International Journal of Research in Advent Technology, Special Issue, RAECE-2K19
E-ISSN: 2321-9637
Available online at www.ijrat.org

Design of an Efficient and Low Cost Wireless Oscilloscope on Android Smart Devices

[1]P.A.Harsha Vardhini , [2]A.Krishna Veni
[1,2]Department of ECE
[1]VITS, Deshmukhi.Telangana
[2] MRCEW, Medchal, Telangana

Abstract—Smartphone is one that electronic gadget that everyone carry with them. This work is intended to develop an application for a Android device as a replacement to the conventional cathode ray oscilloscope (CRO). This work illustrates the design and development of oscilloscope on smart devices like Phones, Tablets..etc., which not only provides the feature of portability but also reduces the equipment cost. This makes use of the core java coding and XML for developing the desired application by running the total code in the eclipse environment. This work includes - digitize the analog samples, Data transmission through Wi-Fi, Android application development to display the signal, scaling the signal.

Keywords—Oscilloscope, Android, PIC16F72, Wi-fi, ADC

1. INTRODUCTION

Generally the conventional oscilloscope as shown in Fig.1 has many features like voltage amplification, two channel operations, time division, triggering and many other features. But a conventional CRO utilization is limited because of the following limitations- Portability is the main limitation as it is not easy to carry everywhere. Secondly, the equipment cost high around Rs. 25,000 and much higher with improved features. Proposed work overcomes these limitations of portability and high cost. The features of oscilloscope are intended to be developed on the android platform.

Fig.1 Conventional Oscilloscope

Bluetooth Oscilloscope transmits the data through Bluetooth. Since Bluetooth has a very slow transmission rate it is not that useful. But the programming which was used for Bluetooth connectivity to the display unit can be implemented for future reference. So we took that code as a reference code for the development of our actual application. The Bluetooth device is named as Bluetooth oscilloscope.[1]

Osciprime is a device is a bit advanced than the Bluetooth oscilloscope. It gave many other features which are almost present in a conventional oscilloscope. But thiswas not useful because the transmission was done through USB in main. But there were advanced features than the Bluetooth oscilloscope. This device was named as osciprime.[2]

This work deals with displaying waveforms by an application developed using Android OS and helps to reduce the cost of lab equipments, increase accuracy and precision in obtaining, recording and storing the waveforms.

2. DESIGN PROCEDURE-WIRELESS OSCILLOSCOPE

Implementation of Wireless Oscilloscope is performed with the following procedure [3,4]

1. Digitize the analog samples.
2. Data transmission through Wi-Fi.
3. Application development to display the signal.
4. Scaling the signal.

For the hardware development, a single ADC chip with serial communication to the Wi-Fi module is enough. Basically an ADC and a serial port are required and a microcontroller to be chosen which is suitable for the implementation. PIC16F72 microcontroller is chosen and the datasheet of the pic microcontroller [8] is verified for suitability of required standards and specifications.

According to the literature survey done, as transmission can be done through Bluetooth but with less data transmission rate when compared to any other transmission media, hence Wi-Fi transmission is preferred as it transmits data at a faster rate.

This work enhances the multi-stream applications that are to use the open source ANDRIOD platform and built embedded applications using some communication concepts As the proposed oscilloscope overcomes limitation of portability and high cost, and as it became inevitable that every person carry a smart device, wireless oscilloscope on smart phone/tablet is of great use. This development concern reduction of cost of lab equipments, increase accuracy and precision in

obtaining, recording and storing the waveforms.

The first part of the project is to convert the analog input which is from the signal source to digital signal. This is necessary because the display unit which is used in the project is able to receive only digitized data. There are many devices available which have in-built ADC. But the other requirement was to transmit the data to the Wi-Fi module. For this purpose the microcontroller is preferred. Simple ADC can also be used but there is a problem of transmitting
the data from ADC to the Wi-Fi module, hence a PIC microcontroller is chosen in order to send the data through its serial port.

Next step is the development of an application to receive the data that is obtained from the Wi-Fi module. This work illustrates the application developed on a android platform is user friendly and also an open source

3. IMPLEMENTATION- HARDWARE & SOFTWARE

Proposed Wireless Oscilloscope consists of two modules, the hardware module consists of conversion of analog signal to digital signal and the software module is an application to display the signal on the screen This deals with development of an Android application, i.e. the application source code is written in the XML file and is run in the eclipse IDE which is a platform that runs the android applications.

A. Hardware

The device used for the analog to digital conversion part of the work is the PIC microcontroller which is named as PIC16F72. This is used because it has inbuilt analog to digital convertor and also the serial port to transfer data to Wi-Fi module.

Fig.2 shows the implemented hardware, input signal is given to the ADC of the microcontroller and by converting the analog signal to digital signal and routing the sampled digital values to the tablet through Wi-Fi. So a micro controller is used which performs the action of analog to digital conversion and also serial transmission of digital data to a Wi-Fi module. Wi-Fi module transmits the data to the SmartPhone/ Tablet through a router. Analog to digital conversion takes place through a clock speed of R_{osc} i.e. the variable clock pulse. Then this digital pulses are transferred from Wi-Fi module to the display device where the application retrieves the obtained values and canvasses the graph and this transfer is done through a router which has the address of the tablet and the Wi-Fi module used stored in it.

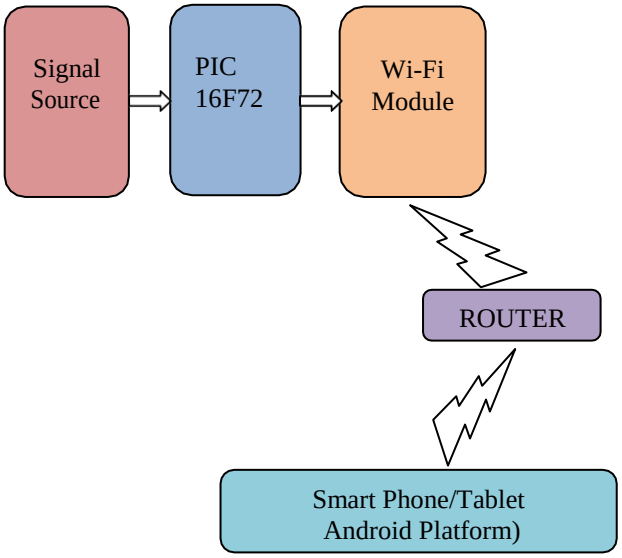

Fig.2. Wireless Oscilloscope on Android Platfrom

B. Software Flow Diagram

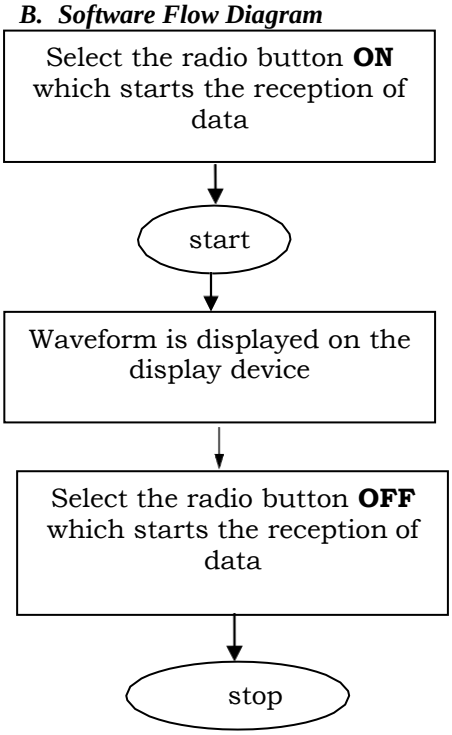

Fig.3. Software flow chart

This is the basic front end of the software and the back end of the application is explained from the following.

C. Application Development

A central feature of Android is that one application can make use of elements of other applications(provided those applications permit it). For example, if your application needs to display a scrolling list of images and another application has developed a suitable scroller and made it available to others, you can call upon that scroller to do the work, rather than develop your own. For this to work, the system must be able to start an application process when any part of it is needed and instantiate the Java objects for that part.

Therefore, unlike applications on most other systems Android applications don't have a single entry point for everything in the application. There are 4 types of components

- Activities
- Services
- Broadcast receivers

Content providers

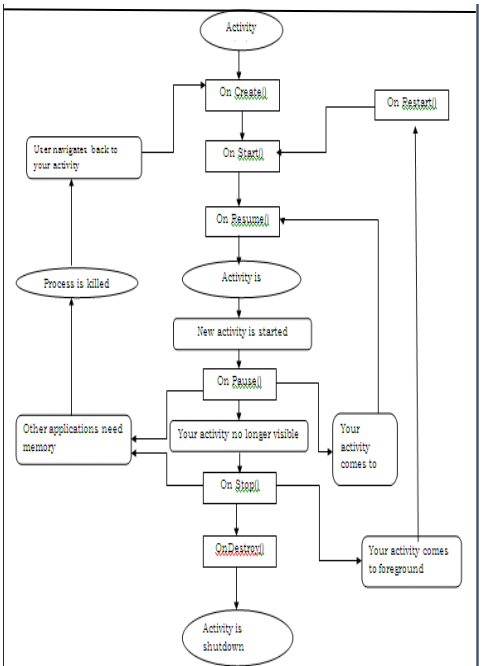

Fig.4. Activity life cycle(source http://developer.android.com)

ANDROID JAVA CODE

```
package android.harsha.WiFiHome;
// to import the required files import android.content.Context; import android.graphics.Bitmap;
import android.graphics.Bitmap.Config; import android.graphics.Canvas;
import android.graphics.Color; import android.graphics.Paint; import android.util.AttributeSet; import
android.view.View;
public class GraphView extends View
{
 private Bitmap mBitmap;
// to create the canvassing of the incoming data private Canvas mCanvas = new Canvas(); private int mColor;
 private float mLastValue; private float mLastX;
 private Paint mPaint = new Paint(); private float mScale;
 private float mSpeed = 1.0F;
private float mWidth; private float mYOffset;
private float maxValue = 1024.0F;
public GraphView(Context paramContext)
{
super(paramContext); init();
}
public GraphView(Context paramContext, AttributeSet paramAttributeSet)
{
//calling the variables from super class super(paramContext, paramAttributeSet); init();
 }
 private void init()
 {
//to give the dimensions of the display where graph appears
  this.mColor = Color.argb(200, 255, 0, 0); this.mPaint.setFlags(1);
 }
 public void addDataPoint(float paramFloat)
 {
//to draw the graph of incoming values by taking one at a time
  Paint localPaint = this.mPaint;
  float f1 = this.mLastX + this.mSpeed;
  float f2 = this.mYOffset + paramFloat * this.mScale; localPaint.setColor(this.mColor);
  this.mCanvas.drawLine(this.mLastX, this.mLastValue, f1, f2, localPaint);
  this.mLastValue = f2; this.mLastX +=this.mSpeed; invalidate();
 }
 protected void onDraw(Canvas paramCanvas)
 {
  try
  {
   if (this.mBitmap != null)
   {
    if (this.mLastX >= this.mWidth)
    {
     this.mLastX = 0.0F;
     Canvas localCanvas = this.mCanvas; localCanvas.drawColor(-1); this.mPaint.setColor(-8947849);
     localCanvas.drawLine(0.0F, this.mYOffset, this.mWidth, this.mYOffset, this.mPaint);
    }
paramCanvas.drawBitmap(this.mBitmap, 0.0F, 0.0F, null);
   }
   return;
  }
  finally
  {
  }
 }
 protected void onSizeChanged(int paramInt1, int paramInt2, int paramInt3, int paramInt4)
 {
  this.mBitmap = Bitmap.createBitmap(paramInt1, paramInt2, Bitmap.Config.RGB_565);
  this.mCanvas.setBitmap(this.mBitmap); this.mCanvas.drawColor(-1); this.mYOffset = paramInt2;
```

```
this.mScale = (-(this.mYOffset * (1.0F /this.maxValue)));
this.mWidth = paramInt1; this.mLastX = this.mWidth;
super.onSizeChanged(paramInt1, paramInt2, paramInt3, paramInt4);
}
public void setMaxValue(int paramInt)
{
this.maxValue = paramInt;
this.mScale = (-(this.mYOffset * (1.0F /this.maxValue)));
}
public void setSpeed(float paramFloat)
{
this.mSpeed = paramFloat;}}
```

MICROCONTROLLER ADC CODE

```
#include <16F72.h> #include <lcd.h>
#use delay(clock=20000000) //crystall Oscillator speed
#use rs232(baud=9600, xmit=PIN_B0,rcv=PIN_B1)
//UART Serial Interface with WiFi Module void main(void)
{
int my_adc_value;
//Microcontroller Health Check indicator output_high(PIN_C4); //LED ON delay_ms(1000); //wait
for 1 second output_low(PIN_C4); //LED OFF delay_ms(1000);
output_high(PIN_C4); delay_ms(1000); output_low(PIN_C4);

while(1)
{
my_adc_value = read_adc(0); //read adc value from channel 0
putc(my_adc_value); //send data to WiFi module over
UART
output_toggle(PIN_C4); //Led status indicator
}
}
```

4. CONCLUSION

Smartphone is effectively implemented as a wireless oscilloscope that can be carried everywhere. For checking various outputs this work provides the benefit of portability at a very low cost. Application for a Android device as a replacement to the conventional oscilloscope. Digitization of the analog samples, Data transmission through Wi-Fi, Android application development to display the signal is implemented. Further code can be extended for the enhancement of oscilloscope features like scaling of signals can be done.

REFERENCES

[1] www.projectproto.blogspot.com
[2] www.osciprime.com
[3] code.google.com/p/oscdroid
[4] www.developer.android.com
[5] www.gabitronics.com
[6] www.projectproto.blogspot.com
[7] www.microchip.com.