DESIGN FOR CPS SHORT COURSE
ASSIGNMENT 2: AIRCRAFT DOOR SYSTEM

EDWARD A. LEE

Due date: Wednesday, May 18, 2022, AOE

1. Reading

(1) Lee and Lohstroh, "Time for All Programs, Not Just Real-Time Programs," ISoLA, October, 2021.
(2) Lee, "Determinism," *ACM Transactions on Embedded Computing Systems* (TECS), July, 2021.

2. Programming Assignment

Create a Lingua Franca program that emulates a version of the aircraft door example considered in class. Your design should have three components:

- `Cockpit`: This has one output, `open`, that produces a `bool` to issue a command to open or close the door. It should issue this output whenever the user types 'o' or 'c' on the keyboard, followed by Return or Enter.
- `VisionMonitor`: This has one output `ramp_present`, that produces a `bool` that is `true` (in C) or `True` (in Python) when a vision system detects that there is a ramp outside the door. It produces `false` or `False` when it detects that there is no ramp. To emulate the vision system, when the user types 'r', followed by Return or Enter, this means a ramp has been detected. When the user types 'n', this means there is no longer a ramp detected.
- `Door`: This has two inputs, `disarm` and `open`. The `disarm` input tells the component whether to disarm the emergency slide deployment or arm it, and the `open` input tells it to open or close the door. Its behavior should be as follows: If it receives an `open` command, it should ignore it if no ramp is present and open the door otherwise. If, however, it receives two `open` commands in rapid succession (within two seconds of each other), then it should open the door regardless of its armed status, possibly deploying the emergency slides.

For inspiration and coding examples, see the TrainDoor example, which shows one way to accomplish the keyboard inputs you will need:

- **C version:** https://ptolemy.berkeley.edu/∼eal/cps/code/c/TrainDoor.lf
- **Python version:** https://ptolemy.berkeley.edu/∼eal/cps/code/python/TrainDoor.lf

Specific deliverables:

(1) A trace of the output from your program with readable time values relative to the start of the program.
(2) Your well-commented Lingua Franca program.
(3) A short discussion of the design. Is this a good design? How could it be improved? What could go wrong?

Send your deliverables by email to eal@berkeley.edu.