Project Final Report:
Directional Remote Control

by

Luca Zappaterra – xxxx@gwu.edu

CS 297 – Embedded Systems

The George Washington University

April 25, 2010

Project Abstract

In the Directional Remote Control project, a prototype of TV remote control which reacts to the user's hand movement is implemented. Commercial remote controls are characterized by many buttons which send different information to the TV through Infrared Data Association (IrDA) [1] protocol. Conversely, the Directional Remote Control reacts to the user tilting the device on the left or on the right, by transmitting the increase or decrease in value of the feature selected. A button on the remote allows selecting between three functions the user can change: volume, channel and color settings. In example, when the user wants to increase the volume, he will firstly push the button a number of times necessary to select the volume function, and then it will incline the device to the right, thus increasing the value. The selected function and value are shown on onboard display and transmitted to the receiver. The project includes also the implementation of a receiver capable of decoding the function and value remotely received. These values will be displayed on the receiver LED screen.

1. Status

The project works as planned due to the ease of configuring the IrDA communication using the ZNEO Z16 F microcontroller. The accelerometer chosen for the project did not create major issues, since the right component for the project was chosen during the proposal. One possible evolution of the project could be to implement more sophisticated functions as well as test the system with a commercial TV IrDA protocol.

2. Specification

The project enables communication between two modules: a transmitter and a receiver. The transmitter is composed by one accelerometer connected to a ZNEO Z16 F Contest Kit board [3]. It is in charge of decoding the user movements and the functions selected, display them and transmit to the other end of the communication. The receiver, implemented on another ZNEO Z16 F Contest Kit board, decodes the signals received and visualizes the received signals. A block diagram of the system is shown in Figure 1.

Both the two terminals, when firstly connected to power, enters an IDLE state waiting for events to happen. The receiver will only perform the commands received by the transmitter. The transmitter has a button (SW1) which selects the function the user wants to change. When pushed for the first time, it will go from the IDLE state into the VOLUME state, visualizing the “Vo” characters on the display and it enables the IrDA transmitter to send out the modified value for the VOLUME. By pushing the button again, the system goes into CHANNEL state, visualizing the “Ch” characters and transmitting out the related values. The third state possible, reached by pushing the button when in CHANNEL state, is COLOR state. It behaves in the same ways as the other

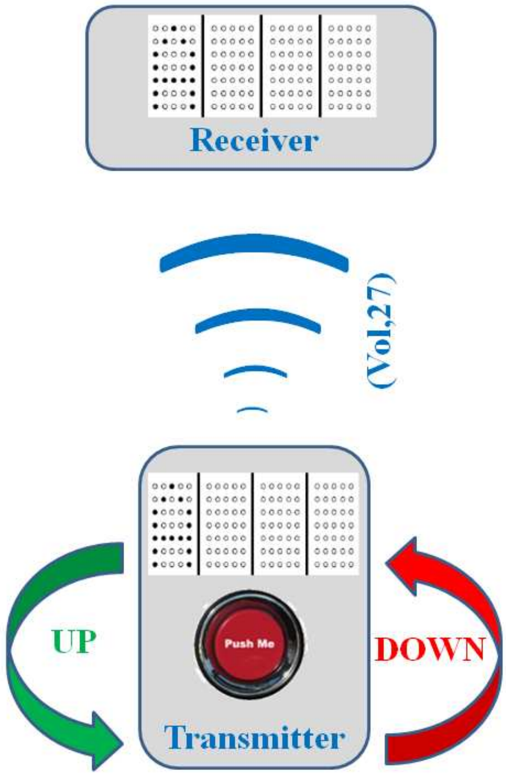


Figure 1. Directional Remote Control Block Diagram.

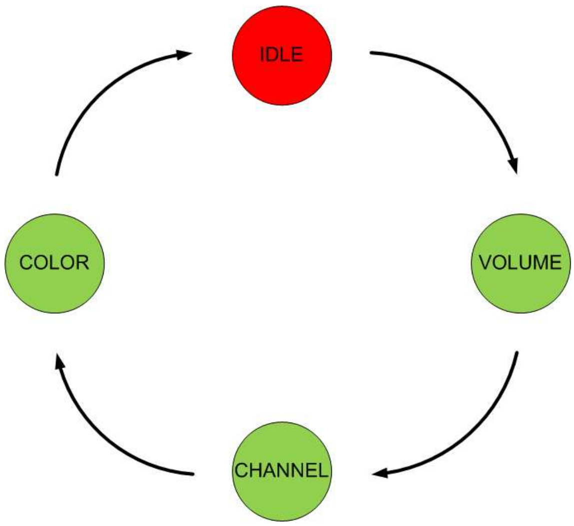


Figure 2. Transmitter StateDiagram.

states, visualizing "Co" on the display and transmitting the values. Finally, if the button is pressed when in COLOR state, it will go back in IDLE, turning off the screen and not transmitting any data. The transmitter state diagram is shown in Figure 2. The values for every function are in the range of integers between 0-40 with default value 20.

The accelerometer connected to the transmitter board detects its inclination in respect to the Y-axis and in case the device is not put in horizontal position, it will increase or decrease the value of the function selected. An increase happens when the accelerometer is rotated clock-wise. Oppositely, the selected value is decreased if rotated counterclock-wise. For every change of value (e.g. from 20 to 19 to18, etc.) the transmitter broadcasts wirelessly through IrDA the current pair Function-Value and it displays it on the four LED matrixes present on the ZNEO board using two matrixes for the function and two for the value (e.g. Vo20, Vo19, Vo18, etc.)

The receiver board, after turning on, keeps its IrDA receiver constantly on, waiting for data sent by the transmitter. When it recognizes that a pair Function-Value has been delivered through the wireless medium, it displays it on its LED array in the same way as described for the transmitter. Additionally, a power saving strategy is implemented which turns off the LED array after some time of inactivity.

3. Implementation & Construction

Figure 3 shows the actual implementation of the system. The three main components are highlighted in red. In this section the hardware and software components of the system are presented in detail.

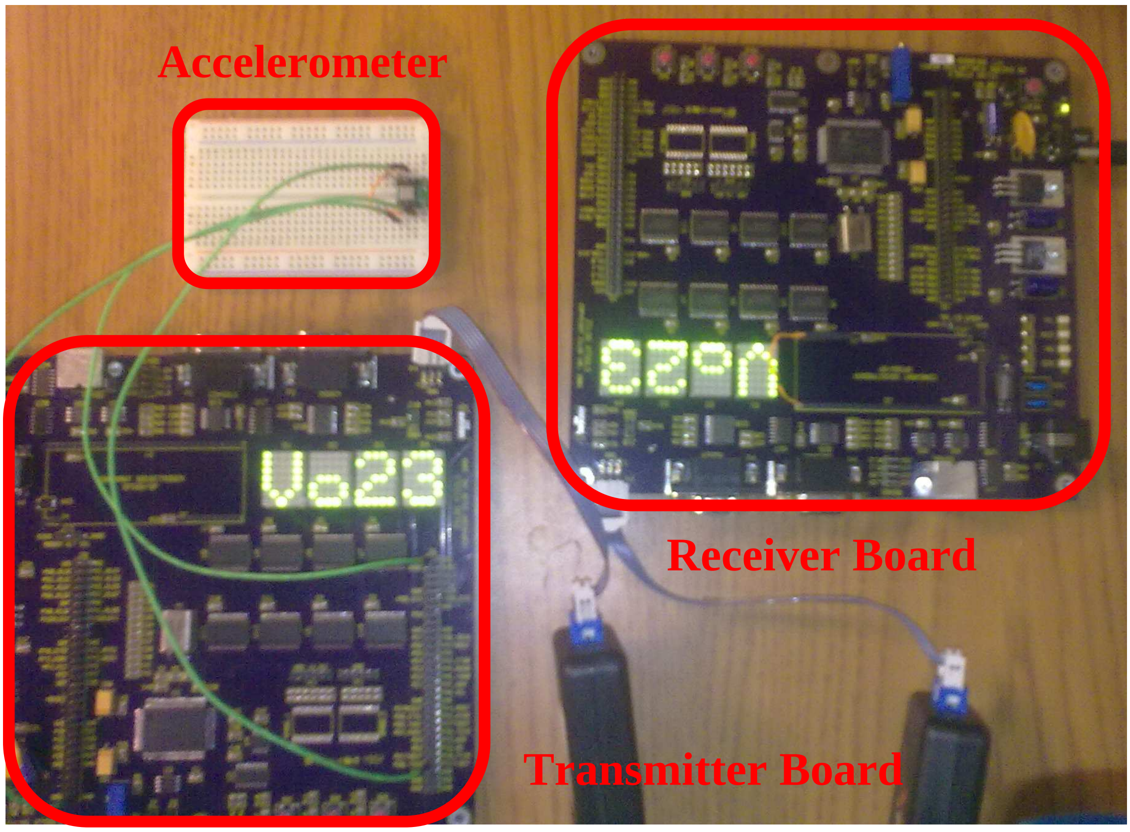


Figure 3. Directional Remote Control Implementation.

3.1. Hardware Components

As highlighted in Figure 3, all hardware components are derived by two a ZNEO Z16 F Contest Kit board and by the externally connected Memsic 2125 Dual-Axis Accelerometer [2]. The accelerometer is mounted on a breadboard. In the following subsections, the single components are described, specifying with letters T and R if used at the transmitter or receiver.

3.1.1. Pushbutton SW1 (T)

The pushbutton SW1 on the ZNEO Z16 F Contest Kit board is directly connected to port PD3 of the Z16 microcontroller, which allows using interrupts for the line. The signal is by default high, and it is pulled to low when the button is pressed.

3.1.2. Memsic 2125 Dual-Axis Accelerometer (T)

The Memsic 2125 Dual-Axis Accelerometer detailed specifications are attached at the end of the document. Here the characteristics relevant to the project are presented. Figure 4 shows how the accelerometer is connected to the microcontroller ports.

- PIN 1 (Tout) gives the analog output of the temperature, which is not used in the project.
- PIN 2 (Y_{out}) provides the inclination value over the Y-axis as a Pulse Width Modulated (PWM) output. See communication protocol section in the attached documents for more details. Basically, the inclination of the device is encoded in the duty cycle value: when the duty cycle is smaller than 50%, the device is tilted to left, otherwise it is tilted to the right. If 50%, the device is horizontally positioned.

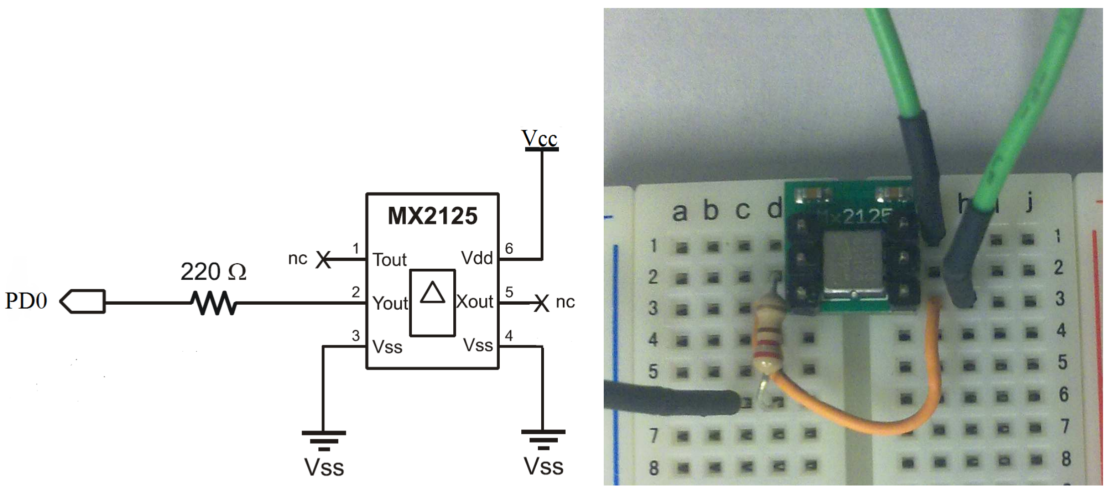


Figure 4. Memsic 2125 Dual-Axis Accelerometer connections.

PIN 2 is connected to port PD0 on the Z16 microcontroller with a resistor of 220 Ω in series, as required by datasheet.

- PIN 3 and PIN 4(Vss) are connected to one of the ground pins (GND) of the Z16 microcontroller.
- PIN 5 (X_{out}) provides the inclination value over the X-axis using the same methodology as PIN 2.
- PIN 6 (V_{cc}) is connected to the V_{cc} pin of the Z16 microcontroller, which provides 3.3 V.

3.1.3. ZNEO Z16F Series (Z16F2811AL) Microcontroller (T,R)

The ZNEO Z16 microcontroller [4] is the core of both the transmitter and the receiver modules. Here only its hardware components relevant for the project are described.

<u>Oscillator</u>

For the specific application, no high frequency oscillator is needed. The 5.5 MHz internal precision oscillator satisfies the requisites of both the transmitter and the receiver.

<u>Timers</u>

Two timers are used in the ZNEO Z16 microcontroller on the transmitter side. One timer is used to sample at a constant rate the input value provided by the accelerometer. The sample rate is set to 43.2 KHz (i.e. reload value = 1, prescale =128), which is more than enough to poll the data line accurately for this application.

The second timer is used both at the transmitter and at the receiver for periodically refreshing the values of the LED display lines. The frequency is set to 1KHz.

<u>UART</u>

The Universal Asynchronous Receiver/Transmitter (UART) is responsible for encoding (at the transmitter) and decoding (at the receiver) data in 8 bit format. In the project implementation, CTS and bit parity check functionalities are not implemented.

At the transmitter side, the UART gets enabled only when there is a value to be sent, based on the accelerometer data. The transmitter waits for the transmission buffer to empty and then it writes to the U0TXD register the byte to be transmitted. When the system goes into IDLE function, the UART is disabled. Oppositely, at the receiver side, the UART is always enabled and ready to decode the received data. It continuously checks the status of the buffer, to see when it is full (i.e. byte received). When a byte is received, it is decoded and the buffer is emptied. In the project, a baud rate of 57.6 KHz has been set but it is possible to modify it, without impacting the application operations.

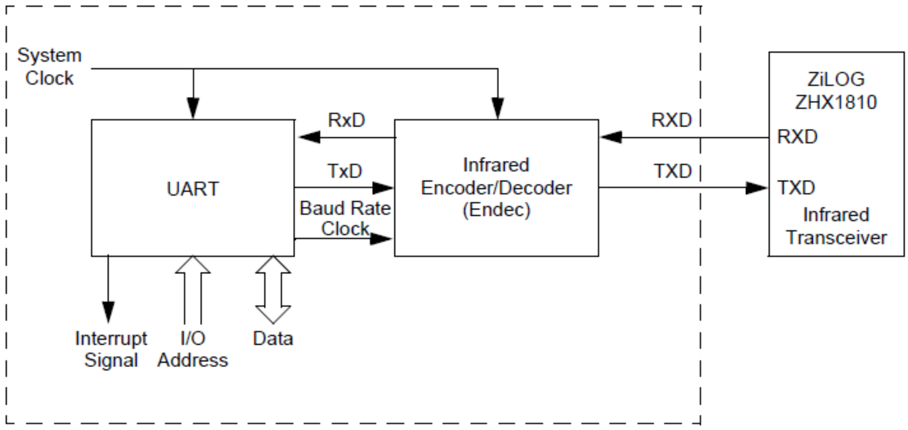


Figure 5. IrDA Data Comunication System Block Diagram.

IrDA

The IrDA encoding/decoding is made possible in both the transmitter and the receiver by setting the IREN bit to 1 in the U0CTL1 register. With this simple configuration, data is encoded in accordance with the IrDA standard and output to the infrared transceiver via the TXD pin. The communication is one-directional, so The ZNEO Z16 microcontroller IrDA-UART architecture is illustrated in Figure 5.

Thanks to the ease of configuration provided by the ZNEO Z16 microcontroller IrDA-UART blocks, there is no need of covering the IrDA transmission protocol. In the project, IrDA communication was achieved by configuring UART to use IrDA

3.1.4. IrDA Transceiver (T,R)

A fully-functional, high-performance UARTs with Infrared Encoder/Decoders (Endec) is present on the ZNEO Z16 F Contest Kit board. The transmitter board uses the transceiver only for data transmission, while the receiver board uses it only for data reception. The IrDA transceiver gets enabled by setting the jumper J10 status to IN.

3.1.5. LED Array (T,R)

The LED array displays the pair Function-Value in the format described above (e.g. Vo20). It is composed by four 7x5 LED matrixes. To light up an LED dot the appropriate Anode bit must be 1, and the correlated Cathode must be 0. All Anodes are addressed by Port G of the Z16 microcontroller, and Cathodes are addressed by Port E of the microcontroller.

3.2. Software Components

The main software building blocks for the project are presented here, divided in transmitter and receiver components.

3.2.1. Transmitter Software Components

The application running on the transmitter ZNEO Z16 microcontroller is shown in Figure 6. The two blocks Init and Timers on the left are necessary blocks for the application but do not implement directly the functionalities of the system. The Button Handler, Accelerometer Handler, UART/IrDA Transmitter and LED Programmer are the functional blocks that implement the system's operation. As the flow in the figure shows, the Button Handler is the root that calls the Accelerometer Handler. From this block, the UART/IrDA transmitter component implements the wireless communication with the receive, whereas the LED programmer displays the current Function-Value on the display. Here the main functionalities of every block are described.

<u>Init</u>

- Set the device to use internal oscillator
- Initialize Timers and set their interrupt vectors
- Set Interrupt Service Routine for when the button is pressed
- Configure GPIO data directions
- Enable interrupts
- Initialize UART for IrDA transmission

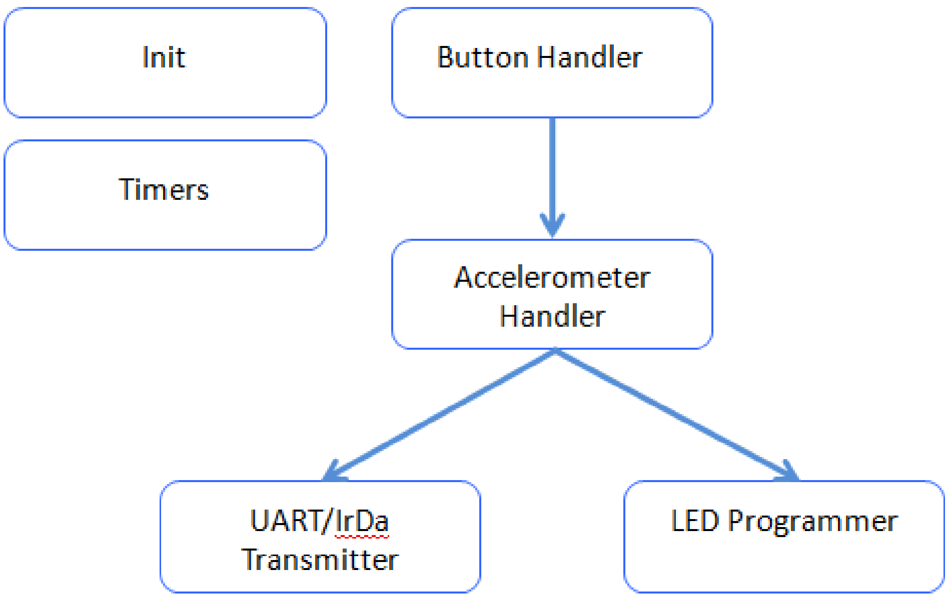


Figure 6. Transmitter Software Components.

Timers

- Set Timer0 as a clock to poll the value at GPIO connected to the accelerometer
- Set Timer1 to periodically refresh the LED display

Button Handler

- Generate an interrupt every time the button SW1 is pressed
- Switch between states, as shown in Figure 2, modifying the current Function
- Enable/Disable IrDA transmission and the display switching from/to IDLE state

Accelerometer Handler

- Continuously poll the accelerometer data line to identify the device inclination
- Three intervals identifying the inclination of the device
- Based on the interval recorded, decide if to decrease, not modify or increase the current value
- If the value is not modified stall, otherwise call the procedures to transmit the new value and to visualize it on the display

UART/IrDA Transmitter

- Add an offset value to encode the Function to be sent
- Check if the transmission buffer is empty
- Transmit the Function-Value within a byte of data

LED Programmer

- Set the four LED matrixes to visualize the Function-Value pair
- Refresh the value periodically

3.2.2. Receiver Software Components

The application running on the receiver ZNEO Z16 microcontroller is shown in Figure 7. Many blocks are a subset of the blocks of the transmitter. With the channel polling procedure, the device checks if there is incoming data on the wireless medium. If there is a communication ongoing, the IrDA/UART Decoder receives the byte of data and decodes the Function-Value parameters, which are displayed on the LED screen through the LED Programmer module.

Init

- Set the device to use internal oscillator
- Initialize Timers and set their interrupt vector
- Configure GPIO data directions

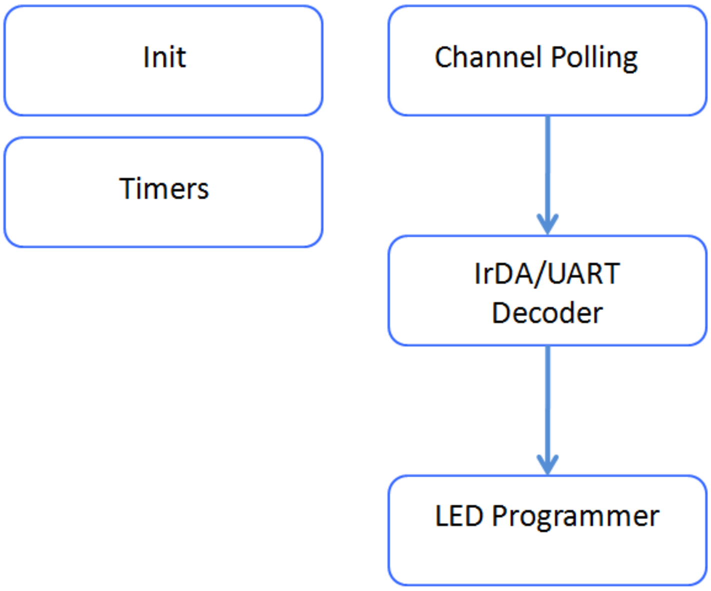


Figure 7. Receiver Software Components.

- Enable interrupts
- Initialize UART for IrDA reception

<u>Timers</u>

- Set Timer1 to periodically refresh the LED display

<u>Channel Polling</u>

- Check for the UART receiver buffer to fill

<u>IrDA/UART Decorer</u>

- Decode the byte received through IrDA
- Extract the Function-Value received pair

<u>LED Programmer</u>

- Set the four LED matrixes to visualize the Function-Value pair
- Refresh the value periodically

4. Retrospective

Probably the most challenging aspects of the project were designing the application correctly and choose the hardware properly. The digital PWM output of the Memsic 2125 Dual-Axis Accelerometer had made the job of retrieving the inclination value simple. Moreover the UART-IrDA implementation on the ZNEO Z16 also facilitates the configuration of a wireless link between two context kit boards.

Another great help in the implementation/debug of the project came from using an oscilloscope to see the output of the accelerometer. Without visually see the data line of the device it would have been harder to retrieve the inclination value from the device.

Debugging possibilities were very limited during the project, since the UART was selected to transmit through IrDA, therefore no RS232 communication (e.g. for printf) was possible. The problem was solved by debugging through the use of on board LEDs and taking advantage of the loopback mode of the UART, which loops the output data back to the input of UART.

The system could be modified by using the inclination level of Axis-X, available on the accelerometer, as a way of changing the function (i.e. volume, channel, etc.) instead of using the button.

Attachments

Memsic 2125 Dual-Axis Accelerometer (#28017) Specifications

The Memsic 2125 is a low-cost thermal accelerometer capable of measuring tilt, collision, static and dynamic acceleration, rotation, and vibration with a range of ±3 g on two axes. Memsic provides the 2125 IC in a surface-mount format. Parallax mounts the circuit on a tiny PCB providing all I/O connections so it can easily be inserted on a breadboard or through-hole prototype area.

Features

- Measures ±3 g on each axis
- Simple pulse output of g-force for each axis
- Convenient 6-pin 0.1” spacing DIP module
- Analog output of temperature (TOut pin)
- Fully temperature compensated

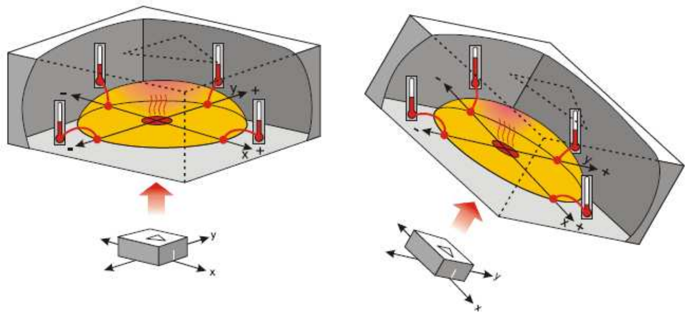
Key Specifications

- Power Requirements: 3.3 to 5 VDC; < 5 mA supply current

- Communication: TTL/CMOS compatible 100 Hz PWM output signal with duty cycle proportional to acceleration
- Dimensions: 0.42 x 0.42 x 0.45 in (10.7 x 10.7 x 11.8 mm)
- Operating temperature: 32 to 158 °F (0 to 70 °C)

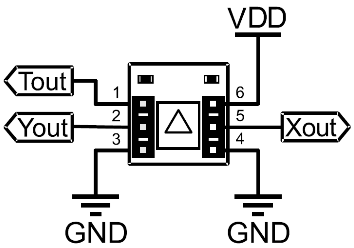
Theory of Operation

The MX2125 has a chamber of gas with a heating element in the center and four temperature sensors around its edge. When the accelerometer is level, the hot gas pocket rises to the top-center of the chamber, and all the sensors will measure the same temperature. By tilting the accelerometer, the hot gas will collect closer to some of temperature sensors. By comparing the sensor temperatures, both static acceleration (gravity and tilt) and dynamic acceleration (like taking a ride in a car) can be detected. The MX2125 converts the temperature measurements into signals (pulse durations) that are easy for microcontrollers to measure and decipher.



Pin Definitions

Pin	**Name**	**Function**
1	Tout	Temperature Out
2	Yout	Y-axis PWM output
3	GND	Ground -> 0 V
4	GND	Ground -> 0 V
5	Xout	X-axis PWM output
6	VDD	Input voltage: +3.3 to +5 VDC



Communication Protocol

Each axis has a 100 Hz PWM duty cycle output in which acceleration is proportional to the ratio tHx/Tx. In practice, we have found that Tx is consistent so reliable results can be achieved by measuring only the duration of tHx. This is easy to accomplish with the BASIC Stamp PULSIN command or with the Propeller chip's counter modules. With Vdd = 5V, 50% duty cycle corresponds to 0 g, but this will vary with each individual unit within a range of 48.7% to 51.3%. This zero offset may be different when using Vdd = 3.3 V.

References

[1] IrDA Protocol specifications - http://www.irda.org/

[2] Memsic 2125 Dual-Axis Accelerometer (#28017) v2.0 1/29/2009 - Copyright © Parallax Inc.

[3] ZNEO™ Z16F Series Flash Microcontroller Contest Kit User Manual PS022006-0207. Copyright ©2007 by ZiLOG, Inc.

[4] High Performance Microcontrollers ZNEO™ Z16F Series PS022006-0207 Product Specification. Copyright ©2007 by ZiLOG, Inc.