MAINLINE

1. Map I2C2
    a. Read **conf_uart1_rtsn** (Address: =0x44E1_097C)
    b. Modify (set bits 0,1, 4, 5) #0x33 for MODE3 (pin 19 of P9)
    c. Write back
    d. Read **conf_uart1_ctsn** (Address: =0x44E1_0978)
    e. Modify (set bits 0,1, 4, 5) #0x33 for MODE3 (pin 20 of P9)
    f. Write back
2. Initialize I2C2 clock (**CM_PER** base: =0x44E0_0000)
    a. Write #0x02 to **CM_PER_I2C2_CLKCTRL** (offset: #0x44)
3. Configure I2C2 module (**I2C2** base: =0x4819_C000)
    a. Write #0x3 to **I2C_PSC** (offset: #0xB0)
        i. Sets clock divider to 4 so that $48 \div 4 = 12\ MHz$
        ii. $¿ = PSC + 1$
    b. Write #0x08 to **I2C_SCLL** (offset: #0xB4)
        i. Sets SCL low time to 50% duty with 400kbps and 12MHz clock
        ii. $tLOW = (SCLL + 7) * ICLK$
    c. Write #0x0A to **I2C_SCLH** (offset: #0xB8)
        i. Sets SCL high time to 50% duty with 400kbps and 12MHz clock
        ii. $tHIGH = (SCLH + 5) * ICLK$
    d. Write #0x40 to **I2C_OA** (offset: #0xA8)
        i. Sets it's own address to 0x40
    e. Write #0x8000 to **I2C_CON** (offset: #0xA4)
        i. Take I2C2 out of reset
4. Initialize I2C2 (**I2C2** base: =0x4819_C000)
    a. Read **I2C_CON** (offset: #0xA4)
    b. Modify (set bits 9 [**TRX**], 10 [**MST**]) #0x600
        i. Set to master transmitter
    c. Write back
5. Configure slave address and DATA counter (**I2C2** base: =0x4819_C000)
    a. Write #0x60 to **I2C_SA** (offset: #0xAC)
        i. Set slave address for current transmission
    b. Write #0x01 to **I2C_CNT** (offset: #0x98)
        i. Set data count to one byte
6. Initiate a transfer (**I2C2** base: =0x4819_C000)
    a. Poll **BB** (bit 12) of **I2C_IRQSTATUS_RAW** (offset: #0x24)
        i. See if line is busy; if bit = 0 continue, if bit = 1 keep polling
    b. Read **I2C_CON** (offset: #0xA4)
    c. Modify (set bits 0 [**STT**], 1 [**STP**]) #0x3
        i. Start transfer
    d. Write back
    e. Go to LOOP

LOOP

1. Create endless loop
    a. To let logic analyzer see results