Designing and Building a Multiple-Transducer Sonar System

Part 2 of 3: Construction and Interfacing

Rick Moll and David J. Musliner

In the first part of this article [Musliner and Moll, 1995] we discussed why sonar is a popular range-sensing method, and showed the tradeoffs involved in the design of multi-transducer sonar systems. We also discussed important crosstalk issues, presenting methods used to deal with the problems encountered. The original plan was to do this article in two parts; however due to the amount of material we have to present, the article has expanded to a total of three installments.

In this second part of the article, we will be presenting the construction techniques needed to actually build the hardware for a multi-transducer sonar system. This article demonstrates how to construct multi-transducer sonar systems around the Polaroid 6500 Series drivers. Detailed schematics are presented so that you can build your own multi-transducer sonar hardware. In the upcoming part three, the series will conclude with software to drive the sonar hardware as well as a detailed discussion of an acoustical crosstalk elimination technique (EERUF), including software implementing the technique.

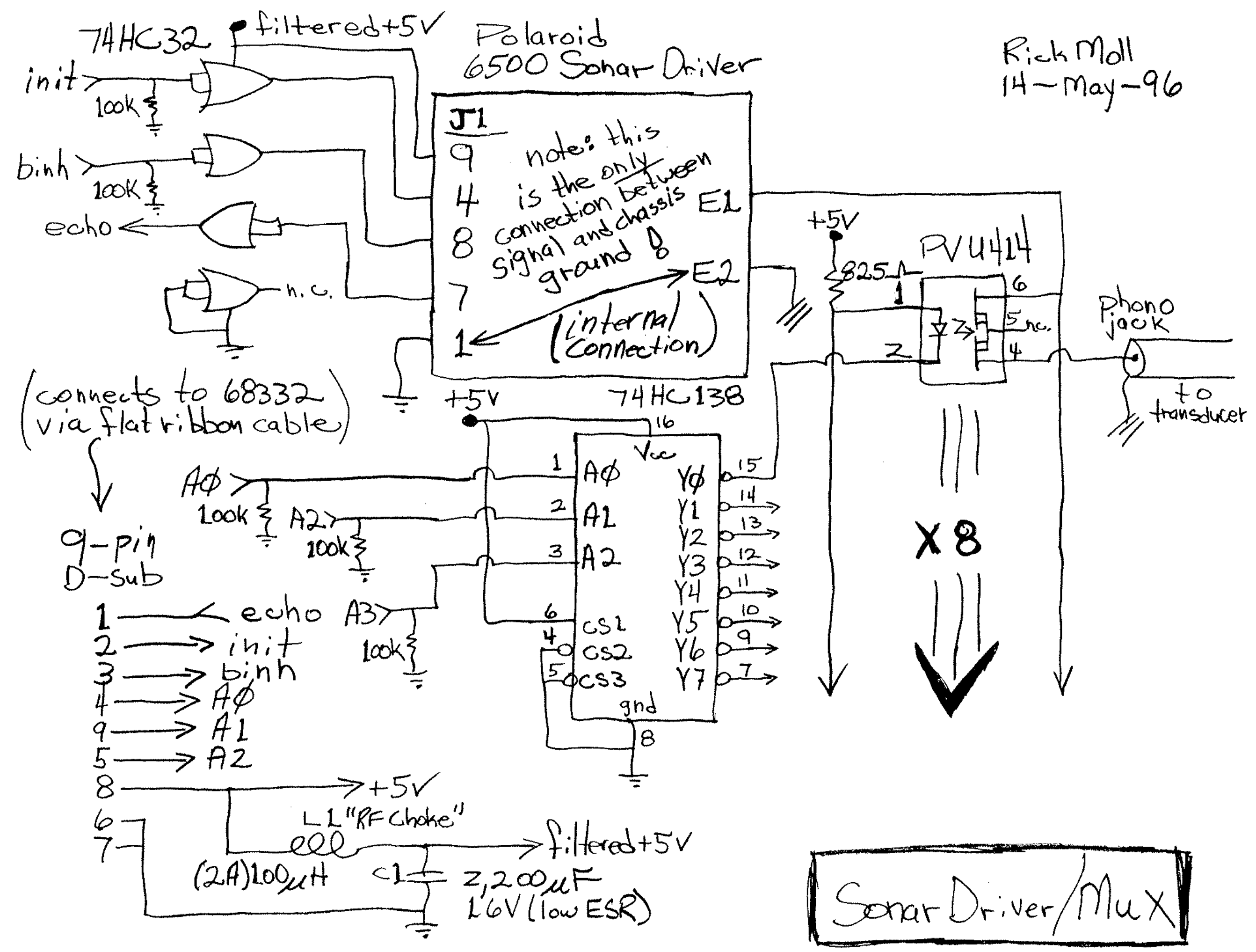
Hardware

The sonar system presented in this article is built from Polaroid transducers and 6500 Series drivers. These can be purchased directly from Polaroid. The OEM kit (part #606783) contains two transducers, and two drivers, as well as datasheets and some technical papers. The transducers (part #616341) are also available separately in multiples of 10 (see the resource section at the end of this article).

Powering the Driver

As we discussed last time, one of the most fundamental problems encountered when building sonar systems is dealing with the crosstalk (both acoustical and electrical). The schematic in Figure 1 presents an

Figure 1: *Sonar Driver/Mux Schematic*



approach that can be used to both filter the electrical noise generated by Polaroid driver, as well as provide the electrical energy required by the driver when it pulses a transducer. A combination filter/energy storage bank is formed by L1 and C1. This is perhaps not the most elegant design. However this solution does provide power for the driver without undue demands being placed on the rest of the system. Without some kind of filter, the 6500 Series driver can easily throw a microprocessor sharing its power bus into convulsions. An important secondary advantage to this filter/energy storage bank is that it works both ways–it also keeps the digital hash on the microprocessor's power bus out of the 6500 Series driver. Another approach that might be taken is to provide a separate power supply for each sonar driver in the system. However just providing a separate regulator, still powered from the same battery, will not necessarily keep the microprocessor's digital hash off of the sonar driver's power bus. Keeping this noise out of a sonar driver becomes increasingly important as it is used to measure longer distances. The returning echoes rapidly weaken as distances are increased [Everett, 1995]. One final point about the filter/energy storage bank is to use a low ESR (Equivalent Series Resistance) capacitor for C1 (Panasonic makes some very fine capacitors for this and other power supply applications, check a Digi-Key catalog). A low ESR capacitor is needed to ensure that it can provide the large current pulse when the driver fires without a significant voltage drop. The inductor L1 should have a 2A rating and is typically referred to as an "RF Choke" or a "Hash Choke" (Radio Shack part #273-102C, or check a Mouser catalog).

Muxing Transducers

The next issue to address is how to multiplex multiple transducers to a sonar driver (for a discussion of why you might want to do this, rather than provide each transducer with its own driver see part 1 of this article). The approach taken (see Figure 1) is pretty straight forward–a 74HC138 drives eight opto-mosfets, allowing a single driver to switch between up to eight transducers. An earlier version of this circuit used in David Musliner's SMARTV robot used opto-triacs instead of mosfets. When Rick Moll tried these he found that they required large amounts of drive current through their LEDs to operate reliably–and further more found that they compromised the performance of the sonar at long distances. The opto-mosfets operated with considerably less drive current (as little as 5mA), thus decreasing the sonar systems' energy requirements, and operated with no discernible degradation of the sonar's performance. The opto-mosfets Rick used in his modular sonar system are manufactured by International Rectifier, part number PVU414. They are available from Digi-Key for under $4.00 each in quantities of 10.

Note also in Figure1 the 100K pull-down resistors on the three address lines of the 74HC138. As we will see below, the multiplexer is mounted in a metal box with the three address lines running to a connector. The three resistors provide some additional protection against static discharges when plugging or unplugging the cable, as well as assuring that the multiplexer will go to a known address should the cable have a loose connection or if the address lines are unconnected on the microprocessor end of the cable.

Interfacing to the Polaroid Driver

Figure 1 also shows the microprocessor to Polaroid driver interface, at least the driver's end of it. As we will see below the driver and multiplexer are remote from the microprocessor, attached via a flat ribbon cable. We use three control signals to interface to the Polaroid driver, two (INIT and BINH) running from the microprocessor to the driver, and one (ECHO) returning back. The INIT signal tells the driver to pulse its transducer. A short time later the microprocessor activates BINH to tell the driver that it is time to start listening for a returning echo. The time delay between the activation of these two signals allows for the transducer to stop "ringing" before the driver begins listening for a returning echo. If this delay is too brief, the driver will sense a false echo when BINH is activated due to the transducer hearing the ringing of its own pulse. The signal running back to the microprocessor is ECHO. When the driver hears an echo it activates this signal,

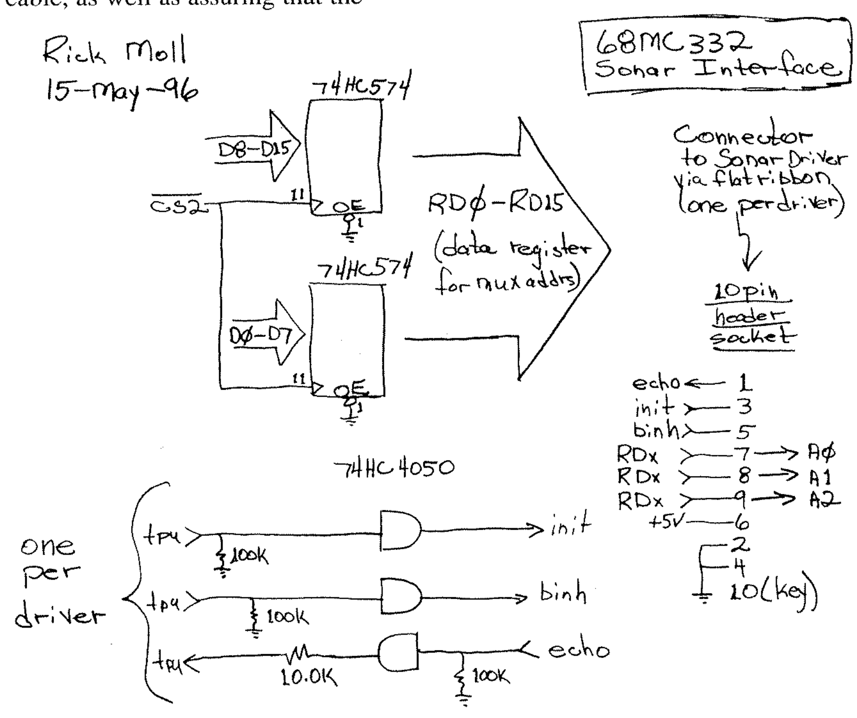


Figure 2: *68MC332 Sonar Interface Schematic*

notifying the microprocessor that it is time to check the clock since the echo has returned. By measuring the time from the activation of INIT to when the driver signals ECHO back, the microprocessor can determine the distance to the sensed object. As we will see in part 3 of this article, the microprocessor, a 68MC332, will make use of its TPU (Time Processing Unit) for measurement, as well as for generating the timing of INIT and BINH.

Note in the schematic that a 74HC32 is used simply as a buffer between the Polaroid driver and the cable. The driver is very sensitive to noise on its control lines, so the 74HC32 is used to buffer the driver from noise on the cable's INIT and BINH lines. Two 100K pulldowns are also used on these two lines as we did for the three address lines of the 74HC138. Besides the reasons mentioned above for the three address lines, the pulldown on the INIT line is also important to prevent the driver from firing randomly should the cable have a loose connection. One of the 74HC32 gates is also used to buffer the ECHO line. The Polaroid driver's echo line is "open-collector" which further requires the 33.3K pullup resistor to be used. Since the only drive for the echo signal when it is high is via this pullup, and since we want to keep the value of this resistor high so as to minimize power consumption, the 74HC32 gate is needed to provide enough current to properly drive the cable running back to the microprocessor.

Figure 2 shows a schematic of the 68MC332's side of the interface. It simply consists of a 16-bit output register for generating multiplexer addresses and some buffers for the control lines. The two 74HC574s form the output register. In the current design only the bottom 9 bits are used–controlling three 1x8 multiplexers. The buffers are 74HC4050s. All the control lines are connected to individual channels of the 68MC332's TPU (Time Processing Unit). Note that INIT and BINH have 100K pulldown resistors on their TPU outputs. When the 68MC332 first boots up all of its TPU pins "float" until the TPU channels are programmed to be inputs or outputs. These pulldowns prevent the sonar drivers from being fired during the boot up period or if the software fails to get the channel properly programmed. The ECHO input channel needs are a little different. First a 100K pulldown resistor is used where the ECHO signal comes off of the cable. This gives a little extra static protection to the 74HC4050 input gate, as well as preventing false echo triggers should the cable be disconnected or have an open fault. Also note the use of a 10K series resistor to connect the ECHO signal to the TPU. This prevents the TPU circuitry from possibly being damaged should the software accidentally program an ECHO TPU channel to be an output. The 68MC332 is often the most expensive part in a design so the use of series resistors on any TPU pins that are used as inputs is a good idea. The 74HC4050s also give some added protection against static discharges at the off-board connections.

Shielding

68MC332 to Sonar Driver Cable

68MC332 10 pin header	Signal Name	Ribbon Cable	Sonar Driver 9-pin D-sub
1	echo	1	1
2	gnd	2	6
3	init	3	2
4	gnd	4	7
5	binh	5	3
6	+5V	6	8
7	A0	7	4
8	A1	8	9
9	A2	9	5
10	key		

Notes:
- echo, init, binh are separated by gnd lines on cable to minimize crosstalk
- key in 68MC332's 10 pin header socket is used to prevent backwards insertion

As we discussed in the first part of this article, sonar systems create a lot electrostatic noise. To prevent the drivers in a sonar system from mistaking electrical crosstalk for a returning echo, not only do we need to filter their power buses, we must also provide a shielding system around each sonar driver and its transducers. This shielding can be implemented in a variety of ways. We could put a single shield around a driver and its transducers–mounting them all in the same metal box for example. Or we could put the driver and each of its transducers in separate boxes, connecting them with shielded cabling. What's important to understand here is that it is each driver/transducer system that you must shield from the other driver/transducer systems. A secondary, but important, advantage of this shielding is that it keeps other electrostatic noise (like from a near-by LCD) out of the sonar systems as well.

Fortunately, because of the frequency range that sonar systems work with (approximately 50 kHz for the Polaroid 6500 Series drivers), the shielding needs are easy to achieve. 50 kHz is high enough that we don't need to concern ourselves with iron, steel or the more exotic shielding materials–a simple aluminum box will work just fine. Also the frequency is low enough that even a crude box with simple butt joints will be adequate.

An Example and Some Tips

Figure 3 shows Rick Moll's first attempt at a modular sonar system for mobile robotics use. A 1x8 multiplexer, a Polaroid driver, and its power filter/energy storage bank are mounted in an aluminum box. Common phono jacks along with cheap off-the-shelf (actually Radio Shack's shelf) audio cables are used to connect the driver box to remote transducers also mounted in aluminum boxes. The interface to the remote 68MC332 microprocessor is accomplished with common 9-pin "D-sub" connectors (like those used for serial ports) via flat ribbon cabling. This approach allows for a great deal of flexibility in the positioning of the transducers.

Figure 4 shows a useful device you can make to help in your debugging. Trying to get a multi-transducer sonar system up and running can be quite difficult with only your ears as diagnostic tools. Once you are working with multiple transducers it can be quite difficult just to verify that they are all firing without a lot of running around getting your ear very close to each transducer, one by one. To help with debugging, try the device in Figure 4 which consists of a NE-2H neon lamp (Radio Shack part #272-1102, or check a Mouser catalog) and a 100K resistor mounted in a phono plug. You can't really see the resistor since it is stuck down the inside of the plug's center pin. By plugging these into the multiplexer outputs you can visually verify operation with a simple glance. If you are building an eight transducer multiplexer it helps to have eight of these on hand so that you can debug the mulitplexer on the bench without going crazy plugging and unplugging transducers or being driven mad with the sonar buzz of them all firing up close.

It should also be pointed out that Polaroid's data-sheet says not to fire the 6500 Series driver without a transducer connected. However Rick Moll fired them a lot without anything attached, or with the neon lamp devices attached and has never had a failure of a 6500 Series driver. The drivers do use zener diodes as clamps across their outputs to keep voltages from exceeding 400V. However we can't promise you that Rick hasn't just been lucky.

Another place where Polaroid's data-sheet appears to be overly conservative is in its requirement that INIT be held inactive for a minimum of 100ms before being re-activated. We have found these drivers to work reliably even if INIT is held inactive for well under 1 ms. This design uses 0.5ms and has worked without problems.

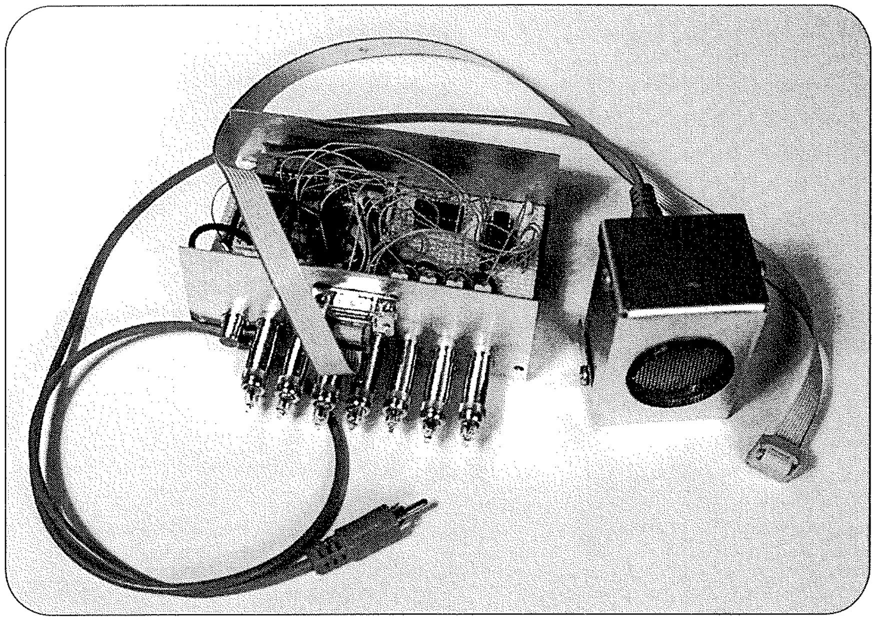
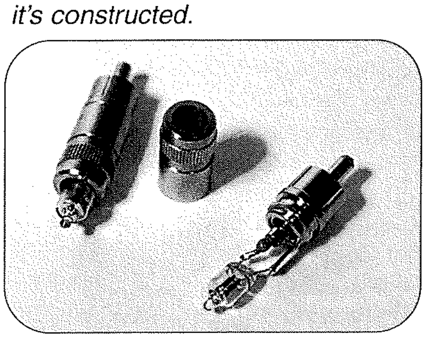


Figure 3: *A sonar driver/1x8 multiplexer shown with cover removed and seven neon test lamps connected. Also shown is a transducer.*

Figure 4: *Two neon test lamps, one disassembled to show how it's constructed.*



Conclusion

In the next, and final, part of this article we will conclude by presenting the software for driving the sonar hardware just built. This software will include an implementation of an acoustical crosstalk elimination technique based on the EERUF method developed by Borenstein and Koren. Armed with the material in the three parts of this article you will then have all you need to know to design and build your own multiple-transducer sonar system.

About the Authors

Rick Moll is currently working on a LISP development environment for use in embedded systems and low-cost machine vision and sonar hardware for autonomous robots. He can be reached at: **rick@footfalls.com**

David Musliner is a Senior Research Scientists with the Automated Reasoning research group at the Honeywell Technology Center. His current research projects focus on real-time intelligent control of autonomous agents and multi-agent distributed scheduling. Prior to joining Honeywell, David spent 16 months as a postdoctoral researcher and lectureer at the University of Maryland. While at Maryland, he led the development of novel low-cost robotc systems, including the design and implementation of guaranteed real-time control software, heuristic navigation algorithms, graphical user interfaces, microprocessor-based onboard control systems, and sonar sensors. He earned his Ph.D. at the University of Michigan in 1993, concentrating on real-time intelligent control. His email address is:

musliner@src.honeywell.com

Resources

Polaroid: (800)225-1000
Digi-Key: (800)344-4539
Mouser: (800)346-6873

References

Musliner, David J. and Rick Moll. "Designing and Building a Multiple Transducer Sonar System: Part 1." *The Robotics Practitioner*, Fall 1995, 39-43.

Everett, H.R. "Understanding Ultrasonic Ranging Sensors." *The Robotics Practitioner*, Fall 1995, 27-38.

Borenstein, Johann and Joram Koren. "Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance." *IEEE Transactions on Robotics and Automation*, February 1995, 132-138.