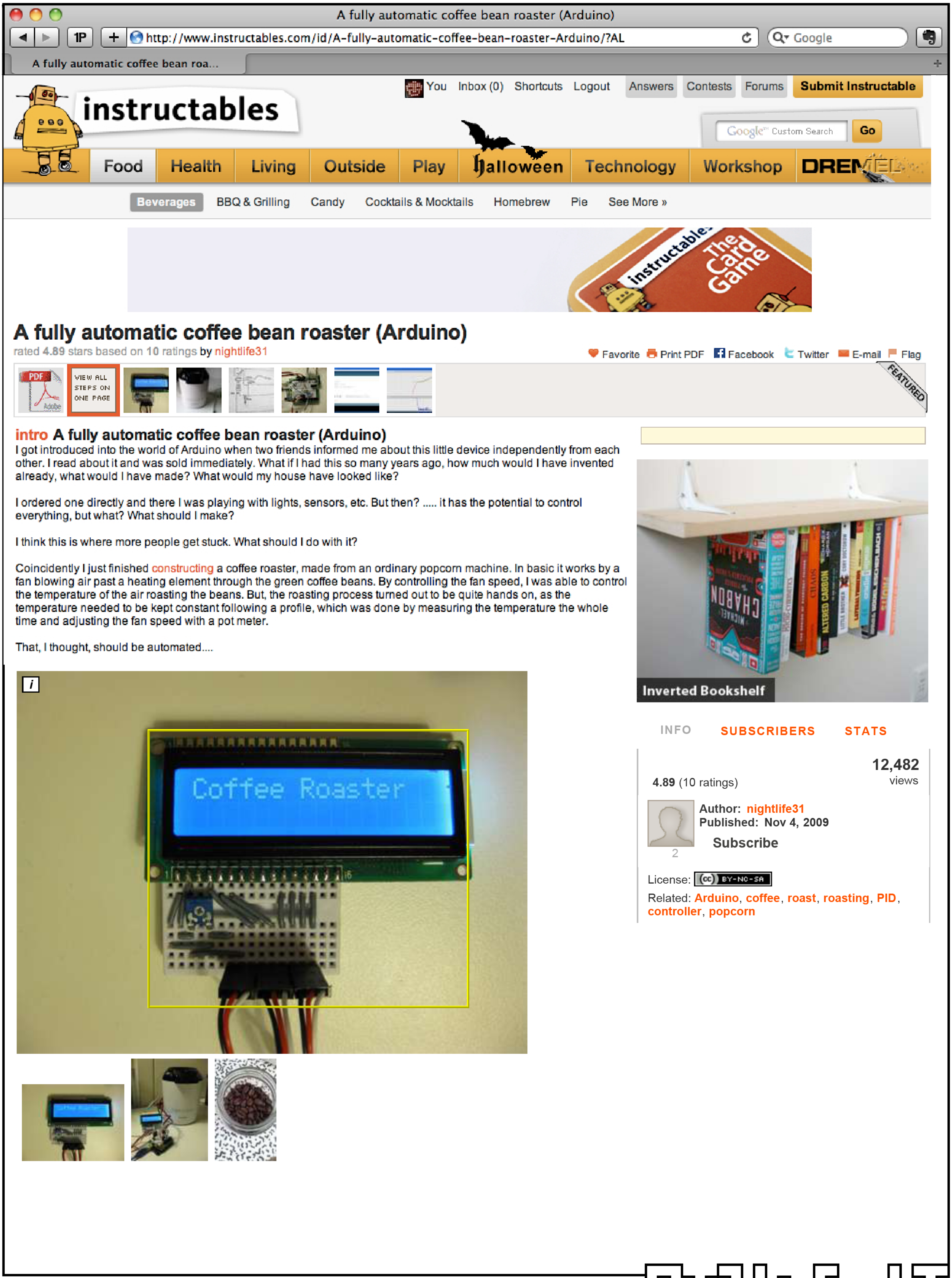


A fully automatic coffee bean roaster (Arduino)

rated 4.89 stars based on 10 ratings by nightlife31

intro A fully automatic coffee bean roaster (Arduino)

I got introduced into the world of Arduino when two friends informed me about this little device independently from each other. I read about it and was sold immediately. What if I had this so many years ago, how much would I have invented already, what would I have made? What would my house have looked like?

I ordered one directly and there I was playing with lights, sensors, etc. But then? ..... it has the potential to control everything, but what? What should I make?

I think this is where more people get stuck. What should I do with it?

Coincidently I just finished constructing a coffee roaster, made from an ordinary popcorn machine. In basic it works by a fan blowing air past a heating element through the green coffee beans. By controlling the fan speed, I was able to control the temperature of the air roasting the beans. But, the roasting process turned out to be quite hands on, as the temperature needed to be kept constant following a profile, which was done by measuring the temperature the whole time and adjusting the fan speed with a pot meter.

That, I thought, should be automated....

12,482 views

4.89 (10 ratings)

Author: nightlife31
Published: Nov 4, 2009

2

License: (cc) BY-NC-SA

Related: Arduino, coffee, roast, roasting, PID, controller, popcorn

CYBERSONICA
OPENLAB WORKSHOPS
Manchester Art Gallery
MANCHESTER CITY COUNCIL

Cybersonica and Openlab Workshops with Manchester Art Gallery
An exhibitionof inventive Arduino and DIY electronic-circuitry projects
accompanying the major solo exhibition *Recorders* by Raphael Lozano Hemmer
18th September 2010 - 30 January 2011
Manchester Art Gallery, Mosley Street, Manchester UK M2 3JL
www.manchesterartgalleries.org

MAKE IT YOURSELF

step 1 Materials used

For this project I used:

-1x Arduino Duemilanove
-1x Arduino protoshield v2 (sparkfun) with breadboard
-1x 12x2 LCD screen (use a parallel one, would recommend a serial LCD)

-1x electric popcorn machine
-4x 1KOhm resistors
-1x small push button
-2x 1KOhm potmeter variable resistor
-1x IRF540N
-1x temperature sensor ZTK 33 V/7 mA
-1x solid state relais S216S02 (sharp)

-19V adapter (used the one of my laptop)
-plugs, wires, etc

step 2 Adjusting the popcorn machine

The popcorn machine itself is adjusted by placing a temperature sensor, disconnecting the heating coil and fan motor contacts and installing wires and a solid state relais. Be sure to read my previous instructable as well, since that was my starting point.

Temperature sensor:

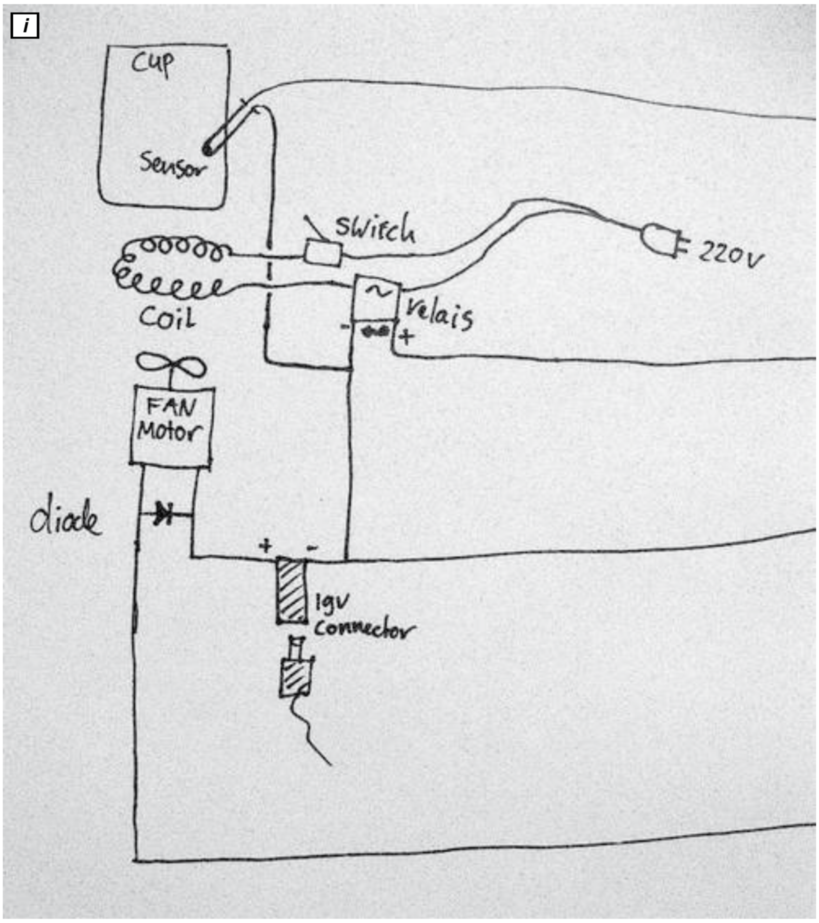
The sensor is installed on the inside of the aluminium cup, just above the sleeves, as it needs to measure the temperature of the beans while roasting. I used a piece of silicon (from a silicon hose) to make a heat resistant housing to keep the sensor in place and prevent short circuit. Drill a hole in the cup and place the sensor as shown on the photo's. Take care the sensor leads don't touch the metal of the cup. Outside the cup, the sensor is connected to wires long enough the lead out of the popcorn machine. The sensor and wires are secured with a tierap. Take care that the wires don't touch the cup either, as they can melt.

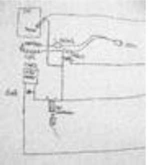
Installing a solid state relais to the heating coil:

The heating coil needs to be disconnected from the motor and from the heat security build into the popcorn machine. The electricity cord of the machine (220V) should be directly connected to the heating coil and the switch. In this circuit you build the solid state relais. Wires to control the solid state relais run out of the popcorn machine to the Arduino.

Fan motor:

The contacts of the fan motor are disconnected from the rest of the electronics and connected to two wires running out of the popcorn machine. A diode is placed between the two contacts to stop electricity being produced by the motor, after it is switched off but still turning. As power supply, a plug is installed to attach the 19v adapter of my laptop.





step 3 Connecting the Arduino

As the roaster should be able to operate without a pc connected to it, I installed it with a LCD screen. The LCD screen I have is parallel and a bit annoying to connect. I therefore made a LCD module I can simply take off and use on other projects. The module is on its own breadboard with a potmeter to control the light and 8 wires connecting to the Arduino. I would recommend getting a serial LCD as you only need 3 wires.

The Arduino is equipped with a protoshield with a small bread board. It consists of:

* The temperature sensor: a simple voltage divider connection with a 1KOhm resistor, using analog port 9.

* The fanspeed controller: pulse width modulator (PWM) of port 3, connected to a mosfet (IRF540N) which controls the fan motor with 19V coming from the laptop adapter. Use 1Kohm resistor between base and digital 3.

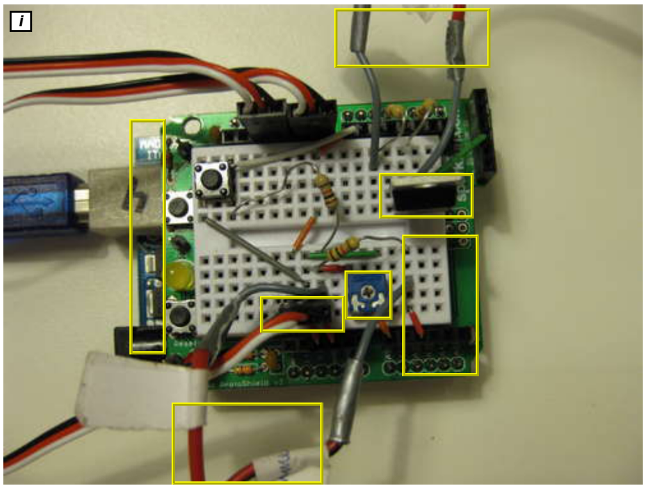
*Heater controller: turned on or off by the solid state relais, connected to digital port 5 (dont forget the 1KOhm resistor or you blow you relais).

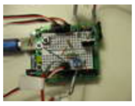
*A potmeter to adjust the minimal fanspeed, connected to 5V, ground and analog port 0.

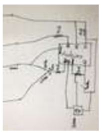
*A button to start the roasting, connected to digital port 6, use another 1KOhm resistor here.

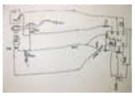
*LCD connection Digital pins 7-13 connected to RS, RW, E, d4, d5, d6, d7 of the LCD screen (and of course 5V and ground)

The popcorn machine is connected to the Arduino through 4 wires, all connected to metal pins to easily stick them in the holes of the breadboard.







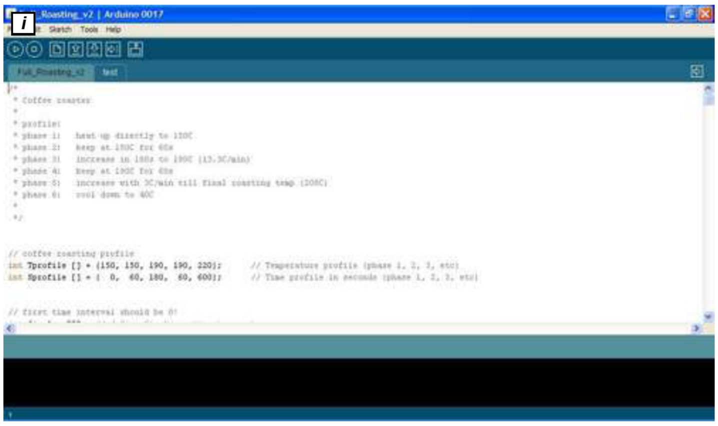


step 4 The Arduino program code

The program to run the Arduino consist of work of others, as it include a library for the LCD display (LiquidCrystal.h) and the PID control (PID_Beta6.h) , and a protocol to communicate with the associated Arduino PID Tuning Front-end, running on Processing (same maker as the PID control library, great work from Brett Beauregard!)

The program does the following:
Display a welcome screen and waiting for you to push the button. Then it turns on the fan but not the heating coil and it allows you to set the minimal fan speed at which the beans still move a little (you can lower it during the roasting as beans get lighter). When you press the button again the heating coil is turned on and the roasting starts. The programcontinuously measures the temperature and adjusts the fanspeed in order to match the set point. It will lower the fan speed to increase the temperature of the beans of increase the fan speed to do the opposite. If the beans get more than 3 degrees above the set point the heating coil is switched of, which will lower the temperature directly. The set point is updated along a profile that is set into the program. First heat up to 150C and stay there for 1 minute, then slowly increase to 190C and stay there for 1 minute. Then slowly increase the temperature again until the final temperature is reached. At this point, the fan is set on max and the heating is turned down. The roaster stops when the beans are lowered to 40C.

After lots and lots of tweaking, the final program code to run the coffee roaster can be found in the file below.



```
/*
 * Coffee roaster
 *
 * profile:
 * phase 1:   heat up directly to 150C
 * phase 2:   keep at 150C for 60s
 * phase 3:   increase in 180s to 190C (13.3C/min)
 * phase 4:   keep at 190C for 60s
 * phase 5:   increase with 3C/min till final roasting temp (208C)
 * phase 6:   cool down to 40C
 *
 */


// coffee roasting profile
int Tprofile [] = {150, 150, 190, 190, 220};     // Temperature profile (phase 1, 2, 3, etc)
int Sprofile [] = { 0,  60, 180,  60, 600};      // Time profile in seconds (phase 1, 2, 3, etc)


// first time interval should be 0!
```

Coffee_Roaster_v1_0.pde 11 KB

step 5 Testing..... and roasting your coffee beans

Ok, so now you connect the popcorn machine to the Arduino, to the 19V and 220V power supply and you connect the Arduino to the LCD screen and your computer, and you are ready to test your roaster.

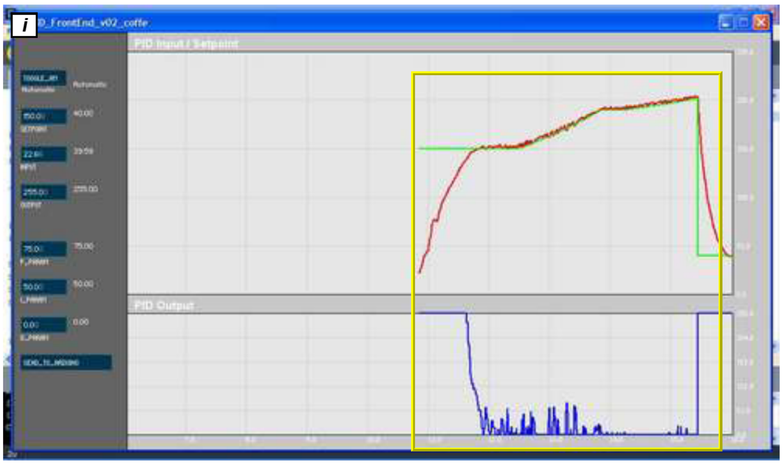
I guess if you use the program, the temperature sensor should be more or less ok. But better be to calibrate is, and if necessary to adjust is. For this you would need a thermometer which can measure up to 220C. Measure at difference temperatures and adjust "a = 0.6271 and b = -213.13"

Next I suggest to try out some things, see how it works and follow the temperature profile with the front-end program. You will need some beans to be able to follow the temperature profile. Take good care that you set a minimal fanspeed that still keeps the beans a little bit moving. Otherwise the beans might burn (can work out dangerously!). As the roasting progresses, the beans get lighter and therefore the minimal fanspeed can be lowered.

If this all works out, I suggest you try your first batch of coffee beans. Mine can take up to 100g of beans each time, which isn't so much (but then I dont drink THAT much coffee). If it works out fine, you can start to experiment with different profiles (I would suggest to start with the final temperature settings)

Good luck and enjoy.

I have thought about an additional controller that would keep an eye on the beans. If they stop dancing then the fan speed should be increased a little bit. Couldn't think of any way to sense this, so if you have any suggestion please let me know.



TOGGLE_AM Automatic
150.00 SETPOINT 40.00
22.60 INPUT 33.59
255.00 OUTPUT 255.00
75.00 P_PARAM 75.00
50.00 I_PARAM 50.00
0.00 D_PARAM 0.00