So at the last meeting we got onto the subject of speech synthesizers and the cost of them.

Disclaimer: A lot of the low cost options are not really speech synthesizers, rather they are ways to have your project talk. While most of the lowest cost solutions require a lot of additional setup, and depending on how you count time invested, low cost could cost a lot more.

The EMIC 2 that I use in my Hero Jr robot is about $60.00. At the time I got it it was about the lowest cost synthesizer that sounded good.

The emic 2 was made by Parallax, and is getting hard to find, it is simple to use, and uses a standard DECTALK command set. It also has a lesser known, but still a standard Epson command set.

Pros: Easy to use, standard command sets

Cons: No busy (talk) line, expensive

(This is not the most expensive on the list)

Lowest Cost (If you use a Chinese UNO clone board): Talkie Library

<https://github.com/going-digital/Talkie>

The talkie library is a software voice/synthesizer system. Modeled after the TI-99/4A speech synthesizer. And additional systems and voices have been added to it.

All you need to use it is an Arduino (168 or 328 processor based) UNO, at 16mhz (I think this becomes important because it modifieds the timers and pwm).

You need a amplified speaker hooked to pin 3.

The speech sounds good, and is very understandable. It reminds me of the old Weather Alert system.

There are really three draw backs to Talkie.

1. You are limited in memory on the 328 based systems.
2. There is a limited amount of words it can say
3. Only works on a limited number of boards. (UNO for sure)

The library will take more memory than the Arduino has if you enable all the words. The best way is just select the words you know you need to say.

Pros: Easy to use, The lowest cost synthesizer (if you use a Chinese UNO clone)

Cons: Limited memory/Limited amount of words. Only works on a UNO

Raspberry PI B+/Raspberry PI 2/Raspberry PI A+/Raspberry PI Zero:

Depending on which model rPI you choose you can build a linux based synthesizer for $30 or less.

RPI 2: $35

RPI B+: $30

RPI A+: $20

RPI Zero: $5

While I’ve not tried this on the Raspberry PI, I have on a couple of other linux based boards.

For some of these boards you’ll need a USB Sound card. You may need a basic understanding of linux and the command line as well. And you’ll have to write software that will communicate with your microcontroller (probably in python, but it could be C, or anything else that gives you access to the serial ports)

You need to enable the audio either on the board (making the default audio to the headphone jack), and install espeak from the package manager.

At least on the B+/A+/Zero there maybe a problem with espeak not always speaking the 1st word. I found a work around for this, but it adds extra overhead to the board.

The issue is known and the people who made espeak seem to be working on it.

PROS: Cost, natural sound

CONS: Lot of time to setup, lot of extra programming, possible “stutter” with espeak. Possible additional cost of USB Sound card.

The $9 CHIP computer:

CHIP is a small linux based computer that costs all of $9.00 - It is brand new, just finishing up from it’s kickstarter campaign.

<https://www.kickstarter.com/projects/1598272670/chip-the-worlds-first-9-computer>

There are some advantages to this over the Raspberry PI. It has audio already directed to the headphone jack, it has on board storage that is also running it’s OS (no need for SD card), it has on board WIFI and Bluetooth. It also has some disadvantages - from what I can tell you can’t plug anything into the USB port while it’s booting, this will cause it to reboot. Espeak also has the same “stutter” that the raspberry pi has. The audio also doesn’t sound great - but it does work. And For $9.00 it’s by far the lowest cost true hardware synthesizer (True meaning it can say anything, it’s not limited to 100 word vocabulary)

Like other linux based systems setup is something of a pain, and additional software is needed to communicate with a 2nd controller.

PROS: COST!, natural sound although the audio quality is not the best

CONS: Stutter, audio quality, lot of time to setup, and extra programming needed.

SpeakJet/Babble Bot/GinSing

<http://www.speechchips.com/shop/item.aspx?itemid=6> $25.00

<http://www.babblebot.net/babblebot.html> $20.00 - $25.00

<http://www.ginsingsound.com/> $35.00

These are interesting projects, that I am not entirely sure if they are still made or not.

The SpeakJet/Babble Bot are both based off a PIC microcontroller, and I believe both need additional hardware to work.

I don’t know much about these chips, other than they really don’t sound that great.

The Ginsing is more of a true synthesizer/wave form generator, it can make sound effects and various wave forms. It sounds good in all of the demos I’ve seen for it.

Ginsing is a shield for the Arduino Uno, and comes as a kit

I think all three are Allophone based, which make them a bit hard to program, as you need to break each word down to it’s key sounds.

The SpeakJet may have a TTS (Text to Speech) build into it. Making it a bit easier to use.

Of the three Ginsing sounds the BEST

PROS: Ginsing makes sound effects & Sounds the best

CONS: May or maynot be supported now, Allophone based, They don’t sound great. May require extra parts or soldering.

\*\*\*Added Jan 6 2016 \*\*\* RoboVoice SP0-512

Robovoice is a Text to speech chip that has 800 or so rules. It sounds about like the SpeakJet/Babble Bot and Ginsing only worse. It uses TTL serial as an interface at 9600 baud.

But to be honest this might be my speaker, most of the demos I heard it sounds ok.

It’s another project that I think is now defunct. But information and demos for it can be found here:

<http://www.speechchips.com/shop/item.aspx?itemid=22>

I bought one when it was still $25.00, and will have it for the demo on the 7th.

It’s now $17.00, it’s pretty easy to use (a little picky, and doesn’t reset itself well)

It can sing, and does a half decent job of that. (website has been out of stock of these for over a year now...I think it’s dead.)

It is a chip, and needs a few resistors, and capacitors to wire it up.

PROS: easy of use, low cost, well documented. Easy to get it to sing

CONS: Maybe a defunct project, doesn’t sound great, needs some additional components to make it work.

MP3 Player:

For under $10.00 (if you don’t count the SD card) you can make your own speech synthesizer based from a MP3 player.

The MP3 Player I would recommend for this would be the Seeed Studio Music Shield.

They can be found on eBay for around $7 or $8 dollars.

Other MP3 players will work as well, there are many from China for around $6 or less.

The main reason I like the Seeed Studio Music Shield is because it lets you call the file by name. Where as the other cheap MP3 will only call by file number.

There are some advantages to using a MP3 and some big disadvantages. One advantage is you can play sound effects, or music.

One of the biggest disadvantages is having to spend time setting up all your “word” files, this could take hours or days.

Like the Talkie Library you will be limited to what it can say by the amount of space on your SD card, and the number of words you make for it. Unlike the Talkie Library it’s easy enough (although time consuming) to add new words. And only uses memory on the SD card, not memory of the controller.

It’s a good low cost solution to solve what could be a expensive problem.

Depending on which MP3 player you use will depend on how you send it a command from the Arduino, and how you address the files.

One other draw back to using MP3 players is they can be a bit slow, and the speech can seem a little broken. It’s not as bad as you’d think it would be thou.

So to setup some sounds, on my linux computer I use a program called Gespeaker which is a front end for espeak - it “makes it easy” to create wav files.

I type in what I want it to say, click record, type a file name in, and click play.

The whole process is time consuming, and the more words you want the longer you’ll be typing one word at time in. Copy all of the words you just made to the SD card, and you now have a way for the arduino to talk cheaply.

PROS: COSTS, the is another under $10 solution. Easy to use once it’s setup, easy to change and add sound effects and words.

CONS: File System maybe limiting on some MP3 players, and Time consuming to setup.

“Low Cost Speech Synthesizers on eBay”

SYN6288 Based boards. $20 and Less

Generally these are for a Chinese market but will sell them to anyone.

There is not a whole lot of documentation about them, other than they speak in Native Chinese.

it appears they will at least say numbers in English. These are something to avoid.

XFS512CE English Speech Synthesizer (recording and playback) $28.00

Most of (if any) documentation is in Chinese, from what little I’ve been able to understand I think this is less of a Speech Synthesizer and more a device that can record (x number) segments. And playback in any order. Something very much like how the MP3 player works, only no SD card.

<http://www.iflytek.com/upload/contents/2014/07/53be5e3ec4047.pdf>

PROS: None That I see

CONS: See above - Bad Documentation, and Chinese Speech.

ISD1820/ISD1700

These boards can be found on eBay for $6 bucks or less.

They are not a true speech synthesizer either, these are more like the MP3 player with addressable memory, depending on model depends on how much memory, and other features, like microphone etc.

My understanding of these are they are easy to use, but can be complicated to setup, and may need additional software to set them up.

I saw a model of these once that had something like 1500 addressable memories, I can’t find it now.

more information can be found here:

<http://www.elecfreaks.com/2215.html>

These are interesting and included as one low cost idea. Thou, limited memory makes them less then useful when compared to other solutions.

PROS: easy to use, low cost

CONS: 8 to 20 second total record time, setup may take a while depending on model you get. Not really a speech synthesizer

WT588D Sound Module $10 or less

These are similar to the above, with the biggest difference I can tell is the amount of memory.

They come in 8m or more packages. These seem to be pretty easy to use from what I can gather watching youtube videos. They do require a programmer, which costs more than the chip itself. It looks like they have addressable memory, and work similar to how the MP3 player solution works, right down to loading wav files on them.

As I’m not overly familiar with these, I’ve never used them, I can’t give a good list of PROS or CONS. They are reasonable in the price range however.

PROS:

CONS:

Using Android as a TTS

There are a few apps in Google play that will let you connect an Arduino over bluetooth and share sensors data from the tablet or phone. Most of these apps will also let you do TTS.

From a cost stand point, these apps are free, some have ads, others don’t. The real cost comes from buying a tablet or phone. (Or buying one just for interfacing to the Arduino)

There is SensorShield

<https://play.google.com/store/apps/details?id=appinventor.ai_g916414.SensorShield>

While this isn’t the prettiest app, it works, I’ve used it for TTS, and it works pretty good. It’s a little strange to use. And it’s documented fairly well - I think the guy who made it is Russian.

SensoDuino (I’m not sure if this one does TTS or not, but I included it because it has alot of other opitions and can be used with lots of sensors)

<https://play.google.com/store/apps/details?id=com.techbitar.android.sensoduino>

Arduino Total Control (Free) - <https://play.google.com/store/apps/details?id=com.apps.emim.btrelaycontrolfree>

There are a few different versions of this app on Google Play, I think for the most part they are the same app. This app is well supported, sample code is in the app, and he has a github with some other examples. It also supports connections over either bluetooth or wifi.

Uses plain ASCII to send commands to it, and is pretty simple to use.

(I will be Demoing this app at the Jan 7 meeting).

There are other apps in the play store that do either control or tts or both.

PROS: COST, for the most part easy to use

CONS: One extra large piece of hardware to carry, need bluetooth, some apps are not documented so well.

Finally The most expensive on the list:

The Arduino YUN $60 and up.

The YUN is one of those surprise you kind of devices. And I was skeptical about it when I first got it. Did not like price of it. And finally found one for $40 on eBay (Which at the time I thought was still too much). The YUN is a combination of a Linux computer, with a Arduino Leonardo. There is a library called Bridge which handles communications between the two halfs. There is also a library called Process which lets the Leonard side execute linux shell commands. On the linux side of things it’s small like 16mb small. It runs OpenWRT, and is a striped down command shell. Linux handles WIFI/Ethernet/SD card and USB

and you can install additional linux software making it possible to do quite a lot with this small computer/microcontroller.

The draw back to all this is on the linux side it only has 16mb to work with - less because the OS takes up part of that space. So installing a lot of extra packages will take up a lot of room.

Lucky Arduino thought of this and lets you use the SD as part of the file system.

<https://www.arduino.cc/en/Tutorial/ExpandingYunDiskSpace>

So to get a text to speech system working this is a must. You need more space.

The tutorial I found on installing espeak to a YUN never says this and ended up causing me a lot of headaches.

The YUN is also slow when it installs some packages (yun-gcc has been installing for about 2 hours now, I’m told to just be patient).

This tutorial also has an error in it - everytime it says “ espeak\_1.48.04\_arm71xx.ipk” it really should say “espeak\_1.48.04\_ar71xx.ipk”

<http://arduinomeetslinux.com/beyond.php?p=1>

He also has some other YUN tutorials on his site, and here is a listing of his download directory.

<http://arduinomeetslinux.com/download/>

His demo video is here:

<https://www.youtube.com/watch?v=Ou4o0prpRh4&feature=youtu.be>

So for this to work, you need a USB Sound card, they are cheap enough, but does add additional cost to this already expensive solution.

From the video, it looks like it works very well, unfortunately, I haven’t gotten to the point of being able to test it out.

The benefit I can see to using a YUN is that it does “nativtly talk” to the Arduino side, so no additional programming would be required on the YUN/Linux side of things.

For this one the PROS and CONS are almost the same.

PROS: programming maybe easier, setup maybe easier.

CONS: COST, setup may take a long long time. COST.

From Lowest To Most:

1. Talkie (Free to $5)
2. ISD1820/ISD1700 (<$6)
3. C.H.I.P. Computer ($9.00)
4. MP3 Player (<$10)
5. WT588D (<$10) (Could be more if you need the programmer)
6. RoboVoice (about $17)
7. “Low Cost Chinese Boards” (<$28)
8. Raspberry PI (Ranges but generally <$40)
9. Parallax Emic 2 (about $60)
10. Arduino YUN (>$60) (unless you are very lucky to find one cheap)
11. Android (Ranges Depending on how you count a pre-owned Android phone or tablet) The cost could be FREE or HUNDREDS.

\*\* Prices are subject to change for no good reason, generally I placed the lowest cost I could find here, so they maybe more for a solution you find.

2nd Disclaimer if you made it this far. I didn’t count the cost of speakers (or getting power to speakers) As almost all of the solutions here require them (with the exception of a few), this would be an additional cost no matter what. Generally a pair of computer speakers (5v powered) and a 5v regulator can be found for under $10.00.

\*\*\* I use a car accessory usb (charger) to power my speakers with no problems, my speakers are usb powered, and cost $6.00 \*\*

Updated Apr 26, 2017

Came across a VOICE RSS (tts) API, I don’t know if it is useful for the arduinos/microcontrollers. It does looks like it can return a wav file, but I’m not sure what to do with it after I have it.

<http://www.voicerss.org/api/documentation.aspx>