

Dear RCL(tm) user:

ALL of the MOST ESSENTIAL functions that RCL provides have been TESTED AND are WORKING PROPERLY on our test robot: Bumpers, Sonar, IR, Lights, Horn, Voice, Motion, Arm, Charger, loops, editing, robot program generation, communications to robot, build, load, and various other support functions. We want you to be aware, however, that modifications are currently being made on SOME of the more COMPLEX RCL functions (such as the Tiny-basic-related commands and Beta-intelligence items), which are UNLIKELY TO AFFECT the MAJORITY of RCL users. Of course, if you do find any errors, or have suggestions for improvements, please let us know.

Many useful commands have been added that are not in previous RCL versions. Liberty has also been taken to change the names of several commands to make them more intuitively descriptive of their functions. Please note that appropriate substitutions and corresponding CHANGES NEED TO BE MADE in the Student's and Teacher's RCL Workbooks. They are as follows:

New RCL Command:	replaces	Old RCL Command:
INVENT	(operation)	DEFINE {can still use this}
PICK A NEW DIRECTION	(motion)	PICK A RANDOM DIRECTION
START A RANDOM MOTION	(motion)	MOVE RANDOMLY
DO ASSIGNED MOTION IF BUMPER PRESSED		ASSIGN A MOTOR CODE
MAKE A PHRASE	(voice)	LOAD THE PHRASE
EXIT IF BUMPER	(bumpers)	EXIT IF THIS BUMPER
REPEAT THE LIMITED LOOP	(seq control)	REPEAT THIS LIMITED LOOP
EXIT IF ECHO	(sonar)	EXIT IF SONAR {uses 'count'}
EXIT IF SONAR	(sonar)	{uses LESS/MORE, 'feet'}
MOVE ARM AT THE SHOULDER	(arm)	MOVE THE ARM FROM THE SHOULDER
FIND THE CHARGER	(battery)	X CHARGER FINDER

As you review the accompanying 6-page RCL COMMANDS LIST document, you'll see both familiar and unfamiliar commands. Most are self-explanatory or follow the old RCL conventions. Here are some notes concerning the new and/or altered commands:

DIAGNOSTICS - run occasionally to check integrity of RCL.SVY file, then immediately run JANITOR to recover garbage space.

JANITOR - run just before exiting with BYE for cleanup.

CLONE A TASK - duplicate a task. The NEW task name CANNOT BE an existing task/function/item/system name.

LOAD - used to download to robot the MOST-RECENTLY built program without having to do BUILD AND LOAD. Good for re-running a task over again during a session or when starting up RCL later.

LIST - displays task commands.

LTP - displays the Tiny-basic program generated by the BUILD process.

DUMBO - communicates with robot to change and show what is in robot's memory.

PRINTOUT - will send a listing of task's commands to a printer.

PRINT TASKS - prints a list of available tasks.

PHONEME GUIDE - shows examples of phoneme usage.

PREPARE THE ROBOT - {no longer required, but okay to use}

RANDOMIZE - seeds the random number generator. 0 uses the same default seed whenever robot is turned ON or Reset, and generates the exact same (REPEATABLE) random-number sequence as random numbers are used. Specifying the seed using a number, N, from 1 thru 1000 causes the robot to generate and dispose of the first N random numbers, so that 1000 different (but REPEATABLE) random sequences are available. (Larger seed numbers take increasing time and, when using 1000, will delay up to 15 seconds before the task command following RANDOMIZE can be processed.) A seed of -1 causes the robot to HONK the horn for 3 seconds, then generate and dispose of random numbers until any BUMPER is PRESSED; it is therefore used to start generating a (typically) NON-repeatable sequence of random numbers.

{\ or / or . or --} - list the task TRACK THE TAPE for example use to create 'white space' between lines to make task logic more readable.

INCLUDE - inserts a specific Tiny-basic-statement-numbered instruction into the Tiny-basic program when building a task. {MUST use a " mark as the first character of the desired text; e.g., "GOSUB5}

COMPILE - similar to INCLUDE, but uses the next available Tiny-basic statement number.

FIND THE TAPE - {NOT functional yet}

TRACK THE TAPE - {not yet tested}

TRACK THE TAPE BACKWARD - {not yet implemented}

SPIN RANDOMLY - starts a CW/CCW-selected spin and stops rotation at a random angle within a specified maximum-angle limit.

DO ASSIGNED MOTION IF BUMPER PRESSED - initiates the one motion (of eight possible) associated with that bumper. Example, pressing Bumper-#1 starts 'move forward'.

Version <sup>2.5</sup> ~~3.00~~ <sup>3/16/94</sup> ~~03/02/94~~

## Student/Teacher/User Operational Commands:

```

----- RCL Operation -----
BYE {or} QUIT {or} SYSTEM
<Esc> key
<Enter> key
DIAGNOSTICS
JANITOR
BACKUP RCL {or} BK
----- TASK Manipulation -----
EDIT the <taskname>
INVENT a new task named <taskname> {or} DEFINE ...
CLONE A TASK
----- TASK Operation -----
BUILD AND LOAD {or} BL
LOAD
RUN
----- TASK Examination -----
SHOW TASKS {or} ST
LIST {or} LST
LIST THE PROGRAM {or} LTP
SHOW VARIABLES
DUMBO
----- TASK Documentation -----
PRINTOUT {or} PR
PRINT TASKS
PRINT ASSOCIATIONS
PRINT TBASIC
----- TASK-Making Reference --
PHONEME GUIDE {or} PG
PHONEME CODES {or} PCODES
SHOW SPEECH
VALUE of the <_item_>
----- TASK Identification ----
ASSOCIATE new phrase <_> with old <_>
FORGET the association named <_>
RENAME old name <_> to new name <_>

```

## Task-editing Control:

```

Q {or} alt-Q      (quit and save task)
E {or} alt-E      (edit line above cursor)
D {or} alt-D      (delete line above cursor)
L {or} alt-L      (list the task)
----- Cursor-movement keys for editing:

```

```

Tab, (shft)+Tab
Cursor keys: -->, <--, (ctrl)-->, (ctrl)<--
Backspace, Ins, Del
Enter

```

*Cursor keys: ^, v*

Task-commands Sequence Control:

```
PREPARE THE ROBOT
RANDOMIZE using a seed of (-1,0,1-->1000): <__>
WAIT for <__> second(s).
WAIT RANDOMLY for up to <__> second(s).
-----
CALL this robot task line <"label"> .
JUMP to the line called <"label"> .
-----
```

----- use the following to leave any of the loops below:  
EXIT THIS LOOP

----- each loop BEGIN must have its paired REPEAT:  
BEGIN A LOOP  
REPEAT THIS LOOP  
-----

BEGIN A COUNTED LOOP called <\_\_> beginning at <\_\_> ending at <\_\_>  
REPEAT THE COUNTED LOOP for the counter <\_\_>  
-----

BEGIN A LIMITED LOOP  
REPEAT THE LIMITED LOOP unless <...> is (<,>,<=>,<>) <...> to <...> .  
-----

----- any of these 'no ops' are used as between-line separators to make  
task logic flow more readable:

- \ [ back slash ]
- / [ forward slash ]
- . [ period ]
- [ two dashes or minus signs ]

Tiny-Basic programming tasks:

```
REMARK that: <"text"> .
INCLUDE at <__> the TINY-BASIC statement <"statement"> which means <"text">
COMPILE the TINY-BASIC statement <"statement"> which means <"text">
-----
```

```
RESERVE the Tiny-Basic variable called <"_"> .
CLEAR VARIABLES {or} CLEAR ALL ITEMS
SET the Tiny-Basic variable <"_"> equal to <__> .
CALCULATE variable <"_"> = <"_"> (+,-,*,/) <"_"> <__> .
-----
```

```
TEST IF the variable <"_"> is (<=>,<,>,<>) <"_"> compared to <__> .
END HERE
OTHERWISE DO
```

of  
-t  
w/te

\_\_\_\_\_  
SENSORS  
\_\_\_\_\_

## Battery-charger Sensing:

## MAINTAIN CHARGE

----- use the following to leave a loop:  
 EXIT IF THE BATTERY IS LOW {or} EXIT IF BATT LOW  
 EXIT IF THE CHARGER IS TOUCHED {or} EXIT IF CHARGER TOUCH

## Bumper Detection:

WAIT FOR ANY BUMPER PRESS and then wait for <\_\_> seconds more.  
 ----- use the following to leave a loop:  
 EXIT IF ANY BUMPER TOUCHED  
 EXIT IF FRONT+REAR BUMPER PRESS  
 EXIT IF BUMPER number <\_\_> is touched.

## Infrared Detection:

TURN OFF THE INFRARED LED {or} TURN OFF IR  
 TURN ON THE INFRARED LED {or} TURN ON IR  
 ----- use the following to leave a loop:  
 EXIT IF THE TAPE IS SENSED

-----  
~~FIND THE TAPE~~

? TRACK THE TAPE

~~TRACK THE TAPE BACKWARD~~

MOVE FORWARD TIL TAPE NOT SENSED

? FOLLOW TAPE TO CHARGER

## Sonar Detection and Distance Measurement:

EXIT IF SONAR distance is (LESS,MORE) <\_\_> than (.8-35) <\_\_> feet.  
 EXIT IF ECHO count is less than (95-350): <\_\_> from sonar.

\_\_\_\_\_  
ACTUATORS  
\_\_\_\_\_

## Lights (LED's) Control:

TURN OFF LED number <\_\_> .  
 TURN ON LED number <\_\_> .  
 TURN OFF THE FLASHING LIGHTS  
 TURN ON THE FLASHING LIGHTS  
 TURN OFF ALL THE LIGHTS  
 TURN ON ALL THE LIGHTS  
 LIGHTS ROUTINE

-----  
 CHECK BUMPER LIGHTS  
 CHECK MOTOR LIGHTS

## Sounds, Horn, and Music Control:

HONK the horn for <\_\_> second(s).  
 TURN OFF THE HORN  
 TURN ON THE HORN

-----  
 WARBLE for <\_\_> cycles using cyclic period of <\_\_> millisecs.  
 CHIRP

## Talking Voice Generation and Control:

PREPARE THE VOICE  
 MAKE A PHRASE called <"phrasename"> with phonemes: <"...sequence...">  
 SPEAK the phrase called: <"phrasename">  
 WAIT TO SAY the phrase <"phrasename"> after waiting <\_\_> sec(s).

## Motion and Wheel Control:

STOP ALL MOTION  
 MOVE BACKWARD {or} GO BACK  
 MOVE FORWARD {or} GO AHEAD  
 MOVE DISTANCE BACKWARD for this many feet <\_\_> .  
 MOVE DISTANCE FORWARD for this many feet <\_\_> .  
 MOVE TIMED BACKWARD for <\_\_> seconds.  
 MOVE TIMED FORWARD for <\_\_> seconds.

-----  
 PIVOT ON LEFT CLOCKWISE {or} PIVOT LCW  
 PIVOT ON LEFT COUNTERCLOCKWISE {or} PIVOT LCCW  
 PIVOT ON RIGHT CLOCKWISE {or} PIVOT RCW  
 PIVOT ON RIGHT COUNTERCLOCKWISE {or} PIVOT RCCW

-----  
 GO CLOCKWISE {or} GO CW  
 GO COUNTERCLOCKWISE {or} GO CCW  
 SPIN CLOCKWISE for <\_\_> degrees. {or} SPIN CW ...  
 SPIN COUNTERCLOCKWISE for <\_\_> degrees. {or} SPIN CCW ...  
 SPIN RIGHT 90 DEGREES {or} SPIN R90  
 SPIN LEFT 90 DEGREES {or} SPIN L90  
 SPIN AROUND CLOCKWISE <\_\_> time(s). {or} SPIN ACW  
 SPIN AROUND COUNTERCLOCKWISE <\_\_> time(s). {or} SPIN ACCW  
 SPIN RANDOMLY (CW,CCW,EITHER) <\_\_> for up to <\_\_> degrees.  
 PICK A NEW DIRECTION

-----  
 START A RANDOM MOTION  
 DO ASSIGNED MOTION IF BUMPER PRESSED {or} DO BUMPER-PICK MOTION  
~~MOVE WITH BETA INTELLIGENCE~~

## Arm Components Control:

MOVE ARM AT THE SHOULDER (UP, DOWN, IN, OUT): <\_\_> for (0-360): <\_\_> degrees.  
 MOVE THE FOREARM (IN, OUT): <\_\_> for (0-360): <\_\_> degrees.  
 TURN THE WRIST (CW,CCW): <\_\_> for (0-360): <\_\_> degrees.  
 MOVE THE HAND (OPEN, CLOSE): <\_\_> for (1-100): <\_\_> %.

Applications - Subsystem Tests:

*TEST Voice Errors*  
 TEST *Spin*  
 TEST ARM  
 TEST BUMPERS  
 TEST BUMPER MOTIONS  
 TEST CHARGER  
 TEST IR  
 TEST LIGHTS  
 TEST MOTION  
 TEST SONAR  
 TEST SONAR BOUNDS  
 TEST VOICE

-----  
 VTEST  
 VTUNE

Applications - Student Examples:

PARTY  
 EXPLORE IR

Applications - Demo's:

~~INTRODUCE THE RB5X~~ *Sonar and Voice Demo*  
 PREPARE THE INTRODUCTION  
 MOTION DEMO  
 ARM DEMO  
 AVA DEMO (Amer. Vocational Assoc. Conference)  
 AVA1,AVA2,AVA3,AVA4 (speech examples)  
 FIND THE CHARGER

Applications - Intelligence Experiments:

*ot*  
*it*  
*ailable* ( )  
 ALPHA  
 ALPHA W/SONAR  
 BETA  
 BETA W/SONAR  
 -----  
 INITIALIZE MEMORY  
 INITIALIZE VARIABLES

## Computer Installation and Configuration Controls:

CONNECT CABLE to computer serial port <\_>  
 SET BAUDRATE [ see BAUDRATE below ]  
 CALIBRATE [ see Parameter list below ]  
 ----- system variables:  
 COMMPORT (1=default) [ 1=COM1, 2=COM2, etc. ]  
 BAUDRATE (1200=default) [ 300, 1200, 4800 options ]

## RCL Calibration Parameters:

S/N (Serial number of RB5X used) [example: 0123456 ]  
 TBASIC.INCREMENT (1=default) [ Tiny-Basic statement-number increments ]  
 -----  
 BUMPER.START (YES=default, NO) [ is Bumper press req'd to start task  
 when motion commands are present? ]  
 DELAY.START (3 secs=default) [ time delay to start task after  
 bumper-press startup ]  
 -----

## Drive-motors runtime-to-movement conversion constants:

MOVE.SECONDS/FT (3 secs/ft=default) [ MOVE FORWARD/BACKWARD ]  
 SPIN.SECONDS/TURN (4.95 or 5.05 secs/360-degs=default) [ SPIN AROUND ]  
 SPIN.SECONDS/90DEG (1.38 secs/90-degs=default) [ SPIN CW/CCW 90 DEG'S ]  
 SPIN.DEG/SEC (62 degrees/sec=default) [ SPIN CW/CCW ]  
 -----

## Sonar distance: feet=(ECHO.count-SONAR.OFFSET)/(SONAR.COUNTS/FT) ]

SONAR.COUNTS/FT (22 counts/ft=default) [ EXIT IF SONAR ]  
 SONAR.OFFSET (73 counts=default) [ EXIT IF SONAR ]  
 SONAR.RECOVER (20 msec=default) [ TEST SONAR BOUNDS min recovery ]  
 -----

IR.RESPONSE (NORMAL=default, REVERSE) [ ... TAPE ..(sense/not sensed) ]  
 CHIRP.PERIOD (1 msec=default) [ WARBLE period for CHIRP ]  
 CHIRP.CYCLES (3 cycles=default) [ WARBLE cycles for CHIRP ]  
 -----

ARM.OFFWAIT (150 msec=default) [ arm power-relay min turn-off time ]

## Arm stepper-motors pulse.count-to-movement conversion constants:

ARM.PULSES/DEG (40.38 pulses/deg=default) [ MOVE SHOULDER ]  
 FOREARM.PULSES/DEG (20.16 pulses/deg=default) [ MOVE FOREARM ]  
 WRIST.PULSES/DEG (19.88 pulses/deg=default) [ TURN WRIST ]  
 HAND.PULSES/PERCENT (4.42 pulses/percent=default) [ MOVE HAND ]  
 -----

## Arm stepper-motors speed-control minimum time-delay counts:

HAND.PWAIT (100 loopings @29-microsecs/loop=default) [ HAND ]  
 WRIST.PWAIT (100 loopings @29-microsecs/loop=default) [ WRIST ]  
 FOREARM.PWAIT (150 loopings @29-microsecs/loop=default) [ FOREARM ]  
 ARMPLEX.PWAIT (250 loopings @29-microsecs/loop=default) [ SHOULDER IN/OUT ]  
 ARMROT.PWAIT (300 loopings @29-microsecs/loop=default) [ SHOULDER UP/DWN ]



TRACK THE TAPE (a Task)

```
1 BEGIN A LOOP
2 .
3 --
4 TURN OFF ALL THE LIGHTS
5 PIVOT ON LEFT COUNTERCLOCKWISE
6 \
7 BEGIN A LOOP
8 EXIT IF THE TAPE IS SENSED
9 REPEAT THIS LOOP
10 /
11 TURN ON THE FLASHING LIGHTS
12 MOVE FORWARD TIL TAPE NOT SENSED
13 EXIT IF ANY BUMPER TOUCHED
14 --
15 TURN OFF ALL THE LIGHTS
16 PIVOT ON RIGHT CLOCKWISE
17 \
18 BEGIN A LOOP
19 EXIT IF THE TAPE IS SENSED
20 REPEAT THIS LOOP
21 /
22 --
23 TURN ON THE FLASHING LIGHTS
24 MOVE FORWARD TIL TAPE NOT SENSED
25 .
26 EXIT IF ANY BUMPER TOUCHED
27 REPEAT THIS LOOP
28 STOP ALL MOTION
End. (Task size is 86 bytes.)
```

CORRESPONDING  
REFER TO TINY-BASIC LISTING  
TO SEE INSTRUCTIONS COMPILED  
TO DOWNLOAD TO THE ROBOT.

← 'NO-OP'S USED TO VISUALLY  
DIVIDE LOGIC FOR ENHANCED  
TASK READABILITY.

← USE OF 'NO-OP' LINE  
SPACERS TO SEPARATE  
A 'LOGIC GROUP' (EG, A LOOP)

← E.G., ANOTHER FUNCTIONAL LOGIC  
GROUPING.

( NOTE: THIS LISTING GENERATED BY THE "PRINTOUT" COMMAND,  
(PRINTER SET TO 96 CHARS/LINE)  
USE "LST" or "LIST" TO VIEW THIS ON THE DISPLAY SCREEN,  
USE "EDIT" TO VIEW OR MAKE CHANGES. )

Stmt #	Tiny BASIC Text	Meaning of Text
1	@#7803=#98:GOTO25	Config 7800-7802 ports, start prog
2	@#7801=@#7801 ORX:RETURN	Turn ON = SET a bit
3	@#7801=@#7801 ANDX:RETURN	Turn OFF = CLEAR a bit
5	@#7802=(#@#7802 AND#FO)ORV:RETURN	Activate wheels using V code
7	IFW<1001DELAYW:RETURN	Wait W msec
8	T=W/1000:W=MOD(W,1000)	Calc T-secs, W-msecs
9	IF(W>0)DELAYW	Wait W msec +
10	FORW=1TOT:DELAY998:NEXTW:RETURN	Wait T seconds
25	REM Begin Prog	Start program here.
26	DO:Y=@#7800:UNTIL(Y<#FF)	Wait for any bumper press
27	T=3:GOSUB10	Wait for T-sec startup delay
30	REM Loop Start	BEGIN a Loop .....
31	X=#83:GOSUB3	Turn OFF all LED's
32	V=#08:GOSUB5	Move RIGHT wheel FORWARD
33	REM Loop Start	BEGIN a Loop .....
34	X=#02:GOSUB2	Turn ON LED 1 (IR emitter)
35	IF(@#7802 AND#40)=0 GOTO37	Exit if TAPE is detected
36	GOTO33	REPEAT this Loop _/
37	REM Loop Exit	
38	X=#40:GOSUB2	Turn ON Flashing Lights
39	U=0	Initialize loop count
40	DO:U=U+1	Begin a limited loop ``\
41	V=#09:GOSUB5	Go FORWARD
42	UNTIL((#@#7802 AND#40)=0 OR(U>99))	Check exit conditions _/
43	DELAY100	Msecs to WAIT
44	Y=@#7800:IFY<#FF GOTO60	Exit if ANY BUMPER contact --v
45	X=#83:GOSUB3	Turn OFF all LED's
46	V=#01:GOSUB5	Move LEFT wheel FORWARD
47	REM Loop Start	BEGIN a Loop .....
48	X=#02:GOSUB2	Turn ON LED 1 (IR emitter)
49	IF(@#7802 AND#40)=0 GOTO51	Exit if TAPE is detected
50	GOTO47	REPEAT this Loop _/
51	REM Loop Exit	
52	X=#40:GOSUB2	Turn ON Flashing Lights
53	U=0	Initialize loop count
54	DO:U=U+1	Begin a limited loop ``\
55	V=#09:GOSUB5	Go FORWARD
56	UNTIL((#@#7802 AND#40)=0 OR(U>99))	Check exit conditions _/
57	DELAY100	Msecs to WAIT
58	Y=@#7800:IFY<#FF GOTO60	Exit if ANY BUMPER contact --v
59	GOTO30	REPEAT this Loop _/
60	REM Loop Exit	
61	V=0:GOSUB5	STOP ALL Motion
9999	REM End of program	End of Tiny-Basic Program

< \_ End of program for TRACK THE TAPE \_ >

NOTE: THIS LISTING GENERATED BY THE "PRINT TBASIC" COMMAND  
(PRINTER SET TO 96 CHARS/LINE)

USE "LTP" COMMAND TO VIEW THIS LISTING ON THE DISPLAY SCREEN

TEST BUMPERS (a Task)

- 1 CALL this robot task line "S" .
  - 2 TURN ON LED number 2 .
  - 3 BEGIN A LOOP
  - 4 EXIT IF BUMPER number 7 is touched
  - 5 TURN ON LED number 3 .
  - 6 REPEAT THIS LOOP
  - 7 TURN OFF LED number 3 .
  - 8 BEGIN A LOOP
  - 9 EXIT IF BUMPER number 8 is touched
  - 10 REPEAT THIS LOOP
  - 11 TURN OFF LED number 2 .
  - 12 MOVE FORWARD
  - 13 BEGIN A LOOP
  - 14 EXIT IF BUMPER number 2 is touched
  - 15 REPEAT THIS LOOP
  - 16 PIVOT ON RIGHT CLOCKWISE
  - 17 BEGIN A LOOP
  - 18 EXIT IF BUMPER number 3 is touched
  - 19 REPEAT THIS LOOP
  - 20 MOVE BACKWARD
  - 21 BEGIN A LOOP
  - 22 EXIT IF BUMPER number 4 is touched
  - 23 REPEAT THIS LOOP
  - 24 PIVOT ON LEFT COUNTERCLOCKWISE
  - 25 BEGIN A LOOP
  - 26 EXIT IF BUMPER number 6 is touched
  - 27 REPEAT THIS LOOP
  - 28 STOP ALL MOTION
  - 29 BEGIN A LOOP
  - 30 EXIT IF FRONT+REAR BUMPER PRESS
  - 31 REPEAT THIS LOOP
  - 32 JUMP to the line called "S" .
- End. (Task size is 168 bytes.)

TEST CYCLE REQUIRES  
BUMPERS TO BE PRESSED  
IN A SPECIFIC SEQUENCE.  
THEN, CYCLE REPEATS AGAIN.

CYCLE:

BUMPER NO.	BEFORE PRESS, LED'S 'ON'	AFTER PRESS MOTION
7	2,3	-
8	2	FORWARD
2	-	PIVOT RCW
3	-	BACKWARD
4	-	PIVOT LCCW
6	-	(STOP) -
1+5	-	-

NOTE: MAY WANT TO PUT ROBOT ON BLOCKS TO DO THIS TEST.

TEST BUMPER MOTIONS (a Task)

- 1 BEGIN A LOOP
  - 2 EXIT IF FRONT+REAR BUMPER PRESS
  - 3 DO ASSIGNED MOTION IF BUMPER PRESSED
  - 4 WAIT for 0.5 second(s).
  - 5 STOP ALL MOTION
  - 6 REPEAT THIS LOOP
  - 7 HONK the horn for 2 second(s).
- End. (Task size is 35 bytes.)

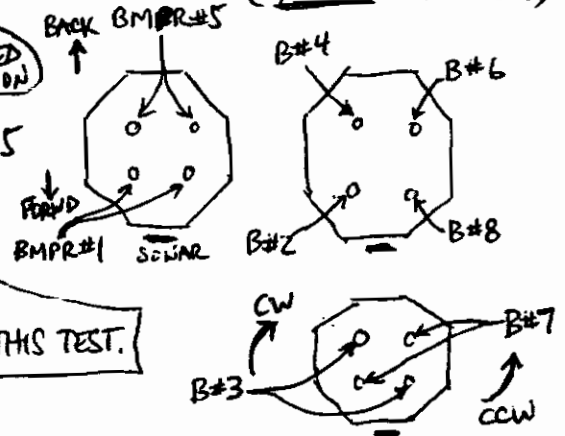
PRESS ANY BUMPER TO INITIATE AN ASSOCIATED MOTION

SIMULTANEOUSLY PRESS BUMPERS 1,5 TO STOP TEST

NOTE: YOU MAY WANT TO PUT ROBOT ON TWO 2"x4"x12" BLOCKS TO DO THIS TEST.

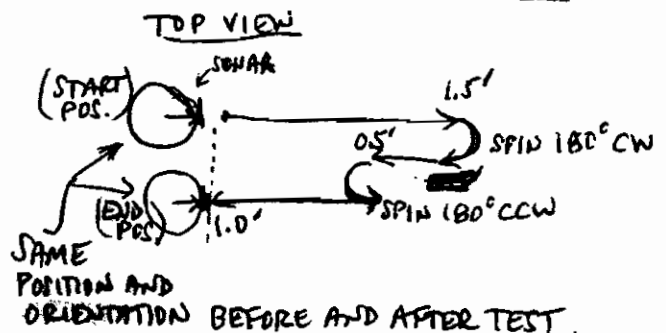
OR 6" OR 2"

LED PANEL TOP VIEW (GREEN MOTOR LED'S)



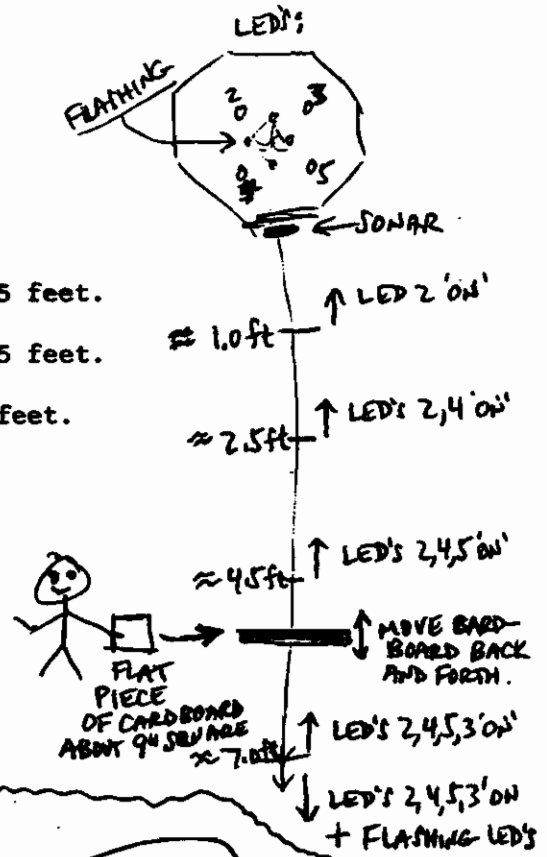
TEST MOTION (a Task)

- 1 HONK the horn for 0.5 second(s).
  - 2 MOVE DISTANCE FORWARD for this many feet 1.5 .
  - 3 SPIN CLOCKWISE for 180 degrees.
  - 4 MOVE DISTANCE FORWARD for this many feet 0.5 .
  - 5 SPIN COUNTERCLOCKWISE for 180 degrees.
  - 6 MOVE DISTANCE BACKWARD for this many feet 1 .
  - 7 HONK the horn for 1 second(s).
  - 8 TURN ON THE FLASHING LIGHTS
- End. (Task size is 71 bytes.)



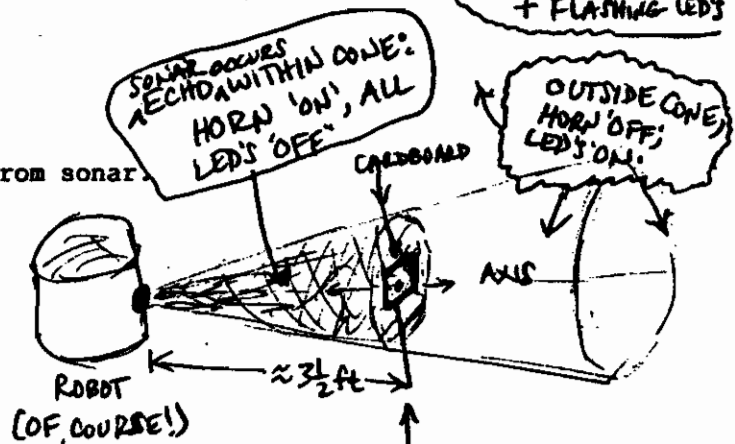
TEST SONAR (a Task)

- 1 LIGHTS ROUTINE
  - 2 WAIT for 1 second(s).
  - 3 BEGIN A LOOP
  - 4 TURN OFF ALL THE LIGHTS
  - 5 TURN ON LED number 2 .
  - 6 BEGIN A LOOP
  - 7 EXIT IF ECHO count is less than (95-350): 95 from sonar.
  - 8 TURN ON LED number 4 .
  - 9 EXIT IF SONAR distance is (LESS, MORE) LESS than (0.8-35) 2.5 feet.
  - 10 TURN ON LED number 5 .
  - 11 EXIT IF SONAR distance is (LESS, MORE) LESS than (0.8-35) 4.5 feet.
  - 12 TURN ON LED number 3 .
  - 13 EXIT IF SONAR distance is (LESS, MORE) LESS than (0.8-35) 7 feet.
  - 14 TURN ON THE FLASHING LIGHTS
  - 15 EXIT THIS LOOP
  - 16 REPEAT THIS LOOP
  - 17 EXIT IF ANY BUMPER TOUCHED
  - 18 WAIT for 0.4 second(s).
  - 19 REPEAT THIS LOOP
  - 20 HONK the horn for 1.5 second(s).
  - 21 TURN OFF ALL THE LIGHTS
  - 22 TURN ON THE FLASHING LIGHTS
- End. (Task size is 150 bytes.)



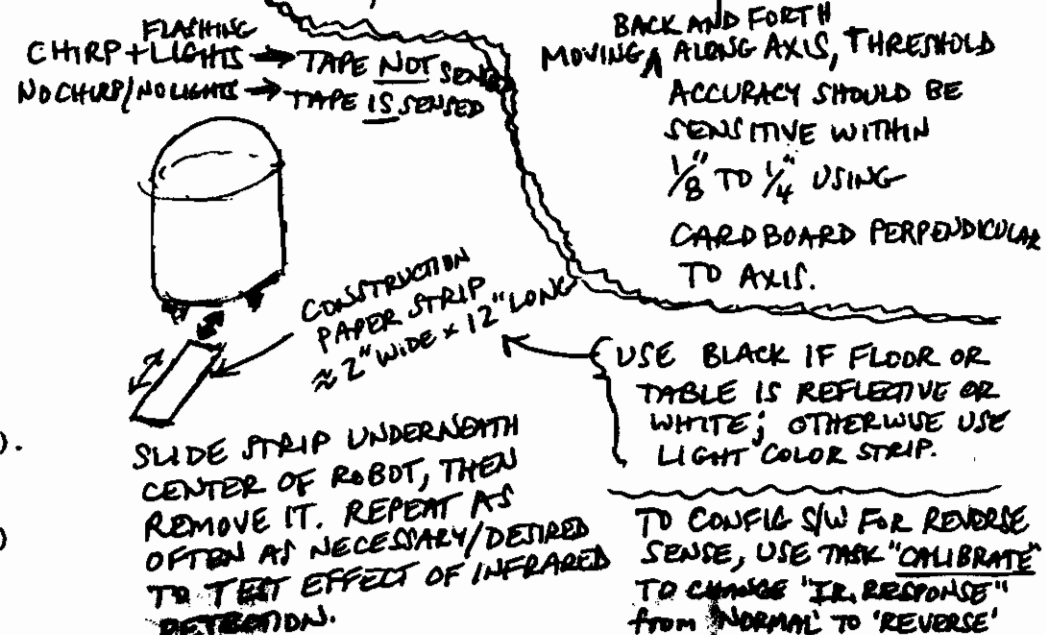
TEST SONAR BOUNDS (a Task)

- 1 PREPARE THE ROBOT
  - 2 CALL this robot task line "S" .
  - 3 BEGIN A LOOP
  - 4 EXIT IF ECHO count is less than (95-350): 145 from sonar.
  - 5 TURN ON ALL THE LIGHTS
  - 6 REPEAT THIS LOOP
  - 7 TURN OFF ALL THE LIGHTS
  - 8 HONK the horn for 0.5 second(s).
  - 9 JUMP to the line called "S" .
- End. (Task size is 53 bytes.)



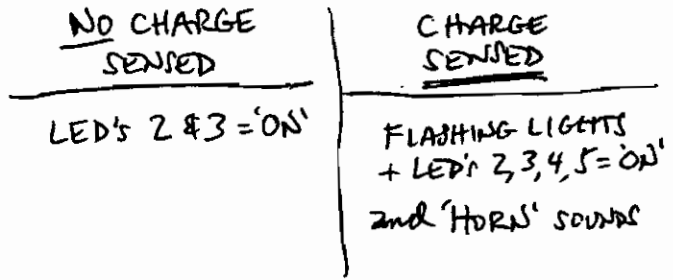
TEST IR (a Task)

- 1 LIGHTS ROUTINE
  - 2 TURN ON THE INFRARED LED
  - 3 BEGIN A LOOP
  - 4 BEGIN A LOOP
  - 5 EXIT IF THE TAPE IS SENSED
  - 6 TURN ON THE FLASHING LIGHTS
  - 7 CHIRP
  - 8 REPEAT THIS LOOP
  - 9 TURN OFF THE FLASHING LIGHTS
  - 10 EXIT IF ANY BUMPER TOUCHED
  - 11 REPEAT THIS LOOP
  - 12 HONK the horn for 2 second(s).
  - 13 TURN OFF ALL THE LIGHTS
  - 14 TURN ON THE FLASHING LIGHTS
- End. (Task size is 50 bytes.)



TEST CHARGER (a Task)

- 1 BEGIN A LOOP
  - 2 BEGIN A LOOP
  - 3 EXIT IF THE CHARGER IS TOUCHED
  - 4 TURN OFF ALL THE LIGHTS
  - 5 TURN ON LED number 2 .
  - 6 TURN ON LED number 3 .
  - 7 REPEAT THIS LOOP
  - 8 TURN ON ALL THE LIGHTS
  - 9 HONK the horn for 0.5 second(s).
  - 10 EXIT IF ANY BUMPER TOUCHED
  - 11 REPEAT THIS LOOP
- End. (Task size is 53 bytes.)



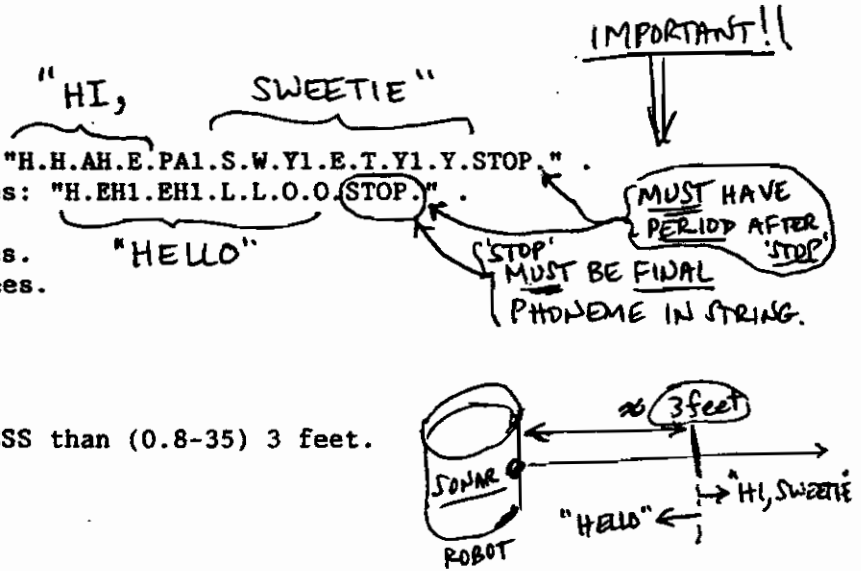
TEST CHIRP (a Task)

- 1 CHIRP
  - 2 WAIT for 1 second(s).
  - 3 WARBLE for 20 cycles using cyclic period of 50 milliseccs.
- End. (Task size is 29 bytes.)

TRY THIS FOR FAMILIARITY W/CHIRP & "WARBLE"

*Sonar and Voice Demo*  
~~TEST VOICE~~ (a Task)

- 1 MAKE A PHRASE called "HI" with phonemes: "H.H.AH.E.PA1.S.W.Y1.E.T.Y1.Y.STOP."
  - 2 MAKE A PHRASE called "HELLO" with phonemes: "H.EH1.EH1.L.L.O.O.STOP."
  - 3 JUMP to the line called "S"
  - 4 TURN THE WRIST (CW,CCW): CW for 45 degrees.
  - 5 TURN THE WRIST (CW,CCW): CCW for 45 degrees.
  - 6 SPEAK the phrase called: "HELLO"
  - 7 CALL this robot task line "S"
  - 8 BEGIN A LOOP
  - 9 WAIT RANDOMLY for up to 6 second(s).
  - 10 EXIT IF SONAR distance is (LESS,MORE) LESS than (0.8-35) 3 feet.
  - 11 SPEAK the phrase called: "HI"
  - 12 JUMP to the line called "S"
  - 13 REPEAT THIS LOOP
  - 14 HONK the horn for 0.5 second(s).
  - 15 SPEAK the phrase called: "HELLO"
  - 16 EXIT IF ANY BUMPER TOUCHED
  - 17 JUMP to the line called "S"
- End. (Task size is 221 bytes.)



*TEST Voice Errors*

~~TESTV~~ (a Task)

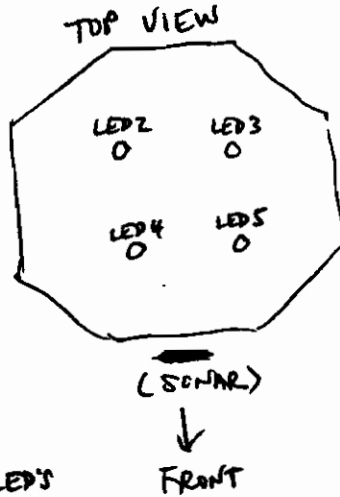
- SPEECH TEST TASK WITH INTENTIONAL PHONEME CODING ERRORS
- NO LONGER REQUIRED (OPTIONAL)
- 1 PREPARE THE ROBOT
  - 2 PREPARE THE VOICE
  - 3 MAKE A PHRASE called "TEST" with phonemes: "EH1'EH1.EH1-EH1'EH1'V'V-V.V'W.W-W'W'E1.E1-E1'E1'Z-Z'Z.Z'UH1.UH3.EH3.EH4.uh1.STOP."
  - 4 SPEAK the phrase called: "TEST"
- End. (Task size is 115 bytes.)
- LOWER CASE NOT ALLOWED!  
NO SUCH PHONEME!  
NO SUCH PHONEME - SHOULD BE UH1 (not UH1)
- PHONEME DELIMITERS:  
INTONATION/INFLECTION
- ERRORS
- Eg, EH1. NO INFLECT.  
EH1-  
EH1'  
EH1^ HIGHEST INFLECT.

VTEST (a Task)

- 1 AVA4
  - 2 SPEAK the phrase called: "ESPANOL"
- End. (Task size is 19 bytes.)

TEST LIGHTS (a Task)

- 1 TURN OFF ALL THE LIGHTS
  - 2 WAIT for 1 second(s).
  - 3 TURN ON LED number 2 .
  - 4 WAIT for 2 second(s).
  - 5 TURN ON LED number 4 .
  - 6 WAIT for 2 second(s).
  - 7 TURN ON LED number 3 .
  - 8 WAIT for 2 second(s).
  - 9 TURN ON LED number 5 .
  - 10 WAIT for 2 second(s).
  - 11 TURN OFF LED number 4 .
  - 12 WAIT for 2 second(s).
  - 13 TURN OFF LED number 2 .
  - 14 WAIT for 2 second(s).
  - 15 TURN OFF LED number 5 .
  - 16 WAIT for 2 second(s).
  - 17 TURN OFF LED number 3 .
  - 18 WAIT for 2 second(s).
  - 19 CHECK MOTOR LIGHTS ← 'GREEN' LED'S
  - 20 CHECK BUMPER LIGHTS ← 'RED' LED'S
  - 21 HONK the horn for 1.5 second(s).
  - 22 TURN ON THE FLASHING LIGHTS
- End. (Task size is 177 bytes.)



TEST ARM (a Task)

- 1 PREPARE THE ROBOT
  - 2 BEGIN A LOOP
  - 3 WAIT FOR ANY BUMPER PRESS and then wait for 0 seconds more.
  - 4 EXIT IF BUMPER number 1 is touched
  - 5 TURN OFF ALL THE LIGHTS
  - 6 TURN ON LED number 4 .
  - 7 TURN ON LED number 5 .
  - 8 MOVE ARM AT THE SHOULDER (UP, DOWN, IN, OUT): OUT for 60 degrees.
  - 9 MOVE THE FOREARM (IN, OUT): OUT for 45 degrees.
  - 10 TURN THE WRIST (CW,CCW): CW for 90 degrees.
  - 11 MOVE THE HAND (OPEN, CLOSE): OPEN for (1-100) 100 %.
  - 12 MOVE ARM AT THE SHOULDER (UP, DOWN, IN, OUT): UP for 20 degrees.
  - 13 HONK the horn for 1 second(s).
  - 14 TURN THE WRIST (CW,CCW): CCW for 90 degrees.
  - 15 MOVE THE HAND (OPEN, CLOSE): CLOSE for (1-100) 100 %.
  - 16 MOVE ARM AT THE SHOULDER (UP, DOWN, IN, OUT): DOWN for 20 degrees.
  - 17 MOVE ARM AT THE SHOULDER (UP, DOWN, IN, OUT): IN for 60 degrees.
  - 18 MOVE THE FOREARM (IN, OUT): IN for 45 degrees.
  - 19 CALL this robot task line "X" .
  - 20 HONK the horn for 1.5 second(s).
  - 21 TURN OFF ALL THE LIGHTS
  - 22 TURN ON THE FLASHING LIGHTS
  - 23 REPEAT THIS LOOP
  - 24 HONK the horn for 3 second(s).
- End. (Task size is 224 bytes.)

{ WILL REPEAT THE COMPLETE ARM CYCLE TEST, UNLESS... }

START EACH ARM MOTION TEST WITH ANY BUMPER, EXCEPT BUMPER #1, WHICH WILL TERMINATE THE TEST.

AVA ~~DEMO~~ (a Task)

← THESE CONTAIN THE "MAKE A PHRASE" COMMANDS.

ALLOWS CONVENIENT GROUPINGS FOR EDIT CHANGES, AND IS ALSO NECESSARY TO STAY WITHIN TASK SIZE BYTE LIMITS.

- 1 AVA1
  - 2 AVA2
  - 3 AVA3
  - 4 AVA4
  - 5 BEGIN A LOOP
  - 6 CALL this robot task line "AGAIN" .
  - 7 BEGIN A LOOP
  - 8 SPEAK the phrase called: "ATTENTION" . ← "SPEAK" EXAMPLE.
  - 9 WAIT for 2 second(s).
  - 0 WAIT RANDOMLY for up to 5 second(s).
  - 1 EXIT IF SONAR distance value is ~~LESS~~ than <sup>4.85 feet</sup> 180 .
  - 2 WAIT for 50 second(s).
  - 3 WAIT RANDOMLY for up to 20 second(s).
  - 4 REPEAT THIS LOOP
  - 5 CALL this robot task line "MOVEIN" .
  - 6 BEGIN A LOOP
  - 7 WAIT TO SAY the phrase called "CLOSER" after waiting 2 second(s).
  - 8 WAIT RANDOMLY for up to 4 second(s).
  - 9 EXIT IF SONAR distance value is ~~LESS~~ than <sup>3.0 feet</sup> 140 .
  - 0 JUMP to the line called "AGAIN" .
  - 1 REPEAT THIS LOOP
  - 2 BEGIN A LOOP
  - 3 WAIT TO SAY the phrase called "THANKS" after waiting 1.5 second(s).
  - 4 WAIT TO SAY the phrase called "BETTER" after waiting 1.5 second(s).
  - 5 EXIT IF SONAR distance value is ~~LESS~~ than <sup>2.35 feet</sup> 125 .
  - 6 JUMP to the line called "MOVEIN" .
  - 7 REPEAT THIS LOOP
  - 8 HONK the horn for 0.5 second(s).
  - 9 WAIT TO SAY the phrase called "HOPE" after waiting 1.7 second(s).
  - 0 WAIT TO SAY the phrase called "FUNHERE" after waiting 2 second(s).
  - 1 WAIT TO SAY the phrase called "HELPKIDS" after waiting 2.5 second(s).
  - 2 WAIT TO SAY the phrase called "ESTEEM" after waiting 0.2 second(s).
  - 3 WAIT TO SAY the phrase called "SUBJECTS" after waiting 1.5 second(s).
  - 4 WAIT TO SAY the phrase called "MODULES1" after waiting 0.5 second(s).
  - 5 WAIT TO SAY the phrase called "MODULES2" after waiting 0.2 second(s).
  - 6 WAIT TO SAY the phrase called "SKILLS" after waiting 2.5 second(s).
  - 7 WAIT TO SAY the phrase called "SKILLS2" after waiting 0.2 second(s).
  - 8 WAIT TO SAY the phrase called "SPEAKANY" after waiting 1.5 second(s).
  - 9 WAIT TO SAY the phrase called "LANGS" after waiting 1 second(s).
  - 0 WAIT TO SAY the phrase called "SPANISH" after waiting 2 second(s).
  - 1 WAIT TO SAY the phrase called "ESPANOL" after waiting 1 second(s).
  - 2 WAIT TO SAY the phrase called "MAKEDAY" after waiting 2 second(s).
  - 3 WAIT TO SAY the phrase called "USEME" after waiting 1 second(s).
  - 4 WAIT TO SAY the phrase called "TALKTO" after waiting 2 second(s).
  - 5 WAIT TO SAY the phrase called "TALKTO2" after waiting 0 second(s).
  - 6 WAIT TO SAY the phrase called "HAVEFUN" after waiting 1.6 second(s).
  - 7 WAIT TO SAY the phrase called "COMEBACK" after waiting 1.3 second(s).
  - 8 HONK the horn for 0.5 second(s).
  - 9 REPEAT THIS LOOP
- nd. (Task size is 640 bytes.)  
 ≈ 700
- 49 WAIT 30 SEC  
 50 WAIT RANDOMLY FOR 40 SECS

NO WAIT  
 THESE ARE SPOKEN BACK TO BACK.  
 (COULD HAVE USED "SPEAK" COMMAND FOR "TALKTO2")

AVA1 (a Task)

- 1 MAKE A PHRASE called "ATTENTION" with phonemes: "M.A1.I3.Y.PAO.AH1.EH3.I3.Y.PAO.P.L.E1.AY.Z.PAO.H.AE1.EH3.V.PAO.Y1.IU.O2.O2.R.PAO.UH1.T.EH1-EH3-N.SH.UH3.N.STOP."
  - 2 MAKE A PHRASE called "BETTER" with phonemes: "D.THV.AE1-EH3.T.Z.PAO.B.EH1-EH3.T.ER.STOP."
  - 3 MAKE A PHRASE called "CLOSER" with phonemes: "P.L.E1.AY.Z.PAO.K.UH1.UH3.M.PAO.UH2-UH1.L.I1-I3-T.UH3.L.PAO.K.L.O1-U1-S.S.ER.STOP."
  - 4 MAKE A PHRASE called "THANKS" with phonemes: "THV.AE1-EH3-NG.K.PAO.Y1.IU.U1-U1.STOP."
  - 5 MAKE A PHRASE called "HAVEFUN" with phonemes: "H.AE1-EH3.V.PAO.UH1.UH2.PAO.F.UH1-UH2-N.PAO.D.A1-I3.Y.STOP."
  - 6 MAKE A PHRASE called "COMEBACK" with phonemes: "Y.UH3-UH1.PAO.AW-L.PAO.K.UH1-UH3-M.PAO.B.AE1-EH3.K.PA1.Y.UH3.UH1.PAO.H.E1-I2-R.STOP."
  - 7 MAKE A PHRASE called "HOPE" with phonemes: "AH1-EH3.I3.Y.PAO.H.O1-U1.P.PAO.Y1.IU.U1.U1.PAO.AH1.R.PAO.EH1.EH3.N.D.J.O1-UH3-I3.AY.I3.NG.PAO.D.THV.UH1.UH3.PAO.K.AH1-UH3.N.F.ER.UH1.EH3.N.S.PAO.AE1.EH3.N.D.PAO.AH1.R.PAO.H.AE1-EH3-V.I1.I3.NG.PAO.UH1.UH2.PAO.T.ER-R.I1-I3-F.I3.K.PAO.D.A1-I3-Y.STOP."
  - 8 MAKE A PHRASE called "FUNHERE" with phonemes: "I1.I3.T.S.PAO.F.UH1-UH2-N.PAO.T.U1-U1.PAO.B.E1-Y.PAO.H.AY-I1.R.PAO.I1.I3.N.PAO.T.EH1-EH3-N.N.UH1.UH2.PAO.S.E1-Y.STOP."
  - 9 MAKE A PHRASE called "MAKEDAY" with phonemes: "M.A1-AY-Y.K.PAO.Y1.IU.O1-O2-R.PAO.D.A1.I3.Y.STOP."
- End.

(Task size is 990 bytes.)

NOTE: TASK SIZE LESS THAN 1000 BYTES!!



AVA3 (a Task)

1 MAKE A PHRASE called "SKILLS2" with phonemes: "AE1.EH3.N.D.PA1.T.AH1-EH3-Y.P.I1.NG.PAO.S.K.I1-I3.L.L.Z.STOP."

2 MAKE A PHRASE called "MODULES1" with phonemes: "I1.I3.N.K.L.U1.U1.D.I1.NG.PA1.E1.Y.L.EH1.EH3.C.T.R.AH1-UH3-N.I1.K.S.PA1.PA1.M.AH1-EH3-Y.K.R.O1.P.R.AH1-UH3.S.EH1.EH3.S.O1.R.Z.PA1.PA1.S.EH1-E13-N.S.O1.R.Z.PA1.PA1.M.EH1.EH3.K.AE1-EH3-N.I1.K.S.PA1.PA1.R.O1.U1.B.AH1-UH3-T.I1.K.S.STOP."

3 MAKE A PHRASE called "MODULES2" with phonemes: "AE1'EH3-N.D.PA1.I1.I3.N.T.EH1-EH3-L.I1.D.J.J.IH3.EH3.N.T.PAO.M.UH2.EH3.SH.E1-Y-N.Z.STOP."

4 MAKE A PHRASE called "SPEAKANY" with phonemes: "D.THV.A1.AY.Y.PAO.K.AE1.EH1.N.PAO.M.A1-AY-Y.K.M.E1.Y.PAO.S.P.E1-AY-K.PAO.EH2-EH2.N.Y.PAO.L.AE1-EH3-NG.G.W.I1.I3.D.J.J.STOP."

5 MAKE A PHRASE called "TALKTO" with phonemes: "T.AW-K.PAO.T.U1.U1.M.AH1-EH3.I3.Y.PAO.K.AH1-UH1.L.E1.Y.G.PAO.T.U1.U1.PAO.L.ER.R.N.PAO.M.O2-O2-O2.R.UH1.B.UH2.AH2-U1.T.Y1.IU.U1-U1-Z.I1.NG.PAO.D.THV.E1.Y.AH1-R.PAO.B.E1-Y.PAO.F.AH1-EH3-Y.V.PAO.EH1-EH3-K.S.PAO.L.U1-ER.R.N.I1.Y.NG.STOP."

6 MAKE A PHRASE called "TALKTO2" with phonemes: "S.I2-S.S.T.UH1.UH3.M.PA1.I1.I3.N.PAO.Y1.IU.O2-O2-R.PAO.K.L.AE1-EH3'S.S.R.U1.U1.M.Z.STOP."

End. (Task size is 937 bytes.)

NOTE: SIZE < 1000 BYTE MAXIMUM!

VA4 (a Task)

1 MAKE A PHRASE called "LANGS" with phonemes: "I1-I3-NG.L.I1.SH.PA1.PA1.S.P.AE1-EH3-N.I1.SH.PA1.PA1.F.R.EH1-EH2-N.D.T.CH.PA1.PA1.D.J.J.ER-M.UH1.N.PA1.PA1.R.UH1-UH2-SH.UH1.N.PA1.PA1.T.CH.AH1-A.I3.Y.N.E1-Y.S.PA1.PA1.AE1'EH3-N.D.PA1.PA1.D.J.J.AE1-EH3-P.UH1.EH3.N.E1-Y.Z.STOP."

2 MAKE A PHRASE called "ISPEAK" with phonemes: "AH1-EH3.I3.Y.PAO.K.AE1.EH3.N.PAO.B.E1.Y.PAO.M.A1-AY-Y.D.PA1.T.U1.U1.PAO.S.P.E1.AY.K.EH2-EH2.N.Y.PAO.L.AE1-EH3-NG.G.W.I1-D.J.STOP."

3 MAKE A PHRASE called "SPANISH" with phonemes: "F.O1.R.PAO.EH1.EH3.K.S.PAO.AE1-EH3-M.P.UH3.L.P.I1.PA1.H.AY-I1-R.Z.PAO.A1.AY.Y.PAO.B.I1-I3-T.PAO.UH1.UH3.V.M.AH1-EH3.I3.Y.PAO.S.P.AE1-EH3-N.I1.S.L.STOP."

4 MAKE A PHRASE called "ESPANOL" with phonemes: "B.W.EH1-AY-N.O.S.PAO.D.E1-AH1.S.PA1.AH.M.E1-Y.O.S.PA1.PA1.PA1.Y1.O-PAO.S.EH3.AY.PAO.AH-B.L.AH1.PA1.EH1.EH3.S.P.AH1.N.Y1-O1-O2-L.STOP."

End. (Task size is 710 bytes.)

SIZE < 1000 BYTES.

VA2 (a Task)

1 MAKE A PHRASE called "HELPKIDS" with phonemes: "AH1.EH1.I3.Y.PA1.K.AE1.EH3.N.PAO.H.EH1-EH3-L.PAO.Y1.IU-O2'O2-R.PAO.S.T.U1-U1.D.EH1.EH3.N.T.S.PAO.D.E1.Y.V.EH2-L.AH1.UH3.P.PA1.M.O2-O2-O2.R.AO.E1.Y.F.EH1-EH3-K.T.I1.I3.V.PA1.P.R.AH1-UH3.B.L.EH1.EH3.M.S.AH1-UH3.L.V.I1.NG.PAO.S.K.I1-I3.L.L.Z.STOP."

2 MAKE A PHRASE called "ESTEEM" with phonemes: "AE1-EH3-N.D.PAO.G.A1-AY-Y.N.PAO.H.AH1-EH3'Y-ER.AO.S.EH1.EH3.L.F.EH1.EH3.S.T.E1-Y.M.STOP."

3 MAKE A PHRASE called "SKILLS" with phonemes: "Y1.IU.O2.O2.R.PAO.K.I1.I3.D.Z.PAO.W.I1.I3.L.I1.3.M.P.R.U1.U1.V.PAO.D.THV.EH2.EH2.R.PAO.L.AE1-EH3-NG.G.W.I1.I3.D.J.J.STOP."

4 MAKE A PHRASE called "SUBJECTS" with phonemes: "D.THV.A1.AY.Y.PAO.W.I1.I3.L.L.ER-R.N.UH1.B.UH.AH2.U1.T.PA1.M.AE-EH3-THV.PA1.PA1.S.AH1-I3-AY.EH1.EH3.N.S.S.PA1.PAO.S.I1-I3-S.S.T.EH1.EH3.M.M.PA1.PA1.AE1.EH3.N.D.PA1.T.EH3.K.N.AH1-L.EH3.D.J.E1.Y.STOP."

5 MAKE A PHRASE called "USEME" with phonemes: "Y1.IU.U1.U1.Z.PAO.M.E1.E.PAO.T.U1.U1.PAO.M.O1-U1.T.I1.I3.V.A1-AY-Y.T.PAO.Y1.IU-O2-O2-R.PAO.K.I1-I3.D.Z.PA1.AE1.EH3.N.D.PAO.B.R.I3.NG.PAO.EH1.EH3.K.S.AH1-EH3'Y-T.M.EH3.EH1.N.T.PAO.T.U1.U1.PAO.D.THV.EH3'A2-EH3-R.PAO.S.K.U.U1.L.PAO.D.A1.I3.Y.Z.STOP."

End. (Task size is 995 bytes.)

SIZE < 1000 BYTES

NOTE: PHONEME STRING LENGTH LESS THAN 255 CHARACTER MAX LIMIT.

