The Home Computer - a Liberty BASIC Retrospective

By Carl Gundel Nov 22, 2015

-
CarlGundel
http://www.libertybasic.com
http://www.runbasic.com

We used to have a different name for personal computers. We called them home computers back in the heyday of 8-bit processors. This was a simpler time when you would buy a computer and plug it into your TV set and save your work on a cassette tape. These computers often didn't have enough memory in them to hold the text of this article.

The home computer was marketed as a fun and friendly device, and there were lots of different kinds of home computers from many different companies. There was not much pressure to make these computers compatible with each other, and so there was lots of creativity everywhere you looked. The most famous of these were the Commodore 64, Radio Shack's TRS-80, and the Timex TS1000, but the list is long and illustrious. These machines were all different from each other, and they were all useful, fun and great. A lot of people built their own computers too.

One of the great things about home computers was that you could turn them on and program them right away, usually in a language called BASIC. There was no beeping or operating system loading message, no hard drive cranking away loading up the operating system. Just turn it on and in a moment the screen would show something like:

```
ComputerPro Pixey Model 2
Pixey BASIC v2.01, copyright 1980
14523 bytes free
Ready.
_
```

Then you just started typing code and the RUN command would set it in motion. :-)

Of course in addition to programming them in BASIC you could also do a thousand other things with them, and this generation of computers was simple enough that a 200 page book was enough to give anyone technical mastery of the computer.

No one really sells home computers anymore. IBM began the end of it all in 1981 with the introduction of the IBM Personal Computer. This was actually a very sterile and uninspired design, and even technically unimpressive for its day. IBM didn't think there was much of a market, so they didn't try to make something great. Because it was IBM's product and because IBM made the mistake of not creating something patentable, everyone wanted to copy it. This was a mistake in the sense that while in the midst of everyone trying to invent what computing was going to be all about, IBM threw their pony into the show and everyone stopped experimenting in order to conform to a crude design. We didn't even know yet what

a computer could be. What a shame, but I'm getting off topic. ;-)

Liberty BASIC is a language that I began developing back in 1991 as a simple way to program Microsoft Windows. There was really nothing else at the time, and even Microsoft did not have Visual Basic yet. People who were programming in QBASIC were not able to properly use it with Microsoft Windows because QBASIC is DOS-based software. Today QBASIC is largely forgotten.

There was also a very personal reason for creating Liberty BASIC. I had always wanted to write my own programming language, and so this was originally meant as a project to prove to myself that I could do it. This became a passion for me.

The first version of Liberty BASIC was meant to be a sort of QBASIC for Windows. It came out in 1992 and it ran on Windows version 3.0. It was a very simple package which supported the core BASIC syntax that was inspired mostly by Microsoft's flavor of 8-bit BASIC languages, but since I also had used other versions of BASIC and so I didn't feel married to any particular dialect. Later on though I did make some adjustments for the QBASIC community.

The strength of simplicity is that it doesn't overwhelm the user. As I developed Liberty BASIC I had to resist the countless requests to add everyone's favorite feature because I didn't want to build a Tower of Babel BASIC. This became even more important as time passed because I wanted to preserve the approachability that encouraged several authors of books and magazine articles to write about Liberty BASIC without my ever approaching them. Liberty BASIC is a language suitable for the hobbyist and beginner and their books and articles are testament to this.

Liberty BASIC eventually gained the ability to call OS functions and use external libraries, it went from 16-bit to 32-bit, and even added graphical sprites for those who like to create video games. The sprites are actually an integration of some excellent work done by Alyce Watson, a well loved member of the Liberty BASIC community who also created programming tools for Liberty BASIC and wrote books.

All the while I was developing Liberty BASIC I was also learning, and reading about language design and also spending time with people who implement programming languages. I began to have ideas about a version of Liberty BASIC developed from scratch. So I and my development partner, Scott McLaughlin began to write new code for a version of BASIC which would be backwards compatible and which would also have an elegant foundation for more exciting programming experience. For me this has more to do with the tools than the language itself. The editor and the debugger should be rich, and one idea that we are developing is an interactive workspace, like an exploratory sandbox for coding.

The original BASIC languages are interactive interpreters. The benefits of this are huge because you can type just a little code to try out an idea and run it instantly. Then play around with that and change it. This was you can learn a lot really fast, and it's fun so it motivates you to keep playing with the system. This is lost in most modern programming languages including Liberty BASIC, but the new development aims to have a more interactive style so the user can 'play around' more.

The tricky part about all this of course is to know how much is too much to add to a language which is supposed to be simple and approachable. It is possible to try to please too many people only to please nobody in the end.

Eventually because of some technical issues related to the Smalltalk development tools that we use, our development of Liberty BASIC v5.0 was put on the back burner. We decided to work on a web version of BASIC while we waited for our tools vendor to get their act together. The result is Run BASIC (see http://www.runbasic.com) which in my opinion is just about the easiest web programming system in existence. Imagine a web programming system easy enough for a ten year old to use.

After this I sat down and wrote a book Beginning Programming with Liberty BASIC, which I self publish and which can be purchased on Amazon. I've been pleased that the book has been well received.

Since then development on Liberty BASIC and Run BASIC has been somewhat slowed as time has permitted. Recently I did release a v4.5.0 of Liberty BASIC to add a few new features and fix some bugs before resuming development on version 5 and also to do some more work on Run BASIC.

Examining the world as it is now I'm very interested in the recent Maker movement. I love to see people interested in making things again. In the computer space this is well represented by the Arduino and Raspberry Pi computers and their respective communities. I would love to develop a version of Liberty BASIC for the Raspberry Pi in particular, or even better a bootable disk image which would turn the Raspberry Pi in to a complete system where BASIC is the operating system. This would be an attempt to bring us back around to the home computer ideal once again.

There are so many exciting possibilities. Long live BASIC!