UPC SCANNER FOR THE SIGHT-CONSTRAINED

Brandon Nicolosi, Christopher Saaidi, Dario Marte
(bmnicolo, csaaidi, djmarte)@uno.edu

Department of Electrical Engineering
The University of New Orleans
2000 Lakeshore Drive,
New Orleans, LA 70148
www.uno.edu

Abstract

The UPC (Universal Product Code) Scanner for the Sight-Constrained is a product which assists individuals in identifying coded items purchased from a store. The scanner allows a consumer to scan the UPC barcode of a product and within a few seconds hear, through a synthesized voice, the contents of the scanned product along with other important product information such as shelf-life, food allergens, and safety notices. The scanner device developed in this project is compact, handheld and interfaces to the user's computer over a wireless link utilizing the IEEE standard 802.15.1, more commonly known as Bluetooth. The product requires a 32-bit personal computer running Windows XP, Vista, or Windows 7 in the user's household.

Introduction

Background

As the rate of the elderly increases in this country, there is a growing need for new and innovating products to meet the specialized clientele who require simple and easy to use products. As people age, they tend to become far-sighted and therefore unable to see near objects clearly. However, there are many more serious eye conditions, regardless of a patient's age, which cannot be easily corrected by the use of reading glasses. The UPC Scanner for the Sight-Constrained, enables anyone to quickly and easily scan the UPC barcode found on cans, packages, and other products and then hear information about the contents of the product, along with other important information such as shelf-life and any product-associated allergens.

UPC is barcode symbology that is widely used to track and scan items at stores in the United States and Canada[1]. For that reason, our product only reads barcodes formatted in the UPC standard because every item (with a few exceptions such as prescription drugs) purchased from a store in the United States will have a UPC formatted barcode. The two types of UPC codes in use are the UPC-A (standard UPC) and the UPC-E (mini UPC). Both UPC types are supported by our product.

Project Description

The three primary components of the device include a UPC Scanner, an interface/database program, coded in C++, which runs on a Windows XP (or newer) 32-bit personal computer, and a voice synthesizer unit.

The UPC scanner, which comes integrated with a Bluetooth transceiver from the factory, scans and transmits the barcode data to a household computer via Bluetooth. The barcode data is then read in and checked, using a search algorithm, against a database of known UPC codes, to determine if a match has been found.

Once a match has been made in the database, the bit-stream of information corresponding to the product (such as description, shelf-life, and food allergens) that is required to make the voice synthesizer "speak" is then transmitted wirelessly from the PC to the voice synthesizer unit via a Bluetooth link. In order to communicate with the voice synthesizer unit via Bluetooth, a Bluetooth←→RS-232 adapter is connected to the serial port of the voice synthesizer unit.

The voice synthesizer unit is connected to a small speaker which outputs the product information. The voice synthesizer unit, speaker, and the Bluetooth←→RS-232 adapter are part of an enclosure from which the scanner can be removed for increased portability.

The standards supported in this project include the IEEE standard 802.15.1 (Bluetooth) for wirelessly interfacing the scanner to the computer and the computer to the voice synthesizer unit (via the use of the Bluetooth ←→ RS-232 adapter), the UPC barcode standard for reading barcodes, and the RS-232 standard for interfacing the voice synthesizer unit to the Bluetooth ←→RS-232 adapter.

Economic Viability

There is currently one product, ScanTalker[2] by Freedom Scientific, already on the market with features comparable to those of the UPC Scanner for the Sight-Constrained; however, the main advantages of our product are price, portability, and size. The price of the ScanTalker, including barcode scanner and the PAC Mate[2] which is required and sold separately, is currently $3,380. The PAC Mate is a separate computer which the ScanTalker connects to using a physical cable and therefore, must always be in close proximity to the actual barcode scanner. Moreover, the PAC Mate weighs two pounds, not including the weight of the scanner. The parts for our product cost approximately $500, and the compact scanner with all other hardware attached is portable and weighs less than half a pound. Therefore, even with research and development costs and the cost of licensing the entire UPC database, we believe our product could be very competitive in the marketplace.

Design Methodology

The IEEE standard 802.15.1 (Bluetooth) was chosen for all wireless communication in this project. This was done due to maximize compatibility, reliability of wireless reception, and range of wireless reception. The wireless scanner chosen for the project is integrated with a Class 1 wireless Bluetooth transceiver enabling it to work up to 100 meters or 330 feet away from the host computer. Also, most computers on the market now come with internal Bluetooth adaptors, or one can buy a USB Bluetooth dongle to add Bluetooth capability to any PC for approximately $20, thereby allowing our product to be compatible with the vast majority of computers.

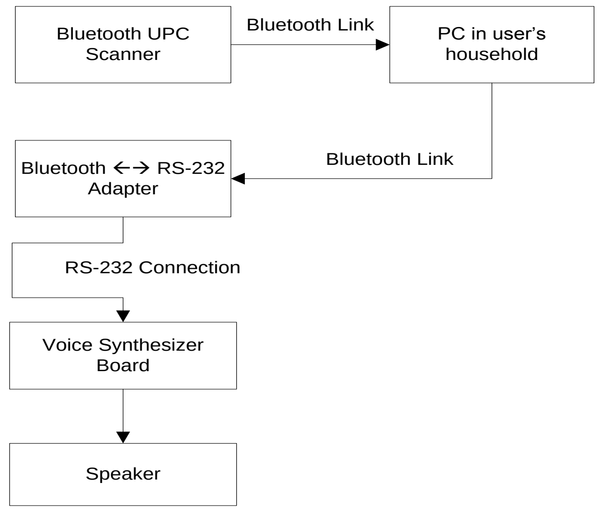
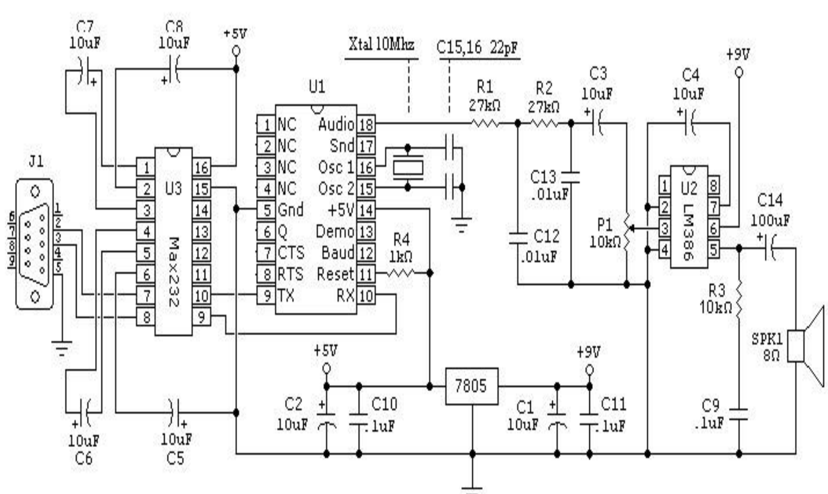


FIG 1: SYSTEM BLOCK DIAGRAM

The SoundGin[3], by Savage Innovations, was chosen as the voice synthesizer because it was available preassembled on a small circuit board that includes the SoundGin PIC, a voltage regulator, audio amplifier and filter, and a RS-232 (serial 9-pin) connection. The circuit is shown in Figure 2. There is at least one product on the market with similar functionality, but with only the individual PIC chip for sale.

The two primary challenges of the project were not at the hardware level, but at the software level of the design. The first major milestone was interfacing the scanner to the computer and the computer to the Bluetooth←→RS-232 adapter (which is connected to the voice synthesizer). Both the scanner and the Bluetooth←→RS-232 adapter support the SPP Bluetooth profile. A Bluetooth profile is a wireless interface specification for Bluetooth-based communication between devices. The use of profiles, as called by the 802.15.1 standard, specifies that in order to use Bluetooth technology, a device must be compatible with the subset of Bluetooth profiles necessary to use the desired services[4]. A Bluetooth profile resides on top of the Bluetooth Core



Specification, and in particular the SPP profile allows the wireless links to be completely transparent and emulate two serial cables, and therefore appear in windows as COM ports, just as actual serial cables would appear. However, reading and writing to COM ports is not straightforward while using C++. Settings such as baud-rate, hand-shaking, and parity cannot simply be entered, and there is no data-type corresponding to a COM port. Another major difficulty is that serial communications is asynchronous. However, there have been several libraries written for C++ and the Win32 API (Windows Application Programming Interface) dealing with serial communications, each with its own pros and cons, but the one used for implementing this project is open-source and written by Ramon de Klein[5].

The second primary challenge consisted of making the voice synthesizer's output sound as human and understandable as possible. In American English, there are 40 phonemes[6]. Phonemes are the different sounds made when people speak and are the smallest units of spoken language. Because variations of these phonemes, called allophones, are also recognized as parts of words and phrases, additional sounds are needed to synthesize intelligible speech[6]. The SoundGin is able to generate 57 different allophones. Examples of allophones would be 'e' as in "Met", "ee" as in "See", or 'er' as in "Fir". Therefore, each word must be sounded out, syllable by syllable, in order to generate a sequence of allophones. Each allophone corresponds to a hex value, and a sequence of hex values sent to the SoundGin generates speech. However, a sequence of allophones with no variations in pitch, speed, bend, density or volume will sound very robotic. Therefore, each word and each sentence must not only be "spelled" correctly using allophones, but must also be appropriately colored with variations to sound as human-like as possible.

FIG 2: CIRCUIT FOR USING SOUNDGIN WITH A RS-232 INTERFACE[6]

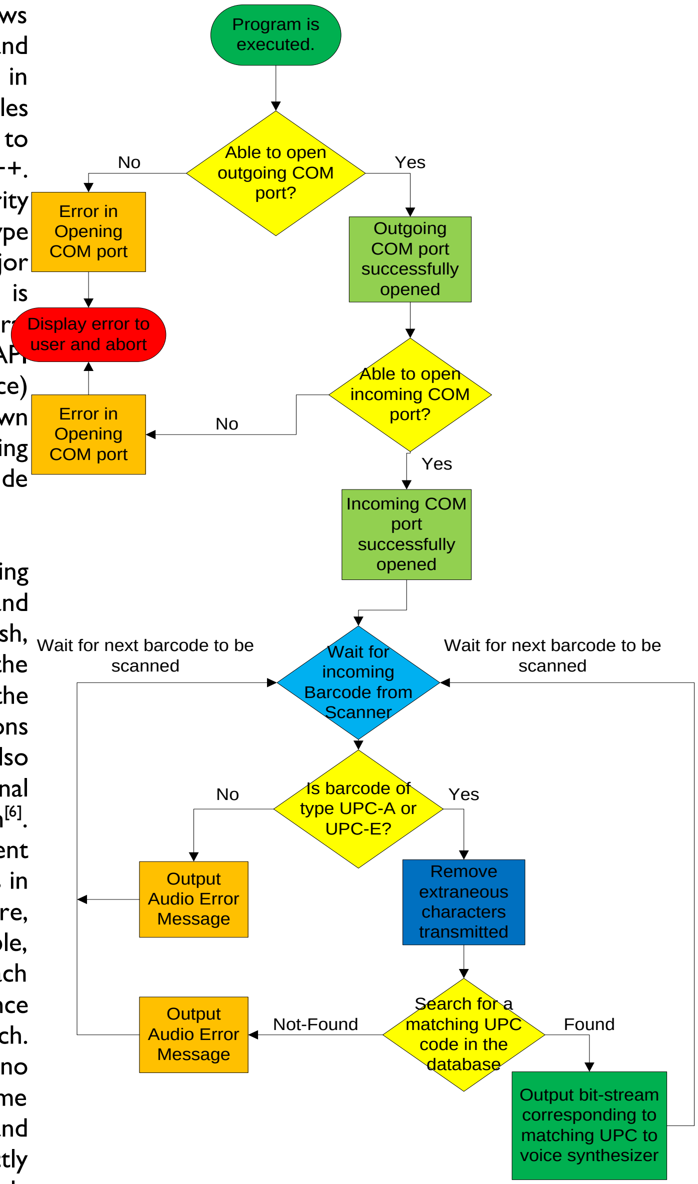


FIG 3: FLOWCHART FOR C++ INPUT/OUTPUT/DATABASE PROGRAM

Barriers and Remedies

Some of the major challenges encountered during the development of this project are detailed next.

One of the first problems was experienced when trying to make the voice synthesizer speak. The volume output was extremely low, even though the volume control potentiometer was set for maximum gain. At first, it was thought to be due to the small size of the speaker and the amount of air it was capable of moving. However, it was discovered that the voice synthesizer board was not connected to the sound amplifier properly. In Fig 2, one can see the SoundGin output is connected to the input (pin 3) of the audio amp (LM386) through a potentiometer. Gain control is modified by changing the resistance between Pins 1 and 8 and should not be done by adjusting the amplitude of the input signal[7]. Therefore, proper gain control and sufficient volume were achieved by putting a potentiometer in series with a 10μF capacitor between those two pins.

Another problem encountered dealt with over-flowing the buffer of the SoundGin. The buffer is 16 bytes, and therefore can hold only 16 phonemes at once. A work-around was to determine the length of time it took the buffer to fill and to empty, and then simply stall the program for that amount of milliseconds. Because the buffer can be filled much quicker than it can be emptied, the user never notices any glitches in speech output when input to the buffer is stalled.

One additional barrier was making the voice synthesizer say certain sounds or words. Words that have a 'u' sound or words that begin with an 'o' sound are the most difficult to have the voice synthesizer say clearly because the voice synthesizer does not have a one to one correspondence to all the vowel sounds in the English language. Therefore, one must carefully use the vowel sounds available on the SoundGin to mimic the sounds it is incapable of producing directly.

Budget

The total amount spent on parts to build the project was $520. With approximately 105 hours spent on completing the project, and a labor cost of $60/hour, the total cost to develop a prototype unit is estimated at $6,820.

Results

With the voice synthesizer board and scanner turned on, the program successfully opens both incoming and outgoing COM ports and waits until an item is scanned. When an item is scanned, its description along with shelf life and food allergen information begins to output from the speaker in less than a second after the item was scanned. The voice synthesizer, or more specifically the Bluetooth←→RS-232 adapter which is connected to the voice synthesizer, must be powered on before the program is launched, or the outgoing COM port will be unable to be opened. However, since the scanner supports Active-Paring, it can be turned off or on when the program is launched, and subsequently be turned off and back on with the program running without any difficulties. Also, the range of the Bluetooth transmission was tested in a one story house with the computer at one end and the scanner at the other with no link issues whatsoever.

Conclusions and Future Work

The project was a success, taking approximately 105 hours for all of the necessary researching, designing, coding, and construction to make one working prototype. Having designed and coded one working model, it would be very easy to replicate another in a much shorter amount of time with manufacturing being the only step required.

The use of the IEEE standard 802.15.1 (Bluetooth) was a good choice of wireless standard enabling our project to both be compatible with the vast majority of PC's currently on the market and also have the required range and power to be able to work throughout a user's household.

Currently, the product can recognize and speak the descriptions of ten different items. Primarily, future work would entail expanding the database to allow the voice synthesizer to be able to speak the descriptions of many more items.

Acknowledgement

We would like thank Dr. Rastgoufard and Mr. Henriquez for their invaluable support for our Capstone design project, and the entire Electrical Engineering faculty at UNO for their support and guidance throughout our undergraduate careers.

Biography

--*Brandon Nicolosi* is an undergraduate majoring in electrical engineering at UNO with a concentration in machinery and an expected graduation date of May 2010. He has a seven year career in the car audio and electronics field and has witnessed the up-rise and evolution of Bluetooth integration into vehicles.

--*Christopher Saaidi* is an undergraduate at UNO majoring in electrical engineering with a concentration in computer engineering and an expected graduation date of May 2010. He was awarded the Chevron Outstanding Junior Award by UNO in May 2007 and the Top Engineering Scholarship by Tulane University in January 2005.

--*Dario Marte* is an undergraduate at UNO majoring in electrical engineering with a concentration in computer engineering and an expected graduation date of May 2010. He can speak both Spanish and English fluently and volunteered for the IEEE PES Transmission and Distribution Conference and Exposition held April 2010.

References

[1] http://en.wikipedia.org/wiki/Universal_Product_Code
[2] http://www.freedomscientific.com/product-portal.asp
[3] http://oopic.com/soundgin/
[4] http://en.wikipedia.org/wiki/Bluetooth_profile
[5] http://www.codeproject.com/KB/system/serial.aspx
[6] http://si.fileburst.com/SoundginDatasheet.pdf
[7] http://www.national.com/mpf/LM/LM386.html