Watt-IZ – A Fast Track to Speech Enabled Embedded Applications

Built on the ESP32-S3 with purpose-designed audio, UI, cloud speech integration, storage, real time clock, and power management.

Introduction

Watt-IZ is a programmable ESP32-S3 based platform designed specifically for speech-enabled embedded applications, combining purpose-built hardware with a comprehensive, ready-to-use software stack. In addition to audio input and output, touchscreen display, SD card storage, wireless connectivity, real-time clock, and power management, Watt-IZ ships with a rich collection of preloaded speech and audio demos, full source code, documentation, and reference designs. This allows users to explore, run, and modify real

speech-interactive applications immediately—without writing code—while still providing a solid foundation for custom development.

The Watt-IZ exists for people who want to explore voice interfaces, conversational systems, language translators, chatbot's, voice assistants, command interpreters, and many other AI-style applications using real hardware without needing a full development system or complex setup just to run and experience the speech-enabled features of the device. Demos and applications can operate standalone from the SD card, while users who want to build or modify firmware can use a normal PC-based development workflow.

Some demo's and app's require API keys from google and openAI to access cloud services such as speech-to-text, text-to-speech, translate, and chat GPT access. See the section *How to Acquire API Keys.*

Table of Contents

Hardware Features

CPU

- Espressif ESP32-S3 SOC, 32-bit LX7 dual core @ clock speed up to 240 MHz.
- 16MB flash (program) memory, 512KB SRAM, 8MB of PSRAM (pseudo static RAM).

- WiFi connectivity supports 2.4GHz, IEEE 802.11/b/g/n. Built-in WiFi PCB antenna.
- WiFi can operate in Station mode (STA), Access Point mode (AP), or both.
- Bluetooth Low Energy (BLE 5.0) supports high speed, long distance, and low power operation.
- ESP-NOW proprietary connection-less low power wireless protocol for direct device-to-device communication using MAC addressing. Used for streaming audio and local control functions.
- Peripherals: PWM, I2C, UART, SPI, SDIO, IrDA, GPIO, LCD, Camera interface, UART, I2S, USB 1.1 OTG, USB Serial/JTAG controller, MCPWM, SDIO host interface, GDMA, TWAI® controller (compatible with ISO 11898-1), 12-bit ADC, touch sensor, temperature sensor, timers, and watchdog.
- DSP optimized routines for FFT / IFFT, FIR / IIR filters, convolution, matrix math, and windowing functions.

Audio Input/Output

- Embedded I2S Mems microphone with 16 bit, 16KHz sampling rate.
- High efficiency class D I2S audio amplifier and speaker driver. Optimized for 4 ohm 5W miniature speakers.

Display and User Interface

- 2.8” IPS LCD with modern, high sensitivity capacitive touch screen. SPI serial interface up to 40MHz. 320X240 pixels with 16 bit color (65K colors).
- Wake button configured to act as a wake-from-deep-sleep trigger or as a general purpose function button.
- Intelligent programmable full spectrum LED for status and error notification. 24 bit color depth and 255 levels of brightness.
- Utilizes the open source LVGL graphics library for event driven UI development. Driver supports DMA transfers to enhance performance.

Storage and Data

- SD-MMC (4-bit mode) SDXC/HC microSD card 32GB.
- Employs the FAT32 file system with full compatibility to the LVGL library file functions.
- Apps and demo programs can be loaded directly from the SD card – no need to download via serial port.

Power Management

- The device is externally powered from a typical 5V AC adapter (phone charger) with a USB-C connector. Alternatively, power can be supplied from the micro-USB connector used for development and debug.
- Full support for battery operation from a single cell lithium-ion battery, 1000 – 3000 MAh capacity. Charging current 900 Ma.
- The device can operate at full power for a minimum of 4 hours using a 2000 Mah battery.
- Battery battery voltage and charging current are monitored by the CPU's Analog-to-Digital Converter.

Real Time Clock

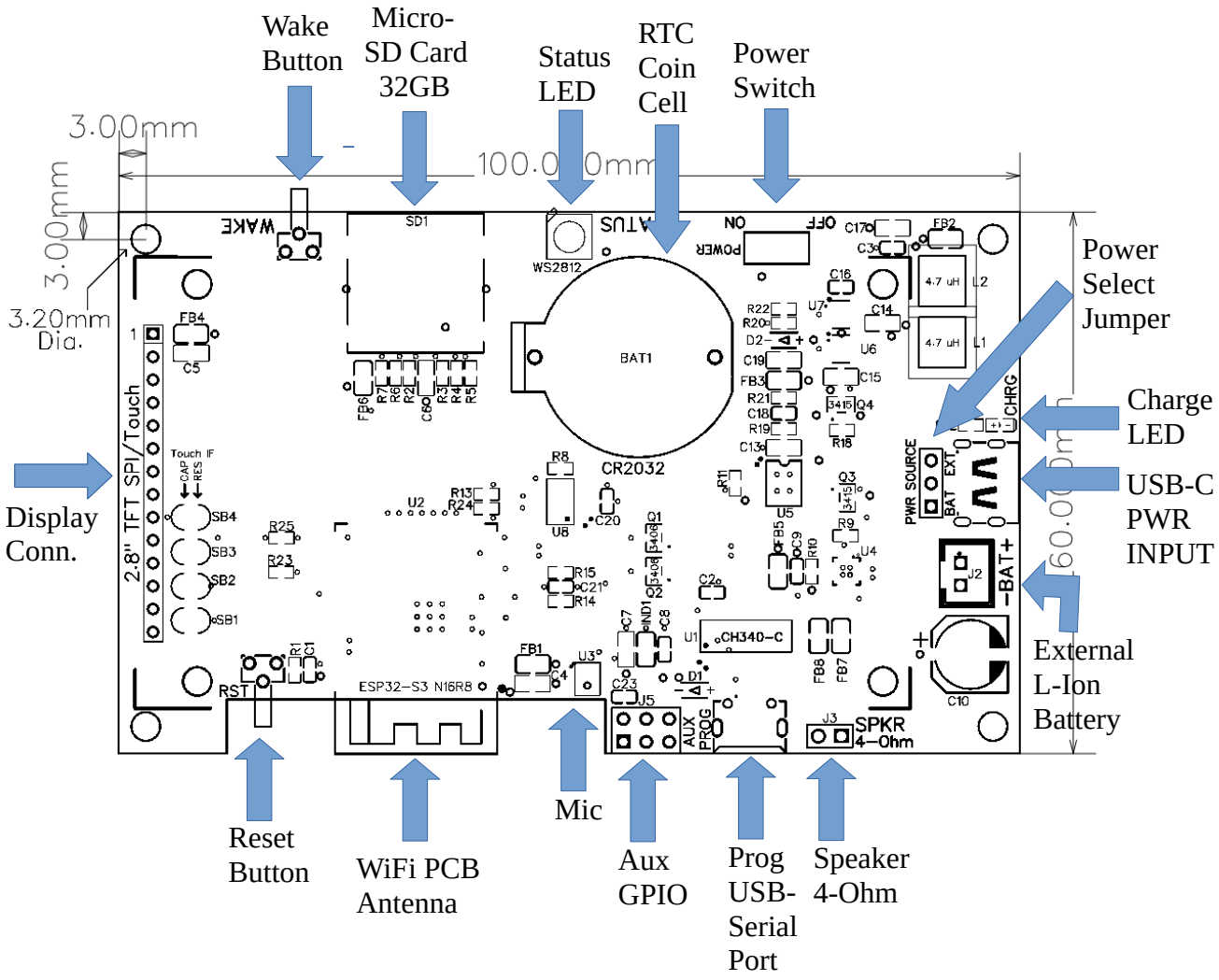
- High accuracy Real Time Clock (RTC) chip with coin cell backup battery for years of continuous timekeeping. Allows accurate time preservation across power failure without the need for manual setting or internet time synchronization after power up.
- Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100.

Firmware Development

USB-Serial port via a micro-USB connector. All hardware necessary to interface with an Integrated Development Environment (IDE) such as Visual Studio Code. Auto download mode is supported – no buttons to push to initiate firmware download.

PCB Physical

The Watt-IZ PCB V1.3 physical dimensions and part locations:



Software Capabilities

Overview

Watt-IZ software support consists of a collection of demonstration programs, user documentation, and a core “Watt-IZ” application that serves to encapsulate many of the platform’s capabilities. Each demo program has an associated README file describing the purpose and operation of the demo.

All demo’s and the core Watt-IZ application can be loaded directly from the SD card without requiring a host development system, download cable, etc.

All programs require functional Watt-IZ hardware, especially the graphics display and a functioning SD card. Some require a connected speaker while others may require an API key for speech conversion, language translation, or chatbot services (see How to Acquire API Keys).

What Can It Do?

GUI Functions (see demo *watt_iz_graphics*)

- GUI basic widgets and touch screen.
- Advanced drag & drop example.
- Create and display QR codes.
- PWM backlight LED control.

Storage and File System (see demo *watt_iz_files*).

- Format an SD card with FAT32.
- Utility to create a system credential JSON file using a text file with your WiFi info and API keys.
- Display file hierarchy.
- Measure SD card write and read speeds in the hardware environment.
- Load demo programs and apps directly from the SD card. All demo’s have code example of how to implement this feature.
- Use case demo’s of Non Volatile Storage using the ArduinoNVS library.

Speech and Audio Functions

- Record and playback WAV (uncompressed) files. See demo ***watt_iz_audio***.
- Audio processing (low pass filtering, FFT).
- Streaming (examples with ESP-NOW to stream audio to a peer).

- Speech detection (use FFT and Formant analysis to detect valid speech vs other noises).
- Sinewave frequency and ring tone functions.

Real Time Clock & Alarm (see demo *watt_iz_clock*)

- Demonstrates timekeeping functions (year, month, day, hour, minute, second).
- Time synchronization with internet NTP server.
- Calender with reverse and forward month scrolling.
- Simple alarm trigger.

AI Cloud Functions

- Google Speech to Text service (see demo ***watt_iz_STT***).
- Google Text to Speech service (see demo ***watt_iz_TTS***).
- Google language translate service (see demo ***watt_iz_translator***).
- OpenAI Chat GPT service (see demo ***watt_iz_chatgpt***).

Intent Classification / Command Parsing

- Break speech into **intent** – **action** – **target** categories.

How to Acquire API Keys

Some demos require cloud based services from Google and/or OpenAI. To use services like Speech To Text, an API key is needed to access those services.

Google API Key

A google API key is needed to obtain access to STT, TTS, and Language Translator services.

- Go to **https://console.cloud.google.com** and sign in (or create a new) Google account.
- Click the project selector (top bar) → **New Project** → give it a name → **Create**.
- Make sure your new project is selected.
- In the left menu, go to **APIs & Services** → **Library**.
- Search for the service you need (e.g., Speech-to-Text, Translate, Maps, etc.).
- Click the API → **Enable**.

- ***Important:*** For the Watt-IZ environment, make sure you enable the following services:

  > Speech To Text

  > Text To Speech

  > Translator
- Go to **APIs & Services → Credentials**.
- Click **Create Credentials → API Key**.
- Your API key is generated—copy and save it securely.
- (Recommended) Click **Restrict Key**:
- Limit which APIs it can access.
- Add application restrictions (IP, HTTP referrer, or app type).
- Save changes.
- If required by the API:
- Go to **Billing** and attach a billing account to your project.
- Copy the API key to the Watt-IZ configuration file (see Watt-IZ User Manual for details).

OpenAI (Chat GPT) API Key

An openAI API key is required to access Chat GPT services. Use the following steps to acquire a new key:

- Go to **https://platform.openai.com** and sign in (or create an account).
- Click your profile icon (top-right) → **View API keys**
  (or go directly to the API Keys page from the dashboard).
- Click **Create new secret key**.
- Give the key a name (optional but recommended).
- Copy the key immediately and store it securely - **You will not be able to view it again!**
- (Recommended) Set usage controls:
- Go to **Settings → Billing** to add a payment method if required.
- Review **Usage Limits** to avoid unexpected charges.
- Use the key in your application:
- Copy the API key string to the Watt-IZ configuration file on the SD card (see Watt-IZ User Manual for details).

- ***Never hard-code your API strings into public source code.***

More Information

For more detailed information, please visit https://github.com/johnny49r/watt-iz.

To purchase: Visit: Tindie Store