

Name: \_\_\_\_\_

ID: \_\_\_\_\_

## ECE6483 EXAM 2, Fall 2021 Take Home

Directions: Answer all questions completely and submit via Brightspace. Open book, notes etc. Please do not collaborate. There will be no credit for correct answers without explanation. **There will be no exceptions for late submissions.**

1. In class we often use the built in functions provided by the MBED framework. This allows us to avoid tedious bit manipulations and register configurations. Often when we use sensors or actuators, there is an accompanying driver which serves the same purpose as the framework does. Let's consider the Azoteq series of gesture/touch controllers (datasheet attached).

This sensor communicates I2C to your host controller. The goal of this question is to make a C file and associated header file that provides a specific driver for that sensor.

Review the I2C section of the datasheet (I have attached the 5 relevant pages).

- a. Build a header file (ASensor.h) that provides the following:
  - a. A #define for the I2C address, #define SENSOR\_ADDRESS xxxx
  - b. Definitions for the following functions
    - i. **Int GetVersion()** //Returns Major Version.Minor Version of the sensor
    - ii. **int GetTotalStrength()** //Returns the total strength (See datasheet table 8.1)
    - iii. **int GetEvent()** //Returns 0 if there is no event or an enumerated value indicating the event type defined by the bits in register addresses 0x000D – 0x000E in the sensor (see datasheet table 8.1)
- b. Build a C file (ASensor.c) that implements the header you built in part a.
  - a. Using any of the MBED I2C functions provided (<https://os.mbed.com/docs/mbed-os/v6.15/apis/i2c.html>), write the code for sections i. through iii. above.
- c. Write the main.cpp in PlatformIO that performs the following:
  - a. Initialize the I2C driver.
  - b. Initialize the Serial driver (to output data to the terminal).
- d. In the while loop in your main(), call the GetEvent() function every 0.5 second and outputs the result of the GetEvent function to the serial terminal.

These are the ONLY requirements. You are free to add features or implement the functions any way that you see fit. You can make assumptions as long as you clearly state and document them.

2. Lets now assume we want to bypass the polling and use a timer/ticker function provided from MBED instead to call the GetEvent() function. See <https://os.mbed.com/docs/mbed-os/v6.15/apis/ticker.html>.
  - a. Write the code to setup a 0.5 second ticker and implement the "attach".
  - b. Write the required code to identify an event and print out the results from the main.cpp while loop. (NOT THE TIMER FUNCTION)

3. For the above example, add `GetTotalStrength()` to your while loop and illuminate the on-board LED with it's brightness relative to the return from `GetTotalStrength()`. Meaning, use MBEDs PWM implementation to "dim" the LED such that weak touch = DIM, strong touch = BRIGHT.