

Pomodoro & Arduino

Mike Stok <mike@stok.ca>

How is this related to perl? Well, you can use Pomodoro to carve up writing a Perl project!

I was planning to maybe use Perl to talk over USB to the serial port so we could display the time ticking down, but my time ticked down.

The structure of the talk is:

- * Pomodoro
- * Arduino hardware / software
- * Fritzing
- * Demos

Pomodoro Technique

- Chose a task to do
- Work on it for 25 mins
- Take 5 min break
- Repeat until task is done - taking a longer break every 4 cycles

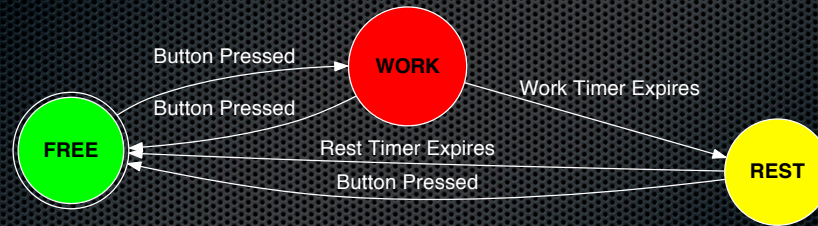


Pomodoro was invented in the late 1980s by Francesco Cirillo as a simple way to manage time to help you work more effectively. Many parallels with agile - simple tools (timer, pencil, todo sheet, activity inventory sheet, records sheet), planned time-boxed activities, review of status, ways of dealing with interruptions.

I have used it at work with a reasonable degree of success, and at home it can help avoid losing hours to a project *without realising it*.

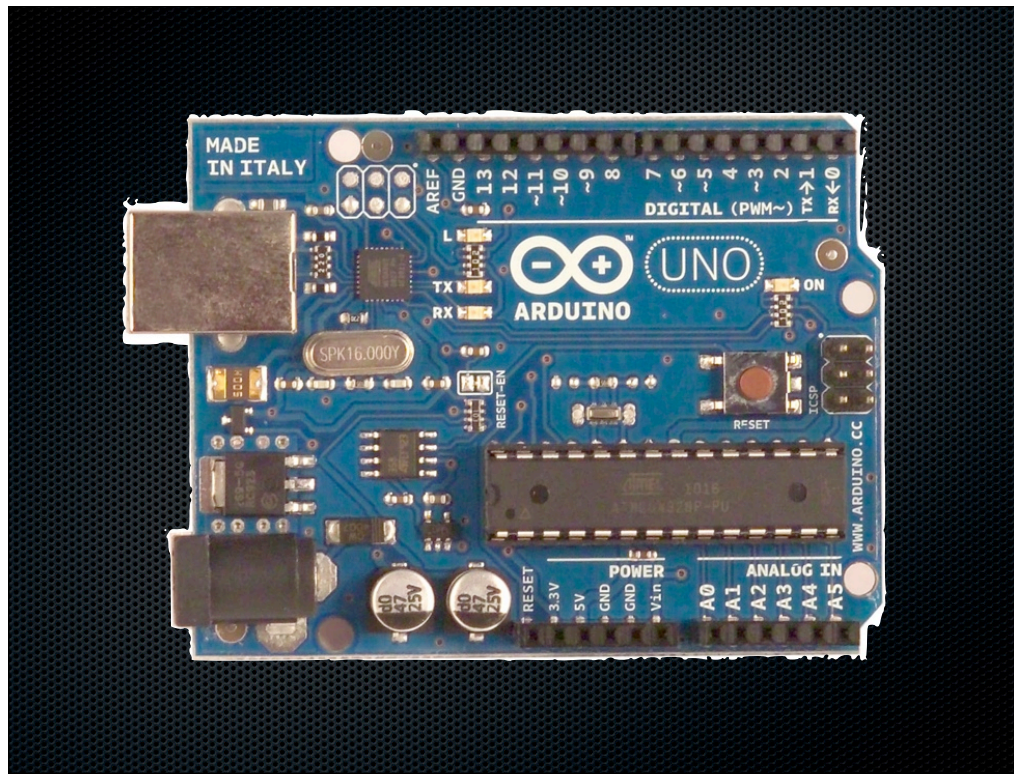
I have a need to communicate current state with potential interruptors at work. If they know that it's at most 30 minutes before they can bug you, then maybe they will not interrupt a work session.

Pomodoro States



Basic idea is to have a simple traffic light style indicator to ward people off with a single button which lets me flip between the states as needed. The lights can be displayed on top of my monitor, and that might be a good first cut.

How hard can that be? :-)

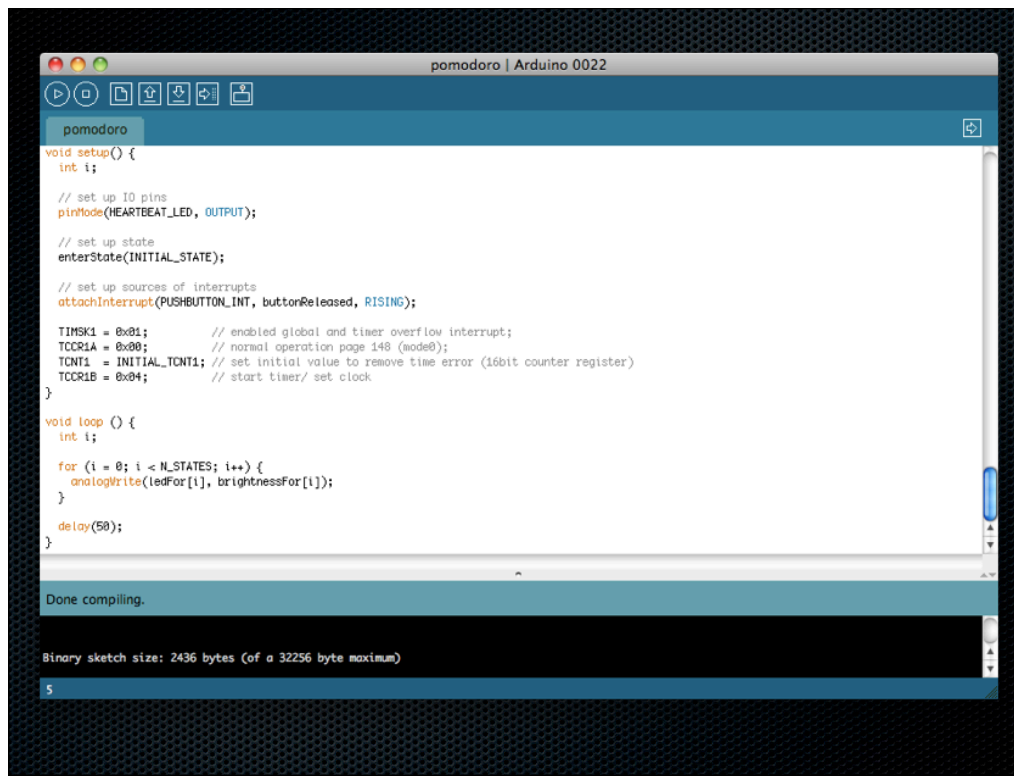


New Arduino Uno - main parts are an Atmel ATmega328, ATmega8U2 programmed as USB to serial, and USB connector which can provide power.

14 digital I/O pins (6 can be PWM outputs), 6 analog inputs, power, analog reference, ...

As I'm not really an electronics person I went to Creatron Inc. <http://creatroninc.com/> on College to buy a box o' bits which included an arduino, breadboard, wires, leds, switches, sensors etc.

Image from http://www.robotronic.co.nz/index.php?main_page=product_info&cPath=1_18&products_id=71

The image shows a screenshot of the Arduino IDE interface on a Mac OS X system. The window title is "pomodoro | Arduino 0022". The main area displays the C++ code for a sketch named "pomodoro". The code includes a setup function and a loop function. The setup function initializes pins, sets state, and configures an interrupt. The loop function iterates through states, writing to an LED and delaying. A status bar at the bottom indicates "Done compiling." and "Binary sketch size: 2436 bytes (of a 32256 byte maximum)".

```
void setup() {
  int i;

  // set up IO pins
  pinMode(HEARTEAT_LED, OUTPUT);

  // set up state
  enterState(INITIAL_STATE);

  // set up sources of interrupts
  attachInterrupt(PUSHBUTTON_INT, buttonReleased, RISING);

  TIMSK1 = 0x01; // enabled global and timer overflow interrupt;
  TCCR1A = 0x00; // normal operation page 148 (mode0);
  TCNT1 = INITIAL_TCNT1; // set initial value to remove time error (16bit counter register)
  TCCR1B = 0x04; // start timer/ set clock.
}

void loop () {
  int i;

  for (i = 0; i < N_STATES; i++) {
    analogWrite(ledFor[i], brightnessFor[i]);
  }

  delay(50);
}
```

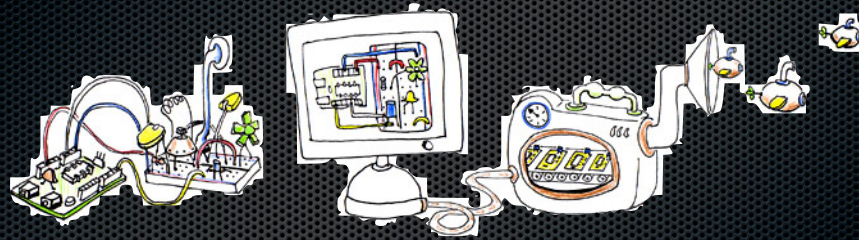
The development environment you can download provides an integrated environment to program and upload code in one pretty simple interface.

This is on OS X, and we can see setup() and loop() here which are the main components of an Arduino program. If time permits we'll look at the code in a second.

See the Pragmatic Programmers April 2011 magazine for alternatives to the Arduino IDE, the code is saved in a text file which can be edited with your favorite editor - in my case ~/Documents/Arduino/pomodoro/pomodoro.pde.

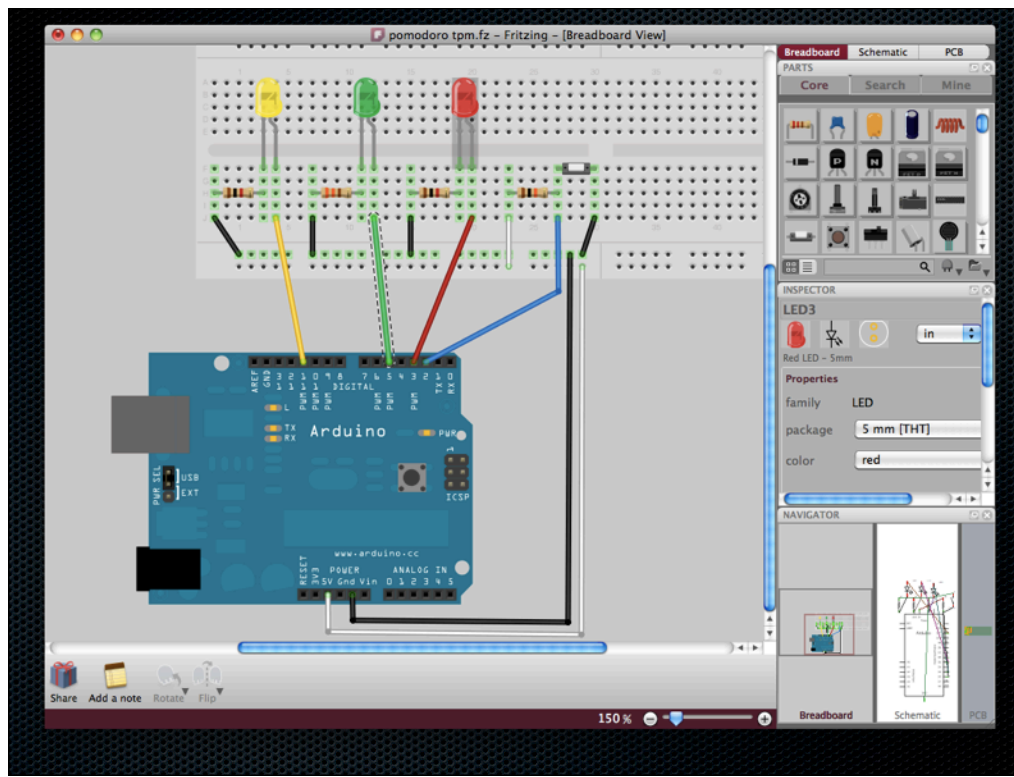
<http://www.electronicblog.net/examples-of-using-arduinoatmega-16-bit-hardware-timer-for-digital-clock/>

Software & Hardware



The software driving the Arduino is half the story, an Arduino project is a combination of hardware and software. I need a way to capture the hardware better than my illegible scribbled (and inaccurate) circuit diagrams.

Image from <http://fritzing.org/> - <http://fritzing.org/media/uploads/headers/home.jpg>



Fritzing maintains three views of a project. Drag & drop components onto the breadboard, wire them up, then you can fix up the schematic by moving the symbols, and there's even an auto-routing PCB gizmo (1 or 2 sides) which can give you PDF to send off and get turned into a real PCB if you like.

It's also a way for me to keep the schematic on the computer alongside the code to drive the Arduino.

Demos

- Look at Fritzing in action to see schematic
- Look at code to see its relationship to the hardware
- Compile and run the code

Resources

- <http://www.arduino.cc/> - Arduino Home Page
- <http://fritzing.org/> - Fritzing Project Page
- <http://pragprog.com/magazines/2011-04/content> - Prag Prog magazine - Arduino issue
- <http://www.atmel.com/> - Atmel for data sheets
- <http://www.pomodorotechnique.com/> - Pomodoro Technique home page