MSP430 Advanced Technical Conference 2006

Hands-on: The New MSP430 Communication Peripherals

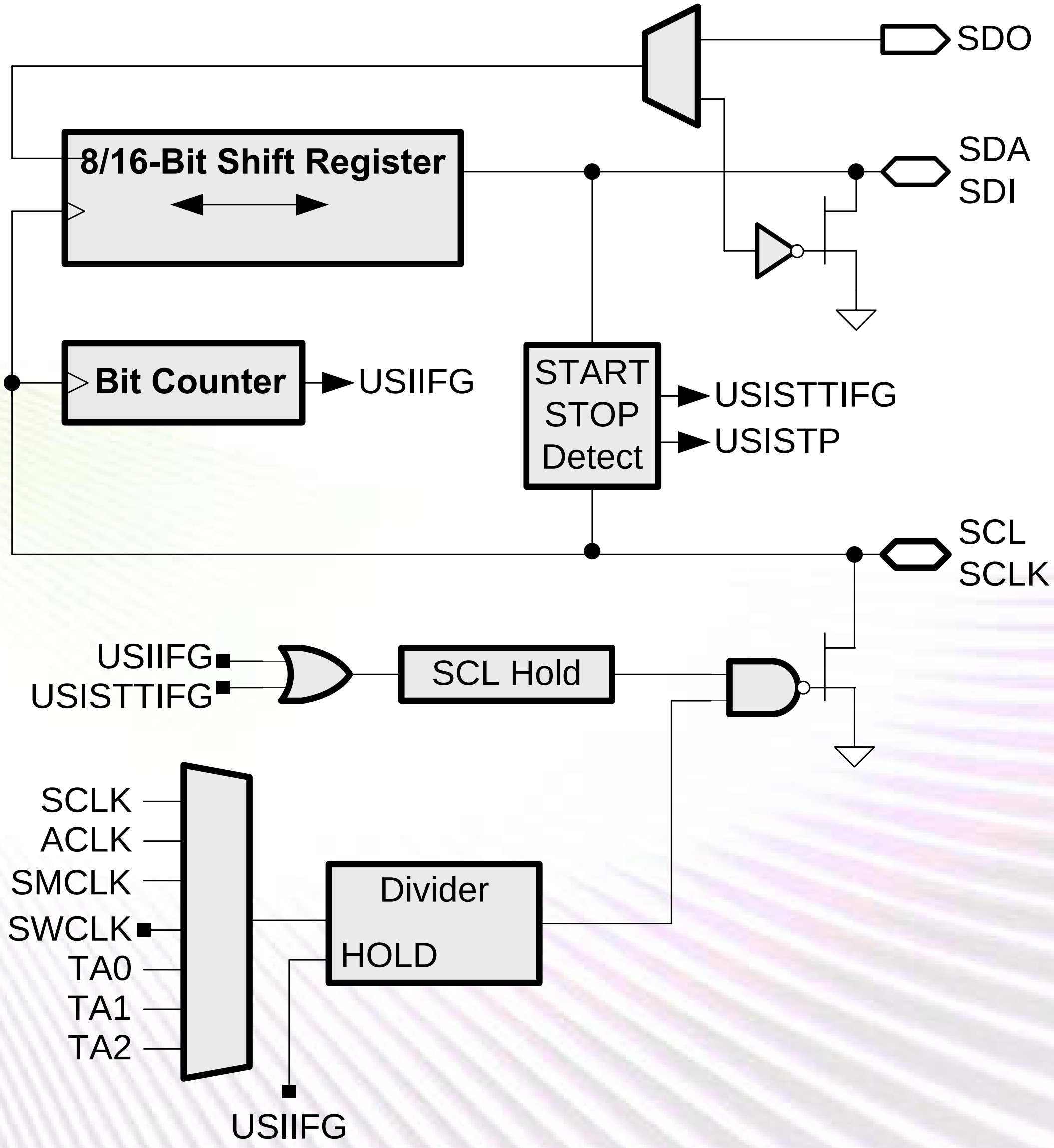
Zack Albus
MSP430 Applications Engineer
Texas Instruments

Agenda

- USI & USCI Basics
- Hands-On Lab: SPI using the USI
- Hands-On Lab: I2C using the USCI
- Wrap-up & Summary

USI Synchronous Data Transfer

- **Universal Serial Interface**
- **SPI Mode**
  - 8/16-bit shift register
  - MSB/LSB first
- **I2C Mode**
  - START/STOP detection
  - Arbitration lost detection
- **Interrupt Driven**
- **Reduces CPU load**

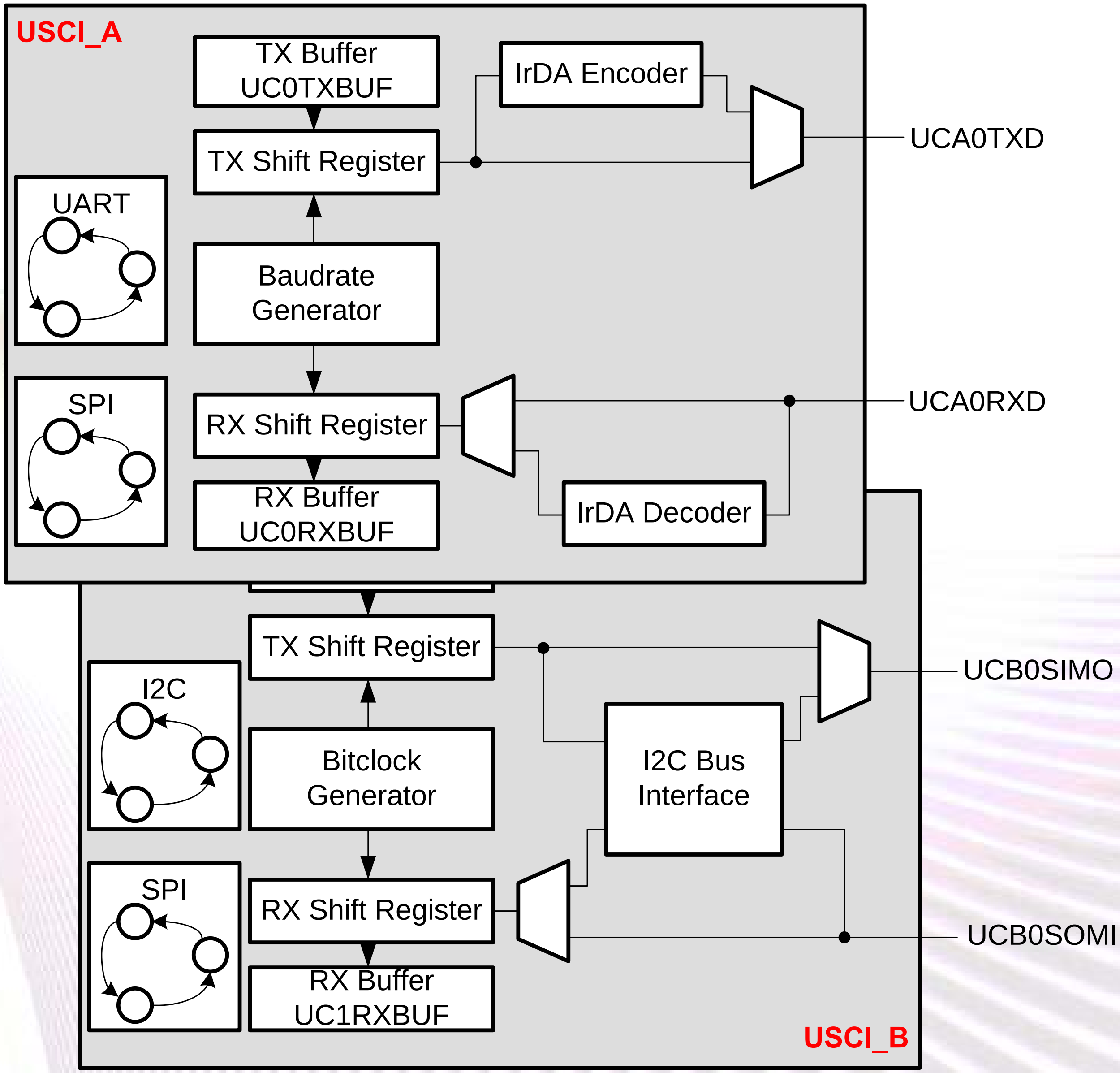


USI Careabouts

- USI provides SPI & I2C support in hardware
  - Protocol in user S/W, timing in USI H/W
- SPI: up to 16MHz clocking & 16-bit data I/O
- I2C: interrupt-driven protocol, critical timing in H/W
- Provides flexibility of 100% S/W solution while maintaining timing in hardware
- Smaller software implementation
- Lower CPU active time
- Faster communication speeds
- Full compatibility
- Internal pull-ups

USCI Universal Serial Comm I/F

- ***1 Module, 2 Independent Blocks***
- **USCI_A**
  - SPI
  - UART
  - LIN/IrDA
- **USCI_B**
  - SPI
  - I2C
- **2 interrupt vectors per A/B pair**
- **Full H/W solution**
- **USCI_A/B simultaneous operation**



© 2006 Texas Instruments Inc, Slide 5

USCI Careabouts

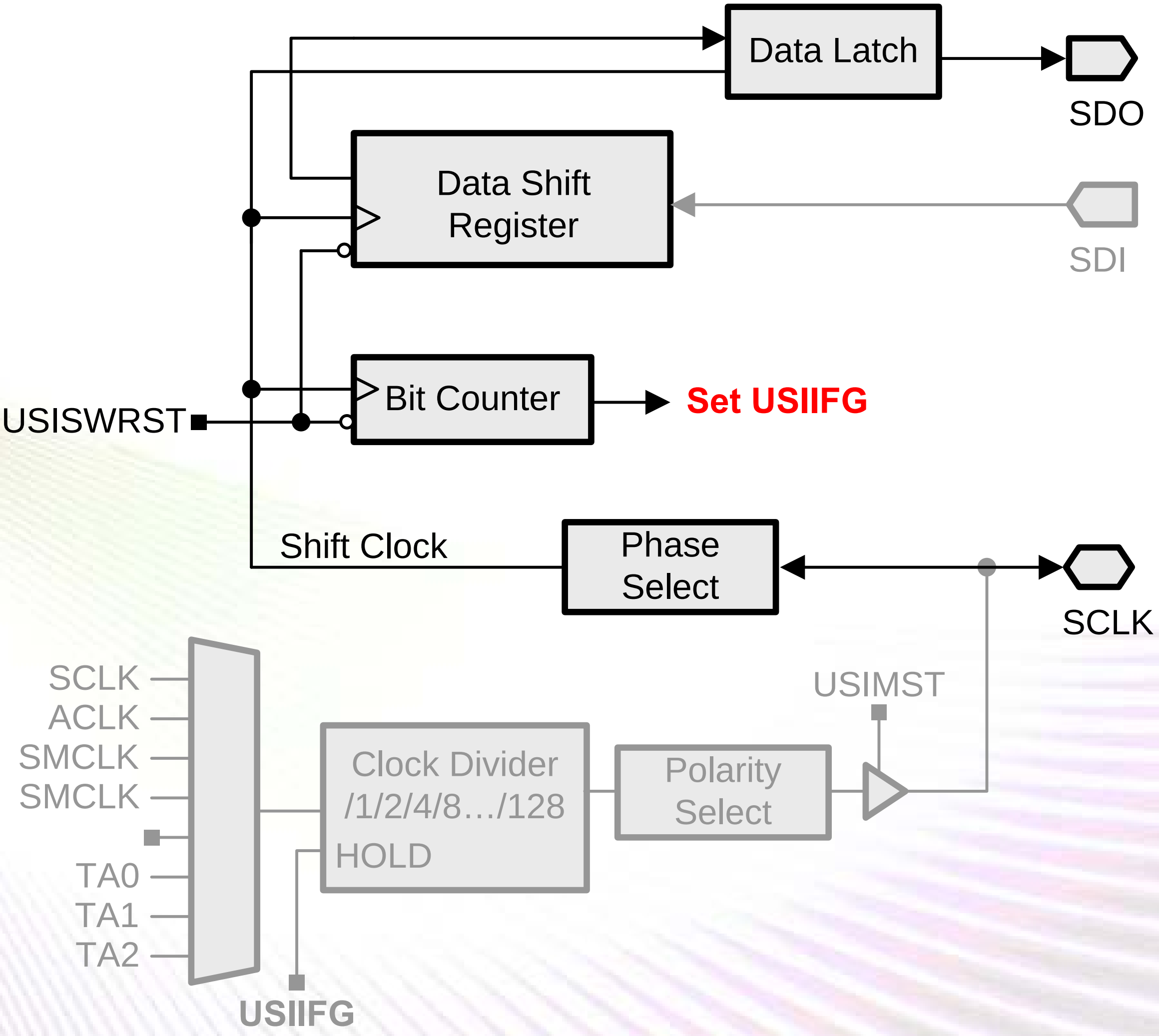
- USCI is the new standard MSP430 serial interface
- Two independent blocks operate simultaneously
- All modes capable of operating from any LPMx
- USCI is interrupt driven
- USCI is DMA enabled
- USCI_A supports SPI, UART, IrDA, auto baud LIN Bus
- USCI_A integrated baud rate generator with modulator for fractional bit rate division support
- USCI_B supports SPI, I2C

Agenda

- USC & USCI Basics
- Hands-On Lab: SPI
- Hands-On Lab: I2C
- Wrap-up & Summary

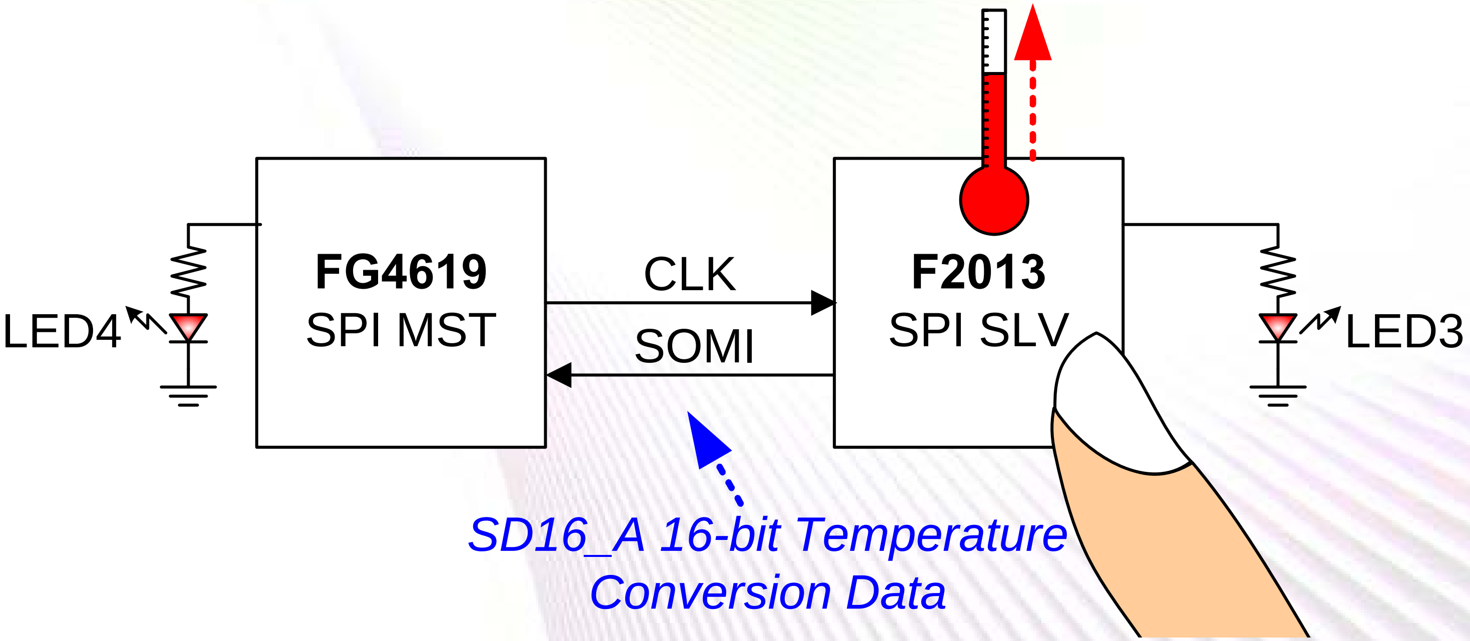
Lab Exercise 1: USI SPI

- **Complete a SPI data link between two MSP430s**
- **Complete partial 'F2013 USI SPI slave code**
  - SPI Slave Transmitter
  - Double check SCLK phase & polarity for compatibility
- **Load ready-to-use USCI SPI master code on the 'FG4619**



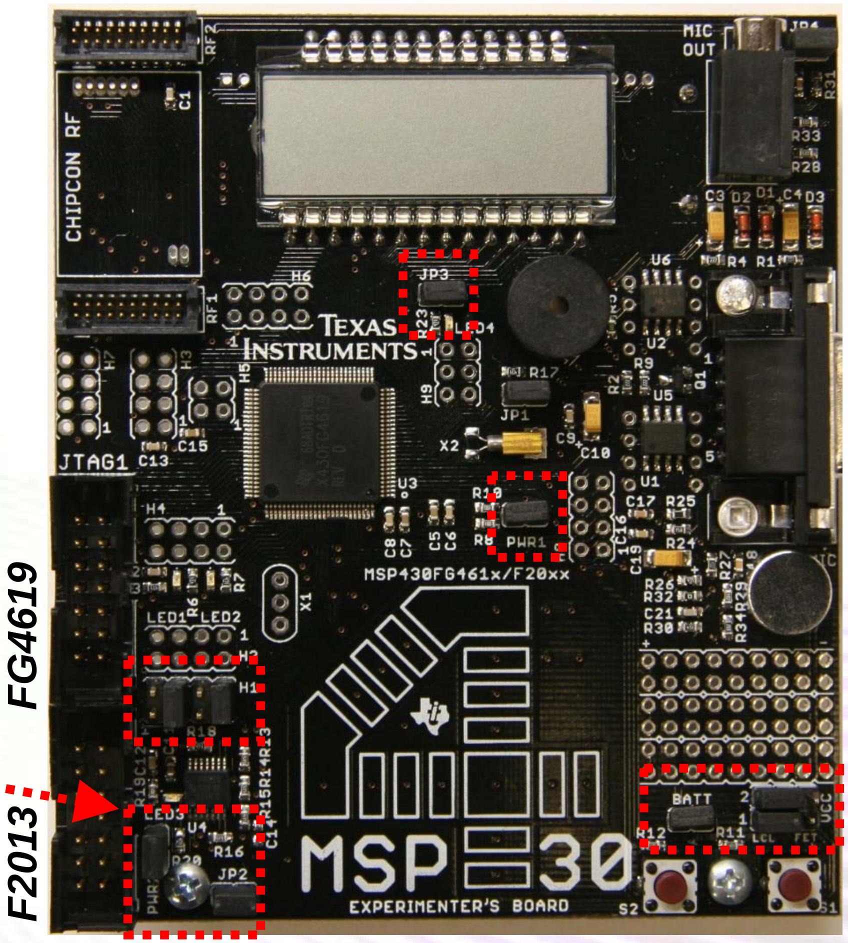
Lab Exercise 1: Overview

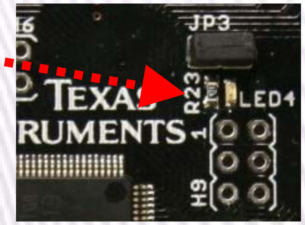
- Implement SPI Slave on the ‘F2013 to TX the 16-bit conversion result to the ‘FG4619 Master
- Slave flashes LED3 with each communication cycle
- When the result-to-result difference exceeds a preset delta, the Master turns on LED4



Lab Exercise 1: Board Setup

- **Needed Jumper Settings:**
  - PWR1/2, BATT, LCL1/2, JP2, JP3
  - SPI: H1- 3&4, 6&7
- **Load ‘FG4916 SPI MST software**
- **“Run” and close debugger**
- **Complete & load ‘F2013 SPI SLV software**
- **LED3 pulses when ‘F2013 is actively communicating**
- **Increase ‘F2013 temperature**
- **LED4 should turn on**

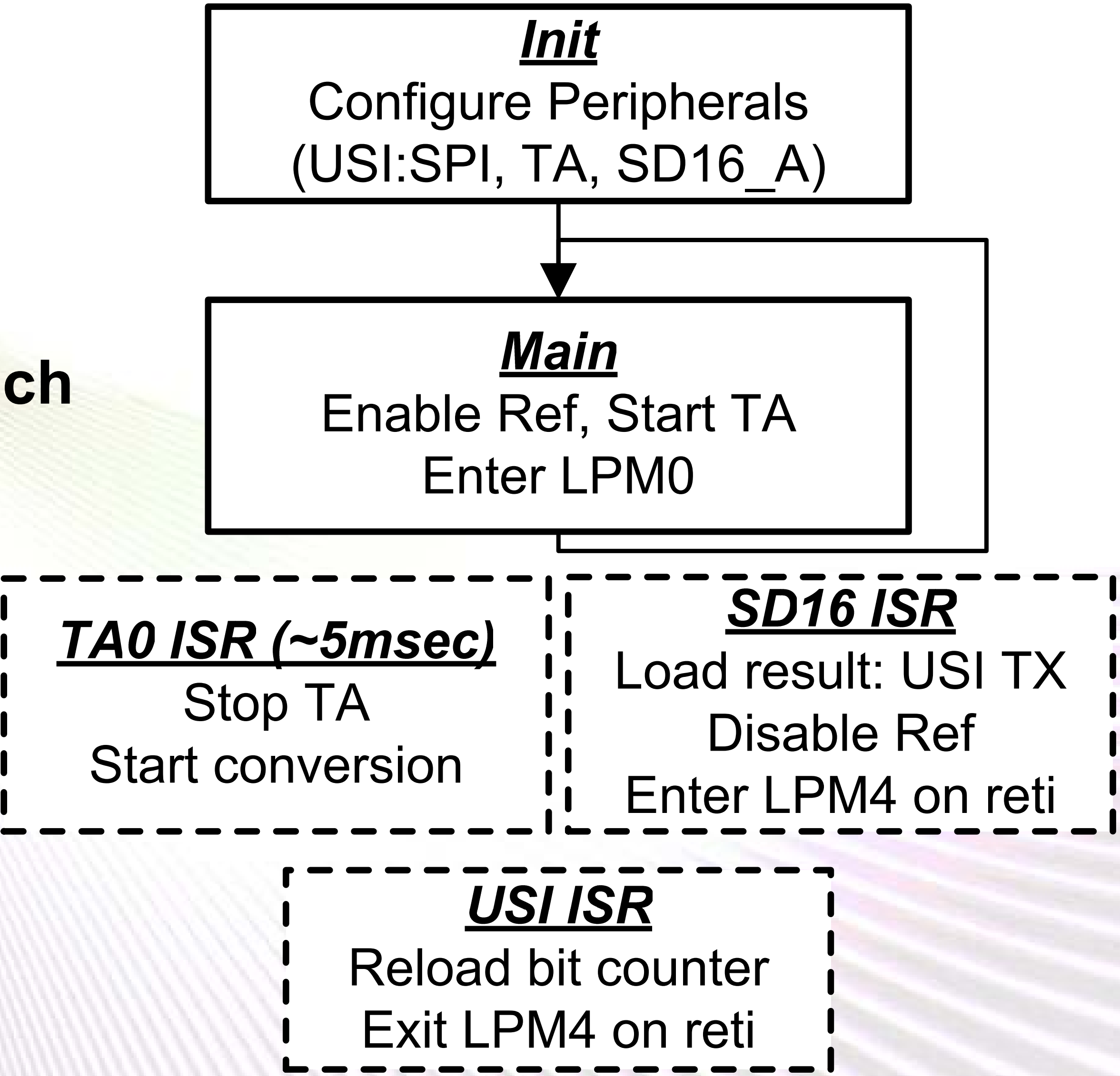




Lab 1 Software Flow

- **Timer_A used for VREF settling timer**
- **USI-triggered wake-up from LPM4**
- **New conversion after each data transfer**

- **USI Configuration:**
  - SPI Slave (SCLK & SDO)
  - Enable data output
  - Enable USI interrupt
  - Load USI bit counter
  - 16-bit data transfer



Init
Configure Peripherals
(USI:SPI, TA, SD16_A)

Main
Enable Ref, Start TA
Enter LPM0

TA0 ISR (~5msec)
Stop TA
Start conversion

SD16 ISR
Load result: USI TX
Disable Ref
Enter LPM4 on reti

USI ISR
Reload bit counter
Exit LPM4 on reti

Lab 1 Exercise: USI Setup

```
// Init USI peripheral

USICTL0 |= __________________;
USICTL1 |= ______;
USICTL0 &= ~_______;
USICNT = __________;
```

- Setup USI for slave mode
- Configure SDO & SCLK pins for USI function
- Enable SDO output buffer
- Enable bit counter interrupt flag
- Release USI module for operation (USISWRST)
- Load bit counter for first data transfer (16-bit data)

Lab 1 Exercise: USI ISR

```
// USI interrupt service routine
#pragma vector=USI_VECTOR
__interrupt void universal_serial_interface(void)
{
  USICNT = ___________;
  ____________________;
}
```

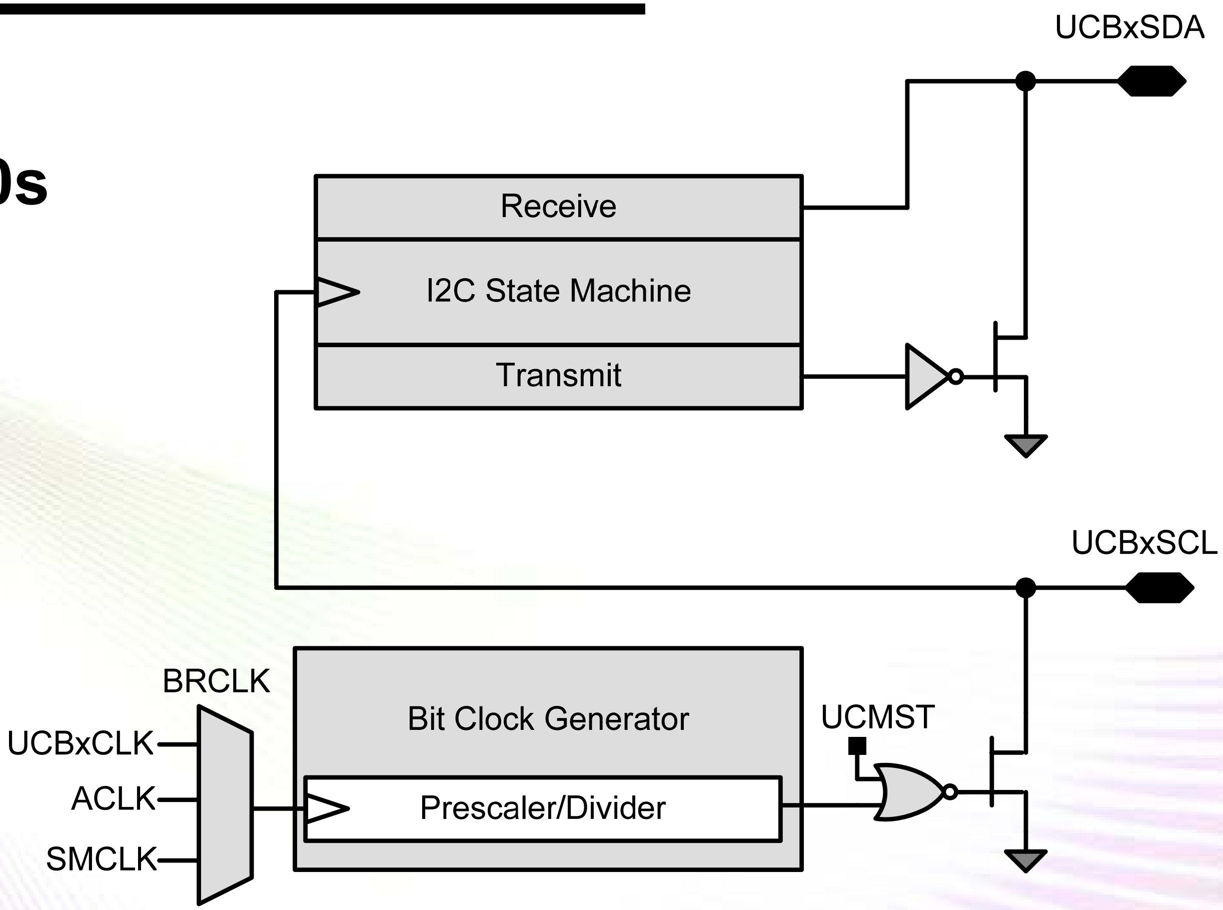
- **Update bit counter**
- **Exit ISR active (exit LPM4 on reti)**
  - *Refer to pg 197 in the MSP430 C/C++ Compiler Reference Guide for additional information on LPMx entry/exit and ISRs (See IAR Help Menu)*
  - *Usage also shown in the “device”.h project include files*

Agenda

- USC & USCI Basics
- Hands-On Lab: SPI
- Hands-On Lab: I2C
- Wrap-up & Summary

Lab Exercise 2: USCI B I2C

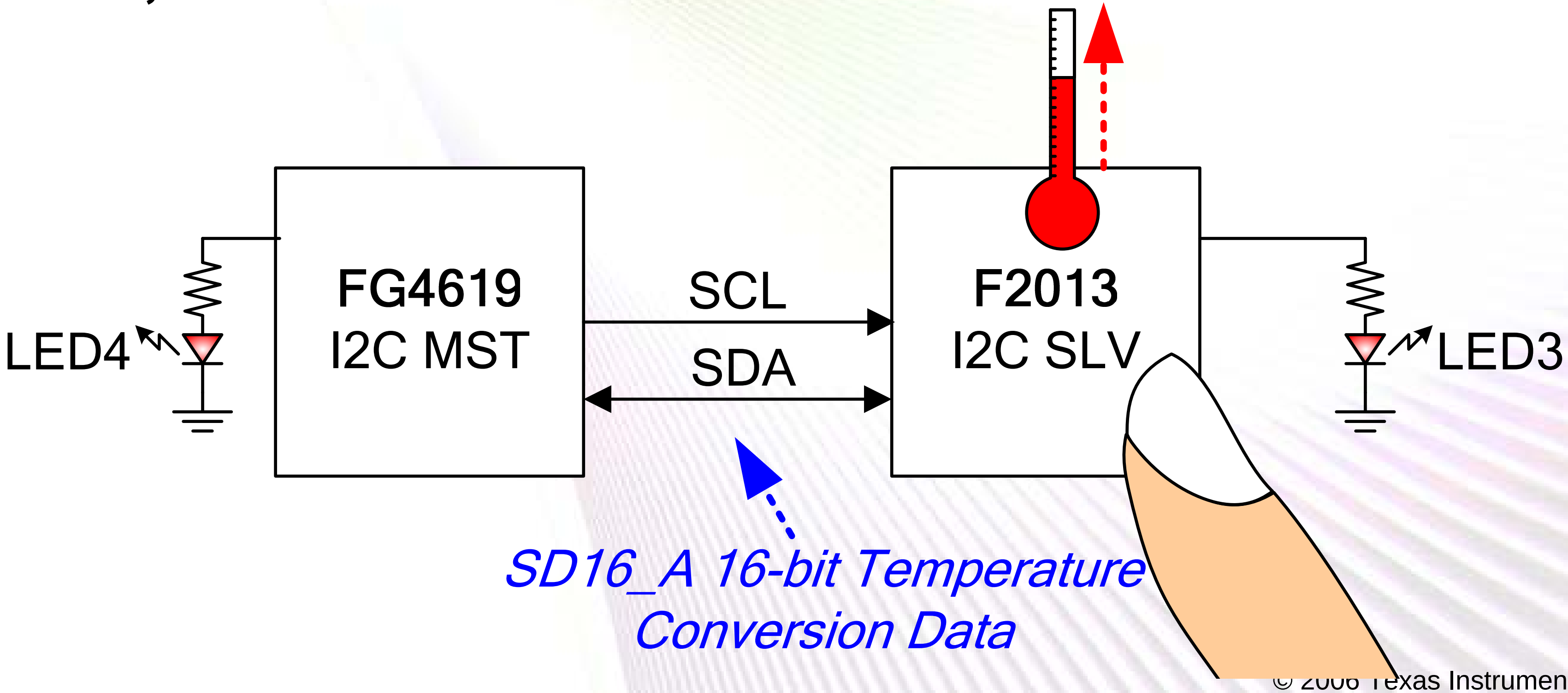
- **Complete an I2C data link between 2 MSP430s**
- **Complete partial ‘FG4619 USCI_B I2C master code**
  - Master Receiver
- **Slave address = 0x48**
- **~100KBPS data rate**
- **Load ready-to-use USI I2C slave code on the ‘F2013**



© 2006 Texas Instruments Inc, Slide 15

Lab Exercise 2: Overview

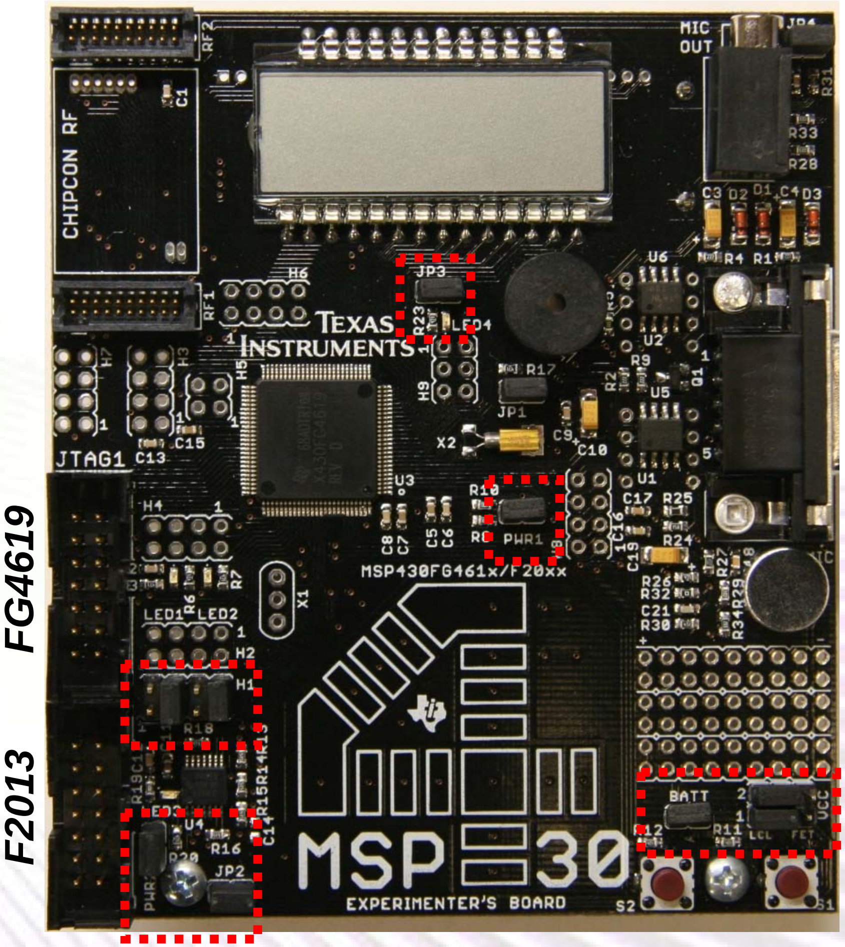
- Implement I2C Master on the ‘FG4619 to RX the 16-bit conversion result from the ‘F2013 Slave
- Slave flashes LED3 with each communication cycle
- When the result-to-result difference exceeds a preset delta, the Master turns on LED4



SD16_A 16-bit Temperature Conversion Data

Lab Exercise 2: Board Setup

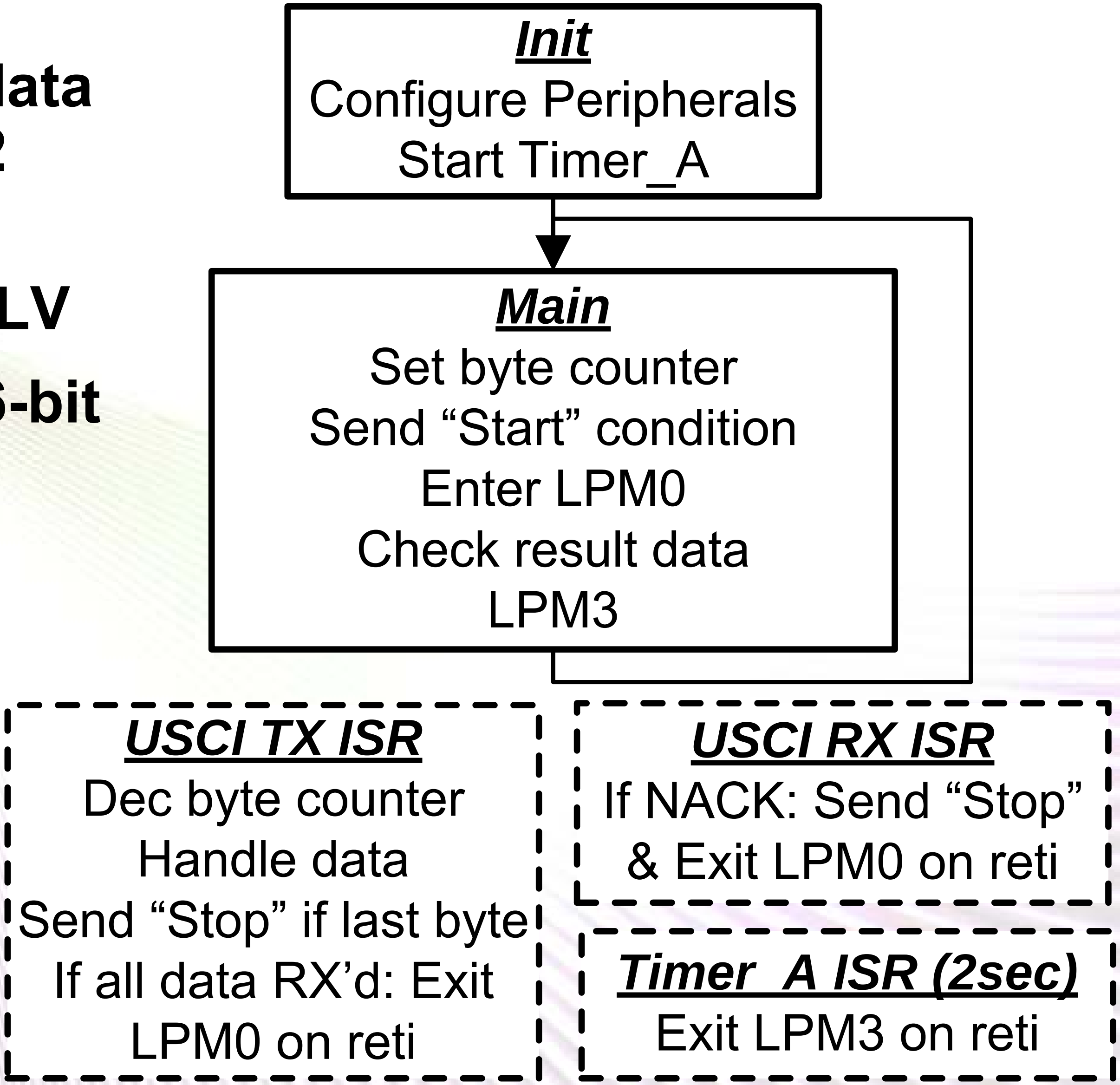
- **Needed Jumper Settings:**
  - PWR1/2, BATT, LCL1/2, JP2, JP3
  - SPI: H1- 1&2, 3&4
- **Load ‘F2013 I2C SLV software**
- **“Run” and close debugger**
- **Complete & load ‘FG4619 I2C MST software**
- **LED3 pulses when ‘F2013 is actively communicating**
- **Increase ‘F2013 temperature**
- **LED4 should turn on**



Lab 2 Software Flow

- **Timer_A triggers new data communication every 2 secs**
- **Two bytes RX'd from SLV**
- **Assemble bytes into 16-bit result**
- **SLV NACK handling**

- **USCI_B Configuration:**
  - PxSEL for SDA & SCL
  - I2C Master Receiver
  - ~100 KBPS data rate
  - Slave address = 0x48
  - Enable NACK & RX IFGs



Init
Configure Peripherals
Start Timer_A

Main
Set byte counter
Send "Start" condition
Enter LPM0
Check result data
LPM3

USCI TX ISR
Dec byte counter
Handle data
Send "Stop" if last byte
If all data RX'd: Exit LPM0 on reti

USCI RX ISR
If NACK: Send "Stop"
& Exit LPM0 on reti

Timer_A ISR (2sec)
Exit LPM3 on reti

Lab 2 Exercise: USCI Setup

```
// USCI Initialization
P3SEL |= ____;
UCB0CTL1 |= ______;
UCB0CTL0 = ________________;
UCB0CTL1 = ________________;
UCB0BR0 = __;
UCB0BR1 = __;
UCB0I2CSA = ____;
UCB0CTL1 &= ~_____;
UCB0I2CIE |= _____;
IE2 |= ______;
```

- Init port pins and assert SW reset
- Setup for MST I2C
- Set clock source & freq
- Define SLV address, clear SW reset & enable RX interrupt

Lab 2 Exercise: USCI Data ISR

```
// USCI Data Handling ISR
#pragma vector = USCIAB0TX_VECTOR
interrupt void USCIAB0TX_ISR(void)
{ RxByteCtr--;
  if (RxByteCtr)
  { RxWord = (unsigned int)________ << 8;
    if (RxByteCtr == 1)
      UCB0CTL1 |= ________;}
  else
  { RxWord |= _________;

  _______________________________;}}
```

- Get data from USCI receive buffer (two bytes)
- Generate “Stop” condition on last byte RX
- Exit LPM0 after all bytes received

Lab 2 Exercise: USCI State ISR

```
// USCI State Handling ISR
#pragma vector = USCIAB0RX_VECTOR
__interrupt void USCIAB0RX_ISR(void)
{ if (________ & ________)
  { UCB0CTL1 |= _________;
    UCB0STAT &= ~________;
    _____________________;
    return;}}
```

- Check if NACK was reason for ISR entry
- Generate “Stop” condition
- Clear required flag
- Exit ISR in active mode (exit LPM0)

Agenda

- USC & USCI Basics
- Hands-On Lab: SPI
- Hands-On Lab: I2C
- Wrap-up & Summary

Summary

USI Peripheral

- Provides SPI & I2C support in hardware
  - Protocol in user S/W, Critical timing in USI H/W
- Provides flexibility of 100% S/W solution while maintaining timing in hardware
- Reduced code size & lower CPU active time

USCI Peripheral

- Two independent blocks can operate simultaneously
- All modes capable of operating from any LPMx
- USCI is interrupt driven & DMA enabled
- USCI_A supports SPI, UART, IrDA, auto baud LIN Bus
- USCI_B supports SPI, I2C

Lab 1 Solution: USI Setup

```
// Init USI peripheral

USICTL0 |= USIPE6+USIPE5+USIOE; // Port init, SPI slave
USICTL1 |= USIIE; // Counter interrupt, flag remains set
USICTL0 &= ~USISWRST; // USI released for operation
USICNT = USI16B + 16; // Load bit counter, clears IFG
```

- Port setup is in the module (PxDIR, etc not needed)
- Only 1 IFG used for SPI mode
- Loading the bit counter clears the USIIFG

Lab 1 Solution: USI ISR

```
// USI interrupt service routine
#pragma vector=USI_VECTOR
__interrupt void universal_serial_interface(void)
{
  USICNT = USI16B + 16; // Load bit counter for next TX
  _BIC_SR_IRQ(LPM4_bits); // Exit LPM4 on RETI
}
```

- **Prep for 16 data bits on next TX**
- **Clear LPM4 bits in SR on stack**
  - CPU & DCO active on exit

Lab 2 Solution: USCI Setup

```
// USCI Initialization
P3SEL |= 0x06; // Assign I2C pins to USCI_B0
UCB0CTL1 |= UCSWRST; // Enable SW reset
UCB0CTL0 = UCMST+UCMODE_3+UCSYNC; // I2C MST, sync mode
UCB0CTL1 = UCSSEL_2+UCSWRST; // SMCLK, keep SW reset
UCB0BR0 = 11; // fSCL = SMCLK/11 = 95.3kHz
UCB0BR1 = 0;
UCB0I2CSA = 0x48; // Set slave address
UCB0CTL1 &= ~UCSWRST; // Clear SW reset
UCB0I2CIE |= UCNACKIE; // Interrupt on slave Nack
IE2 |= UCB0RXIE; // Enable RX interrupt
```

- Setup USCI_B: pins, mode, SCL frequency
- Set slave address properly (see slave code)
- Enable needed INTs: Data RX, Slave NACK

Lab 2 Solution: USCI Data ISR

```
// USCI Data Handling ISR
#pragma vector = USCIAB0TX_VECTOR
__interrupt void USCIAB0TX_ISR(void)
{ RxByteCtr--; // Decrement RX byte counter
  if (RxByteCtr)
  { RxWord = (unsigned int)UCB0RXBUF << 8; // Get data
    if (RxByteCtr == 1) // Only one byte left?
      UCB0CTL1 |= UCTXSTP;} // Generate stop condition
  else
  { RxWord |= UCB0RXBUF; // Get final received byte,
                         // Combine MSB and LSB
    _BIC_SR_IRQ(LPM0_bits);}} // Exit LPM0
```

- Handle data: 2 bytes → 1 word
- Send a “Stop” with the last byte RX’d
- Exit ISR active

Lab 2 Solution: USCI State ISR

```
// USCI State Handling ISR
#pragma vector = USCIAB0RX_VECTOR
__interrupt void USCIAB0RX_ISR(void)
{ if (UCNACKIFG & UCB0STAT)
  { UCB0CTL1 |= UCTXSTP; // Generate I2C stop condition
    UCB0STAT &= ~UCNACKIFG; // Clear IFG
    _BIC_SR_IRQ(LPM0_bits); // Exit LPM0
    return;}}
```

- **USCIAB0RX used for I2C State handling**
  - All I2C flags corresponding to status are here (no data IFGs)
- **On NACK for slave, send “Stop”**
- **Exit ISR active for communication retry**

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265