

# 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.

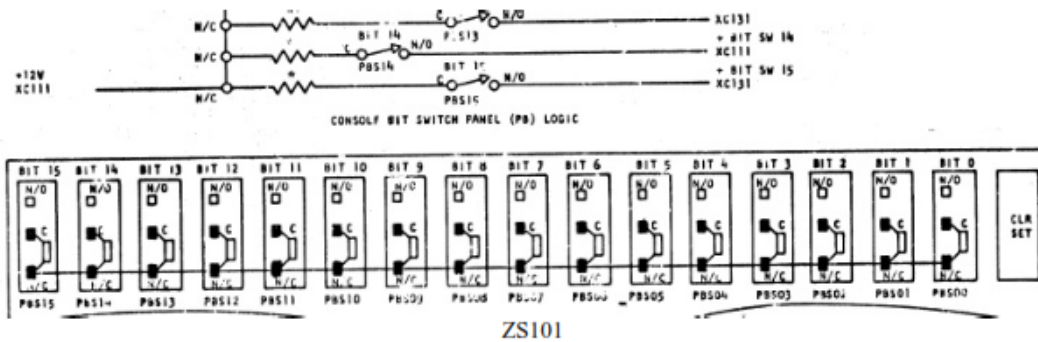
When an address prefixed by = is received, the code saves that address and then when the session is deactivated, it will put it into the CES and press Load IAR so that the 1130 will fetch instructions from that address when Prog Start is pushed.

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.

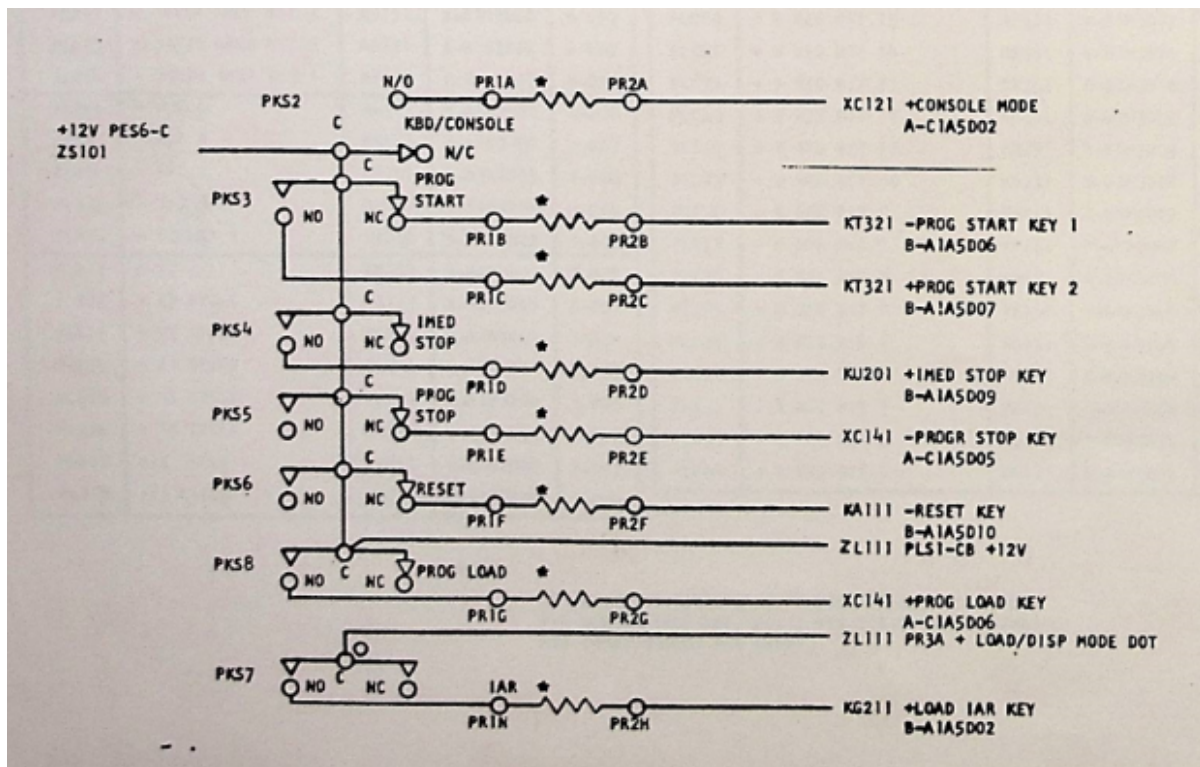
# Core Loading Automaton

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.



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.

The wiring of the Prog Start and Load IAR switches are show on ALD ZK111. Our five wires from the Console Loader are connected to the switches.

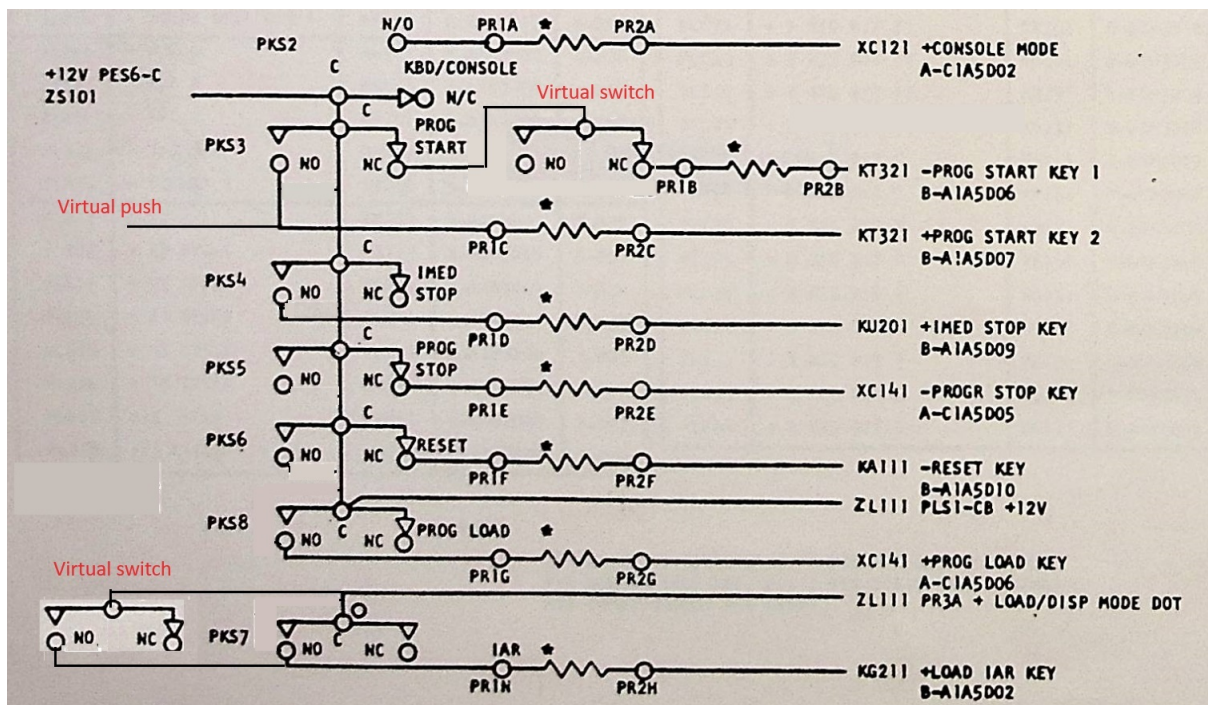


# Core Loading Automaton

We remove the wire from the NC terminal of the Prog Start switch and connect it to the 6 pin connector of our loader. Then the disconnected NC terminal of PKS3 is wired to the loader, so that we route the +12 that comes from the common through the NC of the switch, sending it on to the CPU when the switch is logically not pressed. To logically press the switch, we break that routing from the NC terminal and also deliver +12V directly to the NO terminal of the switch from our loader.

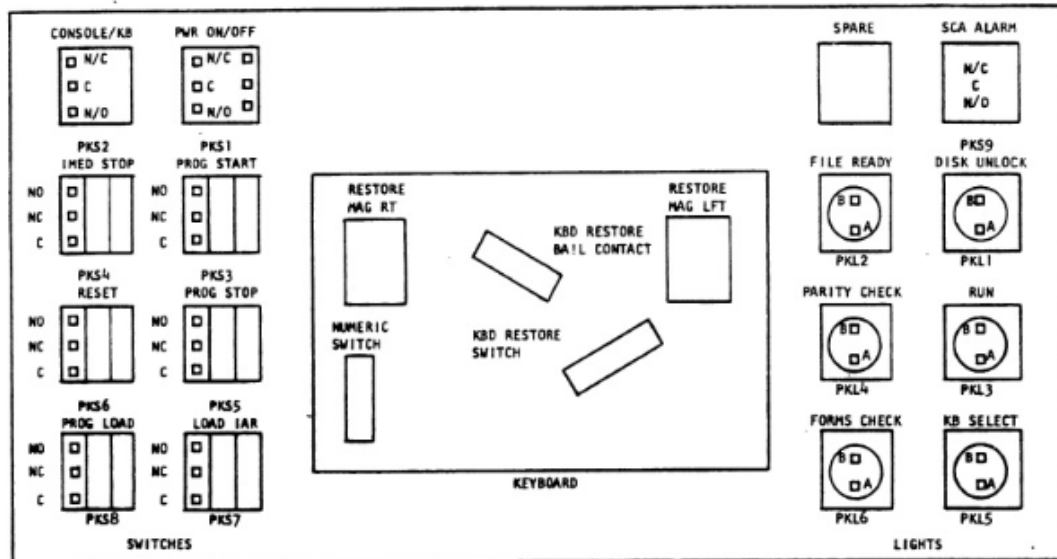
We connect to the +12V power supply from the main 1130 distribution terminal block and use it to provide power to our loader device. This goes to pin 1 of the seven pin connector on the loader board and the ground wire goes to pin 7. We can power the Arduino in one of two ways. A barrel power plug can be wired to pin 1 and 7 of the seven pin connector, or you can hook a diode from pin 1 of that connector to the VIN pin of the Arduino (using the connector end on our PCB).

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.



# Core Loading Automaton

On the updated diagram below, I show two of the photoisolators as virtual switches, plus the third which just delivers +12V is only shown as a wire labeled “Virtual push”.



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 the wire previously soldered to NC). The Load IAR switch is also visible where we connect to the C and NO contacts.

## Software

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 received 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.

# Core Loading Automaton

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 routes +12V to the wire originally connected there, coming from the NC contact since the physical button is not pressed. To press the button virtually we turn off the routing from the NC contact and instead deliver +12V directly to the NO contact via another photoisolator. Releasing the button is the reverse, removing +12V from the NO and allowing the connection from the NC contact to the NC wire.

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 19 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.