# RB ARM USER'S GUIDE

April 1986

General Robotics Corporation Golden, Colorado

Copyright 1986

Copyright 1986 General Robotics Corporation
All Rights Reserved

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise, without the prior written permission of the publisher.

**General Robotics Corporation**
14618 W. 6th Avenue, Suite 150
Golden, Colorado 80401

The following are registered trademarks of the companies indicated:

RB5X: General Robotics Corporation
Robot Control Language (RCL): General Robotics Corporation
Tiny BASIC: National Semiconductor Corporation
Savvy: Excalibur Technologies Corporation

# SECTION A:

# INTRODUCTION

## PHYSICAL DESCRIPTION

The RB Arm is a highly versatile, general-purpose robotic arm designed especially for the RB5X(tm) personal robot. When mounted on the RB5X, this arm has sufficient range of movement to pick up an object from the floor and hand it to the user or deposit it on a table of ordinary height. It has sufficient mechanical power to lift as much as a twelve-ounce load.

The RB Arm duplicates many of the capabilities of the human arm; it can also move in ways yours cannot. Its movement can be illustrated by the following comparison:

- **Shoulder rotation:** Hold your left arm straight out to your side. Then, move your arm in a circle from the shoulder, like an airplane propellor. This is shoulder rotation.

- **Shoulder flexion:** Hold your left arm straight out to your side. Then, move your arm, from the shoulder, so that your left hand is directly in front of you, as if you were pointing at something. This is shoulder flexion.

- **Elbow movement:** Hold your left arm straight out in front of you, with your thumb up. Bend your elbow so that your hand is pointing to your right. If your elbow could also bend so that your hand could point to the left, you would be duplicating the range of elbow movement of the RB Arm.

- **Wrist movement:** Hold your left hand up in front of you, so that you can see the palm. Rotate your hand so that you can see the back of your hand. If you could rotate your wrist in the other direction as well, you would have the wrist movement capability of the RB Arm.

- **Gripper movement:** Put a peanut-sized object on a table. Using your thumb and the first two fingers of your hand, pick up the object with a pinching motion. The RB Arm's gripper works like that.

## TECHNICAL DESCRIPTION

### Electrical

The RB Arm uses the same paired, 6-volt (12-volt total), rechargeable, sealed lead-acid batteries as the base RB5X. In addition, to compensate for the increased power drain of the arm movements, auxiliary batteries are mounted on the arm baseplate and wired in parallel with the main batteries. Programmed movement commands are received from an output port of the INS8073 chip on the CPU (central processing unit) board inside the robot.

### Mechanical

The arm movements are driven by sophisticated, stepper-motor-driven gear trains incorporated into the arm mechanism. (Stepper motors allow much more precise motion control than conventional electric motors.) The electrical characteristics of stepper-drive motors preclude motor burn out if the arm should jam against an obstacle.

# SECTION B:

# GENERAL ASSEMBLY NOTES

## WHAT COMES WITH THE RB ARM PACKAGE

- One arm assembly with two batteries, and attached 50-conductor cable and light-emitting diode (LED) wire harness
- One printed circuit (PC) board
- One arm shell with 16 socket-head screws and one allen wrench
- One arm power wire harness
- One eight-conductor flat cable
- One six-foot, 26-conductor flat cable
- One one-foot, 26-conductor flat cable
- One arm controller box
- One package of miscellaneous hardware
- One arm training software module

## WHAT TOOLS YOU NEED TO INSTALL THE RB ARM

To install the RB Arm on your RB5X base unit, you need the following tools:

- One small flat-blade screwdriver
- One pair of long-nose pliers
- One 3/8" box- or open-end wrench
- One 5/64" allen wrench (provided)

## HANDLING AND INSTALLING COMPONENTS

### Printed Circuit Boards

Printed circuit boards (also called "cards" or "PC boards") are highly reliable, but can be damaged by careless handling or installation. Cracks in the board are the most common damage,

usually resulting from bending or warping during installation or removal. Be careful not to flex a PC board; once cracked, it is unreliable and the whole assembly must be replaced.

Some PC boards can also be damaged by static electricity. Before handling a board, discharge any static electricity in your body by touching some unpainted surface on the robot (the metal, vertical struts, for example).

When installing a board, do not "wiggle" it to seat it in its socket. Push straight down on the board, or gently "rock" it along the long axis. To remove the arm board, lift up on the top levers, which provides an action that gently withdraws the card from its socket.

**Cables and Connectors**

Cables used in the RB5X and on the RB Arm are the flat, ribbon-type. This design reduces electrical noise, allows for easy and accurate assembly, and provides a very flexible cable that is capable of withstanding considerably more flexing than the conventional round cable.

When installing ribbon cables, make sure they are safe from being pinched between two metal parts. Also, don't crease or tightly twist ribbon cables. And **NEVER** pull on the ribbon cable itself to unplug it. That's a sure route to cable failure! Instead, pull on the connector. As with all electronic parts, treat ribbon cables with respect and they'll never give any trouble.

The cable connectors used on the RB5X have a positive retaining mechanism, rather than depending on friction fit to hold them in place.

Some cable connector/socket units (usually the smaller ones) have a hook-like feature to hold them together. When joining these connectors and sockets, the hooks can be used as a guide to proper orientation. When separating these units, hold the cable connector itself, rock it gently from side to side, and pull. **NEVER** pull on the wires coming out of the connector.

Other sockets have latches to hold the connector in positive contact. (For example, see Figure 3.) These latch mechanisms have levers on the lower ends that lift the connectors out of the socket during removal. When inserting connectors in latch sockets, hold the latches open just far enough to let the connector slip past them. When the connector is started into the socket, release the latch and push the connector the rest of the way into the socket. If you hold the latch too far open, the lower part of the extraction lever projects into the connector's path and prevents it from fully seating into the socket.

The connector is fully seated when the retaining latch hooks slip over the top of the connector. To remove the connector, move the latch ears away from the connector. The extraction levers at the bottom of the latches gently lift the connector out of the socket, and the connector can be manually moved out of the way.

Electronic connectors always have some indication of the proper orientation for insertion into a socket. If there is any question about proper orientation of RB Arm connectors, refer to the appropriate photo in this Guide. All cable connections are illustrated.

**Hardware**

In nut and bolt assemblies, star- or split-lock washers go under the nut; flat washers go under the lock washer. If there are two flat washers, one should be installed under the head of the screw and one under the lock washer. A lock washer is only used with a screw head when a nut is not part of the fastening design. When lock washers are used, the nut only needs to be tightened firmly. Don't overtighten.

Machine screw sizes in this Guide are referred to by a standard convention: a series of numbers, such as 6-32 x 3/8" or 4-40 x 1". The number before the dash refers to the diameter of the threaded portion. The larger the number, the greater the diameter. A number 8 screw is larger than a number 4.

The number following the dash refers to the number of threads per inch. Bigger diameters have fewer threads, but we have to give the number to distinguish between coarse and fine threads. The number following the dash describes the length of the part of the screw that goes below the surface to be fastened. A 6-32 x 5/8" screw is called a "six thirty-two by five eighths."

# SECTION C:

# SEQUENTIAL ASSEMBLY INSTRUCTIONS

## INSTALLING THE RB ARM ASSEMBLY

The following information provides step-by-step instructions on installing the RB Arm on your RB5X base unit. We suggest that, if you have the space, you remove all the photographs at the end of this chapter, and spread them out on a table or other flat surface near your work area so that you may more easily refer to the figures as we suggest them to you.

A-1. Switch RB5X off.

A-2. Remove the upper shell/dome assembly by twisting it 1/16 of a turn counterclockwise as you look down on the robot, while holding the lower part (the blue part) of the RB5X steady. This releases the bayonet clips on the inside of the robot. The upper shell/dome assembly may then be lifted off.

A-3. Remove the four, 6-32 x 3/8" screws holding the clear plastic, LED plate to the metal, vertical struts (A, Fig. 1). Disconnect the LED cable from the mother board (B, Fig. 1) by opening the connector latches and pulling the cable free. Set the LED plate and attached cable aside.

A-4. CAREFULLY turn the robot upside down and stand it GENTLY on the vertical struts, making certain not to bend the struts. Be sure that the connector at the top of the interface panel is hanging over the side of the table and carrying none of the robot's weight (Fig. 2).

A-5. Attach the arm power wire harness by removing the 10-32 nuts on the back of the charger pins (A, Fig. 3). Place the connector for the blue wire over the screw that already holds a blue wire; place the connector for the white wire over the screw that already holds a white wire. Replace the nuts.

A-6. Feed the red and black wires and the white connector in the middle of the arm power wire harness through the hole in the robot's baseplate (B, Fig. 3).

A-7. Carefully turn the robot right side up.

A-8. Disconnect the eight-conductor cable on the mother board at J14 and push it aside (A, Fig. 4). DO NOT disconnect this cable from the underside of the robot.

A-9. Connect the eight-conductor wire that comes with the arm package to J14. Make sure that the plugged hole on the cable connector coincides with the missing pin on J14 (A, Fig. 4).

A-10. Move the robot close to a chair or a box about the same height as the RB5X. Place the arm assembly on the chair or box.

A-11. After ensuring that you are grounded, remove the arm PC board from its protective plastic sleeve. Save this envelope for future use or storage.

A-12. Lay the board on top of the card cage with the components facing up.

A-13. Feed the ~~three~~ four-conductor wire harness attached to the arm assembly through the oval slot in the arm baseplate and attach the free end to socket J2 on the top right side of the arm PC board (A, Fig. 5). Note that the connector is keyed and only fits easily in one direction.

A-14. Connect the arm power wire harness (fed through the baseplate in Step A-6) to socket J1 (behind A, Fig. 5) on the back of the arm PC board. Make certain that the lip on the wire connector mates with the tab on J1, and that all four pins are in the socket. Hang the red and black wires to the outside over the card cage side.

A-15. Connect the one-foot, 26-conductor cable to socket AJ3 on the top of the arm PC board (B, Fig. 5), matching the embossed arrows on the socket and the connector. Drape this cable over the arm PC board. You will connect the other end later.

A-16. Slide the arm PC board into slot J3 (C, Fig. 4), with its components facing the interface panel, and push it firmly into place on the mother board.

A-17. Route the 50-conductor cable through the oval slot in the arm baseplate (A, Fig. 9) and connect the free end to socket AJ2 on the top of the arm PC board (B, Fig. 6 and D, Fig. 5). Make certain that the tab on the wire connector fits into the notch on the socket. (If there is no tab, orient the red wire of the 50-conductor cable away from the ribbon cable already connected to this PC board.) Tuck any excess cable between the PC boards in the card cage.

A-18. Remove the LED card from the plastic standoffs that attach it to the clear LED plate (C, Fig.1) by pinching the ears at the top of the standoffs with a pair of pliers. Save this plate with the standoffs still attached in case you ever want to use your robot again without its arm.

A-19. Attach the LED card to the top of the black metal LED plate on the arm assembly (A, Fig. 7), using 4-40 x 3/4" screws, nuts, and lock washers, and the 3/8" standoffs provided.

A-20. Set the arm assembly on the four vertical struts on the robot, with the batteries towards the interface panel and square to it (Fig. 8).

A-21. Secure the arm assembly to the vertical struts with the four, 6-32 x 3/8" screws that you removed in Step A-3.

A-22. Route the LED cable behind the arm, and through the slot in the center of the arm assembly (B, Fig. 7).

A-23. Attach the LED cable to the mother board at socket J9 (B, Fig. 4). Note the tab on the cable connector that fits into the notch on the board connector.

A-24. Route the black and red wires up through the empty round hole between the batteries on the arm assembly (Fig. 8). Connect the terminals on the red wire to the positive terminals on the batteries; connect the terminals on the black wires to the negative terminals on the batteries. (They are marked with embossed "+" and "-" signs.)

**NOTE:** Please make sure to connect the RED wire before connecting the black wire. This prevents the accidental shorting of the batteries, which can occur when one red wire terminal is connected and the other touches metal.

A-25. Insert the two eight-conductor cables from the mother board and the motor control board to the pins on the bottom of the arm board that point toward the robot's motor battery (behind C, Fig. 5). (It doesn't matter which cable connects to which set of pins.) Here again, note the plugged hole in each socket and the missing pin on each connector.

A-26. Route the one-foot, 26-conductor cable from Step A-15 through the oval slot (A, Fig. 9) containing the 50-conductor cable, and attach the free end to the bracket containing the LED and push-button switch (B and C, Fig. 9). Mount this end using two sets of 4-40 x 3/4" screws, nuts, and lock washers; the screws should be inserted from the outside. The mounting lugs on the connector should be on the inside. It is very important that you make sure that the red conductor on the cable is at the bottom of the connector.

A-27. Release the arm by removing the two screws that hold it to the arm tower.

A-28. Remove the plastic tie holding the gripper closed.

A-29. Ensure that the arm ribbon cable (Fig. 10) will not be pinched or stretched during arm movement. Slack in the cable should be drawn out beyond the shoulder joint and tucked between the shoulder joint and the arm electrical bracket (A, Fig. 6).

**INSTALLING THE ARM UPPER SHELL/DOME ASSEMBLY**

B-1. Remove the polycarbonate dome from the upper shell by removing the eight, 6-32 x 1/4" socket-head machine screws, using the provided allen wrench.

B-2. Attach the dome to the arm shell using the eight, 6-32 x 3/8" socket-head machine screws provided with the arm package. (Note that these screws are longer.) Get all of the screws started, but not tight, before you tighten any of them. It may be necessary to shift the shell and dome around a little to get all of the holes lined up. After all the screws are started, tighten them, trying not to distort the dome or shell in the process.

B-3. Put the upper shell on the robot (without the arm shell/dome assembly) and twist clockwise into place, making sure that the bayonet clips on the inside of the robot lock securely.

B-4. Attach the arm shell/dome assembly to the upper shell using the eight, 6-32 x 1/4" socket-head screws that you removed in Step B-1. Orient the arm shell so that the arm reset button and the arm controller socket (Fig. 9) fit through the provided opening. Again, start all the screws before you tighten any of them.

## INSTALLING THE ARM CONTROLLER

C-1. Connect one end of the six-foot, 26-conductor cable to the arm controller box with the red lead on the cable oriented closest to the RB Robot logo on the controller (Fig. 11).

C-2. Plug the other end of this cable into the arm assembly at the socket near the arm reset button (Fig. 9 and 11). Make sure that the red lead on the cable is on the bottom edge, and that the arrows on both connectors match up. If the cable installed in A-26 is correct, the red conductors on that cable and on the arm controller box should match up.

# SECTION D:

# CHECK OUT

## VISUAL INSPECTION

1. Are all plugs properly oriented?
2. Are all plugs fully seated?
3. Make sure that no wires are pinched.
4. Make sure that all battery connections are correct and secure.
5. Make sure there is no loose hardware.

## ELECTRICAL CHECK USING THE ARM CONTROLLER

1. Remove any software module that may be in the socket on the interface panel.
2. Switch RB5X on.
3. Connect the arm controller to the arm assembly (Fig. 11).
4. Switch on the arm power, using the toggle switch at the top right of the arm controller (Fig. 11).
5. Using the various arm movement toggle switches, move each arm joint through its full range of motion:

   Shoulder rotation
   Shoulder flexion
   Elbow movement
   Wrist rotation
   Gripper movement, open and closed

   Each joint should move smoothly and quietly. You may move any number of joints simultaneously.

   **NOTE:** Do not attempt to rotate the shoulder unless the arm is extended beyond the body of the robot, using the shoulder flexion motor.

5. Check the motor wheel operations.

   **Forward movement:** Push the two motor drive switches on the controller box away from you. The robot moves forward, and the right and left front LEDs on the LED plate light up.

**Reverse movement:** Move the motor switches toward you. The robot moves backwards, and the right and left rear LEDs come on.

**Left turn:** Move the right motor switch away from you. The robot turns left; the right front LED comes on. Move the switch toward you. The robot moves backwards to the left; the right rear LED comes on.

**Right turn:** Move the left motor switch away from you. The robot turns right; the left front LED comes on. Move the switch toward you. The robot moves backwards to the right; the left rear LED comes on.

**Left spin:** Move the right motor switch away from you, and the left motor switch toward you. The robot spins to the left. The right front LED and the left rear LED lights up.

Try a right spin. Figure it out for yourself!

# SECTION E:

# CARE OF THE RB ARM

## CLEANING

Wipe or brush dust and dirt off the RB Arm from time to time. The arm is a precision item, and appreciates being clean. Under normal use, the arm is not likely to get dirty enough to require disassembly for cleaning. If you feel you have to disassemble it, BE CAREFUL!

## LUBRICATION

The RB Arm requires no lubrication, and in fact you should avoid it. Grease and oil collect dust. The resulting dirty, oily spots look unsightly, and the dirt caught in the oil is an excellent wear-producing abrasive. **Do not oil anything on the RB Arm!**

## PERIODIC CHECK-OVER

From time to time, you should check that all cable connections are secure and that none are coming loose.

Also check to make certain that cable and wire routing is correct, and that there are no stretched or pinched wires. If a wire or cable won't stay in a safe path, tape it down.

Make sure that all PC boards are fully seated in their sockets on the mother board. Occasionally wiggle parts and joints. If something doesn't seem secure or solid, find out why. Fix it before it gets worse. If you can't correct the problem to your satisfaction, contact General Robotics Customer Service and find out what to do.

# SECTION F:

# PROGRAMMING THE RB ARM

## OVERVIEW

The RB Arm may be programmed either in the robot's native language, Tiny BASIC, or in the optional Robot Control Language(tm) with Savvy(R). We urge arm owners who have RCL to become familiar with the basics of RCL programming before reading the Tiny BASIC section. After you understand how to use the arm-related RCL tasks, you will better understand the detailed discussion in the Tiny BASIC section.

You may also operate the RB Arm using the arm training software module and the arm controller box.

## USING THE ARM TRAINING SOFTWARE MODULE

The arm training software module is used in conjunction with the controller to program the RB Arm to do specific tasks. There is no provision for cancelling a movement command, so you should try to rehearse the arm movements a time or two before actually entering the commands. Use the following procedure:

1. Switch the robot off.

2. Install the arm training software module in the socket on the interface panel and slide the switch to the right of the socket to the position indicated by the label on the module.

3. Connect the arm controller to the panel on the arm shell. The red stripe on the controller cable should line up with the red stripe on the cable at the back of the panel socket.

4. Switch the robot on. The red LED next to the connector comes on with a "click." (Arm power is switched on by software.)

5. Press RB5X's front and rear bumpers (numbers 1 and 5) simultaneously to initiate the training process. (The bumpers are number 1 through 8 as you look down on the robot, starting with the bumper under the sonar and proceeding clockwise.) After a few seconds, the robot sounds its horn and turns on the LEDs to signal that it is ready for input.

6. Using the switches on the controller box, move any motor or combination of motors. The arm repeats each movement that you command. (There is no provision for false starts. If you do make a false start, move on to Step 7 or to Step 12, and then start over.) The blinking LEDs go out as your command is received.

7. Press any bumper to indicate that the movement is complete. The robot's LEDs come back on to acknowledge that the command is in memory. Omit this step if you are rehearsing movements.

8. Repeat Steps 6 and 7 for each arm motion that you wish to program.

9. After the last movement is entered, press bumpers 3 and 7 (the two side bumpers) to indicate the end of your program. The robot sounds its horn and the LEDs go off.

10. Press the front and rear bumpers (numbers 1 and 5) simultaneously to have the robot execute the program you have entered.

11. In addition to arm movements, the arm training software module provides for pauses between movements. To program a pause after a movement, press bumper 2 for a one-second delay, bumper 1 for five seconds, and bumper 8 for 10 seconds. Press several delay bumpers in succession (pressing any other bumper in between) to add several delays together.

12. To start a new training session, press the front and rear bumpers (numbers 1 and 5) a second time. To exit the training mode, press bumpers 2 and 4 (right front and right rear) simultaneously.

13. Switch off power to the arm when you are finished using it, as the motors drain the batteries even when they are not being used.

14. Be sure to switch RB5X off before removing the arm training software module.

**NOTE:** Always turn off power to the arm using the toggle switch on the controller before turning off the power to the robot itself. Otherwise, all the arm motors move for a few seconds until a capacitor in the automatic shutdown circuit discharges.

You can train only the arm motors using the arm training software module, not the robot's motor wheel movements. To operate the wheel motors, manually operate the motor drive switches on the arm controller during the arm program operation.

## USING ROBOT CONTROL LANGUAGE WITH SAVVY

### Getting Started

The optional Robot Control Language with Savvy, available from your local dealer or from General Robotics Corporation, is designed to simplify the writing of robot tasks for the RB5X, and includes arm movement programs. The following discussion assumes that you are at least familiar with programming RB5X using RCL.

The RCL tasks listed below are the specific arm tasks:

```
MOVE THE ARM FROM SHOULDER
MOVE THE FOREARM
TURN THE WRIST
MOVE THE HAND
```

The BUILD function of RCL compiles -- or translates -- these RCL programs into the Tiny BASIC instructions the robot understands.

Each arm-related RCL instruction must include two items of information: 1) the arm joint to be moved, including the direction of movement, and 2) the amount of movement you want.

A typical RCL arm instruction would be:

```
MOVE THE FOREARM OUT for this many degrees:  45
```

or

```
TURN THE WRIST CCW for this many degrees:  15
```

Practice using the four arm commands in RCL programs.

### Getting Good

Study the detail of the four arm tasks listed in Appendix 2 of the RCL Programmer's Manual. Each task includes all of the parameters required by the robot for proper execution of the particular task. Each of these parameters is a variable in the task. The variables, and their use, are explained below.

| Variable | Meaning |
|---|---|
| M, N | Arm joint select and direction |
| D | CPU port |
| P | Movement magnitude |

### Variables M and N

The values assigned to variables M and N are assigned by the RCL task according to the movement direction (in, out, up, down, clockwise, or counterclockwise) written into the task. There are designated high and low statuses for each of the arm's motors (or joints), and by alternating between these statuses, you can make a particular motor move.

| | High (M) | Low (N) |
|---|---|---|
| Motor 1: Shoulder | | |
| Up | 2 | 0 |
| Down | 3 | 1 |
| Motor 2: Shoulder | | |
| In | 8 | 0 |
| Out | 12 | 4 |
| Motor 3: Elbow | | |
| In | 48 | 16 |
| Out | 32 | 0 |
| Motor 4: Wrist | | |
| Clockwise | 192 | 64 |
| Counterclockwise | 128 | 0 |
| Gripper Motor | | |
| Open | #F2 | #F0 |
| Close | #F3 | #F1 |

### Variable D

The value of variable D, which designates the CPU port, is determined within the RCL task:

| | |
|---|---|
| Shoulder, elbow, wrist | #7805 |
| Gripper | #7806 |

### Variable P

The value of variable P, the magnitude of joint movement, is calculated within RCL tasks, based on the degree or percent values that you specify. Take care that no arm joint is driven against a mechanical stop, as damage may result to the gear train. Although the arm tasks in RCL include error traps to ensure that excessive movement is not called for by a single task, it is possible that the cumulative effect of several consecutive tasks could be excessive movement.

| | |
|---|---|
| Shoulder rotation: | 39 to 14040 |
| Shoulder flexion: | 39 to 14040 |
| Forearm movement: | 11 to 3960 |
| Wrist rotation: | 11 to 3960 |
| Gripper: | 4.25 to 425 |

The robot is also capable of moving several joints simultaneously (see Section III, "Using Tiny BASIC"). There are no RCL tasks written to provide for this, but using Savvy alone, you can write your own tasks to get this result.

A sample "compound movement" task follows. This Savvy program has the robot move its elbow in and out, and rotate its wrist clockwise and counterclockwise:

```
MOVE THE ELBOW AND WRIST (IN,OUT) AND (CW,CCW): <1> AND <2>
for this many degrees:  <3>
 1 IF the <1> is IN then
 2    IF the <2> is CW then
 3       COPY the "M=240:N=80" to TEMP
 4       END of test
 5    OTHERWISE
 6       IF the <2> is CCW then
 7          COPY the "M=176:N=16" to TEMP
 8          END of test
 9 OTHERWISE
10    IF the <1> is OUT then
11       IF the <2> is CW then
12          COPY the "M=224:N=64" to TEMP
13          END of test
14       OTHERWISE
15          IF the <2> is CCW then
16             COPY the "M=160:N=0" to TEMP
17             END of test
18          END of test
19    END of test
20 PASTE the "D=#7805:" in front of TEMP
21 COPY from FRONT to TEMP
22 MULTIPLY the <3> by 20.16
23 SPLIT the number PRODUCT
24 COPY from INTEGER to PRODUCT
25 PASTE the "P=" in front of PRODUCT
26 PASTE the TEMP in front of FRONT
27 Z COMPILE the BASIC statement FRONT which means "SET THE
MOTOR, ETC."
28 Z COMPILE the BASIC statement "GOSUB 3400" which means
"MOVE the ARM"
29 Z INSURE ARM PULSE SUBROUTINE IS
30 HALT and return to Ready
31 END
```

See the Savvy manual that came with your RCL package for further details on programming multiple arm movements using Savvy.

The various I/O-port and arm-power-on/off commands are provided for within RCL. The line "Z INSURE ARM PULSE SUBROUTINE IS" is this vital RCL task. It switches the arm power on, initializes the I/O ports, and switches the power off again. **YOU MUST INCLUDE THIS LINE** in all of the arm tasks you write yourself.

**Imbedded Tiny BASIC Commands**

The BUILD task in RCL automatically generates a Tiny BASIC program for the robot, once you have specified the values of the various variables used by Tiny BASIC. For the most part, this is done for you by the appropriate RCL tasks. Look at the RCL task MOVE THE FOREARM in Appendix 2 of the RCL Programmer's Manual. Line 2 says:

COPY from "D=#7805:M=32:N=0:" to TEMP

This line specifies the values of variables D, M, and N. Then some calculations are done, resulting in a number called PRODUCT. Line 13 puts a variable name to PRODUCT:

PASTE the "P=" in front of PRODUCT

Resulting in "P=nnnn" Stored in Front.

Then in Line 14:

PASTE the TEMP in front of FRONT

resulting in "D-#7805:M=32:N=0:P=nnnn" stored in variable FRONT. The part in quotes is a Tiny BASIC language statement. So,

Z COMPILE the BASIC statement FRONT ...

puts the variables and their values in the Tiny BASIC program. Then,

Z COMPILE the BASIC statement "GOSUB 3400"

puts that statement into the program. Finally,

Z INSURE ARM PULSE SUBROUTINE IS

puts more statements into the Tiny BASIC program. Take a look at Z INSURE ARM PULSE SUBROUTINE IS, as detailed in Appendix 2 of the RCL Programmer's Manual.

When you begin writing your own RCL tasks for the RB Arm, you will have many opportunities to use this information in specifying your own variable values. Increasing your skill in RCL programming is primarily a matter of practice.

## USING TINY BASIC

Full information on programming RB5X in Tiny BASIC is contained in the RB5X Reference Manual and in the Using NSC Tiny BASIC manual, available from General Robotics Corporation. The following information is intended only as an aid to you in developing program segments pertaining to the RB Arm.

**NOTE:** Certain numbers in the following text are prefixed with a pound sign, #. This symbol identifies hexadecimal -- or base-sixteen -- numbers. This identifier must be included if values are intended to be in hexadecimal.

Shoulder and elbow movements are defined as if the arm were folded in under the dome to start. Clockwise (CW) and counterclockwise (CCW) wrist movement is defined by looking down the arm from the shoulder.

All arm movements are caused by electrical pulses received by the appropriate arm motors. The pulse routing depends on two values, high and low, which are loaded into memory address #7805 (for the shoulder, elbow, and wrist) or #7806 (for the gripper) on the INS8073 chip. These values are specified in the arm movement programs we write. The values for the various movements are:

| **Address #7805** | **High** | **Low** |
|---|---|---|
| Motor 1: Shoulder | | |
| Up | 2 | 0 |
| Down | 3 | 1 |
| Motor 2: Shoulder | | |
| In | 8 | 0 |
| Out | 12 | 4 |
| Motor 3: Elbow | | |
| In | 48 | 16 |
| Out | 32 | 0 |
| Motor 4: Wrist | | |
| Clockwise | 192 | 64 |
| Counterclockwise | 128 | 0 |

Several joints can move simultaneously. If we program a value of 60, for example, into #7805 for high and 20 for low, we have used values that are the result of unique combination of numbers (assuming that only one high and one low number for each motor are used). The number 60 can only result from 12 + 48 and 20 can only 16 + 4. The shoulder then moves out (12 and 4) and the elbow moves in (48 and 16). Using this principle, all four joints using location #7805 can be programmed to move at once.

Gripper movement is different:

| Address #7806 | High | Low |
|---|---|---|
| Open | #F2 | #F0 |
| Close | #F3 | #F1 |

A discussion of the meaning of the high and low values is beyond the scope of this Guide; however, the principle is similar to that described in the RB5X Reference Manual for the bumper codes.

Stepper motors, as used in the RB Arm, respond to electrical pulses, one pulse for each increment of rotation. To move a particular joint from one extreme of movement to the other, the following approximate numbers of pulses must be received by the motor:

| | |
|---|---|
| Shoulder rotation: | 14040 |
| Shoulder flexing: | 14040 |
| Forearm movement: | 3960 |
| Wrist rotation: | 3960 |
| Gripper: | 425 |

Pulse commands may be any positive value. Care should be taken not to program so many pulses that the joint jams against a mechanical stop. This can result in damage to the associated gear train.

Arm movement programs must include several command elements:

1. Initialize the I/O port for the arm driver program:
   @#7807 = #90

2. Switch on the arm power:
   @#7806 = #70

3. Enable the I/O port to receive motor pulses:
   @#7806 = #F0

4. Specify high and low motor values:
   @#7805=nn:@#7805=nn

A subroutine to move the elbow out might look like this:

```
1000 FOR I = 1 TO 500        (Specify number of pulses)
1010 @#7805=32:@#7805=0      (Define high and low values)
1020 NEXT I                  (Repeat the loop)
```

The program moves the elbow out 500/3960 of its full range of movement.

If we substitute @#7805=48:@#7805=16 in line 1010, the elbow moves in by the same amount.

To move two motors simultaneously, simply add the motor values. To move the shoulder out and the elbow in at the same time, you might enter:

```
1000 FOR I = 1 TO 1500       (Specify number of pulses)
1010 @#7805=60:@#7805=20     (Define high and low values)
1020 NEXT I                  (Repeat the loop)
```

These are the same high and low values used in the general discussion at the beginning of this section. Because one pulse moves the shoulder and the elbow different amounts relative to their full range of movement (1/14040 and 1/3960), we see considerable elbow movement and relatively little shoulder movement resulting from this particular program element.

A subroutine to open the gripper might look like this:

```
1000 FOR I = 1 TO 425        (Specify number of pulses)
1010 @#7806=#F2:@#7806=#F0   (Define high and low values)
1020 NEXT I                  (Repeat the loop)
```

The program moves the gripper from fully closed to fully open.

Arm stepper motors draw power whether they are actually moving the arm or not. To conserve battery power, switch power off to the arm whenever you are not using it by entering the following:

```
@#7806=#B0
```

A complete Tiny BASIC program to move the wrist and then the elbow might look like this:

```
10 REM PROGRAM TO ROTATE THE WRIST AND MOVE THE ELBOW
20 REM INITIALIZE THE DRIVE I/O PORT
30 @#7807 = #90
40 REM SWITCH ON ARM POWER
50 @#7806 = #70
60 REM INITIALIZE THE MOTOR PULSE I/O PORT
70 @#7806 = #F0
80 REM DEFINE AMT OF WRIST MOVEMENT
90 A=500
100 REM DEFINE WRIST CLOCKWISE HIGH AND LOW
110 B=192:C=64
120 REM GO TO MOVEMENT SUBROUTINE
130 GOSUB1000
140 REM DEFINE AMT OF ELBOW MOVEMENT
150 A=350
160 REM DEFINE ELBOW OUT HIGH AND LOW
170 B=32:C=0
180 REM GO TO MOVEMENT SUBROUTINE
190 GOSUB1000
200 REM TURN OFF ARM
210 @#7806=#B0
220 END
```

```
990 REM BEGIN SUBROUTINE
1000 FOR I = 1 TO A ((Specify number of pulses)
1010@#7805=192:@#7805=64 (Define high and low values)
1020 NEXT I (Repeat the loop)
1030 RETURN
```

As is the case with any language, human or computer, skill is related to practice. Work with Tiny BASIC. As you become familiar with the RB5X Reference Manual and learn to use the individual commands, add new ones to your repetoire. Don't try to make giant leaps; instead, advance step-by-step. Be patient, be persistent, and you'll be successful.

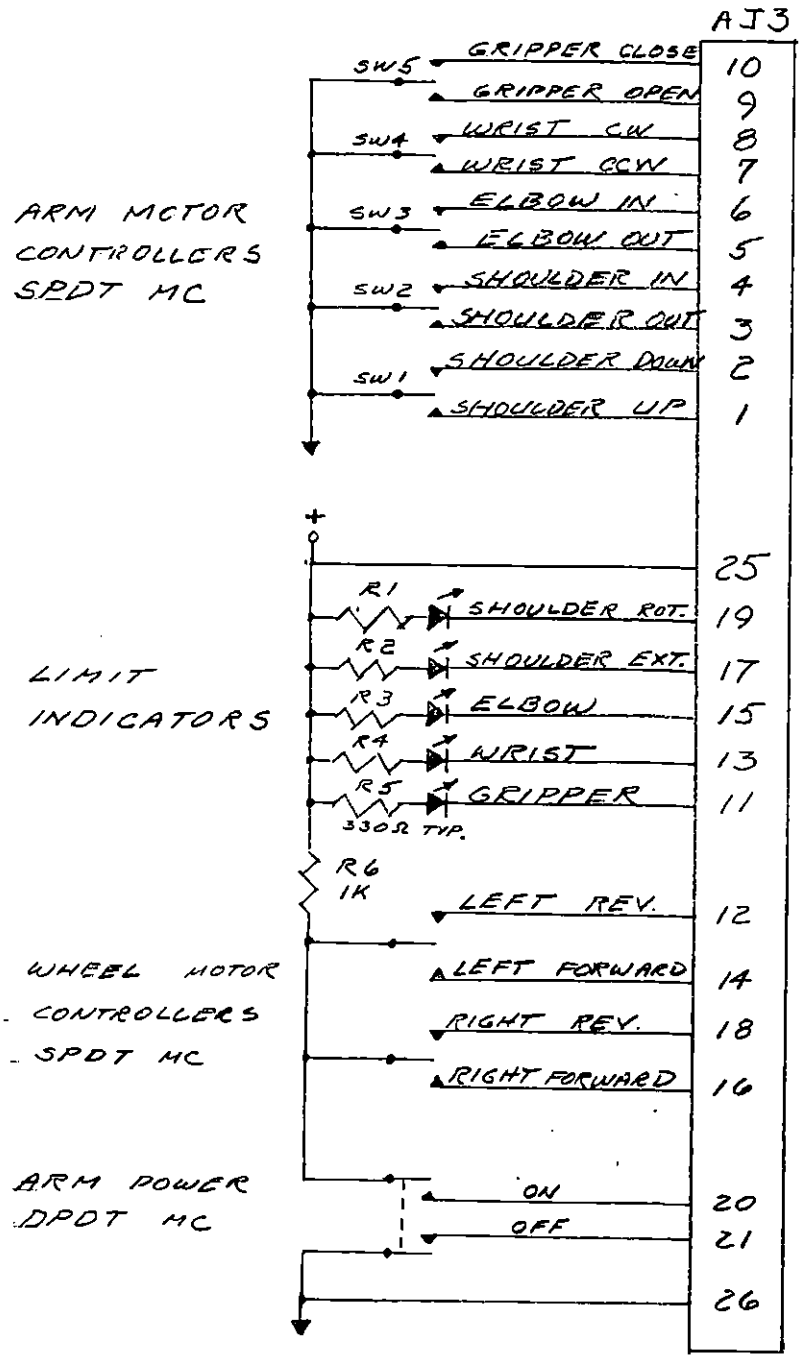
ARM CIRCUIT

PARTS LAYOUT

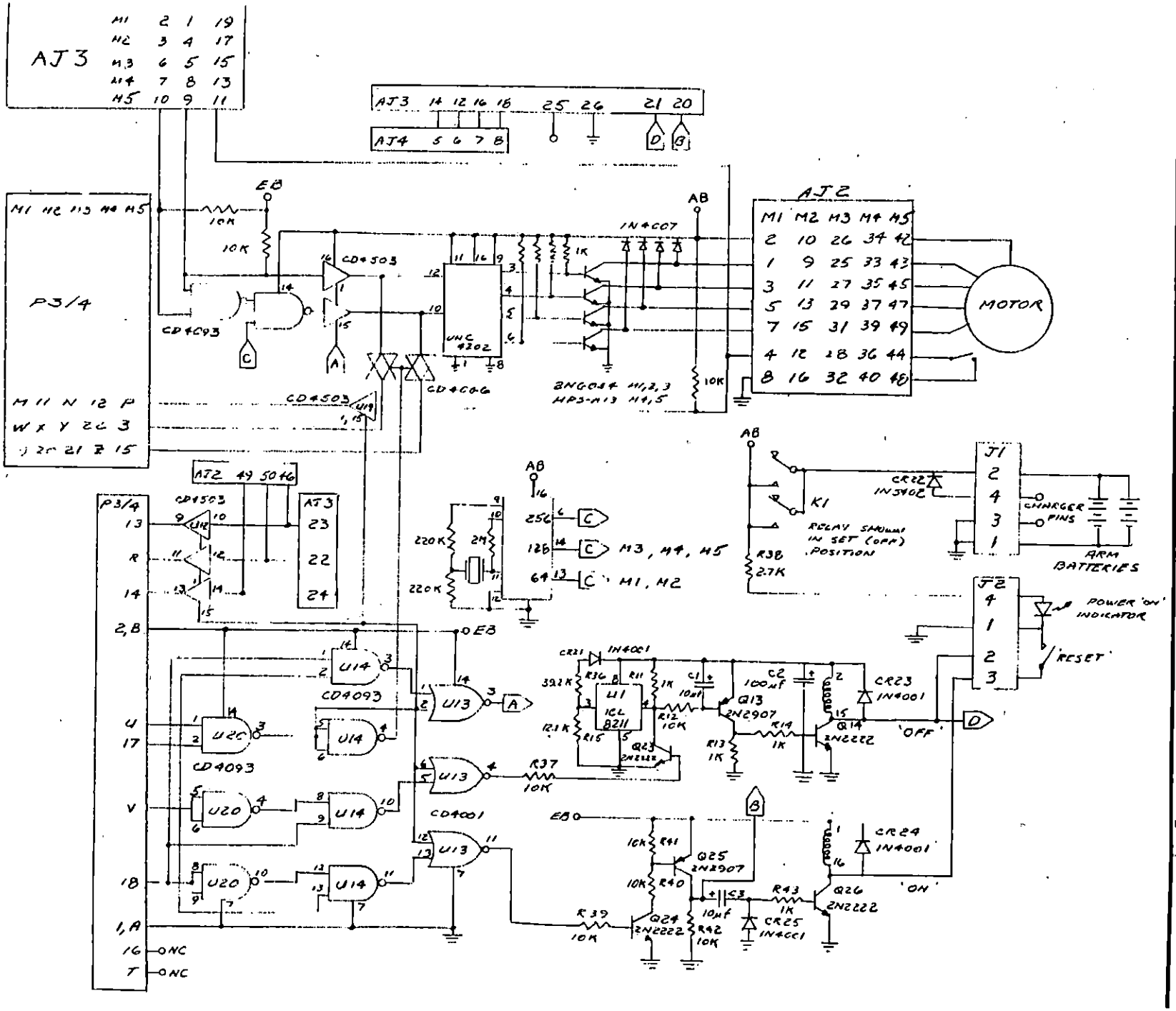
P/N 4010-05000 REV 2

7/10/80

P2/4




| GENERAL ROBOTICS CORP. | | | |
|---|---|---|---|
| ARM PENDANT CONTROLLER | | | |
| ARA | 5/25/89 | | REV Ø |



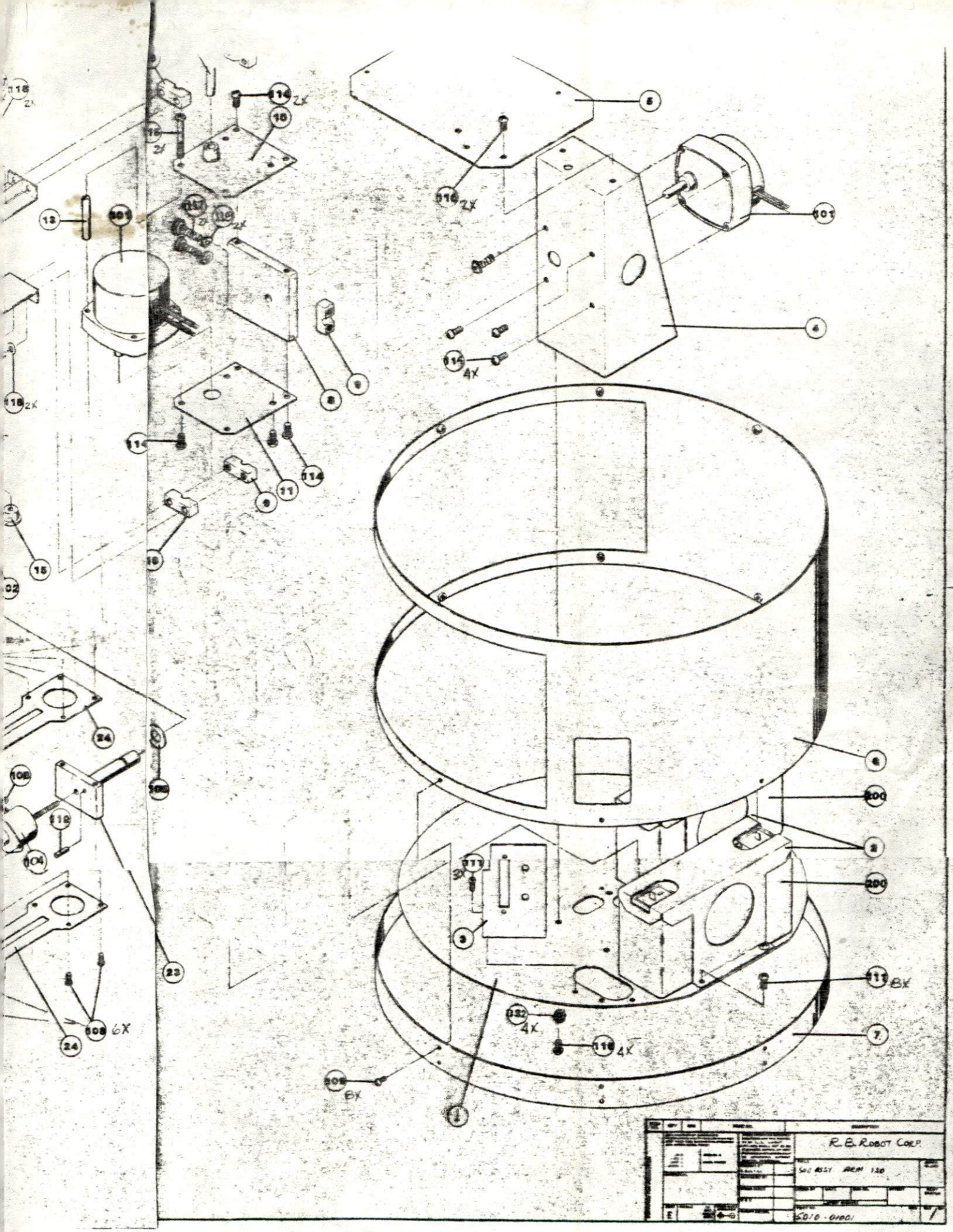

GENERAL ROBOTICS CORP.

ARM CIRCUIT P/N 4010-C5000

| REVISIONS | | |
|---|---|---|
| NO. | DATE | BY |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |

| | | |
|---|---|---|
| DRAWN BY | SCALE | MATERIAL |
| CHK'D | DATE 7/10/9 | DRAWING NO. |
| TRACED | APP'D | |

TOLERANCES (EXCEPT AS NOTED)
DECIMAL ±
FRACTIONAL ±
ANGULAR ±


R.B. ROBOT CORP.
5010-01001

