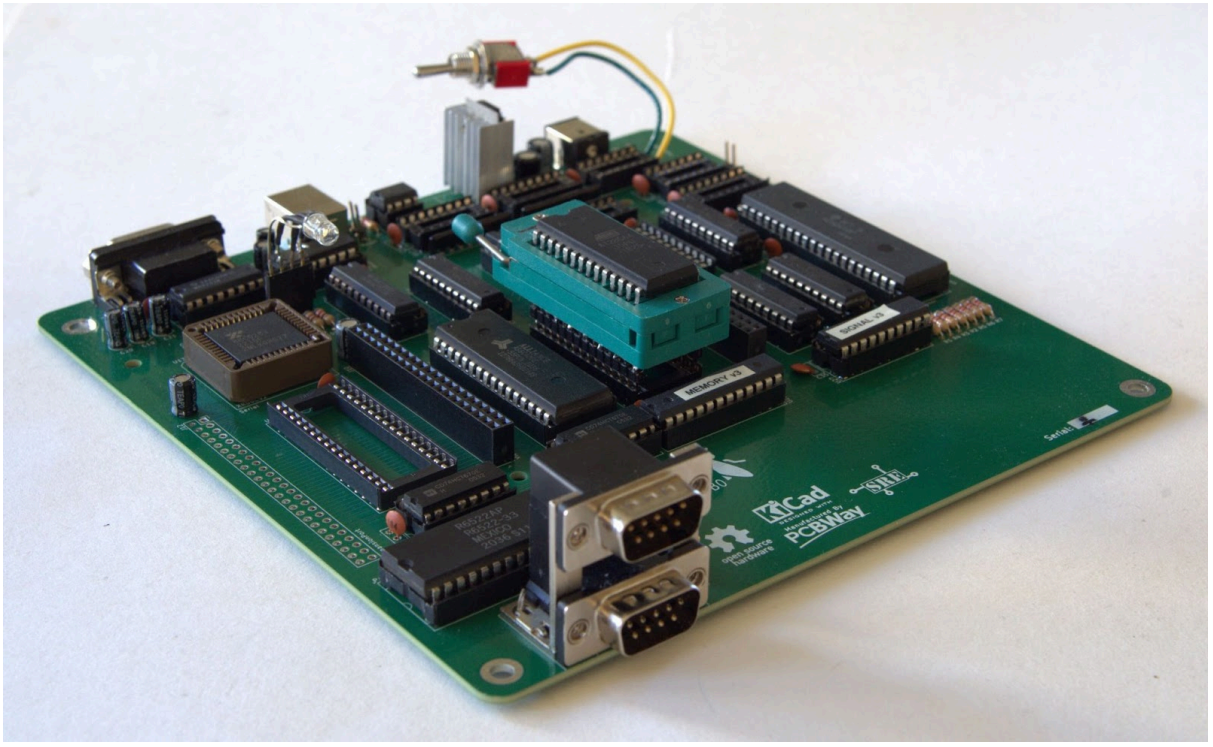
µLind

8-bit Retro Computer



Introduction

This manual is written by me, Eric Lind, as a means to remember how everything in this system is intended to work and also as a means to share that knowledge with you, the reader of this excellent manual. I am not a native English speaker so wordings might be weird sometimes but I hope that the intent shall be clear enough. There might be some factual errors as well but if you find any, please report it back to me so I can fix it. So without further ado, let's get into the business!

Brief project overview

This project started in 2023, after my then 14 year old son found an interesting, and for both of us unknown, microprocessor in a bag of old chips: The 6809! I had since long been curious on how to build a microcomputer and was immediately intrigued by the design of this exotic little chip. Being quite familiar with the 6502/6510 (both me and my son are avid Commodore users. GO 64!) we found the "advanced" features of the 6809 interesting. We started sketching on different ideas of what we would like a computer to be able to do. And my son who is an amazing datasheet reader found several features that made this processor interesting to build a system on.

After a couple of months of discussion and throwing ideas around we settled on a list of features we wanted the system to have:

- Based on the 6x09 processor (later on we decided on the 6309)
- Have at least 1Mb of available RAM (This would require some kind of banking)
- EEPROM for storing BIOS and drivers
- USB based console port (to be able to debug)
- Serial port for peripheral communication
- Joystick ports (C64 compatible, for entertainment)
- PS2 Keyboard interface (and hopefully mouse interface as well)
- External storage for OS and applications (most likely CF, due to 8-bit compatibility)
- Audio support (we want that wonderful SID sound)
- Video capabilities (The 80 col chip in the C128 would fit good)
- And further expansion possibilities…
- All this in a Mini-ITX package, just for lols!

And to be able to manage all this on one board I (I did all of the board layout myself) decided to do it in stages, get everything basic working, add some features, get that board working and add the rest. This way it is easier to isolate what is wrong. We also saw that there was not going to be room for either video controller or audio circuits on the board, so they were moved to addon-ports, to be able to layer the system better and test several different video and audio solutions.

So we planned following stages:

- Stage 1: Basic functionality, processor, some unbanked RAM, EEPROM, basic logic chip setup and a serial controller.
- Stage 2: Stage 1 + banking system, more RAM, power and reset circuit, addon-ports (to be able to start developing video and audio addons), expansion port (with more “all” signals) and a parallel port for joysticks.
- Stage 3: Stage 2 + PS2 interfaces and CF addon port (this was not possible to fit on the board either).

This manual is written for Stage 3, the “final” stage, and will be updated upon correction and changes done to the system during the development (at the time of writing this, the Stage 3 has not been ordered yet, but all parts of stage 2 have been tested).

Purpose and intended audience of the manual

This manual will be useful for anyone that needs to either fault trace the µLind, change settings or write applications for the system. Since this manual also contains a listing of all BIOS routines and memory mapping.

Specifications

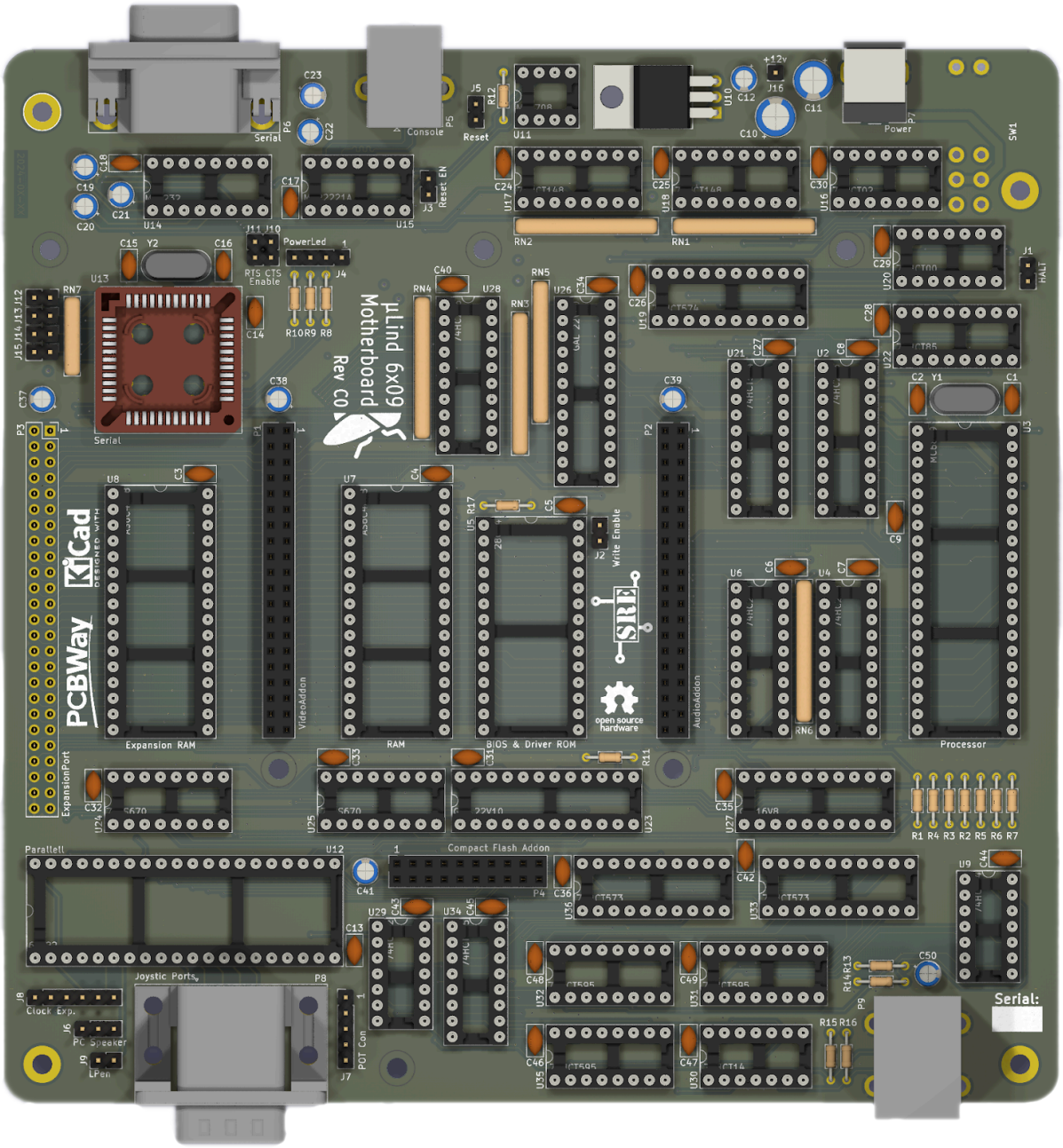
Processor details (HD6x09, clock speed, etc.)

Memory specifications (RAM/ROM configuration)

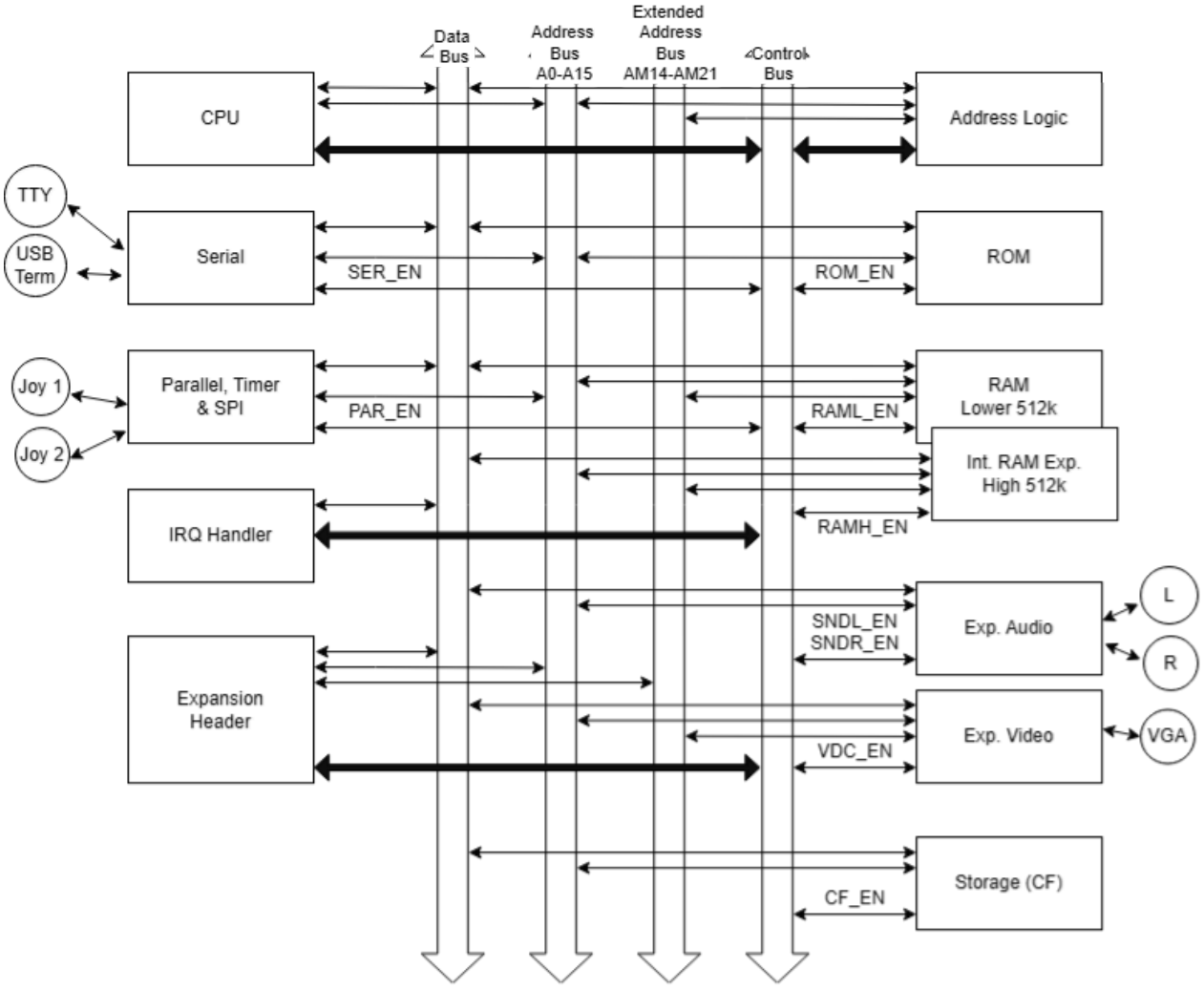
I/O capabilities and peripherals

System Overview

This is the µLind motherboard in all its glory. It is designed to fit in a mini-ITX case with screw holes aligned accordingly. The utility ports are all on the back and user ports, like joysticks, keyboard, mouse and CompactFlash are on the front, easily accessible for the user. The expansion port can be soldered on either upper side of the board for stacking expansion cards on top or on the bottom side, to stack expansions below or to the side. Therefore will the expansion board header not be soldered when the board is delivered to the end user.



Block diagram of the µLind system



The block diagram shows the principle behind the µLind system. It shows what signals each subsystem is using and how they are interconnected. Do note that this is a simplified image and some signals may not be included.

Explanation of key subsystems

To make it easier to explain how µLind is interconnected it will be described as subsystem according to the block diagram.

The CPU subsystem

This is the heart of the µLind system and is connected to all main buses. It has the system clock crystal connected and will generate a E & Q clock, with the Q clock 90° before the E clock.

The Serial subsystem

To be able to connect to the µLind system in a simple way a serial controller with 2 serial ports was added. The first port is used for a USB connection, normally used as a serial terminal. The second port is a user available serial port.

Hardware Components

Description and purpose of each main component:

Processor (HD6x09)

There are two

Memory modules (RAM/ROM)

Clock generation circuitry

Input/output ports

Expansion connectors

Power supply requirements

Configuration & Options

General instructions on handling jumpers.

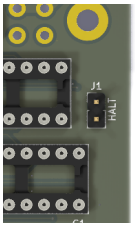
If a jumper is to be changed, do NOT do that with the power on. That can cause an electrical fault in your system and might render your board unusable. Always unplug the power connector before changing any board jumpers.

Jumper Settings and Configuration Options

The µLind board contains 16 different jumpers and on-board connectors that can be used for configuration of board functions.

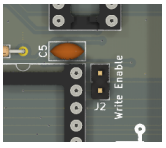
Jumpers and Internal connectors	
ID	Name
J1	Halt
J2	ROM Write Enable
J3	USB Reset Enable
J4	RGBLed
J5	Reset
J6	PWM Speaker
J7	Pot Connector (For SID board)
J8	SPI Port
J9	LightPen (For VDC board)
J10	CTS Enable (Must be used with J11)
J11	RTS Enable (Must be used with J10)
J12	Memory 1M
J13	Expansion Board
J14	Expansion Board 2
J15	(For Future Use)
J16	12v In

J1. Halt connector



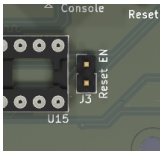
This connector lets the user add a Halt button to the µLind board. The halt button lets you halt execution, for debugging or fault tracing purposes for example.

J2. ROM Write Enable



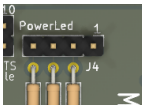
This jumper, when added, lets you write to the EEPROM. It connects the WR line to the EEPROMS WR input. Special considerations need to be taken of how the EEPROM handles writes.

J3. USB Reset Enable



If this jumper is in place the RESET line of the USB Interface is connected to the RST line on the µLind board. This will reset the USB controller every time the RESET button on the µLind board is pressed. Normally (without this jumper) will the USB controller keep the connection to the connected computer terminal during resets.

J4. RGB Led Header



The RGB LED header is used for connecting a RGB LED to show system status. This LED will show green at startup to show all systems OK. It is possible for the user to control the LED after startup.

J5. Reset Header

J6. PWM Speaker Header

J7. Pot Connector (Only for SID)

J8. SPI Port

J9. Light Pen Connector (Only for VDC)

J10. CTS Enable

J11. RTS Enable

J12. 1M Enable

This jumper needs to be added to enable the secondary memory port, expanding the memory to 1 Megabyte.

J13. Expansion board 1 Enable

Short this jumper to enable the first expansion board.

J14. Expansion board 2 Enable

Short this jumper to enable the second expansion board. This requires the first board to also be enabled. (See J13)

J15. Not used

This jumper is currently not used. Leave unused.

J16. 12v Input

This jumper is used by the 12v addon board to deliver 12v to the expansion ports. There are no 12v user on the µLind board it self, but there might be addon/expansion cards that

require higher voltage than 5v. For example a SID audio addon board will require either 12v or 9v depending on SID version.

CTS/RTS Jumper Configuration (Jumpers 2 & 3)

Description of the CTS/RTS signals.

How to configure CTS/RTS using jumpers 2 & 3.

Recommended settings and typical use cases.

Other Jumper Configurations

Explanation and usage of any additional configuration jumpers present on the board.

Compact Flash Add-on Board (Optional)

Introduction to the Compact Flash Add-on

Features and benefits.

Hardware compatibility and installation considerations.

Installation Instructions

Step-by-step installation procedure.

Necessary connections and jumpers.

Configuration and Setup

Compact Flash drive formatting and preparation.

BIOS/Firmware configuration to enable Compact Flash support.

Typical Use-Cases

Storage expansion.

Boot options and file systems supported.

Additional Expansion Options (if applicable)

Description of other possible expansions or optional modules planned or supported.

Information on compatibility and special considerations.

Assembly Instructions

List of required components

Step-by-step assembly guidance

PCB layout considerations

Testing and Troubleshooting

Initial power-up tests

Diagnostic routines

Common hardware troubleshooting scenarios and solutions

Programming and Firmware

Overview of supported software environments

Loading and updating firmware instructions

Appendices

BIOS Routines

To simplify and minimize issues when programming for the µLind there is a collection of BIOS routines that can help with a lot of standard tasks, from setting memory bank to loading a file from CF to a memory position.

Memory Map

The entire I/O system is memory mapped (some functions like the video display might be pseudo memory mapped) and therefore is every register available according to this table:

IO Unit	Adress	Nnemonic	RD: Function	WR: Function
Memory Mapper	$F400	MB0	Read current set memoryblock for page at $0000-$3FFF	Set MemoryBlock for page at $0000-$3FFF
	$F401	MB1	Read current set memoryblock for page at $4000-$7FFF	Set MemoryBlock for page at $4000-$7FFF
	$F402	MB2	Read current set memoryblock for page at $8000-$BFFF	Set MemoryBlock for page at $8000-$BFFF
	$F403	MB3	Read current set memoryblock for page at $C000-$DFFF	Set MemoryBlock for page at $C000-$DFFF (only 8k)
IRQ Handler	$F404	IRQR/IRQM	RD: low nibble current active IRQ, high nibble currently configured IRQ mask	WR: high nibble to set IRQ mask, only IRQs higher than mask will trigger IRQ
Power LED	$F405	PLS		0 = Off, 1 = Red, 2 = Green, 3 = Yellow, 4 = Blue, 5 = Violet, 6 = Cyan, 7 = White
PS2	$F406	KPR	Get current key press	
	$F407	MR	Get current mouse byte	
Compact Flash	$F408	DR	DataRegister	

	$F409	ER/FR	ErrorRegister	FeatureRegister
	$F40A	SCR	Sector count register	
	$F40B	SNR	Sector number register	
	$F40C	CLR	Cylinder low register	
	$F40D	CHR	Cylinder high register	
	$F40E	DHR	Drive head register	
	$F40F	SR/CR	Status register	Command register
Parallel (VIA)	$F410	ORB/IRA	Input Register B	Output Register B
	$F411	ORA	Input Register A	Output Register A
	$F412	DDRB	Data Direction Register B	
	$F413	DDRA	Data Direction Register A	
	$F414	T1L-L / T1C-L	T1 Counter Low	T1 Latch Low
	$F415	T1C-H	T1 Counter High	
	$F416	T1L-L	T1 Latch Low	
	$F417	T1L-H	T1 Latch High	
	$F418	T2C-L	T2 Counter Low	T2 Latch Low
	$F419	T2C-H	T2 Counter High	
	$F41A	SR	Shift Register	
	$F41B	ACR	Aux Control Register	
	$F41C	PCR	Peripheral Control Register	
	$F41D	IFR	Interupt Flag Register	
	$F41E	IER	Interupt Enable Register	
	$F41F	ORA/IRA	Same as register 1 without "handshake"	
Serial (XR88C92)	$F420	MRA	Mode Register A	
	$F421	SRA/CSRA	Status Register A	Clock Select Register A
	$F422	CRA	Reserved	Command Register A
	$F423	RXA/TXA	Receive Buffer A	Transmit Buffer A
	$F424	IPCR/ACR	Input Port Change Register	Auxiliary Control Register
	$F425	ISR/IMR	Interrupt Status Register	Interrupt Mask Register
	$F426	CUR/CTPU	Counter / Timer Upper Register	C/T Pre-load Value Upper Register
	$F427	CLR/CTPL	Counter / Timer Lower Register	C/T Pre-load Value Lower Register
	$F428	MRB	Mode Register B	
	$F429	SRB/CSRB	Status Register B	Clock Select Register B
	$F42A	CRB	Reserved	Command Register B
	$F42B	RXB/TXB	Receive Buffer B	Transmit Buffer B
	$F42C	GPR	General Purpose Register	
	$F42D	IPR/OPCR	Input Port Register	Output Port Control Register
	$F42E	STCR/SOPR	Start C/T Command	Set Output Port Register
	$F42F	SPCR/ROPR	Stop C/T Command	Reset Output Port Register

Graphics	$F440 - $F47F		Depending on graphics addon	
Audio L	$F480 - $F4BF		Depending on Audio addon	
Audio R	$F4C0 - $F4FF		Depending on Audio addon	
Expansion Port	$F500 - $F7FF		Depending on Expansions (Se expansion register mappin)	

And the memory map for the expansion board are as follow:

Expansion Board Address Logic											
		A8	A7	A6	A5	A4	A3	A2	A1	A0	
F600	F63F	0	0	0	x	x	x	x	x	x	BaseBoard 1
F640	F67F	0	0	1	x	x	x	x	x	x	Slot 1 (7 Bit)
F680	F6BF	0	1	0	x	x	x	x	x	x	
F6C0	F6DF	0	1	1	0	1	x	x	x	x	Slot 2 (5 Bit)
F6E0	F6E7	0	1	1	1	0	0	x	x	x	Slot 3 (3 Bit)
F6E8	F6EF	0	1	1	1	0	1	x	x	x	Slot 4 (3 Bit)
F6F0	F6F7	0	1	1	1	1	0	x	x	x	Slot 5 (3 Bit)
F6F8	F6FF	0	1	1	1	1	1	x	x	x	Slot 6 (3 Bit)
F700	F73F	1	0	0	x	x	x	x	x	x	BaseBoard 2
F740	F77F	1	0	1	x	x	x	x	x	x	Slot 1 (7 Bit)
F780	F7BF	1	1	0	x	x	x	x	x	x	
F7C0	F7DF	1	1	1	0	1	x	x	x	x	Slot 2 (5 Bit)
F7E0	F7E7	1	1	1	1	0	0	x	x	x	Slot 3 (3 Bit)
F7E8	F7EF	1	1	1	1	0	1	x	x	x	Slot 4 (3 Bit)
F7F0	F7F7	1	1	1	1	1	0	x	x	x	Slot 5 (3 Bit)
F7F8	F7FF	1	1	1	1	1	1	x	x	x	Slot 6 (3 Bit)

Parts list and suppliers

Pinout tables

Circuit Diagrams & Schematics

Annotated schematic diagrams

Detailed circuit descriptions

References and additional resources