Direction	J4 P6	Digital input	Select winder motor direction up/down	Low = Down High = Up
0V	J4 P7	Power rail	0V	
F1 UP LED	J18 P1	Digital input	Drives LED by Floor 1 UP request pushbutton to indicate elevator is on its way	Low = LED off High = LED on
F2 DOWN LED	J18 P2	Digital input	Drives LED by Floor 2 DOWN request pushbutton to indicate elevator is on its way	Low = LED off High = LED on
F2 UP LED	J18 P3	Digital input	Drives LED by Floor 2 UP request pushbutton to indicate elevator is on its way	Low = LED off High = LED on
F3 DOWN LED	J18 P4	Digital input	Drives LED by Floor 3 DOWN request pushbutton to indicate elevator is on its way	Low = LED off High = LED on
F3 UP LED	J18 P5	Digital input	Drives LED by Floor 3 UP request pushbutton to indicate elevator is on its way	Low = LED off High = LED on
F4 DOWN LED	J18 P6	Digital input	Drives LED by Floor 4 DOWN request pushbutton to indicate elevator is on its way	Low = LED off High = LED on
+24V	J10 P1	Power rail	24V supply to use as a pull-up. Protected by resettable fuse	+24V @ 200mA
Down LED	J10 P2	Digital input	Drives elevator 'down' front panel arrow	Low = LED off High = LED on



ELEVATOR TRAINER

External Controller Interface

Signal Name	Connector	Type	Description	Scaling/Logic sense
Up LED	J10 P3	Digital input	Drives elevator 'up' front panel arrow	Low = LED off High = LED on
F1 LED	J10 P4	Digital input	Drives LED by elevator car Floor 1 request pushbutton to indicate elevator destination	Low = LED off High = LED on
F2 LED	J10 P5	Digital input	Drives LED by elevator car Floor 2 request pushbutton to indicate elevator destination	Low = LED off High = LED on
F3 LED	J10 P6	Digital input	Drives LED by elevator car Floor 3 request pushbutton to indicate elevator destination	Low = LED off High = LED on
F4 LED	J10 P7	Digital input	Drives LED by elevator car Floor 4 request pushbutton to indicate elevator destination	Low = LED off High = LED on
Sounder	J10 P8	Digital input	Drives sounder – can be used to indicate arrival at requested floor (momentary) or fault condition (pulsing)	Low = off Short pulse high = sound once High = sound continuously
+24V	J12 P1	Power rail	24V supply to use as a pull-up. Protected by resettable fuse.	+24V @ 200mA
Pull-up rail	J12 P2	Power rail	Connect to an external pull-up rail in the range +3V → +24V	
+5V	J12 P3	Power rail	5V supply to use as a pull-up. Protected by resettable fuse.	+5V @ 200mA
Call F1 U	J12 P4	Digital output	Calls elevator to go up from Floor 1 pushbutton	High = inactive Low = pushbutton pressed
Call F2 U	J12 P5	Digital output	Calls elevator to go up from Floor 2 pushbutton	High = inactive Low = pushbutton pressed
Call F2 D	J12 P6	Digital output	Calls elevator to go down from Floor 2 pushbutton	High = inactive Low = pushbutton pressed
Call F3 U	J12 P7	Digital output	Calls elevator to go up from Floor 3 pushbutton	High = inactive Low = pushbutton pressed
Call F3 D	J12 P8	Digital output	Calls elevator to go down from Floor 3 pushbutton	High = inactive Low = pushbutton pressed
Call F4 D	J12 P9	Digital output	Calls elevator to go down from Floor 4 pushbutton	High = inactive Low = pushbutton pressed



Appendix B

ELEVATOR TRAINER

External Controller Interface

Signal Name	Connector	Type	Description	Scaling/Logic sense
Car Call F1	J14 P1	Digital output	'Inside elevator car' – selects Floor 1	High = inactive Low = pushbutton pressed
Car Call F2	J14 P2	Digital output	'Inside elevator car' – selects Floor 2	High = inactive Low = pushbutton pressed
Car Call F3	J14 P3	Digital output	'Inside elevator car' – selects Floor 3	High = inactive Low = pushbutton pressed
Car Call F4	J14 P4	Digital output	'Inside elevator car' – selects Floor 4	High = inactive Low = pushbutton pressed
F1 sensor	J20 P1	Digital output	Output of opto sensor to indicate elevator at Floor 1	High = inactive Low = at floor
F2 sensor	J20 P2	Digital output	Output of opto sensor to indicate elevator at Floor 2	High = inactive Low = at floor
F3 sensor	J20 P3	Digital output	Output of opto sensor to indicate elevator at Floor 3	High = inactive Low = at floor
F4 sensor	J20 P4	Digital output	Output of opto sensor to indicate elevator at Floor 4	High = inactive Low = at floor
Door open	J20 P5	Digital output	Elevator door open indicator	Low = closed High = open
Door closed	J20 P6	Digital output	Elevator door closed indicator	Low = closed High = open
0V	J20 P7,P8	Power rail	0V	



17 Appendix C Interfacing with Allen-Bradley SLC-500 Series

The elevator outputs will interface directly with Allen-Bradley 24 V dc current source input modules e.g. 1746-IV8, 1746-IV16 and 1746-IV32. The PLC terminals labelled COM or DC COM should be commoned together and connected to the elevator 0 V. The elevator outputs need a pull-up voltage which is supplied by the elevator 24 V supply output according to the connections shown in Table 17-1.

The elevator inputs will interface directly with Allen-Bradley 24 V dc current sourcing output modules e.g. 1746-OB8, 1746-OB16 and 1746-OB32. The terminals labelled COM or DC COM should be commoned and connected to the elevator 0 V. The terminals labelled VDC should be connected to the elevator 24 V supply. The connections given in Table 17-1 use the 24 V supply from the elevator to power the relay outputs.

Table 17-1 shows the connections for the 1746-IV16 16 digital input module and 1746-OB16.

Refer to Allen-Bradley publication number 1746-2.35 *Discrete Input and Output Modules* for further information.

Table 17-1: Allen-Bradley SLC-500 connection table

PLC I/O Point	34-150 Connector	Function	Detail
	J12 P1 ³	Power rail	+24V rail output
	J12 P2 ³	Output pull-up	Output common pull-up point
IN0	J12 P4	Floor call pushbuttons	Floor 1 call pushbutton – UP
IN1	J12 P5		Floor 2 call pushbutton – UP
IN2	J12 P6		Floor 2 call pushbutton – DOWN
IN3	J12 P7		Floor 3 call pushbutton – UP
IN4	J12 P8		Floor 3 call pushbutton – DOWN
IN5	J12 P9		Floor 4 call pushbutton – DOWN
IN6	J14 P1		Pushbuttons 'inside' car (mounted on panel LHS)
IN7	J14 P2	Car panel – select Floor 2 pushbutton	
IN8	J14 P3	Car panel – select Floor 3 pushbutton	
IN9	J14 P4	Car panel – select Floor 4 pushbutton	
IN10	J20 P1	Floor location sensors	Car at Floor 1 sensor
IN11	J20 P2		Car at Floor 2 sensor
IN12	J20 P3		Car at Floor 3 sensor
IN13	J20 P4		Car at Floor 4 sensor
IN14	J20 P5	Door open switch	Car door open
IN15	J20 P6	Door closed switch	Car door closed
DC COM	J20 P7	Power rail	0V



PLC I/O Point	34-150 Connector	Function	Detail
OUT0	J4 P1	Door control	Open/close car door
OUT1	J4 P2	Brake control	Release car brake
OUT2	J10 P3	Car going up	Front panel 'going up' arrow
OUT3	J10 P2	Car going down	Front panel 'going down' arrow
OUT4	J18 P6	Elevator 'on its way' indicators	Illuminate Floor 4 down indicator LED
OUT5	J18 P5 ¹		Illuminate Floor 3 up indicator LED
OUT5	J18 P4 ¹		Illuminate Floor 3 down indicator LED
OUT6	J18 P3 ²		Illuminate Floor 2 up indicator LED
OUT6	J18 P2 ²		Illuminate Floor 2 down indicator LED
OUT7	J18 P1		Illuminate Floor 1 up indicator LED
OUT8	J10 P7		Elevator destination
OUT9	J10 P6	Illuminate Floor 3 indicator LED in car	
OUT10	J10 P5	Illuminate Floor 2 indicator LED in car	
OUT11	J10 P4	Illuminate Floor 1 indicator LED in car	
OUT12	J4 P4	Logic motor control	Motor 60% speed demand
OUT13	J4 P5		Motor 30% speed demand
OUT14	J4 P6		Motor direction up/down
OUT15	J10 P8	Sounder	Bell to announce arrival of car at floor
VDC	J10 P1	Power rail	+24V rail output
DC COM	J4 P7	Power rail	0V

Notes

1. Link together to common Floor 3 indicator LEDs
2. Link together to common Floor 2 indicator LEDs
3. Link together to connect the elevator output pull-up rail to the +24 V supply terminal on the elevator



18 Appendix D Interfacing with Allen-Bradley MicroLogix 1000 Series

The MicroLogix 1000 series has a limited number of I/O points. The only suitable version is the 1761-L32BWA with 20 inputs and 12 outputs.

The elevator outputs will interface directly with the MicroLogix 1000 24 V dc inputs connected in current source configuration. The PLC terminal labelled 24 V should be commoned with both PLC DC COM terminals and connected to the elevator 0 V. The elevator outputs need a pull-up voltage which is supplied by the elevator 24 V supply output according to the connections shown in Table 18-1.

The elevator inputs will interface directly with the MicroLogix 1000 24 V relay outputs connected in current sinking configuration. The PLC terminals labelled VAC/VDC should all be commoned and connected to a 24 V dc supply. The connections given in Table 18-1 use the 24 V supply from the PLC to power the relay outputs.

Table 18-1 shows the connections. Note that the floor up/down request LEDs have been commoned with the car floor indicator LEDs in order to reduce the number of PLC outputs needed.

Table 18-1: Allen-Bradley MicroLogix 1000 connection table

PLC I/O Point	34-150 Connector	Function	Detail
	J12 P1 ⁶	Power rail	+24V rail output
	J12 P2 ⁶	Output pull-up	Output common pull-up point
I/0	J12 P4	Floor call pushbuttons	Floor 1 call pushbutton – UP
I/1	J12 P5		Floor 2 call pushbutton – UP
I/2	J12 P6		Floor 2 call pushbutton – DOWN
I/3	J12 P7		Floor 3 call pushbutton – UP
I/4	J12 P8		Floor 3 call pushbutton – DOWN
I/5	J12 P9		Floor 4 call pushbutton – DOWN
I/6	J14 P1	Pushbuttons 'inside' car (mounted on panel LHS)	Car panel – select Floor 1 pushbutton
I/7	J14 P2		Car panel – select Floor 2 pushbutton
I/8	J14 P3		Car panel – select Floor 3 pushbutton
I/9	J14 P4		Car panel – select Floor 4 pushbutton
I/10	J20 P1	Floor location sensors	Car at Floor 1 sensor
I/11	J20 P2		Car at Floor 2 sensor
I/12	J20 P3		Car at Floor 3 sensor
I/13	J20 P4		Car at Floor 4 sensor
I/14	J20 P5	Door open switch	Mimic indication – car door open
I/15	J20 P6	Door closed switch	Mimic indication – car door closed
DC COM	J20 P7	Power rail	0 V



PLC I/O Point	34-150 Connector	Function	Detail	
24V-	J20 P8	Power rail	0 V	
O/0	J4 P1	Door control	Open/close car door	
O/1	J4 P2	Brake control	Release car brake	
O/2	J10 P3	Car going up	Front panel 'going up' arrow	
O/3	J10 P2	Car going down	Front panel 'going down' arrow	
O/4	J18 P6 ¹	Elevator 'on its way'/destination indicators	Illuminate Floor 4 down indicator LED	
	J10 P7 ¹		Illuminate Floor 4 indicator LED in car	
O/5	J18 P5 ²		Illuminate Floor 3 up indicator LED	
	J18 P4 ²		Illuminate Floor 3 down indicator LED	
	J10 P6 ²		Illuminate Floor 3 indicator LED in car	
O/6	J18 P3 ³		Illuminate Floor 2 up indicator LED	
	J18 P2 ³		Illuminate Floor 2 down indicator LED	
	J10 P5 ³		Illuminate Floor 2 indicator LED in car	
O/7	J18 P1 ⁴		Illuminate Floor 1 up indicator LED	
	J10 P4 ⁴		Illuminate Floor 1 indicator LED in car	
O/8	J4 P4		Logic motor control	Motor 60% speed demand
O/9	J4 P5			Motor 30% speed demand
O/10	J4 P6	Motor direction up/down		
O/11	J10 P8	Sounder	Sounder to announce arrival of car at floor	
VAC/VDC ⁵ +24V DC OUT	–	Power rail for PLC relay outputs	+24V	

Notes

1. Link together to common Floor 4 indicator LEDs
2. Link together to common Floor 3 indicator LEDs
3. Link together to common Floor 2 indicator LEDs
4. Link together to common Floor 1 indicator LEDs
5. Link all PLC VAC/VDC terminals together and link them to the +24 V dc OUT terminal on the PLC.
6. Link together to connect the elevator output pull-up rail to the +24 V supply terminal on the elevator



19 Appendix E Interfacing with Mitsubishi FXo-30MR-ES

X = inputs

Y = outputs

The elevator outputs will interface directly with the Mitsubishi FXo-30MR-ES 24 V dc inputs connected in the source configuration. The PLC terminal labelled S/S should be connected to the PLC 0 V terminal and the elevator 0 V. The elevator pull-up rail is connected to the 24 V power terminal.

The elevator inputs will interface directly with the Mitsubishi FXo-30MR-ES relay outputs. The PLC terminals labelled COM0, COM1, COM2, COM3 and 24V should all be connected together. The PLC terminal labelled 0V should be connected to the elevator 0 V.

Table 19-1 shows the connections for the Mitsubishi FXo-30MR-ES .

Table 19-1: Mitsubishi FXo-30MR-ES connection table

PLC I/O Point	34-150 Connector	Function	Detail
	J12 P1 ³	Power rail	+24V rail output
	J12 P2 ³	Output pull-up	Output common pull-up point
X0	J12 P4	Floor call pushbuttons	Floor 1 call pushbutton – UP
X1	J12 P5		Floor 2 call pushbutton – UP
X2	J12 P6		Floor 2 call pushbutton – DOWN
X3	J12 P7		Floor 3 call pushbutton – UP
X4	J12 P8		Floor 3 call pushbutton – DOWN
X5	J12 P9		Floor 4 call pushbutton – DOWN
X6	J14 P1		Pushbuttons 'inside' car (mounted on panel LHS)
X7	J14 P2	Car panel – select Floor 2 pushbutton	
X8	J14 P3	Car panel – select Floor 3 pushbutton	
X9	J14 P4	Car panel – select Floor 4 pushbutton	
X10	J20 P1	Floor location sensors	Car at Floor 1 sensor
X11	J20 P2		Car at Floor 2 sensor
X12	J20 P3		Car at Floor 3 sensor
X13	J20 P4		Car at Floor 4 sensor
X14	J20 P5	Door open switch	Mimic indication – car door open
X15	J20 P6	Door closed switch	Mimic indication – car door closed



PLC I/O Point	34-150 Connector	Function	Detail
0V ⁴ S/S ⁴	J20 P7	Power rail	0V
Y0	J4 P1	Door control	Open/close car door
Y1	J4 P2	Brake control	Release car brake
Y2	J10 P3	Car going up	Front panel 'going up' arrow
Y3	J10 P2	Car going down	Front panel 'going down' arrow
Y4	J18 P6	Elevator 'on its way' indicators	Illuminate Floor 4 down indicator LED
Y5	J18 P5 ¹		Illuminate Floor 3 up indicator LED
Y5	J18 P4 ¹		Illuminate Floor 3 down indicator LED
Y6	J18 P3 ²		Illuminate Floor 2 up indicator LED
Y6	J18 P2 ²		Illuminate Floor 2 down indicator LED
Y7	J18 P1		Illuminate Floor 1 up indicator LED
Y8	J10 P7	Elevator destination	Illuminate Floor 4 indicator LED in car
Y9	J10 P6		Illuminate Floor 3 indicator LED in car
Y10	J10 P5		Illuminate Floor 2 indicator LED in car
Y11	J10 P4		Illuminate Floor 1 indicator LED in car
Y12	J4 P4	Logic motor control	Motor 60% speed demand
Y13	J4 P5		Motor 30% speed demand
Y14	J4 P6		Motor direction up/down
Y15	J10 P8	Sounder	Bell to announce arrival of car at floor
0V	J4 P7	Power rail	0V
COM0– COM3 24V ⁵	-		

Notes

1. Link together to common Floor 3 indicator LEDs
2. Link together to common Floor 2 indicator LEDs
3. Link together to connect the elevator output pull-up rail to the +24 V supply terminal on the elevator
4. Link PLC 0 V and S/S terminals together and connect to elevator 0 V
5. Link together PLC terminals labelled COM0, COM1, COM2, COM3 and 24 V



20 Appendix F Fault/Settings Switching

Position 1– Fault: Elevator position output gain error (Analogue output only)

Position 2 – Setting: Enables Uni directional control of the cart (Analogue input only)

Position 3 – Fault: Elevator door stuck open

Position 4 – Fault: Floor 2 UP call pushbutton stuck on

Position 5 – Fault: Floor 2 Elevator in car pushbutton permanently on

Position 6 – Fault: Floor 2 sensor permanently off

Position 7 – Fault: Floor 3 cart sensor shows car is permanently at floor 3



ELEVATOR TRAINER

Appendix F

Fault Switching Settings

Notes



21 Appendix G Routine Maintenance

The unit requires no routine maintenance other than occasional cleaning. The exterior may be cleaned using a damp cloth. Depending on the environment, any dust build-up around the drive mechanism should be removed using a vacuum cleaner or soft brush (remove the transparent covers as necessary). Do not lubricate any parts of the drive mechanism.

Remember to refit the covers before operating the equipment.

21.1 Refitting the chain

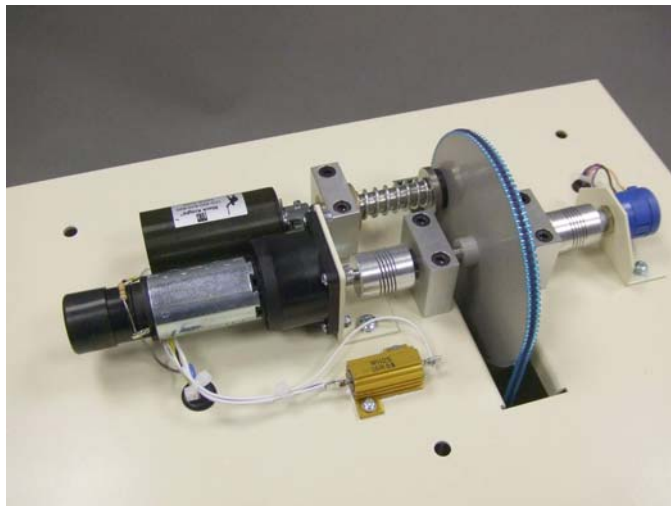


Figure 21-1

Under certain conditions the chain may become detached from the drive sprocket. Refitting is a straightforward procedure but it is very important to follow the instructions given below carefully to avoid disturbing the alignment of the position feedback potentiometer. Refer to Figure 21-1.

Switch off the unit.

Connect a voltmeter, set to a 10 V dc range, between the Position output on the rear panel (connector J2 pin 6) and 0V (connector J2 pin1).

Remove the top transparent cover. You are now going to drive the motor without the chain connected so that the position potentiometer can be aligned correctly. Make sure that the chain is not around the sprocket and the car is free to move.

Ensure that nothing can become entangled in the mechanism whilst the top cover is removed.

Switch on the unit. Watching the position output voltage displayed on the voltmeter, operate the Manual switch either up or down until the voltage reads $0\text{ V} \pm 0.2\text{ V}$. The Manual switch may need to be operated momentarily in either direction to finely adjust the output for zero.

Switch off the unit.

Remove the front transparent cover.



Gently lower the car to the bottom of the shaft so that the counterbalance is now at the top. Put your hand inside the shaft and support the counterbalance so that the chain is slack. Using the other hand, feed the chain around the sprocket then slowly release the counterbalance and allow it to take the weight.

The chain is now refitted correctly. Replace the top and front covers.

Now check and adjust the position offset as described in Section 21.3.

21.2 Load measurement offset adjustment

The zero offset of the load signal will vary over a period of time and may be adjusted if required. This operation is not normally necessary when using an external programmable controller as the offset can be removed by the software.

To manually trim the offset perform the following:

Connect a voltmeter, set to a 10 V dc range, between the Load Cell output on the rear panel (connector J2 pin 9) and 0 V (connector J2 pin1).

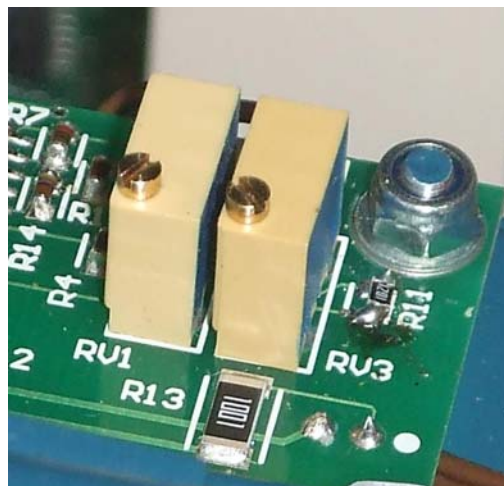


Figure 21-2

Switch on and open the car door either using the PLC (if connected) or by temporarily connecting the door control input (J4 pin 1) to +24 V (connector J10 pin 1). Adjust the trim potentiometer RV3 to give a reading of 0 V \pm 0.1 V. Refer to Figure 21-2. Place both 0.35 kg weights in the cart, then adjust the trim potentiometer RV1 to 2 V \pm 0.1 V. Remove one 0.35 kg weight and the output should show 1 V \pm 0.1 V.

Remove both weights, switch off and remove the voltmeter and any temporary connections. Refit the front transparent cover. The offset is now adjusted correctly.

21.3 Position pot offset adjustment

The zero offset of the position signal will vary over a period of time and may be adjusted if required. This operation is not normally necessary when using an external programmable controller as the offset can be removed by the software.



To manually trim the offset perform the following:

Connect a voltmeter, set to 10 V dc range, between the Position output on the rear panel (connector J2 pin 6) and 0 V (connector J2 pin1).

Switch on and use the Manual switch to move the car to the bottom of its travel so that the Lower Travel Limit LED is illuminated.

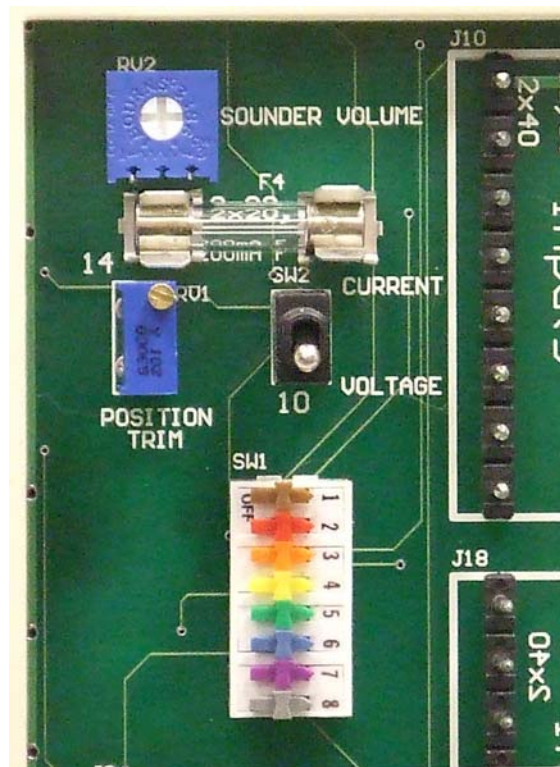


Figure 21-3

Adjust the Position Trim pot located on the rear of the unit so that the voltmeter reads 0 V \pm 0.1 V. Refer to Figure 21-3.

Switch off and remove the voltmeter. The position offset is now adjusted correctly.

21.4 Output protection fuse

The digital outputs are protected by a fuse in case they come into direct contact with a supply rail. The fuse F4 is located on the rear panel and must be replaced by one of the same type (200 mA fast acting type F200 mA). Refer to Figure 21-3.



ELEVATOR TRAINER

Appendix G

Routine Maintenance

Notes