LowPower

Rafal Grasman, Minh-Triet Diep

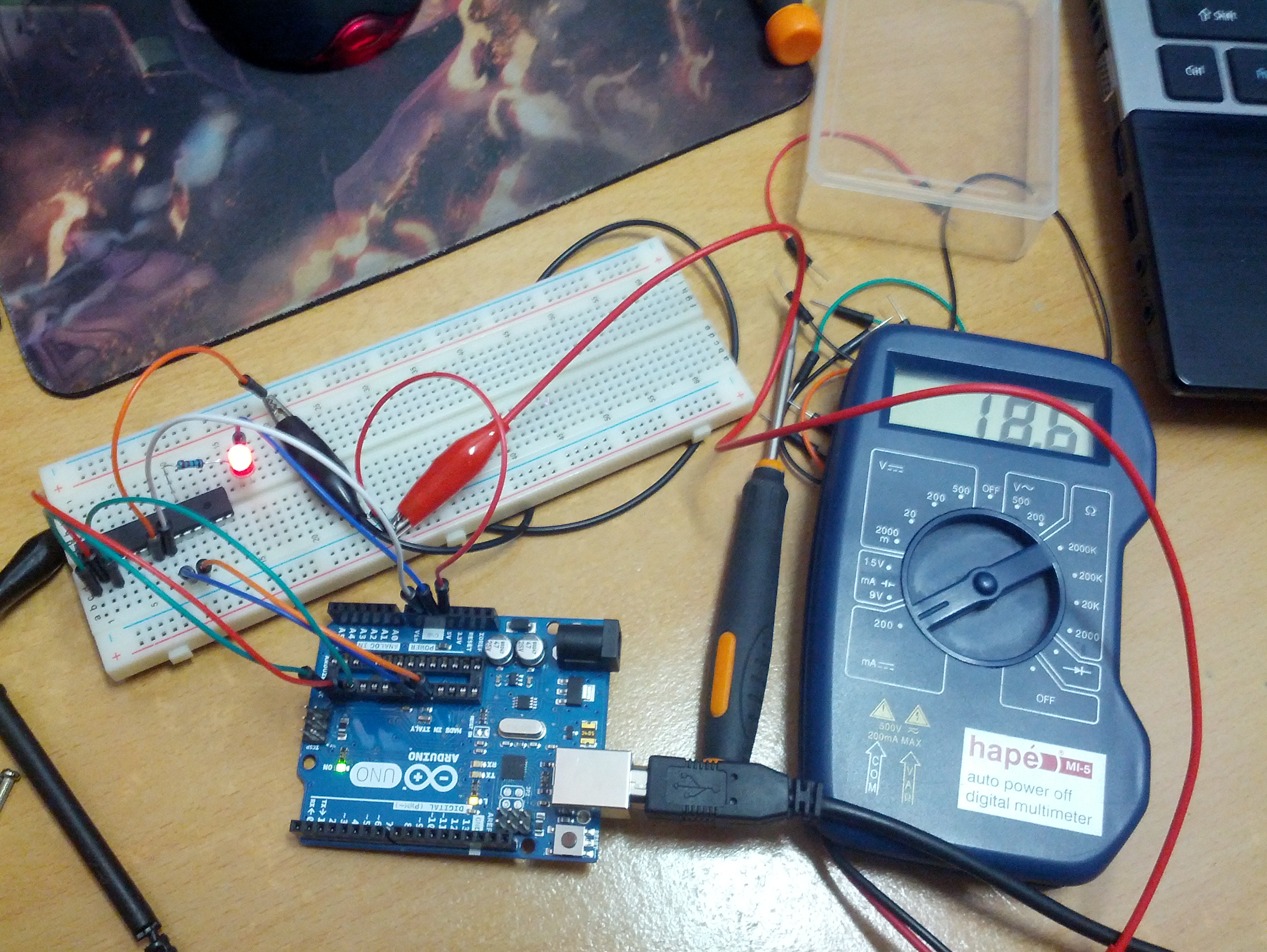
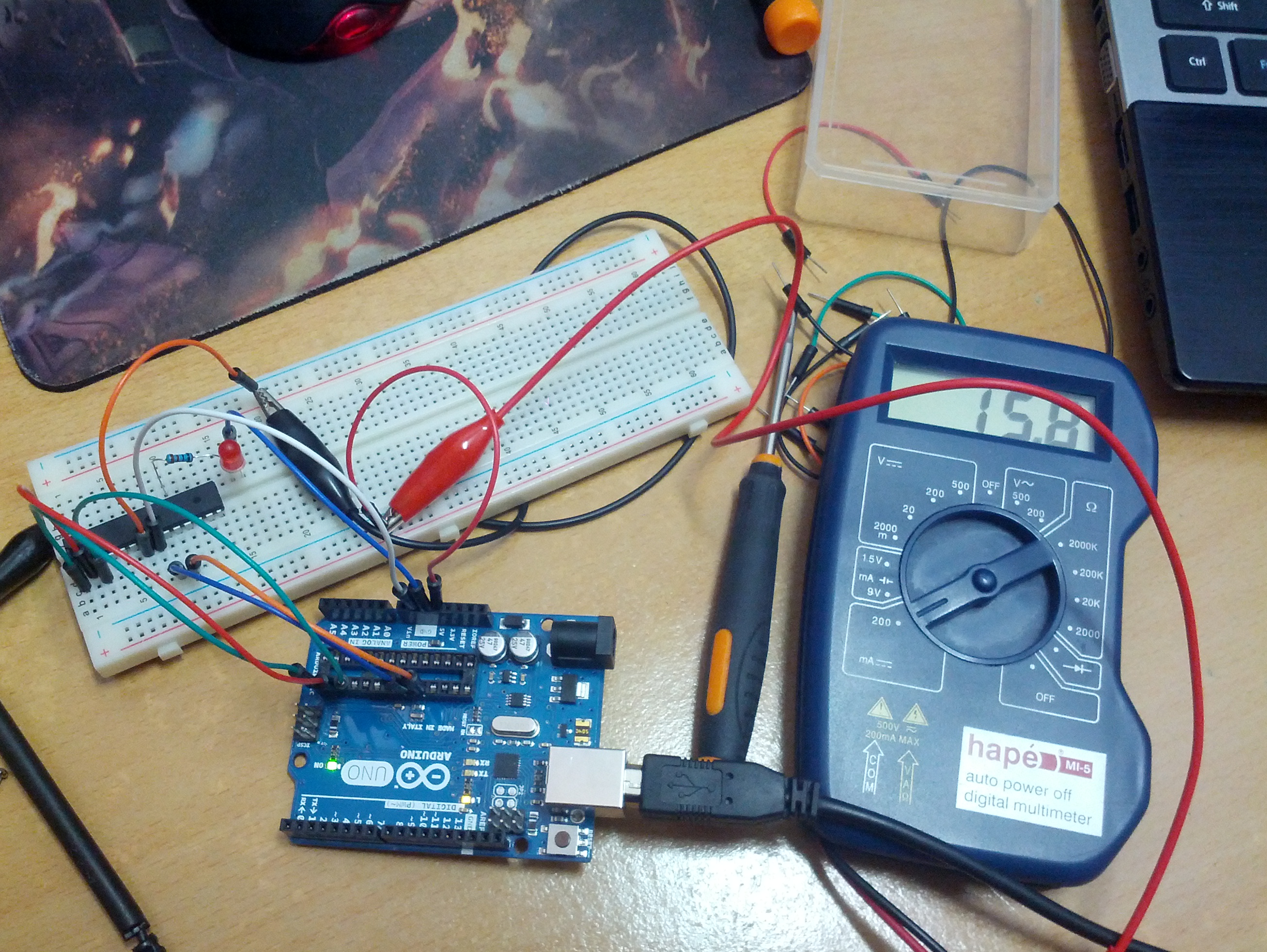
In deze opdracht wordt het energieverbruik van de Arduino onder de loep genomen en wordt deze verminderd door een aantal dingen uit te zetten of te vertragen op de Arduino.

# Peripherals

Arduino Busy waiting, no optimizations:

15.8mA Led Off

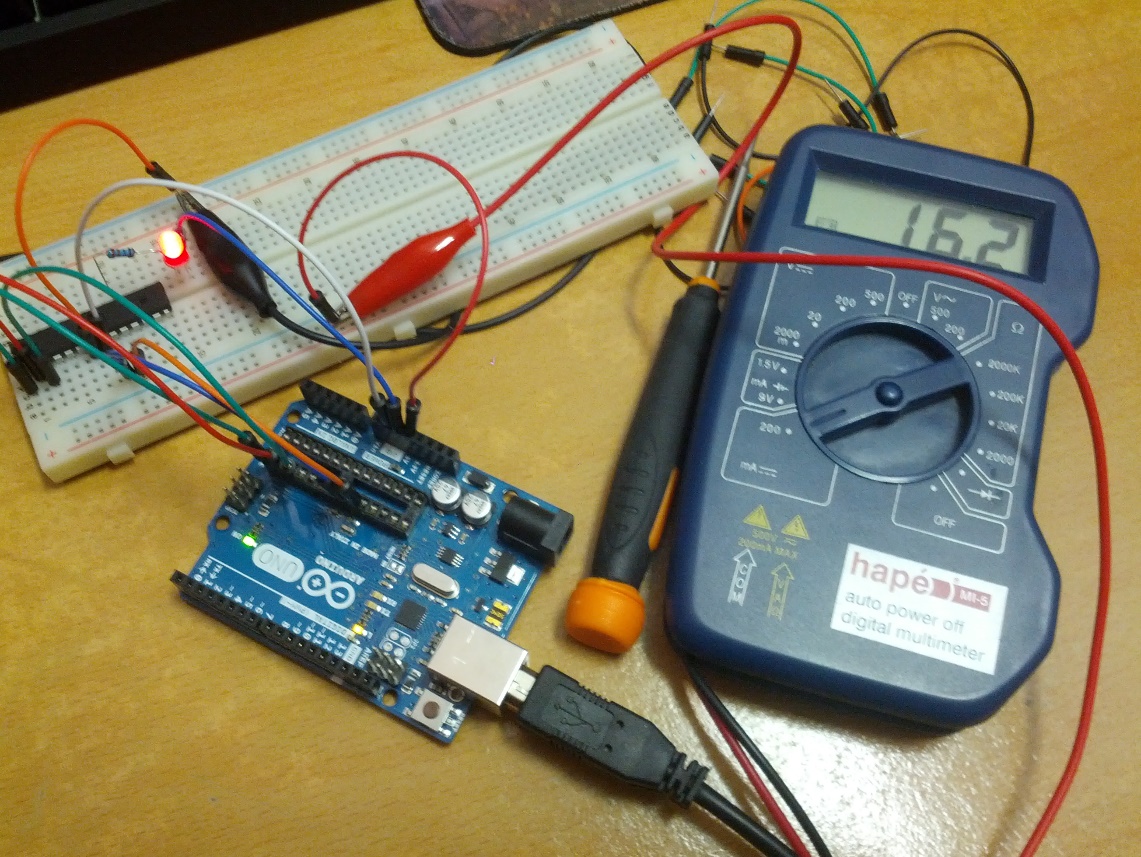
18.6mA Led On

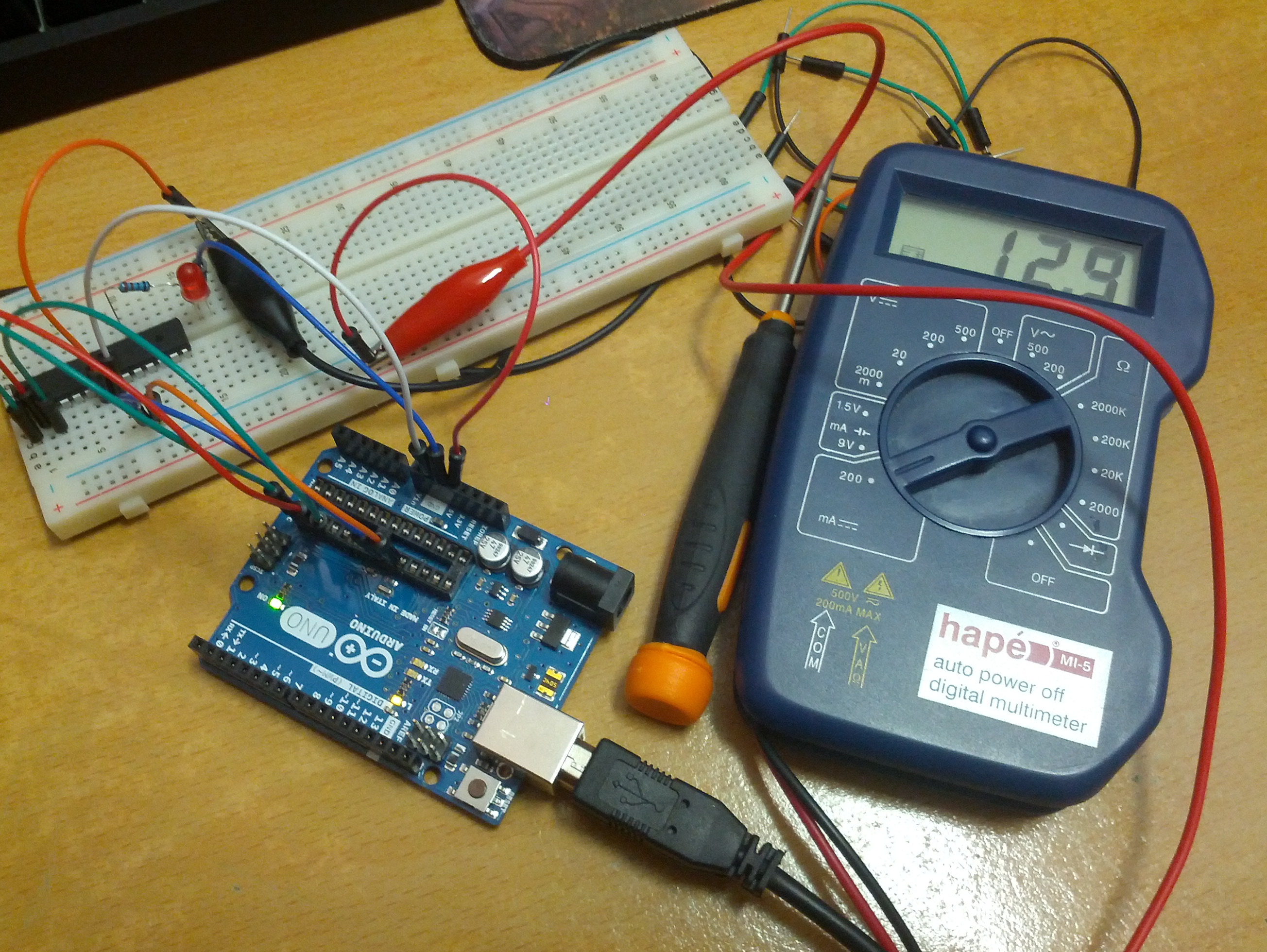


Arduino Busy waiting, turned off ADCSRA, ACSR, DIDR0, DIDR1, PRR

Led Off: 12.9mA

Led On: 16.2mA





Hier is te zien dat het stroomverbruik een heel klein beetje (2 tot 3 mA) omlaag is gegaan.

# Kloksnelheid

Met de prescaler kan worden gespeeld om de kloksnelheid omlaag te krijgen en dus het verbruik te verminderen:

|  |  |  |
| --- | --- | --- |
| **Prescaler** | **Led On** | **Led Off** |
| 1 | 13.5 | 10.6 |
| 2 | 10.3 | 7.4 |
| 4 | 8.5 | 5.8 |
| 8 | 7.3 | 4.3 |
| 16 | 6.7 | 3.6 |
| 32 | 6.3 | 3.3 |
| 64 | 6.1 | 3.1 |
| 128 | 6 | 3 |
| 256 | 5.9 | 2.9 |

Na de Prescaler op 8 te hebben gezet, daalt het energieverbruik niet meer zo hard.

# Slaap

Daarna kan nog de ATMega op slaap worden gezet:

Wanneer de ATMega in Sleep mode is, leest de multimeter 00.0mA. Er moet nog een hele kleine stroom lopen wil de watchdog functioneren, maar dit is niet af te lezen met mijn huidige multimeter. Na het aflopen van de watchdog van 4 seconden (dus 1s aan, 1s uit, 2s slaap) gaat de Arduino weer verder.

Conclusie  
Met het uitzettten van de peripherals en de ACD kan het stroomverbruik een beetje worden teruggedrongen. Door met de prescaler van de clock te spelen kan het stroomverbruik nog meer worden verlaagd. Ten slotte kan de Arduino in slaapmodus worden gebracht waar die nauwelijks stroom verbruikt, en met een watchdog en een interrupt kunnen taken toch worden uitgevoerd.