Question 1

1.a

In the master board, the pattern is round-robin. In the slave board, the pattern is round-robin with interrupts.

1.b

In the master board, the program will just read the values every iteration of the program, and send them over I2C to the slave board.

The slave board has a callback function that is ran every time it receives something over the I2C channel, and that function will read the data, and call functions to update the LEDs accordingly.

Question 2

(see file master.ino)

Question 3

(see file slave.ino)

Question 4

4.a

The mappings are all from [0, 1023] towards another domain, that is specified below:

Temperature sensor: [-50,450] (degrees Celsius).

Potentiometer: [0, 180] (degrees).

Light intensity sensor: [0, 255] (value for the analogWrite to the LED, using PWM).

4.b

Temperature sensor: none.

Potentiometer: none.

Light intensity sensor: 10 seconds before the start of the program, the programs enters a loop in which it just reads the light intensity and from there, the program scales according to the maximum and minimum values seen in those 10 seconds.

4.c

Temperature sensor: if the temperature goes over 24 degrees Celsius, turn the LED on (digitalWrite).

Potentiometer: 0 degrees (turned right) gives a period of 2 seconds for the blink; 180 degrees (turned left) gives a period of 0.2 seconds; scales according to the angle linearly.

Light intensity sensor: 255 (maximum intensity) turns the LED off; 0 (minimum intensity) turns the LED fully on; the intensity of the LED scales inversely with the light intensity (using analogWrite and PWM, as stated before).

4.d

All sensors have their values being outputted via serial, and the values do match. The system is responsive, which shows that it is working accordingly.

Question 5

The timing constraints on the system are related to the I2C protocol. Due to the usage of the Wire library in Arduino, all the low-level timing details regarding the I2C communication are abstracted. The sensor reading/actuator writing time can be neglected since this is much lower than the I2C protocol's timing constraints (e.g., approximately 2 microseconds for the light sensor to react).

Question 6

Data rate: 32 bytes / (3556/2) microseconds ~= 18KB/s

(data rate was measured by sending some bytes over I2C from the master to the slave and back, and dividing the number of bytes sent and received by that round-trip time - after dividing that time by 2)

Latency: 236 microseconds / 2 = 118 microseconds

(latency was measured by sending a byte over I2C and back, and measuring the round-trip time and dividing by 2)

Question 7

The LED starts flickering very quickly. That makes sense, considering that there is a delay between the value read by the sensor and the corresponding LED being turned on or off (with a variable intensity, in this case). Due to this delay, the LED responds very quickly to the sensor's readings, causing the flickering.

Question 8

The system's responses to the sensors' readings are delayed depending on the potentiometer's readings. If the delay was 0 and turned to 1 second, the system

would slow down. If the delay was 1 second and then diminished, the system would stop for a little while (due to the last second, in this case, of delay), and then resume the normal operation.