Core Loading Automaton

A tool I originally developed at the request of Peter Vaughan of The National Museum of Computing, this small modification to an IBM 1130 will allow a PC based file of memory words to be loaded automatically into the core of an 1130.

The device operates by overriding the 16 console entry switches (CES) on the faceplate of the 1053 console printer and by simulating presses of the Program Start button and Load IAR button on the 1130 console.

When the IBM 1130 rotary mode switch is set to LOAD mode, we can enter each data value set on the CES into ascending addresses on each push of Program Start.

Sending a line with # activates the loader. It takes control away from the physical Prog Start button and allows the loader to push the buttons instead. The 1130 rotary mode switch must be set to the LOAD position before this happens. Sending another # will deactivate the loader, then the operator turns the mode switch to RUN.

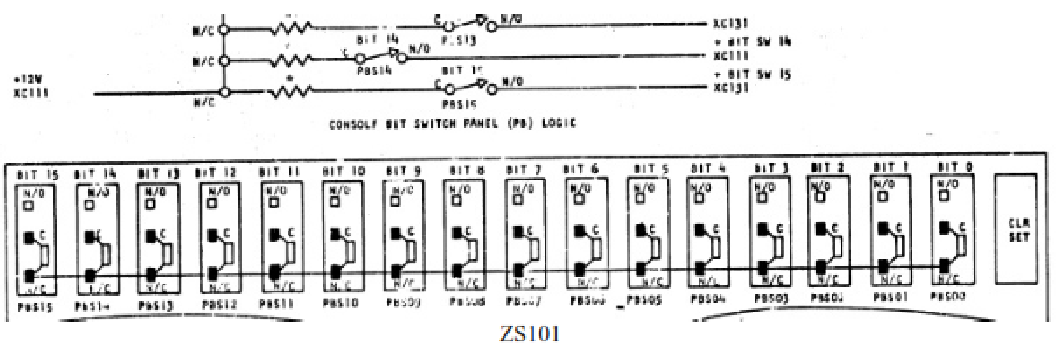
When an address (hex word prefixed by @) is received, it sets up that address in the CES and presses the Load IAR switch virtually. When it receives a hex word without the prefix, it enters it in the CES and pushes the Program Start button.

When an address prefixed by = is received, the code saves that address and then when the session is deactivated, it will put it into the CES and press Load IAR so that the 1130 will fetch instructions from that address when Prog Start is pushed.

If a hex word is prefixed by Z then the hex value is a count for how many words of 0000 should be loaded into contiguous locations in 1130 core.

It communicates over a serial link on a USB cable, interpreting each line of text sent to it as four hex characters with some simple error checking. The line is converted to the 16 values for the CES. The device activates photoisolator switches to activate the CES that have a 1 value, while other photoisolator switches will virtually press the Program Start button.

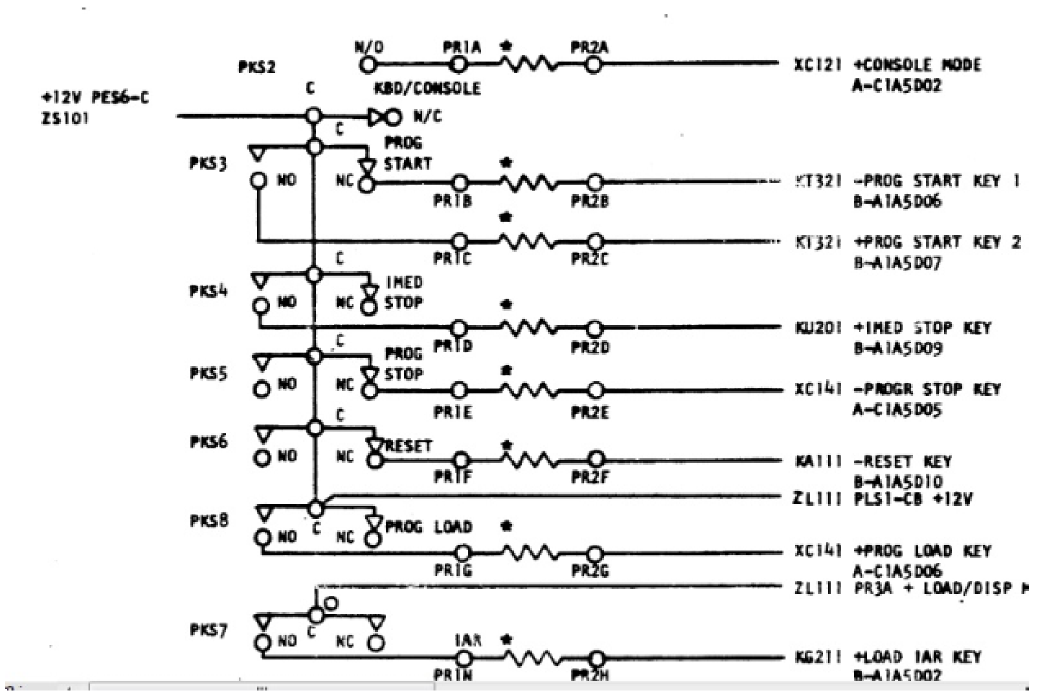
Core Loading Automaton



ZS101

To install the device, we first have to add wires to the Normally Open contacts of each of the 16 CES switches (see ALD ZS101) installed on a 16 pin connector that plugs into the loader device to drive all 16 CES. Another 6 pin connector is wired to deliver +12V power and to make the connections to the buttons on the console.

We disconnect a wire on page ZK111 of the 1130 ALDs that hooks to the Common terminal of switch PKS3 (Program Start) - this is a run of wire carrying +12V to the common terminals of switches PKS2, PKS3, PKS4, PKS5, PKS6 and PKS8, but we break the connection to PKS3 and bridge between PKS2 and PKS4. Then the disconnected common terminal of PKS3 is wired to the 6 pin connector, so that we provide the +12V when the loader is deactivated and remove power to the Prog Start common terminal when we become activated.

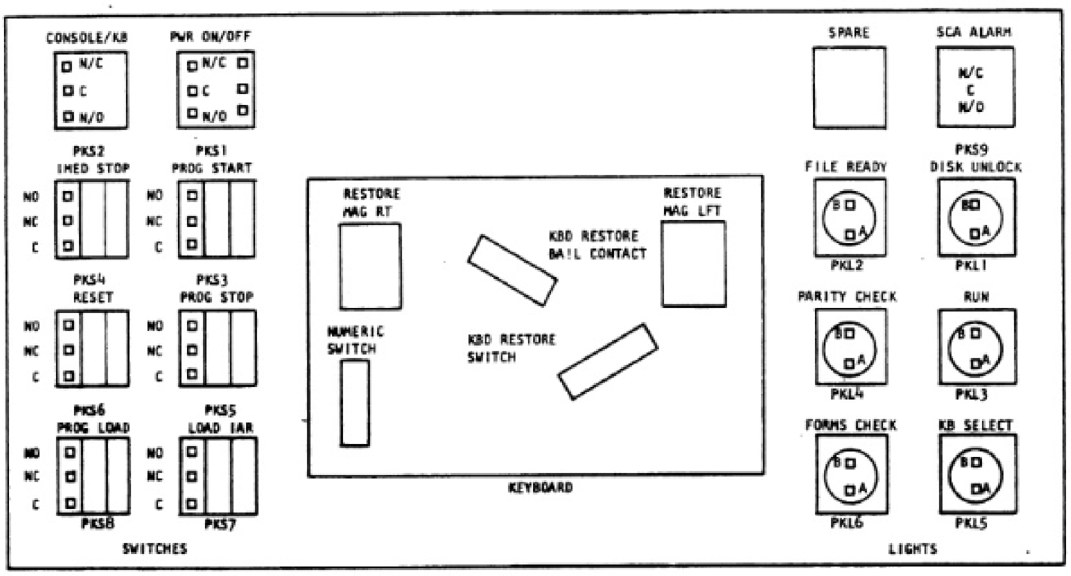


+12V PES6-C ZS101
PKS2
KBD/CONSOLE
PR1A
PR2A
XC121 +CONSOLE MODE A-C1A5D02
PKS3
PROG START
PR1B
PR2B
XT321 -PROG START KEY 1 B-A1A5D06
PR1C
PR2C
XT321 +PROG START KEY 2 B-A1A5D07
PKS4
IMED STOP
PR1D
PR2D
KU201 +IMED STOP KEY B-A1A5D09
PKS5
PROG STOP
PR1E
PR2E
XC141 -PROGR STOP KEY A-C1A5D05
PKS6
RESET
PR1F
PR2F
KA111 -RESET KEY B-A1A5D10
ZL111 PLS1-CB +12V
PKS8
PROG LOAD
PR1G
PR2G
XC141 +PROG LOAD KEY A-C1A5D06
ZL111 PR3A + LOAD/DISP
PKS7
IAR
PR1H
PR2H
KG211 +LOAD IAR KEY B-A1A5D02

Core Loading Automaton

We tap on the +12V line that we just removed and use it to provide power to our loader device.

We run two new wires, one each from the NC and NO contacts of PKS3 to the 6 position connector of our loader. This will act as our virtual Program Start button. Another two wires, from PKS7 NO and C contacts, go to the 6 pin connector to control the LOAD IAR button.



The diagram above shows the underside of the console panel, with our PKS3 Program Start switch clearly shown along with the terminals to which we will connect wires (NO, NC and C) after removing the +12V power wire originally hooked to terminal C.

<u>Software</u>

At startup, the program enables the Program Start physical switch by delivering +12V to its common terminal. Only when we activate the loader using our # command will it remove power rendering the switch inoperable.

The Arduino program loops listening to the serial port (USB link to PC) at 9600-N-8-1 setting. Command characters (#, =, @, or Z) are pulled off the line and handled first. It builds up a line of four characters, if longer it is an error and if shorter the input is ignored. The code skips any NL (x0A) or CR (x0D). It looks for exactly four characters that are valid hexadecimal - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, or F. The letters can be upper or lower case ASCII.

Once it has a good word, it turns on the photoisolator switch for the CES for each switch whose value is 1 from the input we just received. For example, if we recieved the word 3F29 then we want to activate the relays for switches 2, 3, 4, 5, 6, 7, 10, 12 and 15 which correspond to 0x3F29.

After waiting a fraction of a second for the input to the switches to stabilise, we activate the virtual Prog Start button, leaving it pressed for half a second and then releasing it. Once it is released, we go back to give a prompt and accept the next word of entry from the terminal hooked to our USB cable.

To virtually control the Prog Start button one photoisolator switch delivers +12V to the NC contact when the button is not pressed, but when pressing the button virtually we turn off +12V to the NC contact and instead deliver it to the NO contact via another photoisolator. Releasing the button is the reverse, removing +12V from the NO and delivering it to the NC contact.

If the user enters an @ in the line with the four hex characters, it sets the value on the Console Entry Switches and pushes the virtual Load IAR button, with similar timing to how it handles Program Start above. This is accomplished by connecting the wire from the common terminal of Load IAR to the wire from the NO terminal of the button, using another photoisolator switch.

This code is written for an Arduino Mega 2560 board. We need 20 output pins which excludes most of the smaller Arduino boards. The hardware that interfaces this to the IBM 1130 is contained on a shield, a small PCB that is mounted atop the Arduino. This uses the Liteon LTV-847S photoisolator switches and contains 470 ohm resistors for the input of the photoisolators and to activate the CES switch to ground.

The code and other design files is shared on Github as well as to any 1130 restorer wishing to use or modify this device.