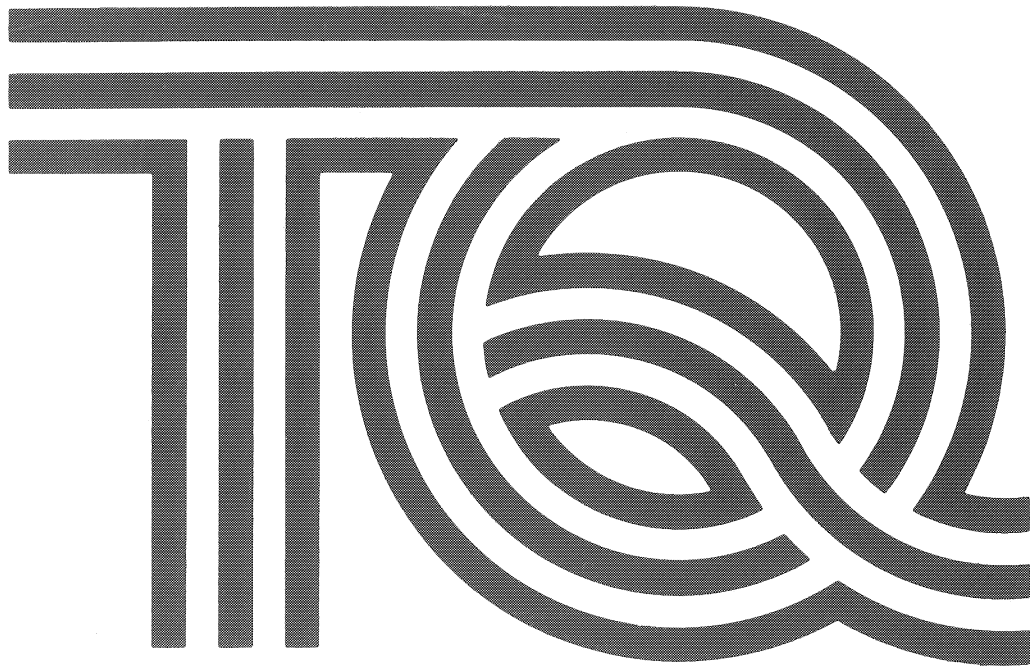


**TecQuipment**

OUTLINE DETAILS of INTERFACE COMMUNICATION PROTOCOL  
for  
MA2000 EDUCATIONAL ROBOT

The equipment described in this manual is  
distributed by the  
T E C Q U I P M E N T  
Group of Companies

Suppliers of Technological Laboratory  
Equipment designed for Teaching

BONSALL STREET, LONG EATON, NOTTINGHAM, NG10 2AN, ENGLAND  
TELEPHONE: LONG EATON 722611    TELEX: 377828

Copyright (c) 1985 TecQuipment International Ltd

All rights reserved. This document is supplied for the sole use of selected users of the MA2000 robot. No part of this manual may be reproduced by any means, nor may the information contained within it be passed to a third party without the prior written permission of the copyright holders.

TECQUIPMENT MA2000 ROBOT  
=====

OUTLINE DETAILS of INTERFACE COMMUNICATION PROTOCOL

=====

GENERAL

The control unit for the MA2000 robot contains a combined interface & motor-controller board, a motor power-amplifier board (on the rear of enclosure), and two front-panel printed circuit boards, together with power supply modules. The control unit is a self-contained controller in so far as the robot mechanics can be operated manually using front panel controls. However it is designed to be driven from a host computer which must supply the controller with required positions for each joint of the robot mechanics. The controller will then perform PID digital servo control to achieve the set positions.

The main interface & motor-controller board is a dedicated single-board microcomputer with a 6502 cpu, operating at 1MHz clock rate. The primary task of this microcomputer is to perform real-time PID control of the six DC-motors which operate each joint of the robot; all six motors are serviced approximately every 20 msec. Secondary tasks are to communicate with the host computer; to obtain position set-points, updated PID control terms, and to report joint positions etc. It is also required to interrogate the front-panel controls, for manual operation of the joints during TEST.

The main task of the host computer is to interpret the requirements of the user as to what posture is required of the robot's joints, and to translate these into a stream of motion set-point commands which it must send to the motor controller interface. Secondary tasks are to scan the teach keypad, (if the host software uses this to obtain the user's commands), to read the process inputs, to set the process outputs, to set the pneumatic gripper, to read the status of the controller front-panel, to instruct the motor-controller micro to report on the current joint positions, control errors & motor powers, and finally to update the PID control terms to be used by the motor-controller.

MEMORY MAP of INTERFACE as VIEWED by HOST COMPUTER

The operating software for the motor controller is held in ROM together with the default PID control parameters. Areas of RAM are used for workspace and as 'Shared Memory' for communication with the host computer. The main interface board also contains a multiplexed 12 bit Analogue-to-Digital convertor, 8 bit Digital-to-Analogue convertors, and various Programmable Interface Adaptors and Digital Latches etc. Four of the latches and 128 bytes of 'Shared RAM' are mapped onto the memory area of the host computer; the rest of the interface is not directly accessible to the host, but information on certain aspects can be relayed by the motor-controller microcomputer via the shared-memory.

For memory mapping a contiguous 256 byte (100Hex) memory slot is required; i.e. the bottom 8 address bits, 8 data bits, Read/NOT-Write, a decoded Page Select, and a memory clock signal.

If a suitable area for external memory is not available on the host memory map then a pseudo memory expansion bus will have to be generated using programmable input/output ports (such as the 8255 on Intel machines). It will be required to generate 8 address lines, 8 data lines, a Read/NOT-Write, and a Select Strobe. Such a method is inevitably slower than direct memory mapping, and may restrict the maximum speed of movement of the robot joints. Link 9/10 at the front left-hand edge of the main interface board must be swapped over if this pseudo memory bus is used.

There are two areas of communication.

- 128 bytes of SHARED MEMORY, Addresses 0 to 7F Hex.  
for data exchange with the MOTOR CONTROLLER MICRO-COMPUTER.
- 2 READ only PORTS, Addresses 80 & 81 hex.
- 2 WRITE only PORTS, also Addresses 80 & 81 hex.  
for PROCESS INPUTS & OUTPUTS  
and for KEYPAD and FRONT-PANEL CONTROLS.

The above addresses are the LOW byte. The HIGH byte of the address may be any to suit the host, provided a unique decoded Select or high address bit is available as an interface select. For the "Acorn BBC" the 1MHz expansion bus is used with addresses FC00 to FCFE (FRED)

Connection to the combined interface & motor-controller board should be made via the COMPUTER ribbon cable connector on the front of the control unit. A table of the pin connections is given at the end of this document.

SHARED MEMORY MAP

For each of the 6 motors:

|                              | Hex address |                 |
|------------------------------|-------------|-----------------|
| Set-points, low byte         | 0 to 5      |                 |
| high byte                    | 8 to D      | range 0 to 4096 |
| Positional errors            | 10 to 15    | signed 7 bit    |
| Measured positions, low byte | 20 to 25    |                 |
| high byte                    | 28 to 2D    | range 0 to 4096 |
| Motor powers                 | 30 to 35    | range 0 to 255  |
| Proportional gains           | 38 to 3D    | range 1 to 19   |
| Integral gains               | 40 to 45    | 0 to 19         |
| Derivative gains             | 48 to 4D    | 0 to 19         |

PORTS

OUTPUT PORT A, address 81 Hex, WRITE only.

|             |                            |
|-------------|----------------------------|
| bits 0 to 4 | KEYPAD COLUMNS 0 to 4      |
| 5 & 6       | not used                   |
| 7           | GRIPPER OUPUT, 0 off, 1 on |

INPUT PORT A, address 81 Hex, READ only.

|             |                       |
|-------------|-----------------------|
| bits 0 to 3 | KEYPAD ROWS 0 to 3    |
| 4 to 7      | PROCESS INPUTS 1 to 4 |

OUTPUT PORT B, address 80 Hex, WRITE only.

|            |                                               |
|------------|-----------------------------------------------|
| bits 0 & 1 | PROCESS OUTPUTS 1 & 2                         |
| 2 & 3      | not used                                      |
| 4 to 6     | STATUS COMMAND <u>to</u> motor controller     |
| 7          | SHARED MEMORY FLAG <u>to</u> motor controller |

INPUT PORT B, address 80 Hex, READ only.

|        |                                                 |
|--------|-------------------------------------------------|
| bit 0  | TEST led on front panel, 1 is on                |
| 1      | AUTO led                                        |
| 2      | STOP led                                        |
| 3      | EMERGENCY STOP, 1 is stopped                    |
| 4 to 6 | STATUS <u>from</u> motor controller             |
| 7      | SHARED MEMORY FLAG <u>from</u> motor controller |

At SWITCH-ON

When the mains to the controller unit is switched on, the motor micro-controller automatically starts-up. When the EMERGENCY STOP is released the micro, reads the positions of the robot's joints, obtains the default PID control terms from it's ROM, and immediately controls the motions with the current positions as set-points.

If the posture of the robot is such that any motion is OUT-of-LIMITS, then the EMERGENCY-STOP will not latch-out and the motions will first have to be driven INTO LIMITS using the front-panel manual controls, and the EMERGENCY-STOP operated and then reset. Whenever the EMERGENCY-STOP is operated the micro-controller resets the set-points to the actual position of the joints.

The above action does not require a host computer to be connected. The controller will initialise itself and maintain safe control of the robot without any action being required from the host.

NOTE If the host computer is connected and goes through an automatic memory checking routine on switch-on or reset, then there is a danger that erroneous data may be inadvertantly written into the shared memory of the robot interface. This can easily be avoided by ensuring that the control unit is powered up after the host, or by ensuring that the front panel does NOT have AUTO selected and that the EMERGENCY-STOP is tripped. Only release the EMERGENCY-STOP when the host has finished it's power-up routine. Do NOT select AUTO until the host software is loaded and running.

## COMMUNICATION PROTOCOL

### SHARED MEMORY ACCESS

Because the shared memory is on the memory map of both the host and the motor-controller a strict communication protocol MUST be followed.

Bit 7 of the READ-only-PORTB and of the WRITE-only-PORTB are used as a hand-shake to control access to the shared memory. Because the motor-controller microcomputer must not be held-up in its task of PID control, it's software is arranged such that if the host is using the shared memory it will not wait for access to the memory. Consequently if the host never relinquishes the memory then the micro will always be denied access to new set-points etc.

Before a read or write to shared memory the host MUST

- 1 Read bit 7 of PORTB.  
If clear (0) then repeat step 1 until bit 7 is set.  
The maximum wait will normally be under a millisec.  
If set (1) then the motor-controller has access.
- 3 Wait until bit 7 of PORTB is clear (0)  
This synchronises the host access to follow micro access.
- 4 Write bit 7 (set to 1) to PORTB
- 5 Read bit 7 of PORTB  
to check motor-controller has not also just grabbed access.  
If clear (0) then you have access, go to step 6.  
  
If set (1) then  
Clear bit 7 (set to 0) and go back to step 3.
- 6 Read or Write data as required to shared memory.  
Note that the micro will not have access to this until step 7.
- 7 Clear bit 7 (set to 0)  
This relinquishes shared memory.

NOTE: To be certain the motor-controller has time to access the shared memory the host should not access it again for 25 msec.

### SET-POINTS for JOINT POSITIONS.

These may be updated into the shared memory at any time when the host has access. Associated LOW & HIGH byte pairs MUST always be sent together within the same access slot. When the host relinquishing shared memory the motor-controller will perform PID control on the new setpoints. If this is significantly different from the previous value the robot will move VERY rapidly as the PID control algorithm will have a large ERROR.

It is the RESPONSIBILITY of the HOST COMPUTER to supply a smooth succession of set-points, such that the robot mechanics do not make unduly violent excursions. See the next section on reading the current position of the robot joints, so that on switch-on the host knows where the robot is, and should then compute a smooth path to the desired start position.

**BECAUSE OF THE HIGH POWER OF THE MOTORS,  
FAILURE TO DO THIS COULD RESULT IN DAMAGE TO THE MOTOR GEARBOXES!!**

The set-points send to the motor-controller must be scaled to suit the ADC data range. The electrical angle of the potentiometers is approximately 330 degrees whilst the normal operating angles of the joints are 270 degrees for the three major axes and 180 degrees for the three minor axes. The 12 bits of the ADC, which gives 0 to 4096, must therefore be scaled accordingly; generally the operating range will be approximately from 500 to 3500.

#### STATUS COMMANDS from HOST to MOTOR-CONTROLLER

To cause the motor-controller microcomputer to report information back to the host, the host must send STATUS commands via bits 4,5 & 6 of the WRITE only PORTB.

STATUS action is indicated by setting bit 6

Status 1: LEAD by NOSE. bit 6 only set.

Instructs motor-controller to follow any movement of the joints, disregarding the set-points last set by the host, and also report back the current joint positions.

Status 2: REPORT ERRORS, POSITIONS & POWERS bits 4 & 6 set.

Control is maintained at the current set-points, but also the current errors, positions and motor powers are reported back into shared memory. Note that after sending this command at least 25 msecs must be allowed for the microcomputer to transfer the information into the shared memory, before the host makes it's accesses to read the data. While the host has the shared memory the motor-controller cannot access it to write data.

This feature should be used at start-up so that the host can find out the joint positions and thus initiate a smooth path to the first required posture.

Status 3: NO INTEGRAL ACTION bits 5 & 6 set.

This causes the motor-controller to use a zero integral term in its PID control algorithm, irrespective of the current value held in it's memory. This can be useful to prevent overshoot due to integral 'wind-up' when executing large or rapid slewing motions.

Status 4: UP-DATE PID CONTROL TERMS bits 4,5 & 6 set

Control is maintained at the current set-points. The current values of the Proportional, Integral & Derivative control gain terms for each motor are reported back into shared memory. If these values are subsequently altered by the host then the new values are used for the control algorithm. The control algorithm in the motor-controller uses look-up tables to determine the gain for each term; there are 19 tables each with a different slope (gain), the number entered is a pointer to one of these gain tables and not an actual value of gain. The PID values sent MUST be in the range 0 to 19; a zero Proportional term should also not be employed.

It is STRONGLY RECOMMENDED that increases over the default values are made in small steps, as it is very easy to make the robot violently unstable if high values are used.

STATUS commands should be cancelled, by clearing bits 4,5 and 6, as soon as the required function is completed. Remember to allow time for each computer to have access to the shared memory where exchange of data is required.

STATUS FLAGS from MOTOR-CONTROLLER

If whilst performing the PID algorithm, the motor-controller finds that the error for any motion reaches 127 or -127 then it sets a status flag to advise the host that no more set-points should be sent for the time being.

Bit 6 of the READ only PORTB will be set to 1 when this flag is set.

The host computer should also take note of the status of the front panel LEDs and emergency-stop as indicated by bits 0 to 3, of this same port.

### WATCH-DOGS

The motor-controller microcomputer board has two hardware watch-dog circuits. One is continually re-set by the microcomputer, and will only time-out if this develops a fault and will automatically trip the emergency stop. The CONTROLLER STATUS FAULT LED will also be lit.

A second watch-dog is re-set by the host Select signal every time it accesses the interface. The time-out is approximately 3 seconds. The only effect is for the COMPUTER FAULT LED to be lit.

### CLOSE-DOWN

Before switching the system off the robot should be driven to a mechanically stable position, either under computer control or manually via the front panel, so that it does not slowly collapse when power is removed.

COMPUTER CONNECTOR  
PIN ALLOCATIONS

SEE DRAWING. SC/OU/20.  
COMPUTER SOCKET IS IN UPPER-LEFT CORNER  
OF CCT. DIAGRAM.

J.B.

MA2000  
Tuning for Various Activities

VEK/AAW/0485/i

© TecQuipment Limited

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without the express permission of TecQuipment Limited.

Whilst all due care has been taken to ensure that the contents of this manual are accurate and up to date, no responsibility, direct or consequential, can be accepted for any errors or omissions.

A Packing Contents List is supplied with the equipment and it is recommended that the customer checks carefully the contents of the package(s) against the list ensuring that no items are missed.

Should there be any items missing, the customer should contact the local TecQuipment agent or TecQuipment direct.

1. Introduction

- (a) Feedback - d.c. drives and potentiometers
- (b) P.I.D. - software equivalent
- (c) The need for tuning

2. Tuning using Function 99

- (a) Continuous Path
- (b) Sort by Weight
- (c) Straight line motion

3. Automatic Tuning

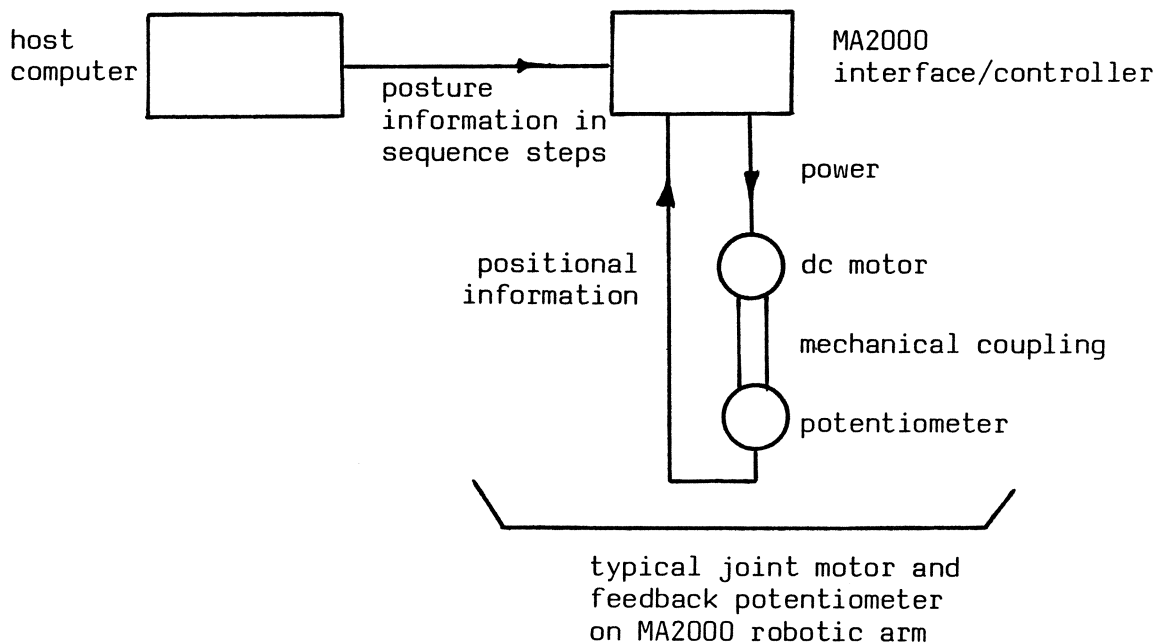
- (a) PROCpath
- (b) PROCTune (MODE 99)
- (c) PROCuser (MODES 7 and 8)

## 1. INTRODUCTION

### 1.1 MA2000 Feedback

The MA2000 robot is a servo-controlled mechanism powered by dc electric motors and using potentiometers to sense position. The position of each joint is used to control the robot by comparing the desired posture with the actual posture. This method of comparison and correction is called feedback.

The MA2000 system comprises host computer, MA2000 interface/controller, and MA2000 robotic arm. The desired posture for each sequence step is passed to the interface/controller by the external host computer. The interface/controller contains the amplifiers which power the dc motors, and also a 6502 microprocessor which performs the comparisons and calculates the corrections for each axis of the robot. The interface/controller also houses the ADC (analogue to digital convertor) which turns the voltage on each potentiometer into a number. That is to say, the feedback loops run through the interface controller.

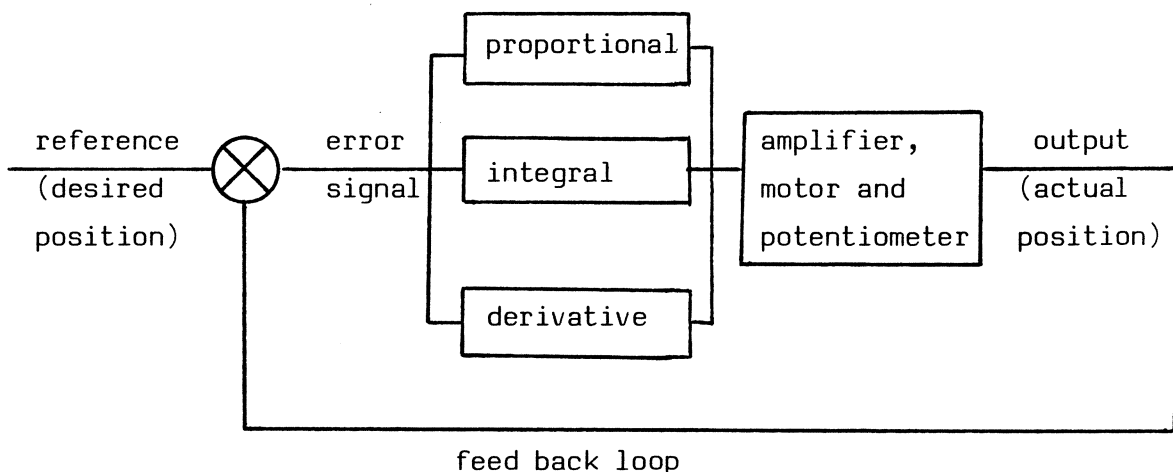


## 1.2 P.I.D. three-term control

In the MA2000 interface/controller 3-term feedback is used - so-called P.I.D. control. In this style of feedback loop the error signal is processed before being amplified to power the motor. The error signal itself is one constituent of the feedback, forming the PROPORTIONAL term. In addition, the error is integrated over time, to form the INTEGRAL term, and is also differentiated with respect to time, to form the DERIVATIVE term. These three terms, P, I and D, are each factored, then added together, and finally amplified to drive the motor. This occurs for each robot axis or joint.

In many servo systems, the integration and differentiation is done using hardware devices, but in the MA2000 interface/controller this is done in software. The proportional term is calculated by simply multiplying the error signal by a factor. The integral term is the result of a summation of the error over a period of time, which happens to be approximately 3 seconds. The derivative term is calculated as the difference between two successive error signals, as the computer cycles through its operating program.

The factors by which these terms are multiplied, before summation and amplification, are held in ROM in the MA2000 interface/controller, but they can be altered, either by using Function 99 on the teach pendant (see MA2000 Manual), by inserting MODE 99 steps within a sequence (see Section 3.2) or by writing some coding in BASIC to exploit the established procedures within the existing software (see MA2000 Experiments Manual). All of these methods are covered in this manual.



### 1.3 The need to alter the tuning

As already mentioned, the factors for the PID terms of each joint are held in the MA2000 interface/controller on a ROM. Whenever the controller is switched on or the RESET button is pressed (after the EMERGENCY STOP button has been pressed), these default values are copied into the interface/controller's RAM, to be available for computation.

The default values of these variables have been chosen to give generally good robot performance in a wide range of activities, but this does not mean they will necessarily be ideal for all applications. In an industrial robot the calculating power of the associated computer or computers is very much greater than the power of the 6502 processor in the MA2000 interface/controller. Indeed, many industrial robots have several processors, possibly as many as one per joint, with one more co-ordinating and administering the others. These robots are obviously very much more expensive than the MA2000, and their computers have the size and the speed to allow the feedback gain to be calculated during motion, dependent on the current posture and the style of movement.

Such a scheme has not been built into the standard MA2000 software, since it would rob the user of some of the memory available for sequence steps, and it would slow down the program to some extent. It is not impossible, though, as we shall see in section 3 of this manual. As a compromise, the standard software does allow specific tuning of the PID factors via Function 99, and we shall see this first, in section 2.

## 2. Tuning using Function 99

Function 99, which is available through the MA2000 teach pendant, gives access to the PID factors. When Function 99 is invoked the current values of the factors for all the motions or axes are displayed on the screen. In addition, if the correct "access code" or software key is typed through the host computer keyboard, the PID factors can be altered.

Wild and extravagant movements can be produced in the robot by inexperienced (or malicious) misuse of this feature. Therefore, to prevent unauthorised use, the access code is not made public, but is notified to the individual who is responsible for the robot at the time the robot is delivered.

The PID factors can take the values 0-20, where the value 20 gives 100% feedback and 0 gives zero %. It is recommended that these terms be altered by no more than 2 units at a time in any exploratory session, so that the effect of the change on the robot's response can be monitored. Once the optimum tuning has been established for a particular sequence the PID factors can obviously be set directly each time the sequence is run, regardless of how great the difference from the default values. Some typical sets of PID factors are given below.

### 2.1 Continuous Path

The MA2000 Continuous Path feature is a compromise between functionality and host computer memory. The path being taught to the robot is discretised at intervals, and the intervals are relatively coarse in order to conserve memory.

If you are playing back a Continuous Path step at RATE 9 it may seem jerky. The motion can be improved by altering the PID terms, and the suggested values in the table are quite a good starting point to begin any search for optimal values. Obviously terms which suit one Continuous Path sequence may not be ideal for another.

P.I.D. terms for RATE 9 with Continuous Path

|                    | Motion 1 | 2 | 3 | 4 | 5 | 6 |
|--------------------|----------|---|---|---|---|---|
| Propn.<br>term     | 4        | 4 | 4 | 4 | 4 | 4 |
| Integral<br>term   | 2        | 4 | 2 | 4 | 4 | 4 |
| Derivative<br>term | 3        | 6 | 3 | 3 | 3 | 3 |

2.2 Sort by Weight

When sorting by weight using the power in the motors (see MA2000 Experiments Kit Manual p.39-41) the choice of algorithm will depend on the exact posture at the time the motor power is checked and also on the motor (or motors) being monitored. Similarly, the time during which the checking needs to be carried out will depend on the style of movement which precedes the check, since the dynamic response of the robotic arm will persist into the checking period. After this response has died away, the robot will be controlled by feedback, as described in Section 1, and inertia forces will still be influential.

Periodically, a correction of posture will take place, under the control of the integral constituent of feedback, and this can perturb the measured value of motor power. It is therefore quite a good idea to remove the integral term completely from the shoulder, the elbow and the pitch axes, irrespective of which motors are being monitored. If this is done, the decision value used in the PROCuser algorithm may have to be different from the value when the integral is present.

Note, the algorithm shown on page 41 of the MA2000 Experiments Kit Manual is relatively insensitive to inertia forces, and the integral terms can be zero or they can be the default values.

### 2.3 Straight Line Motion

The MA2000 Experiments Kit manual describes how the robot can be made to follow a straight line in a radial direction outward from the base of the robot, a few centimeters above the baseboard. Such a line is defined by only 2 consecutive sequence steps, as explained in Section 4.34 pp 27-28 of the Experiments manual.

With the default PID factors, the path which the robot follows will be generally straight but will have an unsymmetrical zig-zag pattern superimposed. This pattern can be reduced by altering the PID factors for the shoulder and elbow joints.

To remove the major periodicity in the pattern, the integral terms should be reduced. To minimise the disymmetry, the proportional term for both of the joints should be equal. To reduce the magnitude of the zig-zag, by increasing its frequency, the differential term should be increased.

The following PID factors have been found to be satisfactory when applied to both the shoulder and the elbow.

P = 9

I = 0

D = 15

### 3. Automatic Tuning

In Section 1.3 above, the concept of automatic or adaptive tuning was introduced, where the feedback terms are altered dynamically as the robot moves, dependent on such things as the range and speed of movement. A very rudimentary form of self-tuning is incorporated in the standard MA2000 software. Section 3.1, which follows, describes this, and suggests how it can be extended. By contrast, Section 3.3 shows how the tuning can be put under the complete control of the software, using posture, speed, amount of movement, weight carried, error in position, etc, to make the choice. Naturally, though, the longer the coding which governs the tuning, the less the memory available for sequence steps. A compromise is achieved with MODE 99, which is described in Section 3.2.

You will need to refer to a listing of the standard MA2000 software. In addition you will need to be aware of the function of the routine called PROCstatus. This procedure is used to prepare the combined system of host computer and interface/controller for a change in activity. This is usually associated with a change in the nature of data transfer between the two micro processors - that is between the host computer micro and the interface/controller micro. PROCstatus requires one parameter, as follows:-

PROCstatus (1) allows lead-by-the-nose.

PROCstatus (2) makes ready for CALL REPORTEPP, which lists the errors, positions, and motor powers for each joint.

PROCstatus (3) completely turns off the integral action for all the joints.

PROCstatus (4) makes ready for CALL REPORTG which lists the current PID factors.

PROCstatus (0) restores the standard communication between the two micros, and must always follow the use of other PROCstatus calls.

### 3.1 PROCpath - Switching off the Integral Dependent on Range and Speed of Move

PROCpath, which is defined at line 1400 in the standard MA2000 software, organises the movement of the robot in those MODES where point-to-point motion is required, i.e. MODE 2, MODE 3, MODE 4 and MODE 5. The movement is produced by the assembly code routine SAMOVE, which is CALLED at line 1435. Before SAMOVE is CALLED, PROCstatus is invoked, with a variable saint% as its parameter. This variable takes the value 0 or 3, dependent on a conditional test on line 1434. That is to say, the integral action is turned off in certain circumstances.

The basis of the conditional statement is a test of the range of increment, as measured by the variable V%, and of the rate of movement, as defined by the variable rate%. The determination of V% takes place on line 1415 and, as can be seen, V% is set to the maximum value of joint angle movement, expressed in terms of the difference between the posture numbers for the current step and the step about to commence.

The precise condition set in the standard software removes the integral for all movements where RATE is greater than 4 and the maximum posture number change is 20. This has the effect of toughening up the robot's muscles for small slow moves where greater precision might be required, and removing the dampening effect of the integral when fast moves over greater distances are called for.

For specific tasks, the conditional statement can be altered. The values of rate% = 4 and V% = 20 can obviously be changed very easily, and when this is complemented by careful choice of the integral terms (using Function 99), then the MA2000 robot can be very finely tuned.

Alternatively, the nature of the conditional statement can be changed. For instance, a different variable\* (say V1%) could be used to calculate the extent of movement of one or two specific joints, rather than all the joints, when some particular task is being undertaken.

---

\* V% as calculated on line 1415 is used elsewhere in the program.

Though this style of coding is very simple, it can be used to great effect in tuning the robot.

---

### Postscript

MODE 6 was left out of the list of MODEs at the beginning of this section, because continuous path movements are organised by PROCcontpath not PROCpath. In the standard MA2000 software PROCcontpath is defined at line 1370, and the integral action is switched off at line 1375 using PROCstatus(3). The integral action can be restored by changing this to PROCstatus(0), in which case the integral terms in the table on page 5 will be suitable.

Be wary of simply editing out the characters :PROCstatus(3) which would obviously have the same effect. Continuous path sequences which have been SAVED to disc or cassette using one version of the program will be corrupted if LOADED by another version which is different in length. The versions need not be identical, but they must be the same size as determined by the number of available sequence steps.

Providing an automatic means of deciding whether to include the integral action is much more complicated for a continuous path step, and has not been provided in the standard software.

### 3.2 PROctune - changing PID during sequence execution

In Issue 3.75 of the standard MA2000 software there is a new MODE - MODE 99. This has been created to allow the PID factors to be changed while a sequence is running. In a MODE 99 step, the RATE is used to define the axis or joint to be altered, and INPUT, OUTPUT and WAIT are used for P, I and D respectively. JUMP can still be used in the normal way.

To change all the PID factors, six MODE 99 steps are required, but it is more likely that only one or two axes will need to be altered at any stage in a running sequence. This will obviously depend on the nature of the activity generated by the sequence.

A listing of the program lines which incorporate the introduction of MODE 99 is given on page 12, and an example sequence is listed below. Note the difference in the way the robot arm stops at the end of step 4 during the first pass through the sequence when the default values of PID exist, compared with the same activity on subsequent passes when the rotate PID terms have been changed at step 5.

#### SEQUENCE 99

|         |     |     |     |     |     |     |
|---------|-----|-----|-----|-----|-----|-----|
| STEP 1) | 7   | 2   | 0   | 0   | 0   | 0   |
|         | 500 | 350 | 100 | 500 | 500 | 500 |
| STEP 2) | 7   | 2   | 0   | 0   | 0   | 0   |
|         | 175 | 350 | 100 | 500 | 500 | 500 |
| STEP 3) | 7   | 2   | 0   | 0   | 0   | 0   |
|         | 175 | 803 | 426 | 500 | 500 | 500 |
| STEP 4) | 9   | 2   | 0   | 0   | 0   | 0   |
|         | 595 | 803 | 426 | 500 | 500 | 500 |
| STEP 5) | 1   | 99  | 4   | 2   | 15  | 1   |

### 3.3 PROCuser or PROCpath - complete adaptive control

The style of programming used in PROctune (see lines 9905 and 9910 on page 12) could be put into PROCuser, or could even be written into a non-standard version of PROCpath. PROCuser is described in the MA2000 Experiments manual, and PROCpath is described in Section 3.1 above.

In such usage, the PID terms could be reset dependent on any of the variables associated with a forthcoming step, for example V%. The posture of the robot could be examined, by making some sort of calculation based on the joint angle values which are held in array sa%. Alternatively, the calculation could be based on the difference between the forthcoming joint angles sa%, and the current ones lsa%, much as V% is calculated within PROCpath (see lines 1400 and 1420). The PID terms could be calculated (in the way that different values of K% are calculated on lines 5911 to 5915) or they might be stored in an array of preset values which are utilised dependent on the results of the posture calculations. The possibilities are limitless, but the robotic task will govern which approach is applicable.

```
50%=2:CLS:PRINT'TAB(3)"TECQUIPMENT INTERNATIONAL Ltd"'TAB(7)"Nottingham, ENG
LAND."''TAB(13)"MA2000"''TAB(11)"Issue 3.75"':REM MODE 99
```

```
1325IF mode%=99 PROCtune(rate%,mode%,input%,output%,wait%)
```

```
1330 IF input%=0 OR mode%=4 OR mode%=5 OR mode%=99 THEN GOTO1355
```

```
1355IF output%<>0 AND mode%<>99 PRINT"SETTING OUTPUT"output%:PROCout(output%)
```

```
1360IF wait% AND NOT(mode%=1 AND NOT(input%=0)) AND mode%<>99 THEN PRINT"WAITIN
G FOR"wait%" 'SECS'":J=TIME+wait%*wai%:REPEAT UNTIL TIME>J OR ?hkey OR FNkey>14
```

```
3940PRINT' "New Values "':A$=INKEY$(1500):IF A$<>"Y" PROCstatus(0):@%=5:ENDPROC
ELSE VDU11:PRINT' "Please enter ACCESS CODE ":VDU21:INPUT access$:VDU6:IF access
$<>code99$ VDU11:PRINTTAB(30):ENDPROC
```

```
4070access$=" " :code99$="0184"
```

```
5910V=FNvalue:IF R THEN ON K% GOSUB 5911,5912,5913,5914,5915,5916:ENDPROC ELSE
ENDPROC
5911K%=6+(3*ABS(SGN(mode%-99))):IF V<0 OR V>K% RETURN ELSE rate%=V:RETURN
5912IF V=6 RETURN ELSE IF V=99:mode%=V:RETURN ELSE IF V<0 OR V>9:RETURN ELSE mo
de%=V:IF V=1 OR V>5 rate%=0:RETURN ELSE RETURN
5913K%=20-(16*ABS(SGN(mode%-99))):IF ABS(V)>K% RETURN ELSE IF mode%=99 AND V<0:
RETURN ELSE input%=V:RETURN
5914K%=20-(18*ABS(SGN(mode%-99))):IF ABS(V)>K% RETURN ELSE IF mode%=99 AND V<0:
RETURN ELSE output%=V:RETURN
5915K%=20+(235*ABS(SGN(mode%-99))):IF V<0 OR V>K% RETURN ELSE wait%=V:RETURN
5916IF V<-1 OR V>TS% RETURN ELSE jump%=V:RETURN
```

```
9900DEF PROCtune(R,M,p,i,d)
```

```
9901REM =====
```

```
9905IF M<>99 ENDPROC ELSE IF R AND R<=N%:R=R-1:PROCstatus(4) ELSE ENDPROC
```

```
9910WSPA%(R)=p:WSPB%(R)=i:WSPC%(R)=d: ?drivem=R:CALL SETG:PROCstatus(0):ENDPROC
```

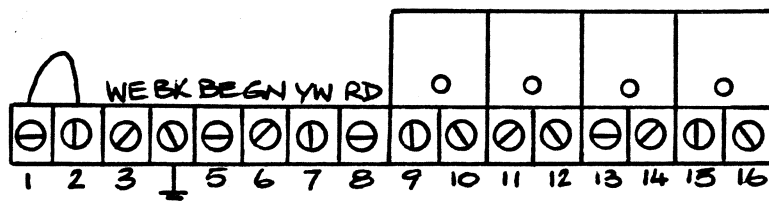


## MA2000p BUTTON BOX

### CONNECTION AND USE

The TecQuipment robot control button box, MA2000p, which is often referred to as the 'decision box', can be used to control running sequences and to simulate external switches. The box contains four normally open (push-to-make) switches, labelled 1 2 3 +4, and one normally closed (push to break) switch labelled -4. A toggle switch is used to select either button +4 or button -4.

The cable attached to the box has six cores, red (RD), yellow (YW), green (GN), blue (BE), black (BK), and white (WE). These should be connected to the rear of the robot micro controller/interface as follows:



Terminals 1 and 2 form the emergency stop loop.

Terminal 3 is the return from the signal wire on the robot end effector.

Terminal 4 is the ground or earth.

Terminals 5 through 8 are INPUT channels 4,3,2,1 respectively.

Terminals 9 through 16 are the four OUTPUT pairs.

When the MA2000 sensing grippers are used, the toggle switch on the decision box should be switched to -4, and the INPUT value -4 should be used in sequence steps to detect an object within the gripper (see MA2000a Experiments Kit Manual, section 7.2, but note that the decision box takes the place of the wire from terminal 3 to terminal 5).

With the toggle switched to +4, the button box can be used to control a decision loop, as in Note G, p.28 of the MA2000 manual "The Open University Robot". This style of sequence control is used in SEQUENCE 1234 supplied with MA2000 software Iss.3.80 onwards.

An alternative use in simpler sequences is to halt the execution at certain critical phases, by introducing an INPUT request to be satisfied by the operator via the button box when he is sure that it is safe or useful to proceed. For example, putting INPUT 1 in STEP 1 of a sequence which cycles (i.e. last step has a JUMP 1) will mean that the sequence will halt after every cycle until button 1 is pressed.

AAW/1285/i



**TecQuipment International**  
Equipment Suppliers, Consultants and Project Contractors for  
Education, Training and Industry  
**Bonsall Street, Long Eaton, Nottingham NG10 2AN, England**  
**Telephone: Long Eaton (0602) 722611**  
**Telex: 377828**

---

ISSUE B

14

Dear sir,

We trust that you are, or will be, pleased with your recently acquired MA2000 robot. We are delighted by the market response, which has been overwhelming and only serves to prove that it is one, if not the best teaching robot available on the market.

As you are no doubt aware, the robot was developed in consultation with the Open University and a version of it is used in their MSC courses "Robotics in Manufacturing" and "Robotics and Computing".

If it was your intention to use your MA2000 robot with the Open University course material it will be necessary for a modification to be made at the TecQuipment factory, as the current version is not compatible with the course material.

In such cases you should contact the TecQuipment sales office for further information as regards the procedure for modification. It should be stressed that this modification is not necessary when the robot is used with the TecQuipment experimental kit and teaching material.

Yours sincerely

Tecquipment Ltd.

# Packing Contents List

**TecQuipment Limited**  
Incorporating  
TQ Electronics

Manufacturers of  
technological laboratory  
equipment designed  
for teaching

Sheet 2



of 3

Registered Office:  
Bonsall Street  
Long Eaton  
Nottingham NG10 2AN  
England

Please note our Telephone  
Number has changed to  
Long Eaton (0602) 722611

Machine

EXPERIMENTS KIT mk II

Reference

MA 2000A

Serial number

124

| Item | Quantity | Part No.  | Description                                       | Check List |
|------|----------|-----------|---------------------------------------------------|------------|
| 16   | 1        | 051-41695 | Type 21 Burr (Cylindrical)                        | X          |
| 17   | 1        | 820-31192 | Drilling Jig                                      | X          |
| 18   | 4        | 000-41784 | Re-usable Wrapping ties                           | X          |
| 19   | 1        | 820-31196 | Aluminium Channel                                 | X          |
| 20   | 1        | 820-42482 | "Humbrol" Air Brush Assembly                      | X          |
| 21   | 8        | 820-41786 | 1/2" Cube                                         | X          |
| 22   | 1yd      | 096-41699 | "Velcro" Hook                                     | X          |
| 23   | 1yd      | 096-41700 | "Velcro" Hoop                                     | X          |
| 24   | 2        | 000-31351 | Crayon Jaw                                        | X          |
| 25   | 1        | 096-41696 | Pen "Major Accent" Fluorescent                    | X          |
| 26   | 1        | 621-42457 | General purpose Sensor Gripper(c/w Screws & nuts) | X          |
| 27   | 2        | 096-42456 | Crayon                                            | X          |
| 28   | 1        | 621-31201 | Drill Lead                                        | X          |
| 29   | 2        | 820-31195 | Tywrap Clamp                                      | X          |
| 30   | 20       | 071-10288 | TYB 25M Tywrap                                    | X          |

Date

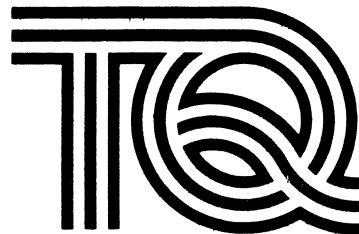
*[Handwritten signature]*

# Packing Contents List

**TecQuipment Limited**  
Incorporating  
TQ Electronics

Manufacturers of  
technological laboratory  
equipment designed  
for teaching

Sheet 1



of 3

Registered Office:  
Bonsall Street  
Long Eaton  
Nottingham NG10 2AN  
England

Please note our Telephone  
Number has changed to  
Long Eaton (0602) 722611

Machine

EXPERIMENTS KIT mk II

Reference

MA 2000A

Serial number

124

| Item | Quantity | Part No.  | Description                                | Check List |
|------|----------|-----------|--------------------------------------------|------------|
| 1    | 6        | 085-41686 | Plastic Bottles                            | X          |
| 2    | 1        | 820-31190 | Bottle Rack                                | X          |
| 3    | 2        | 820-41676 | Bottle Jaw with screws                     | X          |
| 4    | 4        | 820-41687 | 6" x 5/8" Dowel                            | X          |
| 5    | 4        | 820-41688 | 9" x 5/8" Dowel                            | X          |
| 6    | 2        | 820-31186 | Dowel Jaws                                 | X          |
| 7    | 1        | 820-31194 | Microswitch Assembly                       | X          |
| 8    | 1        | 071-41689 | 13 Amp Mains Plug                          | X          |
| 9    | 2        | 096-41690 | 3" x 8 1/2 Oasis Block (Trim to suit jigs) | X          |
| 10   | 2        | 820-31187 | Plug Jaw                                   | X          |
| 11   | 2        | 003-41787 | M6 x 20mm Grubscrew                        | X          |
| 12   | 1        | 051-41691 | Titan Drill with 3 Jaw chuck               | X          |
| 13   | 1        | 051-41692 | 0.7mm Twist Drill                          | X          |
| 14   | 1        | 051-41693 | 1.6mm Twist                                | X          |
| 15   | 1        | 051-41694 | Type 5 Burr (Conical)                      | X          |

Date

*[Handwritten signature]*

# Packing Contents List

**TecQuipment Limited**  
Incorporating  
TQ Electronics

Manufacturers of technological laboratory equipment designed for teaching



Registered Office:  
Bonsall Street  
Long Eaton  
Nottingham NG10 2AN  
England

Please note our Telephone  
Number has changed to  
**Long Eaton (0602) 722611**

Machine

EXPERIMENTS KIT mk II

## Reference

MA2000A

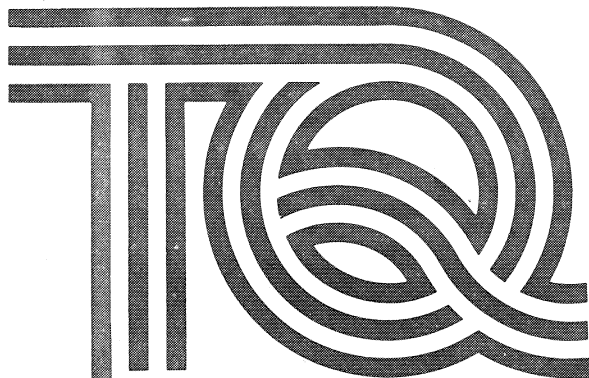
Serial number

124

[illegible]

Date \_\_\_\_\_

AK BLO 1288



## TecQuipment Ltd

Bonsall Street,  
Long Eaton,  
Nottingham NG10 2AN  
England

Telephone:  
Nottingham (0602) 721696

Manufacturers of  
Technological Equipment  
for Education, Training  
and Industry

# TEST CERTIFICATE

**Apparatus**

MACCCC 240V ROBOT & ANCILLRIES

**Serial No.**

**This apparatus has been inspected  
and tests have proved satisfactory**

**Signed**

A handwritten signature in black ink, appearing to be 'J. Clarke'.

**Test Engineer**

The access code for this MA2000 robot is

**0184**

To re-define the access code, change the string  
variable ( code99\$ ) on line 4070. in BBC software  
or on line 8120 in IBM software