

## ***Using the SPI Library for TMS37157***

---

*Stefan Recknagel*

### **ABSTRACT**

This document describes the SPI Commands that are necessary to control the TMS37157 passive low-frequency interface (PaLFI) device and access its EEPROM. The functions of the SPI library are in C and can be easily called from any C program that includes the header files. The header files are optimized for the MSP430F2274 and can be easily adapted to other microcontrollers. The SPI library does not require detailed knowledge of SPI, because the details of communication handling are integrated in the functions.

---

### **Contents**

1	Introduction .....	2
2	Hardware Connections Between MSP430 and TMS37157 .....	2
3	SPI Library - Use from C .....	3
4	Included Library Files .....	3
5	Function Description .....	4
6	MSP Access Commands .....	5

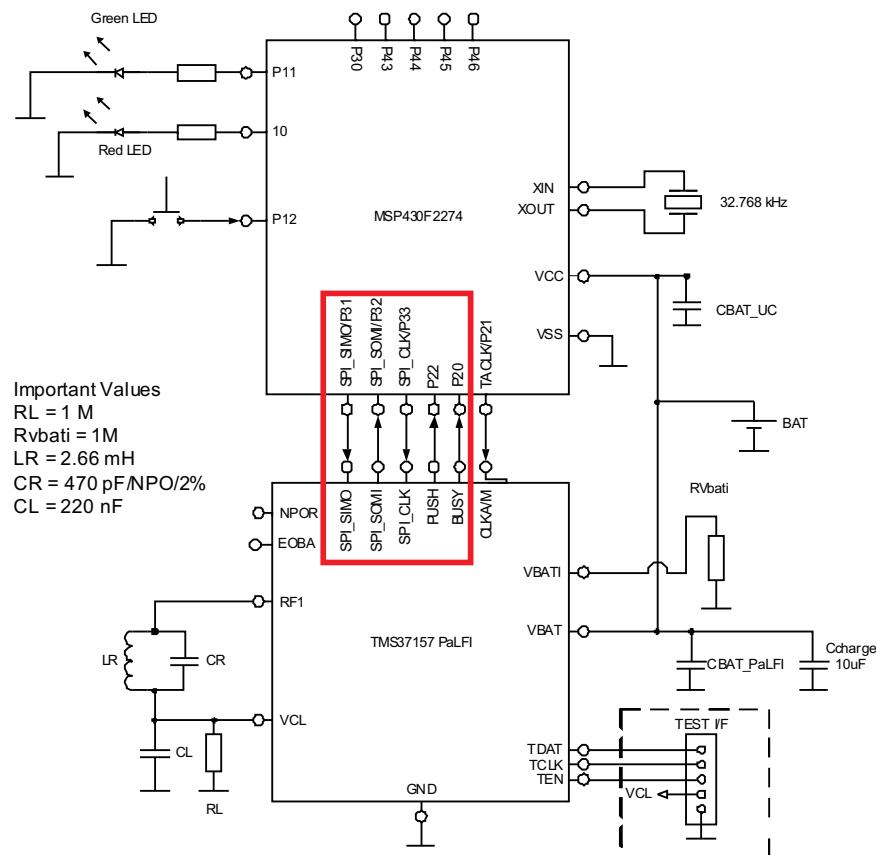
## 1 Introduction

The MSP430F2274 communicates with the TMS37157 through a 3-wire SPI interface. For efficient communication, the busy port of the TMS37157 must be connected to one input port of the MSP430 microcontroller. This library requires the busy port of the TMS37157 connected to one input of the MSP430F2274 and the push port connected to one output port of the MSP430F2274

This application report includes library files that are intended to be compiled in larger projects. These files include all SPI functions that are described in the users guide of the TMS37157; they are identical to the header files used for the eZ430-PaLFI demo tool firmware.

## 2 Hardware Connections Between MSP430 and TMS37157

**Figure 1** shows the necessary connections between the MSP430 and the TMS37157 front end to work together with the SPI library. The port pins of MSP430 can be changed in the `PaLFI_Transponder.h` file. If different ports are used (for example, P1 instead of P2) additional changes must be made in `SPI_Stack.c` and `SPI_LowLevel.c` (for example, P2IN replaced by P1IN), but these changes are trivial.



### Figure 1. SPI Connection, MSP430 to TMS37157

### 3 SPI Library - Use from C

This code example shows how to use the SPI library of the TMS37157 for a memory read and program command.

```
#include "msp430x22x4.h"
#include "PaLFI_Transponder.h"

void main (void)
{
    unsigned char ucPageData[5] = {0};
    Wake_PaLFI();
    SPI_Read_PCU_State();
    SPI_Read_SerialNum();
    SPI_Read_UserPage(Page9,ucPageData);
    SPI_Program_UserPage(Page10,ucPageData);
    SPI_Power_Down();
}
```

## 4 Included Library Files

### 4.1 PaLFI\_Transponder.h

This file contains the definitions for the function and variables used within the SPI library. This file must be adapted if connections between TMS37157 and MSP430F2274 other than those shown in [Figure 1](#) are used.

### 4.2 SPI\_LowLevel.c

This file is for the low level communication between the TMS37157 and MSP430F2274. This file must be adapted if other connections between TMS37157 and MSP430F2274 other than those shown in [Figure 1](#) are used.

### 4.3 SPI\_Stack.c

This file includes the functions for the transponder access commands and the enhanced commands.

## 5 Function Description

Functions that are only used internally by other functions are not described. Initialization functions, transponder access functions, and enhanced command functions are described.

**MSP\_430\_SPI\_Init (void)**— This function initializes the SPI interface of the MSP430F2274. If another microcontroller is used, the function may be adapted. CLK\_AM is not necessary for SPI communication, but it is necessary for trimming. If trimming is not used, CLK\_AM connections can be deleted.

**Wake\_PaLFI (void)**— This function wakes the TMS37157 from standby mode (MSP430F2274 toggles push) and returns when the TMS37157 is ready to receive data (indicated by busy going from high to low).

**NOTE:** This function operates correctly only if the MSP430F2274 is connected to the push port of the TMS37157 and is not powered by VBAT1 of the TMS37157.

**SPI\_Read\_SerialNum (void)**— This function reads the serial number, Page2 and Page1 from the TMS37157. It saves the data in TRP\_Data.

**SPI\_Read\_PCU\_State (void)**— This function reads the PCUsState out of the TMS37157. The PCU state defines if the TMS37157 woke up because of a push interrupt or an MSP access command. If state is 0x01 a push interrupt occurred, if state is 0x02 a MSP access interrupt occurred. The state is saved in TRP\_Data.ucPCU\_Mode.

**SPI\_Read\_UserPage (unsigned char ucPage, unsigned char \*ucData)**— This function reads one user page (ucPage) out of the EEPROM of the TMS37157. The intrinsic values for ucPage are saved in "PALFI\_Transponder.h". The content of the user page is saved in \*ucData.

**SPI\_Program\_UserPage (unsigned char ucPage, unsigned char \*ucData)**— This functions programs \*ucData in one page (ucPage) in the EEPROM of the TMS37157. The intrinsic values for ucPage are saved in "PALFI\_Transponder.h". The content of the user page is saved in \*ucData.

**SPI\_Read\_CU\_Data (unsigned char \*ucData)**— This function should be performed if an MSP access is received by the TMS37157. It receives 6 bytes of data from the front end and saves it to \*ucData. The TMS37157 waits for an answer from the MSP430 (SPI\_Write\_CU\_Data).

**SPI\_Write\_CU\_Data (unsigned char \*ucData)**— This function can only be performed after a SPI\_Read\_CU\_Data command. It sends 6 bytes to the TMS37157 (\*ucData), which forwards this data through its LF interface. The TMS37157 shuts down after sending out the data.

**NOTE:** The data for a program page command is always 5 bytes (except Page 1 and Page 2). The data for an MSP430F2274 access command is always 6 bytes. This must be considered when handing data to the functions.

**SPI\_CLKA\_ON (void)**— This function turns on the LC tank circuit of the TMS37157 and switches it out to the CLK\_AM pin. It introduces a delay for the frequency to stabilize. This is useful for trimming the device.

**SPI\_CLKA\_OFF (void)**— This function turns off the LC tank circuit and the output to CLK\_AM.

**SPI\_TRIM\_WO\_PROG (char data)**— This function programs the trim capacitors of the TMS37157, but it does not program the trim EEPROM. A reset of the TMS37157 restores the values from the EEPROM to the trim capacitors.

**SPI\_TRIM\_W\_PROG (char data)**— This function programs the trim capacitors and the trim EEPROM of the TMS37157.

**SPI\_CRC\_Calc (...)**— This function requests a CRC16 calculation performed by the TMS37157 with the start value 0x3791. ucStart defines which byte of \*ucData is the first byte for the CRC calculation. ucLength defines on how many bytes (including ucStart) the CRC is calculated. The CRC is saved in \*ucCRC.

**SPI\_Power\_Down (void)**— This function turns off the TMS37157. To access the memory of TMS37157, it must be turned on.

## 6 MSP Access Commands

The MSP access commands are special cases—they work only if the TMS37157 receives an MSP access command through its RF interface. The MSP access commands are used to transfer data through the RF interface directly to the microcontroller and back. In the default application, the MSP430F2274 is in LPM4 waiting for an interrupt and the TMS37157 is in standby mode, resulting in overall ultra-low power consumption. If the TMS37157 receives an MSP access command, it sets busy high. This can be used as an interrupt for the MSP430F2274. The TMS37157 shows its readiness by resetting busy. Now the MSP430F2274 can request the data from the TMS37157. The TMS37157 waits until the MSP430F2274 sends 6 bytes of data back to the TMS37157. During this time, the field of the RFID reader must stay on, supplying the TMS37157 with energy. The TMS37157 sends the data back to the RFID reader when the RFID reader switches off the field. The following code example shows how to use the MSP access commands in connection with a busy interrupt. It is assumed that busy is connected to P2.1 of the MSP430F2274.

```
#include "msp430x22x4.h"
#include "PaLFI_Transponder.h"

void main (void)
{
    unsigned char MSP_Access_Data[6] = {0};

    P2OUT = 0; //
    P2DIR &= ~CU_BUSY; // Busy Input P2.1 CU_BUSY = 0x002
    P2IFG &= ~CU_BUSY; // reset busy Interrupt
    P2IE |= CU_BUSY; // busy Interrupt enabled

    While(1)
    {
        if((P2IFG & CU_BUSY) == CU_BUSY) // Test for Interrupt
        {
            While ((P2IN & CU_BUSY) == CU_BUSY); // wait until TMS37157 ready
            SPI_Read_CU_Data(MSP_Access_Data); // read Data from TMS37157
            MSP_Access_Data[1] = MSP_Access_Data[2] + MSP_Access_Data[3]; // change data
            SPI_Write_CU_Data(MSP_Access_Data); // Write Data to TMS37157
            P2IFG &= ~CU_BUSY; // reset Interrupt Flag
            P2IE |= CU_BUSY; // set Interrupt enabled
        }
        __bis_SR_register(LPM4_bits + GIE); // Enter LPM4, global Interrupts Enabled
    }
}

#pragma vector=PORT2_VECTOR
__interrupt void PORT2_ISR(void)
{
    P2IE &= ~CU_BUSY;
    __bic_SR_register_on_exit(LPM4_bits+GIE);
}
```

## 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Mobile Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Transportation and Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

TI E2E Community Home Page

[e2e.ti.com](http://e2e.ti.com)

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2011, Texas Instruments Incorporated