Using AI to write simple software applications

Prof Stewart Bryant
Visiting Professor University of Surrey
Board Chair Radio Society of Great Britain
stewart@g3ysx.org.uk

30-January-2025

This Talk

- Short introduction to Generative AI
- Usage methodologies
- A very simple application
- A more complex application
- Other tools
- A number of other use cases
- Conclusion

©Radio Society of Great Britain

Large Language Models & Gen AI

- Generative AI is a conversational method of generating a descriptive and customised answer to a question
- It is a hyper-fast-moving subject
- OpenAI ChatGPT launched late 2022
- Others rapidly launched
  - Bard/Gemini
  - GitHub
  - Woolfram Alpha
  - …. And the list continues
  - DeepSeek

Overview of ChatGPT

- **What is ChatGPT?**
  - ChatGPT is a large language model developed by OpenAI, based on the GPT (Generative Pre-trained Transformer) architecture.
  - It is designed to generate human-like text based on the input it receives.
- **How Does ChatGPT Work?**
  - Pre-training: ChatGPT is initially trained on a diverse corpus of text data, learning to predict the next word in a sentence.
  - Fine-tuning: After pre-training, it is fine-tuned on more specific datasets with human feedback, improving its ability to follow instructions and generate coherent responses.
  - Inference: When a user inputs a query, the model generates text by predicting the most likely continuation based on its training.

©Radio Society of Great Britain

Key Features of ChatGPT

- **Natural Language Processing (NLP):**
  - Excels in understanding and generating human-like responses.
  - Capable of carrying out conversations, answering questions, and providing creative content.
- **Versatility:**
  - Can assist in a wide range of tasks, including coding, writing, tutoring, and more.
  - Continuous Learning:
  - Although the model itself does not learn in real-time, OpenAI frequently updates versions based on new data and user interactions.
- **Limitations:**
  - May generate plausible-sounding but incorrect information.
  - Sometimes struggles with complex reasoning or context retention over long conversations.

A more recent ask

- Having asked in preperation for an earlier talk I asked again
- Please provide two powerpoint slides explaining what you do and how you work with particular reference to software synthesis. Please use general care rather than the text cases I have been generating.
- *Here are two PowerPoint slides explaining what I do and how I work, with a particular focus on software synthesis. You can download and modify them as needed.*
- I get file not found.
- *I'll generate the PowerPoint file again. Here it is:*
- *Let me know if you need any modifications!*

©Radio Society of Great Britain

What I Do

- I provide AI-driven assistance in various technical fields, including:
- - Embedded systems and microcontroller programming
- - Signal processing and waveform generation
- - Software synthesis for audio and signal applications
- - General computing and algorithm development

- I help users design, debug, and optimize their projects.

©Radio Society of Great Britain

How I Work (Software Synthesis)

1. Understanding Requirements:
- - Identify waveform type (sine, square, etc.) and target application.
- - Define parameters like frequency, amplitude, and modulation.

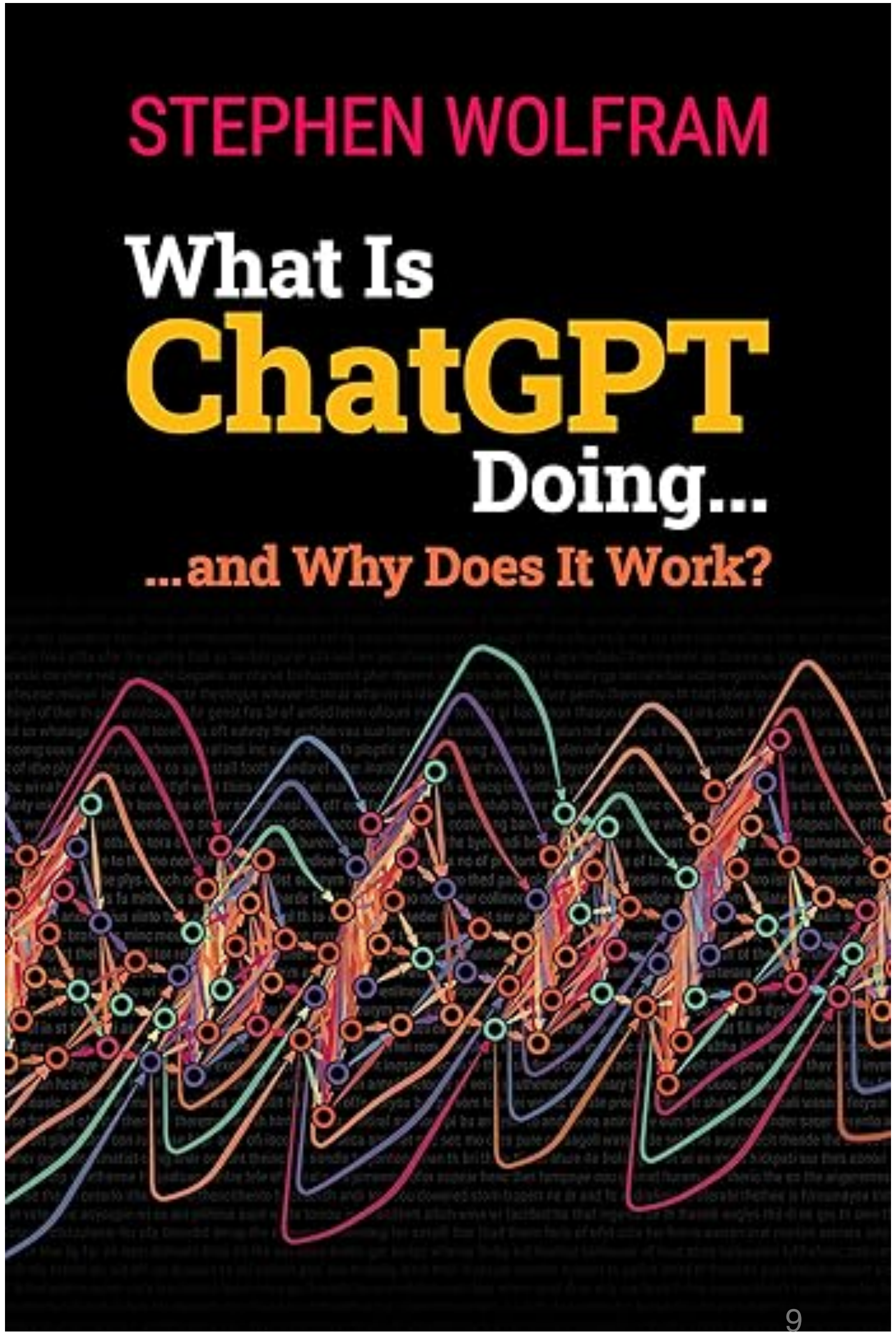
- 2. Generating Waveforms:
- - Use mathematical models (e.g., sine wave tables, DDS techniques).
- - Implement in software using efficient algorithms.

- 3. Output and Optimization:
- - Convert digital data to analog via DACs or PWM.
- - Optimize for real-time performance and minimal latency.

- I assist in implementing and refining these processes for embedded systems.

How does it really work?

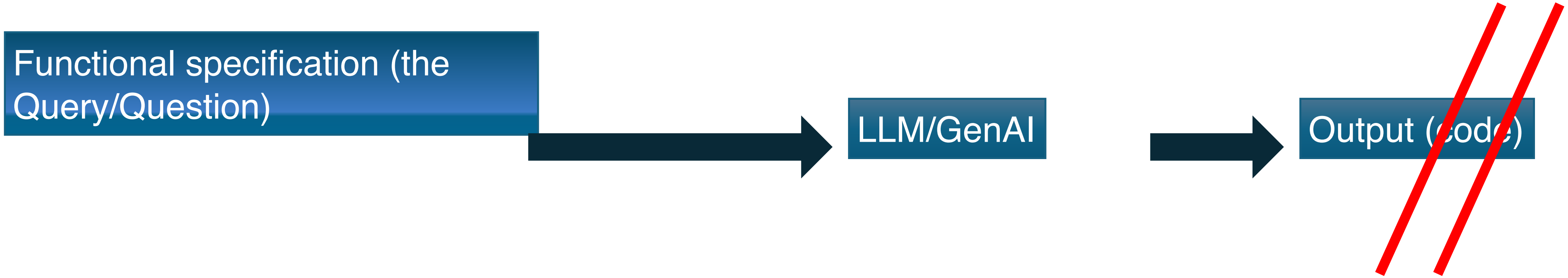
- Key is association of text
  - Constructs sentences by concatenating the word statistically likely to follow the current word
  - Applies a biological evolutionary approach by making random changes to the expected next text.
- Note of caution the author of the book advocates an alternative package that he has an interest in.



Mixed reputation

- Generative AI often applied to composition problems:
  - Writing reports
  - Summarizing findings
- As we will see these systems make mistakes and may be incomplete
- The output needs checking word by word
- We are going to consider the constrained problem of software engineering
  - The formality in the composition of the solution and
  - The formality and automation in verification
  - Makes this a more tractable case

The (broken) Methodology



Functional specification (the Query/Question)

LLM/GenAI

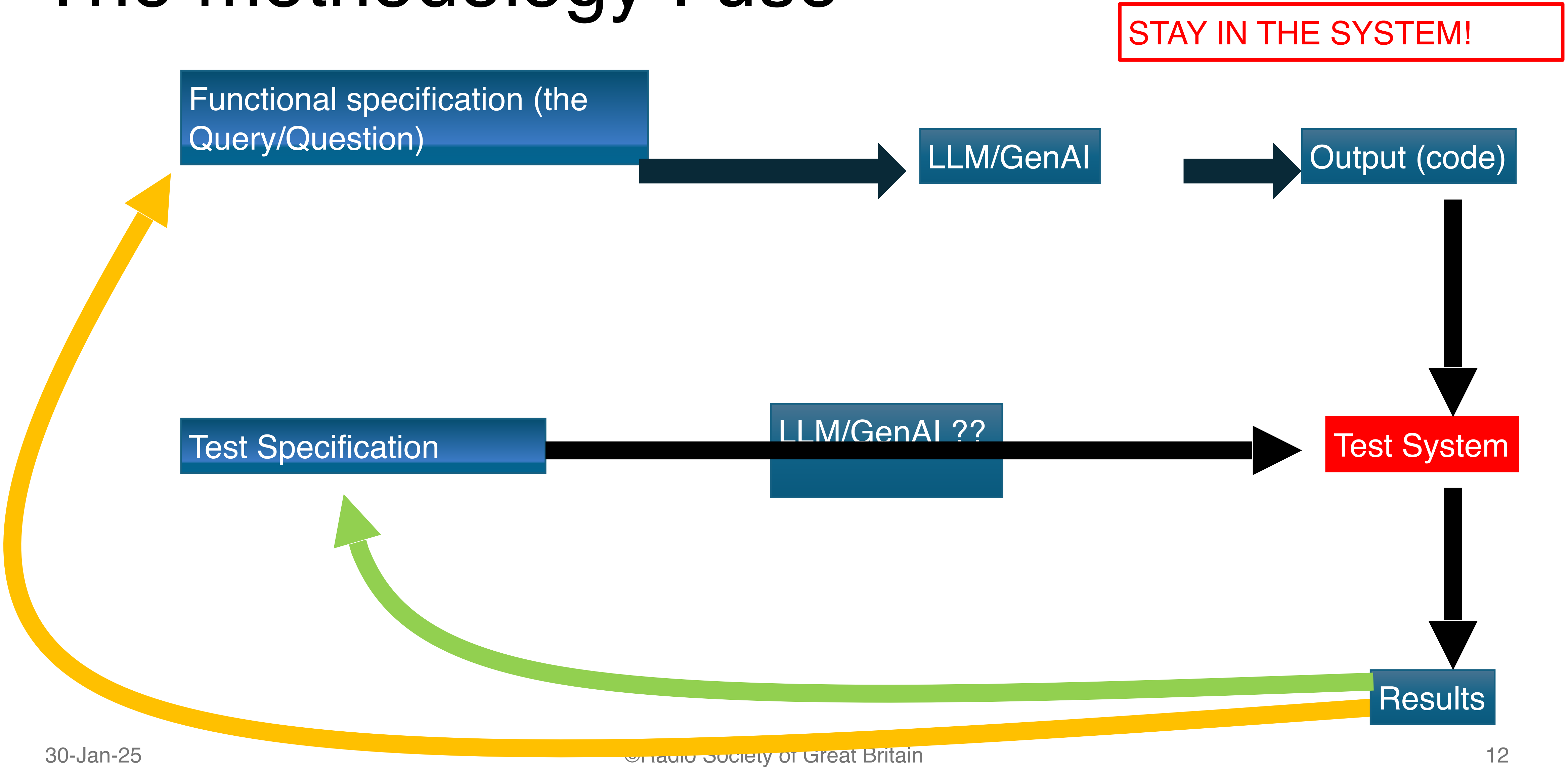
~~Output (code)~~

https://meta.stackoverflow.com/questions/421831/policy-generative-ai-e-g-chatgpt-is-banned

ChatGPT is putting Stack Overflow out of business traffic is down over 50%
https://www.reddit.com/r/ChatGPT/comments/15ju114/

Stack Overflow and OpenAI Partner to Strengthen the World's Most Popular Large Language Models (6/May/2024)

The methodology I use



STAY IN THE SYSTEM!

Functional specification (the Query/Question)

LLM/GenAI

Output (code)

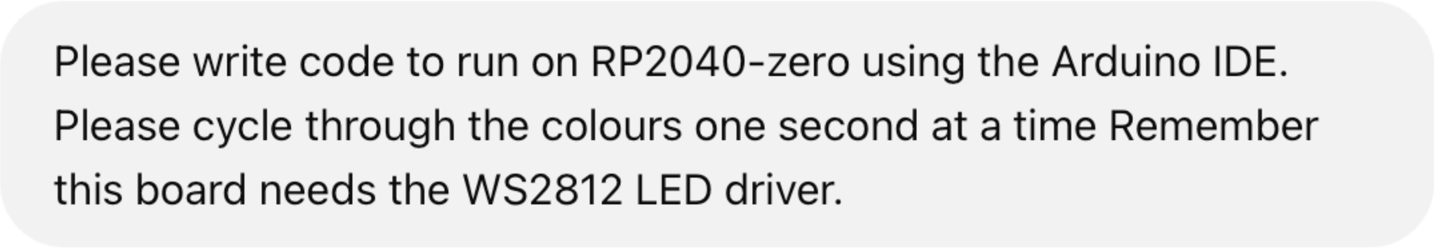
Test Specification

LLM/GenAI ??

Test System

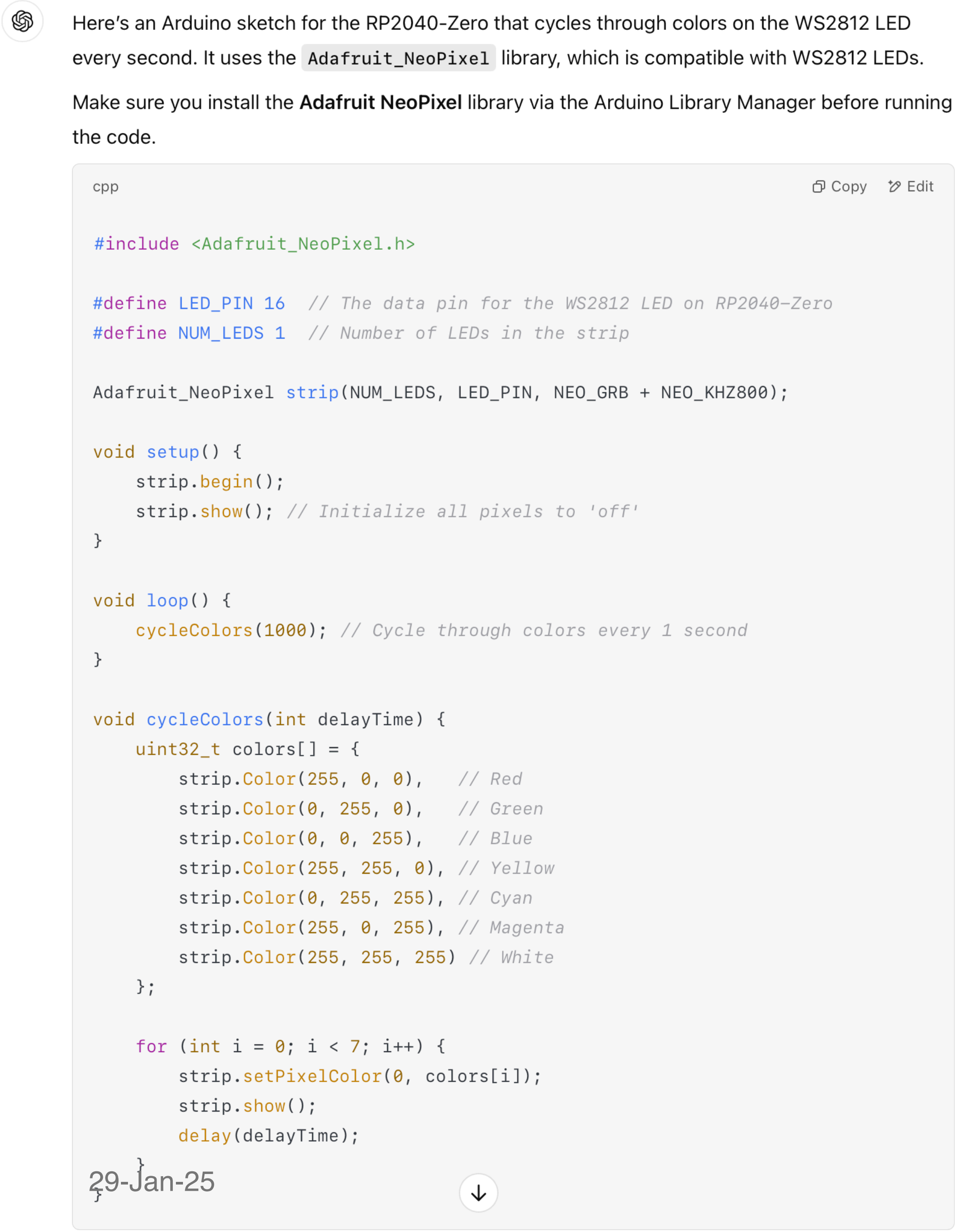
Results

A very simple example



Please write code to run on RP2040-zero using the Arduino IDE. Please cycle through the colours one second at a time Remember this board needs the WS2812 LED driver.

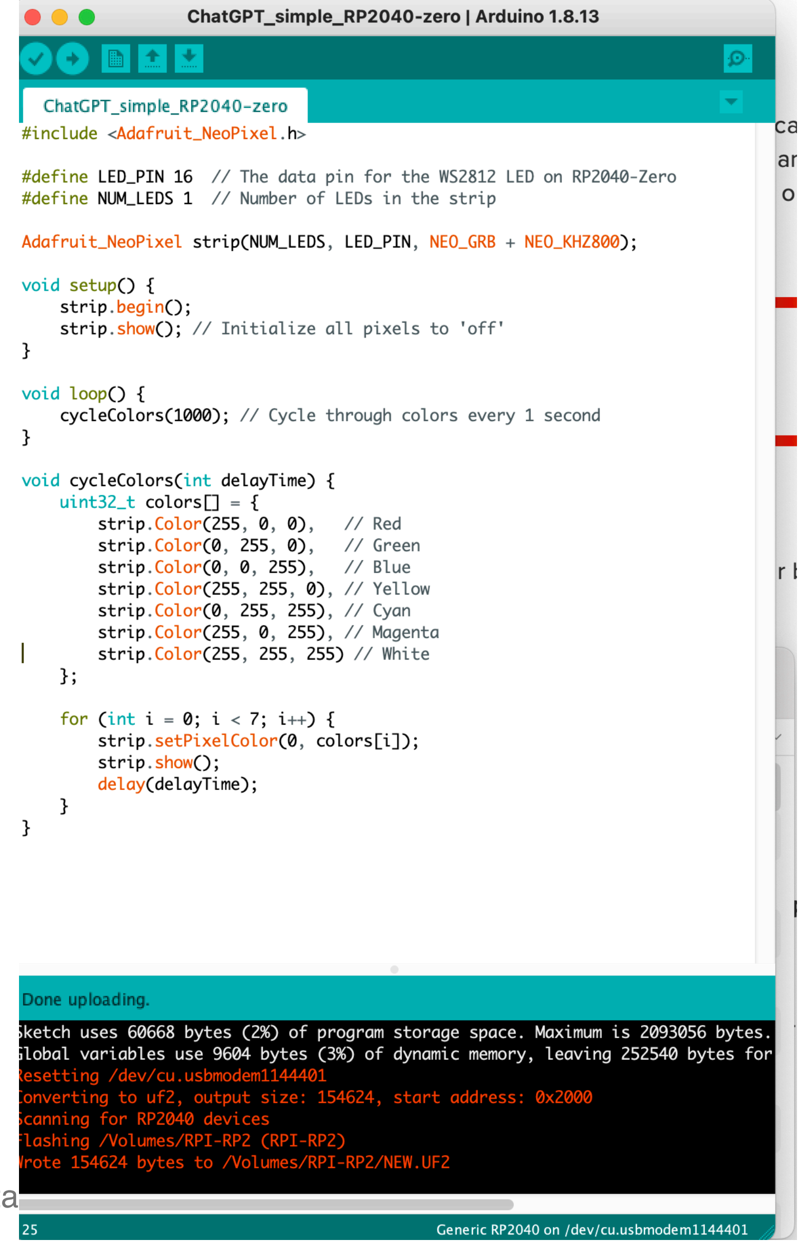
Example - Result



Here's an Arduino sketch for the RP2040-Zero that cycles through colors on the WS2812 LED every second. It uses the `Adafruit_NeoPixel` library, which is compatible with WS2812 LEDs.

Make sure you install the **Adafruit NeoPixel** library via the Arduino Library Manager before running the code.

```cpp
#include <Adafruit_NeoPixel.h>

#define LED_PIN 16  // The data pin for the WS2812 LED on RP2040-Zero
#define NUM_LEDS 1  // Number of LEDs in the strip

Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    strip.begin();
    strip.show(); // Initialize all pixels to 'off'
}

void loop() {
    cycleColors(1000); // Cycle through colors every 1 second
}

void cycleColors(int delayTime) {
    uint32_t colors[] = {
        strip.Color(255, 0, 0),    // Red
        strip.Color(0, 255, 0),    // Green
        strip.Color(0, 0, 255),    // Blue
        strip.Color(255, 255, 0), // Yellow
        strip.Color(0, 255, 255), // Cyan
        strip.Color(255, 0, 255), // Magenta
        strip.Color(255, 255, 255) // White
    };

    for (int i = 0; i < 7; i++) {
        strip.setPixelColor(0, colors[i]);
        strip.show();
        delay(delayTime);
    }
}
```



ChatGPT_simple_RP2040-zero | Arduino 1.8.13
ChatGPT_simple_RP2040-zero

```cpp
#include <Adafruit_NeoPixel.h>

#define LED_PIN 16  // The data pin for the WS2812 LED on RP2040-Zero
#define NUM_LEDS 1  // Number of LEDs in the strip

Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    strip.begin();
    strip.show(); // Initialize all pixels to 'off'
}

void loop() {
    cycleColors(1000); // Cycle through colors every 1 second
}

void cycleColors(int delayTime) {
    uint32_t colors[] = {
        strip.Color(255, 0, 0),    // Red
        strip.Color(0, 255, 0),    // Green
        strip.Color(0, 0, 255),    // Blue
        strip.Color(255, 255, 0), // Yellow
        strip.Color(0, 255, 255), // Cyan
        strip.Color(255, 0, 255), // Magenta
        strip.Color(255, 255, 255) // White
    };

    for (int i = 0; i < 7; i++) {
        strip.setPixelColor(0, colors[i]);
        strip.show();
        delay(delayTime);
    }
}
```

Done uploading.

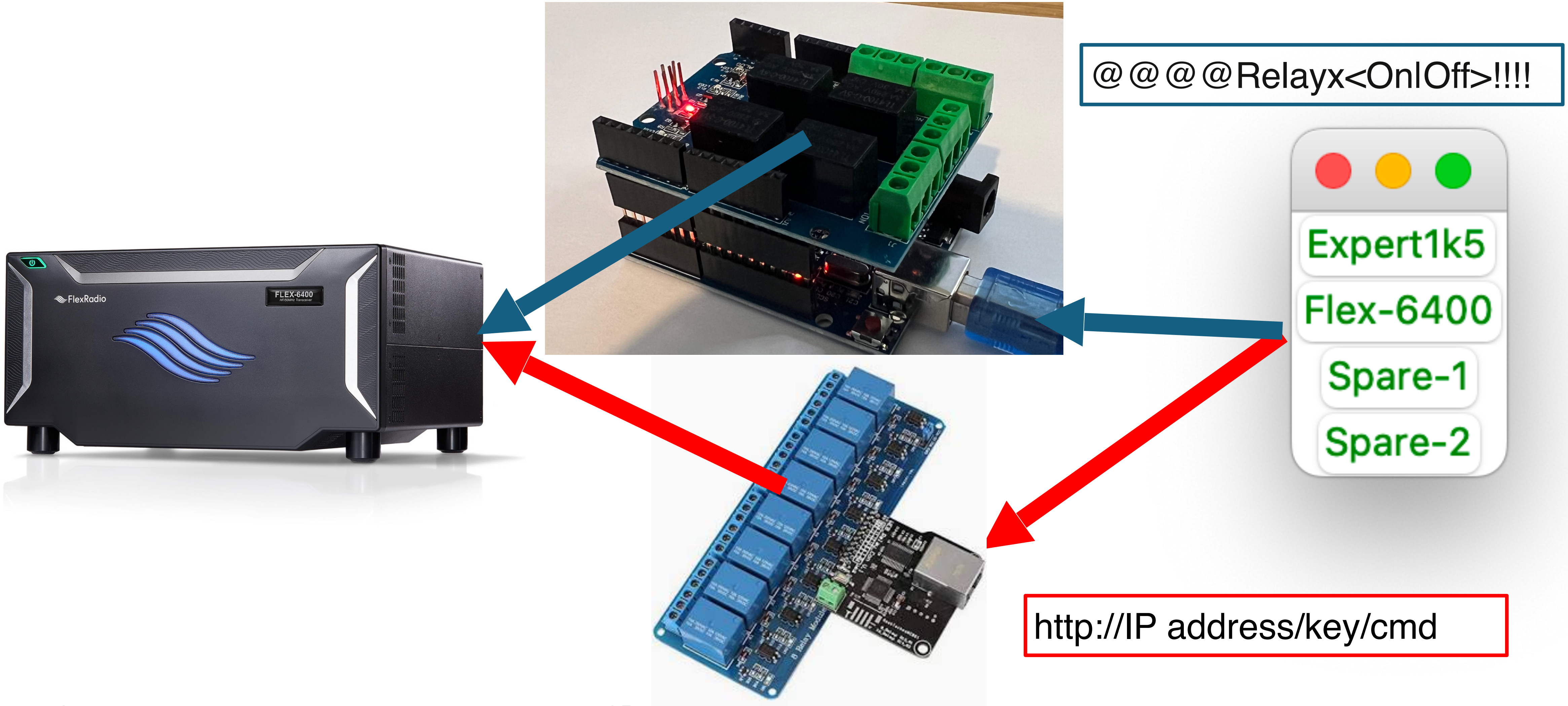
```
Sketch uses 60668 bytes (2%) of program storage space. Maximum is 2093056 bytes.
Global variables use 9604 bytes (3%) of dynamic memory, leaving 252540 bytes for
Resetting /dev/cu.usbmodem1144401
Converting to uf2, output size: 154624, start address: 0x2000
Scanning for RP2040 devices
Flashing /Volumes/RPI-RP2 (RPI-RP2)
Wrote 154624 bytes to /Volumes/RPI-RP2/NEW.UF2
```

Details in RadCom Plus Autumn 2024

rsgb.org/emerging-technology

©Radio Society of Great Britain

A more complex task



@@@@Relayx<On|Off>!!!!

http://IP address/key/cmd

The instruction to generate Arduino code

“Write arduino code that listens on the console port for packets of the form @@@@Relayx <OnIOff>!!!! Where x is a relay in the range 1..4 and the command is On or Off. if the command is on set the corresponding relay digital port to 1 otherwise set it to 0. If the packet is correct reply on the console port with <packet> OK otherwise reply on the console port with <packet>”

After a few iterations and some discussion with ChatGPT about 70 lines of functional, well-commented, code was produced.

It was tested by typing commands on the Arduino console port in the IDE

The code is included in the article but is too long for here.

©Radio Society of Great Britain

Generate python code 1/2 (Ethernet)

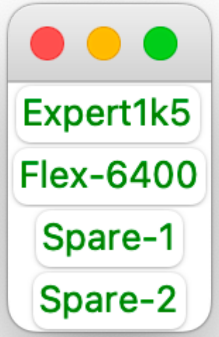
Please write a python3 script to read two parameters from the command line plus an optional third parameter. The first is IP Address the second is the key which is a 0..65535. The third parameter specifies debug output is to be provided. Validate the IP address and the key number. If the command line is not properly formatted print an error with the correct format. To change the state of a relay send an HTTP packet(not HTTPS) request of the form http://IP address/key/cmd. The program should create a column of four buttons which control relays 0..3. Get the program to open text file button.txt where on each line read as CSV text values the label for the button the command to turn the corresponding relay on and then the command to turn the corresponding relay off then whether the button is toggle button (T) or a momentary button (M). The button text for all buttons is initially green and the relays are all off. When a relay is off colour the button text green and when it is on colour the button text red. (Continued over)

Generate python code 2/2 (Ethernet)

When a button that is green is pressed send the corresponding HTTP request to turn it on and change the text colour to red. When a button with red text is pressed send the corresponding off HTTP request and change the button text colour to green. When the top button is pressed turn relay 0 on, chang the button colour to red wait two seconds and then turn the relay off changing its colour back to green. It the button is a momentary button set the relay on, wait two seconds and set the relay off, setting the button colour red when it is on and green when it is off.
Please include this ChatGTP request as comment at the start of the the code. If debug is requested also write the URL to the console.

Button.txt →

```
Expert1k5,01,00,M
Flex-6400,03,02,T
Spare-1,05,04,T
Spare-2,07,06,T
```



Results

- Too difficult to show code on slides – please see article.
- Fully function code was produced but in different styles on different GenAI platforms.
- Occasional crashes during development – fixed by pasting crash into into chat

```
Appleton:EtherRelay stewartbryant$ python3 BardEtherRelay4.py 192.168.8.5 30000 debug
Exception in Tkinter callback
Traceback (most recent call last):
 File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/tkinter/__init__.py", line 1705, in __call__
  return self.func(*args)
 File "BardEtherRelay4.py", line 90, in <lambda>
  command=lambda data=button_data: relay_control(data)
 File "BardEtherRelay4.py", line 69, in relay_control
  button_data["button"].config(fg=button_data["color"])
KeyError: 'button'
```

Experiences with ChatGPT (version 3)

- Took about 37 attempts to get functioning code python for the relay application
- The code was well written though not as elegant as Bard
    - This might have been the result of the design debug conversation
- It was willing try again if you asked it to do so and would produce different code.
- It did (if asked) include the instructions as comments (i.e. include the “source code” in the generated code.
- It is (as is Bard) incredibly polite.

Experiences with Bard

- Bard was Google’s first offering. It is replaced by Gemini.
- Example had the advantage that I had tuned the functional specification
- It produced compact sophisticated code for the Ethernet controlled relay in 7 iterations some of which were requests to output a complete python script and not just edits.
- It was very polite but incredibly wilful
  - ‘I just did that for you”
  - It never included the input chat as comment
- It became my preferred code generator, but ChatGPT 4 is extremely good and (for me) took the leading spot Q1 2025

©Radio Society of Great Britain

Other AI approaches

- I have focused on the tools I studied in H1 2024
- There are many new GenAI applications emerging to support software engineering.
- Many vendors appearing on the bandwagon, with many playing catchup.
- Copilots, for example Git
- Code repository analyzers
- Combined copilot and GenAI applications.
- There is a huge body of sales literature and (usually sales oriented) tutorial material appearing.

©Radio Society of Great Britain

Other Use Cases

- Data Analysis
- Digital Signal Processing
- Hardware Synthesis

©Radio Society of Great Britain

Data Analysis

95GB of CSV files in directory tree, with a mixture of templates and some corrupt data

Extract all fields that start with 2, M or G

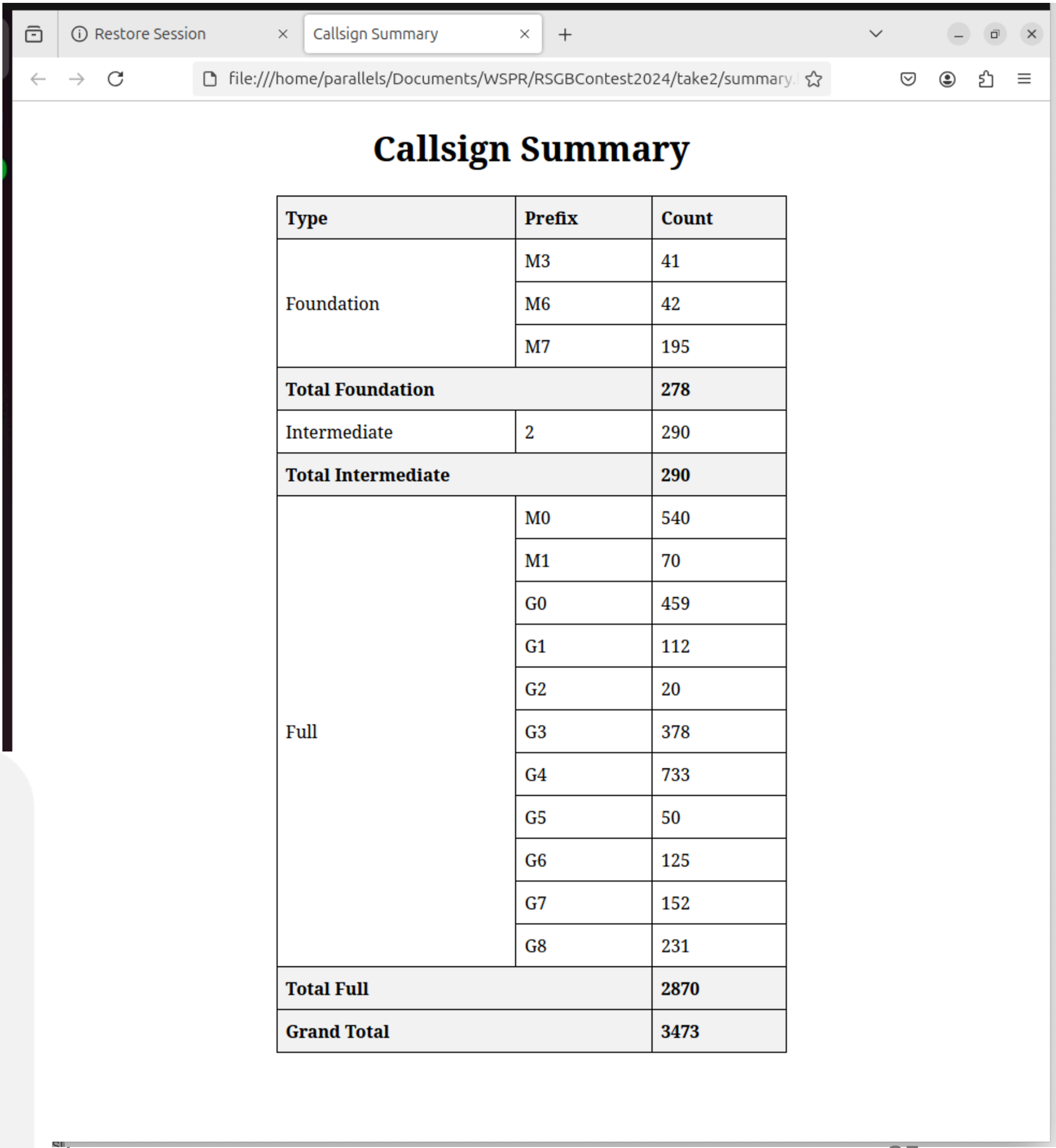
Convert to canonical format

Remove values not appearing N times

Sort and remove duplicates

Apply a number of filters

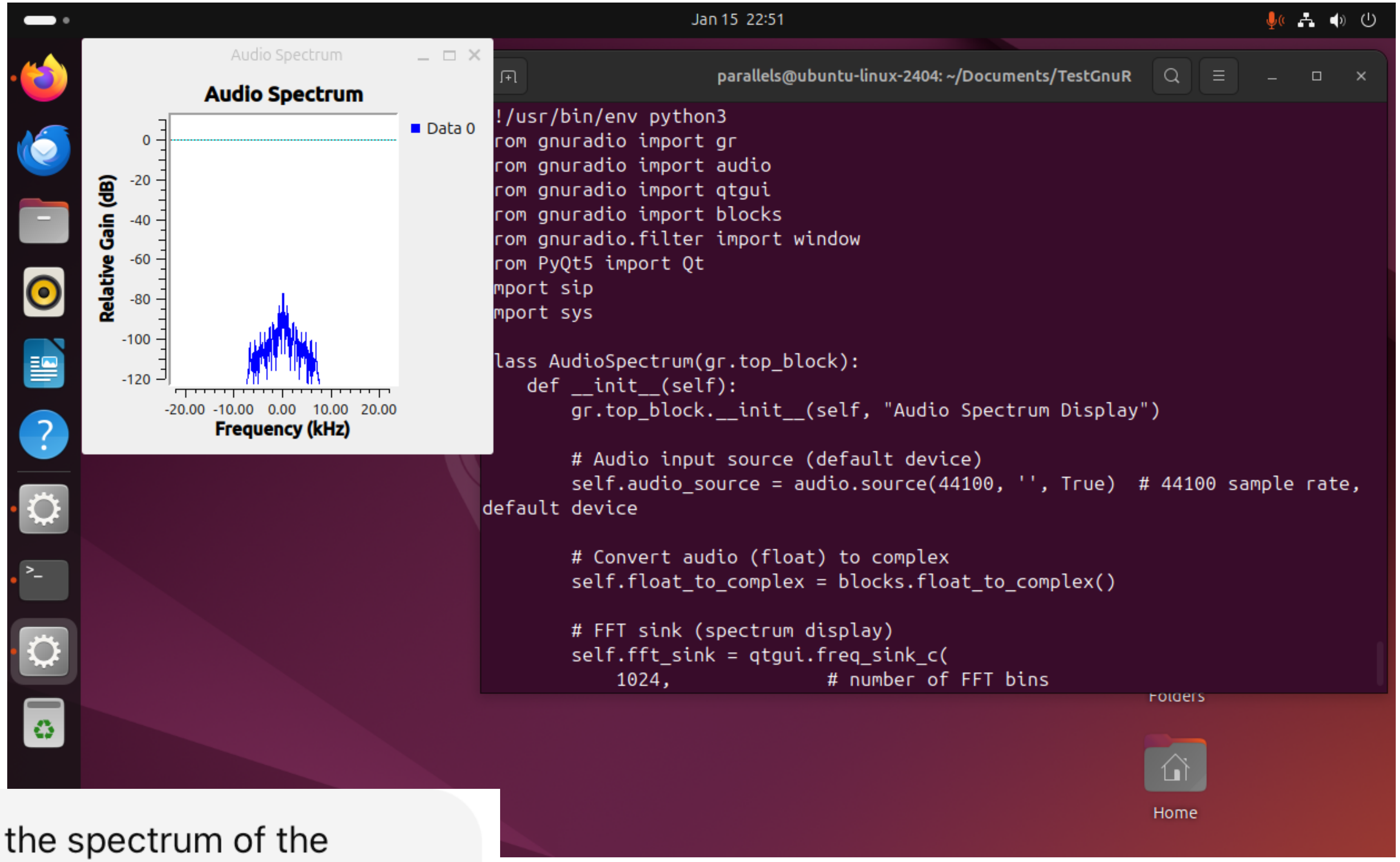
Please write a shell script there takes an input file and an output file as parameters. From the input file extract the number of foundation callsigns (these start with M3, M6 or M7) the number of intermediate callsigns (these start with 2) the number of full callsigns (these start with M0, M1 or G0..8). In the output create a HTML display showing the total number of callsigns by type and the number by constituent prefix and the grand total


Restore Session
Callsign Summary
file:///home/parallels/Documents/WSPR/RSGBContest2024/take2/summary.

Callsign Summary

Type	Prefix	Count
Foundation	M3	41
	M6	42
	M7	195
Total Foundation		**278**
Intermediate	2	290
Total Intermediate		**290**
Full	M0	540
	M1	70
	G0	459
	G1	112
	G2	20
	G3	378
	G4	733
	G5	50
	G6	125
	G7	152
	G8	231
Total Full		**2870**
Grand Total		**3473**

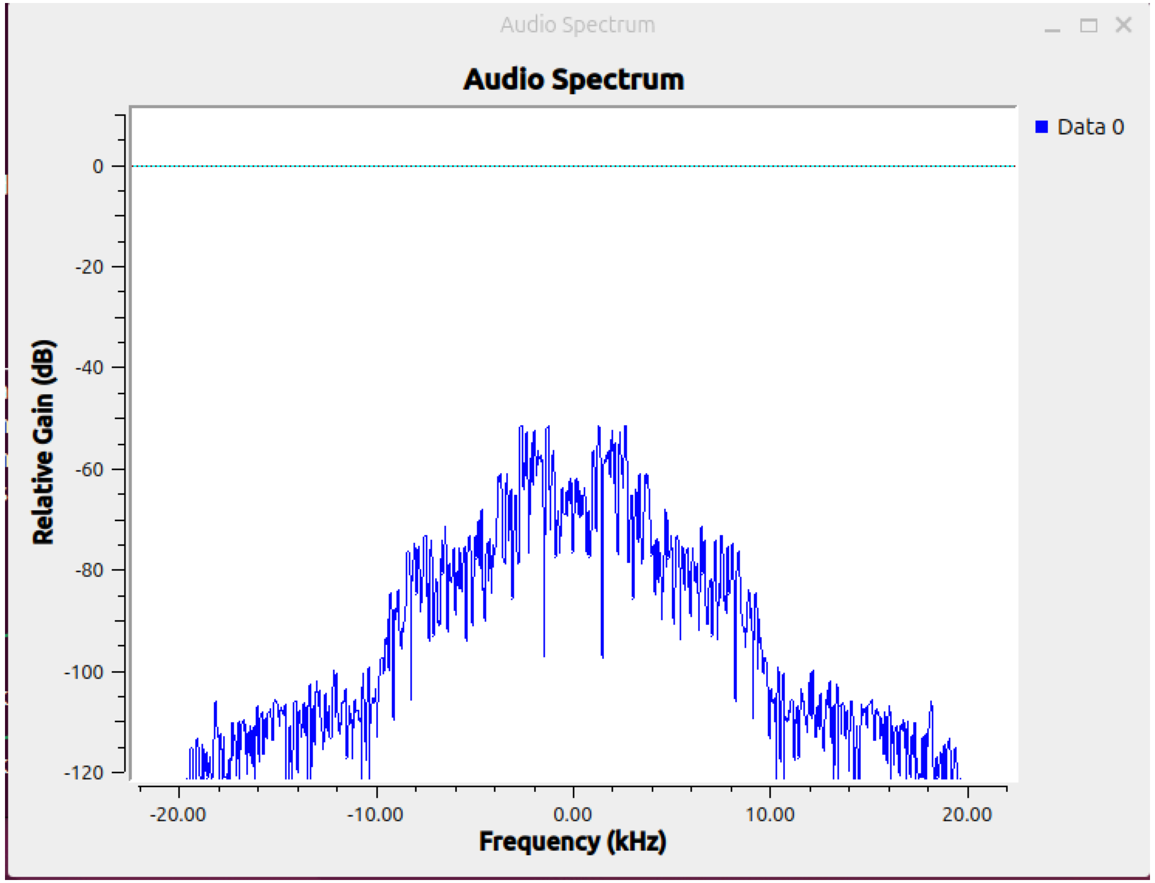
GNUradio

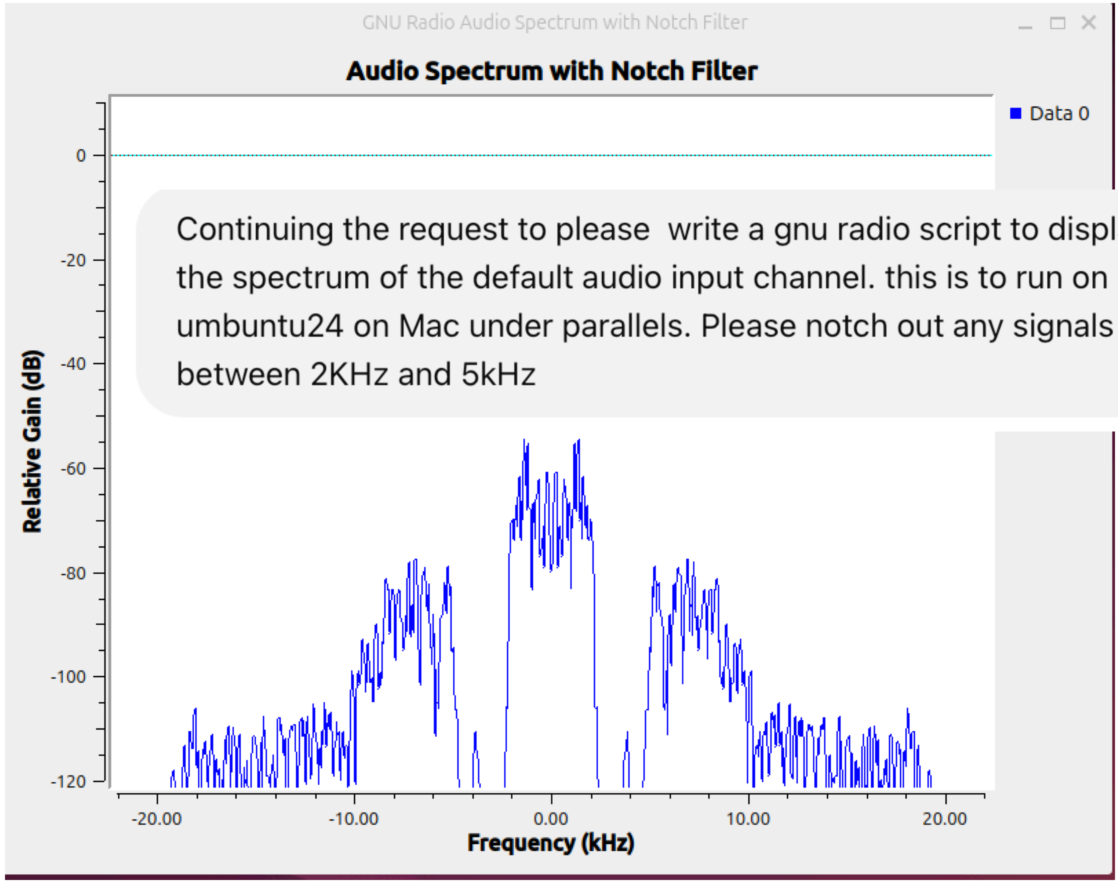


```
!/usr/bin/env python3
rom gnuradio import gr
rom gnuradio import audio
rom gnuradio import qtgui
rom gnuradio import blocks
rom gnuradio.filter import window
rom PyQt5 import Qt
mport sip
mport sys

lass AudioSpectrum(gr.top_block):
   def __init__(self):
       gr.top_block.__init__(self, "Audio Spectrum Display")

       # Audio input source (default device)
       self.audio_source = audio.source(44100, '', True)  # 44100 sample rate,
default device

       # Convert audio (float) to complex
       self.float_to_complex = blocks.float_to_complex()

       # FFT sink (spectrum display)
       self.fft_sink = qtgui.freq_sink_c(
           1024,              # number of FFT bins
```

Pressed write a gnu radio script to display the spectrum of the default audio input channel

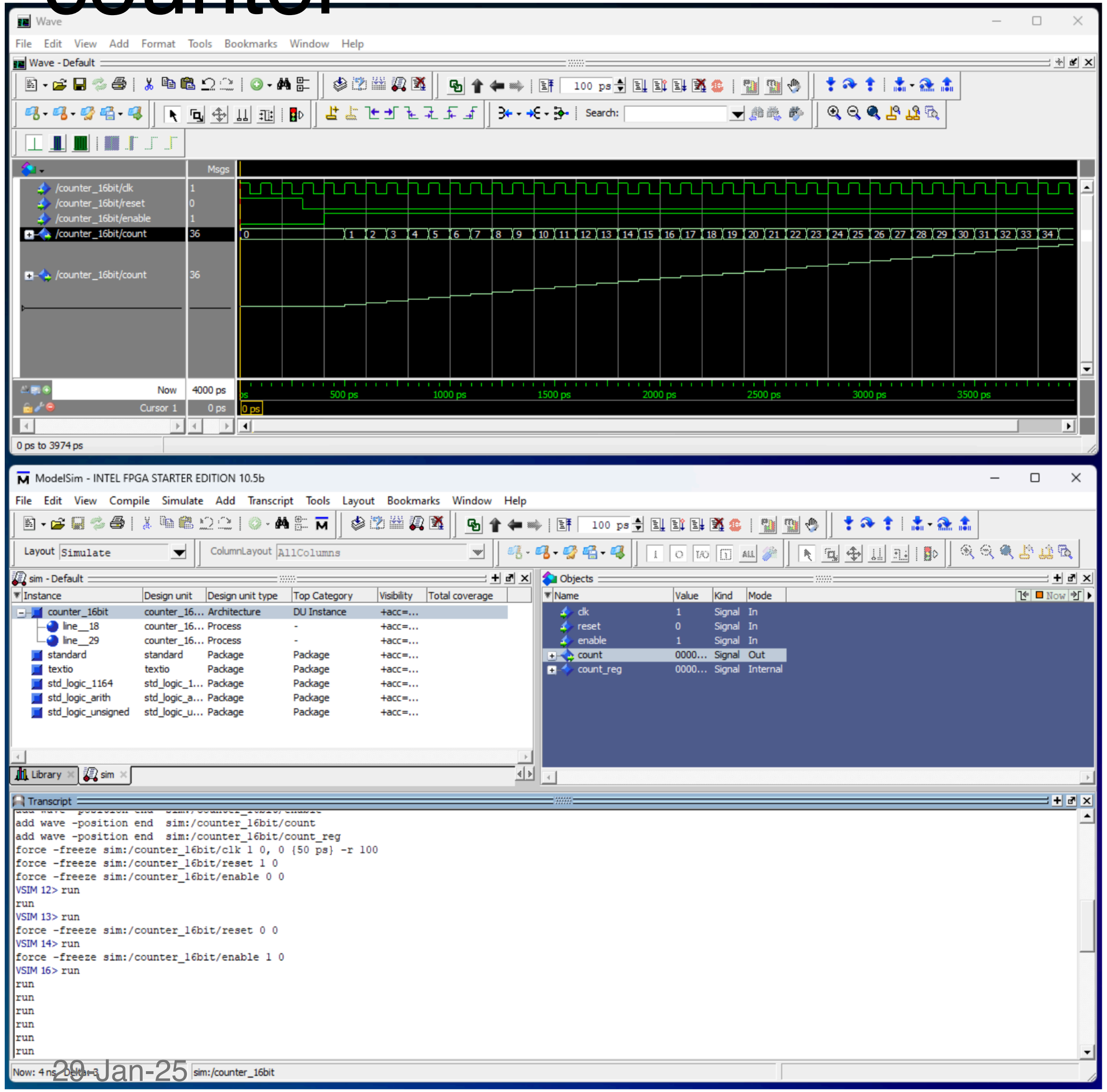
Audio spectrum with & without notch





Continuing the request to please write a gnu radio script to display the spectrum of the default audio input channel. this is to run on umbuntu24 on Mac under parallels. Please notch out any signals between 2KHz and 5kHz

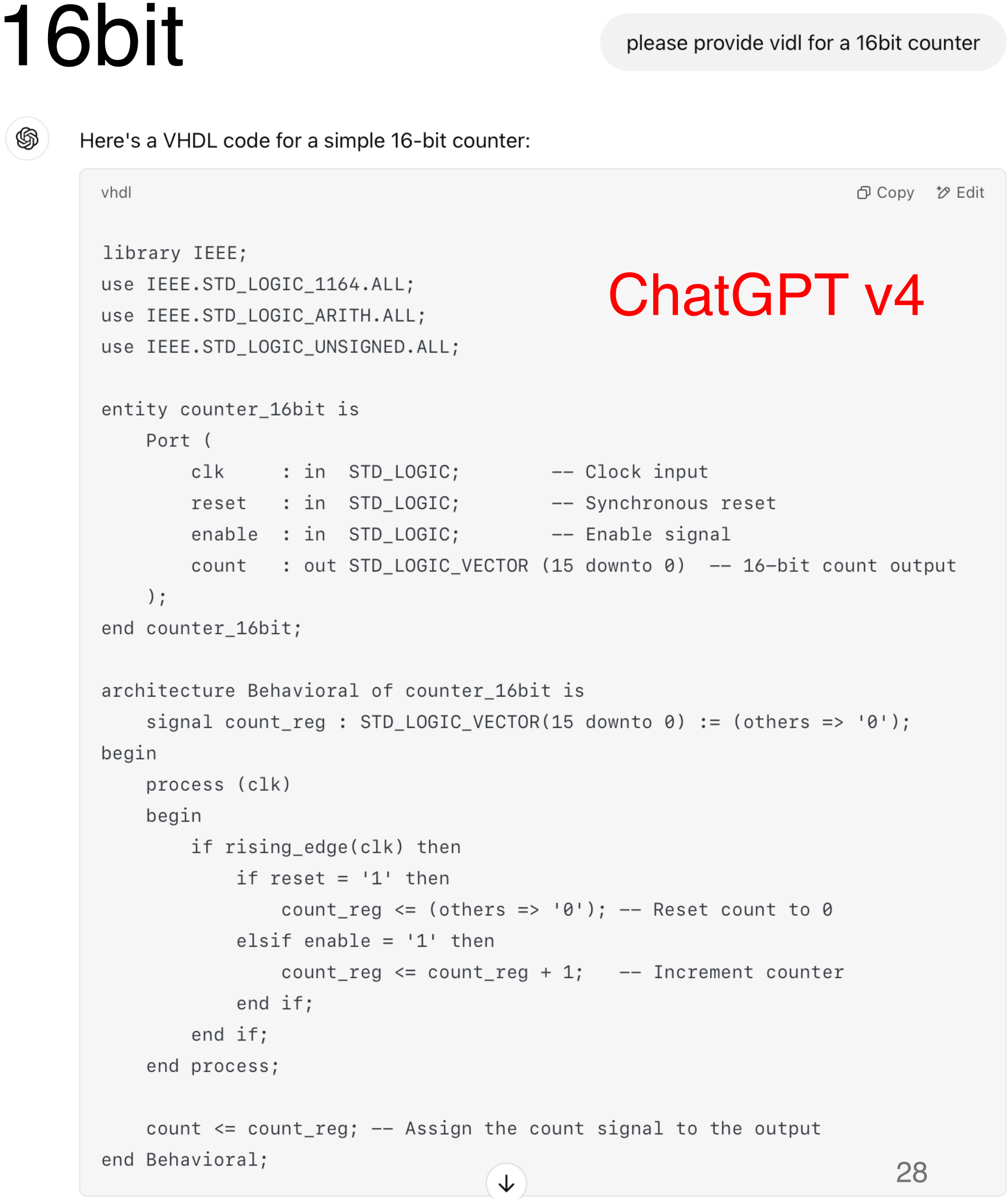
Please provide VHDL for a 16bit counter



Instance	Design unit	Design unit type	Top Category	Visibility	Total coverage
counter_16bit	counter_16…	Architecture	DU Instance	+acc=…	
line__18	counter_16…	Process	-	+acc=…	
line__29	counter_16…	Process	-	+acc=…	
standard	standard	Package	Package	+acc=…	
textio	textio	Package	Package	+acc=…	
std_logic_1164	std_logic_1…	Package	Package	+acc=…	
std_logic_arith	std_logic_a…	Package	Package	+acc=…	
std_logic_unsigned	std_logic_u…	Package	Package	+acc=…	

Name	Value	Kind	Mode
clk	1	Signal	In
reset	0	Signal	In
enable	1	Signal	In
count	0000…	Signal	Out
count_reg	0000…	Signal	Internal

```
add wave -position end  sim:/counter_16bit/count
add wave -position end  sim:/counter_16bit/count_reg
force -freeze sim:/counter_16bit/clk 1 0, 0 {50 ps} -r 100
force -freeze sim:/counter_16bit/reset 1 0
force -freeze sim:/counter_16bit/enable 0 0
VSIM 12> run
run
VSIM 13> run
force -freeze sim:/counter_16bit/reset 0 0
VSIM 14> run
force -freeze sim:/counter_16bit/enable 1 0
VSIM 16> run
run
run
run
run
run
run
```



please provide vidl for a 16bit counter

Here's a VHDL code for a simple 16-bit counter:

ChatGPT v4

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity counter_16bit is
    Port (
        clk     : in  STD_LOGIC;         -- Clock input
        reset   : in  STD_LOGIC;         -- Synchronous reset
        enable  : in  STD_LOGIC;         -- Enable signal
        count   : out STD_LOGIC_VECTOR (15 downto 0)  -- 16-bit count output
    );
end counter_16bit;

architecture Behavioral of counter_16bit is
    signal count_reg : STD_LOGIC_VECTOR(15 downto 0) := (others => '0');
begin
    process (clk)
    begin
        if rising_edge(clk) then
            if reset = '1' then
                count_reg <= (others => '0'); -- Reset count to 0
            elsif enable = '1' then
                count_reg <= count_reg + 1;   -- Increment counter
            end if;
        end if;
    end process;

    count <= count_reg; -- Assign the count signal to the output
end Behavioral;
```

Intellectual Property Rights

- Describing a problem to a public GenAI system is a data leak. It would be unwise apply public GenAI technology to a sensitive project.
- The search and tuning of the answer is almost certainly feeding the AI machine and making this available to others.
- The solution received back from a GenAI system may contain copyright or otherwise IPR encumbered material. You should discuss with your lawyer.
- GenAI services are now being marketed with varying degrees of privacy and selected data set learning.

Conclusion

- GenAI based code generators are going to be a game changer in engineering.
- Unlike natural language generation systematic verification of output is possible.
- They are already as the point where they are useful in at least small applications and procedure/function design.
- This is a VERY fast-moving area, possibly in the hype cycle, but the evidence is that they are already a benefit in:
  - Allowing non-specialists to competently solve problems
  - Improving the productivity of existing specialists.

©Radio Society of Great Britain

Questions?

References

- Radom plus, Autumn 2024, Vol 9 #2
  - rsgb.org/emerging-technology
- https://chatgpt.com
- https://gemini.google.com/app
- stewart@g3ysx.org.uk

©Radio Society of Great Britain