Protocol for Serial Communication for Netron, Revision 1.01

Confidential

February 23, 2001

Sony Electronics, Inc
Display Systems of America
Netron Project

-- Confidential --

TABLE OF CONTENTS

Revision History

Revision 0.1 8/11/99
- Original draft.

Revision 0.2 (Draft #2) 8/16/99
- Replaced Key Scan message header 81 with 90.
- Limited Key Scan Code to 7F to reserve FF as terminator.

Revision 0.3 (Draft #3) 9/14/99
- Removed reference of ST92175 from figure 2.
- Organized the document into numbered sections.
- Replaced User Increment/Decrement commands with general ECS service mode commands.

Revision 0.4 (Draft #4) 9/24/99
- Message format is normalized as this: for unsolicited messages, the packet header is 08H; for responses, the packet header is 90H.
- Revised the code assignment for front panel buttons according to the new Netron Marketing Requirements (Revision 1.2).
- Added revision history.

Revision 0.5 (Draft #5) 10/15/1999
- Corrected typos on page 4.
- Added expected response for each message from master side.
- Added subsections (2.3.6, 2.3.7, and 2.3.8) about monitor information save/retrieve.
- Modified Key Scan subsection (2.3.9) according to new marketing spec.
- Added definition of buffer for received message (Section 2.5).
- Added Chapter 3 Flowcharts for Communication Routines
- Added Chapter 4: Monitor States and Key-panel

Revision 0.6 (Draft #6) 10/21/1999
- Added subsection 2.3.10 for "Power State Control Messages ".
- Revised section 4.1 for monitor power states.
- Revised section 4.3 to reflect the new spec on LED indicators.
- Revised subsection 2.3.3 about interpretation of LED data.
- Revised section 4.4 "Monitor States and Key-panel".

Revision 0.7 (Draft #7) 11/24/1999
- Added register control/inquire for Pin, Pin Balance, Key and Key Balance.
- Removed "Deep Sleep" state. Redefined "Standby" state.

Revision 0.8 (Draft #8) x/xx/xx
- Added Test Pattern Communication Protocol, Please refer to page 14
- Standby Mode Change to Off Mode, Please refer to page 13, 19, 20.

Revision 0.83 (Draft #8) 3/02/00
- Modify Register Control Message (Web Eging to Display), Please refer to page 07
- Modify Response to Register Inquire Message (Display to Web Egine) , Please refer to page 09
- Modify Monitor Information Save Message (Web Egine to Display), Please refer to page 10
- Modify Response to Monitor Information Retrieve Messages (Display to Web Engine), Please refer to page 12

Revision 0.84 3/10/00

- Applying new protocol with 8B code for LCC (Landing Correction Coil) Driver. Please refer to page 14.

Revision 0.85 5/12/00

- Modifying power indicator from ‘OFF’ to ‘Amber’ at OFF mode. Please refer to page 21.

Revision 0.86 (By Y. Hirose) 6/29/00

- Changed name on the cover page from Libiao Jiang to Yoshi Hirose.
- Revised section 2.3.10, Parameter Sub-code 1 “04” from “GO TO AGING” to “GO TO TEST PATTERN”
- Revised section 2.4 External Control Sequence Messages for boot to Bios Diagnostic and Test Pattern, Revised Test Pattern
- Added section 3.3 Flowchart for Boot to Diagnostic and Test Pattern
- Revised section 4.2 Key-Panel , Deleted Calendar& iRadio and added Media, Changed button number from nine to six for ID change
- Revised section 4.4 Monitor Status and Key-panel for added Test Pattern

Revision 0.9 (By Y. Hirose) 7/31/00

- Revised 2.2 to add a MSB restriction.
- Got section 2.3.3 back to Rev 0.8 for solving Syntax error.
- Got section 2.3.5 back to Rev 0.8 for solving Syntax error.
- Got section 2.3.6 back to Rev 0.8 for solving Syntax error.
- Got section 2.3.8 back to Rev 0.8 for solving Syntax error.
- Deleted “GO TO TEST PATTERN” in section 2.3.10, which is not used.
- Revised section 2.3.9 for addition of Key Map Code On/Off function.
- Revised section 2.4 for Key Map Code addition

Revision 0.91 (By Y. Hirose) 8/2/00

- Revised 2.3.9 to add Key Map Request Command.
- Revised 2.4.1 for addition of Key Map Request Command, and for clear definition.
- Changed Total Page from 22 to 23.

Revision 0.92 (By Y. Hirose) 8/16/00

- Revised section 2.3.1 for addition of Completion Message from Web to Display.
- Revised section 2.3.2 for addition of Error Message from Web to Display.
- Revised section 2.4.2 for addition or change of Test Patterns, and change of the Response Message.

Revision 0.93 (By J. Freitas) 8/29/00

- Revised Title of document to read Protocol for Serial Communication for Netron.
- Removed author’s name from document
- Reformatted revision history page.
- Added Table of Contents page.
- Revised section 2.3.3 and section 2.3.4 and included Register Table to identify all registers capable of being controlled by the protocol
- Revised section 2.3.9 to clarify Key Map Message and Key Map Request Control and Response messages
- Revised section 3.3 to describe BIOS Boot flowcharts only.
- Added BeIA Boot Flowchart
- Added Front Panel User Control matrix.
- Added communication control description for display, web engine and a factory adjustment system.
- Revised section 4.4 to eliminate E-mail LED monitor status indication description

Revision 0.94 for **DVT** (By M. Kimoto/J. Freitas) 9/18/00

- Add Recall Basic and Recall Geometry for Memory Control Messages
- Correct command structure of Memory Control Messages
- Add Monitor ID command
- Add Register Attribute Inquire Command
- Update data range of registers on Register Table
- Added note to section 2.2 (added by Joe Freitas)
- Corrected Register number range from 35 to 34 (sec 2.3.3 and 2.3.4)
- Corrected typo on duplicate Sub-code 1
- Included/corrected Register Table item 05H, 29H and 2E.
- Re-arranged BeIA Boot Flowchart diagram (no changes made).

Revision 0.95 for **DMT** (By M. Kimoto) 10/30/00

- Add Burning flag on/off in sub-code 1 for Power State Control Messages
- Add Go to Stand-by in sub-code 1 for Power State Control Messages
- Add STAND-BY state in 4.5 Monitor States and Key -panel
- Add Power State Inquire Message
- Correct Event B definition excluding VOLUME UP and DOWN button in 4.5 Monitor States and Key -panel
- Add initialization to OFF path in 4.5 Monitor States and Key -panel
- Add OFF to STAND-BY path in 4.5 Monitor States and Key -panel

Revision 0.95A for **DMT - Revised**(By M. Kimoto) 11/28/00

- Add Power On Reset Message to inform MCU gets power on reset to web engine
- Add Timeout definition for handshake in section 3.1
- Change the data range of Contrast and Brightness registers in Register Table
- Add Contrast DAC register in Register Table
- Add 2 patterns in Test pattern Code for Test Pattern Message

Revision 0.96 for **PVT** (By M. Kimoto) 1/22/01

- Add data definition for E-mail LED register in Register Table
- Change Burnin mode on/off to Warm-up mode on/off for Power State Control Message and disable this functionality

Revision 1.00 for **Pre-Production** (By M. Kimoto) 2/21/01 ***Deleted due to Error***

- No change from revision 0.96

Revision 1.01 for **Pre-Production** (By M. Kimoto) 2/23/01

- Error correction revision for revision 1.00
- Add ETI Clear and Inquiry commands

1.0 Hardware:

On the display side, the Serial Communication Interface (SCI) of the micro-controller is used for communication between the display and the Web Engine. SCI runs in full duplex asynchronous mode. The settings are:

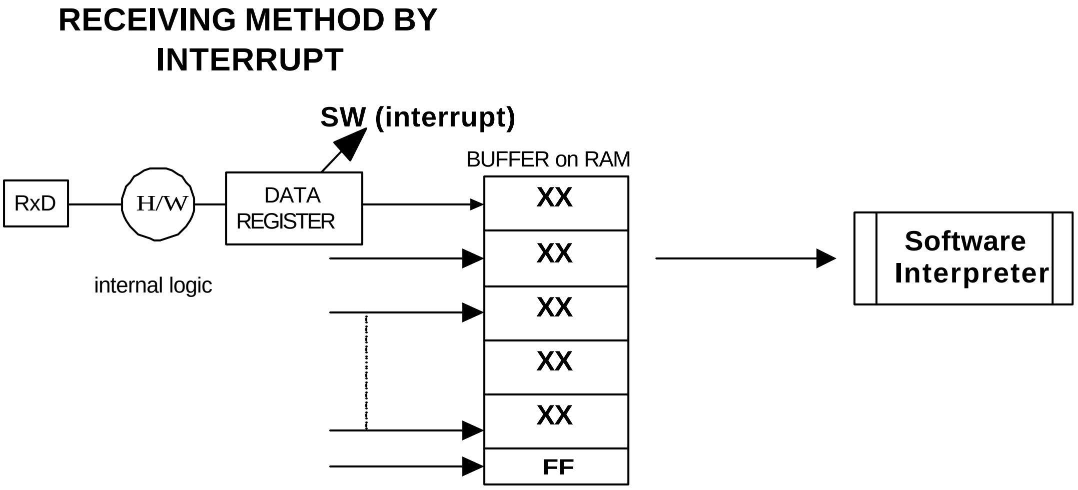
Baud rate: 9600
Data bits: 8
Parity: None
Stop bits: 1
Hardware error correction: None

Electrical characteristics: V_{DD} = 5V±10%. High = 1. Low = 0.
Quiescent marking state: Constant high.
START condition: a high to low transition.

2.0 Message Format:

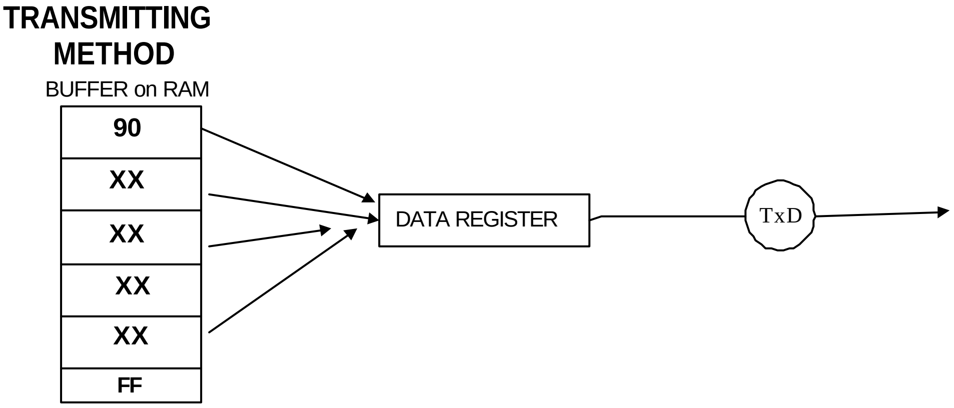
2.1 TX/RX Specification

External connections are by means of two I/O pins - TxD for Transmit Data Output and RxD for Receive Data Input.



NOTE: Please refer to the External Control Sequence Messages for Buffer RAM DATA. FF means the end of transmitting from the outside world.

Figure 1 Receiving Command



NOTE: Please refer to External Control Sequence Messages for Buffer RAM DATA. FF the means end of transmitting from the outside world.

Figure 2 Transmitting Commands

2.2 Two Types of External Control Sequence Messages

There are two types of ECS messages. The first type covers messages that are initiated by either the Web Engine or the MCU. For example, Web Engine can send a message to the MCU to control/inquire the state of the display. Another example is, when the MCU detects that the user has pressed a front panel button, a corresponding message will be sent to the Web Engine. The second type is those messages that are sent in response to incoming messages.

The first type of messages has 81H as the packet header. The second type has 90H as the packet header. One exception to this method pertains to Key Map Request response messages, where the second type also has 81H as the packet header (refer to section 2.3.11).

Therefore, on the both directions, there could be two types of messages. The MCU should not mix bytes of two different messages, i.e. care should be taken to make sure that a new message should not be sent to the Web Engine until the previous one has been completely sent out.

If a confirmation response is needed, the sender of the messages should make sure that it gets the confirmation.

MSB of all the Code and Data have to be “0” except Header and Terminator code to distinguish it from the terminator (FF).

2.3 External Control Sequence Messages (ECS) for Display Control by Web Engine

2.3.1 Completion Message

Display or Web Engine will send this message after receiving a valid message and performing the appropriate job successfully.

Type Completion Message

Syntax

Header		90	Packet Header
Body	Byte 1	50	Completion
Terminator		FF	Terminator

2.3.2 Error Message

The display or Web Engine will check the validity of the incoming messages from either one. If the incoming message is not one of those defined in this protocol, the receiver will return the following error message.

The display or Web Engine will also return the following message when it fails to do the appropriate job after receiving a valid message from the sender.

Type Error Message

Syntax

Header		90	Packet Header
Body	Byte 1	60	Error Code
	Byte 2	xx	Sub-code
Terminator		FF	Terminator

Parameter

Sub-code

02	No Function Error

2.3.3 Register Control Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to control brightness, contrast, horizontal position/size, vertical position/size, and rotation of the display. For example, if the Web Engine sends in hexadecimal 81 01 20 00 00 00 FF to the display, the display will set its vertical size to the minimum (because 0 is sent as the parameter). If the display succeeds, it will send 90 50 FF (Completion Message) back to the Web Engine. Otherwise the display will send 90 60 02 FF (Error Message).

Type Register Control

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	20	Register Control Code
	Byte 2	xx	Sub-code 1
	Byte 3	0x	Upper 4bit data
	Byte 4	0x	Lower 4bit data
Terminator		FF	Terminator

Parameter

Sub-code 1

00	Refer to Register Table
...	
34	

Response

Completion Message or Error Message

Data range values provided in the Register Table have to be converted to hexadecimal. The upper 4 bits are transmitted in byte 3 of the message body, while the lower 4 bits in byte 4.
The reason to split it into two bytes is to avoid the confusion of data FF and the terminator.

2.3.4 Register Inquire Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to get the current settings of brightness, contrast, horizontal position/size, vertical position/size, and rotation of the display. For example, if the Web Engine sends in hexadecimal 81 09 20 00 FF to the display, the display will send back to the Web Engine with a message with its current setting of vertical size in it (please refer to 2.3.5).

Type Register Inquire

Syntax

Header		81	Packet Header
Format Type		09	Read Code
Body	Byte 1	20	Register Control Code
	Byte 2	xx	Sub-code 1
Terminator		FF	Terminator

Parameter

Sub-code 1

00	Refer to Register Table
...	
34	

Response

Response to Register Inquire Messages

2.3.5 Response to Register Inquire Messages (Display to Web Engine)

Type Response to Register Inquire

Syntax

Header		90	Packet Header
Body	Byte 1	50	Completion
	Byte 2	0x	Upper 4bit data
	Byte 3	0x	Lower 4bit data
Terminator		FF	Terminator

The lower 4 bits of byte 2 and byte 3 are combined to give the hexadecimal format of the returned data. For Example, if the display returns 90 50 03 0E FF, the data is 3E. The reason to split it into two bytes is to avoid the confusion of data FF and the terminator.

2.3.6 Register Attribute Inquire Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to get the factory default settings of brightness, contrast, horizontal position/size, vertical position/size, and rotation of the display. For example, if the Web Engine sends in hexadecimal 81 09 23 04 00 FF to the display, the display will send back to the Web Engine with a message with its factory default setting of vertical size in it (please refer to 2.3.7).

Type Register Attribute Inquire

Syntax

Header		81	Packet Header
Format Type		09	Read Code
Body	Byte 1	23	Register Control Code
	Byte 2	xx	Sub-code 1
	Byte 3	xx	Sub-code 2
Terminator		FF	Terminator

Parameter

Sub-code 1

04	EEPROM value of factory default setting

Sub-code 2

00	Refer to Register Table
...	
34	

Response

Response to Register Attribute Inquire Messages

2.3.7 Response to Register Attribute Inquire Messages (Display to Web Engine)

Type Response to Register Attribute Inquire

Syntax

Header		90	Packet Header
Body	Byte 1	50	Completion
	Byte 2	0x	Upper 4bit data
	Byte 3	0x	Lower 4bit data
Terminator		FF	Terminator

The lower 4 bits of byte 2 and byte 3 are combined to give the hexadecimal format of the returned data. For Example, if the display returns 90 50 03 0E FF, the data is 3E. The reason to split it into two bytes is to avoid the confusion of data FF and the terminator.

Register Table (Refer to section 2.3.3 and section 2.3.7)

00	Vertical Size (0-255)
01	Horizontal Size (0-127)
02	Vertical Position (0-255)
03	Horizontal Position (0-127)
04	Pincushion (0-127)
05	*Reserved for later use*
06	Keystone (0-127)
07	Pincushion Balance (0-127)
08	Keystone Balance (0-127)
09	Rotation (0-255)
0A	Contrast (50-127)
0B	Brightness (44-150)
0C	E-mail LED (1: on, 7: off)
0D	H Driver Duty
0E	
0F	H-Moire
10	*Reserved for later use*
11	HFocTrac *(Dynamic HFoc Tracking with HSize)*
12	HFocPh *(H. Focus Keystone)*
13	BOutPol *(B+ out polarity)*
14	*Reserved for later use*
15	*Reserved for later use*
16	SSel *(Vertical S-Correction Linearity)*
17	CSel *(Vertical C-Correction Linearity)*
18	V-Moire
19	*Reserved for later use*
1A	*Reserved for later use*
1B	CornTop *(Corner Top Correction)*
1C	CornBot *(Corner Bottom Correction)*
1D	*Reserved for later use*
1E	*Reserved for later use*
1F	*Reserved for later use*
20	HTopCorn *(Horizontal Top Phase Corner)*
21	HBotCorn *(Horizontal Bottom Phase Corner)*
22	Vfocus *(Vertical Dynamic Focus Parabola)*
23	*Reserved for later use*
24	*Reserved for later use*
25	*Reserved for later use*
26	Contrast DAC
27	R-Drive
28	G-Drive
29	B-Drive

2A	*Reserved for later use*
2B	*Reserved for later use*
2C	*Reserved for later use*
2D	*Reserved for later use*
2E	R-CUT
2F	G-CUT
30	B-CUT
31	Band Width *(NS IC, to determine Cut-Off level.)*
32	ABL
33	G2
34	LCC

2.3.8 Monitor Information Save Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to save model name, serial number, and production week and year into EEPROM. For example, if the Web Engine sends in hexadecimal 81 01 71 02 02 01 02 FF to the display, the display will store 12 as the production week into EEPROM. If the display succeeds, it will send 90 50 FF (Completion Message) back to the Web Engine. Otherwise the display will send 90 60 02 FF (Error Message).

Type Monitor Information Save

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	71	Monitor Information Code
	Byte 2	xx	Sub-code 1
	Byte 3	xx	Sub-code 2 (Data length)
	Byte 4	0x	Upper 4bit data
	Byte 5	0x	Lower 4bit data
	…	…	…
Terminator		FF	Terminator

Parameter

Sub-code 1

00	Model name (10 Bytes ASCII)
01	Serial Number (8 Bytes ASCII)
02	Production Week (2 Bytes ASCII)
03	Production Year (4 Bytes ASCII)

Sub-code 2

01	Data length is one byte
02	Data length is two bytes
…	…
10	Data length is 16 bytes

Response

Completion Message or Error Message

2.3.9 Monitor Information Retrieve Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to retrieve information about model name, serial number, production week, and production year. For example, if the Web Engine sends in hexadecimal 81 09 71 02 02 FF to the display, the display will retrieve production week number from EEPROM and send it back to the Web Engine.

Type Monitor Information Retrieve

Syntax

Header		81	Packet Header
Format Type		09	Read Code
Body	Byte 1	71	Register Control Code
	Byte 2	xx	Sub-code 1
	Byte 3	xx	Sub-code 2 (Data length)
Terminator		FF	Terminator

Parameter

Sub-code 1

00	Model name (10 Bytes ASCII)
01	Serial Number (8 Bytes ASCII)
02	Production Week (2 Bytes ASCII)
03	Production Year (4 Bytes ASCII)

Sub-code 2

01	Data length is one byte
02	Data length is two bytes
…	…
10	Data length is 16 bytes

Response

Response to Monitor Information Retrieve Messages

2.3.10 Response to Monitor Information Retrieve Messages (Display to Web Engine)

Type Response to Monitor Information Retrieve

Syntax

Header		90	Packet Header
Body	Byte 1	50	Completion
	Byte 2	0x	Upper 4bit data
	Byte 3	0x	Lower 4bit data
	…	…	…
Terminator		FF	Terminator

The actual number of bytes is variable, depending on the actual length of the data.

2.3.11 Switch Control Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to activate the degauss or to request the key map information of the display. If the display succeeds, it will send 90 50 FF (Completion Message) back to the Web Engine. Otherwise the display will send 90 60 02 FF (Error Message).

Type Switch Control

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	60	Switch Control Code
	Byte 2	xx	Sub-code 1
	Byte 3	xx	Sub-code 2
Terminator		FF	Terminator

Parameter

Sub-code 1

00	Degauss
01	Key Map Message On/Off *
02	Key Map Request

Sub-code 2

00	Off **
01	On

* Key Map Message is "Off" by default.
** Sub-code 00 is used to turn OFF Key Map Messages only.

Response

For Degauss and Key Map Messages On/Off:
Completion Message or Error Message
For Key Map Request:
Completion Message as below or Error Message

Syntax

Header		81	Packet Header
Body	Byte 1	01	Write/Response Code
	Byte 2	61	Key Map Message
	Byte 3	xx	Key Map Data Byte***
Terminator		FF	Terminator

*** 7 bits - see section 2.4.1 – Key Map Messages

2.3.12 Power State Control Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to control power state of the display. If the display succeeds, it will send 90 50 FF (Completion Message) back to the Web Engine. Otherwise the display will send 90 60 02 FF (Error Message).

Type Power State Control

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	00	Power State Control Code
	Byte 2	xx	Sub-code 1
Terminator		FF	Terminator

Parameter

Sub-code 1

02	GO TO POWER ON
03	GO TO OFF
04	WARM-UP FLAG ON *(not supported)*
05	WARM-UP FLAG OFF *(not supproted)*
06	GO TO STAND-BY

Response

Completion Message or Error Message

2.3.13 Power State Inquire Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to get the current power status of the display. For example, if the Web Engine sends in hexadecimal 81 09 00 FF to the display, the display will send back to the Web Engine with a message with its current setting of power state in it (please refer to 2.3.14).

Type Register Attribute Inquire

Syntax

Header		81	Packet Header
Format Type		09	Read Code
Body	Byte 1	00	Power State Control Code
Terminator		FF	Terminator

Response

Response to Power State Inquire Messages

2.3.14 Response to Power State Inquire Messages (Display to Web Engine)

Type Response to Power State Inquire

Syntax

Header		90	Packet Header
Body	Byte 1	50	Completion
	Byte 2	0x	Upper 4bit data
	Byte 3	0x	Lower 4bit data
Terminator		FF	Terminator

Parameter

Data definition

02	POWER ON
03	POWER OFF
06	STAND-BY
10	FAILURE

The lower 4 bits of byte 2 and byte 3 are combined to give the hexadecimal format of the returned data. For Example, if the display returns 90 50 00 02 FF, the data is 02. The reason to split it into two bytes is to avoid the confusion of data FF and the terminator.

2.3.15 Memory Control Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to save changes or to restore original settings to the display. If the display succeeds, it will send 90 50 FF (Completion Message) back to the Web Engine. Otherwise the display will send 90 60 02 FF (Error Message).

Type Power State Control

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	7C	Memory Control Code
	Byte 2	xx	Sub-code 1
	Byte 3	xx	Sub-code 2
Terminator		FF	Terminator

Parameter

Sub-code 1

20	Save to User Area
21	Save to Factory Area
47	Recall All Factory Area to User Area (and Save)
48	Recall Basic Factory Area to User Area (and Save)
49	Recall Geometry Factory Area to User Area (and Save)

Sub-code 2

00	All

Response

Completion Message or Error Message

2.3.16 Monitor ID Inquire Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to get the software ID and software version of display CPU. For example, if the Web Engine sends in hexadecimal 81 09 76 01 FF to the display, the display will send back to the Web Engine with a message with its chassis ID and firmware version in it (please refer to 2.3.15).

Type Monitor ID Inquire

Syntax

Header		81	Packet Header
Format Type		09	Read Code
Body	Byte 1	76	ID Code
	Byte 2	xx	Sub-code 1
Terminator		FF	Terminator

Parameter

Sub-code 1

01	Software Version

Response

Response to Monitor ID Inquire Messages

2.3.17 Response to Monitor ID Inquire Messages (Display to Web Engine)

Type Response to Monitor ID Inquire

Syntax

Header		90	Packet Header
Body	Byte 1	50	Completion
	Byte 2	0x	Upper 4bit data (1/2)
	Byte 3	0x	Lower 4bit data (1/2)
	Byte 4	0x	Upper 4bit data (2/2)
	Byte 5	0x	Lower 4bit data (2/2)
Terminator		FF	Terminator

The lower 4 bits of byte 2 , byte 3, byte 4, and byte 5 are combined to give 2 bytes of hexadecimal format of the returned data. For Example, if the display returns 90 50 0C 01 01 00 FF, the data is C101. The reason to split it into two bytes is to avoid the confusion of data FF and the terminator.

The data should be converted the format indicated below.

Data Format of Software Version

Software ID = fixed (C1 for Netron)	Version: 4 bits (0 - F)	Revision: 4 bits (0 - F)

eg) Version 1.0 should be C110H.

2.3.18 ETI Clear Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to clear ETI data memorized in monitor MCU. If the display succeeds, it will send 90 50 FF (Completion Message) back to the Web Engine. Otherwise the display will send 90 60 02 FF (Error Message).

Type ETI Clear

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	7F	EEPROM Control Code
	Byte 2	60	Switch Code
	Byte 3	11	ETI
	Byte 4	00	Clear
Terminator		FF	Terminator

Response

Completion Message or Error Message

2.3.19 ETI Inquire Messages (Web Engine to Display)

The following byte sequences sent from the Web Engine to the display are used to get the ETI data. For example, if the Web Engine sends in hexadecimal 81 09 60 11 FF to the display, the display will send back to the Web Engine with a message with the current ETI data in it (please refer to 2.3.20).

Type ETI Inquire

Syntax

Header		81	Packet Header
Format Type		09	Read Code
Body	Byte 1	60	Switch Code
	Byte 2	11	ETI
Terminator		FF	Terminator

Response

Response to ETI Inquire Messages

2.3.20 Response to ETI Inquire Messages (Display to Web Engine)

Type Response to ETI Inquire

Syntax

Header		90	Packet Header
Body	Byte 1	50	Completion
	Byte 2	0x	Upper 4bit data (1/3)
	Byte 3	0x	Lower 4bit data (1/3)
	Byte 4	0x	Upper 4bit data (2/3)
	Byte 5	0x	Lower 4bit data (2/3)
	Byte 6	0x	Upper 4bit data (3/3)
	Byte 7	0x	Lower 4bit data (3/3)
Terminator		FF	Terminator

This 12 bits value shows the ETI data in 30 minutes unit.

eg) 128.5 hours is data of 000101H.

2.4 External Control Sequence Messages (ECS) for Pressed Button Decoding.

2.4.1 Key Map Message (Display to Web Engine)

The following are unsolicited messages from display to Web Engine when Key Map Message On/Off is "On" and a front panel button is pressed down or released.
These codes are generated from D board to web Engine through serial port.

Type Key Map

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	61	Key Map Message
	Byte 2	xx	7 bits Key Map
Terminator		FF	Terminator

Key Map Code

Bit 0	Power SW On/Off
Bit 1	E-mail On/Off
Bit 2	Netron On/Off
Bit 3	Web On/Off
Bit 4	Volume Up On/Off
Bit 5	Volume Down On/Off
Bit 6	0 *(Reserved for future use)*
Bit 7	0 - *Must be 0*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	Volume Down	Volume Up	Web	Netron	E-mail	Power SW

On: 1
Off: 0

Note 1: If any keys changes state of On or Off, the display will send Key Map information to Web Engine except if Key Map Message On/Off is "Off". (Default is "Off")

Response

None.

2.4.2 Test Pattern Message (Display to Web Engine)

The following are unsolicited messages from Test system to Web Engine through serial port on D board when Test system button is pressed.

Type Test Pattern

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	62	Test Pattern Message
	Byte 2	xx	Pattern Code
Terminator		FF	Terminator

Test Pattern Code

00	All White
01	Cross Hatch (12x16 - HxV)
02	Inverted Cross Hatch (12x16 - HxV)
03	mE Pattern
04	Color Bar
05	Gray Scale (16 steps)
06	60% White
07	Dot
08	1-Dot ON, 1-Dot Off
09	Black (Inverted All White)
0A	CG-NEW(5x7)
0B	CG-NEW (7x9)
0C	CG-NEW (9x11)
0D	CG-REF(5x7)
0E	CG-REF (7x9)
0F	CG-REF (9x11)
10	Run H Pattern Program
11	CG(5x7)
12	CG(7x9)
13	CG(9x11)
14	FRAME
15	V-LINER
16	Block WHITE
17	H-LINER
18	CROSSH
19	FOCUS
1A	PURITY
1B	Cross Hatch (6x8 - HxV)
…1F	*Reserved for future use*
20	All Red

21	All Green
22	All Blue
…2F	*Reserved for future use*
30	Inverse
31	Red On/Off
32	Green On/Off
33	Blue On/Off
…3F	*Reserved for future use*
40	*LCC Driver (From Test system to D board only)
…7F	*Reserved for future use*

Note: the MSB of Key Scan Code is always “0” to distinguish it from the terminator (FF).

Response

Completion Message or Error Message

2.4.3 Power On Reset Message (Display to Web Engine)

The following are unsolicited messages from display to Web Engine when 6.5 seconds pass after MCU gets power on reset in order for Web Engine to send enable key message command of MCU, regardless of the power state. This command must send just once when MCU gets power on reset and shouldn't send more than 2 times. These codes are generated from D board to web Engine through serial port.

Type Power On Reset

Syntax

Header		81	Packet Header
Format Type		01	Write Code
Body	Byte 1	70	Status Message
	Byte 2	50	Power On Reset
Terminator		FF	Terminator

Response

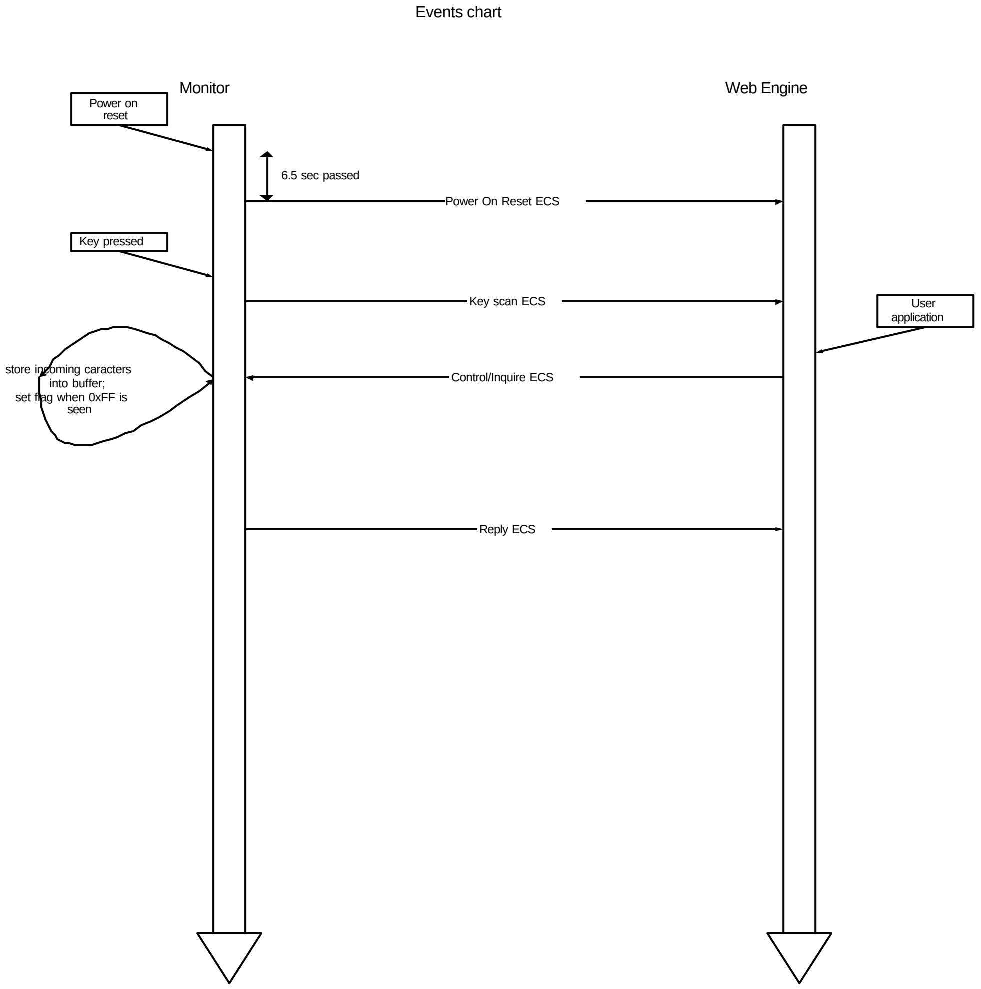
None.

2.5 Size of Buffer for Received Message

64 Bytes.

3.0 Communication Method and Flowcharts

3.1 Flowcharts for Communication Routines

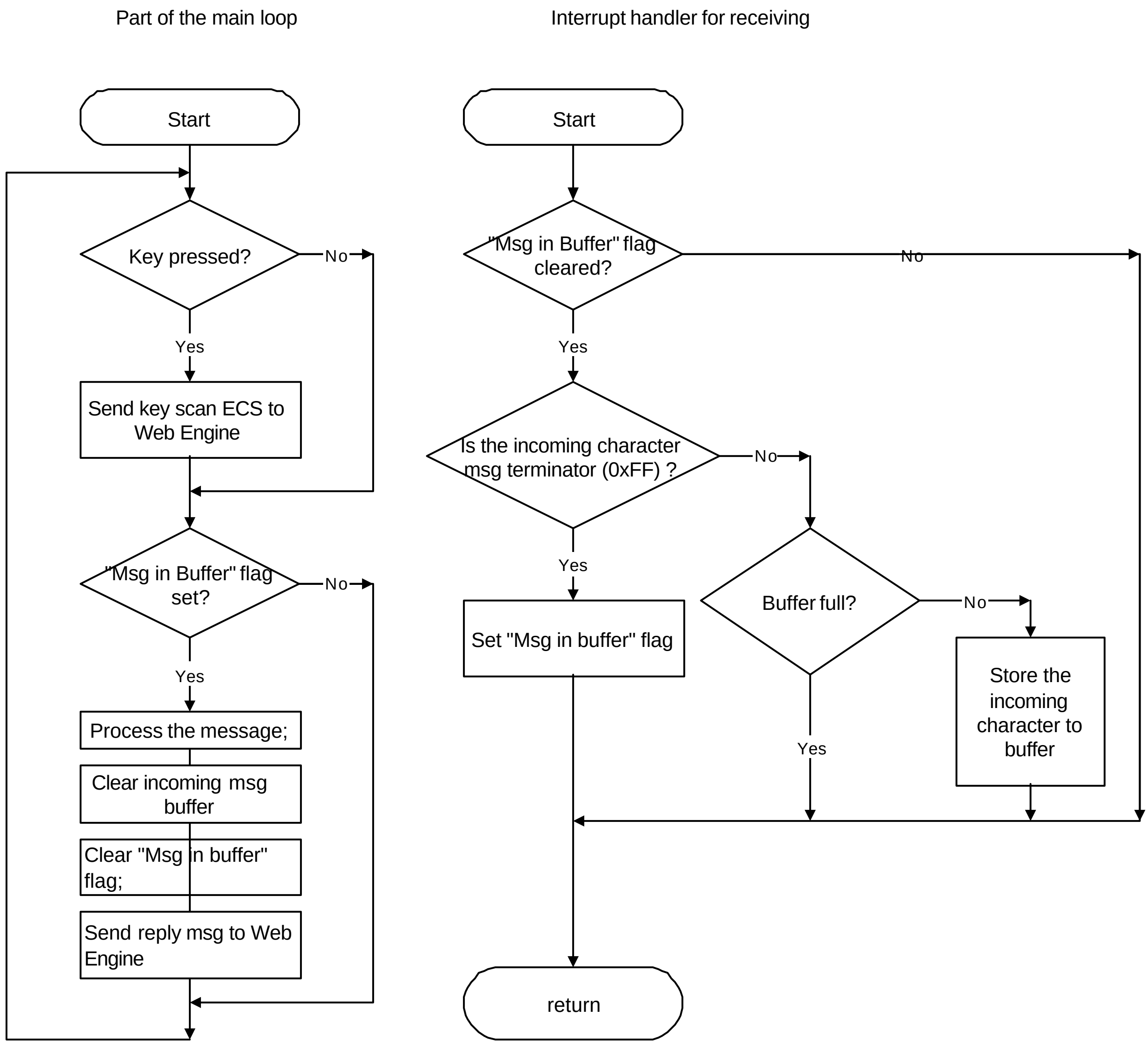


<< Definition of Time Out >>

Web engine must wait for the reply from monitor MCU before issuing next command unless MCU fails to respond the original message in the specified time as below.

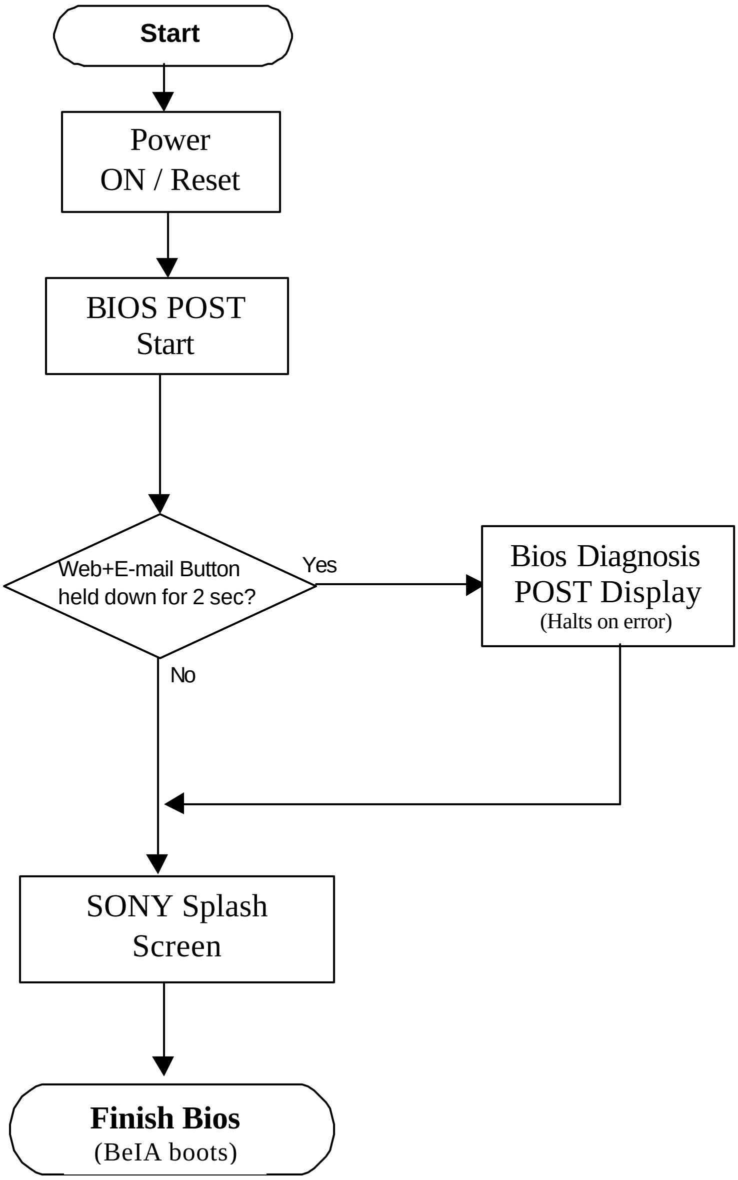
- Power State Control Message	1700ms
- Others	250ms

3.2 Flowchart for Communication with Web Engine

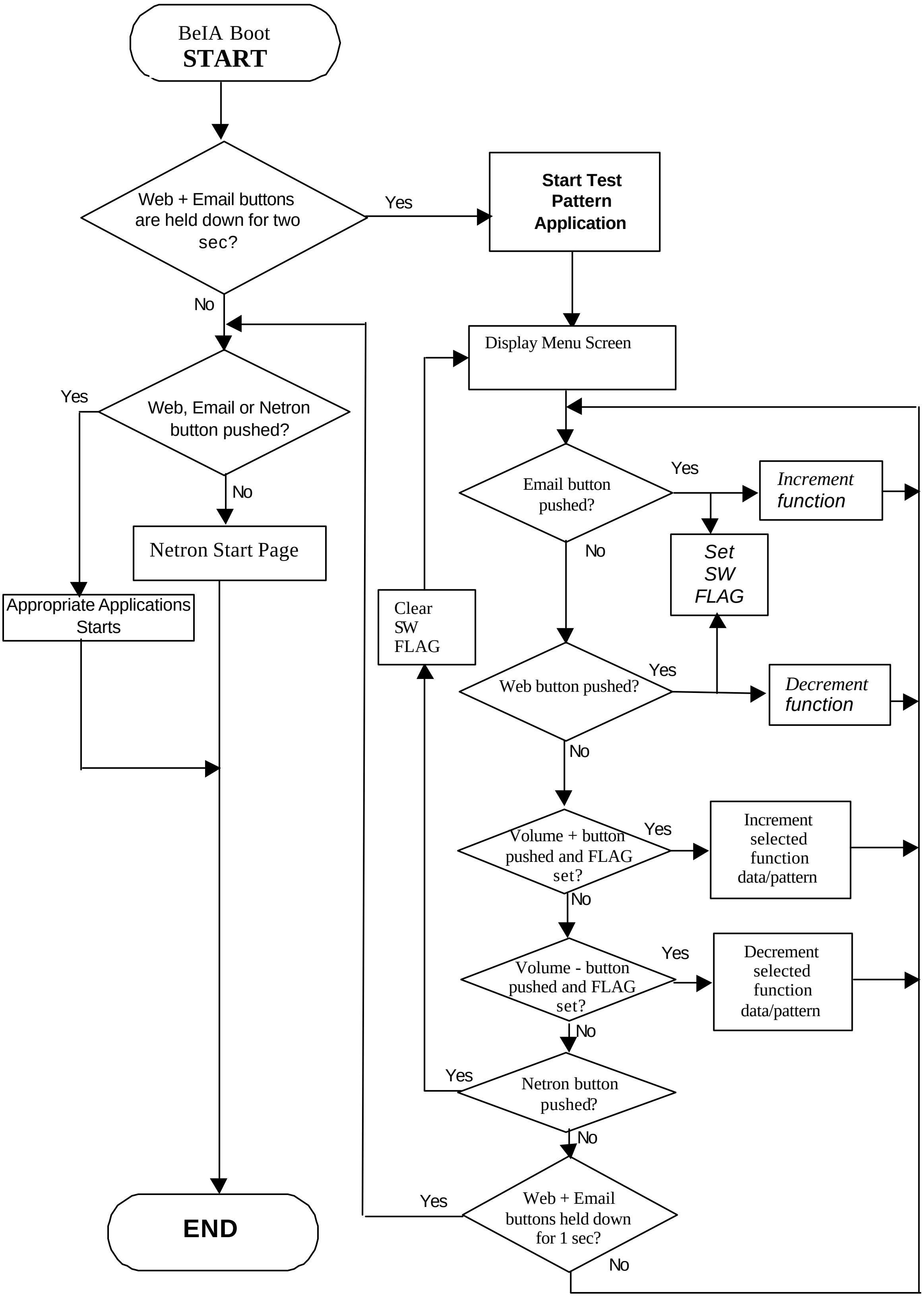

Part of the main loop
Interrupt handler for receiving
Start
Key pressed?
No
Yes
Send key scan ECS to Web Engine
"Msg in Buffer" flag set?
No
Yes
Process the message;
Clear incoming msg buffer
Clear "Msg in buffer" flag;
Send reply msg to Web Engine
Start
"Msg in Buffer" flag cleared?
No
Yes
Is the incoming character msg terminator (0xFF) ?
No
Yes
Set "Msg in buffer" flag
Buffer full?
No
Yes
Store the incoming character to buffer
return

Send reply msg to Web

3.3 BIOS Boot Flowchart



3.4 BeIA Boot Flowchart

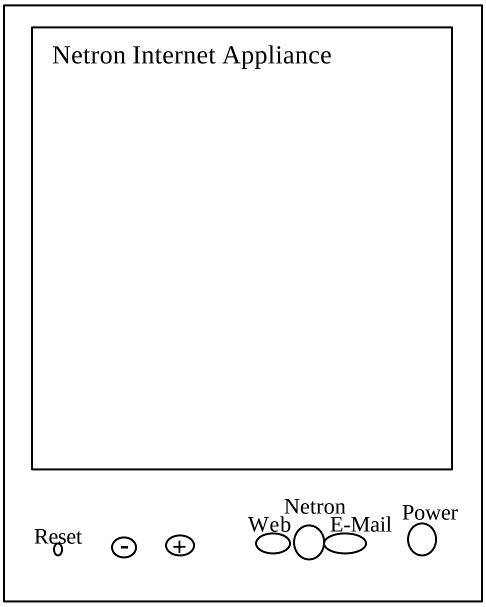

BeIA Boot
START
Web + Email buttons are held down for two sec?
Yes
Start Test Pattern Application
No
Display Menu Screen
Yes
Web, Email or Netron button pushed?
No
Netron Start Page
Appropriate Applications Starts
Email button pushed?
Yes
Increment function
Set SW FLAG
No
Clear SW FLAG
Web button pushed?
Yes
Decrement function
No
Volume + button pushed and FLAG set?
Yes
Increment selected function data/pattern
No
Volume - button pushed and FLAG set?
Yes
Decrement selected function data/pattern
No
Yes
Netron button pushed?
No
Yes
Web + Email buttons held down for 1 sec?
No
END

4.0 Monitor States and User Control

4.1 Monitor States

	Power On	Off Mode	Stand-by Mode	Failure
Heater	ON	OFF	OFF	OFF
Video	ON	OFF	OFF	OFF
HV	ON	OFF	OFF	OFF
Deflection	ON	OFF	OFF	OFF

4.2 Front Panel User Control



FUNCTION	FRONT PANEL	KEYBOARD
Start Test Pattern Program(after BIOS)	Web + Email (2 sec)	N/A *(Note: During boot up only)*
Increment function	Email	→
Decrement function	Web	←
Increment selected function data/pattern	Volume +	↑ or Space key
Decrement selected function data/pattern	Volume –	↓
Quit Test Pattern Program	Web + Email (2 sec)	‘q’ or ‘Q’
Menu	Netron	‘m’ or ‘M’

4.3 Control Description for Test Patterns

There are three possible methods of changing test patterns and modifying MCU register values on the Netron system. Each method communicates with the MCU and the CPU in a different manner as described below:

Definitions:
MCU: The micro controller in the display unit.
CPU: The Micro processor in the web engine board.

4.3.1 Netron front panel button push

The MCU senses the key and sends *ECS for Pressed Button* to CPU.
CPU interprets the key press and processes appropriate command.
If scanned key required a pattern display, the CPU processes image display.
If scanned key requires a change to a MCU register value, the CPU sends *Register Control Messages* to the MCU.

4.3.2. Netron Keyboard key press:

The CPU interprets the key press:
If the key pressed requires an image change, the CPU process image display.
If the key press requires a change to a MCU register, the CPU sends *Register Control Messages* to the MCU.

4.3.3 Factory Adjustment Control:

The factory control equipment sends messages to the CPU, via the MCU using the *Test Pattern Message* command.
The CPU processes image display, and on success, returns an OK or error message.
If factory requires changing a MCU register, the factory equipment communicates directly with the MCU. No communication to the CPU is made.

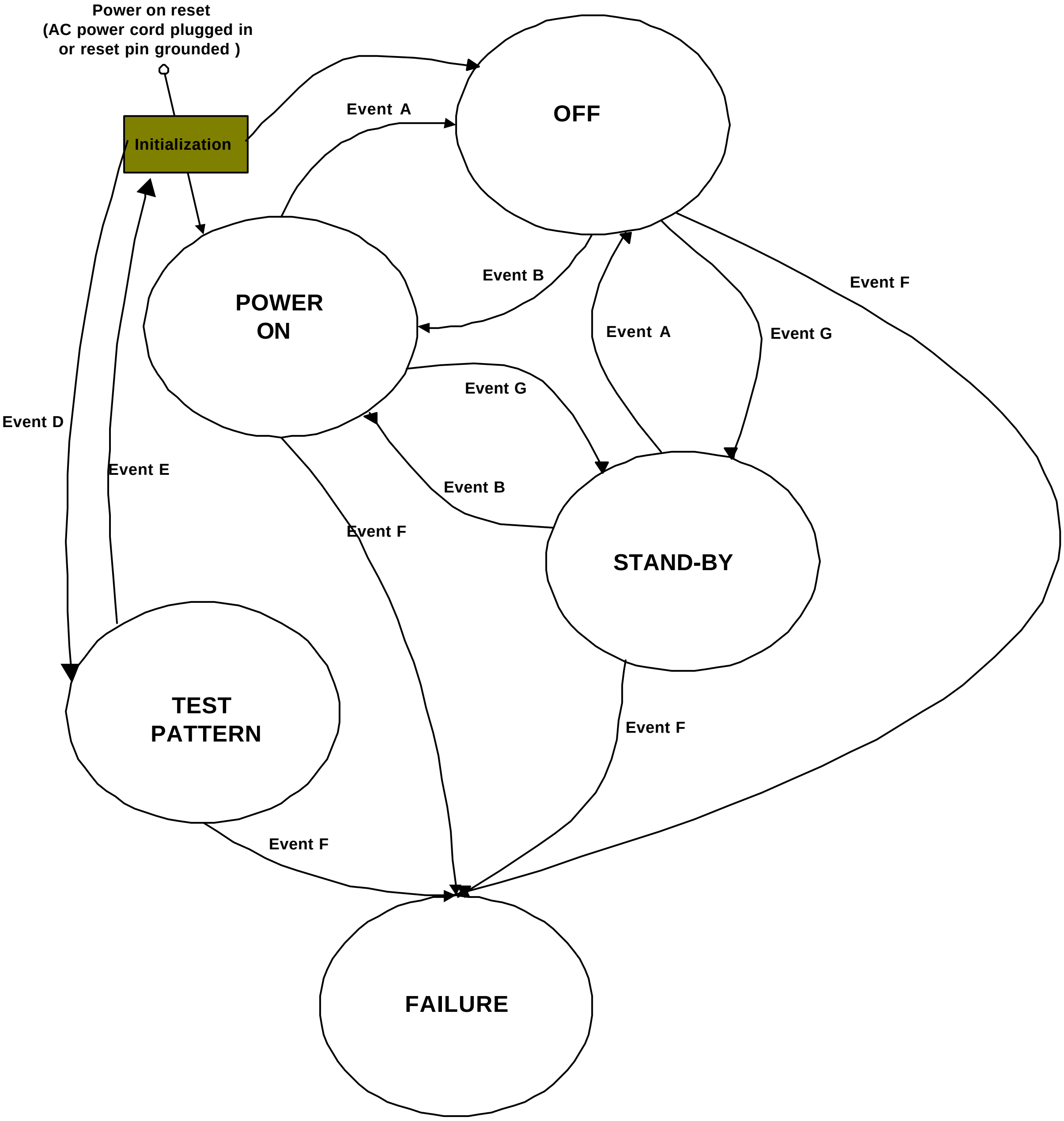
4.4 LED Indicators

According to ECS message received from the Web Engine, the micro-controller will set this LED to one of the following states:

LED Pattern	Power Indicator	Meaning
1	Green	Normal On
2	Amber	Stand-by mode
3	Amber (0.5 sec)/Off (0.5 sec)…	Failure 1
4	Amber (1.5 sec)/Off (0.5 sec)…	Failure 2
5	Amber (0.5 sec)/Off (1.5 sec)…	Failure 3
6	Amber (0.5 sec)/Green (0.5 sec) …	Failure 4
7	Off	Off mode AC Power Off

Failure 1: HV or +B failure
Failure 2: H stop or V stop failure
Failure 3: Excessive anode current failure
Failure 4: No sync (Web Engine failure)

4.5 Monitor States and Key-panel



Event A: A "GO TO OFF" ECS command is received.
Event B: Any front panel button except the VOLUME UP, VOLUME DOWN, or RESET is pressed, or a "GO TO ON" ECS command is received.
Event D: "Web" and "E-mail" buttons are held down for two seconds, or a "GO TO TEST PATTERN" ECS command is received.
Event E: "Netron " button is pushed, or a "GO TO ON" ECS command is received.
Event F: Any of the failure conditions occurs.
Event G: a "GO TO STAND-BY" ECS command is received.