Smart Car

Roger Grayson
Stephen Haug
CENBD 451
Spring 2003

Avoid this fate with the Smart Car!

Smart Car

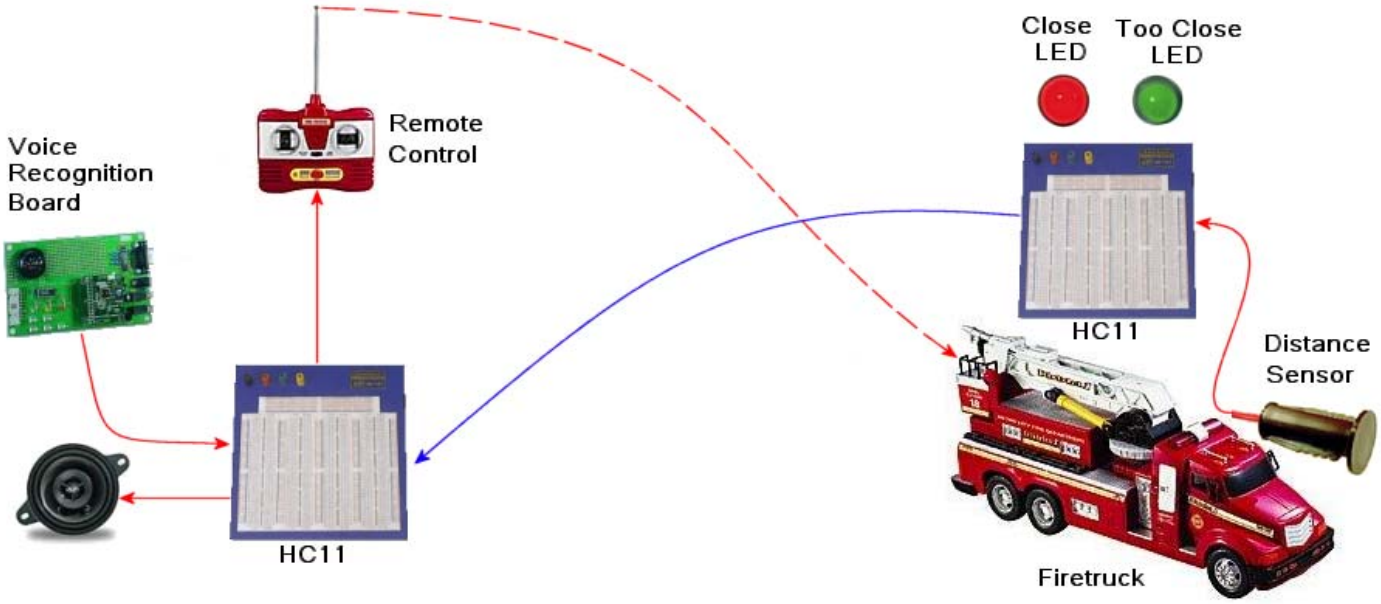
Product Description

The Smart Car is a voice activated, radio controlled car, with an auto-braking system. The Smart Car recognizes human voice input through a Sensory Voice Extreme Module. Distance from the front of the car is measured using a distance sensor. The sensor's and the Voice Extreme's outputs are sent to an HC11 microcontroller. Since the remote for the RC car operates using 4 simple switches, with 5Vdc as 'on', the HC11 controls the remote through 4 relays. The remote is used to transmit the motor controls to the car.

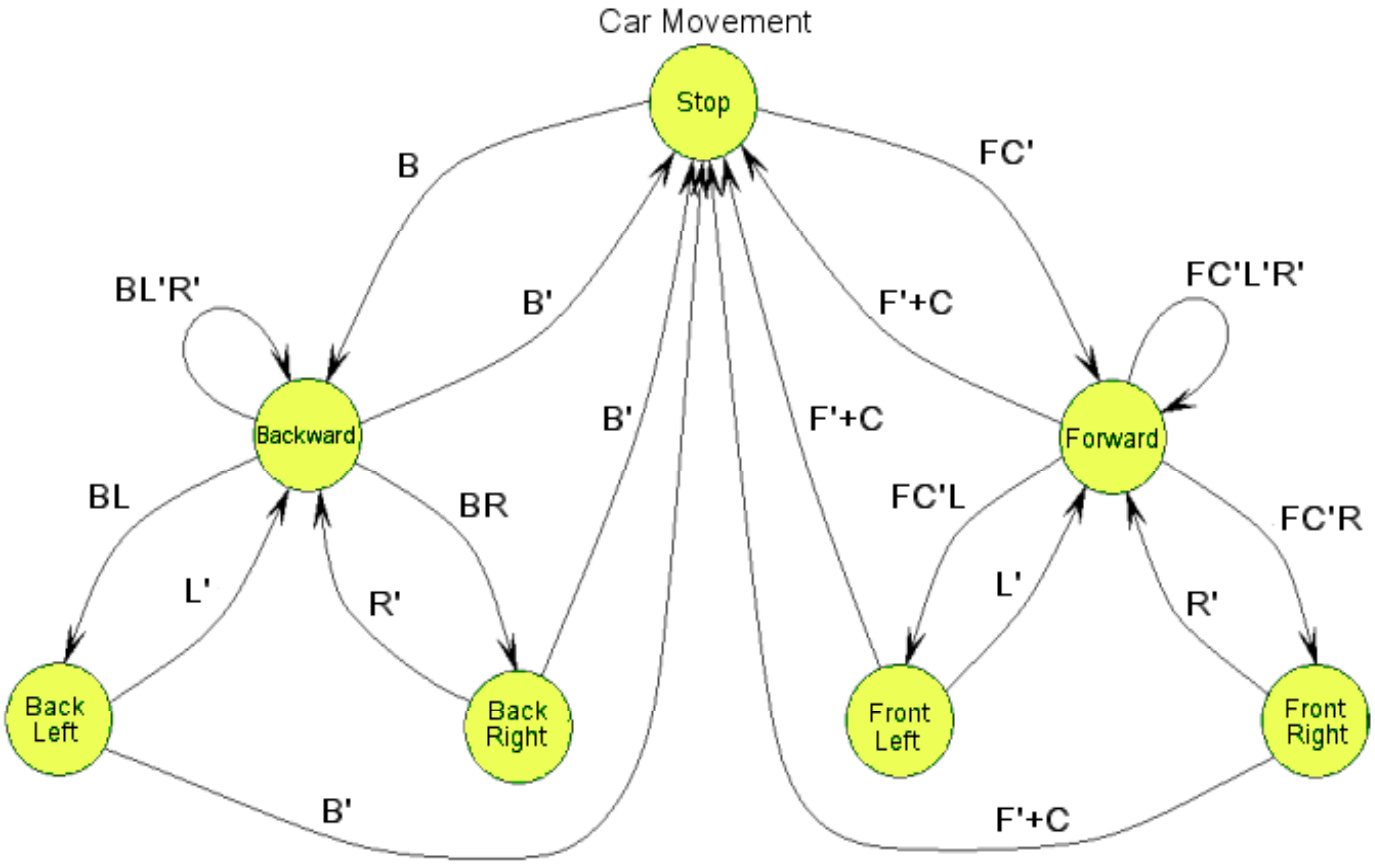
When started, the RC car will be motionless. It will remain so until the Voice Extreme recognizes either of the words "forward" or "reverse," at which point the Voice Extreme will then send an output pin high. The HC11 will see its input pin go high, and activate either the "forward" or "reverse" relay to use the remote to move the car. If the Voice Extreme recognizes the words "left" or "right," a similar output pin rise will occur, causing the HC11 to activate either the "left" or "right" relay. The last two words the Voice Extreme can recognize are "stop" and "straight." These two, when transmitted to the HC11, will cause the HC11 to shut off either the "forward" and "backward" relays, or the "left" and "right" relays, whichever are appropriate.

Throughout all of the interactions between the Voice Extreme and the HC11, the distance sensor on the front of the car will be transmitting the distance of any object in front of the car to the HC11. The HC11 will use this to determine if it should obey the user or not. When an object is detected while the car is moving forward, the HC11 will output a verbal warning through an Archer Voice Recording and Playback chip. If the object is too close to the car, the HC11 will stop the car from moving forward. The user can then direct the car to move backwards, and continue around the object.

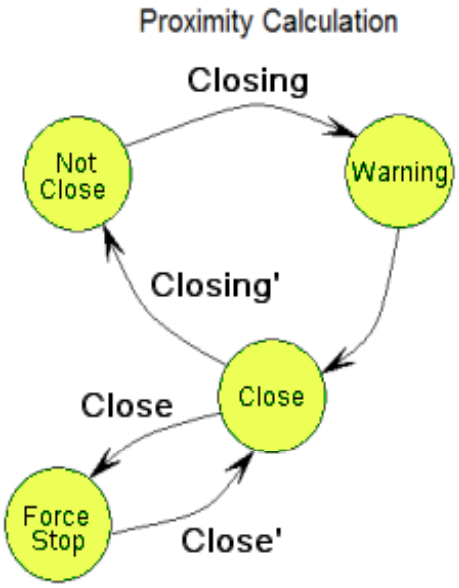
High Level Block Diagram

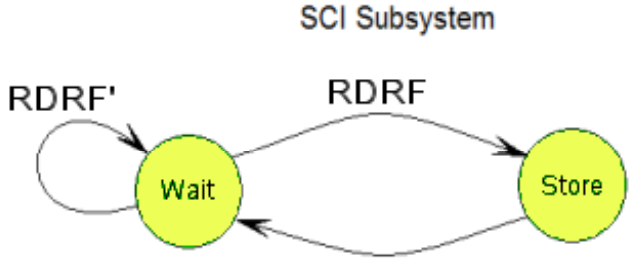


State Diagrams



Legend:
B = Backward
C = Close
F = Forward
L = Left
R = Right





Chapter 2: Technical Details

The main hardware components of the Smart Car are two HC11s, a Sensory Voice Extreme Module (SVEM), a Radio Shack RC Fire Engine, an ultrasonic distance sensor, and an Archer Voice Recording and Playback chip (AVRP). To connect the various pieces, one PAL, six relays, two diodes, many capacitors and resistors, several batteries and LEDs, a microphone, a speaker, and a plethora of wire were used.

The Smart Car was broken down into two components – the Control Unit (Figure 1) and the Fire Engine Unit (Figure 2). The Fire Engine unit consists of the Fire Engine, one HC11, and the distance sensor. The remaining parts form the Control Unit.

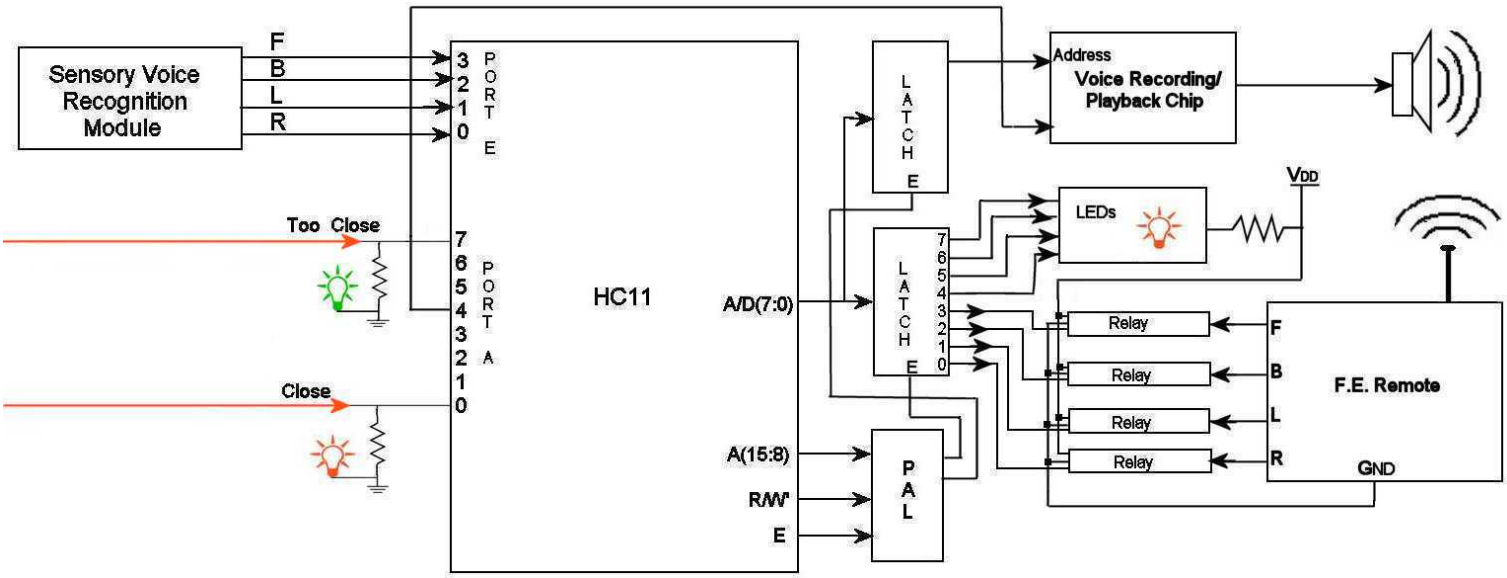


Figure 1: Control Unit

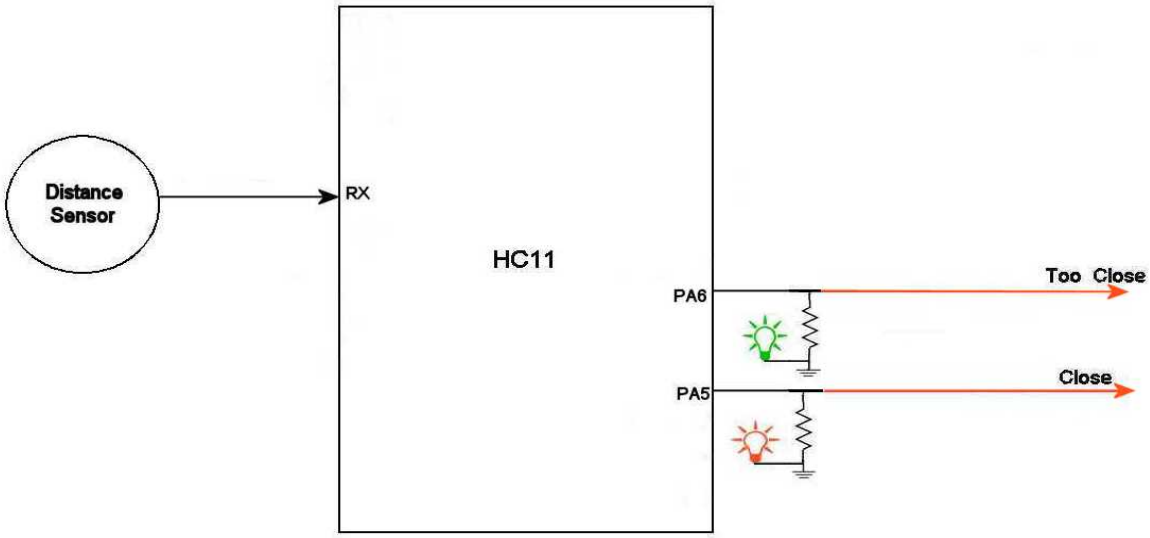


Figure 2: Fire Engine Unit

Control Unit

The heart of the Control Unit is an HC11. The HC11 takes as input four signals from the SVEM, and two signals from the Fire Engine Unit, and outputs four signals to four relays, an eight bit address and one bit enable pulse to the AVPR. The HC11 also outputs four signals to the LEDs, turning each one on according to the signals output to the relays.

The beginning of the Smart Car control starts with the SVEM. A microphone attached to the module allows a user to speak and be recognized. The SVEM was preprogrammed using the Voice Extreme IDE software that came with the module; the program is reproduced on the CD accompanying this report. When powered up, the SVEM asks the user to say six words. They are, in order, "Stop," "Go," "Back," "Straight," "Left," and "Right." If these are spoken out of order, the Smart Car will not function correctly. After the words are stored, the SVEM is capable of recognizing any of these commands.

The program then instructs the module to wait for one of the above words to be recognized, and pull one or more of its output pins high as appropriate. The four output pins used are associated with Forward (or "Go"), Back, Left, and Right. As would be expected, the recognized word "Go" turns on the Forward pin, "Back" turns on the Back pin, "Left" turns on the Left pin, and "Right" turns on the Right pin. The program will not allow the Forward and Back pins to be on simultaneously, or the Left and Right pins to be on simultaneously, but any other combination is allowed. This allows the Smart Car to move Forward and Right, for example, but not Forward and Back. The word "Stop" turns off the Forward and Back pins, and "Straight" turns off the Left and Right pins.

One quirk of the SVEM is that the high output is only 3V. Fortunately, the HC11 recognizes this as a logic one, even if it is operating at 5V.

The program for the HC11 instructs it to look for the logic level of the SVEM output pins, which are attached to the HC11 on port E. Following that, the HC11 checks for the value of Close and Too_Close, on pins PA7 and PA0, respectively.

The Close and Too_Close signals are initiated in the Fire Engine Unit, and are described in that section. The two signals are transmitted to the Control Unit via two wires. The Close signal indicates that the Smart Car is close to an object, and that a warning should be output to that effect. The Too_Close signal indicates that the Smart Car is very close to an object, and should be stopped.

When the Close input goes high, the HC11 outputs a warning to the user through the AVRP. The AVRP is a EEPROM, capable of recording and playback of up to 20 seconds of sound. All work needed to store and output sound, including pre-amp for playback, is done automatically inside the chip. The AVRP is fully addressable, enabling playback of many messages. An End Of Message pulse is output after a message is finished playing, but the Smart Car ignores this, as only one message is played, and never played quickly enough to overlap.

The PAL attached to the HC11 is used to map the AVRP to memory address 0x8200 and the Car/LED combo to address 0x8100. To output the warning, the HC11 writes 0 to the address 0x8200, to start the playback at the first address space on the AVRP, which is where the warning message was prerecorded. Then, a 1 E clock low pulse is sent to the chip enable of the AVRP, starting the playback. The playback continues until the end of the message, without any further monitoring by the HC11.

Once playback has been initiated, a 'Warned' flag is set, preventing repeated warnings. The flag remains set until the car backs up enough to send the Close pin low.

When the Too_Close input goes high, the HC11 stops the forward motion of the Smart Car. The forward motion remains impeded until the Too_Close pin goes low again. Any other direction is allowed. If the user tries to send the car Forward and Right, only the Right command will be transmitted.

The Smart Car's motion control is transmitted by the Fire Engine remote. To send the appropriate signal, as determined by the inputs, the forward, reverse, left, and/or right pin from the remote must be pulled to the remote's ground. The easiest and most reliable method to achieve this is through relays. One side of all four control relays is tied to the remote's ground; the other side of each relay is tied to one of the remote's control pins. One side of each relay's coil is tied to V_{DD}; the other side is tied to the latch mapped to address 0x8100. To allow any control to be transmitted, the appropriate relay is turned on by pulling the coil pin low, which is done by writing to address 0x8100. As a result, current flows through the coil, opening the relay. Since a low signal turns on the relay, the first command in the HC11's program is to write 0xFF to address 0x8100, turning all of the relays off.

The control signal output to 0x8100 is selected by a switch statement, using the Too_Close input and the SVEM outputs as the selector. Only the allowed states, as defined above, are defined in the switch statement. If an error occurs, the default case will catch it, turning all relays off.

Since only four relays are used, and the latch has eight outputs, the upper four bits of the latch are tied to LEDs. When a relay is opened, the corresponding LED is turned on. This allows for visual confirmation of the Smart Car's controls, making debugging much simpler.

Fire Engine Unit

The Fire Engine Unit is composed of the distance sensor, a second HC11, and two transmission wires. All of these are mounted on top of the Fire Engine, on a large block of styrofoam. For obvious reasons, the sensor is placed on the front of the car.

The distance sensor outputs to the HC11 in RS232 format, via the MAX232 chip. The HC11 reads the SCI port, then extracts from that information the distance between the sensor and any object in front of the sensor. The distance sensor outputs three bytes for each distance it calculates. The first is a sync byte, followed by the lower data byte, then the upper data byte.

The HC11 reads all of these, throws away the sync byte, and concatenates the data bytes to get the distance data.

The HC11 then compares the distance read to two values – close (about four feet) and too_close (about 1 foot). If the value is greater than close, no output is sent. If the distance is less than close, the Close output is sent by pulling PA5 high. If the distance is less than too_close, the Too_Close output is sent by pulling PA6 high.

To communicate the Close and Too_Close signals between the two HC11s, the two transmission wires are used. One wire, Close, is connected from PA5 of the Fire Engine HC11 to PA0 of the Control Unit HC11. The second wire, Too_Close, is connected between PA6 of the Fire Engine HC11 and PA7 of the Control Unit HC11. To ensure that 5V (logic one) on each board is the same, a third wire connects to the ground of both boards.

Chapter 3: Unique Attributes

Rather than being controlled like a normal remote control car, the Smart Car responds to six different voice commands: “Stop,” “Go,” “Back,” “Left,” “Right,” and “Straight.” This is accomplished through a voice recognition chip. When the chip recognizes one of the commands, it sends the appropriate signal through the remote to the car in the same manner that pressing or releasing Forward, Reverse, Left or Right would.

The Smart Car is not only called smart because it responds to voice commands. It also has an ultrasonic distance sensor mounted on the front of it. When it detects an object within 4 feet of it, the distance sensor sends a signal to the HC11, which in turn outputs a warning, verbally and via an LED, to the user letting him/her know that the car is about to crash. If the user fails to heed to the warning and the car continues to approach the object, the distance sensor will send another signal to the HC11 which will cause the car to stop, preventing it from crashing and causing any unwanted damage.

When we first started designing the Smart Car, we were only planning to use one HC11 to control everything. While this would have been entirely possible, we would have had to mount the breadboard on the car itself, and in order to control the car, the user would have had to chase the car around yelling commands at it. How did we get around this? We used two HC11's, and two breadboards. One breadboard is mounted on the car itself (the Car HC11), and controls the distance sensor and warning LEDs. The other breadboard is part of the remote (the Control HC11), and controls the voice recognition chip and the remote’s relays. We needed a way to send data from the Car HC11 to the Control HC11. Originally, we had planned to have another transmitter connected to the Car HC11. This transmitter would have been responsible for sending two signals to the Control HC11. Due to complications, we had to use a rather long wire to connect the two HC11s. This wire accomplishes what the remote would have, sending the two signals to the Control HC11. The Control HC11 interprets these signals, and takes the appropriate action (i.e. stops if the car is too close to an object).

Chapter 4: Constructive Criticism

Originally, we had planned to use a second remote to transmit data from the mounted HC11 back to the remote HC11. As stated earlier, we ended up having to use a long wire. Ultimately, we would have liked to use transmitters, probably infrared, to transmit the data. This would have eliminated the need for a long wire and condensed the design.

The use of two breadboards was probably unnecessary. Using one HC11 to control the distance sensor simplified the design on our part, because it was right next to the sensor and it only had to send two signals back to the Control HC11. If we had tried to send the raw distance sensor output back to the Control HC11, it might have caused problems, because it outputs data serially, in RS-232 format. It outputs three bytes of information: a sync byte, the least significant byte of the distance, and the most significant byte of the distance. This happens every 50 milliseconds, which might have been too fast for the second remote to handle. Since we decided to use wire rather than a remote to transmit the distance data back to the Control HC11, we might have been able to connect the distance sensor directly to the wire, and send that back to the remote. A good solution for this would have been an infrared transmitter, which is plenty fast enough to send the raw distance sensor data back to the remote HC11.

The Control HC11 is comparative to a large multiplexer. It takes data in from the voice recognition chip and the Car HC11, and from that data determines the logic levels for the output pins. In this manner, we might have been able to use some combination logic, rather than a microcontroller. If one were to produce this car, combinational logic would be both cheaper and smaller to build. However, due to time constraints, we used the HC11.

The voice recognition board can not only recognize verbal commands, but also has an onboard speaker and is capable of outputting sound. Rather than use a separate sound chip and speaker to output a verbal warning if the car is close to an object, we could have used the voice recognition chip to accomplish this. The main reason we used the separate sound chip was to make things easier for us. We already had previous knowledge of this sound chip, so we knew how it worked and how easy it would be to implement. Again, if one were to produce this car, it would be more compact and cheaper to use the voice recognition chip to not only take in verbal commands, but also output the verbal warning.

The ultra-sonic distance sensor seemed to have a few problems. Although the data it output was easy to interpret because it was in RS-232 format, it seemed to be extremely sensitive to noise. The sensor sends out ultra-sonic sound waves and detects their return, then calculates the distance from how long the return trip was. Any other sound in the room seemed to affect how the sensor acted. When the room was completely quite, the sensor seemed to work as it should; however, if there was background noise, such as people carrying on a conversation, the sensor's output seemed to be sporadic at times. Another noise issue was echoes – the labs are small and crowded with equipment, causing the sound waves to bounce and echo repeatedly. Had we had our choice of distance sensors, we would have chosen another model, such as infrared, that could not be affected so easily by the environment.

The voice recognition chip is also very sensitive to sound. This is especially evident when you try to program it in a quiet environment, and then try to use it in an environment with background noise. The chip does not recognize commands as well with the background noise, and sometimes just ignores them because it of that. The best solution to this is to program the voice recognition board in the same environment that you will be using it in. This will produce the best results.

Appendix A: Duplicate This Project

To reproduce the demonstration from this project, a user needs a Sensory Voice Extreme Module, an Archer Voice Recognition/Playback chip, two working HC11 boards, a distance sensor, several relays, and a remote control car. In order to use the programs on the CD, several other conditions must be met:

1. The AVRP address must be memory mapped to address 0x8200 of the control HC11, and the chip enable must be wired to PA4.
2. The distance sensor must output RS232, with the output as sync byte, low data byte, sync byte, high data byte. Further, the sync bit should be 0x55.
3. The relays must be memory mapped to a latch at 0x8100 of the Control HC11.
4. The car must be hacked as explained below.
5. The car must be both large enough to hold an HC11 board, and slow enough to be stopped before crashing. The Radio Shack Fire Engine With Pump is well suited for this task.
6. The SVEM pins must be attached to port E of the control HC11, as explained below.

The AVRP is very easy to install on a breadboard. The instructions that come with it show very clearly how to install it. Start by following the schematic on page 6, the "Simple Play/Record Circuit." Only a few changes need to be made:

1. Instead of dip switches and pull up resistors for the address pins, use the output of the memory mapped latch.
2. The active low chip enable should be attached to PA4, instead of the momentary switch. Since it is active low, the signal must be inverted to prevent constant playback on power up.
3. Capacitor C2 is backwards in the schematic – negative should be towards ground.

Additionally, the AVRP should have a warning message prerecorded. Otherwise, the warning will be static!

The SVEM is also easy to install. Simply install the software from the CDs provided, then follow the instructions in the quick start guide. To install the program used in this project, open a new project and add the program from the CD to the project. The program also requires sddemoa.veo, sddemo.ves, and hello.ves, all provided by the Sensory Voice Extreme IDE. After adding all the files, simply build and download the project as per the instructions.

After programming, the SVEM will retain the program until reprogrammed, even when disconnected from power. To use the SVEM, attach the output pins to port A of the control HC11 as shown in the table on the next page.

Variable	Sensory Module Pin	HC11 Pin
Forward	P0.2	PE3
Back	P0.3	PE2
Left	P0.4	PE1
Right	P0.7	PE0

Table 1: Pin assignments for Sensory Module and HC11

When powered up, the SVEM will ask for six words. In order to work correctly, these word must be:

1. Stop
2. Go
3. Back
4. Straight
5. Left
6. Right

Note that Go corresponds to Forward. Forward is a hard word for the SVEM to recognize, so Go was used instead.

If the distance sensor outputs RS232, the output should be attached to PD0 (Rx) of the HC11 on the car, hereafter referred to as the Car HC11. If the sensor outputs some other format, it must be interfaced to the Car HC11 in the appropriate fashion. In that case, the program on the CD for the car HC11 will need to be modified to read the sensor, and the data stored in the distance variable. The sensor should then be mounted on the front of the car.

The hardest part of the reproduction will be hacking the car. The car purchased must have a full function remote, with separate forward/back and left/right operations. If turning is tied to a forward or backward operation, the car will not work. Also, the car must have only one speed. A variable speed car can be used, but only if the speed control is independent of directional control.

The remote must be disassembled and its operation determined. Hopefully, it will operate as simple switches. If the remote operates as all of the remotes we found, the switches connect a pin to ground, causing the remote to transmit. If this is the case, a wire should be attached to each of the directional controls, and to ground. This may require creative soldering.

Once hacked, the remote can be attached to the relays. One side of all four relays should be connected to the remote's ground; the other side of each relay should be connected to one control pin. To work with the given program, the order should be:

Direction	Bit of Latch at 0x8100
Forward	3
Back	2
Left	1
Right	0

The code provided for the Control HC11 expects LEDs to be attached to the upper four bits of the relay latch. If these are used, they should be active low, with less than 24 mA current. If not used, the latch outputs should not be attached to anything that is not active low. Whatever is attached will mirror the relays. In the case of the LEDs, when a relay is open, the matching LED is on.

Once all of these steps have been accomplished, the Car HC11 should be attached to the car. The distance sensor should be attached to the front of the car. Batteries need t be used to power the HC11 board and the sensor. Nine volt batteries are simplest, with connectors that can be found at Radio Shack. Make sure to have a LOT of batteries.

To pass the Close and Too_Close signals from the Car HC11 to the Control HC11. The easiest method is with wires. Make sure to choose flexible, light wire. Telephone wire works well. The connections made should be Car PA5 to Control PA0, and Car PA6 to Control PA7. To ensure proper voltage readings on the two HC11s, the grounds of each should also be connected. Again, since telephone wire has four wires in a sheath, it is an excellent choice, as long as the solid wire, not stranded, is used.

If all of the above has been followed, the only remaining step is to power up the boards, speak the requested words to the SVEM, and turn on the remote and car. It is a good idea to have two people to experiment with the Smart Car – one to speak the commands, and one to chase the Smart Car when it runs away!

Sources

The Fire Engine With Pump picture was taken from the Radio Shack web site, April 12, 2003, <http://www.radioshack.com/product.asp?catalog_name=CTLG&category_name=CTLG_009_008_001_000&product_id=60-4319>
Note: The above URL is all on long URL spanning two lines. If used, make sure to copy the entire address into the address line of your browser

All remaining pictures used were obtained form the Microsoft Clip Art Gallery, April 12, 2003 < http://dgl.microsoft.com/mgo1en/home.asp>

Documentation for the Sensory Voice Extreme Module can be found on the Sensory web site, <http://www.voice-extreme.com>