

main.c

```
#include<avr/io.h>
#include<util/delay.h>
#include<avr/pgmspace.h>
#include<compat/deprecated.h>
#include "lcd.h"
#define uchar unsigned char
#define forw 0x50
#define back 0xA0
#define xright 0x60
#define xleft 0x90
#define left 0x40
#define right 0x10
#define brake 0xF0
#define stop 0x00

#define pwmmaxspeed 240

volatile uint8_t Reye;
volatile unsigned char current_cell,next_cell;
uint8_t midpoint[5] = { 0,0,0,0,0 }; /* sensor trip points */
/*
PD0 - switch 1 set
PD1 - switch 2 start
PD2 - switch 3 ok
*/

#include "move.h"
uint8_t adc(unsigned char channel)
{uint8_t adc_value;
    ADMUX = 0x60|channel; // channel 0, left-justified result
    ADCSRA = _BV(ADEN) | _BV(ADPS2) | _BV(ADPS0);
    /* read output from ADC */
    ADCSRA |= _BV(ADSC);
    while (!(ADCSRA & _BV(ADIF)));
    adc_value = ADCH;
    ADCSRA |= _BV(ADIF);

    return adc_value;
}

void eye_read()
{ uint8_t adc_value,i;
  Reye=0;
  for (i=0; i<5; i++)
  {
      Reye = Reye << 1;
      adc_value = adc(i);
      if (adc_value >= midpoint[i])
      {
```

```

main.c
    Reye |= _BV(0);
}
else
{
    Reye &= ~_BV(0);
}
}

}
void motor_pwm_init()
{DDRC=0xFF;
 PORTC=0x00;
 DDRB=0xFF;
 DDRD=0xF0;
 PORTD=0xFF;
//start timer with prescaler 8

PORTC=forw;
TCCR0=0x6D;
TCNT0=0x00;
TCCR2=0x6F;
TCNT2=0x00;
}

uint8_t readobstacle()
{uint8_t p;
 p=(PINA&0xc0);
 if(p==0x00)
 return 0; //both white
 else if(p==0x40)
 return 2; //upblack
 else
 return 1; //no block
}
void steer(unsigned char l,unsigned char r)
{
OCR2=l;
OCR0=r;

}
void pos_calculate()
{unsigned char r,side,t;
 side=(Reye & 0b00010001);
 r=(Reye & 0b00011111);
 t=readobstacle();
 if(side!=0b00010001 && t==1)
 {
 switch(r)
 {case 0x01: drive_motor(0, pwmmaxspeed); break; //xrightmost95
 case
0x03: drive_motor(pwmmaxspeed-210, pwmmaxspeed); break; //xrightmost95
 case 0x06: drive_motor(pwmmaxspeed-80, pwmmaxspeed); break; //slight

```

main.c

```
right35
    case 0x04: drive_motor(pwmmaxspeed, pwmmaxspeed); break; //center
    case 0x0C: drive_motor(pwmmaxspeed, pwmmaxspeed-80); break; //sl
left35
    case
0x18: drive_motor(pwmmaxspeed, pwmmaxspeed-210); break; //xl eftmost95
    case 0x10: drive_motor(pwmmaxspeed, 0); break; //xl eftmost95

}
}
else
{
if(t==1)
decide();
else
{
PORTC=stop;
_delay_ms(2000);
t=readobstacle();
if(t==0)
{
turnright();
turnright();
}
else
{
turnright();
}
while(bit_is_set(PIND, 0));

}
}

}
void avoid_zun()
{unsigned char r, side;
eye_read();
side=(Reye & 0b00010001);
r=(Reye & 0b00001110);
while(side!=0x00)
{eye_read();
side=(Reye & 0b00010001);
r=(Reye & 0b00011111);
switch(r)
{

case
0x01: drive_motor(pwmmaxspeed-120, pwmmaxspeed); break; //xrightmost95
case
0x03: drive_motor(pwmmaxspeed-80, pwmmaxspeed); break; //xrightmost95
case 0x06: drive_motor(pwmmaxspeed-60, pwmmaxspeed); break; //slight
right35
```

```

                                main.c
    case 0x04: drive_motor(pwmmaxspeed, pwmmaxspeed); break; //center
    case 0x0C: drive_motor(pwmmaxspeed, pwmmaxspeed-60); break; //sl
left35
    case
0x18: drive_motor(pwmmaxspeed, pwmmaxspeed-80); break; //xleftmost95
    case
0x10: drive_motor(pwmmaxspeed, pwmmaxspeed-120); break; //xleftmost95
    }

}

```

```

}
void drive_motor(unsigned char Lspeed, unsigned char Rspeed)
{
    PORTC=forw;
    steer(Rspeed, Lspeed); //L-R
}

```

```

void update_robot()
{
    unsigned char hb, lb;
    hb=(current_cell>>4)&0x0f;
    lb=(current_cell&0x0f);
    maze[hb][lb].visited=1;
}
void decide()
{
    update_robot();
    avoid_zun();
    tracepath(current_cell);
    if(current_cell==0x05)
    {
        PORTC=stop;
        while(bit_is_set(PIND, 0));
    }
    current_cell=search_next_move(current_cell);
}

```

```

void calibrate_sensors()
{
    uint8_t adc_value;          /* ADC value */
    uint8_t i, j;               /* loop counter */
    uint8_t thresh_high[5] = { 0,0,0,0,0 };
    uint8_t thresh_low[5] = { 255,255,255,255,255 };
    PORTC=xleft; steer(pwmmaxspeed-40, pwmmaxspeed-40);

    for (i=0; i<100; i++)
    {
        for (j=0; j<5; j++)

```

```

                                main.c
        {
            adc_value = adc(j);
            if (adc_value < thresh_low[j])
thresh_low[j] = adc_value;
            if (adc_value > thresh_high[j])
thresh_high[j] = adc_value;
        }
        _delay_ms(50);
    }

    for (i=0; i<5; i++)
    {
        midpoint[i] = (thresh_low[i] + (thresh_high[i] -
thresh_low[i]) / 2);
    }
    PORTC=stop;
}
void eye_show()
{
    uint8_t t;
    lcd_cmd(0x80);
    lcd_puts("L1-L2- C -R1-R2");
    lcd_cmd(0xc0);
    for(t=0; t<5; t++)
    {
        if((Reye&(1<<t))>0)
        {
            lcd_puts("1  ");
        }
        else
        {
            lcd_puts("0  ");
        }
    }
}

void motor_test()
{
    PORTC=forw;
    steer(64, 64); //right
    //_delay_ms(5000);
    //steer(128, 128); //equal
    //_delay_ms(5000);
    // steer(192, 192); //right
    //_delay_ms(5000);
    //steer(255, 255); //left
    _delay_ms(5000);
    //PORTC=0x00;
}

int main(void)
{

```

main.c

```
DDRD=0xF0;
PORTD=0x0F;
DDRA=0x00;
PORTA=0xC0;
lcd_init();
motor_pwm_init();
lcd_cmd(0x0c);
lcd_cmd(0x01);
lcd_puts("Nexus 2010");
lcd_cmd(0xc0);
lcd_puts("Initialising...");
_delay_ms(2000);
//motor_test();
lcd_cmd(0x01);
calibrate_sensors();
while(bit_is_set(PIND, 0));
current_cell=0x00;
_delay_ms(2000);
while(1)
{

    eye_read();
    //eye_show();
    pos_calculate();
    // }
}
}
```