Abstract - LogiScope is based on the concept of a Logical Analyzer. A Logic Analyzer is an instrument which is electronic in nature, which captures and displays signals from a digital circuit. After which the Logic Analyzer will convert the captured data into various timing diagrams and graphs, etc. according to the data input from the logical or digital circuit. The Logic Analyzers have advanced triggering capabilities, which are useful to a user when he needs to see the timing and frequency relationships between many signals in a digital system. After the data is captured from the digital circuit, the results can be displayed in several ways, from showing the waveforms or state forms and even graphs to any complex form. The digital circuit is connected to any device running an Android OS or a computer with the help of the LogiScope Board via a USB connector. The logic probes on the LogiScope Board can be connected to various points on the digital circuit to get the input and display it in the form of a graph. When the input is fetched from the circuit, the LogiScope Application on the device will output the waveform in the form of a graph. Thus, this application will act as a portable circuit debugger, which will help in debugging any complex circuit with the help of a device having the LogiScope Application and the LogiScope Board.

Keywords - Android, ATmega, Graph, Logic Analyzer, USB

I. INTRODUCTION

Earlier and even today cathode ray oscilloscope (CRO) is a very useful and versatile laboratory instrument used to study waveforms of alternating currents (AC) and voltages as well as for measurement of current, frequency; in fact, almost any quantity that involves amplitude and waveform. A CRO will allow any user to see the amplitude of electronic signals as a function of time on the screen. It is widely used for troubleshooting and laboratory work involving research and design. It is also, at times employed for studying the waveforms of a signal with respect to amplitude distortion and its deviation from the normal or even to debug a circuit, if there are any wrong or faulty connections.

Therefore, a cathode ray oscilloscope has proved to be one of the most important tools in the design and development of modern electronic circuits.

It has always been an active part of a laboratory, to test the waveforms of a particular circuit and thereby understanding it’s working.

Now the use of such a device becomes a bit old-fashioned. The CRO is not at all portable and it needs high voltage to operate. It is bulky in nature and a bit too complex to configure. LogiScope aims at building an application which will work with any device that is compatible, thus capable of doing somewhat the same job as done by the traditional CRO.

Many logical analyzers available nowadays are portable, but are expensive and they force you to use the Operating System that they are built on. So, we wanted to have platform independency and multiple device compatibility.

This system needs any compatible Android device or a computer, which will be connected to the LogiScope board via USB. Logic probes can be connected to the board, which in turn are connected to the digital circuit to debug. There can be 8 or 16 channels for debugging. The ports can be either set as input or output using the LogiScope application.

We decided to use Android as the base platform for the application, as today each and every individual holds a smart phone. Thus we made an Android Application in which the users can set the Input or Output depending on their needs to get a graphical representation the same way as a logic analyzer [7, 9].

If there is any problem with the circuit, i.e. if you are not getting your desired output due to some fault, like a faulty IC, burnt or broken pins of an IC, wrong circuit connections, etc.

Therefore, the main focus of this system is that you don’t have to go in a laboratory for doing all these proceedings, but you can simply get your Android Application working and start debugging the circuit with your LogiScope Board.

II. LITERATURE SURVEY

A. Methodology

The flow of LogiScope is conveyed with the help of a diagram shown below in Figure I.
As one can observe the diagram, we can see that there is a Microcontroller, which is a part of the hardware, controlling the whole functional behavior.

The Input/Output controller on microcontroller takes the Input from the ports via logical probes connected to the circuit being tested. The Microcontroller [5] takes the Input from the Input/Output Controller and transfers the data to the USB 2.0 Host IC module situated on-board. The USB module will transfer the data to the device connected to it. The device should have the LogiScope application running which will take the input from the USB device module and display the output accordingly on the screen.

B. Execution of LogiScope Board

The LogiScope board is a customized board which is designed according to the needs of this system. The working of the LogiScope Board with its basic pin connections is shown in Figure 2. The Board consists of ATmega32U4 [15] IC which is used to interface with other various components of the Board.
After the USB initialization, the IC ATmega32U4 [15] will wait for the Start packet. When it gets the packet, it will set the I/O Pins accordingly, i.e. if its Input or Output.

After setting the I/O Pins the System will initialize the timer, say for example 10ms to 100ms. If the Timer overflows, it will reset the Timer and send the respective packets.

An Interrupt will only occur when a packet is received, at this time it will set the I/O Pins according to the data in the packet. For the channels that have been set as output, the microcontroller will set the channel to HIGH or LOW [3] accordingly.

Thus, this procedure will go on till all the packets have been transferred or till the application sends a stop packet. When this happens the connection will be terminated and the microcontroller will reset. Further, we will explain about the connections of ATmega32U4 [15] with all the other components of the LogiScope Board and how they communicate with each other.

C. Working of Android Application

As you can see in the flow diagram below, when the system initializes it will first check whether the accessory i.e. the LogiScope board is connected or not [6]. If it’s not connected it will wait till the board is connected.

When the Board is connected and start button is pressed the application will send a data packet which will consist of a header and data which will initiate the transmission.

There are two modes in use that is the input and output mode. When the channels are in input mode the application will only display the current state of the channels. If the user sets any of the channel in output mode then a toggle will come up by which the user can set a high/low value at that channel.

If the values of the toggle pins are changed, it will send the current state of the Input/Output Pins to the LogiScope Board.

When the packet is received by the LogiScope Board, it will set all the Input/Output Pins and according to the packet received.

This will go on till the stop button is pressed by the user, and it will send the stop packet to the LogiScope board and shut the application.

D. ATmega32U4

LogiScope Board uses the Atmel AVR [15] 8-bit low power microcontroller, ATmega32U4. It comprises of all the basic features and parameters which are required to run this system.
The ATmega32U4 microcontroller is interfaced with the other components on LogiScope board to get the system running. There are a total of 26 Input/Output pins on the microcontroller for communication of which 8 Input/Output ports are used in this system. These 8 pins/ports are connected to the digital circuit being tested with the help of logic probes connected between the circuit and the LogiScope Board. The Input/Output pins are secured by the protection circuit which protects the pins from over-voltage of input.

The over-voltage protection circuit deals with the input from the digital circuit being tested. The microcontroller can accept input voltages in the range of 0V to 0.7V to indicate a Logic Low input (Binary 0) and 2.5 to 5.3V to indicate a Logic High input (Binary 1).

The circuit in Figure V shows the layout for protection of over-voltage from the input so that the microcontroller pins do not get affected by the high voltage input.

The circuit in Figure V is connected to the digital circuit on one side and to the I/O pin of the microcontroller on the other side. The flow of the voltage is marked in shades of red and shades of green.

The red flow of the voltage indicates sourcing current i.e. from Ground (GND) to the microcontroller pins and the green flow of voltage indicates the sinking of the current from the digital circuit being tested, to the Ground (GND).

The input from the digital circuit to the microcontroller pins should be in the range of 2.5V to 5.3V which means that any voltage above these ranges may not be accepted and may even damage the microcontroller pins. Therefore, we use the concept of an over-voltage protection circuit.

This kind of a circuit consists of a zener diode connected to the Ground and connected to a 510Ω resistor. If the incoming voltage is higher than 5.3V, the zener diode will sink the excess current and pass the required amount of current to the microcontroller pins which will in turn protect the pins from getting damaged.

If the input voltage is very less the excess voltage will be provided by Ground and thereby will source the voltage.

E. Communication Protocol

The Accessory Development Kit (ADK) [13, 14] is a reference implementation which we have used as a starting point for building our accessory for the Android Application. Each ADK protocol released by Google is provided with source code and hardware specifications which we used to design our accessory.
Our accessory uses the Android Open Accessory (AOA) protocol [12] to communicate with the Android device, via USB cable. We have used the AOA protocol to establish communication between the microcontroller and the Android Application.

We are using the Android Accessory Development Kit (ADK) for 2012 which is the latest implementation of an AOA device. The ADK 2012 hardware specifications are based on the Arduino open source electronics prototyping platform. We have made some hardware and software extensions that allows the Application to communicate with Android device.

There are a few experiments which were performed to test the speed of data transfer between the Android device and the LogiScope board. The data transmission was done for a varied amount of data, ranging from 1,000 bytes to 2,00,000 bytes in which different data speeds were recorded.

![Data Size vs Speed Graph]

**Figure VI. Data Size vs Speed of Transfer**

The different data speeds (bytes per second) with respect to the number of data transferred (bytes) are shown with the help of a graph in Figure VI.

As we can see that the speed of transfer of data fluctuates with the increase in the number of data size which is transmitted. For example, if we want to transmit 1,000 bytes of data then the speed at which the data will get transmitted is 3484 Bytes per second. Similarly for 50,000 bytes the speed is 3474 Bytes per second and so on.

III. USER INTERFACE

A. Channel Layout

The channel can be either an 8-channel or a 16-channel layout. In this scenario we have taken an 8-channel layout in which the 8 channels can act as an Input or an Output mode depending on the setup.

![Channel Layout Diagram]

**Figure VII. Channel Layout in Android**

Figure VII shows an example of the channel layout as Input or Output mode. From this example we can observe that 4 channels are set in an Input mode while the other 4 channels are set in an Output mode. It is not necessary to assign the first 4 channels as Input or Output or assign the last 4 channels as Input or Output; rather any of the 8 channels can be assigned any mode depending on the requirement of the digital test circuit.
The channel layout in the above scenarios has been assigned 4 channels as Input (Channel 1 to Channel 4) and 4 channels as Output (Channel 5 to Channel 8), which can be toggled. The input will come from the digital circuit as a Logic Low or a Logic High and depending upon the input the 4 output channels will show the output of the logical operations performed, which can either be a Logic High or a Logic Low.

B. Graph Representation

The graph will consist of an x and y axis. X-axis will depict the time frame of the incoming digital wave input. The Y-axis will depict the range from 0 to 1 i.e. low or high respectively.

IV. Conclusion

The LogiScope system presents a new dimension to portable, low-powered and cost-effective logic analyzers. The proposed technique uses ready ADK protocol for an errorless continuous communication. The low power components and by displaying on an Android device makes the system cost effective.

The system is compatible with multiple platforms thus making it easier to use and cross compatible. The system has effective safety mechanism as it is not damaged due to over-voltage.

As of now we have only made an Android application. In the near future we will also make an application that is compatible with windows and does the same function as the Android application. We have been working to make the windows application. We can also add support by making an application which will be compatible with Mac OS, Linux and IOS. We can increase the number of input/output channels so as to debug larger circuits. We can add other functionalities to our applications like UART, I2C bus debugging. We can increase the refresh rate of the system by using an ARM microcontroller.

REFERENCES


701


