Botball Hacking 101 (Part 1)
Jeremy Rand
Team SNARC (Sooners / Norman Advanced Robotics Coalition)
jeremy.rand@ou.edu

Botball Hacking 101 (Part 1)

1 Introduction

I've been involved with Botball hacking since 2004. I've either successfully hacked or gotten reasonably far trying to hack the SRF04 (Handy Board / XBC sonar), the XBC (collaboration with Fahrzin Hemmati) [1], the iRobot Create, the CBC (collaboration with Matthew Thompson and Braden McDorman) [2] [3] [4], and the Parrot AR.Drone. Outside of Botball, I've done significant hacking on the Nintendo GameCube and Wii game consoles [5] [6]. A lot of this research has been presented at GCER, but often in the Q&A after those presentations, I'm asked a more fundamental question: "How do you hack all this stuff?" I never really had an answer ready for that question (I was expecting questions about the specific hacks I had developed), but at some point I realized that there is indeed a certain skill set which facilitates hacking in general.

By popular demand, in *Botball Hacking 101*, I will attempt to describe some general procedures for hacking in Botball. The goal is to get more Botballers into the Botball hacking scene, which is critical for the scene's survival since all the major players (Jorge Villatoro, Fahrzin Hemmati, Matthew Thompson, Braden McDorman, and I) are no longer Botball students. My hope is that this will result in an awesome paper about *Hacking the CBCv3* being published at GCER 2013 by some new faces.

And now, the obligatory disclaimer:

DISCLAIMER: hacking anything in Botball carries an inherent risk of bricking equipment, and such bricks are not covered by KIPR's warranty. Usually, this risk can be mitigated by simply being very careful, but unexpected things do happen. If you're using your team's equipment, make certain that it is agreed upon who is responsible for repair costs, and that your team's success in competition will not be impacted by potential bricking of equipment. If this concerns you, that should be a hint that hacking is not for you.

2 Why Hack?

There are several reasons why gaining some hacking skills is a good idea.

2.1 Winning at Botball

This is the most short-term advantage. The teams I've worked with have directly benefited from hacking knowledge, both in tournament raw score and in Judges' Choice Awards. Having capabilities that no other team has gives you a serious advantage against those other teams.

2.2 Career Enhancement

If you work at a technology company, and you need some kind of hardware or software component, you may look around and find something from another company that *almost* works, but not quite. You can either build your own component from scratch and burn through your R&D budget, or you can take the other company's component and hack it to have that last feature you need. The other company gets business from you, you save money on your business, and everyone's happy (except your hacking-illiterate competitors who just went bankrupt due to inefficient use of their R&D budget).

2.3 Make the World a Better Place

For the big-picture people among you, hacking skills allow you to help improve the world in ways that others can't. For example, the hacker group Telecomix [7] has used their knowledge of communication systems to assist in the Arab Spring by getting information across borders of countries whose governments are trying to suppress free communication (IRC chat to Twitter relays, proxies and mirrors, dial-up modem pools, fax services, and ham radios). If you saw videos of Egypt after Hosni Mubarak hit the Internet kill-switch during the revolution, there's a good chance that you saw them courtesy of Telecomix.

As another example, The Tor Project develops software which allows human rights workers and ordinary citizens to communicate freely and securely without being tracked by repressive governments. Tor is heavily used by Iranians and Syrians who don't want to be subject to government oppression due to their Internet speech. Developing Tor definitely requires some hacking skill, particularly when a repressive government suddenly deploys a new wiretapping system which aims to block Tor users, and the users need a fix immediately.

Note that I'm not saying that all Botball hackers should join Telecomix or Tor when they graduate (doing so tends to invite death threats from dictators... not kidding). But it's an example of the good that can only be done for the world by hackers.

2.4 Freak Out Your Grandma

In addition to the practical benefits of hacking knowledge, it's also a great way to freak out your computer-illiterate grandma when she overhears you talking about hacking CBC's and thinks you're stealing credit cards or something equally nefarious.

3 Use Google

Google is your best friend when trying to hack something, and has been a prerequisite for all of the hacking I've done. Using Google effectively takes a lot of practice, but there are some key tips which will help.

- Look up model numbers, part numbers, and any phrase written on a circuit board or (if you have source code) in source code comments. You may find datasheets, reverse engineering analysis, or even previously unknown source code.
- Look up manufacturers, companies or people who are known to have been involved. They may have documentation posted.
- Use key phrases which you find with Google, and Google for those.
- Add keywords such as “hacking”; “disassembly”; “source code”; “firmware”; “Linux” (the CBC and AR.Drone both run the Linux operating system); “Telnet” or “SSH” (the two remote login systems used by Linux); “video4linux2” or “v4l2” (the camera API used by Linux); and “Git” or “SVN” (common source code repositories). Being able to guess these keywords is a good skill to develop.
- If your search isn’t successful, try another search. There is almost certainly some good information on Google for whatever you’re looking up.

4 Find and Read Source Code

Hacking doesn’t necessarily require source code for the device you’re hacking, but source code does facilitate the process, and often enables more complex hacks to be developed in less time. For example, Matthew Thompson and I were independently trying to hack the XBC in 2008. I found the XBC firmware’s source code, while Matthew used an ARM disassembler. In the end, I successfully implemented several hacks, and published a 13-page paper at GCER 2008. Matthew was largely unsuccessful. It’s not that I was better at hacking than Matthew; that’s completely untrue. Matthew got much further with his disassembly wizardry than I would have given the same tools. The difference was that I had more information available, because I had the source code, so I needed less work to get results. Moral of the story: always look for source code.

5 Use the Search Feature

Computers have search features, and once you have some source code, searching through it can be a timesaver. For example, if you see a function called from one file, and you want to analyze it further, search for that function name in other files. The same goes for variables of interest. Searching for numbers can also be useful in some cases; for example, I was curious how to change the resolution of the CBC camera, and I found it by searching for the numbers 160 and 120 (the default CBC camera resolution).

6 Use Notepad

If you happen to find a file whose function you don’t understand, your first course of action should be to open it in a text editor such as Notepad. This will often help you see what the file does, and may assist in your hacking activities.

For example, the CBC’s userhook0 files seem quite mysterious, until you open them in Notepad.

If you've used Linux, you can easily tell by looking in Notepad that the userhook0 is a simple shell script. If you can't tell by looking, try Googling for parts of it.

The userhook0 begins with this line:

```
#!/bin/bash
```

Googling this line gives a quite large number of hits, all of which are about shell scripts.

As another useful tool, try a programmer-oriented text editor such as Programmer's Notepad [8]. Programmer's Notepad can often auto-detect the filetype of text files, and has useful syntax highlighting features.

7 Find Documentation

Usually software and hardware have documentation. The technical, development, and repair documentation is probably separate from the end-user documentation which you've been given (if documentation came in your Botball kit, it's probably not what you're looking for). Documentation serves different benefits than source code, and is desirable regardless of whether you've found source code. In particular, pinout diagrams, compilation instructions, algorithm descriptions, and easter eggs are all very valuable things you can often find in documentation. Remember, if you think you know everything there is to know about something's workings, you're almost certainly wrong.

8 Go to the Source Company (Not Just KIPR)

KIPR is somewhat unique in that they develop their products in-house. However, even KIPR typically uses existing products as starting points. For example, the XBC was distributed by KIPR, but it was heavily based on the XRC, a product manufactured by Charmed Labs. As a result, XBC hackers gained a lot of knowledge by checking Charmed Labs's website [9]. But why stop there? The XRC utilized the Game Boy Advance, and fittingly, XBC hackers found useful information on GBA hacking websites like GBADev [10]. Similarly, the CBC was heavily based on the Chumby, manufactured by Chumby Industries, and (surprise) CBC hackers found large quantities of technical documentation on the Chumby website [11].

9 Don't Be Afraid to Talk to Staff

As is inevitable, sometimes hackers will get stuck. In 2006, I was having massive trouble with the XBC firmware, getting hundreds of confusing errors when trying to build it. Finally, I asked KIPR's programmer, Jorge Villatoro (a former Botball hacker before being hired by KIPR), for help. He didn't get mad at me for wasting his time; on the contrary, he seemed impressed that Botball students were interested in this kind of thing. I explained over AIM when I had done,

and he correctly guessed what my problem was within a few minutes. (I had copied the wrong folders between the official Xport devkit release and the latest code that Jorge was working on.) There was no documentation on how this was supposed to work, and had I not asked Jorge, I probably would have been stuck for weeks or months (if I had figured it out at all).

Similarly, when I couldn't understand the function of some Charmed Labs library for the XBC, I asked Charmed Labs CEO Rich LeGrand for help (this happened probably 10-15 times), and each time, he was happy to answer questions.

The message to take away is that if the information you want isn't findable with Google and reading the existing documentation, asking someone with experience is a good choice. Worst case, they'll ignore your post or say they don't support what you're doing, which doesn't make you any worse off, but there's a good chance they'll be able to assist. Remember, they're probably making money off people using their product, and they want to keep users happy.

On the other hand, don't get impatient if you don't get help promptly. Hacking is not officially supported by KIPR, so while KIPR's programmers are often willing to help you, their ability to do so is often dependent on their work schedule. I've found that August and September are the most productive times to ask for help. Whatever you do, do **not** demand prompt help or repeatedly pester people; this is an easy way to make people ignore or yell at you.

10 Direct Your Questions Appropriately

If a company is aiming their product at consumers rather than hobbyists, you may be better off directing your inquiries to a hacking community (e.g. GBADev [10], Robot Reviews [12], AR Drone Flyers [13]). In particular, if a company gives excuses about confidentiality or copyrights, that shouldn't deter you from hacking, but it should give you a hint that other hackers will be able to tell you more than the company will.

11 Collaborate with Other Teams

As tempting as it is to try to keep your hacking top-secret until a GCER release, it can be very beneficial to find another team with whom to collaborate. Bouncing ideas off someone who has a differing area of expertise is a good way to solve problems which might otherwise take a large amount of time to solve on your own. Both times that I collaborated with another team on a hacking project (La Jolla for the XBC and Nease for the CBC), the alliance turned out to be the key to a successful hacking paper.

However, be sure that the other team isn't just leeching off of you. It's not a good situation if your collaborator isn't contributing his/her fair share of the research.

12 References

[1] Jeremy Rand and Fahrzin Hemmati. Hacking the XBC Firmware: Programming the XBC in Standard C++ (Parts 1 and 2). Proceedings of the 2008 Global Conference on Educational

Robotics. July 2008.
[2] Jeremy Rand, Matt Thompson, Braden McDorman. Hacking the CBC Botball Controller: Because It Wouldn't Be a Botball Controller if It Couldn't Be Hacked (Parts 1 and 2). Proceedings of the 2009 Global Conference on Educational Robotics. July 2009.
[3] Jeremy Rand, Matt Thompson, Braden McDorman. CBC Hacking 2010 (Parts 1 and 2). Proceedings of the 2010 Global Conference on Educational Robotics. July 2010.
[4] Jeremy Rand. CBC Hacking 2011: Vision Enhancements and Sensor Speedups (Parts 1 and 2). Proceedings of the 2011 Global Conference on Educational Robotics. July 2011.
[5] Jeremy Rand and Martin Michelsen, with Dr. Deborah Trytten. GCARS-CS and GeckoTunnel: Using Video Game Enhancement Technology to Enable Online Play of Modern Offline Multiplayer Games. 2011 University of Oklahoma Undergraduate Research Day Conference, March 2011.
[6] Jeremy Rand and Martin Michelsen, with Dr. Deborah Trytten. GCARS-CS and GeckoTunnel: Using Video Game Enhancement Technology to Enable Online Play of Modern Offline Multiplayer Games (New Developments for 2012). 2012 University of Oklahoma Undergraduate Research Day Conference, March 2012.
[7] Wikipedia Contributors. Telecomix. Wikipedia, The Free Encyclopedia, https://secure.wikimedia.org/wikipedia/en/wiki/Telecomix . May 2012.
[8] Simon Steele. Programmer's Notepad. http://www.pnotepad.org/ , Retrieved June 2012.
[9] Charmed Labs. http://www.charmedlabs.com . August 2008.
[10] GBADev. http://www.gbadev.org/ , May 2012.
[11] Chumby Industries. http://www.chumby.com/ , June 2012.
[12] Robot Reviews. http://www.robotreviews.com/ , June 2012.
[13] AR Drone Flyers. http://www.ardrone-flyers.com/ , June 2012.

13 See You in Part 2!

That's all I could fit into Part 1. See you in Part 2!