

Contents

Introduction.....	2
1.Aim.....	2
2.Objective.....	2
Design Information &Performance&Equations.....	2
1.Basic knowledge.....	2
2.Required experimental equipment.....	3
3.Product performance.....	3
Software design strategy.....	3
1.software programing logic	4
Hardware design strategy.....	4
1.Hardware architecture.....	5
2.Souce generator design.....	5
Circuit Design and Simulation.....	6
Design for testing.....	6
Software and Hardware design process and result.....	7
Conclusion and future recommendations.....	7
Appendix: complementation code.....	9

Chart Catalogue

Figure1-1(Logic block diagram of software programming).....	4
Figure1-2 (Overall architecture diagram of the system).....	5
Figure1-3 (Source generator circuit system).....	6
Figure1-4 (Digital signal waveform generated by the source)	7
Figure1-5(circuit simulation)	7
Figure1-6 (serial display data window).....	9
Figure1-7 (circuit simulation).....	9
Table 1-1 (experiment data).....	7

Introduction

1. Aim (Analysis of hardware and software requirements)

- (1) Reading two analog input port voltages with Arduino
- (2) Arduino internal ADC conversion based on Algorithms
- (3) Serial communication between two blocks of ArduinoBoard
- (4) There is an indicator to display the converted digital number

2. Objective(expected product function)

Complete all the expected above, and add more display methods if possible (Bluetooth this time) .

Design information &Performance & Equations

1. Basic knowledge used:

(1)The ADC on the Arduino is a 10-bit ADC, meaning it has the ability to detect 1,024 (2^{10}) discrete analog levels.

(2)Serial data is transmitted between devices one bit at a time using agreed-upon electrical signals.

2.Required experimental equipment

- (1) Two Arduino development board
- (2) A number of DuPont lines, Potentiometers
- (3) Bakery board
- (4) HC-05 Bluetooth module (For future improvement)

3.Product performance

- (1) Serial communication frame format: eight data, one stop bit.
- (2) The version of communication is stable
- (3) Only use part of the development board, the final product can be made tinier.

4.Product performance

The formula in the ADC process :

$$V_{\text{input}} * (5/1023) = \text{Digital number of voltage}$$

Software design strategy

1. Software programming logic

The software consists of two parts. They are receiving part and sending part. Figure 1-1 shows the overall logic block diagram of the software program, And the whole software design idea can be split into the following steps.

(1)Sending Code

1. Read analog signals at the analog input port
2. Calculation of voltage values by internal ADC algorithm
3. Sending digital signals through RXD and TXD ports

(2)Receiving Code

1. Receive digital signal
2. Send to a computer or other display port

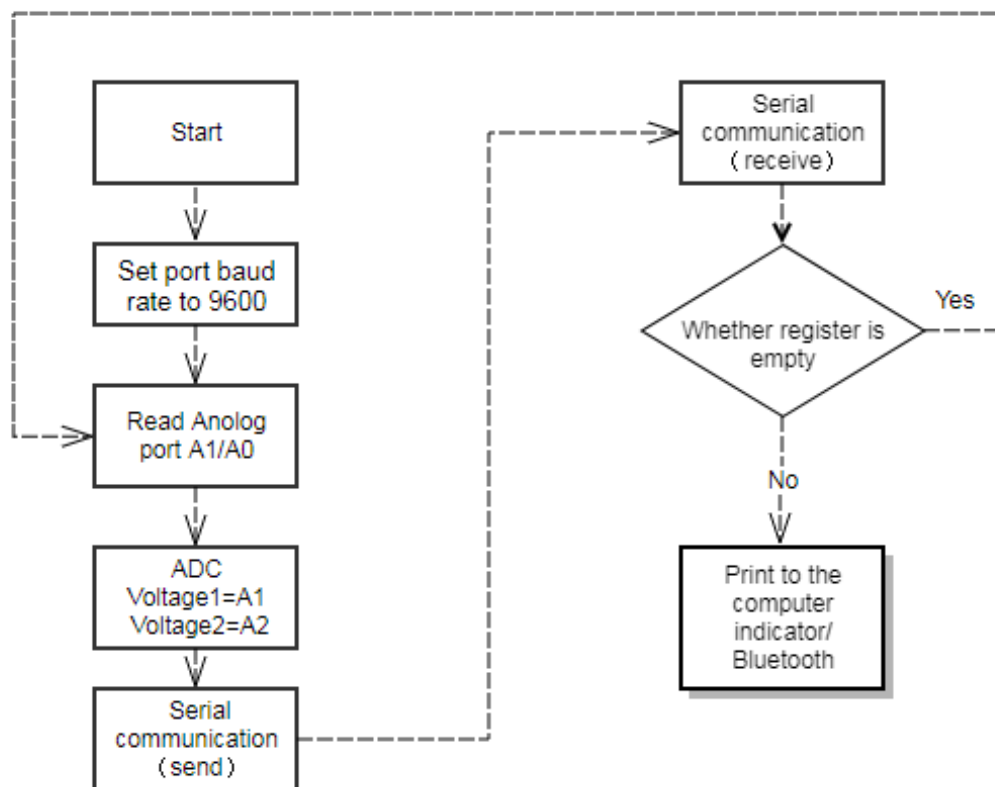


Figure 1-1 Logic block diagram of software programming

Hardware design strategy

1. Hardware architecture

The overall frame of the hardware circuit is shown in Figure 1-2, which includes:

1. Source generator circuit
2. Two Arduino development board
3. Serial communication circuit.

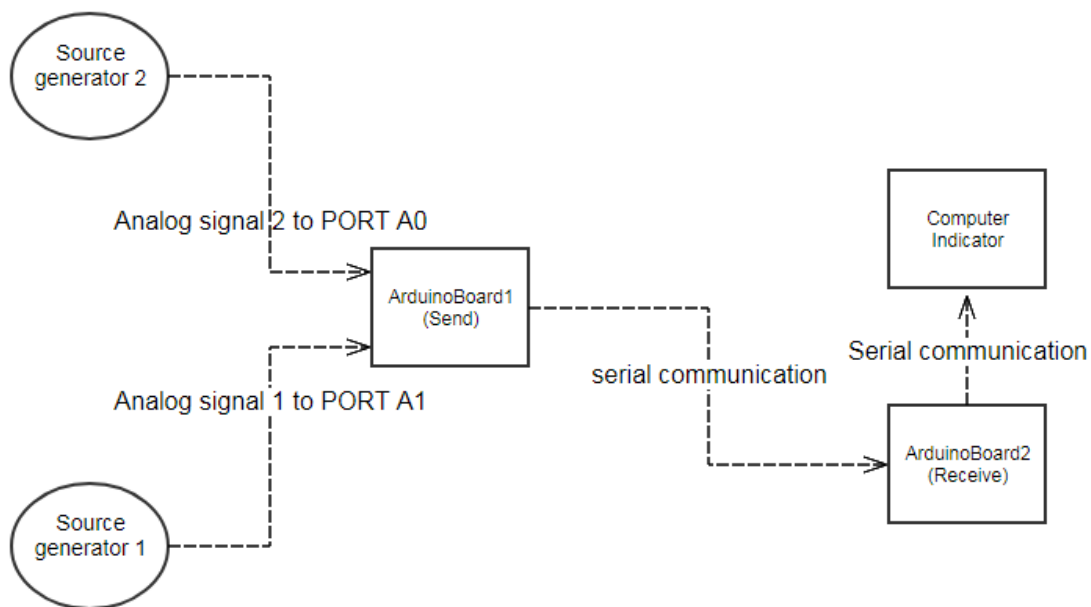


Figure 1-2 : Overall architecture diagram of the system

2. The design process of the source generator

Since the limited experimental conditions, there is no source generator. So a simple circuit shown in Figure 1-3 is built to meet the requirements of the experiment., and source requirements :

1. 0-5 V
2. analog signals, arbitrarily advisable between 0-5V

The signal waveform of the circuit is shown in Figure 1-4, which perfectly meets the source requirements of the experiment.

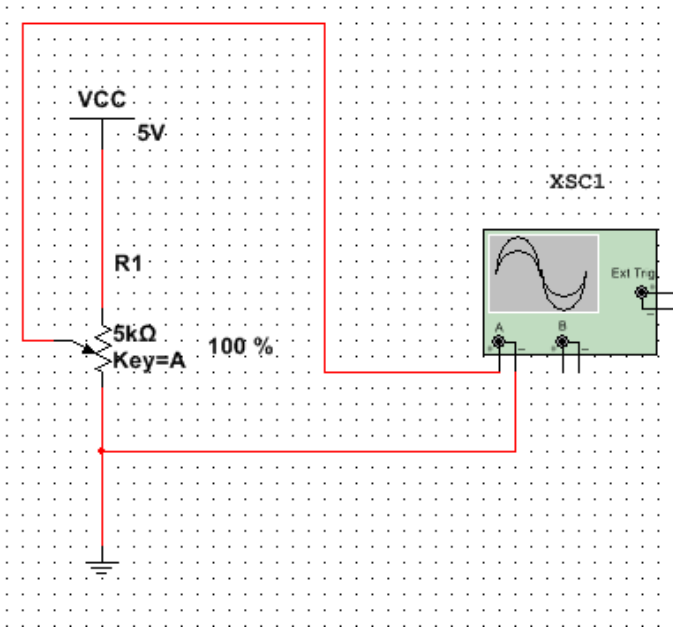


Figure 1-3 : Source generator circuit system

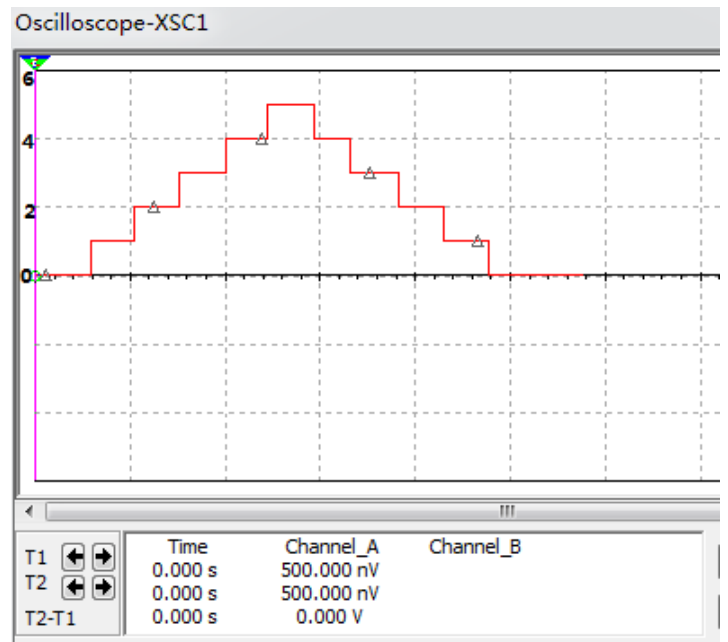


Figure 1-4 : Digital signal waveform generated by the source

Circuit Design and Simulation

After completing the design of the software and hardware part, we can carry out circuit simulation, code simulation, and circuit simulation according to the actual situation. The hardware circuit simulation is shown in Figure 1-5.

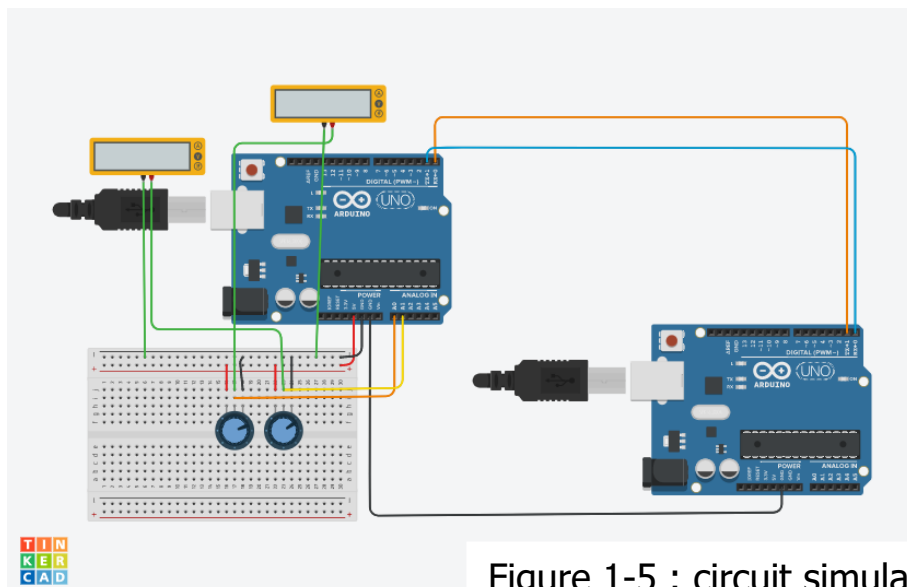


Figure 1-5 : circuit simulation

Design for Testing

Software and hardware testing process and result

Simulation test:

After completing the circuit simulation and the code simulation, it is possible to see whether the results are the same as expected at the output interface. The experimental reliability parameters include:

1. The actual analog voltage values sent by the oscilloscope are measured by an oscilloscope and compared with the received digital signals.

Experimental data is shown in chart 1-1, and the serial display data window is shown in Figure 1-6.

	Actual A0	Calculated A0	Actual A0	Calculated A0
Voltage	4.20V	4198mV	2.4V	2399mV
Voltage	2.50V	2497mV	3.60V	3597mV
Voltage	0.4V	400mV	1.90V	1897mV

Table 1-1 : experiment data

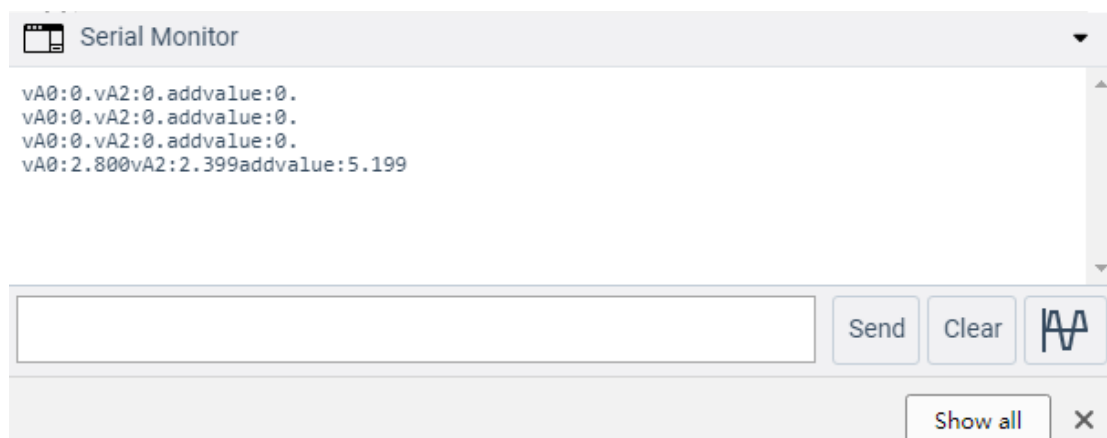


Figure 1-6 : serial display data window

Actual circuit test:

The actual circuit is built in the laboratory, and the actual circuit model is shown in Figure 1-7.

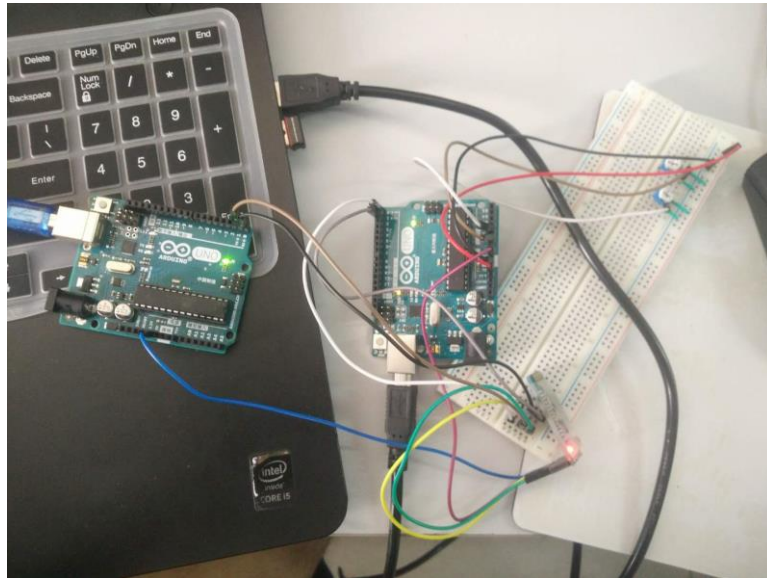


Figure 1-7 : circuit simulation

The specific testing process is consistent with the simulation test process, and the accuracy of measurement is greater than 99%.

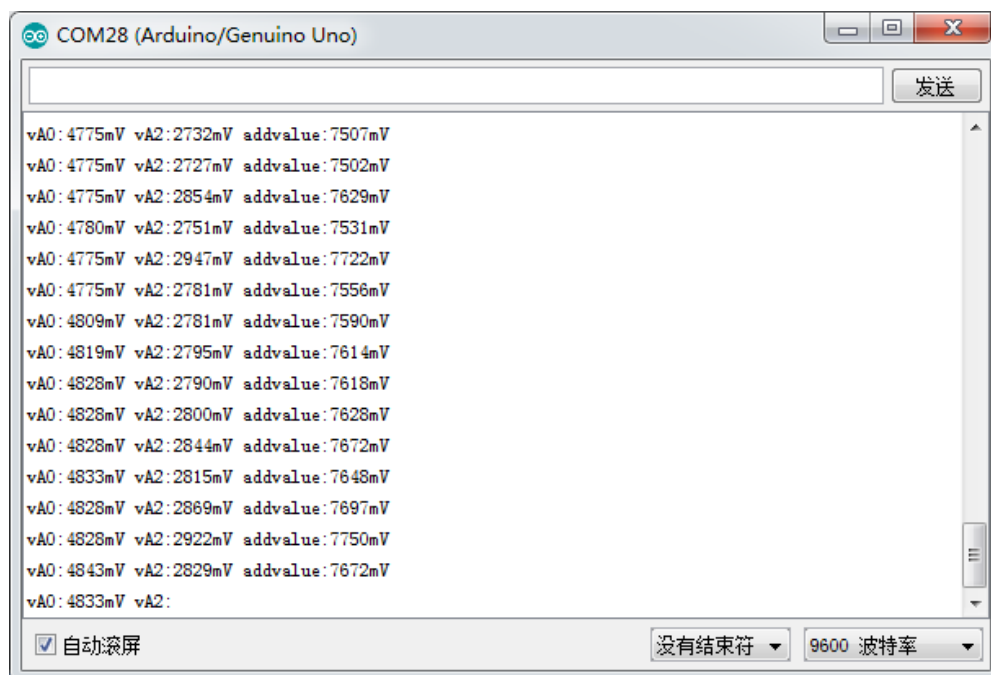


Figure 1-8 : Serial output results

At the same time, this design also adds an HC-05 Bluetooth module to view the measured data on the mobile client.

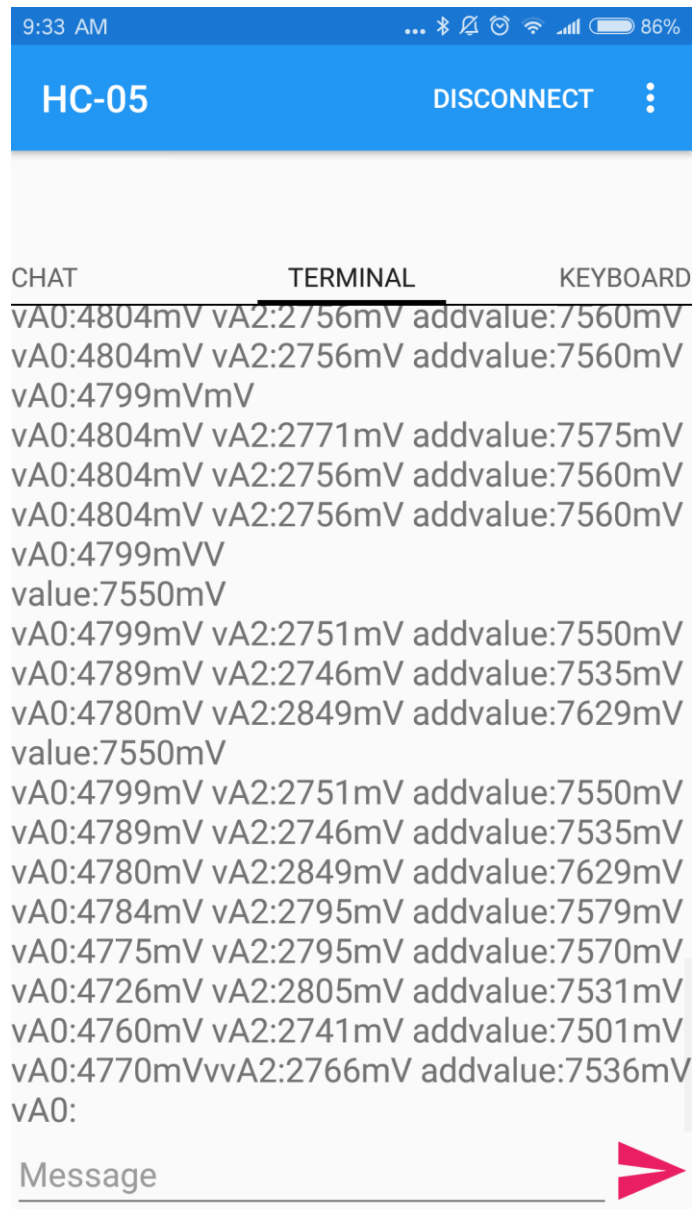


Figure 1-9 : Bluetooth serial port output on mobile

Conclusion and future recommendations

The experimental model completed the following functions perfectly and added a Bluetooth model as one of the indicators:

- (1) Reading two analog input port voltages with Arduino
- (2) Arduino internal ADC conversion based on Algorithms
- (3) Serial communication between two blocks of ArduinoBoard
- (4) There is an indicator to display the converted digital number

The design of the experiment fully realized the expected goal and realized the Bluetooth reading voltage value, which is more in line with the characteristics of mobile devices in today's society. It can be used in the following similar scenarios if it can continue to develop.

1. Using the voltage output sensor to measure the environmental index such as temperature and humidity.
2. Bluetooth-based Bluetooth module App development (such as Bluetooth unlocking).

Appendix: Implementation Code

1.ArduinoBoard1(Send code)

```
1. #define BAUD_PRESCALER 0x0067 //9600 baud Design By Wangzhe
2. 05151154
3. #include <stdio.h>
4. void USART_init(void)
5. {
6.     UBRR0 = BAUD_PRESCALER;
7.     UCSR0C = ((0<<USBS0)|(1<<UCSZ01)|(1<<UCSZ00));
8.     UCSR0B = ((1<<RXEN0)|(1<<TXEN0));
9. }
10.
11. void USART_send( unsigned char data)
12. {
```

```

13.    //while the transmit buffer is not empty loop  while(!(U
    CSR0A & (1<<UDRE0)));
14.    //when the buffer is empty write data to the transmitted

15.    UDR0 = data;
16.    }
17.
18.    void USART_putstring(char* StringPtr)
19.    {
20.        while(*StringPtr != 0x00)
21.        {
22.            USART_send(*StringPtr); StringPtr++;
23.            delay(200);
24.        }
25.    }
26.    void setup()
27.    {
28.        USART_init();
29.    }
30.
31.
32.
33.    // main
34.    void loop() {
35.        // analogread (A0)
36.        int sensorValue1 = analogRead(A0);
37.        int sensorValue2 = analogRead(A1);
38.        // 0-1023 transform to 0-5V
39.        int voltage1 = sensorValue1 * (5.0 / 1023.0)*1000;
40.        int voltage2 = sensorValue2 * (5.0 / 1023.0)*1000;
41.        int voltageadd = voltage1+voltage2;
42.        // send
43.        char s1[40];
44.        char s2[40];
45.        char s3[40];
46.        sprintf(s1,"%d",voltage1);
47.        sprintf(s2,"%d",voltage2);
48.        sprintf(s3,"%d",voltageadd);
49.        USART_putstring("vA0:");
50.        USART_putstring(s1);
51.        USART_putstring("mV");
52.        USART_putstring("vA2:");
53.        USART_putstring(s2);
54.        USART_putstring("mV");
55.        USART_putstring("addvalue:");
56.        USART_putstring(s3);
57.        USART_putstring("mV");
58.        USART_send('\n');
59.        delay(200);
60.
61.    }

```

2.ArduinoBoard2(Receive code)

```
1. #define BAUD_PRESCALER 0x0067 //9600 baud Design by Wangzhe05
   151154
2. void USART_init(void)
3. {UBRR0 = BAUD_PRESCALER;
4. UCSR0C = ((0<<USBS0)|(1<<UCSZ01)|(1<<UCSZ00));
5. UCSR0B = ((1<<RXEN0)|(1<<TXEN0));
6. }
7.
8. void USART_send( unsigned char data)
9. {
10.     //while the transmit buffer is not empty loop while(!(U
   CSR0A & (1<<UDRE0)));
11.     //when the buffer is empty write data to the transmitted
12.     UDR0 = data;
13. }
14.
15. void USART_putstr(char* StringPtr)
16. {
17.     while(*StringPtr != 0x00)
18.     {
19.         USART_send(*StringPtr);
20.         StringPtr++;
21.         delay(200);
22.     }
23. }
24.
25. void setup()
26. {
27.     USART_init();
28. }
29. unsigned char USART_Receive(void)
30. {
31.
32.     while (!(UCSR0A & (1<<RXC0)));
33.     /* Get and return received data from buffer */
34.     return UDR0;
35. }
36. void loop()
37. {
38.     USART_send( USART_Receive());
39. }
```