AVIX

AVIX Extended Demo Application
for the Microchip Explorer16
Development Board

AVIX-RT

Introduction
The 'AVIX Extended Demo Application for the Microchip Explorer16 Development Board' illustrates many advanced AVIX features together providing a highly modular application. The demo works 'out of the box' and can be used to become acquainted with AVIX. When new to AVIX it is advised to start by reading the AVIX documentation and work with the Tutorial application which is also part of the distribution.

How to run the demo and what does it show

Optionally connect UART2 to a PC running HyperTerminal or a comparable terminal emulator using the settings as shown in the figure. Connect the board to a development PC running MPLAB8x using a Microchip Real-ICE or ICD3 debugger.

Open the following project in MPLAB8x: (<root> is the AVIX install directory you have chosen during installation).

<root>_dev_board_demos_demo1_Explorer16_ExtendedDemo_mplab8x_project_files\extDemo.mcw

Compile the application and run in the debugger. Control the application behaviour as indicated in the figure.

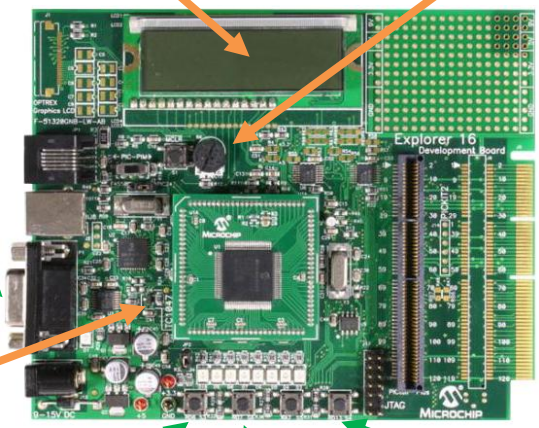
Connect to PC running Hyper-Terminal using settings:
- **115200** Baud
- **8** bit data
- **1** stop-bit
- **no** parity
- **no** hardware handshaking

The application transmits active channel values to the PC.

Using Hyperterminal also allows controlling the application. Typing:
- **'U'** *Increases sample rate*
- **'D'** *Decreases sample rate*
- **'C'** *Toggles active channels*

Display channel states, channel values and sample period.

Pot meter is connected to analog input 5. Application reads this input and displays on LCD and transmits on UART2 labeled as CH.2



The demo is available for:

- PIC24EP512GU810
- PIC24FJ128GA010
- PIC24FJ256GB210
- PIC24HJ256GP610(A)
- dsPIC33EP512MU810
- dsPIC33FJ256GP710(A)
- PIC32MX360F512L

Temperature Sensor is connected to analog input 4. Application reads this input and displays on LCD and transmits on UART2 labeled as CH.1

Switch S3: *Press to in-crease sample rate from 2ms to 10ms*

Switch S6: *Press to de-crease sample rate from 10ms to 2ms*

Switch S4: *Press to toggle the active channels between both, 1, 2 and none.*

Demo structure

This demo consists of four highly autonomous software components responsible for sampling the ADC (temperature sensor and pot meter), reading the switches, showing the sampled values on the LCD and sending the sampled values to a serial link (UART2). The high level of modularity is obtained by using the AVIX Exchange mechanism, a mechanism especially developed to obtain independent software components. This mechanism makes it easy to test individual components, extend your application without making changes to existing code and re-use components in other applications.

A detailed description of the demo application structure can be found here: (<root> is the AVIX install directory you have chosen during installation).

<root>_AVIX\Doc\AVIX_ExchangeMechanism.pdf

When using this demo, it is strongly advised to have this document at hand.

Demo Application Files

Document AVIX_ExchangeMechanism.pdf presents four software components that are 'connected' to each other using two Exchange Objects. Every component and Exchange object is coded in its own file. For initialization and supporting functionality a number of additional files are present:

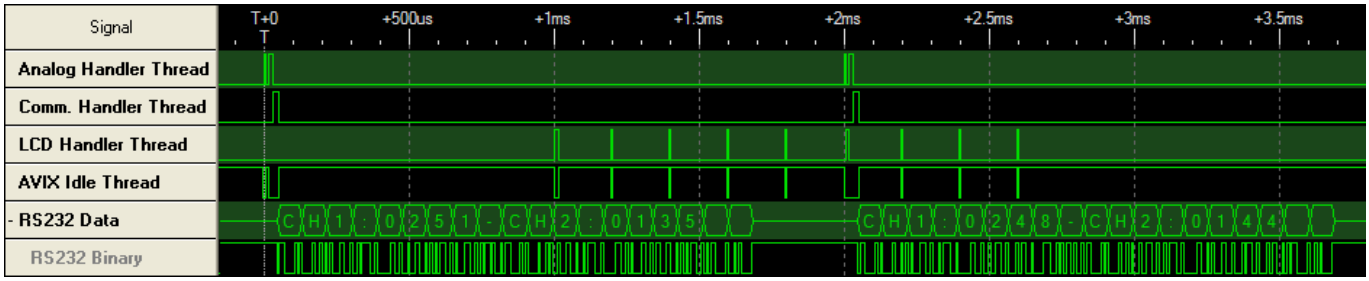
File	Description
extDemoMain.c	Contains function avixMain, the AVIX main function used for initialization. In this function the Exchange objects are created and the initialization function of the four major components are called.
compAdcSampling.c	Contains all functionality of software component Analog Handler.
compUart.c	Contains all functionality of software component Communication Handler.
compLcd.c	Contains all functionality of software component LCD Handler.
compSwitchHandling.c	Contains all functionality of software component Switch Handler.
controlExchange.c	Contains Control Exchange supporting functionality. The Control Exchange holds the active channels and the sample period. For changing both values the components do not access the Exchange directly but through support functions that are present in this file. Also creating and initializing the Control Exchange is present in this file.
adcExchange.c	Contains the functionality to create and initialize the Analog Exchange.
support.c	Supporting function to convert an integer value to a string.

Background

The figure below shows a snapshot of the activity of three of the application threads and the idle thread using AVIX Thread Activation Tracing. Below that the RS232 data on the connection to the PC is shown. Illustrated is the Analog Handler Thread waking up every 2ms. It becomes active for a little while to initiate the AD conversion. Once this is done, it wakes up again to process the result. Since the analog data is written to the Exchange Object observed by the Communication Handler this wakes up next, converts the data to the format send over the serial line and is done. The serial data starts to be send out immediately, a process further taken care of using interrupt handling. The snapshot also shows the LCD Handler Thread becoming active to write some characters to the LCD. Since this Handler Thread has a lower priority it only uses processing power when available. The AVIX Idle Thread finally shows when the application has nothing do, which is quite a substantial amount of time for this application.

Based on the figure it shall also be clear why the shortest sample period is chosen to be 2ms. Sending all data using an 115200 baud connection has to fit in this period.

Finally the last two characters in the serial data do not show any content. These are a CR and a LF character and the representation of these just did not fit in the Logic Analyzer display, which is why these character positions are blank.



When using an AVIX Free Demo distribution, the application will only work for a maximum time of 30 minutes. After this time the application will stop working. This is not an error but a mechanism deliberately built into a Free Demo distribution. Furthermore a Free Demo distribution has limitations on the maximum number of kernel objects that can be created. For the applicable values see the Port Guide in <root>_AVIX\Doc or the AVIX-RT website (www.avix-rt.com).