Nine Tiles

Aubergine platform

Introduction

The Aubergine box is intended as a testbed for the Flexilink routing protocols. It includes a small number of network interfaces, and interfaces for professional digital audio and video standards. Thus it can be used as a switch, or as an edge device, or both.

Flexilink provides two services, an AV service for time-critical continuous media such as live audio and video and an IT service for data applications such as file transfer and web surfing. AV flows are unidirectional and multicast, while IT flows are unicast and in the current implementation are bidirectional. (A "quasi-connectionless" IT service, which is unidirectional and many-to-one, is also defined for supporting routing directly on IP addresses, but is not currently implemented. Where a bidirectional AV service is required, as in a telephone conversation, two flows are used, each with a single destination.)

The AV service needs to keep a tight control on the latency, and therefore uses a different form of routing to the store-and-forward paradigm used in IT networks. This requires a different MAC layer format on links between network elements (i.e. between Aubergines), though standard physical layers are used. We use the following terminology:

a "**physical link**" is one that uses the Flexilink MAC layer

a Flexilink "**island**" is one or more network elements connected together by physical links

a "**virtual link**" tunnels the Flexilink protocols over another network technology

a Flexilink "**cloud**" is one or more islands connected together by virtual links

Virtual links can be over Ethernet (layer 2) or UDP/IP (layer 3). AV flows will experience whatever QoS the underlying network provides and the de-jitter buffers which are inserted at the receiving end will, of course, add to the latency.

Within an island, "house sync" can be distributed accurately enough to meet AES11.

Audio interface standards supported by the Aubergine are AES3 (via AES59, "Tascam" cables) and AES10 (MADI). Support for AES67 as an interface to audio equipment, including synchronisation via PTP, is on the "to do" list.

Connections to the Aubergine box

The power input is on the back panel (12 to 48v DC ±20%, centre positive).

The ports on the front panel are as follows:

1-4: 10/100/1000 Mb/s Ethernet

When connected to an Ethernet port of another Aubergine, or to anything that is outputting the format we use on physical links, they switch to using the Flexilink MAC layer at 1Gb/s. Otherwise the standard Ethernet MAC layer is used, and an IP address is requested via DHCP. See below for more details, including the LED indications.

If port 4 is a physical link it implements a debug server; this uses an ad hoc protocol which we call Simple Control Protocol on an IT flow with a label value that's out of range for normal IT flows.

5-6: USB (not implemented)

These ports were intended to be used for control and monitoring protocols that had previously been implemented over RS232, as well as for "virtual sound card" functions. However, in practice it has been found to be easier to implement these features using UDP on the network ports, and there is no support for USB in the current software.

7: MADI out

This was disabled in the software that was current in 2016, because it used an out-of-date packet format on the network. I'm not sure whether it has been updated yet.

8: MADI in

As for MADI out.

9: Word clock in

If a 44.1 or 48 kHz signal is present on one unit in an island, it will provide audio sync for the whole island. See also "audio synchronisation" below.

10,11: 1Gb/s Ethernet (standard Ethernet modules)

This has only been tested with copper; fibre should be OK for a link between two Aubergines but it's possible that more of the link-up protocols need to be implemented to connect to a layer 2 or 3 switch. The behaviour is the same as ports 1 to 4, except that there are no LEDs.

12,13: 3G-SDI (MSA-compliant SDI modules)

These ports do not do anything useful with the software that identifies itself to the Controller program as a Flexilink switch. There is a separate software which identifies itself as a video interface; this omits most of the support for IP, so as to leave room in the FPGA for the video support, which includes light compression (with a delay of about 8 picture lines) reducing 1080p60 from its raw 2.97 Gb/s to about 750 Mb/s on the network. It supports most HD formats,

but SD is not currently implemented. It was developed using HDMI modules from Embrionix, but should also support SDI. The input must be on port 13; the output is provided on both 12 (for unidirectional modules) and 13 (for bidirectional).

14-21: AES3 (14-17 in, 18-21 out)

The front panel connection uses the pinout specified in Annex D of AES59, usually referred to in catalogues as the Tascam pinout.

The inputs will recognise any frame rate up to 384kHz. The software that was current in 2016 assumed 44.1 or 48kHz when setting up connections on the network; I'm not sure whether other rates are supported now.

The outputs sync up to the word clock on port 9 if present, else to a word clock on port 9 of another unit in the same island, else to the incoming stream from the network.

Synchronisation

There are three timing domains, as follows. Firstly, the Flexilink frames on physical links are all phase-aligned; this is required to make the routing of AV flows work. Frame alignment is independent of the other two timing domains, and is entirely automatic.

Secondly, there is "network time" which within an island is carried in a 32-bit field at the front of each frame. It is also carried in the IT packets on a virtual link, but wouldn't be as accurate as PTP on those links. As with PTP, the system chooses the best source of time that is available; in the case of the Aubergines this may be an audio word clock.

Thirdly, in the Aubergines there are the media clocks (audio and video), which can be aligned to a word clock input, to network time, or (if no other reference is available) controlled to match the sample rate of a stream being received across the network. Note that the current software doesn't do any sample rate conversion.

It's best if you plug a house sync signal into port 9 on one of the boxes in each island (it doesn't matter which) and allow about 15 seconds for the audio clocks to sync up to it before setting up any audio flows; it's also supposed to recognise a DARS on the AES3 inputs (by checking the channel status), but that isn't implemented yet. We don't sync up to non-DARS AES3 inputs in case they are in turn taking their timing from the unit's outputs. Taking the timing from PTP isn't implemented yet either. If there isn't any external reference, it will take its timing from an incoming network connection, but it can take a while to sync up to it, during which there will be sample slips.

Controller program

The Controller provides a "user" interface for one or more target devices, communicating with the Flexilink cloud over IP, and can update the target's software and control and monitor various functionality within the targets. It is intended to provide an "engineering" interface to the system, and to show developers of control systems how a back-end interface to the network would function. It is not intended as an interface for non-technical users, although in a small network it provides an easy way to set up and tear down connections between audio equipment.

It communicates with the Flexilink cloud via an IP network, using an external virtual link (see “Network port details” below) over UDP. It uses an SNMP-like protocol to read and display information from the target's MIB, and can update the target's software and control various functionality within the target. This description applies from version 2.1.1.

Versions up to and including 2.0 were developed in Visual Studio 2008 on Windows XP but (somewhat to my surprise) work in Windows 8.1 too, and probably in all the versions in between. Versions from 2.1 to 2.3.3 were developed in Visual Studio 2015 on Windows 8.1, and from 2.3.4 in Visual Studio 2019 on Windows 10.

There isn't any formal "installation" process, just double-click on the .exe file in Windows Explorer to run it. However (1) you will probably have to tell Windows Firewall to allow it to access the network and (2) it should be in a separate directory (or folder) along with the product file, software images, and configuration files; these other files aren't necessary unless you want to use the facilities they implement, such as automatic software updates.

There are three ways you can connect a PC to the Flexilink network:

(a) if the PC is on an IP network that has a DHCP server, just connect a port on one of the Aubergines into the IP network

(b) connect a port on one of the Aubergines into an IP network that has a DHCP server, and then connect the PC's Ethernet port to an Aubergine; this will also create a layer 2 tunnel from the PC to the IP network

(c) connect the PC's Ethernet port into port 4 of one of the Aubergines; you may also need to give the PC a fixed IPv4 address that begins 192.168.4 (but avoid 192.168.4.10 and 192.168.4.11).

When the program is run it broadcasts a request on the IP network to which the Aubergine responds. If it fails to connect, it retries every 7 to 8 sec.

As soon as the virtual link is connected, the Controller sets up a management connection to the unit at the other end of the link and downloads various objects from its MIB. If any of those show it has a link to another unit, the Controller also sets up a management connection to that unit; this process continues until it has management connections to all the units it has discovered. If there is, or recently was, another Controller connected to the Flexilink cloud, there may be a delay of up to 20 seconds before the information arrives; this is because the Controller uses information from periodic Status Broadcast messages, and if the unit is already sending out status broadcasts it won't restart the cycle when the new Controller joins it.

If there is a product file in the Controller's directory that matches a unit's product code but the software the unit is running does not match the version numbers in the product file, the controller puts up a dialogue box asking whether to upload new software.

There are three windows; “sources” lists all the media inputs it has found, and “destinations” lists the media outputs. (Note that the software running in the Aubergine will not tell it about any physical interfaces it doesn't support.) If there is a signal on the input there is an indication such as “44100” for 44.1kHz audio or “1080p60” for full HD video. Left click on a source to select it; left click on a destination to tell it to take audio or video from the selected source; right click to tear the connection down. The selected port is displayed with a yellow highlight. Note that the AV service is inherently multicast; if you left click on several outputs, they will all get a copy. If

you select an output that is already receiving from another source, it will be disconnected from the other source; this may cause a click because we simply switch from one digital stream to the other, without doing any kind of fade-across.

There is also a “heading” line for each unit; the part in parentheses reports the sources of the audio sync (see 'v' command below), network time (which is similar to PTP; first number is sync source type, e.g. 02 = local, 8C = from word clock, second is source unit, third is distance from source measured as estimated uncertainty in the time in ns), and the alignment of Flexilink frames (numbers similar to network time; “*m*+*n*” as distance means *m* virtual links + physical links with a combined uncertainty of *n* ns). Currently the “uncertainty” is fixed (in the Aubergine software) at 80 ns for a physical link; I think in practice the performance is better than that, and there are some things that could be done to improve the information, such as exchanging PTP packets before switching to the Flexilink format.

The colours for the header lines are: magenta = trying to connect, green = connected and is running the software versions in the product file, blue = connected and is running different software (or product file not found), red = some kind of failure (but usually it switches to magenta almost immediately, as it tries to reconnect).

Colours for media outputs in the “destinations” window depend on their “importance” value (see “LED indications” below); if connected to a network flow they are: green for 1-63, blue for 64-127, brown for 128-191, red for 192-255. If not connected, they are black for 1-127, magenta for 128-255.

Media inputs in the “sources” window are displayed in black if not connected, magenta if the flow to which they are connected cannot be identified. Otherwise they show the importance of the flow, using the same colours as for outputs.

Each flow has a 16-byte globally-unique identifier, which is shown against the port that is transmitting it in the “sources” window. Destinations that are connected show the transmitting source if it is in the MIB, the flow identifier otherwise.

If a management connection is lost, the Controller tries to re-establish it, repeating the attempt every 10 seconds if unsuccessful. The information in the “sources” and “destinations” windows for that unit is frozen at the state it had when the connection was lost, and the header line changes to red or magenta. This happens immediately if a connection within a Flexilink island is lost, but there is a delay of several seconds if the lost connection is over an Ethernet or IP network. Right click on the header line (in either the “sources” or the “destinations” window) to tell the Controller to give up trying to connect to the unit and forget the information it was holding about it. In the current version, if you have replied “yes to all” or “no to all” to the “update software” dialogue (for any unit), right clicking on any header line also makes the Controller forget your reply, and ask again if it discovers another unit which is not running the software versions in the product file.

Left click on a unit's header line to select it in the lefthand window. Typing a space in the lefthand window switches it between the “console” text display and a display of the unit's status, including all the MIB objects the program reads and a dump of the messages (green for incoming, black for outgoing, red for errors, yellow background for connection management messages relating to the connection between the unit and the controller program). Typing anything else switches to the “console” display and sends the character to the unit.

Console commands are all single characters; any that aren't recognised (including '?') output a list of commands. There may be other commands that don't appear in the list. The ones most likely to be interesting are:

d - outputs the table of Flexilink flows

D - lists IP addresses etc

i - outputs table of internal routing

p - shows what has been detected at each input, including the state of each network port; for ports using the Ethernet MAC it also shows the state of each virtual link connected through the port and a dump of the filtering engine's code

v - shows the state of the process of adjusting the audio clock; “adj” shows how often it is monitored (in milliseconds), larger values indicating it is more stable; the first number after the colon shows what it syncs to as: -1 = nothing, 0 = word clock, 3 = network time, 4-7 AES3 inputs, >500 = incoming audio flow on the network

Some of the others can stop various parts of the system working, e.g. stop an input decoding the incoming signal by changing parameters. 'Q' crashes the software so that external debug software can access the memory; on a crash it's supposed to save some diagnostics in flash and restart, but that isn't implemented yet.

Left click on a debug server in the Sources window opens it in the lefthand window; again, ‘?’ produces a list of available commands.

Configuration files

Software releases

A software release consists of the product file and binary images of the FPGA logic and the code for the soft processor which looks after protocol processing and configuring the routing tables.

Configuration to be uploaded by the Controller

This file specifies MIB objects to be set by the Controller in specific units. There are two kinds of action that can be requested: configuration to be applied when the management session is connected, and connection of a virtual link on a specific network port when it comes up in Ethernet mode. This is mostly intended as a stopgap until “sticky” configuration (held in the unit's flash memory) is implemented. The Controller reads this file when it starts up, and stores the information internally.

The filename is `dynamic_config_9t.txt`, and it contains “selector” lines which set the context and “VarBind” lines which specify what is to be uploaded. The term VarBind is borrowed from SNMP in which it associates a MIB object (identified by an object identifier or “OID”) with a value.

Lines that don't fit the formats below are ignored, likewise anything that appears before the first unit selector line. If a line begins with text that matches the format it won't be ignored but may be misinterpreted, e.g. in a VarBind line a decimal value followed by extra text that contains a '-' may be interpreted as being in hex. A *unit selector* line has the form

`unit <unit_id>`

where <*unit_id*> is the unit's identifier, which may be represented as either a hex number consisting of exactly 16 digits and at least one '-' (but no spaces) or a decimal number; in the latter case it selects an Aubergine unit, so for example `15` is the same as `0090A899-0000000F`. The *selected unit* is the unit identified in the most recent unit selector line. A *link selector* line has the form

`link <port_number>`

where <*port_number*> is a decimal number which is the physical port's number on the front panel. If the most recent selector line was a link selector, the *selected port* is the indicated port number within the selected unit; if it was a unit selector there is no selected port.

A VarBind line when there is no selected port applies to the selected unit; it specifies a value to be set in the unit's MIB when the management session is connected, and has the form

<*oid*> = <*value*>

where <*oid*> is in dotted-decimal form and <*value*> may be another OID (which will always include at least one dot, because all OIDs have at least two arcs), or a character string in double quotes, or a decimal number. (Other formats may be defined later.) For each such line, the Controller issues a request to set the MIB object to the indicated value as soon as it has established the management connection; the object will have the new value until either it is changed again or the unit is reset. The requests are issued in the order in which they appear in the file.

The following objects are the ones most likely to be useful:

1.0.62379.2.1.1.1.1.5.*n* where *n* is an audio port number (7, 8, or 14 to 21) is the port's name, for example 1.0.62379.2.1.1.1.1.5.14 = “Studio 2 presenter's microphone”

1.0.62379.2.1.1.1.1.6.*n* where *n* is an audio output port number (7, or 18 to 21) is the port's “importance” (see “LED indications” below)

A VarBind line when there is a selected port applies to a virtual link; it specifies a value to be set in the virtual links table in the selected unit's MIB, and has the form

<*arc*> = <*value*>

where <*arc*> is a decimal number which is the column number in the table of virtual links and <*value*> is either a decimal number or an octet string coded in hex (with an even number of digits) and including at least one '-'. For instance, a line setting a value for column 9 might be coded as `9 = 0090A899-0000000F`.

When the Controller discovers that a port for which there is at least one link selector line has transitioned into a state in which virtual links are supported, it issues requests to connect an internal virtual link as described by the VarBind lines that apply to that port. If there is more than one link selector line for the port, it will issue a set of requests to connect each of them, in the order in which they appear in the file.

The following columns may be used; column 3 must be present for each link:

3: encapsulation: the Ethernet and (optionally) IP and UDP headers to be used on transmitted AES51 packets, as an octet string (in hex), see below for details

4: retry interval: if this is set to a nonzero value *r*, any time the link is not up the unit will make an attempt to connect it every *r* seconds; if set zero or not set, no attempt to re-establish is made

9: link partner's identifier: must be in the format with 16 hex digits and at least one '-'

For a layer 2 connection, the encapsulation must be an Ethernet MAC header (which may include one or more VLAN tags). The destination address (the first 12 digits) must be either the MAC address of the link partner (i.e. of the port that is to be at the other end of the link) or the broadcast address (coded as all-Fs); in the latter case column 9 must also be present. The source MAC address must be coded as all-zero. There may be one or more VLAN tags, and the Ethertype must be the last four digits, and must be coded as `88DD`. The format with the data length followed by a SNAP header is not currently supported.

For a layer 3 connection, the encapsulation must consist of a MAC header, an IPv4 header, and a UDP header. The MAC header is as in the layer 2 case, except that the Ethertype must be `0800`. In the IP header, the length, datagram identifier, checksum, and source address must all be zero and the “do not fragment” bit must be set; the destination address must be either the IP address of the link partner or a broadcast address, and in the latter case column 9 must also be present. The UDP header must be coded as `88DD-88DD-0000-0000`.

Network port details

The main states that a network port can be in are as follows:

Reset (or “down”): no network connection

Active: Flexilink physical link

Outwards: has acquired own IP address and a gateway IP address via DHCP

Inwards: connected to a “client” (see below)

Local and Passive: other states that use the Ethernet MAC

Other states occur transiently, during the process of configuring a Flexilink physical link; this normally completes within a second or two of the link coming up.

If no reply is received to the request to switch to the Flexilink MAC layer, nor to the DHCP request, and Ethernet packets are only seen from one source, the port enters Inwards state, on the assumption that a PC or similar device, which we refer to as the “**client**”, has been plugged into it. If there is a port in the same island that is in Outwards state, an IT flow is connected to it (or to the nearest of them, if there are several) and is identified as a tunnel carrying Ethernet packets. The port uses the same IP address as the Outwards port at the other end of the tunnel; its only use of this address is for communication with the client.

The Aubergine's Ethernet MAC includes a “**filtering engine**” which examines the headers of incoming packets and assigns them to an IT flow. This is similar to the way SDN works, though it doesn't currently include all the facilities of OpenFlow. For an Inwards port, packets addressed to the port's own MAC address, and also multicast packets, are routed to the unit's processor, all

others are routed into the tunnel. In the case of multicast packets, they are processed locally and also copied to the tunnel; this means they are liable to be overtaken by unicast packets, but that doesn't seem to cause any problems in practice. Similarly, the filtering engine at the Outwards port routes unicast packets for the client's address into the tunnel, and multicast packets to its processor, which copies them to all tunnels connected to the port. Thus the Flexilink island looks to the rest of the system like a layer 2 switch.

Virtual links

A virtual link carries Flexilink packets over an IT network. There are two kinds:

internal: a connection between two Flexilink islands

external: connecting an application such as the Controller into a Flexilink cloud

Internal links carry both AV and IT flows, as well as packets that convey information about the alignment of frames between the islands.

External links are asymmetrical, with the external device being referred to as the “client” and the other as the “switch”. They carry IT flows, and may also carry AV flows towards the client if the client supports it.

See “configuration” above for details of the encapsulation on the IT network. There can only ever be one link between a pair of ports using layer 2 encapsulation, though there may be multiple links between different pairs of ports of the same two units, whereas with UDP there can be several separate links between a pair of ports. External links will normally use UDP, so that each client process running in a computer can have its own link; internal links should use Ethernet encapsulation where possible, to reduce overheads.

LED indications

The LEDs on ports 1 to 4 indicate their state; the lefthand LED shows the major state and how the righthand LED should be interpreted, as detailed below.

Flows have an “importance” value in the range 1 to 255 associated with them; this was originally provided for use in broadcast studios, where “on air” signals have the highest importance values, those that are a fader away from being on air the next highest, and so on. Audio and video outputs are assigned an importance (see “configuration” above), for instance one connected to a transmitter will be given a high importance. On each link across which a flow passes, the flow's importance is the largest of the importances of the outputs that are downstream of the link.

The colours are chosen such that both LEDs are red if the port carries a signal that is on air; more generally, a red light indicates a cable that should not be unplugged. There are also some error conditions that are indicated by the righthand LED flashing.

There are some indications that appear on all four ports during start-up and if the software has crashed. All the LEDs flash red and green together if the logic is in reset, red and yellow if the soft processor is halted. Note that if a port is not connected (or is connected to something that is powered off) both LEDs will be off, so on switching on a unit that doesn’t have any of ports 1 to 4 connected the LEDs will all flash briefly and then turn off.

States of the lefthand LED are:

Off: standard Ethernet, no other features, no IP address

The righthand LED is coded as: off = link is down; yellow = connected at 1Gb/s, green = connected at a slower speed.

Green: standard Ethernet, Inwards or Outwards state, no virtual links

The righthand LED is yellow if the port is in Outwards state (see “Network port details” above). If it is in Inwards state, the righthand LED is: green if incoming packets are tunnelled to an Outwards port (in the same unit or elsewhere on the network), in which case it uses the Outwards port's IP address; else flashing yellow if there is no tunnel but it has an IP address; else flashing green. In standard Ethernet states other than Inwards and Outwards, the lefthand LED is off.

Flashing yellow: transitioning to Flexilink physical link

The colour of the righthand LED shows different stages of the process.

Yellow: Flexilink, no AV flows

The righthand LED is red for a physical link; otherwise the port is in standard Ethernet mode and carries one or more virtual links, none of which carries any AV flows, and the righthand LED shows the port's state as yellow = Outwards, green = Inwards, off = other.

Red: carrying at least one Flexilink AV flow

This may apply to either a physical link or an Ethernet port carrying at least one virtual link. The righthand LED shows the highest importance of an AV flow that traverses the link as: off = 1 to 63, green = 64 to 127, yellow = 128 to 191, red = 192 to 255.