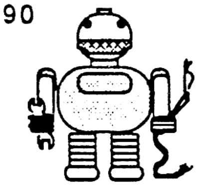
July1990

ROBOT BUILDER



The official publication of the Robotics Society of Southern California
P.O. Box 3227, Seal Beach CA 90740, Meetings the 1st Tuesday @ 7:00 PM at MTI College

UPCOMING EVENT CALENDAR:

June 30:	RSSC Robot Project Workshop, *The Robot Company*
July 3:	RSSC July Meeting, MTI College: Topic - TBD
July 7?:	RSSC Robot Project Workshop, *The Robot Company*
July 11-22	Orange County Fair
July 29:	Computer Swap Meet at Advanced Computer Products
August 7:	RSSC August Meeting, MTI College: Topic - TBD
August 11:	RSSC Robot Project Workshop, *The Robot Company*
September 4:	RSSC September Meeting, MTI College: Topic - TBD
September 8:	RSSC Robot Project Workshop, *The Robot Company*

JUNE 5th RSSC MEETING

We had another good turnout at our June 5th meeting with about 25 people in attendance. Our featured speaker for the evening, Andy Coyle from ***Globe Motors***, was very informative. He passed out a catalog on their line of small dc motors.

During his presentation, he answered a variety of questions about motors from the group.

Joe McCord led a discussion on the Society's participation at the Orange County Fair. He prepared a sign-up schedule for manning our booth and circulated it for members to fill in. There is still some times that need to be filled, so check your schedule for free time and help out the Society!

Tom Carroll presented a videotape from the Artificial Intelligence Laboratory at MIT on several robots that students have constructed.

Jerry Burton distributed lists of extra robot parts and components available from SynPet, including various parts and assemblies used in their SP2 personal robot. The membership voted to purchase several components for the Society's Robot project.

JUNE 16th RSSC ROBOT PROJECT WORKSHOP

On June 16th, the RSSC had another robot project workshop at ***The Robot Company*** shop in Costa Mesa with about 10 members present.

The motors donated by ***Globe Motors*** for the drive wheels have been received.

We discussed the Society's purchase of a high performance controller (HPC) board, motor drivers, dc-dc converter, the TI speech board, and a Polaroid sensor harness for the RSSC robot.

Now that we have hardware to work with, it was agreed to try to have workshops approximately twice a month. Therefore, our next workshop will be held on June 30th at 10:00 a.m., again at ***The Robot Company***, 881 West 18th Street, Costa Mesa.

THE ROBOTEER

(Editors note: Jerry Burton has answered my plea for contributed material with the following two fine articles.)

*-Turning **DUMBOT** in to **SMARTBOT***

Scott and Tom have been asking for input to the newsletter, so I decided to start submitting some thoughts under the title 'The Roboteer.' If a Roboticist is someone who designs and builds robots, then a Roboteer is someone who makes them do something. Just as puppeteers make nonliving puppets appear to have life, Roboteers make robots appear to have intelligence.

In this series of articles, I will attempt to share some of the frustration and joy that I am encountering as I try to turn my DUMBOT in to a SMARTBOT.

First, a little history on the name "DUMBOT." I contracted with SynPet to provide the navigation software for their Newton Robot using C++. They provided a robot and I started to learn C++ (which is an extended version of C) and began to formulate a design based on the capabilities of Newton.

As most of you know, Newton uses the TI speech recognition subsystem and gets most of its input from voice commands. You must train it to recognize your voice. I found out very quickly that it also trained you. You have to learn to speak the same way each time you talk to the robot or it just sits there and does nothing. When you train it, you get visual feedback as to whether the robot "heard" you, and how well it understood you (it gives a confidence or correlation factor). Negative values are bad and positive values are good, the larger the better. I found that it didn't recognize the way I said "Newton" very consistently so I renamed it "Fred."

After many hours and a fairly high level of frustration, I finally got to the point where it would follow my commands most of the time. I'm a Roboteer and live and breathe robots, but found the time investment far exceeded the payoff. In the training, mode I could "see" how well I was doing, whereas when in operational mode it would only give a response when it actually recognized what I said. It would have been easier if the robot gave some response when it didn't understand so I could vary my pitch, speed, or volume to get a match (this is a flaw in the TI subsystem that needs to be looked in to). The speech training process was obviously designed by engineers for engineers, not real people! Once the Society Robot is running, you will all get to experience the "fun" of teaching it to understand you and you to understand it. Perhaps one of you will take on the challenge of making the interface easier to use.

Most people just won't mess with a machine unless it works with very little effort on their part. If they have to work too long and hard to get results, they become agitated and toss it out or put it in the closet (how many of the 30 million plus PCs in the U.S. today do you think are actually being used for meaningful work ??).

My girlfriend and partner, a very pragmatic German lady, saw how much time I was putting in to this new "toy" and got interested in what it could do, so she asked, "Can it vacuum my house or do the dishes?"

I said, "Not yet," and then patiently explained all of its capabilities and what it "could" do when properly programmed. She listened intently and finally said, "That's really good, but what does it do now?" I hemmed and hawed and tried to convince her that it really did do "something" - after all, it could recognize my voice and tell stories and .... well, lots of stuff.

She finally said, "It's just another dumb robot like all the other ones you already have, that don't do anything meaningful. Why don't you call him "DUMBOT." At first, I was really mad that she didn't see the potential of this new robot and was quite defensive about the name "DUMBOT."

After much reflection, I had to admit that it was really pretty dumb, even though it was infinitely more capable than the RB5X, Hero Jr, Hubot, and TOPO that I already had. Even though I considered Newton far superior to these other 'toys' (masquerading as real robots), I could not avoid the conclusion that normal people measured a robot by what it could do, rather than what inherent capabilities it may possess, and what it 'might' do in the future.

We Roboteers sometimes get so immersed in the details of our craft that when asked by our friends "What can it do?", and we tell them and they aren't impressed, we get miffed or dismiss them as not being smart enough to understand the grand and wonderful things we envision the robot will someday do! We know that to get a machine to do even simple things is incredibly difficult and are willing to work on the "problem" day and night to make even modest progress. Can we expect non-Roboteers to appreciate the blood, sweat, and tears required to do even the most mundane task?

I have therefore come to call one of the most advanced robots available today "DUMBOT" to remind me that until I can provide the appropriate software "smarts" it is nothing more than a really expensive PC that can talk and move around bumping into things. The major problem with personal robots is not hardware; Newton is quite capable from a hardware viewpoint. The real problem is that there is no software available to allow the average person to get the robot to do anything useful.

With the group purchase we just made plus the robots that Joe and I have, I figure we will probably have about six or seven Newton class robots available for development within the next year. Hopefully, by our joint development efforts and the exchange of software and hardware ideas we can reach a critical mass and turn them all into SMARTBOTs.

Jerry Burton

ROBOT ARCHITECTURE FOR THE SOCIETY'S ROBOT

With the decision of the members present at last month's meeting to buy a high performance controller (HPC), motor control board, and power supply from SynPet's surplus inventory, the architecture of our robot has changed and we have significantly shortened the time needed to get a working robot. We no longer need the STD bus; everything will be oriented to the AT bus donated by Don Golding. I am also happy to announce that we were able to negotiate a TI speech subsystem for the Society as part of the group purchase we made, so our robot will have speech recognition and synthesis for those of you who wish to play with voice control. We were able to get the HPC, motor board, power supply, TI speech, and a sonar subsystem at a cost of only $300 to the Society (the speech system alone retails for $1200!).

The HPC is based on the HPC CPU by National Semiconductor and provides control for four separate sonar circuits, a passive infrared receiver, three infrared remote control receivers, 30 individual lights, pulse width modulation for three separate motors, optical encoders for three separate motors, and inputs from temperature, voltage, tilt, and smoke sensors, plus one relay output. It certainly has enough control functions to meet our needs for quite awhile.

The HPC board gets its commands directly through the bus via a set of assembly language modules provided by SynPet. They can be accessed via C, C++, or even Forth (this should delight the Forthians among us). Diagram A shows the overall architecture we now will have.

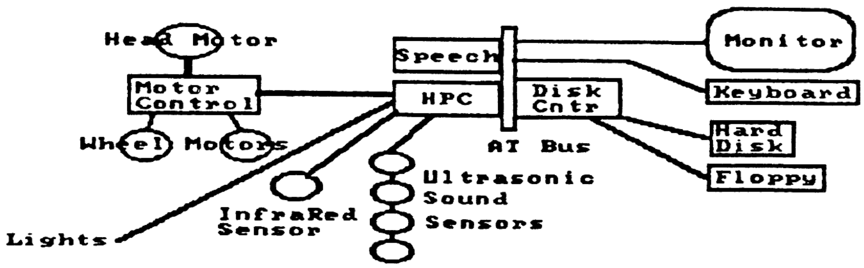
The HPC can control three motors independently, i.e. two drive motors and a head motor. The motor control board is capable of supplying 12 amps, which should be sufficient to drive the Globe motors that were donated. The HPC also has four independent sonar circuits as well inputs for infrared, smoke detection, temperature, voltage, and tilt.

SynPet will provide the HPC source code as well as the HPC development software so we can upgrade or modify the way the HPC works as much as we want. All we have to do is burn a new 256K ROM to test any new code. It would be helpful to have a few interested members focus their attention on the HPC processor so we have expertise in the processor, and can interface new sensors in the future or even change the way it handles the existing sensors. If anyone is interested in pursuing this, contact me at the next meeting and I'll make sure you get all the documentation necessary.

The existing HPC functions are broken into categories and options as follows:

Category		Option	Description
Move	0	0	Rotate and then move forward/backward
		1	Move dd.d. feet in an arc
		2	Move dd degrees in an arc
		3	Follow the wall (not implemented)
Self-test	1	0	Short
		1	Regular
		2	Extensive
Lights	2	0	Turn on/off a specific light
		1	Turn on/off all lights
		2	Turn on/off eye lights
Control	3	0	Pause
		1	Continue
		2	Sleep
		3	Wake-up
		4	No Operation
		5	Obstacle & dropoff avoidance on/off
		6	Sonar on/off
		7	Remote control on/off
		8	Set remote control address
		9	Remote control duplicate filter on/off
		10	Receive remote keystroke
		11	Sonar obstacle/dropoff avoidance on/off for each of the four sonars individually.
		12	Set sonar distance for each of four sonars
		13	Set IR reference level
		14	Auto head calibration on/off
Read-Internal	4	0	Temperature
		1	Battery voltage percentage
		2	Tilt (yes/no)
		3	Smoke sensor
Read-External	5	0	Head sonar distance
		1	Any sonar distance
		2	IR change
		3	Scan head seeking IR change
		4	Current head position
		5	Light intensity
		6	IR status
Move-Head	6	0	Move to specified angle
		1	Home
Beacon	7	0	Follow
		1	Turn to face beacon
Light Show	8	0	Turn on/off pattern #n
		1	Set order of patterns
		2	Turn on/off

Most of these functions are already operational and can be used immediately; others will require some work (the beacon stuff is in the prototype stage). Now is the time to stake out territory of interest so that we don't duplicate our efforts too much.



Robot Overall Architecture
Diagram A

Personally, I am concentrating on the navigation software and have a preliminary version already running. I can give the robot a series of X,Y points to go to and it will go to them one after another. If it detects an obstacle along its path, it goes into local avoidance mode and tries to work its way around the obstacle and ultimately reach its objective. It's quite rudimentary, but works reasonably well. I can hardly wait to see it running on the Society Robot - maybe by OC Fair time?

We need to improve the sonar processing from merely getting the rather inaccurate distance measurements the Polaroid sensors provide - hopefully some of you hardware types will put on your thinking caps and come up with something really good.

We need a few more things to get the robot to an operational stage. As a minimum, we need a monitor card (preferably EGA but even a mono to start), a floppy controller and disk, and ideally a hard disk and controller. If any of you have a monitor and or keyboard to donate or loan it would be appreciated. Scrounge through your old equipment boxes and see what you might be willing to donate. Bring it to the next meeting (regular or lab) and we will add it to the growing collection of stuff on the Society's shelf down at Joe's place.

Well enough of this. I have to go work on "DUMBOT" some more - see you at the next meeting.

Jerry Burton

UPCOMING RSSC EVENTS

For our July 3rd meeting, we are still making arrangements to have a company representative provide a presentation.

On July 11th through 22nd, the RSSC is planning to have a booth at the Orange County Fair. The Society still needs individuals to help man the booth.

It is tentatively planned to go back to the ACP swap meet on July 29th to continue attracting new members and individuals in robotics. We hope to have the Society's robot ready to show and in addition have more of our members' robot projects.

JULY 3RD MEETING AGENDA

1) Business agenda
   a. Plans for staffing OC Fair booth
   b. Contest announcement for the RSSC T-shirt design. The winner will be awarded the donated multimeter
2) Presentation of RSSC robot design and construction
3) Presentation?
4) RAM (Random Access Meeting) - bring something of interest to share with the membership!

Again, I hope to see the entire membership there, along with any interested individuals or business representatives! Pass the word around about the RSSC!!

Scott MacGillivray, Editor

The following was found in *NASA Tech Briefs*, and may be of interest to the membership.

Stabilizing Wheels for Rover Vehicle

Extra wheels would prevent tipping, aid in climbing banks, and help in righting after overturning.

NASA's Jet Propulsion Laboratory, Pasadena, California

A proposed articulated, normally-four-wheeled vehicle would hold an extra pair of wheels in reserve. The extra wheels could be deployed to lengthen the wheelbase on slopes, thereby making the vehicle more stable. The extra wheels could also be deployed to aid the vehicle in negotiating a ledge or to right the vehicle if it has turned upside down. The concept promises to make remotely controlled vehicles more stable and maneuverable in such applications as firefighting, handling hazardous materials, and carrying out operations in dangerous locations.

The extra wheels would be drive wheels mounted on arms so that they can pivot on the axis of the forward drive wheels. Both the extra wheels and the arms could be driven by chains, hydraulic motors, or electric motors.

During ordinary travel, the extra wheels would be kept out of the way and idle. During a dangerous turn on a steep slope, however, the extra wheels would be deployed like outriggers to make contact with the ground (see Figure 1). The deployed wheels would increase the horizontal distance between the center of gravity and the point of contact farthest downhill, thereby enhancing stability.

The extra wheels could also be deployed when the forward main wheels have encountered a steep bank and are stalled (see Figure 2). The arms of the extra wheels would be rotated upward and forward to bring the extra wheels to the top of the bank. As soon as the extra wheels would make contact with the terrace, they would start driving, along with the main wheels, lifting the front of the vehicle to the terrace. The arms of the extra wheels would rotate rearward until the extra wheels make contact with the ledge behind the front wheels. They would start driving again to help the rear wheels boost the entire vehicle to the terrace. When the vehicle has fully mounted the bank, the extra wheels would be returned to their original position. The main wheels would once again drive the vehicle forward.

When the vehicle has overturned, the arms would be rotated to bring the extra wheels to the front of the vehicle. At the same time, the forward main wheels would be rotated on their arms toward the rear. The extra wheels would drive so as to raise the front of the vehicle and continue to drive and rotate on their arms until they make contact with the rear main wheels. At that point, the extra wheels would lock and the rear main wheels would drive, climbing the extra wheels. The rear wheels would rotate forward on their arms until they would climb on the forward main wheels, turning the vehicle over. The extra arms and wheels would brace the vehicle during this maneuver. Finally, the arms of the main wheels would return the main wheels to their normal position, and the extra arms and wheels would return to their original position.

This work was done by Earl R. Collins, Jr., of Caltech for **NASA's Jet Propulsion Laboratory**. *For further information, Circle 75 on the TSP Request Card.*

This invention is owned by NASA, and a patent application has been filed. Inquiries concerning nonexclusive or exclusive license for its commercial development should be addressed to the Patent Counsel, NASA Resident Office-JPL [see page 16]. Refer to NPO-17495

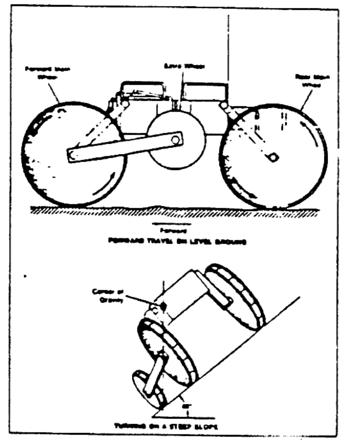
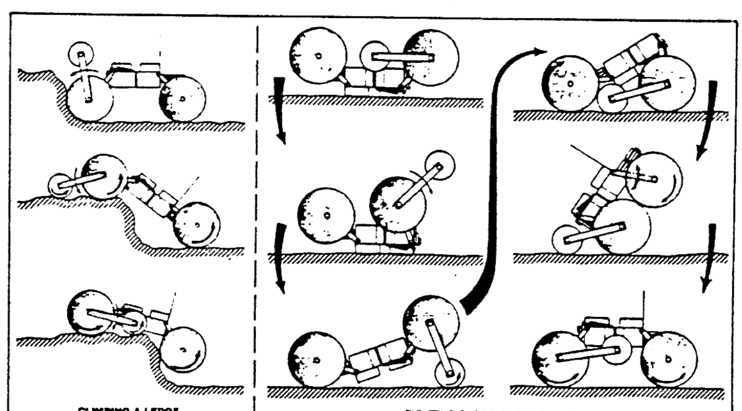


Figure 1. **When the Vehicle Travels** on a relatively-flat, level surface, the extra wheels are nested out of the way. On a steep slope, arms move the extra wheels downward so that they prevent the vehicle from tipping over. In the configuration shown here, the extra wheels provide support on slopes up to 40°.

Figure 2. In this **Ledge-Climbing Sequence** (above), the extra wheels move forward, pull up the front main wheels, and rotate to help the rear main wheels climb. To right the upside-down vehicle (below), the extra wheels move forward, raise the front of the vehicle, and lock, allowing the rear wheels to climb forward. The rear main wheels continue their movement until they restore the vehicle to its upright orientation and the extra wheels have been stowed in their out-of-service position.



The following was taken from the *Olympic Robot Building Manual* prepared by the MIT Artificial Intelligence Laboratory, and may be of interest to the membership.

The Zemco Flux-Gate Digital Compass

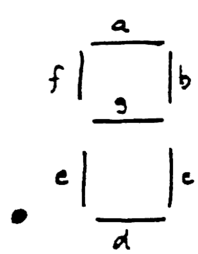
Peter Ning

The Zemco Digital Compass will provide a reasonably accurate measure of orientation with respect to the earth's magnetic field. You can extract 4-bits of resolution from the compass. There are basically two ways of interfacing the compass with your main CPU, either digitally or analog; the former requires no hardware modification to the compass electronics but new data may take up to 3 second to stabilize. The latter method provides data on the order of 1/10 second but requires another level of data translation from analog to digital. Here are all the hints you need to use this compass:

1. <u>Digital Interface Method</u>

There is essentially one chip in the compass electronics that you need to be concerned with, the microprocessor (COP421-MLA). This processor sends out three signals D0 (Pin 24), SK (Pin 16), and SD (Pin 15). D0 is low when SK and SD signals are valid. The SK signal is used to clock in serial data on the SD line on the RISING-EDGE.

What is this data? - The first four bits are starting bits and should be ignored. Then come three 8-bit numbers, each representing a digit of degrees (0-360) for the direction. The last 8-bit number is always zero and should be ignored. Unfortunately, the 8-bit numbers are encoded in 7-segment led display format.



Number	Bits: . c b a f g e d
0	0 1 1 1 1 0 1 1
1	0 1 1 0 0 0 0 0
2	0 0 1 1 0 1 1 1
3	0 1 1 1 0 1 0 1
4	0 1 1 0 1 1 0 0
5	0 1 0 1 1 1 0 1
6	0 1 0 1 1 1 1 1
7	0 1 1 1 0 0 0 0
8	0 1 1 1 1 1 1 1
9	0 1 1 1 1 1 0 1

Scope these signals on an oscilloscope or a logic analyzer to verify the data bits as you move the compass around. For both methods be sure to have proper GROUND signals between the interface of the compass electronics and your CPU circuitry. Remember that the compass is NOT foolproof; there will be occassional spurious readings when the compass nears metal objects, as with most needle-based hand held compasses.

2. <u>The Analog Method</u>

You will not have to deal with the COP421-MLA processor in this case. All you need from the compass are the analog outputs X (Pin U1-8, U1 is the CA3403 quad op-amp) and Y (Pin U1-14). As you rotate the compass 360 degrees you should see these two outputs produce sine waves with a phase shift of 90 degrees.