

# Appendix 4 - Extra komponenter

## Adafruit CCS811 Air Quality Sensor

<https://learn.adafruit.com/adafruit-ccs811-air-quality-sensor/overview>

En sensor som mäter Total Volatile Organic Compound (TVOC) och equivalent carbon dioxide reading (eCO<sub>2</sub>).

Behöver kalibreras mot känd källa om absoluta värden ska användas (som alla liknande sensorer).

Detta ship använder så kallad I2C (Inter-Integrated Circuit) koppling. Detta är en typ av dataöverföring som funkar på vissa pin-s, liknande hur SPI fungerar. De flesta Arduino-kort har dessa pin-s markerade på kortet med namnen SCL och SDA. Kolla dokumentationen om du är osäker.

Detta kort ska kopplas enligt följande tabell.

Pin på kort	Pin på Arduino
Vin	+5V
GND	GND
SCL	SCL
SDA	SDA
WAKE	GND

3V3, RST och INT behöver inte användas. 3V3 matar ut 3,3V från den inbyggda spänningsregulatorn.

För att använda sensorn behövs biblioteket **Adafruit CCS811**. Ladda ner det via biblioteketshanteraren på samma sätt som tidigare. Du kan då ladda exemplet CCS811\_test för att se att allting fungerar som det ska och vilka funktioner som används.



# Adafruit VEML6070 UV Sensor Breakout

<https://learn.adafruit.com/adafruit-veml6070-uv-light-sensor-breakout/overview>

Detta är en sensor som mäter UV-ljus. Den mäter styrkan av UV-ljus den tar emot men ger inte en mått på UV-indexet. Denna sensor använder I2C-koppling enligt följande tabell.

Pin på kort	Pin på Arduino
Vin	+5V
G	GND
SCL	SCL
SDA	SDA

Biblioteket heter **Adafruit VEML6070**. Ladda ner det via biblioteketshanteraren som vanligt. Ladda exemplet `vemltest`. Om allt är kopplat rätt borde sensorn fungera.

Kommandot `VEML6070_1_T` kan bytas ut till

`VEML6070_HALF_T`,

`VEML6070_2_T`,

`VEML6070_4_T`,

för att ändra hur lång tid varje mätning tar. Längre tid ger högre precision men färre mätningar.

# Adafruit MSA301 Triple Axis Accelerometer

<https://learn.adafruit.com/msa301-triple-axis-accelerometer/overview>

En treaxlig accelerometer med ett antal extra funktioner.

Denna sensor använder I2C och har kontakter för en sladd, så man kan koppla upp den utan att löda något alls, men den kan också kopplas på vanligt vis.



Pin på kort	Pin på Arduino
Vin	+5V
GND	GND
SCL	SCL
SDA	SDA

Denna sensor använder två bibliotek som båda kan laddas ner via biblioteketshanteraren, **Adafruit\_MSA301** och **Adafruit\_BusIO**. I biblioteket Adafruit MSA301 ligger två exempel, `acceldemo` och `tapdemo`. `Acceldemo` är bra för att kontrollera att kortet är kopplat korrekt och se dess enkla funktioner. `Tapdemo` detekterar och skriver ut där sensorn blir knackad på. För mer detaljer om koden, se dokumentationen här, [https://adafruit.github.io/Adafruit\\_MSA301/html/class\\_adafruit\\_\\_\\_m\\_s\\_a301.html](https://adafruit.github.io/Adafruit_MSA301/html/class_adafruit___m_s_a301.html).

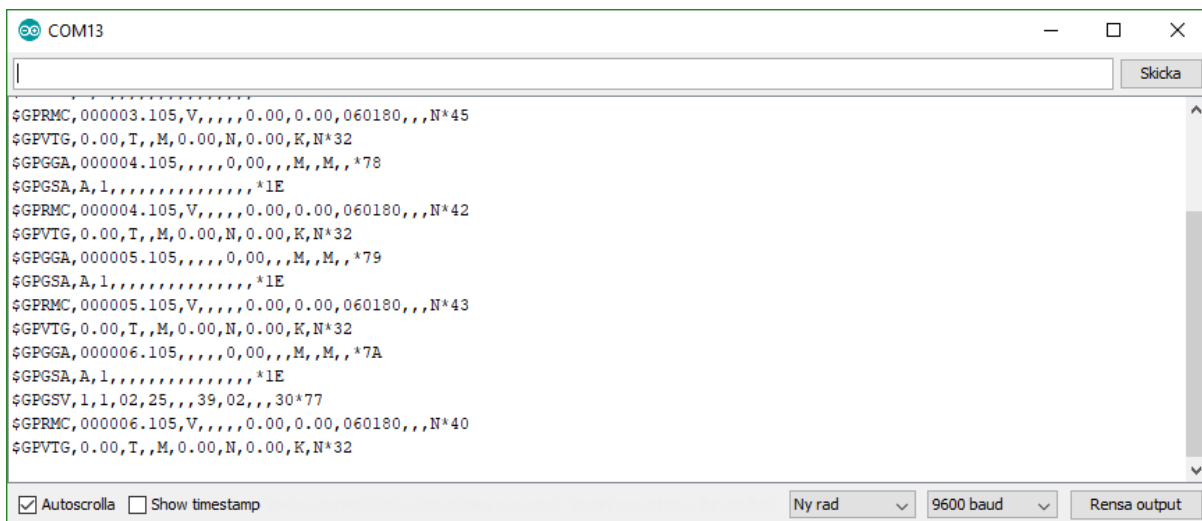
## Adafruit Ultimate GPS

<https://learn.adafruit.com/adafruit-ultimate-gps/overview>

En GPS-modul är ett attraktivt tillägg till en CanSat. Det är ett större kodningsprojekt som kan vara väldigt belönande eftersom att kunna se var ens satellit är gör det lättare att bestämma hur den rör sig.

Det finns GPS-moduler på många nivåer av användarvänlighet. Den enklaste vi kunde hitta var Adafruit Ultimate GPS. Den kräver bara två datakopplingar och har ett bibliotek för användande.

En GPS-modul som har ström kommer alltid att ge data, även om den inte har en kontakt med någon satellit. Det enklaste sättet att kontrollera funktionen på GPS-modulen är att låta informationen gå direkt till en dator (OBS! Detta kommer inte funka om ni använder en Arduino Leonardo. Hoppa i så fall över detta stycke). Detta gör man genom att koppla GPSens TX till Arduinos TX och GPSens RX till Arduinos RX, de kallas ofta pin 0 och 1 på Arduinon, och ladda upp en tom skiss till Arduinon (bara `void setup() {}` och `void loop() {}`). Koppla sedan in ström och öppna Serial monitor med 9600 baud. Ni borde se något som liknar detta



```
COM13
$GPRMC,000003.105,V,,,,,0.00,0.00,060180,,,N^45
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N^32
$GPGGA,000004.105,,,,,0,00,,M,,M,,^78
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPRMC,000004.105,V,,,,,0.00,0.00,060180,,,N^42
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N^32
$GPGGA,000005.105,,,,,0,00,,M,,M,,^79
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPRMC,000005.105,V,,,,,0.00,0.00,060180,,,N^43
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N^32
$GPGGA,000006.105,,,,,0,00,,M,,M,,^7A
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,1,1,02,25,,39,02,,30^77
$GPRMC,000006.105,V,,,,,0.00,0.00,060180,,,N^40
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N^32
```

Alla kommatecken betyder att det inte finns någon data eftersom modulen inte har kontakt med någon satellit. Datan innehåller olika typer av "meningar". Den som är mest intressant för oss är den som börjar \$GPRMC (Global Positioning Recommended Minimum Coordinates), den innehåller data om position och tid. Om man kan placera GPS-modulen utomhus eller vid ett fönster så den kan se himlen kommer den få kontakt med en satellit och börja ge faktisk data. Beroende på omgivningen och andra faktorer kan det ta längre tid att få kontakt.

Här är ett exempel på en GPRMC-mening när modulen har kontakt.

\$GPRMC,092535.000,A,5950.3561,N,01738.8539,E,1.09,157.56,220520,,A\*61

092535.000 = klockan i GMT

09 = 0900

25 = 25 min

35 = 35 s

.000 = millisek

A = Active, GPS har fix på satelliter

5950.3561,N = 59 grader, 50,3561 decimalminuter Norr

01738.8539,E = 18 grader, 38,8539 decimalminuter Öst


Notera! Inte decimalgrader, utan decimalminuter.

1.09 = Hastighet över marken i knop

157.56 = Färdriktning. Grader.

220520 = Dagens datum, 22 Maj 2020

A\*61 = Data checksum



Man kan kontrollera positionen genom att skriva in +59 50.3561', +17 38.8539' i sökfältet i Google Maps.

Att göra om detta till användbar data är *väldigt* mycket jobb, så vi använder ett bibliotek till det. Biblioteket Adafruit GPS Library finns i bibliotekshanteraren och hör ihop med denna modul. Det första exemplet vi ska prova är EchoTest. Beroende på vilken typ av mikrokontroller ni använder ska ni välja GPS\_SoftwareSerial\_EchoTest (nästan alla typer av mikrokontroller) eller GPS\_HardwareSerial\_EchoTest (Flora och FeatherWing). Läs i exempelkoden för att se vilka kopplingar som ska göras (TX - 8, RX - 7, eller TX- TX, RX- RX) och vilken hastighet Serial monitor ska ha, om denna är fel kommer den skriva ut massor av konstiga tecken. Dessa exempel borde ge samma resultat som direktkopplingen ovan, men här kan man se lite hur koden fungerar.

För att se exempel på hur koden faktiskt ska kunna användas, ladda exemplet som heter Parsing (föregås av lämplig typ av Serial). Denna kod kommer visa hur man använder GPS-klassen som biblioteket bygger upp.

I exemplet anropas `GPS.read()` varje varv i loopen för att läsa in data från modulen. Sedan avgör `GPS.newNMEAreceived()` om datan är ny, i så fall används `GPS.parse(GPS.lastNMEA())` för att tyda datan. Sedan kan en rad olika metoder användas, t.ex. `GPS.year`, `month`, `day`, `hour`, `minute`, `seconds`, `milliseconds` för att få tiden, `GPS.latitude` eller `longitude` för att få platsen, `GPS.altitude` för att få höjden osv.

## Sparkfun LSM6DD3

Detta är en treaxlig accelerometer och ett treaxligt gyroskop. Den är mycket flexibel och kan göra väldigt mycket. Den använder I2C.

Notera att denna sensor inte har en inbyggd spänningsregulator så den kommer bli förstörd om man matar den med 5V, den kan bara använda 3,3V.

Det enklaste sättet att använda denna sensor på är att koppla enligt följande tabell



Pin på kort	Pin på Arduino
Vin	+3.3V
GND	GND
SCL	SCL
SDA/SDI	SDA

och ladda ner biblioteket **Sparkfun LSM6DD3**. Ladda exemplet MinimalistExample för att se att kortet fungerar. För en mycket mer detaljerad beskrivning, se den officiella sidan här, <https://learn.sparkfun.com/tutorials/lsm6ds3-breakout-hookup-guide>.

## Sparkfun MPL3115A2 Pressure Sensor

Denna lufttryckssensor har hög precision och använder I2C. Den kan även mäta temperatur och höjd, vilket ger möjlighet att klara hela primary mission med ett kort.

Notera att denna sensor inte har en inbyggd spänningsregulator så den kommer bli förstörd om man matar den med 5V, den kan bara använda 3,3V.

Det enklaste sättet att använda denna sensor på är att koppla enligt följande tabell

Pin på kort	Pin på Arduino
Vin	+3.3V
GND	GND
SCL	SCL
SDA	SDA

och ladda ner biblioteket **Sparkfun MPL3115A2 Altitude and Pressure sensor breakout**. Ladda exemplet Pressure för att se att kortet fungerar. För en mycket mer detaljerad beskrivning, se den officiella sidan här, <https://learn.sparkfun.com/tutorials/mpl3115a2-pressure-sensor-hookup-guide>.