EECS 373 Midterm 2

2 March 2024

80 minutes permitted.

No cellphones, internet, or communicating with others about the exam (except course staff). One double-sided 8.5"×5.5" note sheet is permitted. No other access to course material is permitted.

Name

UM Uniqname

Sign below to acknowledge the Engineering Honor Code: "I have neither given nor received aid on this examination, nor have I concealed a violation of the Honor Code." "Concealed" should be interpreted as "have failed or will fail to report".

Signature

1 [14 pts.] Serial communication

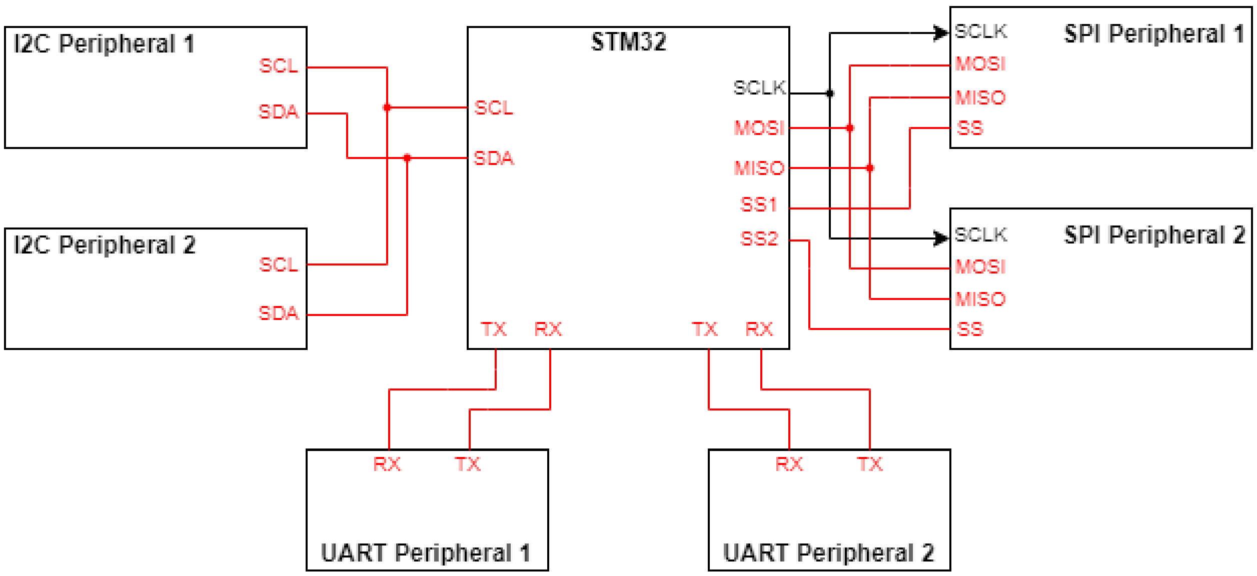


Figure 1: Block diagram.

1. [2 pts.] You have an STM32 and six different peripheral devices. Two devices use I2C, two devices use SPI, and two devices use UART. In Figure 1, draw in and label the appropriate wires needed to communicate with each device individually. Assume the I2C peripherals have different device addresses. Minimize the number of wires for full credit. The SCLK line has been drawn for you. Omit power and ground wires. There is no need to draw arrowheads on the wires.

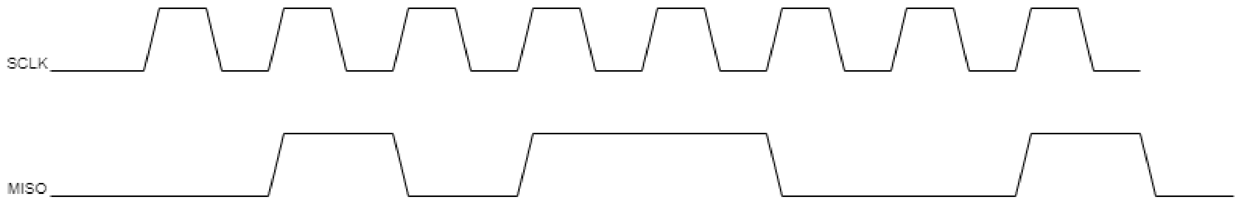


Figure 2: Timing diagram.

2. [2 pts.] Consider the clock and data signals in Figure 2. Which clock edge should the data be sampled on?

   ◯ Rising.
   ◯ ✓Falling.
   ◯ Both.

3. [2 pts.] Use at most three sentences to describe what could go wrong if data are sampled on the incorrect clock edge.

   If the data is sampled on the incorrect clock edge, the data line could be in the process of transitioning. Because of this, there is no guarantee as to what value the data would be read in as, leading to potentially faulty data.

4. [2 pts.] Which serial communication protocol is typically the fastest?

   ◯ I2C.

   ◯ ✓SPI.

   ◯ UART.

5. [2 pts.] Which serial protocol is best for communicating over long distances (15 m+)?

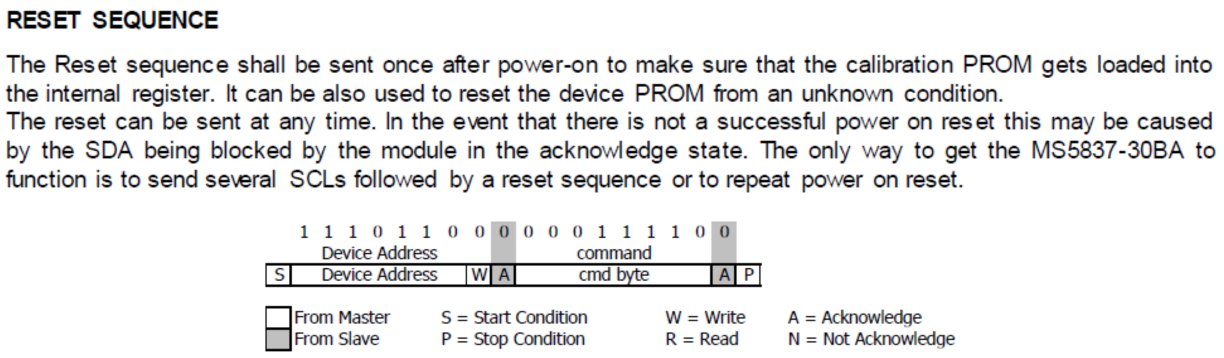
   ◯ ✓UART.

   ◯ SPI.

   ◯ I2C.

6. [2 pts.] Use at most two sentences to explain the difference between full-duplex and half-duplex.

   Full-duplex communication allows transmission of data two-ways simultaneously. Half-duplex only allows two-way communication, but not simultaneously. Half-duplex requires devices to switch between a sending and receiving mode.



RESET SEQUENCE

The Reset sequence shall be sent once after power-on to make sure that the calibration PROM gets loaded into the internal register. It can be also used to reset the device PROM from an unknown condition.
The reset can be sent at any time. In the event that there is not a successful power on reset this may be caused by the SDA being blocked by the module in the acknowledge state. The only way to get the MS5837-30BA to function is to send several SCLs followed by a reset sequence or to repeat power on reset.

Figure 3: Material from datasheet.

7. [2 pts.] Based on Figure 3, which protocol does this sensor support?
   - ◯ ✓I2C.
   - ◯ SPI.
   - ◯ UART.
   - ◯ Parallel.
   - ◯ 1-Wire.

2 [15 pts.] Sampling, ADCs, and DACs

You are given an ultrasonic distance sensor that outputs a voltage from 0 V to 3.3 V corresponding to the distance measured. To interface with the sensor, you decide to use a 10-bit ADC with a 0 V to 4 V input range. The ADC is not half-LSB compensated.

1. [3 pts.] Which of the following changes to the ADC would decrease step size while preserving functionality for this application? Consider each option individually, and assume all other ADC characteristics are held constant. Select all that apply.
   - ◯ ✓Change the range to 0 V–3.3 V.
   - ◯ Change the range to 0 V–2 V.
   - ◯ ✓Increase number of bits to 12.
   - ◯ Decrease number of bits to 8.
   - ◯ Apply half LSB compensation.

   The ADC remains 10-bit with range 0 V–4 V without half LSB compensation for the remainder of the question.

2. [5 pts.] What is the maximum quantization error? Show your work. Your final answer may be unsimplified.

   Max quantization error is 1 LSB; $4\,\mathrm{V}/2^{10} = 0.00390625\,\mathrm{V}$**.**

3. [3 pts.] Indicate what to replace ??? with in the following C function to enable it to calculate the distance measured by the sensor.

```
float get_distance(void) {
  volatile const int adc_val;
  float voltage = ???;
  return voltage_to_distance(voltage);
}
```

   ??? = **(4.0 / 1024.0) * adc_val;**

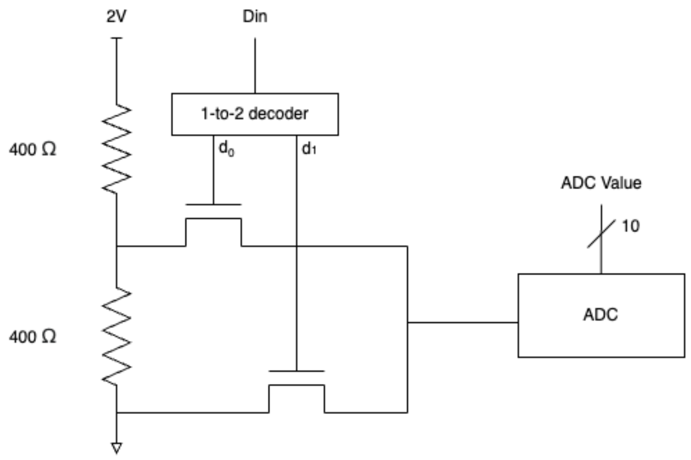


Figure 4: DAC.

4. [4 pts.] Unfortunately, your distance sensor doesn't work. To verify the functionality of your ADC, you connect a 1-bit DAC in place of the sensor, shown in Figure 4. What should the ADC output values be for all possible inputs to the DAC? Please give ADC values in decimal, and show your work.

   For Din = 0: **Dout = 1 V, ADC val =** $1/(4/2^{10}) = 2^8 = 256$

   For Din = 1: **Dout = 0 V, ADC val = 0**

3 [8 pts.] Prototyping

Indicate the correct ways to complete the following sentences.

1. [2 pts.] A convex solder connection for a through-hole component...

   ◯ will ◯ ✓may ◯ will not

   ...electrically connect the pad/trace and the wire/lead.

2. [2 pts.] ◯ will ◯ ✓may ◯ will not

   ...produce a highly capacitive, but not resistive, electrical connection between the pad/trace and wire/lead.

3. [2 pts.] ◯ will ◯ ✓may ◯ will not

   ...electrically connect (with low resistance) the pad/trace and the wire/lead. **This one was kind of repetitive with the first. Oops.**

4. [2 pts.] Use at most one sentence to describe the proper method for handing an ESD-sensitive component to a collaborator. You do not need to explain why it is the correct method, just write the method.

   "Handshake" or "touch the collaborator's hand before and while giving them the component".

4 [8 pts.] Memory

For each of the following application descriptions, select the most appropriate memory type to use.

1. [2 pts.] An energy scavenging application without any battery, in which power will be unpredictably lost and in which state must persist across power outages.

   ◯ Quantum phase relaxation memory.

   ◯ PROM.

   ◯ SRAM on the same die as the processor.

   ◯ Magnetic core memory.

   ◯ DRAM.

   ◯ ✓Flash memory.

   I had intended the content to change during operation, but I didn't say that so "PROM" is also correct and received full credit.

2. [2 pts.] An application in which speed is of critical importance, but high cost is tolerable and power is alway available.
   - ◯ Quantum phase relaxation memory.
   - ◯ PROM.
   - ◯ ✓SRAM on the same die as the processor.
   - ◯ Magnetic core memory.
   - ◯ DRAM.
   - ◯ Flash memory.
3. [2 pts.] An application requiring a lot of memory, where cost is important and power is always available.
   - ◯ Quantum phase relaxation memory.
   - ◯ PROM.
   - ◯ SRAM on the same die as the processor.
   - ◯ Magnetic core memory.
   - ◯ DRAM.
   - ◯ ✓Flash memory.

   Flash is the best answer, but DRAM also received full credit. DRAM would have been the best answer if speed were somewhat important.
4. [2 pts.] An extremely high security application in which it is unacceptable for the executed code to be modified, where price is no object.
   - ◯ Quantum phase relaxation memory.
   - ◯ ✓PROM.
   - ◯ SRAM on the same die as the processor.
   - ◯ Magnetic core memory.
   - ◯ DRAM.
   - ◯ Flash memory.

5 [5 pts.] PCBs

1. [2 pts.] What is the most common method for producing copper trace and pad patterns on PCBs?
   - ◯ ✓Optical lithography.
   - ◯ Milling.
   - ◯ Cyanacrylate bonding.
   - ◯ Sputtering.
   - ◯ Electroplating.

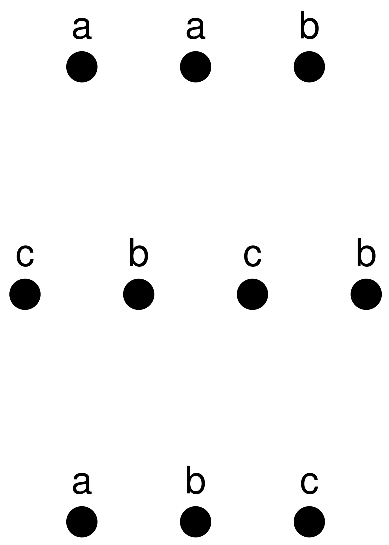


Figure 5: PCB routing.

2. [3 pts.] See Figure 5. Each dot represents the position on a PCB of a through-hole to which a component pin will be soldered. Dots sharing the same letter must be connected by copper traces, but dots with different letters must not be electrically connected. What is the minimum number of copper layers to appropriately connect all the dots? You can show your work directly on Figure 5.

   Minimum number of layers: **1. There are some arrangements of nodes that do not admit planar graphs but this is not one of them.**

6 [15 pts.] Shaft encoders and motors

1. [4 pts.] Match the following components to their best uses. One is completed for you to make the format clear.

   A. Stepper motor. B. Pen. C. Servo. D. Solenoid. E. DC motor.

   - Writing. **B**
   - Move to two discrete positions. **D**
   - Precise control at low speed without feedback. **A**
   - Precide control at high speed with feedback. **C**
   - High speed but imprecise positioning, without built-in feedback. **E**

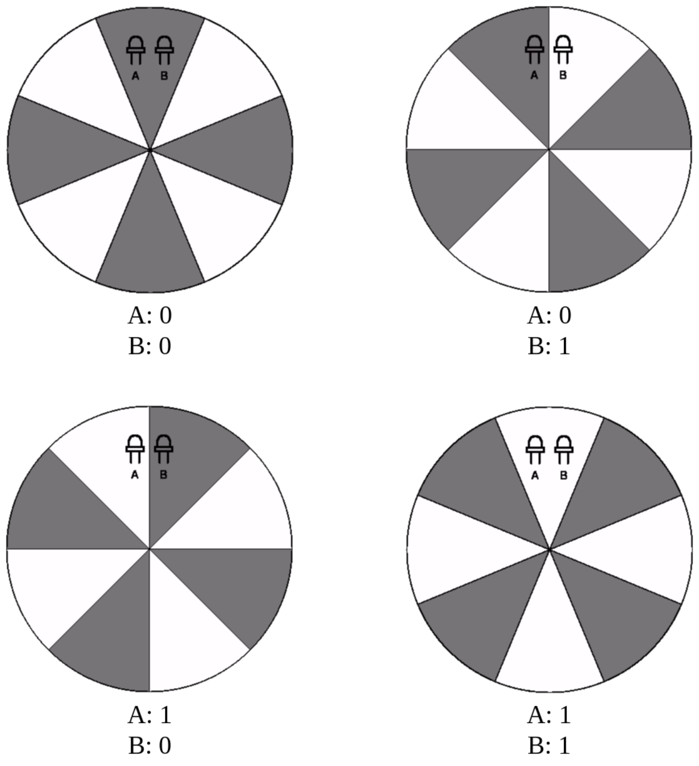


Figure 6: Shaft encoder.

2. You are tasked with developing a system to interpret the outputs of the shaft encoder. The wheel has a circumference of 7 cm and is connected directly to the shaft of the encoder. In other words the wheel travels roughly 7 cm during one full rotation. Counterclockwise rotation of the wheel is considered positive motion and clockwise rotation is considered negative motion. The following definitions and functions have already been implemented and are available for you to use.

```
// gpio_num should be a valid gpio number
// gpio_val is 0 for LOW and 1 for HIGH
void gpio_write(uint8_t gpio_num, uint8_t gpio_val);

// gpio_num should be a valid gpio number
// return value is 0 for LOW and 1 for HIGH
uint8_t gpio_read(uint8_t gpio_num);
```

You have an encoder and must write code to keep track of the wheel's position over time. This encoder differs somewhat from the one presented in lecture, but the same sort of reasoning presented in lecture will guide you to and understanding of this encoder. Figure 6 shows the encoder's four output states. It has a two-bit output.

There is a photosensor for each LED on the other side of the encoder (not shown). The LEDs and photosensors are fixed in place and the disk rotates between them. Each photosensor goes high when light from the LED shines on it and low when light from the LED is blocked. Light from the LED shines through the white and is blocked by the gray. Photosensor A is connected to GPIO 0 and photosensor B is connected to GPIO 1. Both GPIOs are configured as inputs that produce interrupts on rising and falling edges.

In the following steps you will write code to track the relative position of the wheel. Declare any global variables and implement helper functions you would like here.

There are many different ways to approach this problem. Two of the more common solutions used are shown below, but they are not the only methods to correctly solve the problem. There is a subtle difference between the two solutions when it comes to determining whether to increment or decrement the position. This depends on whether the previous state or current state is being used in the logic.

```
// Solution 1:
uint32_t encoder_counts;
uint8_t prev_a;
uint8_t prev_b;

// Solution 2:
uint32_t encoder_counts;
```

[3 pts.] Implement the reset function. When this function is called, the relative position should be reset to 0 cm. You may assume reset() will be called first before any of the other functions.

```
void reset(void) {
// Solution 1
encoder_counts = 0;
 prev_a = gpio_read(0);
 prev_b = gpio_read(1);

 // Solution 2:
 encoder_counts = 0;
}
```

3. [4 pts.] Implement the ISR handlers for the two interrupts.

```
void ISR_PHOTOSENSOR_A(void) {
// Solution 1:
 if (prev_a == prev_b) {
 --encoder_counts;
   } else {
    ++encoder_counts;
   }
   prev_a = !prev_a;

   // Solution 2:
   a_val = gpio_read(0);
   b_val = gpio_read(1);
   if (a_val == b_val) {
    ++encoder_counts;
   } else {
    --encoder_counts;
   }
}
```

```
void ISR_PHOTOSENSOR_B(void) {
// Solution 1:
 if (prev_a == prev_b) {
++ encoder_counts;
   } else {
    -- encoder_counts;
   }
   prev_b = !prev_b;

   // Solution 2:
   a_val = gpio_read(0);
   b_val = gpio_read(1);
   if (a_val == b_val) {
    --encoder_counts;
   } else {
    ++encoder_counts;
   }
}
```

4. [4 pts.] Implement encoder_get_position(). This function should return the most accurate, up-to-date position possible in cm relative to the wheel's starting position. This may be fractional, i.e., the resolution limit comes from the number of shaft encoder segments.

```
double encoder_get_position(void) {
// Solution 1 and 2 (same either way):
return (double)(encoder_counts) * 7.0 / 16.0;
}
```

7 [15 pts.] Power

You are designing a battery-powered embedded system that must operate at least a day without maintenance. Select a wireless interface and battery to meet the battery lifespan requirement with minimal cost.

The processor will need to be active 10% of the time and sleep 90% of the time. It costs \$10, has an active power consumption of 150 mW, and a sleep mode power consumption of 30 mW. The wireless interface will need to transmit 100 Mb per hour (that's 100,000,000 bits). Assume instant transitions among power states. Be careful with your units, e.g., seconds, minutes, hours, days, joules, and watts. This is a pretty realistic problem, and mixing up units will cause serious errors.

Indicate which two components to use.

1. [2 pts.] Wireless interface options.
   - ◯ ✓Transceiver A costs \$20 and transmits at 500 kb/s, has an active power consumption of 1 W, and a sleep mode power consumption of 0 W.
   - ◯ Transceiver B costs \$50 and transmits at 2 Mb/s, has an active power consumption of 2 W, and a sleep mode power consumption of 0 W.
2. [2 pts.] Battery options.
   - ◯ Battery C costs \$10 and has a capacity of 600 mA-H and a nominal voltage of 3 V.
   - ◯ ✓Battery D costs \$18 and has a capacity of 800 mA-H and a nominal voltage of 3 V.
3. [5 pts.] Indicate the total cost of the system.

   \$48
4. [6 pts.] Indicate the battery lifespan of the system in days.

   1.0251 days

It was necessary to account for the processor, even though its use was not a choice. The different units (days, hours, seconds) were fiddly, but that's life. Mistakes on these are common and consequential in real system design. The transceivers had zero power consumption when off, so their average powers were their on state power consumptions multiplied by their duty cycles, which depended on their transmission rates.

Show your work on this and the following page. For partial credit, indicate the meanings of partial solutions.

This page may be used for work. Please hand it in with the exam and reference it from the associated questions if you would like it to be considered when determining partial credit.