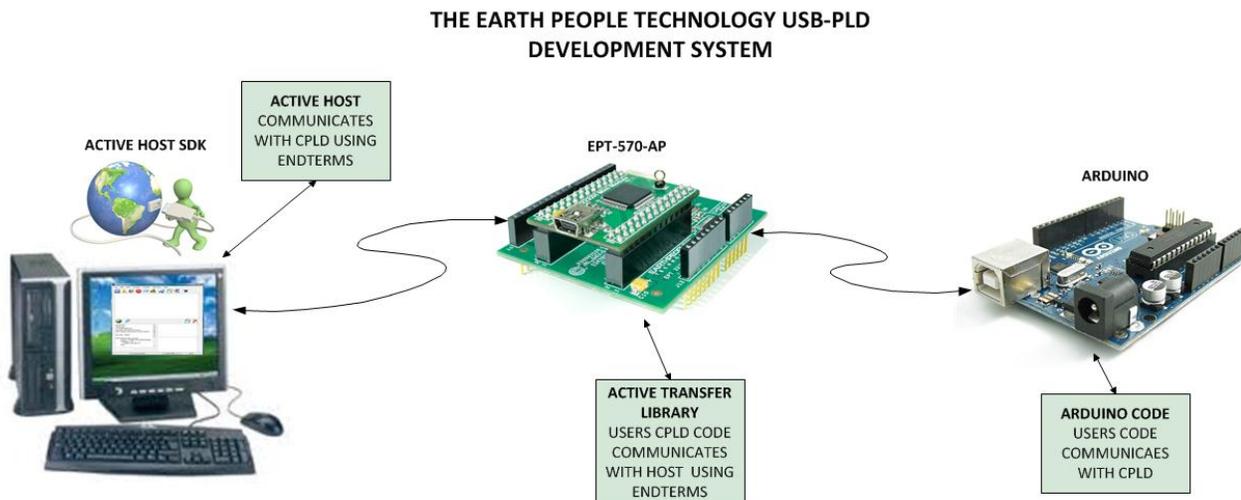


## USB-CPLD DEVELOPMENT SYSTEM FOR THE ARDUINO

The EPT USB-CPLD development system provides an innovative method of developing and debugging programmable logic code. It also provides a high speed data transfer mechanism between an Arduino board and a host PC. The EPT USB-CPLD development system provides a convenient, user-friendly work flow by connecting seamlessly with Altera's Quartus II software. The user will develop the code in the Quartus environment on a Windows Personal Computer. The programmable logic code is loaded into the CPLD using only the Quartus Programmer tool and a standard USB cable. The Active Host SDK provides a highly configurable communications interface between Arduino and host. It connects transparently with the Active Transfer Library in the CPLD code. This Active Host/Active Transfer combination eliminates the complexity of designing a USB communication system. No scheduling USB transfers, USB driver interface or inf file changes are needed. The EPT USB-CPLD development system is a unique combination of hardware and software.

### The System Overview

The Earth People Technology USB-CPLD development system hardware consists of a High Speed (480 Mb/s) USB to parallel (8 bit) bus chip and a CPLD. The USB interface provides both JTAG programming of the CPLD and a High Speed transfer path. The software consists of the Active Host SDK for the PC. The firmware includes the Active Transfer Library which is used in the CPLD to provide advanced functions for control and data transfer to/from the Arduino.



The user's Arduino code is developed to perform particular functions required by the user (such as reading a temperature sensor). The code is downloaded to the microcontroller using the Arduino IDE system provided as part of the microcontroller development system. The EPT USB-CPLD Development System allows users to write HDL code (either Verilog or VHDL) that will implement any digital logic circuit. The user's HDL code is compiled and synthesized and packaged into a programming file. The programming file is programmed into the CPLD using the JTAG channel of the USB to Serial chip, the FT2232H. The Active Host SDK contains a dll which maintains device connection, polling, writes and includes a unique receive mechanism that automatically transfers data from EPT-570-AP when data is ready. It also alerts the user code when the dll has stored the transfer and the data is available to the software GUI (graphical user interface). Users do not need to interface with the USB Host Driver or any Windows drivers. They need only to include the Active Host dll in their projects. The Active Transfer Libraries must be included in the CPLD project to take advantage of the configurability of the Active Host SDK. All of the drivers, libraries, and project source code are available at [www.earthpeopletechnology.com](http://www.earthpeopletechnology.com).

## Hardware Features

The EPT USB-CPLD Development System consists of a CPLD base board onto which is connected the FT2232H USB to Serial board. These boards comprise the hardware components of the EPT USB-CPLD development system. The EPT- 570-AP board is equipped with an Altera EPM570 CPLD; which is programmed using the Altera Quartus II software. The CPLD has 570 Logic Elements which is equivalent to 440 Macrocells.

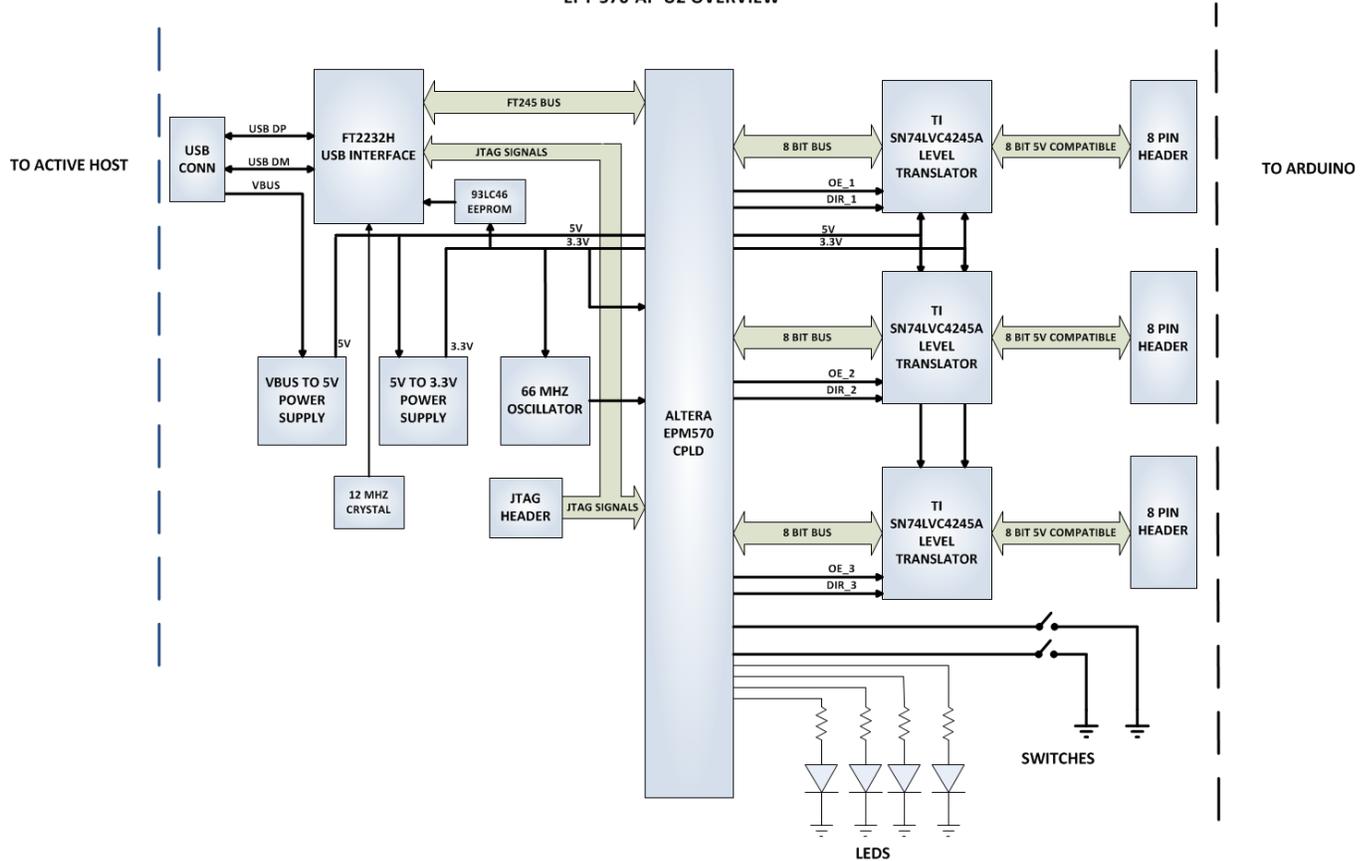


An on board 66 MHz oscillator is used by the EPT Active Transfer Library to provide data transfer rates of 8 Mega Bytes per second. Twenty Four I/O's from the CPLD are attached to three 8 bit transceivers to provide 5 Volt compatible I/O's. These 74LVC245 bidirectional voltage translator/bus transceivers are controlled by one enable and

direction bit per transceiver. This means the direction of the individual bits of each transceiver cannot be selected; the direction is selected for all eight bits per transceiver. There are four green LED's and two Push Buttons that are controllable by the user code. The hardware features are as follows.

- Altera EPM570 with 440 Macrocells
- 66 MHz oscillator for driving USB data transfers and users code
- Three bidirectional voltage translator/bus transceivers
- 24 user Input/Outputs available as three 8 bit ports, 5 Volt compatible
- Four Green LED's accessible by the user
- Two PCB switches accessible by the user
- All I/O connectors stack into the Arduino Uno
- Two 13x2 connectors for accepting the FT2232H Mini Module.

EPT-570-AP-U2 OVERVIEW



## The Software

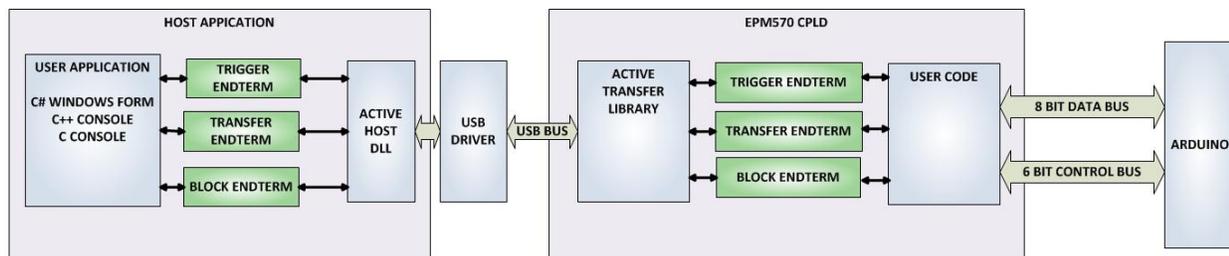
The Active Host SDK is provided as a dll which easily interfaces to application software written in C#, C++ or C. It runs on the PC and provides transparent connection from PC application code through the USB driver to the user CPLD code. The user code connects

## USB-CPLD Development System for the Arduino

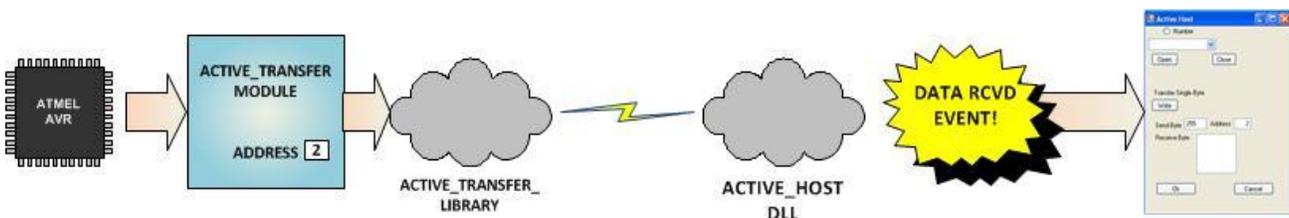
to “Endterms” in the Active Host dll. These Host “Endterms” have complementary HDL “Endterms” in the Active Transfer Library. Users have seamless bi-directional communications at their disposal in the form of:

- Trigger Endterm
- Transfer Endterm
- Block Endterm

User code writes to the Endterms as function calls. Just include the address of the individual module (there are eight individually addressable modules of each Endterm). Immediately after writing to the selected Endterm, the value is received at the HDL Endterm in the CPLD. The Trigger Endterms are used as “switches”. The user code can set a Trigger bit in the CPLD and cause an event to occur. The Transfer Endterm sends one byte to the CPLD. The Block Endterm sends a block of bytes. By using one of the Active Host Endterms, the user can create a dynamic, bi-directional, and configurable data transfer design.



The Active Host SDK is designed to transfer data from the CPLD when it becomes available. The data will be stored into local memory of the PC, and an event will be triggered to inform the user code that data is available from the addressed module of the CPLD. This method of automatically moving data from the user code Endterm in the CPLD makes the data transfer transparent.



The data seamlessly appears in Host PC memory from the Arduino. The user code will direct the data to a control such as a textbox on a Windows Form. The transparent receive transfer path is made possible by a Callback mechanism in the Active Host dll. The dll calls a registered callback function in the user code. The user code callback can be designed to generate any number of events to handle the received data.

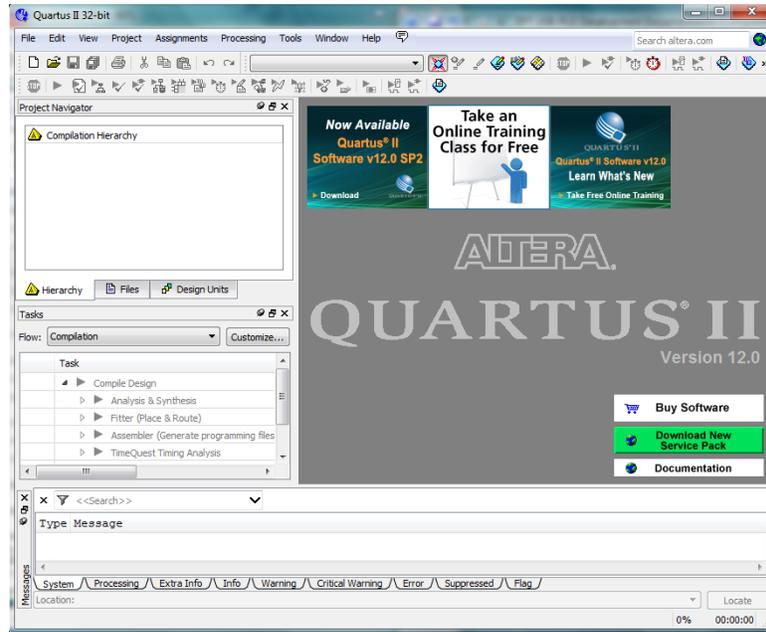
## CPLD Code Development Environment



The user will build the CPLD project using Quartus II software from Altera. The top level file defining the user's CPLD project connects the user code, the Active Transfer Library, and the Endterms together. In order to route the pins of the Arduino to the CPLD, the Quartus Pin Planner tool is used. This tool allows the user to match internal net names to the pins of the CPLD. The user will compile, synthesize, and program the CPLD project from Quartus.

## Project Creation and Compiling

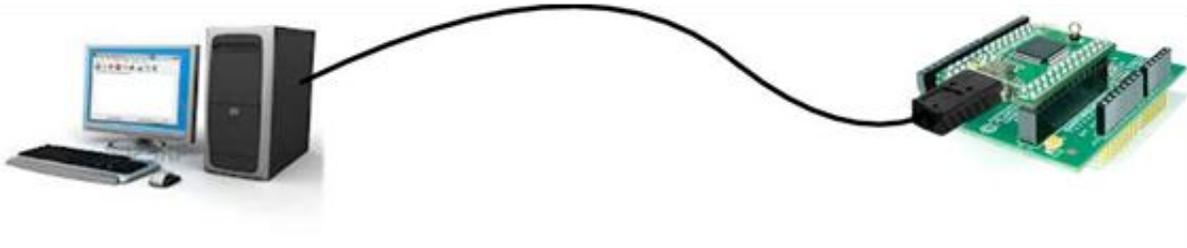
Once the HDL code (Verilog or VHDL) is written and verified using a simulator, a project can be created using Quartus II. Earth People Technology supports the latest version of Quartus II. The EPT-570-AP has a detailed user manual which will walk the user step by step to create a project.



Once the project files are complete, just hit the Compile and Synthesize button. Quartus will output a programming file ready for download into the EPT-50-AP.

## Programming the CPLD

The CPLD on the EPT-570-AP can be programmed with the project created by the user. Programming the CPLD is quick and easy. All that is required is a standard USB cable with a Mini Type B connector, and the EPT Blaster Driver DLL installed on the PC. There are no extra parts to buy - just plug in the USB cable and connect the EPT-570-AP to the PC.



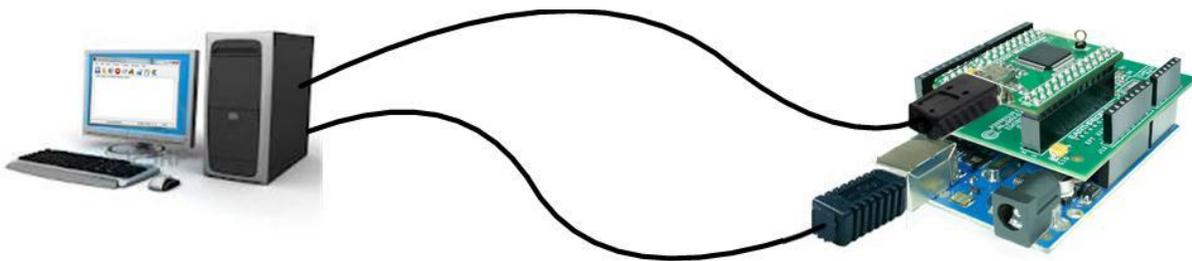
Open the programmer tool, and click the Start button. When the tool is finished, the EPT-570-AP is ready for use with the Arduino.

## USB-CPLD Development System for the Arduino

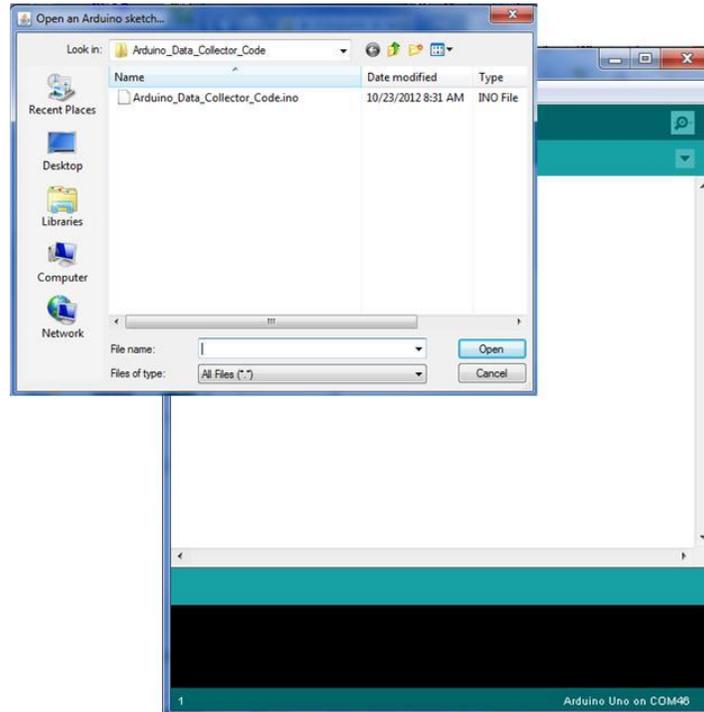


### Connecting the Arduino and EPT-570-AP Together

To complete the project, just snap the Arduino and the EPT-570-AP together, and connect each to the PC. The user can create an Arduino project and load it into the flash using the



Arduino IDE. The project can take advantage of the CPLD's large number of Macrocells. Since the Arduino is limited with internal SRAM, it can use the CPLD for storage. The Arduino is processing constrained with only 16 MIPS. The CPLD can be used to perform redundant processing tasks which will free up the Arduino for other tasks. For High Speed data transfer, the Arduino can take advantage of the Active Host dll.

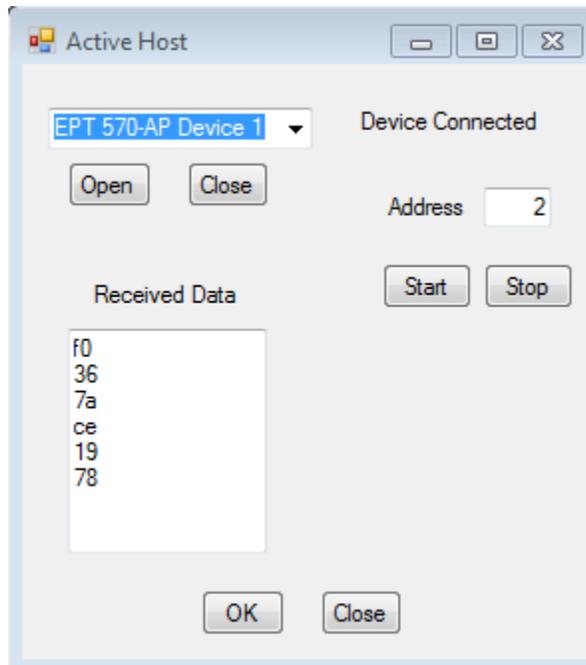


The user will load the sketch into the Arduino IDE window. Compile and Download the sketch into the Arduino microcontroller using the Upload button.

## Testing the Project

Testing the project exercises all of the components. One of the difficult aspects of testing embedded electronics is comparing the timing of signals inside the chips and searching for bugs. The Active Host SDK makes troubleshooting more dynamic. The Trigger Endterms can be used to notify the Host PC when selected signals change state in the Arduino or CPLD. The Transfer Endterms can be used to transfer the value of an internal bus while it is operating, much like a logic analyzer.

## USB-CPLD Development System for the Arduino



The Active Host dll allows the user to write Host PC applications in C#, C++ or C language. The user manual includes easy step-by-step directions to guide the user to create a powerful Windows application in C# with a minimal amount of code writing. There are several projects the user can use as a base to create much larger projects. You can easily make a volt meter, data logger, or more advanced projects like multiple motor controllers with feedback loops. The possibilities are endless.