Raspberry Pi – termostaatti

Tämän projektityön tarkoitus on rakentaa järjestelmä, jossa Raspberry Pi:llä ja lämpötila-anturilla simuloidaan ilmastointijärjestelmän termostaattia. Kommunikaatio Raspberry Pi:n ja anturin välillä tapahtuu käyttämällä I2C-väylää. Yhteys Raspberry Pi:hin tulee ottaa SSH yhteyden avulla. Anturia ohjaava ohjelmakoodi kirjoitetaan Python-ohjelmointikielellä. Järjestelmän tulee ilmoittaa mittausinformaatio konsolissa, mutta myös ledien avulla. Järjestelmällä mitataan huoneilmaa, joka tulee pitää 24-25°C välillä. Käytännössä tämä tarkoittaa sitä, että lämpötilan ollessa alle 25°C oletetaan huoneilman lämpötilan nousevan tietyllä nopeudella. Kun lämpötila nousee 25°C tasolle, ilmastointi kytkeytyy päälle ja pysyy päällä kunnes huoneilman lämpötila on laskenut 24°C tasolle. Ohjaus tulee hoitaa rautatasolla hyödyntäen lämpötila-anturin ominaisuuksia (ohjelmallisesti luettuun lämpötila-arvoon perustuva ohjaus ei ole sallittua).

Ledien ohjaus:

- huoneilman lämpötila OK: vihreä ledi palaa, punainen ei (ilmastointi ei päällä)
- huoneilman lämpötila liian korkea: vihreä sammuu, punainen ledi syttyy ja palaa kunnes lämpötila on halutulla tasolla (ilmastointi päällä)

Lediohjauksen lisäksi järjestelmän tulee näyttää konsolissa käyttäjälle selkeästi tämän hetkinen lämpötila-arvo celsius-asteina yhden desimaalin tarkkuudella. Lämpötilan tarkkuus sekin täytyy perustua rautatasolla säädettyyn mittaustarkkuuteen.

Raspberry Pi – Käyttöönotto

Kattavat käyttöönotto-ohjeet löytyvät esimerkiksi osoitteesta http://elinux.org/RPi_Beginners.
Käytettävät muistikortit ovat esiasennettuja Raspbian-käyttöjärjestelmällä. Esiasetetut tunnukset ovat,

user: pi
password: raspberry

Rpi kannattaa pitää kiinni Internetissä, jotta pakettien ja päivitysten asentaminen onnistuu helposti. Käyttöjärjestelmä kannattaa ensin päivittää komennolla **sudo apt-get update**.

I2C

Tutustu I2C väylän toimintaan (esim. http://en.wikipedia.org/wiki/I%C2%B2C)

RPissä on vakiona I2C väylä poistettu käytöstä. Kirjoita käsky **sudo nano /etc/modprobe.d/raspi-blacklist.conf**. Avautuvasta tiedostosta voidaan I2C väylä ottaa käyttöön kommentoimalla **blacklist i2c-bcm2708** risuaidalla pois. Tämän lisäksi I2C-moduli on lisättävä kernelin boottilistaan. Kirjoita **sudo nano /etc/modules** ja lisää avautuvaan tiedostoon rivi **i2c-dev**.

I2Ceen käyttö vaatii myös pakettien asentamisen. Kirjoita **sudo apt-get install i2c-tools**, jolla voidaan skannata I2C-väylää (http://www.lm-sensors.org/wiki/I2CTools). Toinen asennettava paketti saadaan

kirjoittamalla **sudo apt-get install python-smbus**, jolla voidaan käyttää I2C-väylää Pythonilla.Ohjelman konfigurointia varten lisätään käyttäjä Pi I2C käyttöoikeusryhmään käskyllä **sudo adduser pi i2c**. Käynnistä järjestelmä uudelleen **sudo reboot**.

Rpi:ssä on kaksi I2C kanavaa, 0 ja 1. Kurssin Rpi mallit käyttävät oletuksena 1 kanavaa. Testataksesi onko väylässä kiinni mitään kirjoita **i2cdetect –y 1**, joka näyttää seuraavalta, jos mitään ei ole kytketty

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
00:         -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Anturin kytkeminen ja ohjaaminen

Sensorissa on siis neljä linjaa: käyttöjännite (punainen), maa (musta), data (sininen, SDA) ja kellosignaali (punakeltainen, SCL). Lämpötilasensori toimii 3.3V jännitteellä (tasokonvertteria ei siis tarvita!). Raspberry Pi:n GPIO-pinoutin löytää esimerkiksi osoitteesta http://elinux.org/RPi_Low-level_peripherals. Kytkennän helpottamiseksi on järkevää kytkeä kaikki ensin koekytkentälevylle, josta alkaa jakaa johdotuksia eri komponenteille. Kun kytkentä on valmis, kannattaa ajaa i2cdetect, josta nähdään missä osoitteessa sensori sijaitsee. Jos sensori ei tunnistaudu tarkista kytkentä ja kokeile vielä vaihtaa tarkistettavaa kanavaa.

LEDien kytkeminen

Ledit voidaan kytkeä etuvastuksella RPin GPIO-pinneihin. Raspbianiin on esiasennettuna GPIO-kirjasto Pythonille, jonka avulla pinnien ohjaus käy helposti. Low-level Peripherals –sivulta löytyy lisää tietoa.

SMBus-komennot

function	description	parameters	return value
SMBus Access			
write_quick(addr)	Quick transaction.	int addr	long
read_byte(addr)	Read Byte transaction.	int addr	long
write_byte(addr,val)	Write Byte transaction.	int addr,char val	long
read_byte_data(addr,cmd)	Read Byte Data transaction.	int addr,char cmd	long
write_byte_data(addr,cmd,val)	Write Byte Data transaction.	int addr,char cmd,char val	long
read_word_data(addr,cmd)	Read Word Data transaction.	int addr,char cmd	long
write_word_data(addr,cmd,val)	Write Word Data transaction.	int addr,char cmd,int val	long
process_call(addr,cmd,val)	Process Call transaction.	int addr,char cmd,int val	long
read_block_data(addr,cmd)	Read Block Data transaction.	int addr,char cmd	long[]
write_block_data(addr,cmd,vals)	Write Block Data transaction.	int addr,char cmd,long[]	None
block_process_call(addr,cmd,vals)	Block Process Call transaction.	int addr,char cmd,long[]	long[]
I2C Access			
read_i2c_block_data(addr,cmd)	Block Read transaction.	int addr,char cmd	long[]
write_i2c_block_data(addr,cmd,vals)	Block Write transaction.	int addr,char cmd,long[]	None