Embedded Systems in Clocks

Bob Roan
April 2008
Robert.Roan@post.harvard.edu

I'm Bob Roan. Thanks for letting me talk with you.

Today, I want to speak about an impending revolution in clock design due to embedded systems. An embedded system is a computer inside a clock, the way a computer is in a car or a telephone. It's still a car or a telephone but very different. Embedded systems will spread deep and wide through clocks, as they have many other fields.

This is the beginning of something very big, a once in a lifetime opportunity for you as clock makers and clock collectors to get in at the very beginning and help set the direction of a major movement.

I'll use my work as an example, but the important thing is to see me as a harbinger of the future.

Have any of you ever added a computer to a clock?

Technology is all around your work and it's only a matter of time before it permeates the clock case.

This is a great area in many dimensions from the creative to the financial. I recommend it to all of you. It's robotics in a clock. The rewards for the curious hobbyist are amazing, and reason enough for many of you to embark on this adventure.

Before I talk about the creative aspects, I want to say a few words about this as a business opportunity.

An article in the December 2007 issue of the Harvard Business Review described a methodology for evaluating innovation opportunities which asks three broad questions:

Is it real?
Can we win? and
Is it worth doing?

The first question, "Is it real," asks about both the market and the product. Just because they can live and prosper in our imagination doesn't mean they can be made real.

The relevant questions about the reality of the market are:

Is there a need or desire, and
Is the size of the potential market adequate?

There is a desire. Time is nature's heartbeat. Clocks go deep into our psyche and that's why they're durable luxury markets. People are always looking for tasteful, innovative and grounded gifts for themselves or others.

We live in an age of innovation, an impulse which has not touched clocks.

If you boil clocks down to their essence, there are two, rotating hands and digital numbers.
There hasn't been much innovation in clocks since 40 years ago when the seven segment display showed up.

Popular culture exalts the new and disparages the staid. No matter what your product, this is one of the first judgments people will make about it.

For most people, clocks fit solidly into the staid category. They're a commodity

Embedded computers will let people like you move clocks out of the staid and into the new category by challenging assumptions about clocks and pushing their boundaries, thus satisfying what I believe is pent up demand for innovation.

The authors have three questions to help determine if the product is real:

Is there a clear concept,
Can the product be made, and
Will the final product satisfy the market?

A clear concept requires that you have a concrete idea of exactly how a computer will add to a clock. This will take a bit of R&D on your part to learn about the possibilities as well as the way technology leverages you personally.

My clear concept has four aspects:

The essential similarity between computers and clocks,
The relationship between clocks and timelessness,
Loosening the constraints on clock design, and
An alchemical approach to perception.

Whether the product can be made depends on the manufacturing process and the availability of components.

The product is a clock, which you know a lot about making.

The components and technology are widely available.

Whether you can win depends on competitive advantages and understanding of the market and the competition

Your sustainable advantage relates to

What it is that people like about your clocks,
how many other people can do what you do, and
If that's a direction you'd like to develop.

The job of the clockmaker is changing dramatically. It used to be about holding something, either six LED digits or three tightly connected rotating hands. Now it's about reinventing clocks the next step forward, back into their rightful place of wonder.

If you enjoy psychological and perceptual boundaries, a strategy based on innovation may make sense for you.

Whether it's worth doing depends on the profitability and risk of this product and how well it fits your overall growth strategy.

A profitability and risk analysis looks at two areas:

Are returns greater than costs?

Are the risks acceptable?

The potential returns relate to how much extra you can charge for a computer with a clock.

Your costs will include:

Extra expense for each clock. You'll have to buy the digital clock movements and displays from someone like me or make your own,
New equipment if you decide to make your own digital movements, and
Getting started. It will take time and money to learn new skills and experiment and build and rebuild prototypes.

Does it fit your growth strategy? Do computers interest you? Do you have an aptitude for them? Do you imagine clocks that can do things or look ways that are not currently possible?

(point to my work)

That's the business case for embedding computers in clocks. Now I'll talk about how I've done it.

I'm doing some very ambitious things here, which I point out to show that it is possible to do very ambitious things with the available technologies.

As I discuss the details of my system, think about what you could create with the power behind this example.

My system includes a central computer and peripheral boards that take keypad input, turn lights on and off, and control motors.

The peripherals have their own computers. The central computer calculates the intensity of each dot, segment or LED, as well as in which position each motor should be.

It gives that information to the controllers on the peripheral boards, whose job is to make that happen.

Communication takes place via serial signals. All the peripheral boards have a max 232 integrated circuit which translates the 12 volt signal of my controller board into the 5 volt signal used by the peripherals.

All the boards also contain debugging and programming circuitry so I can reprogram the software and debug my programs without removing the computer chips.

The keypad contains a 4 by 4 keypad, a 4 line by 20 character display, clock chip, battery backup, and a microcomputer.

The keypad and display can each be removed from the board and connected via a cable so you can place them anywhere.

There are many types of LEDs. Right now, I can control 4 of them:

Individual, discrete LEDs,
The 7 segment, numeric LEDs you find in clock radios,
The kind of high power LEDs that are replacing light bulbs, and
A digital chip that contains 8 rows of 8 columns of LEDs

Display boards contain a microcomputer, the LEDs and a number of maxim 7219s, integrated circuits that controls 64 dots, individual LEDs or segments.

The microcomputer takes the configuration from the central computer and assigns it to the various maxim 7219s on the board.

Each motor board has a microcomputer, which works through a ULN 2803 integrated circuit to send the pulses that turn a stepper motor.

The motor can be mounted on the board or connected via its cable in case the motor needs to go somewhere that the board won't fit.

Everything is controlled by software. I'm programming the peripheral boards in Basic and the central computer in C++.

I use configuration files so the same software can operate different size displays and different mixtures of components.

An addressable and reliable coordinate system is essential. The software must be certain about the correspondence between a light's virtual coordinates and its physical ones or the intended outcome will not occur.

The components are widely available.

My central computer is made by JK Microsystems, a small company, and the computers on the peripheral boards by microchip, a large company.

The max232 for my serial signals and the max7219 display chips are from maxim.

I get my LEDs from China and my motors and gears domestically. I have about 5 or 6 suppliers but get 80% of my stuff through one of them, Jameco.

Here's how I make my circuit boards.

I designed an excel spreadsheet which tracks the position of each component, the number of which vary. The spreadsheet does all the calculations for determining the exact positions and also does some consistency checks.   Then a Microsoft excel macro generates visual basic commands describing that configuration.

I copy those commands into a Microsoft Access database I created which then adds the specific components and generates the electrical connections between them. A database macro generates commands describing this layout in a language that can be read by EAGLE, a circuit board design program which I purchased.

From EAGLE, I get another script which can be read by circuit board manufacturers. From them, I get a circuit board, to which I solder the components.