

Name: _____ ID: _____

EE4144

Quiz 2-Take Home

1. Suppose you would like to implement I2C, but unfortunately you do not have access to the I2C peripheral. Therefore, you have to implement the protocol in software. Assuming you are using Arduino pin 6 for SCK and Arduino Pin 7 for SDA write the necessary code for a., b., and c. (you can assume the pins are already set up as outputs)

- a. `void IICWriteClockLine(int value)`
`//Write the I2C clock line, value=(0,1) means (low,high)`
`{...}`
- b. `void IICWriteDataLine(int value)`
`//Write the I2C data line, value=(0,1) means (low,high)`
`{...}`
- c. `int IICReadDataLine(void)`
`//Read the I2C data line, return=(0,1) means (low,high)`
`{...}`

You also have setup these two timer ISRs that fire at transitions of a 100kHz time and an I2C Hail function: (They are already attached to the interrupts, no need to set that up)

- d. `void TimerTransitHighToLow(void) //ISR`
`{...}`
- e. `void TimerTransitLowToHigh(void) //ISR`
`{...}`
- f. `int HailIIC(unsigned char address)`
`{...}`

`//Send a 1 byte write request to see if "address" //acknowledges.`

`//return=(0,1) means (nack,ack)`

Implement both the ISRs in (d.) and (e.) as well as the HailIIC function in (f.) such that the HailIIC function uses the ISRs to control the I2C Data and Clock line. It should generate a start condition, followed by a write request to slave I2C address 0x74, then

reads the ack/nack bit, and finally ending with a stop condition. The function should return 0 if the I2C slave nacks or a 1 indicating an ack was received. You can add global or local flags and variables as necessary to keep track of the state or any other conditions. You do not have to implement the functions in (a.), (b.) or (c.). Also, be sure to explain any assumptions that you make in the I2C protocol.

2. **Read the attached datasheet for an I2C gesture pad.** Suppose you were using the above functions to control this gesture pad which is able to register swipes and taps on a solid state pad. Write a function called GetInfo() that uses the above functions to determine which event was performed - a swipe X (X+ or X-), swipe Y (Y+ or Y-), a tap, or a “tap and hold” (See section 8.9 of the datasheet for details). The GetInfo() function can return a value or an enumerated type to indicate which event occurred. Its OK to use any variables, structures or routines that you wish, however, please note any assumptions you make.
3. Notice the definition of Data Ready (RDY) in section 8.1 of the datasheet. It would be very efficient to use this to trigger to call the GetInfo() function you made in 2. Write the code to set up the pin change interrupt (any pin you want) and implement the ISR so that this ISR can be used to call GetInfo(). Can you think of any hazards in using this approach to determine if a gesture was made? Would any loop function be required?