Computer History Museum

IBM 1620 Jr.
Project Description
Version 1.2 (8/31/2017)

The IBM 1620 Jr. project is a volunteer effort of the Computer History Museum. Its goal is to produce an operational version of the IBM 1620 that recreates as much as possible, the experience [visual, auditory, tactile, and visceral] of operating a real IBM 1620. It will allow visitors hands-on access to run legacy IBM 1620 programs from the museum's software collection, as well as newly written code, under the direction of the education department. While a software simulator such as SIMH lets one run IBM 1620 code, it doesn't come close to the experience of pushing buttons, typing commands, and watching the blinking lights of a real machine. CHM has restored several historic computers – an IBM 1620, PDP-1, and IBM 1401 – but these must be operated and maintained by trained volunteers. Visitors can only observe, not interact, with the machines. The IBM 1620 Jr. will require little maintenance and be able to stand up to heavier use.

Unlike the original IBM 1620 Restoration Project, this is not an historic restoration. That is, it does not have to preserve the historic authenticity of an artifact. Using a spare IBM 1620 front panel from the museum's study collection, the IBM 1620 Jr. will ***look*** and ***behave*** like a real IBM 1620, but be built from modern components as appropriate for reliability and low maintenance. It will come as close as possible to passing Dave Babcock's "parroty check" – that is, placed next to the front panel of a real, operating IBM 1620, one cannot distinguish any difference between them. Again, the goal is to recreate the experience of operating the machine, not hardware preservation.

At a minimum, the IBM 1620 Jr. will consist of:

1. An actual IBM 1620 front panel. The study panel is from an IBM 1620 Model 1, Level C, paper tape system. It is housed in a wooden, table-top cabinet. The incandescent lights, many of which are missing or broken, will be replaced with similar-looking, variable-intensity LEDs for both power and longevity reasons. Interface circuitry will connect the panel's lights and switches to the Raspberry Pi. The broken "Instant Stop / SCE" button will be repaired and the damaged toggle switches will be replaced with more durable counterparts.

2. A console typewriter. The console typewriter was a required component of the IBM 1620, as the machine couldn't be operated without one. The sight, feel, and sound of the console typewriter are important to recreating the experience of the real machine. However, the electro-mechanics of an IBM Executive typewriter would require a large restoration effort and an ongoing maintenance problem for the IBM 1620 Jr. An IBM/Lexmark Wheelwriter 1000 will be adapted for use as the console typewriter. This is a readily available, reliable, and easy-to-maintain typewriter whose printing can be made to look like the output of the IBM 1620 console typewriter. The only modification will be the addition of a team-designed circuit board interposed between the typewriter's keyboard and motherboard.

3. A "card reader/punch" device. The bulk of the IBM 1620 software library that the museum has, the Maniotes collection, was originally on punched cards and is now stored in disk files. For this software to load and run correctly, an IBM 1622 Card Read-Punch must be simulated. The initial approach will be to represent punched card decks using USB memory sticks – an idea borrowed from Oscar Vermeulen's PDP-8/I replica. While a small amount of the museum's IBM 1620 software library is on paper tape, the IBM 1620 Jr. project doesn't plan on simulating the IBM 1621 Tape Reader and IBM 1624 Tape Punch devices in the initial phase.

4. A Raspberry Pi 3 Model B running IBM 1620 emulation software. A new IBM 1620 cycle-level simulator will be written in C, as all existing simulators are instruction-level and not capable of properly

driving the front panel. This emulator will interface with the front panel and console typewriter, simulate the card reader/punch devices, and execute code as an IBM 1620 Model 1 Level F card system with 60,000 digits of memory and the automatic divide, indirect addressing, floating point, and MF/TNF/TNS options. A later phase of the IBM 1620 Jr. project may add support for the IBM 1311 Disk Storage Drive.

5. The IBM 1620 software library. The card-based IBM 1620 software in the museum's collection, principally the Maniotes collection, will be converted into a format usable with the IBM 1620 Jr. The software that can be verified to run correctly on the IBM 1620 Jr. will be packaged, along with its corresponding documentation, for use. Software that cannot be run due to unsupported hardware, might be addressed in a later phase.

6. An operations guide. The primary references for programming and operating the IBM 1620 Jr. will be the official IBM 1620 documentation. A brief guide to the unique elements of the simulated machine will be produced.

This is a dynamic project whose definition will be refined as the work proceeds.

All project discussions, decisions, documents, and status will be captured for historic purposes. This project may become the template for other such work in the future, just as the original IBM 1620 restoration project was the model for subsequent restorations by the museum.

There is no pre-set schedule or completion date for the project as all of the work is being done by museum volunteers on a part-time basis.

There are a number of ways that the IBM 1620 Jr. could be used by the Computer History Museum. Once the project reaches a demonstrable stage, then the various options can be explored.

The IBM 1620 Jr. project team currently consists of:

Name	**Position**	**Role**
David Brock	Director, Center for Software History	Project sponsor
Len Shustek	Chairman, Computer History Museum	Advisor
Dag Spicer	Senior Curator	Advisor
Dave Babcock	Volunteer	Project lead, software engineer
Steve Casner	Volunteer	Hardware engineer
Joe Fredrick	Volunteer	Hardware engineer
David Wise	Volunteer	IBM 1620 expert
Lee Courtney	Volunteer	Software library