Lab 5: IrDA Communication Protocols

Revised March 13, 2017
This manual applies to Unit 5, Lab 5.

1   Introduction

The format of this lab differs from the rest in this series in that the hardware is not entirely specified. The student is intended to use an IR remote control device of his or her own choosing. For this reason, all background information is kept to the Unit 5 text. This lab is also the only lab that requires the use of the Analog Discovery 2.

2   Problem Statement

This lab requires the student to analyze the IrDA signal from an arbitrary device remote control unit and generate an application that can exchange IrDA messages with that device.

3   Lab 5

3.1   Requirements

1. Using the Analog Discovery 2, capture the IrDA signal sent from the remote control device and analyze the IrDA protocol.
2. Develop a PIC32 project that can decode the IrDA signal.
3. Display the decoded IrDA messages on the LCD and send to the workstation terminal.
4. Send a fixed IrDA message corresponding to keys "1" through "8".

3.2   Design

1. Analyze the IrDA remote control protocol.
    a.   Determine the carrier frequency as shown in Fig. 7.3 of Unit 5.
    b.   Determine the synchronization timing from a capture as shown in Fig. 7.5 of Unit 5.
    c.   Determine the bit timing for a logic 1 and logic 0 from a capture as shown in Fig. 7.6 of Unit 5.
    d.   Identify the encoding method by comparing to waveforms shown in Reference 3.
2. Generate a data flow diagram representing the tasks that you will use to decode an IrDA message and generate such a message.
3. Generate a control flow diagram that represents the control flow of decoding and encoding an IrDA message for your particular remote control device.

3.3 Construction Phase

1. Select a remote control device. Record the manufacturer's name, model number, and serial number along with the device the remote was made to control (if known).
2. Generate an MPLAB X project that includes files to configure the PIC32MX370 bits, the UART 4 at 19200 Baud (no parity), and the LCD.
3. Configure the Analog Discovery 2 to display the IR_TX (DIO 10) and IR_RX (DIO 11) signals and trigger on the negative edge of the IR_RX signal.
4. Copy the code from Listing 7.3 of the Unit 5 text into a new project file name *irda.c*.
5. Develop the necessary functions to capture and decode the IrDA input bit stream.
6. Develop a function that will send the IrDA code for keys "1" through "8." The key value is to be set using the Basys MX3 slide switches and the IrDA transmission initiated whenever BTNR is pressed.

3.4 Testing

1. Develop a table of IrDA key and control codes for all the keys on your IrDA remote control device.
2. Demonstrate that IrDA codes generated register on the device controlled by the IrDA remote.

4 Questions

1. What IrDA protocol does your IrDA control device use?
2. What is the carrier frequency?
3. How many different control bytes does your IrDA device generate?
4. What type of error checking does your device use?
5. What code is generated if a key on the remote control device is held depressed for 10 seconds?

5 References

1. Remote Control, https://en.wikipedia.org/wiki/Remote_control
2. Consumer IR, https://en.wikipedia.org/wiki/Consumer_IR
3. Data Formats for IR Remote Control, http://www.vishay.com/docs/80071/dataform.pdf
4. AN #157 Implementation of IR NEC Protocol, http://www.mcselec.com/index.php?option=com_content&task=view&id=223
5. Remote Control with IrDA® Transceivers, http://datasheet.octopart.com/TFBS4710-TT1-Vishay-datasheet-12514678.pdf
6. SB –Projects NEC Protocol, http://www.sbprojects.com/knowledge/ir/nec.php
7. PIC32MX330/350/370/430/450/470 Data Sheet, http://ww1.microchip.com/downloads/en/DeviceDoc/60001185E.pdf
8. Understanding Sony IR remote codes, LIRC files, *and the Arduino library,* http://www.righto.com/2010/03/understanding-sony-ir-remote-codes-lirc.html