Wireless DSO

by
Ananda Kundu (18307R025) anandakundu13@gmail.com
Aswin P Ajayan (183079032) aswinp2008@gmail.com
Sunny Mehta (183079023) sunnymehta78669@gmail.com

July 12, 2019

1 Abstract

The project aims at creating wireless DSO modules for measuring differential voltage signals with high common mode signals, observing signal variations and waveforms and monitoring them from a remote location. We implemented a circuit to get the differential output without inputs getting overloaded by common mode signal. For this purpose, common mode signal is attenuated to avoid the saturation of output, both the inputs are separately attenuated and get the differential signal. But recognizing in the process of attenuating the common mode signal, differential signal is also getting attenuated, and so need to be amplified the differential signal.

2 Introduction

The measurements usually made using normal oscilloscope probes are single ended, that is the low line of the probe is connected to ground and the high line is connected to test node. If it is needed to make differential measurement, then both the channels of scope were used and make single ended measurements of two test points with respect to ground and by subtraction operation the difference signal is measured. Though with this technique the purpose is solved, it needs two CRO channels to make a single differential measurement. With a differential probe, the differential measurement can be made with the help of a single channel. Another advantage of using differential probe is it can measure differential output with very high common mode without saturating. The differential signal is then converted to fit the ADC range (0-3.3 V) using the circuit described later. This signal is then given to the ADC port of TIVA C launchpad which has an inbuilt ADC with 10-bit resolution. The data from ADC is then sent to the PC via Wi-Fi using CC3100. The data received by the PC is used to plot the signal in GUI. In GUI we had included three sliders to control voltage scaling, time scaling and level triggering.

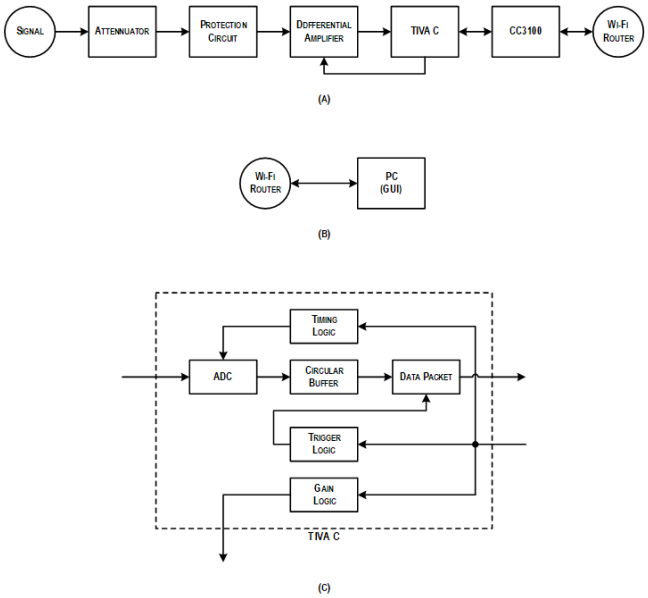

Figure 1: Block Diagram: a,b) hardware blocks c)logical blocks

3 Circuit Description

3.1 Reference Generator

The circuit comprises of op-amp and POT as shown in schematic. As we are using a single ended power supply (0-5 V), this is required to generate a ground reference for the amplification stage. We used POT instead of resistors so that we can calibrate the output for zero.

3.2 Attenuator

The attenuator is used in the input side so that signals with larger peak to peak values can be plotted. It will increase the dynamic range of our system.

3.3 Protection circuit

A diode protection circuit is implemented as shown in the schematic; its purpose is to prevent the input of the amplifier stage going beyond the supply voltages. This protects the op-amp circuitry from any over voltages.

3.4 Differential amplifier with variable gain

We have used instrumentation amplifier over general differential amplifier, as it has better CMRR. Further, we have used INA118 as it has inbuilt overvoltage

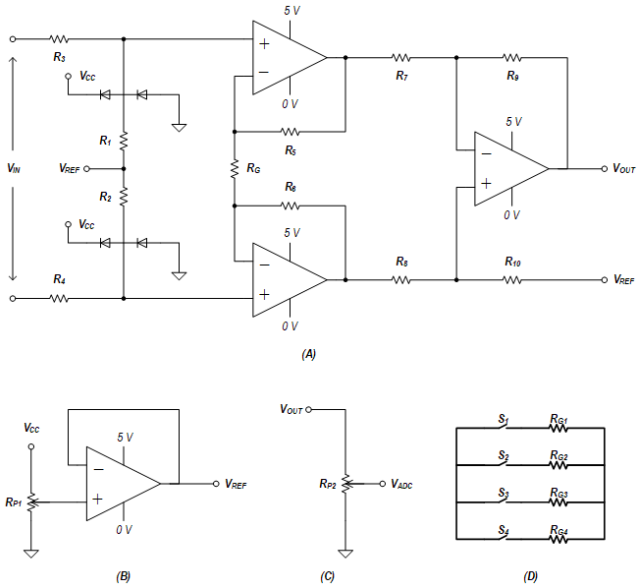

Figure 2: a) Diff Amplifier with input attnennuator and protection circuit b) refernce voltage generator c) output attennuator d) Rg for variable gain

protection and better output swing. We have used analog switch (4066), to vary the resistor R g , which in turn varies the gain

3.5 Output attenuator

At output stage we have connected a POT as an attenuator to match the output range (0-5 V) of the amplifier stage with the input range (0-3.3 V) of the ADC.

4 Controller and Firmware

We chose Tiva C Launchpad board for our project. It uses **TM4C1236GHPM** MCU . It doesn't have embedded wifi capabilities , so we were using **CC3100** add on board for the same . has 2 inbuilt ADC's. TM4C123GH6PM has in-built ADC module with sampling frequency approaching 1 Mega samples per second and with a resolution of 12 bits . It supports various triggering modes.

4.1 Sampling and Timing

In any Oscilloscope, the task of setting up correct sampling interval is of utmost importance. To ensure that stringent timing requirements are met ADC was configured in *timer triggered* mode . Inbuilt ADC module in TM4C123GH6PM was used.Given below are the cofigurations used by us for setting up the sampler

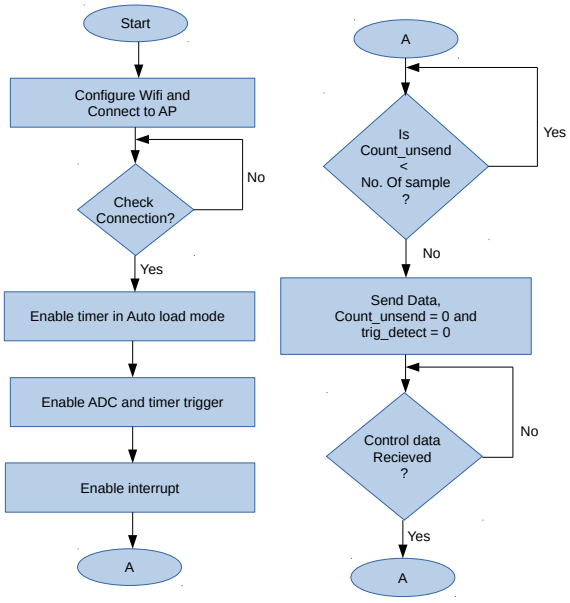
Start
Configure Wifi and Connect to AP
Check Connection?
No
Yes
Enable timer in Auto load mode
Enable ADC and timer trigger
Enable interrupt
A
A
Is Count_unsend < No. Of sample ?
Yes
No
Send Data, Count_unsend = 0 and trig_detect = 0
Control data Recieved ?
No
Yes
A

Figure 3:

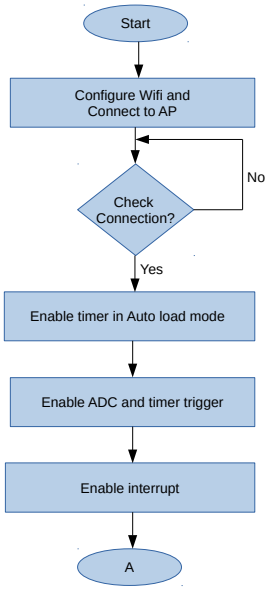

Figure 4: Flowchart: interrupt handler

ADC Channel : ADC_CTL_CH4
Triggering mode : Timer Trigger
Sampling Freq : 512 kHz (varied later by downsampling)

Timer A is configured to generate periodic interrupts at a rate of 512 kHz which inturn triggers the ADC module.
source files : *Timing and ADC configuration main.c*
Interrupt handler needs are enabled in tm4c123gh6pm_startup_ccs.c

4.2 Circular buffer and Triggering

The sampled data is then stored in a circular buffer of size 512 * 16(see to timing control section). The value is set by the macro

```
#define NO_OF_SAMPLES 16384
```

A seperate buffer of the same size is kept for taking the first order difference as

```
diff_buf[i] = inb_buf[i] - in_buf[i-1]
```

At the time of sampling itself, the sampled value is checked for a **trigger level** (set from ui) and a caliberated difference value in diff buf(to distinguish edges). once the trigger is detected a **counter is incremented for every sample acquired** . The process is continued till the counter exceeds the NO_OF_SAMPLES. After that the interrupt is disabled and **DATA PACKET** is pushed to the server (laptop) and waits for the reply back from laptop. Once the Controller receives the data from computer it reconfigures the sampler and re-enables the interrupt. And the cycle continues.
When the trigger condition is not detected till a specified pre-defined threshold , it sends the data captured to the Laptop without any synchronisation. Thus resulting in a running wave form.(see trigger control)

4.3 Wifi Connection and UDP Client

Tiva provides seperate SDK for their wifi modules. In our case it was the **CC3100 SDK** . (basically the simple link driver library). We builded upon the *getting_started_with_wlan_ap*. The MCU can be programmed to be in Station Mode or access point mode(*sl_common.h*). When acting as station it gets connected to the router using the SSID and password specified in sl_common.h

To communicate data over Wi-fi we are using UDP protocol. The MCU acts as a UDP client and PC acts as UDP server . *IP_ADDRESS* of the laptop and the *PORT* needs to be specified in main.c

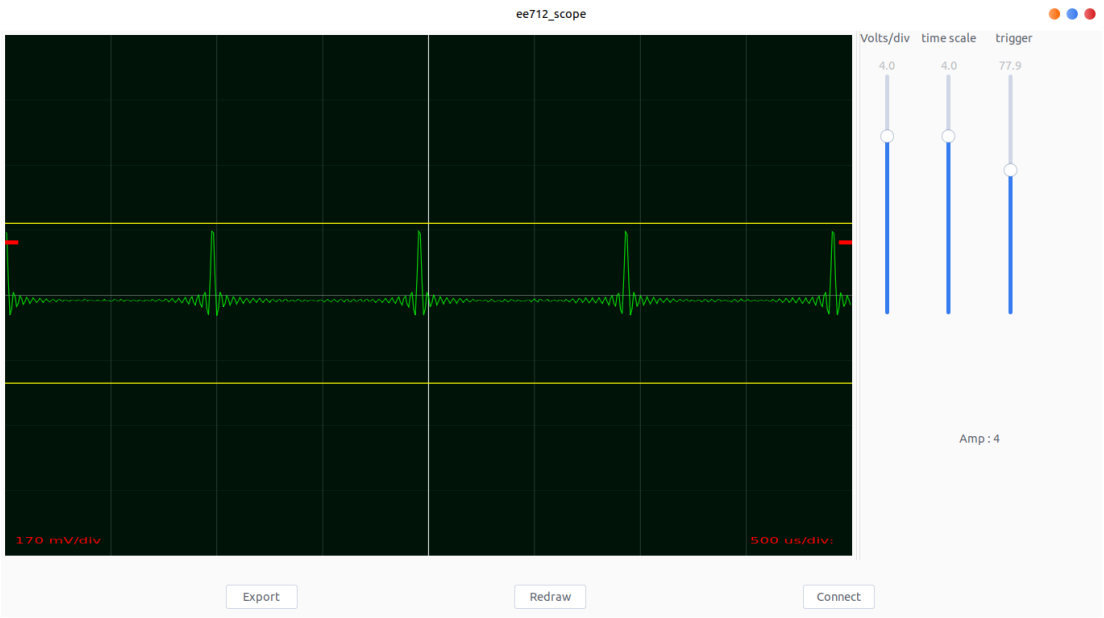

Figure 5: Oscilloscope Ui depicting the features

5 DSO-User-Interface and UDP Server

C Libararies Used	
User Interface	gtk 3
Plotting Wave form	cairo
Multi Threading	pthreads
UDP	linux system libraries

Since the UDP server side uses linux system libraries, the ui might work in linux systems only.
The program for UI is purely event driven. Various events and controls in ui are described as in the following sections

5.1 Connect Button:

The botton spawns two threads , UDP socket thread and Async Plotter.
Multithreading ensures the responsivity of UI

5.2 Redraw Button:

The button is to replot the graph ion case when the user interface is stuck.

5.3 Export Button:

The button is used to invoke the function to collect the samples recived and save them in a .csv file. The data can be used to check the plot observed or as a saved analysis.

5.4 Threads

```
void* socketThread(void *arg)
```

This thread implements a UDP socket server and waits for an incoming packet from MCU . Recieved data is copied to shared buffer. Then control data from UI is send back to the Micro controller . It triggers the async Plotter thread to start plotting the data.

```
void* async_plotter_thread(void *arg);
```

Copies the data from shared buffer to a dedicated buffer and request the gtk drawing area for a redraw by invalidating the draw area

```
gtk_widget_queue_draw(ui_pointer->w_drawing_area);
```

Redrawing of the ui is done by the function

```
static void do_drawing(cairo_t *cr, double WIDTH, double HEIGHT)
```

5.5 Voltage scale slider :

The slider can be used to change the voltage per division . By default it is set to the minimum value. The change in scale will send the gain value to the controller and the respective resistor value will be selected.The selection of resistor is taken care by analog switch .

5.6 Time scale slider :

The slider can be used to change the time per division on the display.By default the sampling frequency is 512 Khz which is the maximum(for simplicity). The change in scale will send the requried sampling frequency by the user and controller will take care of it based on the input recieved. Down sampling is done in the micro controller itself to reduce data transmission.

5.7 Trigger level slider :

On the interface , thje trigger slider can be used to trigger the waveform at a particular value.By defalut the trigger level is 0 V.To implement the trigger we are using 128 different levels to accomodate different 1024 voltage levels.The respective trigger value will be send to the controller before receiving the data to be plotted.

6 Source code

Source code for this project is spread across two public repositories in github
User interface ee712-Scope
Firmware for MCU firmware

7 Conclusion

This wireless DSO module can be used for measurements over Wi-Fi so that it can be observed anywhere in the range of Wi-Fi module. Further since we use UDP. The data can be sent to any node in an intranet. This will be helpful in carrying out filed measurements from a lab. However, the features of DSO included in this project is very limited. Also, for increasing the range of frequencies that the DSO module can measure, an ADC with high sampling rate is recommended. Wi-Fi technology is used in this application. Hence it works about 10-100m at very high-speed data transmission of 10-105 MBps. Wi-Fi doesn't consume high power compared to normal DSO. Now a days, tablet computers are used very much among people. When this GUI interface is once installed in the tablet computers, we can use whenever we are in need of Oscilloscope and its remote control is possible. Wi-Fi transmitter is also portable. And this interface also makes so easier to observe response of the circuit as graphical waveform.

8 References

- `http://users.ece.utexas.edu/~valvano/arm/wirelessconnectivity.html`
  for making tiva board wifi connection
- TIVA C peripheral library users manual
- `https://www.geeksforgeeks.org/udp-server-client-implementation-c/`
- `https://www.gtk.org/documentation.php`
- `https://developer.gnome.org/gtk3/stable/gtk-getting-started.html`
- `https://www.cairographics.org/samples/`
- linux man pages