

#define FOSC 16000000 // Clock Speed

#define BAUD 9600

#define MYUBRR FOSC/16/BAUD-1

void USART\_Init(unsigned int ubrr)

{

/\*Set baud rate \*/

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

/\*Enable receiver and transmitter \*/

UCSR0B = (1<<RXEN0)|(1<<TXEN0);

/\* Set frame format: 8data, 2stop bit \*/

UCSR0C = (1<<USBS0)|(3<<UCSZ00);

}

void USART\_Transmit(char b)

{

while (!(UCSR0A & (1<<UDRE0)));

UDR0 = b;

}

unsigned char USART\_Receive(void)

{

/\* Wait for data to be received \*/

while (!(UCSR0A & (1<<RXC0)));

/\* Get and return received data from buffer \*/

return UDR0;

}

void USART\_Flush() //USART function to clear the data register

{

unsigned char dummy;

while (UCSR0A & (1<<RXC0)) dummy = UDR0;

}

void dc\_pwm(int OP)

{

DDRD |= (1<<PD6)|(1<<PD4); //Fast PWM output at OC0A pin: PD6

PORTD&=~(1<<4);

TCCR0A |= (1<<COM0A1) | (1<<WGM01) | (1<<WGM00); //Non-Inverting Fast PWM mode 3 using OCRA unit

TCCR0B |= (1<<CS00); //No-Prescalar

OCR0A =((255\*OP)/100);

}

//2.

void dc\_motor()

{

int i,k,l;

char q[2];

char a[]="DC motor selected: ";

char b[]="Enter the duty cycle: ";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

\_delay\_ms(100);

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;b[i]!='\0';i++)

{

USART\_Transmit(b[i]);

\_delay\_ms(100);

}

for(k=0;k<2;k++)

{

q[k]=USART\_Receive();

USART\_Transmit(q[k]);

}

q[k]='\0';

l=atoi(q);

dc\_pwm(l);

}

//

void stepper\_motor()

{

int i,m,k,l;

char q[3];

char c[]="Stepper is selected: ";

char d[]="Enter the angle: ";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;c[i]!='\0';i++)

{

USART\_Transmit(c[i]);

\_delay\_ms(100);

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;d[i]!='\0';i++)

{

USART\_Transmit(d[i]);

\_delay\_ms(100);

}

for(k=0;k<3;k++)

{

q[k]=USART\_Receive();

USART\_Transmit(q[k]);

}

q[k]='\0';

l=atoi(q);

stepper\_clock(l);

}

void stepper\_clock(int r)

{

int i;

DDRD=(0xf);

PORTD&=~(0xf);

for(i=0;i<(r/7.2);i++)

{

PORTB=(1<<0);

\_delay\_ms(100);

PORTB&=~(1<<0);

PORTB=(1<<1);

\_delay\_ms(100);

PORTB&=~(1<<1);

PORTB=(1<<2);

\_delay\_ms(100);

PORTB&=~(1<<2);

PORTB=(1<<3);

\_delay\_ms(100);

PORTB&=~(1<<3);

}

}

//4

void manual\_mode()

{

int i;

char a[]="Enter the motor to select: ";

char b[]="Enter s for steppermotor d for dc motor: "; char d[]="Do you want to continue any motor: ";

char ch,ch1; char v[]="No motors ";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

\_delay\_ms(100);

}

l1: USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;b[i]!='\0';i++)

{

USART\_Transmit(b[i]);

\_delay\_ms(100);

}

ch=USART\_Receive();

USART\_Transmit(ch);

if(ch=='s')

{

stepper\_motor();

}

else if(ch=='d')

{

dc\_motor();

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;d[i]!='\0';i++)

{

USART\_Transmit(d[i]);

\_delay\_ms(100);

}

ch1=USART\_Receive();

USART\_Transmit(ch1);

if(ch1=='y') goto l1;

else

{

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;v[i]!='\0';i++)

{

USART\_Transmit(v[i]);

\_delay\_ms(100);

}

}

}

int main()

{

USART\_Init(MYUBRR);

int i;

char p[]="select a mode: ";

char q[]="Enter a for auto mode b for manual mode: ";

char r;

USART\_Transmit('\r'); USART\_Transmit('\n');

for( i=0;p[i]!='\0';i++)

{

USART\_Transmit(p[i]);

\_delay\_ms(100);

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;q[i]!='\0';i++)

{

USART\_Transmit(q[i]);

\_delay\_ms(100);

}

r=USART\_Receive();

USART\_Transmit(r);

//if(r=='a')

// active\_mode();

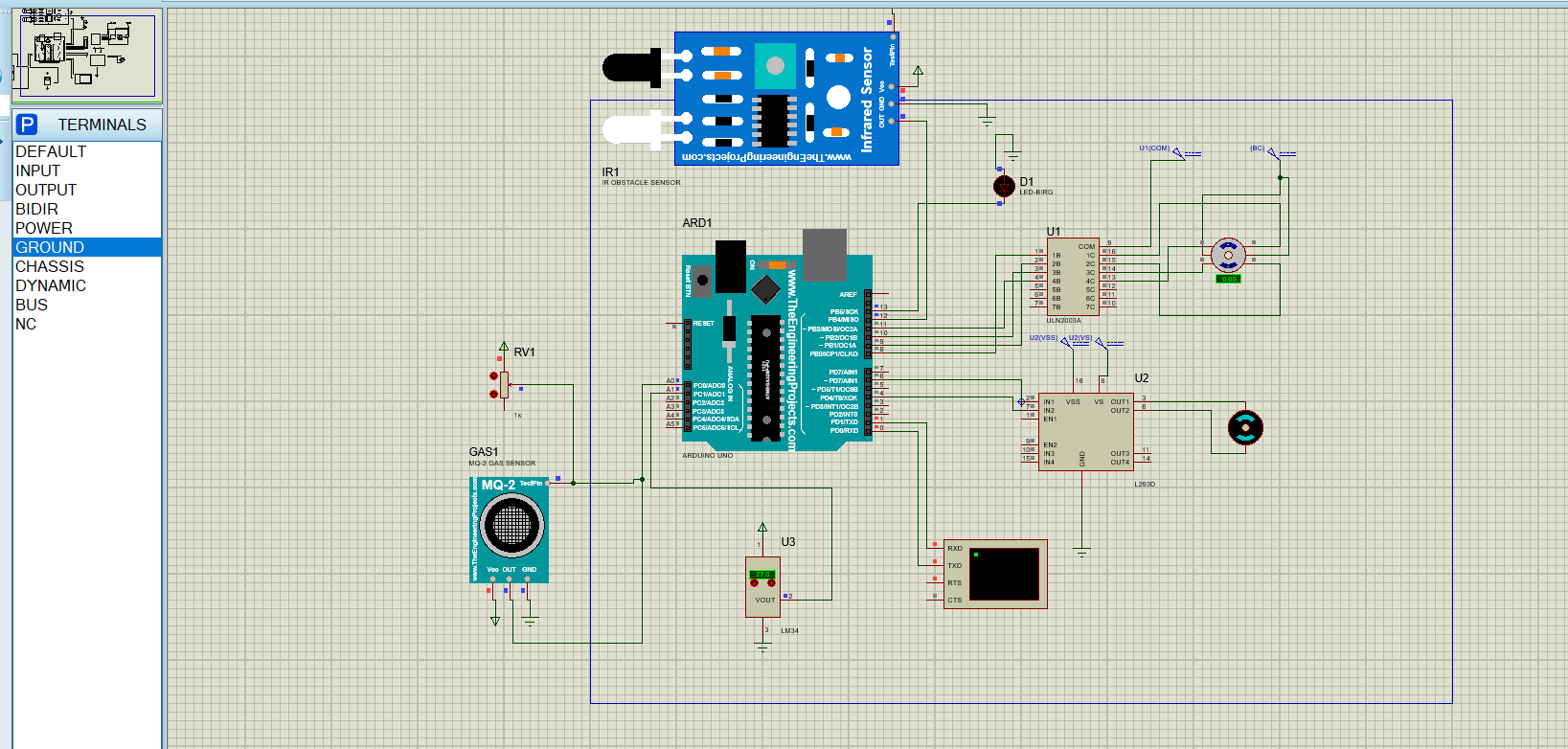
if(r=='b')

{

manual\_mode();

}

}



#define FOSC 16000000

#define BAUD 9600

#define MYUBRR FOSC/16/BAUD-1

void USART\_Init(unsigned int ubrr)

{

/\*Set baud rate \*/

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

/\*Enable receiver and transmitter \*/

UCSR0B = (1<<RXEN0)|(1<<TXEN0);

/\* Set frame format: 8data, 2stop bit \*/

UCSR0C = (1<<USBS0)|(3<<UCSZ00);

}

void USART\_Transmit(char b)

{

while (!(UCSR0A & (1<<UDRE0)));

UDR0 = b;

}

unsigned char USART\_Receive(void)

{

/\* Wait for data to be received \*/

while (!(UCSR0A & (1<<RXC0)));

/\* Get and return received data from buffer \*/

return UDR0;

}

void USART\_Flush() //USART function to clear the data register

{

unsigned char dummy;

while (UCSR0A & (1<<RXC0)) dummy = UDR0;

}

void dc\_pwm(int OP)

{

DDRD |= (1<<PD6)|(1<<PD4); //Fast PWM output at OC0A pin: PD6

PORTD&=~(1<<4);

TCCR0A |= (1<<COM0A1) | (1<<WGM01) | (1<<WGM00); //Non-Inverting Fast PWM mode 3 using OCRA unit

TCCR0B |= (1<<CS00); //No-Prescalar

OCR0A =((255\*OP)/100);

}

void dc\_motor()

{

int i,k,r;

char q[2];

char a[]="DC motor selected";

char b[]="Enter the duty cycle";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

\_delay\_ms(10);

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;b[i]!='\0';i++)

{

USART\_Transmit(b[i]);

\_delay\_ms(10);

}

for(k=0;k<2;k++)

{

q[k]=USART\_Receive();

USART\_Transmit(q[k]);

}

q[k]='\0';

r=atoi(q);

dc\_pwm(r);

}

void step\_motor()

{

int i,r,k;

char q[3];

char c[]="stepper motor selected";

char d[]="Enter the angle";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;c[i]!='\0';i++)

{

USART\_Transmit(c[i]);

\_delay\_ms(10);

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;d[i]!='\0';i++)

{

USART\_Transmit(d[i]);

\_delay\_ms(10);

}

for(k=0;k<3;k++)

{

q[k]=USART\_Receive();

USART\_Transmit(q[k]);

}

q[k]='\0';

r=atoi(q);

step\_clock(r);

}

void step\_clock(int r)

{

int i;

DDRB=(0X1e);

PORTB&=~(0X1e);

for(i=0;i<(r/7.2);i++)

{

PORTB=(1<<0);

\_delay\_ms(100);

PORTB&=~(1<<0);

PORTB=(1<<1);

\_delay\_ms(100);

PORTB&=~(1<<1);

PORTB=(1<<2);

\_delay\_ms(100);

PORTB&=~(1<<2);

PORTB=(1<<3);

\_delay\_ms(100);

PORTB=~(1<<3);

}

}

void step\_anti(int r)

{

int i;

DDRB=(0X1e);

PORTB&=~(0X1e);

for(i=0;i<(r/7.2);i++)

{

PORTB=(1<<3);

\_delay\_ms(100);

PORTB&=~(1<<3);

PORTB=(1<<2);

\_delay\_ms(100);

PORTB&=~(1<<2);

PORTB=(1<<1);

\_delay\_ms(100);

PORTB&=~(1<<1);

PORTB=(1<<0);

\_delay\_ms(100);

PORTB=~(1<<0);

}

}

void manual\_mode()

{

int i;

char a[]="enter the motor to select ";

char b[]="Enter s for steppermotor d for dc motor ";

char s[]="Do you want to continue any motor y/n: "; char t[]="No motors";

char c,d;

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

\_delay\_ms(10);

}

l1: USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;b[i]!='\0';i++)

{

USART\_Transmit(b[i]);

\_delay\_ms(10);

}

c=USART\_Receive();

USART\_Transmit(c);

if(c=='s')

{

step\_motor();

}

if(c=='d')

{

dc\_motor();

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;s[i]!='\0';i++)

{

USART\_Transmit(s[i]);

\_delay\_ms(10);

}

d=USART\_Receive();

USART\_Transmit(d);

if(d=='y') goto l1;

else

{

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;t[i]!='\0';i++)

{

USART\_Transmit(t[i]);

\_delay\_ms(10);

}

}

}

void temperature\_sensor()

{

int i,x;

char res[5];

char a[]="threshold of temperature=26 c";

char b[]="Temperature is:";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

\_delay\_ms(10);

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;b[i]!='\0';i++)

{

USART\_Transmit(b[i]);

\_delay\_ms(10);

}

ADMUX=(0x01);

ADMUX=(1<<6);

ADCSRA=(7<<0);

ADCSRA|=(1<<7);

ADCSRA|=(1<<6);

while((ADCSRA & (1<<4))==0);

x=ADC;

x=((x\*5\*100)/1024);

sprintf(res,"%d",x);

for(i=0;res[i]!='\0';i++)

{

USART\_Transmit(res[i]);

}

if(x>15)

{

dc\_pwm(90);

}

else

dc\_pwm(60);

}

void gas\_sensore()

{

int x,i;

char res[5]; char a[]="Threshold of gas is 4%"; char b[]="Gas is : ";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

\_delay\_ms(10);

}

USART\_Transmit('\n'); USART\_Transmit('\r');

for(i=0;b[i]!='\0';i++)

{

USART\_Transmit(b[i]);

\_delay\_ms(10);

}

DDRB=(1<<5);

ADMUX=(1<<6);

ADCSRA= (7<<0);

ADCSRA|=(1<<7);

ADCSRA|=(1<<6);

while((ADCSRA & (1<<4))==0);

x=ADC;

x=(x\*100/1024);

sprintf(res,"%d",x);

for(i=0;res[i]!='\0';i++)

{

USART\_Transmit(res[i]);

}

if(x>4)

{

PORTB=(1<<5);

}

}

void ir\_sensore()

{

char a[]="Object is detected! "; char b[]="Object is not Detected! "; int i;

if((PINB & (1<<0))!=0)

{

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

}

step\_clock(90);

}

if((PINB & (1<<0))==0)

{

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;b[i]!='\0';i++)

{

USART\_Transmit(b[i]);

}

step\_anti(90);

}

}

void active\_mode()

{

int i;

char a[]="Auto mode is selected ";

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;a[i]!='\0';i++)

{

USART\_Transmit(a[i]);

\_delay\_ms(10);

}

temperature\_sensor();

gas\_sensore();

ir\_sensore();

}

int main()

{

USART\_Init(MYUBRR);

int i,x,y;

char p[]="Select a mode ";

char q[]="Enter a for active motor m for manual mode "; char v[]="THANK YOU";

char s[]="Do you want to continue any mode y/n: "; char t[]="No mode";

char w[]="do you want to exit enter q or n ";

char r;

USART\_Transmit('\r'); USART\_Transmit('\n');

l1: for(i=0;p[i]!='\0';i++)

{

USART\_Transmit(p[i]);

\_delay\_ms(10);

}

USART\_Transmit('\r'); USART\_Transmit('\n');

l2: for(i=0;q[i]!='\0';i++)

{

USART\_Transmit(q[i]);

\_delay\_ms(10);

}

r=USART\_Receive();

USART\_Transmit(r);

if((r=='a')|(r=='A'))

{

active\_mode();

}

if((r=='m')|(r=='M'))

{

manual\_mode();

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;s[i]!='\0';i++)

{

USART\_Transmit(s[i]);

\_delay\_ms(10);

}

x=USART\_Receive();

USART\_Transmit(x);

if(x=='y') goto l1;

else

{

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;t[i]!='\0';i++)

{

USART\_Transmit(t[i]);

\_delay\_ms(10);

}

}

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;w[i]!='\0';i++)

{

USART\_Transmit(w[i]);

\_delay\_ms(10);

}

y=USART\_Receive();

USART\_Transmit(y);

if(y=='q')

{

PORTB&=~(1<<5);

OCR0A=0;

USART\_Transmit('\r'); USART\_Transmit('\n');

for(i=0;v[i]!='\0';i++)

USART\_Transmit(v[i]);

\_delay\_ms(10);

}

else goto l2;

}