

Abstract

High Speed Universal Serial Bus (USB) 2.0 is a recently developed transfer protocol robust enough for most data acquisition applications, making it an ideal platform for sensor to PC interface. While USB sensor interface systems are available for many sensor types such as audio, economical data acquisition boards are not currently available for video.

Existing USB cameras are not open systems and are composed of hardware and software designs tailored only for their single specific application. It is the goal of this project to create a USB 2.0 interface that can be generalized for various custom video sensors, providing a platform that can easily be adapted for multiple sensor technologies.

The design approach for the platform divides the implementation into three primary elements—camera, DAC microcontroller, and PC host. The camera provides the system data, including the pixel information and synchronization signals necessary to produce video. This data is collected and packaged by an FPGA microcontroller board, which is responsible for converting the raw digital data into a USB 2.0 data stream. This stream is routed into the PC host, which coordinates the system operation and displays the data via a GUI.

This development of a USB video interface allows various camera sensors to be attached. The data acquired from the camera sensors is relayed and properly displayed according to the camera type. The generic capability of the USB system is demonstrated via the implementation of two different grayscale custom video sensors. The result is a robust platform that leverages the utility of USB 2.0 to provide an efficient and accessible means of interfacing custom video sensors with personal computers.

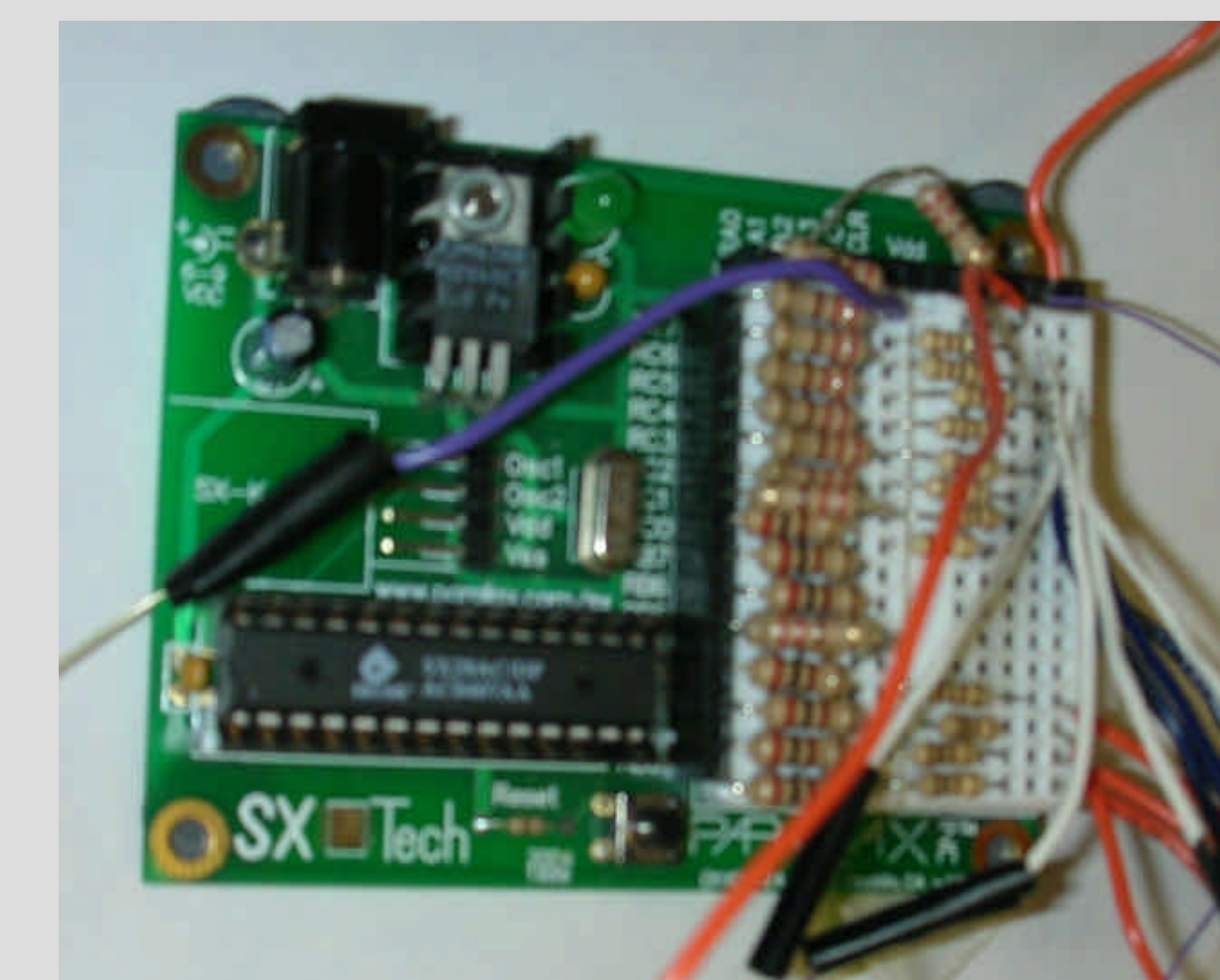
Generic USB 2.0 Custom Video Sensor Platform

Camera

The system is designed to be compatible with any grayscale camera operating at up to 5 MHz with screen resolution up to 512 x 384. Digital camera output conforms to a standard format, providing a parallel array of raw data bits demarcated by pixel and frame synchronization signals. These sync signals are responsible for indicating when new data has arrived and when an entire frame of pixel data has been transmitted. This standardized format allows for the generic interface of custom digital cameras with the USB 2.0 platform hardware.

A simulation of this standardized camera output was generated via a test microcontroller, which produced a predictable eight-bit counter pixel data output coupled with a synchronized frame sync. This simulation was used to verify the capabilities of the platform prior to the connection of the camera demonstration applications.

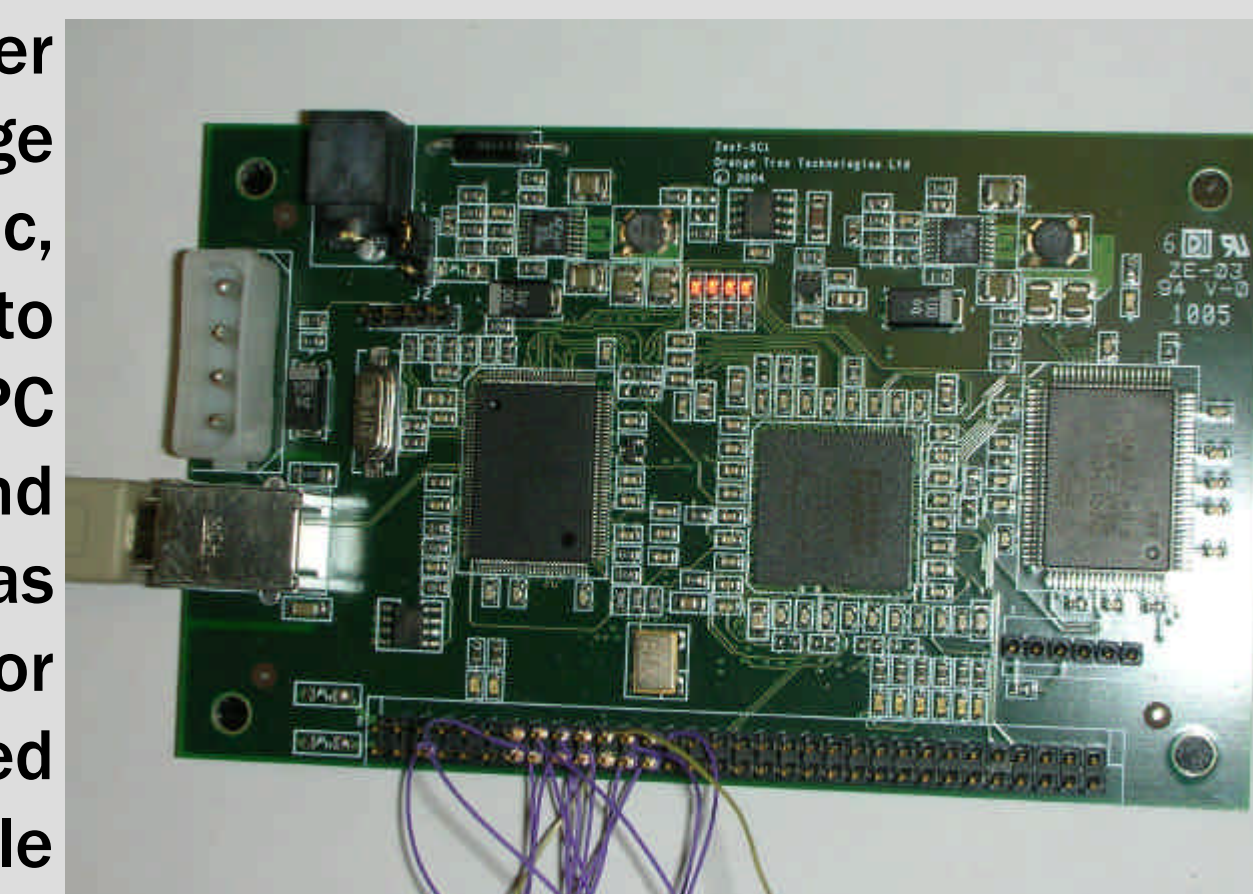
Two actual camera applications were implemented to demonstrate the generalization capabilities of the board. Each of these applications are custom grayscale sensors that adhere to the standardized method of video output, providing an output interface that can be wired directly to the I/O ports of the platform's FPGA hardware.



Microcontroller Used for External Counter Test Application

FPGA

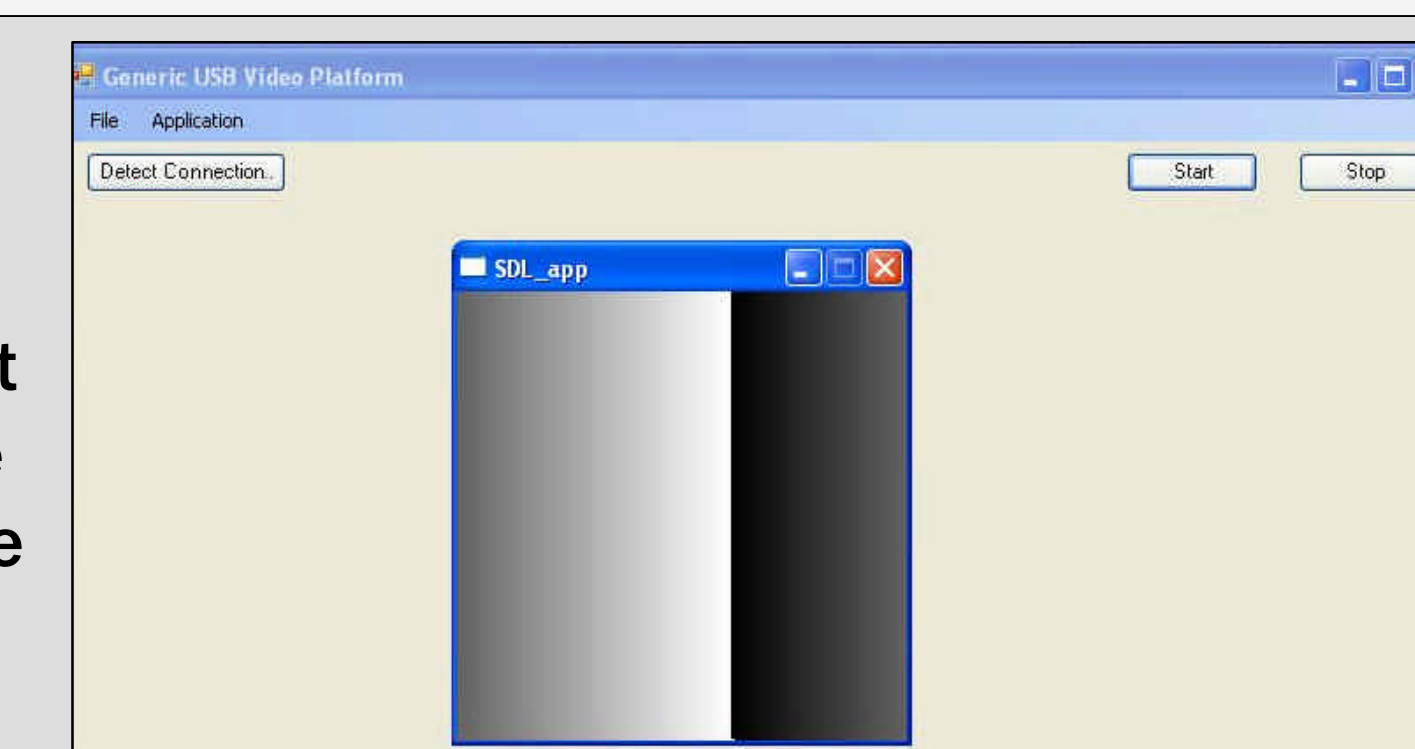
The FPGA is tasked with obtaining the pixel data from the camera unit and sending the information to the PC via USB 2.0. Both the pixel sync and frame sync trigger the FPGA to collect and send data. On the falling edge of the pixel sync, or the rising edge of the frame sync, the FPGA turns on the streaming mode of the FPGA to send data to the PC. The data that is sent to the PC consists of the 8 pixel data bits, the frame sync, and the FPGA's internal error counter. After the data has been sent, the streaming mode is shut off and the error counter is incremented. The error counter is designed to ensure that if any single pixel is transmitted multiple times the PC is aware that it is not a new pixel and thus will disregard it. The FPGA operates in a constant loop which is not dependant on the status of the PC.



Orange Tree ZestSC1-400 FPGA

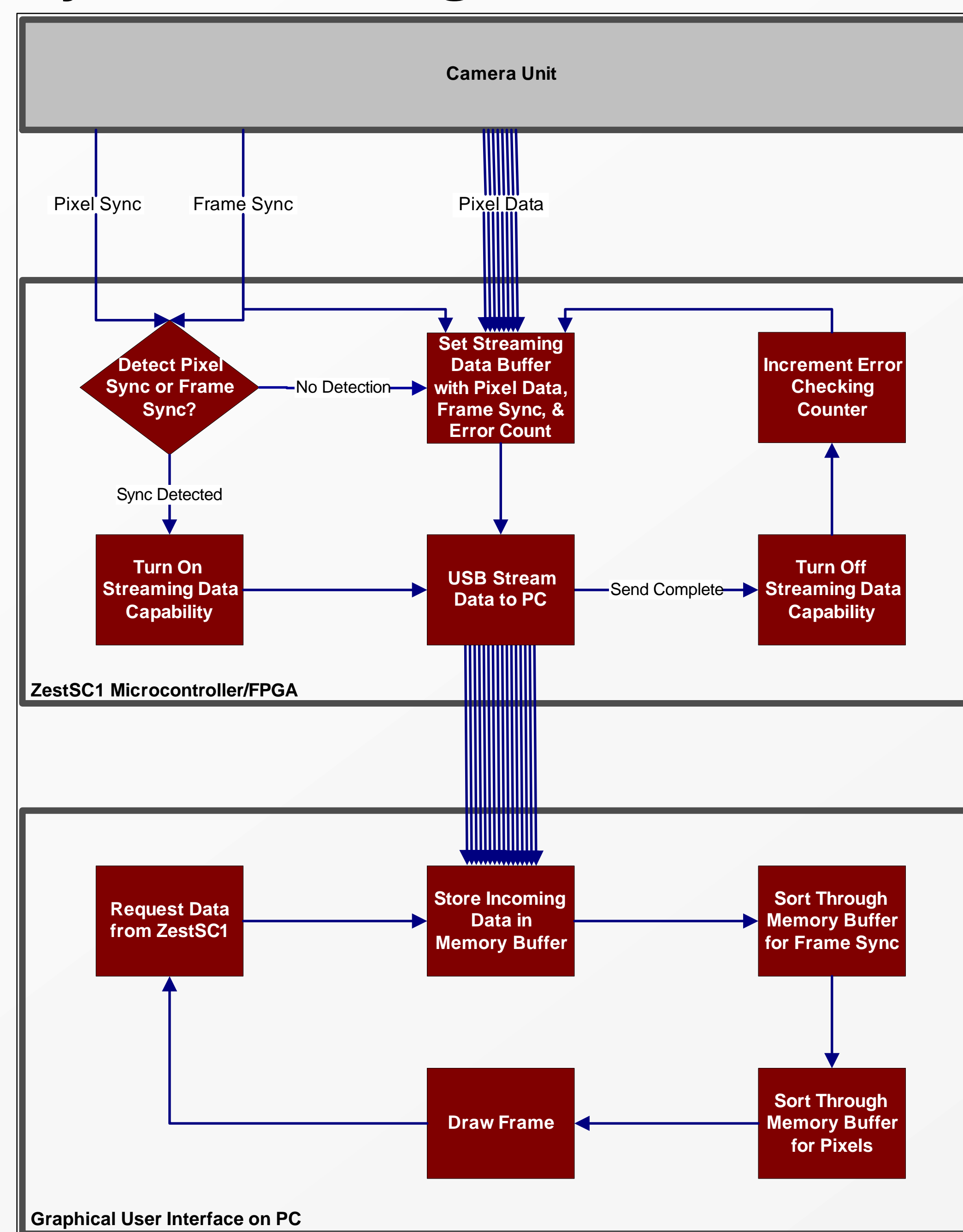
PC Host

The PC Host is controlled by a GUI built upon the Microsoft .NET framework. The GUI coordinates the operation of the system by establishing communication with the FPGA and programming it according to the selected camera input. When the system is active, the PC uses the ZestSC1 interface support functions to read data via the streaming USB 2.0 interface into an internal memory buffer. This internal memory buffer is analyzed to detect the transmitted frame and pixel syncs, which are used to parse the individual frame and pixel boundaries necessary to compose the raw data into displayable video. The SDL.NET video support package is then used to access the display drivers and print to the screen the parsed data as an individual video frame. The GUI has been designed to be both modular and scalable, allowing for almost instantaneous addition of camera applications.



Screen Capture of GUI Running External Counter Test Application

System Flow Diagram



Group Members:

#14

Mark Dweck
Louie Huang
Daniel Koch

Advisors:

Dr. Jan Van der Spiegel
Dr. Viktor Gruev

Demo Times:

10:00 AM – 12:00 PM
April 18, 2006