The INPs and OUTs of Talker/80 -

A Modern Voice Synthesizer for the TRS-80

Michael Wessel

October 24th

Talk Outline

- What…
  - Is Talker/80, and how do I get one?
    - Project Design Goals
    - Hardware Overview
    - Features, Demo
    - Background
    - (DIY) Resources
- How…
  - Do you program it (external perspective)
  - Does it work (internal perspective)
- Why… [BACKUP SLIDES]
  - Did I make it?
  - Personal Background

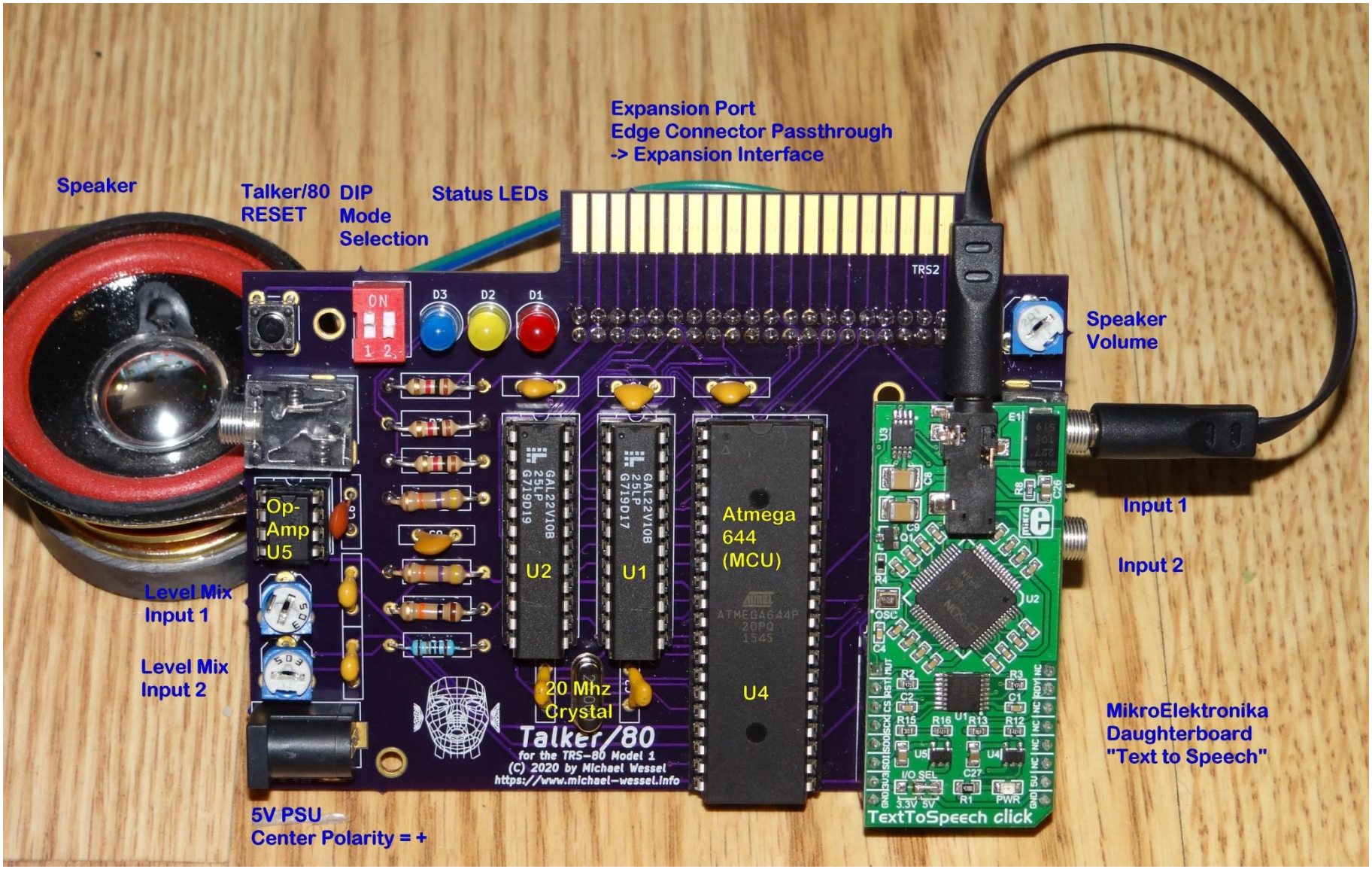
„What“ Part

What is Talker/80 ?

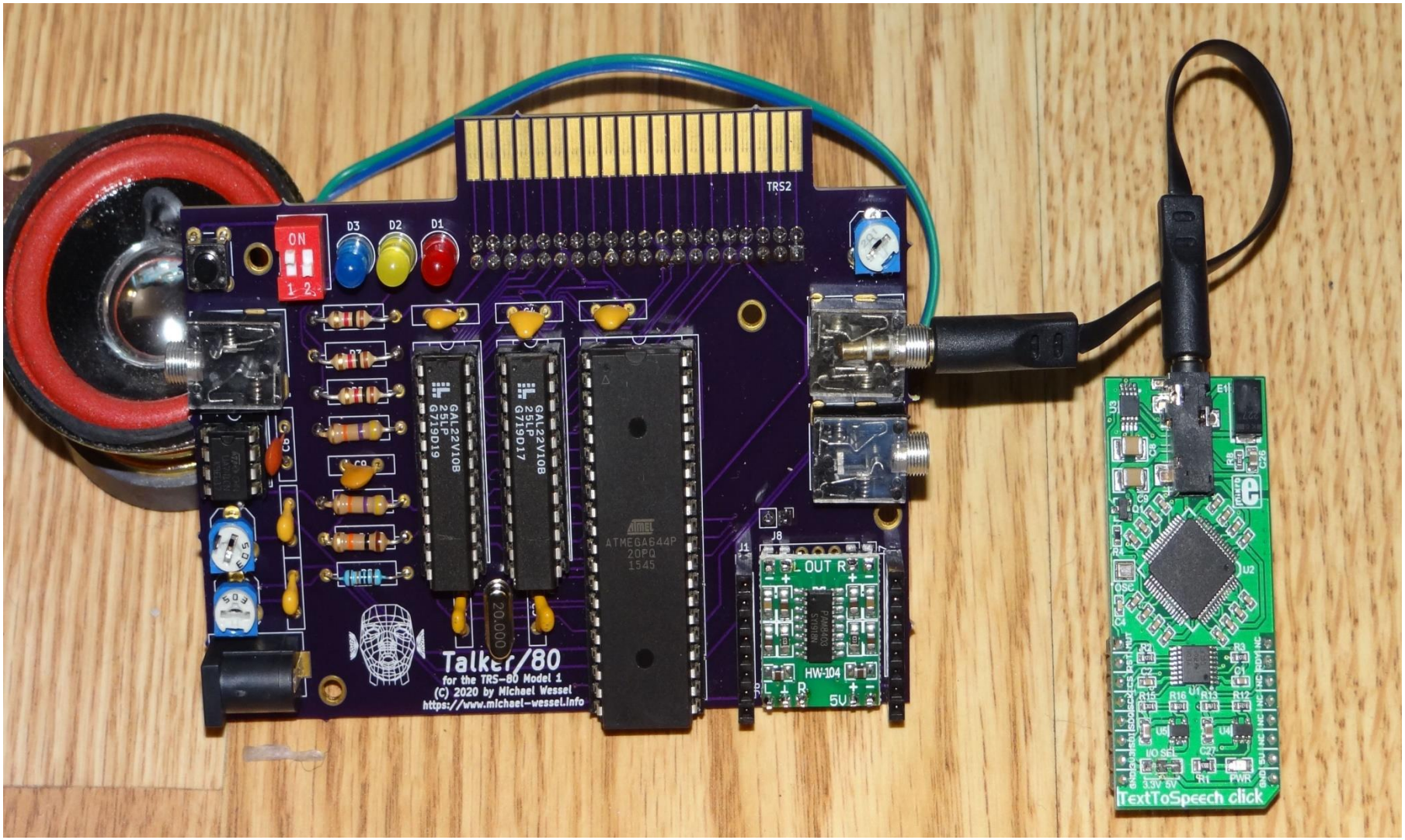
- In a nutshell: A New Voice Synthesizer for the TRS-80 Model 1, III & 4
  - Not a „clone“or „reimplementation“ of existing designs
  - Some unique new features
  - A blend of „new & old“
- Design Goals
  1. (Almost) Natural Sounding Speech (ALMOST „Alexa Quality“)
  2. Easy to program from BASIC (no machine code drivers needed)
  3. DIY / Maker-friendly (no SMD, no Unobtainium like the **VOTRAX SC-01A**, …)
  4. Compatible with Expansion Interface, FreHD, & Quinterface
  5. Expansion Port Passthrough Connector
  6. Speaker & amplifier
  7. Not for profit / OpenSource: GPL3 License
- Stretch Goals
  - TRS Voice Synth Emulation (VOTRAX SC-01A)
  - VS-100 Alpha Products Emulation (dito)
  - Existing voice synthesizer software should run, as compatible as possible (but not necessarily sound the same)

Talker/80 Model 1 (May 2020)

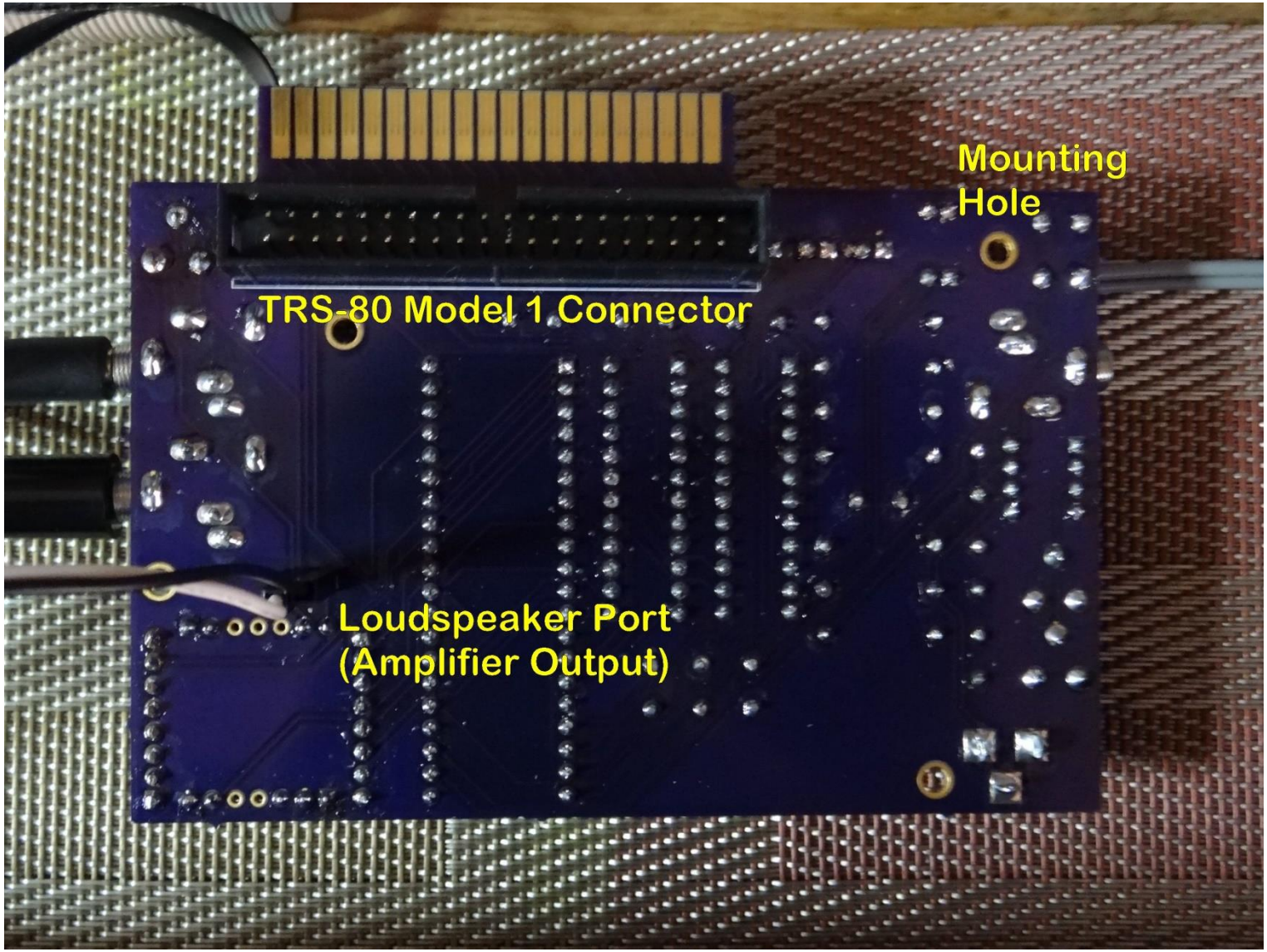
Talker/80 Model 1



Talker/80 Model 1



Talker/80 Model 1



DIP Switch & LEDs

The DIP switches are used for initial boot / startup mode selection (note that the mode can be changed later by sending a control byte); for switch 1, 2 from left to right:

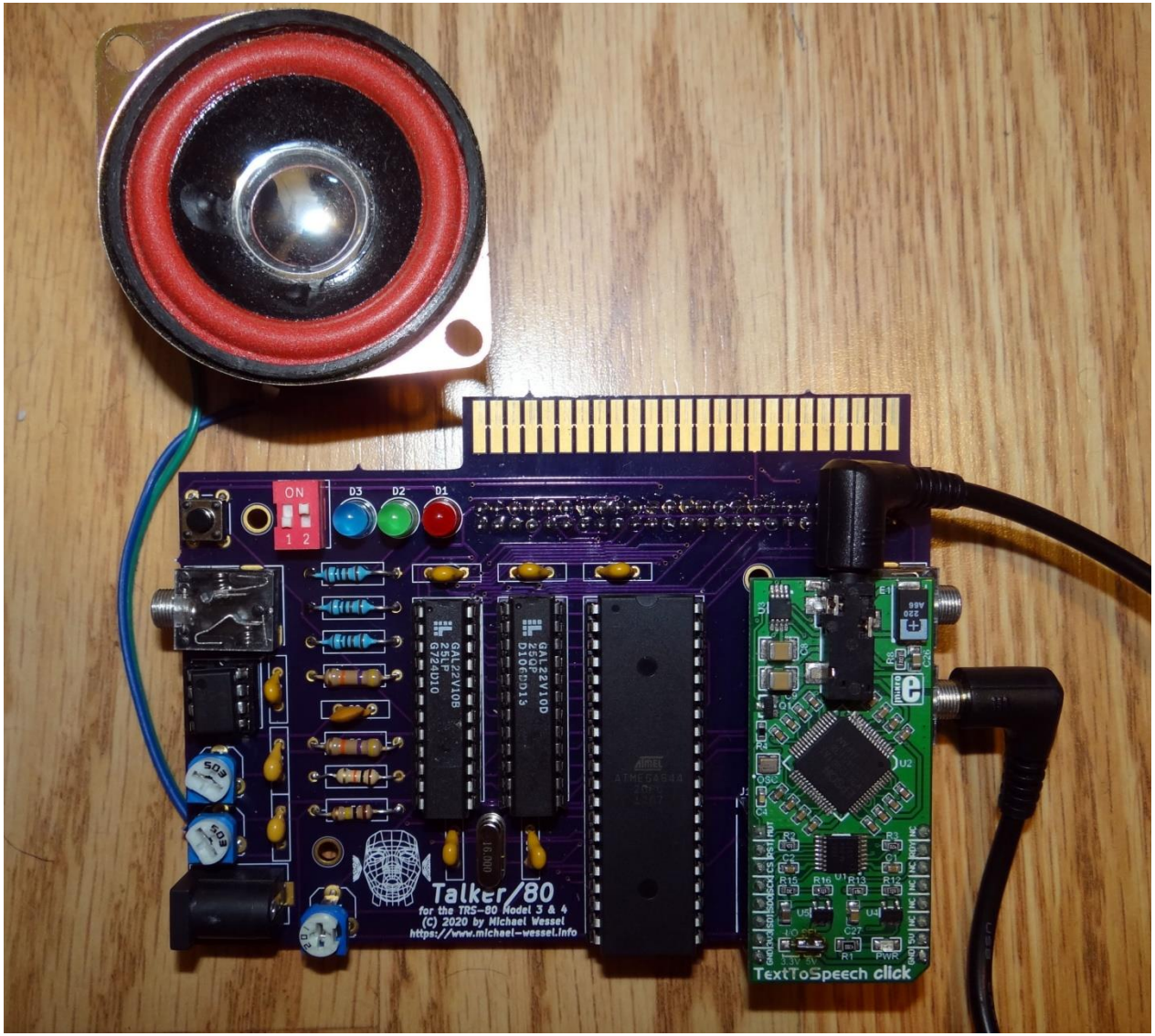
- **On On**: EPSON Mode.
- **On Off**: TRS Voice Synthesizer Mode.
- **Off On**: VS-100 Voice Synthesizer Mode.
- **Off Off**: DECtalk Mode.

The LEDs have the following meaning - during startup and after mode selection, they briefly indicate the selected mode. The blink in case of errors. In normal operation, they have the following meaning:

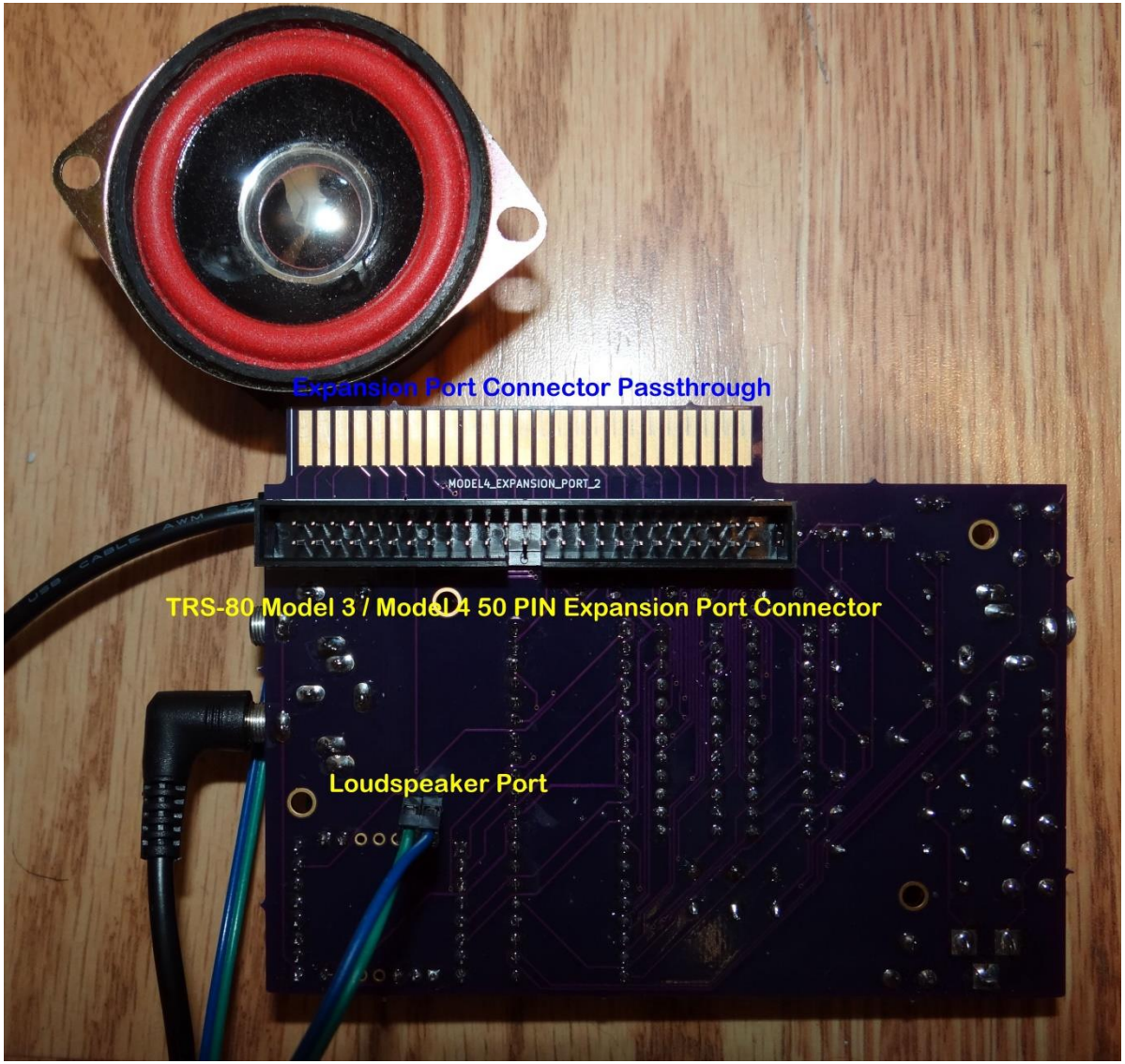
- **D1**: Incoming data.
- **D2**: On if a single phoneme is being uttered, and when the "stop speech" command is being processed.
- **D3**: Actively speaking.

Talker/80 Model III & 4 (July 2020)

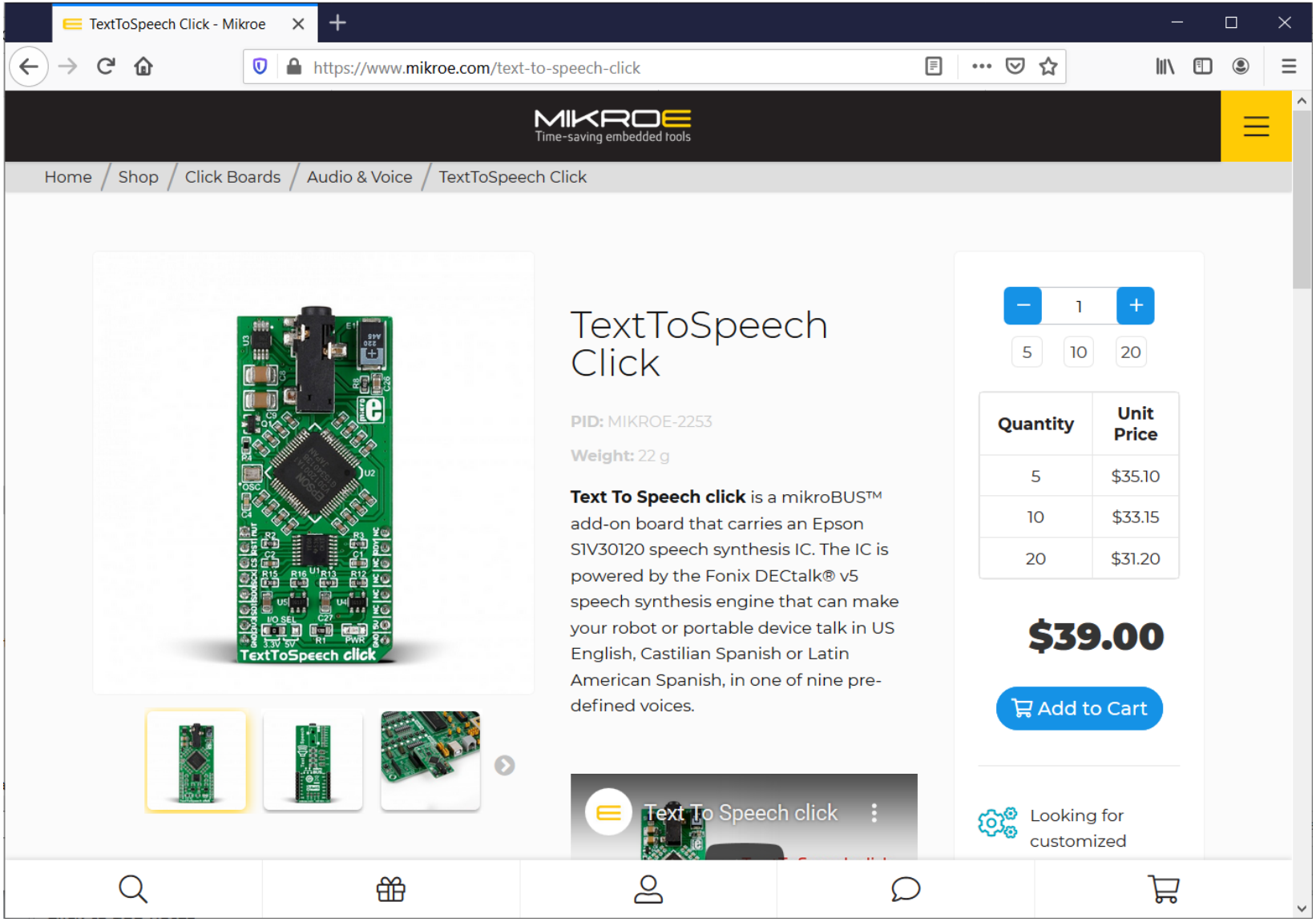
Talker/80 Model III & 4



Talker/80 Model III & 4



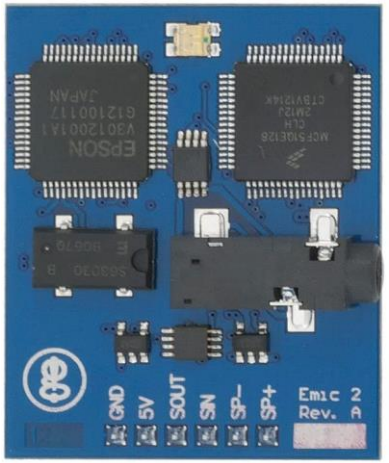
MikroE TextToSpeech Click



MikroE Speech Daughter Board
5V SMD Jumper Mod

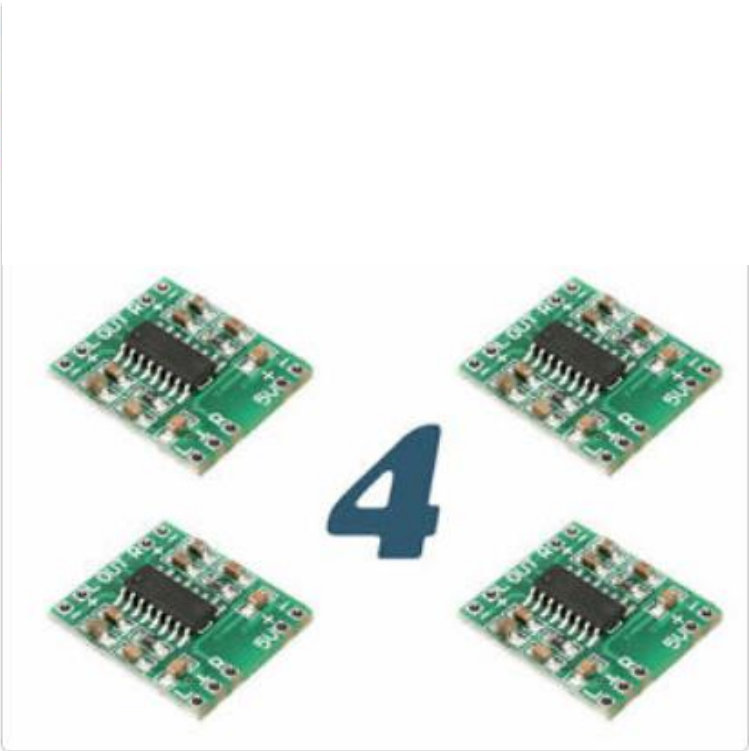
Speech Chip Epson S1V30120

- **Epson S1V30120** speech chip runs **DECtalk v5**
  - SPI (Serial Peripheral Interface)
- Unlike other speech chips, the S1V30120 is **not easy to use:**
  - 32 KB firmware image needs to be uploaded over SPI
  - This is why we need a big MCU for Talker/80
  - Complicated SPI protocol for communication
- Other more DIY-friendly S1V30120 boards exists
  - EMIC 2 from Parallax has a UART Interface, but 59 $
  - Additional TMS MCU -> Easy to use via UART (TX, RX)
- Btw – there are MANY more neat little click! boards from Mikroelektronika
  - MP3 Player, OLEDs, Sensors (Weather), Clocks, WIFI, …
  - Either I2C („I-squared-C“), SPI, or UART
  - Future project: Generic Mikrobus plug&play board for the TRS-80?



Alternative S1V30120 Board: **Emic 2** from Parallax (Grand Idea Studios)

PAM8403 Amplifier Board (Instead of an LM386 Circuit…)



4PCS PAM8403 Mini Digital Power Amplifier Board Class D 2*3W 2.5-5V input USA

Condition: New

Quantity: 1 More than 10 available

19 sold / See feedback

Price: **US $4.34**

Buy It Now

Add to cart

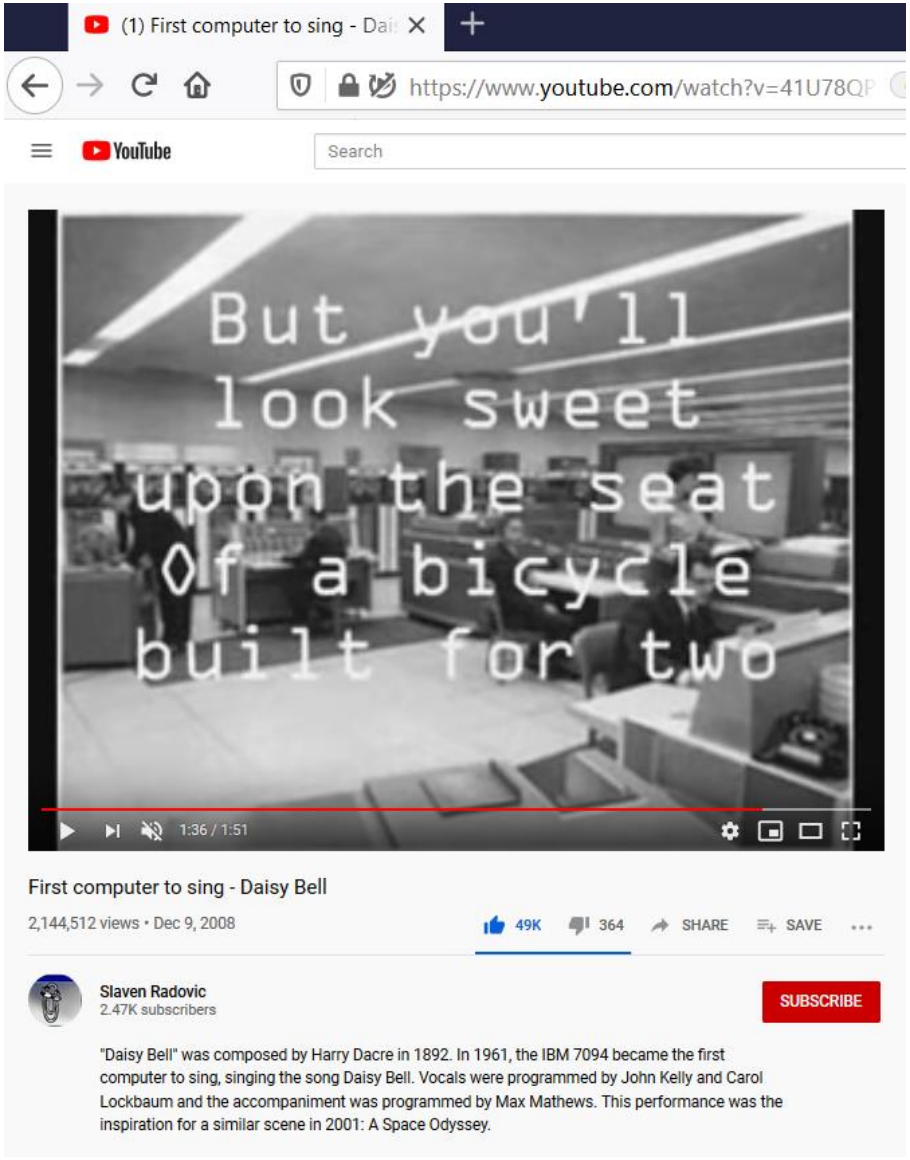
Feature Overview Part 1
„Native“ Modes: DECtalk & EPSON

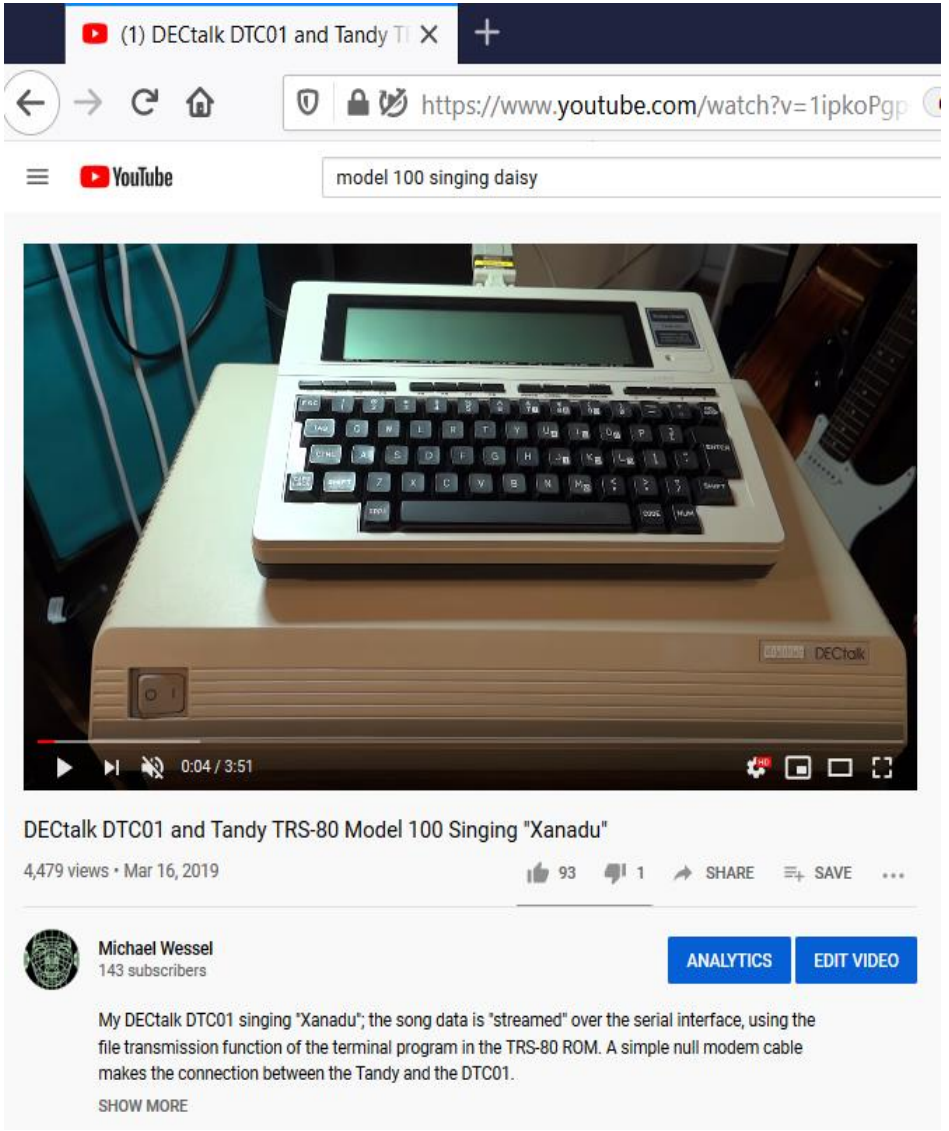
- High Quality, almost natural sounding speech
  - DECtalk mode (DECtalk v5)
  - Epson mode is a more user friendly DECtalk mode
  - English & Spanish (Epson mode)
  - Singing (DECtalk mode)
    - Unfortunately, many songs from „The Flames of Hope“ require changes, because they are for older DECtalk versions
  - Different
    - Voices, Pitches, Volumes, Speeds, …
    - All modes (below)
  - Note: DECtalk & Stephen Hawkings voice: „Klatt Speech Synthesizers“

Demo DECtalk / EPSON Mode

- `WELCOME.BAS`
- `ENGLISH.BAS`
- `SPANISH.BAS`
- The first singing computer was an IBM 7094 (1961): „Daisy Bell“
  - `OUT 11,225`
- `ELIZA80.BAS`

Singing Computers / Inspiration for HAL 9000 Shutdown Scene: Daisy Bell - IBM 7094 (1961) & Xanadu - Model 100 (2019)





History Corner – The Master Mind

Dennis H. Klatt - Wikipedia

https://en.wikipedia.org/wiki/Dennis_H._Klatt

Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute

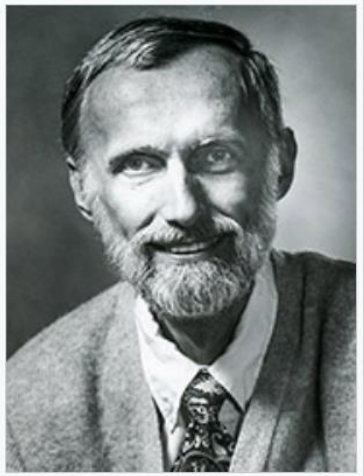
Help
Learn to edit
Community portal
Recent changes
Upload file

Dennis H. Klatt

From Wikipedia, the free encyclopedia

Dennis H. Klatt (March 31, 1938 – December 30, 1988) was a noted American researcher in speech and hearing science. Klatt was the pioneer of computerized speech synthesis and created an interface which allowed for speech for non-expert users for the first time. Prior to his work, non-verbal individuals would need specialist support to be able to speak at all.[1] The DECtalk speech synthesizer, best known as Stephen Hawking's voice, was based on Klatt's own voice.[2][3][4]

Klatt was born in Milwaukee, Wisconsin, on March 31, 1938. He received the B.S. and M.S. degrees in electrical engineering from Purdue University, West Lafayette, Indiana, in 1960 and 1961, respectively, and the Ph.D. degree in communication sciences from the University of Michigan, Ann Arbor, in 1964. He joined the Massachusetts Institute of Technology as an assistant professor in 1965, becoming a



Dennis H. Klatt

Feature Overview Part 2
Emulated Modes: TRS & VS-100 Synths

- Classic Voice Synthesizer Emulations
  - TRS Voice Synth
  - VS 100
  - „Goodies“, imperfect, „add ons“
- Compatible with existing voice synthesizer software
  - Model 1
    - „Talking Eliza“
    - TRS Voice Synthesizer Demo Program
  - Model 1 & Model III / 4
    - VS-100 Driver & Demo Software
  - Question
    - Any other good „standard“ voice synthesizer software I should know about? More test cases?
- Audio Mixer Demo

TRS-80 Classic Voice Synthesizers
See TrashTalk Episode 2 for Info!

Trying to get one of these desperately!
(But don‘t want to pay 400 $)

Demo TRS Voice Synthesizer / VS-100 Mode

- TRS Voice Synthesizer Demo Program
- VS-100 Demo Program
  - VS48.CMD doesn‘t work with FreHD boot ROM
  - Use floppies

Audio Mixer Demo

- `FROGGER1.CMD`

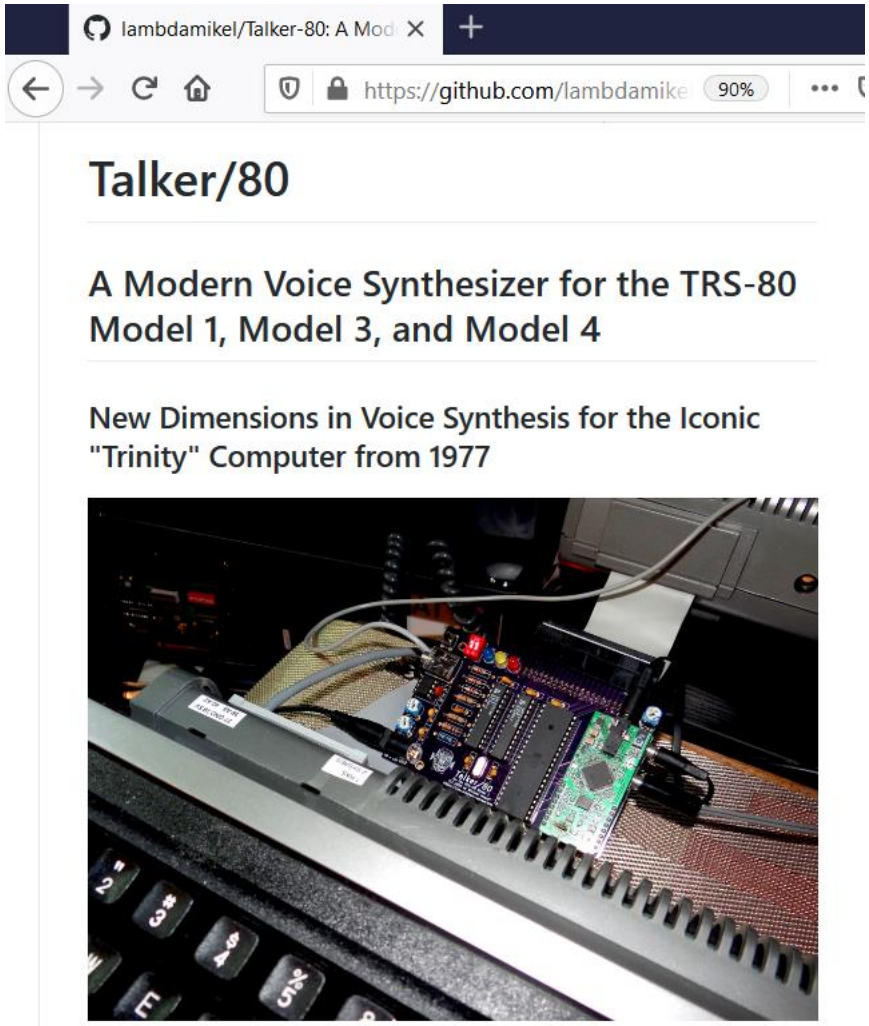
How Part

HOW can I get one???

- DIY
  - From sratch: Github resources **https://github.com/lambdamikel/Talker-80**
  - eBay: DIY Kit with programmed chips
  - „#SepTandy“ Model III / 4 assembly video **https://youtu.be/yHVgSqcxh_4**
- Assembled
  - eBay (occassionally), or drop me a mail. Search for „Talker/80“ on eBay
  - Vintage Computer Forum

Github Project Repository

- Source Code
- HEX / JED Files
- Gerbers
- BOM
- Software
- PDF Manuals
- Documentation
- Schematics
- …

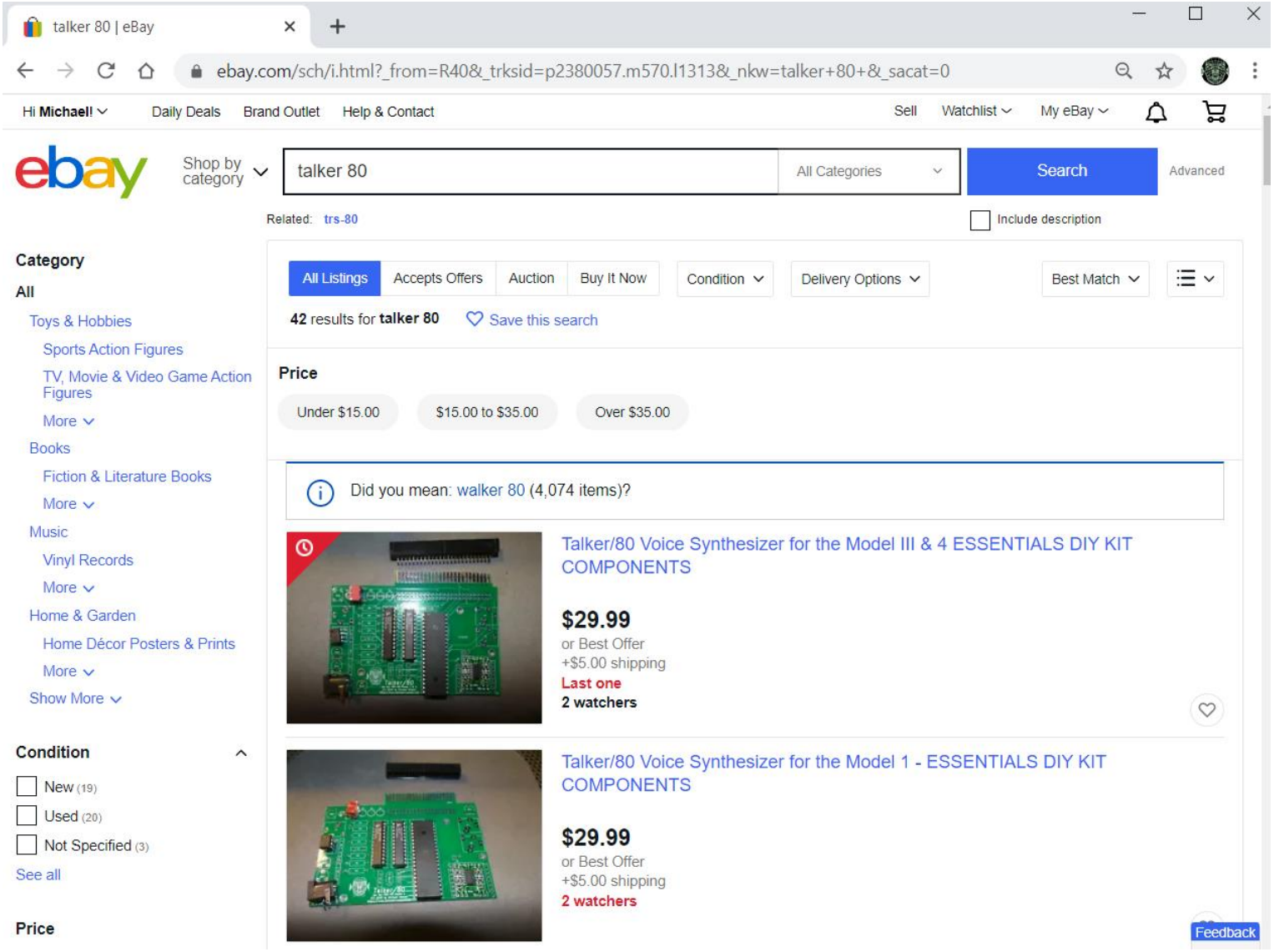


https://github.com/lambdamikel/Talker-80

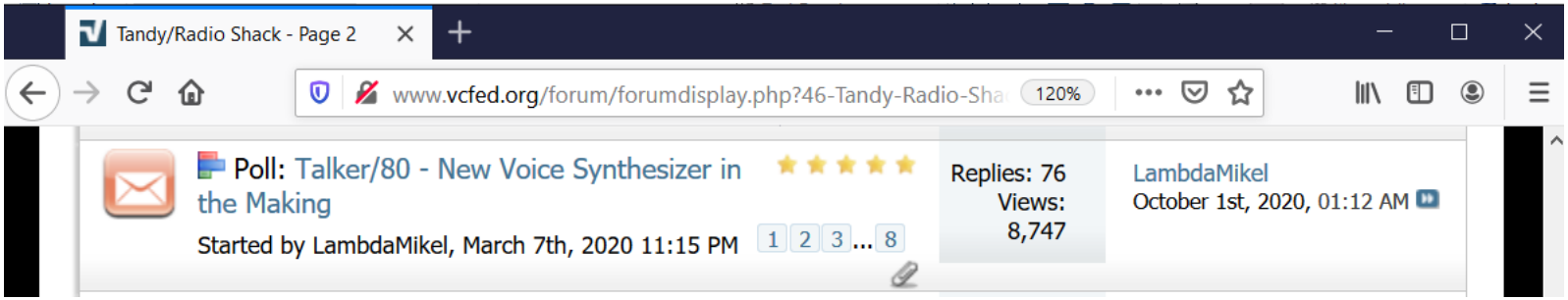
YouTube – #SepTandy Video

https://youtu.be/yHVgSqcxh_4

eBay- DIY Kits



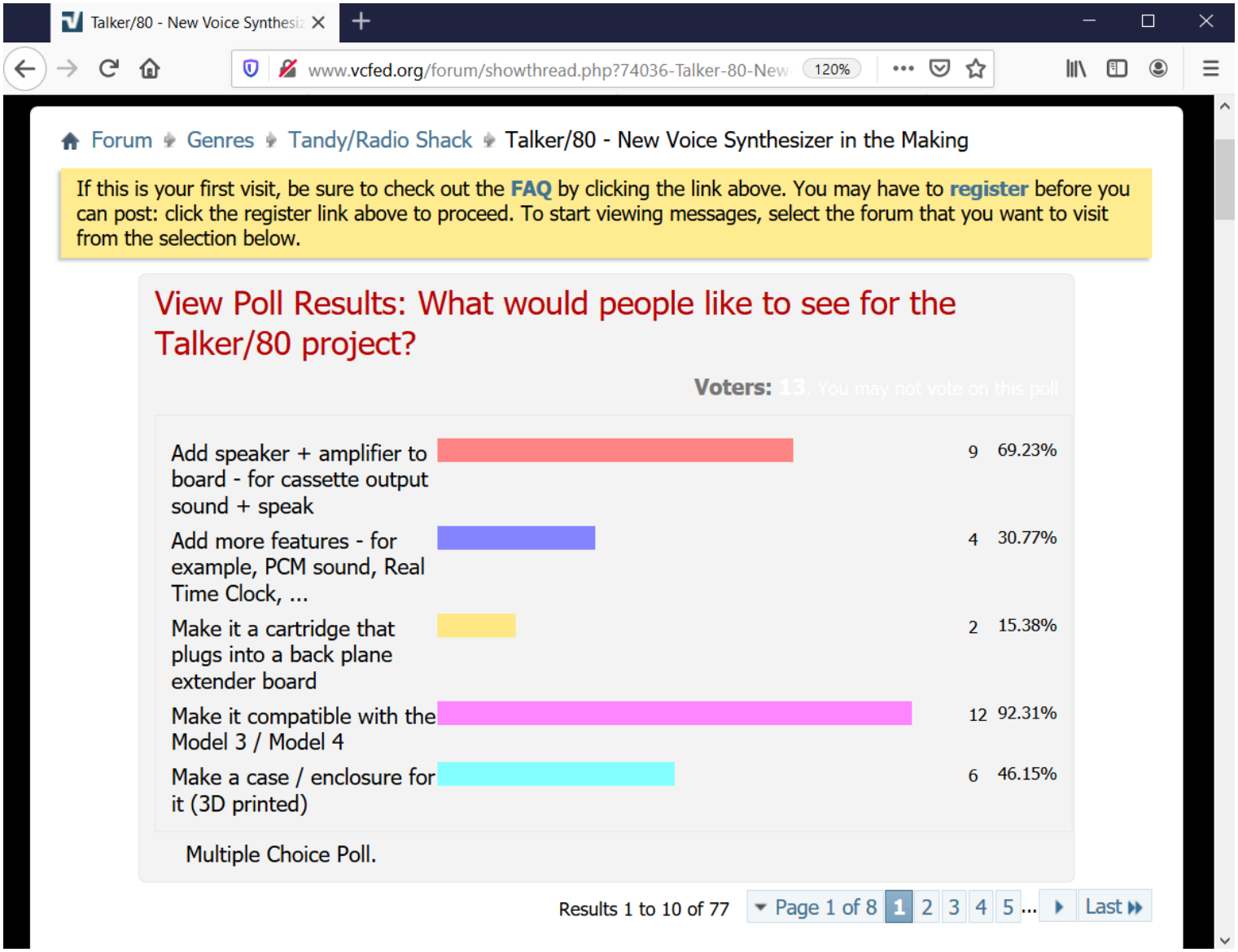
Reception by the Community



FEEDBACK	FROM
The Talker/80 is a great product. Works perfect. Would recommend! Talker/80 Voice Synthesizer for the TRS-80 M III/4 + Cable, Disk, Deluxe Speaker (#193679838973)	Buyer: azdesertdweller (12★) US $99.99
Excellent product, works very well. Talker/80 Voice Synthesizer for the TRS-80 Model 1 (#193454931652)	dittman1 (881★) US $80.80
Great product & seller. Thanks for supporting the classic M1. Talker/80 Voice Synthesizer for the TRS-80 Model 1 - Last Chance (#193565612155)	Buyer: caryg0 (466★) US $80.80
Excellent transaction! Thank you. AAA+ Talker/80 Voice Synthesizer for the TRS-80 M III/4 + Cable, Disk, Deluxe Speaker (#193633916335)	Buyer: gulfracing (4785★) US $99.99

Thank you all - very gratifying!

Community Input



Community Reception

Talker/80 - New Voice Synthesi

www.vcfed.org/forum/showthread.php?74036-Talker-80-New-Voice- 120%

Reply With Quote

August 31st, 2020, 04:45 AM #65

Avia
Senior Member

Join Date: Jan 2016
Location: Saint Augustine, FL
Posts: 194

Just a quick shout out to Michael and his Talker/80 for the M1. The product is great and his after sale support is excellent. Given the rarity of the original RS device and the astronomical prices when one does come up (not to mention possible reliability after all these years), I never thought I would hear my M1 speak.

A product this cool deserves a neat set of clothes. Having a 14 year with a 3D printer was all that was needed.

Overall I am very impressed with the Talker/80 - a great addition to the M1 ensemble.

Cary

Talker80sm.jpg

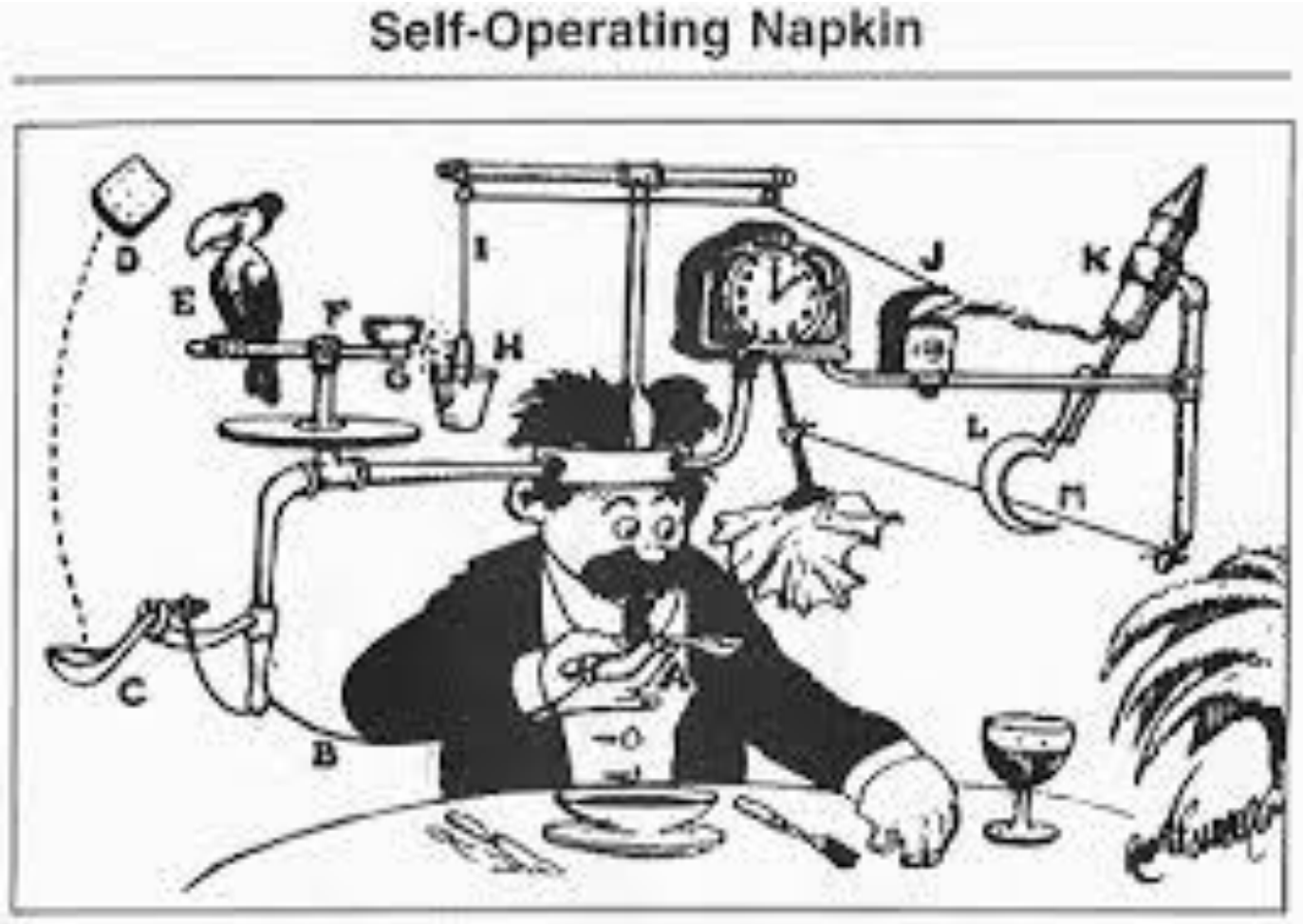
8-Bit Family: Epson QX-10 w/ ComFiler 10MB HD | Kaypro 2X | TRS 80 Model I | TRS 80 CoCo 1,2 & 3 | TRS 80 M4 & 4D

Reply With Quote

Talker/80 Case Made by Cary & Logan

**Occasionally,
folks ask me for this,
and I always refer
them to Aviva on VCF**

Getting Technical…



Talker/80 Programming - 1

- Start by looking at `WELCOME.BAS` on the disk
- Simple Text-to-Speech – No Phonemes!

```
120 A$="HELLO, I AM TALKER 80. NICE TO MEET YOU! WHAT'S YOUR NAME?"
130 GOSUB 590
```

Next Slide

- Very easy to program: I added Talker/80 speech to Jeff Shrager's ELIZA in 30 minutes (`ELIZA80.BAS`)
- DECtalk Singing – ARPABET Phoneme Syntax

[Phoneme mode on] [Speakrate 200] [Voice :n1] [Phoneme<Pitch,Duration> …]

```
210 A$="[:phone arpa speak on][:rate 200][:n1][hxae<300,10>piy<300,10> brr<600,22>th<100>dey
<600,19>dih<600,15>rdeh<600,14>ktao<600,12>k_<120>_<120>][:n1]"
220 A$=A$+"[:phone arpa speak on][:rate 200][:n3][hxae<300,10>piy<300,10> brr<600,22>th<100>dey_
<120>_<120>][:n3]"
230 GOSUB 590
```

- DECtalk / ARPABET is NOT so easy… **RTF(DECtalk)M**
- **Great resource for DECtalk songs: The Flames of Hope**
  http://www.theflameofhope.co/DECTALK.html

Talker/80 Programming - 2

- EPSON & DECtalk modes: `OUT 11,X`
- `INP(11)` Ready Signal
- Output Routine

```
595 FOR I=1 TO LEN(A$)
600 C$=MID$(A$,I,1)
610 X=ASC(C$):OUT 11,X
620 NEXT
630 OUT 11,13
640 PRINT:PRINT A$
650 A=INP(11)
660 IF INKEY$=" " THEN OUT 11,254:PRINT "quiet!":PRINT
670 IF NOT A AND 192 THEN GOTO 650
680 RETURN
```

Output ASC of A$ character by character

Carriage Return (13) initiates Text-to-Speech from Buffer

Ready Signal: 192 = 0b11000000

Control (Command) Bytes – General

- Control Bytes in Epson / DECtalk mode
  - DECtalk / EPSON ASCII is 7 Bit
  - Bytes with 8th Bit set (Bit #7) are Control Bytes

Control Byte	Description
0xFF / 255	Reset Talker/80. DIP settings determines mode.
0xFE / 254	Immediately stop speaking.
0xFD / 253	Disable Talker/80 until hardware reset.
0xFC / 252	Pull Z80 WAIT states while speaking.
0xFB / 251	**Don't pull Z80 WAIT states while speaking.**

Control Bytes – Mode Control

Control Byte	Description
0xEF / 239	**Enable EPSON mode.**
0xEE / 238	Enable DECtalk mode.
0xED / 237	Enable TRS Voice Synth mode. "?"-based segmentation of input.
0xEC / 236	Enable TRS Voice Synth mode 2. Timer-based segmentation of input.
0xEB / 235	VS-100 mode.
0xEA / 234	**Enable audible command confirmations (for control bytes).**
0xE9 / 233	Disable audible command confirmations (for control bytes).
0xE8 / 232	**Enable English.**
0xE7 / 231	Enable Castilian Spanish.
0xE6 / 230	Enable alternate VS-100 / TRS Voice Synth pronounciation.
0xE5 / 229	**Use normal VS-100 / TRS Voice Synth pronounciation.**

Control Bytes – Fun & Testing

Control Byte	Description
0xE4 / 228	Announce current mode.
0xE3 / 227	Speak copyright info.
0xE2 / 226	Quote HAL9000.
0xE1 / 225	DECtalk singing demo - "Daisy".
0xE0 / 224	Speak current version number.

Control Bytes – Voice Settings

Control Byte	Description
0xD0 / 208	**Use default pitch of voice.**
0xD1 - 0xDF / 209 - 223	Use corresponding voice pitch.
0xC0 / 192	**Use default voice.**
0xC1 - 0xCF / 193 - 207	Use corresponding voice. Not all voices are defined.
0xB0 / 176	**Use default volume.**
0xB1 - 0xBF / 177 - 191	Use corresponding voice volume.
0xA0 / 160	**Use default speak rate.**
0xA1 - 0xAF / 161 - 175	Use corresponding speak rate.

Other Modes – „Goodies“ / Add-Ons

- Control Bytes not supported in these modes
  - Send Control Bytes for voice adjustments while in Epson / DECtalk mode, then switch to VS-100 or TRS Voice Synthesizer Mode
- VS-100 Emulation Mode
  - IO over Port 11, and same „ready“ signal
  - Pitch Control Bits not supported (impossible with DECtalk?)
- TRS Voice Synthesizer Emulation Mode
  - Model 1
    - Memory-Based IO: VIDEO RAM Snooping (last 32 Screen Characters) `0x3FE0 ... 0x3FFF`
    - Speech window open / close via `PRINT("?")`
    - READY Signal on Port 11, but not used by Voice Synth Software
  - Model III / 4
    - No Memory-Based IO, Using Port 11 again
    - **You can port existing BASIC TRS Voice Synthesizer Software by replacing `PRINT <X>` with `A$=<X>:GOSUB <SPEECHROUTINE>`**

How Does it Work?

- Two phoneme mapping tables
  - Mapping of TRS Voice Synth „printed“ phonemes to VOTRAX SC-01 phonemes
  - Mapping of SC-01 phonemes to DECtalk / ARPABET phonemes
  - Check the `talker.c` firmware file ;-)
- Challenges
  - Single Phoneme Soundeffects
  - Impossible with DECtalk? (Also tried DECtalk singing)
  - Easy with a SP0256-AL2 or SpeakJet…
  - Votrax Pitch Control Bits not supported (singing didn‘t work)

Table IV. Decimal/ASCII/Phoneme Conversion

Decimal	ASCII	Phoneme Symbol	Decimal	ASCII	Phoneme Symbol
ØØ	@	A1	32	Space	PA1**
Ø1	A	AH2	33	!	12
Ø2	B	B	34	“*	I
Ø3	C	CH	35	#	I3
Ø4	D	D	36	$	OO
Ø5	E	E1	37	%	OO1
Ø6	F	F	38	&	Y
Ø7	G	G	39	‘	U
Ø8	H	H	4Ø	(	IU
Ø9	I	I1	41	)	A2
1Ø	J	J	42	*	AY
11	K	K	43	+	NG
12	L	L	44	,	AW
13	M	M	45	-	Ø DEC.
14	N	N	46	.	E
15	O	O1	47	/	ER
16	P	P	48	Ø	PAØ**
17	Q	DT	49	1	AW1
18	R	R	5Ø	2	AW2
19	S	S	51	3	EH1
2Ø	T	T	52	4	EH2
21	U	U1	53	5	EH3
22	V	V	54	6	UH1
23	W	W	55	7	UH2
24	X	ZH	56	8	UH3
25	Y	Y1	57	9	AE1
26	Z	Z	58	:*	AE
27	↑	O2	59	;	AH1
28	↓*	O	6Ø	<	THV
29	←*	AH	61	=	TH
3Ø	→*	A	62	>	SH
31	–*	Null	63	?	open /close window

* Cannot directly input these characters from keyboard.
**PA1 and PAØ are pauses which result in silence.

SC-01 Phonemes

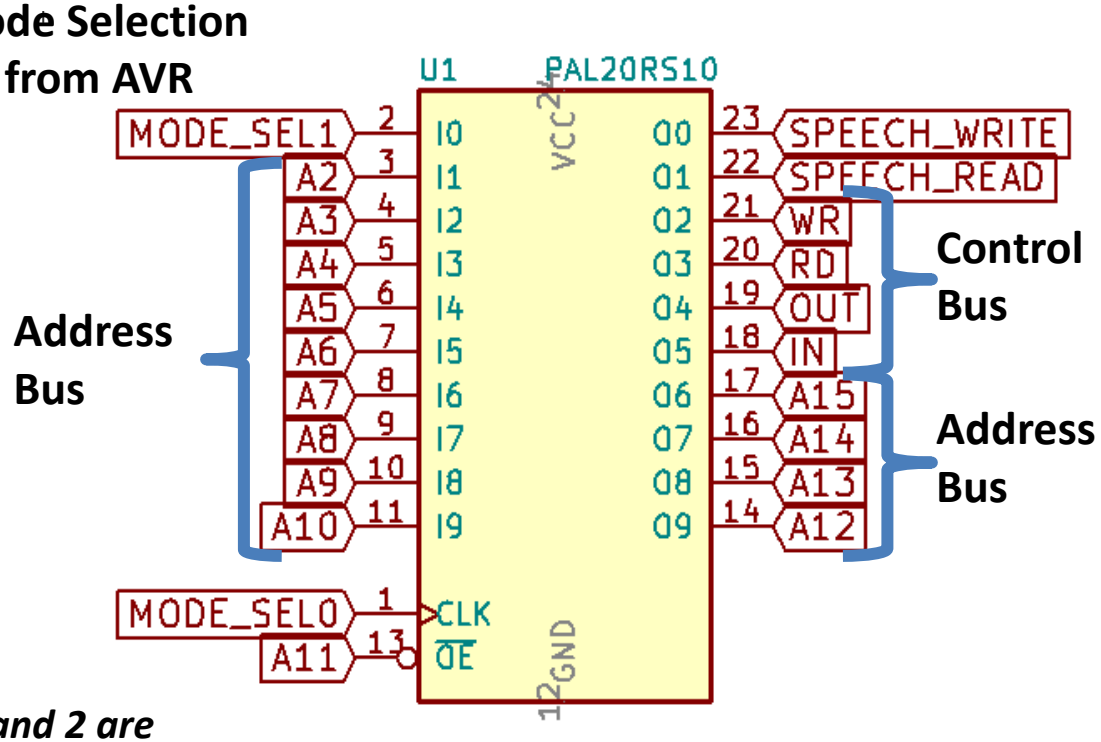
Phoneme Code	Phoneme Symbol	Duration (ms)	Example Word
00	EH3	59	jacket
01	EH2	71	enlist
02	EH1	121	heavy
03	PA0	47	no sound
04	DT	47	butter
05	A2	71	made
06	A1	103	made
07	ZH	90	azure
08	AH2	71	honest
09	I3	55	inhibit
0A	I2	80	inhibit
0B	I1	121	inhibit
0C	M	103	mat
0D	N	80	sun
0E	B	71	bag
0F	V	71	van
10	CH*	71	chip
11	SH	121	shop
12	Z	71	zoo
13	AW1	146	lawful
14	NG	121	thing
15	AH1	146	father
16	OO1	103	looking
17	OO	185	book
18	L	103	land
19	K	80	trick
1A	J*	47	judge
1B	H	71	hello
1C	G	71	get
1D	F	103	fast
1E	D	55	paid
1F	S	90	pass

Phoneme Code	Phoneme Symbol	Duration (ms)	Example Word
20	A	185	day
21	AY	65	day
22	Y1	80	yard
23	UH3	47	mission
24	AH	250	mop
25	P	103	past
26	O	185	cold
27	I	185	pin
28	U	185	move
29	Y	103	any
2A	T	71	tap
2B	R	90	red
2C	E	185	meet
2D	W	80	win
2E	AE	185	dad
2F	AE1	103	after
30	AW2	90	salty
31	UH2	71	about
32	UH1	103	uncle
33	UH	185	cup
34	O2	80	for
35	O1	121	aboard
36	IU	59	you
37	U1	90	you
38	THV	80	the
39	TH	71	thin
3A	ER	146	bird
3B	EH	185	get
3C	E1	121	be
3D	AW	250	call
3E	PA1	185	no sound
3F	STOP	47	no sound

/T/ must precede /CH/ to produce CH sound.
/D/ must precede /J/ to produce J sound.

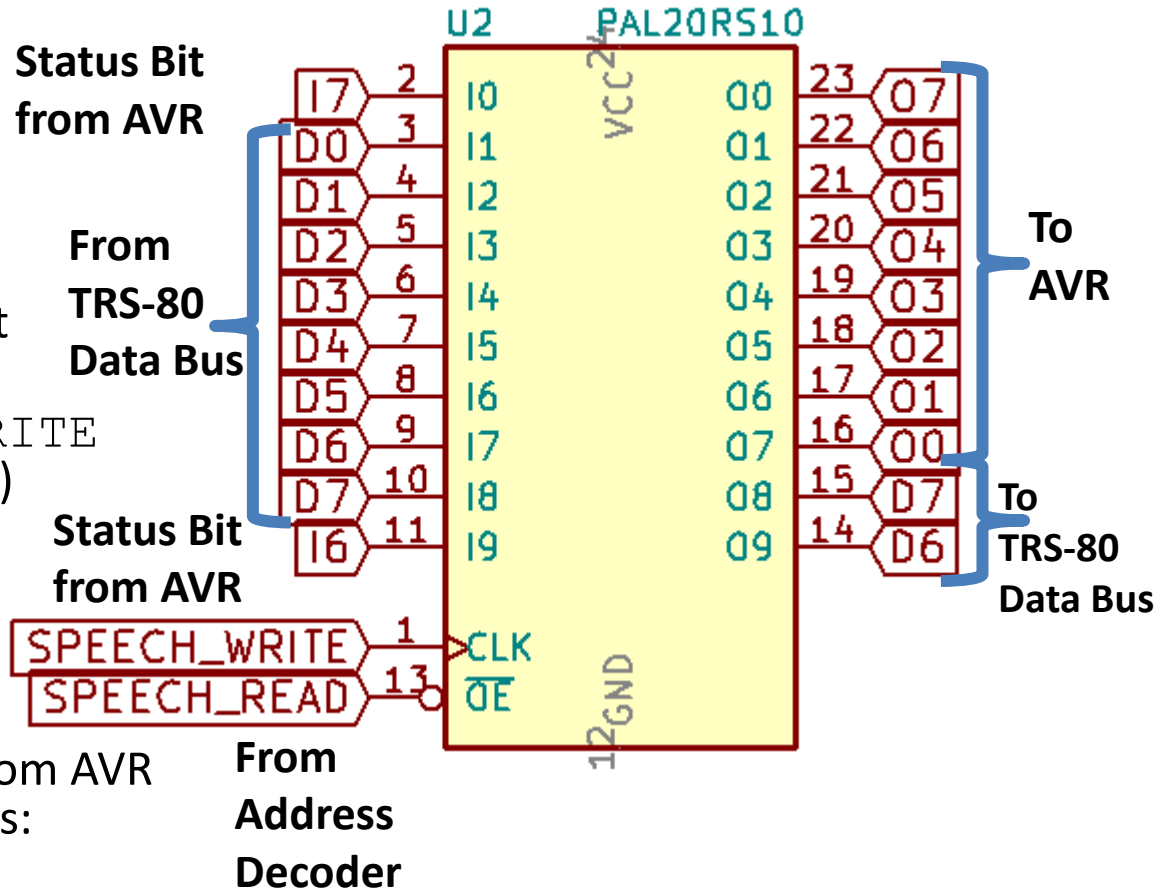
Hardware – M1 Address Decoder (U1)

- GAL22V10(B, D)
- Can be difficult to program
- WinCUPL (> 20 years old…)
- Beware of fake GALs on ebay…
- Combinational Logic
- Depending on mode
  - Video RAM-based IO
  - Port-based IO
  - Mode is input to the GAL (`MODE_SELx` lines)
  - Partial Decoding (A0, A1 don‘t care)
  - VS-100 ***“responds if bits 7, 5, 4 and 2 are 0, don't care for others” (JoeZwerko)***
  - Same for DECtalk / Epson
- Generates `SPEECH_WRITE SPEECH_READ` signals



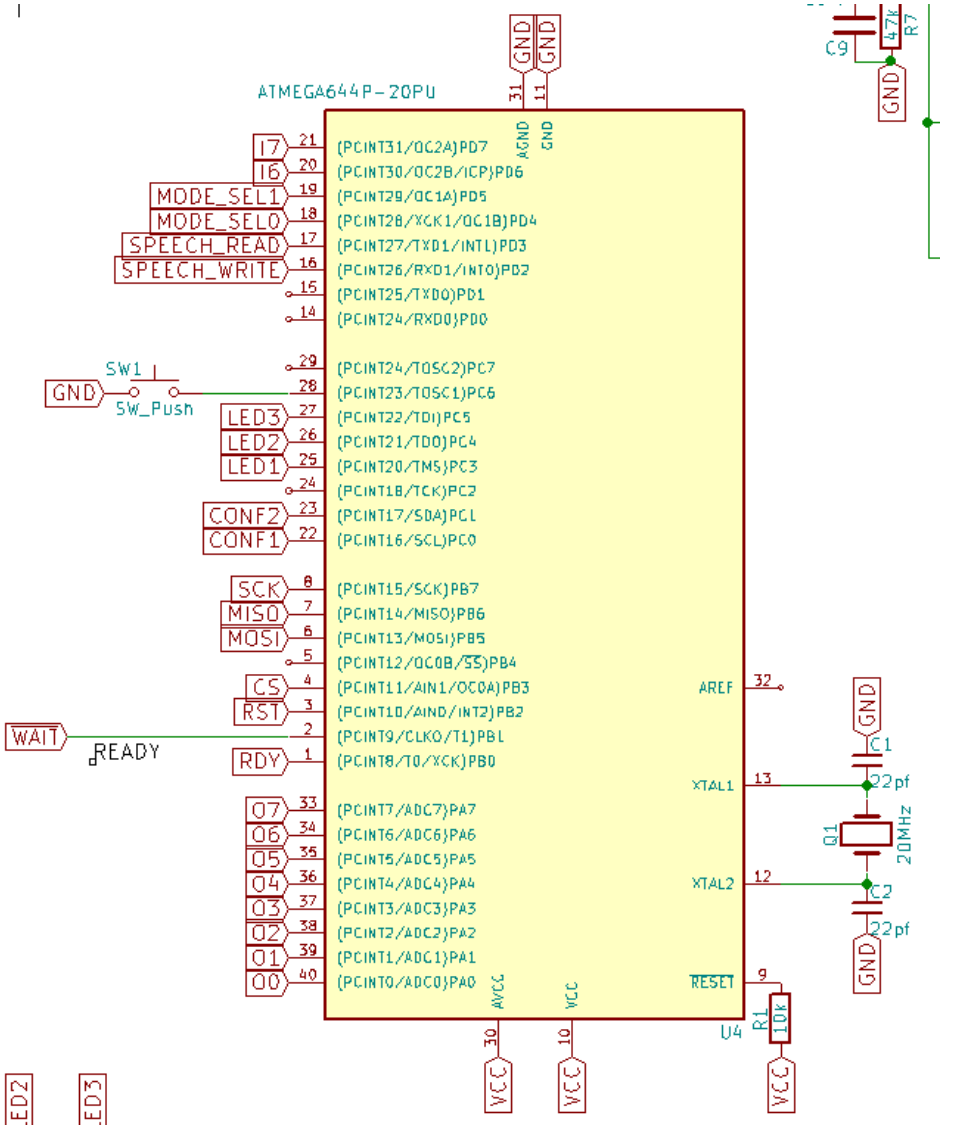
Hardware – M1 Databus Latch (U2)

- GAL22V10(B, D)
- Registered Logic
- Upon `SPEECH_WRITE`
  - Latch databus and present to AVR microcontroller
  - AVR also got `SPEECH_WRITE` (Interrupt Service Routine)
  - AVR fetches latched bytes and buffers it
- Upon `SPEECH_READ`
  - Read `I6, I7 (I9, I0)` status bits from AVR and put on TRS-80 databus: `D6, D7 (O9, O8)`
  - `O9, O8` are TriState (**Z** elsewise)
  - There is an ISR on the AVR, but not needed (`I0, I9` are constantly updated by the AVR)



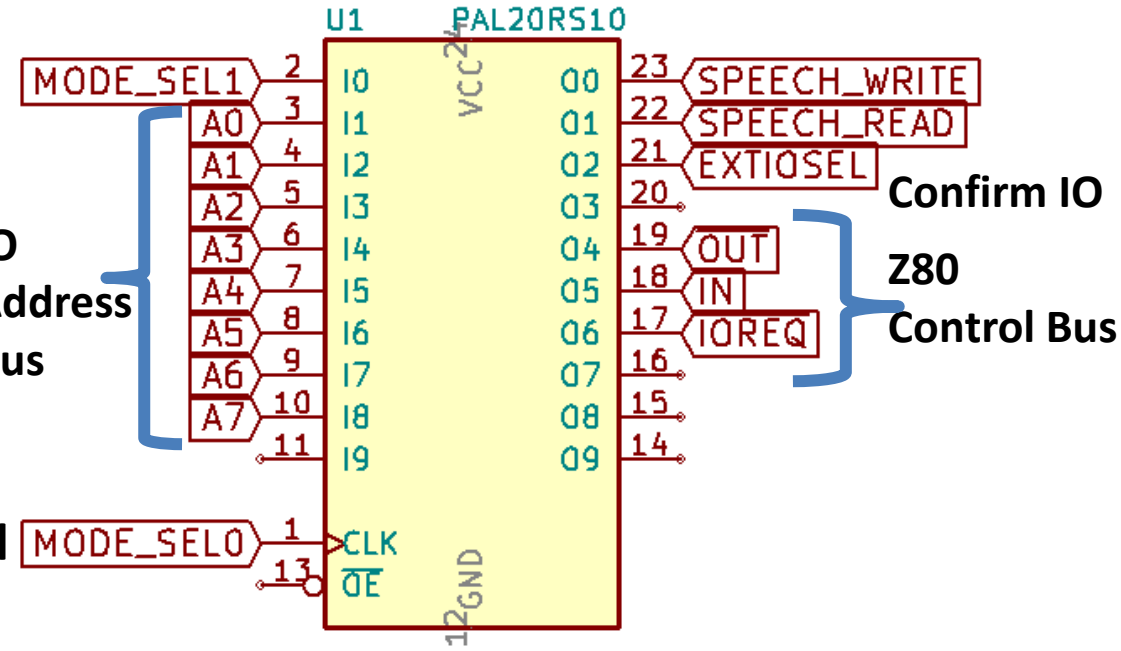
Hardware – AVR MCU (U4)

- Proven old AVR 8bit ATmega 644-20
- Talker/80 used 16 Mhz these days
  - Started with 20 MHz, but reliability problems with some crystals
- 64 KB Program ROM
  - Enough for Epson DECtalk firmware
- 4 KBs of SRAM
- SPI (-> Speech Board) and more
- Enough GPIOs – all but 3 pins ar used



Hardware – M III/4 Address Decoder (U1)

- GAL22V10(B, D)
- Simpler than the M1 version!
- Mode bits from AVR are input, but don‘t matter
- Only 8 bit IO address bus
- Full decoding for IO port 11
- Generates
- `SPEECH_WRITE`
  `SPEECH_READ`
  signals
- **U2 (Databus Latch) is identical**
- **AVR and firmware is identical**



Biggest Engineering Challenge?

The TRS Voice Synthesizer Emulation

- Main problem
  - DECtalk does not allow „realtime speech“
  - Phonemes / text gets buffered, and spoken when flushed (CR)
  - TRS Voice Synth utters Phonemes in real time:
    „The phonemes in the word compute are: K UH1 M P Y1 IU U1 T“
  - No „flush buffer“ token – only the window open/close („?“)
  - But since there is no „ready“ signal / flow control, if we wait too long with uttering phonemes, we will be too late and skip content!
    (and pulling Z80 wait states is not nice)
- Potential Solutions
  - „Single Phoneme“ detector IMMEDIATELY flushes and speaks
  - Timer solution: no new phoneme since the last <n> ms, flush & speak
  - Phonemes don‘t sound quite right if uttered „in isolation“
  - DECtalk singing allows phoneme length spec, but adds vibrato… hmm
- **The TRS Voice Synthesizer emulation is an „extra“**

Talker/80 Milestones Timeline

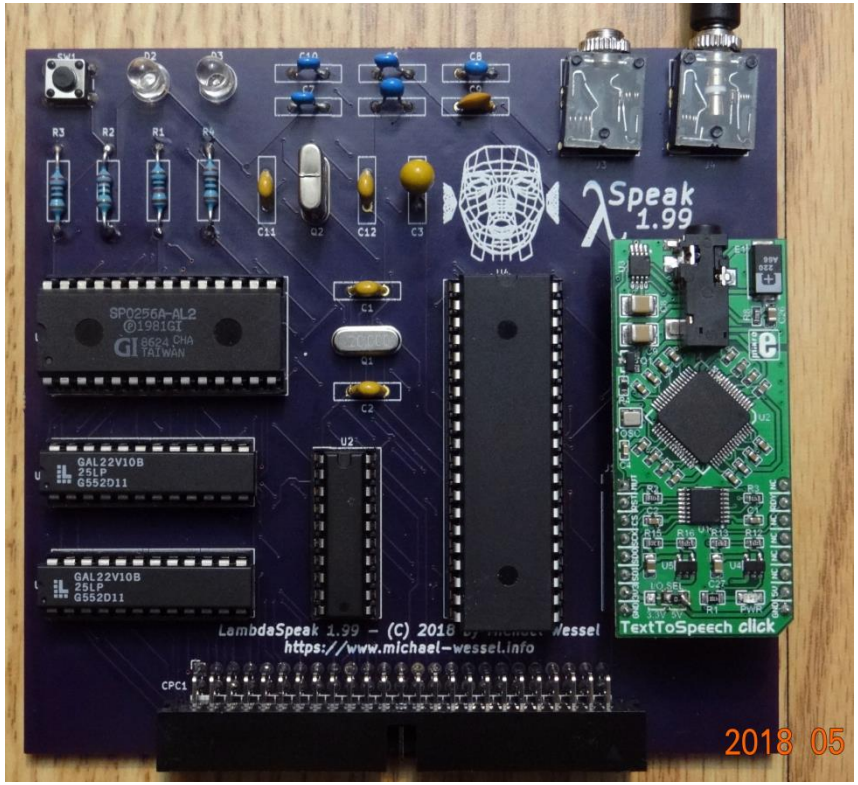
- 2/2020: My M1 is fully operational (FreHD, FreHD Bootrom, LowerCase mod)
- 3/2020: First Breadboard Protoype (DECtalk, Epson)
- 4/2020: TRS Voice Synth & VS-100 Emulation
- 5/2020: Talking Eliza & First 3 Customers
- 6/2020: TrashTalk Eric‘s Review
- 7/2020: M III / 4 Version

It took only 2 month of spare time to develop it?!

Thanks to the very similar LambdaSpeak ancestor for the Amstrad CPC, hardware development was straight forward (re-used existing bits & pieces).

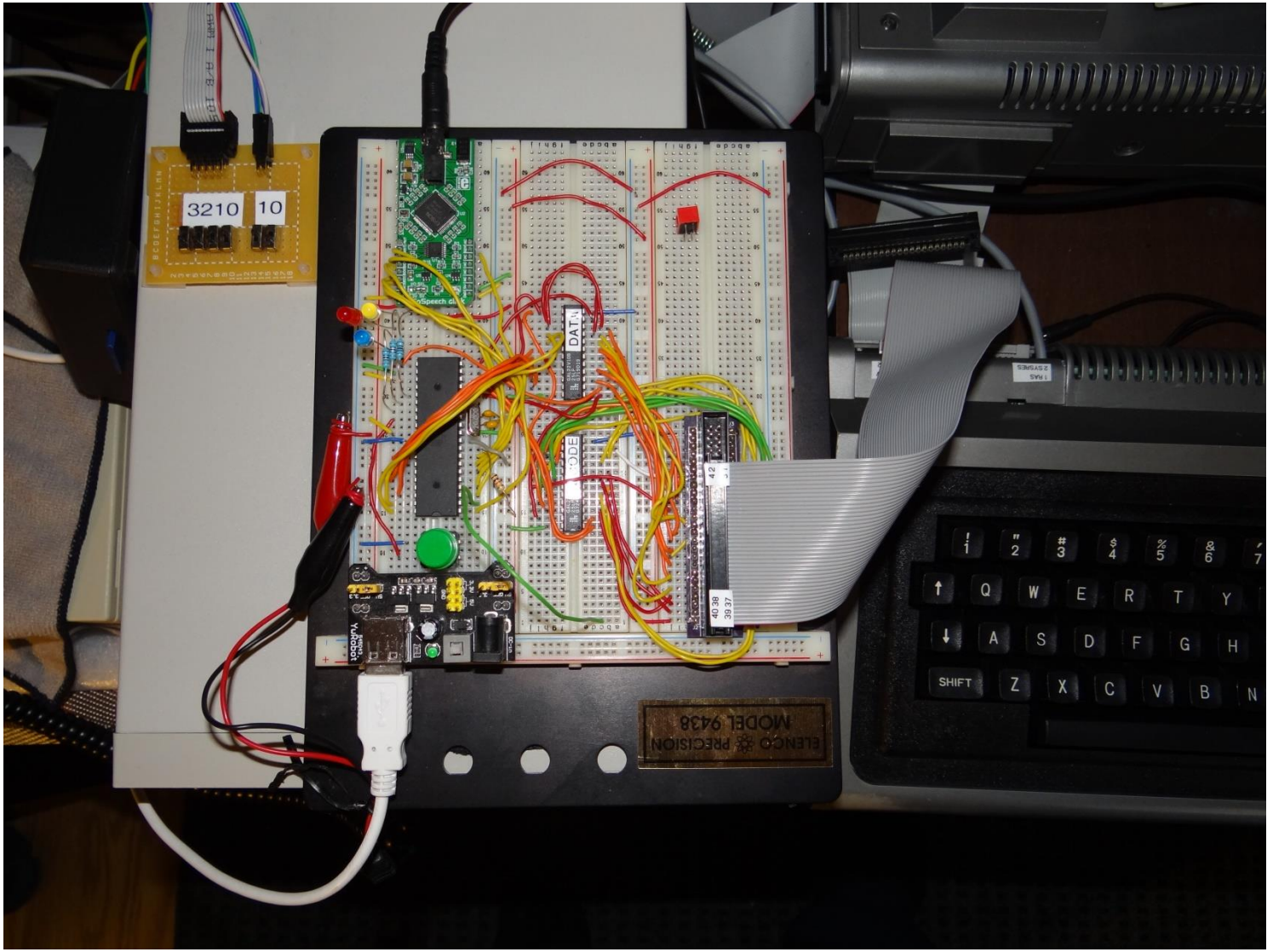
At that point, Talker/80 development was MOSTLY firmware programming and learning about the TRS-80. MUCH more time went into LambdaSpeak.

I confess that I used using autorouting ;-)



LambdaSpeak 1.99 for the Amstrad CPC (2018)

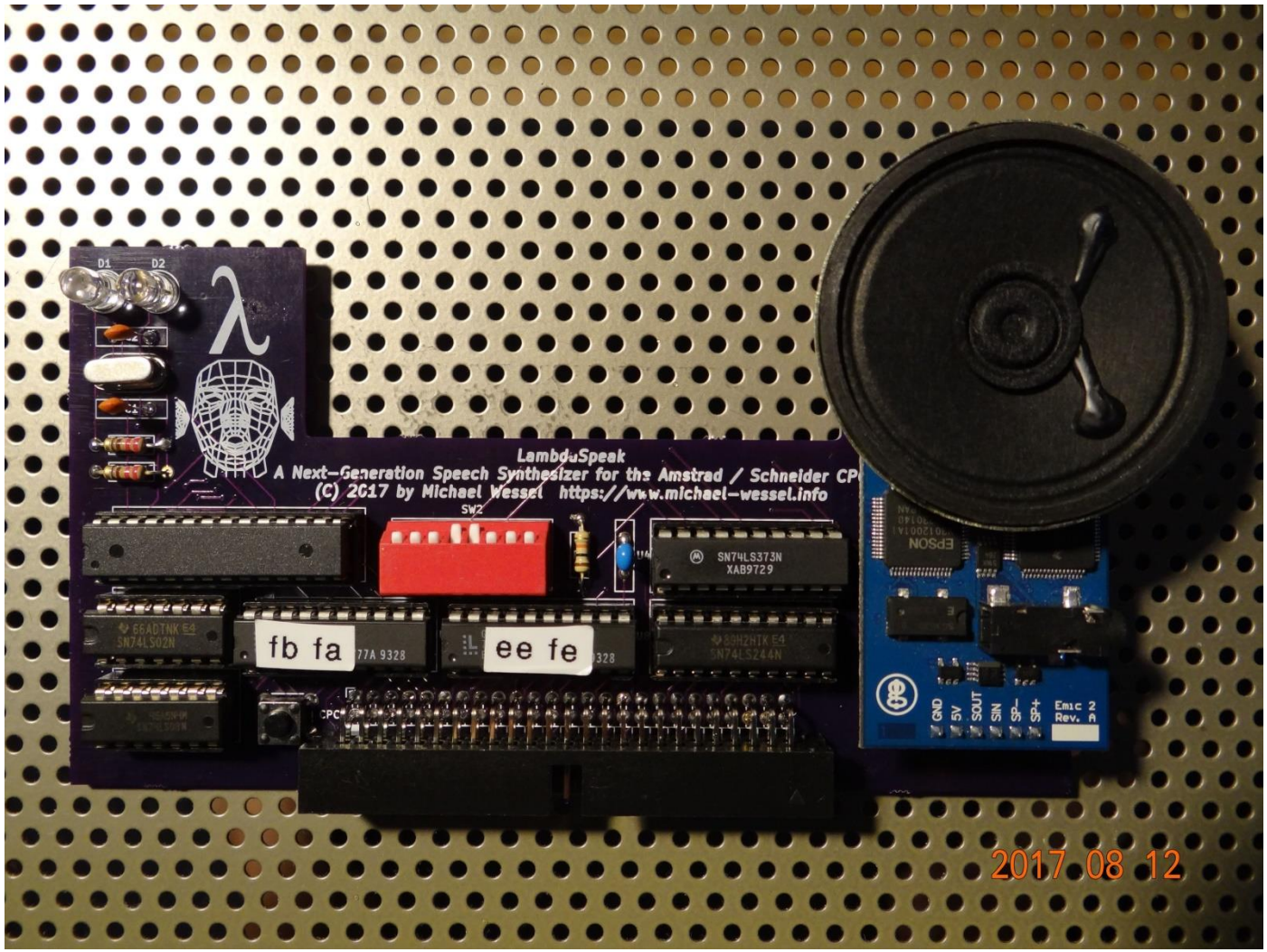
Talker/80 – Early Breadboard Version



Acknowledgements

- Grand Idea Studios (Joe Grand) for making the EMIC 2
  - started my interest in DECtalk v5
- Teodor Costachioiuo
  - Talking Clock Tutorial with the click speech board (but wasn‘t in WinAVR code)
  - Demonstrated how to load the DECtalk firmware and how to program it
- Ian Mavric / Mav for getting my TRS-80s outfitted and (remote) repair help!
- Terry Stewart (Tezza) for his „Driving Instructions“ and disk images
- Arno Puder & Pete Cetinski for inviting me to TrashTalk
- JoeZwerko (Peter Bartlet)
  - Phoneme tables
  - „Mass production tips“
- Eric Dittman
  - First customer of the M1 & MIII /4 Versions
  - TrashTalk reviews
- Members of the Vintage Computer Forum
- Last, but not least - all Talker/80 customers!

LambdaSpeak 1 with Emic 2 (2017)



Great Books

- *DECtalk 5.01 User Guide*
  https://www.digikey.com/htmldatasheets/production/1122220/0/0/1/dectalk-guide.html

- *Practical Electronics for Inventors, Fourth Edition 4th Edition*
  Paul Scherz and Simon Monk

- *Make: Electronics Vol. 1 & 2*
  Charles Platt

- *Make: AVR Programming*
  Elliot Williams (from Hackaday)

- *Hardware Interfacing with the TRS-80*
  John Uffenbeck (Covers M1 & M III / 4)
  (an error regarding Memory-Based IO?)

  > "This experiment cannot be done on a Model III.... The experiment also cannot be done using the expansion interface and a Model 1 computer. This is because the RD and WR control signals are not enabled on the expansion interface bus (you may want to disconnect the expansion interface temporarily for this experiment)".

Summary

- Talker/80 hardware is straightforward
  - No ingenious hacks being pulled – „keep it simple, stupid“
  - Old, proven, slow MCU, BUT „fast enough“ with some additional GAL glue chips / logic
  - Some folks build such a device without an address decoder and latch -
    Feasible with very fast MCUs (ESP32, Cortex M4, .STM32, …)!
  - „Retro“ & DIY / Maker Friendly
  - Main feature natural sounding speech (Epson & DECtalk v5)
  - Some goodies (audio mixer, TRS Voice Synthesizer & VS-100 Emulations)
- Old GALs instead of CPLDs
  - GALs are cheap and still available, WinCUPL is easy to learn
    (Verilog or VHDL requires more learning, and more complex tool suites like ISE)
  - Xilinx CPLDs and Maker-friendly ThroughHole PLCC sockets are expensive
  - „more retro“
- Community effort & welcoming reception
  - Good input and help from the community, thanks a lot guys!
  - ~ 30 fully assembled Talker/80 have been sold
  - ~ 8 kits have been sold