

ESS: Exercise Set 6

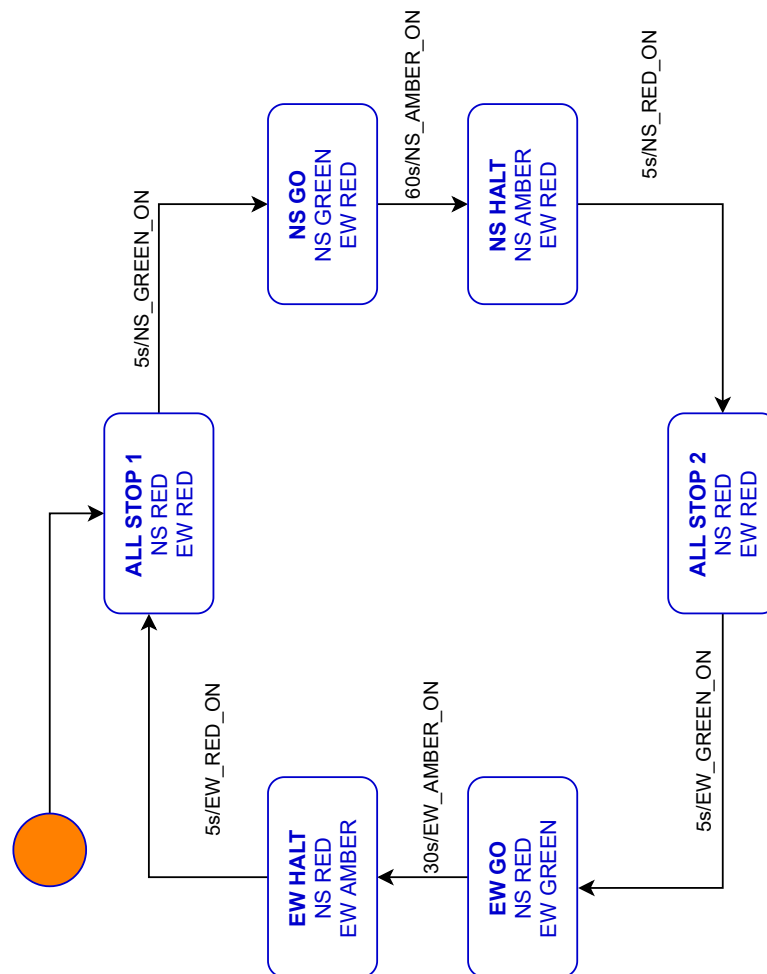
Reactive Embedded Systems

Question 1:

Design an FSM to control the flow of traffic over a cross-road intersection.

Solution:

Here is a very simple solution. It assumes there are two important directions for the cross-road: traffic which is flowing North-South (NS) and traffic which is flowing East-West (EW). When a certain amount of time has elapsed, it transitions to a new state. Note that the ALL STOP state is duplicated - one of the major issues of FSMs is state duplication. This is obviously a very high level example, and there could be much more enhancement e.g. to use sensors to detect whether cars are present or not, and whether or not a pedestrian wants to cross the road. However, even from this simple diagram, it is possible to guarantee that both NS and EW directions will never both be green at the same time.



Question 2:

NMEA is a format for sending data from navigational equipment (such as GPS receivers) over a serial port to a microcontroller. It is necessary to check that an NMEA message is received correctly and is not garbled, or a ship/aeroplane could be put on an incorrect course. The following rules specify the syntax of an NMEA message:

- Each message's starting character is a dollar sign.
- The next five characters identify the sender (two characters) and the type of message (three characters).
- All data fields that follow are comma-delimited.
- Where data is unavailable, the corresponding field remains blank (it contains no character before the next delimiter – see Sample file section below).
- The first character that immediately follows the last data field character is an asterisk.
- The asterisk is immediately followed by a checksum represented as a two-digit hexadecimal number.
- <CR><LF> ends the message.

These are examples of correct NMEA sentences.

\$GPAAM,A,A,0.10,N,WPTNME*32

\$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,1.72,1.03,1.38*0A

This is an incorrect NMEA sentence (missing asterisk) before two digit checksum

\$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,76

Write an FSM to parse an NMEA message and determine its validity.

Solution:

Here is an example GPS parser - it is from an open-source GPS NMEA sentence parser: <https://visualgps.github.io/NMEAParser/>. The red arrows denote error transitions i.e. that if there is any failure in the message construction, it returns to the initial state.

