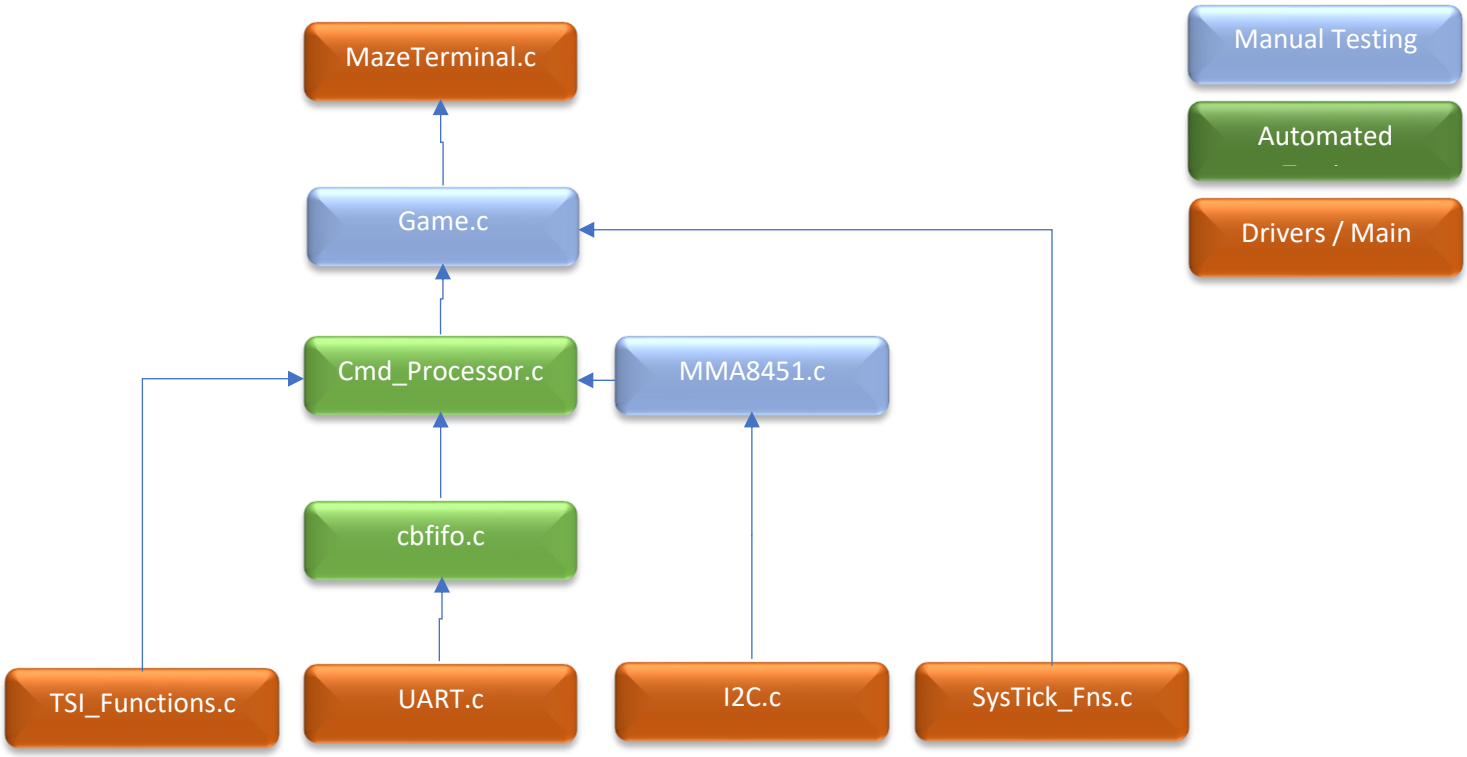
Final Project Test Plan

Principles of Embedded Systems (ECEN – 5813)

Spring 2021

Dhruv Mehta

Test Outline



As shown in the modular diagram above, the modules in green are unit tested using automated test cases previously written and new test cases written in uCUnit framework. The Drivers section is not tested as a suitable test plan cannot be derived for these sections. The Manual testing components are the upper application layers which consist of the following:
MazeTerminal.c – This is the main file which calls the driver initializations and the game. This module cannot be tested as it has no functionality of its own.

Game.c – This consists of the state machine cycling through the four available states of the game. This module is manually tested by playing the game and through incremental development method. All accessible states of the game are tested manually by playing the game and observing whether the functionality meets what is expected.

MMA8451.c – This consists of the accelerometer data gathering and conversion to roll and pitch angles. This was again manually tested by physically verifying that the obtained angles were correct by moving the accelerometer through a range of angles.

Test Results

Cbfifo.c -> Extensively tested for different range of cases included extreme cases such as dropped bytes, enqueuing beyond capacity, dequeuing beyond length etc. All 5 test cases have passed.

Cmd_Processor.c -> Tested using uCUnit framework (specifically isEqual function) to check the result of each movement operation and all handlers in the command processor. It is tested for the normal cases as well as edge cases such as movement of the cursor beyond the terminal borders and moving the cursor into objects. All 20 test scenarios passed.

Game.c -> Number of states (4) is known and entered into as per the sequence in the game. Tested for different scenarios manually (touching the sensor in between the game/holding the accelerometer upside down). Since the states are programmed to follow a particular sequence, there is no deviation from the expected functionality. Also the code has no interrupts apart from the UART handler which does not interfere with Game.c (no shared variables)

MMA8451.c -> Tested by moving the accelerometer through a range of values. The final roll and pitch angles are clipped off to +- 45 degrees before interfacing with the game logic.