

Mobiilsed robotid MHK0020 nädalaaruanne		
Gruppi nimi	Ratasjalg	Aruande number
Grupi juht	Pent Laineste	3
Grupi liikmed	Pent Laineste, Marek Laul, Andrei Porõvajev, Vladimir Kuts, Vladimir Zaplava	Kuupäev
Grupi otsene juhendaja	Raivo Sell	09.03.2011
Aruanne tuleb laadida grupi juhi poolt e-õppe keskkonda vastava nädala juurde. Nädalaaruande tähtaeg on neljapäev kell 9:00 (hilinenud aruanded saavad vähem punkte). Küsimuste korral võtke ühendust juhendajaga: Raivo Sell V308 (6203201)		

Püstitatud ülesanded (juhendajate poolt või eelmise nädala grupi plaanid)

- Ratta avamissüsteemi joonised, vajalike detailide valmistamine
- Alusraami joonis ja valmistamine
- Mootori- ja andurikinnituste detailid
- Andurite ja mootorite testimine, programmeerimine

Tehtud tööd (Kirjeldage töökäiku, probleeme, ideid)

- Ratta avamissüsteemi prototüüp
 - Ühe servo kasutamisel oluline avamisvarraste sümmeetrilisus, mõlemad peavad sobituma avamiskettaga, võimalik korrigeerida varda pikkuse muutmisega
 - Servo peab liikuma täpselt vajaliku nurga alla, liiga väikse nurga korral varras ei ulatu avamiskettani, liiga suure nurga korral varras puudutab ratast ja kiilub kinni
 - probleemiks võib kujuneda kulumine, kuna toimub hõõrdumine varda ja ketta vahel
 - esialgsete variantide katsetamine, võimalikud muudatused
- Alusraam ja komponentide kinnitusdetailid
 - Ultrahelianduri, -mootorite ja rataste kinnitused
 - lahtine veel tagaratta tüüp ja aku asukoht
- Andurite ja mootorite testimine
 - avamissüsteemi jaoks sobiva liikumiskiiruse ja –aja leidmine
 - servo ratta avamiseks vajaliku pöördenurga leidmine

Tulemused (konkreetsed tulemused ja valmidusaste (algstaadiumis -A, Prototüüp -P, Valmis -V))

- Ratta avamissüsteemi konstruktsioon – P
- Alusraam - V
- Mootorite, andurite, elektroonika paigutus/kinnitused alusel – P
- Esialgne juhtimisalgoritm - P

Probleemid (Kiiret lahendamist vajavad probleemid)

- Avamissüsteemi töökindluse tõstmine: kinnikiilumise vältimine, samuti on oht kruvide mutrite lahti tulemiseks, kuna avamissüsteemi mõningaid kruvisid ei saa tugevalt kinni keerata

Grupist sõltumatud (vajavad juhendaja abi/otsust)

- ei saa wiki leheküljele faile üles laadida

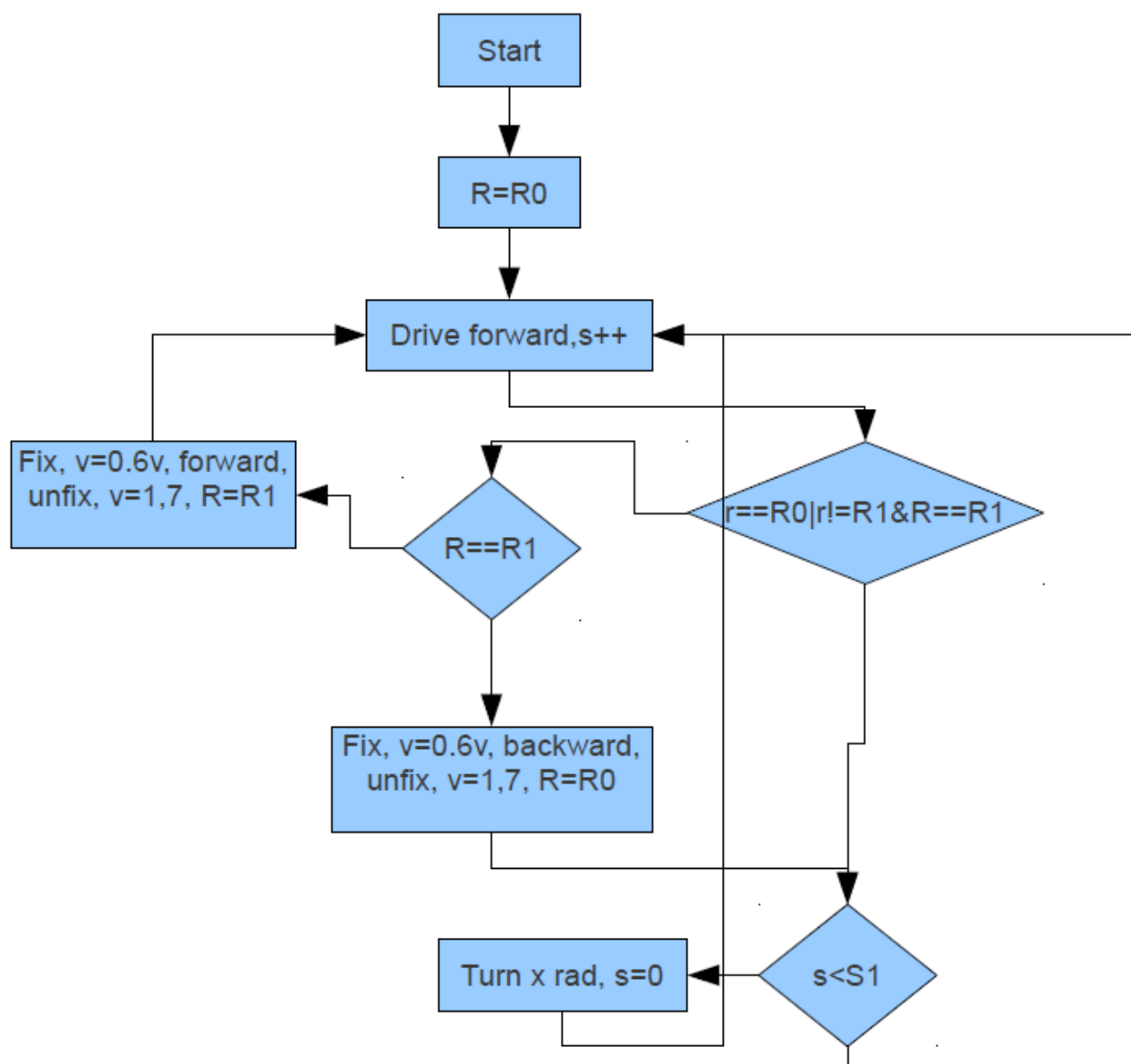
Järgmise nädala tööd (töö nimetus, vajalikud vahendid, planeeritud tulemus koos valmidusastmega)

- Konstruktsiooni täiustamine
 - tagaratta kinnitus
 - aku asukoht ja kinnitus
 - muud muudatused konstruktsioonis roboti toimivuse parandamiseks
- Takistuste tuvastamise ja rataste avamise algoritmi testimine
 - roboti töövõime hindamine erinevates olukordades

Kommentaariid (jms, mis ei sobi eelpool toodud alajaotiste alla)

Lisad (graafiline materjal, tabelid, kirjeldused, jne.)

Flow chart



Testprogramm:

```

#include <avr/io.h>
#include <avr/interrupt.h>
#include <homelab/pin.h>
#include <homelab/delay.h>
#include <homelab/module/motors.h>
#include <homelab/module/lcd_alpha.h>
#include <homelab/module/sensors.h>
#include <stdio.h>

pin pin_trigger = PIN(G, 1);
pin pin_echo    = PIN(G, 0);
static unsigned short clock=0;
static unsigned short ts=0;

ISR(TIMER0_COMP_vect) {
    if(TCCR0&(1<<COM00))
        PORTD|= (1<<4);
    else
        PORTD&=~(1<<4);
}

ISR(TIMER0_OVF_vect) {
    if(TCCR0&(1<<COM00))
        PORTD&=~(1<<4);
    else
        PORTD|= (1<<4);
}

ISR(TIMER2_COMP_vect) {
    cli();
    clock+=1;
    sei();
}

inline bool get_value(unsigned char port, unsigned char pin){
    return ((port)&(1<<pin))>0? true: false;
}

inline void ForWard()
{
    TIMSK |= ((1<<OCIE0) | (1<<TOIE0));
    PORTB&=~(1<<7);
    TCCR0|= (1<<COM01);
    TCCR0&=~(1<<COM00);
    PORTD&=~(1<<5);
}

inline void BackWard()

```

```

{
    TIMSK |= ((1<<OCIE0) | (1<<TOIE0));
    PORTB|= (1<<7);
    TCCR0|= (1<<COM01);
    TCCR0|= (1<<COM00);
    PORTD|= (1<<5);
}

inline void TurnRight()
{
    PORTB &= ~(1<<7);
    TCCR0|= (1<<COM01);
    TCCR0 &= ~(1<<COM00);
    PORTD &= ~(1<<5);
    TIMSK &= ~((1<<OCIE0) | (1<<TOIE0));
}

inline void TurnLeft()
{
    TIMSK |= ((1<<OCIE0) | (1<<TOIE0));
    PORTB&=~(1<<7);
    TCCR0&= ~(1<<COM01);
    TCCR0 &= ~(1<<COM00);
    PORTD &= ~(1<<5);
}

inline void Stop()
{
    TCCR0&= ~(1<<COM01);
    TCCR0 &= ~(1<<COM00);
    TIMSK &= ~((1<<OCIE0) | (1<<TOIE0));
    PORTD&=~(1<<5);
    PORTD&=~(1<<4);
    PORTB &= ~(1<<7);
}

bool OpenLegs()
{
    Stop();
    OCR1A=1843;
    OCR0=120;
    sw_delay_ms(1000);
    ForWard();
    sw_delay_ms(1000);
    OCR1A=3686;
    OCR0=255;
    Stop();
    sw_delay_ms(1000);
    return true;
}

bool CloseLegs()

```

```

{
    Stop();
    OCR1A=1843;
    OCR0=120;
    sw_delay_ms(1000);
    BackWard();
    sw_delay_ms(1000);
    OCR1A=3686;
    OCR0=255;
    Stop();
    sw_delay_ms(1000);
    return true;
}

int main()
{
    TCCR0|=(1<<COM01)|(1<<WGM00)|(1<<WGM01)|0x05;
    OCR0=255;
    lcd_alpha_init(LCD_ALPHA_DISP_ON);
    lcd_alpha_clear();
    char text[16];
    unsigned short distance=0;
    DDRC|=0x38;
    PORTC &=~(0x38);
    DDRF|=(1<<1);
    DDRF &= ~((1<<0)|(1<<2)|(1<<3)|(1<<4));
    PORTF|=(1<<0)|(1<<2)|(1<<3)|(1<<4);
    PORTF&=~(1<<1);
    TCCR1A|=(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11);
    TCCR1B|=(1<<WGM13)|(1<<WGM12)|(1<<CS11);
    ICR1=36864;
    OCR1A=1843;
    OCR1B=1843;
    pin_setup_output(pin_trigger);
    pin_setup_input_with_pullup(pin_echo);
    TCCR2|=(1<<WGM21)|(1<<CS20)|(1<<CS22);
    OCR2=144;
    TIMSK|=(1<<OCIE2);
    dcmotor_init(0);
    dcmotor_init(3);
    servomotor_init(0);
    servomotor_init(1);
    sei();
    bool flag2=true;
    bool flag1=true;
    bool flag0=true;
    unsigned short t0=clock;
    while(1)
    {
        if(t0<clock)
        {

```

```

        distance=ultrasonic_measure(pin_trigger, pin_echo);
        t0=clock+10;
    }
    if(bit_is_clear(PINC,2)&&flag2)
    {
        flag2=false;
        if(OCR0<255)
            OCR0+=10;
    }
    else if(bit_is_set(PINC,2))
        flag2=true;
    if(bit_is_clear(PINC,1)&&flag1)
    {
        flag1=false;
        if(OCR0>120)
            OCR0-=10;
    }
    else if(bit_is_set(PINC,1))
        flag1=true;
    if(bit_is_clear(PINC,0)&&flag0)
    {
        static char count=0;
        flag0=false;
        if(count&0x01)
        {
            OCR1A=1843;
            OCR1B=1843;
            count+=1;
        }
        else
        {
            OCR1A=2600;
            OCR1B=2600;
            count+=1;
        }
    }
    else if(bit_is_set(PINC,0))
        flag0=true;
    if(bit_is_clear(PINF,2))
        ForWard();
    /*else if(bit_is_clear(PINF,3))
        TurnLeft();
    else if(bit_is_clear(PINF,0))
        TurnRight();*/
    else
        Stop();
    sprintf(text,"%d %d %d ",OCR0,clock/100,distance);
    lcd_alpha_goto_xy(0, 0);
    lcd_alpha_write_string(text);
}
/*while(1)

```

```

{
    if(t0<clock)
    {
        distance=ultrasonic_measure(pin_trigger, pin_echo);
        t0=clock+10;
    }
    ForWard();
    sprintf(text,"%d %d %d ",OCR0,clock/100,distance);
    lcd_alpha_goto_xy(0, 0);
    lcd_alpha_write_string(text);
    if(distance<30)
        OpenLegs();
    if(bit_is_clear(PINC,2)&&flag2)
    {
        flag2=false;
        OpenLegs();
    }
    else if(bit_is_set(PINC,2))
        flag2=true;
}*/
return 0;
}

```