

M4 Robotika ja mehhatroonikarakenduste loomine mikrokontrollerite baasil



Õppematerjali koostas ITT Group OÜ 2014/05

Koostajad: Raivo Sell, Rain Ellermaa, Heiko Pikner

Litsents: CC BY-SA 3.0 EE



Robotika ja mehhatroonikarakenduste loomine mikrokontrollerite baasil annab ülevaate ühe näiteprojekti lahendamise kohta kahel arendusplatvormil. Näiteprogramme uurides on võimalik tutvuda kontrollerite eripäradega ning erinevate võimalustega.

Praktilise tööna tuleb luua lihtne temperatuuri jälgiv süsteem, mis temperatuuri tõustes üle normi lülitab süsteemi jahutamiseks sisse ventilaatori.

Arduino Uno rakendus

Näiteprogramm, mis temperatuurinivoo ületamisel käivitab ventilaatori. Temperatuurinivoo seadistatakse potentsiomeetriga. Nupule vajutus seiskab ventilaatori koheselt.

Ülesande lahendamiseks on vaja kasutada Arduino kontrolleriplaati ning mootorite juhtplaati, kuhu oleks võimalik ühendada ventilaator. Temperatuuri mõõtmiseks on vaja kasutada termoandurit ning ventilaatori töö peatamiseks nuppu.



```
/*
Nimetus: Näide
Kirjeldus:   Luua programm, mis temperatuurinivoo ületamisel käivitab ventilaatori.
              Temperatuurinivoo seatakse potentsiomeetriga.
              Nupule vajutus seiskab ventilaatori koheselt.
Autor: Rain Ellermaa
Kuupäev: 24.04.2014
*/

int pote_sisend = A1; // määrame potentsiomeetri sisendviigu
int NTC_sisend  = A0; // määrame temperatuurianduri sisendviigu
int led         = 13; // määrame LED ühendusviigu
int ventilaator = 3;  // määrame ventilaatori ühendusviigu
int nupp        = 2;  // määrame nupu ühendusviigu

void setup()
{
  pinMode(led, OUTPUT); // seadistame LED viigu väljundiks
  pinMode(ventilaator, OUTPUT); // seadistame ventilaatori viigu väljundiks
  pinMode(nupp, INPUT); // algväärtustame nupu viigu sisendiks
}
```

```

    Serial.begin(9600);          // Käivitame jadaühenduse arvutiga, kuhu kuvame analoogväärtused
}

void loop()
{
    // muutujad potentsiomeetri ja temperatuuri väärtuse salvestamiseks
    int pote = analogRead(pote_sisend);    // loeme potentsiomeetri väärtuse
    int temperatuur = analogRead(NTC_sisend); // loeme temperatuuri andurit

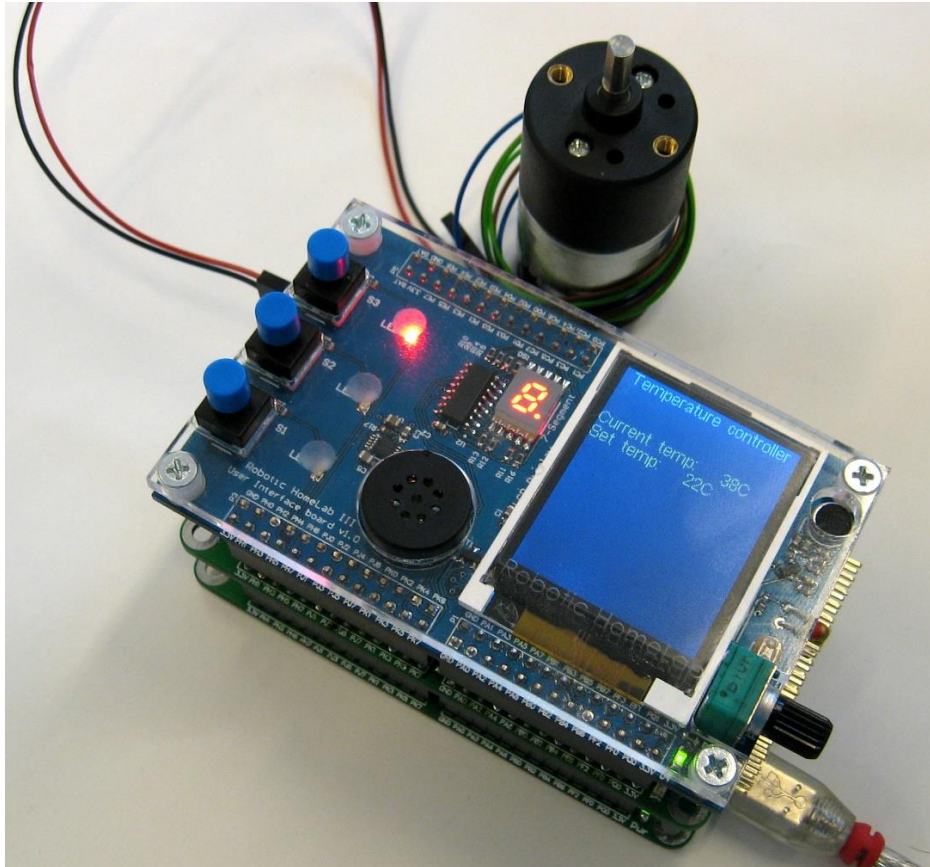
    // Kui temperatuur on kõrgem potentsiomeetriga seatud väärtusest ja nuppu pole vajutatud
    if((temperatuur > pote) && (digitalRead(nupp) == HIGH))
    {
        digitalWrite(led, HIGH); // süütame LED-i
        // Kui võimendatud ADC väärtuste vahe on üle 8-bitit
        // seatakse ventilaatori kiiruseks maksimaalkiirus
        // Muul juhul on ventilaatori kiirus seda suurem,
        // mida suurem on potentsiomeetri ja temperatuurianduri pinge vahe
        if(((temperatuur - pote)*4) > 255)
            digitalWrite(ventilaator, HIGH);
        else
            // Ventilaatori kiirust reguleeritakse madala sagedusega tarkvaralise PWM signaaliga
            analogWrite(ventilaator, (temperatuur - pote)*4);
    }
    else
    {
        digitalWrite(led, LOW); // vastasel korral kustutame LED-i
        digitalWrite(ventilaator, LOW); // ventilaator kinni
    }

    // Saadame arvutisse silumise eesmärgil loetud andurite väärtused
    Serial.print("Pote: ");    Serial.print(pote);
    Serial.print("    Temp: "); Serial.print(temperatuur);
    Serial.println(""); //Tekita reavahetus
}

```

Robootika Kodulabori rakendus

Ülesande lahendamiseks Robootika Kodulaboriga tuleks kasutada III generatsiooni kontrolleri puhul Kontrollerplaati, Komboplaati ning Kasutajaliidese plaati. Vanemate generatsioonide puhul tuleb kasutada lisaks Kontroller- ja Kasutajaliidese plaadile ka Komboplaati või Mootori ja Anduriplaati.



```
/* Nimetus: Näide
   Kirjeldus: Luua programm, mis temperatuurinivoo ületamisel käivitab ventilaatori.
   Temperatuurinivoo seatakse potentsiomeetriga.
   Nupule S1 vajutus seiskab ventilaatori koheselt.
   Hetke temperatuur ning temperatuurinivoo kuvatakse LCD ekraanile.
   Autor: Rain Ellermaa
   Kuupäev: 28.05.2014 */
#include <avr/io.h>
#include <homelab/pin.h>
#include <homelab/delay.h>
#include <homelab/xmega/clksys_driver.h>
#include <homelab/module/lcd_gfx.h>
#include <homelab/module/motors.h>
#include <homelab/module/sensors.h>
#include <homelab/adc.h>

#define FAN 1
#define THERMISTOR 14
#define POT 15

long map(long x, long in_min, long in_max, long out_min, long out_max)
{
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
```

```

int main(void)
{
    // Kodulabori III generatsiooni mooduli taktsageduse määramine
    Homelab_clock_init();
    // LEDide väljunditeks seadmine
    pin_setup_output(led_green);
    pin_setup_output(led_yellow);
    pin_setup_output(led_red);

    // LCD ekraani algseadistamine
    lcd_gfx_init();
    lcd_gfx_goto_char_xy(0,0);
    lcd_gfx_write_string(" Temperature controller ");

    // Ventilaaatori mootorikontrolleri seadistamine
    dcmotor_drive_pwm_init(FAN,TIMER_NO_PRESCALE);

    // ADC muunduri seadistamine
    adc_init(ADC_REF_AREF, ADC_PRESCALE_8);
    while(1)
    {
        char text[50];
        static uint8_t ventilaator = 1;
        // Andurite lugemine
        uint16_t temperature = thermistor_calculate_celsius(adc_get_value(THERMISTOR));
        uint16_t potensiometer = map(adc_get_value(POT),0,2047,0,100);

        // Teksti kuvamine LCDl
        sprintf(text, "Current temp: %4dC \nSet temp: %4uC ",temperature,potensiometer);
        lcd_gfx_goto_char_xy(0, 3);
        lcd_gfx_write_string(text);

        // Kontrollime nupule vajutust
        if(button_read(S1))
            ventilaator = 0;

        // Ventilaaatori juhtimine.
        if((temperature > potensiometer) && (ventilaator == 1))
        {
            // Käivitame ventilaatori ja paneme põlema punase LEDi
            dcmotor_drive_pwm(FAN,1,0xFF);
            led_off(led_green);
            led_on(led_red);
            FgColor = RED;
        }
        // Kui temperatuur on kõrgem seatud väärtusest ja ventilaator pole lubatud
        else if(temperature > potensiometer)
        {
            // Seiskame ventilaatori, paneme põlema punase ja vilgutame kollast LEDi
            dcmotor_drive_pwm(FAN,1,0);
            led_off(led_green);
            led_on(led_red);
            pin_toggle(led_yellow);
            FgColor = YELLOW;
            sw_delay_ms(100);
        }
        else
        {
            // Seiskame ventilaatori, paneme põlema roheline LEDi ja lubame ventilaatori
            dcmotor_drive_pwm(FAN,1,0);
            led_on(led_green);
            led_off(led_red);
            led_off(led_yellow);
            ventilaator = 1;
            FgColor = GREEN;
        }
    }
}

```

Praktiline töö

Praktiline töö Robotika Kauglaboriga – Ventilaatori käivitamine vastavalt temperatuurile ja valgusele.

Ülesandeks on ventilaatori (mootori) tööle lülitamine vastavalt eelseadistatavale temperatuurile. Ventilaatori juhtimine peab olema lahendatud mitmeastmeliselt, kus väike temperatuurierinevus paneb mootori tööle aeglasemalt ja suurem temperatuurierinevus suurendab ka ventilaatori kiirust. Hetke ning seadistatud temperatuur tuleb kuvada LCD ekraanil kraadides. Samuti tuleb ekraanile kuvada mootori töökiirus protsentides. Lisaks temperatuurile tuleb mõõta ka valgustugevust ning valgustugevus lux-ides tuleb esitada samuti LCD ekraanil.

Töö tuleb vormistada lühiaruandena, kus asuane sisaldab töö kirjeldust, kommenteeritud tarkvara lähtekoodi ja järeltust.

Kokkuvõte

Antud mooduli eesmärgiks on näitlikustada ühe rakenduse lahendamist erinevatel kontrollritel. Ülesannet saab alati lahendada erinevatel kontrollritel ning antud moodulis kasutatud Arduino ja Kodulabori platvormid on vaid valik võimalikest lahendustest.

Lisalugemist ja viited

1. Robotika Kodulabori platvorm – <http://home.roboticlab.eu/et/kit>
2. Robotika Kodulabori tarkvarateek - <http://home.roboticlab.eu/et/software/homelab/library>
3. Robotika Kauglabor - <http://distance.roboticlab.eu/>
4. Arduino Reference library - <http://arduino.cc/en/Reference/HomePage>