

# Projektrapport Väderstation Grupp 3



Projektmedlemmar:

Hannes Camitz, Andreas Åkesson, Markus Gustafsson, Johan Lamberth, Fredrik Karlsson

## Sammanfattning

Projektarbetet gick ut på att bygga en väderstation som kunde presentera temperatur och vindhastighet allt enligt 'Projekt - Väderstation' <sup>[1]</sup>. Datan skulle komma från två olika källor men presenteras gemensamt. Datan för vindhastighet kommer från en anemometer via mikroprocessor (Arduino) och datan för temperaturen kommer från en tempsensor via Raspberry Pi. Rapporten kommer att visa hur kommunikationen från olika källor samlas till en presentation.

# Innehåll

<b>Sammanfattning</b>	<b>2</b>
<b>Innehåll</b>	<b>3</b>
<b>Produktbeskrivning</b>	<b>4</b>
Material/utrustning	4
Systemöversikt	4
Kodbeskrivning	5
Arduino UNO	5
Raspberry Pi	5
Resultat	6
<b>Referenser</b>	<b>7</b>

## Produktbeskrivning

[1]Projektet går ut på att skapa en väderstation som består av en Raspberry PI (RPI) som tar emot data från flera sensorer och gör den tillgänglig för en internet-ansluten klient till exempel en laptop. Vinddata mäts av en analog anemometer med hjälp av en Arduino och temperaturen digitalt med RPin, vinddata förs över till RPin som tillgängliggör den på datorn via ssh.

## Material/utrustning

Mikrodator Arduino

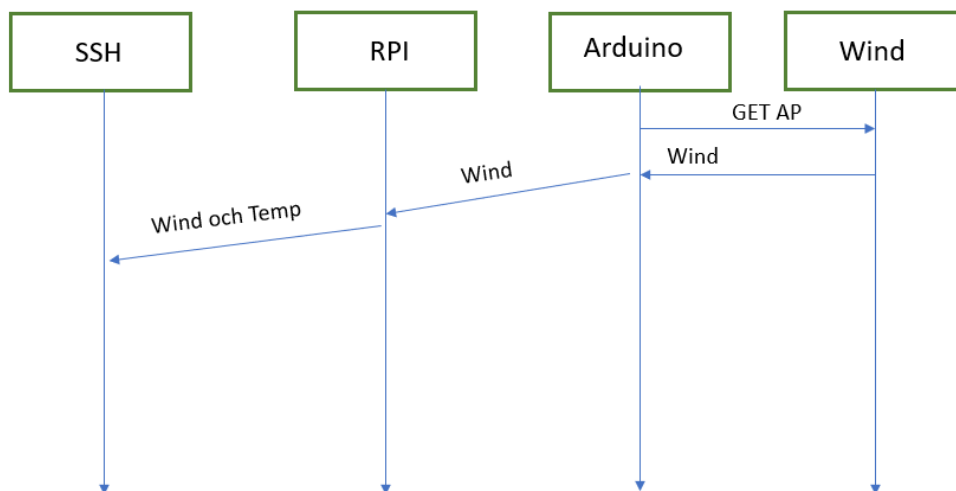
Enkortsdator Raspberry Pi

Breadboard

Temperatursensor DS1820

Anemometer Vindsensor (Art.nr: 41012900 elektrokit)

## Systemöversikt



Systemet består huvudsakligen av två delar, en Arduino UNO och en Raspberry pi.

UNO läser av anemometern analogt via analog-ingång A0.

Detta sänds via UART och USB från UNO till RPi, RPi behandlar och formaterar data-utskriften och skriver ut denna i terminalen. Språken är C++ (Arduino IDE) och Python 3.

RPi läser av tempsensor DS18B20 digitalt via dess 1-wire protokoll, denna sensor skickar värdet i 1000-tal i celsius. Informationen vilken skrivs ut i terminalfönstret eller motsvarande via protokollet SSH (Secure SHell) i detta fall. Både vind och temp beräknas för att ge rätt värdet vid utskriften, temp divideras med 1000 och vind med 10 och avrundas till lämpligt antal decimaler.

För att se resultatet se bifogad länk: <https://youtu.be/3ppya-GEBS4>

Exempel på utskrift: Temperaturen är: 22,5° C och vindhastigheten är 12,7 m/s.

## Kodbeskrivning

### Arduino UNO

Arduino UNO:n läser av anemometern analogt via analog-ingång A0. Gränsvärden sätts på det inkommande värdet med en constrain-funktion. Sedan sker en mappning som omvandlar värdet till en siffra mellan 0-324, vilket motsvarar vindhastigheten dm/s.

Detta skickas via UART/USB från UNO:n till RPin.

### Raspberry Pi

Pythonkoden som körs i RPin inleds med att ta emot seriell vinddata från Ardunion. Datan som tas emot ankommer kodad i ett format som är svårt att hantera i programmet. Därför körs funktionen `decode('ascii')` för att utvinna en sträng med det heltal och värde som önskas. Strängen innehåller dock fortfarande specialtecken som t.ex nyradstecken och vagnretur så dessa måste filtreras bort innan datan kan typomvandlas till integer, detta sker via två `.replace`-funktioner.

Då den seriella datan ankommer med hög uppdateringsfrekvens så blir det ibland fel vid avläsningen och korrupt eller inkomplett data kunde ibland läsas in. Detta löstes av en if-sats med `isnumeric`-funktion, genom att kolla om inkommande sträng endast består av siffror. Om strängen innehåller något annat än siffror gör inte programmet något alls. Detta förhindrar att bland annat oönskade nyradstecken, som följde med korrupt/ogiltig data, kommer ut i printsatsen. För att utskriften inte ska sakna en vindhastighet, skrivs det senaste giltiga värdet ut om korrupt data kommer in.

För att programmet ska skriva ut värden i terminalen på samma rad och skriva över senaste utskriften används en printsats. I printsatsen är första tecknet ett vagnreturstecken och det automatiskt utskrivna nyradstecknet i slutet av strängen är bortplockat. På så vis syns endast den senast uppdaterade utskriften.

## Resultat

Väderstationen fungerar och projektets kravspecifikation är uppfylld. Systemet är stabilt och de buggar som tidigare orsakade kraschar har motverkats genom att filtrera bort störningar som mätfel och kommunikationsstörningar i koden. Funktionalitet visas upp vid gruppens presentation av projektet.

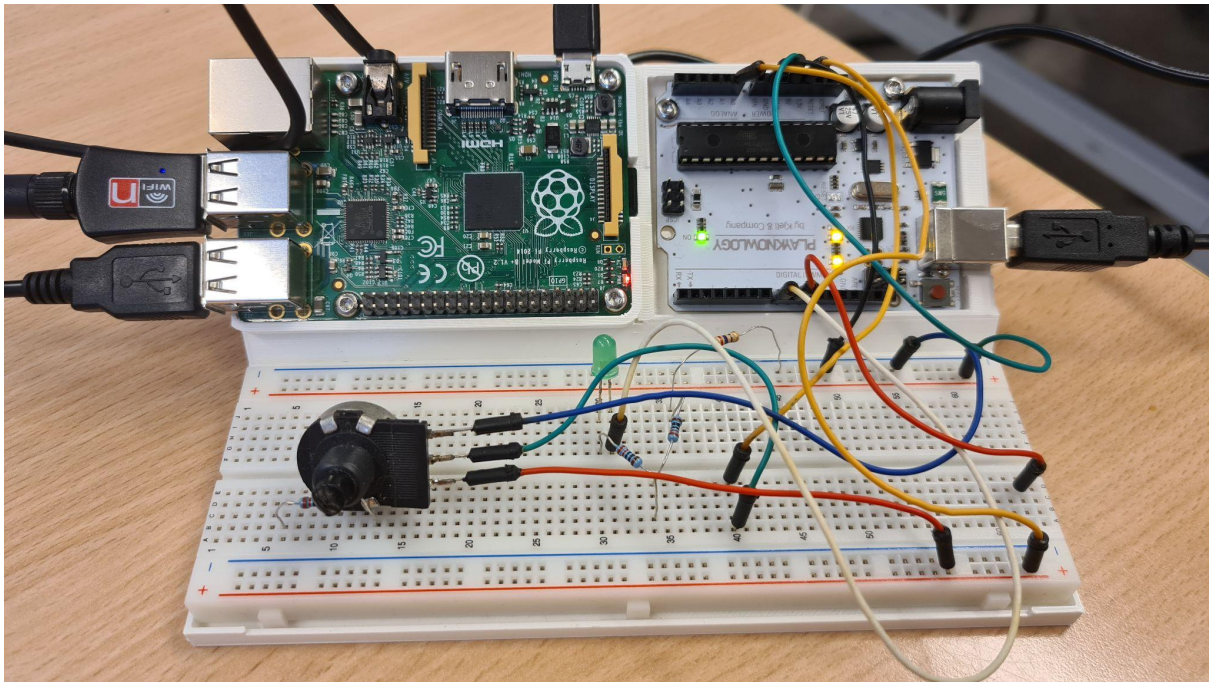


Bild på uppkoppling, potentiometern ersätter i detta fall anemometern.

# Referenser

[1] Erik Jagre, Inbyggda system 2 - Ei21 ht22, Projekt - Väderstation

<https://docs.google.com/document/d/1MRqNYjUiiVkmi8emllpr5zMccDobkZloldH49aSZcPo/edit>