

# Mandatory Assignment 1

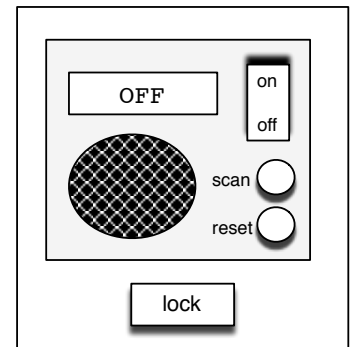
INF 2140, Spring 2012

Due: 03.02.2012

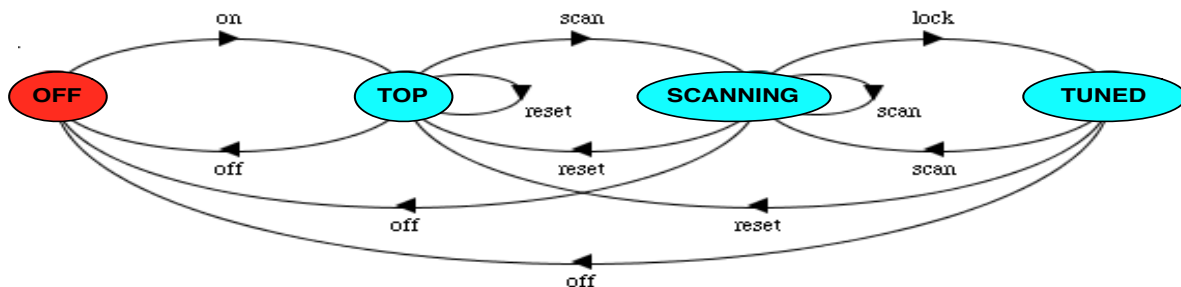
Published Date: 27.01.2012

## The Portable FM Radio

A miniature portable FM radio has three controls. An on/off switch turns the device on and off. Tuning is controlled by two buttons `scan` and `reset`, which operates as follows. When the radio is turned on or `reset` is pressed, the radio is tuned to the top frequency of the FM band (108MHz). When `scan` is pressed, the radio scans (between 108MHz and 88MHz) until it locks on to a station (event expressed by pressing the `lock` button). If the radio is currently tuned to a station and `scan` is pressed, then it starts to scan toward the next station, when `reset` is pressed the receiver tunes to the top.



Below is a proposed state machine of this model, which uses the alphabet {`on`, `off`, `scan`, `reset`, `lock`}. Here we ignore the current frequency and scanning is done by pressing `scan` repeatedly until the `lock` button is pressed.



## Your Tasks

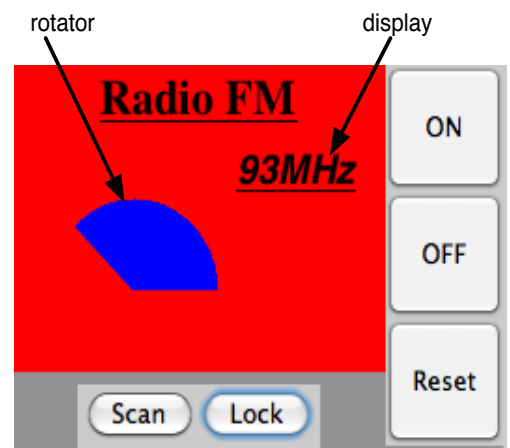
- Implement the FSP that generates the state machine above.

**HINT:**

OFF = (on -> TOP),

...

- Complete the implementation of the portable FM radio in Java, based on the state machine of the model and using the graphical interface provided in the zipped Eclipse Project "inf2140assignment1.zip". Your task consists on implementing the file "RadioController.java" in order to animate `display` and `rotator` that should change state when the events `on`, `off`, `reset`, `scan` and `lock` happen.



**IMPORTANT NOTE:** You are free to modify the user interface, but you **MUST NOT** modify the files “IRotator.java”, “IDisplay.java”, “IController.java” and “AssignmentOne.java”. And the signature of the constructor of the RadioController class (RadioController(IRadioDisplay display, IRotator rotator) **MUST NOT** be modified.

## Deliver to

The assignment should be carried out individually or with one or two other students (Note that groups of more than three students are not allowed) and delivered to the teaching assistant responsible through <https://devilry.ifi.uio.no/>.

Only one answer is delivered if two or three students work together. The answer must be marked with full name(s) and username(s) of the contributing students.

## How to deliver

- You have to implement the FSP using the LTSA Tool (the tool is installed on the lab machines, or you can download it from: <http://www.doc.ic.ac.uk/~jnm/book/ltsa/download.html>). Your implementation should compile and generate a state machine isomorphic to the one above.
- You have to complete the implementation of the FM radio in JAVA. We recommend to use the Eclipse tool
  - 1.- Unzip the file “inf2140assignment1.zip”
  - 2.- Import the project “inf2140assignment1” (into Eclipse).
  - 3.- The project runs with the file “AssignmentOne.java”.
  - 4.- Implement the file “RadioController.java”.

**Note:** Your final implementation must run when your project is imported into Eclipse, otherwise you must clearly state in the README file the exact commands to run your implementation.
- The delivery must only be in one file: either a .tgz or .zip archive (with one of those two file extensions). First you copy the contents of your assignment to a folder with the same name as your username (for groups with two or three students, choose one username of the two or three usernames), then delete all compiled files from the java project and pack the folder in a .tgz or .zip archive.
- Your archive should contain
  - The source file of your FSP implementation.
  - The source files of your JAVA implementation (the whole project without the compiled files).
  - A report, named either report.txt or report.pdf , in which you should include the full name(s) and username(s) of the contributing students together with a clear explanation of your FSP and JAVA implementation.
  - A file named README in which you should describe any peculiarities of your solution, for instance, things that may be missing, or assumptions made, or questions for the group teacher. If everything runs fine, state it in the README file.

**Evaluation:** This assignment is graded pass or fail. You must pass this assignment in order to take the final exam.

*Good luck!*