

RC ohjattu 600l roska-astia

ref Kennel Helsinki

Arduino RC-vastarikoodi nyt myös githubissa:

https://github.com/rambo/Arduino_rcreceiver/blob/master/Arduino_rcreceiver.ino

Moi

Saanan sanoa muutaman sanan.

Teimme Robosodan puitteissa joskus vähän vastaavan projektin, mm radio-ohjattava jääkaappi ja pesukone. Käytimme noissa akkuporakoneen moottoreita. Moottorit pitää modata jotta saadaan momenttisäätö pois.

Täällä hyvä ohje: http://www.kolumbus.fi/rex/akkuporan_modaus_ohjeet.html

Nopeudensäätimien tyyppiä en muista mutta jokin jenkkiläisen robokaupan säädin, hyvä mutta kallis.

Akkuporakonee moottorille voisi riittää hyvin esim. Robbe Rookie 25 säädin. Edullinen pakilla varustettu säädin. Jokaiselle moottorille yksi. Ohjaus noille säätimille esim GWS:n V-tail mikserin kautta, jos radio-ohjaimessa ei ole tuota ominaisuutta. V-mikserillä saadaan aikaiseksi ns. tankki ohjaus. Minulta löytyis ainakin tuo V-mikseri ja varmaan pari rookie säädintäkin lainaan. Rungon moottoreille, pyörille ja muulle roipeelle teimme yksinkertaisesti puusta. Akkuna 12V 18Ah lyijyakku.

Muuten olen niin kiinni muissa projekteissa etten pysty auttamaan rakennusprojektissa, vaikka mieli tekisikin.

T:Petri

v-mikseri tehty Arduino softassa (kts padin loppu), H-sillat (aka nopeudensäätimet) on Arduinon päällä shieldissä (12A jatkuvaa virtaa per moottori kertaa 2 moottoria)

/Rambo

Tilanne 2013.05.11

- Näillä näkymin **mekaniikka ei tule tapahtumaan** (klo 18 ja kukaan ei ole käynyt läbillä osoittamassa minkäänäköistä kiinnostusta asian konkreettiseen edistämiseen)
- Tutkailun jälkeen todettu: Takapyörien (eli vetävien pyörien) z-akselin ympäri pyörimisen lukitseminen on pysyvä muutos (pitää hitsata tai porata laakerien läpi, näihin ei ole "undo- nappia") ja tätä **ei kannata tehdä** ellei ole 100% varmaa että kaikki muukin tulee valmiiksi ajoissa.
- Moottorit laitettu takaisin porakoneiden runkoihin ja johdot tuotu rungosta läpi.

Tilanne 2013.05.09

- Akkuporakoneet purettu
 - Tosin kannattanee laittaa ne moottorit takaisin porien runkoihin ja vetää johdot ulos niin saa helpommin kiinni.
- Rambo testannut Arduinolla RC-vastaanottoa ja porista purettujen moottorien ajoa
 - Hyvin toimii, softa padin lopussa, ardu, shieldi yms pöydällä moottorien vieressä.
- Rambo toimittanut auton akun läbille jota voi käyttää virtalähteenä.
- Mekaniikka yhä täysin vaiheessa
 - Rambo sopinnut Stadi Mallarien Henttosen kanssa että la on mahdollista käydä Inarintiellä

työstämässä metallia mikäli tarpeen.

Tilanne 2013.05.07

Astia läbillä, ja pohdittu on:

- Ei voi laittaa isoja renkaita tai kitkarullia koska näkyy kuvissa (kuvakulma sellaisia)
- Omat renkaat ehkä pidoltaan riittävät
-

Kokeillaan: polkupyörän rattaat kiinni takarenkaisiin josta ketju sisään ja Biltema CD12 ruuvinvääntimellä pyöritellään toista ratasta

rambolla on pololun motor driver shieldi (muistaakseni VNH5019) jolla voi ohjata ruuvinvääntimien moottoreita kuhan kaivellaan johdot esiin (ja otetaan syöttö autonakusta).

jaronekolta saatiin lainaa radiolaitteet, kytketään arduun joka tulkkaa signaalit moottoriohjaimelle.

Otin yhden pyörän irti tarkasteltavaksi (pajassa). jos ruuvinvääntimessä on kiristettävä sokka voisi olla mahdollista kiinnittää se suoraan pyörän akseliin ilman välityksiä jolloin vetäviä pyöriä voisi kääntää. Olen helatorstaina paikalla jos rakennetaan hökötytys silloin. Mulla on hakattu lennokkisofta odroidille jolla voi ohjata pololulua joystickillä wlanin yli.

Mitä jos polkupyörän ratas ottaisi suoraan kiinni maahan eikä pyörään?

<http://www.biltema.fi/fi/Tyokalut/Sahkotyokalut/Porakone-akkukayttoinen/Porakone-ja-ruuvinvaannin-CD-12-V-17358/>

<http://www.pololu.com/catalog/product/2502>

Voisin tuoda torstaina vähän omia robokamoja labille, niistä voisi olla jotain iloa. Löytyy tehokkaita moottoriohjaimia ja yksi robotti minkä saisi esim. suoraan pultattua roskiksen pohjaan kiinni. --Eipou

Heitin keskiviikkona labille kaksi vanerilevyä ja kaksi noita tusinavääntimiä Biltemasta..

Moottorit renkaiden vieressä ei mielestäni toimi, koska ne saattaa ottaa kipeetä kuvaustilanteessa ja näkyvätkin aika ilkeästi. Rakennelman olisi hyvä olla sellainen, että koko Labin varikkotiimin ei tarvitse päivystää kuvauspaikalla.

Hankittavaa olisi ainakin vielä:

- Pyörän ketjua,
- 4 x identtisiä ketjurattaita
 - Kannattaisko kuitenkin ottaa joku 1:2 välitys tms <- rambo

Lanttu

-----BEGIN: Arduino RC vastaanotto ja Pololu ohjaussofta ----

// Get this from <https://github.com/pololu/Dual-VNH5019-Motor-Shield>

```

#include "DualVNH5019MotorShield.h"
DualVNH5019MotorShield md;
// Get this from https://github.com/rambo/PinChangeInt\_userData
#include "PinChangeInt_userData.h"

#define SERVO_MIDDLE 1500

// To hold data for reach RC input pin.
typedef struct {
    const uint8_t pin;
    volatile unsigned long start_micros;
    volatile unsigned long stop_micros;
    volatile boolean new_data;
    uint16_t servo_position; // microseconds
} RCInput;

// Initialize the inputs to an array
RCInput inputs[] = {
    { A4, 0, 0, false, SERVO_MIDDLE }, // Forward/backward
    { A5, 0, 0, false, SERVO_MIDDLE }, // Left/right
};
const uint8_t inputs_len = sizeof(inputs) / sizeof(RCInput);

// Speeds for the left & right motors
int16_t left_speed = 0;
int16_t right_speed = 0;

// Called whenever a control pulse ends
void rc_pulse_low(void* inptr)
{
    RCInput* input = (RCInput*)inptr;
    input->stop_micros = micros();
    input->new_data = true;
}

// Called whenever a control pulse starts
void rc_pulse_high(void* inptr)
{
    RCInput* input = (RCInput*)inptr;
    input->new_data = false;
    input->start_micros = micros();
}

// Calculates the servo position, called from the mainloop whenever there is new data
void calc_servo_position(void* inptr)
{
    RCInput* input = (RCInput*)inptr;
    input->servo_position = (uint16_t)(input->stop_micros - input->start_micros);
    input->new_data = false;
}

int16_t clip_md_speed(int16_t spd)
{

```

```

    if (spd > 400)
    {
        spd = 400;
    }
    if (spd < -400)
    {
        spd = -400;
    }
    return spd;
}

int16_t map_rc_to_speed(uint16_t servo_position)
{
    return clip_md_speed(servo_position - SERVO_MIDDLE);
}

void stopIfFault()
{
    if (md.getM1Fault())
    {
        Serial.println(F("M1 fault"));
        md.setSpeeds(0,0);
        while(1);
    }
    if (md.getM2Fault())
    {
        Serial.println(F("M2 fault"));
        md.setSpeeds(0,0);
        while(1);
    }
}

void setup()
{
    Serial.begin(115200);
    Serial.println(F("RC controlled dual VNH5019 Motor Shield"));
    md.init();

    // Attach pin change interrupts for the RCInputs
    for (uint8_t i=0; i < inputs_len; i++)
    {
        PCintPort::attachInterrupt(inputs[i].pin, &rc_pulse_high, RISING, &inputs[i]);
        PCintPort::attachInterrupt(inputs[i].pin, &rc_pulse_low, FALLING, &inputs[i]);
    }
    Serial.println(F("Booted"));
}

volatile unsigned long last_report_time = millis();
int16_t lrspeed = 0;
int16_t fbspeed = 0;
void loop()
{

```

```

// Stop and hang if motor driver has fault
stopIfFault();

// Check inputs for new data
for (uint8_t i=0; i < inputs_len; i++)
{
    if (inputs[i].new_data)
    {
        calc_servo_position(&inputs[i]);
    }
}

lrspeed = map_rc_to_speed(inputs[1].servo_position);
if (lrspeed != 0)
{
    left_speed = lrspeed;
    right_speed = -1 * lrspeed;
}
else
{
    left_speed = 0;
    right_speed = 0;
}
fbspeed = map_rc_to_speed(inputs[0].servo_position);
left_speed += fbspeed;
right_speed += fbspeed;

// Set speeds
left_speed = clip_md_speed(left_speed);
right_speed = clip_md_speed(right_speed);
md.setSpeeds(left_speed, right_speed);

// Report positions every second
if (millis() - last_report_time > 200)
{
    for (uint8_t i=0; i < inputs_len; i++)
    {
        Serial.print(F("Channel "));
        Serial.print(i, DEC);
        Serial.print(F(" position "));
        Serial.print(inputs[i].servo_position, DEC);
        Serial.print(F(" speed "));
        Serial.println(map_rc_to_speed(inputs[i].servo_position), DEC);
    }
    Serial.print(F("Left motor speed "));
    Serial.println(left_speed, DEC);
    Serial.print(F("Right motor speed "));
    Serial.println(right_speed, DEC);
    last_report_time = millis();
}
}

```

-----END: Arduino RC vastaanotto ja Pololu ohjaussofta ----

