

Väderstation

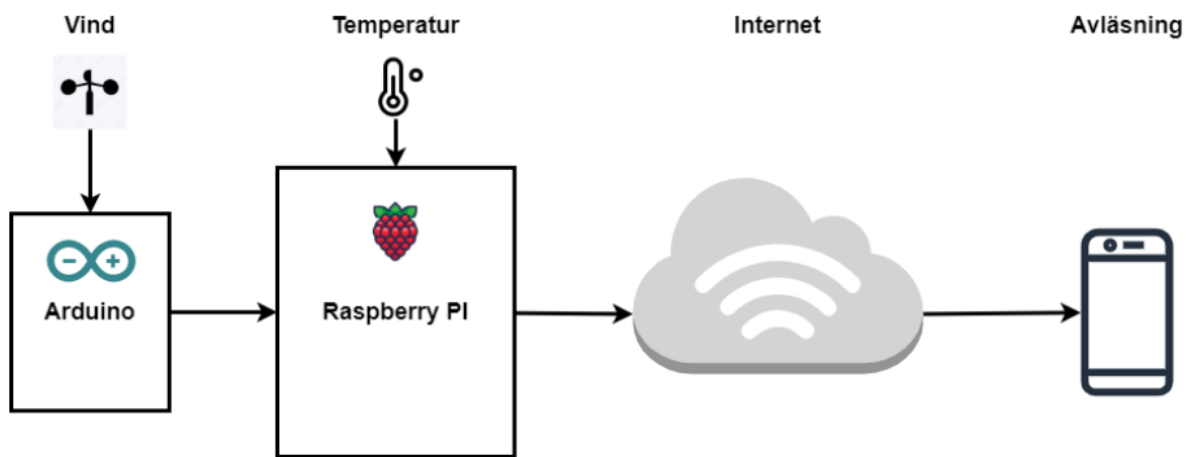


Utförare: Daniel Jovanovski, Roni Omari & Jetmir Salihi
Handledare: Erik Jagre
Datum: 2022-09-08

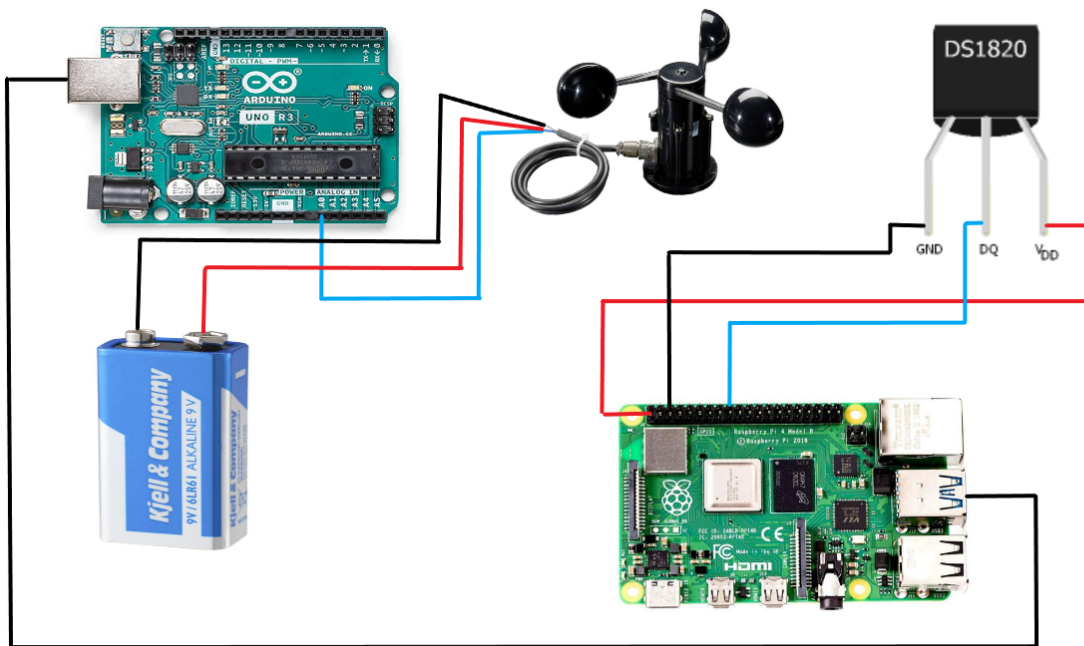
1. Introduktion

I detta projektet har en väderstation skapats där det är möjligt att läsa av temperatur och vindhastighet över internet. För att få detta system att fungera har en Raspberry PI (RPI) använts som är huvudet i projektet och tar emot data från två olika sensorer, där ena är en temperatursensor som är direkt kopplad till RPI:n.

Den andra sensor är en analog vindsensor som är kopplad till en arduino. Arduinon används för att hämta data från temperatursensor och skickar över datan till RPI:n. Nedan visas bild på hur systemet är kopplat.



2.1 Kretsschema på väderstation



2.2 Beskrivning av temperatursensorn

Temperatursensorn som används i projektet är en digital temperatursensor med namnet DS1820. Den kommunicerar med en 1-wire buss som är enkel att använda då bara en kabel behövs för signalen, en kabel till ground och en kabel till spänning.

Temperatursensorn har en power supply mellan 3.0V till 5.5V vilket passar bra till RPI:n då det finns alternativ för båda.



2.3 Beskrivning av vindsensorn

Vindsensorn som används i projektet är en analog sensor. Även denna sensorn är enkel att använda då endast tre kablar behöver kopplas. En brun kabel till en spännig mellan 7V-24V, en svart till jord och den blå kabeln är där vindhastigheten presenteras mellan 0.4V (0m/s) och 2.0V (32.4m/s). Vindsensorn har ett mätområde mellan 0.5 – 50m/s.

Vindsensorns kablar kopplas till arduinon som läser in datan och skickar sedan datan vidare till RPI:n.



2.4 Beskrivning av kod för vindsensor(arduino)

```

/*****
Seriell överföring initieras för att seriell kommunikation ska fungera.
*****/

void setup()
{
    Serial.begin(9600);
}

/*****
Data läses från analogPin (A0) och sparas som en float i variabeln sensorValue.
Sensors analoga värde omvandlas sedan till en spänning och sparas som en float i
variabeln voltage.
*****/

void loop()
{
    float sensorValue = analogRead(A0);
    float voltage = (sensorValue / 1023) * 5;

    /*****
Vindhastigheten startar på 1m/s vid 0,4V med max 32,4m/s runt 2.
Arduino har en inbyggd map() funktion, men map() fungerar inte för float, så vi har en enkel
mapFloat() funktion.
Mappningen sparas som en float i variabeln wind_speed.
En if-sats med argumentet om wind_speed är mindre än "0" så printas "0,00" ut för att
undvika negativa värden.
Annars så printas korrekt uppmätt värde ut.
*****/

    float wind_speed = mapfloat(voltage, 0.4, 2, 0, 32.4);
    if (wind_speed < 0)
    {
        Serial.println("0,00");
    }
    else
    {
        Serial.println(wind_speed);
        delay(100);
    }
}

/*****
Mapp funktionen används för att spara variabeln som en float.
*****/

float mapfloat(float x, float in_min, float in_max, float out_min, float out_max)
{
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

```

2.5 Kombinerad kod för tempsensorn och vindsensorn

Först försökte vi läsa ut tempsensorn i terminalen utan någon kod alls. 1-wire enablades i raspi-config. För att sedan läsa ut temp sensorn gick vi till `/sys/bus/w1/devices/28-0000095d19c6` och körde `cat` på `temperature`. Då lästes temp sensorn ut i terminalen, men det fanns inte några decimaler. Detta fixades genom att göra ett script för tempsensorn, scriptet skapades också för att inte behöva gå till temperaturens filsökväg hela tiden, datan som ficks av tempsensorn lagrades i en `.txt` fil. Koden för scriptet finns här: <https://github.com/jRoniO/weather> (temp.py)

För att föra över seriell info från arduinon till RPi:n så användes en USB kabel. Genom att gå till `/dev/ttyACM0` så kunde vi få ut seriell data i Ubuntu. Ett script skapades för vindsensorn för att slippa gå till den filsökvägen. Data som togs emot från vindsensorn sparades i `.txt` fil. Koden för scriptet finns i här: <https://github.com/jRoniO/weather> (vind.py)

Sedan kombinerades båda koderna. Vi använde trådar för att exekvera koden parallellt. Utan trådar så läste bara tempsensorn in. Datat för båda sensorerna lagrades i varsin `.txt` fil. Vi testade att lagra allt i samma `.txt` fil men då blev det bara problem och båda sensorerna försökte skriva över varandra.

```
#!/usr/bin/python3
import serial
import time
from w1thermsensor import W1ThermSensor
import threading

sensor = W1ThermSensor()

if __name__ == '__main__':
    ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
    ser.flush()

    def print_wind_speed():
        while True:
            try:
                if ser.in_waiting > 0:
                    line = ser.readline().decode('utf-8').rstrip()
                    file = open("/var/www/html/vindData.txt", "w") #Öppnar vindData.txt
                    file.write('Vindhastigheten är: %s m/s' % line) #skriver vindhastigheten
                    file.close() #stänger filen
                    #Om vindsensorn inte rörde på sig vid start så fick vi ett Unicode error, koden nedan
                    #hjälpte till med att undvika detta.
            except UnicodeDecodeError:
                ser.readline().decode('utf-8').rstrip()
    return
```

```

def print_temperature():
    while True:
        temperature = str(sensor.get_temperature())
        avrundning = temperature[0:4] #temperaturen avrundas till två decimaler
        file1 = open("/var/www/html/tempData.txt", "w") #öppnar tempData.txt
        file1.write("Temperaturen är %s grader celsius' % avrundning) #skriver till filen
        file1.close() #stänger filen
    return

t1 = threading.Thread(target = print_wind_speed) #tråd för vindhastigheten
t2 = threading.Thread(target = print_temperature) #tråd för temperaturen
"""
Startar båda trådarna och sätter ihop dom
"""
t1.start()
t2.start()

t1.join()
t2.join()

```

2.5 Skicka info över internet

Vi skapade en local hemsida för att skicka sensorernas info live över internet. Vi ladda ner apache på RPi:n som gör så att RPi:n IP-adress kan användas som en hemsida. Vi installerade också PHP som är ett skriptspråk som främst körs på webbservrar för att driva internetsajter med dynamisk innehåll.

Skapade en fil som hette vindApp.php

```

<meta http-equiv="refresh" content="1"> #Refreshar sidan varje sekund

<?php

#Hämtar info från vindData.txt och tempData.txt
$temp = file_get_contents("vindData.txt");
$temp1 = file_get_contents("tempData.txt");

#Info i vindData skrivs ut i blå
if ($temp){
    $color="blue";
}

#Info i tempData skriv ut i röd
if ($temp1){
    $color1="red";
}

```

#Skriver ut vår titel

```
print "<h1 style='text-align:center;'>Avläsning för temperatur och vindhastighet</h1>";  
print "<br>";
```

#Skriver ut infon från filerna på våran hemsida

```
print "<p style='text-align:center; font-size:100px; margin-top:10px; margin-bottom:10px; color:'.Scolor.';'>".Stemp."</p>";  
print "<p style='text-align:center; font-size:100px; margin-top:20px; margin-bottom:20px; color:'.Scolor1.';'>".Stemp1."</p>";
```

```
?>
```