Digital Control Systems in CADAC Consoles

1) The System.
   a) J-type
   b) F/M-type
   c) Concert
   d) The IBM PC
2) The Central Control Module.
   a) Main Processor
   b) Communications Processor
   c) Concert's CAM module
3) Simple Faders.
   a) VCA Control
   b) Logic control
   c) Communications.
      i) J-type
      ii) F/M-type
4) Motorised Faders
   a) Micro-processor
   b) VCA Control
   c) Servo
   d) Communications
5) The Module Bus
   a) J-type Programmable Routing Modules
      i) Mk I type 1
      ii) Mk I type 2
      iii) Mk II
   b) J-type Programmable Group Module
   c) J-type Programmable Channel Module
   d) M-type matrix module.
6) The Global Bus
7) The Fast Copper Comms card
   a) The communications link to the console
   b) The control link to the IBM PC
   c) The data link (DMA) to the IBM PC
8) Optical comms cards

1. The System

The purpose of the digital control system in CADAC consoles is to transfer data to and from the various modules in the console. This data represents the state of switches, the positions of faders and potentiometers, and various other items. An ideal control system would transfer data instantly from one module to another but practical circuitry has to be a compromise between required speed and "un-required" cost. As our needs have changed over the years various communications systems have been devised and implemented and these will be examined.

1.a The J-type system

The J-type console marked a clear change from previous CADAC control systems, which had required a considerable amount of hand wiring. This dramatically increased the cost of the console and made it difficult to build them quickly or re-configure the layout of the modules. J-type was devised to use ribbon bussing throughout and to allow modules to be dropped into the frame(s) anywhere.
To allow ribbon bussing to work for data communications, we require that each unit have an "address" that messages may be "posted" to. Every module receives the message but only that module which matches the address, included within the message, acts upon it. The required address is achieved using a small PCB mounted on connector 'D' of the module cradle[1]. This PCB carries 6 pads, which may be connected to ground in various combinations. When a module is plugged in, pull-up resistors generate high logic levels, except where the pads on the card have been "solder-shorted" to ground. These lines (MA0 to MA5) together with two extra lines in the bus (FA0 and FA1), which are generated by a switch on the rear of the console frame, form an address, which may be read by the digital control hardware. Addresses within a frame start as 1 for the left-most module position and increment to a maximum of 63.[2]
The switch on the rear of the frame allows for up to four frames to be connected together and have different addresses. Because of an accident in the wiring of the first frame we built, the switch positions produce the following addresses;

Frame 1	positions 1 to 63	
Frame 2	positions 129 to 191	(add 128)
Frame 3	positions 65 to 127	(add 64)
Frame 4	positions 193 to 255	(add 192)

Rather than have one rogue frame out there we have continued to wire frame switches in the same way.

These addresses are a vulnerable part of the communications system and we have several checks during the building procedure to make sure they are correct. First the bare frame is checked with a simple test unit which plugs into the cradle's 'D' connector, or the end of the fader ribbon, to show the address.[3] Because shorting pins to ground sets the address, it should be clear that a copper etch, or solder splash fault on a module could change the apparent address when the module is plugged into one of the cradles. Worse, it could change it in one frame position but not in another (if the address card already shorts the pin to ground then the fault will not be seen). We therefore check the modules on the bench at enough addresses to make sure all the module address lines can be either high or low.[4] Even modules that are not themselves controlled by the system could have shorts on these pins so they must also be checked.

[1] The module cradle carries four connectors, labelled A,B,C,D respectively, from the back of the console.
[2] We have made several 64 way frames. In these cases we *always* leave the right-most module position without an address card and tell customers that one of the non-automated modules must be fitted here.
[3] This unit shows the addresses in ***hexadecimal***. If you are unfamiliar with this numbering scheme please check with a computer engineer for details. Hexadecimal numbers are written as 0x.. to distinguish them from normal decimal numbers.
[4] This really only requires the two hexadecimal addresses 0x55 and 0xAA.

J-type and Concert are complicated further by the fact that the address setting pins are "seen" by both the module and the fader so either could generate an address fault, affecting both units.

1.b The F/M-type system

The F/M-type is a derivative of the J-type system and so most of the bus structure developed for J-type is also in the F-type frame. It has been "condensed" onto one 64-way connector and there are a few new signals but it should all be obvious if you are familiar with the J-type.
There are some controls "switches" in the F-type that are turned on or off in all modules simultaneously. These switches are controlled by the Global Bus (G-Bus).

1.c The Concert system

It was with Concert that we introduced the new high speed RS485[5] busses for communications. Two busses appear on a 10-way ribbon that connects to all of the faders. One bus provides the timeline, which is a 100 Hz pulse waveform used to synchronise the actions of different modules. The other bus is the high-speed communications needed to keep motorised faders moving smoothly. A further two RS485 busses appear on the module connector providing similar facilities for the control of the modules.

1.d The IBM PC

The IBM PC provides the user interface and mass storage facilities for the automation system. The first J-type systems communicated using the PC serial port (COM 1) at 19200 baud but as modules became more complex, requiring larger amounts of data, we developed the MK II CCM for J-type. This was a derivative of the Concert CCM and so inherited all its ability to control motorised faders. Initially these CCM's used fibre optic cables and communicated at 1 Mbit/second. The optical system has been replaced with the Fast Copper Comms system, which is identical apart from the cable and the devices that drive that cable.

[5] RS485 is a two-wire balanced standard with signal levels of 0 to +5 volts. Driver devices are such that even when powered down they present a 12kohm load to the bus and are therefore ideal for our use. Most logic devices have outputs which, when their power rail goes away, appear as a low impedance to ground thus preventing other devices on the same bus from communicating.

2. The Central Control Module

The CCM is the main point of communication between the console and the IBM PC. It receives data in blocks from the PC and distributes that data to individual modules via the various communications busses designed into it. It also collects data from the individual modules and sends it as a block to the IBM PC when requested to do so. The front panel of the CCM provides a secondary user interface, giving the person at the console most of the facilities of the automation system. The following descriptions are true for J-type Mk II[6] and F/M-type CCMs, the Concert system is slightly different and will be examined in a different document.

2.a The Main Processor

The main processor is on PCB 7249 and controls the front panel buttons and displays, communicates with the IBM PC and drives simple (non-motorised) faders and the Module Bus. The processor may be link selected to run at 11.0592 MHz for old style serial communications to the IBM PC, or at 16 MHz for the newer "fast copper" communications. The change in crystal frequency is detected by software at start-up and the correct communications system selected. The 80C152 microprocessor (µP) is the "brains" of the board and access EPROM, RAM and peripheral devices using a 16-bit address bus and an 8-bit data bus. The µP multiplexes the 8 lower address lines with the 8 data lines and these are extracted (de-multiplexed) by U2. General purpose RAM is provided by U3 (32K bytes) and the program is stored in EPROM U4 (64K bytes).
128K bytes of battery backed memory is provided by U8. This memory is accessed as pages, each 8K bytes in size using a "paging" register U15. The first 10 pages of BB memory are used as Cue Memory stores accessed as 0 to 9 via the keypad controls (see the operations manual). The remaining 6 pages are used for internal requirements.
U16 is a serial communications device specifically configured to send and receive MIDI information. MIDI is too large a topic to cover in detail here but suffice it to say it is a way of sending control messages to other pieces of hardware in the audio system. MIDI runs at 31250 baud and U16 is clocked from the 16 MHz crystal clock, even if the µP is running at 11.0592 MHz. It is this feature which allows the software to work out which clock the µP is running from as it can compare the down-counters in U16 and in the µP to see if one is slower than the other.

The remaining peripheral devices are on the motherboard.

2.b Communications Processor

The communications processor (PCB 7173) has the same basic µP circuit of U1, U2, U3 and U4. It has one main job, to exchange data with up to 128 motorised faders. It is connected to the main processor via two FIFO (first-in first-out) memory devices (U8 and U9) and two command/status latches (U7 and U10). When the main processor receives a block of data from the IBM PC that is intended for the motor faders, it transfers the data into the relevant FIFO device (U9). The main µP then writes a command byte into U7. The action of writing to U7 generates an interrupt to the communications µP (U1) telling it to read the data from the FIFO. A similar process works in the other direction through U8 and U10) to allow the comms µP to send blocks of data back to the main µP.

When the comms processor receives the interrupt it unloads data from the FIFO and then transmits the relevant bytes to each motor fader in turn. There are four "fader" busses provided by this PCB, each consisting of a copy of the RS485 timeline and an RS485 communications bus running at 500K baud. Bus 1 is used for communications within the frame that the CCM is in, busses 2,3 and 4 are available on 9 pin 'D' connectors for use with extension frames. For each fader in turn the µP enables one of the bus drivers and transmits an address. If there is a fader at the specified address it will respond by transmitting 3 bytes of data, the µP then ends the transaction by sending the fader 3 bytes of new data. In this way the µP distributes the data originally sent by the IBM PC, and collects data from all the faders into a block to send back to it.

[6] The Mk I CCM for J-type should now be considered obsolete. If you require detailed information about its operation please consult R&D.

2.c Concert's CAM module

The Central Assignment Module (CAM) in Concert is similar to any of the CCMs but it deals specifically with the module switches and potentiometer recall data. Concert is the only console that has RS485 communications for the modules as well as for the faders. The CAM has it's own main processor board (PCB 7224) to control the front panel buttons and LEDs, communicate with the IBM PC and other "house-keeping" chores. Another 7173 communications card is used, running different software, to control the communication with the modules.

3 Simple Faders

The "simple" fader (non-motorised) is best represented by PCB6962, this contains all the hardware that has then been developed further to provide the faders for F/M-type. Each fader is connected to its corresponding channel module via a 20-way ribbon cable providing power and frame address. The fader drives its signals back up this cable to control the module. A 26 way ribbon bus (the Fader Bus) connects along all the faders, providing simple serial communications, overall control wires and DC Master busses.

3.a VCA Control

The VCA Control circuit is taken directly from the E-type faders. A 10-volt reference is generated by D3/D4 and applied across the Fader (an audio taper track). U13A buffers the voltage from the fader wiper and then U13B and C and U11 form an anti-log amplifier. This changes the audio taper law into a voltage scale, which may be applied, to a volts/dB VCA. U12A and B form two separate VCA drive signals buffer by U13D and the output from U13C. U13D also adds the contribution from the selected DC Master bus and the Mute control circuit.
U3 is a counter that may be clocked up or down between 0 and F to select one of the DC Master busses. This selection is done with U5, adding the bus voltage to U13D as described above. The four outputs from the counter U3 are optically isolated to drive U5 preventing any digital noise contamination between the digital and analog parts of the system.

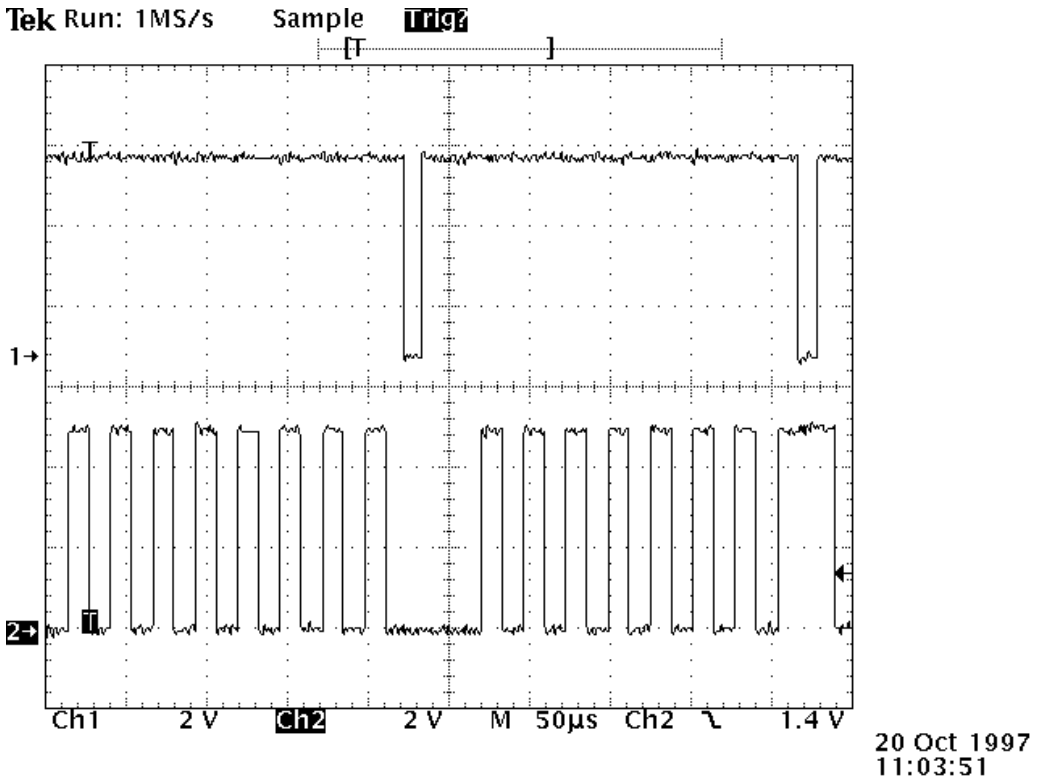
3.b Logic control

The logic control is simply a few flip-flops that may be toggled by a push button. "Mute" and "Aux from VCA" may be set or cleared from the communications system, while Isolate and Bypass may be set or cleared from control lines that are part of the Fader Bus.

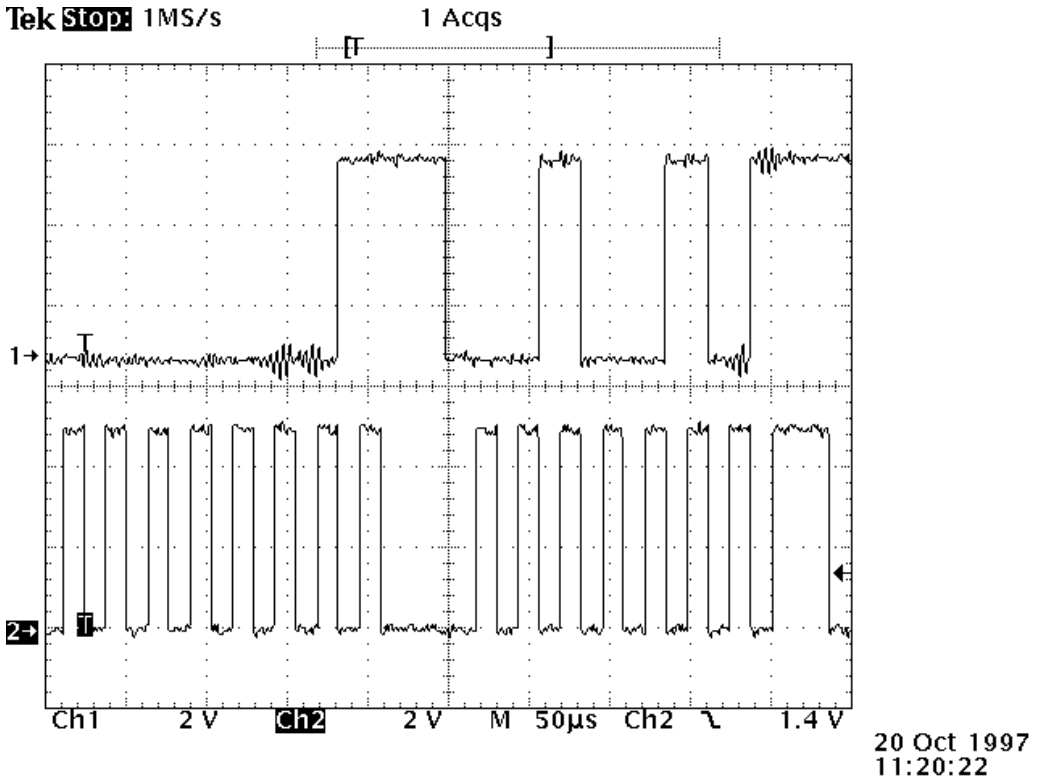
3.c Communications

i)

Communications are provided on a simple 4 wire serial interface, the signals being Clock, Strobe, "Data to Fader" (DTF) and "Data From Fader" (DFF). The Clock and Strobe lines have RC filters on their inputs to guard against reflection noise generating false edges, the waveforms are then "squared-up" again by U14. The data lines are not edge sensitive and so do not require these filters. All four wires are fed through relay contacts that open if the fader is isolated or if the power is removed, thus protecting the bus wires from a rogue or un-powered fader.



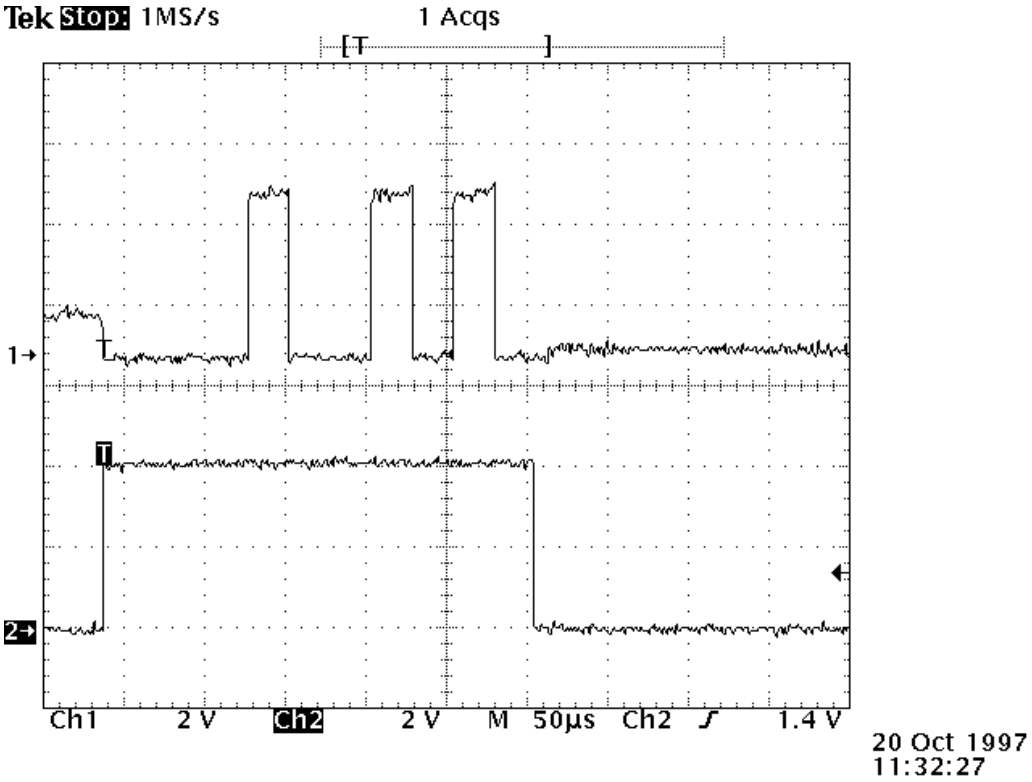
The picture above shows the Clock (lower) and Strobe (upper) lines as seen in a channel fader on P5025 pins 8 and 9. Note the first strobe pulse occurs while the clock line is low, this is an ***Address Strobe***. The second strobe occurs while the clock line is high and this is a ***Data Strobe***.



This picture shows the Data To Fader (DTF) waveform in comparison to the Clock. Note the first 8 clock pulses shift an address into the fader (in this case 00000001) and then the second 8 clock pulses

shift the data in (00100101) Note the data is “Most Significant” bit first and is "read" on the rising edge of the clock.

The data to the fader is shifted one bit at a time into the P5027-2 chip, which also contains two four-bit comparators. When the data in the shift register is the same as that presented on the MA0-7 lines then the two outputs EQU1 and EQU2 will both be low. These two levels together with an Address Strobe (as described above) cause a flip-flop in P5025 to be set generating a high logic level on output DEN. This output in turn enables the tri-state driver in P5026 allowing it to drive data back down the Data From Fader (DFF) wire during the second group of 8 clock pulses. These clock pulses have also shifted new data into the P5027-2 shift register and a Data Strobe together with the high level on DEN generate a clock pulse which writes the data into the counter (U3), the Mute flip-flop and the Aux from VCA flip-flop. The data that the fader sends out, comes from the shift register U6.



The picture above shows the fader emitting data (P5026 pin 19) referenced to the DEN signal (P5025 pin 14). Note the upper waveform shown at a non-logic voltage as the scan starts, this is characteristic of a High Impedance bus that is not being driven by anything. As soon as DEN turns on the bus assumes a correct voltage.

ii)

The F/M-type system used the same method but the number of data bits have been extended to 16.

4. Motorised Faders

The motorised fader were first developed for Concert, but development of the Mk II CCM for J-type along side the Concert CCM allowed J-type to immediately include the same faders for it’s own use. The motor fader is best demonstrated with PCB730 and PCB7358, other motor faders exist but the technology is much the same as these.

4.a Micro Processor

The motorised fader uses a microprocessor (µP) to perform the complex mathematics required for closed-loop servo control and to achieve the high-speed communications required for smooth movement. The basic circuit is similar to that which we have seen in the CCM, but we use a different member of the C51 family (the 80C51FA). This µP is a lower cost device (important when we put 10's or even 100's of them in one console) and has a slightly different serial interface peripheral for communications (see 4.d for details). Much of the general logic provides output latches (74HC573), for driving LEDs or switching devices, and input buffers (74HC541) for reading switches and status.

4.b VCA control

The VCA control circuit exists on the "meter" board but is very much the same as that in a non-motorised fader, although there is some extra complication around U3B, C and D and U5A. In a non-motorised fader the contribution from a selected DC Master bus is ***always*** added to the output that drives VCA1. In a motorised fader we have a facility called **"Group Link"** where the channel fader will physically move up and down to follow the position of the DC Master fader. If this is the case then we have to switch off the voltage contribution to the VCA drives using U5A. If we didn't then the DC Master fader would affect the VCA twice, once through the voltage drive and once by moving the channel fader. U3D and U6A set the DC Master bus voltage to a range that can be read by an Analog to Digital Converter (ADC) so that the µP can do the necessary **"Group Link"** calculations.

4.c Servo

Much of the servo loop in the motorised fader is performed by the µP. This is quite a step away from many servo systems which have highly tuned amplifiers to achieve good speed and positional response. These amplifiers however require a lot of components and careful set-up procedures to work well. The beauty of a digital (software) control loop is that it uses few components and once the software is tuned to the mechanics of the system they work without any set-up procedure.
The µP reads the position of the fader using an ADC (U15) and a special servo track in the fader. It performs calculations based on where the fader is and where the computer says it should be to decide if the fader should be moving and if so how fast it should be going. A voltage is generated by Digital to Analog Converter (U16) which is amplified by a power op-amp on the meter board (U11 on PCB7358) in order to drive the motor on the fader. The fader speed is approximately proportional to motor drive voltage and so as the fader gets closer to the point it should be at, the speed is reduced so that we avoid "overshooting".

4.d Communications

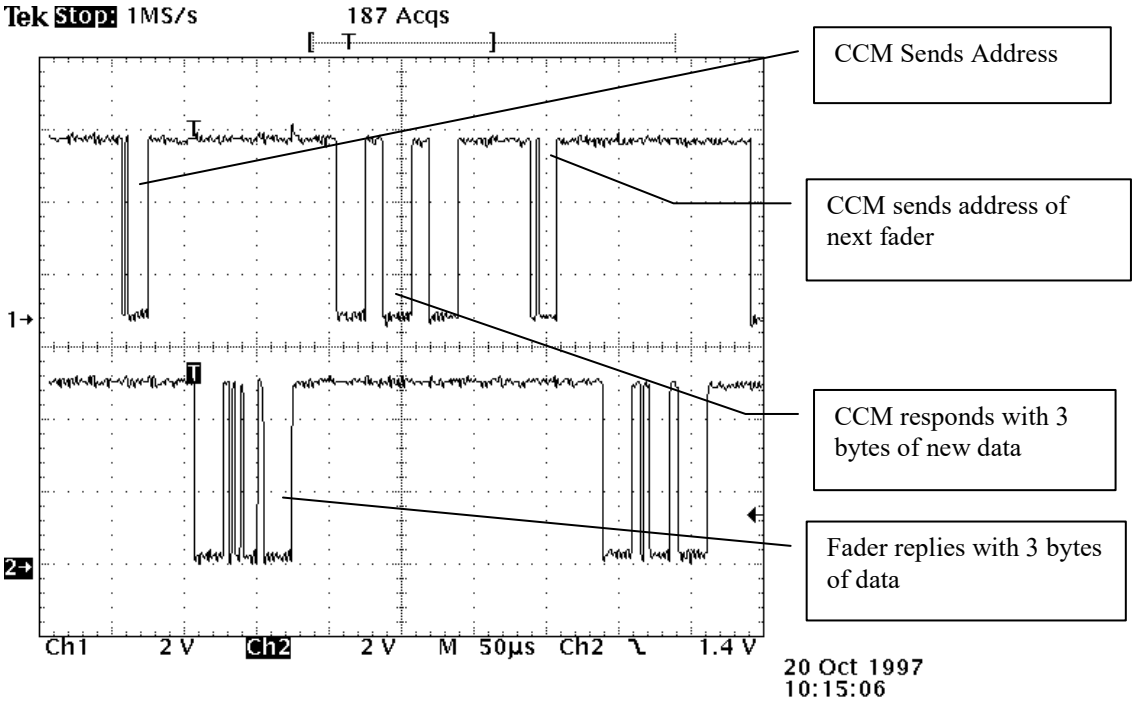
Motor faders require a much higher speed communications than the non-motorised variety, this is because we must give a motorised fader a new positional value at least 25 times per second to maintain smooth motion. We designed the system to have a maximum of 128 faders so the system must be able to send messages to all 128 faders, 25 times per second. The communications processor in the CCM has been described in 2.b). The receiving end of this communications is the motorised fader. Communications runs at 500K baud and is very tightly controlled by the µP programs to maintain this rate. The communications bus is bi-directional, which is another reason for the tight software control. What we would "see" on the bus is as follows;

1) The CCM comms processor sends out an address byte. This is an 8 bit number with a 9th data bit appended which is set to '1'
2) If there is a fader which matches this address[7] it will reply by sending 3 bytes of data, these bytes will have the 9th bit set to '0' to mark them as data. The fader has approximately 150 uS to begin

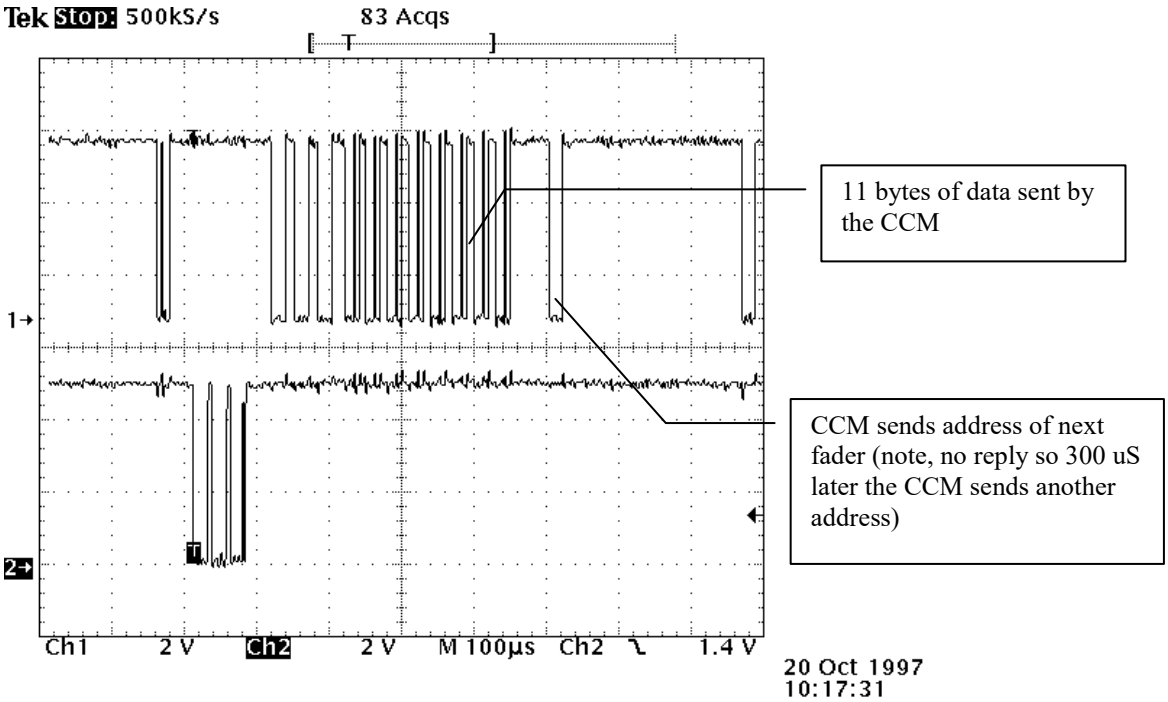
[7] The address is set by the Module Cradle Address card, described in section 1.a

this reply or the CCM will assume the fader does not exist. You should note that even if the fader is isolated it will reply, with a data bit set to indicate it's isolated state. If the CCM sees no reply then it will set a bit within it's own memory store to say the fader was isolated.

3) If the CCM receives a reply within the correct period it then sends 3 bytes of new data to the fader, again these bytes have there 9th bit set to '0' to mark them as data. This 9th bit set to '0' stops any other fader from thinking they may be address bytes and responding to them. If the fader is a DC Master then it is given 11 bytes of data (the extra 8 being the alphanumeric display data).



The picture above shows the waveforms seen on PCB7173, U14 pins 1 and 4. These are the individual transmit and receive lines before being combined into the bi-directional bus by U14 (on pins 6 and 7). The data shown is that of a channel fader (3 bytes sent and received).



This drawing shows similar data transfer, but for a DC Master fader. Note the larger block of data sent to the fader by the CCM.

5) The Module Bus

Some of the modules for the J-type system required more than 1 byte of data and the speed penalty of using a serial bus would have been too great so a parallel bus system was devised for the modules. We use the same addresses in the frames, giving us 256 different module addresses. The data is read and written in bytes requiring an 8-bit bi-directional data bus. Each module may have 16 byte locations within it to read or write data from/to, therefore we require a 12-bit address bus to uniquely access each location of each module position. There are also four control lines whose use will be described as we progress. It was our intention to design a number of different types of module which would all work on the same bus so we defined some general principals for all modules.

1) a module is selected by the 8 upper address lines (BA4 to BA11) being equal to the cradle address that the module is sited in, and by the SEL line being low. The direction of data is controlled by the R/N_W line, which is high if the CCM is reading data from the module and low if the CCM is writing data to the module.
2) The lower 4 address lines (BA0 to BA3) select one of 16 locations (referenced as internal address 0 to F) within the module to be read or written to.
3) The module must present a code number (Module Identity) in the lower 4 bits of location F to indicate what kind of module it is.
4) A module may use the CON1 line to request attention from the CCM. If it does then it must echo this with a flag in bit 7 of location F that is high as long as the module is holding the CON1 line low.
5) If the module is holding CON1 low then this must be reset by the CCM writing to location F
6) If the module has a SEL LED on its front panel then this is turned on and off by the CCM writing bit 7 to location `E`. The other bits of location `E` are undefined for all existing module types and are always written as 0 by the J-type CCM software.

5.a.i) J-type Programmable Routing Module

The first J-type console had PRM Mk I (type 1) modules designed for it. These modules were really just a big switching matrix where two audio inputs could be connected to any or all of the Sub-Group or Matrix Group busses. For digital control this repesented 96 bits of data (12 bytes) each bit switching one audio "route". The 12 bytes of data appeared in locations 0x00 – 0x0B. Locations C and D were unused and locations E and F where used for module mapping purposes. This unit is shown in schematic 6991, there are none of these units still in the field so we will progress to their replacement.

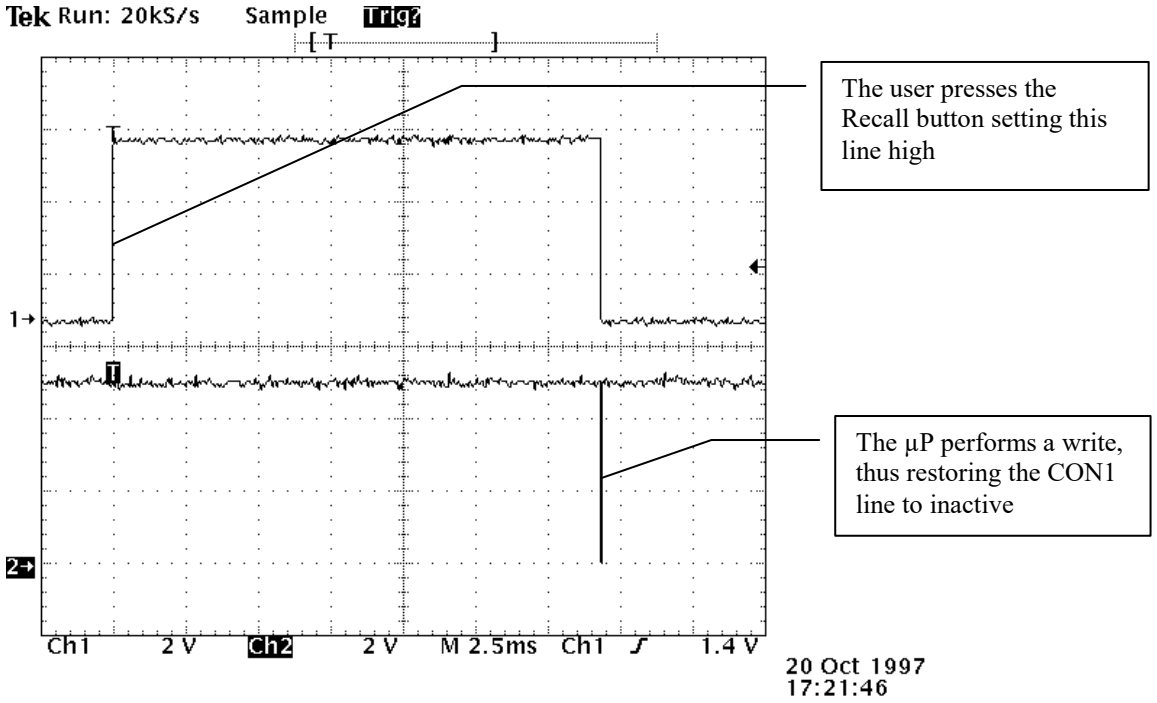
ii) Mk I PRM (type 2)

This unit is Schematic 7062 and is the current Mk I PRM. The logic circuit controls the same 96 audio route switches and there are only a few minor differences. The 12 bytes of data appear at locations 0 – 5 and 8 – 0x0D. This change made the logic for the two separate Isolate switches easier to implement. U4 generates specific strobe signals (BLK0 to BLK13) which are used to write data into the 8-bit latch on the relevant relay control card (PCB7024).

When location F is read be the CCM then U7 is enabled. This returns a Module Identy of '1' as required under rule 3 above. It also returns the CON1 repeat flag in bit 7 as required by rule 4. The button marked ***recall*** is used to set the flip-flop U3A which pulls down the CON1 line through D2. This CON1 line is bussed through the console to the CCM where is tells the main µP that a module required attention. At the same time the flip-flop state may be read by the CCM through location F, in this way the CCM finds the module that requires attention. If the CCM writes to location F (any data) then the BLK15 strobe from U4 resets the flip-flop allowing the CON1 line to go high.

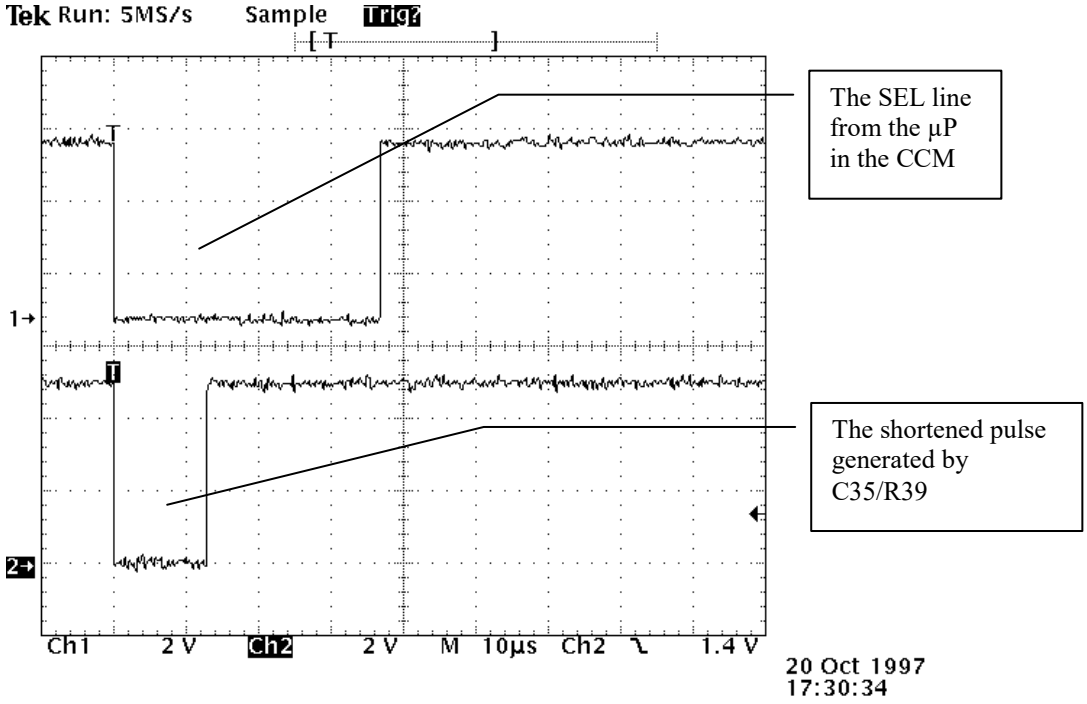
This method of calling attention was devised so that we could have a multiple of Slave PRMs and a Master PRM which would be equiped with all the buttons to "toggle" the audio routes on or off. A slave would be recalled to the Master using its Recall button and then pressing buttons on the master

would cause the corresponding audio "route" to change in the selected Slave. The selected Slave would be indicated by lighting a Recall LED. The LED is controlled by writing to bit 7 of location E as specified in rule 6. The ***Master Routing Module*** has never been implemented so the Recall button and LED are only used to map the units.



The picture above shows U3 pin 5, which goes high when the Recall button is pressed, then some tens of milliseconds later the µP performs a write (the narrow pulse on the lower trace) which clears the line back to a logic low.

One point to note – The SEL strobe is used to write data into the latches, it also turns the data buffer, U2, on and off. This causes a problem because the data is written on the rising edge of the strobe, just as the data buffer is being turned off !! It is therefore possible for incorrect data to be written to the latches. To correct this the SEL line is fed via a CR circuit (C35/R39) to generate a shorter write strobe so the data is written before the SEL line is turned off.



iii) Mk II PRM

The Mk II PRM has been a further development of the PRM idea, it is very much the same as other PRMs as far as the digital control is concerned but the audio side is markedly different. It is not within the scope of this document to explore the audio circuitry of this module.
As with all modules the Mk II PRM returns a code number in it's location 15 to identify what kind of module it is, this code is 3.

5.b) J-type Programmable Group Module

The programmable Group is a very simple module consisting of a single byte of data which controls 6 switches. It returns an ID code of 2 in its "location 15". Each switch is a 74HC74 flip flop wich can be toggled by a push-button (de-bounced by resistors, capacitor and one gate of a 74HC14). The flip-flops may also be Set/Reset by a byte of data written to them, the circuit based around each 74HC139 half is just a convenient way of getting two logic functions in one package. The remaining decoding logic is in a programmable logic chip (16V8) apart from the address comparator (74HC688) which is the same as we have seen in all the other "parallel bus" modules so far.

5.c) J-type Programmable Channel Module(s)

These modules, although based upon the same ideas and technology as described above, are quite complex in their implementation. They are therefore described in a separate document.

5.d) J-type Programmable Group Module (type 2)

These modules are described in a separate document.

5.e) J-type Programmable Aux Module

These modules are described in a separate document.

5.f) M-type Matrix Module

6) F/M-type Global Bus

The "Global Bus" is a simple way to provide logic switching that operates simultaneously in all the modules of a console. This is used for facilities like choosing whether an audio bus is stereo or dual mono. Each module has a 16 bit shift register and some control logic, which is driven by the CCM via CLOCK, STROBE and DATA line. The CCM shift 16 bits of data into each module (all modules receiving the same data) and then the STROBE line is used to "latch" the data. This is very similar to the non-motorised faders which we have examined previously. The 16 logic outputs in each module are then used to drive relays etc. to perform the required audio functions.

7) The Fast Copper Comms Card