User's Guide
TIDCBV2–March 2016

Capacitor Bank Switching and HMI Subsystem Reference Design for Automatic Power Factor Controller

Firmware Description

The TIDA-00737, which encompasses the TIDA-00454, contains an analog front-end for measuring three voltage channels and three current channels. It uses the MSP430F67791A, which is a low-power MSP430™ MCU from TI with a 32-bit multiplier to perform all energy calculations. The MSP430F67xx1A family executes the energy measurement software library from TI, which calculates voltage RMS, current RMS, active power, reactive power, and power factor [3]. The energy measurement software library is available along with the MSP430F67xx1AEVM [4]. For details on interfacing the TIDA-00454 board with the TIDA-00737 board, refer to the TIDA-00737 design guide (TIDUB67) [1].

The TI MSP430™ family of MCUs supports the standard JTAG interface, which requires four signals for sending and receiving data. The JTAG signals are shared with the GPIO. The TEST/SBWTCK signal is used to enable the JTAG signals. In addition to these signals, the RESET signal is required to interface with the MSP430 development tools and device programmers. For further details on interfacing to development tools and device programmers, see the *MSP430 Hardware Tools User's Guide* (SLAU278) [2].

The TIDA-00737 focuses on using the computed values from the energy library for capacitor bank switching. It also focuses on the HMI subsystem that includes a 4-digit 7-segment LED display (SSD), LEDs, relays (for contactors), and transistor (for thyristors) output drivers.

This document will cover firmware aspects of the HMI including pin assignment, peripheral initialization, functional behavior and communication (SPI or I^{2}C). Table 1 shows the TIDA-00737 at a glance.

Table 1. TIDA-00737 MCU Interface

MSP430F67791A PERIPHERAL	INTERFACED WITH	APPLICATION (USED FOR)
SPI0 (USCI A0)	TLC5916	4-digit, 7-segment display (SSD)
SPI1 (USCI A1)	TLC59025	LED indications
GPIO	TPL7407L	Output relays (for contactors) and Transistors (for thyristors)
I2C0 (UCSI B0)	OPT3001	Ambient light sensing for varying the intensity of SSD
I2C1 (UCSI B1)	TMP451 EVM	To detect remote temperature used for over temperature alarm
UART (UCSI A2)	—	To communicate with GUI and display voltage RMS, current RMS, Active power, reactive power, and power factor

Copyright © 2016, Texas Instruments Incorporated

1 4-Digit 7-Segment LED Display (SSD)

The SSD is driven using TLC5916 Constant-Current LED Sink Drivers. The TLC5916 converts serial input data into a parallel output. This design uses two TLC5916 devices that are cascaded. To update the SSD data, two bytes of data is used:

- SSD digit selection
- SSD data corresponding to value to be displayed

The first byte is used by the TLC5916 (U5) to select the digit (position) as shown in Table 2.

Table 2. Digit Selection

TO SELECT DIGIT	DATA
1 (MSB)	0x80
2	0x40
3	0x20
4 (LSB)	0x10

Other bits are not used in this design. The TLC5916 is used to select four digits as shown in Table 3:

Table 3. Position Format (Left to Right)

POSITION1	POSITION2	POSITION3	POSITION4
Digit1	Digit2	Digit3	Digit4

The second byte is used by the TLC5916 (U4) to display the parameter values. Each digit can display numbers from zero to nine and an optional decimal point. The segments of SSD are referred by the letters A to G and decimal point (referred to by letter H). By switching the segments on and off, various characters and numbers can be displayed. For example, to display zero, the segments A to F are turned ON and segment G is turned OFF (see the Lumex LDQ-M286RI datasheet). Table 4 shows the values for the corresponding value to be displayed.

NOTE: The SSD used is a common anode type and the driver used is a constant-current sink driver type.

Table 4. Data Displayed on SSD Mapped With LED Driver Data

VALUE TO BE DISPLAYED	SSD DATA SENT THROUGH SPI
0	0xFC
1	0x60
2	0xDA
3	0xF2
4	0x66
5	0xB6
6	0xBE
7	0xE0
8	0xFE
9	0xF6

Each SPI message updates one digit data. SPI messages are updated every 250 µs using the ADC sampling interrupt. The circular display sequence is refreshed every 1 ms (250 µs × 4 digits).

Copyright © 2016, Texas Instruments Incorporated

1.1 Initialization

1.1.1 GPIO Pins Initialization

Table 5. Function and Port Mapping

FUNCTION	MCU PORT PIN
LE_SEG	P1.2
OE_SEG\	P2.4

1. Set P1SEL [2] as 0 to select GPIO mode of operation.
2. Set P1DIR [2] to 1 to set the GPIOs as outputs.
3. Enable the internal resistors by making P1REN [2] as 1.
4. Default (during start up): P1.2 as low.
5. Set P2SEL bit4 as 0 to select GPIO mode of operation.
6. Set P2DIR bit4 to 1 to set the GPIOs as outputs.
7. Enable the internal resistors by making P2REN bit4 as 1.
8. Default (during start up): P2.4 as high.

1.1.2 SPI0 Initialization

1. Set P3SEL0 [0:2] bits as 1 to initialize the pins P3.0, P3.1, and P3.2 as the UCA0SOMI, UCA0SIMO, and UCA0CLK.
2. Reset the universal serial communication interface (USCI) module using register UCA0CTLW0 and clear all the other bits.
3. Set the SPI clock source as SMCLK (set UCA0CTLW0 bit 6 and bit 7).
4. Select the master mode using UCA0CTLW0 bit 11.
5. Select synchronous mode enable using UCA0CTLW0 bit 8.
6. Select the SPI mode as three-pin (UCA0CTLW0 bit 9 and bit 10 clear).
7. Set SPI speed to 9 Mbps using register UCA0BRW.
8. In the register UCA0CTLW0, clear bit 13 (LSB first option) and bit 15 (data captured on the first and changed on the next).
9. In the register UCA0CTLW0, clear bit 14 (the SPI clock polarity as inactive to be low).
10. Enable the SPI module by clearing UCA0CTLW0 bit 0.
11. Clear interrupts using the UCA0IFG register.
12. Enable the SPI receive interrupt using the register UCA0IE bit 0.
13. Enable global interrupt using status register (SR) bit 3 (GIE).

Copyright © 2016, Texas Instruments Incorporated

1.2 Scrolling Selected Parameters

Parameters like current, voltage, and power factor for R, Y, and B phases are displayed in a cyclic sequence. Each parameter is displayed for 4 seconds before changing to the next parameter. Corresponding LEDs glow to indicate the type of parameter (voltage, current, or power factor) and the phase (R, Y, or B) being displayed on SSD.

1.2.1 Selecting the SSD Digit to Update Data

NOTE: Position_Y (where Y = 1 to 4)

Data is updated one digit at a time. The function Position_1() is used to update Digit1 (First = 0x80;) in the SPI message sent to TLC5916. Likewise, Position_2() is used to update data in Digit2, and so on.

1.2.2 Updating Data

NOTE: For Digit_X (where X = 0 to 9)

Different segments (within SSD) have to be turned ON to display the numbers 0 to 9. See Table 4 for the values corresponding to these numbers.

The function Digit_0() is used to update the second byte of data in the SPI message sent to TLC5916 (Second = ZERO; //set G and clear others to display 0 //0xFC). To display digit1, call Digit_1() and so on.

NOTE: To display dot, BIT0 is set along with digit value.

1.2.3 Selecting Parameter to be Displayed

The function UpdateReadings() is called every four seconds to update the SSD. The following parameters are displayed (in a cyclic order):

- Voltage R (first four seconds)
- Voltage Y (next four seconds)
- Voltage B
- Current R
- Current Y
- Current B
- Power factor R
- Power factor Y
- Power factor B

1.2.4 Refreshing SSD Display Data UpdateReadingsRepeat()

The parameter selected by UpdateReadings() is refreshed on the display by repeatedly calling this function. This function is called 4096 times per second (when ADC sampling is initiated).

Copyright © 2016, Texas Instruments Incorporated

1.3 *Adjusting SSD Brightness*

When the ambient light sensed by the OPT3001 is above 100 lux, SSD current is increased and the SSD brightness is higher. When the ambient light intensity level decreases to less than 100 lux, the SSD current is reduced and the SSD brightness is lower. The TLC5916's configuration can be changed by switching to a special mode. SSD brightness depends on the configuration value. (Configuration value used in special mode: 0xFE). After updating the configuration, the driver must be configured to normal mode to update the data.

1.3.1 TLC5916 (Constant Current LED Sink Driver)

During normal system functioning, the TLC5916 is configured for normal mode. In the normal mode phase, the serial data can be transferred into TLC5916 through SDI, shifted in the shift register, and transferred out through the pin SDO. LE latches the serial data in the shift register to the output latch. OE\ enables the output drivers.

Table 6. TLC5916 Operation

CLK	LE(ED1)	$\overline{\text{OE}}$(ED2)	SDI	$\overline{\text{OUT0}}$ to $\overline{\text{OUT7}}$	SDO
↑	H	L	Dn	Dn to Dn – 7	Dn – 7
↑	L	L	Dn + 1	No change	Dn – 6
↑	H	L	Dn + 2	Dn + 2 to Dn – 5	Dn – 5
↓	X	L	Dn + 3	Dn + 2 to Dn – 5	Dn – 5
↓	X	H	Dn + 3	Off	Dn – 5

1.3.1.1 *Cascading of TLC5916 Drivers*

Two numbers of TLC5916 (U4 and U5) have been cascaded in this design. U4 is used to drive each segment of a digit (SEG-A to SEG-H). U5 is used for digit selection (DIG1 to DIG4).

When data is sent by the SPI master, the first byte of data is for U5 and the second byte for U4. (The last device cascaded receives the first byte.)

SPI data format: Position (first byte, see Table 2), data (second byte, see Table 5)

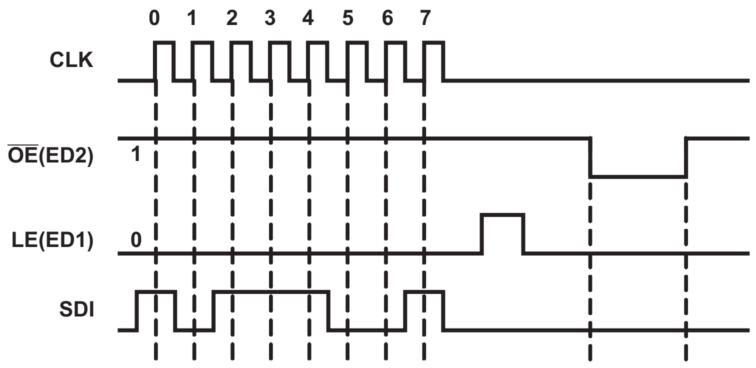

Figure 1. TLC5916 Normal Mode

After sending the data to TLC5916 through SPI, enable output by making P1.2 (LE_SEG: high and low) and P2.4 (OE_SEG\: low). These pins are shared by U4 and U5.

Copyright © 2016, Texas Instruments Incorporated

1.3.2 Switching to Special Mode

NOTE: ToSpecial(); In special mode, configuration can be updated. Using the configuration register, it is possible to vary the SSD brightness by varying the current.

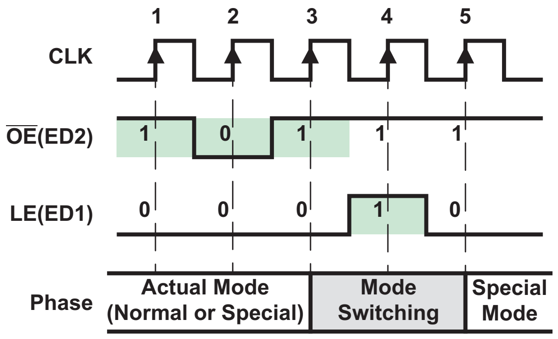

Figure 2. Switching to Special Mode

The function ToSpecial() calls SCLK2GPIO() and configures P3.2 as GPIO. P3.2 is used to toggle states (0 and 1 alternatively) and generates five clocks. OE_SEG\ (P2.4) is toggled 1-0-1 at the falling edge of the first 3 clock cycles (see Figure 2). LE_SEG (P1.2) is set to 0 at first clock cycle (or before). LE_SEG is set to 1 at the falling edge of the third clock cycle. ToSpecial() calls the function SendSplModeConfig() to change configuration register values in the TLC5916.

NOTE: To achieve clock phase alignment, SPI peripheral pin is configured as GPIO for switching into special mode. After switching to GPIO mode, bit banging is used to simulate clock and data, while toggling OE_SEG\ and LE_SEG as shown in Figure 2.

1.3.3 Writing New Configuration Data in Special Mode

ToSpecial() calls SendSplModeConfig(). The TLC5916 must be in special mode to call this function. This function toggles bits to send out the configuration data to the TLC5916. P3.2 is used as clock, and P3.1 is used as data. The first byte is sent as configuration data (0xFF). This byte is shifted to U5. The second byte (0xFE) is sent as configuration data (sent to U4) is of interest. The CM bit is made as 0 to make the data 0xFE. LE goes high when bit0 is transmitted as shown in Figure 4.

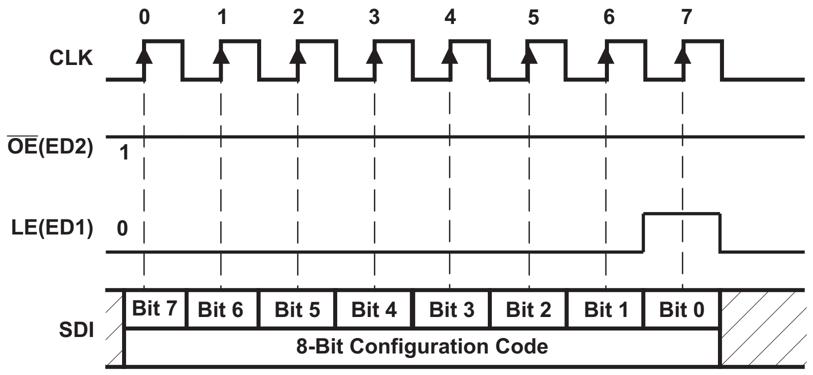

Figure 3. Writing Configuration Code in Special Mode

Copyright © 2016, Texas Instruments Incorporated

1.3.4 Switching Back to Normal Mode

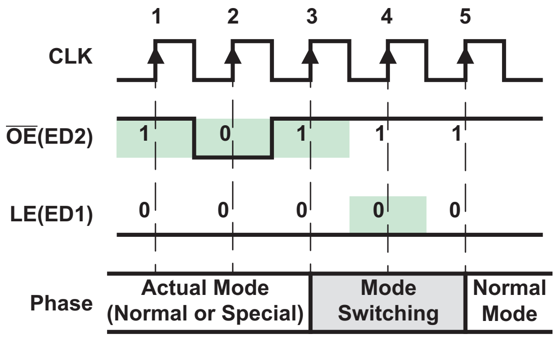

Figure 4. Switching to Normal Mode

ToNormal() calls SCLK2GPIO() and configures P3.2 as GPIO. P3.2 is used to toggle states (0 and 1 alternatively) and generates five clocks. OE_SEG\ (P2.4) is toggled 1-0-1 during the first three clock cycles (see Figure 4). LE_SEG (P1.2) is set to 0 at first clock cycle (or before). ToNormal() calls the function GPIO2SCLK() to reinitialize SPI0 peripheral. The SPI clock and data pins are reverted back to their SPI peripheral functionality.

1.3.5 Switching From Normal to Special Mode

SCLK2GPIO() configures UCA0CLK (P3.2) into GPIO:

1. Set P3SEL0 bit 2 as 0.
2. Set P3DIR bit 2 as 1.
3. Set P3REN bit 2 as 1.

This function is used to configure TLC5916 modes (normal to special, special to normal).

1.3.6 Switching From Special to Normal Mode

GPIO2SCLK() reconfigures the SPI0 peripheral.

1.3.7 Enabling Output After Sending Data Using SPI

To use the function Enable7SegLEDOutput():

1. Set P1OUT bit 2 as 1 to make LE_SEG high.
2. Set P1OUT bit 2 as 0 to make LE_SEG low.
3. Set P2OUT bit 4 as 0 to make OE_SEG low.

1.3.8 Calling SetTxData()

This function is called after the digit and display data value are set. This function has two input parameters, data1 and data2. The function prepares the SPI data and assigns a pointer at the beginning.

The SPI message follows this sequence for a 7-segment display:

1. Position: data1
2. Value (to display numbers): data2

1.3.9 Transmitting SPI Data From SPI Master to TLC5916

The input parameters for TransmitData_SPI() are the base address of the SPI peripheral and pointer to data. The user needs to call SetTxData() before calling TransmitData_SPI(). This function waits for data transmission to get complete. Once data transmission is complete, the function copies data (using the data pointer) to UCA0TXBUF. This process is repeated until all data is copied to UCA0TXBUF.

Copyright © 2016, Texas Instruments Incorporated

2 Indication LEDs

16 LEDs are used in this TI Design for indicating alarms, capacitor bank status, and for the parameter selected.

- 8 LEDs indicate phase, parameter selected, and PF (lead/lag) information
- 5 LEDs indicate the capacitor bank status
- 3 LEDs indicate the alarm condition

2.1 Indicating Display Parameter Status

LED status is updated in tandem with SSD. Every four seconds, the LED status is updated according to the measured data. The parameters like current, voltage, and power factor for R, Y, and B phase are displayed in a cyclic sequence. Each parameter is displayed for four seconds. When displaying the parameter, a corresponding LED glows to indicate which parameter is displayed.

Table 7. Mapping LED to Parameter Being Displayed on 7-Segment LED

D14	D15	D11	D12	D13	D16	D10	D7
VOLTAGE	CURRENT	PF	PHASE R	PHASE Y	PHASE B	LAG	LEAD

Copyright © 2016, Texas Instruments Incorporated

2.2 Alarm Status

LEDs indicate the measured reactive power. LED status is updated in tandem with output relays. When more than one output relay and LED (within D1, D4, D5, D8, D9) are to be operated in a sequence, every four seconds, one LED and one relay operate until the present status reaches the desired status.

2.2.1 GPIO Pins Initialization

Table 8. Function and Pin Mapping

FUNCTION	MCU PORT PIN
OE\	P1.0
LE	P1.1

1. Set P1SEL [0:1] as 0 to select GPIO mode of operation.
2. Set P1DIR [0:1] to 1 to set the GPIOs as outputs.
3. Enable the internal resistors by making P1REN [0:1] as 1.
4. Default (during start up): P1.0 high and P1.1 low.

2.2.2 SPI1 Initialization

1. Set P3SEL0 [4:6] bits as 1 to initialize the pins P3.3, P3.4, and P3.5 as the UCA1CLK UCA1SOMI and UCA1SIMO.
2. Reset the universal serial communication interface (USCI) module using register UCA1CTLW0. Set bit 0 as 1 and clear all the other bits.
3. Set the SPI clock source as SMCLK (set UCA1CTLW0 bit 6 and bit 7).
4. Select the master mode using UCA1CTLW0 bit 11.
5. Select synchronous mode enable using UCA1CTLW0 bit 8.
6. Select the SPI mode as three-pin (UCA1CTLW0 bit 9 and bit 10 clear).
7. Set SPI speed to 5 Mbps using register UCA1BRW.
8. In the register UCA1CTLW0, clear bit 13 (LSB first option) and bit 15 (data captured on the first and changed on the next).
9. In the register UCA1CTLW0, clear bit 14 (the SPI clock polarity as inactive to be low).
10. Enable the SPI module by clearing UCA1CTLW0 bit 0.
11. Clear interrupts using the UCA1IFG register by making the value to 0.
12. Enable the SPI receive interrupt using the register UCA1IE bit 0.
13. Enable global interrupt using status register (SR) bit 3 (GIE).

2.2.3 TLC59025 (16-Channel Constant Current LED Sink Driver)

The serial data is transferred into TLC59025 through SDI, shifted in the shift register, and transferred out through SDO. LE can latch the serial data in the shift register to the output latch. OE enables the output drivers to sink current [see Table 1 in the TLC59025 datasheet (SLVS934)].

To display LEDs, send the desired data to TLC59025 using SPI and enable the output by making P1.1 high and P1.0 low.

Copyright © 2016, Texas Instruments Incorporated

2.2.4 SetTxData()

The input parameters for SetTxData() are data1 and data 2. SetTxData() prepares the SPI data and assigns pointer at the beginning of data. The SPI message follows this pattern for LEDs:

1. LEDs (D1 to D8): data1, the first byte of data on SPI. D1 glows when bit7 is set and D8 glows when bit0 is set
2. LEDs (D9 to D16): data2, the second byte of data on SPI

2.2.5 Transmitting SPI Data From SPI Master to TLC59025

The input parameters for the function TransmitData_SPI1() are the base address of the SPI peripheral and pointer to data. The user needs to call SetTxData() before calling TransmitData_SPI().

This function waits for data transmission to get complete. Once data transmission is complete, the function copies data (using the data pointer) to UCA1TXBUF. This process is repeated until all data is copied to UCA1TXBUF.

2.2.6 Calling EnableLEDOutput() After Transmitting Data to TLC59025 Using SPI

1. Set P1OUT bit 1 as 1 to make LE high.
2. Set P1OUT bit 1 as 0 to make LE low.
3. Set P1OUT bit 0 as 0 to make OE low.

2.2.7 UpdateLEDs()

NOTE: In the current implementation, R phase power values are used for relay switching.

This function is called every 4 seconds to update the LEDs to the desired state. The LEDs are illuminated based on the scenarios described in Table 9:

Table 9. LED Functions

LED	DESCRIPTION
D9	ON if reactive power (VAR) > 200
D8	ON if reactive power (VAR) > 400
D5	ON if reactive power (VAR) > 600
D4	ON if reactive power (VAR) > 800
D1	ON if reactive power (VAR) > 1000
D12	R phase
D13	Y phase
D16	B phase
D2	Indicate > 110% Un (Un = 230 V) overvoltage
D3	Indicate remote temperature > 50°C (over temperature)
D6	Indicate ambient light condition (< 100 lux)
D7	PF lead
D10	PF lag
D11	PF
D14	Voltage
D15	Current

The LEDs and outputs (relays or transistors) are switched in tandem. The outputs are switched on and off sequentially (one every 4 seconds).

NOTE: The LEDs D1, D4, D5, D8, and D9 are updated when voltage is > 23 V.

Copyright © 2016, Texas Instruments Incorporated

3 Relay and Transistor Outputs

Depending on the computed reactive power, output relays (for contactors) and transistors (for thyristors) can be turned ON (which in turn switches ON capacitor banks). The LEDs D1, D4, D5, D8, and D9 are operated along with their respective relay pair OUT_2, OUT_1, K1, K2, and K3 output relays. The outputs operate in a sequence as described in Table 10.

Table 10. Output Relays Operation is Based on Measured Reactive Power

REACTIVE POWER (VAR)	RELAY CONTROL OUTPUT STATUS (ON)
> 200	K3
> 400	+K2 (four seconds after K3)
> 600	+K1 (four seconds after K2)
> 800	+OUT_1 (four seconds after K1)
> 1000	+OUT_2 (four seconds after Out_1)

When the reactive power reduces, the relay outputs operate (switched off) in a sequence described in Table 11.

Table 11. Output Relays Operation During PF and Reactive Power Recovery Sequence

REACTIVE POWER (VAR)	RELAY CONTROL OUTPUT STATUS (OFF)
< 1000	OUT_2
< 800	–OUT_1 (four seconds after OUT_2)
< 600	–K1 (four seconds after OUT_1)
< 400	–K2 (four seconds after K1)
< 200	–K3 (four seconds after K2)

3.1 GPIO Pins Initialization

1. Set P2SEL bit7 as 0 to select GPIO mode of operation.
2. Set P2DIR bit7 (RESET_DRV) as 1 to set the GPIOs as outputs.
3. Enable the internal resistors by making P2REN bit7 as 1.
4. Set P2OUT bit7 (RESET_DRV) as zero. (default).
5. Set P7SEL bit1 as 0 to select GPIO mode of operation.
6. Set P7DIR bit1 (nENBL) as 1 to set the GPIOs as outputs.
7. Enable the internal resistors by making P7REN bit0 as 1.
8. Set P7OUT bit1 (nENBL) as zero.
9. Set P7SEL bit3 and bit5 as 0 to select GPIO mode of operation.
10. Set P7DIR bit3 and bit5 to 1 to set the GPIOs as outputs.
11. Enable the internal resistors by making P7REN bit3 and bit5 as 1.
12. Set P7OUT bit3 and bit5 as zero (default).
13. Set P8SEL [0:2] as 0 to select GPIO mode of operation.
14. Set P8DIR [0:2] to 1 to set the GPIOs as outputs.
15. Enable the internal resistors by making P8REN [0:2] as 1.
16. Set P8OUT [0:2] as zero (default).

Copyright © 2016, Texas Instruments Incorporated

3.2 *ProcessRelays()*

This function is called every four seconds to update the status of digital outputs.

- 5 outputs [(1)] divided into 5 steps from ≥ 200 VAR to ≥ 1000 VAR.
- OUT_2 (Transistor) = in sync with D1
- OUT_1 (Transistor) = in sync with D4
- K1 (Relay) = in sync with D5
- K2 (Relay) = in sync with D8
- K3 (Relay) = in sync with D9

(1) 3 relays + 2 transistors = 5 total outputs

Copyright © 2016, Texas Instruments Incorporated

4 Ambient Light Sensor

The OPT3001 ambient light sensor is used to measure the ambient lighting condition in which PFC controller is installed. The OPT3001 is connected to the MCU using an I^2C interface. The ambient light intensity is read every four seconds.

Depending on the ambient light condition, the display brightness of SSD is adjusted.

4.1 Write Process

Writing to a register starts with the first byte transmitted by the master. This byte is the slave address with the R/W bit low. The OPT3001 then acknowledges the receipt of a valid address.

The next byte transmitted by the master is the address of the register that data are to be written to. The write command follows this format:
[Register address + W] [data (MSB)] [data(LSB)].

4.2 Read Process

When reading from the OPT3001, the last value stored in the register address by a write operation determines which register is read during a read operation. To change the register address for a read operation, a new partial I^2C write transaction must be initiated. This partial write is accomplished by issuing slaves address byte with the R/W bit low, followed by the register address byte and a stop command. The master then generates a start condition and sends the slave address byte with the R/W bit high to initiate the read command:
[Register address + R/W].

4.3 Order of Commands and Function Calls

- Command1Write(): This function is called once to update the configuration register (I^2C initialization for UCB0 is a prerequisite.)
- Command1Poll(): This function is used to read back the value of the configuration register.
- Command2Write(): This function performs a partial write to update the register address.
- Command2Poll(): This function is called every 4 seconds. This function retrieves the result (2 bytes) from OPT3001.

4.4 I^2C Initialization

1. Set P4SEL [5:6] to 1 to enable UCB0_SCL and UCB0_SDA.
2. Reset the universal serial communication interface (USCI) module using register UCB0CTLW0. Set bit 0 as 1 and clear all the other bits.
3. Set the SPI clock source as SMCLK (set UCB0CTLW0 bit 6 and bit 7).
4. Select the master mode using UCB0CTLW0 bit 11.
5. Set the communication speed at 100 kbps by making the UCB0BRW register as 0xF0. (uses SMCLK)
6. Calls the function SetSlaveAddress() (Slave address for ALS OPT3001: 0x44)
7. Clear the reset bit UCB0CTLW0 bit 0 to start the I^2C module.
8. Enable transmit and receive interrupts by making UCB0IE [0:1] bits as 1.
9. Enable global interrupt using status register (SR) bit 3 (GIE).

4.5 SetSlaveAddress()

1. Set up the slave address for the master to communicate.
2. Set the I^2C slave address using register UCB0I2CSA (the slave address for the OPT3001 is 0x44).

Copyright © 2016, Texas Instruments Incorporated

4.6 I²C Transmit Interrupt

When UCB0IFG bit1 is set, transmit interrupt is detected. Data from TXData array is copied to UCB0TXBUF and TXByteCtr is decremented. This process is repeated until all the desired bytes in TXData is transmitted (TXByteCtr is zero). Enable stop flag using UCB0CTL1 bit2.

4.7 I²C Receive Interrupt

When UCB0IFG bit0 is set, receive interrupt is detected The received data is copied from UCB0RXBUF iteratively until expected number of bytes (RXByteCtr) is received.

4.8 Setting up the Configuration Register Using Command1Write()

The command sent to the OPT3001 is {0x01, 0xC6, 0x00}, which follows the format {register address, data (MSB), data(LSB)}. The address of configuration register is 0x01.

The next two bytes are written to the register addressed by the register address. The OPT3001 acknowledges receipt of each data byte. The master may terminate the data transfer by generating a start or stop condition.

This function is used to set up the configuration register (0x01) of the OPT3001. RN[12:15] is chosen to be automatic scaling mode. The mode of operation [9:10] is chosen as continuous by making the value 11b.

1. Set the TXByteCtr as 3.
2. Set the UCB0CTLW0 bit4 as 1 to write the command.
3. Ensure the UCB0CTLW0 bit2 (stop bit) is set.
4. Send the start condition by making the UCB0CTLW0 bit1 as 1.
5. Once the last data is copied to UCB0TXBUF set the UCB0CTLW0 bit2 (stop).

4.9 Reading Back the Configuration Using Command1Poll()

1. Set the RXByteCtr as 2.
2. Set the UCB0CTLW0 bit4 as 0 to read/ poll data.
3. Ensure UCB0CTLW0 bit2 (stop bit) is set.
4. Send thr start condition by making the UCB0CTLW0 bit1 as 1.
5. Once the last data is copied to the UCB0TXBUF, set the UCB0CTLW0 bit2 (stop).

4.10 Reading Data Using Command2Write()

To be able to read the data, write from the results register address using the function Command2Write(). The function Command2Write() is used to do a partial write of the register address. The address of result register is 0x00.

1. Set the TXByteCtr as 1.
2. Set the UCB0CTLW0 bit4 as 1 to write the command.
3. Ensure the UCB0CTLW0 bit2 (stop bit) is set.
4. Send the start condition by making the UCB0CTLW0 bit1 as 1.
5. Once the last data is copied to the UCB0TXBUF, set the UCB0CTLW0 bit2 (stop).

Copyright © 2016, Texas Instruments Incorporated

4.11 Reading Register Value Results Using Command2Poll()

This function is used to read the value of results register.

1. Set the RXByteCtr as 2.
2. Set the UCB0CTLW0 bit4 as 0 to read/ poll data.
3. Ensure the UCB0CTLW0 bit2 (stop bit) is set.
4. Send the start condition by making the UCB0CTLW0 bit1 as 1.
5. Once the last data is copied to the UCB0TXBUF, set the UCB0CTLW0 bit2 (stop).

The two-byte value received is of the format Exponent[12:15] and fractional result [0:11].

4.12 Interpreting OPT3001 Readings

When the output of ALS OPT3001 reads above 100 lux (0x29C4), output current of the driver is increased. When the ambient light level decreases to less than 100 lux, the output current of the driver is reduced.

Copyright © 2016, Texas Instruments Incorporated

5 Interface to Remote Temperature Sensor (Using EVM)

The remote temperature sensor monitors the panel temperature or capacitor bank temperature (temperature alarm setting: 50°C). The TMP451 EVM is connected to the MSP430F67791A through the I^{2}C interface. Default configuration values are used for reading the temperature.

5.1 Read/Write Process

When reading data from the temperature sensor, the last value stored in the register address by a write operation determines which register is read during a read operation. To change the register address for a read operation, a new partial I^{2}C write transaction must be initiated. This partial write is accomplished by issuing slaves address byte with the R/W bit low, followed by the register address byte and a stop command. The master then generates a start condition and sends the slave address byte with the R/W bit high to initiate the read command:
[Register address + R/W].

5.2 Order of Commands and Function Calls

- TempCommandWriteMSB(): This function updates the register address of temperature MSB result register (precondition I2C initialization for UCB1 is complete).
- TempCommand2PollMSB(): This function is used to read the temperature result (MSB).
- TempCommandWriteLSB(): This function performs a partial write to update the register address to LSB.
- TempCommand2PollLSB(): This function retrieves the result (2 bytes) from temperature sensor.

The temperature reading is read every four seconds. The MSB of the remote temperature sensor result are stored in register 01h, and the LSBs are stored in register 10h. The MSB value is read first followed by LSB. When MSB is read first, this causes the LSB value to be locked until the value is read (ADC does not write value to LSB until then).

5.3 I^{2}C Initialization

1. Set P4SEL [4:5] to 1 to enable UCB1_SCL and UCB1_SDA.
2. Reset the universal serial communication interface (USCI) module using register UCB1CTLW0. Set bit 0 as 1 and clear all the other bits.
3. Set the SPI clock source as SMCLK (set UCB1CTLW0 bit 6 and bit 7).
4. Select the master mode using UCB1CTLW0 bit 11.
5. Set the communication speed at 100 kbps by making the UCB1BRW register as 0xF0 (uses SMCLK).
6. Calls the function SetSlaveAddress() (Slave address for temperature sensor TMP451: 0x4C).
7. Clear the reset bit UCB1CTLW0 bit 0 to start the I^{2}C module.
8. Enable transmit and receive interrupts by making UCB1IE [0:1] bits as 1.
9. Enable global interrupt using status register (SR) bit 3 (GIE).

5.4 SetSlaveAddress()

1. Set up the slave address for the master to communicate.
2. Set the I^{2}C slave address using register UCB1I2CSA (the slave address of the TMP451 is 0x4C).

5.5 I^{2}C Transmit Interrupt

When UCB0IFG bit1 is set, transmit interrupt is detected. Data from TXData array is copied to UCB1TXBUF and TXByteCtr is decremented. This process is repeated until all the desired bytes in TXData is transmitted (TXByteCtr is zero). Enable stop flag using UCB1CTL1 bit2.

Copyright © 2016, Texas Instruments Incorporated

5.6 I^{2}C Receive Interrupt

When UCB1IFG bit0 is set, receive interrupt is detected. The received data is copied from UCB1RXBUF iteratively until expected number of bytes (RXByteCtr) are received.

5.7 TempCommandWriteLSB()

1. Set the variable CmdPollFlag to 0xFF.
2. Set TXData[0] as 0x10. The pointer register (of TMP451) is set with every write command. A write command must be issued to set the proper value in the pointer register before executing a read command. This command (0x10) is used to read the remote temperature (low byte).
3. Set the TXByteCtr and RXByteCtr as 1.
4. Set UCB1CTLW0 bit4 as 1 to write the command.
5. Ensure UCB1CTLW0 bit2 (stop bit) is set.
6. Send start condition by making UCB1CTLW0 bit1 as 1.

5.8 TempCommandPollLSB()

1. Send the command TempCommandWriteLSB() before TempCommandPollLSB() is called to poll for data.
2. Set the variable CmdPollFlag to 0xFF.
3. Set TXData[0] as 0x10. This command is used to reads the remote temperature (low byte). The resolution of this register is 0.0625°C.
4. Set the TXByteCtr and RXByteCtr as 1.
5. Set UCB1CTLW0 bit4 as 0 to read poll data.
6. Ensure UCB1CTLW0 bit2 (stop bit) is set.
7. Send start condition by making UCB1CTLW0 bit1 as 1.

NOTE:

1. The eight MSBs of the remote temperature sensor result are stored in register 01h, and the four LSBs are stored in register 10h (the four MSBs of register 10h).
2. The four LSBs of the remote sensor indicate the temperature value after the decimal point (for example, if the temperature result is 10.0625˚C, the high byte is 0000 1010 and the low byte is 0001 0000). These registers are read-only and are updated by the ADC each time a temperature is measured.

5.9 TempCommandWriteMSB()

1. Set the variable CmdPollFlag to 0xFF.
2. Set TXData[0] as 0x01. The pointer register (of TMP451) is set with every write command. A write command must be issued to set the proper value in the pointer register before executing a read command. This command (0x01) is used to read the remote temperature (high byte).
3. Set the TXByteCtr and RXByteCtr as 1.
4. Set UCB1CTLW0 bit4 as 1 to write the command.
5. Ensure UCB1CTLW0 bit2 (stop bit) is set.
6. Send start condition by making UCB1CTLW0 bit1 as 1.

Copyright © 2016, Texas Instruments Incorporated

5.10 TempCommand2PollMSB()

1. Send the command TempCommandWriteMSB() before TempCommandPollMSB() is called to poll for data.
2. Set the variable CmdPollFlag to 0xFF.
3. Set TXData[0] as 0x01. This command is used to reads the remote temperature (high byte). The resolution of this register is 10C.
4. Set the TXByteCtr and RXByteCtr as 1.
5. Set UCB1CTLW0 bit4 as 0 to read poll data.
6. Ensure UCB1CTLW0 bit2 (stop bit) is set.
7. Send start condition by making UCB1CTLW0 bit1 as 1.

5.11 Interpreting Remote Temperature Readings

The function TempCommand2PollMSB() sends a request to temperature sensor to read the value of MSB results register. The temperature reading (MSB) is stored in the variable "Temperature". For example, when "Temperature" reads 0x16, the temperature reading is 22°C. This MSB register value is used to compare the over temperature alarm setting.

6 UART Initialization (for GUI)

UART is used to communicate with the GUI (TIDM-3PH-ENERGY5-ESD software). The GUI displays the voltage, current, power factor, and other energy parameters.

To initialize UART:

1. Set P3SEL [6:7] to 1 to enable UART operation for pins UCA2RXD and UCA2TXD.
2. Set UCA2CTL1 bit0 to 1 to reset the UART peripheral.
3. Set UCA2CTL1 as 0x80.
4. Set UCA2CTL0 as 0x00.
5. Set UCA2MCTL as 0x5500 and set UCA2BR register with 0x0A3D for setting up baud rate of communication (9600bps).
6. Clear UCA2TXBUF by making it 0.
7. Enable UART by making UCA2CTL1 bit0 to 0.
8. Enable receive interrupt by making UCA2IE bit 0 as 1.

Copyright © 2016, Texas Instruments Incorporated

7 References

1. Texas Instruments, *Capacitor Bank Switching and HMI Subsystem Reference Design for Automatic Power Factor Controller*, TIDA-00737 Design Guide (TIDUB67)
2. Texas Instruments, *MSP430 Hardware Tools User's Guide* (SLAU278)
3. Texas Instruments, *AC Voltage and Current Transducer With DC Analog Outputs and Digital Output Drivers*, TIDA-00454 Design Guide (TIDUAH1)
4. Texas Instruments, *Implementation of a Three-Phase Electronic Watt-Hour Meter Using the MSP430F677x(A)*, MSP430F677x(A) Application Report (SLAA577)

8 About the Author

VIVEK GOPALAKRISHNAN, Firmware Architect at Texas Instruments India, is responsible for developing reference design solutions for Grid Infrastructure within Industrial Systems. Vivek brings to his role his experience in firmware architecture design and development. Vivek earned his master’s degree in sensor systems technology from VIT University, India. He can be reached at vivek.g@ti.com.

Copyright © 2016, Texas Instruments Incorporated

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions of TI 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated