# Project 2:
# "Build Your Own Arduino Clone"

PCB Drawing (and send to fab.) Due Tuesday 3/18
Breadboard Test Due Thursday 3/20
Completed PCB Due Thursday 3/27
(Late Penalties will Apply)

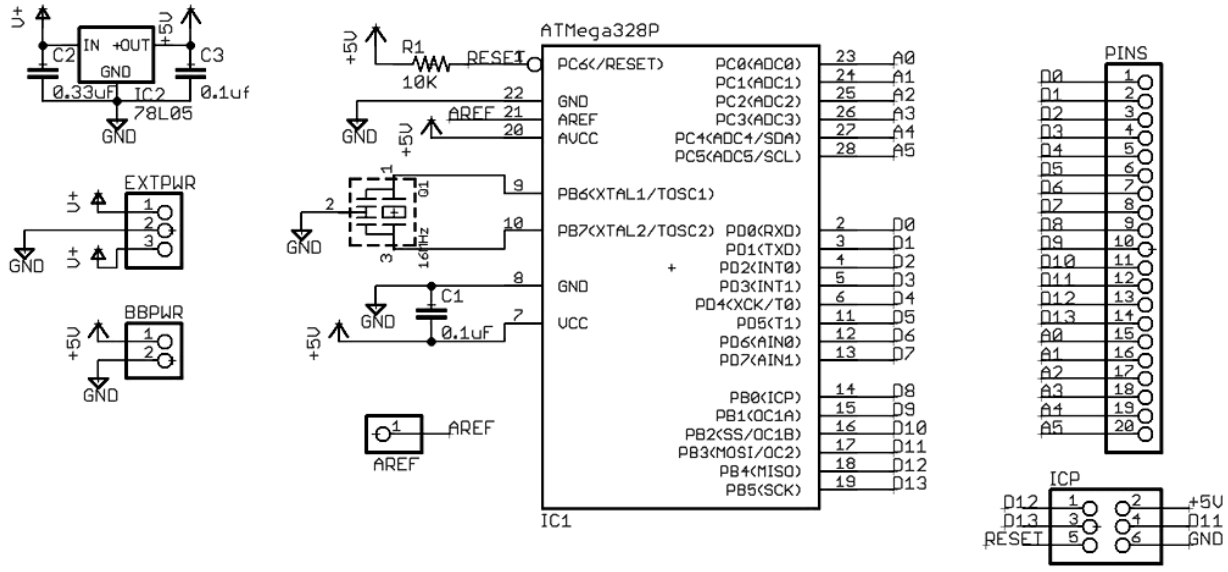
**HCI-833 Applied Gadgets, Sensors and Activity Recognition in HCI**
**Spring 2014**

---

**Overview**

This project is to build a slightly simplified, but fully functional Arduino clone using a custom printed circuit board (PCB) that you design. The project will have three parts. You will test the circuitry on your breadboard and use your current Arduino to burn the Arduino bootloader into the ATMega328P chip for your new Arduino. You will specify a custom PCB for your project, have it manufactured, and finally assemble your circuit on it to create your own Arduino clone. (Note due to delays likely associated with getting your board fabricated, you will do these steps out of what would be the normal order: specifying the board first and then testing your circuit.)

The circuit for your project should match the following schematic:




Parts List

| | |
|---|---|
| IC1 | ATMega328P-PU (AVR micro-controller) |
| IC2 | 78L05 (5v voltage regulator) |
| Q1 | 16MHz Resonator (with internal caps) |
| C1 | 0.1µF capacitor |
| C2 | 0.33µF capacitor |
| C3 | 0.1µF capacitor |
| R1 | 10KΩ Resistor |

**Breadboard Test**

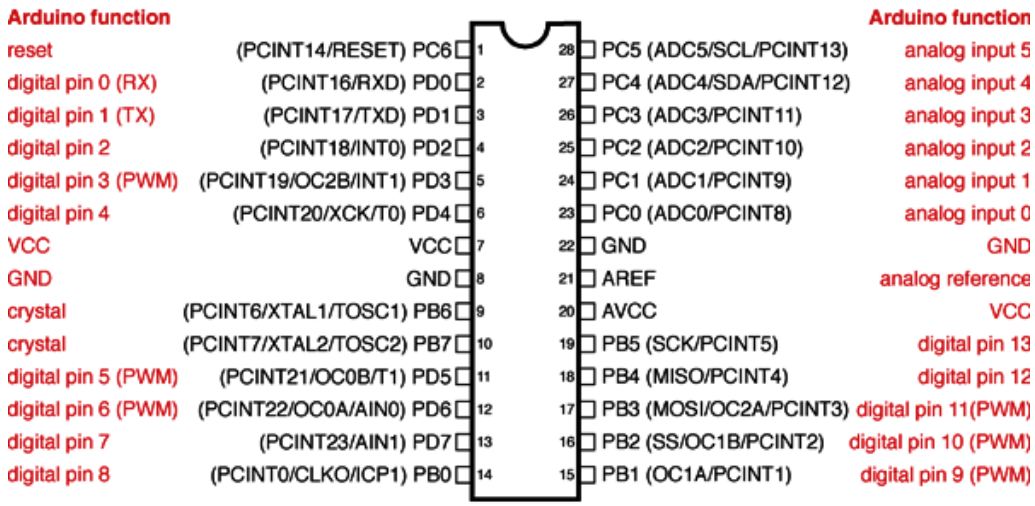
This part of the project is to test your circuit on a breadboard, where you will burn the Arduino bootloader onto the ATMega chip and verify that it works by loading a simple

Arduino test program into it. You should create a circuit on your breadboard next to your existing Arduino Nano which matches the schematic diagram above (but without the header pins: PINS, ICP, EXTPWR, and BBPWR).

Test that the voltage regulator circuit works by applying 7v or more (perhaps with a lab power supply or a 9v battery) between the "V+" connection and ground. Then verify with a multi-meter that you see 5v between the "+5V" connection and ground. Alternately, you can attempt to power your existing Arduino Nano with this circuit (connect the outputs to your breadboard power bus, but don't connect it to the USB port at the same time). Once you have tested your power circuit, you can remove that and power the next step from the breadboard bus driven by the USB connection to your existing Arduino Nano.

Next you should burn the Arduino bootloader into your ATMega chip. You will do this using your existing Arduino Nano as an In-System Programmer (ISP) which is capable of loading programs directly into the flash memory of the ATMega chip. Do this with the following steps (note: similar instructions with pictures can be found here http://arduino.cc/en/Tutorial/ArduinoISP):

1. Wire two LEDs (and 470Ω current limiting resistors) to your Nano, one driven by D9 and the other by D7. You may wish to test that you have wired the LEDs in the proper direction with a simple "blink" program that flashes them.
2. Load the "ArduinoISP" firmware sketch (found in "File>Examples") into the IDE, then load it into your Nano like any other program. If this is successful you should see the LED attached to D9 fade in and out with a "heartbeat" pattern.
3. Wire the main circuit (involving IC1, Q1, C1, and R1 only) on your breadboard as shown in the schematic above.
   Notes:
   - IC1 (the ATMega328P µcontroller chip) has the following physical pinout (see : http://arduino.cc/it/Hacking/PinMapping168):




   You should take note of where pin 1 is found relative to the half circle indentation at the top of the chip, and the "U" numbering pattern around the chip.

- The resonator (Q1) contains internal capacitors (as indicated inside the dotted lines of the component in the schematic). The center pin of the resonator package should be connected to ground and the outer two pins to IC1 pins 9 and 10.

4. Place a 0.1μF capacitor between the Nano RST pin and Ground.
5. Wire the following connections between your Nano and the AVR chip:

| Nano Pin Name | AVR Pin Number (Name) |
|---|---|
| D10 | 1 (Reset) |
| D11 | 17 (MOSI) |
| D12 | 18 (MISO) |
| D13 | 19 (SCK) |

6. In the IDE, select "Tools>Board> Duemilanove w/ ATMega328", then "Tools>Burn Bootloader". This will burn the bootloader into your ATMega chip. During this process the Tx/Rx and D13 LEDs on your Nano will flash and the LED you connected to D7 should be lit continuously. When this process is complete, the D7 LED should go dark and the D9 LED should return to its heartbeat display (fading in and out).

To test that your ATMega chip is now acting as an Arduino, do the following:

7. Place an LED and current limiting resistor between pin #19 (which should now be the equivalent of Arduino D13) and ground.
8. Load the "Examples>Basic>Blink" sketch into the Arduino IDE.
9. Leaving the "Tools>Board" setting on "Duemilanove w/ ATMega328", select "Tools>Programmer>Arduino as ISP".
10. Invoke: "File>Upload Using Programmer". LEDs should light similar to the way they did when you burned the boot loader. On completion, the LED attached to pin #19 of your chip should blink.
11. To double check that you didn't just accidentally load the "Blink" program into your Nano rather than the AVR chip, temporarily remove the wire from Nano D13 and the chip pin #19. The LED attached to your chip should continue to blink.

At this point your ATMega chip should be able to load and run any Arduino sketch using steps 9 and 10 above. Once the chip is programmed with a sketch, it can run independently as long as you have C1, Q1, and R1 attached and provide a regulated 5v power supply.

Remember to set the board to "Duemilanove w/ ATMega328" when you are loading with "Upload Using Programmer", but to set it back to "Arduino Nano w/ ATMega328" if you want to reprogram your Nano for another purpose (using the normal "Upload" command). The "Tools>Serial" setting should remain the same throughout.

**Breadboard Test Turn In (Due Th 3/20) (NOTE: this is the second thing you turn in!)**
This part of the project will be graded with a peer demo (just like a *μ*Project). Demo your code by showing correct operation of the candle1 code from the first day of class. Turn a *μ*Project Peer Demo Certification Sheet in no later than Th 3/21 at the start of class. Note

that unlike $\mu$Projects which have no late penalty, a late penalty of 10% per day (with a maximum of 30% off) will be charged for late assignments.

**Printed Circuit Board (NOTE: the first ½ of this is due first: Tuesday 3/18)**
This part of this assignment is to use the Eagle Schematic and PC Board editing program to specify a custom Printed Circuit Board for your circuit, have it manufactured, and then assemble your board.

A slightly limited (but highly functional) copy of the Eagle system is available free for non-profit use from CadSoft at http://www.cadsoftusa.com/downloads/. You should download and install this program now.

A lecture section will be devoted to how to use the Eagle system and how to have boards manufactured by a PCB fabrication house. Once we have covered the system, you should use it to first enter the schematic at indicated above, then to draw a PC Board for it. These should be both turned in on Blackboard and sent to be fabricated no later than the start of class Tuesday 3/18. Late penalties will be as indicated above. Next, you will solder a socket for your ATMega chip and the other components of your circuit onto the board, place the chip in the socket and demonstrate that it functions as an Arduino by running the candle2 code from the first day of class on it.

**PCB Construction**
After your PC board has been manufactured you should visually inspect it for any problems (like traces which are close, but don't actually connect to pads). It may be necessary to cut through traces or solder extra wires to the board to "patch up" errors. You should solder your components to it only after you have carefully inspected it.

You should not directly solder your ATMega processor chip to the board. Instead solder on a socket then place the chip in the socket. Pay careful attention to ensure you solder the socket with the notch in the right direction and especially that it is on the correct side of the board (this second error is more common than you might think and is hard to recover from, so double check this). Similarly ensure that you have the voltage regulator in the correct orientation on the right side of the board before you solder it in place. The remaining components are orientation independent.

The PINS, BBPWR, and EXTPWR components in the circuit are "header pins" which are spaced at 0.1" and can be pressed into the holes of a standard breadboard. The PINS connector is intended to be placed along the bottom of the board and connect the 20 Arduino pins (D0…D13 and A0…A5) from your board to the breadboard. The BBPWR connector is designed to connect to your breadboard power rails. This pair of header pins should be placed at the end of two wires sized to allow them to reach the breadboard rails. You may wish to find red and black (or blue) wire for this purpose since it will be critical that you don't plug BBPWR in backwards. Finally, the EXTPWR header pins are for "external" power. These are designed to accept an external power source such as a 9v battery to drive your circuit. (Note that the EXTPWR connector is wired in a (+) (-) (+)

configuration so that – as long as you have a matching (+) (-) (+) wired connector – you can't plug it in backwards.)

The ICP component in the circuit is intended to be header sockets. These are to be connected back to your Arduino Nano which will serve as an In-System-Programmer to load new programs (connecting to the same pins and extra capacitor as you used in Part 1, steps 4 and 5, of this project).

The AREF component listed in the circuit can normally be left out (left as simply a hole in the circuit board, with no header pins or sockets installed) since it would rarely be used and a pin or socket can always be added later.

**Turning In Your Final Result (Due Tues 3/27)**

The final result of the project will also be graded with a peer demo form. Much of this assessment will simply be an indication that the board works. However, part of this assessment will be an indication of whether the board worked as originally designed, or whether it had to be "patched" before it worked. Turn peer demo forms in no later than Tues 3/27 at the start of class. As indicated above, a late penalty of 10% per day (with a maximum of 30% off) will be charged for late assignments. However, anyone who finds that they must have a second version of their board fabricated will be given an additional 7 days of grace period with no late penalty to allow for manufacturing time.