Multi-Channel Digital Oscilloscope Implementation over Android Device

Chongyi Yang[1,2], Chanpin Chen[1,3], Zhenhao Wu[1,2], Jiashun Jiang[1,2], Sicheng Su[1,4], Xue Kang[1,3] & Qingping Dou[3]

[1] Institute of Internet of Things and Logistics Engineering, Jinan University, Zhuhai, China

[2] Department of Electronic Information Science and Technology, School of Scientific Intelligence and Engineering, Jinan University, Zhuhai, China

[3] Department of Internet of Things, School of Scientific Intelligence and Engineering, Jinan University, Zhuhai, China

[4] Department of Software Engineering, School of Scientific Intelligence and Engineering, Jinan University, Zhuhai, China

Correspondence: Qingping Dou, School of Scientific Intelligence and Engineering, Jinan University, Zhuhai 519070, China. E-mail: tdouqingping@jnu.edu.cn

Abstract

Signal monitoring and waveform analysis play a significant role in state-of-the-art signal processing and electronic measurement. Traditional oscilloscopes tend to be heavy and huge, which makes it impossible for outdoor signal measurement. In addition, most of those oscilloscopes can measure merely two signals simultaneously. This article proposes the design of multi-channel digital oscilloscope over common Android mobile device. In our system we use STM32 development board to implement up to eight input channels, data processing and wireless transmission. In addition, an Android application is designed for Wi-Fi data reception, respective waveform demonstration and derivation of each signal's amplitude and frequency. In order to transmit up to eight digital signals simultaneously as fast as possible, we designed an algorithm where all signals' data can be transmitted within a surprisingly small amount of wireless data. In our system test, wireless data transmission is implemented and each waveform can be recovered and demonstrated basically.

Keywords: digital oscilloscope, Android, multi-channel, wireless transmission, STM32

1. Introduction

As a necessary instrument for signal processing and analysis in academic research, oscilloscope can measure waveform of input signal and display parameters such as signal frequency and amplitude (Sun, 2012). However, oscilloscopes are large in size and expensive in price, which bring troubles to the majority of electronic enthusiasts(Liu, Zeng & Liu, 2017). For teaching organizations, oscilloscopes are usually applied in school laboratories, but the utilization rate in daily life is relatively low. Users' platform and system interface become an important trend of the development of oscilloscopes with the increasing demand of oscilloscopes (Liu & Zhang, 2015). At the same time, with the continuous development of the Internet of things technology, the application of embedded system is more and more extensive, ARM processor-based instrument control, data display applications can be found everywhere, which makes can make oscilloscopes more compact and convenient.

Through documentary research and data collection, we find research over oscilloscopes is still in the leading position in foreign countries (Wang & Li, 1999) (Yu, 2014). There are few devices in China that use Android device to observe waveforms. At present, digital oscilloscopes mainly apply the system based on ARM, STM32 and STC series. Although these technologies are mature, they still have disadvantages such as large error rate of experimental results, big size of experimental equipment, high price of experimental equipment and inconvenient observation of waveforms. With the continuous development of improvement research, tektronix launched three new oscilloscope applications for android smart phones in 2013, which can verify that the network and oscilloscope can realize remote monitoring of waveform and measurement from smart phones. At the same time, without operating the oscilloscope, users can complete the positioning and scaling of the waveform on smartphone. In

addition, some research institutions began to study portable virtual dual trace oscilloscopes to implement digital measurement (Zhu, Sheng, Dai & Guo, 2017), processing and display of analog signals. The main idea of the study is that after the signal conditioning circuit processes the input signal, it is converted into a digital signal by AD converter. Through WIFI communication (Cai, Kang & Zhang, 2016), FPGA sampling control center (Qu & Zhang, 2017) sends original data to android APP and obtains configuration information of working mode from APP. At last, the data processing and display are completed by the android APP. The user's interface is easy to understand and easy to operate. Through the update and maintenance of android APP, function expansion and secondary development can be achieved.

Differ from traditional oscilloscopes, virtual oscilloscopes, generally implemented on android APP and ARM processor, are relatively cheap, convenient and easy to use (Wang, 2013) (Li, 2014). The update of the program is more convenient than the that of the traditional experimental equipment, which can save a lot of money in equipment purchase. At the same time, for beginners and electronics enthusiasts, Android oscilloscope simulation software can meet their basic requirements, which can reduce their cost in purchasing expensive traditional oscilloscope.

Based on existing research results of mobile phone oscillograph, our team propose a multi-channel digital signal oscillograph based on STM32F407 and Android (Song, 2013) mobile phone program software. The main content of the study is divided into the following two parts:

Hardware design of signal acquisition board and wireless data transmission. The hardware system applies STM32F407ZGT6 chip as the core processor with high integration and high performance. Furthermore, it provides 168MHz cortex-m4 kernel for system clock. Up to eight input channels, one USB-232 interface and one Wi-Fi module are used on the development board where data acquisition and Wi-Fi transmission can be implemented simultaneously on one board.

(2) Android intelligent mobile terminal app software design. The software intends to run the oscilloscope application on any Android phone. The application receives the Wi-Fi data from the signal acquisition board, separates the received multi-channel digital signals, displays the waveform of each signal separately and stores data locally. Through the interactive interface, users can scale the waveform, add or subtract functions, and view amplitude, frequency and other parameters.

2. Design of Digital Oscilloscope on Android

We divide our whole design into hardware system and software system. Here we adopt STM32F407 (He & Yu, 2013) development board as solution of hardware system, which enables us to process multi-channel input signals simultaneously. Furthermore, we apply a simple but efficient method where multi-channel digital signals can be quickly transmitted within a surprisingly small amount of data through a Wi-Fi module connected with the development board. When input signal is analog instead of digital, however, the signal will primarily be converted via A/D module and the identical procedure is followed thereafter.
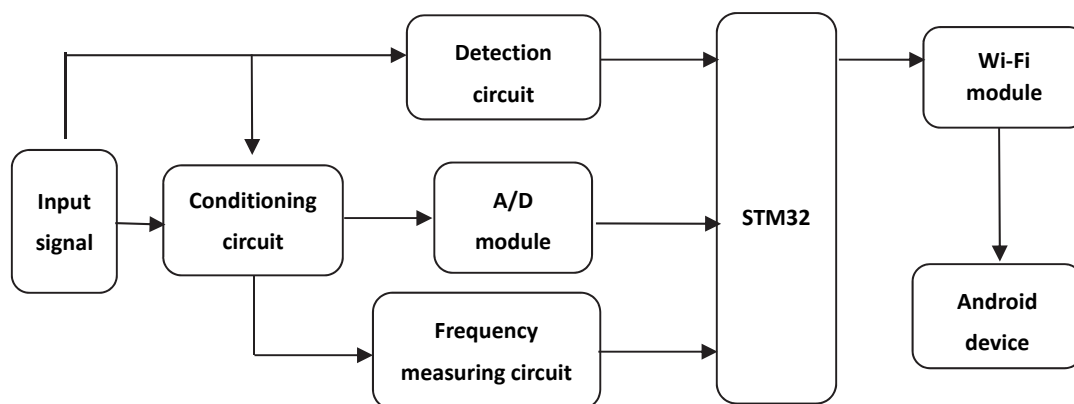

Figure 1. Digital oscilloscope system structure

Figure 1 above demonstrates the thorough design of hardware system. At the beginning, each signal is sent to the conditioning circuit, which outputs the signal that conforms with A/D input requirements (e.g. signal's amplitude, maximum voltage). Then the output signal is sent to A/D module, which converts the output signal into digital signal. The converted digital signal is then connected to the development board (notice that digital input signals are not required to be delivered to A/D module). In addition, frequency measurement is indispensable in waveform measuring domain, so we add a frequency measurement circuit module where the frequency of original signal can be derived. After a series of signal processings in STM32, all signals' data from eight channels is transmitted simultaneously via ESP8266. To summarize, the input of the hardware system is eight signals while it outputs a continuous wireless data.

2.1 Sources of Digital Signal

Unlike analog signal, digital signal is discrete, which enables us to represent a digital signal with 1s and 0s only (Larsen, 1980). For one thing, digital signal widely exists in A/D sampling module, base station of telecommunication; for another, it is widely used in household electrical appliances such as stereo set and DVD player. In A/D system, analog signal is converted into digital signal so that it can be recognized and processed by other digital apparatus. In modern telecommunication system, audio signal is converted into digital signal so as to reduce distortion during the whole transmission. Computer generates digital signals in order to transmit varieties of encoded information.

Theoretically, all of these digital signals can be adopted as our system input. For simplicity, in this article, however, we use the digital signal generated from signal generator. Such signal is fairly appropriate to be used in our system because (a) all the potential glitch and noise has been filtered within signal generator, which ensures the precision of our system input; and (b) a set of parameters such as amplitude and frequency over the signal can be manually adjusted.

2.2 Central Processing Board

In our hardware system design it is necessary to choose a high-performance development board as a central processing circuit board where multiple functions can be implemented, and various interfaces can be accessed. STM32F4xx series is an ideal alternative because it works in a high speed, provides lots of I/O interfaces and lots of modules have been integrated in the board. In this section we introduce some important functions of the board to be used in our hardware design.

2.2.1 Development Board Processor: STM32F407ZGT6

STM32F407 series is based on high-performance ARM Cortex-M4 core running with 168MHz system clock. Compared with Cortex-M3, the previous structure of ARM, Cortex-M4 core supports Floating Point Unit (FPU) calculation with single precision, which is widely adopted in ARM data-processing operation. Furthermore, lots of I/O interface and existing interface resources makes it available for users to implement all sorts of functions. Due to its high frequency, we can process eight channels of input signals simultaneously at a relatively high speed. Other STM32 series such as STM32F103 may also implement our system design. However, processing latency will increase due to its low system clock frequency.

2.2.2 Keil as a Programming Software

Keil (Zheng, Han & Li, 2009) is a core Integrated Development Environment (IDE) facing embedded system development. It includes some developing tools such as single-step debugging, code revisor. In addition, users can debug their own development board via Keil with an emulator after choosing the matched ARM chip in Keil. Furthermore, this software is very tiny and easy-to-use where many embedded system programmers develop their board using Keil.

2.2.3 Processing Multiple Input Signals via GPIOs

The fundamental function of GPIO is the implementation of basic input or output where input and output contain high voltage level and low voltage level. In our central processing development board, it is required to configure related registers so that the designated port may work under the specified working mode.

Step 1 Specify the ports to be configured

In STM32F407 there are 112 GPIO ports where users can configure them respectively. In our system design we need to process eight signals, and each signal is brought to its own input channel. Thus, we need to specify up to eight GPIO ports where each port acts as an input channel. For convenience, in this step we use PA0~PA7 as our system input channel.

Step 2 Input Mode Configuration

In this project, you need to configure the GPIO port to enter float mode.After configuring this mode, its input is completely determined by external factors. In data communication, both the receiver and the transmitter need to accurately obtain voltage level between them without external intervention.After programming the GPIO port as a multiplexing function, the input data register will sample the data on the pin every 1 AHB1 clock cycle, and the read access to the input data register can obtain the state of GPIO.

Step 3 Specify the Working Speed of GPIO (Gao, Xu & Wang, 2017)

GPIO has two speeds include the pin speed of GPIO and the turnover speed of GPIO.GPIO pin speed is also known as the response speed of the output drive circuit, which can be understood as the bandwidth of the output drive circuit, namely the maximum non-distorted frequency when a drive circuit passes through the signal.If the signal frequency than the response speed of the drive circuit, it is possible to distortion, and if the unlimited increase corresponding speed of drive circuit, and will make noise increases, so the speed of GPIO pin needs to be set in a reasonable range, in order to ensure the signal transmission speed and efficiency, we put the GPIO pin speed adjustment to 50 MHZ.The turnover speed of GPIO refers to the speed at which the 0,1 value of the input/output register is reflected in the high and low level of the external pin (APB2). The manual of STM32 indicates that the maximum turnover speed of GPIO can reach 18MHz.

2.2.4 LCD Module

TFT-LCD (Thin Film Transistor-Liquid Crystal Display) could be a solution of our hardware system design. Differ from passive TN-LCD and STN-LCD, each of the pixel in the displayer contains a thin-film transistor where interference during non-gated state can effectively be avoided. Additionally, static characteristics of such liquid displayer is irrelevant with scan lines, which greatly improves image quality. TFT-LCD module adopts TFT-LCD panel and true color image can be displayed in 16 colors. In our hardware design we use the on-board LCD interface to switch on TFT-LCD, display ASCII characters in different colors, print the identification number of LCD controlloer via COM port and display the number on LCD at the same time.

2.3 Derivation of Signal Preprocessing Circuit

Generally, analog-to-digital converter (ADC) can only collect signals within a range between 0 and 3.3V, otherwise it may produce inaccuracy during conversion or may even damage the converter. Additionally, owning to the frequency measurement module in our hardware design, input signal also needs to be preprocessed before frequency measurement can be carried out. Therefore, signal preprocessing is the primary stage over the project. Here we introduce a conditioning circuit where input signal can be effectively preprocessed (see fig. 2).
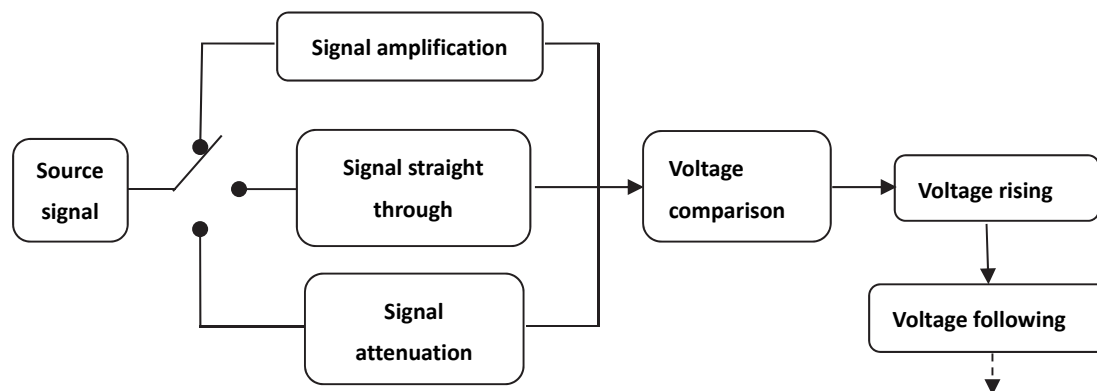

Figure 2. Structure of the conditioning circuit

Step 1 Through-Path Circuit Design over Input Signal

"Through path" means the output signal is derived via voltage comparison, lifting and follow without amplification or attenuation over input signal. Then the output signal is processed by ADC and MCU and finally outputs the corresponding amplitude and frequency.

Step 2 Design of Small Signal Amplifier

As we can see, the appropriate input voltage range for MCU is 0~3.3V. In order that MCU can identify weak signals effectively, a small signal amplifier circuit should be added here. To ensure the input voltage appears to be

consecutive, here we set the amplification factor to be 5, and the voltage range of input signal is between 0 and 0.5V.
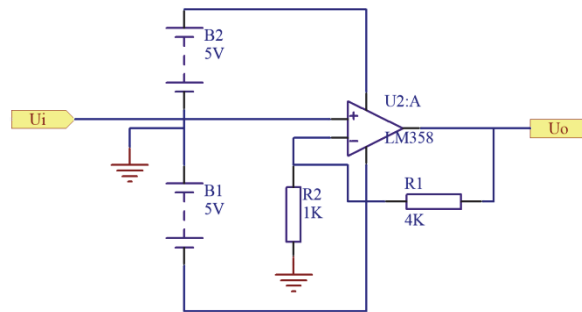
Figure 3. Small signal amplifier circuit structure

Step 3 Design of Large Signal Attenuation Circuit

Contrary to small signal processing above, we apply signal attenuation over large signal input. First, we reduce the amplitude of large signal to the voltage range of through-path mode. Then we execute step 1 above and finally send it to ADC. The process is done by MCU where we can obtain our expected output via programming. From the circuit diagram above, we can see the input signal is attenuated five times through the circuit. To ensure the voltage measurement range is consecutive, voltage range of the input signal should be 3-15V.
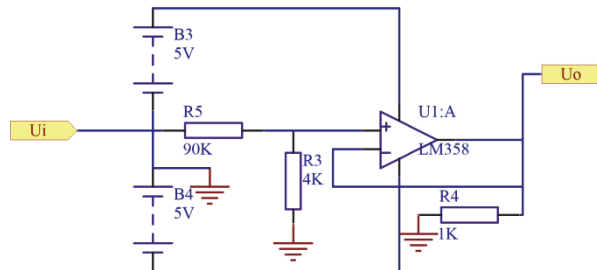
Figure 4. Large signal attenuation circuit structure

Step 4 Design of Voltage Comparator

Here we implement a voltage comparator in the conditioning circuit so as to measure the frequency of input signal. In this step we measure signal frequency using two internal timers of MCU: one acts as a timer, the other acts as a counter so that the number of pulses can be calculated in a unit time. Therefore, in order to measure the frequency of the input signal, it is necessary to transform input signal into pulse signal via a voltage comparator, and zero comparison is applied in our design.
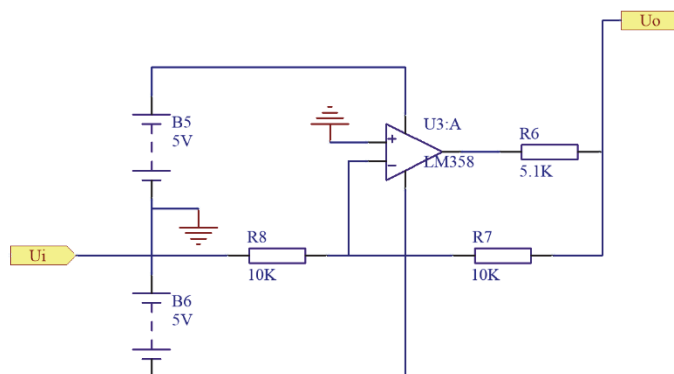
Figure 5. Voltage comparator circuit structure

Step 5 Implementation of Voltage Lifting Circuit

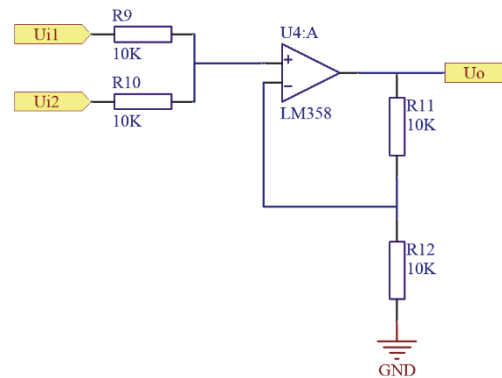In this step we apply an adder which enables to lift the negative voltage towards the range of through-path mode.

Figure 6. Voltage rising circuit structure

Step 6 Design of Voltage Follower

Voltage follower contributes to stability of output voltage. Even if output load is added, amplitude and waveform fluctuation will be trivial. Figure 7 shows the design of voltage follower design applied in our system.
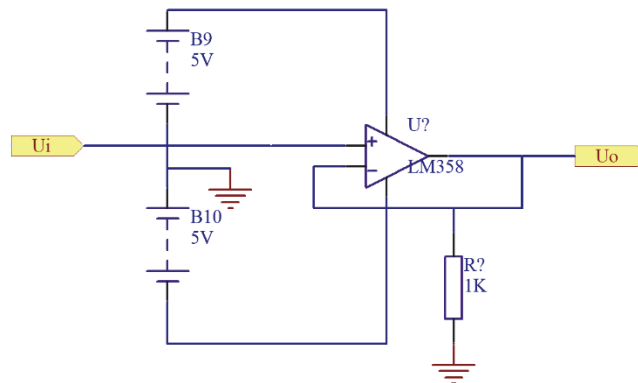

Figure 7. Voltage Follower Circuit Design

2.4 Design of Signal Sampling Circuit

Contrary to STC80C51 and STM32F1, STM32F4 applies Cortex-M4 core, adds FPU and DSP instructions physically, and its system clock reaches up to 168MHz, which makes it especially suitable for high-speed signal sampling. Here are the key parts of signal sampling module applied in our system design.

①**Function Generator**

Function generator outputs different kinds of signal waveform including sinusoidal, square, triangular, sawtooth and even arbitrary waveform. Some function generators can implement AM, FM, PM, PWM and VCO control. In our project, function generator is mainly used to generate standard signals for sampling input as well as adjust the precision of sampling module.

②**ADC Module**

In this part we use the ADC module of STM32. It converts analog signal to digital signal via 12-bit successive approximation. Each of the ADC contains 19 channels and up to 16 external, 2 internal sources and V_{bat} channel can be measured in four different modes. The result can be stored in a 16-bit data register in either left-aligned or right-aligned format. The maximum conversion rate of ADC in STM32F4 is 2.4 MHz, that is, the fastest time for each data acquisition is 0.41us.

③**DMA Module**

Directly Memory Access (DMA) module transmit data very quickly without assistance of CPU, nor does it reserve or restore the field. It can greatly improve the speed of data transmission via direct data transmission path commonly used in RAM and I/O device. When sampling with ADC, the rate data generation may exceed CPU processing rate, resulting in decline of signal acquisition accuracy. Therefore, applying DMA module becomes a solution to high-speed transmission and high-precision signal acquisition. Once a converted data is acquired, it is directly transmitted to Wi-Fi module via DMA channel without intervention of CPU, which greatly improves the efficiency of data transmission.

2.5 Solution towards Wireless Data Transmission

There are many options where wireless data can be transmitted. Considering the reduction of system cost as much as possible, here we introduce two typical transmission protocols: Bluetooth and Wi-Fi.

2.5.1 Bluetooth Transmission (Liu & Li, 2011)

Bluetooth technology aims to transmit wireless data in a short distance for electronic device using fairly small amount of power. It supports very short data package. For instance, in Bluetooth 4.0, the highest data transmission rate is only 24 Mbps. Additionally, connection establishment can be completed within 3 ms implemented by an application. and with milliseconds of transmission speed to complete the approved data transfer, immediately cancel the connection. Bluetooth low power technology uses 24-bit cyclic repetitive check to ensure the stability of all packages in case of interference. It is the benefits of Bluetooth transmission that make it possible to become a way of data transmission in our system design. However, it limits the length of data package in Bluetooth transmission. Thus, Bluetooth transmission fails to satisfy our requirement.
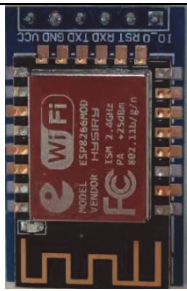
2.5.2 Wi-Fi Transmission (Tang & Chen, 2013)

At present, the main ways of communication between smart devices and other peripherals in the market are Wi-Fi wireless communication, Bluetooth, and USB communication. Compared with Wi-Fi and Bluetooth, USB communication is wired, while wireless communication is more convenient. Wi-Fi wireless communication works in the 2.4ghz frequency band, and the supported rate can reach 54Mbps. The distance of large WiFi in shopping malls is about 300 meters, while Bluetooth is a point-to-point or point-to-multipoint transmission, generally 30 meters, and the transmission rate is generally 1Mbps, which is far less than WiFi communication. After comparison, the Wi-Fi communication mode is more reasonable. In this experiment, the atk-esp8266 WiFi module is used.

ATK-ESP8266 is a high-performance Wi-Fi (serial port -- Wi-Fi) module launched by ALIENTEK.The ATK-ESP8266 module uses a serial port (LVTTL) to communicate with MCU (or other serial devices), with built-in TCP/IP protocol stack, which can realize the conversion between serial port and WIFI.

Through the ATK-ESP8266 module, the traditional serial device only needs a simple serial port configuration, which can transmit its own data through the network (WIFI). The ATK-ESP8266 module supports LVTTL serial port and is compatible with 3.3v and 5V single-chip microcomputer system. The module supports the modes of serial port to WIFI STA, serial port to AP and WIFI STA+WIFI AP, so as to quickly build the data transmission scheme of serial port to WIFI.

The ATK-ESP8266 module is very tiny (29mm*19mm only). The module is connected with the outside through 6 rows of pins with an interval of 2.54mm. The appearance of the module is shown in table 1 below.

Table 1. Appearance of ESP8266 Module

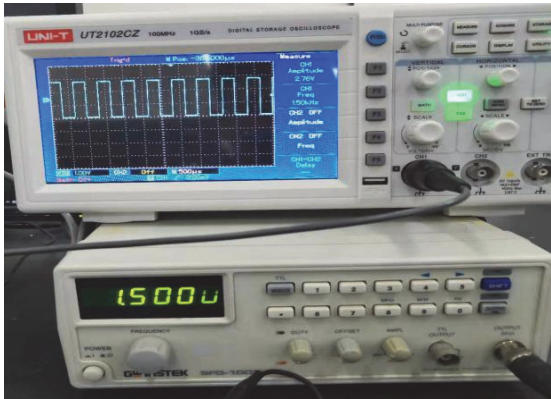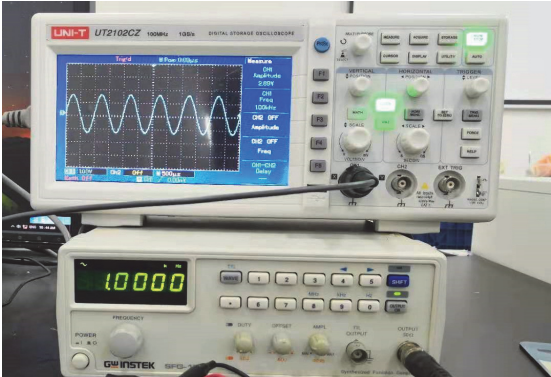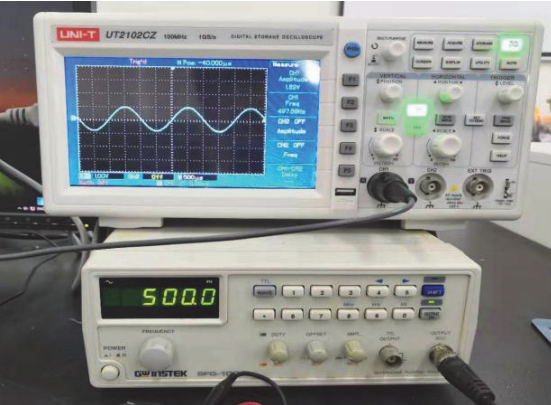| Positive Side	Negative Side

3. Waveform Demonstration and Analysis

In this section, we intend to demonstrate a couple of results step by step where each result, including signal preprocessing, signal sampling, wireless data transmission and waveform demonstration via Android, is orderly described and analyzed in detail.

3.1 Selection of Input Signals

In this section we choose three signals generated by signal generator as system input (see table 2). We first measure each of the signal and observe the waveform respectively via an analog oscilloscope as a reference. Also ensure each signal does not contain glitch.

Table 2. Waveform Demonstration of System Input

No.	Sort of Signal	Signal Waveform
1	**Square Wave** (1.5000 KHz)	
2	**Sinusoidal Wave** (1.0000 KHz)	
3	**Sinusoidal Wave** (0.50000 KHz)	

3.2 Multiple Signals' Measurement

After these three signals are connected to STM32F407, the board samples the data, stores each signal's value temporarily and generates data to be transmitted byte in byte. In order to observe the data, we deliver real-time data to the computer through USB-232 port where the processed digital signal can be displayed and supervised by serial debugger from computer. Compared with the waveform of three signals on the oscilloscope, we expect to find the consistency between altered waveform voltage and data displayed in serial debugger. Here we use three of eight channels (PA0, PA1, PA2) as three signals' input. The process and part of the data display are shown in table 3.

Table 3. Data Generation Test

Three of Eight Channels connected to Signal Generator	Generation of Data

3.3 Signal Data Transmission Test via Wi-Fi

In this section we are about to verify the validity of wireless data transmission using ESP8266 module. Using our original data generation algorithm, we attempt to transmit up to eight signals simultaneously. Firstly, we connect the ESP8266 module with computer via USB device. Configure the module by sending a set of AT instructions so that the module works under "AP+STA" mode. Once configured successfully, connect an electronic device (here we use an Android phone) with the module via Wi-Fi network so that the signal data sent by STM32 can be received by phone. Here we apply a Wi-Fi debugging software on the phone where all data sent by STM32 can exactly be visualized the analyzed by users. Table 4 below shows wireless signal data received by phone. From these data we can verify the correctness of Wi-Fi transmission itself as well as data consistency with original input data.

Table 4. Received Data Test via Wi-Fi Debugger

State 1	State 2

We can make a brief analysis towards the results shown above. Each byte contains eight bits, and each bit represents the voltage of converted digital signal. Each byte is separated by a white space so that we can observe the content of each byte in binary form more easily. Consider the most significant bit (i.e. the leftmost bit) of each byte and we find the data basically conforms with the input signal via PA0. Consider the second bit from the left of each byte and we also obtain the similar result, and so on.

Furthermore, a more comprehensive statistical analysis over the sampling count for each channel worth to complete in this step. For one thing, derivation of precise sampling frequency is necessary; for another, since the sampling frequency may vary depending on the performance of STM32 and ESP8266, sampling stability and transmission stability are essential for system evaluation.

3.4 Android Application Framework Demonstration

Table 5 below is the layout of our application design on Android. Two buttons, namely "connect" and "receive from message" are designed here. Click "connect" and users can establish a Wi-Fi connection with STM32, and click "receive from message" to receive data sent by STM32 through Wi-Fi network. Eight entrances of waveform monitors from Channel 1 to Channel 8 are implemented below the buttons. Click any horizontal bar below and we can see the waveform monitor for corresponding input channel.
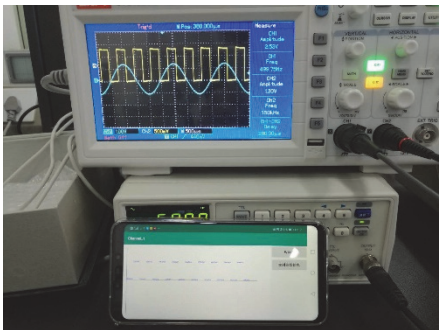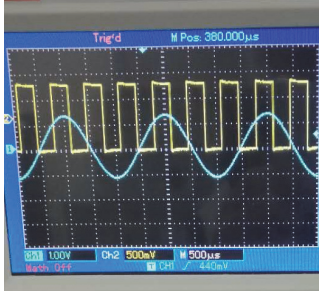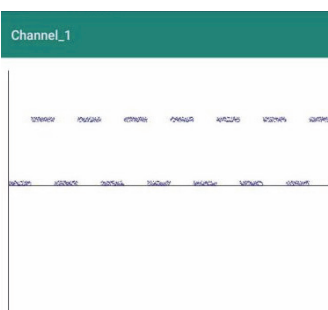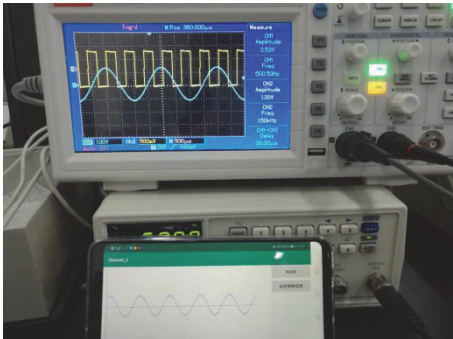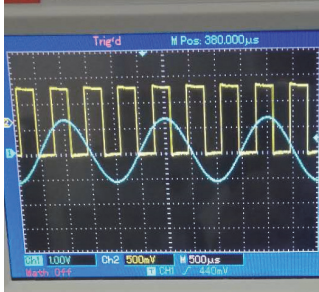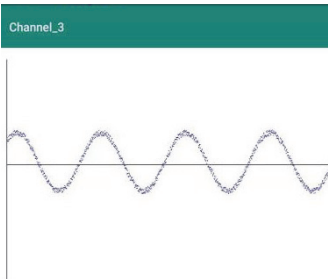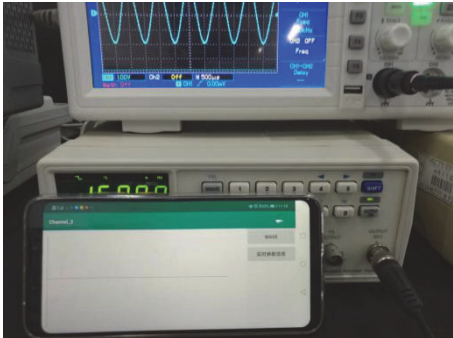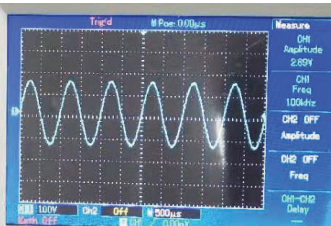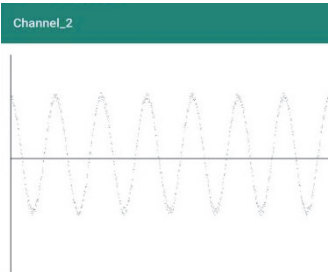
Table 5. Android Application Design

Homepage	Waveform Monitor for each Channel

3.5 Waveform Demonstration on Android

The input signal is received by mobile phone Wi-Fi , and each signal value which has been received will be

captured and drawn inside its own waveform monitor, so the signal waveform can be recovered and obtained. To enhance the experiment rigor, input signal will also be sent to a traditional oscilloscope to display waveform as a reference so that the validity of our system waveform will be verified by comparison. Table 6 below shows three input signals and their corresponding waveform. Similar to the system input in the previous steps, here we choose a square waves and two sinusoidal waves with different amplitude as three inputs.

Table 6. Waveform Demonstration

No.	Comparison of Real Waveform & System Waveform	Real Waveform (Reference)	Our Waveform
1		(see the yellow item)	
2		(see the blue item)	
3			

Through comparison between two waveform of each signal above, we can draw some conclusions listed below:

1. Our system's waveform output is basically in line with our expectation. Except for some noise, each of the waveform output is roughly identical to the corresponding reference waveform.

2. From the first row we find the square wave none of the sharp variation of square wave is drawn in our system's waveform, which makes our waveform non-consecutive. In other words, a line should be added between any two adjacent points in the waveform.

3. Compared with the second row and the third row, signal's altered amplitude and frequency can be reflected on our Android device.

4. Though the result is basically correct, the waveform tends to be light-colored once input signal's frequency becomes higher. We can deduce the points that composites the waveform may become very sparse when input signal's frequency is relatively high. It often happens when sampling frequency is not high enough. One solution towards such problem is to connect every two adjacent points with a line so that the waveform seems to be consecutive. Another solution is to increase sampling frequency so that more sampling points will be derived and waveform will be more compact and precise.

4. Discussion

Waveform demonstration and analyses on ordinary Android mobile device are a novel attempt which brings us convenience and simplicity. While those conventional heavy oscilloscopes have been widely used in laboratories, a lightweight, portable and practical substitute, on the other hand, is playing an important role in state-of-the-art Internet of Things (IoT) system. In order that the functions (i.e. waveform demonstration, amplitude & frequency measurement) provided by users can basically satisfy their requirements, we need to optimize the system further so that measurement precision can be maximized, waveform distortion can be minimized, and wireless data amount can be minimized as much as possible.

In this article we introduce an original idea where multiple signals' input can be quickly transmitted and visualized simultaneously. However, we haven't implemented amplitude and frequency measurement in our current system design, which could be the primary task of our research. In addition, we find some noise exists in the result waveform. Suppose the frequency of input signal is relatively high, such noise will damage the signal waveform and will intervene analysis over the signal. How to remove the noise or improve SNR becomes a critical problem for future research.

Another problem is further improvement on sampling rate (2007). As digital signals can be represented using 1s and 0s, we can obtain real-time voltage without A/D conversion, which saves a lot of time during data generation. Whereas latency caused by A/D conversion cannot be ignored if many input channels need to be converted. Adopting a more advanced ADC or processor may improve performance, but it will be much more expensive. Thus, taking a better algorithm into use is the destination for future improvement.

Upsampling is a new idea to solve the sampling problem when sampling rate is limited. In a DSP system, some approximate values will be interpolated into a derived sequence where those values are expected to be equal to the discrete signal generated at a higher sampling rate. However, probably a DSP chip will perform the algorithm much better than STM32 due to the large amount of computation, say, the implementation of DTFT.

Nevertheless, from the summaries and results above, our tentative system design can still be applied in various occasions. For one thing, the unparalleled flexibility of our system enables users to implement outdoor measurement; for another, various signals can still be demonstrated roughly. Since the most unsatisfactory problem is the limited sampling rate, applying DSP chip instead of STM32 may improve our system performance greatly.

Acknowledgments

References

(2007). Research on Gain Controllable Sampling/Holding Circuits. (Doctoral dissertation, University of Electronic Science and Technology)

Cai, X. W., Kang, B. Q., & Zhang, K. Q. (2016). Design of portable Wi-Fi virtual oscilloscope. *Journal of chifeng university (natural edition)*, *32*(11), 29-31. https://doi.org/10.3969/j.issn.1673-260X.2016.11.014

He, Y., & He, Y. (2013). Single-phase electrical parameter measuring instrument based on stm32f407. *Journal of Chengdu University (Natural Science Edition)*, *32*(4), 386-388.

Highland, X. C., & Wang, Y. (2017). High-performance DSP processor GPIO port design. *Chinese integrated circuit*, *26*(z1), 24-27.

Larsen, R. S. (1980). An introduction to digital and analog integrated circuits and applications. *Proceedings of the IEEE, 69*(9), 1172-1173.

Li, W. Zh. (2014). The study of the digital storage oscillograph based on the embedded system (Doctoral dissertation, shandong university of science and technology).

Liu, D., Zeng, S., & Liu, X. (2017). Design of portable oscilloscope based on stm32. *Technology innovation and*

application, (12), 77-77.

Liu, Shuai, & Zhang, H. (2015). Design of portable multifunctional digital oscilloscope based on stm32. *Microcomputer & application*, *34*(15), 37-39. https://doi.org/10.3969/j.issn.1674-7720.2015.15.011

Liu, X., & Li, H. (2011). Main problems and current status of Bluetooth transmission performance research. *Electronic World,* (14), 15-15.

Qu, B., & Zhang, X. (2017). Design and implementation of digital storage oscilloscope based on FPGA. *Automation and instrumentation,* (5).

Song, F. (2013). The development of the oscillograph application based on the Android system (Harbin Institute of Technology). Tektronix launched three android applications for oscilloscopes. *Foreign electronic measurement technology,* (2), 121-122.

Sun Dengliang. (2010). Principle and application of digital oscilloscope. Shanghai jiaotong university press.

Tang, J., & Chen, J. (2013). Application of Wifi transmission technology in control field. *Communication and information technology,* (4), 61-63.

Wang, S., & Li, D. (1999). Development status and trend of foreign oscilloscopes. *Foreign electronic measurement technology,* (1), 3-5.

Wang, Z. (2013). The design and study of the virtual oscillograph in the electronic experimental teaching system of vocational schools (hunan normal university).

Yu, Y. (2014). Development status and countermeasures of domestic oscilloscopes. *Electronic products world,* (5), 10-11.

Zheng, C., Han, T., & Li, X. (2009). Application of compiler software keil in the teaching of single chip computer courses. *Higher Education Forum,* (12), 96-97.

Zhu, D., Sheng, Q., Dai, S., & Guo, Y. (2017). Portable virtual dual trace oscilloscope. *Electronic production,* (20), 51-52. https://doi.org/10.3969/j.issn.1006-5059.2017.20.023

Copyrights