

Details of a multi-core problem:

The Train Crossing:

A function launched with `cog_run`:

- Can not require parameters passed to it (use a global variable if a parameter is needed)
- Can not return a value (again, use a global variable if something needs to be passed back)
- Should have all its instructions inside a `while(1)` loop, unless your program is also going to deliberately stop the cog (which you will see in the Stopping Cogs lesson)
- Should not contain a `print`, `scan`, or other function call that uses the SimpleIDE Terminal, unless your program is going to specifically manage that with additional functions (which we'll do in the Print from a Different Cog lesson)

Starting a cog:

<http://learn.parallax.com/multicore-approaches/simple-multicore>

Stopping a cog:

<http://learn.parallax.com/multicore-approaches/stopping-cores>

Approach:

1. Write a function to blink two lights.
2. Write a function to move the gate down.
3. Write a function to move the gate up.
4. Write a function to make the sound.
5. Write the main function which checks if a train is present. If not, check again.
 - If the train is present, start all the other functions.
 - If the train is still present, check again
 - If the train is not present, stop all the other functions and go back to step 5.