elektor
design > share > earn
Declassified BONUS EDITION
534B
SINCE 1961

AI Guest-Edited Edition

Free Brand-New Bonus Articles Every Week!

Week 50

Available in week 51

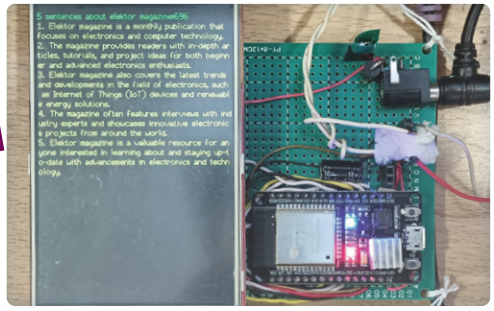
Available in week 1 (2025)

Available in week 51

High-Performance Edge AI

The New STM32N6

Available in week 1 (2025)



Available in week 52

A Personal AI Terminal

ChatGPT Gets a Voice

ASUS IoT PE8000G

REVOLUTIONARY EDGE AI PERFORMANCE

Robust and reliable

The PE8000G complies with the MIL-STD-810H standard and is therefore protected against extreme temperatures, vibrations and voltage fluctuations - ideal for use in harsh industrial environments.

Powerful dual GPU support

Process multiple neural networks simultaneously thanks to the support of two 450-watt GPUs - seamlessly and in real time. This allows you to complete even complex tasks efficiently and quickly.

Wide range of applications

From AI-driven factory automation to intelligent video analytics, the PE8000G delivers the computing power and flexibility you need.

Flexible power supply

With a DC input of 8-48V and integrated ignition control, the PE8000G is ideally equipped for a wide range of industrial requirements.

FORTEC INTEGRATED

Interested in our products?

Volume 50
Guest-Edited AI Digital Bonus Edition
ISSN 0932-5468

Elektor Magazine is published 8 times a year by
Elektor International Media b.v.
PO Box 11, 6114 ZG Susteren, The Netherlands
Phone: +31 46 4389444
www.elektor.com | www.elektormagazine.com

For all your questions
service@elektor.com

Become a Member
www.elektormagazine.com/membership

Advertising & Sponsoring
Büsra Kas
Tel. +49 (0)241 95509178
busra.kas@elektor.com
www.elektormagazine.com/advertising

International Editor-in-Chief
Jens Nickel

Content Director
C. J. Abate

Publisher
Erik Jansen

Bonus AI Projects and Articles

Did you really think we were done delivering AI-related content after we announced the 2024 Guest-Edited AI edition of ElektorMag? No way! As regular Elektor readers know, we always have new projects, tutorials, and background articles in the pipeline. With this Bonus edition, which we'll unveil over the course of four weeks, we aim to inspire you to design new AI-related solutions for weeks and months to come.

If you are looking for a project to start experimenting with AI, the article, "AI-Based Universal IR Remote Control," is a great place to start! With the help of a Raspberry Pi 5, Google's MediaPipe Studio platform, and a tiny plug-in interface board, you can have a versatile gesture recognition system to control your TV or other devices operated through IR remote controls.

Can a modern build capture the spirit of Turing's original experiment? In "A Personal AI Terminal," discover how an ESP32 module, paired with a TFT display and an I²S amplifier, becomes the gateway to a 21st-century version of the Turing Test, communicating directly with ChatGPT.

In "AI for Product Concept Design" we look at how AI has revolutionized image creation. Designers can transform sketches and ideas into photorealistic renderings with just a well-crafted text prompt. Dive into this exploration of AI-powered design tools.

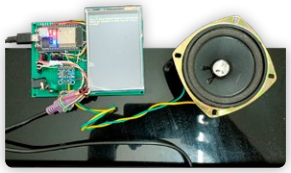
Curious about the history of AI in the pages of Elektor? In this edition, we take you on a journey through Elektor's archives, showcasing how artificial intelligence has evolved within the electronics community. Dive into our editors' top recommendations from past editions of your favorite magazine.

And much more awaits you in this free Bonus edition!

After you dive into this issue and start your own AI-related projects, be sure to share your experiences on the Elektor Labs online platform: *www.elektormagazine.com/labs*. We look forward to learning about your creations!

C. J. Abate (Content Director, Elektor)

THIS EDITION



Available in week 51

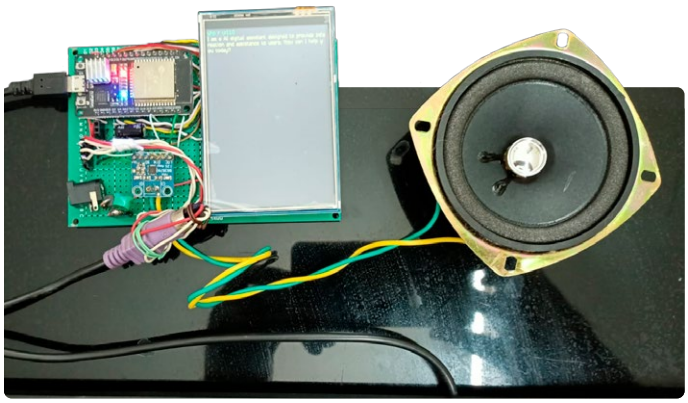
Available in week 52

Available in week 1 (2025)

The Guest-Edited AI Edition 2024 of ElektorMag is available at newsstands and in the Elektor Store.
www.elektor.com/ep-0534

A Personal AI Terminal

ChatGPT Gets a Voice



By Somnath Bera (India)

Whether AI solutions such as ChatGPT can pass the Turing test is still up for debate. Back in the day, Turing imagined a human operator judging replies to questions sent and received using an electromechanical teletype machine. Here we build a 21st-century version of Turing's original experimental concept, using an ESP32 with a keyboard and TFT display to communicate exclusively with ChatGPT via the Internet. In addition, Google text-to-speech together with a tiny I²S amplifier module and speaker lets you listen in to the conversation. In our case, it's obvious from the start that we are communicating with a machine — isn't it?

There is no doubt that AI tools such as OpenAI's ChatGPT and Google's Gemini can be real game changers in so many situations. I have used ChatGPT to develop quite complex control solutions. I provide the initial idea, and as I give more inputs, it refines the code, making it better with each iteration. It can even convert Python code to MicroPython or an Arduino sketch. The key is to guide the process carefully to prevent it from straying too far off course. There are times when it does deviate and repeats the same mistakes, but I actually enjoy finding these bugs and steering ChatGPT's output to more closely match my target outcome.

Hardware

The heart of this build is Espressif's ESP32 development module. Its dual-core architecture has enough processing reserve to take care of Wi-Fi communications, handling the serial input from a PS2 keyboard, sending data to the 3.5-inch TFT display and outputting digital audio data to the I²S module. The complete schematic for the ChatGPT terminal is shown in **Figure 1**.

You may be surprised at the use of an old PS2 keyboard; the only reason is that I was not successful at implementing a USB port for this purpose on the ESP32. In the limited time available, I opted for the simpler PS2 type of interface, which uses far fewer resources. A PS2 socket for the keyboard can be sourced from several web-based electronic component distributors. The PS2 keyboard pin assignments can be seen in the circuit diagram. No doubt there will be developers out there itching to upgrade the system to USB, and I welcome their input.

The 3.5-inch TFT touch display used has a parallel interface rather than the alternative SPI version. This inevitably uses more interconnect wires, but in this application, we have enough spare GPIOs and the interface offers a significantly faster response time. The touch screen features are not used in this application.

A MAX98357A I²S amplifier (**Figure 2**) module takes the digital audio I²S signal generated by the ESP32 and converts it into analog audio. A built-in, 3-W class D amplifier boosts the signal to achieve a good level of sound. The gain input is tied to ground, which produces maximum volume from the amp. The class D amp output can drive a 4-Ω speaker.

Figure 2: The audio amp module converts I²S digital audio to 3 W of audio.

Don't Lose the Key

ChatGPT was developed by OpenAI. It generates text-based responses to prompts or questions entered by the user. Usually, it works through a browser interface. The OpenAI API has been designed for use by developers and businesses so that they can integrate AI features into their own applications, software, or websites. It's not just a chatbot interface, but a flexible programming interface allowing businesses to embed chatGPT's capabilities into their products. Developers send data to the API programmatically and get responses back. To use the OpenAI API, you will need a unique key, which is generated for your account when you sign up. Go to the OpenAI website [1] and click on the *Sign up* button.

Fill out the registration form with your email address, password, and other required information. Once you have entered these, navigate to your OpenAI account dashboard and click on the *New Project* button. Give your project a name and description (optional). In your project

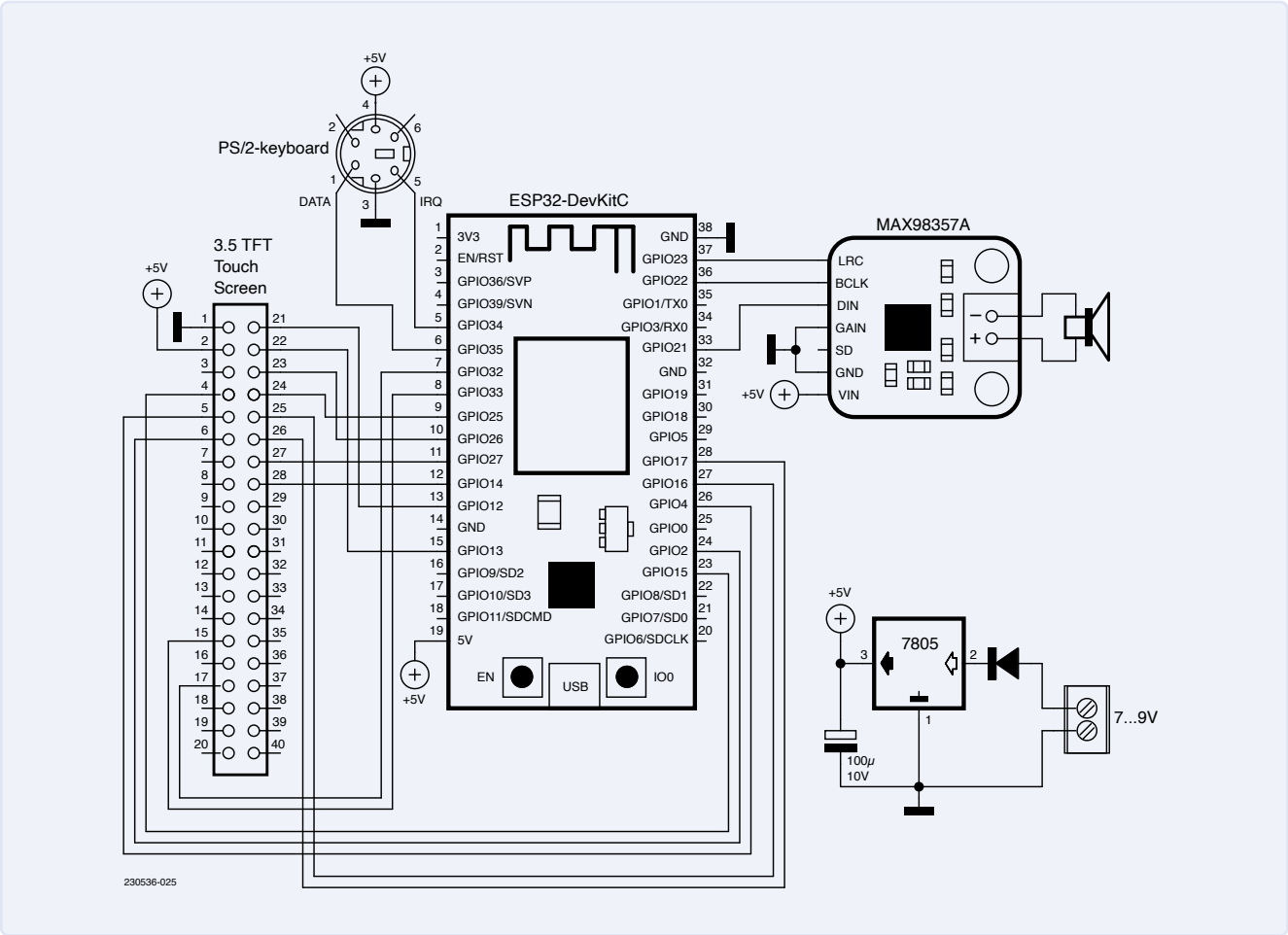


Figure 1: Schematic of the talking ChatGPT terminal.

settings, click on the *API Keys* tab. You'll see a list of existing secret keys (**Figure 3**). Click the *Create new secret key* button to generate a new key. Make a copy of this generated API key and save it securely, as you won't be able to retrieve it again for security reasons. You'll need this key to authenticate your applications with OpenAI's services. On their site there is also a developer's quickstart tutorial which guides you through the process of generating a response via the API using the generated key to authorize your access.

At the time of writing, OpenAI provides free credits when you first sign up, which you can use to experiment with the API. After you use up all these credits, you will need to pay based on your usage. If you haven't already set up a payment method for billing, it will be necessary to do so. Read through OpenAI's usage guidelines and terms of service.

Text to Speech

A Text to Speech (TTS) API is used to convert the textual response from OpenAI into a digital audio data stream. There are a couple of text-to-speech APIs we can use for this. OpenAI has its own TTS API offering a number of different voice alternatives which sound quite natural. To access this API, you use the same key you were allocated to use the OpenAI API.

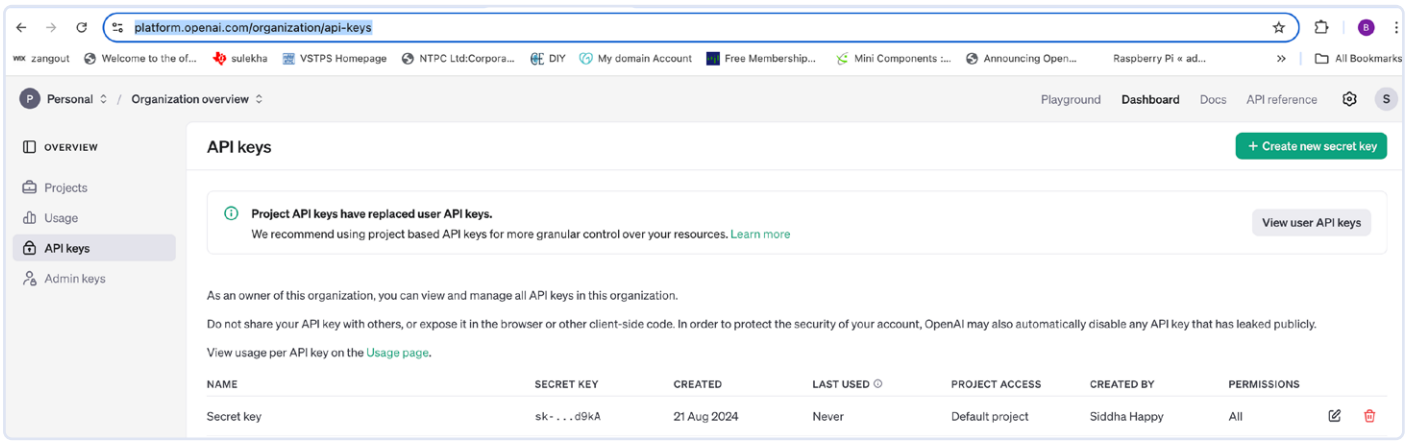


Figure 3: The OpenAI page to register for the OpenAI API.

For this project, we make use of the Google Cloud Text-to-Speech API. It offers a diverse range of voices in various languages and dialects. Compared to OpenAI TTS the voices sound a little more mechanical. Long text strings make the ouput break up. This API is however free to use at the time of writing, whereas OpenAI TTS API incurs a charge.

To start using the Google TTS, we first need to create the project on Google Cloud and enable the Google TTS API to get our API key. Text strings can now be posted to the API using an HTTP POST request along with the key. The resulting I²S digital audio stream is then stored and played back to produce an analog audio signal via the MAX98357A.

Software: Libraries

The Arduino sketch is included [2]. Check all the libraries referenced in the header of the sketch listing to make sure they are installed in your environment, if not install them now using the Arduino IDE library manager. Using the methods available in the *Audio.h* library, it was a simple job to produce the audio output to accompany the word output written to the TFT display. It was only necessary to add a few lines in the loop to generate the audio. Check it out in the Arduino sketch.

```
#include <PS2Keyboard.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <SPI.h>
#include <TFT_eSPI.h> // Hardware-specific library
#include "Audio.h"    //Audio header file

// GPIO connections to I2S board
#define I2S_DOUT  21
#define I2S_BCLK  22
#define I2S_LRC   23

Audio audio;  //audio instance creation
```

The *ArduinoJSON.h* is used to parse the JSON-formatted response data from the OpenAI API into a format that can be used by the Arduino code.

Software: ChatGPT API request

The process of interacting with the OpenAI API is contained in the `makeApiRequest(String prompt1)` function:

1. First, we set up an `HTTP client`:

```
HTTPClient http;
http.setTimeout(24000); // 24 seconds timeout
http.begin("https://api.openai.com/v1/chat/completions");
http.addHeader("Content-Type", "application/json");
http.addHeader("Authorization", "Bearer " +
                  String(api_key));
```

2. Next, prepare the payload which will be JSON formatted:

```
StaticJsonDocument<1024> jsonPayload;
    // setting a maximum size of 1KB
jsonPayload["model"] = "gpt-3.5-turbo";
    // model is gpt-3.5-turbo,
    // change if you have access to 4 or 4.5
jsonPayload["temperature"] = 0.8;
    // randomness of response, the higher
    // the value, the higher the randomness
jsonPayload["max_tokens"] = 2000;
    // maximum words & punctuations
    // limit to be generated by response
```

3. Prepare the message (a nested array of messages), serialize to JSON format and send the request:

```
JsonArray messages = jsonPayload.
    createNestedArray("messages");
    //put in a nested format not random
JsonObject userMessage = messages.createNestedObject();
userMessage["role"] = "user";
userMessage["content"] = prompt1;
String payloadString;
serializeJson(jsonPayload, payloadString);
```

4. Send the request and receive the response:

```
int httpResponseCode = http.POST(payloadString);
    //transfer to the open page
if (httpResponseCode == HTTP_CODE_OK) {
  String response = http.getString();
    // if everything goes OK get the
    // reply = output and put in a string
...
```

5. The web page will generally include a jungle of spurious information (all in HTML code) which is not useful for our application and can be stripped off. Here we normalize the response — remove the tags JSON etc. so that it can read on the TFT display:

```
StaticJsonDocument<1024> jsonResponse;
    //parse the tags etc of the response string.
deserializeJson(jsonResponse, response);
    // and put in a simple stripped string & return
String assistantReply = jsonResponse
    ["choices"][0]["message"]["content"].
    as<String>();
    // select the first part which contains our reply
return assistantReply;
```

Software: Setup and Loop

In the `setup` function we are initialising the TFT, I²S board, connected to the Internet with our Wi-Fi credentials.

```
void setup() {
  delay(300);
  …
```

```
  audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
//I2S board initialised
  audio.setVolume(50);
```

Inside the `loop` function, we sent questions to ChatGPT. When the session is complete we remain inside `loop`:

```
String response = makeApiRequest(msg);
    // sent to Chatgpt
...
if (l1>200) {
  response = response.substring(0, 200);
    // truncate first 200 characters
}
audio.connecttospeech(response.c_str(), "en");
    //speak up the 200 characters
```

The Google TTS has a 200-character limit and will not output any text beyond this limit. To work around this, the answer string is trimmed to 200 characters for Google TTS. This ensures that while the full answer is displayed on the screen, only the first 200 characters are vocalized. For longer answers, the screen scrolls, but this can be adjusted by making slight modifications to the sketch.

Project Testing

The delays I've used inside the software loops are quite specific. You may adjust them, but I recommend starting with the default values used in the code. Once you're comfortable with the responses, feel free to tweak them. I began with simple questions like "Who r u?" ChatGPT responded appropriately, displaying the introduction on the screen, while the speaker articulated it clearly.

I then tested the system using questions such as: "Write 5 sentences about Elektor magazine" (**Figure 4**) and even asked it to write blink sketches for Arduino UNO, ESP32, and Raspberry Pi Pico.

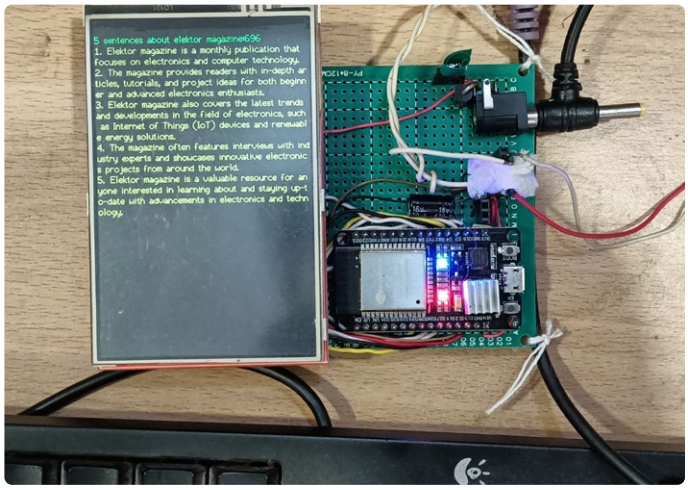


Figure 4: Testing the system with prompts asking for blink sketches for Arduino UNO, ESP32, and Raspberry Pi Pico.

In every case, ChatGPT performed flawlessly, understood the context perfectly and answered accurately, with the speaker delivering the voice output loud and clear.

To Wrap Up

All files associated with this project can be found at [2]. The Internet and web browsers such as Google have totally revolutionized our access to information and replaced those bulky encyclopedias that used to line our bookshelves at home. Now we see the rise of AI software and machines based on ChatGPT API, TensorFlow Lite Micro, Edge Impulse, OpenMV, TinyML which are poised to disrupt existing business models and more traditional methods of problem-solving. We live in interesting times.

230536-01

About the Author

Somnath Bera, a mechanical engineer from Jalpaiguri Govt. Engg. College, India, worked as a General Manager at NTPC, the largest power producer in the country. He has a profound passion for electronics, evidenced by his 60+ innovative projects on Elektor Labs, over 10 of which have been featured in *ElektorMag*. His projects are often focused on problem-solving in areas like waste and natural resource management. Somnath likes to use innovative approaches and platforms like Arduino, Raspberry Pi, and ESP32 coupled with various kinds of sensors and wireless systems to create efficient and cost-effective solutions.

Questions or Comments?

If you have technical questions or comments about this article, feel free to contact the author by email at berasomnath@gmail.com or the Elektor editorial team at editor@elektor.com.

Related Product

- **Espressif ESP32-DevKitC-32E**
  www.elektor.com/20518

WEB LINKS

[1] OpenAI website: https://platform.openai.com
[2] Elektor web page for this article (downloads): https://www.elektormagazine.com/230536-01

High-Performance Edge AI: The New STM32N6

A Game-Changer in Performance and AI Capabilities

Contributed by STMicroelectronics

The new STM32N6 is designed to meet the growing demands for high-performance edge AI, enabling new use cases in computer vision, audio processing, and other advanced applications. It continues to elevate the standards for video and multimedia applications, enhancing user experiences with its rich multimedia capabilities.

The STM32 microcontrollers — based on Arm Cortex CPUs from M0 to M7 — are well known in the electronics community, as they are equipped with an abundant set of peripherals and there is a vast ecosystem of development boards, libraries and software tools for all kind of applications. The new STM32N6 addresses the needs for pre-processing data "on the edge", with the help of AI in form of a Neural Processing Unit (NPU). It opens up new fields of applications for Arm Cortex M processors (**Figure 1**).

STM32N6 — Features and Benefits

- The embedded NPU (ST Neural-ART Accelerator) is the heart of the new STM32N6. It offers an impressive 600 GOPS at a very low-power consumption. This NPU is a game-changer, enabling high-end AI performance (**Figure 2**) on an MCU with a small footprint, low power consumption, and low costs.
- Powered by an Arm Cortex®-M55 core operating at 800 MHz, the STM32N6 provides 1280 DMIPS and 3360 CoreMark, marking the highest performance for an STM32 MCU to date. This makes it ideal for advanced vision and graphics applications, supported by a Chrom-ART Accelerator, NeoChrom Accelerator, H264 encoder, and JPEG encoder/decoder.
- The MCU includes 4.2 MB of embedded RAM, supporting real-time data processing, multitasking, and efficient execution without the need for external memory in most cases. Additionally, it offers a rich peripheral set, including dual

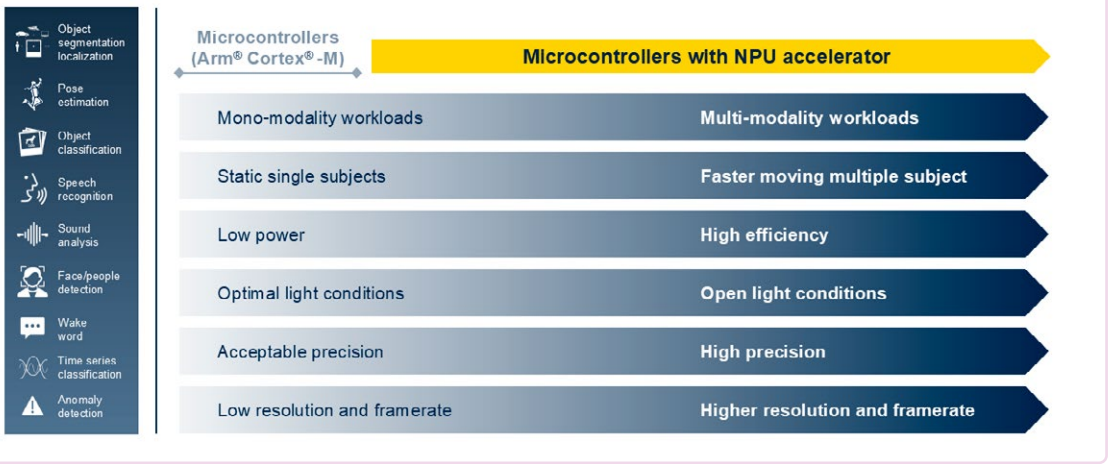


Figure 1: The Neural Processing Unit (NPU) is opening up new fields of applications.

USB, Gbit Ethernet, I3C, and seven UART interfaces, ensuring versatile connectivity options for various applications.

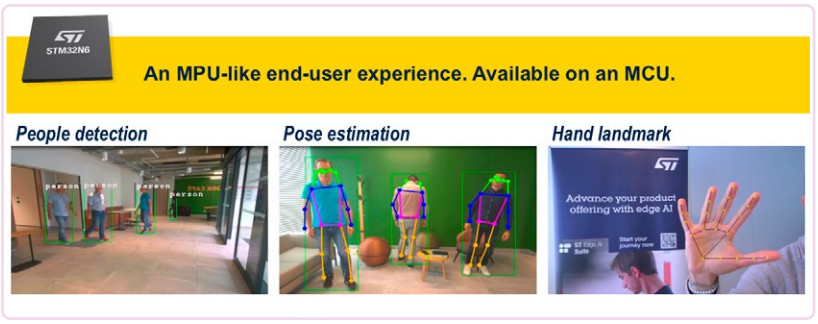
- A dedicated image signal processor (ISP) is dimensioned for a 5 MP camera at 30 fps. Embedded firmware for the Arm Cortex processor allows auto white-balance and auto exposure. A software tool permits fine-tuning of the ISP (**Figure 3**).
- Fully integrated within the STM32 ecosystem, the STM32N6 is supported by the ST Edge AI Suite, including tools like STM32Cube.AI, an AI model zoo, various resources, and case studies. It is also backed by numerous partners, including Edge Impulse, Nota.AI, and EmbedUR, providing a comprehensive development environment.

Comprehensive Software Ecosystem

STMicroelectronics is at the forefront of the edge AI movement, providing comprehensive solutions that cater to the varying needs of engineers and developers across industries. Through its Edge AI Suite, ST offers a wide range of software tools tailored to different user profiles and expertise levels, along with a large collection of case studies to inspire developers.

STM32Cube.AI and ST Edge AI Developer Cloud are designed for users who need great flexibility and customization. These tools provide access to advanced resources for optimizing and deploying custom models (**Figure 4**).

ST also developed a comprehensive model zoo that comes with additional resources like retraining scripts and application code examples. This is a great starting point for developers who want to quickly implement



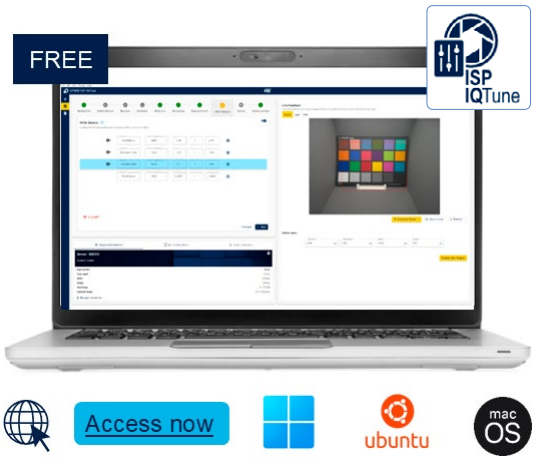


Figure 2: The ST Neural-ART Accelerator enables the execution of advanced AI applications.

Figure 3: A dedicated image signal processor (ISP) is dimensioned for a 5 MP camera at 30 fps and can be fine-tuned via a software tool.

The STM32 Edge AI Contest

Want to use the capabilities of the new STM32N6 in a project of your own? Participate in the **STM32 Edge AI Contest** powered by STMicroelectronics and Elektor! If you are a professional engineer, maker or student, apply for a free STM32N6 development board, and develop your own AI application by the end of April 2025. More information about the contest is available at *www.elektormagazine.com/stm32ai* and in January/February 2025 edition of Elektor.

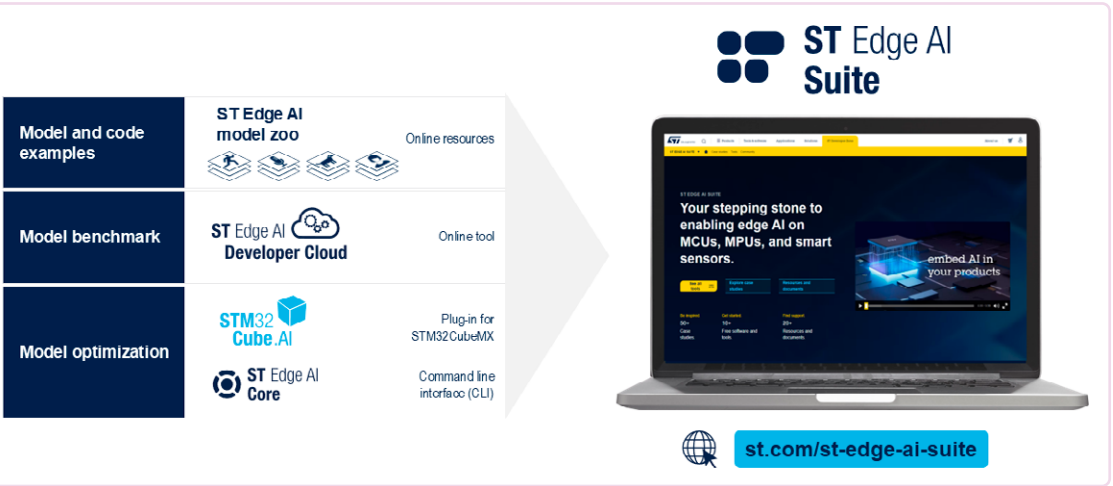
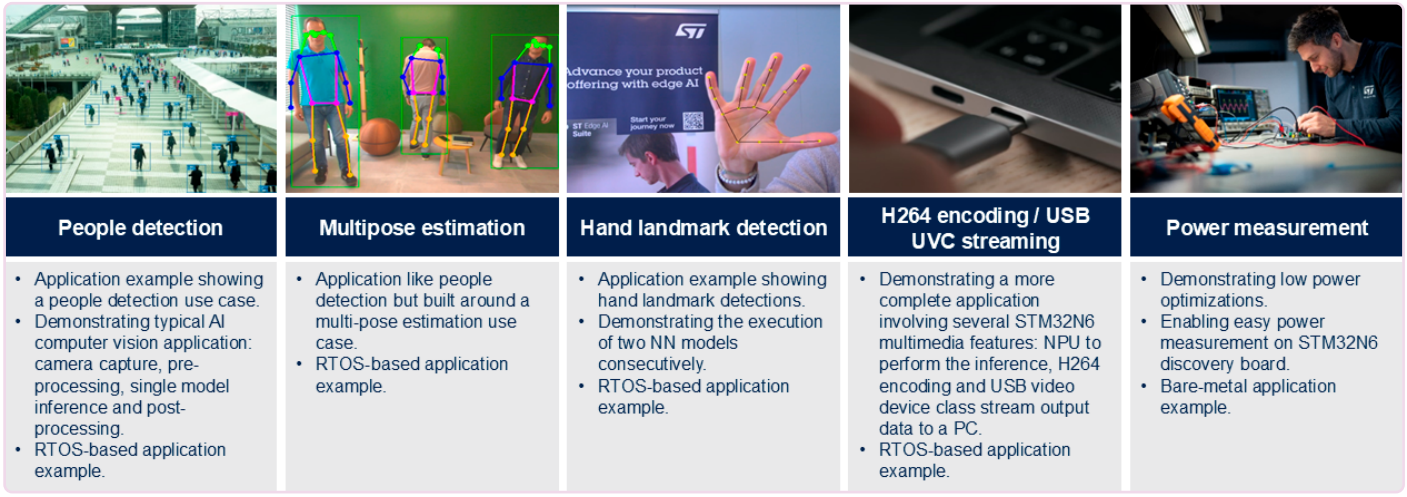


Figure 4: STM32Cube.AI and ST Edge AI Developer Cloud are designed for users who need great flexibility. The comprehensive model zoo with application code examples is a great starting point for developers who want to quickly implement AI solutions.



▲

Figure 5: STM32N6 comes with a set of code examples to kick-start your project.

AI solutions without starting from scratch. The model zoo provides pre-trained models for various applications, such as object detection, image classification, and audio scene recognition, which can be fine-tuned to meet specific requirements. The retraining scripts and application code examples make it easier for developers to adapt these models to their unique use cases, accelerating the development process and reducing time-to-market (**Figure 5**).

It's an STM32!

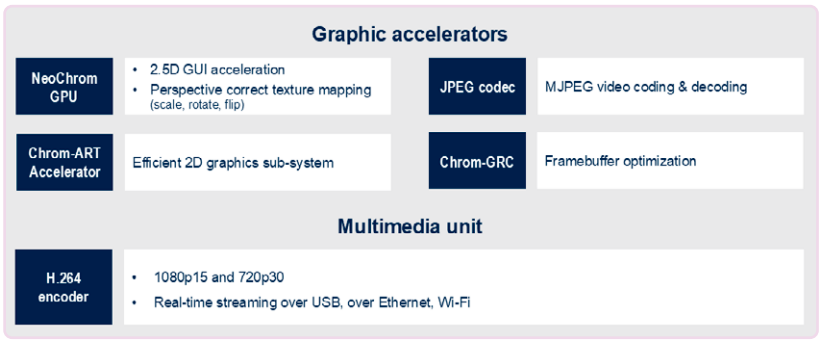
Developers will benefit from the great ecosystem and legacy of STM32 products to create their own design. The STM32N6 offers a host of advantages for developers, opening up new possibilities and simplifying the development process:

Seamless AI Integration

The STM32N6 is fully integrated into the STM32 ecosystem, making AI deployment straightforward. Developers can leverage the ST Edge AI Suite, including tools like STM32Cube.AI, to optimize and deploy their neural network models with ease. This integration reduces the complexity of AI development and accelerates time-to-market for new products.

Figure 6: The STM32N6 — which is also available without the Neural Processing Unit — is the most powerful STM32 MCU in terms of graphics capabilities.

▼



Cost and Power Efficiency

By providing MPU-like AI user experience with a smaller footprint, lower power consumption, and reduced costs, the STM32N6 allows developers to create innovative AI-enabled applications without the high expenses typically associated with MPU and GPU solutions.

Enhanced Performance

The high-performance Arm Cortex®-M55 core, combined with the embedded NPU and ample RAM, ensures that developers can run complex AI algorithms and multimedia applications efficiently. This opens up new possibilities for creating cutting-edge applications that stand out in the competitive market. The STM32N6's performance capabilities enable developers to push the boundaries of what is possible with MCUs.

Rich Graphics

The STM32N6 stands out as the most powerful STM32 MCU in terms of graphics capabilities. It features the Chrom-ART™ Accelerator for 2D graphics acceleration and the NeoChrom™ Accelerator for 2.5D acceleration, including advanced drawing, perspective correction, and texture mappings (**Figure 6**). Additionally, the Chrom-GRC™ serves as a graphic resource cutter for non-square displays. With 4.2 MB of integrated SRAM, the STM32N6 provides ample memory to handle large graphical assets and complex animations without the need for external memory, ensuring faster and more responsive user interfaces.

Comprehensive Development Environment

The STM32N6 is supported by a robust ecosystem and numerous partners, providing developers with a comprehensive development environment. Leveraging the STM32Cube software suite, it offers

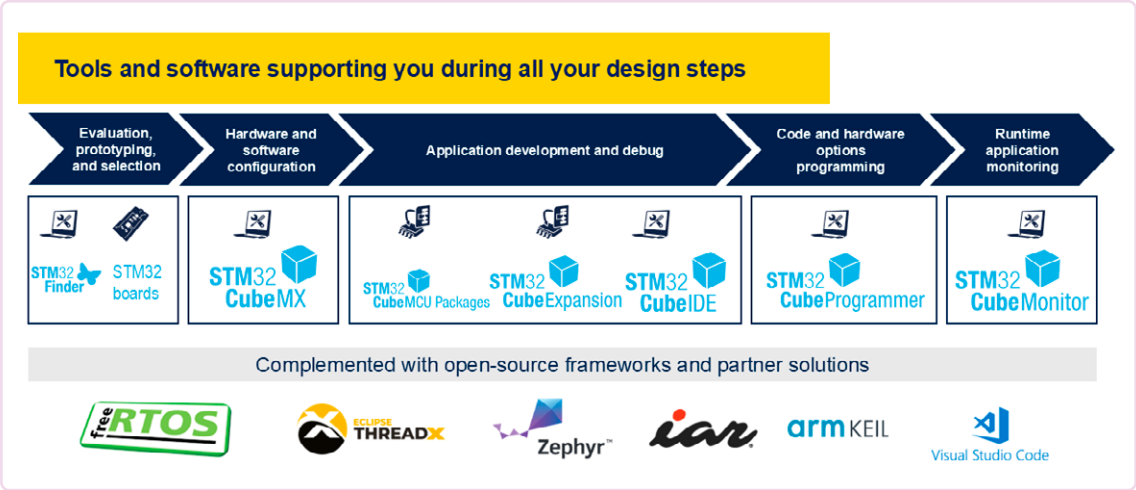


Figure 7: The STM32N6 is supported by a robust ecosystem and numerous partners.

a range of software tools for configuration, development, programming, and monitoring. For embedded software, it includes packages, middleware, and driver support, complemented by RTOS options like FreeRTOS, Zephyr, and Eclipse ThreadX (**Figure 7**).

In the AI domain, developers have access to tools, resources, case studies, and support from partners like Edge Impulse, Nota.AI, and EmbedUR. For graphics development, the STM32N6 is supported by TouchGFX, simplifying the creation of sophisticated graphical user interfaces (GUIs). This extensive support network ensures that developers have the assistance they need at every stage of the development process, enabling the creation of visually stunning and highly responsive applications.

Flexibility and Versatility

With its rich peripheral set and extensive connectivity options, the STM32N6 offers developers the flexibility to design a wide range of applications. Whether it's computer vision, audio processing, or advanced multimedia, the STM32N6 provides the necessary tools and capabilities to bring innovative ideas to life. Additionally, the STM32N6 will also be available in a version without the NPU for applications requiring high-performance computing, graphics, and multimedia but not demanding AI features. This versatility ensures that developers can choose the right configuration for their specific needs, making the STM32N6 suitable for a variety of industries, including automotive, healthcare, consumer electronics, and industrial automation.

Exciting Possibilities for Developers

The STM32N6 opens up new horizons for embedded AI applications, offering unmatched edge AI performance on an MCU. Its small footprint, lower power consumption, and reduced costs make it an ideal choice for a wide range of applications, from audio processing to computer vision. The STM32N6 empowers developers to create smarter, more efficient, and more secure devices that can operate at the edge.

Developers can leverage the STM32N6 to create intelligent home automation systems that recognize and respond to various commands and situations, enhancing convenience and security. In healthcare, this STM32 can be used to develop advanced medical devices that monitor patient health in real-time, providing accurate diagnostics and timely alerts. For industrial automation, the STM32N6 enables the design of robust systems that perform complex tasks with high precision and efficiency, improving productivity and safety. In the consumer electronics space, this microcontroller allows for the innovation of new gadgets that offer enhanced multimedia experiences, from high-definition video playback to immersive gaming.

The STM32N6 is a testament to STMicroelectronics' commitment to innovation and excellence. By redefining microcontroller performance and integrating advanced AI capabilities, the STM32N6 empowers developers to create cutting-edge applications that stand out in the competitive market. With its seamless integration into the STM32 ecosystem and support from numerous partners, the STM32N6 is poised to revolutionize the world of microcontrollers and embedded AI.

For more information, visit *www.st.com/stm32n6*.

240676-01

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

AVAILABLE IN WEEK 51

PROTEUS DESIGN SUITE

Driving forward with Manual Routing

Labcenter Electronics

www.labcenter.com

info@labcenter.com

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52

AVAILABLE IN WEEK 52


AVAILABLE
IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

AVAILABLE IN WEEK 1

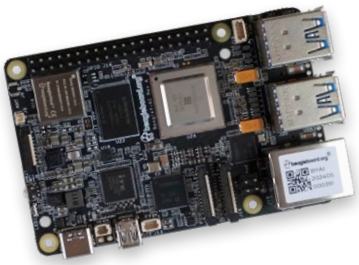
Get your hands on cutting-edge AI Hardware!

Raspberry Pi AI Kit

The Raspberry Pi AI Kit includes the Raspberry Pi M.2 HAT+ and a Hailo AI acceleration module for use with the Raspberry Pi 5. It provides an accessible, cost-effective, and power-efficient way to integrate high-performance AI. Explore applications including process control, security, home automation, and robotics!
www.elektor.com/20879

Unitree Go2 Pro Quadruped Robot

The Unitree Go2 is a quadruped robot designed for the research & development of autonomous systems in the fields of human-robot interaction (HRI), SLAM & transportation. Due to the four legs, as well as the 12DOF, this robot can handle a variety of different terrains.
www.elektor.com/20357



BeagleY-AI SBC with GPU, DSP and AI Accelerators

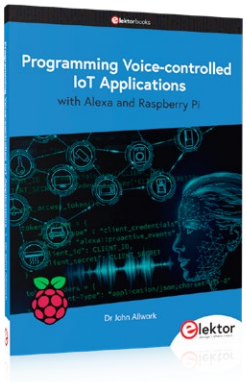
BeagleY-AI is a low-cost, open-source, and powerful 64-bit quad-core single-board computer, equipped with a GPU, DSP, and vision/deep learning accelerators, designed for developers and makers. Users can take advantage of BeagleBoard.org's provided Debian Linux software images, which include a built-in development environment.
www.elektor.com/20991

Raspberry Pi AI Camera

The Raspberry Pi AI Camera is a compact camera module based on the Sony IMX500 Intelligent Vision Sensor. The IMX500 combines a 12 MP CMOS image sensor with on-board inferencing acceleration for various common neural network models, allowing users to develop sophisticated vision-based AI applications without requiring a separate accelerator.
www.elektor.com/20953

Waveshare Jetson Orin Nano AI Development Kit

This AI Edge Computing Development Kit is based on the Jetson Orin Nano Module providing rich peripheral interfaces such as M.2, DP, USB, etc.
The kit also comes with a pre-installed AW-CB375NF wireless network card that supports Bluetooth 5.0 and dual-band WIFI, with two additional PCB antennas, for providing high-speed and reliable wireless network connection and Bluetooth communication.
www.elektor.com/20762



Programming Voice-controlled IoT Applications with Alexa and Raspberry Pi

This book is divided into two parts: creating Alexa skills and designing Internet of Things (IoT) devices using a Raspberry Pi. It covers topics like Alexa skill development, in-skill purchasing, and using AWS Lambda, as well as building smart home devices controlled by Alexa. Readers also learn about MQTT messaging, creating proactive notifications, and turning a Raspberry Pi into a stand-alone Alexa device.
www.elektor.com/20400

Join the Elektor Community

Take out a elektor GOLD membership!

- The Elektor web archive from 1974!
- 8x Elektor Magazine (print)
- 8x Elektor Magazine (digital)
- 10% discount in our web shop and exclusive offers
- Access to more than 5,000 Gerber files

Also available

The Digital elektor GREEN membership!

- The Elektor web archive from 1974!
- 8x Elektor Magazine (digital)
- 10% discount in our web shop and exclusive offers
- Access to more than 5,000 Gerber files

www.elektormagazine.com/member

elektor design > share > earn