# KNOBLOMAT

# Technical Manual

**DTV-ONLINE**

# Hardware



The *Knoblomat* is designed around an ESP32 microcomputer using simple USB 5V power bank.

## ESP32

The ESP-WROOM-32 module has 4 MB of flash memory, which is used to store the bootloader, the user application and nonvolatile (flash) memory for storage. An onboard CP2102 – Silicon Labs single-chip USB-to-UART bridge provides serial communication over USB.

## Summary

| **Board** | **ESP32 Devkit** |
|---|---|
| **Microcontroller** | Extensa® Dual-core LX6 600 DMIPS |
| **Architecture** | 32 Bit |
| **Operating Voltage** | 3.3V |
| **Input Voltage** | 7-12V |
| **Digital I/O Pins (DIO)** | 25 |
| **Flash Memory** | 4 MB |
| **SRAM** | 520 KB |
| **Clock Speed** | 240 MHz |
| **Wi-Fi** | IEEE 802.11 b/g/n/e/i |
| **PWM Output** | 16 |
| **Power Consumption** | 80 mA |
| **PCB Size (L x W x H)** | 5 x 3.7 x 1.2cm |
| **Weight** | 13 g |

The Wi-Fi has an integrated TR switch, balun, LNA, power amplifier and matching network and supports WEP or WPA/WPA2 authentication, or open networks.

# Serial Port

The USB drivers provides a virtual com port to download software to the microcontroller:

(https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers).

When a computer is connected via a Micro USB cable, the *Knoblomat* writes various messages during the execution of the program (startup, WiFi, HTTP etc.). Any terminal program can be used with the settings shown here.

| Parameter | Value |
|---|---|
| Baud Rate | 115200 |
| Parity | None |
| Data Bits | 8 |
| Stop Bits | 1 |

The *Knoblomat* provides access to the two buttons (EN or reset, Boot or load) and to the LEDs (red power, blue application). A flashing blue led (1 second on, 1 second off) is shown in normal operation. Slower or faster blinking indicates an error. When the *Knoblomat* goes into sleep mode (after 10 minutes of no interaction) only the red led in on.


# Setup

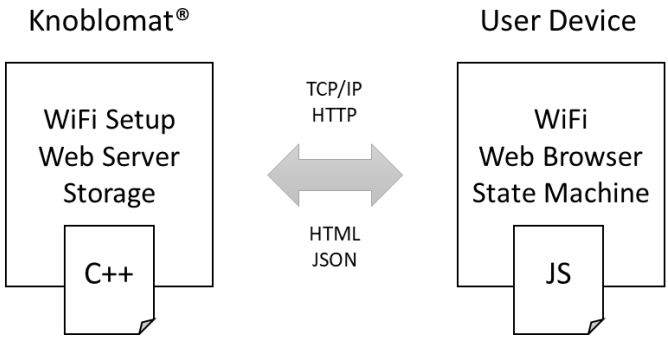
The *Knoblomat* WiFi network allows the use of any web browser enabled device to communicate.

The web interface (HTML-5) adapts to the different screen sizes and supports mouse or touch as input. To power the Knoblomat a 5V capable USB charger (not supplied), or a computer USB port with charging capabilities is required.

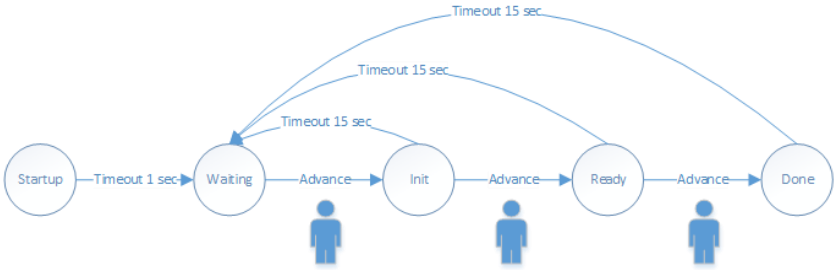
# Software

## Architecture

The software architecture is based on a classical client/server setup using a HTTP web server at the backend, and a HTML-5 user interface running in any standard web browser.

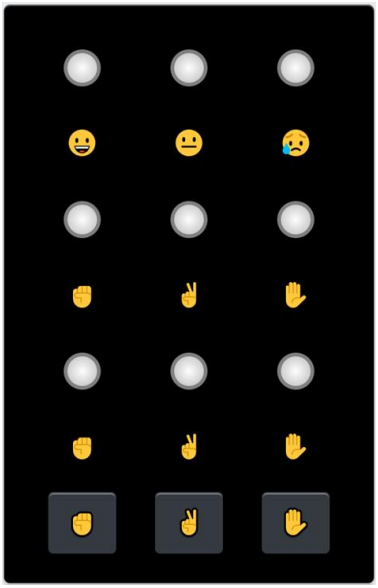



The backend implements the WiFi setup and configuration and the asynchronous web server providing HTML pages and the REST based JSON data access. The client frontend runs the user interface (*Knoblomat* display, user help, and configuration). The JavaScript application implements the *Knoblomat* state machine for the operation.



## HTML 5 User Interface

Various (simulated) LEDs and buttons are used to operate the *Knoblomat*.



3rd LED row
(results).

2nd LED row
(machine selection)

1st LED row
(user selection)

Push Button row.

In order to simplify the programming several special JS libraries are used:

- Bootstrap (getbootstrap.com)
- jQuery (jquery.com)
- Popper (popper.js.org)
- State-Machine (github.com/jakesgordon/javascript-state-machine)

The software for the *Knoblomat* is using those additional libraries for the HTML-5 user interface and implementing a finite state machine. All interactions with the backend are performed using *jQuery* `ajax()` method to perform an AJAX (asynchronous HTTP) request calls (GET, POST) to exchange JSON data without refreshing the browser UI.

## Backend - HTTP Web Server

The web server is implemented using the ESP Async Webserver library. It provides access to the various web pages and resources (CSS files, JS scripts, images etc.). Additionally, a REST interface to configuration and settings data is used to exchange data to the browser client application.

| HTML Pages: | REST (GET): | REST (POST): |
|---|---|---|
| http://knoblomat/ | http://knoblomat/settings | http://knoblomat/ap |
| http://knoblomat/index.html | http://knoblomat/ap | http://knoblomat/wifi |
| http://knoblomat/help.html | http://knoblomat/wifi | http://knoblomat/game |
| http://knoblomat/about.html | http://knoblomat/game | http://knoblomat/smart |
| http://knoblomat/config.html | http://knoblomat/server | http://knoblomat/clear |
| | http://knoblomat/system | http://knoblomat/reset |
| | | http://knoblomat/reboot |

## Backend – WiFi Operation

The standard WiFi configuration is an open access point (no password) with a standard fixed SSID to allow client connection to the web server at a fixed IP address:

- SSID: KNOBLOMAT_XXXXXXXXXXXX
- IP Address: 192.168.4.1

Note that the SSID always starts with KNOBLOMAT_ and ends with a unique chip ID (the MAC address).

Additionally, a connection to an existing WiFi network can be established, when the SSID and password are provided to allow the connection.

## Smart Config

*Smart Config* communicates information (the network SSID and key phrase) from a secure WiFi network to a *Smart Config* enabled device. This is typically performed by using a Smartphone and a special application - **EspTouch**. The *Knoblomat* is a *Smart Config* enabled device to facilitate the network setup.

Check for apps on the **Apple Store** or on the **Google App Store**. On the Google App Store, the *ESP8266 SmartConfig* App can also be used.

# Configuration

Using the web browser, the main (home) page displays the *Knoblomat*. Besides the **About** page and the **Help** page, the configuration page allows the *Knoblomat* to join an existing WiFi network.

## Web Server

The current HTTP web server settings are displayed. Note that if the Knoblomat is also connected to a WiFi network, two IP addresses are displayed (AP and WiFi).

```
Web Server:
    Address (AP):   192.168.4.1
    Address (WiFi): 10.0.1.109
    Hostname:       knoblomat
    Port:           80
    URL:            http://knoblomat
```

## WiFi Access Point

The WiFi access point provided by the *Knoblomat* has a fixed SSID name. A password can be provided by the user in the *Access Point Setup* dialog.

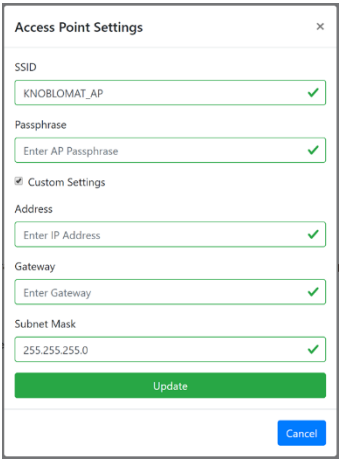
```
WiFi Access Point:
    SSID:      KNOBLOMAT_XXXXXXXXXXXX
    PASS:
    NetworkID: 192.168.4.0
    Hostname:  knoblomat
    Address:   192.168.4.1
    Clients:   0
    MAC:       3C:71:BF:4F:B7:99
```

The default at startup is an open network with no password. Connect the client to the “KNOBLOMAT_” access point and open the web page at the default IP address. This address is also provided by a QR-Code. Simply scan the code with a smartphone and the *Knoblomat* webpage will open.

Scan Me

## User Dialog Access Point Settings

The access point settings can be changed using the corresponding user dialog. When the custom option is enabled the IP address, the gateway (should be the same IP address) and the subnet mask can be provided). This should be done only by the experienced user knowing what the implications are.




Since no critical user data (except the disappointing results of the game) are exchanged with the *Knoblomat*, the open access point is a valid option for most users.

## WiFi Network

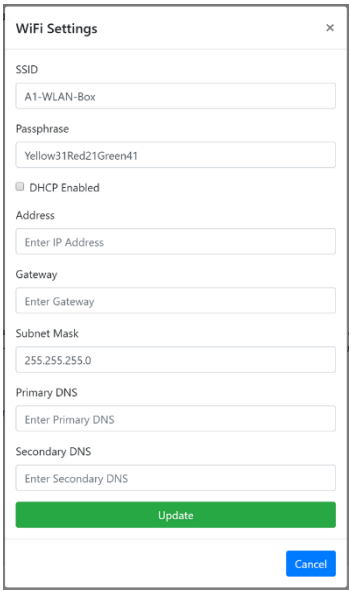
The WiFi network configuration shows the current settings. If the existing WiFi network provides DHCP, the *Smart Config* option allows a simple setup using the smartphone app.

```
WiFi Network:
    SSID:       YOUR_SSID
    PASS:       YOUR_PASSWORD
    NetworkID:  10.0.1.0
    Hostname:   knoblomat
    Address:    10.0.1.109
    Gateway:    10.0.1.138
    Subnet:     255.255.255.0
    DNS:        10.0.1.138
    BSSID:      E0:28:6D:79:9F:97
    MAC:        3C:71:BF:4F:B7:98
```

Note that the IP address shown is the second address for the web server and allows the access to the *Knoblomat* within your home network.

## User WiFi Network Settings

The network settings option allows the experienced user to change a set of various parameters (IP address, gateway, subnet mask, DNS etc.).




For the typical home network, the WiFi access point provides DHCP services for the network clients, so normally only the SSID and the password for the WiFi network must be provided.

As mentioned before, the *Smart Config* setup using a smartphone is also a supported option. In both scenarios the necessary parameters are:

- SSID
- Password

Using the *Smart Config* setup, it is also assumed that the WiFi access point has DHCP enabled.

Note that only WiFi networks using IEEE 802.11 standard security feature WPA/WPA2 are supported (WEP is not currently supported).

Note that the *Knoblomat* has a unique MAC address which might be needed to allow the connection to the WLAN router in Your home network (the last 12 characters from the access point name).

## Nonvolatile Storage

All settings are stored in nonvolatile storage and are again available at startup time. All settings can be cleared with the *Reset WiFi Config* option. When all settings are cleared, after restart the *Knoblomat* defaults to the single access point “KNOBLOMAT_XXXXXXXXXXXX”.

# Development

The code can be easily customized using freely available development tools, such as the Arduino IDE, Visual Studio Code, or other editors.

## IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux). It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

Other options are a Visual Studio Code plugin or the *Visual Micro* Arduino IDE for Visual Studio plugin.

The *espressif SDK* is also available if even more control is necessary or if new firmware is to be developed. Currently only 64-bit versions of Windows are supported. ESP-IDF requires some prerequisite tools to be installed so you can build firmware for the ESP32. The prerequisite tools include Python, Git, cross-compilers, menuconfig tool, CMake and Ninja build tools.

## Editor

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. The Visual Studio 2019 Community Edition is also freely available from the Microsoft Visual Studio home page.

## GitHub

GitHub is a company that provides hosting for software development. GitHub includes version control using Git, which allows software projects to keep track of all versions and revert to previous versions if necessary.

## Libraries

All libraries used are available on GitHub and can be downloaded from their respective repositories or the integrated Arduino library manager:

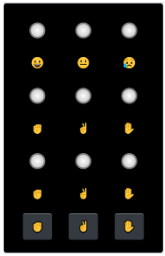
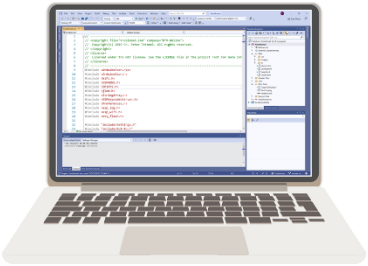
- Utilities (ArduinoJson, jled, neotimer, SPIFFS, Preferences)
- Networking (NetBIOS, ESPmDNS, ESPasyncWebServer, WiFi)

## Source Code

The complete source code (HTML, C++, Arduino etc.) is available at:

https://github.com/dtv42/Knoblomat

Just download the project from GitHub, install a development environment, change the code, upload the new code using the USB cable to the *Knoblomat* and enjoy (if You have any questions, don't hesitate to contact us).

*Happy Coding*

## Contact Us

Grabenweg 4
3683 Yspertal, Austria

Phone: +43 699 1968 1006
Email: peter.trimmel@live.com

Web: dtv-online.net

DTV-Online
Grabenweg 4
3683 Yspertal, Austria