RELAY-BEE

Automation Adaptor

Installation and Users Manual Ver 3.3
(Including Bee-Step*14* Software)

Available exclusively from
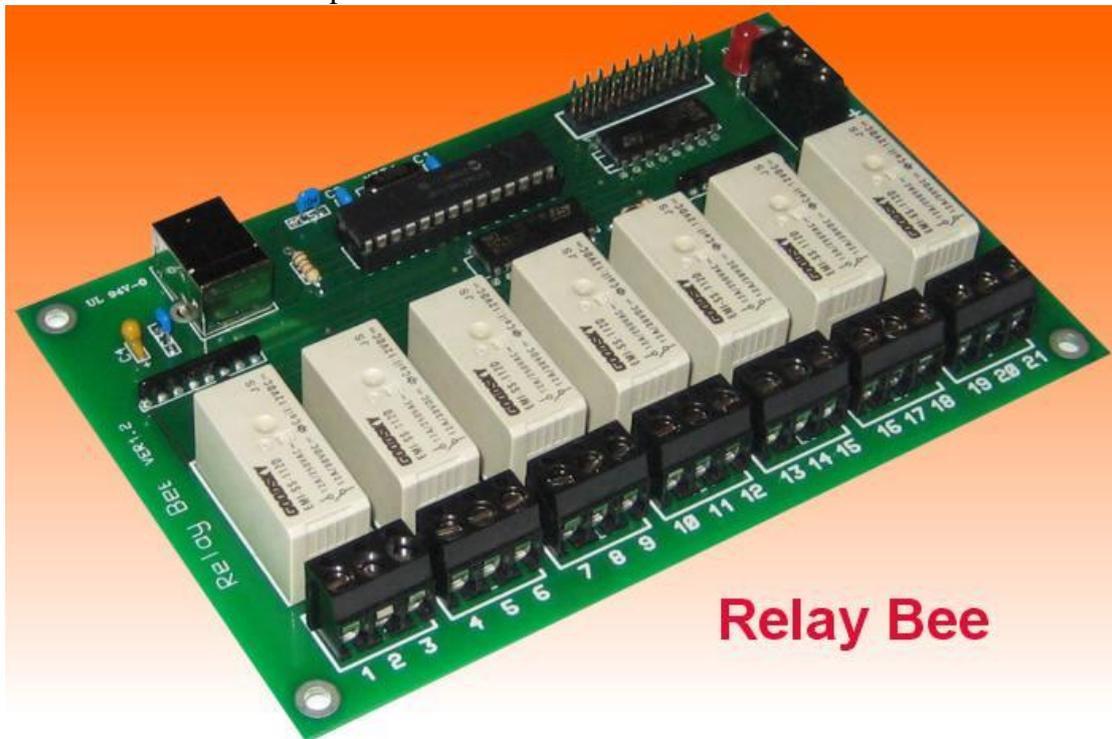PC Control Ltd.
www.pc-control.co.uk

Contents

1.   Introduction

The RELAY-BEE is a versatile USB adaptor, which allows the PC User to explore the world of real time control and automation. It is a tool, which is attractive to both the novice and experienced user.

For the absolute beginner it can be used straight from the box as a flexible controller for a wide range of projects. The beginner can take advantage of the ease of connectivity of USB, making connection to the PC simple. Screw terminals for the relay connections mean that external devices to be controlled can be attached without any need for soldering. The included Bee-Step*14* application software allows the beginner to quickly create elaborate control sequences without any need for prior programming knowledge or PLC techniques.

For the intermediate user a DLL is provided to allow the programmer to construct their own software applications to take advantage of the RELAY-BEE hardware without having to know the details of USB communication protocols etc..

WARNING: The Relay-Bee adaptor board is fitted with relays that are capable of switching high voltages and high currents. Use of the Relay-Bee in any intended application needs careful design paying particular attention to the safety aspects, including a suitable enclosure. i.e. if used for high voltages, it should not be used simply as supplied. No provisions have been made to safeguard the potentially dangerous exposed relay terminals on the board or the board tracks connecting them. If you intend to switch high voltages or currents in your application then you need to address this issue. If you have any doubts about how to use the board safely then you should consult a qualified electrician.

2   Hardware Installation

Simply connect the RELAY -BEE to any available USB port (*This will require a standard USB cable*). Although it will operate from bus powered hubs it is recommended that you connect it to a primary USB socket or a self powered hub.

This allows RELAY -BEE to take full advantage of the available 500mA from such a connection. Bus powered hubs are limited to 100mA. Windows operating system will automatically detect and install the appropriate device drivers. The RELAY -BEE is regarded by Windows as a standard HID (Human Interface Device) which makes it very easy to install.

3. Bee-Step*14* Application Software

3.1 Overview

When the Bee-Step*14* application is started, click on the "Run" menu option to initiate the control dialog box (shown below). The first time Bee-Step14 is run the customary disclaimer agreement will ask for your agreement. Click "I Agree" to never see it again and to start Bee-Step*14*.

Bee-Step*14* provides the user with a means to edit, save, restore and run a sequence of "patterns" of outputs that will be used to directly control outputs on the attached Relay-Bee adaptor board. This is known as a Directed Sequencer. The sequence is made up of up to 10000 individual "steps". Each step has three elements

- A duration:
- A pattern of outputs (on or off for each of the outputs)
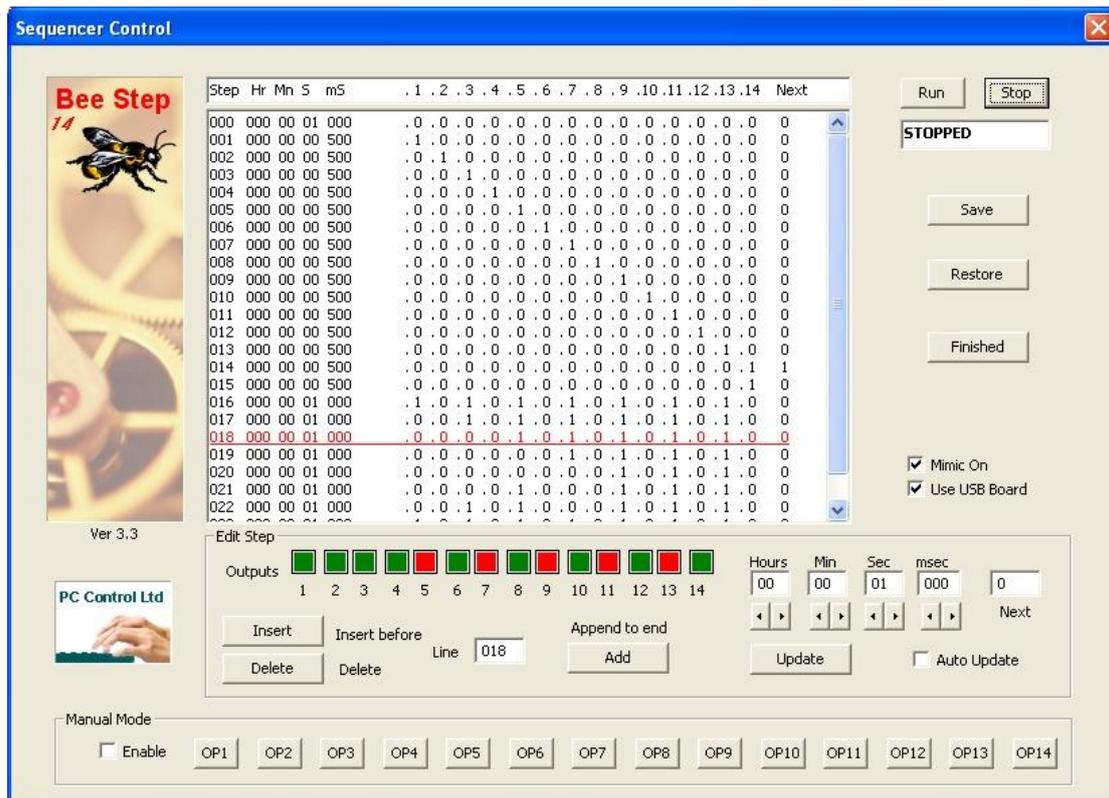- The number of the next step to be executed

The duration is specified in Hours, Minutes, Seconds and milliseconds. The overall resolution of the time interval is 0.1 seconds. This provides a realistic level of control for devices in the "real world" of automation and robotics.

The pattern of outputs corresponds to the on / off status of the actual control outputs on the Relay-Bee adaptor. All outputs are independent and any desired pattern can be used for each step.

A simple sequence controller sets the outputs for each step in turn finishing when it reaches the end of the list. Bee-Step*14* can operate this way if you simply ignore the "next" element of the step. However, to create more elaborate sequences including continuous loops, you can use the "next" element to specify which step to go to when the current step is complete (i.e. duration reached).

3.2 Bee-Step*14* Main Screen

When you first run BeeStep14 you will start from the main menu screen. Click on the menu item "Run" to open the sequence editor. We can now look at each of the elements of the screen in turn….

Sequencer Control

Bee Step
14

Ver 3.3

PC Control Ltd

Step	Hr	Mn	S	mS	.1 .2 .3 .4 .5 .6 .7 .8 .9 .10 .11 .12 .13 .14	Next
000	000	00	01	000	.0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0	0
001	000	00	00	500	.1 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0	0
002	000	00	00	500	.0 .1 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0	0
003	000	00	00	500	.0 .0 .1 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0	0
004	000	00	00	500	.0 .0 .0 .1 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0	0
005	000	00	00	500	.0 .0 .0 .0 .1 .0 .0 .0 .0 .0 .0 .0 .0 .0	0
006	000	00	00	500	.0 .0 .0 .0 .0 .1 .0 .0 .0 .0 .0 .0 .0 .0	0
007	000	00	00	500	.0 .0 .0 .0 .0 .0 .1 .0 .0 .0 .0 .0 .0 .0	0
008	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .1 .0 .0 .0 .0 .0 .0	0
009	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .0 .1 .0 .0 .0 .0 .0	0
010	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .0 .0 .1 .0 .0 .0 .0	0
011	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .1 .0 .0 .0	0
012	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .1 .0 .0	0
013	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .1 .0	0
014	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .1	1
015	000	00	00	500	.0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .1	0
016	000	00	01	000	.1 .0 .1 .0 .1 .0 .1 .0 .1 .0 .1 .0 .1 .0	0
017	000	00	01	000	.0 .1 .0 .1 .0 .1 .0 .1 .0 .1 .0 .1 .0 .1	0
018	000	00	01	000	.0 .0 .0 .1 .0 .1 .0 .1 .0 .1 .0 .1 .0	0
019	000	00	01	000	.0 .0 .0 .0 .1 .0 .1 .0 .1 .0 .1 .0 .1 .0	0
020	000	00	01	000	.0 .0 .0 .0 .0 .1 .0 .1 .0 .1 .0 .1 .0	0
021	000	00	01	000	.0 .0 .0 .1 .0 .1 .0 .1 .0 .1 .0 .1 .0	0
022	000	00	01	000	.0 .0 .1 .0 .1 .0 .0 .1 .0 .1 .0 .1 .0	0

Run   Stop

STOPPED

Save

Restore

Finished

☑ Mimic On
☑ Use USB Board

Edit Step

Outputs
1 2 3 4 5 6 7 8 9 10 11 12 13 14

Insert   Insert before   Append to end   Line 018   Add
Delete   Delete

Hours 00   Min 00   Sec 01   msec 000   0

Next
Update   ☐ Auto Update

Manual Mode
☐ Enable   OP1 OP2 OP3 OP4 OP5 OP6 OP7 OP8 OP9 OP10 OP11 OP12 OP13 OP14

3.2.1 Main Sequence Display Window Sequence Summary List

The sequence summary window shows each individual step in a programmed sequence in summary form. Each line in the window corresponds to a single step in the sequencer. On the left of each line is the line number. This will always be consecutive from 1 to your last entered line. When editing the sequence list using insert or delete (discussed later), the line numbers will be automatically adjusted so that they remain in strict numerical sequence. This line number is used as the reference for the "next" facility (also discussed later) to control sequence loops and jumps. Immediately to the right of the line number is the duration. This is split into Hours, Minutes, Seconds and Milliseconds. The smallest duration is 100msec (0.1sec). The next 14 columns show the on / off status that the Relay-Bee board outputs will use during this step. Remember that the first 7 outputs are the relays and the second 7 are the ordinary switching outputs.   On is shown as '1' and off as '0'. On the right hand end of the line is the "next" selection. This determines which line will be used after this one. A value of 0 in the next position indicates that the next line immediately below the current one will be used. For sequences larger than the sequence summary window can display, the window may be scrolled up and down using the scrollbar on the right

3.2.2 Step Editing

The step editing section immediately below the summary window allows individual steps to be added, deleted and edited.   Each of the elements of a step is available in this area in editable form. The row of boxes along the top corresponds to the step's on/off pattern of outputs. Green is off and red is on. Clicking on the box will toggle it between on and off.   The duration of the step has individual edit boxes for hours, minutes, seconds and milliseconds. Using the adjacent arrows allows new values to be specified. In a similar way the value of the next step to be run can be

edited. Note that the last step in a sequence will always cause the sequence to stop even if the "next" box is set to another location (eg to loop back). To avoid this add a dummy step after the one in which you wish to loop back.

To edit a step that already exists in the summary window simply click on that line in the window. This will cause a red underline to appear at that line and a copy of the step elements to appear in the edit area. Once editing is complete you can update the step in the sequence list by pressing the "Update" button. Alternatively you can check the "Auto Update" box and see the changes you make updated immediately.

To create a new step to be added to the end of the current sequence make sure the "Auto Update" box is unchecked, construct your step in the edit area and then click on the "Add" button.

To insert a new line somewhere in the middle of the current sequence, again make sure the "Auto Update" box is unchecked, click on the line BELOW the intended point of insertion and, when your editing of the step is complete, click on the "Insert" button.

To delete a line currently in the sequence list, simply click on the line and press the "Delete" button. To preserve the consecutive line numbering, all lines below the deleted line will be automatically moved up one and correspondingly re-numbered. Note that step 0 in the sequence list cannot be deleted or have other steps inserted above it.

Once a control sequence has been created to your specifications it can be saved to hard drive using the "Save" button. Files created in this way can later be restored using the "Restore" button.

3.2.3   Run Control

On the right of the main screen there are two buttons for run control, which are virtually self-explanatory. Clicking "Run" starts execution at the first step (0) in the sequence. It will continue to "run" until either the last step is reached or the "Stop" button is pressed. If the stop button is pressed, the current state of the outputs will be maintained. Care should be taken when arbitrarily stopping the running sequence since it may stop with an output permanently on that shouldn't be (for example it may leave a motor running continuously).

While running it is possible to see the current state of the outputs changing on the screen in real time mimicking the real outputs. This is selected by checking the "Mimic Display On" box. Normally there is no problem in running the mimic concurrently with the real time control sequence, however, if the control sequence involves very short duration steps and/or your PC is relatively slow it is better to turn off the mimic in favour of the real time control outputs.

It is good practice to test your sequence before actually applying it to the Relay-Bee outputs. To do this, simply uncheck the "Use USB Board" box. The mimic will run but no outputs will change on the Relay-Bee board.

3.2.4   Manual Mode

At the bottom of the screen you will see a row of buttons labelled OP1 to OP14. These correspond to the 14 outputs on the Relay bee. Pressing one of these buttons will "toggle" the corresponding output immediately. i.e. if the output is currently "on" it will be turned "off" and if "off" it will be turned "on". This provides a simple mechanism to test your connected hardware before attempting to construct sequences of operations as described above.

In order to make use of Manual Mode, you must first check the box labelled "Enable" within the manual mode area on screen. This temporarily disables the normal editing facilities provided in the automatic sequence area. i.e. you can't use both auto and manual at the same time. Clicking on "Mimic On" or "Use USB Board" will revert back to normal automatic mode enabling sequence editing again.

4.   Connecting Devices to Relay-Bee

The Relay-Bee board is designed to be flexible in its uses and has a number of electrical features that make it easy to use in many applications. The following descriptions are merely suggestions on suitable ways to use the unit.

4.2   Using the Relay Outputs

There are 7 relay outputs available. These are referred to as outputs 1 to 7 in the BeeStep software and the DLL function library. Each relay provides a SPDT (single pole double throw) type output, otherwise known as a "changeover" type. This means that each relay has three external connection points. One for the common, one for the normally open and one for the normally closed connections. This is illustrated below…
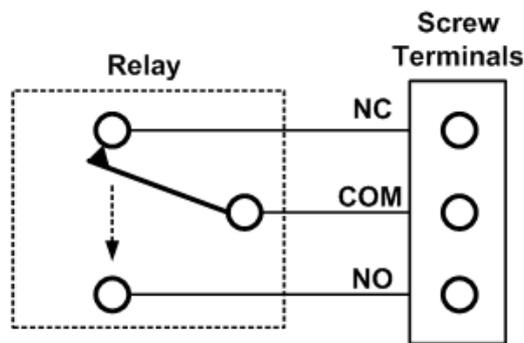
For those unfamiliar with relay terminology the state of the connections are as follows. In the un-energised state (i.e. power off or output off) the common will be connected to the normally closed. And, of course, in the energised state (output on) the common will be connected to the normally open. For example, when output 1 is off screw terminal 2 is connected to terminal 3. When output 1 is on , screw terminal 2 is connected to terminal 1. (similarly for the other sets of 3 terminals 4,5,6 …… 19,20,21)

Connecting external devices to any or all of the relay outputs is very simple. First ensure your device is within the specified voltage and current specifications of the Relay Bee. Then decide if you need a normally open or normally closed switch. Refer to the pinout details supplied later in this manual and choose the terminals to connect to.   When using the relay outputs you will need to connect an external 12v DC supply to the terminals TL8 labelled as (12v + and −) near the top right of the board. This supply is needed to provide the coil voltage for the relays. The current requirements for this supply is a modest 300mA max.   The red LED (D8) will illuminate when your external 12v supply is on. As an example, the connection of an external device to relay 1 is shown below..
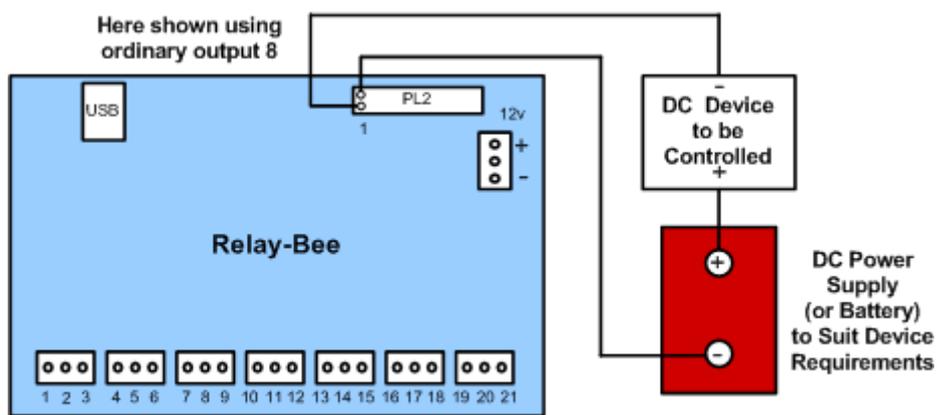
Connecting External Devices to Relay-Bee

Here shown using the normally open contacts of relay 1
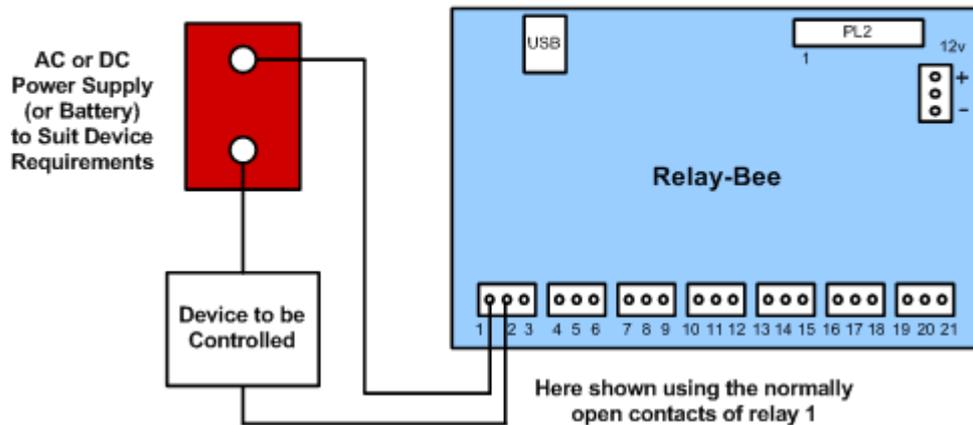
When switching inductive loads such as motors or solenoids, particularly those requiring high currents, it may be necessary to fit contact suppressors across these loads to prevent unwanted arcing within the relay contacts and electrical interference emissions. Due to the wide nature of devices that can be attached, the type of suppressors that may be needed are beyond the scope of this manual.

4.2   Using the Ordinary Switching Outputs

In addition to the relay outputs there are 7 ordinary DC switching outputs available on the 20 way header (PL2) with each output taking the form of an "open collector" driver. These are outputs referred to as 8 – 14 within BeeStep software and the DLL function library.   It is recommended to use a 20way ribbon cable terminated with a standard 20-way IDC type plug to connect to PL2. For these outputs to operate correctly it is necessary to link the 0v (or Ground) connection of the Relay-Bee to the 0v connection of the external power supply which is being used to "drive" the device to be controlled (eg a motor, solenoid, lamp etc…). The 0v connection is available on all even numbered pins on PL2 (i.e. 2 – 20)

Connecting Lower Power DC Devices to Relay-Bee

The device being controlled is then connected between one of the Relay-Bee control outputs (e.g. OP8 on PL2/1) and the external positive end of the supply (e.g. +12v).

When the Relay-Bee output is turned on (by the control step in BeeStep14 or by your own software) the terminal becomes a low impedance path to ground and current flows in the external circuit through the attached device. For low power non-inductive devices such as lamps, that is all that is necessary. If you are directly attaching an inductive load (such as a motor or relay) it is advisable to take precautions against switching transients.
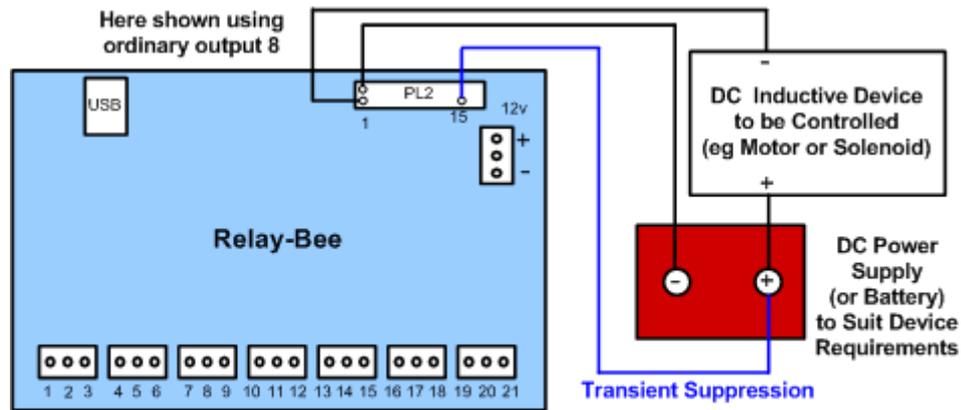
Connecting Lower Power DC Inductive Devices to Relay-Bee

Switching transients are spikes in the voltage that occur when an inductive load is turned off and can be high enough to cause damage to attached circuitry. Relay-Bee has built in facilities to suppress these transients by using suppressor diodes connected to PL2 pin 15). To make use of this simply connect this pin to the external positive supply being used to "drive" your inductive load.

The ordinary output switching component used by Relay-Bee is the DS2003 High Current / High Voltage Darlington Driver. These devices are capable of being switched up to 50v and 500mA. However, although capable of these operating limits, the recommended application of the Relay-Bee is for voltages up to 24v. The current capability on each output is 500mA. As the ULN2003 data sheet suggests, this can be extended by connecting outputs in parallel. If doing this, it is obviously essential that your control sequence ensures that all outputs that are paralleled are always in the same state (on/off) at the same time or you run the risk of one output taking all of the current and exceeding maximum limits. You must also take account of the overall power handling capability of the ULN2003 when using multiple outputs each with high current. Refer to the graph of "peak collector current vs duty cycle and number of outputs" in the ULN2003 data sheet included on the installation CD.

5   Connector Pinouts

Pinout of the Relay Outputs On Screw Terminals(1 - 21)

Terminal	Signal description
1	Relay 1 Normally Open
2	Relay 1 Common
3	Relay 1 Normally Closed
4	Relay 2 Normally Open
5	Relay 2 Common
6	Relay 2 Normally Closed
7	Relay 3 Normally Open
8	Relay 3 Common
9	Relay 3 Normally Closed
10	Relay 4 Normally Open
11	Relay 4 Common
12	Relay 4 Normally Closed
13	Relay 5 Normally Open
14	Relay 5 Common
15	Relay 5 Normally Closed
16	Relay 6 Normally Open
17	Relay 6 Common
18	Relay 6 Normally Closed
19	Relay 7 Normally Open
20	Relay 7 Common
21	Relay 7 Normally Closed

6.   Pinout of the Ordinary Switching Outputs On PL2 (20 Way Header)

Pin	Signal description
1	Switching Output 7
2	GND
3	Switching Output 6
4	GND
5	Switching Output 5
6	GND
7	Switching Output 4
8	GND
9	Switching Output 3
10	GND
11	Switching Output 2
12	GND
13	Switching Output 1
14	GND
15	Transient Suppression
16	GND
17	N/C
18	GND
19	N/C
20	GND
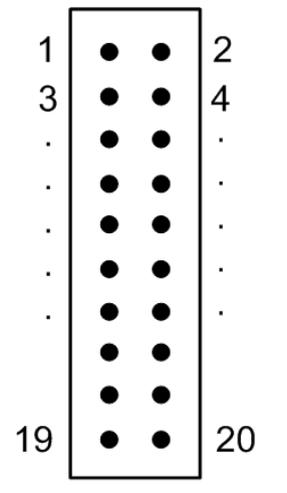
PL2
Pin Arrangement

7. Writing your own software for Relay-Bee

To use Relay-Bee straight from the box does not require any programming other than entering the step details into BeeStep14. However, if you prefer to design you own software then the following information will be of use.

Provided with Relay-Bee is a DLL (dynamic link library) called "mb.dll". This encapsulates the functions used by BeeStep14 in communicating with Relay-Bee across the USB interface into a few simple functions easily understood and used in custom software. Although the DLL was written in 'C' it can be used (called) by programs written in a number of popular languages including BASIC (visual BASIC etc..).

7.1 Writing Visual Basic Programs for Relay-Bee

The DLL provides a general-purpose interface that greatly simplifies the task of writing programs for a USB device. It can be tricky manipulating the USB comms into sending and receiving messages to and from a device, which can easily be plugged and unplugged at any time. The DLL eliminates all of these headaches by simplifying the task into two library functions.

InitMbee()   and   SetOutputs(outputs)

InitMbee () is called somewhere near the start of your program and takes care of all of the USB comms initialisation and prepares the Relay-Bee for receiving messages.

SetOutputs(outputs) can then be called at any time during your program to set the output pattern of on's and off's. The parameter Outputs is simply a 32 bit integer value where bit0 corresponds to output 1, bit 1 to output2, etc... Where a logic value of 1 turns the output on and a value of 0 turns it off. For example the statement below would turn on the first three outputs (ie the first three relays)...

SetOutputs(7)

The only other thing that a VB program must do is to declare the functions that it is going to use within the DLL and the name of the DLL itself. This must be done at the start of your program or at least before any references to the two functions are made. The following is an program excerpt showing how this is done...

Declare Function InitMbee Lib "mb.dll" () As Boolean
Declare Function SetOutputs Lib "mb.dll" (ByVal Outputs As Integer) As Boolean

The first declaration states that the function BeeInit has no parameters, is found in mb.dll and returns a boolean value. The second states that SetOutputs has one integer parameter passed by value rather than reference, is found in mb.dll and also returns a boolean value. It should be noted that the   ….Lib "mb.dll"….   part lets the program know where to find the mb.dll file. When written like this it assumes that since there is no path information that the bee.dll file can be found in the windows system directory   c:\windows\system32   If you like you can copy the file mb.dll on the installation disk to the system32 directory and the above statement will work perfectly. Alternatively you can copy the file to some other location and give that location in the declaration as below...

Declare Function InitMbee Lib "c:\library\mb.dll" () As Boolean

To speed up your development of software for the Relay-Bee, a complete working example is available in the directory VBmini. It creates a very simple form based program that has individual buttons for various functions such as initialising the Relay-Bee and setting various patterns on the outputs. The main (and only) screen is shown on the right. This has been written using Microsoft Visual Studio .net and the directory contains the full workspace (solution) details to allow you to immediately open and start editing or running.

Even if you don't have Visual Studio, the source code is virtually self explanatory with the most relevant sections being in "Form1.vb" which can even be opened in a simple text editor such as notepad.
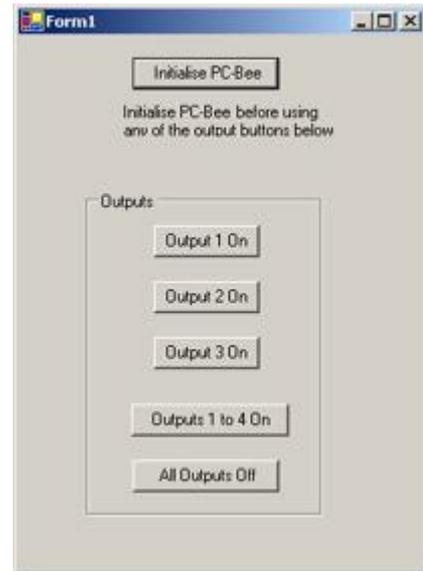
7.2 Writing Visual C++ programs for Relay-Bee

Ignoring some of the formalities in the construction of a Visual C++ program for the windows environment the techniques in using "mb.dll" consists of four main tasks….

Loading the DLL into memory

Before any functions within the DLL can be used it is necessary to instruct windows to load it into memory. This is done by calling the LoadLibrary() function. i.e.

……

HINSTANCE BeeHandle;   // declaration of variable to hold the handle to the dll

….

BeeHandle = LoadLibrary("mb.dll");// load the dll into memory and return handle

The declaration of the variable BeeHandle used to store the handle to a DLL , uses a built in type definition which is called HINSTANCE in this particular 'C' compiler, but you should use the appropriate one defined in your own compiler for this purpose.

The LoadLibrary() function return a handle to the DLL if the load is successful otherwise NULL. Ideally your own program should check for a NULL returned and give an error message. Make sure the function parameter is the full pathlist to where you copied the bee.dll file from the installation CD.

Get the addresses of the functions within the DLL

Using the DLL handle returned above you can now obtain pointers to the functions within the DLL. Using the following

Type_InitMbee   InitMbee;
Type_SetOutputs   SetOutputs;

……..

InitMbee = (Type_InitMbee)GetProcAddress(BeeHandle, " InitMbee ");
SetOutputs=(Type_SetOutputs)GetProcAddress(BeeHandle, "SetOutputs");

………………

The Type_InitMbee and Type_SetOutputs type definitions are contained in the header file "mbdll.h" and defines the correct type of function pointer to reference the DLL function. mbdll.h should be included in your own source file eg.

…………..

#include "mbdll.h"

…………..

The call to GetProcAddress() returns a pointer to this function if found within the DLL otherwise NULL. Once the functions pointers have been obtained in this way the internal functions within the DLL are simply accessed like ordinary function calls e.g.

…………..

InitMbee ();
SetOutputs(0x1234);

……………

Initialising The DLL

Once the addresses of the DLL functions are obtained as above the remaining functions required to use them are very simple. The first step is to initialise the DLL using….

int status;

……..

status = InitMbee();

Your program should check to see if a value of zero has been returned by InitMbee (). Any other value indicates an error. E.g. Relay-Bee not connected etc…

Using the SetOutputs() Function

The majority of your programming will use the SetOutputs() function. This simply applies the pattern 1's and 0's of the first fourteen bits of the 32 bit parameter directly to the outputs. For example: to create a pattern of alternate on and off over all outputs use..

SetOutputs(0x00001555);   // hexadecimal number

Or to turn on just output 1 only use …

SetOutputs(0x00000001);   // hexadecimal number
Etc……….

More generally…..

unsigned long bits;
………..
bits = 0x00001234;   // hexadecimal number
SetOutputs(bits);
………..

This example will turn on outputs 13, 10, 6, 5 and 3.

Although this only gives a glimpse of the possibilities of writing your own programs, it should be apparent that the use of the DLL functions greatly simplifies this process. It frees the programmer from the task of getting to know the fine details of programming USB interface communications and lets him concentrate on the main function of controlling switching outputs.

8.   Minimum PC System Requirements

Relay-Bee and BeeStep*14* software do not require a high spec PC for correct operation, but the following system is suggested as a sensible minimum

Processor	500MHz Pentium
Memory	64MB
HDD	10MB free space required
Screen Resolution	1024x768 (256 colours)
Interface	One free USB socket (1.0 or 2.0)
Operating System	Windows XP (or later)

9   Board Specifications

Dimensions	140mm x 90mm
Mounting Holes	4 x 3.5mm Diameter
Weight	250g

Relay Outputs (Outputs 1-7)

Number	7 Independent
Voltage Rating	250v AC / 110v DC Max
Current Rating	6A Max
Contact Form	SPDT (Changeover)
Operate Time	12ms Max
Release Time	8ms Max

Ordinary Outputs (Outputs 8-14)

Number	7
Voltage rating	24v DC Max
Current Rating	500mA Max
Form	Open Collector

Board Power Requirements

5v	From USB
12v/300mA	For relay coils. Requires external supply

USB	1.0

IMPORTANT PLEASE READ CAREFULLY

WARNING: The Relay-Bee adaptor board is fitted with relays that are capable of switching high voltages and high currents. Use of the Relay-Bee in any intended application needs careful design paying particular attention to the safety aspects including a suitable enclosure. i.e. if used for high voltages, it should not be used simply as supplied. No provisions have been made to safeguard the potentially dangerous exposed relay terminals on the board or the board tracks connecting them. If you intend to switch high voltages or currents in your application then you need to address this issue. If you have any doubts about how to use the board safely then you should consult a qualified electrician.

RelayBee
Regulatory Compliance and Safety Information

Product Name: RelayBee
Part No. BRD008

IMPORTANT PLEASE RETAIN THIS INFORMATION FOR FUTURE REFERENCE

Warnings
- This product should not be connected to mains voltages under any circumstances.
- This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.
- The connection of non CE compliant devices may affect overall compliance or result in damage to the unit and invalidate the warranty.

Instructions for safe use
- To avoid malfunction or damage to your board please observe the following:
- Do not expose it to water, moisture or place on a conductive surface whilst in operation.
- Do not expose it to heat from any source; the MotorBee is designed for reliable operation at normal ambient room temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Avoid handling the board while it is powered. Only handle by the edges to minimize the risk of electrostatic discharge damage.
- All peripherals used with the board should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met.

Compliance Information
- The board complies with the relevant provisions of the RoHS Directive for the European Union.

WEEE Directive Statement for the European Union
- In common with all Electronic and Electrical products the board should not be disposed of in household waste. Alternative arrangements may apply in other jurisdictions.

EMC Compliance Statements
European Union (EU) Electromagnetic Compatibility Directive Compliance Statement
This product is in conformity with the protection requirements of EU Council Directive 2004/108/EC on the approximation of the laws of the Member States relating to electromagnetic compatibility.
Warning: This is equivalent to an EN 55022 Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

PC Control Ltd. 18 Beech Close, Desborough, Kettering, Northants NN14 2XQ, UK
www.pc-control.co.uk

Terms of Use for all Goods Supplied

Definitions

'Supplier' shall mean PC Control Ltd.

'Buyer' shall mean the person, company or any other body that purchases or agrees to purchase Goods.

'Goods' shall mean all goods and services which the Buyer agrees to buy from the Supplier including replacements for defective Goods, hardware, documentation and software products licensed for use by the Buyer.

Use of the Goods in any way by the Buyer constitutes acceptance of these terms and conditions.

Terms and Conditions

1. The Goods are intended to be part of the buyer's own design of apparatus and not a finished product in their own right.
2. The Goods supplied are not to be used in any design where there is a risk, however small, either directly or indirectly, of death or personal injury.
3. The Buyer will be responsible for ensuring the fitness for purpose of the Goods for the Buyer's application.
4. To the extent permitted by law, the Supplier accepts no liability whatsoever or howsoever arising in respect of loss, damage or expense arising from errors in information or advice provided whether or not due to the Supplier's negligence or that of its employees, agents or sub-contractors save for any loss or damage arising from death or personal injury.
5. To the extent permitted by law, the Supplier shall not be liable to the Buyer by reason of any representation (unless fraudulent), or any implied warranty, condition or other term, or any duty at common law, or under the express terms of any Contract with the Buyer, for any indirect, special or unforeseen loss or damage (whether for loss of profit or otherwise), costs, expenses or other claims for compensation whatsoever (whether caused by the negligence of the Supplier, its employees or agents or otherwise) which arise out of or in connection with the supply of the Goods or their use or resale by the Buyer.
6. The entire liability of the Supplier under or in connection with the Contract with the Buyer shall not exceed the price of the Goods except as expressly provided in these terms and conditions.
7. These terms are an important part of the full terms and conditions of business as published on the website at www.pc-control.co.uk/general-terms.htm which also apply.

If you cannot agree to the terms and conditions of use of the RelayBee then you should return the RelayBee to the supplier within 7 days of receipt to receive a refund. Your use of the board or the associated software in any way whatsoever will be regarded as an acceptance of these terms and conditions.