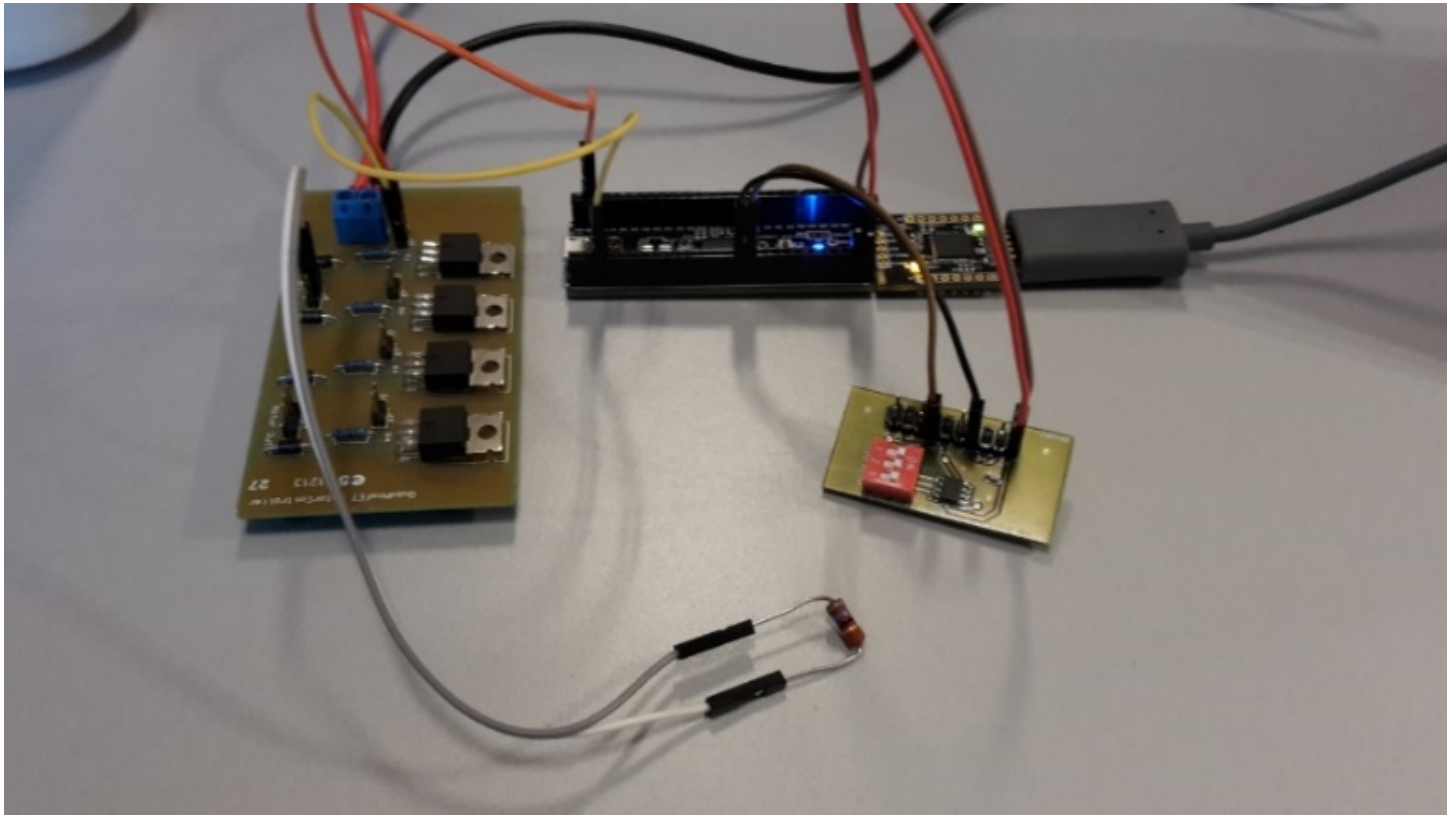
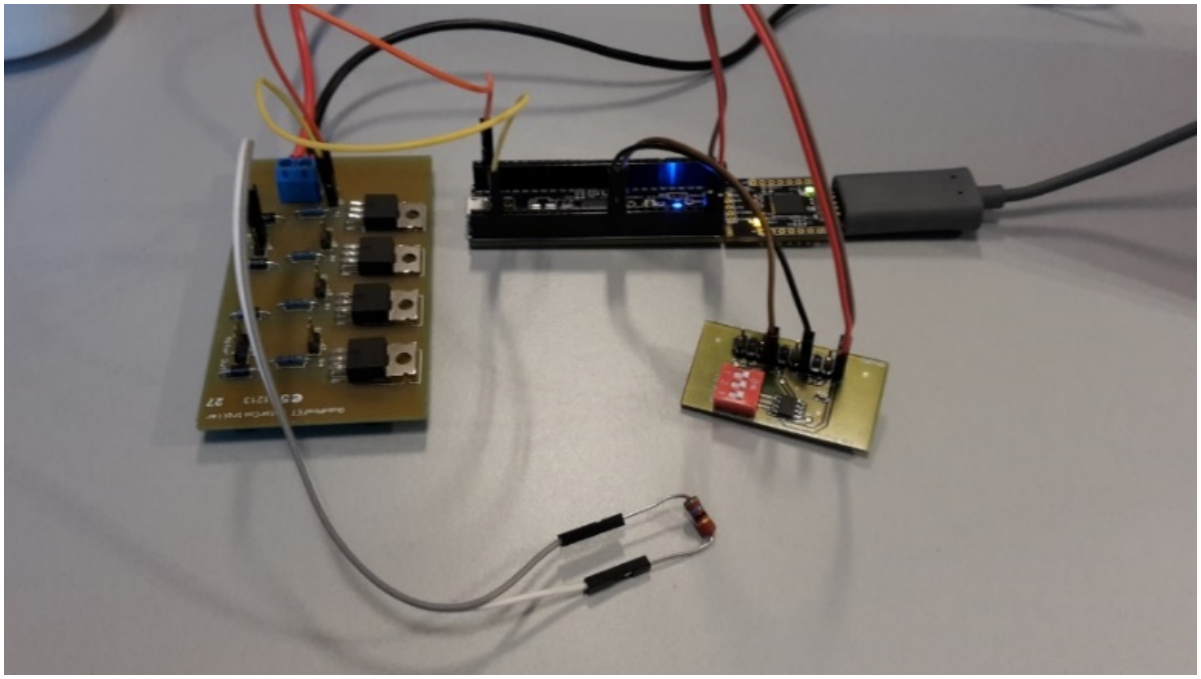
Automatic temperature regulator lab experiment



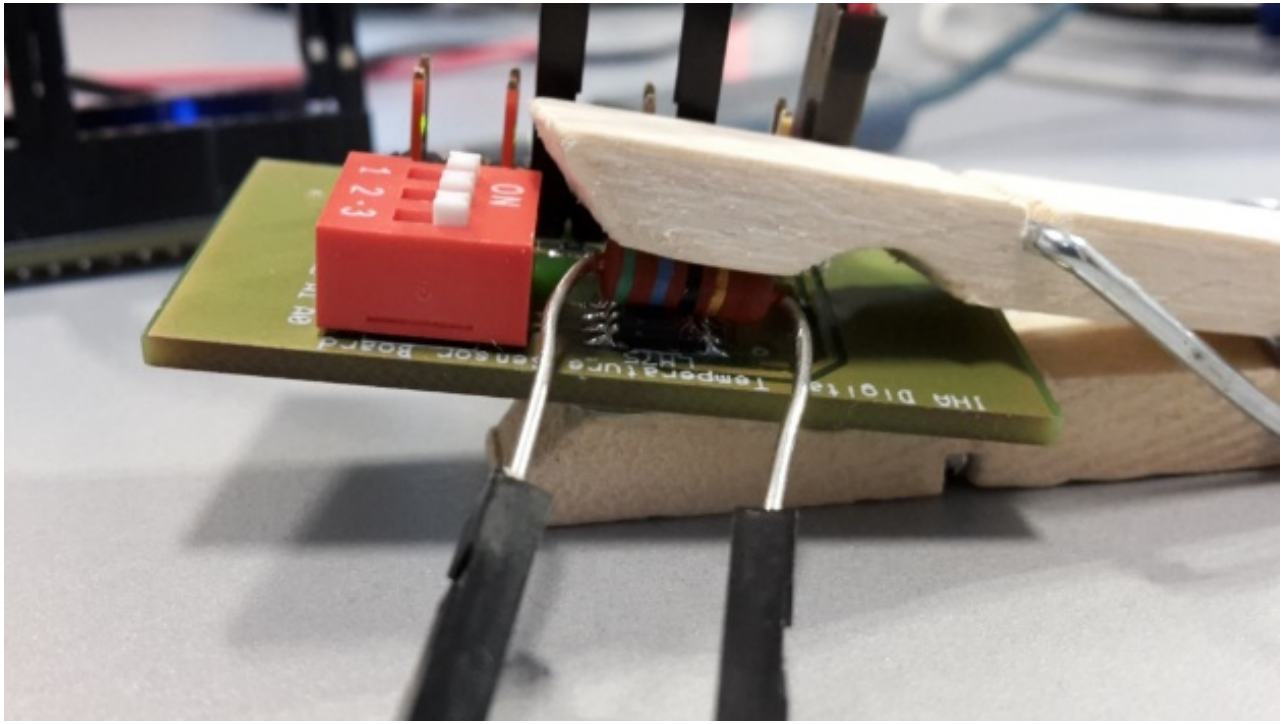
1 experiment

Use a software PID controller in the PSoC to obtain and maintain a constant temperature, given a setpoint.

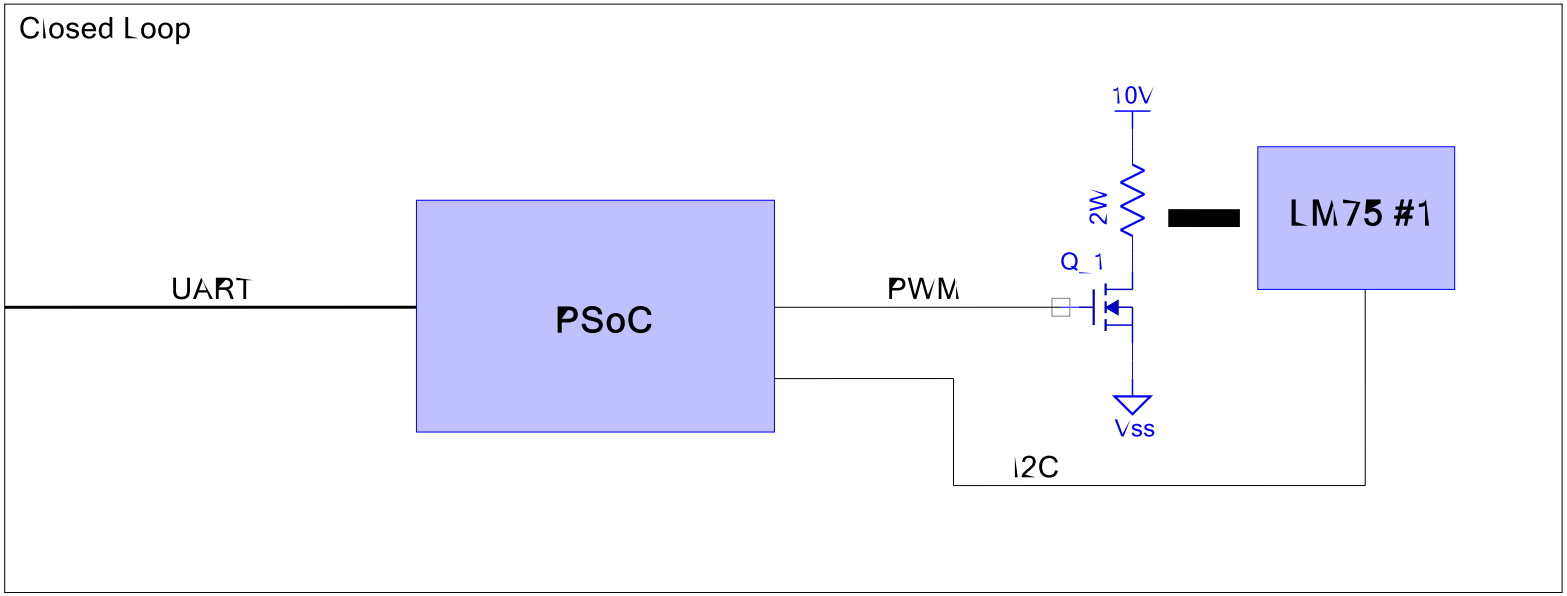


PSoC
Power Resistor
MOSFET PCB
LM75 temperature sensor

Don’t short any pins



Closed loop system



PID starting point

You can use the provided PSoC project on Blackboard as a starting point.

It contains an implementation of a PID controller.

As a starting point for the controller coefficients use:

$$K_p = 2, \qquad K_i = 1/30, \qquad K_d = 0$$

LM75 delay

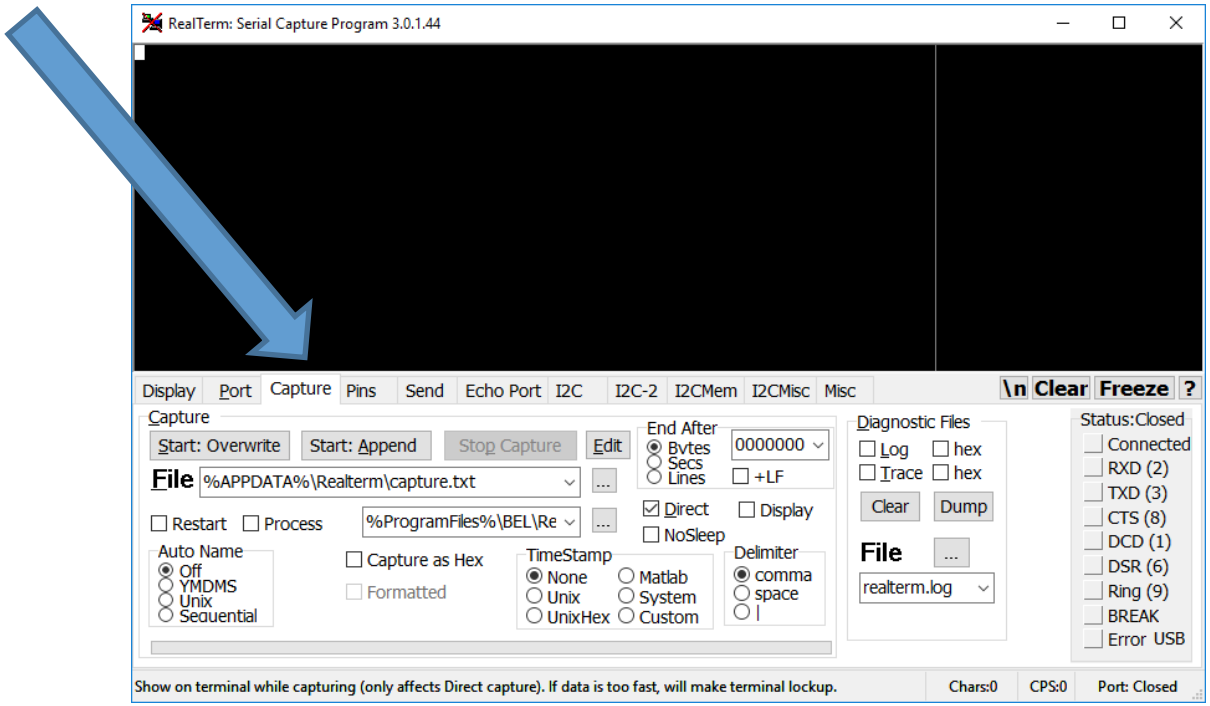
Minimum delay between reading the LM75 is given by the time it takes the LM75 to finish a temperature conversion.

If you request the temp too fast (the datasheet only guarantees >300ms), you will get the old converted value and the temperature never changes.

Common mistakes

- Don’t start the regulation loop before you have turned on power.
  - The integral part of the controller will build up a large integral because the error is unchanged.
- Don’t sum or integrate twice.
  - The output from the PID controller ***is*** your control signal.
- Always have a stable temperature and a running controller, before making the change in setpoint.
  - otherwise you can’t compare controller performance

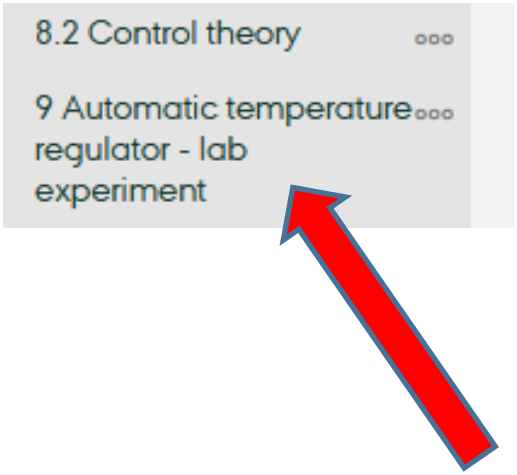
Hints!



- You may benefit from plotting the control signal and maybe even the individual outputs of the P, I and D parts of the controller, to better understand how the controller works.

- If you use the provided PID controller code, you can write the console output to a file e.g. using RealTerm and import the comma separated values directly into Excel.

Important reading



Martin Kjær has written a 2 page document on how you should conduct and document your experiments.

Read it!

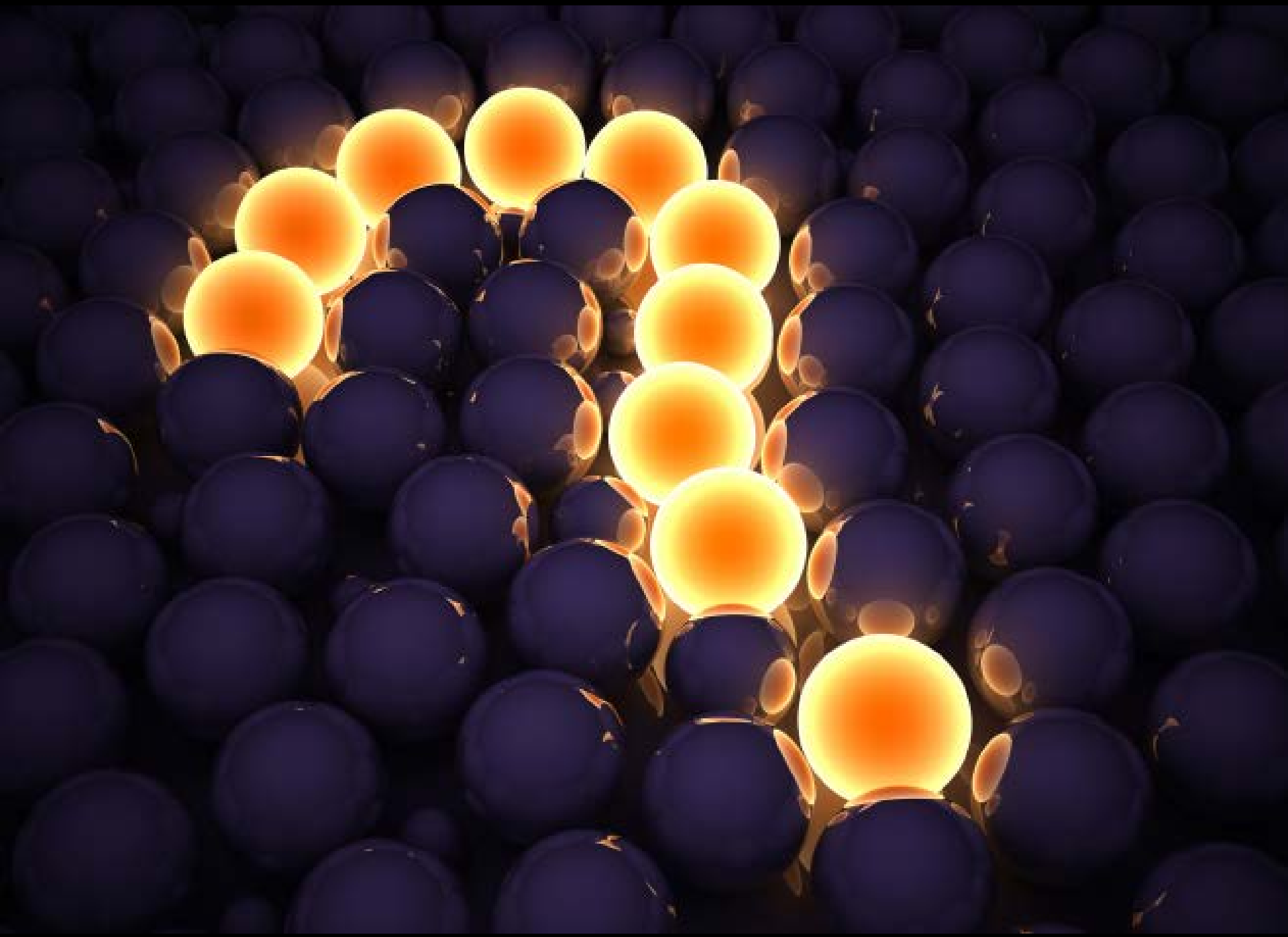


Image resources

- Question mark: https://wall.alphacoders.com/big.php?i=437563