Computer History Museum

# IBM 1620 Jr.
## Project Status
Report #1 (3/29/2017)


This is the first status report on the IBM 1620 Jr. project. An overview of the project is detailed in *IBM 1620 Jr. Project Description, Version 1.1 (3/21/2017)*. The project is structured as several sub-projects, each with a set of volunteers working on it. Note that this is only a summary of the work done to date. Details are available in email & phone conversations which will be available on the website.

After many email and face-to-face discussions between David Brock, Dave Babcock, Dag Spicer, and Lee Courtney (and later Steve Casner and Joe Fredrick) throughout most of 2016, the project formally began on Monday, November 21, 2016 when Dave Babcock picked up the spare IBM 1620 front panel from Adriane Tafoya at the museum warehouse. The condition of the panel was assessed, it was measured and photographed, and turned over to Steve Casner on Saturday, January 21, 2017 for all of the hardware work.

**Front Panel** (Steve Casner, Joe Fredrick)

The front panel sub-project consists of all physical and electrical work done on the spare IBM 1620 front panel, the Raspberry Pi 3 (RPi), and the interface circuitry connecting the two.

Great progress has been made on the front panel including:

- The top-level architecture of the interface was designed – 3 identical daughter boards stacked on the RPi 40-pin connector, communicating via an I2C serial bus, each driving 64 variable-intensity LEDs and 16 bi-directional switch/button/LED I/O ports.
- The interface circuitry was designed, reviewed, and revised.
- Sample LEDs were ordered and compared to a real IBM 1620 incandescent light (driven by a custom-designed/built circuit).
- Sample parts (ICs and edge connectors) for the breadboard and IBM 1620 Jr. were ordered.
- A Raspberry Pi 3 was bought, software installed and configured.
- The equivalent of $3/5^{th}$ of a single daughter board (2 of 4 PCA9635 variable-intensity and 1 of 1 PCA9675 driver ICs) was breadboarded and tested via prototype driver software on the RPi.
- The timing of a complete display update was measured (~ 4 microseconds) and can be supported by the RPi.
- A layout of the daughter board was done, reviewed, and revised.
- The 3 daughter circuit boards were fabricated and are in the mail.
- The broken “Instant Stop / SCE” button was repaired.
- Options for replacing the broken toggle switches are being explored.

Next steps:

- Select the LEDs to use and order them.
- Determine the value(s) of the resistor packs and order them.
- Assemble the daughter boards.
- Disassemble, clean, and reassemble the front panel.
- Finalize the additional lights and front panel labelling needed to emulate a Level F card system with all internal options.
- Replace all of the front panel lights with LEDs.

- Fabricate a mounting bracket and attach the RPi with its daughter boards to the back of the front panel.
- Connect all of the front panel wires to the RPi daughter board connectors.
- Acquire a 5v power supply.
- Determine the LED intensity levels which correspond to incandescent lamp duty cycles.
- Resolve and replace the toggle switches.
- Possibly repair the secondary contacts on a broken momentary switch.
- Possible work on the wooden cabinet.
- Test, test, test.

**Console Device** (Dave Babcock, Dag Spicer)

The console device sub-project consists of all physical and electrical work done on the console I/O device, also known as the console typewriter.

For the project development work, a keyboard and display will be used as the console device. The question of what to use for the "public" IBM 1620 Jr. console device has had some discussion and thought, but is not resolved. The IBM 1620 simulator software could be modified to accommodate any of the possible solutions.

Dag correctly points out that the most accurate experience would be to use a real IBM Executive Typewriter as the console device. There is a spare IBM 1620 console typewriter in the collection (and one currently on eBay for $10K), but it would need to be restored, an interface circuit designed & built to connect it to the Raspberry Pi, and a case made for it. It would require a lot on on-going maintenance.

One solution would be to use a not-IBM-1620-authentic, but vintage, printing terminal such as a Teletype Model 43, Diablo 1620 [a fitting product number], Anderson Jacobson 830/840/841, IBM 2740/2741, or Trendata 1000. These devices were designed for more rugged computer use and might be easier to restore, interface, and maintain.

Another approach would be to avoid any electromechanical device and focus on replicating the look, sound, and speed of the console device. A typewriter-like keyboard, such as the QwerkyWriter, could be used with a monitor. The monitor would display actual IBM 1620 Console font characters, play the sounds of the real IBM 1620 Console typing, and operate at 10 characters-per-second. Note that the same IBM Executive Typewriter was used on the DEC PDP-1, so we could easily capture the sounds of it typing and the character font used.

The console device requires a lot more thought, discussion, and work. It can also have multiple solutions over time and different uses of the IBM 1620 Jr.

**Card Read/Punch Device** (Dave Babcock)

The card read/punch device sub-project consists of emulating the physical punched card device in the software simulator using USB memory sticks to represent individual card decks.

For the initial phase of the IBM 1620 Jr. the card read/punch will be a virtual device. We might consider 3D-printing a miniature version of the IBM 1622 Card Read-Punch and equipping it with USB sockets. That way the user could "load" their "punched card decks" into the machine.

Depending on the use made of the IBM 1620 Jr., it would be easy to interface a table-top punched card reader, such as the Documation M-200/600/1000, to the IBM 1620 Jr. That way users could punch their programs and/or data using a restored IBM 026/029 and read physical decks of cards into the IBM 1620 Jr.

**Simulator** (Dave Babcock, Steve Casner)

The simulator sub-project consists of developing a new IBM 1620 cycle-level simulator, coded in C, that interfaces with the front panel hardware. It will simulate 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 processor options.

The simulator will consist of initialization code and two separate threads, each locked onto their own RPi CPU core. One [execution] thread will simulate each machine cycle as defined by the IBM 1620 Customer Engineering Reference Manual. The state of each front panel control switch and momentary button will be queried every machine cycle. The ON/OFF state of every indicator light will be tracked and counted every machine cycle. Development of this execution simulation thread has not started.

The second [display] thread in the simulator will periodically sample the ON/OFF counts of the simulated indicator lights and proportionally set the intensity of each LED on the front panel. This approach will produce realistic looking lights when simulated code is running, which "pulse and throb" like a real IBM 1620. An initial version of this code, and the initialization code, was written and used in testing the breadboarded daughter board.

**Software Library** (Lee Courtney, Dave Babcock)

The software library sub-project consists of converting all of the museum's IBM 1620 software collection to a format usable by the IBM 1620 Jr., assembling the associated documentation, and testing the programs on the IBM 1620 Jr.

Work on the software library has not begun.

**Operations Guide** (Dave Babcock)

The operations guide sub-project consists of writing a brief document on the operation of the IBM 1620 Jr.

It is premature to begin this effort.

**Website** (Dave Babcock, Team)

The website sub-project consists of selecting, configuring, and populating a dedicated website on the IBM 1620 Jr. project which will document the entire project.

Preliminary discussions and investigation about which platform to use (Slack, Google, Wordpress, etc.) have occurred, but a decision has not yet been made. Once a platform is selected, past email threads and documents will be uploaded. Going forward, all discussions will be done through the website and all documents will be stored there.