ECE 461 Project 1 Report Template

Date

Your Name and Unity Email Address

Your Partner’s Name and Unity Email Address

*For extra credit, include the I2C bus signals in the screenshots.*

# Blocking Code Implementation

1. Screenshot showing three debug bits over duration of a read message.

# Finite State Machine Implementation

1. Explanation of how you pass input data to the FSM, and handshaking used (if any)
2. Explanation of how you get result data from the FSM, and handshaking used (if any)
3. Drawing of control flow graph (flow chart) with states identified
4. Drawing of finite state machine showing states and labeled transitions
5. Screenshot showing three debug bits over duration of a read message with no delay between calls to FSM
6. Screenshot showing three debug bits over duration of a read message with maximum delay between calls to FSM
7. What is the maximum delay between FSM calls which works?

# Interrupt Service Routine Implementation

1. Explanation of how you pass input data to the ISR, and handshaking used (if any)
2. Explanation of how you get result data from the ISR, and handshaking used (if any)
3. Screenshot showing three debug bits over duration of a read message
4. How much processor time is used to execute the I2C ISR to read the accelerations? What is the duration of the message on the bus?

ECE 561 Project 1 Report Template

Date

Your Name and Unity Email Address

Your Partner’s Name and Unity Email Address

*For extra credit, include the I2C bus signals in the screenshots.*

# Blocking Code Implementation

1. Screenshot showing three debug bits over duration of a read message.
2. How much processor time is used to execute the i2c\_read\_bytes\_fsm function to read the accelerations? What is the duration of the message on the bus?

# Finite State Machine Implementation

1. Explanation of how you pass input data to the FSM, and handshaking used (if any)
2. Explanation of how you get result data from the FSM, and handshaking used (if any)
3. Drawing of control flow graph (flow chart) with states identified
4. Drawing of finite state machine showing states and labeled transitions
5. Screenshot showing three debug bits over duration of a read message with no delay between calls to FSM
6. Screenshot showing three debug bits over duration of a read message with maximum delay between calls to FSM
7. What are the durations of the shortest and longest calls to the FSM function?
8. What is the maximum delay between FSM calls which works?
9. With that maximum delay, how much processor time is used to execute the i2c\_read\_bytes\_fsm function to read the accelerations? What is the duration of the message on the bus?
10. Extra Credit: Reduce the number of states which need to be executed to read the acceleration. Explain your approach and quantify the time benefits.

# Interrupt Service Routine Implementation

1. Explanation of how you pass input data to the ISR, and handshaking used (if any)
2. Explanation of how you get result data from the ISR, and handshaking used (if any)
3. Screenshot showing three debug bits over duration of a read message
4. How much processor time is used to execute the I2C ISR to read the accelerations? What is the duration of the message on the bus?