

Speech Activated Automation

Rui Antunes

Abstract—This article presents a simple way to perform programmed voice commands for the interface with commercial Digital and Analogue Input/Output PCI cards, used in Robotics and Automation applications. Robots and Automation equipment can “listen” to voice commands and perform several different tasks, approaching to the human behavior, and improving the human-machine interfaces for the Automation Industry. Since most PCI Digital and Analogue Input/Output cards are sold with several DLLs included (for use with different programming languages), it is possible to add speech recognition capability, using a standard speech recognition engine, compatible with the programming languages used. It was created in this work a Visual Basic 6 (the world’s most popular language) application, that listens to several voice commands, and is capable to communicate directly with several standard 128 Digital I/O PCI Cards, used to control complete Automation Systems, with up to (number of boards used) x 128 Sensors and/or Actuators.

Keywords—Speech Recognition, Automation, Robotics.

I. INTRODUCTION

THERE are an increasing number of companies that are providing speech recognition engines for several natural languages. One of them is Microsoft, with the Speech SDK 5.1 (Software developer kit) available on-line (for free) on Microsoft’s web-page. The use of this kind of software is far more simple then the creation of the entire speech recognition algorithms.

This work uses this speech recognition engine, with a specially developed Visual Basic Application, to control all the 128 Digital I/Os of a well known PCI Board (The Meilhaus ME-1000). This low-cost board can be used for many automation and robotics applications, making possible, affordable and simple to implement the concept of “speech activated automation”.

II. THE SPEECH SDK 5.1 ENGINE

The SDK 5.1 tool [1] became very popular to speech recognition researchers, because it was free and also quite accurate.

Before starting to recognize the user’s voices, the Speech SDK 5.1 engine must be trained. This operation only needs to be done once.

Manuscript received May 20, 2005. This work was supported by Escola Superior de Tecnologia de Setúbal (ESTSetúbal), Portugal.

Rui Antunes is with the Electrical Engineering Department of Escola Superior de Tecnologia, at the Setúbal Polytechnic Institute, Rua Vale de Chaves, Estefanilha, 2910-761 Setúbal, Portugal (phone: 351-265-790000; fax: 351-265-721869; e-mail: rantunes@est.ips.pt).

During training, the user is invited to read and speak a set of dictations through a microphone, in a certain language. The precision of the engine increases, as more sentences are read.

III. VB6 VOICE COMMANDS PROGRAMMING

In order to program voice activated commands in Visual Basic 6 [2], it was used the Microsoft Direct Speech Recognition (DirectSR) ActiveX. There are several events related with this component, like the PhaseFinish event, that is called when DirectSR ends the processing of the voice command. The VB6 code for DirectSR declaration is:

```
Private Sub  
DirectSR1_PhaseFinish(ByVal flags As Long, ByVal beginhi  
As Long, ByVal beginlo As Long, ByVal endhi As Long,  
ByVal endlo As Long, ByVal Phrase As String, ByVal parsed  
As String, ByVal results As Long)
```

```
End Sub
```

During the Onload Procedure of the created VB6 Form, it is necessary to add the set of all the words (or phrases) to recognize. In order to do that, we can use the DirectSR1.GrammarFromString function, which uses a string type variable, containing all the words to be recognized. The code above shows the Form Load event:

```
Dim jk As String  
jk = "[Grammar]" & vbCrLf & _  
"langid = 1033" & vbCrLf & _  
"type=cfg" & vbCrLf & "[<Start>]" & _  
vbCrLf & "<start>=Set Output sixty" & _  
Vocal & _  
"<start>=Read Input twenty four" & _  
vbCrLf & "<start>=Reset Board" & _  
vbCrLf  
' langid = 1033 English (US)  
DirectSR1.GrammarFromString (jk)  
DirectSR1.Activate
```

The DirectSR1.Activate function enables the speech engine and the DirectSR1.Deactivate function disables it. In the langid field stays the code for the natural language voice that will be used. In Table 1 are shown the codes for six different languages. For now, only the English (US) language is freely distributed.

TABLE I
LANGID CODES FOR SIX LANGUAGES

Code	Language
1033	English (US)
1031	German
1036	French
3082	Spanish
2070	Portuguese
1040	Italian

To finish the process, the word(s) recognized must be linked with the right I/O board action. If the sound matches the set of the words stored in the program, an I/O board command will be executed.

In this particular example, the digital Output 60 will be set to "High", the state of the digital Input 24 will be read, and also it is possible to reset all digital I/Os of the single board used (board 0).

```
Dim jk As String
Boardnumber As long
Port_X As Long
Bit_X As Long
Private Sub
DirectSR1_PhraseFinish(ByVal Flags As
Long, ByVal beginhi As Long, _
ByVal beginlo As Long, ByValendhi As
Long, _
ByVal endlo As Long, _
ByVal Phrase As String, ByVal parsed As
String, _
ByVal results As Long)
On Error GoTo jk:
Dim var As Long
Select Case Phrase
Case "Set Output sixty"
Boardnumber = 0
Port_X = 1 ' Second 32I/O Port (Port 1)
Bit_X = 28 ' Output 60 (bit 28, Port 1)
Function0 = me1000DOSetBit(Boardnumber,
Port_X, BIT_X, 1)
Case "Read Input twenty four"
Boardnumber = 0
Port_X = 0 ' First 32I/O Port (Port 0)
Bit_X = 24 ' Input 24 (bit 24, Port 0)
Var1 = me1000DIGetBit(Boardnumber,
Port_X, BIT_X)
Case "Reset Board 0"
Boardnumber = 0
Function1 = me1000DIOReset (Boardnumber)
End Select
End Sub
```

Note that the PCI I/O board DLL file (me1000.dll) must be stored in the system, and the three ME-1000 functions used have to be declared in the VB Form module as:

```
Declare Function me1000DOSetBit Lib
"me1000.dll" Alias
```

```
"_VBme1000DOSetBit@16" (ByVal
iBoardNumber As Long, ByVal iPortNumber
As Long, ByVal iBitNo As Long, ByVal
iValue As Long) As Long
```

```
Declare Function me1000DIGetBit Lib
"me1000.dll" Alias
"_VBme1000DIGetBit@16" (ByVal
iBoardNumber As Long, ByVal iPortNumber
As Long, ByVal iBitNo As Long, ByRef
iValue As Long) As Long
```

```
Declare Function me1000DIOReset Lib
"me1000.dll"
Alias "_VBme1000DIOReset@4" (ByVal
iBoardNumber As Long) As Long
```

IV. DIGITAL I/O PCI BOARD PROGRAMMING

The hardware used in this work, is the low-cost 128 Digital I/O PCI Board (available from Meilhaus Electronic), that is directly plugged inside the PC. It supports Windows and Linux, and several programming languages like Visual basic, Visual C++, Delphi and Borland C. It has also LabVIEW compatibility.

With four PCI slots available, it is possible to equip a single PC or Microprocessor with up to 512 Digital I/Os, making it (at a small price) suitable for use in large Automation and Robotics applications.

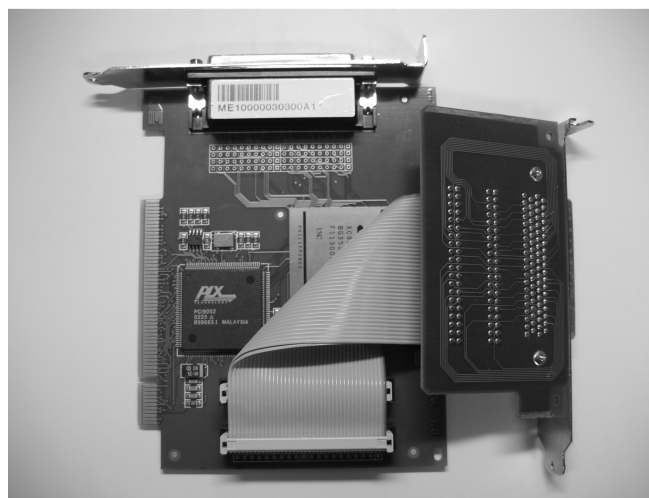


FIGURE 1
ME-1000 PCI BOARD

ME-1000 boards [3] have four bi-directional 32 bit ports witch can be accessed (at bit, byte, word, and long word level) by special software functions declared in the program. These I/O functions are:

- me1000SetPortDirection;
- me1000DIGetBit;
- me1000DOSetBit;
- me1000DIGetByte;
- me1000DIGetWord;

- me1000DOSetByte;
- me1000DOSetWord;
- me1000DIGetLong;
- me1000DOSetLong.

All other Meilhaus boards, including the analogue I/Os, and the fast counters board can also be used and programmed in the same way, with new functions available.

V. EXPERIMENTAL RESULTS

The experimental results obtained in this work reveal that the speech recognition engine used can, with proper previous training, reach a maximum of 92% words matched. We can conclude that it is a quite accurate engine.

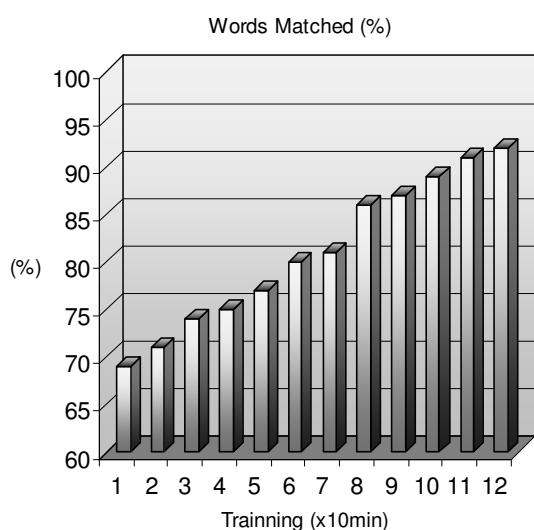


FIGURE 2
PERCENTAGE OF MATCHED WORDS, VERSUS TRAINING TIME

The audio quality of the microphone used, is also very important to achieve these results.

VI. CONCLUSIONS

This work uses the Speech SDK 5.1 engine, with a specially developed VB6 application (that can stay in memory), which allows any user to control with voice, a Robotic or Automation system, turning robots to “active listeners”, improving the human-machine Automation interfaces, and increasing also security.

With the development of Telerobotics, the World Wide Web is now being used to control Robots over the Internet. A new application, developed in Visual Studio.Net is currently being tested. This new program uses special web pages that are able to recognize microphone speech commands [4], turning the Internet Automation and Robotic applications sensible to speech.

REFERENCES

- [1] Microsoft, 2005. Speech SDK5.1 for Windows applications. Available: <http://www.microsoft.com/speech/download/sdk51>
- [2] J. Smiley, “Learn to Program with Visual Basic 6”, Active Path Ltd, 2002.
- [3] *Electronic Manual ME-1000 PCI/cPCI 1.4E*, Meilhaus Electronic GmbH, 2002.
- [4] G. Bailey. (2005). “Voice-Activated Web Browsing”. Available: <http://www.codeproject.com/audio/WebVoicePkg.asp>



Rui Antunes was born in Lisbon, in 1970. He received the degree in electronic and computer engineering from the Technical University of Lisbon, Instituto Superior Técnico (IST), Portugal, in 1993, and the MSc degree in electronic and computer engineering, in 1999, from the Technical University of Lisbon, Instituto Superior Técnico (IST), Portugal. His actual research interests involve electronics, robotics, automation and speech recognition.

He has worked at Ford Electronics (Palmela), and he is currently Adjunct Teacher at Escola Superior de Tecnologia de Setúbal, in the Setúbal Polytechnic Institute (Portugal).

Prof. Antunes was a member of the Portuguese Engineering Association.