Project Proposal

Directional Remote Control

February 16, 2010
Luca Zappaterra – lucaz@gwu.edu

Project Abstract

In the Directional Remote Control project, a prototype of TV remote control which reacts to the user's hand movement will be implemented. Commercial remote controls are characterized by many buttons which send different information to the TV through infrared. Conversely, the Directional Remote Control will react to the user tilting the device on the left or on the right, by transmitting via infrared the increase or decreasein value of the feature selected. Additionally, a button on the remote will permit to change the function the user wants to modify: volume, channel, color settings, etc. 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. Possibly, a verification module could be added to the system. In example, a display showing the function and value transmitted can be implemented on the Directional Remote Control or at a receiver module.

Strategy

The Directional Remote Contol device is composed of four hardware blocks, shown in Figure 1 on the next page.

The **Function Button** is the input that enables the system: after being pushed, allowing the user to select the function, the device will retrieve the inclination data from the accelerometer. It will be implemented using the user configurable pushbutton *SW1* embedded in the *ZNEO™ Z16F Series Flash Microcontroller Contest Kit.* A de-bouncing technique will need to be implemented to avoid unexpected results.

The **Accelerometer** represents the command interface between the user and the device. In a simple implementation, it will measure the tilt over a single axis and recognize the progressive movement of the user. *Memsic 2125 Dual-Axis Accelerometer* is a known digital output solution which produces a 100 Hz PWM output signal with duty cycle proportional to the acceleration in the determined direction. Its DIP module availability makes it convenient to be inserted on a breadboard for prototyping.

The **ZiLog Microcontroller** *ZNEO™ Z16F* is the core of the system and it is responsible for processing the information from the function button and the accelerometer and encoding the data for the infrared transmitter and the LED display.

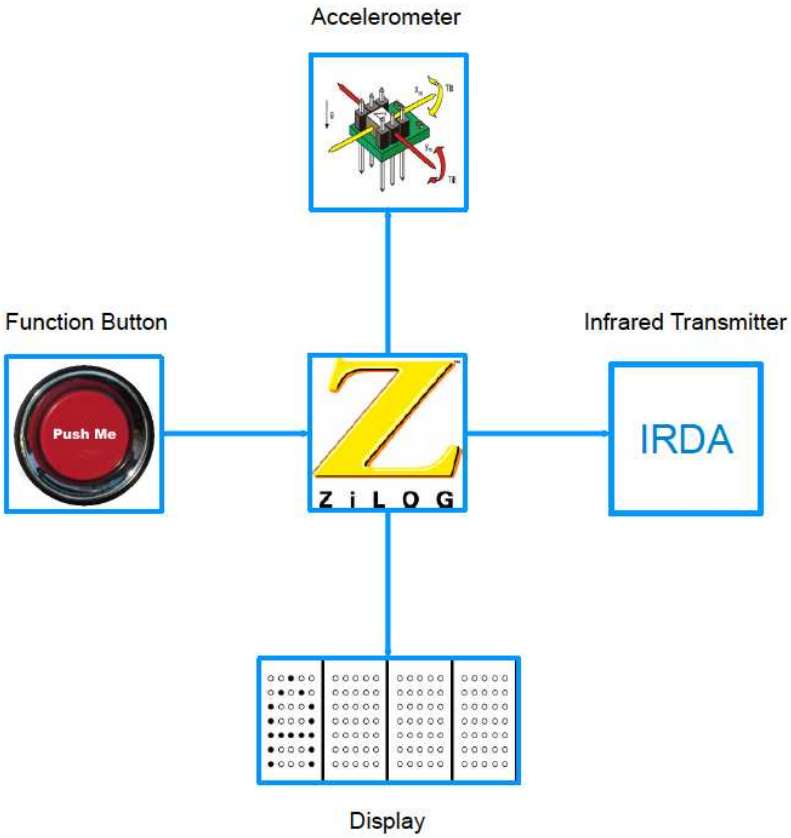

Figure 1. Hardware Block Diagram

In order to interact with the four external components, it will make large use of its GPIO ports and timers and it will need to be programmed carefully to handle the interrupts. Data transfer to the infrared transmitter will be handled by the *Universal Asynchronous Receiver/Transmitter (UART)* and *Infrared Encoder* blocks embedded on the Z16. The data to transmit will go though the UART which creates the transmit signal and the baud rate clock for the infrared encoder to generate the modulated signal.

The **Infrared Transmitter** component present on *U21* of the Z16 contest kit implements the infrared communication transmitting the modulated signal produced by the infrared encoder, according to the protocol physical specifications. Both the function selected and the values triggered by the accelerator need to be transmitted.

The **Display** is composed by four 7x5 LED matrixes which can be used to show the values that will be transmitted by the remote control. The first matrix could write a function number, while the other three matrixes could be used to visualize a range of values representing the movement of the user (e.g. from –X to +X).

Different software modules need to be implemented on the ZiLog Microcontroller. They are schematically drawn in Figure 2. In the block diagram are present six components, namely: **Button Handler**, **Accelerometer Handler**, **Data Processing**, **LED Programmer**, **IRDA**

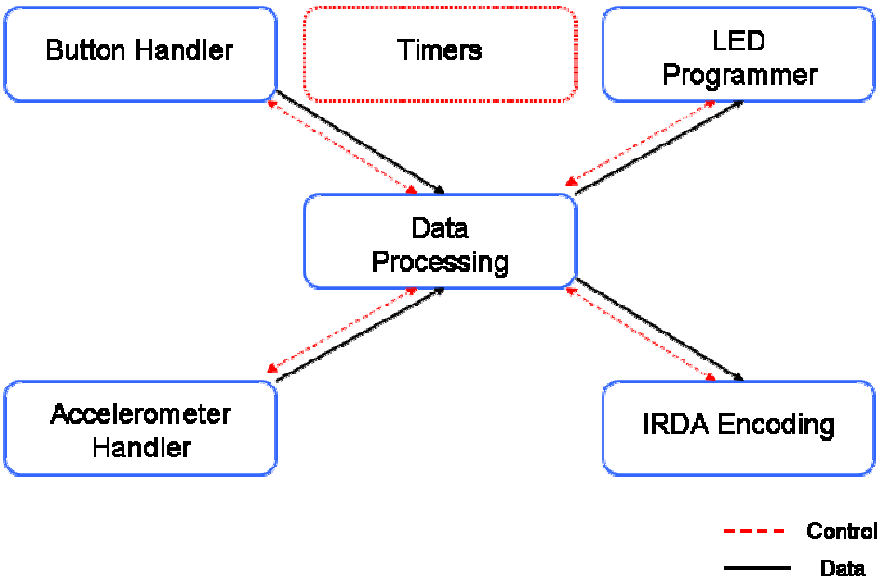

Figure 2. Software Block Diagram

Encoding and **Timers**. The control and data flow between components are drawn with dashed-red and solid-black lines. The timer block is represented differently, since it is not involved in the data flow, but it provides control for the other blocks.

Unknowns

- The decision of using a digital output accelerometer is motivated by the possibility of providing already to the microcontroller a digital signal instead of an analog to be converted. Nonetheless, the way to extract the direction information from the duty cycle of the accelerator output is unclear.
- The UART/Infrared Decoder implementation seems challenging and many aspects need further investigation.
- At the receiver side of the transmission two data need to be received: the function selected and the increase/decrease driven by the accelerometer). How to transmit them in a way that the receiver knows how to decode the symbols in the transmitted sequence? Do I need to specify a protocol?

Implementation Plan

After having purchasing the accelerometer component, in the first part of the project I would like to study how to retrieve and process its data. In parallel I could reuse and adapt concepts and code seen in class for the Button Handler, LED Programmer and Timers.

Later, a significant amount of time needs to be spent to investigate and implement the infrared transmission. Additionally the proper encoding of data in the modulated stream requires time.

Based on the difficulties encountered during the project implementation and the time available, the project could be changed in one of the following ways:

- Remove the Display component (and the LED Programmer) and visualize the transmitted data with an oscilloscope.
- Implement a receiver module with another ZiLog Microcontroller connected to the display, to decode and visualize the received commands.

Resources

- ZNEO™ Z16F Series Flash Microcontroller Contest Kit (given by University)
- Memsic 2125 Dual-Axis Accelerometer (to be purchased from www.parallax.com)
- Possibly oscilloscope or another ZNEO™ Z16F Series Flash Microcontroller Contest Kit