LS/7 Kernel Documentation

Gabriel Weingardt

November 10, 2023

1 General

1.1 Introduction

The LS/7 Kernel is the building ground of the LS/7 Computer and it's applications. It holds simple and complex modules, made to make programming the LS/7 Computer easier.
The Kernel is sub-divided into multiple files, housing many specific functions. Since, the target CPU is the 65c02, memory is a important key in programming. The Kernel is designed in a way, that it doesn't interfear with memory, that isn't designated specificaly to the Kernel.
The 65c02's Stack is also a vital limitation to programming the system. That's why the Kernel uses a seperate Kernel stack, to offload variables onto, so it can be assured, that up to 127 sub-routines can be nested into each other.
The Kernel has tight hardware implementation, to make targeting this proprietary platform easier.

1.2 Legal

1.3 Installation

Dependencies

- vasm6502
- minipro (optional)

Building
Use the Makefile to build the binary. It compiles it and automaticly uploads it to a EEPROM programmer, to the "39SF010" Flash IC, found in the LS/7 Computer.

We use vasm with the 6502 target and the oldstyle syntax, with the "-Fbin", "-dotdir" and "-wdc02" options.
main.s houses initialisation components and also acts as the global binder for the Kernel

$vasm6502\_oldstyle - Fbin - dotdir - wdc02\ main.s$

2 The Kernel

This chapter describes the individual parts, the Kernel is made of. It documents the usecases and gives a small discription. The Kernel is made out of smaler sub-sections as described in 1.1. It houses both simple and complex functions, to make programming the LS/7 Computer easier and more effecient.

2.1 Hardware.s

Houses platform specific functions, such as a video interface, status LED control, scanning the keyboard and beeping the speaker.

2.2 Event_Handler.s

Executes, Adds and Removes Event Handlers from a List. These Event Handlers are executed as sub-routines

2.3 Memory_Manager.s

Handles memory allocation. Here are two types of [] used. A software stack (to not interfeer with the hardware stack of the 65c02), and a form of external fragmantation.
The memory manager also cleans up "free" space, when a block is given free, so it's highly reccomented to use the Index page to address this memory (we explain this section in more detail in **??**). It's also possible to allocate a static chunk, that cannot be moved by the manager.

2.4 Console.s

This module handles the text mode. It provides functions specifically, to work in cunjunction with the text mode. Everything from printing text, getting the keyboard input, loading fonts and much more.

2.5 Graphics.s

This module handles the graphics mode. Everything from drawing content on the screen. Drawing rectangles, polygons or just setting pixels. It's not reccomented to use this module in the Text mode.

2.6 Video.s

Houses functions for the video card. These work both in text and graphics mode. For copying memory, setting the mode and accent, etc. .

2.7 Irq.s

Handles the IRQ request, starts a keyboard scan and starts the IRQ event-handler. It also keeps track of software-timers and invokes them.

2.8 Keyboard.s

Scans the keyboard and converts it to ASCII. It also handles key-press delay and repeat-rate. It returns the pressed key (Formatted and not), pressed modifier keys (such as CTRL, Shift etc.) and the pressed arrow keys.