

Ultra-Low-Power 27-MHz Wireless Mouse Reference Design

Randy Wu

AEC MSP430 Products

ABSTRACT

The Texas Instruments (TI) MSP430 is a family of microcontrollers typically used in many high-performance low-power human interface device (HID) applications, such as a wireless mouse or keyboard. This application report describes a fully-functional production-ready wireless mouse reference design consisting of TI's MSP430 ultra-low-power microcontroller, the Avago Technologies' ultra-low-power ADNS-3040 optical mouse sensor, and TI's TRF7950 27-MHz transmitter module. To cover all parts of the solution, the corresponding USB receiver dongle solution is also discussed. Complete hardware schematics, Bill of Materials, and MSP430 transmitter source code are all provided as a complete reference design package with this application report.

The source code that is included with this application report can be downloaded from this link: <http://www-s.ti.com/sc/psheets/slaa302/slaa302.zip>.

Contents

1	Introduction	2
2	Mouse Transmitter: Hardware Architecture	2
3	USB Receiver Dongle: Hardware Architecture	5
4	Mouse Transmitter: Software Architecture	6
5	Conclusion	10
6	References	10
Appendix A	Wireless Mouse Transmitter Schematic	11
Appendix B	Wireless Mouse Transmitter Board Bill of Materials (BOM)	12
Appendix C	USB Receiver Dongle Schematic	14
Appendix D	USB Receiver Dongle Bill of Materials (BOM)	15
Appendix E	Transmitter Module Data-Packet Formats	16

1 Introduction

A wireless mouse design must be implemented with a strategy of maintaining a high-performance system, while keeping power consumption to an optimum level to prolong battery life. The MSP430F1222 was chosen as the microcontroller for this design because of its relatively low pin count (28 pins) and production volume cost. In this specific software design, the MSP430 operates in two modes: ACTIVE (250 μ A at 1 MHz) and DEEP SLEEP (0.1 μ A). Whenever the mouse detects motion, the MSP430 stays in its ACTIVE mode for 80 ms to process various events: gathering X-Y displacement data from the optical sensor, turning on the transmitter module while shifting out encoded data packets on a single serial line, detecting scroll wheel movement (Z displacement), and sensing button presses. After the 80 ms of constant inactivity, the MSP430 automatically goes into a DEEP SLEEP mode until mouse movement is detected by the optical sensor, or buttons have been pressed. While the mouse is in its DEEP SLEEP mode, the entire unit draws less than 4 mA.

2 Mouse Transmitter: Hardware Architecture

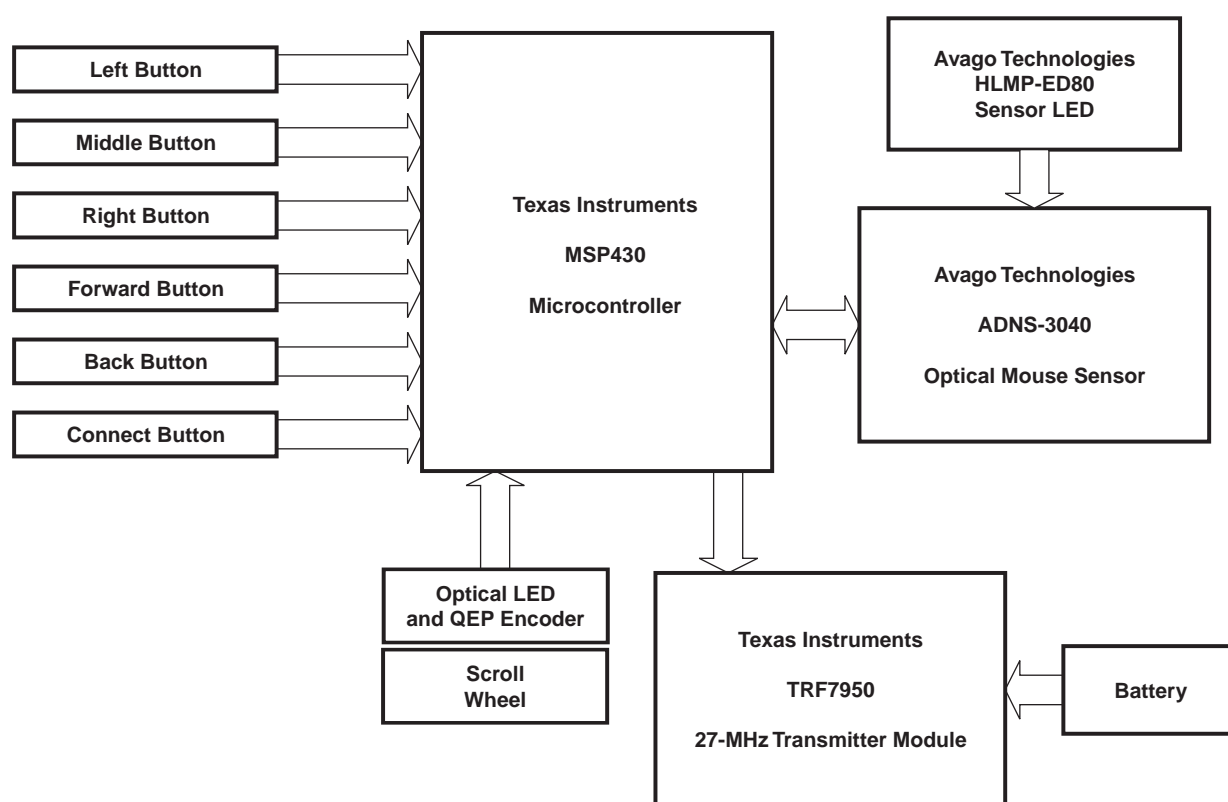


Figure 1. Mouse Transmitter System Block Diagram

2.1 Texas Instruments MSP430 Microcontroller

The TI MSP430 family of ultra-low-power microcontrollers consists of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that attribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 1 μ s.

The MSP430x11xx (20-pin devices with reduced peripherals) and MSP430x12x2 (28-pin devices to

accommodate additional I/O) series are ultra-low-power mixed-signal microcontrollers with a built-in 16-bit timer, 10-bit A/D converter with integrated reference and data transfer controller (DTC), and 14 (20-pin package) or 22 (28-pin package) general-purpose I/O (GPIO) pins. The MSP430x12x2 series microcontrollers have built-in communication capability using asynchronous (UART) and synchronous (SPI) protocols.

2.1.1 Oscillators and Timers

During MSP430 ACTIVE mode (and upon waking up from DEEP SLEEP mode via motion or button interrupt), the internal DCO is recalibrated based on an external 32.768-kHz watch crystal, which is energized only upon wake-up to conserve power. The DCO is software configurable for a default frequency of 4 MHz, which is then used to clock the CPU (for processing all the various pending events) and the Timer_A module. This 16-bit timer module is used as the time base to periodically interrupt at variable intervals based on the timing of the serial data going out to the transmitter module. During ACTIVE mode, the external 32.768-kHz auxiliary clock (ACLK) remains on as the time base to drive the watchdog timer to generate an interrupt every 250 ms, checking for potential changes in the scroll-wheel position, as well as keeping track of the inactivity time-out timer. Once the time-out timer has expired, the MSP430 turns off the CPU, all peripherals, and all clocks before going into DEEP SLEEP.

2.1.2 Serial Interface

The MSP430F1222 has an on-board universal synchronous asynchronous receiver transmitter (USART) module which can be configured as a hardware synchronous serial peripheral interface (SPI) or universal asynchronous receiver transmitter (UART). The Avago ADNS-3040 optical mouse sensor communicates with any SPI-compliant microcontroller, thus the MSP430 USART is configured as a Master SPI device to initiate all communications activity with the sensor and provide the master clock input to the sensor.

2.1.3 Debugger Interface

A JTAG-compliant emulation module is built into the MSP430 to support real-time debugging and flash memory updates. The JTAG-specific lines are brought out to a connector on the wireless mouse main board to facilitate an easy connection to a host development PC.

2.1.4 General-Purpose and Interrupt Port Pins

Port pins on the MSP430 are used as either interrupt for detecting button presses or as GPIOs used for enabling and disabling the other smaller devices in the system (e.g., QEP encoder enable and outputs, scroll wheel optical LED, sensor shutdown, and sensor chip select).

2.2 Avago ADNS-3040 Optical Mouse Sensor

The ADNS-3040 is an ultra-low-power optical navigation sensor. It has new low-power architecture and automatic power-management modes, making it ideal for battery and power-sensitive applications, such as cordless input devices. The ADNS-3040 is capable of high-speed motion detection (up to 20 ips and 8g). In addition, it has an on-chip oscillator and LED driver to minimize external components.

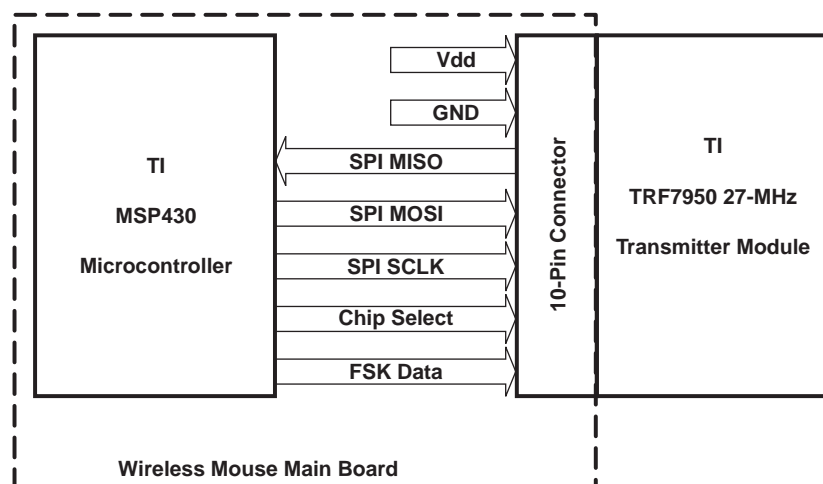
The ADNS-3040, ADNS-3120-001 lens, ADNS-2220 clip, and HLMP-ED80 LED form a complete and compact mouse tracking system. There are no moving parts, which leads to higher reliability and less maintenance for the end user. In addition, precision optical alignment is not required, facilitating high-volume assembly. The sensor is programmed via registers through a four-wire SPI port. It is packaged in a 20-pin DIP.

The ADNS-3040 is based on optical navigation technology (ONT), which measures changes in position by optically acquiring sequential surface images (frames) and mathematically determining the direction and magnitude of movement. The ADNS-3040 contains an image acquisition system (IAS), a digital signal processor (DSP), and a four-wire serial port. The IAS acquires microscopic surface images via the lens and illumination system. These images are processed by the DSP to determine the direction and distance of motion. The DSP calculates delta-X and delta-Y relative displacement values that can be understood by any standard mouse driver.

An external microcontroller reads the ΔX and ΔY information from the sensor serial port. The microcontroller then translates the data into PS2, USB, or RF signals before sending them to the host PC or game console. See Appendix A of this application report for a complete schematic of the main wireless mouse reference design board, including the power components and MSP430-to-ADNS-3040 optical-sensor interconnections.

2.3 TI 27-MHz TRF7950 Transmitter Module

The MSP430 communicates with the TRF7950 transmitter module via a standard three-wire SPI interface, two GPIO pins (see Figure 2) used to enable the TRF7950 device (Chip Select), and another pin for the serial data, which is fed into the modulator (FSK Data).



NOTE: Transmit board stacks on top of main board.

Figure 2. MSP430 Interface to Transmitter Module

2.4 Mechanical Components (Scroll Wheel and Buttons)

In addition to optical mouse movement and motion sensing, the mechanical components typically include a scroll wheel and various buttons (the scroll wheel can also act as a button if depressed). In this reference design, a total of six buttons are implemented: Left, Scroll (middle), Right, Forward, Back, and Connect. Based on the host-PC mouse driver implementation, the Left, Scroll, and Right buttons may have standard and/or customized functions. The Connect button is used to associate the mouse with its corresponding receiver dongle on the PC end.

From a hardware standpoint, all buttons are implemented with standard push-button switches that are tied between a 100-k Ω pullup resistor (going to V_{CC}) and an MSP430 GPIO pin that is configured as an edge-triggered interrupt. Each time a button is pressed or released, the input logic state at the GPIO pins changes between V_{CC} (released) and GND (pressed), which generates an interrupt to the MSP430. Whenever a rising or falling edge is detected at any of these button input pins, the MSP430 can gather the various button states and send the status back to the receiver dongle.

Scroll-wheel position sensing is implemented using a standard two-channel quadrature encoder chip (which contains two photo sensors) and an optical LED aimed at the midway point of each spoke on the scroll wheel. An incremental quadrature encoder is a digital (angular) position sensor using photo-optics. Within the MSP430, an internal quadrature encoder pulse (QEP) state machine is maintained, so that it can increment or decrement a counter based on the direction and movement of the scroll wheel.

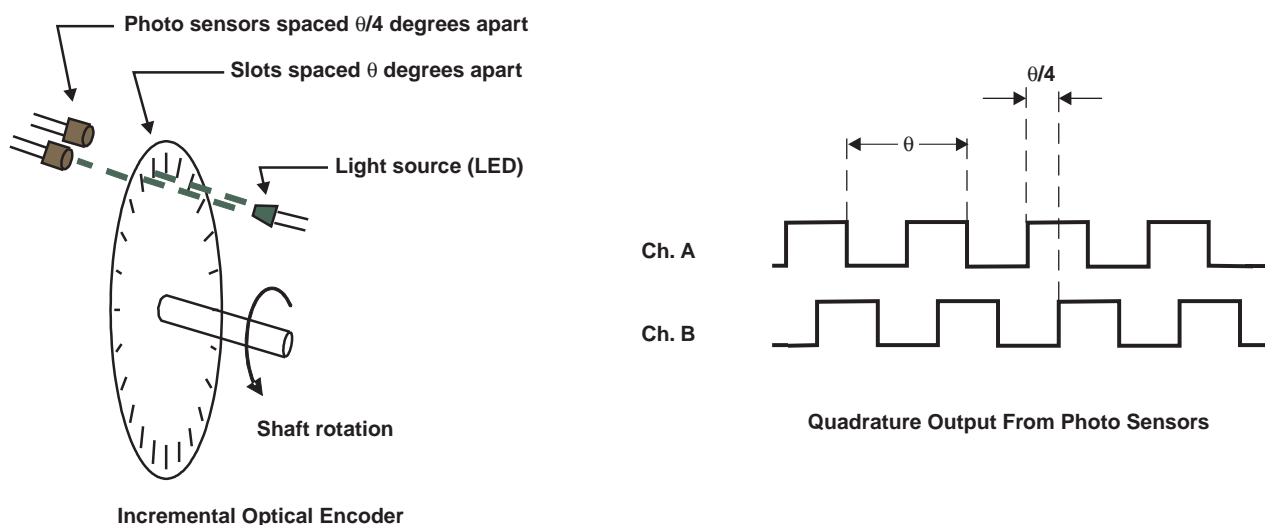


Figure 3. Quadrature Encoder Implementation

3 USB Receiver Dongle: Hardware Architecture

3.1 TI TRF7900 27-MHz RF Receiver

The TRF7900 27-MHz RF receiver is used in the USB receiver dongle. The incoming RF data is demodulated and presented as an asynchronous data stream to the microcontroller of the USB dongle connected to the host PC.

This device is a dual-channel integrated RF receiver designed for human interface devices (HIDs). Operating at 27 MHz, it integrates multiple components (PLLs, RF mixers, simulated SAW filters, tuning circuits, and miscellaneous passive components) to provide frequency selection from eight discrete channels. This integration reduces component, manufacturing, and system costs, while simplifying the system-level design.

The receiver generates eight frequencies (utilizing a single 12-MHz crystal) that can be programmed via the I²C interface bus for the mouse channel. Another 6-MHz clock is internally generated to clock the USB microcontroller. This enables a single crystal to be used to supply the clock needs for the receiver channels and the microcontroller.

The optimized receiver design enables simultaneous reception on both parallel channels up to 5 kHz (10-kbps encoding) per channel. A receive strength signal indicator (RSSI) register can be read via the I²C bus to determine the signal strength of each independent channel. The end user can then utilize their software to correlate the RSSI reading to the packet error rate (PER), which enables the receiver to determine if the transmit signal is too weak (e.g., weak battery or out of range), or if there is too much noise on the selected channel in the user's area.

3.2 Cypress CY7C63743 USB Microcontroller

The Cypress CY7C63743 device is used as the integrated microcontroller and USB solution to process the incoming data packets received by the TI TRF7900 receiver chip and to decode the encoded packets to present the mouse information to the mouse driver on the host PC.

See Appendix C and Appendix D for detailed schematics and Bill of Materials (BOM) of a USB receiver dongle solution that works in conjunction with the MSP430 wireless mouse implemented in this reference design. The fully functional and tested Cypress microcontroller code has also been provided with this application report.

4 Mouse Transmitter: Software Architecture

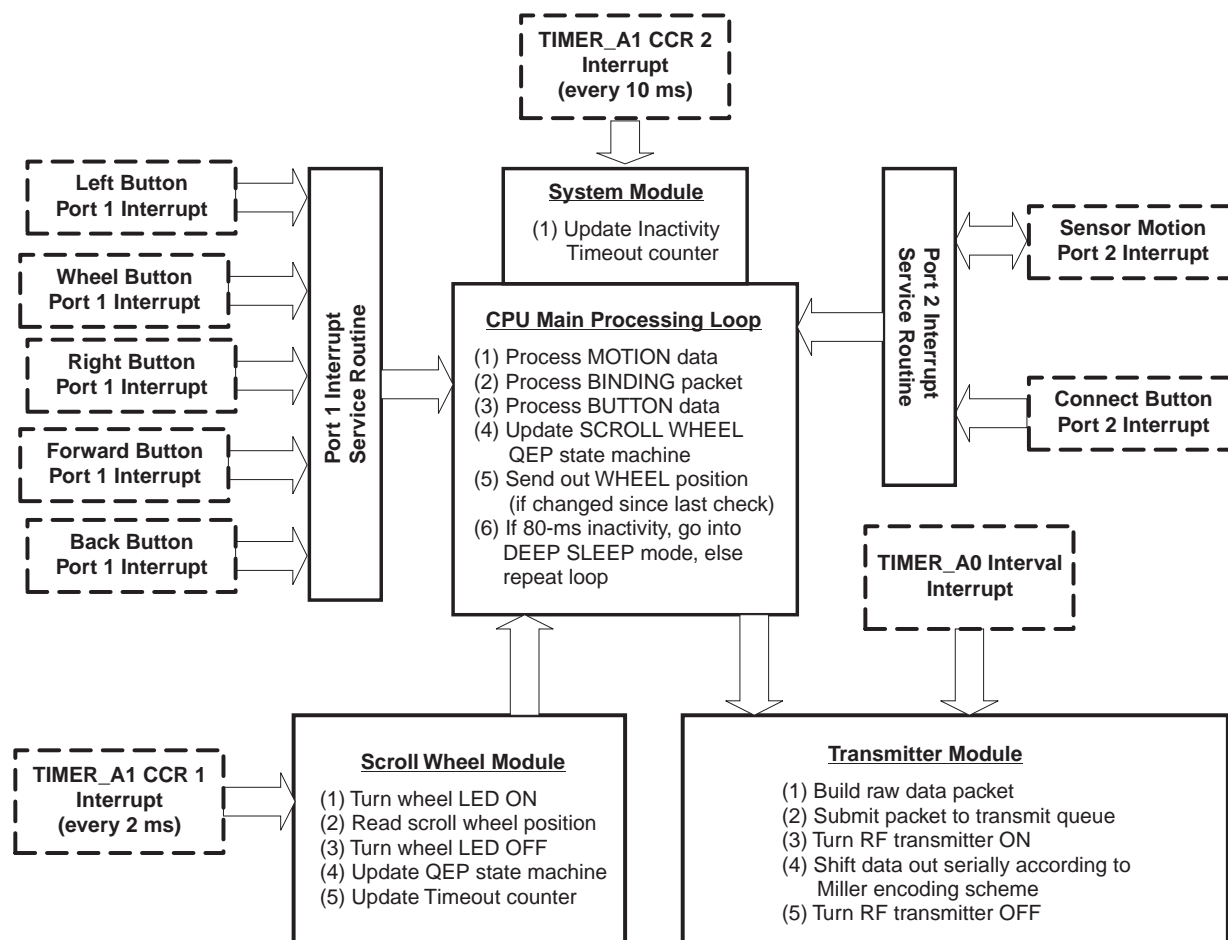


Figure 4. Mouse Transmitter Software Flow Diagram

4.1 MSP430 Low-Power Modes

The MSP430 can operate in one of five low-power modes (LPM0 to LPM4), LPM4 being the deepest sleep (with RAM retention) drawing only 0.1 μA . While in LPM4, the CPU, peripherals, and all internal and external clocks are completely shut off. Only an external interrupt can wake up the MSP430 from its DEEP SLEEP mode in less than 1 μs . Once an external interrupt is detected, the CPU can be switched on and perform the necessary processing for sensor motion detection, button clicks, and scroll-wheel movement. When the CPU is in ACTIVE mode, it only draws 250 μA for every 1 MHz. The MSP430 stays in ACTIVE mode until 80 ms of constant inactivity is recorded. At that point, the MSP430 goes back into its DEEP SLEEP mode, waiting for another interrupt via optical sensor motion detection or button clicks. Scroll wheel movements are updated only during ACTIVE mode; however, once a scroll wheel change has been detected, the MSP430 stays in ACTIVE mode for 5 seconds, to allow for subsequent changes in the scroll wheel movement.

4.2 Avago ADNS-3040 Optical Mouse Sensor Device Driver

A complete device driver layer for the ADNS-3040 optical sensor has been provided as part of the MSP430 reference design software framework. The device driver takes advantage of the MSP430F1222 on-board USART configured as a synchronous master SPI port to control the optical sensor. This highly modular and portable software component was written completely in C to promote reusability in actual production software designs. The following APIs are provided:

```
void SPI0_writeByte(unsigned char data);
// Usage:      Transmits an 8-bit data value out via the SPI0
// Parameters:  8-bit data to be transmitted
// Returns:     Nothing

void ADNS3040_delayOperation(void);
// Usage:      Execute a set delay to separate each access to the ADNS-3040
// Parameters:  None

unsigned char ADNS3040_readRegister(unsigned char regAddress);
// Usage:      Reads a byte from the ADNS3040
// Parameters:  Desired register address
// Returns:     Most currently received data

void ADNS3040_writeRegister(unsigned char regAddress, unsigned char data);
// Usage:      Writes a byte from the specified ADNS3040 register
// Parameters:  Desired register address & data
// Returns:     Nothing

void ADNS3040_readMotionBurst(unsigned char *motion, unsigned char *deltaX, unsigned char *deltaY);
// Usage:      Reads the Motion, Delta_X, and Delta_Y registers in a burst sequence
// Parameters:  Addresses for the locations for motion, deltaX, & deltaY data to be modified
// Note:       This sequence of operations will automatically clear the Motion bit

void ADNS3040_enable(void);
// Usage:      Asserts (active low) /NCS for the ADNS-3040 sensor
// Parameters:  None
// Returns:     Nothing

void ADNS3040_disable(void);
// Usage:      De-asserts (deactive high) /NCS for the ADNS-3040 sensor
// Parameters:  None
// Returns:     Nothing

void ADNS3040_shutdown(void);
// Usage:      Completely shuts down the ADNS-3040 sensor
// Parameters:  None
// Returns:     Nothing

void ADNS3040_init(void);
// Usage:      Executes proper sequence to bring up the ADNS-3040 sensor
// Parameters:  None
// Returns:     Nothing
```

4.3 TI TRF7950 Transmitter Device Driver

A complete device driver layer for the TI TRF7950 transmitter has been provided as part of the MSP430 reference design software framework. The device driver takes advantage of the MSP430F1222 on-board USART configured as a synchronous master SPI port to control the transmitter. This highly modular and portable software component was written completely in C to promote reusability in actual production software designs. The following APIs are provided:

```
void TRF7950_delayOperation(void);
// Usage:      Execute a set delay to separate each access to the TRF7950
// Parameters:  none
// Returns:     nothing

unsigned char TRF7950_readRegister(unsigned char regAddress);
// Usage:      Reads a byte from the TRF7950
// Parameters:  Desired register address
// Returns:     Most currently received data

void TRF7950_writeRegister(unsigned char regAddress, unsigned char data);
// Usage:      Writes a byte from the specified TRF7950 register
// Parameters:  Desired register address & data
// Returns:     nothing

void TRF7950_enableCS(void);
// Usage:      Asserts (active low) CS for the TRF7950 device
// Parameters:  none
// Returns:     nothing

void TRF7950_disableCS(void);
// Usage:      De-asserts (deactive high) CS for the TRF7950 device
// Parameters:  none
// Returns:     nothing

void TRF7950_enableTX(void);
// Usage:      Turns on the transmitter of the TRF7950 device
// Parameters:  none
// Returns:     nothing

void TRF7950_disableTX(void);
// Usage:      Turns off the transmitter of the TRF7950 device
// Parameters:  none
// Returns:     nothing

void TRF7950_disableCLO(void);
// Usage:      Turns off the clock output of the TRF7950 device
// Parameters:  none
// Returns:     nothing

WM430_SYS_Bool TRF7950_checkLowBattery(void);
// Usage: Checks to see if the BAT input voltage drops below the low batt setting
// Parameters:  none
// Returns:     TRUE if battery voltage is low, FALSE otherwise

WM430_SYS_Bool TRF7950_init(void);
// Usage:      Executes proper sequence to bring up the TRF7950 device
// Parameters:  none
// Returns:     TRUE if success, FALSE otherwise
```

4.4 Scroll-Wheel Decoding

Since power consumption is the main concern in battery applications, the optical LED is enabled only when the MSP430 needs to read the current states of the QEP inputs to the MSP430 port pins while in ACTIVE mode. The optical LED is pulsed on for approximately 40 μ s every 2 ms while in ACTIVE mode to read the current position of the scroll wheel, thus saving power, since the optical LED is on for a duty cycle of only 2%. See Figure 5 for the state machine decoding logic.

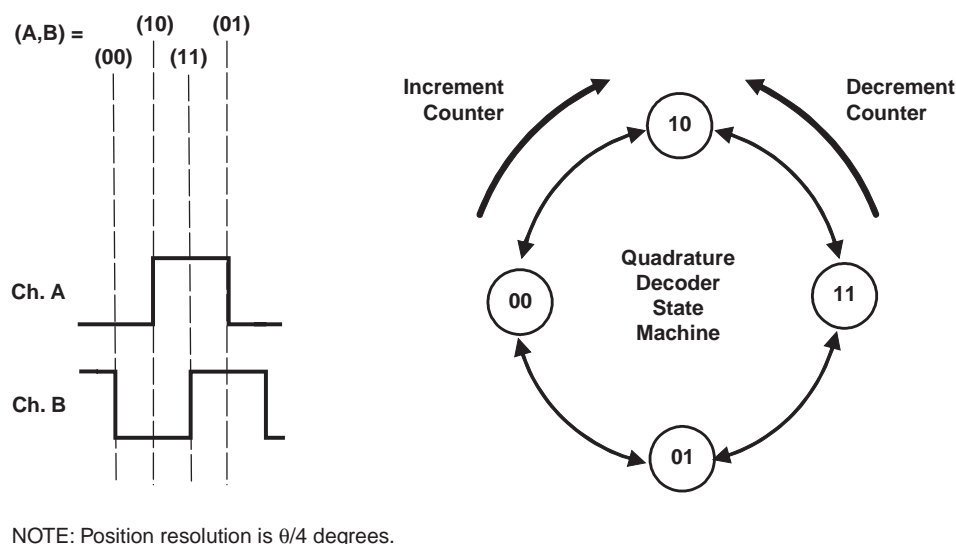


Figure 5. Quadrature Decoder State Machine

4.5 Miller Encoding and Cyclic Redundancy Check (CRC) for Reliable Data Transmission

A serial data rate of 10 kbps was chosen for this design (every transmit bit period = 100 μ s). As with all data transmissions, there is always a possibility of interference or noise that could corrupt the bits in each data packet sent from the wireless mouse to its corresponding receiver module on the host PC side. Thus, every data packet generated by the MSP430 includes an 8-bit CRC value appended at the end of the packet to ensure data integrity when the wireless mouse receiver is decoding the incoming data packet and presenting the information to the mouse driver. See Appendix E for a detailed description of the actual data packet format and protocols.

Because the data transmission is via RF link, an appropriate and efficient RF encoding scheme is applied to the raw serial data before it is shifted out to the modulator input of the transmitter module. The Miller encoding scheme has been implemented in this software reference design.

In the telecommunications world, Miller encoding is the encoding of binary and text files. The binary data is used to form a two-level signal such that (a) a logical 0 causes no change of signal level unless it is followed by another logical 0, in which case a transition to the other level takes place at the end of the first bit period, and (b) a logical 1 causes a transition from one level to the other in the middle of the bit period. Miller encoding is used primarily for encoding radio signals, because the frequency spectrum of the encoded signal contains less low-frequency energy than a conventional non-return-to-zero (NRZ) signal and less high-frequency energy than a biphase signal. The benefit of the Miller encoding scheme is that it does not add any extra bits to the original data stream being encoded. The Miller encoding scheme is also known as Delay encoding.

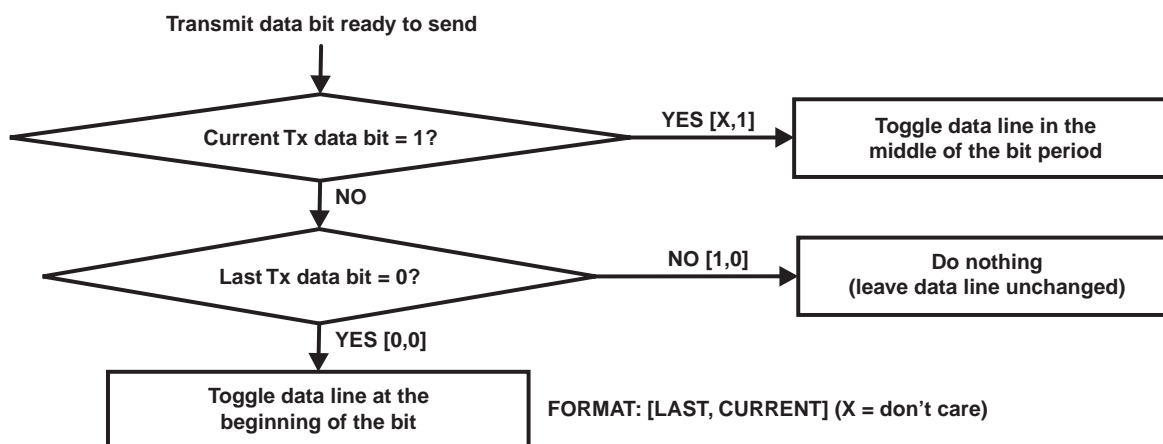


Figure 6. Miller Encoding Scheme Flow Chart

5 Conclusion

The purpose of this application report is to provide any low-power wireless mouse designer a hardware and software template for creating a customized application involving an ultra-low power microcontroller and optical mouse sensor. The fully-functional and production-ready hardware and software footprints provided with this application report serve to jump start any OEM wireless mouse design where high performance in conjunction with optimum ultra-low-power conservation is required.

6 References

1. MSP430x1xx Family User's Guide (SLAU049)
2. MSP430x11x2, MSP430x12x2 Mixed Signal Microcontroller data sheet (SLAS361)
3. TRF7900 27-MHz RF Receiver data sheet (SWRS031) (www.ti.com)
4. TRF7950 27-MHz RF Transmitter data sheet (www.ti.com)
5. Avago ADNS-3040 Ultra Low Power Optical Sensor data sheet (www.avagotech.com)

Appendix A Wireless Mouse Transmitter Schematic

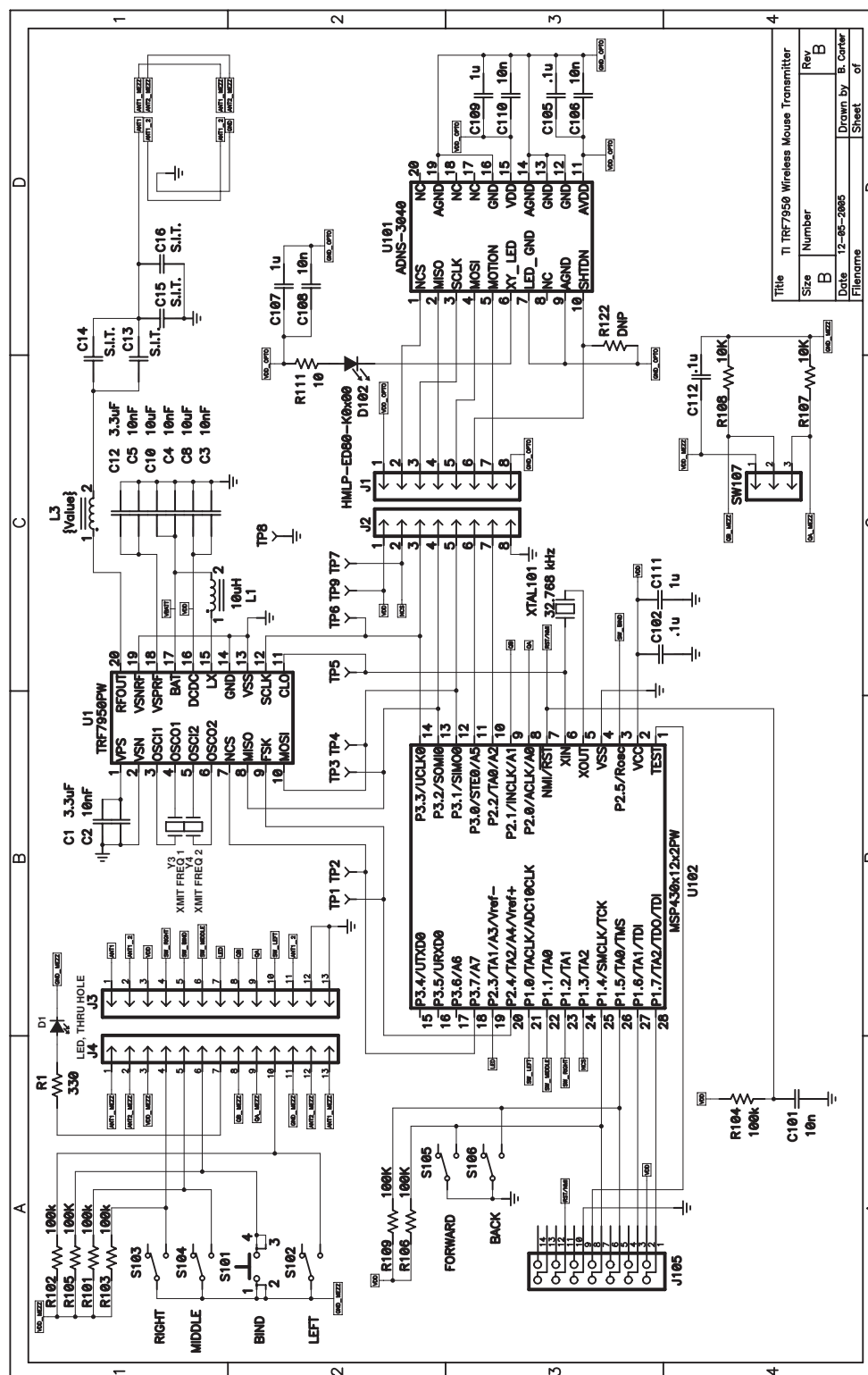


Figure A-1. Wireless Mouse Transmitter Schematic

Appendix B Wireless Mouse Transmitter Board Bill of Materials (BOM)

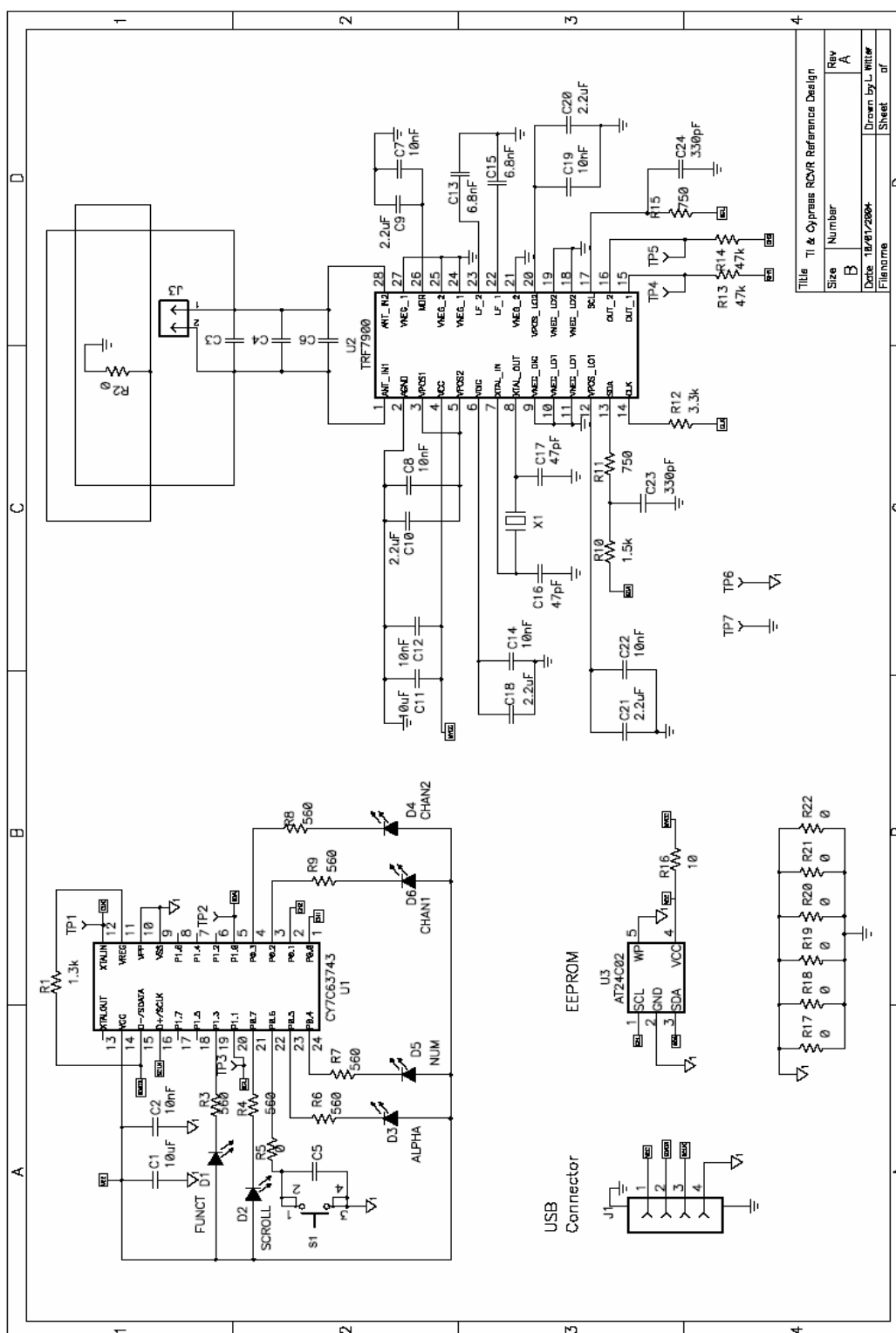
Table B-1. Wireless Mouse Transmitter Board BOM

ITEM	COUNT	REF DESIGN	VALUE	MFR	PART NO.	DESCRIPTION	SOURCE	SOURCE PN	NOTES
1	8	C101	10 nF			Capacitor, Ceramic, 0603 case			
		C106							
		C108							
		C110							
		C2							
		C3							
		C4							
2	3	C102	.1 μ F			Capacitor, Ceramic, 0603 case			
		C105							
		C112							
3	3	C107	1 μ F			Capacitor, Ceramic, 0603 case			
		C109							
		C111							
4	2	C1	3.3 μ F			Capacitor, Ceramic, 0805 case			
		C12							
5	2	C8	10 μ F			Capacitor, Ceramic, 0805 case			
		C10							
6	4	C13	S.I.T.			Capacitor, Ceramic, 0603 case			Antenna matching components - see TRF7950 data sheet for calculation
		C14							
		C15							
		C16							
7	1	J105		Mil-Max	833-93-100-10-001000	Header, Double Row 2 mm, 14 pin	DigiKey	ED93100-ND	Comes in 50 pin lengths - snap into 14 pin sections (7 per)
8	1	J1		Amp	535541-6	Header, 8-pin, 100 mil spacing, F, low profile	DigiKey	A26477-ND	
9	1	J2		Sullins	PTC36SFAN	Header, 8-pin, 10 0mil spacing, M	DigiKey	S1212-36-ND	Comes in 36 pin lengths - snap into 8 pin sections (4 per)
10	1	J4		Sullins	PTC36SAAN	Header, 14-pin, 100 mil spacing, (36-pin strip)	DigiKey	S1012-36-ND	Comes in 36 pin lengths - snap into 8 pin sections (2 per)
11		J3							DO NOT INSTALL - Placeholder for pad pattern only! Mates with pins from Mezzanine board.
12	1	L1	10 μ H	Murata	LQH32MN100J23L	Inductor, SMT	DigiKey	490-2505-1-ND	
13	1	L3	S.I.T.			Inductor, SMT, 1608			Antenna matching component - see TRF7950 data sheet for calculation
14	1	R111	10 Ω			Resistor, 0603 case			
15	1	R1	330 Ω			Resistor, 0603 case			
16	2	R107	10 k Ω			Resistor, 0603 case			
		R108							
17	7	R101	100 k Ω			Resistor, 0603 case			
		R102							
		R103							
		R104							
		R105							
		R106							
		R109							
18	1	R122	open			Resistor, 0603 case			DO NOT INSTALL – for test purposes ONLY
19	1	S101		Panasonic	EVQPAC07K	Switch, SPST, push button	DigiKey	P8011S-ND	

Table B-1. Wireless Mouse Transmitter Board BOM (continued)

20	3	S102		Omron	D2F-01F	Switch, SPDT, snap action, vertical mount	DigiKey	SW502-ND	
		S103							
		S104							
21	2	S105		Omron	D2F-01F-A1	Switch, SPDT, snap action, side mount	DigiKey	D2F-01F-A1-ND	
		S106							
22	1	S107		Panasonic	EVQVX____	Switch, encoder, 11 mm height			Rotary switch - scroll wheel. Part number incomplete - consult Panasonic for more details: http://www.panasonic.com/industrial/components/pdf/ATC000QCE20.pdf
23	8	TP1		Keystone	5005	Test Point, 0.062 Hole, Red	DigiKey	5005K-ND	
		TP2							
		TP3							
		TP4							
		TP5							
		TP6							
		TP7							
		TP9							
24	1	TP8		Keystone	5006	Test Point, 0.062 Hole, Black	DigiKey	5006K-ND	
25	1	U1		TI	TRF7950PW	IC, 27 MHz Wireless Mouse and KB System			
26	1	U102		TI	MSP430F1222IPW	IC, Mixed Signal Microcontroller	DigiKey	296-13781-5-ND	
27	1	U101		Avago	ADNS-3040	IC, Ultra Low-Power Mouse Sensor			Used in conjunction with items 28 to 30
28	1	U101		Avago	ADNS-3120-001	Lens			
29	1	U101		Avago	ADNS-2220	Clip			
30	1	D102		Avago	HLMP-ED80-PS000	Diode. LED, Sensor-Based			
31	1	D1				Diode. LED, Red, T1	DigiKey	67-1066-ND	
32	1	XTAL101	32.768 kHz	Citizen	CMR200TB32.768KDZFTR	Crystal, 32.768 kHz	DigiKey	300-2065-2-ND	DO NOT INSTALL - for test purposes ONLY
33	1	Y3	XMIT FREQ 1		ATS256SM-T	Crystal			Frequencies selected by application – see TRF7950 data sheet
34	1	Y4	XMIT FREQ 2		ATS256SM-T	Crystal			

Appendix C USB Receiver Dongle Schematic



Appendix D USB Receiver Dongle Bill of Materials (BOM)

Table D-1. USB Receiver Dongle BOM

COUNT	REF DESIGN	DESCRIPTION	SIZE	MFR	PART NUMBER
1	C1, C11	Capacitor, Ceramic, 10 μ F, 6.3V, X7R, \pm 5%	805	AVX	0805D106KAT2A
16	C2, C7, C8, C12, C14, C19, C22	Capacitor, Ceramic, .01 μ F, 50V, X7R	603	AVX	06035C103KAT2A
2	C23, C24	Capacitor, Ceramic, 330 pF, 50V, NPO	603	AVX	06035A331JAT2A
6	C5, C9, C10, C18, C20, C21	Capacitor, Ceramic, 2.2 μ F, 16V, Y5V	805	AVX	0805YG225ZAT2A
2	C16, C17	Capacitor, Ceramic, 47 pF, 50V, NPO	603	Sams	CL10C470JBNC
2	C13, C15	Capacitor, Ceramic, 6800 pF, 50V, X7R	603	AVX	06035C682KAT2A
3	C3, C4, C6	Open	603		
2	D4, D6	Diode, LED, Amber	603	Liteon	160-1183-1-ND
4	D1, D2, D3, D5	Diode, LED, Green	603	Liteon	160-1180-1-ND
1	J1	Connector, USB, (Type A)	0.472 \times 0.821	Conec	33UBA-RS1-04-P-W-1
1	J3	Open			
1	R2	Open	603		
1	R5	Resistor, Chip, 0 Ω , 1/16W	603	Sams	RC1608J000CS
1	R16	Resistor, Chip, 10 Ω , 1%, 1/10W	603	KOA	RK73H1JLTD10R0F
4	R3, R4, R6, R7, R8, R9	Resistor, Chip, 562 Ω	603	PHYC	9C06031A5620FKHFT
2	R11, R15	Resistor, Chip, 750 Ω , 1%, 1/16W	603	ROHM	MCR03F7500EZF
2	R13, R14	Resistor, Chip, 47.5 k Ω	603	KOA	RK73H1JTDD4752F
1	R1	Resistor, Chip, 1.3 k Ω	603	KOA	RK73H1JTDD1301F
1	R10	Resistor, Chip, 1.5 k Ω	603	KOA	RK73H1JTDD1501F
1	R12	Resistor, Chip, 3.32 k Ω	603	PHYC	9C06031A3321FKRFT
6	R17, R18, R19, R20, R21, R22	Open	805		
1	S1	Switch, 1P1T, PB Momentary, 100 mA, SM		Digi-Key	7914G-000ECT-ND
5	TP1, TP2, TP3, TP4, TP5	Test Point, Red	1 mm		
2	TP6, TP7	Test Point, Black	1 mm		
1	U1	Open			
1	U2	TRF7900		TI	TRF7900PW
1	U3	EEPROM		Microchip	24LC02BT-1OT
1	X1	Crystal, 12 MHz, 18 pF		Digi-Key	300-6127-1-ND
1	–	PCB, 0 in \times 0 in \times .062 in		Any	Cypress/TI HID Receiver

Appendix E Transmitter Module Data-Packet Formats

Table E-1. Standard Data-Packet Format (Total 48 Bits = 6 Bytes)

6 Bits	8 Bits	8 Bits	8 Bits	8 Bits	8 Bits	2 Bits
Sync Pattern	ID Byte	Header Byte⁽¹⁾	Most Significant Data Byte⁽¹⁾	Least Significant Data Byte⁽¹⁾	Checksum Byte	Stop Pattern

⁽¹⁾ **Bolded** fields indicate that more detailed information is available below.

Table E-2. Header 8-Bit Byte Format

M/K ⁽¹⁾	Reserved	Not Used	Not Used	Not Used	Reserved	Type ⁽²⁾	Type ⁽²⁾
--------------------	----------	----------	----------	----------	----------	---------------------	---------------------

⁽¹⁾ 1 = Mouse, 0 = Keyboard

⁽²⁾ 00 = Mouse ID change
01 = Mouse X-Y displacement
10 = Mouse button and Z displacement
11 = Manufacturing test data

Data Bytes Format

(1) Binding/Identification

- Most significant data byte = Frequency code
- Least significant data byte = New ID code

Table E-3. Frequency Code 8-Bit Byte Format

Reserved	Old FC	Old FC	Old FC	Reserved	New FC	New FC	New FC
----------	--------	--------	--------	----------	--------	--------	--------

Table E-4. Frequency Codes

CODE	FREQUENCY (MHz)
000	27.xxxxx
001	27.xxxxx
010	27.xxxxx
011	27.xxxxx
100	27.xxxxx
101	27.xxxxx
110	27.xxxxx
111	27.xxxxx

(2) X-Y Displacement

- Most significant data byte = Signed X displacement
- Least significant data byte = Signed Y displacement

(3) Button and Z Displacement

- Most significant data byte = Button states
- Least significant data byte = Signed Z displacement

Table E-5. Button States 8-Bit Byte Format (0 = Button Released, 1 = Button Pressed)

Not Used	Not Used	Not Used	Forward	Back	Middle	Right	Left
----------	----------	----------	---------	------	--------	-------	------

Table E-6. Z-Displacement 8-Bit Signed Byte Format (–127 to +127)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-------	-------	-------	-------	-------	-------	-------	-------

Notes:

- Every data packet must be preceded with a Start Pattern consisting of at least ten (10) ones (1s) in a row.
- Sync Pattern = 1 0 0 1 1 1 (an intentionally illegal Miller encoded pattern)
- ID = 8-bit value (cannot be 0xFF) created by a random number generator (at startup, the ID defaults to 0xFF)
- Checksum byte is calculated by applying the polynomial $x^8 + x^2 + x^1 + x^0$ (seed value = 0) on the previous four bytes of the overall data packet (ID, Header, Most Significant Data Byte, then Least Significant Data Byte)
- Stop Pattern = all 1s

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.

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
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Copyright © 2006, Texas Instruments Incorporated