

## Real-time data acquisition serial communication

Jinji Zheng<sup>1, a</sup>, Wei He<sup>1, b, \*</sup> and Huaifu Li<sup>1, c</sup>

<sup>1</sup>School of Physics and Electronic Information, Yunnan Normal University, Kunming, China

\*Corresponding author: he99wei@aliyun.com

**Keywords:** Serial communication; data collection; visualization

**Abstract.** The serial data acquisition is based on MATLAB. It can achieve real-time communication between the computer and the microcontroller. And we analyzed data in real time through MATLAB library functions. Finally, we used visual graphics window to display the results of the data analysis. This article described how to upload MATLAB read by the microcontroller serial port to PC to the height of the ultrasonic data acquisition and real-time rendering height versus time. Connected through a computer and external acquisition, after the acquisition through the serial port or an external network to collect the sensor data to the computer, computer obtains the collected data, then mathematical modeling and theoretical study can be carried out based on mathematical model. So that one can reduce the difficulty of product development and shorten the development cycle.

### Introduction

The basic algorithm based on MATLAB platform for simulation[1], data can not be directly read peripheral data acquisition component. In this case, the user needs a computer connected to the external acquisition component and then transmitted through the network or serial port to collect sensor data to a computer. Finally, we need use MATLAB software on a computer analysis of sensor data[2][3]. MATLAB is not only a powerful scientific computing functions, but also can create a graphical user interface GUI (Graphical User Interface) to achieve the interaction between users and computers. MATLAB supports all GUI user controls integrated in this environment and provides the interface appearance, properties and behavior response mode setting method, along with improved versions. This capability will continue to be strengthened[4]. And, you can easily get a higher quality of the curve in FIG with powerful graphics capabilities[5].

Firstly, we realized the computer and the next crew of the serial communication[6]. And serial port is used to gather collect ultrasonic height sensor acquisition position. Then call the GUI controls to time the X-axis, Y-axis height, and curve display.

### Lower Computer Design

Lower machine is relative to the PC. In the field of electronics, which run on the Windows platform is usually called the host computer, with respect to the subsidiary in the following Windows called role-bit machine. The lower machine is mainly concluded MCU or ARM, which is mainly responsible for the data collection and some simple processing and transmitting or control. It mainly consists of hardware and software components of two modules.

### System Components

In this design, the microcontroller read the ultrasound data. We used serial communication way to send the data to the PC, the PC side through MATLAB to collect the data in real-time. Then it was displayed and analyzed the collected data in real-time.

The overall framework of the system was shown in Fig1.

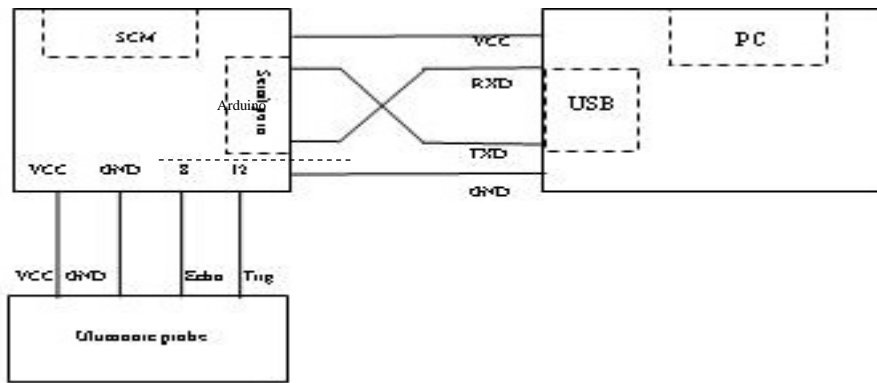


Fig.1 System Block Diagram

## Hardware Design

The hardware consists of an Arduino microcontroller and an ultrasound module. Arduino is an open source code based on the simple i / o platform, and has a similar java, C language development environment. So you can quickly use the Arduino language with Flash or processing software to make interactive works. Arduino can be used to complete the development of electronic components, such as switch or sensors or other controller, LED, stepper motor or other output devices. Arduino can also operate independently as a communication with the software platform. Arduino features are as follows:

- Creative CC schematic design based on open source code.
- Download freely, also following their own needs modification, subject to compliance attribution.
- Arduino can use ICSP lines burned into the device, and the bootloader burned into the new IC chip.
- Can be based on the official circuit diagram, simplify Arduino modules, complete microprocessor control to operate independently.
- Simply connected a variety of electronic components to the sensors, such as infrared, ultrasound, thermistors, photoresistor, etc. ... servomotor.
- The use of low-cost micro-controller (ATMEGA8-328).
- USB interface, no external power supply. Another provides 9V DC power input.
- Applications using arduino, breakthrough interactive content that would otherwise require a mouse, keyboard, CCD and other input devices can be more easily reached in single or multiplayer interactive games.

## Software Design

Ultrasonic sensors can be used to transmit sound waves above 20 KHz, 20 KHz or detect whether there are more acoustic sensing components. Ultrasonic sensors commonly used piezoelectric and telescopic two categories. Here is a selection of voltage type, the module can simultaneously support several output modes: serial, analog, pulse, so that we can learn how to use Arduino read the information.

Given the level of potential changes in serial mode from the RX and BW pins, the TX output will be at a rate of 9600 data, and the data will begin with the character R to the front, and the middle of three characters represents the size of the distance. The last character is CR (0x0D).

The Arduino analog signal with a resolution of 4.9mv, we Arduino to read the value multiplied by 1.27 to get the results in centimeters, and in the process the need for such conversion before they can obtain more accurate numerical results wiring time as long as connected to the power, the other is connected to the analog signal output pins on the Arduino corresponding analog signal detection pin, the core code reads:  $Val = \text{analogRead}(0) * 1.27$ .

The last one is measuring the pulse width, the module will change depending on the length of the

distance pulse of high potential, in terms of the way: 147us equal to about 1 in, when the wiring needs attention, you need to PW pin to the Arduino digital pin, general selection is the third Digital I/O, core code: `pulse = pulseIn (3, HIGH); pulse = pulse / 147; Cm = inches * 2.54;` here is the way I use three. Arduino directly to us to measure the pulse length function, and this function will return directly to the length of time data.

Wherein value `cm = float (pulse In (EchoPin, HIGH) * 17) / 1000`. Echo time is converted into cm, read a pin pulse (HIGH or LOW). For example, if the value is HIGH, `pulse In ()` will wait pin goes HIGH, start time, and then wait for the pin becomes LOW and stops timing. Then, return the pulse length in microseconds. Time ranges from 10 microseconds to 3 minutes. (1 second = 1000 ms = 1000000 microseconds). Accordingly, the high level of the received time (us) \* 340m / s / 2 = received high time (us) \* 17000 cm / 1000000 us = received high time \* 17/1000 (cm), followed by the definition of another variable `VALUE_0` by, and make the following conversion, `Value_0 = int (Value_cm * 100)`.

Finally, the detected data to the serial port shown above, and then add the serial port code, and then only need to initialize the serial port, you can call the appropriate port function.

## PC Design

GUI is called the graphical user interface (Graphical User Interface, referred to as the GUI, also known as the graphical user interface) is a graphical display of the computer operating the user interface. Compared with the early computer use the command line interface, graphical interface for users to visually is more acceptable.

## Serial Port Initialization

Through the above drop-down menu, we can apply for a serial port, its format is: `my_SerialPort = serial ('com1');` in addition we have some extra serial port for setup and initialization, the code is:

```
set(my_SerialPort,'BaudRate',9600);
set(my_SerialPort,'DataBits',8);
set(my_SerialPort,'StopBits',1);
set(my_SerialPort,'InputBufferSize',11000);
set(my_SerialPort,'terminator',10);
set(my_SerialPort, 'BytesAvailableFcn',{ @receiveData, hObject});
handles.mycom = my_SerialPort;
guidata(hObject, handles);
```

Open And Close The Serial Port. Add to open part of the code is :

```
function open_serial_Callback(hObject, eventdata, handles)
try
    fopen(handles.mycom);
    msgbox('Serial Open success! ');
    fprintf('=====Readytoreceivedata.=====\\n')
catch
    msgbox('Serial port does not exist! ');
    fprintf('=====Serial port does not exist! =====\\n')
    return;
end
```

Add to close part of the code is:

```
function close_serial_Callback(hObject, eventdata, handles)
try
    fclose(handles.mycom);
    delete(handles.mycom);
    clear handles.mycom;
```

```

        msgbox('Serial success Close! ');
catch
        msgbox('Serial port can not be closed! ');
return;
end

```

## Serial Interrupt Reading Data

Typically read serial manner inquiries approaches and disruptive manner, due to the way the query more waste CPU resources, so the way here taken to interrupt the way to read the serial data. Interrupt-driven serial port interrupts, that is, each receives a carriage return to read the data again. Specific operation is as follows:

First, specify the serial port interrupt initialization and callback function. Details Code:

```

set(my_SerialPort,'terminator',10);
set(my_SerialPort,'BytesAvailableFcn',{ @receiveData, hObject});

```

Second, the use of serial tool detects the next bit machine to send back data shown in Figure 2:

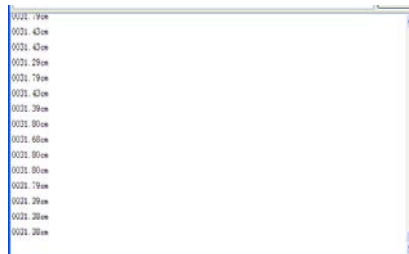


Fig.2 lower computer to send serial data detection Figure

Third, the serial data read function

```

function receiveData(hObject,eventdata,hfigure)

```

```

handles = guidata(hfigure);

```

```

number = get(hObject, 'BytesAvailable');

```

```

RawData = fread(handles.mycom, number, 'uchar');

```

```

RawData1=RawData(number-10)-48;

```

```

RawData2=RawData(number-9)-48;

```

```

RawData3=RawData(number-8)-48;

```

```

RawData4=RawData(number-7)-48;

```

```

RawData6=RawData(number-5)-48;

```

```

RawData7=RawData(number-4)-48;

```

```

RawData_out=1000*RawData1+100*RawData2+10*RawData3+RawData4+0.1*RawData6+0.

```

```

01*RawData7

```

Fourth, the data is displayed in a separate window inside.

## Data Analysis Algorithm Design

The method employed here to parse data from the array inside a bit out, eventually combined into a complete data. How to make PC to parse the data depends entirely on how the next crew to the data, the foregoing lower machine part has specified data format is 1234.56cm + Enter, so here you have the time to resolve in accordance with this agreement to resolve. Detail of the resolution process is as follows:

```

RawData1=RawData(number-10)-48;

```

```

RawData2=RawData(number-9)-48;

```

```

RawData3=RawData(number-8)-48;

```

```

RawData4=RawData(number-7)-48;

```

```

RawData6=RawData(number-5)-48;

```

```

RawData7=RawData(number-4)-48;

```

```

RawData_out=1000*RawData1+100*RawData2+10*RawData3+RawData4+0.1*RawData6+0.

```

## MATLAB Data Plotted Versus Tim

By defining a counter in the serial interrupt function, whenever the serial data received a counter variable is incremented by one to obtain a set of drawing parameters. In addition, the need to coordinate the range of X-axis and Y-axis do initialization value can be initialized through the command window to observe the counter, with the increasing of the counter value of the data will be read into the successive increases, the effect shown in Figure 3 after running shown.

After the show over a period of time to observe the changes of less data through GUI, because the specified range is displayed at initialization time from 0 to 100, so when the counter value is greater than 100, the need to change the coordinate axes to move it to the left, so that brings visual effect curve is constantly moving to the right, which is constantly changing over time. So we have the program running in the process of constantly updating the maximum and minimum X-axis and Y-axis, when the timer value exceeds the maximum value initialization, the maximum and minimum X-axis worthwhile to follow the timeline adjusted accordingly by optimization and operating results after the modification shown in Figure 4.

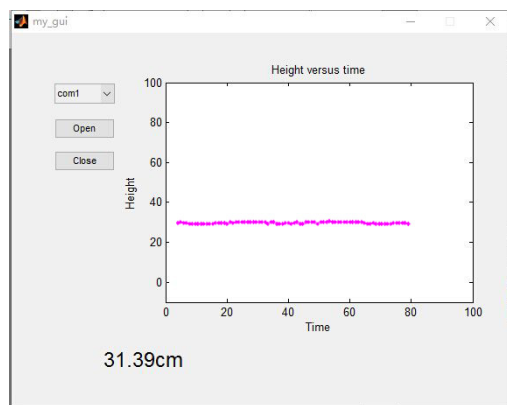


Fig.3 GUI test objects

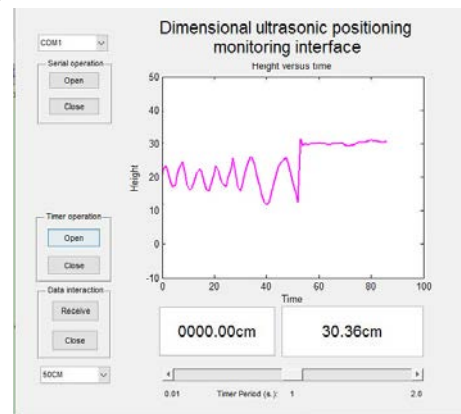


Fig.4 GUI data reading curve

## Conclusion

Connected through a computer and external acquisition, after the acquisition through the serial port or an external network to collect the sensor data to the computer, computer obtains the collected data, and then mathematical modeling and theoretical study can be carried out based on mathematical model. So that one can reduce the difficulty of product development and shorten the development cycle. Application of MATLAB powerful graphics capabilities and computing combined well to receive serial data transmission over. MATLAB-based serial data acquisition software enables data acquisition and transmission lower machine and PC. Then we get some results via MATLAB serial tools to achieve the communication between SCM and PC, and the collection of real-time data through an ultrasonic sensor. It has a certain value of MATLAB-based serial data acquisition application software.

## Acknowledgements

This project was supported by National Natural Science programs (51267021) and College students' innovative entrepreneurial training programs (2015).

## References

- [1]Zhang Defeng and others. MATLAB numerical methods [M] Beijing: Mechanical Industry Press, 1999,2009.
- [2]Zhao Yunpeng. MATLAB serial communication data acquisition application [J] Microcomputer

Information, 2006,22 (1-1): 111-112.

[3]Wang Guodong, Liu Mingyu, Yang Jiping. Based on Matlab hysteresis loop experimental serial program [J] physics experiments, 2004,24 (3): 25-27.

[4]Wang Zhan-Jun, Shen Ming. Achieve Matlab GUI Serial Communication Programming [J] Modern electronic technology, 2010,320 (9): 38-44.

[5]Nanyang, Zhou Jing and others. MATLAB-based GUI interface [J] oil equipment, 2008,22 (6): 76-78.

[6]Yang Jue. Serial communications [M] Beijing: Higher Education Press, 1999.