

MSP430 Advanced Technical Conference 2006



Hands-on: The New MSP430 Communication Peripherals

Zack Albus
MSP430 Applications Engineer
Texas Instruments

Technology for Innovators™



Agenda

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

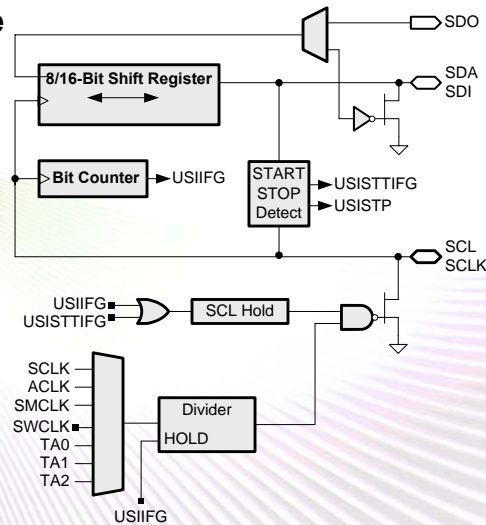
© 2006 Texas Instruments Inc, Slide 2

Technology for Innovators™



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**



© 2006 Texas Instruments Inc, Slide 3

Technology for Innovators™

TEXAS INSTRUMENTS

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**

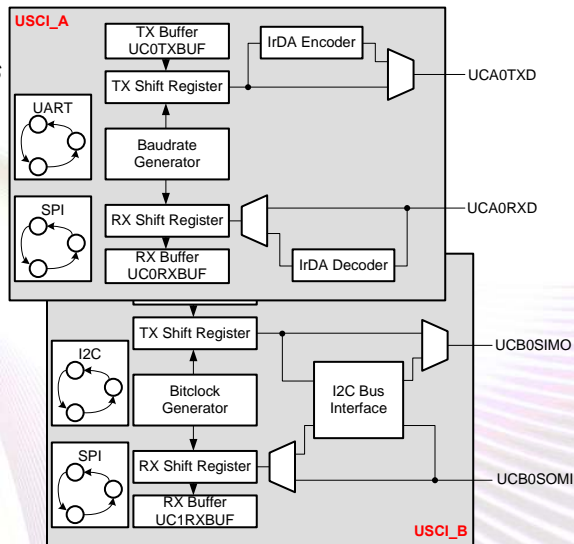
© 2006 Texas Instruments Inc, Slide 4

Technology for Innovators™

TEXAS INSTRUMENTS

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**



Technology for Innovators™

TEXAS INSTRUMENTS

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**

© 2006 Texas Instruments Inc, Slide 6

Technology for Innovators™

TEXAS INSTRUMENTS

Agenda

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

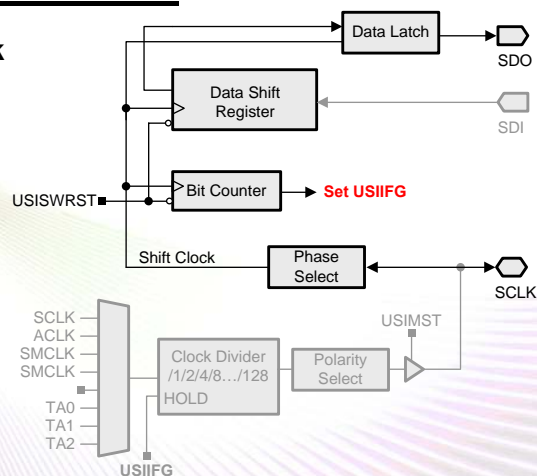
© 2006 Texas Instruments Inc, Slide 7

Technology for Innovators™

TEXAS INSTRUMENTS

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



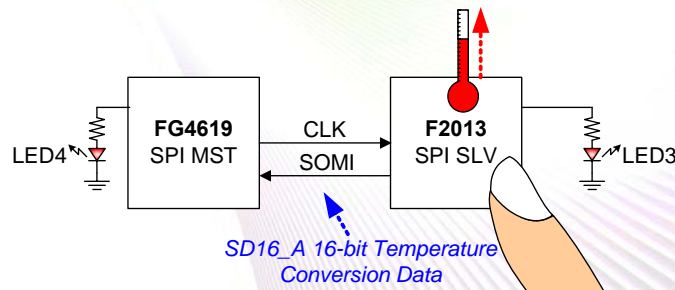
© 2006 Texas Instruments Inc, Slide 8

Technology for Innovators™

TEXAS INSTRUMENTS

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



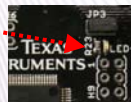
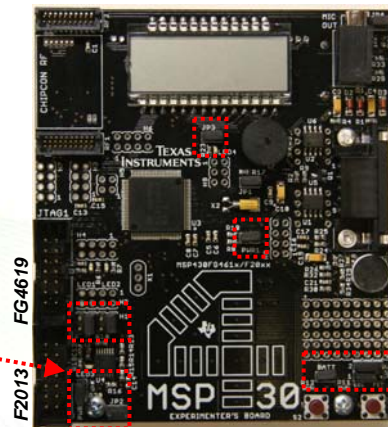
© 2006 Texas Instruments Inc, Slide 9

Technology for Innovators™

TEXAS INSTRUMENTS

Lab Exercise 1: Board Setup

- Needed Jumper Settings:
 - PWR1/2, BATT, LCL1/2, JP2, JP3
 - SPI: H1- 3&4, 6&7
- Load 'FG4619 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



© 2006 Texas Instruments Inc, Slide 10

Technology for Innovators™

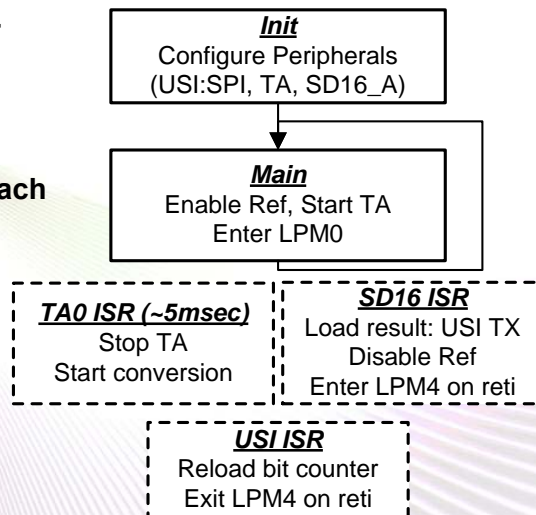
TEXAS INSTRUMENTS

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



© 2006 Texas Instruments Inc, Slide 11

Technology for Innovators™

TEXAS INSTRUMENTS

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)

© 2006 Texas Instruments Inc, Slide 12

Technology for Innovators™

TEXAS INSTRUMENTS

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

© 2006 Texas Instruments Inc, Slide 13

Technology for Innovators™



Agenda

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

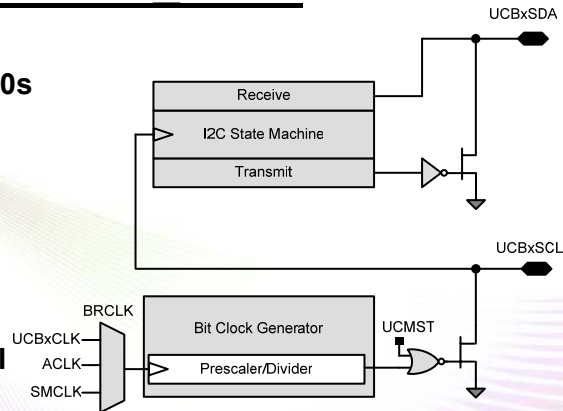
© 2006 Texas Instruments Inc, Slide 14

Technology for Innovators™



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



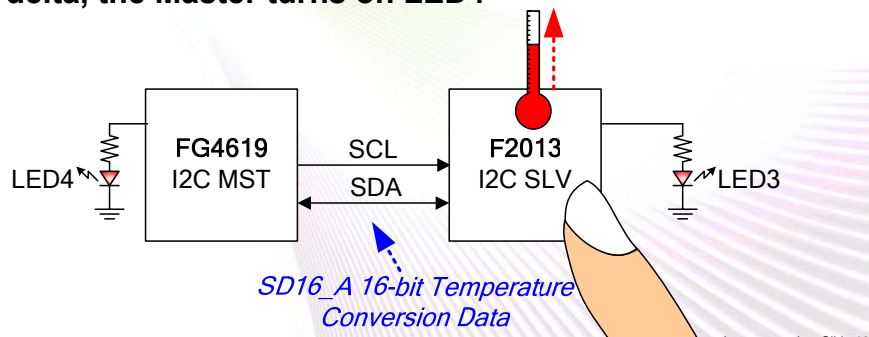
© 2006 Texas Instruments Inc, Slide 15

Technology for Innovators™

TEXAS INSTRUMENTS

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



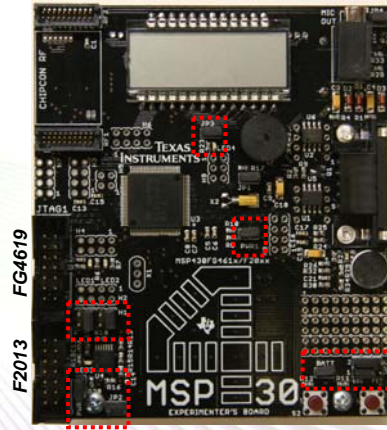
© 2006 Texas Instruments Inc, Slide 16

Technology for Innovators™

TEXAS INSTRUMENTS

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



© 2006 Texas Instruments Inc, Slide 17

Technology for Innovators™

TEXAS INSTRUMENTS

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

© 2006 Texas Instruments Inc, Slide 18

Technology for Innovators™

TEXAS INSTRUMENTS

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

© 2006 Texas Instruments Inc, Slide 19

Technology for Innovators™



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

© 2006 Texas Instruments Inc, Slide 20

Technology for Innovators™



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)

© 2006 Texas Instruments Inc, Slide 21

Technology for Innovators™

 TEXAS INSTRUMENTS

Agenda

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

© 2006 Texas Instruments Inc, Slide 22

Technology for Innovators™

 TEXAS INSTRUMENTS

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

© 2006 Texas Instruments Inc, Slide 23

Technology for Innovators™



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

© 2006 Texas Instruments Inc, Slide 24

Technology for Innovators™



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

© 2006 Texas Instruments Inc, Slide 25

Technology for Innovators™



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

© 2006 Texas Instruments Inc, Slide 26

Technology for Innovators™



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

© 2006 Texas Instruments Inc, Slide 27

Technology for Innovators™

TEXAS INSTRUMENTS

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

© 2006 Texas Instruments Inc, Slide 28

Technology for Innovators™

TEXAS INSTRUMENTS