BlueTwiddler:
Designing a Twiddler for the Future

Steven Akins steven.akins@gatech.edu
Tarandeep Gill taran@gatech.edu
Michael Hewner hewner@gatech.edu
Camilo Vargas vargascamilo@gatech.edu

INTRODUCTION

The Twiddler2, developed by the Handykey Corporation, is a chording keyboard with a built in pointing device used primarily with wearable computers due to its portable design. The Twiddler design featured a 3x4 matrix of buttons that required users to press multiple keys simultaneously to generate characters or predefined shortcut sequences.

In late July of 2007, Handykey sold their last Twiddler2. [3] They have stopped production on that model, mainly because the mouse pointer component is no longer available. The cost of developing a new Twiddler was more than the company could afford, even though they had some fresh ideas for the old keyboard. For many involved in wearable computers, a new, updated Twiddler would fill the gap left behind by Handykey.

Our team decided to take up the challenge of building a new Twiddler that had updated features including Bluetooth wireless access and a more ergonomic case design. We also wanted to build the Twiddler in a reproducible way so that Twiddler enthusiasts could use our design to produce their own copies of our Twiddler or other similar devices.

PREVIOUS WORK

In the Georgia Tech Fall 2006 Ubiquitous Computing Class, a team modified an existing Twiddler device to use Bluetooth and wrote firmware for the BlueCore processor to use the Twiddler chording scheme. [1] The firmware and the modified hardware from this project formed the starting point for our own work. The prototype and firmware were still basically functional but significant work was required to produce a reliable, reproducible version.

For the Twiddler in general there has been significant work measuring its productivity as a typing device and its learnability. [4] The Twiddler's longevity as a chording device – over 14 years since its original release – is a testament to its usefulness as a chording system.

OUR WORK

Our work consisted of three major parts: the electronics, the firmware, and the case.

Electronics

The Twiddler2 has a PS/2 interface, which could be converted to USB. The previous Georgia Tech team had used an existing Bluetooth accelerometer with a modified firmware to drive their Twiddler.

Figure 1: Two views of new Twiddler 3D models

We wanted to move to a more modern, reproducible solution, so we developed our own Bluetooth circuit board using the Bluecore4 chip. [2] Using standardized breakout headers to connect to the various inputs and outputs of the Bluecore chip, we not only created a Twiddler board, but a modular, reusable Bluetooth board (see appendix 1 and 2). An on-board EEPROM was added to enable easy programming, key-shortcuts, and reconnecting with previously used computers. We also developed circuit diagrams for all of the input keys of the keyboard. These input boards can easily be modified and printed by an individual with little extra equipment, allowing for the modification of the button layouts (see appendix 3).

In addition to the Bluetooth interface, a mini-USB port was placed on the Bluetooth board, allowing the unit to be interfaced with using USB instead of connecting wirelessly with Bluetooth. By making the device wireless, it now needed power. To recharge the battery from USB, we introduced a lithium-ion battery charging circuit.

Firmware

After developing a new board, we needed to create new firmware, while keeping the familiar Twiddler functionality intact. We started with the firmware developed by the previous Georgia Tech team. The original firmware was designed to act as a Human Interface Device (HID) eliminating the need to install special drivers. However, that firmware was lacking in several key areas.

The first problem we discovered was that the original Bluetooth firmware did not properly support standard keyboard authentication. On most versions of Windows that we attempted to connect with it was impossible to authenticate. We modified the firmware to support authentication fully using standard Twiddler key combinations.

Secondly, the original firmware had no support for USB. We added code so that the BlueTwiddler now authenticates as a HID-compliant keyboard and mouse. Keypresses register appropriately on the host computer, but are oddly slowed – see the discussion section.

Finally, the original firmware did not work under Linux. This was due to a variation in the authentication between Linux and Windows. We were able to update the BlueTwiddler to operate with both operating systems.

Case Design

Many people who have used the Twiddler2 will tell you that the case design leaves something to be desired. One of the advantages of the Twiddler2 is that it can be used in either hand. The downside of this approach is that it isn't very comfortable for either hand. Taking the liberty of making it a left-hand only device (for now) allowed us to shape it to fit more naturally in the users hand. We also were able to give the new case a form that followed the function of being a chording keyboard. The blocky design of the Twiddler2 didn't give much intuition as to the use, having to strap it on at just the right part of the hand to be able to use it, etc. Our more organic redesign of the case should allow new users to pick up typing on the BlueTwiddler quicker than ever before (see appendix 4 and 5).

DISCUSSION

Going from a prototype device to something reproducible has significant hurdles. Now that we know the issues involved, it should be possible to make this transition easier in the future.

Electronics

Getting prepared to print a circuit board was an intensive manual process. Building a viable circuit design, selecting and creating footprints for the necessary components, actually laying out those components and validating the layout is too long of a process to be undertaken lightly. For this reason we took special pains to design our circuit board to be as generic as possible. Hopefully, future projects can use our design in order to avoid the process of creating a new board.

Firmware

Using the BlueCore APIs was challenging. The documentation was inadequate and frequently inexplicable breakages would occur. Most of our process involved modifying example code and then attempting to recover from the resulting errors. Also, the BlueCore's debugging mode has odd performance effects which we did not discover until later in the project. In future devices, we would consider using a better understood processor that interfaces with the BlueCore chip and handles all non-Bluetooth operations. USB functionality is only partly effective because communication to the USB port seems to only work sporadically. We believe this must be some bug in the firmware, but the APIs in this area are limited and there are no good examples to work from. We hope to talk to CSR for future projects in order to resolve this issue and overhaul the USB functionality to bring the device up to speed for later works.

Case Design

Constructing a 3D-model for the case involved several steps. Interviews were conducted with both Twiddler novices and experienced users to get a feel for the functioning of the device. From there several sketched mockups were produced. Examination of the physical ergonomics of the hand gave the dimensions of the device, with consideration given to the future physical requirements of the circuit boards. Finally, the 3D-model was built. It is our intention to use this as the basis for a 3D-printed prototype that we can use in future interviews and analysis to settle on a finalized case design.

FUTURE WORK

While our prototype has several functional components, they do not fit together perfectly. Improving the case design and adjusting it to support the boards is a top priority. More work needs to be done on the firmware side to make sure the device behaves properly on a variety of systems, and testing needs to be done. Validating all the functionality of our Bluetooth board is also crucial. We would also like to refine the project so that others can use this same work to develop their own Bluetooth devices easily.

CONCLUSIONS

While we made impressive progress towards a Bluetooth-based replacement for the Twiddler2, our designs are not quite ready for full production. We feel that while a marketable device is not physically in our hands, our goals of producing new circuitry, including the Bluetooth functionality, and redesigning the case have all had significant success. We are excited about pursuing this project further and accomplishing our long-term goal of a fully reproducible Bluetooth Twiddler.

ACKNOWLEDGMENTS

Thanks to Peter Presti for helping us through the circuit design and Daniel Ashbrook for helping us get off the ground.

REFERENCES

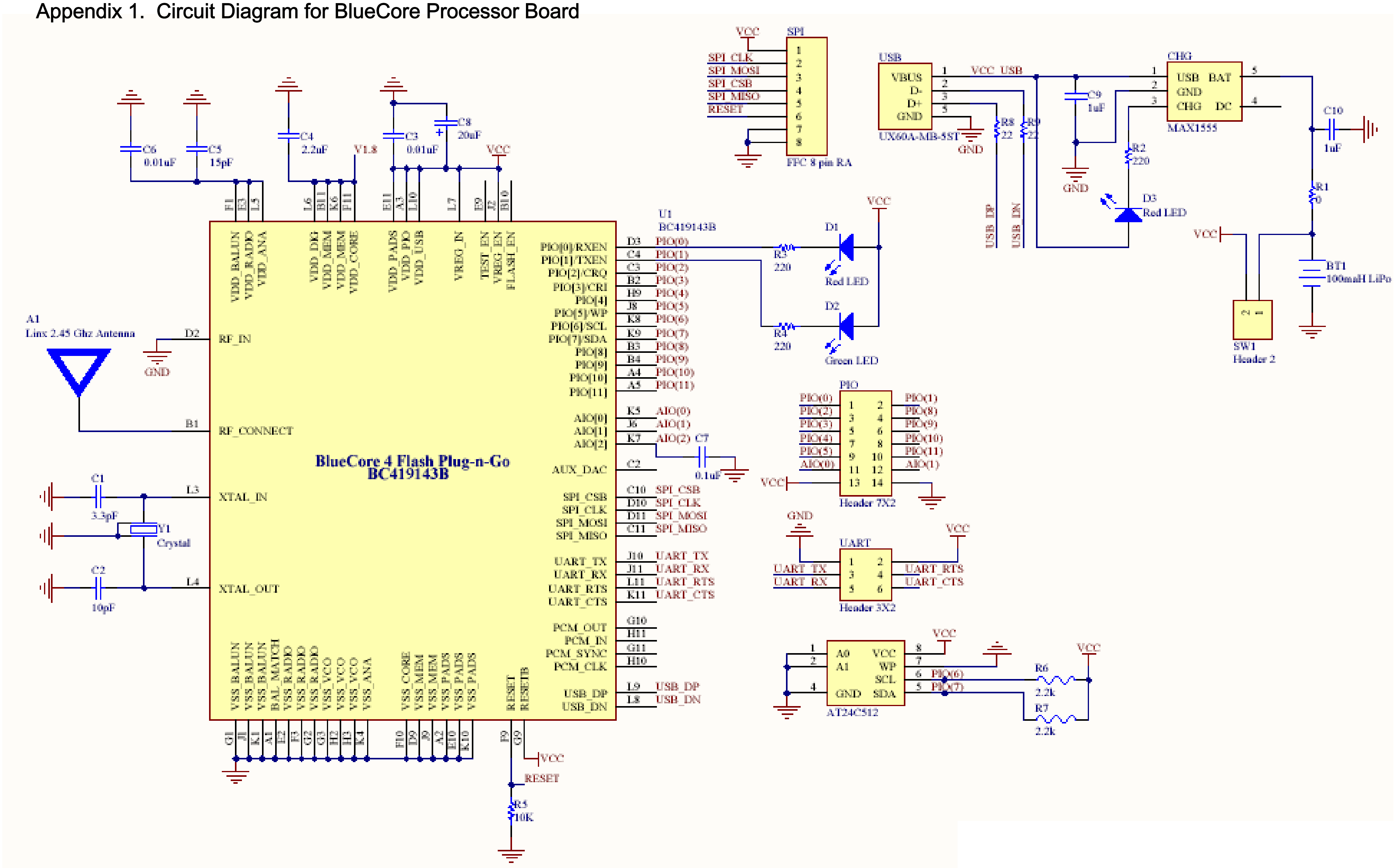
1. Charania, A. and Rogers, J. Bluetooth Twiddler (2006), accessed October 1, 2007; http://wiki.cc.gatech.edu/ccg/classes/7470/7470-f06/bluetooth_twiddler.

2. Cambridge Silicon Radio Support Website; http://www.csrsupport.com/.

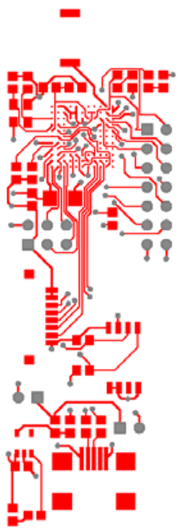
3. Handykey Corporation; http://www.handykey.com/.

4. Lyons, K., Plaisted, D., and Starner, T. Expert Chording Text Entry on the Twiddler One-Handed Keyboard. In *Proceedings of the Eighth International Symposium on Wearable Computers 2004* (October 31-November 03, 2004, Arlington, VA) IEEE, Los Alamitos, CA. 2004, pp.94-101.

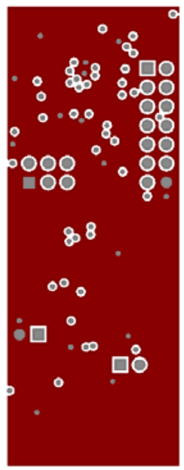
Appendix 1. Circuit Diagram for BlueCore Processor Board



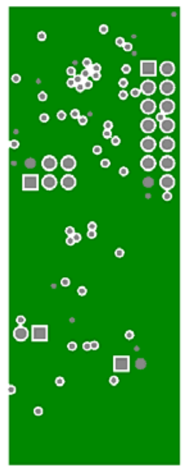
Appendix 2. Circuit Layout for BlueCore Processor Board



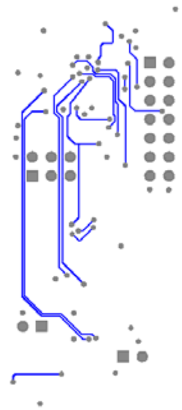
Top Layer



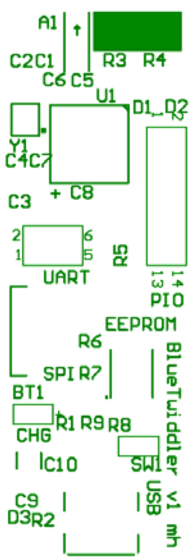
Mid-Layer (GND)



Mid-Layer (VCC)

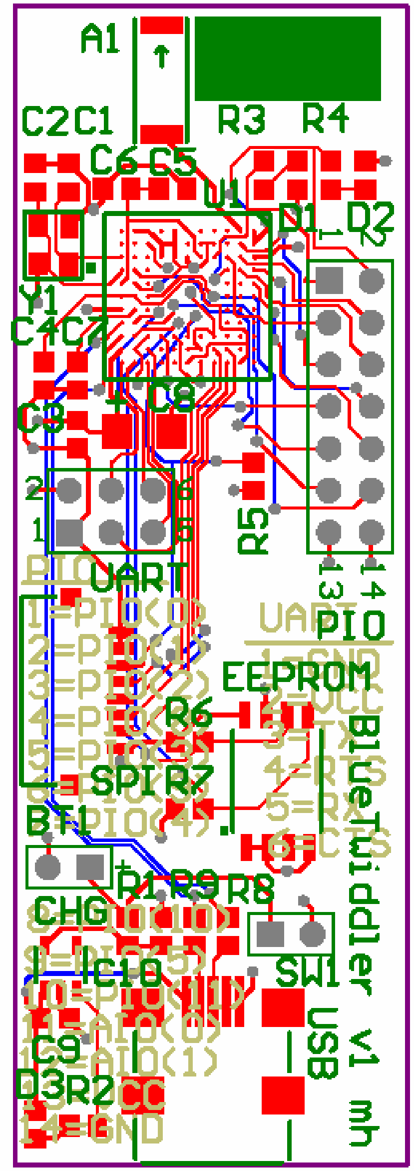


Bottom Layer



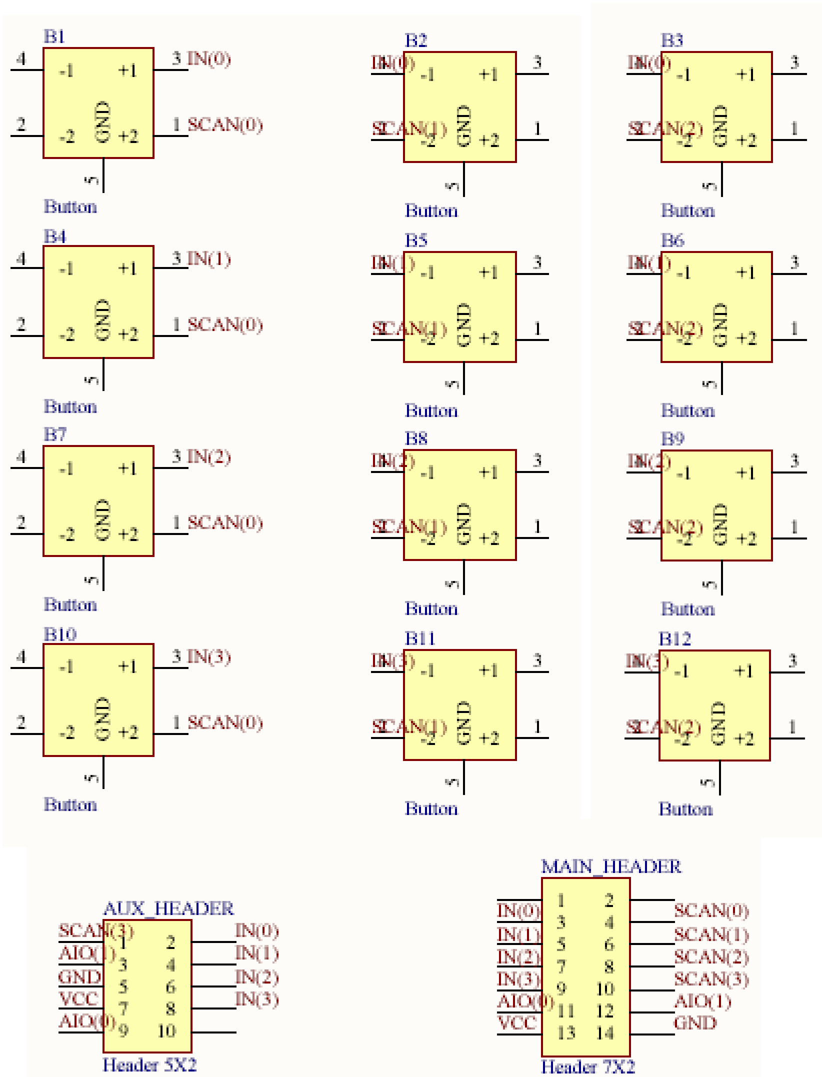
Top Screen

Bottom Screen

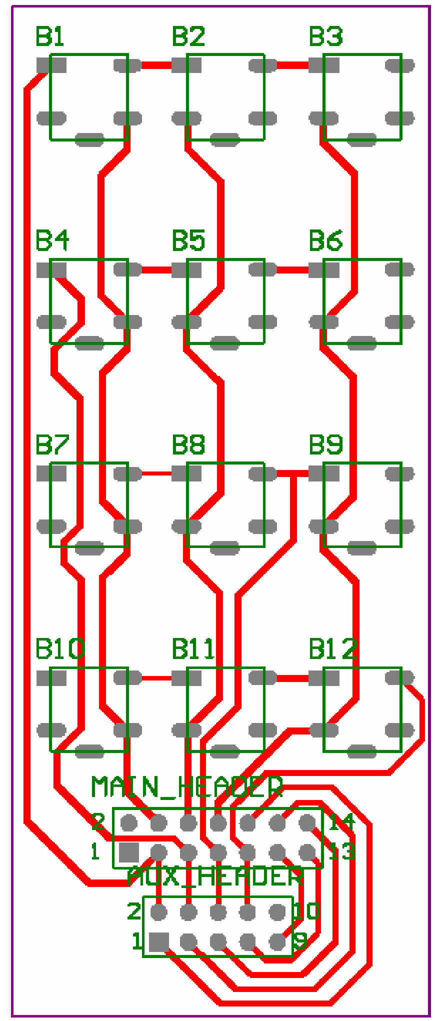


Composite View (no power planes)

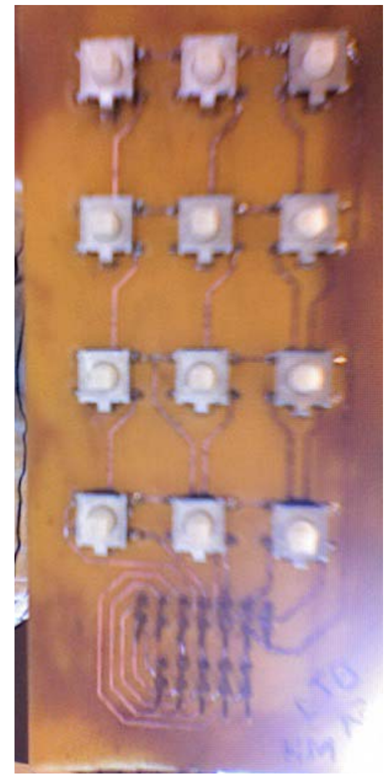
Appendix 3. Twiddler Button Board



Diagram



Layout



Printed Version

Appendix 4. 3D Case Renderings

Front Side

Back Side

Top

Appendix 5. Case Renderings showing hand position