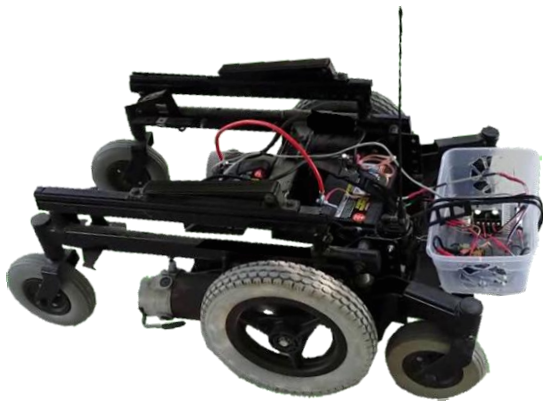


The HamRover 2.0: A Raspberry Pi-Based STEM Education Project

Terence C. Paddack (K5DXD)

Cowtown Hamfest 2020



STEMania @ WCWC

Science

Technology

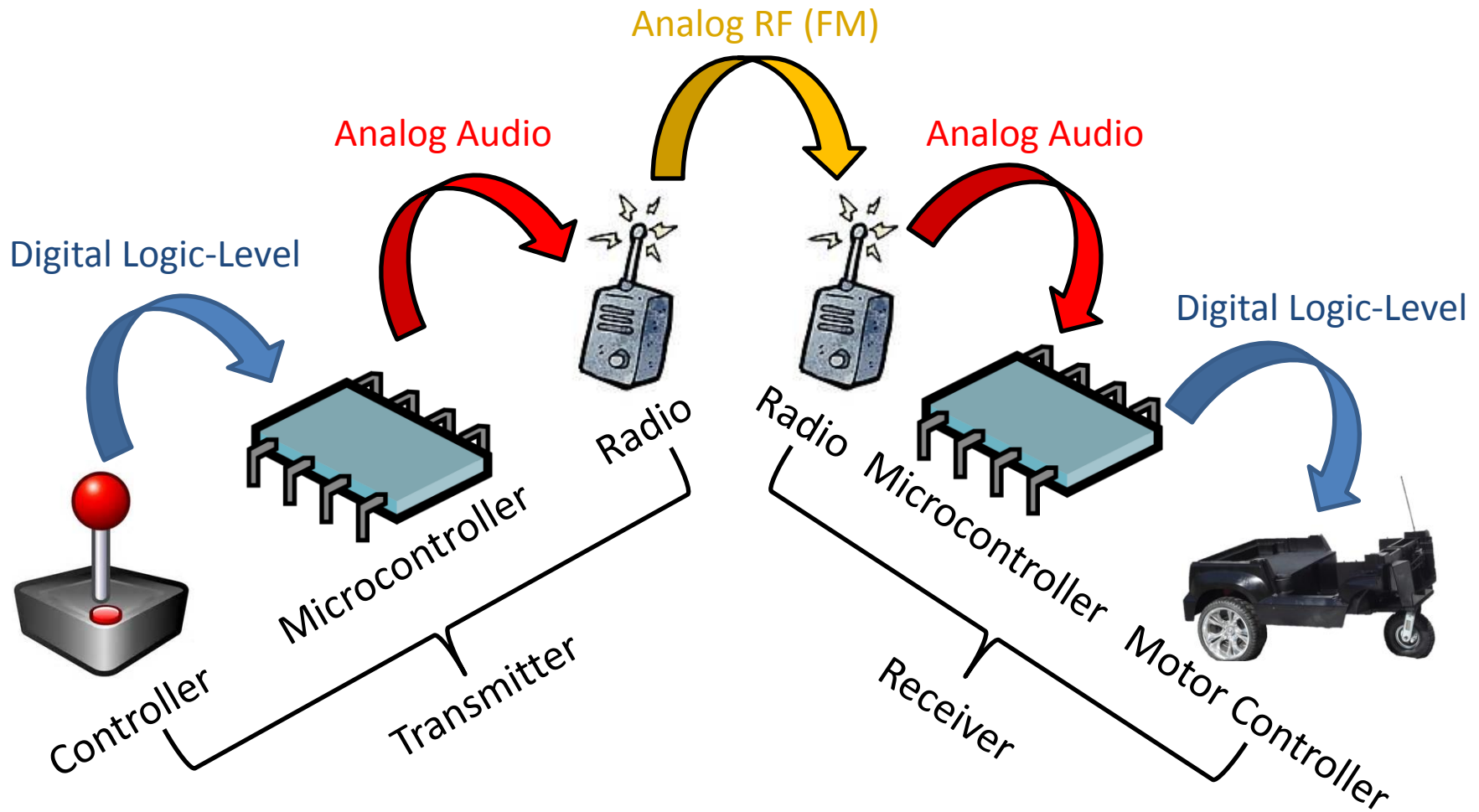
Engineering

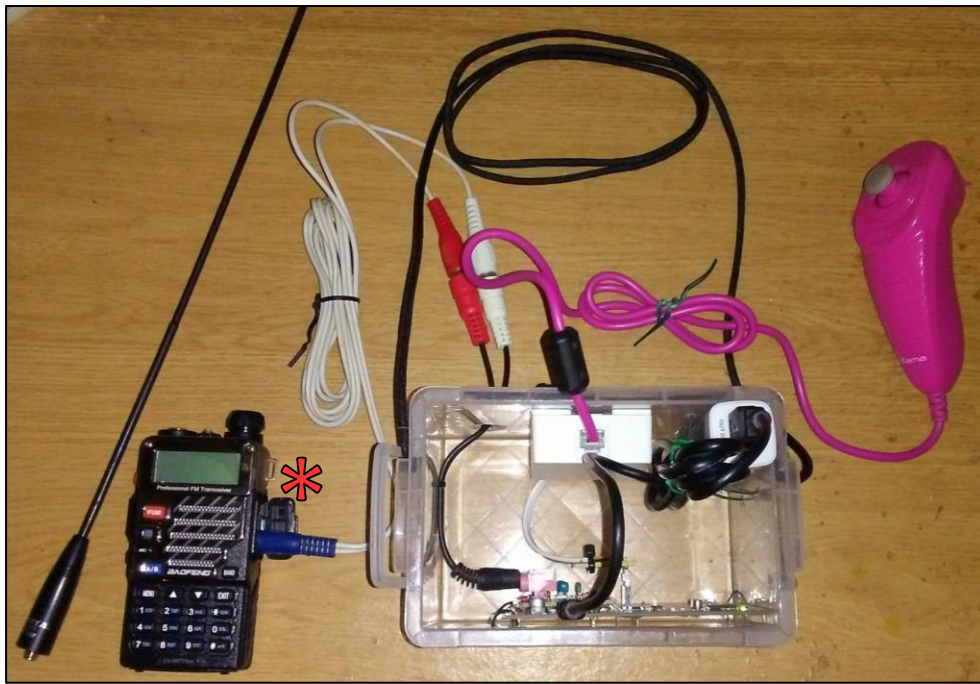
Mathematics

} Ham radio,
anyone?



Original HamRover Concept





Transmitter

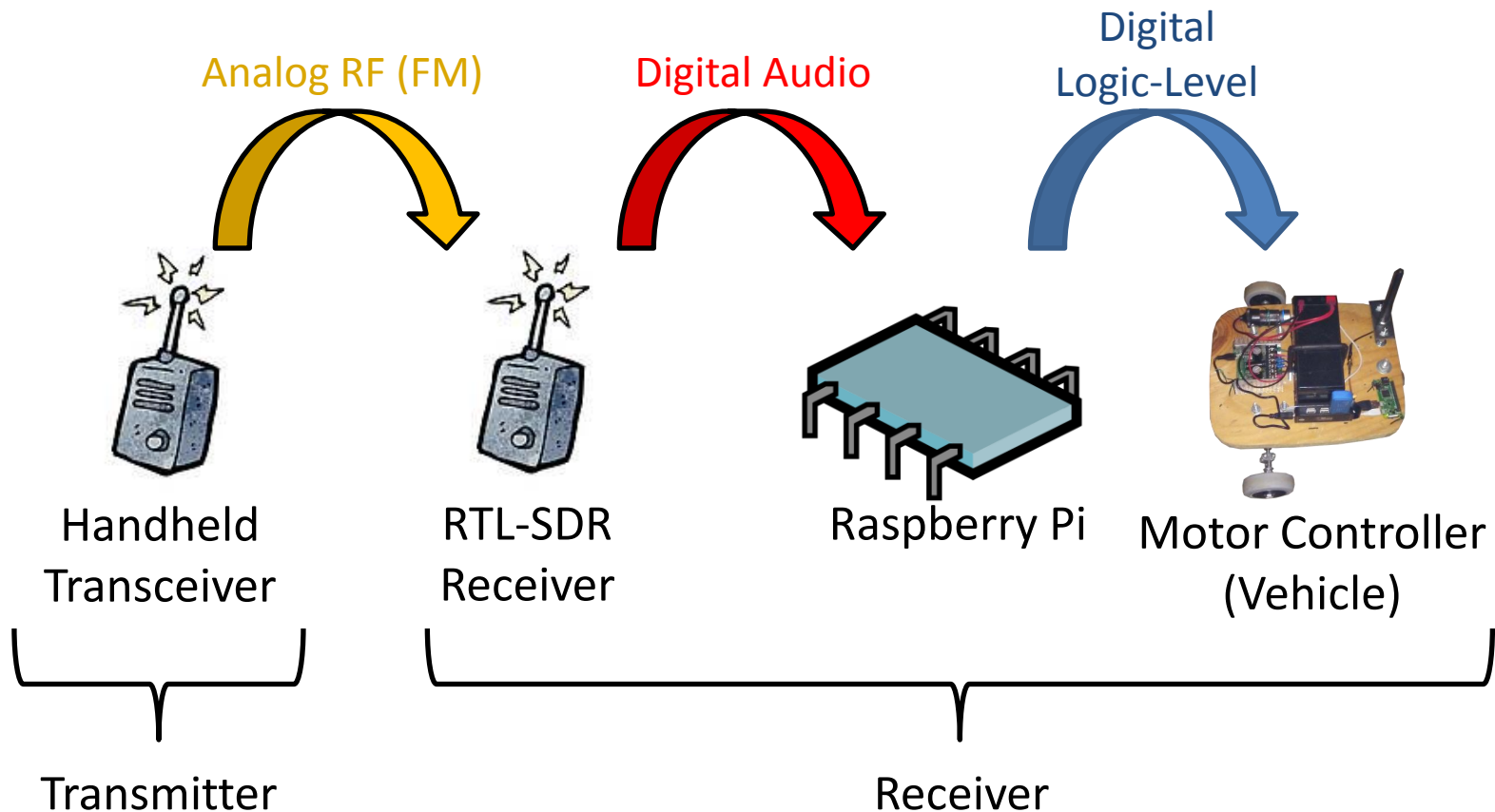
- Nintendo Wii Nunchuck
- Parallax Propeller Microcontroller
 - Emulating a Bell 202 Modem
- Baofeng UV-5R Transceiver
- * Transmitter label for station ID purposes!



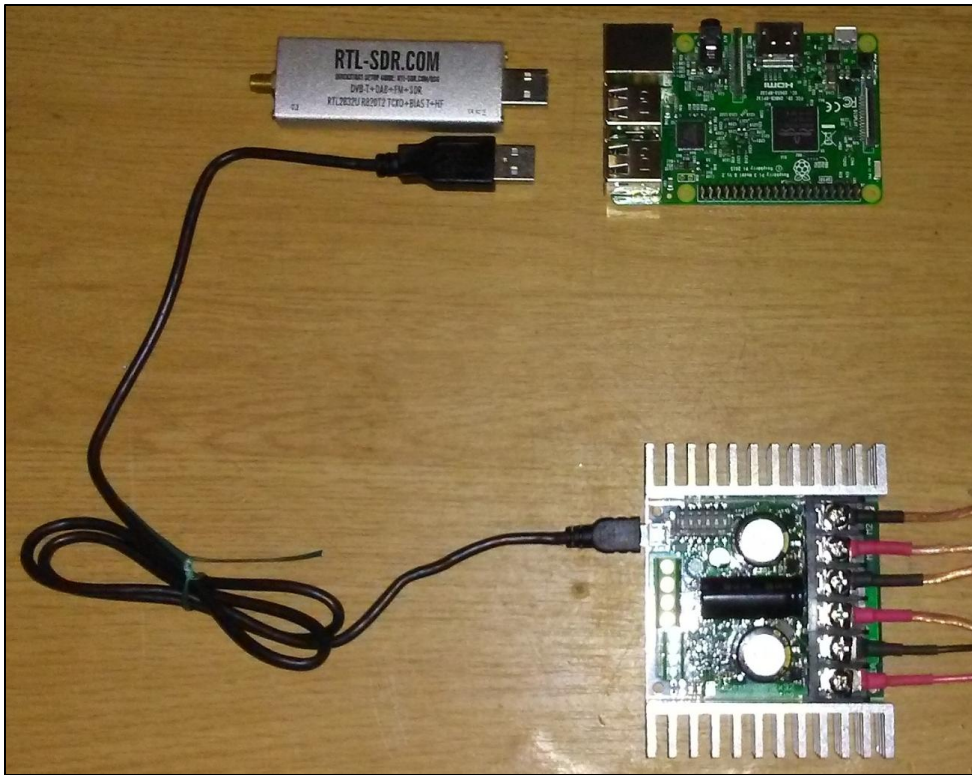
Receiver

- Baofeng UV-5R Transceiver
- Parallax Propeller Microcontroller
 - Emulating a Bell 202 Modem
- Sabertooth 2x32 Motor Controller
 - For DC motors 6 – 30 volts at 32 amps continuous (64 amps peak)

New HamRover Concept



New Concept – Raspberry Pi



Raspberry Pi (~ \$35 US)

- A Linux-based single-board computer with open-source SDR and packet software

RTL-SDR USB Dongle (~ \$15 US)

- A DVB-TV receiver hacked to function as an SDR receiver

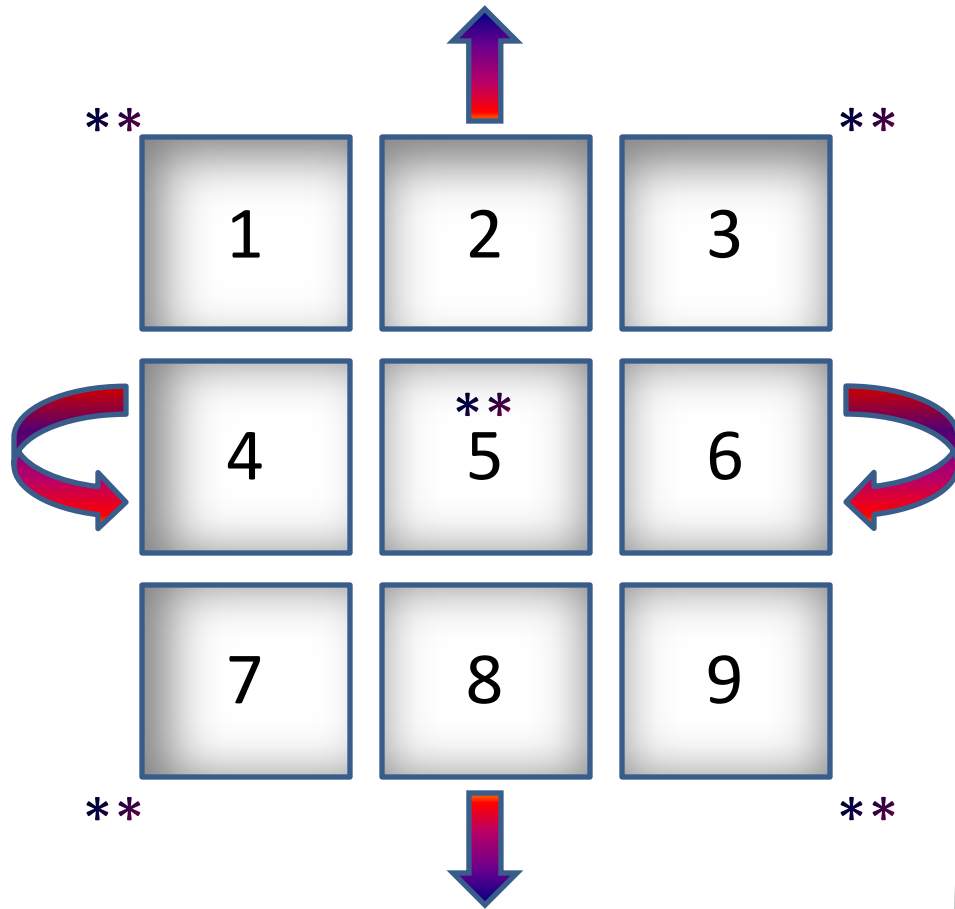
Motor Controller (Prices Vary)

- Small/low-power units = \$
- Large units (pictured) = \$\$\$

Goal

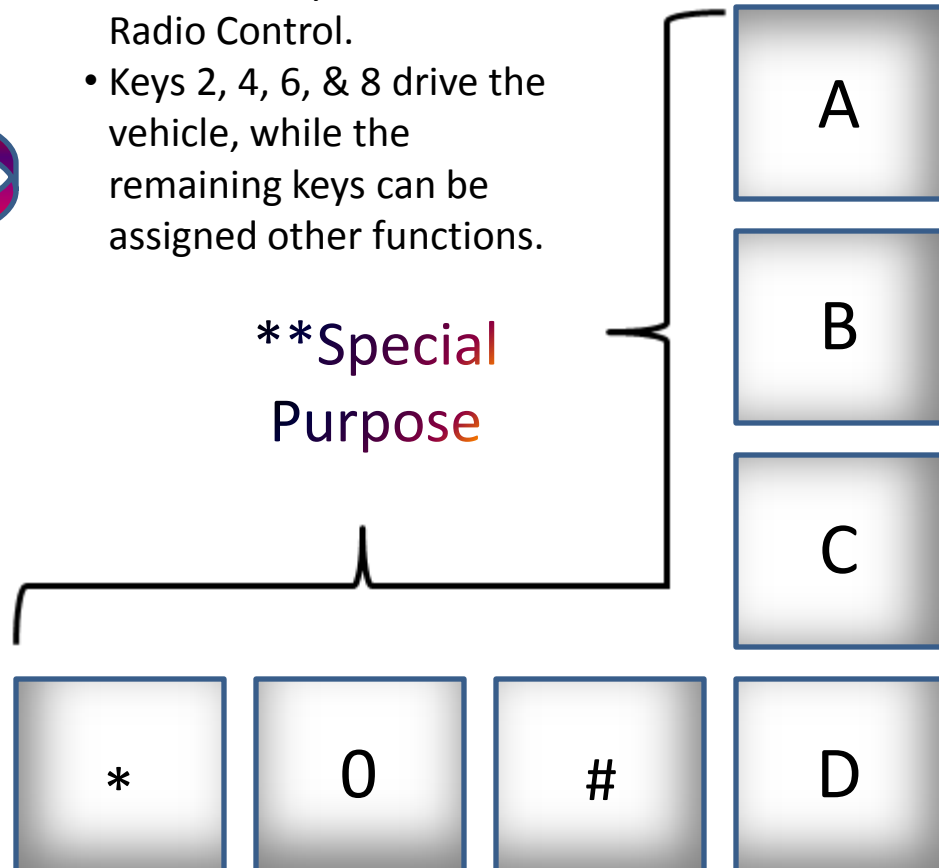
- Use this system in place of dual HTs and modems to decrease cost and complexity.

DTMF Control



- Use DTMF keys on the transmitter radio to control the rover directly
- Eliminates the need for bulky/costly transmitters that are only useful for Radio Control.
- Keys 2, 4, 6, & 8 drive the vehicle, while the remaining keys can be assigned other functions.

****Special Purpose**



Legality & Part 97 Compliance

§97.215 Telecommand of model craft.

An amateur station transmitting signals to control a model craft may be operated as follows:

- (a) The station identification procedure is not required for transmissions directed only to the model craft, provided that a label indicating the station call sign and the station licensee's name and address is affixed to the station transmitter.
- (b) The control signals are not considered codes or ciphers intended to obscure the meaning of the communication.
- (c) The transmitter power must not exceed 1 W.



Considerate Operation



Band and Frequency Selection:

- 6m has designated channels for Radio Control
- Unfortunately, the cheapest and most readily available radios use 2m and/or 70 cm
- Fortunately, 2m and 70 cm regional band plans usually designate frequencies for low power FM simplex or experimental modes
- Texas VHF FM Society Band Plans at <http://www.txvhffm.org/coordination/bandplan.php>
- Monitor frequencies prior to use to ensure that you will not interfere with QSOs or automated systems
- Although station identification by voice is not required for RC under 97.215, it is not forbidden either.

Hardware - Transmitter

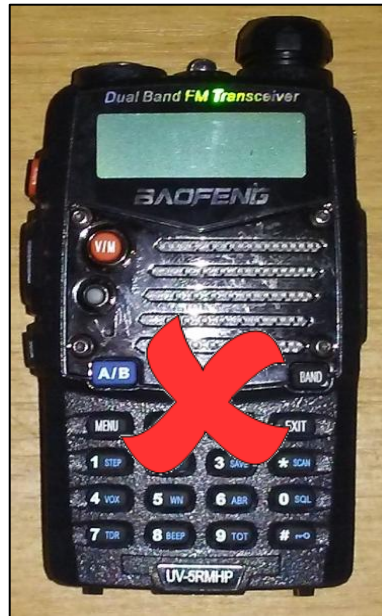
Baofeng UV-5R⁺

Settings:

- VOX: Off
- Low Power setting: 1 W
- FM Simplex on 2m or 70 cm
- Transmit Overtime Timer: 600 seconds
- CTCSS/DCS: Off
- Capable of continuous transmission with brief interruptions every 10 minutes
 - Reliable battery life > 4 hours
- UV-5RMHP transmits > 1 W on low setting
 - This violates Part 97 Section 215
 - Make sure your radio model is compliant!



UV-5R⁺



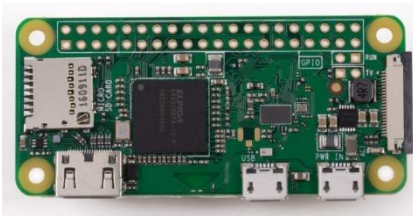
UV-5RMHP

Hardware RTL-SDR USB Dongle



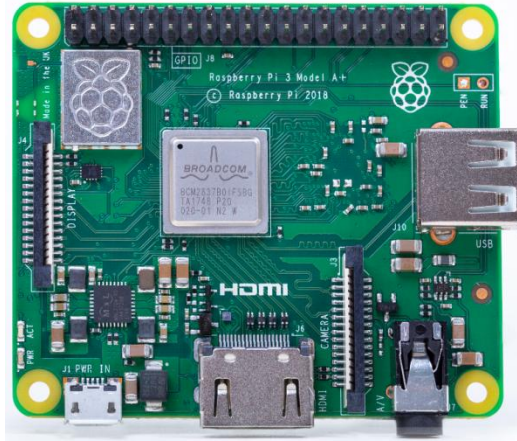
- Available online for \$10 and up (depending on form factor)
 - Differences include:
 - Enclosure construction/material
 - Chipset (Realtek RTL2832U, Elonics E4000, Rafael Micro R820T, etc.)
 - Antenna connector MCX, SMA, etc.
- Generally capable of tuning VHF, UHF (and *some* high HF bands)
- Supported by free software available on GNU/Linux, Windows, Mac
 - GNU Radio (Linux)
 - Linrad (Linux/Windows/Mac)
 - SDR# (Windows/Linux?)
 - HDSDR (Windows)
 - RTL_FM (Linux/Windows)

Hardware - Raspberry Pi



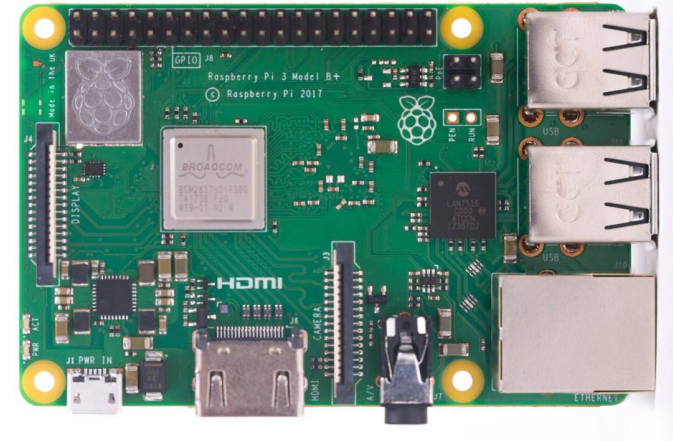
Pi Zero

- \$5-\$15
- Single-core, 1 GHz
- 512 MB RAM
- ~ 1W power draw
- Smallest form factor
- 1 micro USB port
- Pi Zero W has:
 - 802.11n
 - Bluetooth 4.1



Pi 3A+

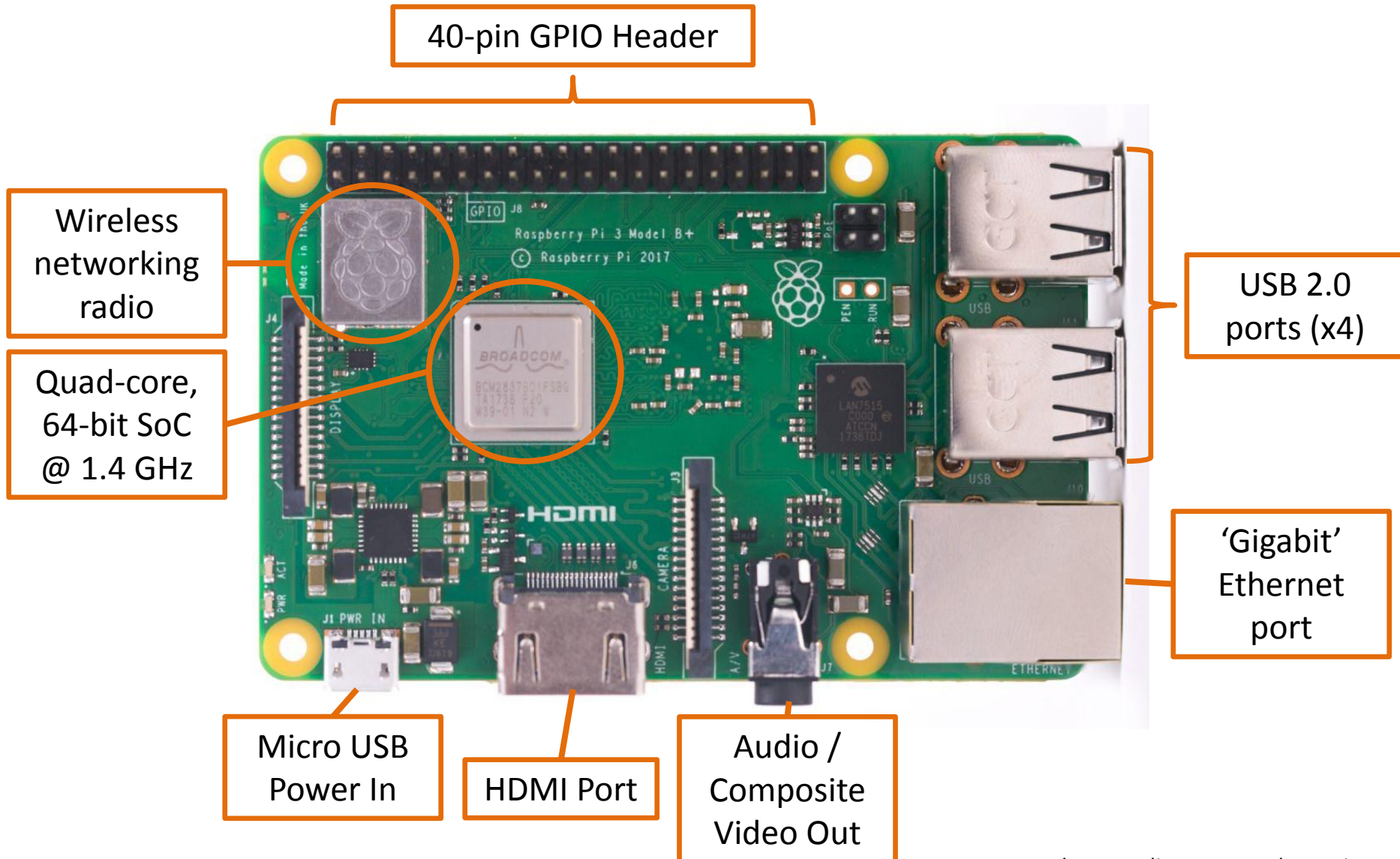
- \$25
- Quad-core, 1.4 GHz
- 512 MB RAM
- ~ 4W power draw
- Medium form factor
- 1 USB 2.0 port
- 802.11b/g/n/ac
- Bluetooth 4.2



Pi 3B+

- \$35
- Quad-core, 1.4 GHz
- 1 GB RAM
- ~ 5.6W power draw
- Largest form factor
- 4 USB 2.0 ports
- 802.11 ac/n
- Bluetooth 4.2

Anatomy of a Raspberry Pi3B+



40-pin GPIO Header

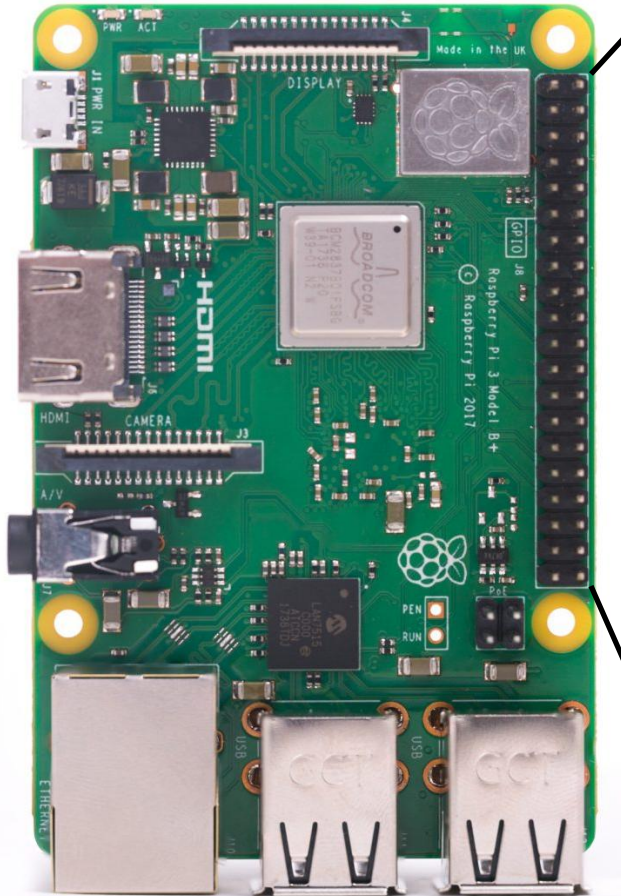
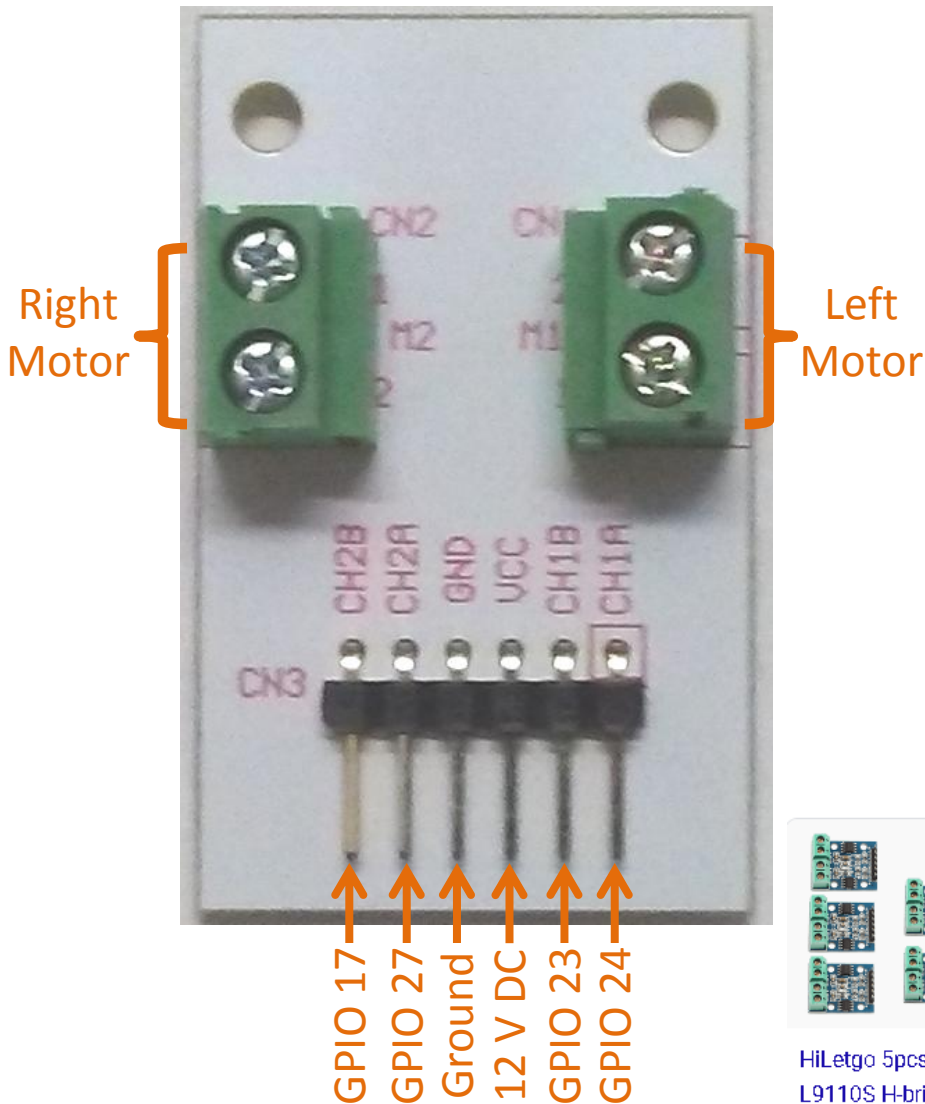


Photo credit: www.raspberrypi.org

		+3.3 V	1	2	+5 V			
I ² C	SDA	GPIO 2	3	4	+5 V			
	SCL	GPIO 3	5	6	GND			
		GPIO 4	7	8	GPIO 14	TXD	UART	
		GND	9	10	GPIO 15	RXD		
		GPIO 17	11	12	GPIO 18	CLK	PCM	
		GPIO 27	13	14	GND			
		GPIO 22	15	16	GPIO 23			
		+3.3 V	17	18	GPIO 24			
SPI	MOSI	GPIO 10	19	20	GND			
	MISO	GPIO 9	21	22	GPIO 25			
	SCLK	GPIO 11	23	24	GPIO 8	CE0	SPI	
		GND	25	26	GPIO 7	CE1		
		ID SD	27	28	ID SC			
		GPIO 5	29	30	GND			
		GPIO 6	31	32	GPIO 12			
		GPIO 13	33	34	GND			
PCM	FS	GPIO 19	35	36	GPIO 16			
		GPIO 26	37	38	GPIO 20	DIN	PCM	
		GND	39	40	GPIO 21	DOUT		

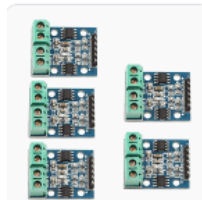
Hardware - Motor Controllers

Front View

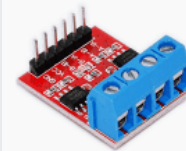


Dual L9110 Motor Driver Board

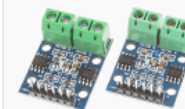
- Small package
 - ~ 1.75 inches X 1 inch
- Dual Channel
 - Differential drive
- Can also control a single 4-wire (bipolar) stepper motor
- Easily interfaces with Raspberry Pi
- Inexpensive, available online, many brands



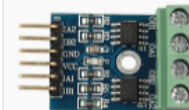
HiLetgo 5pcs
L9110S H-bridg...
\$7.49



L9110s 2
Channels Stepp...
\$0.99



NOYITO Dual
L9110S DC...
\$8.99



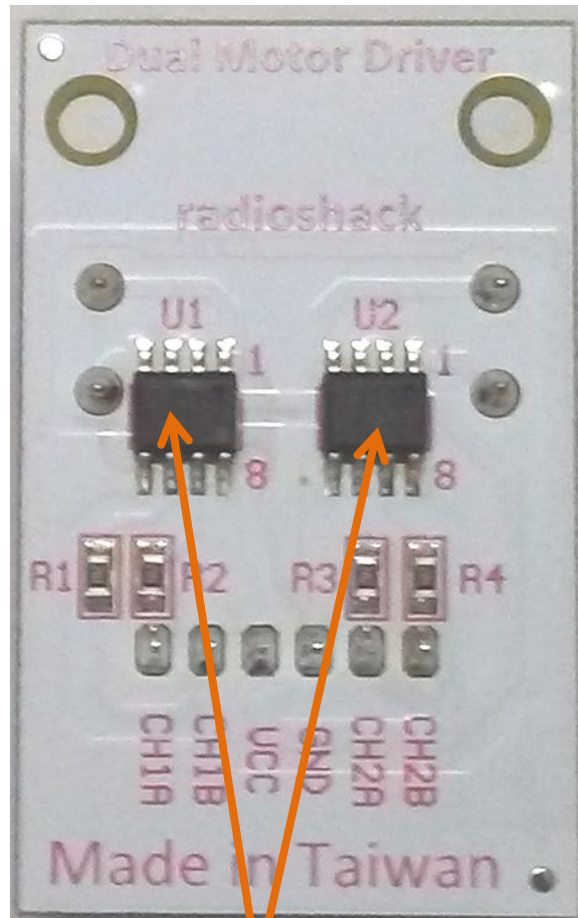
L9110S Dual-
Channel H-bridg...
\$2.77



Stepper Motor
Dual Motor Driv...
\$1.52

Hardware - Motor Controllers

Back View



**L9110
Integrated
Circuits**

L9110 Integrated Circuit

- 3 – 12 V Supply Voltage Range
- Low Power:
 - ≤ 600 mA (Continuous)
 - 1.5 to 2 A (Peak)
- Operating Temperature:
 - 0 – 80 Celsius
- Logic –Level Device
 - Can be directly connected to:
 - Raspberry Pi (3.3 volts)
 - Arduino (5 volts)

L9110 Truth Table			
Input A	Input B	Output A	Output B
High	Low	High	Low
Low	High	Low	High
Low	Low	Low	Low
High	High	High	High

Software: rtl_dtmf

rtl_dtmf:

- a high-level python library (written by K5DXD)
- manages acquisition and analysis of incoming audio signals from the RTL-SDR dongle
 - Starts the rtl_fm software-based receiver as a subprocess on a designated frequency
 - Pipes audio data from rtl_fm into a buffer
 - Analyzes the data to determine which DTMF tones are present
 - Provides a class-based interface that updates in real-time to:
 - Identify which DTMF button is being transmitted
 - Identify the state of the button (pressed, held, released)
 - Store and recognize sequences of DTMF tones
- Is Free Open-Source Software under the GPL v3 license
- Available online (with sample “demo” scripts) at:
 - https://github.com/terence-paddack/rtl_dtmf

Software: rtl_dtmf

Dependencies (must be installed for the library to work properly):

- **Python3**: python language interpreter and virtual machine
 - Runs the script that receives/decodes the signals
 - Generally pre-installed on Raspbian images
- **numpy**: python library that contains objects and methods for the necessary mathematical operations
 - Used by the rtl_dtmf library to perform the FFT
 - python3 -m pip install numpy
- **rtl_fm**: included in the rtl-sdr package for GNU/Linux
 - Used by rtl_dtmf to receive incoming FM RF signal
 - Converts the signal to a stream of bytes that represent the demodulated audio signal
 - sudo apt-get install rtl-sdr
 - Sometimes the following fix is needed for a permissions issue later on:
 - https://github.com/merbanan/rtl_433/issues/740

Decoding DTMF Tones

DTMF keypad frequencies (Hz)				
	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Dual Tone Multi Frequency

- 8 possible audio frequencies
- Arranged in a 4x4 grid
 - Each code is represented by a pair of frequencies
- 16 valid combinations
- Arrangement allows for optimum spacing between frequencies
 - 4 “low” frequencies
 - 4 “high” frequencies
 - Each code is composed of one “low” & one “high” frequency

Software – What is FFT?

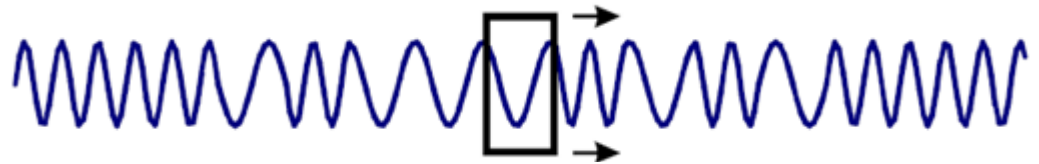


Jean-Baptiste Joseph Fourier

https://en.wikipedia.org/wiki/Joseph_Fourier#/media/File:Fourier2.jpg

Fast Fourier Transform (FFT) is an algorithm that converts a complex waveform into a series of coefficients that represent the relative intensity of each pure sine wave in the signal.

An optimized version of the **Discrete Fourier Transform (DFT)**, which is based on the concept of Fourier Analysis used by Jean Baptiste Joseph Fourier while studying the propagation of thermal energy.



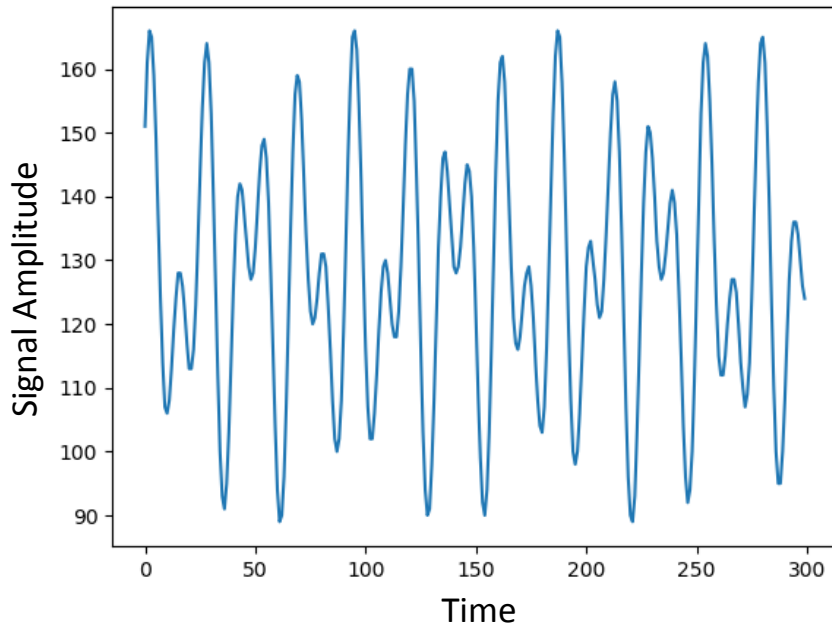
$$\text{Power}_f(t) = \left[\sum_{i=t-15..t} \text{Sig}_i \cdot \sin(2\pi \cdot i/16 \cdot f/1200) \right]^2 + \left[\sum_{i=t-15..t} \text{Sig}_i \cdot \cos(2\pi \cdot i/16 \cdot f/1200) \right]^2$$

© Phil Pilgrim

Okay, so What is FFT?

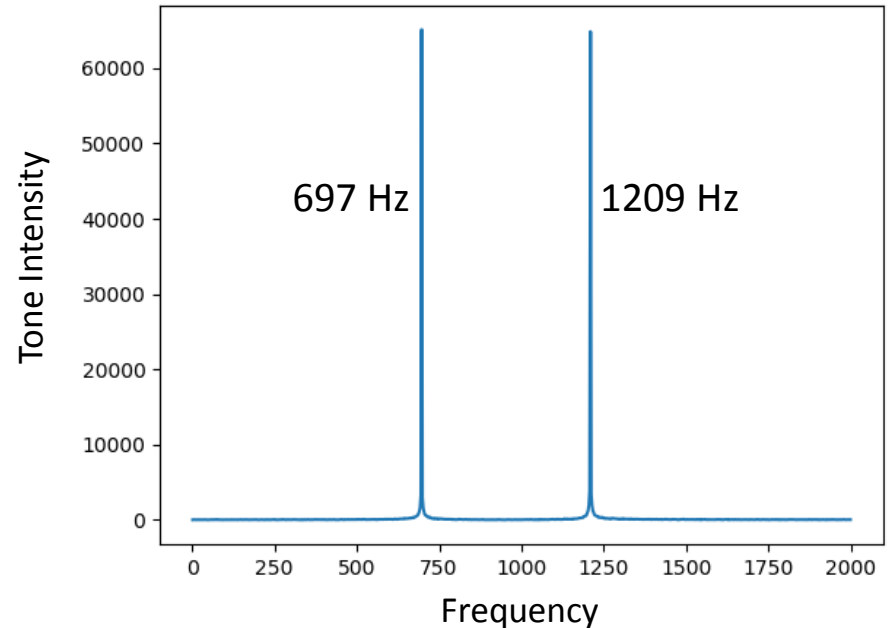
- It converts the input from the **time domain** to the **frequency domain**.
- Takes an **oscilloscope** trace and converts it into a **spectrum analyzer** trace.
- Allows us to determine which audio tones are present in an audio signal.
- Routinely used in digital signal processing.

FFT Input



Audio Waveform of DTMF Key 1

FFT Output

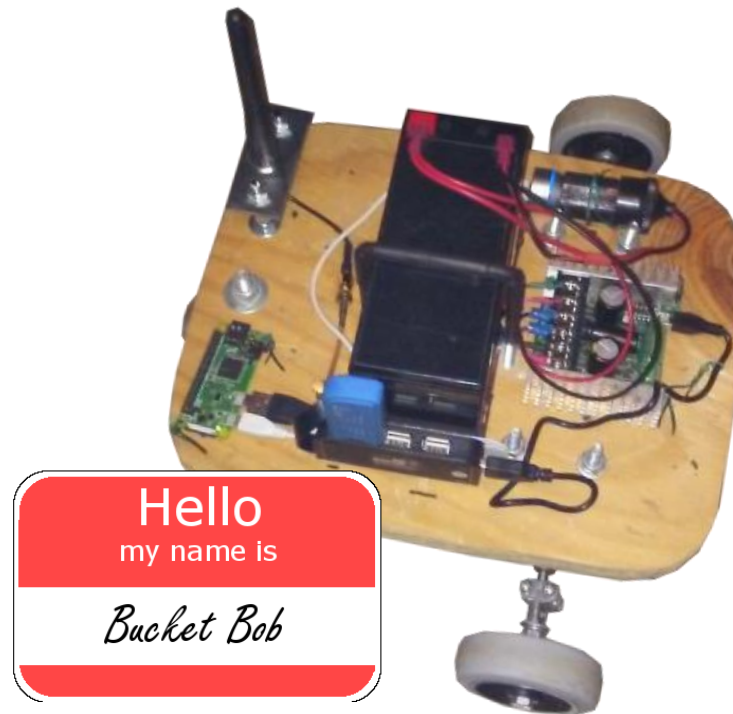
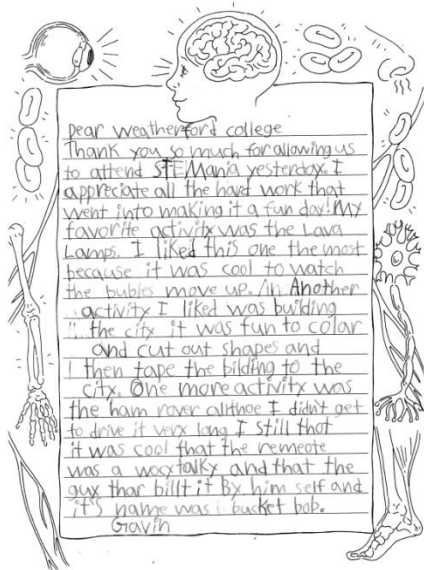


Fourier Series of DTMF Key 1

4th Grader Feedback from STEMania

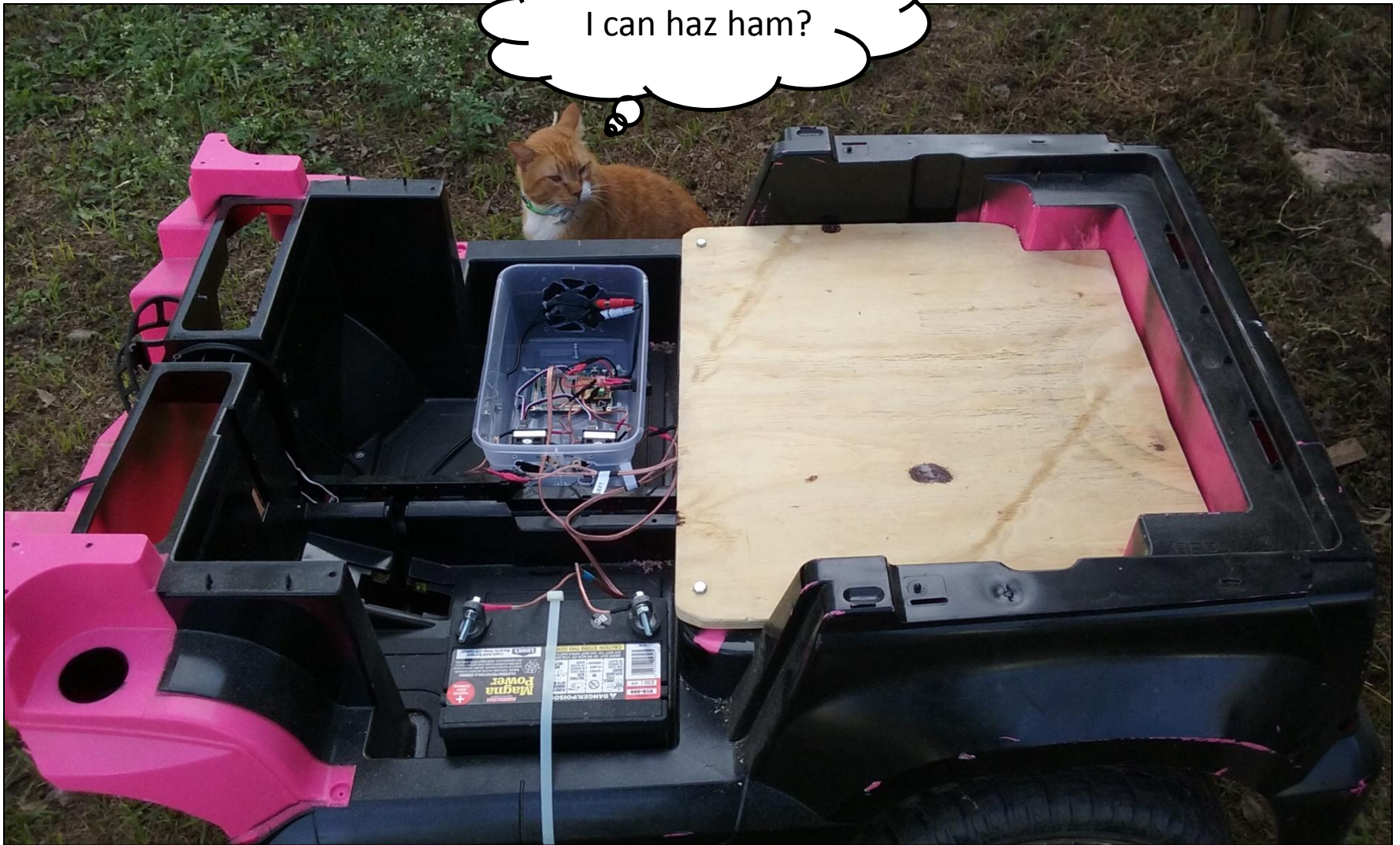
“Another activity I liked was [. . .] the ham rover althoe I didn’t get to drive it very long I still thot it was cool that the remeote was a wocy talky and that the guy that billt it by him self and it’s name was bucket bob.”

– Gavin

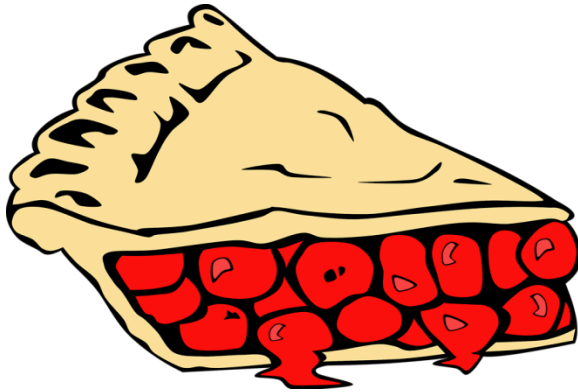


Questions?

I can haz ham?



Licensing



This presentation is licensed under CC BY-SA. You are free to copy and redistribute this material in any medium or format. No warranties are given.

Photos of Raspberry Pi products were generously provided by The Raspberry Pi foundation under this license.

For more information see:

<https://creativecommons.org/licenses/by-sa/4.0/legalcode>

Source code and this presentation are available at:

https://github.com/terence-paddack/rtl_dtmf