

BLUETOOTH EMBEDDED PORTABLE OSCILLOSCOPE

H.M.D.B. Seneviratne¹ and K.N. Abhayasinghe²

¹ Department of Electronic and Computer Engineering, SriLanka Institute of Information Technology (SLIIT), SriLanka. Email: seneviratne.dushan@gmail.com

² Department of Electronic and Computer Engineering, SriLanka Institute of Information Technology (SLIIT), SriLanka. Email: nimsiri.a@sliit.lk

ABSTRACT

Portable oscilloscopes currently in the market are very expensive, less power efficient and have small low resolution displays. This paper presents the design and implementation of a low cost, portable, light-weight; low power, dual-channel oscilloscope, consisting of a hardware device and a software application. The device is equipped with a Bluetooth module to provide connectivity to a device with Bluetooth, running the Android operating system (OS), in order to display the waveforms. Android OS is selected because there are a decent number of Android device users and most of these devices satisfy the requirements of the oscilloscope's software application. The hardware device includes circuitry to capture the input voltage signals and an embedded Bluetooth module for transmitting the captured signal information to an Android device for displaying the waveform. The Software application developed for Android receives the data transmitted from the hardware device and plots the waveform according to the display settings configured by the user. These display configurations are transmitted to the hardware device once they are set by the user, and are used by the hardware device to set the sampling rate and the values of samples. For optimal use of the available bandwidth, the application provides two modes of operation, namely single channel mode where only channel 1 is operational and dual channel mode where both channels 1 and 2 are operational. The user can select a mode from the application, which in turn sends a message to the microcontroller which then changes the sampling frequency accordingly: a higher sampling rate for single channel and half of that for dual channel.

Key words: Android, Bluetooth, Oscilloscope

1. INTRODUCTION

This paper presents an oscilloscope implemented using new technology, with low power consumption. It is a Bluetooth embedded device which will capture input voltage signals and transmit them to an external device, such as a smart-phone, running the Android operating system. Since the device is wireless it can make use of the smartphone's display and processing power.

The implementation of an oscilloscope with Bluetooth was previously reported, by Yus in 2010 [1]. It is an open source prototype project called the "Android Bluetooth Oscilloscope", which consisted of a Bluetooth enabled transmitter circuit to send data to an Android phone which draws the waveforms on its screen. The transmitter circuit uses Microchip's dsPIC33FJ16GS04 and an LMX9838 Bluetooth 2.0 SPP module. The maximum input voltage to the circuit is +8 V to -8 V. However, there is no mention about the bandwidth of the device. Furthermore, it is stated that the application had been tested only with a Samsung Galaxy GT-i5700 Spica (rooted Android 2.1 OS) phone.

2. METHODOLOGY

2.1. System Overview

The design and implementation stage of the project, involved the Bluetooth embedded hardware device implementation and the software application development for Android. The Bluetooth embedded device is a microcontroller based system. Figure 1 gives a block diagram of the overall system undertaken in this project and Figure 2 shows a block diagram of the Bluetooth embedded device.



Figure 1: Block diagram of the Overall system



Figure 2: Block diagram of the Bluetooth Embedded device

As in figure 2, the input signal is directly

connected to the front end circuit. The maximum voltage range allowed into the front end circuit is $33 V_{p.p.}$. Voltages above this will be regulated at the stated maximum. If higher voltages need to be input, a probe with a simple resistive divider or a high voltage probe can be used. The protection mechanisms used are a Zener diode AC regulator and Schottky voltage clamp protection.

2.2. Bluetooth Data Rate Limitations

From research carried out it was found that data rates of 2 Mbps [2] are not achievable with the existing software stacks implemented on the module's controller. Therefore, the approach suggested to fully utilize the Bluetooth bandwidth, was to use the module in Host Controller Interface (HCI) mode [2].

The standard mode refers to the on-board stack running on the Bluetooth module. In HCI mode, the on-board stack is bypassed and the module is put in a state that runs the Bluetooth baseband. Therefore, the Bluetooth stack is not on the module, so it needs to be implemented on the interfacing host processor. So the module is used as a radio, performing the lower level MAC functionalities, while the application stack runs on the host processor. The hardware interface between the host processor and the module can be either UART or USB. The advantage of HCI mode is that it allows to obtain maximum throughput and also to implement custom profiles on the Bluetooth module.

2.3. Issues with HCI Mode

However practical issues exist due to the lack of resources, on how HCI mode can be implemented. It is required to be a registered developer at the Bluetooth module's chip manufacturer, and have documentation and support on how to access the HCI layer. Implementing the Bluetooth stack is a complex process which requires a powerful, high-end microcontroller with complex firmware and hardware design. The Android device communicates in the RFCOMM layer [3] using the SPP profile, but in HCI mode it will have to work in the lower baseband layers. The support currently available for such an approach in Android is inadequate.

2.4. Alternative Approach

A widely used, more reliable Bluetooth module, RN42 by Roving Networks was selected. The module was used in normal mode due to the high level of complexity involved in HCI mode. With

the standard firmware having the Bluetooth stack on board (Serial Port Profile), a throughput of 240 kbps [2] can be achieved in slave mode.

2.5. Basic Specifications of the Bluetooth Embedded Device

- Number of input channels: 2
- Bandwidth: 2.5 kHz (dual channel mode) and 5 kHz (single channel mode)
- Input voltage range: -16.5 V to +16.5 V with 1:1 probe
- Sampling frequency: 10 kHz (dual channel mode) and 20 kHz (single channel mode)

2.6. Android Application

The Android application is developed based on the software development kit (SDK) of Android 2.2 Froyo OS, API (Application Program interface) level 8.

The Android device used for testing the application is a Samsung Galaxy Y (GT-S5360) smartphone with Android 2.3.5 Gingerbread OS, an 832 MHz processor and 290 MB of RAM. The smartphone has a 3.0" QVGA 240x320 LCD screen with Bluetooth v3.0 + HS (High Speed). It is one of the low cost Android smartphones available in the market to-date. Figure 4 shows the layout of the application on the phone's screen.

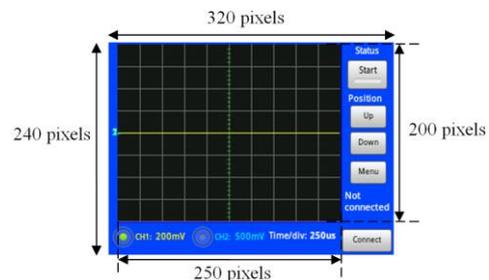


Figure 4: Layout of the Oscilloscope Android Application

The screen measures 240 by 320 pixels and, the waveform grid area for drawing waveforms is 200 by 250 pixels. The rest of the screen is used by the various buttons as shown in Figure 4.

The waveform grid area contains 10 horizontal divisions and 8 vertical divisions, similar to a standard oscilloscope. Hence, each division in both the vertical (voltage) axis and horizontal (time) axis consists of 25 pixels (200 px / 8 and 250 px / 10). The ground (0 V) line is the horizontal line at the centre of the waveform grid. The waveform plotting is done as defined by the canvas class in the android graphics package,

where the drawing is done with reference to pixels, since it is the smallest element on the screen. As the voltage axis consists of 200 pixels, 200 voltage levels can be represented on the grid. Hence 8-bits are capable of representing these 200 levels. Similarly the time axis has 250 pixels so 250 samples of one input channel can be displayed on the grid.

A sample from the dsPIC's ADC is 10-bits which represents 1024 voltage levels. Because the phone's waveform grid has only 200 voltage levels, the 1024 levels should be compressed into this range. Because dsPIC's ADC cannot handle negative voltages, if a 3.3 V_{p,p} sinusoidal signal shown in Figure 5 (a), needs to be sampled, it has to be first level shifted by a dc offset of 1.65 V, so that it is within 0 V and 3.3 V as in Figure 5 (b).

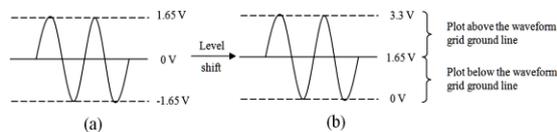


Figure 5: (a) 3.3 V_{p,p} sinusoidal signal (b) Level shifted 3.3 V_{p,p} sinusoidal signal

When the signal is plotted on the waveform grid it should look like Figure 5 (a), centered on zero. Therefore, as in Figure 5 (b), the voltages greater than 1.65 V should be above the ground line in the waveform grid and those less than 1.65 V to be below the ground line. Essentially, this process is what is used to convert the 10-bit sample to 8-bits to represent the 200 levels on the screen. Figure 6 shows this processing algorithm;

```

Voltage = ADC reading x ADC resolution
if ADC reading < 512
    Voltage = ADC midvoltage - Voltage
    New voltage = (Voltage x Pixels per division) / Volts per division
    Transmit value = Display ground point + New voltage
    if Transmit value > 230
        Transmit value = 230
else
    Voltage = Voltage - ADC midvoltage
    New voltage = (Voltage x Pixels per division) / Volts per division
    if New voltage >= Display ground point
        Transmit value = 0;
    else
        Transmit value = Display ground point - New voltage
    
```

Figure 6: Processing Algorithm

2.7. Modes of Operation

The device provides two modes of operation, as in Figure 9, namely single channel mode where only channel 1 is operational, with the full bandwidth available for channel 1 and, dual channel mode where both channels are operational, with the bandwidth shared between the two channels. The sampling frequency in single channel mode is 20 kHz which is double the sampling frequency per channel of 10 kHz in dual channel mode. The user can select a mode

from the application, which in turn sends a message to the dsPIC to notify the selected mode, so it can change the sampling frequency. This helps to make efficient use of the available bandwidth when only one channel is required.

By default, the hardware is in dual channel mode, so both input channels are sampled simultaneously. After sampling and conversion, Channel 1 (CH1) and Channel 2 (CH2) samples are converted to a byte and are immediately written to the dsPIC's UART buffer for transmission via Bluetooth. Figure 7 shows the sequence in which the samples are written to the UART buffer.

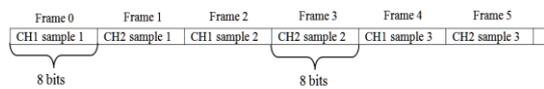


Figure 7: Data transmission sequence in dual channel mode

The even numbered and odd numbered frames contain channel 1 and channel 2 data, respectively. There is no accumulation of samples as each sample is sent soon after it is processed. In single channel mode the frames contain samples of channel 1 only.

2.8. Waveform Plotting Algorithm

The transmission format in Figure 7 also makes the reception of data easier. The application reads the data stream from its Bluetooth data buffer and separates out the channel 1 and channel 2 data into two arrays. Reading this continuous data stream with the un-rooted Android OS and synchronizing with the transmission rate was the major challenge in developing the application. Thereafter, the waveform of the particular channel is plotted on the grid by joining the data points (samples) together with straight lines.

2.9. Android Application User Interface

Figure 8 shows a screen shot of the application's user interface.

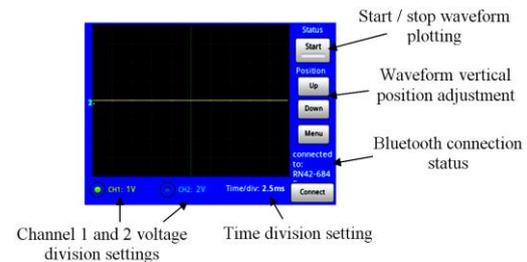


Figure 8: Main interface

Figure 9 shows the Menu screen of the application on Menu button press on the main

interface. Here, the user can change the settings as desired, such as the voltage and time division settings, the channel to be displayed on the grid and the operating mode.

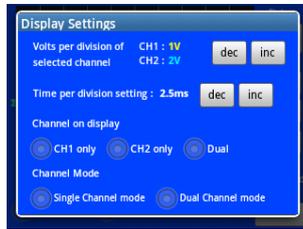


Figure 9: Menu interface

When the user does a change in the volts-per-division setting from the android application, it sends a message to the dsPIC notifying the change. The dsPIC then adjusts the values of the samples to be transmitted based on the current setting in order to plot the waveform correctly.

3. RESULTS

The operation of the device was tested by providing test signals from the signal generator and dc voltages from the dc power supply. Both modes of operation of the device were tested with different voltages and frequencies applied to the channels. Figure 10, 11 and 12 show screen shots of the application displaying the waveforms of three test conditions.

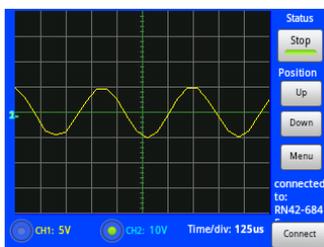


Figure 10: 5 V_p sinusoidal signal at 2.24 kHz (single channel mode)

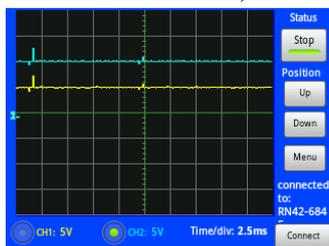


Figure 11: DC input, channel 1: 5 V, channel 2: 10 V (dual channel mode)

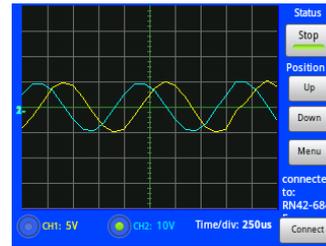


Figure 12: Channel 1: 5 V_p sinusoidal signal, channel 2: 10 V_p sinusoidal signal at 1 kHz (dual channel mode)

4. CONCLUSION AND FUTURE WORK

The paper presented a Bluetooth embedded oscilloscope that selects optimum sampling rate and attenuation/gain according to the setting selected. This design has the advantage that, the Android application can be upgraded to provide more features without doing changes to hardware, which helps to improve the device standards and user experience.

Bluetooth 3.0+HS which has a much higher data rate of 24 Mbps can be used instead of v2.1+EDR. It can improve the bandwidth of the device. Still similar issues related to the effective data rate may arise due to the limitations in communication profiles and software stacks used. Wi-Fi (802.11) could be the other option, to achieve higher data transmission speeds and increase bandwidth.

The prototype contains majorly through-hole components. Surface mount components could be used to reduce the size of the PCBs and make the device more portable.

5. REFERENCES

[1] Yus. (2010, Sep. 23). “Android Bluetooth Oscilloscope” [Online]. Available: <http://projectproto.blogspot.com/2010/09/android-bluetooth-oscilloscope.html> [Feb 24, 2012].

[2] Roving Networks. (2012, Sep.). “RN-42/RN-42-N Class 2 Bluetooth Module” [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/rn-42-ds-v2.2r.pdf> [March 4, 2013].

[3] Android Developers, “BluetoothSocket”, Mar. 5, 2013. [Online]. Available: <http://developer.android.com/reference/android/bluetooth/BluetoothSocket.html>. [Mar. 6, 2013].