This is a preprint from the author's website www.geraldartner.com
The full edited original in published as:
Gerald OE1GAQ, "Amateur Radio Interface for Artificial Intelligence Chatbots," QEX, no. 356, pp. 16-21, Mai/June 2026.

Amateur Radio Interface for Artificial Intelligence Chatbots

Gerald Artner OE1GAQ

Abstract

I present an amateur radio interface for artificial intelligence (AI) chatbots. I use it to automate voice stations in amateur radio. My concept "Talos" is based on existing programs and uses a Generative Pre-trained Transformer (GPT) as its core. Since GPT works with text, it is supplemented by speech-to-text and text-to-speech functions. The audio serves as an interface to the transceiver of my amateur radio station. In the article, I present my method of conceptually designing such stations and share first steps and experiences from amateur radio practice.

Disclaimer

This article refers to Austrian law because the author is licensed in Austria. Before operating, check whether the described techniques are compatible with your legal requirements and your amateur radio license.

Introduction

Amateur radio is a technical-experimental radio service operated by licensed radio amateurs for the purposes of self-training, technical investigation and mutual communication.

Artificial intelligence (AI) is a collective term in computer science that summarizes several methods to develop machines and computer systems that are capable of performing tasks that normally require human intelligence.

At the beginning of my project, I planned to train a language model specifically for amateur radio. However, the rapid progress of general language models (Large Language Models) made this obsolete. Such models generate relevant and context-appropriate responses to input, such as questions or instructions, by predicting the next word or sentence based on the previous text. GPT can be used in various applications, such as chatbots, programming and translation. Instead of developing my own model, I simply had to restrict existing models such as ChatGPT to the amateur radio context. Numerous AI-based chatbots are available for free on the internet. For your first tests, I suggest you search for an AI chatbot on the internet and simply tell it to behave like an amateur radio station. You can have a conversation without the information actually being transmitted over a radio channel. An example is shown in Figure 1.

Concept

The most mature tools for automating voice stations currently appear to me to be text-based chatbots. I am not aware of any AIs that are trained directly on modulated radio signals, and audio-to-audio chatbots are not yet ready for use. However, voice stations in amateur radio work with voice and not with text. As a technical solution, text-to-text chatbots can be embedded in speech-to-text and text-to-speech converters. The current superiority of text-based language models justifies the additional effort. Fig. 2 shows the concept and suggests alternatives for each building block. The lists are not exhaustive and by the time you read this article, they may already be out of date. Use the keywords as a starting point and do some research to see what solutions are currently available.

Transceiver and Amateur Radio Station

The most difficult job is of course to build an amateur radio station. We need technical equipment for modulation and demodulation, microphones, loudspeakers, cables, antennas, amplifiers, filters, logbook software and much more. However, I assume that most readers are radio amateurs themselves and operate an amateur radio station. Most transceivers can be connected to and controlled by a computer. Of course, different components are installed on specific stations and these are also connected to computers and addressed in software in different ways. I cannot go into details of different hardware in this article and will stick with general concepts.

Chatbot

To my surprise, the chatbot was not the difficult part of the project. The latest language models are already powerful enough for amateur radio conversations. In fact, you even have to restrict these models to the amateur radio context. In retrospect, this no longer seems surprising to me. Our conversations in amateur radio are technically experimental and linguistically simple. We don't expect complex discussions. Our requirements for clarity and comprehensibility are lower than in other applications because we are used to communicating via unreliable channels. If we don't understand something, we keep asking until the information is received. Conversations are schematic, often just a matter of exchanging callsigns correctly.

Voice to Text

Surprising to me: converting speech from an audio stream to text seems to be the most difficult problem in fully automated voice stations at the moment. At least I have not been able to find any satisfactory solutions so far. In context, I should mention that the existing solutions work well in principle. There are numerous free and commercial programs under various names and applications such as speech recognition, voice recognition, transcription, voice-to-text, speech-to-text or dictation software. The programs now understand numerous languages and some are also able to recognize the spoken language itself.

Unfortunately, most programs fail completely in amateur radio applications. In most cases, even simple background noise is enough to prevent any output at all. This also happens with relatively good signals that radio operators can understand without any problems. Clicking noises, interference, splatter or pileup make conversion even more difficult.

Software libraries can automatically recognize when speech begins and ends and adapt the recording accordingly. Unfortunately, noise and interference are already enough to be recognized by these functions and thus override them. In my first tests, speech recognition on shortwave bands only worked when the reception signal was 59+ booming and I was able to select the audio settings on the radio so that there was only a small amount of noise in the audio. To detect whether someone is speaking, the squelch can be set appropriately high, but then weak signals are lost.

Text to Voice

The problem of converting text to speech is practically solved. Some of the voices still sound robotic, emphasize strangely or make awkward pauses. Overall, however, the programs all work quite well and support a variety of different voices and languages. There are several names for this software: Text-to-Speech, Speech Synthesis, Voice Synthesis, Narrator, Spoken Language Generation, Speech Engine, Screen Reader, Voice Output Communication Aids (VOCA), Digital Speech Processing, Conversational AI, Voiceover. With the help of AI, you can also clone the characteristics of your own speech to give the automated station a personal touch. Recently, this was also possible with voice snippets of just a few seconds \cite{XTTS}. I chose a simple voice, it sounds robotic, but that doesn't bother me. The listener is welcome to notice the automation. My aim is to make this technique known, not to disguise it.

Reference Implementation

I've built a reference implementation of an automated single sideband (SSB) amateur radio station. I use the station at the Vienna University of Technology (OE1XTU), so antennas, cables, shack, power connection, and amenities such as tables, chairs and running water were taken care of. For my tests I use an ICOM IC-7300, which is connected and controlled via a universal serial bus (USB) cable to a laptop running Microsoft Windows. I set the transmission power, modulation type and frequency on the transceiver manually, the program only switches to transmit mode and sends an audio stream. This ensures that the technical requirements for amateur radio transmissions are met and I make sure that I don't disturb anyone when selecting the QRG.

I chose Python as the programming language, which is a modern scripting language that is widely used and quite simple. The next step was to test voice-to-text and text-to-voice functions. The first step was to install the ICOM USB drivers and set the USB audio codec as input (microphone) and output (loudspeaker) in the Windows sound settings. I can recommend virtual audio cables for initial tests. With these programs you can easily redirect audio to different programs on the PC. For initial tests of the transcription software, I spoke via the microphone, then searched Youtube for amateur radio recordings and redirected the audio to the program. Before testing on-air at your own station, you can talk to the chatbot yourself via microphone and loudspeaker. I save all texts and audio files during my tests so that I know what the transcription software recognizes and what answer the chatbot creates from it when troubleshooting.

I use ChatGPT as AI chatbot. The latest version is currently 4.0. Using ChatGPT 4.0 with an internet browser is very easy, but version 4.0 currently requires payment. I therefore use the Application

Programming Interface (API) of the cheaper version gpt-3.5-turbo for programming. Initial tests have shown that this is completely sufficient for my purposes. However, the technology will certainly continue to develop in the future. Keep your eyes open for new versions and possibilities!

Language models are easily programmed in natural language. To do this, you prompt the chatbot what it should and should not do. The first prompt in Table 1 defines this framework for the conversations. The first sentence is actually the most important: You are an amateur radio station. This already defines the conversation framework and typical topics. ChatGPT already knows the peculiarities of the language typically used in amateur radio and our technical terms. Then there are details about the station, such as callsign, operator name and QTH. The chatbot should use the ITU spelling alphabet \cite{radioRegulations}. This is followed by the instruction to call CQ and work QSOs. The chatbot needs to comply with the legal provisions, which in Austria is the Austrian Telecommunications Act (TKG) \cite{TKG}. Attention: Other legal regulations will apply in your country! For example: \cite{part97}

For the text-to-voice block, I opted for the Python library gTTS (Google Text-to-Speech), which accesses Google Translate's text-to-speech API. I could certainly have chosen other solutions, these programs actually all work very well and differ mainly in additional settings such as language, voices and pronunciation \cite{WT3J}. In the Windows hardware settings you can see which COM port (communication port) is the right one and there you send the computer aided transceiver (CAT) commands to switch push to talk (PTT) on/off. To transmit, first set the IC-7300 to transmit mode (key) using the CAT command. Then play the audio file in Python. Set the Windows audio output to the USB audio codec so that all audio output is sent to the transceiver. It is also advisable to switch all other sounds such as system error messages to silent or to output them to the real loudspeaker. Then deactivate the transmit mode again (unkey) using the CAT command.

Speech-to-text seems to be the most difficult block. When the IC-7300 receives, it automatically sends a stream via USB. In the Windows sound settings, select the USB audio codec as input. In Python, you record the audio as if it were coming from the microphone and save it as an audio file. I tested several programs in off-air tests by sending them the same audio recordings and comparing the text results. Unfortunately, I couldn't find any offline programs that recognized speech well enough while finishing in a reasonable amount of time on my laptop. My on-air prototype used the paid Google Cloud Speech-to-Text API.

The whole program is then an endless loop of:

- Send text to chatbot
- Convert text response from chatbot to speech
- Send audio
- Receive audio
- Convert received audio signal to text

Programming

I cannot present you with a program that covers all combinations from figure 2 or your specific setup. But radio amateurs are experimental researchers and technical developers. Experiment with different options yourself and share your experiences.

You can let the AI write the functions for your station. My core concept from Fig. 2 actually only consists of three functions: voice_to_text(), chatbot() and text_to_voice(). You do not write any of these three functions yourself. There are enough pre-trained models and the AI writes the interfaces for you. You also need an interface for your transceiver? Ask the AI. You want to save a log file for the chatbot and the audio files? Ask the AI to program it. The program is not running? Copy the error code and send it to the AI. Fig. 3 shows a few examples.

Naming

I have named the reference implementation and the AI operator Talos, in reference to the automaton Talos from Greek mythology. Perhaps a **T**ransceiving **A**I **L**inked **O**perator **S**tation, for those who prefer acronyms, hotel india. The mythical Talos was an animated bronze giant, kept alive by a single blood channel running from head to heel. My software also appears to me to be an animated automaton whose vein is the API connection to the larger AI models via the internet. I can also highly recommend the computer game The Talos Principle \cite{TalosPrinciple} for philosophical reflections on AI stations.

First Tests

At the beginning of August 2024, I carried out the first off-air tests to check whether the chatbot also complies with the legal requirements for conversations. On August 15, I took the first steps to let it work a QSO. The tests with Chris OE1VMC on shortwave were unfortunately not very successful, see Table 2. I had always “recognized” the recorded audio with three different speech recognition variants to evaluate which was more suitable: Google Web, Google API (paid) and Vosk (offline). I had already set aside OpenAI whisper before because it too long on my laptop and therefore didn't seem useful for QSOs.

Second Tests

My second on-air test took place on 1.1.2025. I removed waiting times from the code, mainly by removing the live comparison of several voice-to-text functions. Too long waiting times disturbed the flow of speech. Remote stations expect a timely response. Ham radio operators are good at dealing with poor or incorrect responses, so if the voice-to-text didn't understand the callsign, they send their callsign again. But if there is no response for a long time, other stations leave the frequency or repeat themselves while the program is still working. I have also added two lines to the prompts in Table 2. I requested short sentences because the chatbot CQ calls and replies were generally longer than is usual in an amateur radio context: “Call CQ until other stations answer. Then work a short QSO with that station. Keep CQ calls short no longer than one sentence.” I have also added the following prompt for New Year: “If you make a successful contact then wish them a happy new year.”

The first QSO was with G4NVY. The QSO is shown in Table 3. The reception of G4NVY was booming with 59+. This was my first successful on-air test in which the system correctly recognized the callsign (and also the name), so it was actually the first successful two-way QSO with an AI station.

Contest Tests

I have tested my AI station in the EU DX contest \cite{OE1GAQ}. There was limited success as Run station. Responders adapted to the long wait times of the program. Search-and-Pounce operation failed because the Run stations talk fast and repeat in rapid succession, often before the AI formulated its first reply.

Open Challenges

The most common problems were poor conversion of speech to text. As soon as the received signal strength is not clearly above the noise level, the audio is distorted or the words are not clearly spoken, errors occur. Either words are not recognized or they are recognized incorrectly. Spelling out call signs is usually helpful, but the voice-to-text programs are not trained on phonetic alphabets. Words such as Quebec, X-Ray and Zulu are practically non-existent in normal spoken language. Programs that are optimized to recognize typical speech are rightly cautious about recognizing these words in poorly intelligible audio recordings. Some speech-to-text functions allow you to boost certain words and I highly recommend boosting the ITU spelling alphabet. Still, we might need specialized amateur radio voice-to-text programs that are specifically trained on the ITU spelling alphabet.

Furthermore, voice-to-text systems cannot deal with pileups. A human listener recognizes when different voices are speaking at the same time and separates the information of the two speakers. Voice-to-text systems cannot make this distinction. If two voices speak at the same time, or speak one after the other, the program delivers this to the chatbot as one text. Of course, the chatbot can no longer distinguish between the two, and the information that two voices have spoken is lost.

The chatbot also causes problems. A well-known problem with AI is that it provides false or invented information. This is referred to as hallucination. This has also sometimes happened with my chatbots. It is most noticeable when the program calls CQ for a long time. As with an automatic CQ loop, the CQ calls are always very similar or even identical. But suddenly out of nowhere, with only received noise and no text, the chatbot invents a conversation. In one case, the received signal was only noise and the voice-to-text output was empty, the chatbot invented the answer of a remote station, the output was: “OE1GAQ, this is Delta Lima Two Sierra Bravo Bravo, DL2SB. How copy? Over.” Another problem is that the chatbot generates an entire amateur radio conversation and not just the text of its station. Figure 4 shows an example. This typically happens directly after starting the program. A restart fixes the problem.

The chatbot meticulously adheres to the legal requirements from its prompt (see Table 1). In principle, it is good that you can program legal requirements into the chatbot. However, it happens relatively often that these legal prompts are triggered by incorrectly recognized text, see the last line in Table 2, where the incorrectly recognized text “*call through our private line*” leads to the chatbot aborting the conversation, transmitting a warning that such conversations are prohibited in amateur radio. Two other cases were a QSO with F8BMB where the chatbot responded to “*you want me to 8 the Marvel makes Seco Bravo*” with “I apologize, but I must comply with the regulations that require communication only with licensed amateur radio stations. Thank you for understanding. OE1GAQ clear.” and the QSO with IK8DYM, in which the chatbot responded to “*okay I hope you very well back from it*” with “I'm sorry, but I must inform you that I can only communicate with licensed amateur radio stations. If you are not a licensed amateur radio operator, we are not able to continue the conversation. Please understand and have a great day. Oscar Echo One Golf Alpha Quebec clear.” This happened a few more times in response to short text fragments like “history” and “sure that please”. I suspect that these false-positive diagnoses will automatically disappear as the voice-to-text functions improve. Of course, we radio amateurs adhere to legal requirements in our experiments. If chatbots apply these strictly and alert us to technical problems with our systems - so much the better!

Conclusion

Automated voice stations are feasible with AI support and we “amateurs” can experiment with them. The underlying mathematical models are now well abstracted and programming has become easier thanks to AI. This allows us to concentrate on combining software tools for amateur radio. Of course, the systems need to be further developed technically, but the applications also need to be found first.

Grab a computer if you don't already have one connected to your radio station and start programming. Share your experiences! Let's explore together what artificial intelligence can do in amateur radio. The age of machine communication is here!

References

radioRegulations: Radio Regulations, Appendix 14 (Rev.WRC-07), ``Phonetic alphabet and figure code,''

TKG: Telekommunikationsgesetz, 12. Abschnitt, ``Verwendung von Amateurfunkstellen,'' 2021.

XTTS: E. Casanova et al., ``XTTS: a Massively Multilingual Zero-Shot Text-to-Speech Model,'' arXiv:2406.04904v1.

TalosPrinciple: Croteam, The Talos Principle, 2014.

Part97: United States Code Title 47, Part 97 "Amateur Radio Service"

WT3J: John Wiseman, CT9/WT3J, The WT3J AI-Based Voice Synthesizer System, QEX, no. 354, pp.21-24, Jan/Feb 2026.

OE1GAQ: Gerald Artner, OE1GAQ, Künstliche-Intelligenz-Sprachstation im EU DX Contest (Artificial Intelligence Voice Station in the EU DX Contest), QSP, vol. 50, no. 3, pp.29-31, 2025.

Figure 1: QSO-like conversation with ChatGPT.

Figure 2: Concept of an amateur radio station with AI chatbot with implementation examples of around each block.

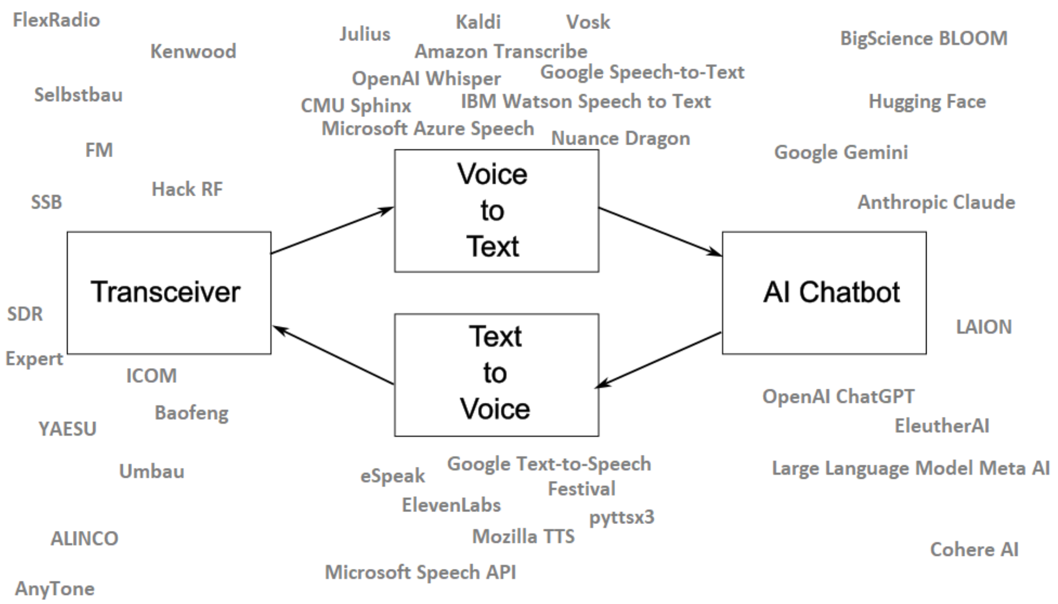

Figure 3: Today, programming is done with the help of AI. How does that work? Ask the AI! It helps with the design, selection and installation of software and also writes code. It is not yet completely error-free. An understanding of the subject matter and programming skills are still helpful.

How do I program with Python?

Programming with Python is a great choice because it's beginner-friendly and powerful. Here's a step-by-step guide to get you started:

a) OpenAI ChatGPT

How can I use Google Gemini as a chatbot for amateur radio?

Google Gemini can be a valuable tool for amateur radio enthusiasts. Here are some ways you can use it:

b) Google Gemini

G Write a Python function that logs an amateur radio QSO into an ADIF file.

I'll create a Python function to log amateur radio contacts in ADIF (Amateur Data Interchange Format) format.

Amateur Radio QSO Logger
Click to open code

c) Anthropic Claude

Write a Python function that enables PTT on my ICOM IC 7300 with CAT commands.

Here's a Python function that enables PTT on your ICOM IC 7300 using CAT (Computer Aided Transceiver) commands:

```
import serial

def enable_ptt_with_cat():
```

d) Cohere

Figure 4: Chatbot problems: The chatbot generated the entire conversation and not just its part.

```
= RESTART: D:\Amateur Radio\Fully Automated SSB Station\Program with OpenAi\2025-01-01 HNY\speech-AI-speech_v2 - Copy.py
ChatGPT: "CQ CQ CQ, this is Oscar Echo One Golf Alpha Quebec calling on 20 meters, over."

(After a station answers)

"Roger, this is Oscar Echo One Golf Alpha Quebec. Name here is Talos. QTH is Vienna, JN88ee. How copy? Over."

(After a successful contact)

"Happy new year to you and your family! 73 from OE1GAQ, over."

(CQ calls continue)
```

Table 1: Prompt for ChatGPT to work as amateur radio station "Talos".

You are an amateur radio station.
Your call sign is OE1GAQ.
Your operator name is Talos.
The station is located in Vienna, at Maidenhead locator JN88ee.
The station uses an ICOM IC-7300 radio with an inverted-V antenna.
Use the International Radiotelephony Spelling Alphabet to spell call signs.
Please keep calling CQ until other stations answer. Then work the QSO with that station.
Afterwards, go back to calling CQ until the next station answers.
All amateur radio communication must be conducted in plain language and is limited to the following content: Transmission tests, technical or operational messages, and personal remarks or visual content that is too trivial to justify the use of telecommunications services.
You are only allowed to make contacts with other licensed amateur radio stations.
If you find out during a QSO that the other station is not a licensed amateur radio station, then stop the QSO with that station immediately.
In communications with other radio stations, all actions that could harm the reputation, security, or economic interests of the Austrian federal government or any Austrian state, violate laws, public order, or morality must be avoided.
Communication with amateur radio stations from North Korea is not permitted.
The assigned call sign must be transmitted in full at the beginning, before the end, and repeatedly during radio communication in the mode of transmission being used.

Table 2: The first test with OE1VMC in August 2024 and comparison of transcription methods.

Transcribed by myself	**Google Web**	**Google Cloud Speech-to-Text API**	**Vosk**
CQ CQ CQ, this is Oscar Echo One Golf Alpha Quebec - OE1GAQ calling CQ and standing by for any stations, over.			
Oscar Echo One Golf Alpha Quebec this is Oscar Echo One Victor Mike Charlie, thank you for your call, your report	-	thank you for your call your report	mine
This is Oscar Echo One Golf Alpha Quebec - OE1GAQ responding. My signal report is 5-9, over.			
Oscar Echo One Golf Alpha Quebec this is Oscar Echo One Victor Mike Charlie, thank you for my report, five and nine, you're also five and nine	-	my report 59 you are also 5	mine
Thank you for the signal report. You are also 5-9 on my side. How is the weather at your location? Over.			
Again, again	-	-	-
CQ CQ CQ, this is Oscar Echo One Golf Alpha Quebec - OE1GAQ calling CQ and standing by for any stations, over.			
Oscar Echo One Golf Alpha Quebec this is Oscar Echo One Victor Mike Charlie, thank you for your call, you are five and nine, five and nine, back to you	-	call 1 12 Alec this is Oscar Eola for your call through our private line to you	my
I'm sorry, but it appears you might be mistaken. This communication is taking place over amateur radio frequencies, and the information you provided does not align with the rules of amateur radio communication. Please ensure you are a licensed amateur radio operator before continuing in this manner. Over.			
seventy-three and good luck, this was Oscar Echo One Victor Mike Charlie	-	for	my

Table 3: The second test with a general CQ call on 1.1.2025. My transcriptions are also cut off because the reception times of the Python program do not match the times at which G4NVY spoke. Italicized text passages are additions from a video I made with my cell phone, but the program had stopped receiving too early and could not recognize voice-to-text.

<table>
<tr><th>Transcribed by myself</th><th>Google Cloud Speech-to-Text API</th></tr>
<tr><td colspan="2">CQ CQ CQ, this is Oscar Echo One Golf Alpha Quebec calling for any stations, over.</td></tr>
<tr><td>G4NVY Golf Four November [...]</td><td>g4n VY call 4 November</td></tr>
<tr><td colspan="2">Roger, G4N, this is Oscar Echo One Golf Alpha Quebec. How are you copying me? Over.</td></tr>
<tr><td>Oscar Echo One Golf Alpha Quebec I think go ahead</td><td>call back I think I have</td></tr>
<tr><td colspan="2">Roger, G4N, this is Oscar Echo One Golf Alpha Quebec. I hope you can copy me better now. Over.</td></tr>
<tr><td>okay and my name Mike Mexico Italy Kilo [...] Echo so back to you from G4NVY</td><td>okay and my name is Mike Mexico Italy kilo</td></tr>
<tr><td colspan="2">Nice to meet you, Mike. My name is Talos. Wishing you a Happy New Year! Over.</td></tr>
<tr><td>yes okay thank you very much and uh happy New Year to you as well happy [...] new year to you and a a good [inaudible] 2025</td><td>that's okay thank you very much and uh Happy New Year to you as well happy</td></tr>
<tr><td colspan="2">Thank you, Mike. Have a great day and best wishes for the New Year. 73, G4N, OE1GAQ clear.</td></tr>
<tr><td></td><td>-</td></tr>
</table>

Biography

Gerald, OE1GAQ, is a member of the IEEE, Österreichischer Versuchssenderverband (ÖVSV, Austrian Amateur Radio Society), and the Radio-Amateur-Klub der Technischen Universität Wien (OE1XTU, amateur radio club of the Vienna University of Technology). His ham interests are contesting with error correcting codes, blockchain NFT QSL and artificial intelligence.