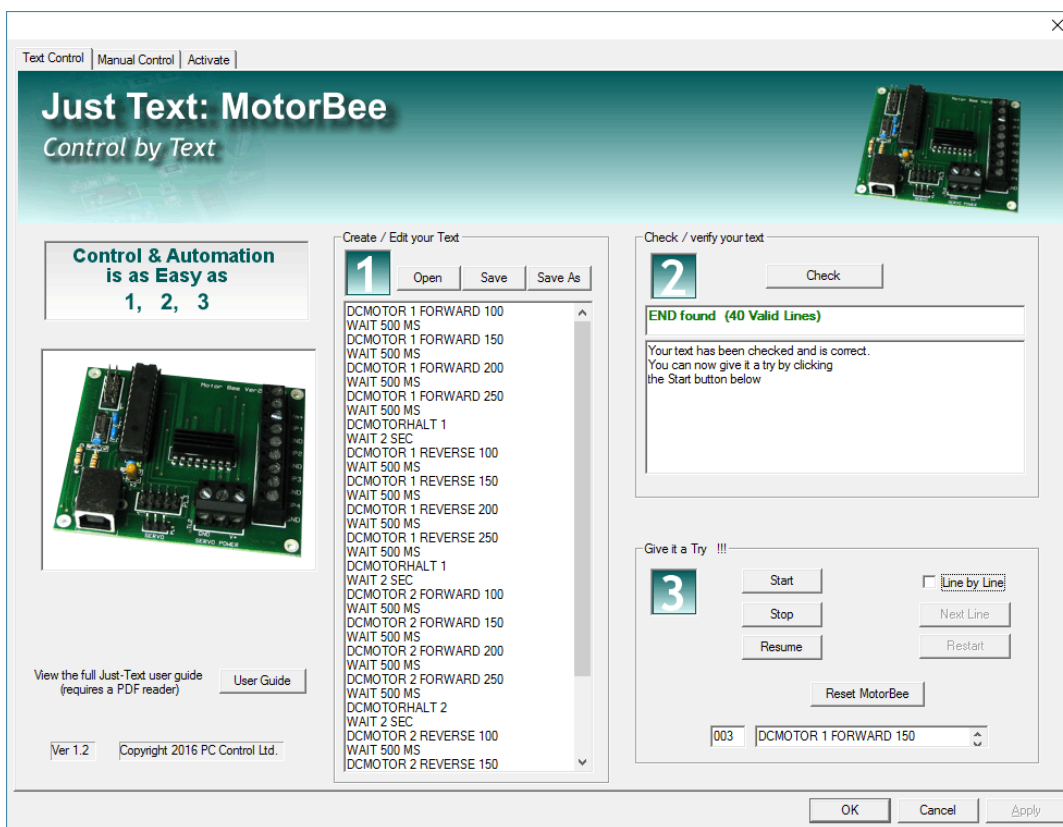


# Just-Text : MotorBee

## "Control by Text"

### Installation and Users Manual



Available exclusively from  
PC Control Ltd.  
[www.pc-control.co.uk](http://www.pc-control.co.uk)

© 2016 Copyright PC Control Ltd.

Revision 1.2

## **Contents**

- 1. Introduction**
- 2. Software Installation**
- 3. Using Just-Text Software**
- 4. Text Reference**
- 5. Text Examples**
- 6. Product Activation**
- 7. Minimum PC System Requirements**
- 8. Terms of Use**

## 1. Introduction to Just-Text

The world of control and automation using motors, solenoids, lamps and all sorts of other devices has long been restricted to those who can write computer programs. "Just Text" opens up this world to non-programmers (and programmers who want it a bit easier). If you can create a simple text file on your computer then you can operate this world of devices and build your own automation system.

### What do we mean by "Control and Automation"

It sounds a glamorous title but it's simply a way of making devices such as motors, solenoids, lamps etc. behave in a pre-determined way automatically. This can be for something as simple as an on-off timer for lights to a more complex motion control application where motors move objects around in a precisely defined sequence. Although widely used in the industrial environment, automation is becoming increasingly used in the "hobbyist" area. A good example would be as a model train controller, where the motion of the trains and their speed is governed by the position of the track points and signals with horns and lighting added to the mix. It is also quite popular with photographers who want to move the camera at regular intervals in a pre-defined sequence taking photos at each position. Astronomers may use it to alter the position of the telescope making it travel automatically to a specific portion of the sky. Shop owners very often employ window displays which automate all sorts of animatronic figures, especially during the festive periods. The scope of control and automation is nothing short of vast. It is really down to the imagination and creativity of the user what can be done.

In most cases the control system includes three main elements: namely, inputs, outputs and control logic. Inputs are the means to let the controller know what is happening in the real world and can be as simple as a switch to a complex motion detector or temperature sensor. Outputs are the "things" which are controlled. These can be simple on-off lights to speed and direction of motors. The control logic is the "rules" by which these elements know what to do. Decades ago, these "rules" were actually "hard wired" into arrangements of relays and switching devices. Today the vast majority rely on computing power and a set of instructions for the computer. The latter offering the best option for flexibility and ease of change.

### How does Just-Text perform "Control and Automation" for the MotorBee

The MotorBee has electronic components which provide for "reading" inputs, switching outputs, controlling motors in both speed and direction and positioning a servo. Specifically....

Inputs: - These are standard digital type; which means that it will recognise if an input is on or off. This can be from a simple switch or a more complex device which provides a voltage output corresponding to it being on or off.

Outputs:- These are simple switching type outputs which offer a means of switching on and off external devices of all types. Although the switching outputs are limited to fairly low powered devices, they can be used to operate relays which in turn can then operate almost any power of device.

Motor Control:- Control for DC motors providing speed and direction control for up to 4 motors (depending on the number of motors used some may be uni-directional).

Servo Control:- Most servos rely on a standard interface which allows it to be instructed to move to a given angular position. The Motor Bee provides this control for one Servo.

The power and simplicity of Just-Text really begins to show when it comes to specifying the control requirements for the automation. As previously mentioned, this is a simple text file. It can be created using the "built-in" text editor which is part of the Just-Text software or any other text editor which doesn't involve fancy formatting or font and colour enhancements, i.e. simple , plain text. A good example is the text editor which comes with all Windows systems called "Notepad".

Each separate line of this text file has a single instruction which will be used by Just-Text to perform the desired operation on the connected MotorBee. For example....

- **OUTPUT 2 ON** will turn on switching output number 2.
- **WAIT 10 SEC** will wait for 10 seconds before moving on to the next line of text.
- **DCMOTOR 2 FORWARD** will start DC motor number 2 running in the forward direction.

As you can see the instructions are designed to be as close to plain English as possible. Just-text software provides a checking facility to check your text for any errors and explain what is wrong. Once the text is correct you simply click on "Start" to see it operate via the MotorBee which would be connected to a USB port.

## 2. Software Installation

To install Just-Text software simply download the zip file from the PC Control Website at ...

<http://www.pc-control.co.uk/just-text-motorbee.php>

Unzip the files contained within it using your preferred "unzip" program and then double click on the file named "setup" (also shown as setup.exe). This will begin the installation process. On some windows systems, including Windows 10, you may be asked to confirm that you want to install uncertified software. You will need to confirm to allow this to proceed. The installation will create a desktop icon which can then be used to start the program.

At this stage, the software will be fully functional except for actually running the text instructions. This requires the software to be activated. Activation requires a purchase from the Control Shop online. See details at....

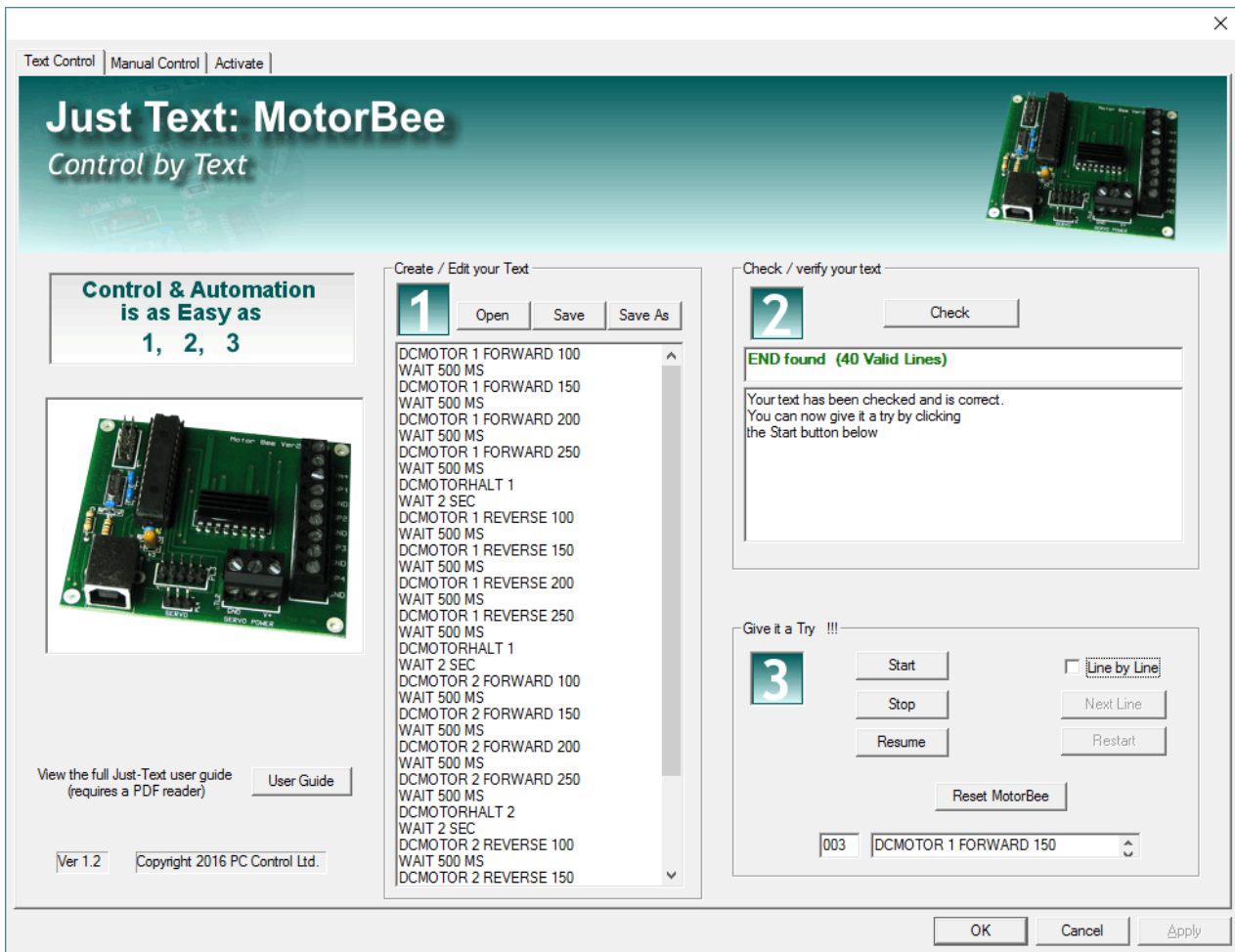
<http://www.pc-control.co.uk/ocart>

All you have to do now is connect the MotorBee to a free USB port.

### **There is no "Windows Driver" to be installed**

All Windows operating systems since XP already have the appropriate driver installed, so there is nothing more to do. As soon as you connect the board to a USB port, Windows will recognise the board and configure it automatically, ready to use. This applies to Windows XP, Vista, and Windows 7,8 and 10.

### 3. Using the Just-Text Software (*It's as easy as 1,2,3*)



Just-Text software is divided into 3 main functions.

#### **1 Create and Edit your Text**

Before you can control anything you need to create your text file. This will contain the instructions that you want the MotorBee to follow. The simplest way is to directly type your text into the text edit area on the Just-Text main screen, but, you can also create your text file using any editor you like as long as it is simple text without any type of formatting. By formatting we mean, fonts, colours and any other embellishments aimed at presentation. A good example of this type of editor is "Notepad" which is included with all installations of Windows. If you use this type of editor to create your text file then you can simply "Open" it with Just-Text and it will appear in the Edit Area ready for the next step. Whichever way you create it, you can then save it as a ".txt" file to your computer as normal.

The instructions in the text file correspond to a range of operations that can be performed by the MotorBee. The full list of these operations/instructions can be found in the reference section. In all cases, the text of the instruction is written in CAPITAL letters with a single space between the individual parts of the instruction. For example to start the DC motor number 1 running forward at speed 128 you would use

**DCMOTOR 1 FORWARD 128**

where '1' is the motor number and '128' is the speed. Each separate line is a separate instruction. Just-Text will start at line 1 and progress down the list one line at a time executing each instruction in turn. (*there are instructions which can make it "jump" to other lines but we will cover that later*).

Just-Text executes the instruction very quickly. In the case above, the DC motor will be turned on within 100ms seconds of processing that instruction (i.e. 0.1 sec). You can see, therefore that the following sequence of instructions may initially be misleading...

**DCMOTOR 1 FORWARD 100**  
**DCMOTOR 1 REVERSE 60**  
**DCMOTOR 1 FORWARD 100**  
**DCMOTOR 1 REVERSE 60**

You would think that the motor will go forward then reverse then forward again and then reverse. No. Since the first 3 instructions are processed within 300 milliseconds (100 ms each) the only one that has time to take effect is

the last one which will see the motor moving in reverse. What we need is a way of delaying the progress from one step to the next. For this we use the WAIT instruction.

The above example now becomes...

```
DCMOTOR 1 FORWARD 100  
WAIT 5 SEC  
DCMOTOR 1 REVERSE 60  
WAIT 5 SEC  
DCMOTOR 1 FORWARD 100  
WAIT 5 SEC  
DCMOTOR 1 REVERSE 60
```

Now we will see motor 1 go forward for 5 seconds then reverse for 5 seconds then forward (5 seconds) and finally reverse. As you can see the WAIT instruction is very useful and can be used with units of milliseconds, seconds and minutes. Full details on these and all other instructions can be found in the reference section.

**2**

### **Check / verify your text**

There is always the possibility of typing errors and incorrect use of instructions etc.. so your text must be checked before running it. Pressing the "Check" button will allow Just-Text to perform a thorough examination of your text and let you know of any errors. It will also provide additional information to help you correct any problems.

**3**

### **Give it a Try !!!**

You can now start the text instructions "running" by pressing the "Start" button. The current instruction being processed will appear in the "active" window. This will continually be updated until your text instructions are completed. Note that it is possible to create a continuous loop where your processing will never finish. If you want to manually stop the board then press the "Stop" button. This will cause it to stop immediately.

Pressing the resume button will cause the text processing to resume from where it left off. Re-pressing the "Start" button at any time will "re-start" your text from the beginning.

For convenience there are also some other facilities for running the text instructions. You can select "Line by Line" mode. In this mode a single instruction (line of text) will be performed each time the "Next Line" button is pressed. This can help you see exactly what your instructions are doing in a slow, controlled way and allow you to see where your logic is working or not. The "Restart" button will force the "Next Line" to be the first one again.

The "Reset MotorBee" button will return all settings on the MotorBee to a default condition where all outputs and motors are off and the servo is at mid position. This can be useful if you need to turn off quickly !.

## Just-Text Manual Mode

Text Control Manual Control Activate

### MotorBee Manual Control

Test your connected devices

Select Configuration Configuration 3

**Motors and Servo Control**

Motor Speed: 117, 144, 79, 61

Servo Position: 127

On/Off: , , ,

Set motors and Servo: Set

**Input and Output Control**

Digital Outputs: 1 , 2 , 3 , 4

Digital Inputs: 1 , 2 , 3 , 4 , 5 , 6

Read Inputs and Set Outputs: Update

Connecting Four Independent Motors

OK Cancel Apply

When you first have your devices connected to the MotorBee it can be useful to operate them individually to verify everything is connected and working correctly. Manual control mode provides this. Click on the "Manual Control" tab to access this facility.

The screen area is divided up into separate sections for inputs, outputs and motor control. The Output Control section is self explanatory. i.e. click on the "OP1 On" button to turn on output 1 and OP1 Off for off etc..

The motor control section has a range of controls to cover all possible configuration choices. You should select the hardware configuration you are using from the drop-down menu box. This will enable the appropriate controls for that configuration. It will also show an example photo of such a configuration. Operating a DC motor is very simple. Once you have selected direction and speed just click "Set".

The current state of the inputs can be displayed by clicking on the "Update Inputs" button. This will take a single snapshot of their current state

#### 4. Text Reference

The following is a list of all the "Keywords" that can be used in constructing your text file with a description of what they do and how to use them. Each description shows the general format of the instruction line containing the keyword and an example of its use.

##### OUTPUT

The OUTPUT keyword provides control of the seven switching type outputs available. The outputs can have one of two possible settings (i.e. ON or OFF).

Instruction Format:     **OUTPUT** <*output number*> <*onoff*>

**output number:**     This is the specific output to be turned on or off and must be in the range 1 to 4

**onoff:**             Self explanatory: use either of the words **ON** or **OFF**

Example:             **OUTPUT 2 ON**  
                       this turns on output 2

##### WAIT

Normal progress from one line of text to the next is virtually immediate (approx 100ms). The WAIT keyword allows a delay to be introduced while you wait for things to happen or complete. The delay period can be specified in milliseconds, seconds or minutes.

Instruction Format:     **WAIT** <*duration*> <*units*>

**duration:**         The time to wait

**units:**            Can be either **MS**, **SEC** or **MIN**, corresponding to milliseconds, seconds or minutes

Example:             **WAIT 100 SEC**  
                       this will cause a delay of 100 seconds

Note:                The smallest unit of time used is 100 milliseconds. If you specify something else , such as 125 MS, it will be rounded.

##### DCMOTOR

The DCMOTOR keyword provides control of a DC motor . Up to four DC motors can be connected at any one time.

Instruction Format:     **DCMOTOR** <*motor*> <*direction*> <*speed*>

**motor:**            This is the motor number it refers to: It can be either 1 ,2,3 or 4. Note that the actual number of motors available depends on the configuration currently being used. The default configuration has two DC motors present with numbers 1 and 2. See the CONFIGURATION keyword for other options.

**direction:**       The direction of rotation of the motor. Can be either **FORWARD** or **REVERSE**

**speed:**            The speed of rotation. This can be in the range 1 to 250

Example:             **DCMOTOR 2 FORWARD 250**  
                       DC motor 2 will rotate forward at maximum speed.



## DCMOTORHALT

The DCMOTORHALT keyword will stop a currently running DC motor . Up to four DC motors can be connected at any one time.

Instruction Format:     **DCMOTORHALT** <*motor*>

**motor:**                 The motor number it refers to: Can be either 1 ,2,3 or 4. Note that the actual number of motors available depends on the configuration currently being used. The default configuration has two DC motors present with numbers 1 and 2. See the CONFIGURATION keyword for other options.

Example:                 **DCMOTORHALT 2**  
DC motor 2 will be stopped.

## PLACE

The text file has a separate instruction in each line. Normal operation is to process each line and then to move on to the next one immediately after it. Sometimes it can be useful to be able to "Jump" to another part of the text file (i.e. over the intervening instructions). To be able to do this you need to be able to specify which line to go to. The PLACE keyword gives a particular line a numerical "Name" that can then be referred to as a destination to "jump to". This instruction does not carry out any function other than naming the line it is on. When this keyword is encountered in the text file , processing continues immediately on to the next line. (*see GOTO and IFINPUT keywords which both use place names*)

Instruction Format:     **PLACE** <*place number*>

**place number:**       The number which can be used to refer to this line in the text file. This can be any number in the range of 1 to 900. Note that each place number used must be unique. i.e. duplicate place numbers are not allowed.

Example:                 **PLACE 25**  
This line in the text file can now be referred to as 25 .

## **GOTO**

The text file has a separate instruction in each line. Normal operation is to process each line and then to move on to the next one immediately after it. Sometimes it can be useful to be able to "Jump" to another part of the text file (i.e. over the intervening instructions). The GOTO instruction will do this. It can be used to "jump" to any line which has a PLACE keyword. When this keyword is encountered in the text file , processing continues immediately after the line with the specified place number. (*see PLACE keyword* )

Instruction Format:     **GOTO** <*place number*>

*place number*:        The number of the PLACE statement to go to in the text file. This can be any number in the range 1 to 900 (as long as there is a line with the corresponding PLACE statement)

Example:               **GOTO 7**  
Go to the line in the text file which contains the PLACE 7 statement and continue from there.

## **IFINPUT**

The text file has a separate instruction in each line. Normal operation is to process each line and then to move on to the next one immediately after it. Sometimes it can be useful to be able to "Jump" to another part of the text file (i.e. over the intervening instructions) depending on the state of an input (eg a switch ). The IFINPUT instruction will do this. It can be used to "jump" to any line which has a PLACE keyword if the input it refers to is in the required state (i.e. ON or OFF). If the input matches the required state then processing will "jump" to the specified place number. If not then it will continue processing at the next instruction immediately after. (*see also PLACE keyword* )

Instruction Format:     **IFINPUT** <*input number*> <*onoff*> <**GOTO**> <*place number*>

*input number*:        The number of the input to check. MotorBee has 6 inputs

**onoff**:                Self explanatory: use either of the words **ON** or **OFF**

**GOTO**:                Always use the word "GOTO" in this position

*place number*:        The number of the PLACE statement to go to in the text file. This can be any number in the range 1 to 900 (as long as there is a line with the corresponding PLACE statement)

Example:               **IFINPUT 4 ON GOTO 32**  
If input number 4 is ON then go to the line in the text file which has the PLACE 32 statement and continue from there.

## LOOP

The text file has a separate instruction in each line. Normal operation is to process each line and then to move on to the next one immediately after it. Sometimes it can be useful to be able to repeat a section of text a number of times before proceeding. The LOOP instruction is used with a corresponding ENDLOOP instruction to define the lines of text to be repeated. (i.e. the text between the LOOP and ENDLOOP instructions).

Instruction Format:     **LOOP** <*number of loops(repeats)*> <*TIMES*>

*number of loops*:     The number of times the text between this instruction and the next ENDLOOP instruction will be repeated. This can be any number in the range 1 to 5000000. Once the specified number of loops has been completed normal processing will resume immediately after the ENDLOOP instruction.

*TIMES*:             Always use the word "TIMES" in this position

Example:             **LOOP 15 TIMES**

..

.. (*other instructions*)

..

**ENDLOOP**

..

This instructions in the text file between the LOOP and ENDLOOP will be repeated 15 times before continuing with the instruction immediately after the ENDLOOP.

## ENDLOOP

This defines the end of the area of text to be repeated. See description of LOOP instruction above.

Instruction Format:     **ENDLOOP**

## WAITINPUT

The WAITINPUT keyword will pause the execution of the text instructions until the specified input matches the requirement (ON or OFF).

Instruction Format:     **WAITINPUT** <*input number*> <*onoff*>

*input number*:     The number of the input to check. MotorBee has 6 inputs

*onoff*:             Self explanatory: use either of the words **ON** or **OFF**

Example:             **WAITINPUT 3 ON**

pause at current text instruction until input number 3 is ON

## **CONFIGURATION**

The MotorBee can be used to control up to four DC motors. The factory set default configuration is two DC motors. This is considered configuration 1. If you want to use one of the other possible combinations of motors (see below) then this instruction should be included in your text file. **IT MUST BE USED ON THE VERY FIRST LINE OF YOUR TEXT FILE.** If the default configuration is to be used then you don't need any configuration instruction.

Motor Numbering: To refer to each motor in your text file you need to know the number to use:

- When Configuration 1 is used (default) the two DC motors are DC Motors 1 and 2. Both are bi-directional.
- When Configuration 2 is used the motors are numbered 1,2 and 3. Only motor 1 is bi-directional
- When Configuration 3 is used the DC motors are simply numbered 1,2,3 and 4 (all are uni-directional)

Instruction Format:     **CONFIGURATION** <*configuration number*>

*configuration number*: The configuration to use. This can be one of three possible configurations.

1.     Two DC Motors (both bi-directional)
2.     Three DC Motors (Motor 1 is bi-directional, 2 and 3 are uni-directional)
3.     Four DC Motors (All motors are uni-directional)

Example:               **CONFIGURATION 2**  
Configuration 2 will now be used .

Note: The inputs, outputs and servo controls are always available regardless of configuration. The configuration only affects the DC motor arrangement.

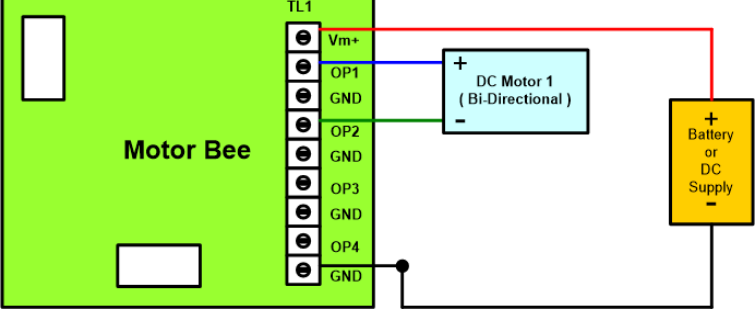
## **END**

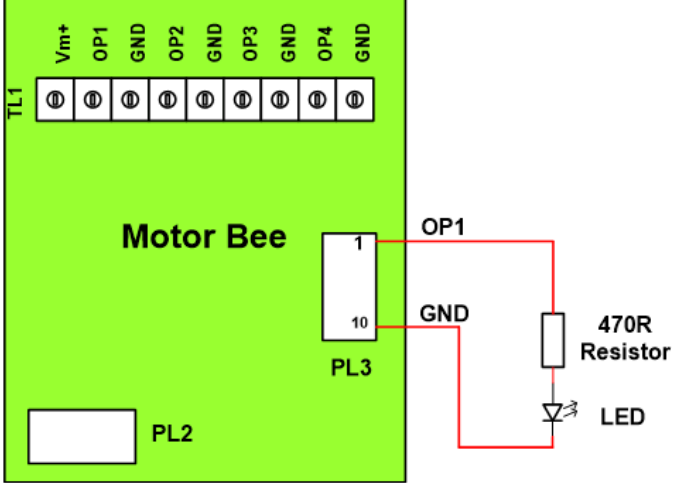
This defines the end of the text file and should be the last instruction in your file. No further processing of instructions will take place as soon as this instruction is executed.

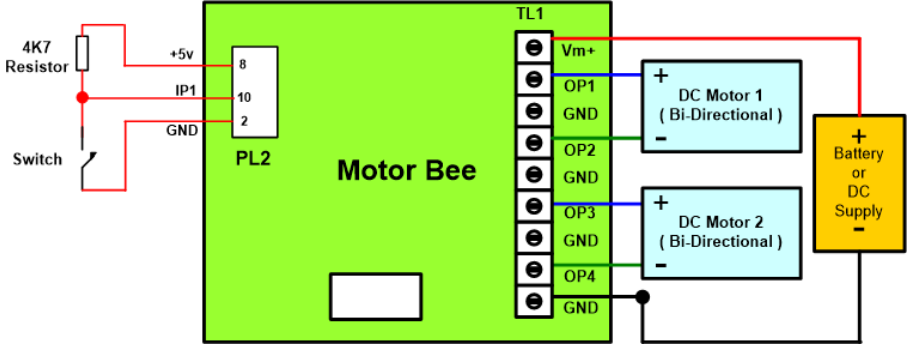
Instruction Format:     **END**

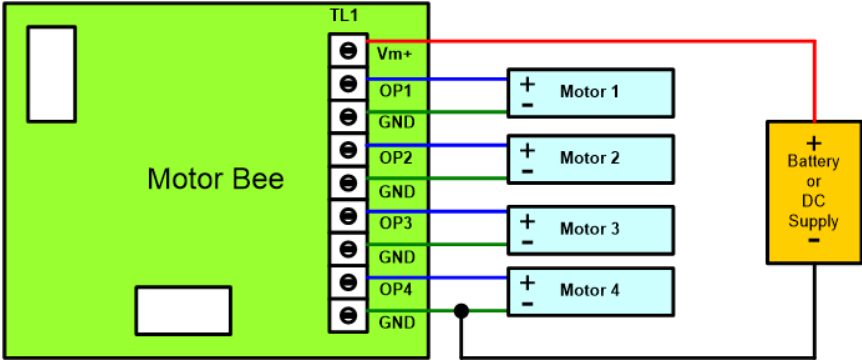
## 5. Text Examples

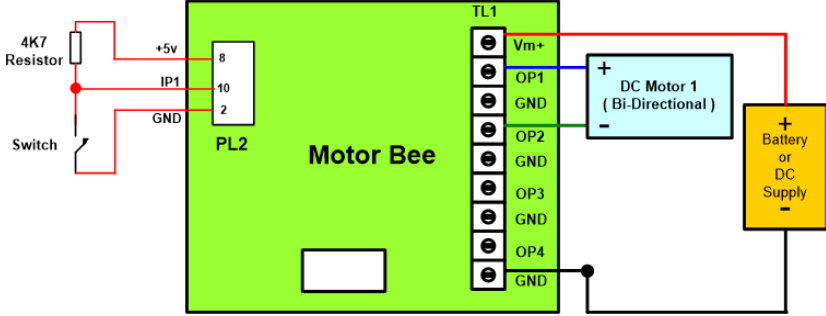
The following texts are examples of very simple control applications that demonstrate common hardware configurations and instruction usage.

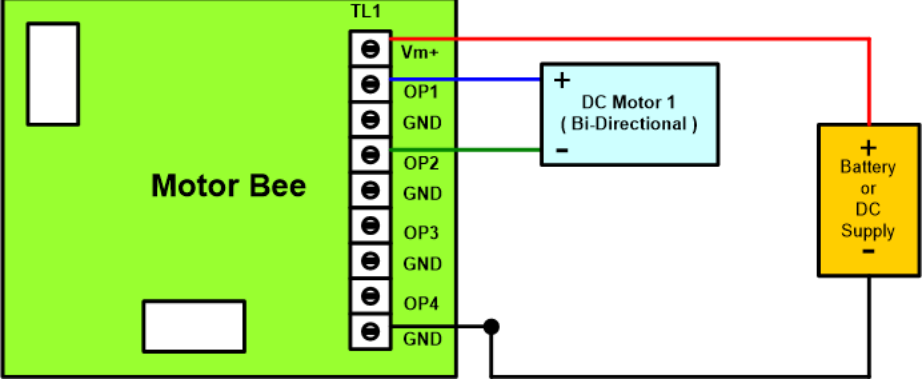
Text	Description
<pre>DCMOTOR 1 FORWARD 250 WAIT 5 SEC DCMOTORHALT 1 WAIT 5 SEC DCMOTOR 1 REVERSE 125 WAIT 5 SEC DCMOTORHALT 1 END</pre>	<p><b>DC Motor</b></p> <p>A DC motor which is connected to MotorBee TL1 terminals OP1 and OP2 will rotate in the forward direction at its maximum speed for 5 seconds. It will then stop for 5 seconds before rotating in the reverse direction at half speed for 5 seconds.</p> 

Text	Description
<pre>OUTPUT 1 ON WAIT 4 SEC OUTPUT 1 OFF WAIT 4 SEC OUTPUT 1 ON WAIT 4 SEC OUTPUT 1 OFF END</pre>	<p><b>Switching a digital output on and off</b></p> <p>Digital Output 1 (OP1) will turn on for 4 seconds then off for 4 seconds. It will do this twice.</p> <p>A simple way to test this is to use an LED (with a resistor) as shown below.</p> 

Text	Description
<pre> WAITINPUT 1 ON DCMOTOR 1 FORWARD 250 WAITINPUT 1 OFF DCMOTORHALT 1 WAITINPUT 1 ON DCMOTOR 2 FORWARD 250 WAITINPUT 1 OFF DCMOTORHALT 2 END </pre>	<p><b>Using an Input to Control the Operation of two DC Motors</b></p> <p>A digital input can be used to affect the flow of instructions. In this example digital input one is used to start and stop the two DC Motors in sequence. Initially off, when the switch is on DC motor 1 will rotate in the forward direction at its maximum speed until the switch is turned off. When the switch is turned on again DC motor 2 will rotate in the forward direction at its maximum speed until the switch is turned off again.</p> 

Text	Description
<pre> CONFIGURATION 3 DCMOTOR 1 FORWARD 250 WAIT 5 SEC DCMOTORHALT 1 WAIT 5 SEC DCMOTOR 2 FORWARD 250 WAIT 5 SEC DCMOTORHALT 2 WAIT 5 SEC DCMOTOR 3 FORWARD 250 WAIT 5 SEC DCMOTORHALT 3 WAIT 5 SEC DCMOTOR 4 FORWARD 250 WAIT 5 SEC DCMOTORHALT 4 END </pre>	<p><b>Controlling Four DC Motors</b></p> <p>Since the default configuration is for two DC motors with bi-directional control, it is necessary to include the CONFIGURATION 3 instruction in the first line of the text. This will allow the TL1 terminals to be used for 4 Uni-Directional DC motors..</p> <p>DC motor 1 will run at maximum speed in the forward direction for 5 seconds and then stop for 5 seconds. DC motors 2,3 and 4 will then do the same as motor 1.</p> 

Text	Description
<pre> IFINPUT 1 ON GOTO 10 DCMOTOR 1 FORWARD 250 GOTO 20 PLACE 10 DCMOTOR 1 REVERSE 250 PLACE 20 WAIT 5 SEC DCMOTORHALT 1 END </pre>	<p><b>Using an Input to Alter the Sequence of Control</b></p> <p>Slightly more complex control sequence in which the normal flow of instructions is altered by a GOTO instruction controlled by a switch on input 1</p> <p>If digital input 1 is on (i.e. the switch is open and the input is "pulled up" to +5v), DC motor 1 will run in the forward direction for 5 seconds at full speed. If the input is off (switch closed) it will run in the reverse direction for 5 seconds.</p> <p>Remember that the PLACE instructions don't actually do anything other than provide a "place" for the GOTO instruction to go to.</p> <p>Note that the "GOTO 20" instruction in line 3 is necessary to avoid the next instruction being the "DC MOTOR REVERSE 250"</p>  <p>The diagram shows a Motor Bee microcontroller board. On the left, a 4K7 resistor is connected between pin 8 (+5v) and pin 10 (IP1). A switch is connected between pin 10 (IP1) and pin 2 (GND). On the right, the TL1 header has pins Vm+, OP1, GND, OP2, GND, OP3, GND, OP4, and GND. The Vm+ pin is connected to the positive terminal of a DC Motor 1 (Bi-Directional). The OP1 pin is connected to the positive terminal of a Battery or DC Supply. The OP2 pin is connected to the negative terminal of the DC Motor 1. The OP3 pin is connected to the negative terminal of the Battery or DC Supply. The OP4 pin is connected to ground.</p>

Text	Description
<pre> LOOP 20 TIMES DCMOTOR 1 FORWARD 250 WAIT 2 SEC DCMOTOR 1 FORWARD 125 WAIT 2 SEC DCMOTOR 1 FORWARD 62 WAIT 2 SEC ENDLOOP DCMOTORHALT 1 END </pre>	<p><b>Using LOOP and ENDLOOP</b></p> <p>It is very often useful to repeat a sequence of instructions a number of times before progressing to the next section of text. This example will run DC Motor 1 forward at full speed for 2 seconds then half speed for 2 seconds and finally quarter speed for 2 seconds. It will then repeat this sequence 20 times before finally stopping motor 1.</p> 



## 6. Product Activation

This software is restricted to all functions except actually controlling the MotorBee. If you decide you want to use all facilities you can purchase an activation code at the link provided below. To make your purchase you need to obtain your personal activation request code by clicking on the button on the Activation page (the tab next to the Manual control tab). This will then show the code you need to use when making your purchase. Please copy the digits carefully into the comments section of your order. You will receive your activation code by return e-mail (usually same day but please allow up to 24 hours). This code should be entered into the "Activate Software" section. Clicking on "Activate" will then activate your JustText to the full version.

NOTE: This code is specifically for your computer AND your particular hard drive currently installed. It may not be used on any other computer or hard drive.

Text Control | Manual Control | Activate

# Just Text: MotorBee

Product Activation

Product is NOT Activated

Get Request Code

**1**

- Click on the button to generate an activation request code.
- Add this number to the comments area when you place your order in the Control Shop ([www.pc-control.co.uk/ocart](http://www.pc-control.co.uk/ocart))
- Your activation code will be sent to you by e-mail (usually same day)

NOTE: If you have already placed your order before obtaining this code then simply send the code by e-mail quoting your order number

Activate Software

**2**

When you receive your activation code enter it in the box and click Activate

OK Cancel Apply

You can make your purchase at the "Control Shop" website at.....

<http://www.pc-control.co.uk/ocart>

**Direct link to Just Text Software Purchase**

[https://www.pc-control.co.uk/ocart/index.php?route=product/product&path=72&product\\_id=155](https://www.pc-control.co.uk/ocart/index.php?route=product/product&path=72&product_id=155)

## **7. Minimum PC System Requirements**

Just-Text does not require a high spec PC for correct operation, but the following system is suggested as a sensible minimum

Processor	500MHz Pentium
Memory	640MB
HDD	100MB free space required
Screen Resolution	1024x768 (256 colours)
Interface	One free USB socket (1.0 or 2.0) to connect the MotorBee
Operating System	Windows XP (or later) (includes Win XP, 7,8 and 10, 32 & 64bit)

## 8. 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. Before proceeding with the installation and use of this software, carefully read the following terms and conditions of this license agreement and limited warranty ("the agreement"). By installing or using this software you indicate your acceptance of this agreement. If you do not accept or agree with these terms, you may not install or use this software. This software, including documentation is owned by PC Control Ltd. This agreement does not provide you with title or ownership of the software, but only a right of limited use as outlined in this license agreement. When purchased, PC Control Ltd. will grant you a non-exclusive, royalty free license to use the software on the single computer on which it is installed. The license is also restricted to the specific hard drive on that computer that is present during installation. The software is supplied solely for use in conjunction with our range of boards. Without the written consent of PC Control Ltd., you may not rent or lease the software, or sell any portion of the software. This software is provided "as is" without warranty of any kind either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. PC Control Ltd. do not warrant that the functions contained in the software will meet your requirements or that the operation of the software will be uninterrupted or error free. In no event shall PC Control Ltd or any other party who may have distributed the software be liable for damages, including any general, special incidental, or consequential damages arising from the use or inability to use the software, including, but not limited to, loss of data or losses sustained by you or third parties.
8. 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](http://www.pc-control.co.uk/general-terms.htm) which also apply.