The Hand-Held Bone / WiFi Viability Project

There are two physical pieces for this project: A hand-held unit and a receiver unit. For brevity I'll call them HxB and RxB respectively. To avoid creating a new user interface for the initial part of the project, the HxB will use the keypad and enclosure for our current wireless hand-held baseball switch assembly. It is safe to assume that future incarnations of the HxB will include more diverse capabilities than incrementing Balls, Strikes, and Outs. The primary function of RxB will be to receive keystrokes from HxB, to keep track of Game data, and to send display data (MP architecture current loop) to a scoreboard display. RxB will be housed in a ScoreLink receiver enclosure.

Communication between the HxB and RxB will be via WiFi. The main goal of the project is to demonstrate the effectiveness of WiFi in a realistic environment. We will want to install the RxB in a small scoreboard (perhaps LX1320) and test it on baseball fields, on football fields, and in gymnasiums. In addition to the basic scoreboard functions (Scores, Time, etc.) we will need to be able to call on some special test functions to evaluate the WiFi performance. I do not know enough about WiFi to describe the best test methods and most useful measurements to make. Signal Strength and Network Latency come to mind. I do not know how to best measure or express these concepts.

Wifi has a signal strength value that I can make go to the scoreboard like our current system.

Network Latency won’t be an issue because there is only one network so no varying amount of jumps.

The bare minimum distance for useful wireless operation is 300 feet from the hand-held unit to the scoreboard. A better goal would be a minimum of 1,000 feet with both pieces at three feet above ground level. For the initial project, we can assume a one-to-one relationship between the HxB and RxB -- no Group Management or similar concepts.

Then it is very easy. lol

We need to test security and stability issues. Again, these are topics I know almost nothing about, so I cannot recommend specifics. But there must be industry practices for determining how difficult or easy it is to disrupt a WiFi system. We need to test how the system responds under stressful conditions. What happens when you separate the pieces enough to break connectivity? How quickly can they link up again? How long does it take to reset, reboot, or power up (both in terms of the device getting up to speed AND the time required to reconstitute connectivity)?

John and I have discussed this and we have the plan for security that we are happy with. My current code for the HxB reboots forever till it joins the scoreboard network. It connects very fast and appears to work immediately. I will keep it that way for any release at all costs. Rebooting with micropython is so fast no one will know. All variables have to be defined before this can be totally know and then only at the very end will John add the security layer.

The HxB is supposed to be a battery operated device. Whether it uses AA batteries or some other power source is not critical. For initial tests, battery life is not very important. For the production version, a life of 8 hours would be acceptable for a battery that can be easily a cheaply replaced (such as an AA battery). If the battery is not easily replaceable, 16 hours (and a recharge time of less than 3 hours) would be an acceptable minimum.

Can do.

Let's assume the initial demo system passes all tests with flying colors:

- Communication is stable over distances much greater than 1000 feet.

- All security concerns are addressed.

- It FEELS responsive, and we can also MEASURE responsiveness to confirm the feeling.

- Recovery from problems is graceful.

- Boot ups are quick.

- Batteries last all day.

I expect nothing less of myself sir!

So, what happens next?

“Party time!”

I think the next critical question will be: Is a tiny box with ten buttons the way we want to go with this? We may not have the luxury of asking this question, because we may discover that we can't make everything work in that box. But, if the tiny version is an option, we will need to be certain that we are developing the right product at the right price point for the market we wish to pursue. There are two things that worry me about the tiny box. One is I don't see a good (comfortable, aesthetically pleasing, affordable, etc.) way to add a mechanical rocker switch for super-tactile clock control. The other is that our three biggest competitors -- Dak, Nevco, and Fair-Play -- went for bigger hand-held devices with LCD text displays. I'm afraid we are going to end up with a solution that costs as much or more than what our competitors offer and has no distinguishing feature other than smaller size.

How much do each one of those cost?

On the other hand, Electro-Mech has always specialized in producing simple and sturdy products aimed at customers who are not interested in bells and whistles. If the tiny box can give us a product that is simple to use, reliable, and cheap, then we may have a winner.

The way to have the best product it to get everyone to look at it through the right filters.

One way to think about pricing is to look at what we have in an MM console + embedded transmitter + embedded receiver. Because we include the price of the console in a scoreboard, the customer only sees the additional cost of the ScoreLink system -- typically discounted to $600. Because the hand-held device is less substantial than an MM console, I believe we should try to stay well under this price level -- maybe $480 ($640 list) for a set of HxB + RxB. This implies a material cost of about $200 for both pieces combined. Having a material cost well over $200 would be an indicator that we should NOT design the product to be a tiny device with limited functionality.

Basic price:

RxB = RPi3 $35 + Embedded box w/socket/plug $14 + Antenna w/wifi module $14-19 + MIC or current loop converter $7-18 = $70-86

HxB = Module $30-40 + Enclosure $7 + Keypad $12-20 = $49-67

Total = $119-153

Looks like I can get some sweet bells and whistles now. Yee-Haww!!!!!!!

What next?

Lunch?

Once we have proven that we can build a functioning hand-held WiFi-based control device in a package that we like, what refinements do we need to make to take it to market and what additional features should we roll out in the future? Lots. This is where the project hits the ChapMan Tar Pit, because I am interested in designing the User Interface. This includes the keypad layouts as well as describing what functions are available and how they work (from a User perspective). While I cause delays pursuing this goal, there is another avenue that can be explored independently: Giving the HxB the ability to pass files back and forth between the RxB and a PC (or other device). We talked about implementing this by having the HxB act like a USB drive. There may be other approaches. This will inevitably lead to a separate (graphical) UI development process, but we can worry about that part later.

Part of the goal of developing the file transportation capability (and the UI associated with it) is to provide a method of configuring the HxB and the RxB without the limits of a UI with just a (a just) few buttons, and some LED indicators. A PC-based (graphical) configuration UI can open up ways to implement MP-console features like dimming, ETNs, Practice Segment Timers, and Group Management. It can also lead us to features that we would never have dreamed of adding to an MP console. This would include the ability to move data files from the scoreboard (RxB) to a PC and upload them to the Internet -- making them accessible to Electro-Mech. And, of course, we could push information in the other direction -- for example, allowing customers to download firmware updates or special configuration files, copy them to the HxB and then to the RxB.

File transfer will already be implemented for version 1. It will just need hooks and rules for when to activate.

All you have laid down in this tome conforms to the grand vision I (Sith Lord Darth Awesome) have foreseen.