CML Microcircuits
COMMUNICATION SEMICONDUCTORS

4th February 2013

(CMX7011 – Demonstrator 5)

Transmit and receive packet data using one PE0101-7011 evaluation kit

Introduction

This document is intended to assist with the evaluation and demonstration of the CMX7011 Digital Voice Processor and the Digital Voice Encryption Function Image™ 7011FI-1.x.x.x. It describes the use of the PE0002 Interface Card for easy connection to a computer via a USB cable and a single PE0101-7011 (CMX7011 evaluation kit) connected to the PE0002 C-BUS1 port.

General description

These scripts demonstrate the use of the packet data facility provided by the device. Messages of different lengths can be sent and received. Using digital audio editing software such as Audacity®, the transmitted data may be saved, and then played back into the device to recover the data packets.

Three scripts have been provided for this demonstration. They all transmit four data packets for being recorded, and recover the data when the recorded transmission is played back. The last data packet transmitted is a zero-byte packet which, when received, will cause the receive part of the script to finish. The difference between the scripts is in how the data are supplied, and the maximum length of the packets supported.

pkttx.pes – This script transmits data packets where the data are contained in arrays within the script. To change the data transmitted, the script must be edited. This script allows data packets of up to 127 bytes to be transmitted and received.

pkttxf.pes – This script transmits data packets where the data are contained in files. To change the data, the files must be edited. The script reads the data from three files and stores them in buffers at the start of the script. The data format is ASCII, hexadecimal, with one byte value per line. This script allows data packets of up to 127 bytes to be transmitted and received. If the files contain more data than this, the extra will be ignored.

pkttxfl.pes – This script is the same as pkttxf.pes except that data packets of up to the device maximum of 255 bytes are supported.

A fourth script has been included, setlevel.ps, which may be used to help set the analogue signal levels.

Key Items Required

- 1 off PE0002 Interface card
- 1 off PE0101-7011 evaluation board
- USB audio card with mic/line and speaker connections
- 5V power supply
- Digital audio editor software such as Audacity® (http://audacity.sourceforge.net)
- PE0002 GUI Software - ES000243.exe
- Scripts – setlevel.pes, pkttx.pes, pkttxf.pes, pkttxfl.pes
- Data files – dblock_a.ped, dblock_b.ped, dblock_c.ped
- Function Image™ 7011FI-1.x.x.x – (fi7011_1030.ped or later)

Connection Diagram

Demonstration Procedure

1. Make connections as shown in the diagram
2. Turn on power supply
3. Start the audio editing software
4. Start the PE0002 GUI software (ES000243.exe) and select the «Script Handler» tab
5. Set up sound card input signal level
    a. In the PE0002 GUI window, click the «Select Script» button, locate the setlevel.pes script file then click the «Open» button.
    b. Click «Run». This should instruct the device to produce an FFSK signal.
    c. In the audio editing software, check the level and record the FFSK signal. Adjust the signal input level for maximum amplitude without clipping.
    d. If a higher signal level is required from the device, each script has an AOGAIN constant that may be changed to increase the level (in 2dB steps). If the signal level from the device is too high with the AOGAIN constant set to 0 (which is the default for these scripts), then an external attenuator must be employed.
    e. In the PE0002 GUI window, click on «OK» to stop the FFSK signal.
6. Set up sound card output level
    a. Play back the final test recording made in step 5 and adjust the output level so the signal level is ~2Vp-p. If this level cannot be achieved, each script has an AIGAIN constant that may be changed to increase the effective level of the signal.
    b. The AIGAIN value represents 1.5dB gain steps. If the sound card will, for example, only deliver 1Vp-p, set the AIGAIN value to 4 (6dB)
7. Transmit packet data
    a. In the PE0002 GUI window, click the «Select Script» button, locate the pkttx.pes script file, then click the «Open» button.
    b. Click the «Run» button. The script will download the Function Image^{tm} to the device and configure the device for transmitting.
    c. The script will display «Start recording» in a dialogue box with «Yes», «No» and «Abort» buttons. Set the audio software to record and then press the «Yes» button.
    d. The device will then transmit 4 data packets. Once the transmission has finished, the script will display «Stop recording». Stop the audio software from recording and then press the «OK» button.

8. Receive packet data
    a. After step 7 has been completed, the device will be configured for receiving and be actively searching for a frame head.
    b. Set the audio software to play back the recording made in step 7
    c. As the FFSK recording is being played, the script will display the progress of decoding the data packets and their contents in the console.
    d. The last data packet which has a data length of 0 bytes will cause the script to finish.
9. Receive recorded packet data again
    a. In the PE0002 GUI window, click the «Select Script» button, locate the pkttx.pes script file, then click the «Open» button.
    b. Click the «Run» button. The script will download the Function Image[tm] to the device and configure the device for transmitting.
    c. The script will display «Start recording» in a dialogue box with «Yes», «No» and «Abort» buttons. Press the «No» button, this will skip over the transmitter section.
    d. Set the audio software to play back the recording made in step 7
    e. As the FFSK recording is being played, the script will display the progress of decoding the data packets and their contents in the console.
    f. The last data packet which has a data length of 0 bytes will cause the script to finish.

Steps 7, 8 and 9 can be followed with the other two scripts, pkttxf.pes and pktxfl.pes.

Example Audacity screen shot with all four recordings loaded