Fangyao Liu
No partner
Wednesday 3:30 P.M.
Sep.19, 2016

Exercise 3

Decimal	Binary	Analog Voltage
00	0000	0.303
01	0001	0.542
02	0010	0.874
03	0011	1.124
04	0100	1.456
05	0101	1.705
06	0110	2.038
07	0111	2.287
08	1000	2.385
09	1001	2.634
10	1010	2.966
11	1011	3.216
12	1100	3.553
13	1101	3.802
14	1110	4.134
15	1111	4.384

Exercise 4

Decimal	Binary	Analog Voltage
00	0000	0.303
01	0001	0.527
02	0010	0.869
03	0011	1.099
04	0100	1.436
05	0101	1.671
06	0110	2.003
07	0111	2.233
08	1000	2.438
09	1001	2.575
10	1010	2.629
11	1011	2.649
12	1100	2.668
13	1101	2.683
14	1110	2.697
15	1111	2.707

Exercise 5

Decimal	Binary	Analog Voltage
00	0000	2.297
01	0001	0.742
02	0010	0.742
03	0011	0.742
04	0100	1.520
05	0101	1.348
06	0110	1.505
07	0111	1.505
08	1000	2.463
09	1001	2.492
10	1010	2.864
11	1011	2.976
12	1100	3.147
13	1101	3.103
14	1110	3.103
15	1111	3.103

1. How does the buffer affect the results? What do you think the reason is?

   At first, the voltage grows steadily, reaches cutting off voltage and keeps the same. The reason is that buffer and diode consists a super diode.

Appendix

Exercise 1

```
//Fangyao Liu Exercise 1
//including necessary head file

#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>
#include "oi.h"

//define certain ports

#define USB 1
#define CR8 2

//function declration
```

```
void setSerial(uint8_t com);
uint8_t getSerialDestination(void);
void writeChar(char c, uint8_t com);
void delay(void);
void byteTx(uint8_t value);
void Init_Uart(void);
void SendStringtoPC(char *Message);


void Init_Uart(void)
{
    DDRC &= ~0x30;      //Set C4, C5 as input
    DIDR0 |= 0x30;      //disable digital input on C4, C5
    PRR &= ~0x01;       //disable adc power reduction
    ADCSRA = 0x87;      //enable ADC, prescale = 128
    ADMUX = 0x40;       //set voltage reference

    UBRR0 = 19;         //set the baud rate to 57600
    UCSR0B = 0x18;      //enable the transmit and receive functions of the serial port
    UCSR0C = 0x06;      //selects 8-bit data
    DDRB = 0X10;        //set pin B4 as an output
    PORTB = 0X10;       //set pin B4 to high
}

int intToChar(int meas_c4, char* Str)       //a function to transfer int number to string
{
    int p=meas_c4;
    int i = 0;
    char decitable[]="0123456789";      //create char table for transferring
    while(p != 0)                       //use module and division to get number on each digit
    {
        Str[i] = decitable[p%10];
        p = p/10;
        i++;
    }
    return i-1;                         //return how many digits are in this integer.
}

void main(void)
{

    Init_Uart();
    uint16_t meas_c4;
```

```
	char Str[4];
	int i=0;
	while(1)
	{
		ADMUX &= ~0x0F;				//clear the ADC channel
		ADMUX |= 0x04;				//set the ADC channel to C4
		ADCSRA |= 0x40;				//start the ADC measurment
		while(ADCSRA & 0x40);			//wait until it's done
		meas_c4 = ADC;				//save the result
		i=intToChar(meas_c4, Str);		//get digits of this integer
		SendStringtoPC("The measured value is:");		//send string to computer
		while(i>=0)
		{
			writeChar(Str[i],USB);		//transmit digit from high to low to computer
			i--;
		}
		writeChar(' ', USB);
		writeChar('\n', USB);
		writeChar('\r', USB);
	}
}


void delay(void)
{
int i=0,j=0;

	for(i=1;i<=1000;i++)
	{
		for(j=1;j<=1000;j++)
		{
		}
	}

}

//acquire the destination for WriteChar function

uint8_t getSerialDestination(void)
{
	if (PORTB & 0x10)
	return USB;
	else
	return CR8;
```

```
}

//set the communication interface

void setSerial(uint8_t com)
{
    if(com == USB)
    PORTB |= 0x10;
    else if(com == CR8)
    PORTB &= ~0x10;
}

//WriteChar function takes char and destination. It will use setSerial and getSerialDestination function to build communication interface
//and set com as USB, then transmit the byte. After that,

void writeChar(char c, uint8_t com)
{
    uint8_t originalDestination = getSerialDestination();
    if (com != originalDestination)
    {
    setSerial(com);
    delay();
    }

    byteTx((uint8_t)(c));
    if (com != originalDestination)
    {
        setSerial(originalDestination);
        delay();
    }
}

//When UCSR0A is ready, set UDR0 as the char which will be transmitted

void byteTx(uint8_t value)
{
    while(!(UCSR0A & 0x20)) ;
    UDR0 = value;
}

//Function to write string from robot to computer
void SendStringtoPC(char *Message)
{
    while(*Message)
    {
        while(!(UCSR0A & _BV(UDRE0))) ;
        UDR0 = *Message;
        Message++;
    }
}
```

Exercise 2

```
//Fangyao Liu Exercise 2
#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>
#include "oi.h"

#define USB 1
#define CR8 2

void setSerial(uint8_t com);
uint8_t getSerialDestination(void);
void writeChar(char c, uint8_t com);
void delay(void);
void byteTx(uint8_t value);
void Init_Uart(void);
void SendStringtoPC(char *Message);


void Init_Uart(void)
{
    DDRC &= ~0x30;
    DIDR0 |= 0x30;
    PRR &= ~0x01;
    ADCSRA = 0x87;
    ADMUX = 0x40;

    UBRR0 = 19;
    UCSR0B = 0x18;
    UCSR0C = 0x06;
    DDRB = 0X10;
    PORTB = 0X10;
}

void intToChar(int meas_c4, char* Str)
{
    int p=meas_c4;
    int i = 0;
    char decitable[]="0123456789";
    while(i != 4)
    {
        Str[i] = decitable[p%10];
        p = p/10;
        i++;
    }
}

void main(void)
{

    Init_Uart();
    uint64_t meas_c4;    //define meas_c4 as uint64_t, because a large number is going to be
assigned to meas_c4
    char Str[4];
    int i;
    while(1)
    {
        i=3;
        ADMUX &= ~0x0F;
        ADMUX |= 0x04;
        ADCSRA |= 0x40;
        while(ADCSRA & 0x40);
        meas_c4 = (uint64_t) ADC*5000/1023;    //transfer steps into certain voltage. cast form
into uint64_t so that it won't overflow
        intToChar(meas_c4, Str);
        SendStringtoPC("The analog input voltage is:");
        while(i>=0)
        {
            writeChar(Str[i],USB);
            if (i==3) writeChar('.', USB);
            i--;
        }
        writeChar(' ', USB);
        writeChar('\n', USB);
        writeChar('\r', USB);
    }
}


void delay(void)
```

```
{
int i=0,j=0;

        for(i=1;i<=1000;i++)
        {
                for(j=1;j<=1000;j++)
                {
                }
        }

}



uint8_t getSerialDestination(void)
{
        if (PORTB & 0x10)
        return USB;
        else
        return CR8;
}




void setSerial(uint8_t com)
{
        if(com == USB)
        PORTB |= 0x10;
        else if(com == CR8)
        PORTB &= ~0x10;
}




void writeChar(char c, uint8_t com)
{
        uint8_t originalDestination = getSerialDestination();
        if (com != originalDestination)
        {
        setSerial(com);
        delay();
        }

        byteTx((uint8_t)(c));
        if (com != originalDestination)
```

```
	{
		setSerial(originalDestination);
		delay();
	}
}


void byteTx(uint8_t value)
{
	while(!(UCSR0A & 0x20)) ;
	UDR0 = value;
}

void SendStringtoPC(char *Message)
{
	while(*Message)
	{
		while(!(UCSR0A & _BV(UDRE0))) ;
		UDR0 = *Message;
		Message++;
	}
}
```

Exercise 3

```
//Fangyao Liu Exercise 3
#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>
#include "oi.h"

#define USB 1
#define CR8 2

void setSerial(uint8_t com);
uint8_t getSerialDestination(void);
void writeChar(char c, uint8_t com);
void delay(void);
void byteTx(uint8_t value);
void Init_Uart(void);
void SendStringtoPC(char *Message);


void Init_Uart(void)
{
	DDRC |= 0x0F;		//set PC0,PC1,PC2,PC3 as output
	DDRC &= ~0x30;
	DIDR0 |= 0x30;
	PRR &= ~0x01;
	ADCSRA = 0x87;
	ADMUX = 0x40;

	UBRR0 = 19;
	UCSR0B = 0x18;
	UCSR0C = 0x06;
	DDRB = 0X10;
	PORTB = 0X10;
}

void intToChar(int meas_c4, char* Str)
{
	int p=meas_c4;
	int i = 0;
	char decitable[]="0123456789";
	while(i != 4)
	{
		Str[i] = decitable[p%10];
		p = p/10;
		i++;
	}
}

void intToBi(int k, char* Str)		//function that transfer integer to binary string
{
	char bitable[]="01";
	int i = 0;
	while(i!=4)
	{
		Str[i] = bitable[k%2];
		k=k/2;
		i++;
	}

}

void main(void)
{

	Init_Uart();
```

```
    uint64_t meas_c4;
    char Str[4];
    int i;
    int k;
    int p;
    k=0;
    PORTC &= 0xF0;                    //Initialize and mask different bits of port c
    while(1)
    {

        if(k>15) k=0;             //If k reaches (10000)B, change k to (0000)B again.
        PORTC |= k;                   //Set PC 3,2,1,0 as k
        SendStringtoPC("The analog input voltage for ");
        intToBi(k, Str);          //change k into binary string
        i=3;
        while(i>=0)
        {
            writeChar(Str[i],USB);
            i--;
        }
        k++;

        ADMUX &= ~0x0F;
        ADMUX |= 0x04;
        ADCSRA |= 0x40;
        while(ADCSRA & 0x40);
        SendStringtoPC(" is: ");
        meas_c4 = (uint64_t) ADC*5000/1023;
        intToChar(meas_c4, Str);

        i=3;
        while(i>=0)
        {
            writeChar(Str[i],USB);
            if (i==3) writeChar('.', USB);
            i--;
        }
        writeChar('\n', USB);
        writeChar('\r', USB);
        for(p=0;p<100;p++)        //delay 1 second
        _delay_loop_2(46080);
        PORTC &= 0xF0;            //clear PC 3,2,1,0 to zero
    }
}
```

```
void delay(void)
{
    int i=0,j=0;

    for(i=1;i<=1000;i++)
    {
        for(j=1;j<=1000;j++)
        {
        }
    }

}


uint8_t getSerialDestination(void)
{
    if (PORTB & 0x10)
    return USB;
    else
    return CR8;
}


void setSerial(uint8_t com)
{
    if(com == USB)
    PORTB |= 0x10;
    else if(com == CR8)
    PORTB &= ~0x10;
}


void writeChar(char c, uint8_t com)
{
    uint8_t originalDestination = getSerialDestination();
    if (com != originalDestination)
    {
    setSerial(com);
    delay();
    }

    byteTx((uint8_t)(c));
    if (com != originalDestination)
    {
        setSerial(originalDestination);
        delay();
    }
}


void byteTx(uint8_t value)
{
    while(!(UCSR0A & 0x20)) ;
    UDR0 = value;
}

void SendStringtoPC(char *Message)
{
    while(*Message)
    {
         while(!(UCSR0A & _BV(UDRE0))) ;
        UDR0 = *Message;
        Message++;
    }
}
```