

```

#include <stdint.h>

#include <stdio.h>
#include "i2c2.hpp"
#include "io.hpp"

// TODO: Modify the handleInterrupt at the I2C base class
// 1. Add your slave init method
// 2. Add the "slave address recognized" state in your I2C slave driver and
    print a msg when you hit this state inside the ISR
// 3. To test that your slave driver init is working, invoke "i2c discover" on
    the Master board

// Slave Board sample code reference
int main(void)
{
    printf("Started main\n");
    I2C& i2c = I2C2::getInstance(); // Get I2C driver instance
    const uint8_t slaveAddr = 0xC0; // Pick any address other than an existing
        one at i2c2.hpp
    volatile uint8_t buffer[256] = { 0 }; // Our slave read/write buffer (This is
        the memory your other master board will read/write)

    // I2C is already initialized before main(), so you will have to add
        initSlave() to i2c base class for your slave driver
    i2c.initSlave(slaveAddr, &buffer[0], sizeof(buffer));

    // I2C interrupt will (should) modify our buffer.
    // So just monitor our buffer, and print and/or light up LEDs
    // ie: If buffer[0] == 0, then LED ON, else LED OFF
    uint8_t prev = buffer[0];
    buffer[2] = 0x16;
    while(1)
    {
        if (prev != buffer[0]) {
            prev = buffer[0];
            printf("buffer[0] changed to %#x by the other Master Board\n", buffer[0]);
        }
        if(buffer[2] == 0x16){
            LE.on(1);
        }
        else{
            LE.off(1);
        }

    }

    return 0;
}

```