

Beginning Embedded Electronics for Botballers Using the Arduino

Matthew Thompson

Allen D. Nease High School

matthewbot@gmail.com

1 Introduction

Robotics is a unique and multidisciplinary field, where successful solutions emerge from tricky combinations of software, mechanics, and electronics. In Botball, students gain skills in all three areas as they construct robots to complete each year's challenge, although in the competition a large disparity exists between the level of student involvement in electronics versus their involvement in mechanics and software. For a variety of practical and safety reasons, students are not allowed to modify or create their own electronics^[1], and are instead required to use the largely plug and play solutions that arrive in the kit, in the form of a prebuilt controller and preselected, pre-wired actuators and sensors. While these restrictions make sense to ensure the safety of Botball participants and keep costs low, learning electronics basics outside of the Botball competition can be a useful part in any Botballer's robotics education.

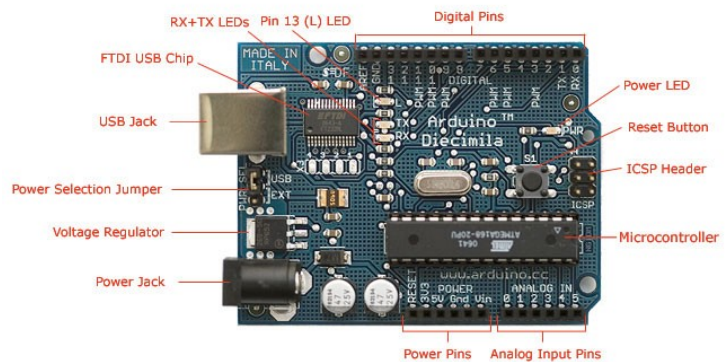
2 Embedded Electronics Primer

Embedded Electronics is a field of electronics where circuits are augmented with a small programmable microprocessor^[2]. They're found nearly everywhere, from nearly any robot, to household appliances like thermostats or microwaves, and even to devices like MP3 players. Although it may seem intimidating to have to how to build both the circuit portion and the microcontroller portion of an embedded design, today many useful shortcuts enable hobbyists to quickly create fun and interesting gadgets armed only with a little patience and a willingness to experiment, and without a formal education in Electrical Engineering or Computer Science.

2.1 The Arduino

The first massive shortcut in the hobbyist's arsenal is the Arduino microcontroller. It's a very inexpensive, easy to use microcontroller with many useful features built in, like power regulation and a USB to serial chip^[3]. All of the pins of the microcontroller are brought out to reusable female headers, which is excellent for prototyping. It also comes with a simple but effective IDE which eliminates the hassle of setting up compilers and programmers for its powerful Atmel Atmega328 microprocessor. The Arduino is virtually plug and play; all that's needed is a DC power supply and you've got the microprocessor portion of your embedded circuit design finished.

The controller itself, however, is no good without proper programming. Fortunately, most Botballers have had some exposure to programming on the CBC, either through programming a bot themselves, or from attending workshops or reading workshop slides. Many of these concepts carry over, but a new way of thinking is sometimes required.



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

2.2 Going from CBC to Arduino

This biggest paradigm shift needed when transitioning from Botball programming to Arduino programming is the level at which your code addresses the hardware. The CBC consists of a microcontroller like the Arduino, but it contains pre-wired devices to its microprocessor, instead of simply exposing the microprocessor pins and enabling you to attach your own devices like the Arduino. Because the creators of the CBC programming environment know exactly which devices its microprocessor is connected to, they can provide useful functions like `fd()` and `set_servo_position()`.

The Arduino, however, can be hooked up to a multitude of different devices through its simple analog and digital pins. Because it is so flexible, the base software includes little additional functions beyond speaking certain serial protocols, and setting outputs and reading inputs from its pins. To use many devices like Botball-style analog sensors and switches, reading the value on a pin is all that's really needed.

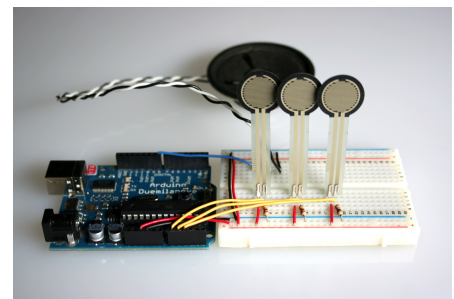
For more complex hardware, that might require a serial protocol, multiple inputs and outputs, or complex timing, like a motor controller or a digital accelerometer, it may be necessary to delve into the datasheet of the device and learn how to communicate with it properly. However, owing to the extreme popularity of the Arduino across the Internet, odds are actually fairly high someone somewhere has already written a driver for the device and published it online. This is the true power of the popular, open source platform, as even complex, feature-rich devices often times remain as plug and play as the old mainstays of analog sensors and buttons.

3 Circuit Design

The next half of embedded design, the construction of the circuit attached to the microcontroller, can also seem a daunting process. Once again however, there are tools and online resources that remove a lot of the difficulty in designing and constructing circuits.

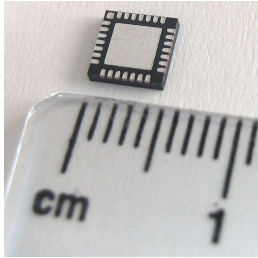
3.1 Basic Tools Needed

The “standard” electrical skills like soldering, schematic design, and board layout aren't actually needed to construct simple Arduino embedded circuits. Arduino users are usually able to get by with a simple breadboard. A breadboard is a large rectangular grid of little holes, which are electrically conductive in small groups, usually individual rows of 5. The breadboard is excellent for prototyping and like the female headers on the Arduino, is entirely re-usable and adjustable. Making circuits on the breadboard is as simple as placing components and wires directly in the holes such that the conductive groups and wires form the desired circuit.



In the above example, three large flex sensors and a speaker are attached to the Arduino through the white breadboard. The touch sensors are read as analog inputs while the speaker is driven by generating by moving an output pin up and down at high speeds, creating a square wave. It could be programmed to make a simple alarm system, or perhaps act as a simple musical instrument.

3.2 Breakout Boards



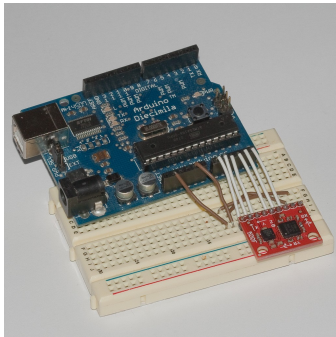
Many of the most exciting new devices that are just now becoming very affordable are unfortunately surface mount devices. This means that they are made on incredibly tiny ICs that don't fit on a breadboard, and are difficult and time consuming to use without making an expensive PCB. Things like gyroscope or accelerometers, wireless transmitters, or even motor drivers often times only come in a surface mount package, an obstacle previous insurmountable to the average hobbyist or electronic newbie.

Today, however, many companies and individuals sell small circuit boards known as break out boards. A breakout board is a professional PCB that holds a surface mount chip, and exposes its pins in standard breadboard spacing, and usually also contains all the additional components needed for the chip to work, like capacitors or resistors. Using a surface mount chip on a breakout board is not much harder than using the simple speaker or touch sensors pictured above. This author has had great success with Sparkfun^[4] breakout boards.



*ITG-3200 gyro
breakout by Sparkfun*

3.3 Putting it all together



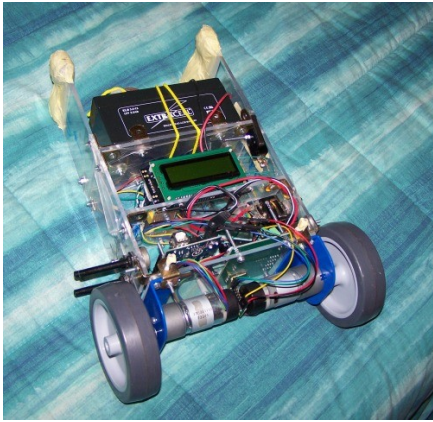
Between the Arduino, breadboard and the breakout boards, the previously complex task of designing and constructing a working circuit has been simplified considerably, almost resembling an electrical connect the dots. All that is required now is to carefully route the Arduino pins to the breakout board pins using the breadboard and wires, and provide the breakout board with a power and ground connection. Armed now with the ability to connect modern SMD devices to the easy to use and easily programmable Arduino, the sky is literally the limit for even the most inexperienced electronics hobbyist, as the following examples show.

4 Example Arduino Projects

The Internet is chock full of creative Arduino based projects. Here's a few cool ones of varying electrical and software complexity.

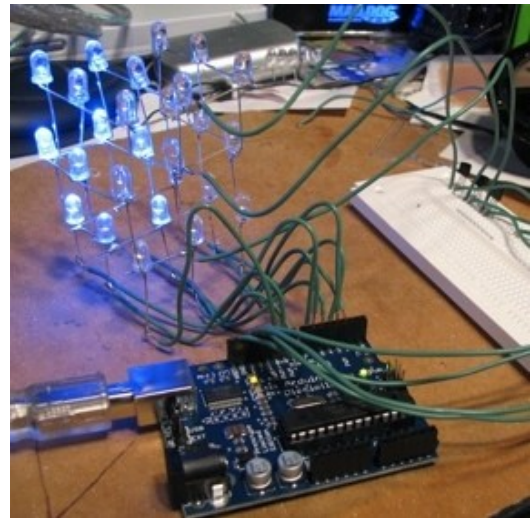
To the right is an example of wiring a Wii Nunchuck to an Arduino^[5]. The Nunchuck contains an accelerometer, two-axis joystick, two buttons, and a small microprocessor that can provide all of the sensor readings over a common serial protocol known as TWI or I2C. The Arduino has built in support for TWI, so you can easily communicate to the Nunchuck using its simple binary protocol. There is also a `WiiChuckLibrary`^[6], which gives easy high level access to the Nunchuck's sensors.





Another cool, though far more complex Arduino based project is this balancing robot^[7]. Though this robot looks complicated and did require both soldering and the design and construction of its frame, the author made great use of the Arduino and a SparkFun breakout board featuring a gyro and accelerometer, as well as an pre-built motor driver board. The Arduino keeps the robot upright by estimating the robots orientation using the gyro and accelerometer sensors, then running a control loop and outputting commands to the motor driver. Once again, this is definitely a complex robot, but the Arduino and breakout boards handled much of the electrical complexity.

A popular and cool looking project that has been undertaken by many Arduino users on the Internet is the LED cube. This cube^[8] here is a simple 3x3x3 cube of LEDs, which are powered by the three transistors on the breadboard. The Arduino drives the transistors, and can control each LED independently through some clever multiplexing. Once the Arduino can control the LEDs, the next step is to program in any variety of fancy light strobing animations. Even a 3x3x3 cube can produce dazzling spinning, pulsing, and rotating animations, though more advanced users build 5x5x5 and 8x8x8 for even greater possibilities.



5 Conclusion

Taking the leap from pre-constructed Botball electronics and controllers into the world of embedded electronics isn't very difficult and can be both educational and fun. The Arduino, and the massive amount of code available for it on the Internet, combined with inexpensive break out boards for nearly every kind of chip imaginable gives a large array of possibilities to even a total beginner, who will get up to speed quickly and soon be capable of making his own cool projects.

6 References

- [1] KIPR. Botball 2010 Teachers' Workshop Game Review v1.0.1. 2010
- [2] SparkFun Electronics. Beginning Embedded Electronics. http://www.sparkfun.com/commerce/tutorial_info.php?tutorials_id=57, 2010
- [3] Arduino Team. Arduino Duemilanove. <http://arduino.cc/en/Main/ArduinoBoardDuemilanove>, 2010
- [4] Sparkfun Electronics. <http://www.sparkfun.com/>
- [5] Read wii nunchuck data into arduino <http://www.windmeadow.com/node/42>
- [6] Arduino Playground: <http://www.arduino.cc/playground/Main/WiiChuckClass>

[7] Arduino Forum - Self Balancing Robot

<http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1225283209>.

[8] Blink on, Blink off <http://markw.us/post/1666>