ECE/CSE 474 WINTER 2020

COURSE EXAM

J. Vining
ECE/CSE 474, Embedded Systems
University of Washington – Dept. of Electrical and Computer Engineering
Winter 2020

5 March 2020

Score (of out 100 pts): ____________________

LAST Name: ____________________________
FIRST Name: ____________________________

1.0 TRANSLATING CODE INTO AN EXECUTABLE (10PTS)

Describe the steps taken to get C code into an executable

2.0 KERNEL SERVICES (10PTS)

Name and ***describe*** THREE primary kernel services

3.0 MEMORY (5PTS)

If a memory address consists of X bits, how many unique memory locations can be addressed?

4.0 Atomic Operations, Part 1 (10pts)

Explain why accessing an 8-bit variable ***is*** or ***is not*** an atomic operation for the Arduino.
Hint: The ATmega chip is 8-bit addressable

5.0 Atomic Operations, Part 2 (10pts)

Why would you want to make a critical code section atomic?

6.0 Analog Input (15pts)

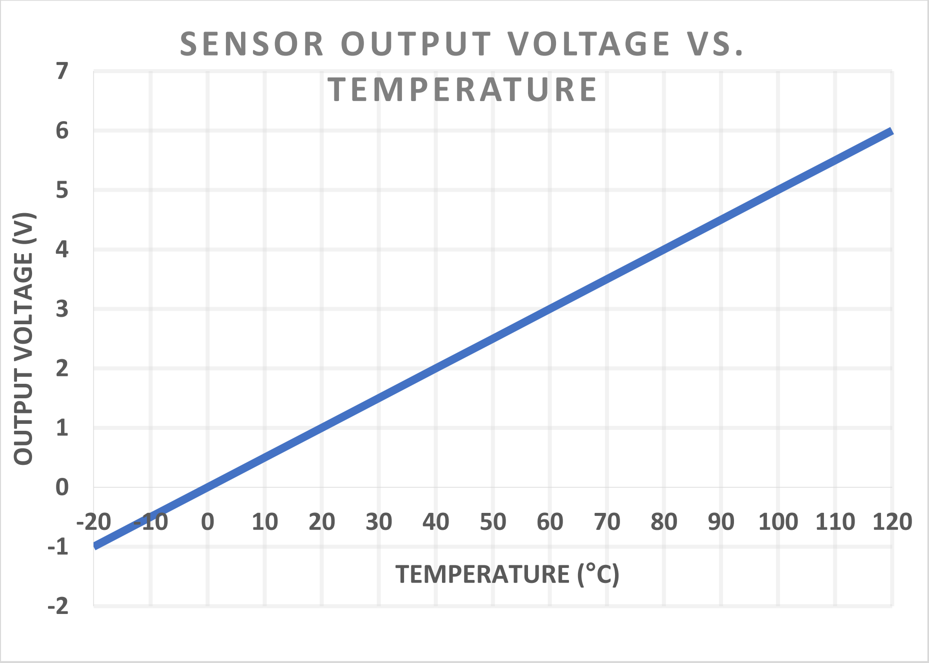
Given the analog input channel and sensor specifications below:

1. What is the **temperature range** that can be measured?
2. What is the **resolution** of the temperature sensor?
   You can leave the resolution calculation in equation form but it MUST HAVE UNITS!

Analog input characteristics:

- Voltage range: [0V, 5V]
- 8-bit ADC

Temperature sensor with the following output voltage characteristic:

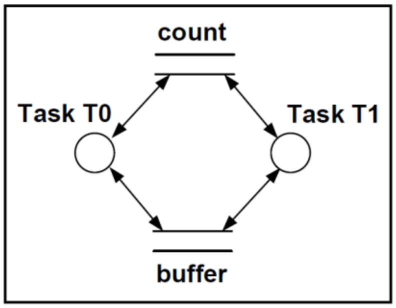


7.0 Resource Sharing & Data Integrity (15pts)

Two tasks, T0 and T1, can run concurrently. Both operate on a **shared buffer** and **shared count variable** as shown in the data flow diagram and pseudocode below. Occasionally the system loses data or receives garbage.

1. Explain the source of the data corruption
2. Explain how to solve the data corruption problem. The solution must work for cases when the tasks run concurrently.
3. Write pseudocode for both T0 and T1 to solve the problem. Include comments in your pseudocode to indicate what it is doing.

Data flow diagram:



Pseudocode for tasks T0 and T1

Task T0: Producer	Task T1: Consumer

`if count!=full`
`  add item to buffer`
`  increment count`
`else`
`  return`
	`if count>0`
	`  get item from buffer`
	`  decrement count`
	`else`
	`  return`

(Question 7 answer space, continued)

8.0 INTERRUPTS (10PTS)

Using the interrupt vector table below for the ATmega, choose the interrupt vector that would be able to produce the specified interrupt. In the case of multiple possible interrupt vectors, ***choose the one with highest priority***.

Interrupt Trigger	**Interrupt Vector Number**
Rising edge on input pin signal	
UART channel 0, data available	
Hardware timer	

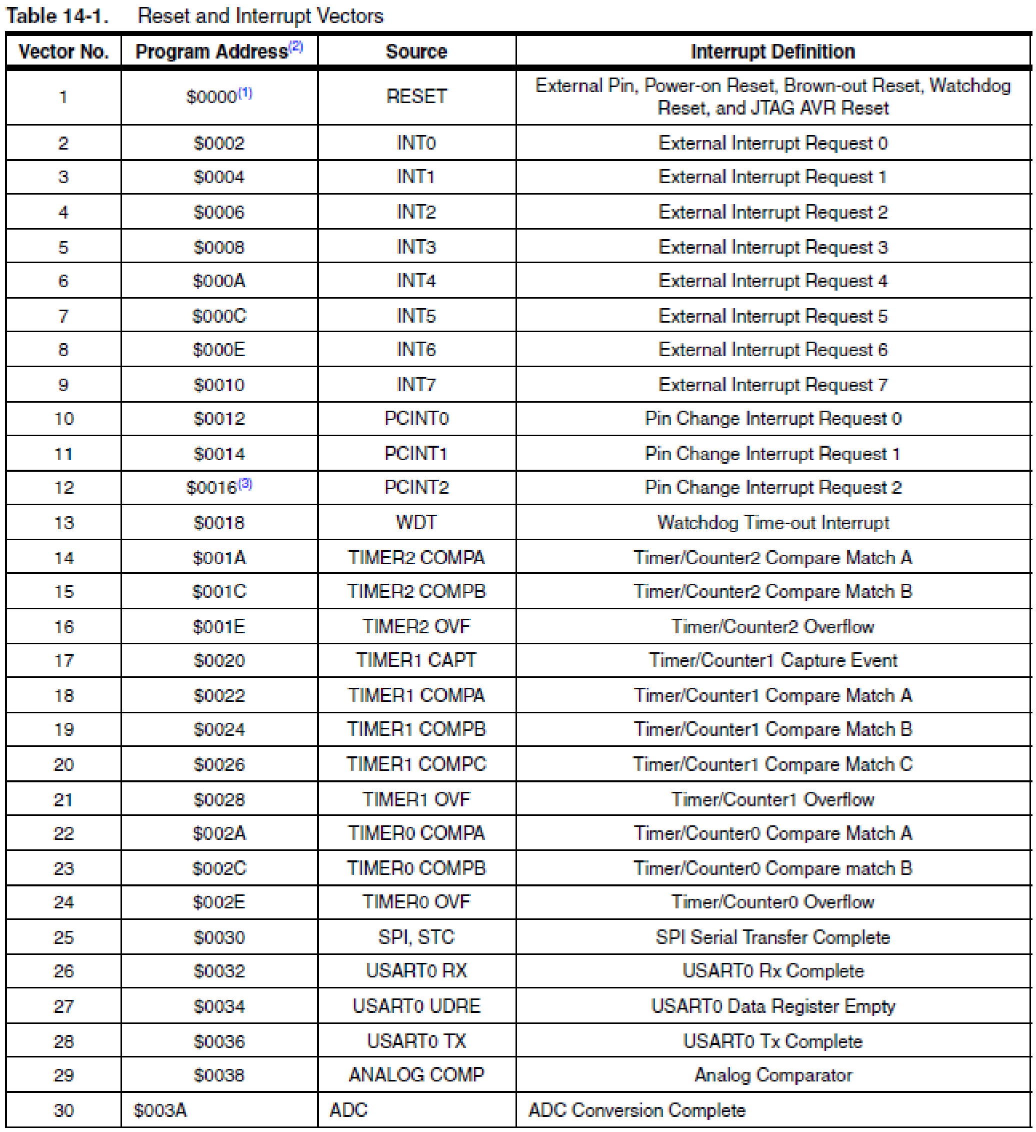


Table 14-1. Reset and Interrupt Vectors

Vector No.	Program Address[2]	Source	Interrupt Definition
1	$0000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	PCINT0	Pin Change Interrupt Request 0
11	$0014	PCINT1	Pin Change Interrupt Request 1
12	$0016[3]	PCINT2	Pin Change Interrupt Request 2
13	$0018	WDT	Watchdog Time-out Interrupt
14	$001A	TIMER2 COMPA	Timer/Counter2 Compare Match A
15	$001C	TIMER2 COMPB	Timer/Counter2 Compare Match B
16	$001E	TIMER2 OVF	Timer/Counter2 Overflow
17	$0020	TIMER1 CAPT	Timer/Counter1 Capture Event
18	$0022	TIMER1 COMPA	Timer/Counter1 Compare Match A
19	$0024	TIMER1 COMPB	Timer/Counter1 Compare Match B
20	$0026	TIMER1 COMPC	Timer/Counter1 Compare Match C
21	$0028	TIMER1 OVF	Timer/Counter1 Overflow
22	$002A	TIMER0 COMPA	Timer/Counter0 Compare Match A
23	$002C	TIMER0 COMPB	Timer/Counter0 Compare match B
24	$002E	TIMER0 OVF	Timer/Counter0 Overflow
25	$0030	SPI, STC	SPI Serial Transfer Complete
26	$0032	USART0 RX	USART0 Rx Complete
27	$0034	USART0 UDRE	USART0 Data Register Empty
28	$0036	USART0 TX	USART0 Tx Complete
29	$0038	ANALOG COMP	Analog Comparator
30	$003A	ADC	ADC Conversion Complete

Assuming that interrupts are enabled and that no interrupts are currently being serviced, if the following interrupts are ***triggered simultaneously***, which will be serviced first? Hint: What is the priority of each interrupt?

Interrupt Trigger Vectors, All Triggered Simultaneously	**Which will be serviced first?**
WDT PCINT2 TIMER2 COMPA TIMER2 COMPB	

9.0 REAL-TIME OPERATING SYSTEMS (5PTS)

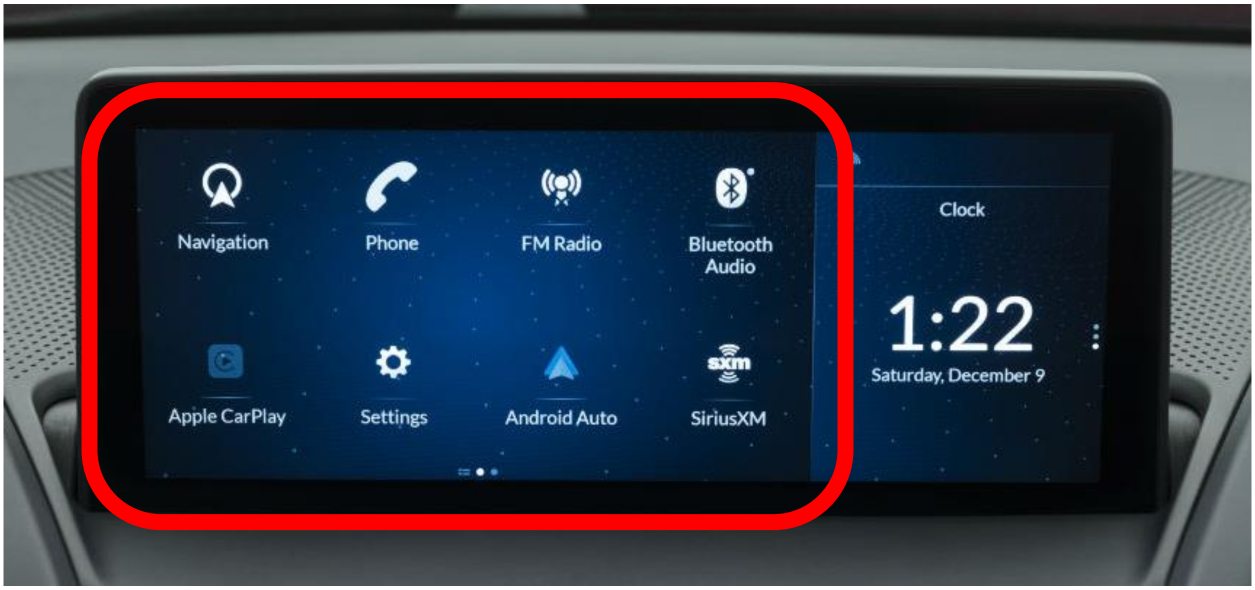
Explain what else needs to be known to determine if the following system is ***hard*** or ***soft*** **real-time**:

Given conditions of the operating system:

- Execution time of tasks in the operating system can be deterministically measured and modeled.
- The system has a predictable execution pattern.
- The system responds to external stimuli.

10.0 System Design & Scheduling (10pts)

You have won a contract to design an automotive infotainment center. The following shows the front panel display:



Functional decomposition: Name and describe the major high-level tasks required to implement the buttons in the box. Tasks must provide kernel functionality, a user interface, and functions to implement the button tasks.

Scheduling algorithm: Once a button has been touched by the user, explain how the system knows that (1) a button has been pressed and (2) which task the user wants to launch. Also explain how the kernel launches the desired task.

Discuss in terms of:

- Task scheduling & task queue (of major tasks listed in previous section)
- Shared variables
- Inter-task communication

Supplement your explanation with a **control/data flow diagram** between the tasks involved.

What **scheduler rating criteria** would be ***noticeable to the user*** for this application?