HÖGSKOLAN
I HALMSTAD

Mekatronikingengör 180 hp

EXAMENSARBETE

Våg till bikupor

Viktor Ringmark

Examensarbete 15 hp

Halmstad 2017-10-29

Sammanfattning

Projektet syftar till att utveckla en prototyp till en våg för bikupor som skall underlätta övervakningen av bins hälsa och honungsproduktion. Den ger också biodlarna en ökad tillsyn över bikuporna eftersom de ibland kan stå på avlägsna platser. Vågen skall automatisk väga bikuporna med jämna mellanrum och spara informationen på ett sådant sätt att man enkelt kan få en överblick över bikupans viktskillnad över tid. I rapporten beskrivs tillvägagångsättet för utvecklingsprocessen av vågen. Resultaten visar att vågen kan mäta vikten på en bikupa med en tillräcklig noggrannhet för att göra meningsfulla mätningar varje dag och vid olika tidpunkter.

Abstract

The project aims to develop a prototype for a scale for beehives that will facilitate the monitoring of bees health and honey production. It also gives the beekeepers an increased overview over the hives because the hives can sometimes be in distant places. The scale will automatically weigh the hives periodically and save information in such a way that one can easily get an overview of the hives weight over time. This report describes the approach of the development process of the scale. The results show that the scale can measure the weight of a beehive with sufficient accuracy to make meaningful measurements at multiple times every day.

Innehåll

1 Inledning

Honungsbin är viktiga för människan eftersom de tillverkar honung och vax samtidigt som de pollinerar växter och frukt. Utan pollineringen så minskar frukten och växterna i antal, kvalité och storlek. Jordbruksproduktionen är alltså beroende av pollineringen och honungsbin[1]. Antalet bin och biodlingar fortsätter att minska i USA och Europa[2]. Det är då viktigt för biodlare att kunna övervaka sina bikupor för att kunna se hur bikuporna mår såväl som hur mycket honung som producerats.

Anledningen till att man vill bevaka bikuporna är för att biodlaren skall få användbar information som kan användas för att sköta sina biodlingar på ett effektivare sätt.

Ett sätt att övervaka bikuporna är att väga dem vid jämna mellanrum för att ta reda på hur snabbt bina producerar honung över tid. Det är vanligt för biodlare att åka ut och väga sina bikupor manuellt med en simpel badrumsvåg eller liknande. Det innebär tunga lyft när man skall ställa bikuporna på vågen och extra transporter för att åka ut och kolla till bikuporna. Det kan förenklas med en våg som bikupan kan stå på konstant och som automatiskt gör regelbundna mätningar över långa perioder. Biodlaren kan sedan hämta den lagrade datan från vågen efter en viss period. Det innebär att man kan mäta mycket oftare utan att biodlaren behöva transportera sig till bikupan. Vågen kan också behandla datan som den får in för att ge biodlaren bättre överblick över hur produktionen i bikupan fortgår.

1.1 Syfte och Mål

Syftet är att bygga en prototyp/demonstrator som kan förenkla vägningen av bikupor för biodlare. Det ska ge biodlaren en noggrannare översikt över aktiviteten i bikuporna än tidigare. Huvuduppgifter för prototypen är som följer.

- Mäta viktskillnad på en bikupa regelbundet för att ge biodlare bättre översikt över deras bikupors aktivitet.
- Lagra mätvärden som visar bikupans vikt för att ge biodlaren en möjlighet att jämföra produktionen av honung under olika perioder.

1.2 Kravspecifikation

Nr	Krav
1	Skall mäta vikt.
2	Skall vara stadig och klara av vind
3	Skall lagra mätvärden
4	Skall automatiskt göra mätningar varje dag
5	Skall kunna användas på olika storlekar av bikupor
6	Skall klara av att väga 100 kg
7	Skall ha en upplösning på 100g som sämst

1.3 Frågeställningar

- Hur gör man tillförlitliga mätningar?
- Hur konstruerar man vågen så att den inte gör bikupan ostadig?
- Vilken mätmetod är bäst?
- Hur kalibreras och tolkas mätvärdena?
- Vad kan man göra för att hålla ner priset på utrustningen?

1.4 Avgränsningar

Att trådlöst kunna överföra mätvärden till biodlaren är en tillämpning som lämnas till ett uppföljande projekt. Resultatet är inte meningen att vara en fullfjädrad produkt utan en prototyp/demonstrator.

2 Bakgrund och Teori

I det här avsnittet så beskrivs olika tekniker och teorier för de olika sakerna som ingår i projektet samtidigt som några olika alternativ till dessa också värderas.

2.1 Kunskapsläge

För att mäta en vikt finns det flera olika metoder som kan användas. En av de äldsta tekniker är att man använder en balansvåg[3]. Balansvågen använder sig av en referensvikt vars vikt redan är känd. Man jämför denna med en okänd vikt genom att balansera de två vikterna mot varandra. På så sätt får man reda på den okända vikten.

En annan teknik är att använda en fjädervåg (dynamometer)[4] som mäter kraft genom att töja ut en fjäder. Genom att använda Hookes lag kan man sedan bestämma kraften som fjädern har utsatts för utifrån dess förskjutning. Tekniken är dock inte alltid så noggrann. Den mer moderna tekniken är att använda olika typer av lastceller. Några vanliga typer är t.ex. hydraulisk, pneumatisk, mekanisk och piezoelektriska lastceller. Dessa olika använder sig ofta av någon form av givare som antingen ger en elektrisk signal ut till en dator för att tolkas eller visar ett analogt värde med visarinstrument. Det är också den teknik som ger bäst noggrannhet och tar minst plats.

Systemen för inhämtning av data består av A/D-omvandlare och en mikrokontroller. Vissa mikrokontrollers har inbyggda A/D-omvandlare. A/D-omvandlare används för att konvertera en analog till digital signal så signalen skall kunna tolkas av mikrokontroll. En mikrokontroller är i stort sett en enkortsdator som kan användas för att styra och läsa av olika typer av elektronik.

Mätvärden som inhämtas av mikrokontrollern kan lagras på olika sätt. Data kan lagras på externa minnen som USB-minnen eller SD-kort. Alternativt kan man använda sig av trådlös kommunikation som Bluetooth, Wifi eller radio för att skicka direkt till användaren. Det kräver dock mer tid och pengar och används inte i detta projekt.

Systemet behöver någon typ av strömförsörjning för att fungera. Det är vanligast att man använder någon form av batteri om man inte har tillgång till det elektriska nätet. Man kan också använda en solcell till batteriet för att förlänga batteritiden.

2.2 Honungsproduktion

En bikupa i Sverige kan producera omkring 25-100 kg honung per år och i snitt ungefär 50 kg [5]. En bikupa kan alltså producera omkring 0.3 kg honung per dygn i optimala fall och i sämre fall omkring 0.075 kg.

2.3 A/D-omvandlare

För att tolka värden som kommer från en lastcell så behöver man en A/D-omvandlare. När en A/D-omvandling sker så omvandlas en analog signal till en

diskret signal (digital). A/D-omvandlingen kan göras på olika sätt. Ett sätt är genom spänning-frekvens omvandling[17]. Det innebär att en spänning omvandlas till ett pulståg med en frekvens som är direkt proportionerlig mot spänningen. Metodens fördelar är att den kan vara väldigt billig och noggrannheten beror på längden av tiden som frekvensen mäts. Det innebär att om man mäter över en längre tid så kan metoden vara mindre känslig för störningar. En annan metod är att använda sig av en A/D-omvandlare med t.ex. 24 bit för att få önskad upplösning. Upplösningen beror då på de antal bitar som A/D-omvandlaren kan hantera samt samplingsfrekvensen. Upplösningen kan beräknas med formeln 1 nedan.

$$V_{LSB} = V_{FS}/2^n \tag{1}$$

V_{FS} är det största spänningsområdet för A/D-omvandlaren där n är antalet bitar som A/D-omvandlaren har. Man kan säga att 2^n beskriver hur många delar som in-signalen delas upp i och den minst signifikanta biten representerar den minsta förändringen som kan ske.

Samplingsfrekvensen bör vara minst två gånger så stor som den analoga signalens bandbredd enligt Nyqvistteoremet[6] för att undvika för mycket brus och mätfel. Samplingen bestämmer hur ofta man tar mätvärden från den analoga signalen och om samplingsfrekvensen är för låg är det risk för vikning (aliasing). Vikning sker då signalfrekvensen är högre än samplingsfrekvensen. Det innebär att signalen som har en högre frekvens än samplingsfrekvensen ser ut som om den har en lägre frekvens än samplingsfrekvensen (frekvensen viks ner). Vikning gör alltså att signalen inte återskapas tillräckligt bra.

Det finns lite olika omvandlare t.ex. finns det omvandlare som använder Nykvist samplingsteoremet så att det samplar en signal två gånger snabbare än signalens frekvens eller mer. Sådana metoder kan t.ex. använda sig av omvandlartyper som flash, successiv approximation medans t.ex. sigma-delta metoden är översamplad.

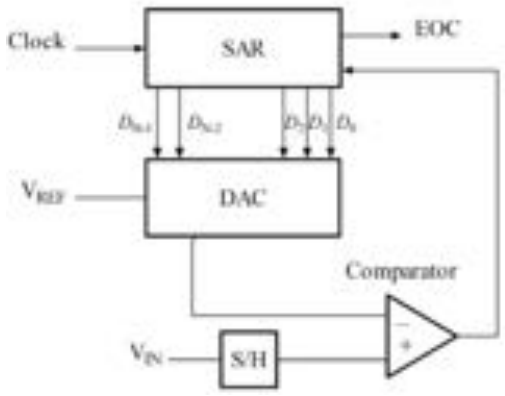
2.3.1 Flash

En metod som används i A/D-omvandlare är flash[7]. Man använder flash metoden för att den är väldigt snabb eftersom alla bitar omvandlas samtidigt. Den kan dock vara ganska dyr då man behöver $2^n - 1$ komperatorer i en flash med n-bitar.

2.3.2 Successiv approximation

Successiv approximation är en metod som används i A/D-omvandlare ger låg strömförbrukning[8]. Den använder sig av en komparator som är kopplad till ett successivt approximationsregister (SAR) som i sin tur är kopplad till en D/A-omvandlare. SAR går igenom n antal bitar i ordningen MSB till LSB och jämför varje bit mot in-signalen. Beroende på om in-signalen är större eller mindre än de värdet som D/A-omvandlaren ger ut så sätts SAR till en etta eller nolla. När

alla bitar i SAR registret har approximerats rätt så skickas det ut. Se figur 1 nedan för illustration.



Figur 1: Successiv approximation blockdiagram

2.3.3 Sigma-delta

En metod som A/D-omvandlare använder sig av är sigma-delta teknik [9] den har ofta hög upplösning och inte så dyr. Denna typ av A/D-omvandlare har tre delar. Den första delen är sigma delta modulatorn, den andra är ett digitalt filter (integrator) och de sista är decimatorn. I sigma delta modulatorn använder man översampling vilket innebär att man samplar många gånger mer än vad man behöver enligt Nykvist kriteriet. Det gör att signalen till brusförhållandet blir högre och signalen representeras då noggrannare. I modulatorn används också teknik som kallas "nois shaping" som används för att flytta bruset till högre frekvenser. Bruset som flyttas till högre frekvenser kan sedan filtreras bort med ett filter. Detta resulterar i en ännu bättre noggrannhet.

Det digitala filtret används dels som ett anti-aliasing filter och dels för att filtrera bort de högfrekventa kvantiseringsbruset som skapades från "nois shaping". Sedan går signalen till decimatorn som tar bort all överflödig information som vi fått från översamplingen. Decimatorn minskar då signalens samplingsfrekvens med 4 gånger.

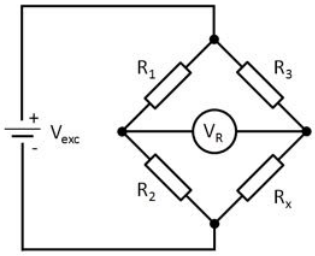
2.4 Signalförstärkning

Lastceller ger ofta ut en liten signal. För att kunna använda signalen ut från lastcellen så behöver man förstärka signalen så att den blir så stor att AD-omvandlaren kan konvertera den analoga signalen till digital. Till att börja med använder man sig av en Wheatstone brygga som sitter i lastcellen. Men ofta är Wheatsone bryggan inte tillräcklig utan man behöver en förstärkarkoppling också. Med en excitations spänning på 10 Volt in till en Wheatstone brygga så ger då bryggan en utspänning som ligger omkring 20 mV. En spänning på

20 mV är en liten spänning som behöver förstärkas för att en mikrokontroller skall kunna använda den. För att förstärka signalen kan man använda en OP-förstärkarkoppling. När man vill förstärka signaler från sensorer och olika typer av givare så är det vanligt att man använder en instrumentförstärkare. En instrumentförstäkarkrets består av tre operationsförstärkare och är gjord för att man skall få ut en noggrann förstärkarsignal för pålitliga mätningar.

2.4.1 Wheatstone brygga

En Wheatstone brygga[10] är en elektrisk krets som ofta används tillsammans med töjningsgivare. En Wheatstone brygga består utav fyra resistanser som är kopplade som två stycken spänningsdelare i en elektrisk krets, se figur 2 nedan. Mellan vardera spänningsdelare går en ledare som ger en utspänning. Spänningen mellan dessa ledare är utspänningen. Resistanserna kan vara varierbara som t.ex. en töjningsgivare och därmed förändras utsignalens spänning med resistansskillnaden i bryggan.



Figur 2: Wheatstone brygga

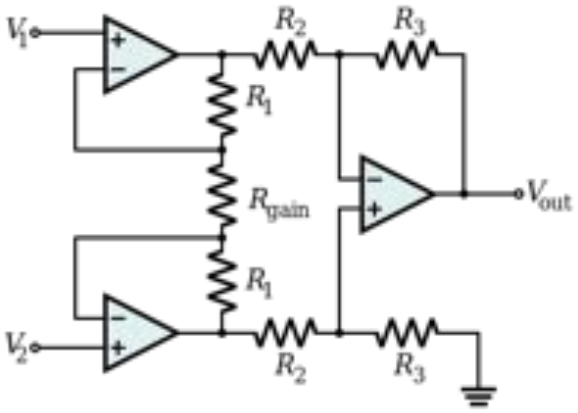
Om man vet de 4 olika motstånden så kan man använda formeln nedan för att få utspänningen.

$$V_R = \left(\frac{R_x}{R_3 + R_x} - \frac{R_2}{R_1 + R_2} \right) V_{exc} \tag{2}$$

Wheatstone bryggan används ofta till att göra om väldigt små resistansförändringar från töjningsgivare till en elektrisk signal. Bryggan kan användas med olika antal av varierbara resistanser t.ex. om två av resistanserna i bryggan är töjningsgivare så kallas det en halvbrygga osv. Halvbryggor används oftare på mindre lastceller som inte har lika mycket utrymme för töjningsgivare.

2.4.2 Instrumentförstärkare

En instrumentförstärkare[11] är uppbyggd av tre differentialförstärkare, se figur 3, och används oftast i sensorsystem. Den har ofta två signal ingångar där den förstärker spänningskillnaden mellan det två signalerna. Fördelarna med en instrumentförstärkare mot en OP-förstärkare är att den har högre in-impedans, hög CMRR och låg brusnivå vilket passar väldigt bra när man vill använda givare och sensorer för noggranna mätningar.



Figur 3: Instrumentförstärkare

CMRR står för Common-Mode Rejection Ratio och beskriver hur bra förstärkaren stöter bort 'common mode' signaler. Common mode signaler är signaler som uppkommer på båda ingångar med samma polaritet, fas och amplitud, dessa signaler vill man oftast inte ha eftersom de skapar brus. CMRR beskriver hur bra förstärkaren är på att undertrycka dessa signaler. Formel 3 ger en matematisk beskrivning av CMRR.

$$CMRR = \frac{A_v(d)}{A_c m} \tag{3}$$

I formeln är $A_v(d)$ spänningsskillnadens förstärkning och $A_c m$ är common mode förstärkningen. Desto högre CMMR desto bättre är förstärkaren på att undertrycka common mode signaler.

2.5 Lastceller

I projektet vill man ha en så billig lösning som möjligt med bra noggrannhet och tålighet. Det finns några olika typer av lastceller att välja mellan.

2.5.1 Töjningsgivare lastcell

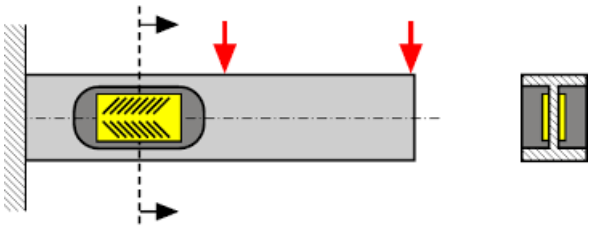
En lastcell[12] med töjningsgivare registrerar deformationen av lastcellen genom att mäta skillnaden i resistans under deformationen genom att använda sig av töjningsgivare. Töjningsgivarnas resistans på lastcellen varierar med deformationen. Dessa resistanser kopplas ofta i en Wheatstone brygga för att få en spänning som man kan mäta istället för en resistans. Lastcellen ger vanligtvis ut en liten spänning som ofta kräver någon typ av förstärkarkrets för att signalen skall bli tillräckligt stor för att hanteras av en mikrokontroller eller A/D-omvandlare.
Fördelar:

- Billigare än pneumatisk/hydraulisk
- Enkel design
- Inga rörliga delar
- Hög noggrannhet

Nackdelar:

- Kräver skydd för väder eller liknande.

I en lastcell sitter ett visst antal trådtöjningsgivare. En vanlig typ av lastcell ser ut som en balk med töjningsgivare på ovansidan och undersidan. För exempel se figur 4 nedan.



Figur 4: Lastcell med trådtöjningsgivare

När balken deformeras av en kraft så kommer de trådtöjningsgivare som sitter på ovansidan ge en större resistans eftersom de dras ut av deformationen. Den trådtöjningsgivare som sitter på undersidan istället ger en mindre resistans eftersom den trycks ihop. Skillnaden i resistans mellan de olika trådtöjningsgivare erhålles med hjälp av Wheatstone bryggan som också omvandlar resistansskillnaden till en spänning.

2.5.2 Hydraulisk lastcell

Principen för en hydraulisk lastcell[12] är liknande som för en balansvåg. I grova drag läggs en kraft på en kolv som trycker ihop ett membran som i sin tur trycker undan ett flytande medium (olja) vilket skapar ett tryck som läses av en tryckgivare.

Fördelar:

- Klarar av starka stötar
- Klarar stora temperaturskillnader

Nackdelar:

- Dyr
- Komplicerad design

2.5.3 Pneumatisk lastcell

En pneumatisk lastcell[12] fungerar så att en kraft läggs på ena sidan av ett membran och ett lufttryck läggs på andra sidan för att få jämnvikt. Luft far då ut ur ett munstycke. Kraften mäts sedan av en tryckgivare i lastcellen. Membranets böjning påverkar trycket i lastcellen såväl som luftflödet genom munstycket.

Fördelar:

- Säkra att använda i industri utan risk för explosion.
- Klarar stora temperaturskillnader

Nackdelar:

- Kräver ren torr luft.
- Komplicerad design

2.5.4 Pizoeletrisk lastcell

När ett Piezoelektriskt material till exempel en kristall utsätts för en kraft så bildas det en laddning på ytan[13]. Laddningen läser man sedan med hjälp av en laddningsförstärkare för att ta reda på kraften. Tekniken är dock inte passande för mätning under längre tid eftersom laddningen avtar med tiden.
Fördelar:

- Bra i dynamiska applikationer.
- Klarar stora temperaturskillnader

Nackdelar:

- Sämre linearitet än trådtöjningsgivare
- Sämre stabilitet över tid än trådtöjningsgivare.

2.6 Mikrokontrollers

För att ta kunna ta hand om datan från lastceller och lagra dessa värden så krävs någon form av mikrokontroller. De finns många att välja bland med olika för-och nackdelar. Vanliga mikrokontrollers är t.ex. Arduino uno eller olika pic controllers.

2.6.1 Arduino uno

ATmega328P[14] är en 8-bit mikroprocessor som ofta sitter på utvecklingsplattformen som kallas Arduino uno. Den har 14 digitala input/output pinnar varav 6 st kan generera PWM utsignaler och 6 st kan läsa analoga signaler med 10-bit upplösning. Arduino uno förenklar arbetet eftersom det mesta är färdig installerat för att man skall kunna programmera den så snabbt som möjligt.

2.6.2 PIC controllers

PIC microcontrollers[15] har en rad olika mikrokontrollers som finns med 8, 16 och 32-bitar. Det är ofta billiga och det finns många varianter med olika många input/outputs, analoga ingångar, PWM mm. Det finns också med stöd för USB kontakt. PIC är dock inte lika lättanvänd som en Arduino då den behöver lite mer arbete för att sättas upp på rätt sätt. Fördelen är att man kan handplocka exakt de komponenter som man behöver. Det innebär att man inte behöver betala för komponenter som man inte har användning för som man annars kan få göra om man köper en Arduino uno t.ex.

2.7 Klockkrets

I projektet behövs det en klockkrets[16] för att man skall kunna få tidsinformation om när mätningar sker. Klockkretsen behövs också för att bestämma när mätningarna ska ske på ett noggrant sätt. En klockkrets som visar tiden i realtid kan användas som tidsreferens för mikrokontrollers. Klockkretsen har ofta ett eget batteri så att den kan hålla tiden oberoende av mikrokontrollern samt ett alarmsystem. Med alarmsystemet kan klockkretsen skicka pulser till en mikrokontroller vid bestämda tider. Det kan användas för att trigga en interrupt på en mikrokontroller t.ex. Ibland har man en temperatursensor till klockkretsen som korrigerar eventuella fel som uppstår när klockkretsen utsätts för temperaturskillnader. För kommunikation mellan klockkretsen och mikrokontrollern så används gränssnittet I^2C.

2.8 Strömförsörjning

För att vågen skall fungera så behöver den någon typ av strömmförsörjning. Biodlare har oftast inte tillgång till det fasta elnätet ute vid deras kupor. Därför kan det vara bra att använda ett batteri. Vill man att batteriet skall hålla längre så kan man lägga till en solcell. Då krävs det ett batteri som är laddningsbart.

Om ingen solcell behövs så kan man eventuellt använda utbytningsbara batterier. Hur länge ett batteri håller beror på batteriets kapacitet som beskrivs i Amperetimmar (Ah). För att beräkna tiden som batteriet kan driva en krets så behöver man veta hur mycket energi som kretsen drar från batteriet. Med formel 4 och 5 nedan kan man då räkna ut hur lång tid batteriet orkar driva kretsen. P är effekten, U är spänning, I är ström samt h är tid.

$$P = U * I <=> \frac{P}{U} = I \tag{4}$$

$$\frac{Ih}{I} = h \tag{5}$$

Formel 5 visar att batteriets kapacitet Ih dividerat med strömmen I som dras från batteriet ger batteritiden h.

2.9 Minne

För att spara informationen som mikrokontrollern får in och bearbetar så behöver man ett minne där informationen kan lagras. Vi vill använda ett externt minne som lätt går att ta med sig. Det finns lite olika typer att välja mellan men mest vanliga är USB-minne och SD-kort.

2.9.1 USB-minne

USB-minnen är flashminnen som en dator kan skriva till och läsa från. Det finns olika storlek på minnena men i vårt fall räcker det med ett relativt litet minne eftersom vi bara behöver lagra ett Excel dokument i princip. Vi har inte heller några krav på skrivhastighet mellan minne och dator vilket gör att man kan använda ett billigare USB-minne.

2.9.2 SD-minne

SD-minne använder sig av SPI för att bli läst och skrivet till. Fördelen med SD-minne är att modulen som krävs till Arduino är billigare än den för USB. SD-kort är också väldigt enkelt att använda tillsammans med Arduino.

2.10 Gränsnitt

Vid kommunikation mellan elektronik så behövs det ett förbestämt sätt att kommunicera på och då använder man olika typer av gränssnitt. Här beskrivs några vanliga gränssnitt.

2.10.1 Serial Peripheral Interface (SPI)

Serial Peripheral Interface eller SPI används oftast för snabb kommunikation över korta avstånd inom elektroniken. SPI är ett gränssnitt som man ofta

använder för kommunikation i båda riktningar mellan mikrokontrollers och olika periferienheter. En av enheterna styr klockan (oftast mikrocontrollern) och kallas då Master och den styrda enheten kallas slave. Man kan bara ha en Master men man kan ha flera slaves. SPI använder sig av 4 signaler som kallas SCKL, MISO, MOSI och SS. SCKL är klocksignalen som styr när mottagaren ska läsa bitarna från inkommande data eller när man skall ta emot data. MISO står för Master IN Slave Out och MOSI står för Master Out Slave In och fungerar som databussar. Sen finns det SS som står för Slave Select och används för att väcka en slave för att kunna läsas/skrivas till. Den används oftast när det finns flera slaves.

2.10.2 I^2C

I^2C (Inter-Integrated Circuit) är en multi-master, multi-slave och seriell datorbuss som Philips Semiconductor har skapat. Det används ofta till kommunikation med låg hastighet mellan periferienheter och mikrokontroller i kortdistans.

2.11 Databehandling

För att kunna förstå informationen som Arduinon får in så behöver man behandla den inkommande datan på ett sätt som förenklar förståelsen för informationen. För att få bra jämna värden kontinuerligt så kan de vara bra att behandla den inkommande datan. Eftersom det finns risk för blåst och regn kan mätvärden variera mer en önskat.

2.11.1 Normalfördelning

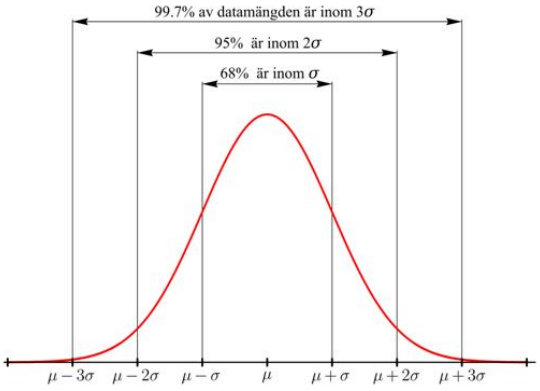
Normalfördelning användas för att göra matematiska analyser inom sannolikhetsläran. Funktionen för normalfördelning kan beskrivas som sannolikhetstäthet. Det är ett mått på hur stor sannolikheten är att ett värde kommer att hamna inom ett visst intervall kring ett medelvärdet. Se formel 6 nedan .

$$P(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{6}$$

För att få reda på i hur stor utsträckning som normalfördelnignsfunktionen är centrerad kring medelvärdet så används formel för standardavvikelsen. Standardavvikelse beskriver hur mycket som värdena avviker från medelvärdet. Se formel 7 nedan för matematisk beskrivning av standardavvikelsen

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}. \tag{7}$$

En kurva som är normalfördelad skall se ut som i figur 5 nedan. Figuren



Figur 5: Normalfördelningskurva

visar att 99.7 procent av alla mätvärden skall ligga inom 3 standardavvikelser, 95 procent inom 2 och 68 procent inom en standardavvikelse för att kurvan skall vara normalfördelad.

2.11.2 Medelvärde

För att minska fel så kan man ta ett flertal mätvärden och genom matematik beräkna vilket tal som är den bästa uppskattningen av mätningen. Man kan då använda sig av medelvärde enligt formel 8 nedan.

$$M(x) = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{8}$$

Medelvärdes beräkning fungerar bäst om man vet att värdena som mäts är normalfördelade. Det betyder alltså att man vet att det absolut flesta värdena kommer att hamna mycket nära medelvärdet.

2.11.3 Median

Ett annat sätt att få ut ett värde som kan representera en typ av medeltal är att beräkna medianvärde. Medieanvärde får man genom att arrangera en serie av värden i nummerordning och sedan ta mitten värdet. Om det finns mer än ett mitten värde tas medelvärdet av dessa till medianvärde. Detta kan användas t.ex. om medelvärdet dras upp eller ner för mycket av ett fåtal värden och ger ett missvisande resultat.

2.11.4 Typvärde

Ett annat sätt är att använda typvärde. Typvärdet är det värde som förekommer flest gånger i en sekvens av värden. För att få ändå bättre värden så görs flera

mätningar under en liten period, flera gånger, var fjärde timma. Detta görs för att upptäcka avvikande mätvärden. Avvikelser kan bero på t.ex. vind eller regn.

Vidare om värdet som ges av en mätning visar ett mindre eller mycket större värde än den föregående mätningen så görs en ytterligare mätning. Den extra mätningen görs ifall att den föregående mätningen gjordes under en tillfällig störning.

2.12 Skydd för väder

För att skydda elektroniken och last cellerna från väder och vind så finns det lite olika metoder att använda. Det finns lastceller och elektrisk utrustning som har olika kapslingsklassningar som visar hur bra den elektriska utrustningen kan hantera fukt, vatten, och damm mm. På engelska kallas det IP code som betyder international protection marketing. Kvalifikationen ges i formen IP följt av två siffror. Den första siffran visar hur bra produkten skyddas mot damm och inträngande föremål. Den andra siffran visar hur bra den tål vatten. Ju högre siffran är desto bättre. Siffran för damm går från noll till sex och siffran för vätska går från noll till nio. Utöver att välja delar med rätt klassning så kan man själv göra inkapslingar för att hålla fukt och damm borta. Det är också viktigt att välja produkter som klarar temperaturen vi har utomhus i Sverige.

2.13 Forskning och relaterade arbeten

Det finns en rad olika forsknings arbeten som på olika sätt mäter och övervakar bikupors aktivitet.

2.13.1 U. Warnke, Bees birds and mankind

I denna artikel[19] skriver man bland annat om hur viktiga bina är för oss och att man tror att bina står för 85% av pollineringen av frukt och växter vilket gör att det kan räknas som ett av de viktigaste produktions djuren för människan. Man skriver också om hur antalet bin minskar och varför det kan bli ett problem. Bikolonier ser en minskning av antalet bin runt om jordklotet och man är inte säker på orsaken. De kolonier som får en så kallad 'colony collaps disorder' gör att bina försvinner och de som är kvar är svaga och orkar inte tillverka lika mycket honung.

2.13.2 System Architectures for Real-time Bee Colony Temperature Monitoring

Artikeln[20] beskriver olika metoder för att övervaka förändringar i temperatur och luftfuktighet i bikupor under real-tid. Man beskriver några olika metoder och väger dem mot varandra. Man menar att mätning av temperatur och luftfuktighet i bikupan är både enkelt och billigt. Bin är kallblodiga och deras aktivitet blir då också beroende av temperaturen i bikupan vilket då gör att det är intressant att mäta temperaturen inne i bikupan.

2.13.3 Design and Development of a Smart Weighing Scale for Beehive Monitoring

Tidigare har det gjorts ett liknande projekt [18] där syftet var att mäta bikupans viktförändring och sedan skicka datan via zigbee radio. I det systemen ingick också luftfuktighetssensor, temperatursensor och solpanel. Sensorerna för att mäta luftfuktighet och temperatur används för att kompensera för lastcellens mätfel vid förändring i luftfuktighet och temperatur. Man använde sig av en 'single point' lastcell som klarar av 750 kg och en noggrannhet på 0.02% av max kapacitet. Till den så användes en 24-bit A/D-omvandlare för en hög noggrannhet. Och man kunde göra mätningar som gav viktförändringar i tiotals gram. Skydd mot väder och stabilitet för vind och stötar utelämnades i projektet.

2.14 Slutsats av bakgrund och teoridel

I teori och bakgrundsdelen framgår det att den är trådtöjnings lastcell som är det smartast valet p.g.a. dess enkelhet, hållbarhet och pris. Arduino är det kortet som är billigast och mest användarvänligt. Det medföljer också bra bibliotek som kan förenkla programmerandet. Strömförsörjningen med batteri kommer att användas främst med tanke på priset och eftersom det är en prototyp så är funktionaliteten viktigast. Som minne så är det SD-kort som är enklast och billigast p.g.a. SPI kommunikation och billig SD-korts modul.

3 Metod

Projektet delades upp i delar som skall göras i en viss ordning. I dessa ingick konstruktionen som innefattade skruv och bult, svetsning, borrning och själva vågen skulle bära upp bikupan, alltså, vågens utformning; Elektronikdelen där AD-omvandling och klockkrets, Modulen för SD-kortet och Arduinon ingick; Minne i form av SD-kort; Programmering där det olika gränssnitt användes t.ex. SPI eller I^2C. I programmerings delen skrivs också kod för spara energi och för databehandling. Det kontrollerar också hur mätresultaten skrivs till SD-kortet på ett strukturerat och lättläsligt sätt. Se tabell 1 nedan över det olika delarna.

Tabell 1: Projektdelar

Delar	Innehåll
Konstruktion	Material och konstruktion där ingår borrning, svetsning, bultar och stålrör
Elektronik	Klockkrets, AD-omvandlare, Arduino, och SD-korts modul
Programmering	Gränssnitt, bibliotek och lätt databehandling.
Minne	Struktur av information på SD-kort

3.1 LIPS Projektmodell

Projektet är i grunden baserat på LIPS projektmodell som ger projektet en grundläggande struktur. Modellen delas in i 3 faser. Det tre faserna är före, under och efter fasen varav varje fas innehåller vissa beslutspunkter och milstolpar. I första fasen planeras projektet och man gör förundersökningar och skriver en kravspecifikation. I andra fasen börjar man utförandet av projektet. Det innefattar programmering, design och mekanisk konstruktion samtidigt som man skriver rapport. I efterfasen så görs tester och man fastställer resultat och en utvärdering av projektet sker.

3.2 Tillvägagångssätt

Under projektets gång dokumenteras framskridandet i en typ av dagbok. Projektet började med att dess olika delar definieras i en grov kravspecifikation. Projektet delades upp i 4 delar bestående av strömförsörjningsdel, databehandlingsdel, datalagringsdel och en vågdel. Utifrån de olika delarna gjordes undersökningar på olika typer av lastceller, A/D-omvandlare, datalagring och strömförsörjning i den ordningen. Sen valdes komponenter till de olika delarna baserat på den information som inhämtats under undersökningarna.

3.3 Val av Elektronik

I denna del beskrivs det val som gjordes i elektronikdelen.

3.3.1 Klockkretsen

För att kunna visa vid vilken tidpunkt som mätningarna tagit rum så användes en realtids klockkrets. Den pratar med Arduinon via I^2C protokoll. Det är en billig krets som är väldigt exakt och har en temperatursensor som korrigerar för eventuella temperaturskillnader som skulle kunna påverka klockan. Alternativen var att använda Wifi eller radio för att synka Arduino till en klocka. Det fanns också ett fulare och inte så exakt sätt där man bara gjorde en klocka genom programmering.

3.3.2 SD-kortsmodul

Kortmodulen kommunicerade med Arduinon via SPI vilket är bra eftersom det inte fanns något rum för fler enheter som kommunicerar med I^2C. Utöver SPI kommunikation så valdes SD-korts modulen eftersom den var enkel och det billigaste lagrings alternativet. En USB-modul övervägdes också men den blev dyrare eftersom den behövde mer elektronik för att kunna kommunicera med Arduinon. Den extra elektroniken innebär också en svårare programmering. Ett SD-kort räcker gott och väl och är billigt.

3.3.3 A/D-omvandling

Vid val av teknik för att läsa av mätvärden med en mikrokontroller så var vanlig A/D-omvandling och spänning till frekvens omvandling alternativen.

Användning av en 555-timer som en spänning frekvensomvandlare testades och vägdes mot en 24-bit A/D-omvandlare. Noggrannheten för spänning/frekvens omvandling bestäms beroende på hur länge man mäter frekvensen från spänning/frekvens omvandlingen. Det gör också att den inte blir så känslig för störningar. Timern fungerade inte med önskad noggrannhet vad gäller spänning mot frekvens värden. Det var också svår att få något bra konstant frekvensvärde värde från Arduinon. AD- omvandlaren som använde sig av sigma-delta teknik fungerade däremot utmärkt, den kan mäta med en noggrannhet på 10-tals gram och har en inbyggd förstärkning. Detta gjorde att valet blev att använda en 24-bit A/D-omvandlare istället för 555-timern och spänning/frekvensomvandling. Om man skulle försökt med en dedicerad spänning frekvensomvandlare så hade priserna för AD-omvandlaren och V/F omvandlaren legat i samma prisklass.

3.3.4 Lastcell

En förstudie gjordes där man inhämtade information om olika typer av lastceller. Typer som pneumatisk och hydrauliska lastceller uteslöts när man vägde dem mot en mycket billigare lastcell med trådtöjningsgivare. Dessa var också onödigt

komplicerade för vårt användningsområde. En lastcell med töjningsgivare i helbrygga valdes eftersom dess IP klassning var hög (IP65) och noggrannheten bra samtidigt som den var lätt att få tag på. Den är också enkel att använda i en konstruktion. Eftersom den redan har hål att fästa i, vilket gör den enkel att integrera i en konstruktion. Valet baserat också på pris och på stabilitet. Det stod till slut mellan en 'singel point' lastcell eller 4 st små lastceller med halvbryggor som är en teknik man ofta använder i vanliga badrumsvågar. Valet blev 'single point' lastcellen p.g.a dess noggrannhet och hållbarhet.

3.3.5 Gränssnitt

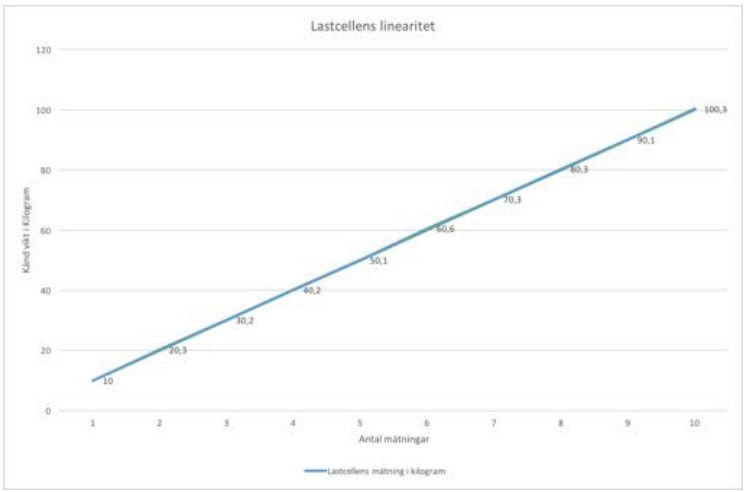
Gränssnitten som används för kommunikation är SPI för SD-korts modulen och I^2C som används för att prata med klockkretsen. Dessa fungerar bra till Arduino-kortet eftersom de inte behöver använda samma sorts pinnar för kommunikationen. Det var de gränssnitt som de olika modulera använde så det fanns inte något annat gränssnitt att välja mellan.

3.4 Val av mikrokontroller

Valet av mikrokontroller styrdes av pris, användarvänlighet och antal I/O pinnar. Arduinon har många pinnar, är billig och enkel att använda direkt. Pic controller är väldigt billig men kräver mer arbete för att börja programeras, den kan också behöva en del extra elektronik som arduinon redan har. Arduino uno valdes p.g.a användarvänligheten och att den möjliggör ett snabbt sätt att komma igång med programmeringen och elektroniken.

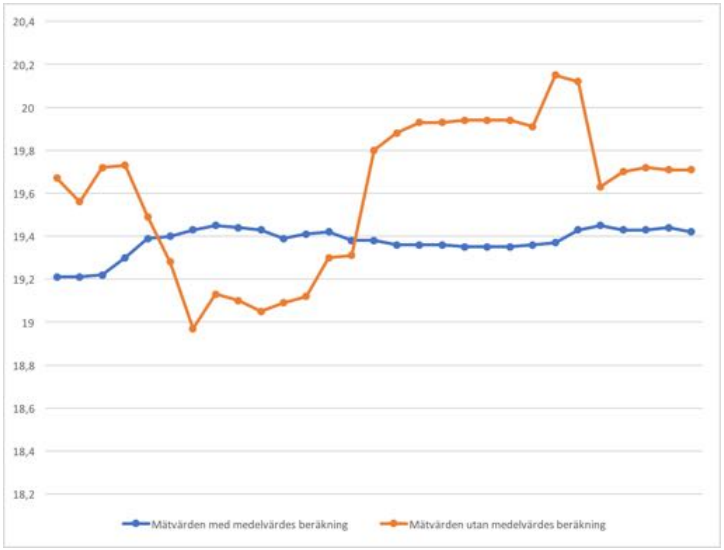
3.5 Databehandling

Databehandlingen sker efter att den analoga signalen från lastcellen har förstärkts 128 gånger och gått igenom A/D- omvandlaren. Värdena går då till en början inte att förstå eftersom de bara visar ett tal som representerar en spänning. Arduinon kalibreras då så att värdena beräknas till vikt i kg. Detta görs genom att lägga 1 kg på lastcellen, vi får då ett värde och vet då att det värdet är samma som ett kilo och på så sätt kan man skriva om så att värdena representeras i kilogram. Test som visar att lastcellens spänning i förhållande till vikt är linjär görs också. Testet görs genom att lägga på flera kända vikter i stigande ordning upp till 100 kg på lastcellen och jämnföra det mot vad lastcellen ger för vikt. Se I figuren 6 nedan.



Figur 6: Lastcellens linearitet

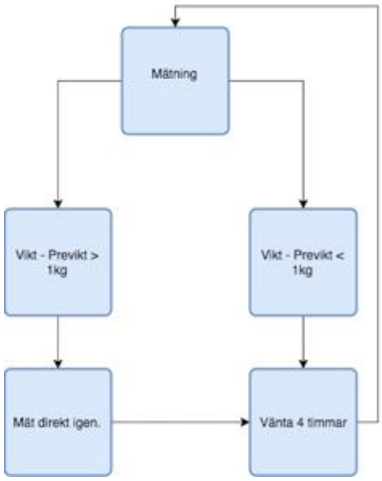
Vid test med konstant vikt så varierar mätvärdena ca 1 kilo. Mätningar görs en gång i minuten för att snabbare få testresultat och mer data. Se figur 7 nedan på mätvärden vid enkel mätning. Eftersom värdena ändrades för mycket så gjordes i stället flera mätningar varje mätnings tillfälle. Av dessa värden räknades medelvärdet ut. Värdena blev därmed stabilare men visade ändå skillnad på ett par hekto under en timme.



Figur 7: Mätvärden med och utan medelvärdesberäkning

För att få ytterligare stabilitet så gjordes tre medel-talsmätningar under en och en halv sekund som man också beräknade medeltalet på. Det resulterade i att värdena bara varierade 10 gram ungefär under en timme. Det illustreras i figur 12 som finns i resultatdelen.

Rättvisast mätningar görs på natten då alla bin är hemma men mätningar görs också under dagen också för en noggrannare överblick. Det görs 6 st mätningar under ett dygn med 4 timmars mellanrum. På så sätt får man en överblick hela dygnet. För att minska fel som beror på vind och regn så används en liten algoritm. Algoritmen säger att om föregående mätvärde har en för stor skillnad mot den senaste mätningen så görs en extra mätning strax inpå. Om det inte är någon stor skillnad så fortsätter vågen med mätningar som vanligt. I figuren 8 nedan visas ett flödes schema.



Figur 8: Mätvärdes algoritm

Medelvärde och att gör flera mätningar ansågs som bästa sättet att få jämna pålitliga mätningar. Andra sätt som att t.ex. beräkna medianen eller ta typvärde övervägdes. Medianen hade vart bra att använda om enstaka värden avviker väldigt mycket i mätningarna. Det ansågs dock inte att värderna ändrades tillräckligt mycket för att användandet av medelvärdet skulle bli missvisande. Typvärde visar ju det värdet som förekommer flest gånger och det tänktes användas främst som ett komplement till medelvärdet. Medelvärdes mätningarna fungerar dock väldigt bra så att använda typvärde också ansågs överflödigt.

3.6 Strömförsörjning

Strömförsörjning väljes p.g.a. budget till laddningsbara batterier som vid senare tillfälle kan laddas med solceller. Solceller används dock ej i detta projekt p.g.a. tid och pengar.

3.7 Konstruktion

Konstruktionen byggs i rostfritt stål eftersom det var enklast att få tag på och håller bra utomhus. En simulering av konstruktionen gjordes i Solidworks för att se om konstruktionen och materialet klara de krafter som den förväntas. Det var även väldigt billigt då de skänktes bort till förmån för projektet. Konstruktionen byggdes så att måtten skulle likna en medelstor bikupa, eftersom det flesta bikuporna har närliggande mått. Bikupor finns i många modeller gjorda på olika sätt så de bestämdes att en medelstor plattform för vågen passar bäst. Det gör inte något att bikupan är lite för liten eftersom det ges möjlighet att förstora plattformen om man har större bikupor.

3.8 Minne

Tanken var att USB-minne skulle användas men efter noggrannare eftertanke så användes ett SD-minne istället. Det visade sig att en Arduino uno är en USB slave och ett USB-minne är också en slave, dessa kan inte prata med varandra. För att Arduinon skall kunna skriva till USB-minnet så måste Arduinon vara en host/master. Det finns host moduler till Arduinon men dessa blir dyrare än SD-korsmodulerna och kräver oftast mer programmering för dem att fungera. Ett SD-kort använder sig av SPI och är mer öppet i sin kommunikation och behöver inte ha en host funktionalitet. Det gör att de SD-korts modulen blir billigare och också enklare att använda än en USB-host modul.

3.9 Programmering

Programmeringen görs i Aruduinos egna IDE i språket C som har lite extra funktioner som är en påbyggnad som är specifikt för Arduino kortet. Vid programmeringen användes bibliotek för kommunikation med SD-kortet och Klockkretsen som använder sig av SPI och I^2C.

3.10 Utförande

In denna del beskrivs utförandet av projektet i sina olika delar.

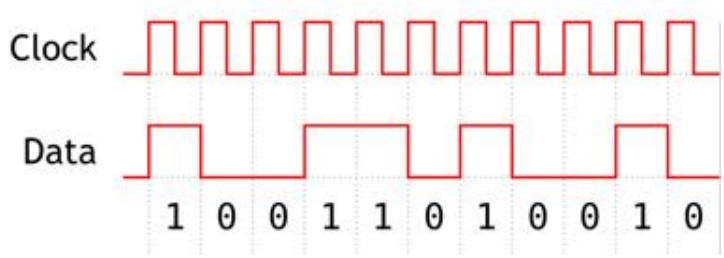
3.10.1 Konstruktion

Den tänkta konstruktionen ritades upp i Solidworks [22]. Där gjordes en kraft/stress simulering som visade att konstruktionen skulle hålla utan att deformeras under 100kg för stål och aluminiumlegering vilket möter det krav som ställts. Erhöll material som fyrkantsrör, bult samt verktyg som vinkelslip och pinnsvets. Delade ett rostfritt rör i 8 delar med vinkelslip samt fasade kanterna. Dessa bitar svetsades sedan ihop till två stycken likadana konstruktioner med pinnsvets. De olika delarna fästes samman genom lastcellen. Konstruktionen är gjord så att man skall kunna stoppa in mindre stänger i fyrkantsrören för att göra plattan större för större bikupor. En platta som lastcellen skall sättas i svetsades också på mittenstängerna med utmätt avstånd så att lastcellens ände kommer i centrum på vågen. Därefter mättes hålen ut på plattan som skall matcha lastcellens hål. Dessa håll mättes noggrant och för-borrades med ett mindre borr och senare med ett 7mm borr. Sedan användes torx bultar till att fästa lastcellen vid plattan och konstruktionen. Konstruktionen blev p.g.a. värmen från svetsen lite sned, det korrigerades så gott det gick med lite knackning med slägga. Se figur 9a och 9b nedan.

Figur 9: Våg konstruktion

3.10.2 AD-omvandling

I projektet används en A/D-omvandlare med Sigma Delta teknik som har en programmable gain amplifier (PGA) där man kan välja mellan 32, 64, och 128 ggr förstärkning. Då 128 ggr förstärkning valdes eftersom lastcellen ger ut en så liten signal. Lastcellen kopplades så att utsignalen går till förstärkaren och sladdar för excitations spänningen går till jord och 5 Volt från Arduinon. Signalen går sedan från förstärkaren till A/D-omvandlingen som sedan skickar värdet till det digitala gränssnittet som pratar med Arduinon via seriell kommunikation. ADCns pinnar för gränssnittet kopplas till vilka två Arduino input/output som helst. Den ena ingången fån Arduion läser signalen från ADCn och den andra styr klockan. Se figur 10 nedan.



Figur 10: Seriell kommunikation

Med den ingång som styr klockan kan man ställa in förstärkning och skifta ut data från signal ingången beroende på antal pulser; pulserna ska ligga mellan 25–27 st, varje positiv puls skiftar ut en bit från utsignalen. Den 25e pulsen sätter utsignalen till hög och sätter därmed utsignalpinnen till hög vilket säger att ADCn inte är redo för att ge nya värden. De antalet pulser utöver de 25 bestämmer förstärkningen på ADCn.

3.10.3 Klockkrets

Klockkretsen kommunicerar med Arduinon genom gränssnittet I^2C. Rätt datum sätts och den konfigureras så att ett alarm går mitt i natten och mitt på dagen genom programmering på Arduinon. Alarmet triggar en puls till Arduinon som är i sovande läge. Det gör att Arduinon gör en interrupt och vaknar ur strömsparläget.

3.10.4 SD-korts modul

SD-korts modulen kopplades in till Arduinon via gränssnittet SPI. Datan som skrivs till SD-kortet struktureras i kronologisk ordning i tabellform. Det ger en tydlig översikt över mätdatan. På SD-kortet lagras datum, tid och vikt vid varje mätning.

3.10.5 Strömförsörjning

Som strömförsörjning används 4 st 1.5V batterier. På dessa görs strömförbrukningsmätningar när de driver Arduinon. Sedan görs beräkningar av batteritid utifrån mätresultaten. Varje batteri har 2700 mAh och dessa batterier parallellkopplas vilket ger 4 gånger 2700, alltså, 10800 mAh. Arduinon drar enligt egna mätningar 36,3 mA vid strömsparläget och 60mA vid vanligt läge.

3.11 Tester

I detta avsnit beskrivs hur de olika delarna i projektet testades.

3.11.1 Test av A/D-omvandling

Elektroniken kopplas upp på en prototyp bräda och testas med hjälp av oscilloskop och sedan med programmeringskod med Mikrokontrollern. SD-kortet, AD-omvandlaren och klockkretsen testas var för sig och efter lyckat test med hjälp av programmeringstester och oscilloskop så sätts elektroniken ihop så att det fungerar tillsammans. Koden som används till var och en av de individuella elektronikmoudlerna sätts ihop och anpassas in det slutgiltiga programmet.

3.11.2 Test av konstruktion

En 3D simulering visade genom färger var belastningar och spänningar skulle ske i konstruktionen samt hur starka. Styrkan på spänningar och belastningar illustrerades med en färgskala. Där röd var starkast visar att konstruktionen böjer sig eller går sönder. I simuleringen visades bara blå och grön färg vilket indikerar att vågen inte kommer gå sönder under belastning. Test gjordes för att visa att det simuleringen stämmer. En 100 kg vikt ställdes på vågkonstruktionen.

3.11.3 Test av lastcell

Lastcellen testas genom att den belastas med olika vikter över en längre tid. Värdena dokumenteras för att se om och hur mycket lastcellen har ändrats. Konstruktionen testas genom att max belasta den med 100 kg och dokumentera hur den beter sig med vikten över tid.

3.11.4 Test av strömmförsörjning

Strömförbrukningen testas med hjälp av multimeter för att mäta hur mycket ström som dras från batterierna. Test görs när Arduinon är i viloläge samt vanligt läge. Eftersom vågen kommer presenteras som en demonstrator så har inte strömförsörjningen haft högst prioritet så man ansåg att det andra funktionerna var viktigare för att visa själva tanken.

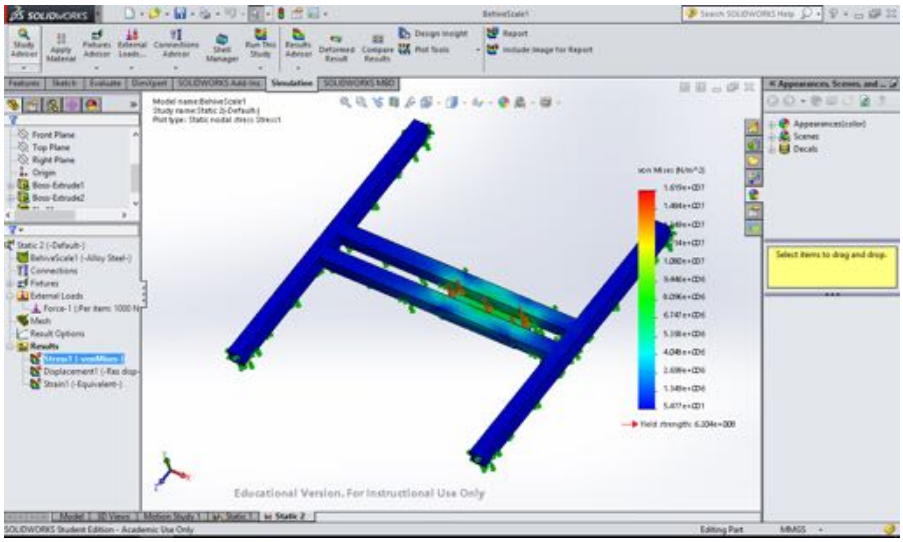
3.11.5 Systemtest

Hela systemet kommer att testas genom att låta vågen mäta en vikt som ökar gradvis under ett dygn. Det görs genom att låta en behållare droppa vatten ner i en hink som står i bikupan. Vågen mäter då dess vikt varje timme.

4 Resultat

4.1 Konstruktion och Lastcell

Lastcellen är stadig och ger bra stabila värden till A/D-omvandlaren. Den är enkel att fästa i konstruktionen med eftersom den redan har gängade hål. Konstruktionen som består av 2 st delar där varje del består 4 st ihopsvetsade fyrkantsrör. Fyrkantsrören har måtten 25x25 mm med tjocklek på 1.5 mm. De två mitten rören är svetsade med en plåt mellan sig som är 2 mm tjock och klarar av vikten på 100 kg som också var kravet för konstruktionen. Se figuren 11 nedan.



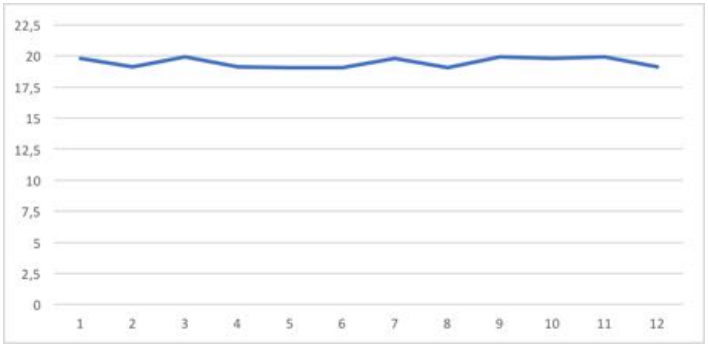
Figur 11: Kraft simulering i Solidworks visar vågens underdel och hur belastningen påverkar konstruktionen, vilket illustreras med hjälp av en färgskala. Röd färg visar att det är stora krafter och grön visar små krafter.

Konstruktionen blev lite sned p.g.a. krypning efter svetsen men fungerar bra. Den klarar att bära 100 kg utan att gå sönder, deformeras eller böjas märkbart.

4.2 A/D-omvandling

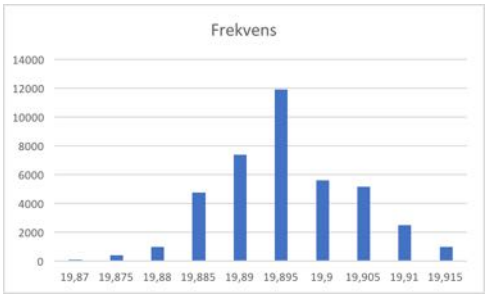
A/D-omvandling med 24 bitar gav en noggrannhet omkring 10 gram när Arduinon är igång hela tiden och med konstant vikt. När strömsparläget används och Arduinon går in och ur strömsparläget så blev dock värdena mindre noggranna, ungefär 10-20 gram. Vid test under några timmar så stabiliseras mätningarna.

Se mätvärden med en konstant vikt som är tagna varje minut i figuren 12 nedan. Att använda flera mätningar och medelvärdes beräkningar stabiliserar värdena bra.



Figur 12: Medelvärden av mätvärden med konstant vikt

Histogrammet i figur 13 visar bruset från lastcellen som varierar kring en vikt på 19,8 kg. Det visar att vid flera mätningar är det stor chans att de största delen av värdena hamnar inom ett 10 grams område omkring medelvärdet. Den vertikala axeln visar vikten och den horisontella axeln visar antal mätningar.

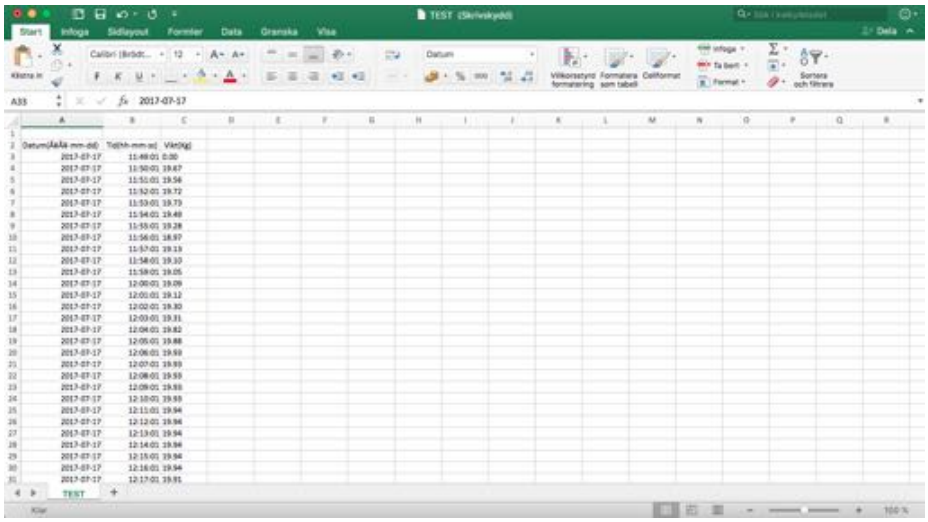


Figur 13: Histogram som visar en stor mängd mätvärdens fördelning med en konstant vikt på 19.8 kg

I histogrammet ser man att de allra flesta värdena hamnar nära kring medelvärdet. Histogrammet ser normalfördelat ut vilket gör att man kan motivera användning av medelvärde när man vill ha stabila mätresultat.

4.3 SD-kort

SD-kortet lagrar värden i ett enkelt textdokument som kan öppnas och analyseras i t.ex. Excel. SPI används för kommunikation mellan SD-modul och Arduino och fungerar bra. Värdena läggs in på ett ordnat sätt så att när de öppnas i Excel så visas datum, tid och vikt på alla mätningar. Se figur 14 nedan.



Figur 14: Mätvärden i excel som visar datum, tid och vikt

4.4 Realtidsklocka

Realtidsklockan fungerar bra den räknar tiden exakt och ger korrekta tidpunkter på viktmätningarna. Den kommunicerar med arduinon via I^2C och skickar och tar emot värden utan några fel. Vid tillfällen då arduinon är i strömsparläget så skickar klockan en puls vid specifika klockslag väcker Arduinon. Arduinon gör då nya mätningar. Noggrannheten på tiden är väldigt hög och man har ej kunnat mäta någon onoggrannhet i tid.

4.5 Strömförbrukning

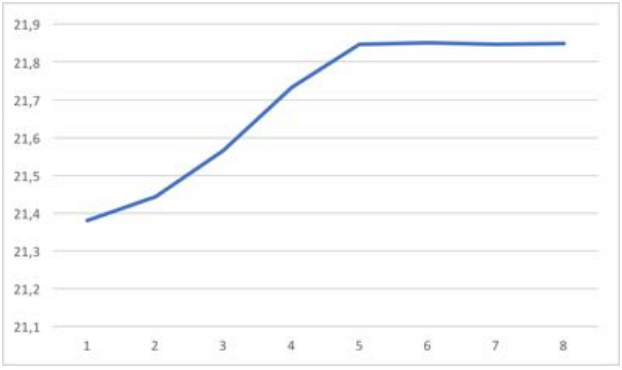
Strömförbrukningen i kretsen i vanligt läge ger 60mA. Strömsparningsläget ger 36.3 mA. Det ger en batteritid på 9 dagar vid användning av 4st parallellkopplade AA batteri. Se figur 15 nedan.

Figur 15: Strömförbruktnings mätning med multimeter när Arduinon är i viloläge

Arduinon väcks utav en puls från den externa klockan. Mätningar görs och arduinon somnar igen. Pulsen skickas var 4 timme så att 3 mätningar görs under dagen och 3 under natten.

4.6 Systemtest

Test gjordes för hela systemet när en vattenfylld 50 centiliters PET-flaska droppade vatten i en hink. Hinken stod på vågen under mer än ett dygn. Detta skulle simulera en viktökning liknade den som skulle ske när bin producerar honung. Mätningar har tagits var 4 timma. I figuren 16 nedan visas mätvärden vid varje mätning.



Figur 16: Mätvärden under systemtestet med långsamt ökande vikt

I figuren visar den horisontella axeln antal mätningar och den vertikala axeln visar vikten i kg.

5 Slutsats/Diskussion

5.1 Om konstruktion

Konstruktionen svetsades med pinsvets. Men det resulterade i att konstruktionen blev något sned p.g.a. den stora värmeutvecklingen i materialet samt att svets pinnen var lite för stor. Om man skulle göra än bättre svets så skulle man använt en TIG svets som gör det enklare att kontrollera värmen och att göra en precisions svets. Alternativt kunde vi gjort en typ av ställning som skulle motverka den krypning som sker i materialet p.g.a. värme. Det hade också tillåtit en noggrannare passform för att fästa lastcellen. För att fästa lastcellen användes torx fattning på skruvarna. Torx används för att få ett bättre grepp när man skruvar och de minimerar slitage på bulthuvudet vilket medför att det tillåter hårdare åt-dragning. Konstruktionen klarar av att bära och mäta 100 kg utan att gå sönder eller deformeras vilket möter kraven.

5.2 Om mätvärden

Mätvärdena är stabila med undantag för första och sista värdet som ibland kan vara ett några hekto fel. Med en konstant vikt så ändras värdena väldigt lite, plus, minus 10-20 gram första timmen men stabiliseras sedan. I de stora hela är vikten konstant och ytterst liten variation i mätningarna. Det gjorde stor skillnad i att mäta vikt flera gånger och beräkna medelvikten. Mätvärdena blev mycket mer stabila.

5.3 Om elektroniken

Elektroniken kopplas i en prototyp-bräda. Varje modul provas separat både med oscilloskop och med hjälp av programmeringskod. Det fungerar bra men om man kunde gjort allt på ett och samma kretskort för att få de mer kompakt och enkelt men det får bli nästa steg då själva funktionen är det viktigaste för denna prototyp.

5.3.1 Klockkretsen

Klockkretsen fungerar väldigt bra och är väldigt exakt. Den drivs av ett eget batteri när Arduinon inte är på. Den tar tiden både när Arduinon är på och av, och styr externa interruppt med en negativ puls. Man skulle egentligen kunna använda den så att man startar en avstängd Arduinon också och bara mäter några sekunder. Men då behövs lite extra elektronik och det blir lite dyrare. Det skulle då egentligen spara avsevärt mycket mer energi och batteriet skulle hålla mycket längre.

5.3.2 SD-kort

SD-kortet fungerar bra och lagrar tid och mätvärden så att man enkelt kan få en bra överblick i ett text dokument eller Excel. nästa steg är att göra så att man

kan rita upp en kurva. Och vid fortsättning av projektet är tanken att använda någon trådlös kommunikation. Men SD-kortet visar ett enkelt sätt att lagra på informationen så att de snabbt kan läsas av en dator.

5.3.3 Arduino

Arduinon fungerar bra till att användas för projektet. Vid fortsatta utveckling kanske man använde en mindre version men den har fungerat bra och har en del bra bibliotek och ganska mycket dokumentation. Man skulle kanske önska en lägre energiförbrukning men tanken är ju att man senare skall kunna ansluta en solcell och använder ett större batteri vilket ökar tiden den kan mäta avsevärt.

5.3.4 Den färdiga vågen

Vågen har en sådan noggrannhet på omkring 10 gram som tillåter den att göra meningsfulla mätningar varje dag. Lastcellen hade en bra linearitet och bruset visade att mätvärdena låg nära ett medelvärde vilket visar att medelvärdes beräkningar kan vara effektiva. Vågen kan göra mätningar under förbestämda tider och spara energi när den inte mäter vilket är väldigt bra för att batteritiden. Det är en demostrator men den visar att konceptet fungerar.

Om mer tid hade getts så hade man kunnat göra mer realistiska tester på riktiga biodlingar. Det finns dock mycket utvecklings möjligheter som att göra vågen ännu smartare och registrera väder och stötar på ett bättre sätt. Detta är något som kan göras under vidareutvekclingen.

5.3.5 Sammanfattning

Projektets möter alla krav som finns i kravspecefikationen. Vågen var relativt billig att bygga då pristet stannar på 1112 kr. Tabell 2 nedan visar noggrannare vad de olika delarna av projektet kostade. För binas skull så används vågen på

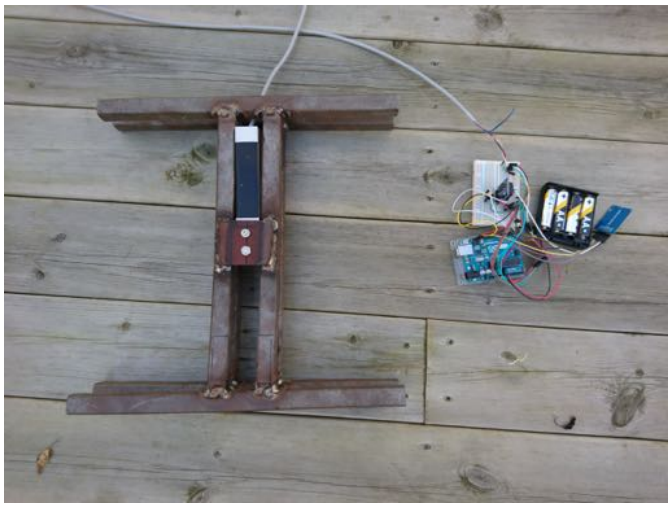
Material och komponenter	Pris
Arduino Uno	230 Kr
Belastningsmätare, AL6N-C3-100kg-3B6, Variohm EuroSensor	600 Kr
RTC-kretsen DS3231 från Luxorparts	80 Kr
Luxorparts Micro-SD-kortläsare för Arduino	90 Kr
SparkFun Load Cell Amplifier - HX711	112 Kr
Fyrkantsrör	240 kr
Bult	80 Kr
Totalt	1432 Kr

Tabell 2: Tabell för projektets kostnad

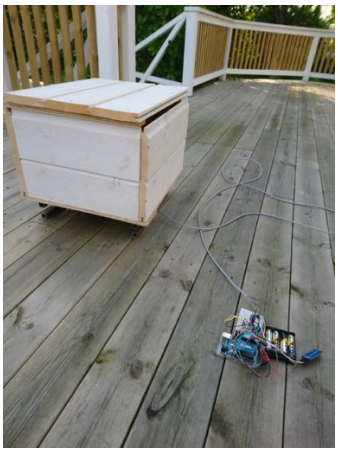
utsidan bikupan och inte inuti. Det är för att elektroniken inte skall påverka eller störa bina på något sätt. I bikupor vill man helst bara ha naturliga material. Konstruktionen är gjord av stål som inte ger någon direkt miljöpåverkan,

Stål går också att återvinna. Elektroniken kapslas in för att inte påverkas eller påverka den yttre miljön.

Vågen försöker spara energi så mycket den kan och förväntas användas med laddningsbara batteri för att minska miljöpåveran av att slänga engångsbatterier. Vid vidaruveckling är tanken att solcell skall användas vilket drar ner energin extra mycket. För att vågen skall var säker att använda så fasades alla vassa kanter och hål av med slipmaskin. Projektet tog längre tid än väntat men alla krav uppnåddes. Nedan visas bild på systemet och systemet med bikupa, se figur 17 och figur 18.



Figur 17: Bild på systemet



Figur 18: Bild på systemet med bikupa. Tanken är att elektroniken sedan skall ligga i en låda som sitter intill ramen till lastcellen.

Referenser

[1] Dennis van Engelsdorp and Marina Doris Meixner.(2010). A historical review of managed honey bee populations in Europe and the United States and the factors that may affect them. *Journal of Invertebrate Pathology* volume 104, S80 - S95

[2] Peter Neumann and Norman L Carreck, (2010).Honey bee colony losses. *Journal of Apicultural Research*, volume 49, p.1-6.

[3] `Våg (instrument)` *https://sv.wikipedia.org/wiki/*

[4] *WHAT IS A DYNAMOMETER AND HOW DOES IT WORK? http://www.setra.com/blog/test-and-measurement-dynamometer*

[5] *Biodling* `http://www.ullalars.se/biodling.htm`

[6] *Sampling Theory For Digital Audio By Dan Lavry, Lavry Engineering, Inc. https://web.archive.org/web/20060614125302/ http://www.lavryengineering.com/documents/Sampling$_T$heory.pdf*

[7] *Understanding Flash ADCs* `https://www.maximintegrated.com/en/app-notes/index.mvp/id/810`

[8] *Understanding SAR ADCs: Their Architecture and Comparison with Other ADCs*

`https://www.maximintegrated.com/en/app-notes/index.mvp/id/1080`

[9] *How delta-sigma ADCs work*

`http://www.ti.com/lit/an/slyt423a/slyt423a.pdf`

[10] *Wheatstone bridge.*

`http://www.electronics-tutorials.ws/blog/wheatstone-bridge.html`

[11] *Practical Uses of Instrumentation Amplifiers* https://www.allaboutcircuits.com/technical-articles/practical-uses-of-instrumentation-amplifiers/

[12] *Introduction to Load Cells*

`http://www.omega.com/prodinfo/loadcells.html`

[13] *Sensorer och elektronik*

`http://www8.tfe.umu.se/courses/elektro/FSE/Kraft/kraft.pdf`

[14] *ARDUINO UNO REV3* `https://www.arduino.cc/en/Main/ArduinoBoardUno`

[15] *Focus Product Selector Guide*

`http://ww1.microchip.com/downloads/en/DeviceDoc/00001308R.pdf`

[16] *What is an RTC?*

`https://learn.adafruit.com/ds1307-real-time-clock-breakout-board-kit/what-is-an-rtc`

[17] *Electronic Devices (Conventional Current Version),9th Edition* Thomas L. Floyd, 2012

[18] D. W. Fitzgerald and F. E. Murphy and W. M. D. Wright and P. M. Whelan and E. M. Popovici, (2015). Design and development of a smart weighing scale for beehive monitoring, *26th Irish Signals and Systems Conference (ISSC)*, p.1-6.

[19] *Birds and mankind Destroying Nature by 'Electrosmog': Effects of Wireless Communication Technologies. A brochure Series by the Competence Initiative for the Protection of Humanity, Environment and Democracy. Kempten; (2009)*

[20] *System Architectures for Real-time Bee Colony Temperature Monitoring; Procedia Computer Science*

[21] Tomas Svensson,Christian Krysander. *Projektmodellen Lips*. Studentlitteratur (ISBN 978144075259)

[22] *Om SOLIDWORKS*

`http://www.solidworks.se/sw/6453_SVE_HTML.html`

A Bilaga

Programmeringskod

```
 #include <SD.h>
#include "HX711.h"
#include <Wire.h>
#include <avr/interrupt.h>
#include <avr/power.h>
#include <avr/sleep.h>
#include <Time.h>
#include <TimeLib.h>
#include <DS3232RTC.h>

#include <DS3232RTC.h>

#define DOUT  3
#define CLK  5
#define interruptPin 2

#define DS3231_ADDRESS 0x68
#define DS3231_SECONDS_REG 0x00

#define DS3231_CONTROL_REG 0x0E
#define DS3231_ALARMS_REG 0x07
double previkt = 0;
DS3232RTC arduino;
HX711 scale(DOUT, CLK);
File myFile;
const int READ_DELAY = 500;
volatile byte state = false;

typedef struct timepar{
  uint8_t ss;
  uint8_t mm;
  uint8_t hh;
  uint8_t dy;
  uint8_t d;
  uint8_t m;
  uint8_t y;
  }Timepar;

  Timepar intime ;
  Timepar outtime;
```

```
static uint8_t convertValueOUT2(uint8_t value)
{
  uint8_t convertedVal = value ;
  return convertedVal ;
}

uint8_t maskBit(int b){
  uint8_t tmp = 2;

  return pow(tmp, b);
}


void setTime(Timepar *timeVals){
  /*
   * From the datasheet
   * 0 - seconds
   * 1 - minutes
   * 2 - hours
   * 3 - day
   * 4 - date
   * 5 - month
   * 6 - year
   */

  Wire.beginTransmission(DS3231_ADDRESS) ;
  Wire.write(DS3231_SECONDS_REG) ;

  Wire.write(convertValueOUT(timeVals->ss)) ;
  Wire.write(convertValueOUT(timeVals->mm)) ;
  Wire.write(convertValueOUT(timeVals->hh)) ;
  Wire.write(0) ;
  Wire.write(convertValueOUT(timeVals->d)) ;
  Wire.write(convertValueOUT(timeVals->m)) ;
  Wire.write(convertValueOUT(timeVals->y)) ;
  Wire.endTransmission() ;
  delay(5) ;

}

void readTime(Timepar *timeVals)
{
  Wire.beginTransmission(DS3231_ADDRESS) ;
  Wire.write(DS3231_SECONDS_REG) ;
  Wire.endTransmission() ;
```

```
  Wire.requestFrom(DS3231_ADDRESS, (byte) sizeof(Timepar)) ;

  timeVals->ss = convertValueIN(Wire.read()) ;
  timeVals->mm = convertValueIN(Wire.read()) ;
  timeVals->hh = convertValueIN(Wire.read()) ;
  Wire.read() ;
  timeVals->d = convertValueIN(Wire.read()) ;
  timeVals->m = convertValueIN(Wire.read()) ;
  timeVals->y = convertValueIN(Wire.read()) ;

  delay(5) ;
}

static uint8_t convertValueIN(uint8_t value)
{
  uint8_t convertedVal = value - 6 * (value >> 4) ;
  return convertedVal ;
}

static uint8_t convertValueOUT(uint8_t value)
{
  uint8_t convertedVal = value + 6 * (value / 10) ;
  return convertedVal ;
}

void setup() {

  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  Wire.begin();

  pinMode (interruptPin, INPUT_PULLUP);
  intime = {
    0,
    49,
    11,
    1,
    28,
    04,
    17,
  } ;

setTime(&intime);


 if (!SD.begin(4)) {
```

```
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  scale.set_scale();
  scale.tare();

  Serial.println("Initializing SD card...");
myFile = SD.open("test.txt",FILE_WRITE);
myFile.println();
myFile.println("Datum(aa-mm-dd)\t Tid(hh-mm-ss)\t Vikt(Kg)");
  myFile.close();
}

void loop() {
delay(50);
   scale.set_scale(42000); //Adjust to this calibration factor
if(!state){Serial.println("alarm");
  arduino.alarm(2);}
  state = true;
delay(1000);
Serial.print("Reading: ");

 double vikt1 = scale.get_units(8);
delay(100);
 double vikt2 = scale.get_units(8);
delay(100);
 double vikt3 = scale.get_units(8);

double vikt = (vikt1+vikt2+vikt3)/3;
if( previkt == 0){previkt = vikt;}
int p =1;
while( p == 1){
  p = 0;
if(vikt < 0.0){
 vikt = 0.0;
  };
 myFile = SD.open("test.txt", FILE_WRITE);
 if (myFile) {

    readTime(&outtime);
    //Serial.println("Writing to test.txt...");
    //myFile.print("xx-xx-xx\t");
    myFile.print(outtime.d);
    myFile.print("/");
```

```
    myFile.print(outtime.m);
    myFile.print("-");
    myFile.print(outtime.y);
    myFile.print("\t");
    //
    myFile.print(outtime.hh);
    myFile.print(":");
    myFile.print(outtime.mm);
    myFile.print(":");
    myFile.print(outtime.ss);
    myFile.print("\t");
    myFile.print(vikt);
    myFile.println("");
    // close the file:
    myFile.close();
    //Serial.println("done.");
  } else {
    // if the file didn't open, print an error:
    Serial.println("error opening test.txt");
  }

  if(vikt-previkt > 1) {p = 1;}
  double previkt = vikt;
}
  scale.power_down();              // put the ADC in sleep mode
  //delay(READ_DELAY);
  sleepNow();
  scale.power_up();
  Serial.println("waking up!");
}

void interrupt(){
  sleep_disable();
  state = false;

  }
void sleepNow()         // here we put the arduino to sleep
{
    sleep_enable();
    attachInterrupt(digitalPinToInterrupt(interruptPin), interrupt, CHANGE);
    set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    cli();
    sleep_bod_disable();
    sei();
    sleep_cpu();
    /*Sleeps here*/
```

```
    sleep_disable();

}
void AlarmSet(){

  time_t current_time;
  current_time = now();

 arduino = DS3232RTC(arduino);

  //arduino.set(current_time);
  byte alarmNumber = 2;
  bool interruptEnabled = true;
  arduino.setAlarm(ALM2_EVERY_MINUTE,0,0,0);
  arduino.alarmInterrupt(alarmNumber, interruptEnabled);
  byte control = arduino.readRTC(RTC_CONTROL);
  Serial.println(control);
  arduino.writeRTC(RTC_CONTROL,(control | 0x04));

  control = arduino.readRTC(RTC_CONTROL);
  Serial.println(control);
  arduino.alarm(2);
   attachInterrupt(digitalPinToInterrupt(interruptPin), interrupt, CHANGE);
  }
```

Komponenter

A/D-omvandlare och förstärkar krets

SparkFun Load Cell Amplifier - HX711

Klockkrets

RTC-kretsen DS3231 från Luxorparts

SD-kortsmodul

Luxorparts Micro-SD-kortläsare för Arduino

Mikrokontroller

Arduino uno rev3

Lastcell

Belastningsmätare 100 kg, AL6N-C3-100kg-3B6, Variohm EuroSensor

Konstruktionens mått

Höjd

6,5 cm

längd

35 cm

Bredd

35 cm

Fyrkantsrörens dimensioner

24*24 mm

Viktor Ringmark

Besöksadress: Kristian IV:s väg 3
Postadress: Box 823, 301 18 Halmstad
Telefon: 035-16 71 00
E-mail: registrator@hh.se
www.hh.se