The HoneyPi Project – A home-brew computer with attitude

Rob Sanders

This article is about the origin of and background to this project which is itself described on my embryo website www.honeypi.org.uk. At a recent CCS meeting someone suggested that it might be of interest to members, so here are the bald facts.

It may seem macabre to start a computer resurrection project with a death, but it is a fact that mother-in-law died and left us with her home to sell and therein lay the original problem. For several decades while I was working for a life assurance company I had acquired much of their cast-off electronic equipment to fuel my electronics hobby. Indeed amongst other things I could map the entire history of word processing with my collection of components from the earliest Flexowriter with its Selectadata dual speed paper tape reader through the Dataplex machine with its magnetic card drives, Honeywell Keytape machines with magnetic tapes and IBM Selectric typewriters, Wang word processor with hard and floppy disk drives and daisy-wheel printer along with the Honeywell Level 6 and DPS6 minicomputers. Even now I can still lay my hands on policy clauses in Flexowriter code on paper tapes and corporate pension fund contracts on magnetic cards, but the problem was that much of my hoard was stored in a shed in mother-in-law's garden and it all had to be moved into our garage to sell the property. Now possibly I could pass a few working minicomputers on to a museum but over a thousand logic boards from Keytape machines were more of a problem. The HoneyPi Project was my solution.

The Keytape machines used Honeywell series 200 components, so it occurred to me that I might be able to build a computer with them, maybe even one compatible with a Honeywell 200. I had started my IT career programming an H200 in the mid-sixties and still remember it with affection. Building the computer wouldn't solve my storage problem, not unless it was so like an H200 that I could give it away to someone as a replica when it was finished, maybe along with a large collection of spare parts. The peripheral devices might turn out to be a dual-speed paper tape reader and punch, two magnetic card drives and a daisy-wheel printer but that would just add to the appeal. Perhaps my idea of using magnetic tape drive vacuum/pressure components to enable it to drink a glass of water while whistling *Land of Hope and Glory* would have to go on the back burner though. No, HoneyPi wasn't started as a worthy reconstruction project but as an exercise in recycling. Nevertheless it has developed since so I will attempt to promote it.

To my knowledge the last series 200 machines in use, maybe in existence, were scrapped in the year 2000 by a company in Pennsylvania which had continued to use them until the Millennium bug forced them to be decommissioned, not because they themselves were unreliable but just because the software was. Even the late Dr. William L. Gordon, chief designer of the H200, hadn't known that they were still in use then, as his daughter told me. The H200 was first marketed fifty years ago and knocked the then popular IBM 1401 off its pedestal, forcing IBM to stop dithering and launch their 360 series before they lost too many customers, so the H200 has its place in history. However, it deserves to be remembered for more than that in my opinion. It wasn't just a clone of the 1401 but a more sophisticated machine that just happened to be compatible with it in many respects. Dr. Gordon's daughter found in his personal papers a document written by him describing the thinking behind many aspects of its design and his own words put the idea of the H200 as the so-called "IBM 1401 Killer" into perspective. She plans to give the document to a museum but I won't pass on its contents without first getting her permission.

So, to assess the H200 in its own right, like the IBM 1401 it was a two address character stream processor in that one operation could process a string of characters of any length as defined by punctuation marks set in memory, but unlike the 1401 it used two bit punctuation marks which enabled it to recognise four level data structures made up of characters, words, items and records. For a newcomer to computers like myself in 1966 this was obviously what a computer ought to be able to do, but few machines have ever achieved it so simply at the hardware level in the way that

that little second generation H200 did. It tackled data in such a naturally human way without the artifice of fixed length words that anyone could grasp the principles and programme it and it was a welcome change from working on spreadsheets with a hand-cranked calculator as I had been doing previously in the Actuaries department. A computer which treated all data just like numerical spreadsheets would hardly have caught my imagination in the same way no matter how fast it was. Only later did I have to learn how to programme machines that didn't give any consideration to humans. That said, its way of working wasn't far off the principle of a Turing machine with its data stream handling approach and, lacking in its simplest form a memory stack where data could be dumped thoughtlessly until needed later, it forced one to think about the machine's own state and use self-modifying code, machine state being another concept envisaged by Turing in his conceptual machine. Therefore despite the human-compatible nature of the H200 it taught me fundamental principles of computing which served me well in later years. I couldn't have found a better introduction to a successful career in software development, for the H200 was not a number-crunching computer nor a data processing machine but genuine information technology from the start. That was not surprising as, apart from being a modest general-purpose machine for smaller businesses it was also intended to be a front-end communications device for much larger people-unfriendly number-crunchers. Nowadays information technology focusses considerably on data stream handling and the H200 would be right at home. Finally, regardless of what went on inside the H200 it was a good looking machine when seen from the outside.

My heap of logic boards from seven Keytape machines contained mostly early integrated circuits from the 1960s with two gates in each package and, oddly I thought, connections for feedback capacitors. Possibly the feedback capacitors provided hysteresis which protected the gates against the high levels of noise present on wire-wrapped backplanes. The H200 was a transistorised machine but I later discovered that at least one other computer in the 200 series, the smaller H120, used ICs, which was why it was also physically smaller. Therefore I am now justified in constructing my machine using those ICs. It is after all not a precise replica of an H200 but a pastiche, a machine in the style of the 200 series, a machine that Honeywell could have built at the time. Apart from the logic boards I had a couple of backplanes with a total capacity of 200 logic boards between them, but my main problem was the magnetic core memory. Each Keytape machine had a single plane core memory capable of storing just a few punched card images. Furthermore for the convenience of the operator they were wired for decimal addressing but the 200 series used binary addressing. In his document Dr. Gordon explained the decision to deviate from the IBM 1401 precedent in this respect as he believed, wrongly as it turned out, that 1401 users only used addresses as meaningless field identifiers and didn't manipulate their values. Those who saw the H200 as a 1401 clone subsequently saw this aspect of its design as a weakness whereas users of the machine in its own right saw it as a benefit. In any case Honeywell had got the business approach right, to make a product similar enough to one's competitor's to gain customers but different enough to keep them. For my part I devised a way of mapping the binary addresses onto decimal ones in a non-sequential one-to-one way which could operate through simple logic in real time without slowing down the machine. I still had the problem of constructing a nine plane memory though.

I had long ago given one memory board to a work colleague, appropriately as a memento, so I had left four planes containing 4200 bits and two containing 2100. It was impossible to modify the wiring of the memories without upsetting their noise cancellation properties so I would have to use them as they were. The only possibility would be to construct a 2k nine-plane memory by accessing the larger planes twice per byte, but that would slow the machine to well below the H200's operating speed and it would have a hard time outperforming the working 1401s still in existence. However, I still had one extra plane in hand, so I went back to basics and reviewed the way that core memories were used. Here I am considering the classic flux reversal mode of operation rather than the exotic differential technique as used in the H200 control memory. In his document Dr. Gordon explained that this control memory was the one extravagance in an otherwise frugal design and I believe that it was also the central reason for the H200 outperforming the 1401 as well as its ultimate downfall. The control core memory ran at four times the speed of the main memory but that generated considerable heat and the differential storage method was heat sensitive, so if the ambient temperature was too high it would malfunction or even burn out, as my own employers found out the hard way. Honeywell claimed that the H200 didn't need a fully air-conditioned environment in which to operate and even demonstrated it in hotel rooms to prove their point, but as their company had supplied environmental temperature control equipment from its earliest days I suspect that they made sure that the hotels they chose for their demonstrations used them. No doubt as that was their key business they felt justified in assuming that their computers would always operate in such an environment.

The H200 was therefore very much a thoroughbred, reliable in familiar surroundings but unpredictable outside of them. This may be the reason why none now exist, for any machine unwittingly operated outside of its comfort zone undoubtedly eventually died. I do now have an original H200 control memory but have no intention of using it; there is a point where authenticity must give way to practicality. An engineer who used to work in the Computer Control Division at Honeywell, which was the continuation of the former company "3C" bought out by them, told me about the rivalry between the series 200 engineers and his division. The series 200 engineers regarded the CCD machines as "toys" while the CCD engineers regarded the 200 machines as unreliable dinosaurs which could easily be wiped out by just a little global warming. He gave training courses on the H716 to series 200 engineers and to gain their respect he would turn off the H716 halfway through a demonstration and then turn it back on with the result that the machine continued working as though nothing had happened. Despite using core memory there was no guarantee what an H200 would do once it had lost power. The H716 also allegedly worked reliably at any temperature from freezing to forty centigrade. Nevertheless success with H200 sales justified Honeywell in calling themselves "The Other Computer Company" in their literature.

Returning to my problem, the classic method of using core memory distinguishes read cycles from write cycles and indeed multiplane memories with common write circuits for all planes and separate inhibit circuits for each plane virtually have to work this way, but my memory planes each had their own circuits, so I could use them how I liked. My solution was to use the fact that the sense amplifiers for a core memory always output the *differences* between what is being stored and what is already stored, therefore one can effectively retrieve the existing contents while storing new ones by recombining the differences output with the inputs. That meant that I could process the memory in just three cycles instead of four. Three cycles matched the speed of an H120, so my machine would still be compatible with a series 200 machine. My technique involved retrieving the first half-byte in the first cycle, then storing it or its replacement in the second half-byte during the second cycle while simultaneously retrieving the second half-byte which would be stored or replaced in the first's location in the third cycle. This meant that bytes would continually switch between being stored big-endian and little-endian whenever they were accessed but I had a spare memory plane in which to store the current state of each byte. The key to the success of this approach was that it needed exactly what I had, two single-sized planes for the parity and state bits and four double-sized planes for the eight data bits. Fifty years ago I might have applied for a patent but as it was I regarded it as a personal victory. I should also mention that 2k was the smallest memory size available in the original 200 series, so my machine would still be within the specification. The project was feasible.

I have mentioned this episode not for self-aggrandisement but because it illustrates a key aspect of such reconstruction projects. This was never meant to be a carbon copy of the construction of an H200 but a personal experience of resolving problems set in a similar context to that at Honeywell in the 1960s. One sees other examples of such reconstructions with people living in conditions as they were in the stone age or the 1950s and they are never entirely genuine, but they are undeniably educational and working out one's own solutions in such an artificial context is more rewarding than just slavishly following in someone else's footsteps. One revelation is how one can see in progressive versions of the series 200 logic boards the way that higher gate densities, brought about by integrated circuits, combined with the existing pin limitations of the standard board format forced the evolution of computer design as the specific structure of a particular machine moved out of the backplane wiring and onto customised printed circuits until ultimately the backplane was reduced to just a universal communications bus between complete subprocessors as in the level 6 minicomputers. For this project we must make similar decisions about whether we use many underpopulated general purpose logic boards or fewer high density boards made to our own specifications. Honeywell did it both ways, so we won't be outside our self-imposed guidelines whatever we decide. Anyway, now I could possibly build a machine to my satisfaction but there remained the problem of whether one with only 2k bytes of memory could do anything worthwhile and also whether I could still programme one. My answer was "The Pi Factory" as described in detail on my website, a programme devised to meet constraints that no IBM 1401 could match, not that I would think of challenging those worthies who keep the last of that breed working in America. Now with a plan and a demonstration programme I did the unbelievable; I shelved the entire project for two years and wrote a novel.

My wife thought that I'd gone mad. I'd never even written a short story before in my life and she couldn't understand why I was in the spare room writing a novel instead of building a 1960s mainframe computer. Evidently when it comes to gauging insanity there is no norm for comparison. The truth was that, just like my HoneyPi project, it was just a pastime. Both the novel and the computer would just be by-products if either was ever completed. Writing the novel did tell me something though, first, that it was only the first story of a trilogy and perhaps life was too short to write the rest *and* build the computer and, second, that it wasn't just what one did but when one did it that was critical. After a two year delay the errant muse that prompted me to start writing suddenly prompted me to return to the HoneyPi project with startling success. Through my meticulous searches on Google I made contact with Dr. Gordon's daughter and she was most interested in my plans. Her father had died a year or so earlier after suffering from Alzheimer's for five years and she'd realised since that she'd never paid enough attention to his boring perpetual enthusing over the H200. Corresponding with me prompted her to sort out his personal papers still stored in her basement and find that seminal document. I also discovered Dr. Marcel van Herk, a relatively young vintage computer enthusiast who possessed at his home in Amsterdam almost all the components needed to build an original H200 memory unit that he'd acquired as a teenager. He was willing to entrust these to me for the project, so he became an active colleague with me in it and I, somewhat sadly, wouldn't have to implement my novel memory design after all.

Having all of Marcel's logic boards to accommodate meant that my backplanes wouldn't be adequate, so I had a new problem. As a contingency plan I had a foundry cast a replica of my existing backplane frame before we started using it but then another miracle occurred. Correspondence about Easycoder, the H200 native assembly language, with a former programmer located in Australia resulted in someone else mentioning the engineer who maintained the last H200s in Pennsylvania in year 2000. It turned out that when he decommissioned them he'd taken home some 30,000 logic boards and around 30 backplanes to sell to scrap dealers to help fund his daughter's forthcoming college fees. To cut a long story short Dr. Gordon's daughter in New Jersey was introduced to the last H200 site engineer in Pennsylvania and a backplane was shipped across the Atlantic to me while Marcel visited the said engineer's home during a trip to give lectures in the States and brought back a selection of extra logic boards. As I have also made contact with a few former site engineers willing to search their memories for answers to pressing questions as well as an enthusiastic man in Yorkshire who happens to run the business manufacturing printed circuit boards that his father started at their home around the year that the H200 was launched, I now feel justified in having given my website that pretentious ".org" suffix.

At present I am busy building the 8k memory unit that Marcel and I agreed was the best option with our limited supply of boards. Even so Roger, our man in Yorkshire, will have to manufacture some replica boards at his factory to fill in the gaps once I determine whether Marcel's core memory modules are actually still functional. The final difficulty that we would have to face is that we don’t have enough of those iconic chunky button tops to go on the control panel as illustrated on my website. Unfortunately a former work colleague who operated our company’s original H200 in the 1960s took its control panel home as a souvenir when the machine was decommissioned but threw it away when he subsequently moved house. Without that memorable control panel it just wouldn’t be an H200, but the next miracle is apparently on its way. Someone has informed me that he owns an original H200 control panel, if he can find it. He believes that it is in the garden shed at his parents’ home. Apparently the climate in California is much better for storing electronic equipment in garden sheds than in England and that must be an appropriate point to end this article.