

Arduino Benchmarking

Nicu-Serban POP

Polytechnica University of Timisoara

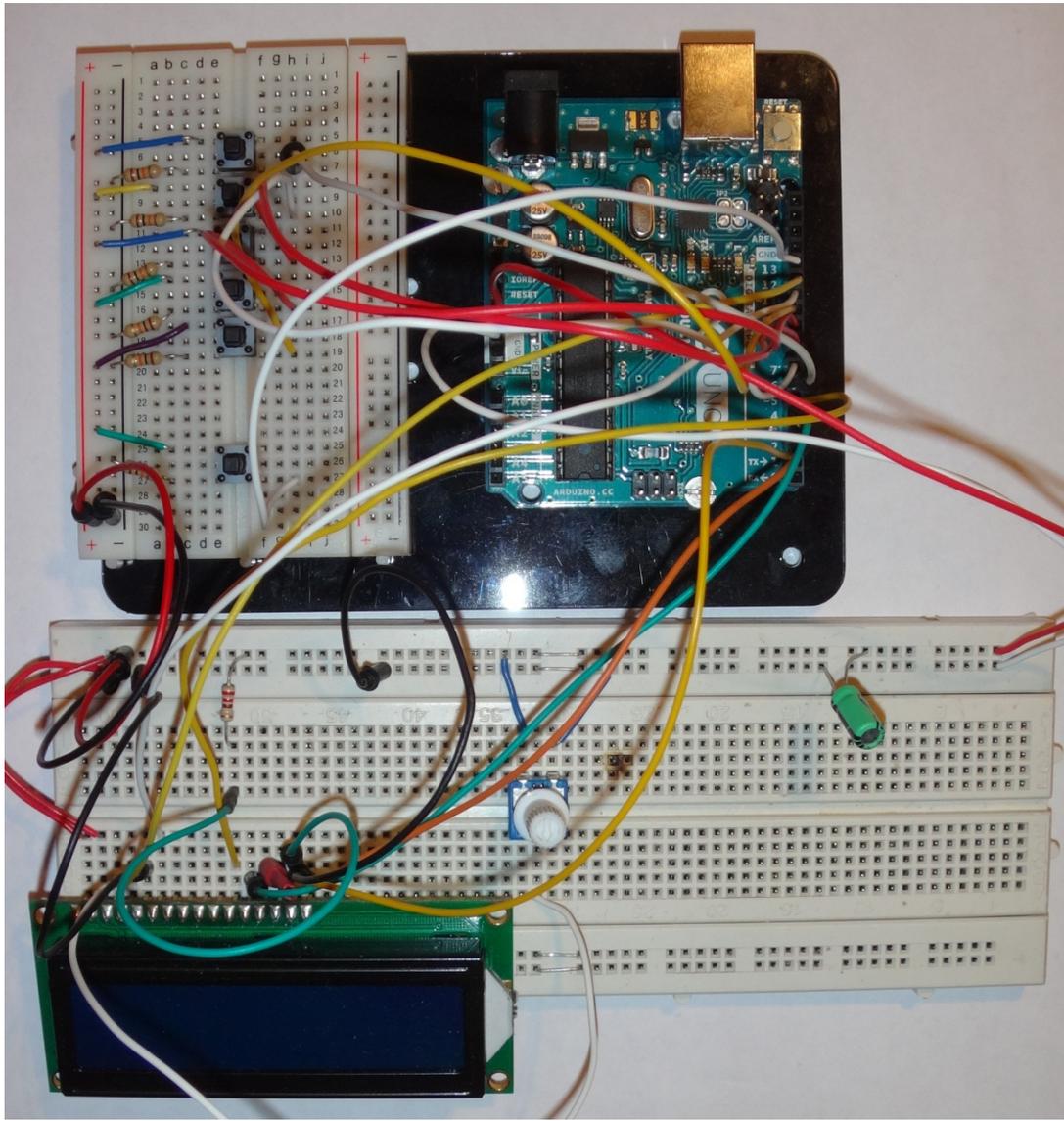
1. Introduction

The application is a fusion of two benchmarks, it tests different components of an Arduino platform (Arduino Uno in this case). Components analyzed are the CPU, tested by a SuperPI like program and the EEPROM tested by writing and reading data timings.

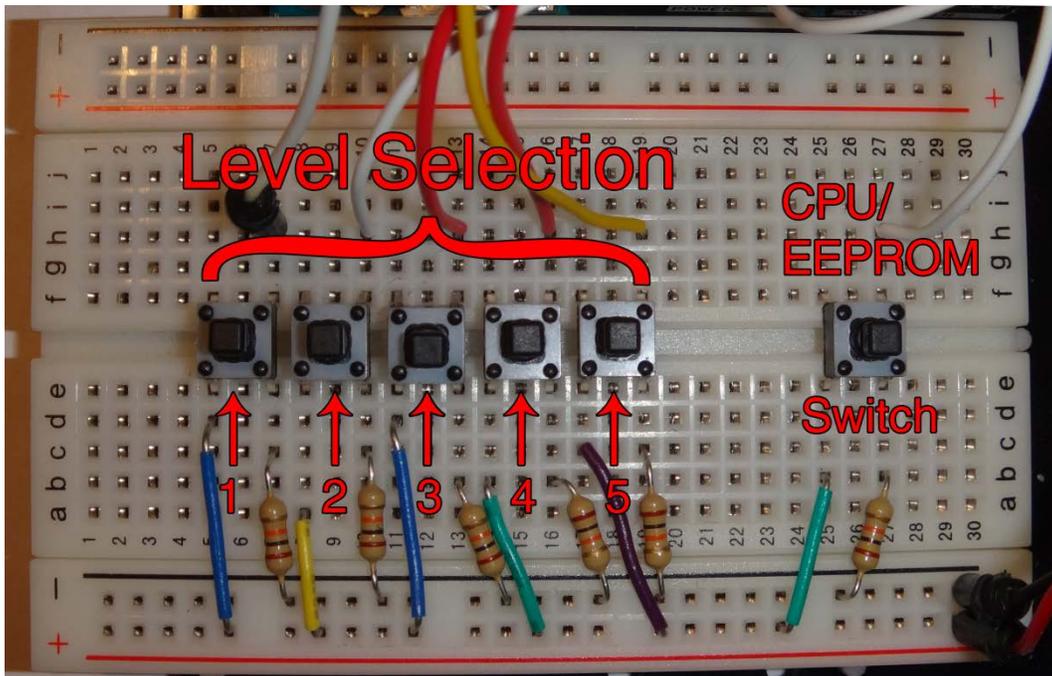
2.1 Implementation

- CPU benchmark
 - method used for computing PI: Newton
 - five levels of evaluation
 - Level 1 – 115.000 iterations
 - Level 2 – 230.400 iterations
 - Level 3 – 345.600 iterations
 - Level 4 – 460.800 iterations
 - Level 5 – 576.000 iterations
- EEPROM benchmark
 - Time for writing random data on 1024 bits
 - Time for reading random data from 1024 bits

- Project Overview:

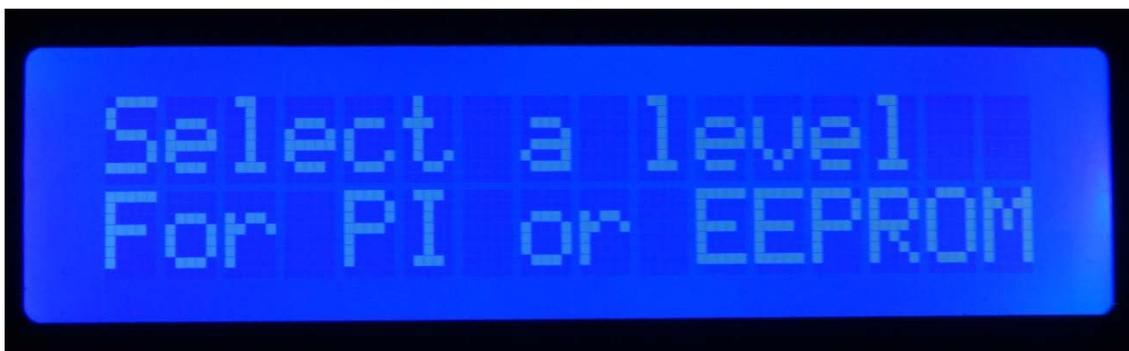


- User Interface



3. Usage and results

After launching the benchmark the user is prompted the default screen requesting a level for PI to be tested, or to initialize the EEPROM assessment which is done by just pressing one of the 6 push-buttons. A higher level for the PI test means a greater precision but also a longer run time.



After pressing any of the six buttons, the corresponding test will start.

- For Pi, press one of the 5 buttons on the left:

This process will start to compute Pi to the selected level and return an approximation of Pi and also the time it took to compute it.

The method used for calculating Pi is a pretty simple and not very precise one but we are more interested in execution time rather than precision.

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

Tested on multiple types of Arduino will return very close timings (few milliseconds) because at such a low level there is almost no difference between components.



- For EEPROM, press the right-most button

After choosing the EEPROM test, Arduino will initiate to write random numbers on its memory displaying how much time it needed for the operation to complete. After ten seconds it will commence to read the written data and will display again the time in milliseconds.

- Writing:



- Reading



4. Conclusions

I learned a few extra things about Arduinos, like the potential of such a tiny device in doing mathematical equations and crunching hard numbers or just how slow the CPU is compared to a desktop CPU.

The easiest part, which I enjoyed the most, was assembling everything together and the most difficult one was getting the idea into my head and figure out how to code it.

As I mentioned before, I like development boards and the biggest joy is seeing your programs work as intended.

5. Q&A

- Where did you get the idea?

Well, I like Arduino and I just tried to test everything that was possible on it.

- Are there any other benchmarks like this?

Not that I could find on the internet.

- Does this work on other types of Arduino/Dev boards?

Yes as long as you have the code. Looking forward to implement a chip version (attiny) for plug-and-play testing.

- What are the specifications of the used Arduino?

8-bit CPU@16 MHz, 32KB flash memory, 1KB EEPROM