# Symobi

## Universal IoT & Embedded OS

Real Microkernel OS

Modular Structure

Easy Portablility

Adaptable, Small Footprint

Drivers, Stacks & Applications

### Stages
- Microkernel only (Multi-Thread)
- Microkernel only (Multi-Task)
- Microkernel & process manager
- System, drivers & services
- Fully equipped Symobi OS
- Customized stages

### Highlights
- Microkernel with ~100 kB footprint
- Available with drivers, stacks & applications
- User Interface: GUI, CLI and headless
- MultiThreading, MultiTasking & MultiCore
- Easily portable to new CPU platforms
- Available with shell, launcher and applets

# Symobi

## The embedded OS Symobi stands out as an IoT platform with its unique architecture and structure

Lightweight Microkernel architecture, combined with a fully modular structure, makes Symobi the universal and versatile OS for IoT and other embedded applications.

### Microkernel Architecture

Symobi is based on Sphere, a real Microkernel with a footprint of only 100kB, which already includes the Nanokernel of about 30 kB for platform abstraction.

### Modular Structure

Symobi's entire structure is modular. This starts with the Microkernel, which contains the Nanokernel component. Every structural building block is a module of its own.

### Symobi is not Linux

Symobi is an entirely unique operating system, with its own architecture. It is in no way related to Linux, neither by birth nor by marriage.

### Portability

Due to Symobi's Microkernel architecture and highly modular structure, it is easily portable.

### General Purpose

This makes Symobi a versatile embedded RTOS, which is extremely and easily adaptable and configurable for universal use on a wide range of different systems.

### Separated Levels

The efficient separation of functional levels was an essential condition in designing Symobi. Each level covers the required mechanisms – but not more. This results in the strict separation of functionality into Nanokernel, Microkernel, process manager and drivers/services/applications. This strict separation into modules allows all components to be kept small and simple.

### User Interfaces

Symobi offers a graphical user interface (GUI) or a command line interface (CLI). Both options cover a number of existing applications and tools. Using the GUI, many features are comfortably available as applets. Symobi can also be used completely "headless," if required.

### Drivers, Services, Stacks, Applications

Symobi is a complete embedded operating system, which includes device drivers, protocol stacks, system-wide services and a variety of applications. All of these can be integrated freely in order to combine and produce maximum performance with a minimal footprint.

### Apps, Applets, Auxiliaries

Symobi applications that use the GUI are grouped into apps, applets and auxiliaries. Apps are visually bound to the Symobi Launcher and integrate with it. Applets use a very minimalistic interface, allowing them to not depend on graphics, while the Symobi's applet framework takes care of displaying them appealingly. Auxiliaries are the most versatile applications, yet special care must be taken to display them correctly. They use the normal graphical and component frameworks of Symobi, but do not integrate with Symobi's launcher.

### Reliable IPC model

In Symobi, message-based Inter Process Communication (IPC) is used for the exchange of data between separate tasks (processes). Therefore, every driver, service and application has full control over the information to be shared or to be made available to other processes.

### Host Platforms

In order to facilitate cross development on different hosts, Symobi supports user mode virtualization on Linux and Windows hosts. Various programs, with the exception of device drivers, can therefore be developed and tested directly on the host.