

Using Strawberry Tree WorkbenchMac and Workbench PC software for data acquisition and control in the animal learning laboratory

IAIN S. MCGREGOR

University of Sydney, Sydney, Australia

Workbench (developed by Strawberry Tree Incorporated) is a commercially available software package for Macintosh or PC which is designed for industrial automation, process control, and data acquisition. Workbench provides a graphical environment in which programs are constructed by joining icons with connecting wires on the computer screen. In recent years, the author and his colleagues have used Workbench extensively to develop programs for the control of experiments in animal learning, psychopharmacology, and behavioral neuroscience. The principal advantage of Workbench is that sophisticated programs can be quickly written and modified by persons with little knowledge of classical computer programming techniques. Programs developed with Workbench provide on-screen charts, alarms, and meters for extensive user feedback during the running of experiments. Workbench operates in conjunction with Strawberry Tree interface cards and terminal panels for Macintosh or PC, allowing an integrated hardware/software package to be bought off the shelf. This article describes the hardware and software elements of Workbench and gives a step by step description of an example WorkbenchMac program for the operant conditioning of laboratory rats.

The widespread availability of inexpensive personal computers has had a major impact on research in experimental psychology, not least in the area of animal learning. The animal learning laboratory was once a place dominated by electromagnetic relays, hardware counters, and chart recorders, the instruments through which operant schedules were controlled and data collected and analyzed. In more recent years, this rather laborious approach has been largely superseded by a variety of computer-based solutions.

In the area of operant conditioning, computer programs for experimental control have been written in many different computer languages, such as BASIC, C, and FORTH (Balsam, Deich, O'Connor, & Scopatz, 1985). In addition, specialized versions of computer languages such as BASIC have been developed specifically for experimental control purposes, such as ECBASIC (Palya & Walter, 1993) and SPIDER (Fray, 1993). The disadvantage of this approach is that the typical psychology student or researcher is not a computer programmer, so writing programs in these languages or modifying existing programs may depend on the availability of expert technical support.

An alternative approach is the use of state-notation-based control languages such as SKED (Butler, 1980; Snapper, 1973) or MED PC (Tatham & Zurn, 1989). These allow operant control programs to be written in specialized languages based on state notation (Snapper, 1973; Snapper, Kadden, & Inglis, 1982). Although this approach is generally easier for psychologists to master compared to classical computer programming languages, it may still entail a considerable learning curve for students with a liberal arts background. In addition, to the author's knowledge, there is currently no state-notation-based system available for the Macintosh computer.

Over the past few years, the author has made extensive use of Workbench, a commercially available programming environment developed by Strawberry Tree, Incorporated (Sunnyvale, CA), which is available for both Macintosh (WorkbenchMac) and PC (Workbench PC) platforms. Originally designed for the automation and control of industrial processes, Workbench has several properties that make it an excellent tool for programming and data acquisition in the animal learning laboratory. Workbench is now used extensively in the author's department for teaching and research in animal learning, behavioral neuroscience, and psychopharmacology. The author's department has an undergraduate laboratory used for hands-on instruction in animal learning that is equipped with 12 PC-compatible computers (33 MHz 386), each using Workbench PC to provide data acquisition and control for a single operant chamber. In addition, the department has three research laboratories where individual Macintosh computers running WorkbenchMac control many

The software development described in this paper was supported by URG and Small ARC grants to I.S.M. The author is grateful to P. Horne for his advice on Workbench programming and to J. Martin (of Strawberry Tree, Inc.) for other technical help. Correspondence should be addressed to I. S. McGregor, Department of Psychology, University of Sydney, NSW 2006, Australia (e-mail: iain@psychvax.psych.su.oz.au).

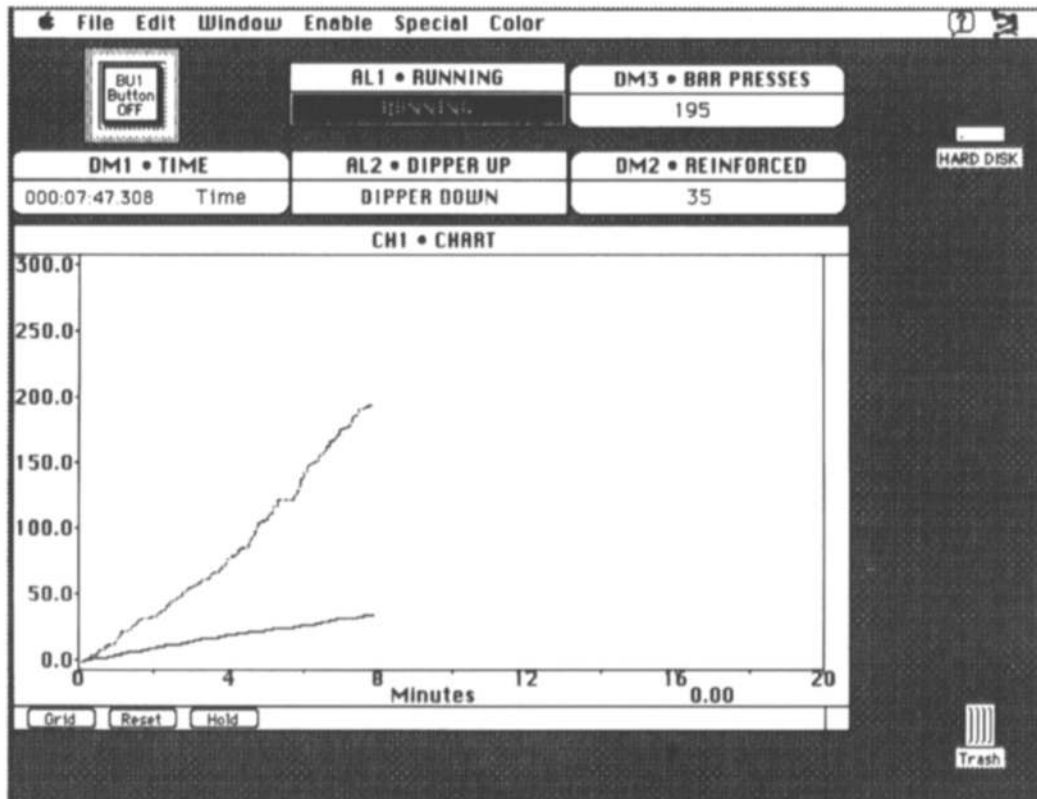
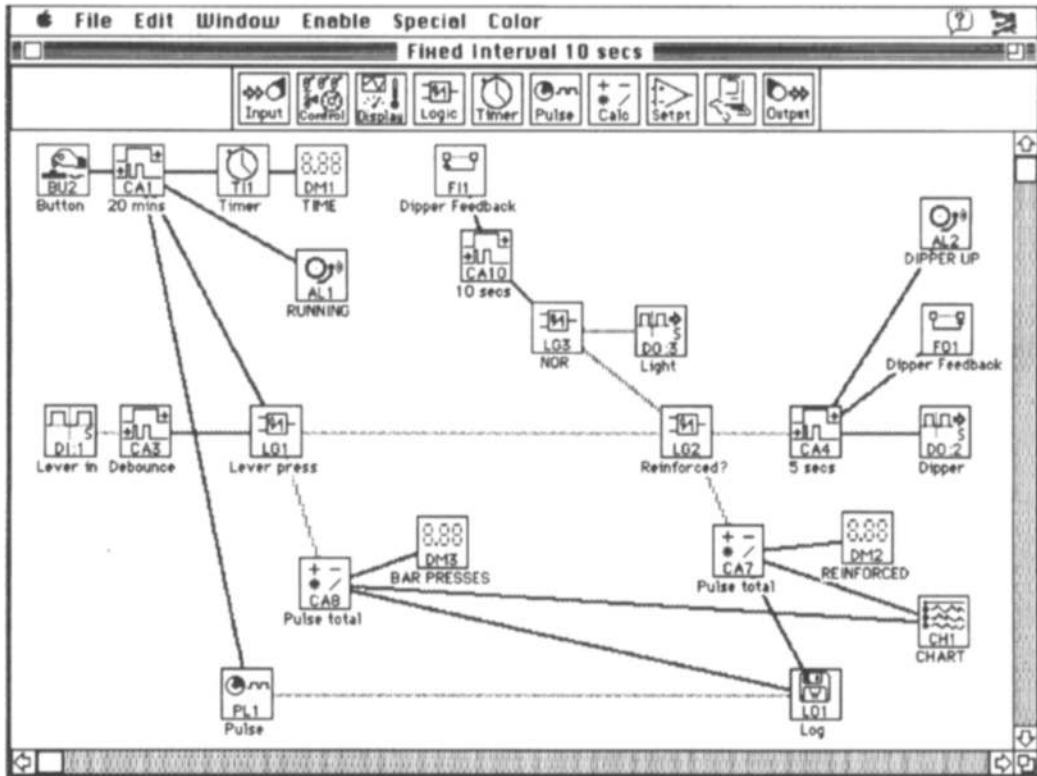


Figure 1. The top picture shows a WorkbenchMac program consisting of various icons connected with wires. Icons are dragged (using a mouse) from the icon palette underneath the title bar onto the main area of the worksheet, where they are connected together with wires. Wires carrying analog signals are black, and wires carrying digital signals are gray. The bottom picture shows what the user sees when this worksheet is “hidden”—namely, the alarms, meters, charts, and buttons that the worksheet generates.

experimental paradigms, including incentive learning, self-stimulation, delayed conditional discrimination, conditioned place preference, aversive contextual conditioning, and latent inhibition.

The primary advantage of Workbench is that complex programming tasks can be accomplished in an easy-to-use graphical environment or "worksheet" in which various function icons are connected using a mouse. A completed program is a graphical entity consisting of several icons and connecting wires (Figure 1, top). Thus, the usual programming requirement of learning a rather obscure new language with a demanding syntax is obviated with this approach. As a result, we find that students are not intimidated by programming tasks using Workbench, and researchers find that new experimental paradigms can be easily programmed and existing programs quickly modified.

The worksheets developed using Workbench provide a pleasant user interface that includes visual feedback on the ongoing progress of an experiment in the form of charts, alarms, and meters (Figure 1, bottom). In addition, data can be readily transferred into spreadsheet programs such as Microsoft Excel for analysis. The ease of use and general user friendliness of Workbench does not come at the sacrifice of power, since Workbench runs complex experimental paradigms comfortably and at acceptable speed.

The present paper gives a guide to establishing a Workbench-based animal learning laboratory. The hard-

ware and software implementation is described first, and then the reader is guided through a sample Workbench-Mac worksheet.

Hardware Requirements

Workbench software runs in conjunction with three separate hardware elements, namely (1) a Macintosh or PC computer; (2) a Strawberry Tree interface card, which resides inside the computer; and (3) a Strawberry Tree terminal panel, which connects the computer and interface card to the equipment being controlled. Figure 2 shows a simple Workbench-based operant conditioning installation where the computer measures a switch closure (leverpress) and controls a discriminative stimulus (light) and reinforcement delivery (in this case a dipper is raised that delivers a small amount of liquid into the operant chamber).

The laboratories used by the author make extensive use of MED Associates (St. Albans, VT) operant equipment because it is inexpensive, easy to set up, and reliable. MED Associates equipment requires 28 V (output) power, and the author uses 5-V circuitry to detect the switch closures occurring with leverpressing. Figure 2 shows how connections are made between the operant equipment, the Strawberry Tree terminal panel, the power supplies, and the computer. The circuitry used is "open collector," where equipment using the same power supply shares a common earth (see wiring of dipper and light in Figure 2).

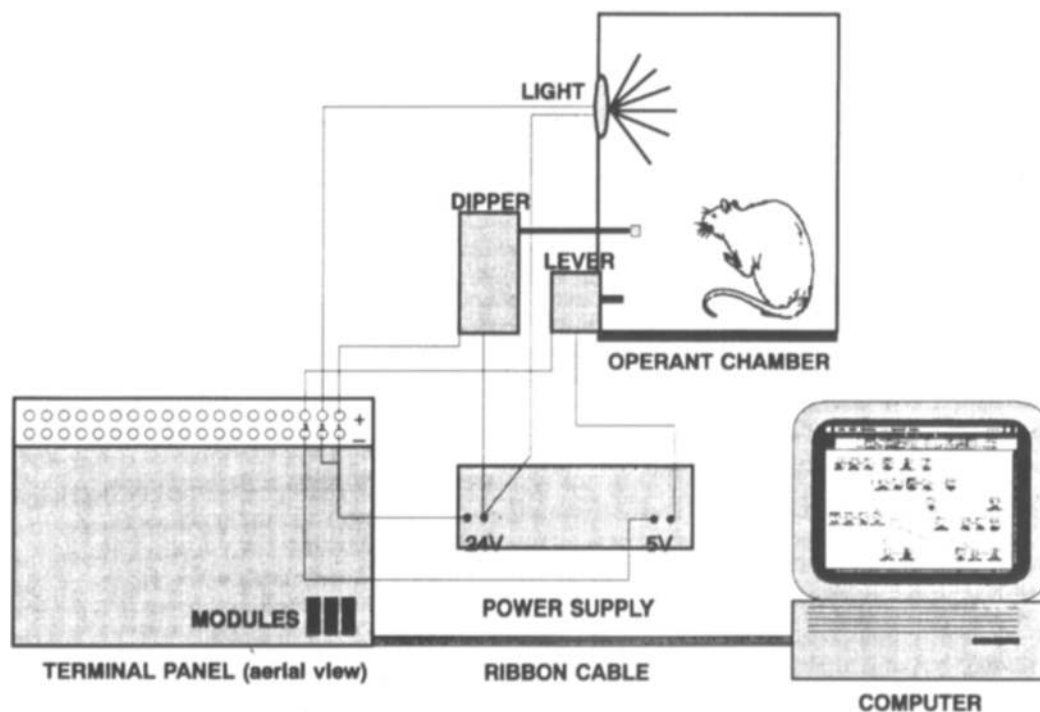


Figure 2. A single-chamber laboratory set up for operant conditioning research showing a typical hardware installation for a Workbench-based system. The Strawberry Tree interface card (which sits inside the computer) is not shown in this diagram. See text for description of other elements.

Computer requirements. WorkbenchMac software and associated interface cards will run on any Macintosh II series, Centris, Quadra, or Power Mac computer that is equipped with at least 3 MB of RAM, a full size NuBus slot, and System 7.1 or higher. Note that with Power Macs, the software currently runs in emulation mode rather than in native mode. In our laboratories, we have WorkbenchMac running on Macintosh IICx, IICi, IISI, Centris 650, and Quadra 800 machines.

Workbench PC runs on any PC compatible from XT upward. However, the graphical overhead of Workbench PC means that it runs more comfortably on faster machines, and the author recommends a 33 MHz 386 PC compatible as a minimum configuration for most laboratory installations.

Interface cards. Strawberry Tree manufactures interface cards in a variety of configurations for both Macintosh and PC. The Strawberry Tree cards for PC fit into the standard PC expansion slot, and interface cards for Mac fit into a standard full size NuBus slot. The interface card required depends on the nature of the data to be acquired (analog versus digital) and the number of input and output channels needed to run the laboratory equipment. In the author's laboratory, a Macintosh IICi running WorkbenchMac controls eight operant chambers (each equipped with two retractable levers, a nosepoke detector, a dipper, a noise generator, and a sonalert) via a Strawberry Tree interface card with 160 digital input/output (I/O) channels. For smaller laboratory installations, cards with 40, 16, or 8 digital I/Os are available. Each digital I/O channel can be configured in software to be either an input (e.g., to detect a leverpress) or an output (to switch on a light or raise a dipper).

Strawberry Tree cards with analog inputs and outputs are also available for laboratory situations in which analog data are being acquired or analog control signals are outputted. Typically, analog inputs or outputs are not required in operant conditioning research.

Terminal panels. The link between the Strawberry Tree interface card inside the computer and the operant equipment is made through a Strawberry Tree terminal panel (Figure 2). These come in a variety of forms (8, 40, and 80 channel) to match the different Strawberry Tree interface cards available. Some interface cards can drive more than one terminal panel. In the author's laboratory, a single 160-digital I/O interface card addresses an 80-channel and a 40-channel terminal panel. Terminal panels attach to the interface card through a ribbon cable, and all of the channels on the terminal panels are readily accessed via screw terminals on the rear of the unit. Operant equipment is wired to the terminal panel as shown in Figure 2.

The larger terminal panels (40 and 80 channels) must be populated with optically isolated input or output modules for each channel being used. Low-cost modules (LDA-210s) can be used for output channel switching or sensing small input signals where the voltages and currents involved are small (< 30 V and < 10 mA). For the

switching of higher powered devices (such as dippers or retractable levers), high-powered output modules (ODC-5s) are recommended. Both high-power and low-cost modules slot manually, without soldering, into numbered slots on the terminal panel.

Typical costs. A typical Strawberry Tree hardware and Workbench software combination for the control of eight operant chambers costs around \$2,500 (current U.S. prices: Workbench PC or WorkbenchMac software \$995; 40-channel Strawberry Tree interface card for Macintosh, \$380; 40-channel Strawberry Tree interface card for PC, \$295; 40-channel Strawberry Tree terminal panel with 40 low-cost modules, \$795; high-powered output modules, \$20 each). Such a system would provide five channels per operant chamber, allowing the use of, say, one dipper, two levers, one light, and one tone in each box. ODC modules would be required for the high-powered outputs (the dipper and light), whereas low-cost modules could be used for the lever inputs and for the output to the tone generator.

Software Description

WorkbenchMac versus Workbench PC. The present article focuses on WorkbenchMac software, the latest version of which is WorkbenchMac 4.0 (released in December 1994). Workbench PC (the latest version of which is 2.2) is similar to WorkbenchMac in operation, except that it has fewer icons (the X-Y, Bar, and Scrolling Y-T charts, analog meter, and lamp icons are found only in WorkbenchMac) and only runs in gray scale, compared to WorkbenchMac's color operation. In addition, only WorkbenchMac allows real-time data transfer between Workbench and other applications using Apple Events.

A new program called Workbench for Windows has recently become available for the PC. Its operation is somewhat different from WorkbenchMac and Workbench PC and has not yet been evaluated by the author. Interested readers should contact Strawberry Tree for more details.

The worksheet and icon palette. Programming in Workbench takes place on-screen in a resizable window called a worksheet (Figure 1, top). Icons are dragged (using the mouse) from the icon palette under the menu bars onto the worksheet, where they can be connected together by wires. The icon palette has several subpalettes of icons that appear when particular icons in the top palette are clicked with the mouse (Figure 3). For example, clicking on the Display icon brings in a subpalette of eight icons, including various chart and meter icons and the lamp and alarm icons. Any of these may then be dragged onto the worksheet (Figure 3).

To connect two icons on a worksheet, the mouse is dragged from the left icon to the right. In WorkbenchMac, there are 27 different types of icon available from the icon palettes. In addition, three further icons, the Random, Cycle Time, and One Shot icons, can be accessed via the dialog box for the Calculation icon. Fig-

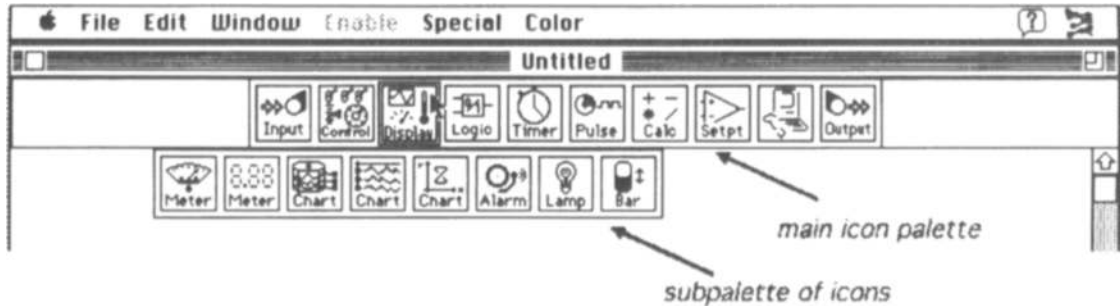


Figure 3. Subpalettes of icons can be accessed under the Input, Control, Display, Macintosh, and Output icons of the main icon palette. Shown here is the subpalette for the Display icon, which is accessed when the Display icon is clicked.

ure 4 shows these 30 icons and briefly describes their functions.

Most icons receive either an analog (number) or digital (1 or 0) input, and output either (1) an analog value (number); (2) a digital value (1 or 0); (3) a screen display (in the case of the meter, alarm, and chart icons); or (4) a data file that is written to disk (in the case of the Log icon). In the worksheet, wires carrying analog signals show up as black, and wires carrying digital signals are gray (Figure 1, top).

Icon dialog boxes. Most of the icons in Workbench have a dialog box associated with them that is accessed by double clicking on the icon (see Figure 5 for example dialog boxes). The dialog box allows the user to define the precise mode of operation of the icon. For example, in the case of the digital input icon, the dialog box allows the selection of the input channel on the interface card that the icon will receive its input from and the rate at which this input will be sampled (Figure 5, top). In the case of the Calculation icon, the dialog box allows the user to select which one of more than 45 different types of calculation the icon will perform on its input (Figure 5, second from bottom). Similarly, the Logic icon dialog box allows one of a number of different logic functions to be selected (Figure 5, second from top). Obviously, the present article cannot give an exhaustive account of the dialog box for each icon, and the reader should refer to the WorkbenchMac manual for further information.

The same icon type can be used multiple times on the same worksheet operating in a different mode each time. Thus, a worksheet might contain three Calculation icons, one performing a $\text{Sin}(X)$ function, another performing an $(X+Y)$ function, and the third computing Pulse Rate (X). Each individual icon can be given its own name to enhance the clarity of the worksheet. Icons are named via their dialog boxes (Figure 5). It should be noted that any individual Workbench icon can have outputs to many other icons in the worksheet (Figure 1, upper).

Program flow and cycle time. The flow of control on a worksheet is from left to right, so that icons on the left of the worksheet pass their data to the icons they are connected to on the right. As Workbench runs, it continuously cycles from the left to the right of the worksheet.

The time taken for the computer to process all the icons from the left to the right of the worksheet is called the "cycle time," and cycle time determination is a function available with the Calculation icon. The cycle time varies according to the number and type of icons on a worksheet and the speed of the computer being used. In our experience, complex worksheets involving control of multiple operant chambers usually have a cycle time of no more than 50 msec when running with a middle-range Macintosh such as a IIci. Although this figure is typical of the processing time required by other operant control programs (e.g., Tatham & Zurn, 1989), it may be a limitation in some time-critical experimental situations where single millisecond temporal resolution is required. Cycle time can be greatly reduced through use of faster (i.e., 68040 or PowerPC-based) Macintoshes.

Logging data in the worksheet does not usually affect the cycle time in WorkbenchMac, since disk operations are buffered with a large memory cache (at least 32K) in System 7. If the disk cache becomes full, under conditions of high-volume data acquisition, processing of the worksheet is temporarily halted while the data in the cache are written to disk. If this presents a problem to the researcher, then the size of the cache can be increased accordingly using the Memory Control Panel of System 7.

Although data flow is usually from left to right on a worksheet, WorkbenchMac allows flow from the right to the left via the Feedback In and Feedback Out icons. Feedback Out icons on the right side of the worksheet can pass their inputs to appropriately configured feedback in icons on the left side. This is an important process for many applications in the animal learning laboratory, where response outcomes must influence ongoing conditions (see example below).

Hiding the worksheet. Once a worksheet has been completed, it can be hidden so that the worksheet and icons are no longer seen and only the meters, alarms, buttons, and charts that the worksheet generates are displayed. This is convenient when the user desires an uncluttered view of the ongoing progress of an experiment being run, or in teaching situations where the instructor wishes to protect the underlying worksheet from student modification.









































TYPE		
	 Digital Input Worksheet input from an interface card digital channel e.g. DI:1 is input on channel 1.Used to sense switch closures from levers etc.	 Analog Input Worksheet input from an interface card analog channel e.g. AI:1 is analog input on channel 1 Sense analog values from thermocouples etc.
	 Feedback In Receives input from a Feedback Out icon, allowing values to be passed from right to left of worksheet (opposite to normal direction).	 Replay Represents file on disk from which values are passed to the worksheet at a user defined rate.
	 Switch Generates an on-screen switch which allows the user to switch between two inputs using mouse.	 Slider Generates analog output set by user via mouse on a variable range on-screen slider
	 Button Generates output when user clicks mouse on an on-screen button.	
	 Meter Generates an on-screen analog meter (i.e. like VU meter) of input.	 Strip Chart Generates an on-screen strip chart of input with user-defined time scale on the X axis.
	 Digital Meter Generates an on-screen digital (i.e. numerical) meter of input.	 Alarm Generates an on-screen display of user-defined message.
	 Chart Generates an on-screen chart of input with user-defined time scale on the X axis.	 Lamp Generates and on-screen lamp that lights whenever icon input is >=1.
	 XY-Chart Generates an on-screen XY chart of two inputs.	 Bar Generates an on-screen bar graph of input.
	 Logic Applies the logic functions AND, OR, XOR, NAND, NOR, XNOR or "Divide by n" to its inputs and generates appropriate output.	
	 Timer Generates elapsed time in secs. Started, stopped or reset by digital input.	
	 Pulse Generates a digital pulse at user-defined rate.	
	 Calculation Performs a user-defined calculation on its input and generates output equivalent to the result of the calculation.	 Random Generates a random number between two user-defined values. Accessed via dialog box for calculation icon.
	 One Shot Generates a "1" output for a user defined duration. Accessed via dialog box for calculation icon.	 CycleTime Reports the time in msec taken for the program to process all the icons in the worksheet from left to right. Accessed via dialog box for calculation icon.
	 Set Point Compares input to a user-defined set point and generates output when specified condition met.	
	 Serial Allows input from or output to a serial device via Mac serial port.	 Events Runs specified Apple Events when triggered.
	 Menu Selects specified menu items in Workbench or Finder when triggered.	
	 Digital Output Worksheet output to an interface card digital output channel. Used to switch real world devices on and off such as tones and lights.	 Analog Output Worksheet output to an interface card analog output channel. Used to provide continuous control signals or to generate analog waveforms etc.
	 Feedback Out Passes input value from right to left of worksheet (opposite to normal direction) to Feedback In icons with the same ID number	 Log Represents a file on disk to which data are written.

Figure 4. The various icons available in WorkbenchMac and a brief description of their function.

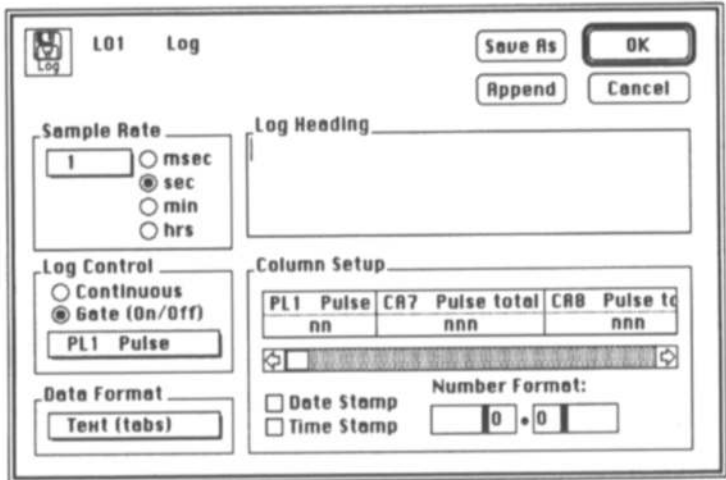
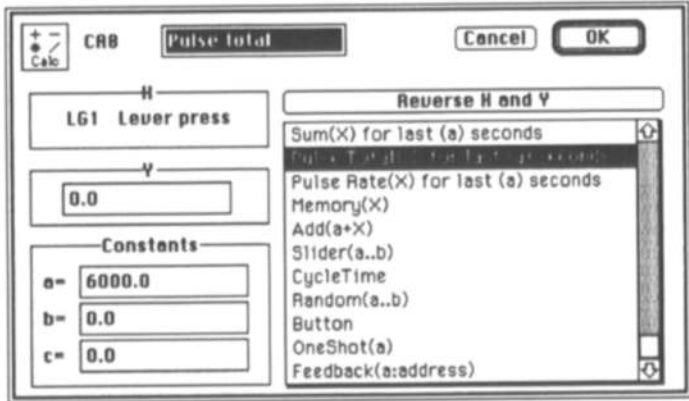
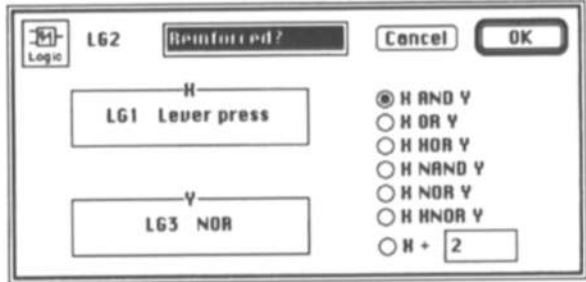
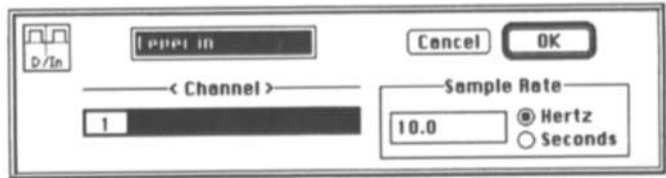


Figure 5. Sample dialog boxes for the Digital In, Logic, Calculation, and Log icons (going from top to bottom of figure). Note that the Calculation icon dialog box is scrollable, and so only a subset of the possible calculation icon settings are displayed in this figure. The dialog boxes shown here all correspond to specific icons in the worksheet shown in Figure 1 and Figure 6. Icons can be named via the highlighted box at the top of the dialog box.

A Simple Example Worksheet

Figure 1 (top) shows a simple example worksheet involving experimental control and data acquisition for an experiment running on the equipment shown in Figure 2, namely, an operant chamber equipped with a lever, dipper, and light. In this experiment, reinforcement is delivered under a fixed-interval 10-sec (FI-10) schedule, meaning that a leverpress will be reinforced only if 10 sec have elapsed since the last reinforcement was delivered. The discriminative stimulus (light) comes on whenever reinforcement is available. Reinforcement consists of the dipper being raised for 5 sec to deliver a small quantity of sucrose solution into the chamber. The experiment duration is 20 min.

The worksheet in Figure 1 (top) controls all of the experimental parameters (e.g., session duration, reinforcement schedule, dipper time) as well as logging data and providing screen displays of the ongoing responding of the rat. Figure 1 (bottom) shows what the user sees when the worksheet is hidden during this experiment, namely, the buttons, meters, alarm, and chart that are generated by the worksheet. Note that with Workbench-Mac, there is a full color user interface that is not reproduced here.

For the purposes of explaining how the worksheet functions, it is reproduced in Figure 6, divided into six sections, each of which is now described in detail.

Section 1. This section of the worksheet controls the session length (20 min) and provides user feedback on the status of the experiment (running or stopped) and the elapsed time. The button icon (labeled “Button”) generates an on-screen button (labeled “BU1 Button”) that the user can press to start the test session (Figure 1, bottom). When this button is pressed, the output of the button icon momentarily changes from “0” to “1.” This causes the one shot icon (labeled “20 mins”) to generate a “1” output for a 20-min period, and it is this icon that thus controls the session duration. The output from the “20 mins” icon starts a timer icon (labeled “Timer”) that is connected to a digital meter icon (labeled “TIME”) and that gives a screen display of the time elapsed since the start button was pressed. In addition, the “20 mins” icon controls an alarm icon (labeled “RUNNING”) that displays a flashing color message, “RUNNING,” on the screen for the duration of the 20-min session. When the input to the Alarm icon is digital “0” (i.e., before and after the 20-min test session), the alarm displays the message “STOPPED.”

Section 2. This section of the worksheet handles the input from the lever in the operant chamber. The switch closure resulting from a leverpress in the chamber is sensed by digital input 1 on the interface card. Whenever the lever is pressed, the digital input icon (labeled “Lever in”) outputs a transient “1” signal (Figure 5, top, for the dialog box for this icon). This output goes to a one shot

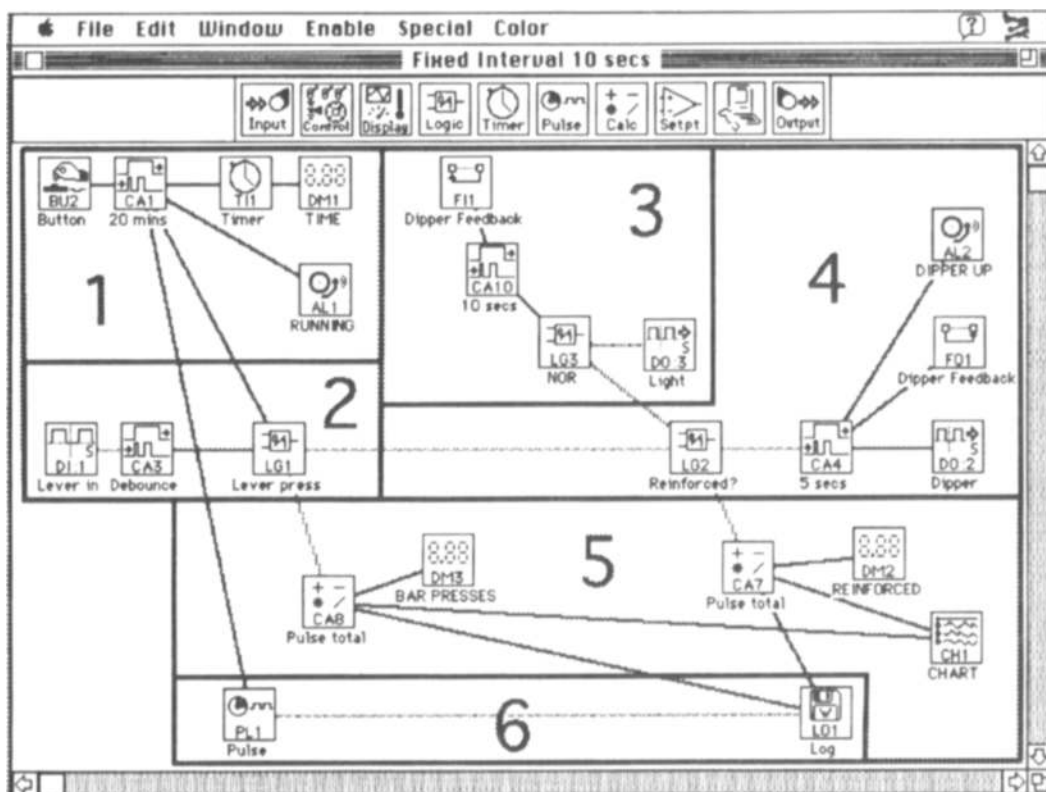


Figure 6. The same worksheet as is shown in Figure 1 (top) but subdivided into six sections for the purposes of exposition.

icon (labeled “Debounce”), which gives a fixed-duration 0.1-sec output whenever its input is “1.” This icon acts as a debounce circuit (Genovese, 1988), meaning that each press the rat makes delivers a single 0.1-sec output signal into the worksheet regardless of the actual number of switch closures detected during this 0.1-sec period. The “Debounce” icon feeds its output to a logic icon (labeled “Lever press”) that performs a logical AND operation between the lever input and the output of the “20 mins” icon discussed in Section 1. This AND function ensures that a leverpress signal is only passed through the worksheet (to charts, meters, and dipper control icons) if the experiment has been started via the on-screen button in Section 1. Thus leverpresses outside the session time will not be reinforced or counted.

Section 3. This section controls the fixed-interval schedule of reinforcement. A feedback in icon (labeled “Dipper Feedback”) receives its input from an identically configured feedback out icon (also labeled “Dipper Feedback”) connected to the dipper circuitry of Section 4. Recall that the feedback icons allow values to flow from the right to the left of a worksheet, opposite to the normal direction. Feedback icons are not joined with a wire; rather, the values flowing into the Feedback In icon (in this case, the “Dipper feedback” icon in Section 4) are reproduced in identical feedback icons placed elsewhere on the worksheet (in this case, the other “Dipper feedback” icon in Section 3).

When the dipper is down, the “Dipper feedback” has a 0 input, and when the dipper is up (i.e., during reinforcement), its input is “1” (see Section 4 for derivation of these values). Whenever the input to the “Dipper feedback” icon changes to “1” (i.e., when the dipper goes up), the one shot icon in Section 3 (labeled “10 secs”) outputs a “1” value for 10 sec. This is fed to a logic icon, which performs a NOR function on its input. This NOR function inverts the value of its digital input, outputting a “1” when the input is “0” and a “0” when the input is “1.” The output of the NOR icon is therefore 1 at all times except during the 10 sec following the dipper being raised, which corresponds to the time of the fixed interval. The output of the NOR icon controls digital output number 3 (labeled “Light”), so that the discriminative stimulus (light) comes on only when the fixed interval has elapsed.

Section 4. This section controls the dipper. A logic icon (labeled “Reinforced?”) performs an AND function on its two inputs (Figure 5, second from top, for the dialog box for this icon). It will output a “1” only if there is a leverpress input (as determined by the output of Section 2) AND the fixed interval has elapsed (as determined by the output of Section 3). If these conditions are met, the One Shot icon (labeled “5 secs”) is triggered and causes the digital output to the dipper on Channel 2 of the interface card to go high for 5 sec. This causes the dipper to be raised for a 5-sec period. In addition, the “5 secs” icon changes the output of the “Dipper feedback” icon to 1, thereby resetting the fixed-interval clock described in Section 3. Finally, an alarm icon (labeled

“DIPPER UP”) connected to the “5 secs” icon provides screen messages as to the current status of the dipper, namely, “DIPPER UP” when its input is 1 and “DIPPER DOWN” when its input is 0 (Figure 1, bottom).

Section 5. This section counts the number of responses and reinforcements occurring in a test session and displays these on the computer screen. Two calculation icons (both labeled “Pulse total”) are set to the “Pulse Total” calculation, which measures the number of pulses received over a user-defined period (in this case, 6,000 sec or 100 min) (Figure 5, second from bottom, for the dialog box setting for this icon). The integration time for pulse counting is set to be much greater than the duration of the experimental session to ensure that the meter values do not decrement at the end of the test session. One of the “Pulse total” icons receives its input from the “Lever press” icon of Section 2 and thus records the cumulative number of barpresses made by the rat across the experimental session. The second “Pulse total” icon receives its input from the “Reinforced?” icon in Section 4 and thereby counts the cumulative number of reinforcements received during the session. Meter icons connected to the two “Pulse total” icons (labeled “BAR PRESSES” and “REINFORCED”) generate on-screen meters that display these cumulative totals (Figure 1, bottom). In addition, the two “Pulse total” icons feed their output to a Chart icon (labeled “CHART”) that generates an on-screen chart upon which the cumulative number of barpresses and reinforcements are displayed across a 20-min axis (Figure 1, bottom).

Section 6. Section 6 of the worksheet controls the logging of the data to disk. The log icon defines a file called “Log” to which data are written. The log icon receives its input from the two “Pulse total” icons in Section 5 and writes the outputs of these icons to disk at a user-defined rate. The dialog box associated with the log icon is shown in Figure 5 (bottom). In this example, the log icon is configured so that logging to disk is gated by a pulse icon (labeled “Pulse”). This pulse icon generates a 1-sec duration digital pulse for every 60 sec of the experimental session. The input to the Pulse icon from the “20 mins” icon of Section 1 ensures that it outputs only a pulse, and that data are logged only when the experiment is running. Whenever the log icon receives an input pulse from the pulse icon, it writes the current values of the two “Pulse total” icons to the disk cache. Thus, a cumulative record of the number of responses and reinforcements for each minute of testing is recorded. Figure 7 shows a view of the actual log file for a test session opened in Microsoft Excel.

The ease of use of Workbench may be gauged by the fact that the above program took the author (who is not a trained computer programmer) approximately 15 min to generate.

Extending the Worksheet

Multiple operant chambers. The worksheet described above can be very quickly extended to control more than one operant chamber. The entire collection of

	A	B	C	D
1	PL1 Pulse	CA7 Pulse total	CA8 Pulse total	
2	1	0	0	
3	1	4	11	
4	1	7	20	
5	1	11	37	
6	1	15	61	
7	1	19	76	
8	1	23	100	
9	1	26	115	
10	1	30	146	
11	1	34	170	
12	1	38	181	
13	1	43	189	
14	1	47	200	
15	1	52	212	
16	1	55	229	
17	1	60	241	
18	1	65	249	
19	1	69	255	
20	1	73	266	
21	1	76	274	
22	1	81	290	
23				

Figure 7. The data collected via the Log icon in the example worksheet. The file has been opened in Microsoft Excel. Each input to the Log icon has its own data column. The first column simply represents the output of the Pulse icon (in Section 6 of Figure 6), which gates logging and so always has the value 1. The second column is the cumulative number of reinforcements for each minute of testing, and the third column is the cumulative number of responses. See text for further details.

icons in Figure 1 (top) can be copied (using "Copy" from the edit menu) and then pasted onto another location in the worksheet. The worksheet is "scrollable" and covers up to 20 times the area of a single screen. The newly pasted set of icons can be adjusted so that the digital input and output channels match the hardware connections of the second chamber with the terminal panel. In addition, a different name can be given to the log file for this second chamber. This copy-and-paste approach ensures that once a program has been successfully written for one chamber, it is a trivial matter to extend the program to control multiple chambers. In the author's laboratory, eight boxes are routinely controlled from the one worksheet.

Altering schedule parameters. The above worksheet runs an FI-10 schedule for 20 min with a dipper time of 5 sec. Altering any of these parameters is as simple as double clicking on the appropriate icon and inserting a new value into the icon's dialog box. Thus the dipper time could be altered to a new value by double clicking on the "5 secs" icon in Section 4 and inserting a new value. Similarly, the session length could be readily altered by changing the "20 mins" icon in Section 1, and the fixed-interval parameter could be changed via alteration of the value of the "10 secs" icon in Section 3. After such changes are made, the worksheet can be saved under a different name using the "Save as" command. In this way a large library of different experiment control

worksheets can be easily established for a variety of different experimental requirements.

Different schedules of reinforcement. The above worksheet can also be modified to provide other schedules of reinforcement. Variable interval schedules can be established using the Random icon (Figure 4) in conjunction with a Calculation icon set to perform a Memory calculation. This allows a random number to be selected and stored for each new trial. Fixed-ratio schedules can be implemented using the Logic icon in its "divide by n " mode. In this mode, the Logic icon outputs a digital "1" for every " n " input pulses. Although presentation of worksheets for different experimental schedules exceeds the scope of the present article, example worksheets for different schedules of reinforcement can be obtained from the author on request.

Simulations. The replay log icon allows data to be read into a worksheet at a specified rate. Thus the leverpress performance of a rat in a test session can be logged by Workbench and this log can then be "played back" into a worksheet. This may be useful for teaching purposes, where simulations of operant performance are required, or in cases where the user wishes to test newly written worksheets without using actual subjects to generate test data.

Incorporating user-defined functions. One strength of Workbench is that users are not constrained to use only the icons that are provided with the program. Users

can write their own external functions and these can be readily incorporated into worksheets. User-defined external functions appear in the dialog box of the Calculation icon and can be selected in just the same way as other calculation functions. There are few limits as to what external functions can do: They can have their own dialog boxes, windows, menus, or icons, or can provide custom on-line analysis. In the author's experience, external functions are not often needed. With some practice and a little ingenuity, most needs can be addressed with the existing palette of icons in Workbench.

With Workbench PC, external functions can be written in Borland's Turbo C++, while for WorkbenchMac, functions can be written in THINK C. Other languages can be used but require additional code or "patches" that are not supported by Strawberry Tree. A patch for Workbench Mac to allow incorporation of external functions written in Think Pascal has recently been described in this journal (Bushe, Vaughan, & Rosenbaum, 1994).

Conclusion

The present article has described the use of Strawberry Tree's Workbench software for control of simple operant conditioning experiments. Workbench provides a user-friendly, powerful, and relatively inexpensive approach to data acquisition and experimental control in animal learning, psychopharmacology, and behavioral neuroscience experiments. It is a very different programming approach compared with traditional techniques and one that may be particularly well adapted to the researcher who is not well versed in classical computer programming. In over 5 years of use, the author and his colleagues have found Workbench to be an excellent tool for teaching and research.

However, the package may not suit everyone. Possible disadvantages of the Workbench approach is that its temporal resolution is not as good as some other packages (see, e.g., Palya & Walter, 1993) and that experienced structured programmers may find the icon-based Workbench approach to programming rather counter-intuitive and inflexible.

Readers interested in Workbench may also wish to consider the LabView software package manufactured by National Instruments, which provides a similar graphical programming environment to Workbench, albeit at greater expense. At least one behavioral application of the LabView package has recently been described (Kling-

Peterson & Svensson, 1993). Also worth noting for a user-friendly approach to schedule control in operant conditioning is the recently released MED Associates Schedule Manager software for Windows.

Although few experimental psychologists are currently making use of Workbench, it is hoped that a growing user base will emerge among which ideas, tips, external functions, and worksheets can be shared.

Availability. Strawberry Tree Workbench software and associated hardware is manufactured by Strawberry Tree Incorporated, 160 S. Wolfe Road, Sunnyvale, CA 94086 (fax: 408 736 1041). Worksheets for the control of animal learning experiments and general advice on using Workbench software in experimental psychology can be obtained from the author (e-mail: iain@psychvax.psych.su.oz.au).

REFERENCES

- BALSAM, P. D., DEICH, J., O'CONNOR, K., & SCOPATZ, R. (1985). Microcomputers and conditioning research. *Behavior Research Methods, Instruments, & Computers*, *17*, 537-545
- BUSHE, M. M., VAUGHAN, J., & ROSENBAUM, D. A. (1994). Pascal external functions for Strawberry Tree's "Analog Connection Workbench" on the Macintosh. *Behavior Research Methods, Instruments, & Computers*, *26*, 461-466.
- BUTLER, F. E. (1980). MicroSKED. *Behavior Research Methods & Instrumentation*, *12*, 152-154.
- FRAY, P. J. (1993). Personal computers and the control of behavioural experiments. In A. Sahgal (Ed.), *Behavioural neuroscience: A practical approach* (Vol. 1, pp. 185-210). Oxford: IRL Press.
- GENOVESE, R. F. (1988). A strategy for microcomputer-controlled measurement of responses in the behavioral laboratory. *Behavior Research Methods, Instruments, & Computers*, *20*, 1-5.
- KLING-PETERSON, T., & SVENSSON, K. (1993). A simple computer-based method for performing and analyzing intracranial self-stimulation experiments in rats. *Journal of Neuroscience Methods*, *47*, 215-225.
- PALYA, W. L., & WALTER, D. E. (1993). A powerful, inexpensive experiment controller or IBM PC interface and experiment control language. *Behavior Research Methods, Instruments, & Computers*, *25*, 127-136.
- SNAPPER, A. G. (1973). Use of a notation system for digital control and recording. *Behavior Research Methods, & Instrumentation*, *5*, 124-129.
- SNAPPER, A. G., KADDEN, R. M., & INGLIS, G. B. (1982). State notation of behavioral procedures. *Behavior Research Methods & Instrumentation*, *14*, 329-342.
- TATHAM, T. A., & ZURN, K. R. (1989). The MED-PC experimental apparatus programming system. *Behavior Research Methods, Instruments, & Computers*, *21*, 294-302.

(Manuscript received March 3, 1994;
revision accepted for publication December 13, 1994.)