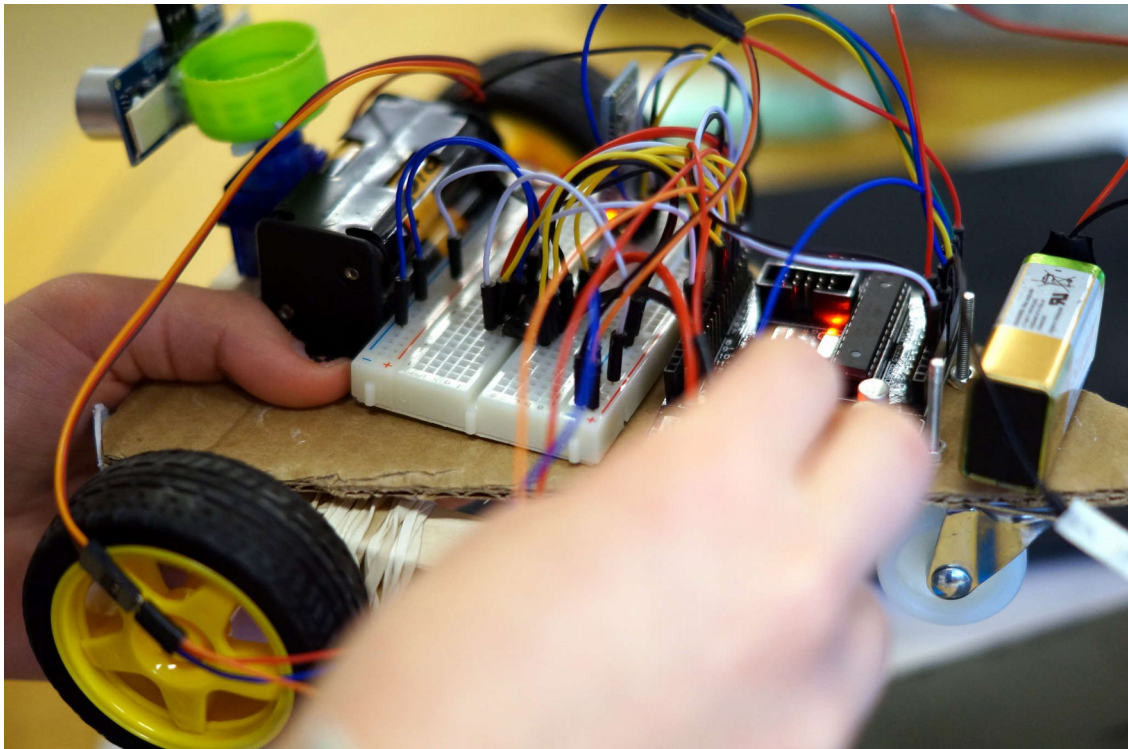




Teknik årskurs 9 - Robotprogrammering

Robotprogrammering

Att bygga och programmera en motordriven robot som kan undvika hinder.



Av Staffan Melin (lärarpraktikant) och Martin Blom (NO-lärare)
Bild & form-skolan, Masthugget, Göteborg
2015



Detta verk är licensierat under en
Creative Commons Erkännande-IckeKommersiell-DelaLika 4.0 Internationell Licens.



Innehållsförteckning

Innehåll

Robotprogrammering.....	1
Innehållsförteckning.....	2
Om projektet.....	4
LPP.....	4
Form.....	4
Tidsåtgång för eleverna.....	4
Dokumentation.....	4
Delar vi använder (komponenter).....	5
Schema.....	6
Förberedelser.....	9
Beställ delar.....	9
Datorer för programmering.....	9
Inför lektion 1.....	10
Inför lektion 2.....	10
Inför lektion 3.....	10
Inför lektion 4.....	10
Verktyg och material.....	10
Lektion 1. Introduktion till Arduino.....	12
Vad är Arduino?.....	12
Vad är en robot?.....	12
Vad är ett program?.....	12
Lektion 2: Servo och sensor.....	15
Komponenter.....	15



Teknik årskurs 9 - Robotprogrammering

Del 1. Servo.....	16
Del 2. Sensor.....	17
Del 3. Servo + sensor.....	19
Läs mer/referens.....	19
Lektion 3: En motor och H-brygga.....	20
Så här styr Arduinon H-bryggan.....	21
Koppla ihop allt.....	22
Program.....	23
Läs mer/referens.....	25
Lektion 4: Bygg roboten.....	26
Schema.....	26
Ritning.....	27
Exempel på färdigt program.....	28
LPP.....	32



Om projektet

Hösten och vintern 2014-2015 konstruerade Bild & form-skolan i Göteborg en robot i årskurs 9. Målet var att eleverna skulle lära sig bygga med digital elektronik samt grundläggande programmering. Detta genomförde vi med Arduino, en så kallad microcontroller (enkel dator) som är billig och duktig på att prata och läsa av omvärlden. Det finns en uppsjö av information, inspiration och projekt på internet, och den bygger dessutom på öppen källkod.

Det hela tog formen av ett teknikutvecklingsprojekt där komplexiteten gradvis ökades. En viktig motivation till projektet var att eleverna skulle nå en insikt om hur digital teknik fungerar, vilka olika delar som ingår och hur de samverkar.

Kostnaden per robot var ca 300 kr.

Vi har i denna skrift försökt att samla vår planering och de erfarenheter vi gjorde under projektet. Med hjälp av detta hoppas vi att fler skolor kan genomföra liknande projekt. Kontakta oss gärna om du har några frågor! Och hör gärna av dig om du gör något liknande!

Staffan Melin, staffan.melin@oscillator.se
Martin Blom, martinblomblom@hotmail.com

LPP

Se slutet av denna manual.

Form

Vi genomförde projektet under 7 lektioner à 2 timmar. Eleverna arbetade i grupp med 3-4 elever i varje grupp. Större delen av projektet arbetade vi i helklass, med två lärare.

Tidsåtgång för eleverna

Lektion 1: 2 timmar.

Lektion 2: 3-4 timmar.

Lektion 3: 2 timmar.

Lektion 4: 6 timmar.

Dokumentation

Eleverna fick hela tiden dokumentera projektet, både skriftligt och individuellt, samt i grupp genom att med sina surfplattor filma och berätta om dagens arbete. De fick ca 15 minuter varje vecka för dokumentationen.

Vi strävade efter att skriva upp "dagens begrepp" på tavlan - dessa skulle finnas med i dokumentationen.

Dokumentationen skulle innehålla dagens mål, vad de genomförs, problemen de stötte på och hur de löste dem (eller planerade att lösa dem).



Komponenter

Arduino. En enkel programmerbar experimentdator som är bra på att kommunicera med omvärlden. Robotens "hjärna".

Batteri-hållare. Eftersom roboten ska kunna åka omkring måste den vara batteridrivna. 4 st så kallade AA-batterier på 1,5 V styck (även kallade LR6) driver motorerna.

Batteri-kontakt: 1 st 9V-batteri driver Arduinon (datorn). När Arduinon är ansluten till datorn för programmering, så får den ström via USB-kabeln som förbinder de två. Då behövs alltså inte 9V-batteriet.

Motorer med hjul och växellåda. Roboten är trehjulig, och svänger genom att hjulet på ena sidan snurrar medan det på andra sidan står still (eller rör sig i motsatt riktning). Det tredje hjulet är ett stödhjul. Robotens "fötter".

H-brygga/L293D. Detta är en liten IC-krets (http://sv.wikipedia.org/wiki/Integrerad_krets) som används för att Arduinon ska kunna styra motorerna. (Kallas också för H-brygga, läs mer om du vill: <http://sv.wikipedia.org/wiki/H-brygga>).

Sensor (avståndsmätare). Avståndsmätare använder ultraljud (ljud som har en högre frekvens än vad det mänskliga örat kan uppfatta) för att mäta avstånd. Robotens "ögon". Fladdermössen använder samma teknik.

Servo. Ett servo innehåller en liten motor som kan vridas, och gör att vi kan rikta avståndsmätaren åt olika håll. Robotens "hals".

Kondensator. Kondensatorn kan lagra ström (som ett batteri, fast ofta under kortare stunder), och används i vårt projekt för att jämna till strömmen över servot (eftersom servot drar lite extra när det sätter igång).

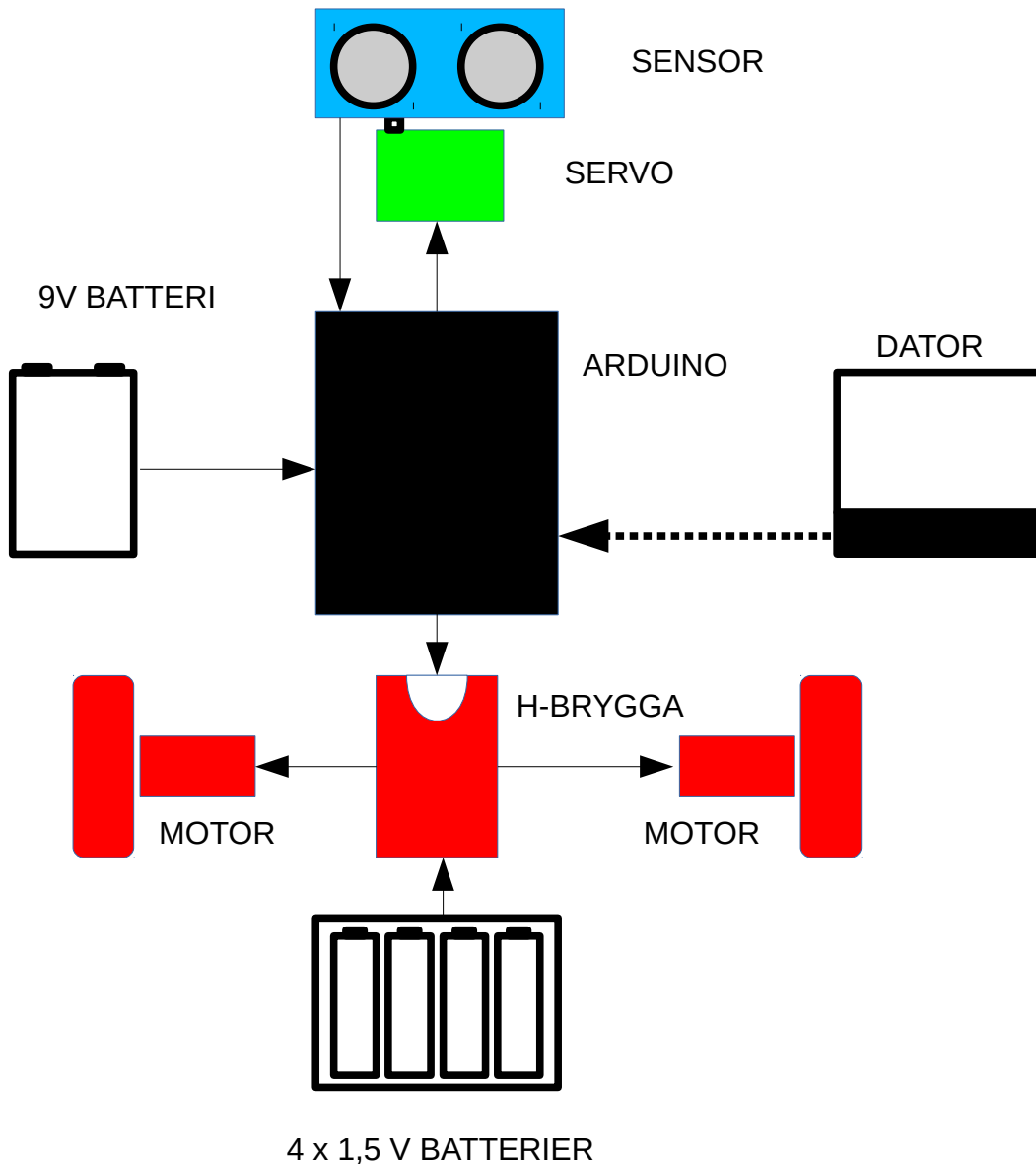
Dator. Programmen skrivs på en dator för att sedan skickas över till Arduinon via en USB-kabel. På datorn har vi installerat en så kallad "utvecklingsmiljö", som är ett program där du kan skriva och ändra Arduino-program och med ett enkelt knapptryck skicka över dem till Arduinon via en USB-kabel.

Breadboard. Används för att koppla ihop komponenter utan att behöva löda.



Schema

Så här sitter delarna ihop:





Teknik årskurs 9 - Robotprogrammering

De olika komponenterna kopplas ihop med sladdar, ofta med hjälp av en **breadboard**. Det finns i huvudsak två olika sorters förbindelser hos varje komponent:

- de som skickar den **ström** som komponenterna behöver för att fungera (strömkrets)
- de som skickar **data** mellan komponenterna och Arduinon

Roboten har dessutom två olika **strömkretsar**:

- En är ström för motorerna (6V). Den skickas från batterihållaren (4x1,5 V = 6V) via H-bryggan.
- En är ström från Arduinon (5V). Den kommer först från datorn via USB-kabeln, senare från det 9V-batteri som ansluts till Arduinon.

Arduinon skickar i sin tur vidare **ström** till:

- sensorn (avståndsmätaren)
- servot
- H-bryggan (som styr motorerna)

Arduinon **skickar data** till

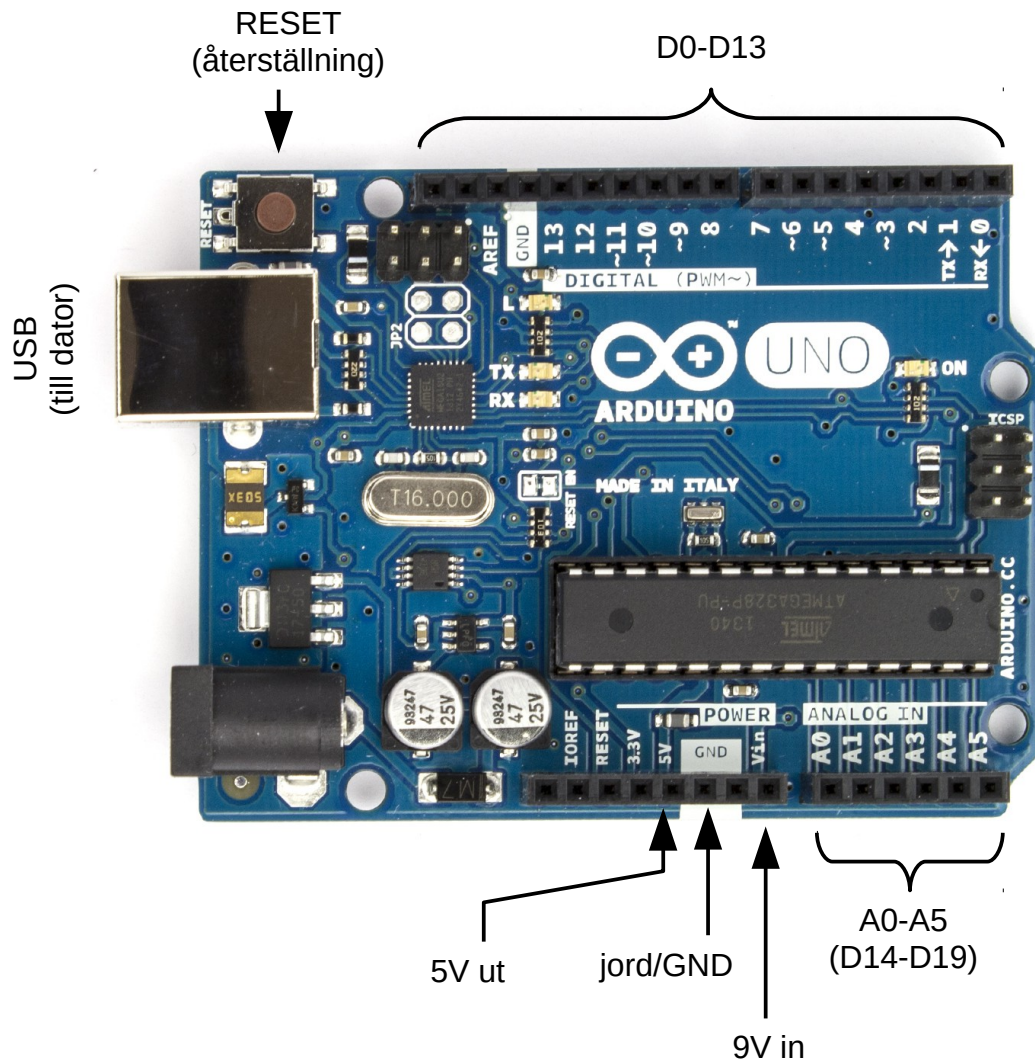
- H-bryggan, för att reglera hur mycket ström som går till motorerna
- servot, för att styra riktningen på avståndsmätaren
- sensorn, för att tala om för den att den ska mäta avståndet

Arduinon **tar emot data** från

- avståndsmätaren, som skickar tillbaka avståndet till närmaste hinder rakt framåt



För att kunna prata med omvärlden har Arduino flera **stift eller portar**, dit du kan koppla komponenter (och sladdar):



- D0-D13: digitala stift/portar. De här kan ta emot och skicka två olika signaler: LOW (av, 0V) eller HIGH (på, +5V).
- A0-A5: analoga stift/portar. De kan ta emot värden från 0-1024 (normalt 0-5V). De kan också användas som digitala portar (och kallas då D14-D19).

Stiften kan ta emot eller skicka information - du talar om vilket i början av ditt program.

Det gäller med andra ord att dina kopplingar stämmer överens med hur du läser och skickar information i ditt program. Sladdar och komponenter samarbetar med programmet!



Förberedelser

Beställ delar

Beställ material i god tid om du beställer dem från utlandet. Vi beställde allt material förutom batterier från Ebay. Det tog upp till 2 månader att få hem allt. Beställ ett par reservsatser, det är alltid något som krånglar eller går sönder.

För varje del har vi skrivit in lämplig sökterm på Ebay.

- Robot-kit: platta, två hjul med växellåda och motor, batterihållare för 4xAA, tredje stödhjul. (Ebay: "arduino robot car chassis", ca 80 SEK.)
- Arduino Uno, inklusive USB-sladd för programmering. (Ebay: "arduino uno R3 usb cable", ca 50 SEK.)
- H-bridge: L293D (liten IC-krets). (Ebay: "H-bridge L293D", ca 20 SEK.)
- ultraljuds avståndsmätare SR03/SR04. (Ebay: "arduino sr04", ca 10 SEK.)
- (micro) servo motor. (Ebay: "arduino micro servo sg90", ca 20 SEK. Finns även i 10-pack.)
- 9V batterihållare for Arduino. (Ebay: "9v battery clip cable", ca 10 SEK. Finns i två varianter. En som ansluts till strömförsörjningsporten på Arduinon, samt en vars kabel slutar i sladdar. Vi har använt den senare.)
- 100 μ F (elektrolytisk) kondensator. (Ebay: "100 μ F electrolytic capacitor", ca 8 SEK.)
- breadboard (400 punkters). (Ebay: "arduino breadboard 400 contacts", ca 20 SEK.)
- sladdar:
 - hane-hona stift (till sensor): 4 st, minst 20 cm långa. (Ebay: "arduino male female cable", ca 10 SEK för 40 st.)
 - hane-hane stift: ca 30 st (15 st x 15 cm, 15 st x 10 cm). (Ebay: "arduino male male cable 65pcs", ca 15 SEK för 65 stycken.)
- batterier:
 - 4x1,5V AA till varje robot (10 grupper = 40 batterier)
 - 1x9V till varje robot (10 grupper = 10 batterier)

Datorer för programmering

Programmen för Arduinon skrivs på en "vanlig" dator med hjälp av programmet **Arduino IDE**. Det kan laddas ner från Arduinons hemsida <http://arduino.cc/en/Main/Software>. Det finns till både Linux, Windows och Mac OSX.



Du behöver också installera biblioteket **NewPing** (vilket förklaras i lektion 2). Du laddar ner det från playground.arduino.cc/Code/NewPing, packar upp det till en mapp, och placerar det i följande mapp: Arduino-1.0.6 > libraries (mappen "Arduino-1.0.6" kan ha olika namn beroende på vilken version du kör). Alla bibliotek ska installeras *innan* du startar Arduino IDE.

Här kan du läsa mer om hur du installerar bibliotek: <http://arduino.cc/en/pmwiki.php?n=Guide/Libraries>

Inför lektion 1

Som förberedelse är det en mycket god idé att bygga en egen robot. Börja lektion 1 med att visa upp den och demonstrera funktionen. Denna robot kan senare fungera som referens för eleverna om de undrar över något.

Låt grupperna börja fundera över hur de vill sätta ihop sin robot.

Inför lektion 2

Inga särskilda förberedelser.

Inför lektion 3

Löd fast sladdar med stift på batteri-kontakterna till 9V-batteriet. Gör gärna även samma sak på de två sladdar som sitter fast i varje motor.

Inför lektion 4

Inför denna lektion gick vi igenom alla robotar och såg till att de var korrekt ihopsatta (alla sladdar på rätt plats etc). Vi skickade över ett program som testade allt.

Ha alltid extra batterier (9V och 4xAA).

Beställ extra motor/växellådor eftersom de verkar vara av låg kvalitet: stiften på motorn lossnar och (plast)kugghjulen i växellådan slits.

Börja lektionen med att gå igenom exempelkoden. Färgerna i koden överensstämmer med färgerna på komponenterna i schemat, så det är bra om du kan visa de båda på samma gång.

Vi avslutade projektet med att bygga en bana med hinder i klassrummet. De olika projektgruppernas robotar fick därefter tävla mot varandra: Vilken robot klarade sig längst utan att krocka?

Verktyg och material

Vi använde oss av följande verktyg och material:

- Kniv
- Sax
- Kniptång



Teknik årskurs 9 - Robotprogrammering

- Ståltråd
- Tejp
- Limpistol
- Lödstation/lödkolv. Vi lät inte eleverna löda något själv utan gjorde det åt dem. Tack vare breadboarden behövs väldigt lite lödning.
- Plattor, burkar från köket, glasspinnar och liknande byggmaterial. Foamboard skulle också kunna vara ett bra byggmaterial.
- Märkpenor och frystejp för att märka sladdar.
- Eltejp
- Gummisnoddar och snöre för ihopsättning.

Vi provade även häftmassa, men den leder ström så den övergav vi.

Även om det är relativt billiga delar, blir de tillsammans dyra. Våra elever skulle därför bygga roboten så att den var återvinningsbar = kunde plockas isär och delarna återvinnas.



Lektion 1. Introduktion till Arduino

Det här ska vi göra! En robot som kan rulla omkring utan att krocka!

Se ett exempel: <http://youtu.be/rtzUJKoj7Vc>

Vad är Arduino?

En enkel dator för experiment. Den är billig och bra på att prata med omvärlden (den har många in- och utgångar = kan kopplas ihop med annan elektronik).

- Läs mer om du vill: <http://www.electrokit.com/arduino.html>
- Om du vill ha ännu fler (tekniska) detaljer: <http://sv.wikipedia.org/wiki/Arduino>
- Arduinons hemsida: <http://arduino.cc/>

Vad är en robot?

"En robot är en teknisk anordning, oftast en elektromekanisk maskin som styrs av elektronisk programmering, som utför fysiska uppgifter." (Wikipedia)

Eller mer kortfattat: en maskin som styrs av en dator.

Läs mer om du vill: <http://sv.wikipedia.org/wiki/Robot>

Vad är ett program?

"En serie instruktioner som styr en dator " (Wikipedia)

Eller: Genom att skriva ett program talar du om för datorn vad den ska göra. Ett program skrivs ofta som en text i ett datorspråk. Det kan vara olika långt, från några rader till miljontals rader (typ JAS).

Läs mer om du vill <http://sv.wikipedia.org/wiki/Datorprogram>

Saker som nästan alla program har

- Funktioner. Några rader kod som kan användas flera gånger. Börjar med att de namnges.
- Variabler, konstanter. Tänk på dessa som små lådor där du kan skriva och läsa värden. Om det är en konstant börjar raden med "const". Då kan du inte ändra värdet senare i programmet.
- Satser: Rader i programmet som gör saker.

Arduinon kör igång sitt program (den kan endast hålla ett program i minnet åt gången) när den får ström. När strömmen avbryts avslutas programmet (men finns kvar i Arduinons minne). Du kan starta om programmet från början genom att trycka på Arduinons RESET-knapp.



Teknik årskurs 9 - Robotprogrammering

Arduino-program har två viktiga funktioner:

- **setup.** Den här delen av programmet körs en gång i början.
- **loop.** Den här delen körs gång på gång tills Arduinon inte längre får ström, eller ett nytt program laddas.

Här är ett enkelt program som får en ljusdiod att blinka. Arduinon har från början en lysdiod ansluten till port 13. Lysdioden sitter alltså redan monterad på Arduinon.

```
const int kLEDPin 13; // lysdioden sitter på stift 13

void setup () {
    pinMode(kLEDPin, OUTPUT);    // ställ in stift 13 för digital
    utmatning
}

void loop () {
    digitalWrite(kLEDPin, HIGH); // Slå på lysdioden
    delay(1000);                 // Vänta 1 sekund (1000 millisekunder)
    digitalWrite(kLEDPin, LOW);  // Slå av lysdioden
    delay(1000);                 // Vänta 1 sekund (1000 millisekunder)
}
```

Skriv in programmet på datorn och ladda upp det till Arduinon via USB-kabeln. Lysdioden ska börja blinka.

Så, vad händer på de olika raderna i programmet:

1. Vi skapar en variabel som kommer att vara konstant och innehålla värdet 13. Arduinon har en LED (lysdiod) ansluten till denna port.
2. Tom rad. Dessa struntar Arduinons i.
- 3-5. Setup-funktionen. Körs en gång när programmet startar. Talar om att Arduinons port med nummer 13 ska användas för att skicka information.
6. Tom rad.
- 7-12. Loop-funktionen. Körs om och om igen. Gör följande:
 8. Sätter port 13 till PÅ. Då går en ström genom lysdioden och den tänds.
 9. Väntar 1000 ms (= 1 sekund).
 10. Sätter port 13 till AV. Då går inte strömmen längre genom lysdioden och den slocknar.
 11. Väntar 1000 ms (= 1 sekund).

Prova att ändra värdena i delay-funktionen. Vad händer? Vad händer om du ändrar ordningen på raderna (satserna) i loop-funktionen (raderna 8-11)?



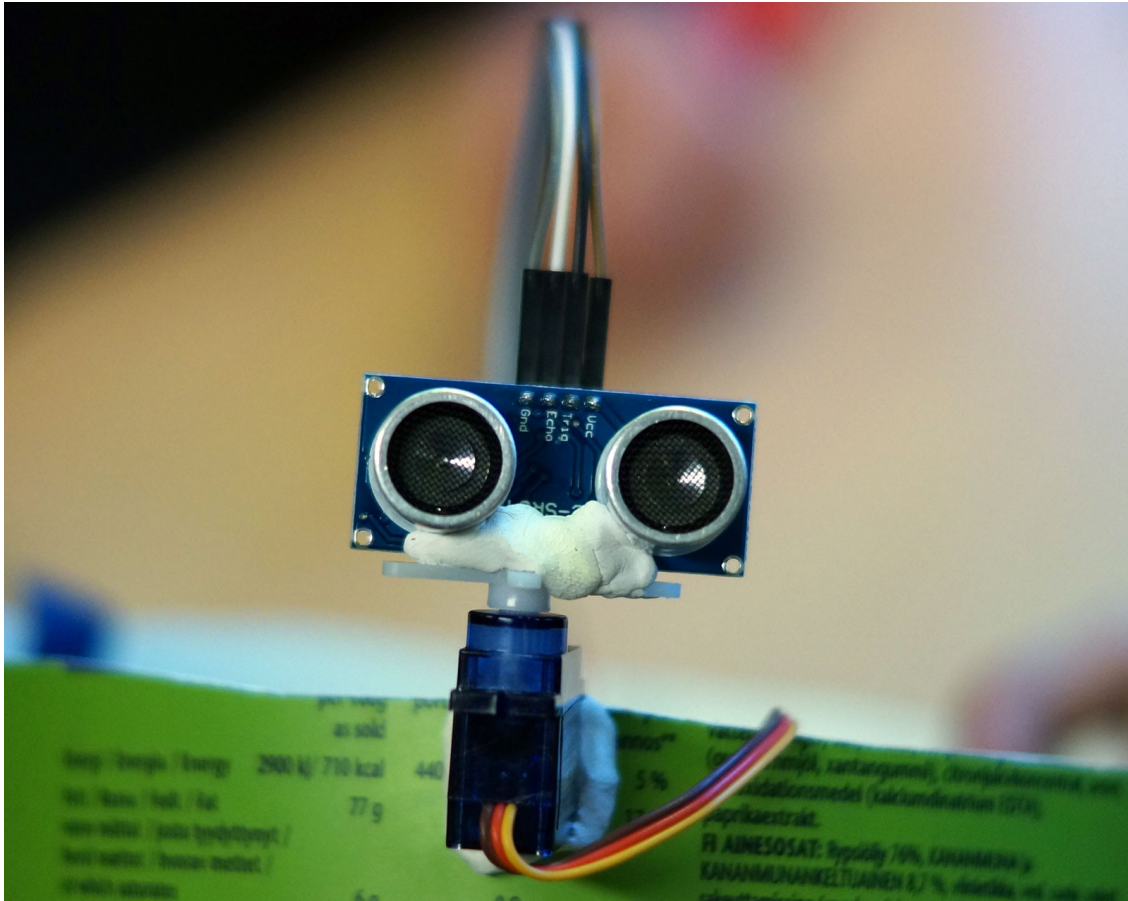
Teknik årskurs 9 - Robotprogrammering

Andra viktiga saker i ett program:

- **Måsvingar/block.** Håller samman flera satser. Se <http://www.arduino.cc/en/Reference/Braces>.
- **for.** Upprepar satser flera gånger. Se <http://arduino.cc/en/Reference/For>.
- **if.** Testar ett så kallat villkor, och kör olika satser beroende på vad testet säger. Se <http://arduino.cc/en/Reference/If> och <http://arduino.cc/en/Reference/Else>.
- **case.** Påminner om if, men låter dig testa flera värden. Se <http://arduino.cc/en/Reference/SwitchCase>.
- **Kommentarer.** Ger möjlighet att skriva in noteringar i programmet som Arduinon sedan struntar i. Se <http://arduino.cc/en/Reference/Comments>.
- **Funktion.** Flera rader med satser, som ges ett namn, och kan användas flera gånger. Se <http://arduino.cc/en/Reference/FunctionDeclaration>.
- **random().** En inbyggd funktion som tar fram ett slumpvärde. Se <http://arduino.cc/en/Reference/Random>.



Lektion 2: Servo och sensor



Denna lektion kommer vi att börja se hur **program och hårdvara jobbar tillsammans**. Det är så vår robot kan känna av och styra världen runt omkring.

Att bygga något som inte är permanent är en viktig del i att ta fram en **prototyp**. Det ger möjlighet att prova sig fram.

Komponenter

Kondensator. Kondensatorn kan lagra ström (som ett batteri, fast ofta under kortare stunder), och används i vårt projekt för att jämna till strömmen över servot (eftersom servot drar lite extra när det sätter igång).

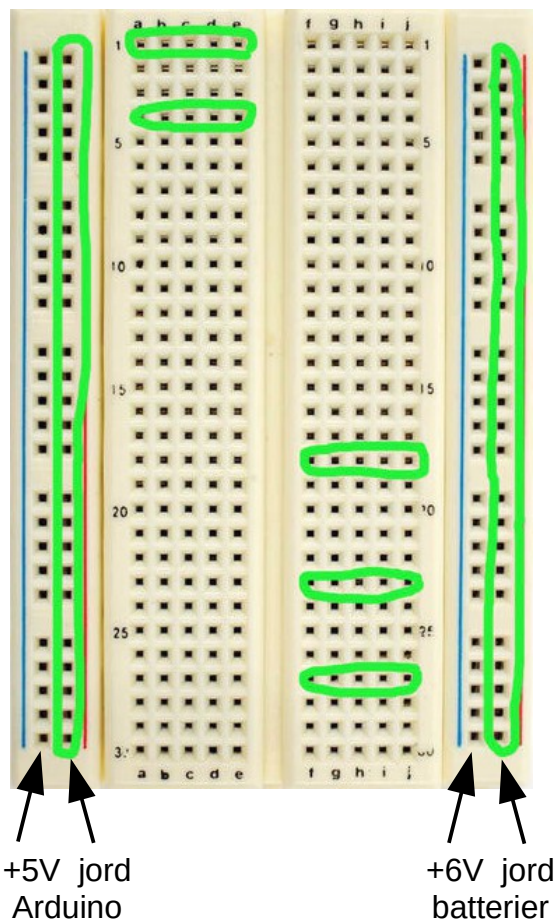
Sensor (avståndsmätare). Avståndsmätare använder ultraljud (ljud som har en högre frekvens än vad det mänskliga örat kan uppfatta) för att mäta avstånd. Robotens "ögon". Fladdermössen använder samma teknik.

Servo. Ett servo innehåller en liten motor som kan vridas, och gör att vi kan rikta avståndsmätaren åt olika håll. Robotens "hals".



Teknik årskurs 9 - Robotprogrammering

En **breadboard** används för att koppla ihop komponenter utan att behöva löda. Hålen är sammanlänkade enligt bilden nedan. Det innebär att alla sladdar som sticks ned i hål som är inom en grön cirkel blir sammankopplade.



Del 1. Servo

Först ska vi lära oss att styra ett **servo**. På det ska vi sedan sätta dit avståndsmätaren (**sensorn**). Ett servo är egentligen en liten motor som vi kan ställa i olika riktningar genom att ange vinkeln i grader.

Först av allt skapar vi en ström- och en jordkolumn på breadboarden:

1. Dra en sladd från +5V på Arduinon till en av breadboardens +-kolumner (en lång vertikal kolumn på breadboardens sida).
2. Dra en annan sladd från GND på Arduinon till --kolumnen på breadboarden (kolumnen bredvid den där du kopplade in +5V).

Nu ansluter vi servot. Servot har tre olika anslutningar:



Teknik årskurs 9 - Robotprogrammering

1. Kontroll. Är ofta orange. Koppla den till port D10 på Arduinon.
2. Jord/GND. Är ofta svart/brun. Koppla den till jord-kolumnen (--kolumnen)på breadboarden
3. Ström, +5V. Är ofta röd. Koppla den till +5V-kolumnen på breadboarden

Vi ansluter också den lilla kondensatorn mellan +5V och GND på servot för att jämna ut spänningen (kondensatorns långa ben ska anslutas till +5V, det korta till GND). Det görs enklast genom att du sticker ned kondensatorn mellan de två kolumnerna som ger +5V och GND på breadboarden.

Testa nedanstående program genom att skriva in det i Arduino IDE (på datorn) och ladda upp det till Arduinon.

```
// Servo
#include <Servo.h> // ett så kallat bibliotek som låter oss använda servot
const int pinServo = 10; // servot är anslutet till D10
Servo myServo; // används för att kommunicera med servot

void setup()
{
  myServo.attach(pinServo);
  myServo.write(90); // antal grader; 90 grader är rakt fram
}

void loop()
{
  myServo.write(0); // antal grader
  delay(1000); // vänta 1000 ms så servot hinner vrida sig

  myServo.write(180); // antal grader
  delay(1000);
}
```

Fundera över hur programmet och kopplingarna samverkar.

Prova att ändra graderna och väntetiden (den anges i ms, millisekunder).

Spara programmet med namnet "robot_servo". Då skapas en mapp med samma namn som programmet.

Del 2. Sensor

Nu ska vi lära oss att mäta avstånd med hjälp av sensorn.

Sensorn har fyra anslutningar. På sensorn kan du se vilken sladd som är vilken anslutning. Koppla in sensorn:

1. Ström (VCC). Koppla den till +5V-kolumnen på breadboarden.



Teknik årskurs 9 - Robotprogrammering

2. Styrsignal (Trig). Koppla den till D19 (A5) på Arduinon .
3. Datasignal/avståndet (Echo). Koppla den till D18 (A4) på Arduinon.
4. Jord. Koppla den till GND-kolumnen på breadboarden.

Programmet skickar en signal till sensorn via Trig-porten. Då skickas en ultraljudspuls ut. Den studsar mot något föremål och kommer sen tillbaka till sensorn.

Eftersom

$s = v * t$, dvs sträckan = hastigheten * tiden

och ljudets hastighet i luft är något vi känner till, kan programmet räkna ut sträckan (avståndet).

Men allt det sker i ett **bibliotek** (engelska library), så vi behöver inte tänka på det. Ett bibliotek är en mängd funktioner som skrivits av någon annan, och som vi kan återanvända. Det bibliotek som vi ska använda heter **NewPing**.

Testa nedanstående program genom att skriva in det i Arduino IDE (på datorn) och ladda upp det till Arduinon.

```
// Sensor
#include <NewPing.h>

int pinLed = 13; // lysdiod

const int pinPingTrig = 19; // sensorn ansluten till stift/port 18 och 19
const int pinPingEcho = 18;
const int pingMax = 500; // hinder längre bort än pingMax centimeter kommer
ej att ses

NewPing myPing(pinPingTrig, pinPingEcho, pingMax); // hjälper oss prata med
avståndsmätaren

int avstandFramat; // avstånd framåt

void setup() {
    pinMode(pinLed, OUTPUT);
    Serial.begin(115200);
}

void loop() {
    avstandFramat = myPing.ping_cm();

    Serial.println(avstandFramat); // skicka tillbaka avståndet till
datorn

    if (avstandFramat < 25) // om avståndet är mindre än 25 cm
    {
        digitalWrite(pinLed, HIGH); // tänd lysdioden
    }
}
```



Teknik årskurs 9 - Robotprogrammering

```
} else {  
  digitalWrite(pinLed, LOW); // släck lysdioden  
}  
  
  delay(100); // vänta 0,1 s innan vi kollar avståndet igen  
}
```

Programmet fungerar så att sensorn mäter avstånd till närmaste föremål och tänd lysdioden om det är mindre än 25 cm till detta. Prova med olika avstånd.

Fundera över hur programmet och kopplingarna samverkar.

Spara programmet med namnet "robot_sensor". Då skapas en mapp med samma namn som programmet.

Programmet använder dessutom något som heter **Serial** för att kunna skicka tillbaka information till din dator.

För att se vad Arduinon skickar tillbaka väljer du Tools > Serial Monitor i Arduino IDE och anger hastigheten till 115200.

Del 3. Servo + sensor

Sätt fast sensorn på servot och sätt fast servot i något fast, exempelvis breadboarden.

Försök kombinera de två ovanstående programmen till ett eget. Det ska göra följande:

1. Titta framåt. Mät avståndet.
2. Titta åt vänster. Mät avståndet.
3. Titta åt höger. Mät avståndet.
4. Om avståndet framåt är minst, tänd lysdioden.

Om vi även har motorer och kan styra dessa, kan vi använda denna programkod för att att välja vilken väg roboten ska ta.

Läs mer/referens

- <http://playground.arduino.cc/Learning/SingleServoExample>
- playground.arduino.cc/Code/NewPing
- <http://arduino.cc/en/Guide/Libraries>



Lektion 3: En motor och H-brygga

I denna lektion ska vi lära oss styra en **motor** som ska:

- köra roboten framåt
- köra roboten bakåt (backa)

Om vi ska styra en elektrisk motor manuellt (alltså utan Arduinon) och välja om den ska snurra åt ena eller andra hållet, så byter vi helt enkelt plats på de två + och - ledningar som ger motorn ström.

Men för att Arduinon ska klara av detta behöver den hjälp av en extra krets kallad **H-brygga**. Med hjälp av olika signaler kan H-bryggan styra motorn så att den

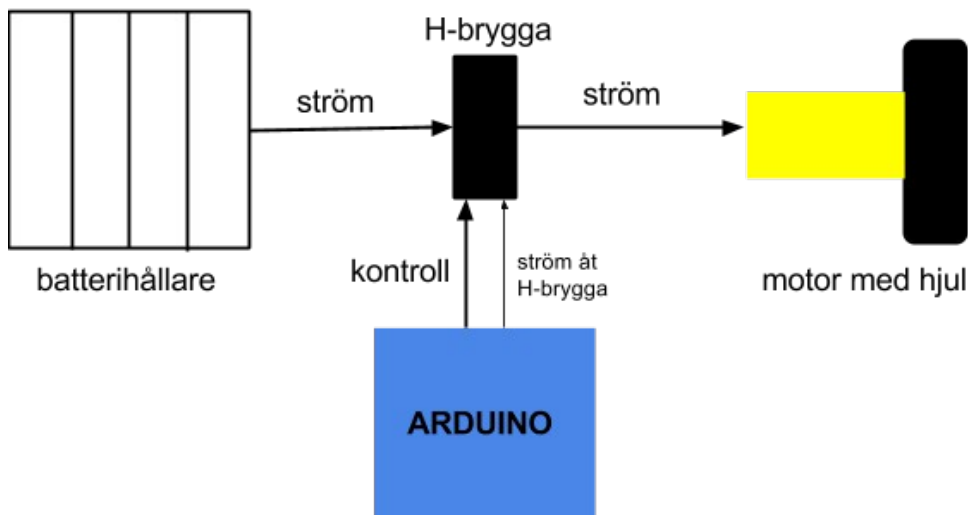
- rör sig framåt
- rör sig bakåt
- stannar

Eftersom H-bryggan kan driva två motorer så kan roboten då

- köra framåt
- köra bakåt
- svänga vänster (genom att stoppa vänster motor och köra den högra framåt)
- svänga höger (genom att stoppa höger motor och köra den vänstra framåt)
- stanna

När du kopplar in H-bryggan så blir det många anslutningar (många sladdar). Det beror på att H-bryggan är ansluten till

- motorn
- batterihållaren (som ger motorerna ström)
- Arduinon (för styrning/kontroll)
- Arduinons +5V (för drift av H-bryggan)



Strömmen från batterierna driver alltså enbart motorerna. Än så länge får Arduinon ström via USB-sladden till datorn, och H-bryggan får ström av Arduinon.

Vår H-brygga kan styra 2 motorer.

Så här styr Arduinon H-bryggan

Arduinon styr motorn genom tre sladdar mellan Arduinon och H-bryggan: Enable, Input 1 och Input 2 (eller Enable, Input 3 och Input 4 för den andra motorn).

Om Enable-porten, som är ett slags av/på, är satt till HIGH (+5V) så motsvarar output (alltså strömmen till motorerna) input (alltså kontrollsignalerna från Arduinon). Kom ihåg, för att en ström ska kunna flöda måste en ledning vara positiv (HIGH) och en negativ (LOW/jord).

Tänk på strömmens + och - som två händer som håller i en ratt. Ratten kan endast snurra om de två händerna drar uppåt respektive nedåt (Input 1 och Input 2). Dessutom måste bilen vara igång för att ratten ska kunna vridas (Enable).

Enable	Input 1	Input 2	Motor
HIGH	HIGH	LOW	bakåt
HIGH	LOW	HIGH	framåt
HIGH	HIGH	HIGH	stopp
HIGH	LOW	LOW	stopp
LOW	betydelslös	betydelslös	stopp



Koppla ihop allt

Se till att Arduinon inte är ansluten till datorn när du gör kopplingarna, och kontrollera dem en extra gång innan du ansluter Arduinon. Var noggrann!

Tejpa fast motorn på en bordskant så att hjulet kan snurra fritt.

Skapa en +5V-kolumn och en jord/GND-kolumn på kopplingsplattans vänstra långsida (på samma sätt som i föregående lektion). Använd Arduinons +5V och GND-stift. De ska driva H-bryggan.

Skapa en +6V-kolumn och en jord/GND-kolumn på kopplingsplattans högra långsida. Sätt i 4 st 1,5V batterier i batterihållaren och anslut den till de två kolumnerna. De ska driva motorn.

Förbind jord/GND hos de båda kolumnerna så att de har en gemensam jord.

Sätt fast H-bryggan på breadboarden på raderna 8-15, med benen på var sin sida om mitten-”diket”. Den lilla gropan på kretsen ena kortsida ska peka uppåt.

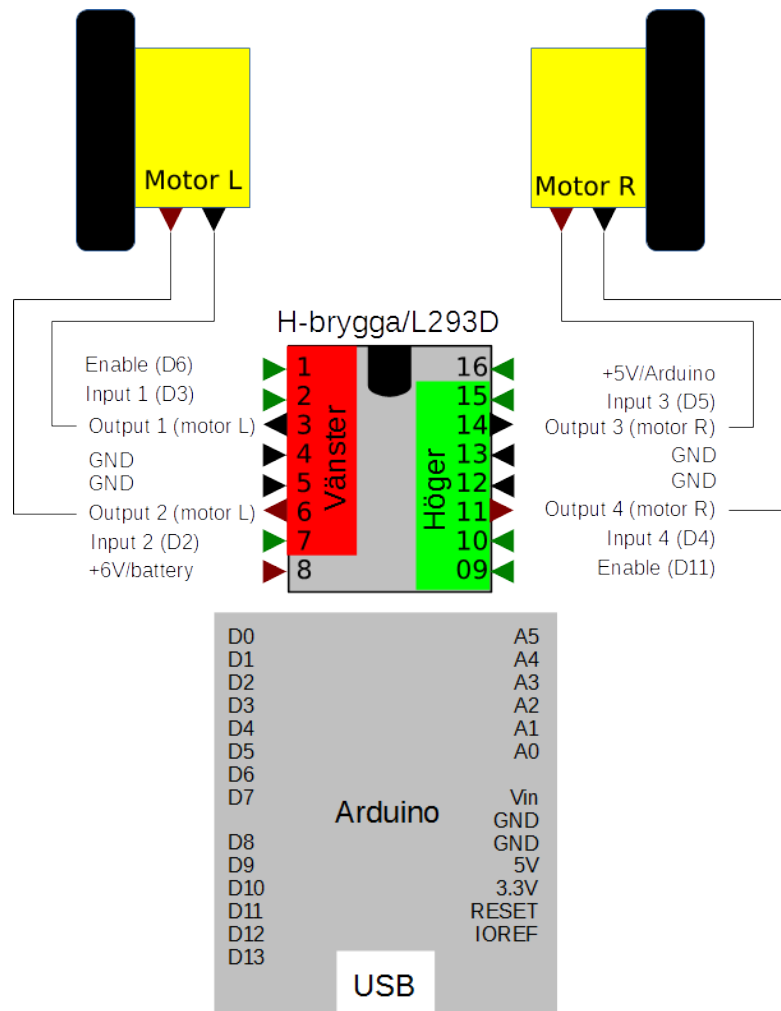
Vi börjar med att ordna anslutningarna för vad som kommer att bli den vänstra motorn:

1. H-bryggan till batterihållarens +-kolumn på breadboarden (ben 8). Detta blir strömen till motorn.
2. H-bryggan till Arduinons +-kolumn på breadboarden (ben 16). Detta är strömmen som driver H-bryggan.
3. H-bryggan till Arduinon (ben 1, 2 och 7 på H-bryggan). Det här är styrsignalerna från Arduinon till H-bryggan.
4. H-bryggan till Arduinons jord-kolumn på breadboarden (ben 4, 5).
5. H-bryggan till motorn (ben 3 och 6). Det här är de två sladdar som sitter fast i motorn.

Nedan följer en ritning över hur motorerna ansluts till H-bryggan. Båda motorerna visas, men nu ska du bara ansluta den vänstra.



Teknik årskurs 9 - Robotprogrammering



Program

Kopiera programmet nedan och för över det till Arduino.

```
/*  
Test av H-brygga och motor  
*/  
  
//  
// konstanter (värden som inte ändras i programmet)  
//  
  
int pinLed = 13; // utgång 13 är kopplat till en LED på Arduino  
  
const int pinVansterMotor1 = 2; // H-bryggan styrs av D2, D3 och D6  
const int pinVansterMotor2 = 3;  
const int pinVansterMotorFart = 6;  
  
const int motorFart = 200; // hur snabbt ska motorerna gå?  
  
//
```



Teknik årskurs 9 - Robotprogrammering

```
// funktionen setup() körs EN GÅNG i början av programmet
//
void setup() {
    // vänster motor
    // den STYR vi (OUTPUT)
    pinMode(pinVansterMotor1, OUTPUT);
    pinMode(pinVansterMotor2, OUTPUT);
    pinMode(pinVansterMotorFart, OUTPUT);
}

//
// funktionen loop() körs om och om igen (tills strömmen bryts)
//
void loop() {
    // framåt
    motorFramat();
    delay(1000);

    // stanna motorn
    motorStopp();
    delay(100);

    // bakåt
    motorBakat();
    delay(1000);

    // stanna motorn
    motorStopp();
    delay(100);
}

void motorFramat(void)
{
    // kör vänster motor
    analogWrite(pinVansterMotorFart, motorFart);
    digitalWrite(pinVansterMotor1, LOW);
    digitalWrite(pinVansterMotor2, HIGH);
}

void motorBakat(void)
{
    // kör vänster motor bakåt
    analogWrite(pinVansterMotorFart, motorFart);
    digitalWrite(pinVansterMotor1, HIGH);
    digitalWrite(pinVansterMotor2, LOW);
}

void motorStopp(void)
{
    // stoppa vänster motor
    analogWrite(pinVansterMotorFart, 0);
    digitalWrite(pinVansterMotor1, LOW);
    digitalWrite(pinVansterMotor2, LOW);
}
```

Läs igenom programmet och studera vilka olika delar (funktioner) som programmet innehåller



Teknik årskurs 9 - Robotprogrammering

och hur det använder dem.

Prova att ändra ordningen på programsatserna i loop()-funktionen.

Prova att ändra värdens för delay()-satserna och motorns fart (variabeln motorFart).

Läs mer/referens

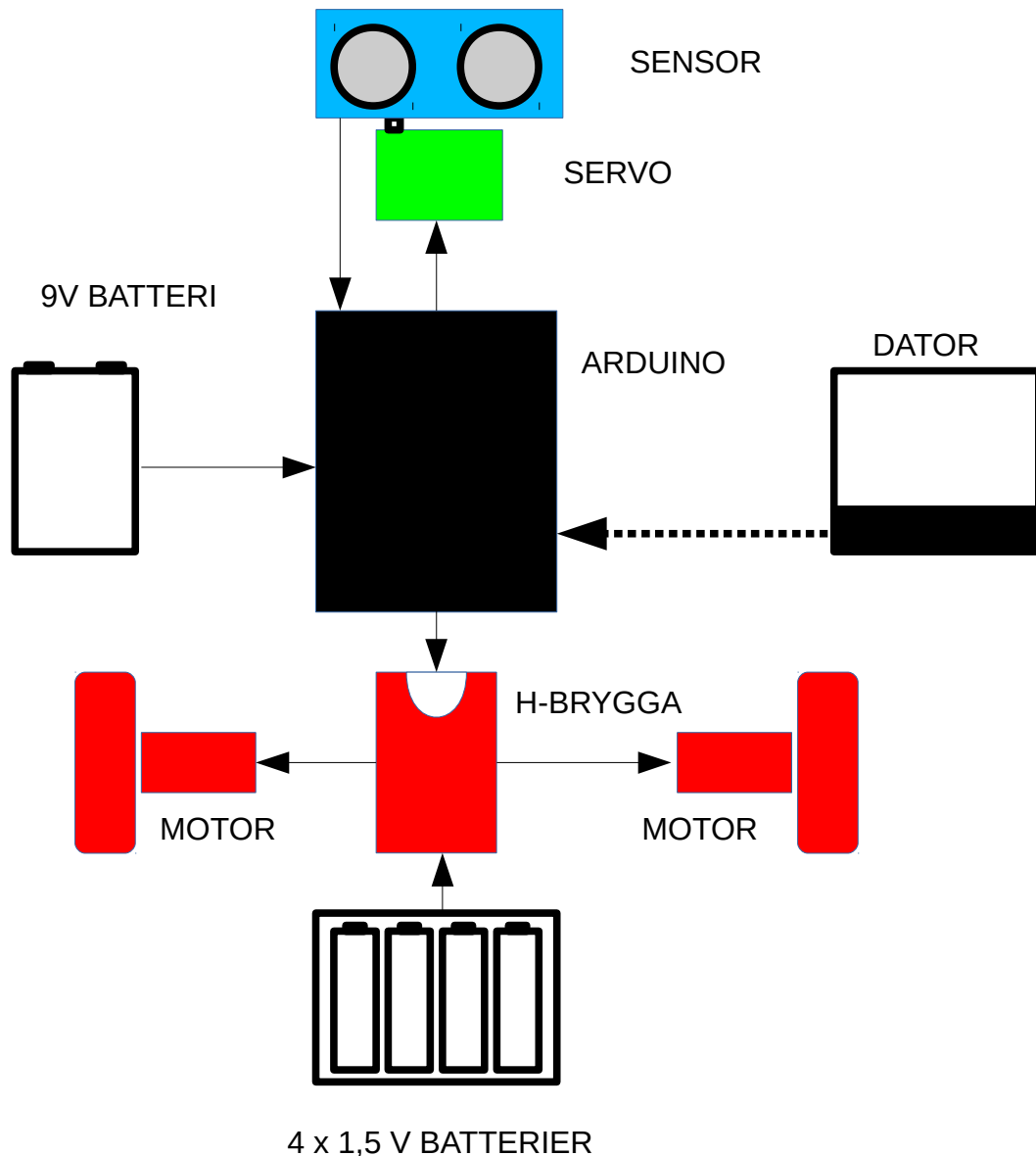
- Hur fungerar en H-brygga inuti?
http://www.robotplatform.com/howto/L293/motor_driver_1.html
- Funktioner i Arduino. <http://arduino.cc/en/Reference/FunctionDeclaration>



Lektion 4: Bygg roboten

Nu ska alla delar sättas samman till en komplett robot!

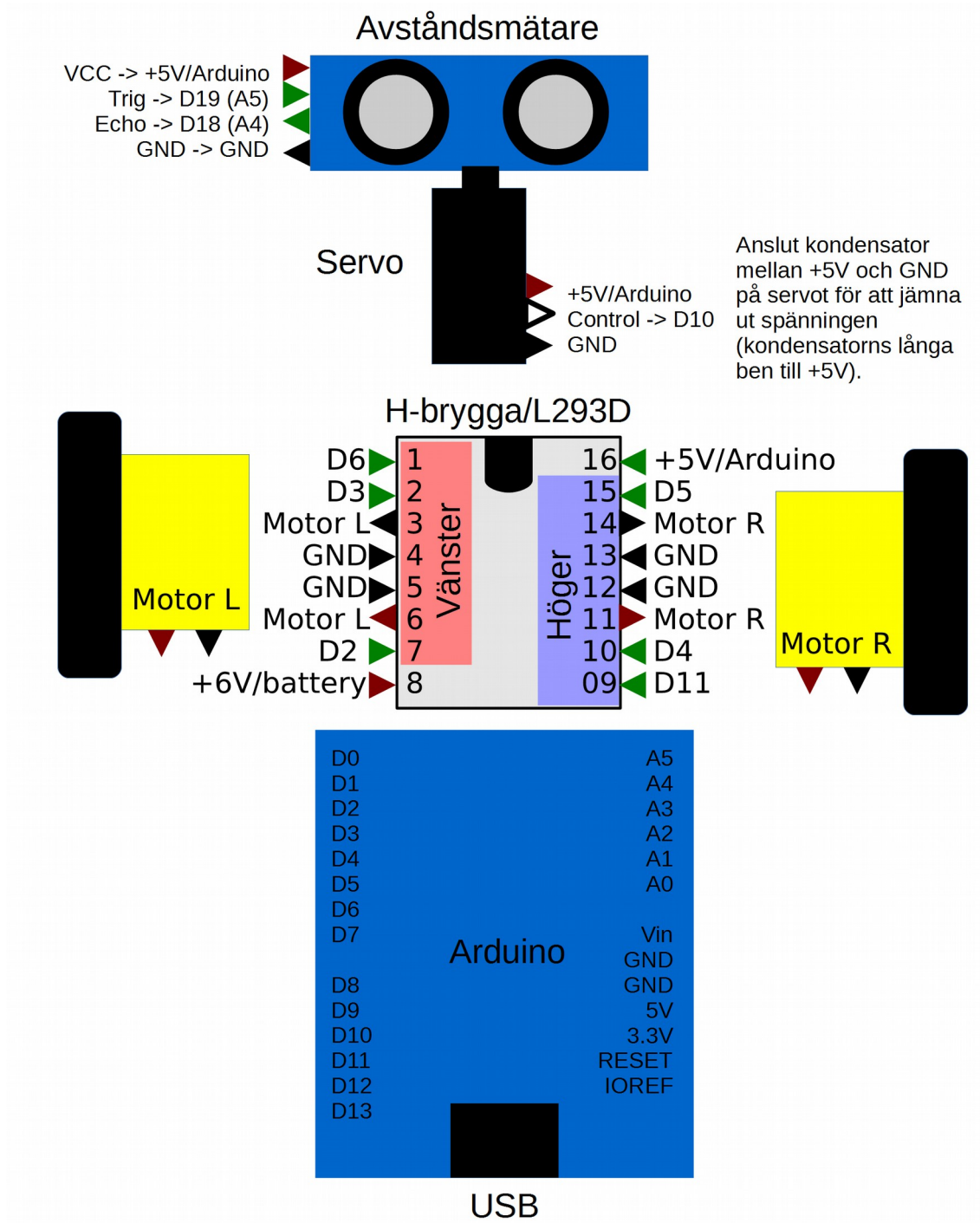
Schema



9V-batteriet ska anslutas till Arduinons Vin (Volt in) och GND (men vänta med att göra det till allt fungerar och roboten är mogen att släppas ned på golvet). Resten av anslutningarna görs enligt tidigare lektioner.



Ritning





Exempel på färdigt program

Programmet är färgkodat på samma sätt som bilden i avsnittet Schema: **motor** (rött), **servo** (grönt), **sensor** (blått). Värderna som är intressanta att experimentera med är markerade med **gult**. Rader i **grått** är kommentarer.

```
/*  
Robot  
Elever: <ditt namn>  
Ursprunglig version: 2014-08-31/Staffan Melin  
Kräver NewPing library  
http://playground.arduino.cc/Code/NewPing  
*/  
  
//  
// Vårt program behöver inkludera (lägga till) lite funktionalitet  
// Det kallas för bibliotek.  
//  
  
#include <Servo.h>  
#include <NewPing.h>  
#include <SoftwareSerial.h>  
  
//  
// konstanter (värden som inte ändras i programmet)  
//  
  
int pinLed = 13; // utgång 13 är kopplat till en LED på Arduinon  
  
const int pinVansterMotor1 = 2;  
const int pinVansterMotor2 = 3;  
const int pinVansterMotorFart = 6;  
  
const int pinHogerMotor1 = 4;  
const int pinHogerMotor2 = 5;  
const int pinHogerMotorFart = 11;  
  
const int pinServo = 10;  
  
const int pinPingTrig = 19;  
const int pinPingEcho = 18;  
const int pingMax = 200; // hinder längre bort än pingMax centimeter kommer ej att ses  
  
const int motorFart = 200; // hur snabbt ska motorerna gå? (160-250 är ok)  
  
//  
// globala variabler (syns i hela programmet)  
//  
  
Servo myServo; // hjälper oss prata med servot  
NewPing myPing(pinPingTrig, pinPingEcho, pingMax); // hjälper oss prata med  
avståndsmätaren  
  
int avstandFramat; // uppnått avstånd framåt  
int avstandVanster; // uppnått avstånd åt vänster
```



Teknik årskurs 9 - Robotprogrammering

```
int avstandHoger; // uppnått avstånd åt höger
```

```
//  
// funktionen setup() körs EN GÅNG i början av programmet  
//
```

```
void setup() {  
    pinMode(pinLed, OUTPUT);  
  
    // avståndsmätare (ping = ultraljudssensor)  
    // den LÄSER vi av (INPUT)  
    // Dessa två inställningar görs av NewPing myPing(...) ovan, så de behöver  
    // vi inte tänka på  
    // pinMode(pinPingTrig, OUTPUT);  
    // pinMode(pinPingEcho, INPUT);  
  
    // servo som styr avståndsmätaren  
    // den STYR vi  
    myServo.attach(pinServo);  
    myServo.write(90); // ställ in servot på 90 grader (= rakt fram)  
  
    // vänster motor  
    // den STYR vi (OUTPUT)  
    pinMode(pinVansterMotor1, OUTPUT);  
    pinMode(pinVansterMotor2, OUTPUT);  
    pinMode(pinVansterMotorFart, OUTPUT);  
  
    // höger motor  
    // den STYR vi (OUTPUT)  
    pinMode(pinHogerMotor1, OUTPUT);  
    pinMode(pinHogerMotor2, OUTPUT);  
    pinMode(pinHogerMotorFart, OUTPUT);  
  
    // sätt igång roboten framåt  
    motorFramat();  
}
```

```
//  
// funktionen loop() körs om och om igen (tills strömmen bryts)  
//
```

```
void loop() {  
    avstandFramat = pingMedelvarde();  
  
    // vi bryr oss inte om avstånd som är 0 (för avståndsmätaren visar ibland  
    // fel)  
    if (avstandFramat > 0)  
    {  
        // har vi ett hinder framför roboten?  
        if (avstandFramat < 25)  
        {  
            digitalWrite(pinLed, HIGH); // tänd - roboten tänker  
  
            motorStopp();  
  
            // titta till vänster och mät avståndet  
            myServo.write(10);  
            delay(1000); // vänta 1 sekund så att servot hinner vrida sig  
            avstandVanster = pingMedelvarde();  
        }  
    }  
}
```



Teknik årskurs 9 - Robotprogrammering

```
// titta till höger och mät avståndet
myServo.write(170);
delay(1000);
avstandHoger = pingMedelvarde());

// vrid servot framåt
myServo.write(90);
delay(1000);

// bestäm vad roboten ska göra nu

// finns det hinder BÅDE till vänster och höger?
if (avstandVanster < 25 && avstandHoger < 25) {
    motorBakat();
    delay(1000); // vänta ett tag medan roboten backar
    if (random(2) == 0)
        motorSvangVanster();
    else
        motorSvangHoger();
    delay(500); // vänta ett tag medan roboten svänger

// är det friare till vänster?
} else if (avstandVanster > avstandHoger)
{
    motorSvangVanster();
    delay(1000); // vänta ett tag medan roboten svänger

// tydligen är det friare till höger!
} else {
    motorSvangHoger();
    delay(1000); // vänta ett tag medan roboten svänger
}

digitalWrite(pinLed, LOW); // släck lysdioden - roboten har
funderat färdigt

    motorFramat());

} else {
    // det finns inget hinder så kör framåt
    motorFramat());
}

}

void motorSvangHoger(void)
{
    // stoppa höger motor
    analogWrite(pinHogerMotorFart, 0);

    // kör vänster motor
    analogWrite(pinVansterMotorFart, motorFart);
    digitalWrite(pinVansterMotor1, LOW);
    digitalWrite(pinVansterMotor2, HIGH);
}

void motorSvangVanster(void)
{
    // stoppa vänster motor
    analogWrite(pinVansterMotorFart, 0);

    // kör höger motor
    analogWrite(pinHogerMotorFart, motorFart);
    digitalWrite(pinHogerMotor1, LOW);
    digitalWrite(pinHogerMotor2, HIGH);
}
```



Teknik årskurs 9 - Robotprogrammering

```
}  
  
void motorFramat(void)  
{  
    // kör vänster motor  
    analogWrite(pinVansterMotorFart, motorFart);  
    digitalWrite(pinVansterMotor1, LOW);  
    digitalWrite(pinVansterMotor2, HIGH);  
  
    // kör även höger motor  
    analogWrite(pinHogerMotorFart, motorFart);  
    digitalWrite(pinHogerMotor1, LOW);  
    digitalWrite(pinHogerMotor2, HIGH);  
  
}  
  
void motorBakat(void)  
{  
  
    // kör vänster motor bakåt  
    analogWrite(pinVansterMotorFart, motorFart);  
    digitalWrite(pinVansterMotor1, HIGH);  
    digitalWrite(pinVansterMotor2, LOW);  
  
    // kör höger motor bakåt  
    analogWrite(pinHogerMotorFart, motorFart);  
    digitalWrite(pinHogerMotor1, HIGH);  
    digitalWrite(pinHogerMotor2, LOW);  
  
}  
  
void motorStopp(void)  
{  
  
    // stoppa vänster motor  
    analogWrite(pinVansterMotorFart, 0);  
    digitalWrite(pinVansterMotor1, LOW);  
    digitalWrite(pinVansterMotor2, LOW);  
  
    // stoppa höger motor  
    analogWrite(pinHogerMotorFart, 0);  
    digitalWrite(pinHogerMotor1, LOW);  
    digitalWrite(pinHogerMotor2, LOW);  
  
}  
  
// Den här funktionen mäter avståndet (i centimeter).  
// Avståndsmätaren ger ibland fel värde, så mät 5 gånger och ta medelvärdet.  
  
int pingMedelvarde()  
{  
    int i;  
    int totAvstand = 0;  
  
    for (i = 0; i < 5; i++)  
    {  
        totAvstand = totAvstand + myPing.ping_cm();  
    }  
  
    return (totAvstand / 5);  
}
```



Arbetsområde: ROBOT - KONSTRURERA & PROGRAMMERA

Centralt innehåll:	Vad jag ska öva på, lära mig och bli bedömd på:	Elevutvärdering:
<ul style="list-style-type: none"> - Styrning av digitala/elektroniska kretsar och komponenter med hjälp av dator. - Elektronik och elektroniska komponenter samt de begrepp som behövs för att beskriva dessa och hur de samverkar. - Undersöka, föreslå, konstruera och prova olika tekniska lösningar. - Dokumentera arbetet med film och skriven loggbok. 	<ul style="list-style-type: none"> - Koppla ihop en robot, motorstyrets, motor, avståndsmätare och batterier. - Bygga ihop de olika delarna till en robot som kan köra med egen kraft. - Förstå och förklara hur de olika delarna kommunicerar med varandra. 	<p>Min förmåga att beskriva hur roboten är konstruerad, och programmerad:</p>
<p>Hur vi ska arbeta:</p> <ul style="list-style-type: none"> - Projektet sker i grupp (3 personer). - Bygga ihop en datorstyrd rullande robot med hjälp av dator(er), motorer, servo, avståndsmätare och annan elektronik. - Programmera datorn så att roboten kan åka framåt och svänga för att undvika hinder. 	<ul style="list-style-type: none"> - Undersöka och modifiera program skrivna i Arduino - Föra över program från en bärbar dator till Arduino - Utvärdera och förbättra konstruktionen. 	<p>Min förmåga att arbeta självständigt och pröva olika tekniska lösningar:</p>
<p>Hur du redovisar dina kunskaper:</p> <ul style="list-style-type: none"> - Filma och sammanfatta dagens arbete med iPad. - Skriva individuell loggbok i GDrive. 		<p>Min förmåga att dokumentera arbetet med roboten:</p>



Teknik årskurs 9 - Robotprogrammering



	Kunskapskrav E:	Summativ bed.	Kunskapskrav C:	Summativ bed.	Kunskapskrav A:	Summativ bed.
<input type="checkbox"/>	Använder NÅGRA tekniska begrepp när du beskriver hur en teknisk lösning är konstruerad och fungerar. (BESKRIVA TEKNISKA LÖSNINGAR)		Använder ALLTID tekniska begrepp när du beskriver hur en teknisk lösning är konstruerad och hur dess delar fungerar och samverkar		Använder alltid, och på ett KORREKT SÄTT tekniska begrepp när du beskriver hur en teknisk lösning är konstruerad och hur dess delar fungerar och samverkar, samt VISAR PÅ LIKNANDE LÖSNINGAR	
<input type="checkbox"/>	Kan bygga enkla konstruktioner och PRÖVA lösningar för detta (BYGGA OCH TESTA OLIKA LÖSNINGAR)		Kan bygga enkla konstruktioner och pröva lösningar för detta samt FÖRÄNDRA dessa lösningar för att nå ett bättre resultat		Kan bygga enkla konstruktioner och SYSTEMATISKT PRÖVA dig fram till lösningar som ger så bra resultat som möjligt	
<input type="checkbox"/>	Kan med LÄRARES HJÄLP ta dig framåt i arbetet (SJÄLVSTÄNDIGT ARBETE)		Kan med VISS HJÄLP från lärare ta dig framåt i arbetet		Kan med SJÄLVSTÄNDIGT ta dig framåt i arbetet	
<input type="checkbox"/>	Gör ENKLA dokumentationer med bilder, texter och skisser (DOKUMENTERA)		Gör TYDLIGA dokumentationer med bilder, texter och skisser		Gör DETALJERADE dokumentationer med bilder, texter och skisser	
<input type="checkbox"/>						
Formativt omdöme:						
Till TUP:						
F <input type="checkbox"/>	E <input type="checkbox"/>	D <input type="checkbox"/>	C <input type="checkbox"/>	B <input type="checkbox"/>	A <input type="checkbox"/>	