Fangyao Liu
No partner
Wednesday 3:30 P.M.
Date: Sep. 24, 2016

Exercise 1
Part A

3 What does the function “**powerOnRobot**” do?
To turn on the robot. Because the robot’s on/off state will toggle whenever there is a low to high transition on D7. Use RobotPwrToggleLow and RobotPwrToggleHigh to turn on the robot. Then set D7 back to low again.

4 Explain in detail the working of the main function.
Please see comments of the code.

Part B

Exercise

<u>Appendix</u>

Exercise 1
Part A

```
//Fangyao Liu Exercise 1 Part A
//Include enough head files
#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>
#include "oi.h"

//Function declarations
void initialize(void);
void powerOnRobot(void);
void baud28k(void);
void delay10ms(uint16_t delay_10ms);
uint8_t byteRx(void);
void flushRx(void);
void Move_Forward(void);
void Move_Stop(void);
void byteTx(uint8_t value);


int main (void)
{
   // Initialize the microcontroller
   initialize();
   // Turn on the Create power if off
   powerOnRobot();

   // Start the open interface
   byteTx(CmdStart);

   // Change to 28800 baud
   baud28k();

   // Take full control of the Create
   byteTx(CmdFull);

   //Get rid of unwanted data in the serial port receiver
   flushRx();

   Move_Forward();
   Move_Stop();

}

void Move_Forward(void)
{
      byteTx(137);    // drive opcode

      //Go forward 100 mm/s
      byteTx(0x00); // velocity high byte
      byteTx(0x64); // velocity low byte

      //Go in a straight line
      byteTx(0x80); // radius high byte
      byteTx(0x00); // radius low byte
      delay10ms(300);
}


void Move_Stop(void)
{
      byteTx(137);    // drive opcode
      //Stop the robot
      byteTx(0x00); // velocity high byte
      byteTx(0x00); // velocity low byte
      //Go in a straight line
      byteTx(0x80); // radius high byte
```

```
    byteTx(0x00); // radius low byte
}


void initialize(void)
{
  // Turn off interrupts
  cli();

  // configure the I/O pins
  DDRB = 0x10;
  PORTB = 0xCF;
  DDRC = 0x02;
  PORTC = 0xFF;
  DDRD = 0xE6;
  PORTD = 0x7D;

  // Set up the serial port for 57600 baud
  UBRR0 = Ubrr57600;
  UCSR0B = (_BV(TXEN0) | _BV(RXEN0));
  UCSR0C = (_BV(UCSZ00) | _BV(UCSZ01));
}

void powerOnRobot(void)
{
  // If Create's power is off, turn it on
  if(!RobotIsOn)
  {
      while(!RobotIsOn)
      {
          RobotPwrToggleLow;
          delay10ms(50);    // Delay in this state
          RobotPwrToggleHigh;    // Low to high transition to toggle power
          delay10ms(10);    // Delay in this state
          RobotPwrToggleLow;
      }
      delay10ms(350);    // Delay for startup
  }
}

void baud28k(void)
{
  // Send the baud change command for 28800 baud
  byteTx(CmdBaud);
```

```
   byteTx(Baud28800);

   // Wait while until the command is sent
   while(!(UCSR0A & _BV(TXC0))) ;

   // Change the atmel's baud rate
   UBRR0 = Ubrr28800;

   // Wait 100 ms
   delay10ms(10);
}

void delay10ms(uint16_t delay_10ms)
{
   // Delay for (delay_10ms * 10) ms
   while(delay_10ms-- > 0)
   {
      // Call a 10 ms delay loop
      _delay_loop_2(46080);
   }
}

uint8_t byteRx(void)
{
   // Receive a byte over the serial port (UART)
   while(!(UCSR0A & _BV(RXC0))) ;
   return UDR0;
}

void flushRx(void)
{
   uint8_t temp;

   // Clear the serial port
   while(UCSR0A & _BV(RXC0))
      temp = UDR0;
}

void byteTx(uint8_t value)
{
   // Send a byte over the serial port
   while(!(UCSR0A & _BV(UDRE0))) ;
   UDR0 = value;
}
```

Part B

```
//Fangyao Liu Exercise 1 Part B
#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>
#include "oi.h"

void initialize(void);
void powerOnRobot(void);
void baud28k(void);
void delay10ms(uint16_t delay_10ms);
uint8_t byteRx(void);
void flushRx(void);
void Move_Forward(void);
void Move_Stop(void);
void byteTx(uint8_t value);
void Move_Backward(void);

int main (void)
{
   // Initialize the microcontroller
   initialize();

   // Turn on the Create power if off
   powerOnRobot();

   // Start the open interface
   byteTx(CmdStart);

   // Change to 28800 baud
   baud28k();

   // Take full control of the Create
   byteTx(CmdFull);

   //Get rid of unwanted data in the serial port receiver
   flushRx();

   Move_Forward();
   Move_Stop();

   Move_Backward();
   Move_Stop();
}

void Move_Forward(void)
{
        byteTx(137);    // drive opcode

        //Go forward 100 mm/s
        byteTx(0x00); // velocity high byte
        byteTx(0x64); // velocity low byte

        //Go in a straight line
        byteTx(0x80); // radius high byte
        byteTx(0x00); // radius low byte
        delay10ms(300);
}

void Move_Backward(void)
{
       byteTx(137);    // drive opcode

       //Go backward 100 mm/s
       byteTx(0xFF); // velocity high byte
        byteTx(0x9C); // velocity low byte

       //Go in a straight line
       byteTx(0x80); //radius high byte
       byteTx(0x00); //radius low byte
       delay10ms(300);
}


void Move_Stop(void)
{
        byteTx(137);    // drive opcode
        //Stop the robot
        byteTx(0x00); // velocity high byte
        byteTx(0x00); // velocity low byte
        //Go in a straight line
        byteTx(0x80); // radius high byte
        byteTx(0x00); // radius low byte
}


void initialize(void)
```

```
{
   // Turn off interrupts
   cli();

   // configure the I/O pins
   DDRB = 0x10;
   PORTB = 0xCF;
   DDRC = 0x02;
   PORTC = 0xFF;
   DDRD = 0xE6;
   PORTD = 0x7D;

   // Set up the serial port for 57600 baud
   UBRR0 = Ubrr57600;
   UCSR0B = (_BV(TXEN0) | _BV(RXEN0));
   UCSR0C = (_BV(UCSZ00) | _BV(UCSZ01));
}

void powerOnRobot(void)
{
   // If Create's power is off, turn it on
   if(!RobotIsOn)
   {
         while(!RobotIsOn)
         {
               RobotPwrToggleLow;
               delay10ms(50);    // Delay in this state
               RobotPwrToggleHigh;    // Low to high transition to toggle power
               delay10ms(10);    // Delay in this state
               RobotPwrToggleLow;
         }
         delay10ms(350);    // Delay for startup
   }
}

void baud28k(void)
{
   // Send the baud change command for 28800 baud
   byteTx(CmdBaud);
   byteTx(Baud28800);

   // Wait while until the command is sent
   while(!(UCSR0A & _BV(TXC0))) ;
```

```
    // Change the atmel's baud rate
    UBRR0 = Ubrr28800;

    // Wait 100 ms
    delay10ms(10);
}

void delay10ms(uint16_t delay_10ms)
{
    // Delay for (delay_10ms * 10) ms
    while(delay_10ms-- > 0)
    {
        // Call a 10 ms delay loop
        _delay_loop_2(46080);
    }
}

uint8_t byteRx(void)
{
    // Receive a byte over the serial port (UART)
    while(!(UCSR0A & _BV(RXC0))) ;
    return UDR0;
}

void flushRx(void)
{
    uint8_t temp;

    // Clear the serial port
    while(UCSR0A & _BV(RXC0))
        temp = UDR0;
}

void byteTx(uint8_t value)
{
    // Send a byte over the serial port
    while(!(UCSR0A & _BV(UDRE0))) ;
    UDR0 = value;
}
```

Exercise 2

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

void initialize(void);
void powerOnRobot(void);
void baud28k(void);
void delay10ms(uint16_t delay_10ms);
uint8_t byteRx(void);
void flushRx(void);
void Move_Forward(void);
void Move_Stop(void);
void byteTx(uint8_t value);
void Move_Backward(void);
void Turn_Right(void);

int main (void)
{
   int i=0;

   // Initialize the microcontroller
   initialize();

   // Turn on the Create power if off
   powerOnRobot();

   // Start the open interface
   byteTx(CmdStart);

   // Change to 28800 baud
   baud28k();

   // Take full control of the Create
   byteTx(CmdFull);

   //Get rid of unwanted data in the serial port receiver
   flushRx();

for(i=0;i<4;i++)      // loop four times to move in a square path
{
     Move_Forward();
```

```
    Move_Stop();

    Turn_Right();
    Move_Stop();
}

}

void Move_Forward(void)
{
     byteTx(137);    // drive opcode

     //Go forward 100 mm/s
     byteTx(0x00); // velocity high byte
     byteTx(0x64); // velocity low byte

     //Go in a straight line
     byteTx(0x80); // radius high byte
     byteTx(0x00); // radius low byte
     delay10ms(1000);
}

void Move_Backward(void)
{
    byteTx(137);    // drive opcode

    //Go backward 100 mm/s
    byteTx(0xFF); // velocity high byte
     byteTx(0x9C); // velocity low byte

    //Go in a straight line
    byteTx(0x80); //radius high byte
    byteTx(0x00); //radius low byte
    delay10ms(1000);
}

void Turn_Right(void)
{
    byteTx(137);    //drive opcode

    //100 mm/s velocity
    byteTx(0x00); // velocity high byte
     byteTx(0x64); // velocity low byte
```

```
      //turn right 90 degrees
      byteTx(0xFF);
      byteTx(0xFF);
      delay10ms(205);
}


void Move_Stop(void)
{
       byteTx(137);    // drive opcode
       //Stop the robot
       byteTx(0x00); // velocity high byte
       byteTx(0x00); // velocity low byte
       //Go in a straight line
       byteTx(0x80); // radius high byte
       byteTx(0x00); // radius low byte
}


void initialize(void)
{
   // Turn off interrupts
   cli();

   // configure the I/O pins
   DDRB = 0x10;
   PORTB = 0xCF;
   DDRC = 0x02;
   PORTC = 0xFF;
   DDRD = 0xE6;
   PORTD = 0x7D;

   // Set up the serial port for 57600 baud
   UBRR0 = Ubrr57600;
   UCSR0B = (_BV(TXEN0) | _BV(RXEN0));
   UCSR0C = (_BV(UCSZ00) | _BV(UCSZ01));
}

void powerOnRobot(void)
{
   // If Create's power is off, turn it on
   if(!RobotIsOn)
   {
         while(!RobotIsOn)
```

```
            {
                  RobotPwrToggleLow;
                  delay10ms(50);    // Delay in this state
                  RobotPwrToggleHigh;    // Low to high transition to toggle power
                  delay10ms(10);    // Delay in this state
                  RobotPwrToggleLow;
            }
            delay10ms(350);    // Delay for startup
      }
}

void baud28k(void)
{
   // Send the baud change command for 28800 baud
   byteTx(CmdBaud);
   byteTx(Baud28800);

   // Wait while until the command is sent
   while(!(UCSR0A & _BV(TXC0))) ;

   // Change the atmel's baud rate
   UBRR0 = Ubrr28800;

   // Wait 100 ms
   delay10ms(10);
}

void delay10ms(uint16_t delay_10ms)
{
   // Delay for (delay_10ms * 10) ms
   while(delay_10ms-- > 0)
   {
      // Call a 10 ms delay loop
      _delay_loop_2(46080);
   }
}

uint8_t byteRx(void)
{
   // Receive a byte over the serial port (UART)
   while(!(UCSR0A & _BV(RXC0))) ;
   return UDR0;
}
```

```
void flushRx(void)
{
   uint8_t temp;

   // Clear the serial port
   while(UCSR0A & _BV(RXC0))
      temp = UDR0;
}

void byteTx(uint8_t value)
{
   // Send a byte over the serial port
   while(!(UCSR0A & _BV(UDRE0))) ;
   UDR0 = value;
}
```

Exercise 3

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

void initialize(void);
void powerOnRobot(void);
void baud28k(void);
void delay10ms(uint16_t delay_10ms);
uint8_t byteRx(void);
void flushRx(void);
void Move_Forward(void);
void Move_Stop(void);
void byteTx(uint8_t value);
void Move_Backward(void);
void Turn_Right(void);
void Turn_Left(void);

int main (void)
{
   int i=0;

   // Initialize the microcontroller
   initialize();

   // Turn on the Create power if off
```

```
    powerOnRobot();

    // Start the open interface
    byteTx(CmdStart);

    // Change to 28800 baud
    baud28k();

    // Take full control of the Create
    byteTx(CmdFull);

    //Get rid of unwanted data in the serial port receiver
    flushRx();

while(1)    //keep moving in a "8" path
{
        Move_Forward();
       Move_Stop();

       Turn_Left();
       Move_Stop();

       Move_Forward();
       Move_Stop();

       Turn_Right();
       Move_Stop();
}

}

void Move_Forward(void)
{
        byteTx(137);    // drive opcode

        //Go forward 100 mm/s
        byteTx(0x01); // velocity high byte
        byteTx(0xF4); // velocity low byte

        //Go in a straight line
        byteTx(0x80); // radius high byte
        byteTx(0x00); // radius low byte
        delay10ms(120);
}
```

```
void Move_Backward(void)
{
      byteTx(137);    // drive opcode

      //Go backward 100 mm/s
      byteTx(0xFF); // velocity high byte
       byteTx(0x9C); // velocity low byte

      //Go in a straight line
      byteTx(0x80); //radius high byte
      byteTx(0x00); //radius low byte
      delay10ms(1000);
}

void Turn_Right(void)
{
      byteTx(137);    //drive opcode

      //0 velocity
      byteTx(0x00); // velocity high byte
       byteTx(0xC8); // velocity low byte

      //turn right at r=500mm
      byteTx(0x01);
      byteTx(0xF4);
      delay10ms(1100);
}

void Turn_Left(void)
{
      byteTx(137);    //drive opcode

      //0 velocity
      byteTx(0x00); // velocity high byte
       byteTx(0xC8); // velocity low byte

      //turn left at r=500mm
      byteTx(0xFE);
      byteTx(0x0C);
      delay10ms(1100);
}
```

```
void Move_Stop(void)
{
        byteTx(137);    // drive opcode
        //Stop the robot
        byteTx(0x00); // velocity high byte
        byteTx(0x00); // velocity low byte
        //Go in a straight line
        byteTx(0x80); // radius high byte
        byteTx(0x00); // radius low byte
}


void initialize(void)
{
    // Turn off interrupts
    cli();

    // configure the I/O pins
    DDRB = 0x10;
    PORTB = 0xCF;
    DDRC = 0x02;
    PORTC = 0xFF;
    DDRD = 0xE6;
    PORTD = 0x7D;

    // Set up the serial port for 57600 baud
    UBRR0 = Ubrr57600;
    UCSR0B = (_BV(TXEN0) | _BV(RXEN0));
    UCSR0C = (_BV(UCSZ00) | _BV(UCSZ01));
}

void powerOnRobot(void)
{
    // If Create's power is off, turn it on
    if(!RobotIsOn)
    {
            while(!RobotIsOn)
            {
                    RobotPwrToggleLow;
                    delay10ms(50);    // Delay in this state
                    RobotPwrToggleHigh;    // Low to high transition to toggle power
                    delay10ms(10);    // Delay in this state
                    RobotPwrToggleLow;
            }
```

```
         delay10ms(350);    // Delay for startup
    }
}

void baud28k(void)
{
    // Send the baud change command for 28800 baud
    byteTx(CmdBaud);
    byteTx(Baud28800);

    // Wait while until the command is sent
    while(!(UCSR0A & _BV(TXC0))) ;

    // Change the atmel's baud rate
    UBRR0 = Ubrr28800;

    // Wait 100 ms
    delay10ms(10);
}

void delay10ms(uint16_t delay_10ms)
{
    // Delay for (delay_10ms * 10) ms
    while(delay_10ms-- > 0)
    {
        // Call a 10 ms delay loop
        _delay_loop_2(46080);
    }
}

uint8_t byteRx(void)
{
    // Receive a byte over the serial port (UART)
    while(!(UCSR0A & _BV(RXC0))) ;
    return UDR0;
}

void flushRx(void)
{
    uint8_t temp;

    // Clear the serial port
    while(UCSR0A & _BV(RXC0))
        temp = UDR0;
```

```
}

void byteTx(uint8_t value)
{
   // Send a byte over the serial port
   while(!(UCSR0A & _BV(UDRE0))) ;
   UDR0 = value;
}
```