

# Fakta och praktiska tips

## Den smartare staden



Med stöd från

**VINNOVA**  
Sveriges innovationsmyndighet

 **Energimyndigheten**

**FORMAS** 

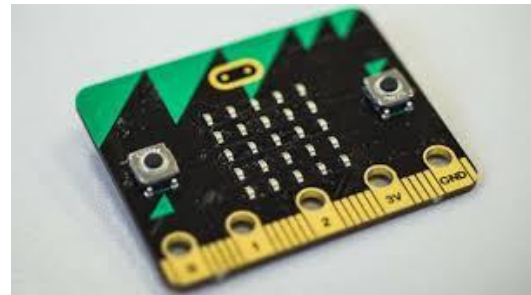
**Strategiska  
innovations-  
program**

## Inledning

Dokumentet innehåller fakta, handledning, praktiska tips mm som stöd till de laborativa momenten inom digitalteknik för ämnet Teknik på grundskolan. Materialet är avsett för främst NTA:s tema-utbildare samt lärare och pedagoger i NTA-temat Den smartare staden. Delar av innehållet kan mycket väl anpassas för eleverna där läraren finner det lämpligt. I ett NTA-tema kallas en uppgift för ett uppdrag.

Uppdragen i de båda temana omfattar moment ur läroplanen, som exempelvis:

- Tekniska lösningar som utnyttjar elektriska komponenter och enkel elektronik för att åstadkomma ljud, ljus eller rörelse för exempelvis larm och belysning
- Hur komponenter och delsystem samverkar i ett större system
- Tekniska lösningar för styrning och reglering
- Samspel mellan människa och teknik, samt människans möjligheter att skapa tekniska lösningar som bidrar till hållbar utveckling.
- Dokumentation och terminologi



## Micro:bit och Teknik

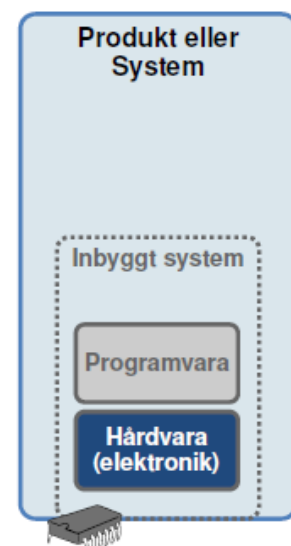
De praktiska momenten i våra NTA-teman Smartare produkter och Den smartare staden utgår från en s k plattform, ett datorsystem, som kallas Micro:bit. En Micro:bit är en enkortsdator och kan användas som ett system för inbyggnad i produkter. Flera namn finns på samma tekniska system, som är komplext men lättanvänt och innefattar ett antal kompetensområden inom digitalteknikområdet. Att arbeta med och lära sig tekniken runt ett inbyggt system innefattar områdena elektronik, datateknik, datavetenskap (programmering) samt kunskap om gränssnittet till sensorer och ställdon (lampor, motorer etc.) som ska anslutas till systemet.

Förutom det rent tekniska innebär det också att förstå andra behov som de olika applikationsområdena ställer på ett elektroniksystem som ska byggas in ett sammanhang.

Micro:bit är framtaget för utbildningsändamål på grundskolenivå, men skulle mycket väl kunna användas som ett system för att exempelvis styra en tvättmaskin och andra vitvaror eller ingå som ett elektroniksystemen i ett fordon. Användningsområdena är många, det är bara fantasin som sätter gränser.

Användningen av elektronik är i högsta grad branschöverskridande och utgör en allt större och viktigare del av de produkter och tjänster som utvecklas. I kombination med programvara är systemen ett oslagbart verktyg i takt med behovet av nya lösningar för att möta de globala utmaningar världen står inför. Det kan gälla minskad energiförbrukning, hållbar miljöutveckling, vård och omsorg för en åldrande befolkning, bara för att nämna några.

**Ett motsvarande dokument finns för temat Smartare produkter.**



## Innehåll

Inledning.....	2
Micro:bit och Teknik.....	2
Elektronikbranschen.....	6
Spetsområden i Sverige.....	6
Tema - Den smartare staden.....	6
Micro:bit med enheter i systemet.....	7
Utrustning och tillbehör.....	8
Förberedelser med dator som programmeringsverktyg.....	8
Inledning.....	8
Terminologi – en sammanfattning.....	11
Val av plats för fil vid nedladdning.....	12
Firmware.....	14
Lite fakta kring Firmware.....	15
Praktiska tips och trix.....	16
Före lektion.....	16
Under lektion.....	16
Efter lektion.....	16
Micro:bit och surfplattor med trådlös överföring av program.....	16
Support – vad finns på nätet?.....	16
Användbara länkar med information.....	17
Allmänt.....	17
Tekniska aspekter och möjligheter.....	17
Micro:bit och nedladdning av program.....	17
Kan något gå fel – en felkälla?.....	17
Strömförsörjning.....	18
ESD – en introduktion.....	19
ESD-påsar.....	19
Uppdragen i temat med fakta och kommentarer.....	20
Symboler flödesdiagram.....	20
Uppdrag 2 – Smartare gatubelysning.....	21
Ljussensor på micro:bit-kortet.....	21
Programmering med den interna ljussensorn.....	22
Extern ljussensor.....	23
Mer om hårdvaran i gatubelysning.....	24
Komponentfakta.....	25

Talsystem.....	26
Decimala talsystemet .....	26
Binära talsystemet.....	26
Digital signal .....	27
Grundprogram för belysning med flödesschema.....	27
Hysteres.....	28
Hysteres vid magnetisering .....	29
Felsökning av gatubelysning (Lamp:bit) .....	30
Trasiga komponenter .....	30
Fördjupning – Fotonik .....	31
Fysikaliska grunder .....	31
Övergången till teknik med nytta.....	32
Energimyndigheten testar .....	33
Uppdrag A – Smartare trafiksystem .....	34
Grundprogram för trafikljuset.....	35
Händelser .....	36
Kitroniks tillägg.....	37
Kitroniks funktioner.....	37
Lite mer om felsökning.....	38
Radio.....	39
Exempel .....	39
Uppdrag B – Smartare äldreomsorg.....	40
Inledning.....	40
Utrustning och inledande val .....	41
Programmering och loopar .....	41
Programmets flödesdiagram.....	42
Fall-larmet .....	43
Använda utvecklingsmiljön .....	44
Simulera.....	44
Mikro- och nanoelektronik.....	46
Micro:bit version 2 .....	47
Testprogram med de nya funktionerna i version 2.....	48
Uppdrag C – Kylkedjan .....	49
Fakta om sensorn och program.....	50
Simulering och Debug Mode .....	51
Fysiska mätningar.....	52

Kalibrering av termometern i kylkedjeuppdraget.....	53
Sammanfattning av programmeringen i Kylkedjan.....	54
Tryckt elektronik.....	55
Uppdrag D – Smartare säkerhetssystem.....	56
Utrustning och Kitroniks funktioner.....	56
Radio.....	58
Mekatronik.....	58
Modell av en Micro:bit.....	59
Kraftelektronik.....	60
Trådlös kommunikation.....	61
Antenner för mikrovågs- och terahertzsystem (AMT).....	61
Inbyggda system och sensorer.....	62
Sensorer och AI.....	63
Sammanfattning.....	64
Utvecklingsmiljön för Micro:bit.....	64
Hårdvarunäva programmering.....	65
Extramaterial.....	66
Fakta om 5g och IoT.....	66
Rekonstruktion.....	68
Diskussionspunkter.....	69
Tekniska system i staden.....	69
Globala mål och intressen kontra Micro:bit.....	69
Innovation och entreprenörskap.....	70

## Elektronikbranschen

Den svenska elektronikbranschen är verksam på en rad teknikområden och i ett stort antal tillämpningsområden. I detta dokument vävs fakta kring elektronikområden och programmering samman med de uppdrag som temat innehåller.

### Spetsområden i Sverige

I programmet Smartare Elektroniksystem har man identifierat sju svenska spetsområden när det gäller elektronik, det gäller:

- Fotonik
- Tryckt elektronik
- Mikro- och nanoelektronik
- Sensorer
- Inbyggda system
- Kraftelektronik
- Antenn-, mikrovåg- och terahertzsystem

Spetsområdena kommer att tas upp och kort beskrivas i dokumentet i syfte att vävas ihop med de uppdrag vi arbetar med i temat *Den smartare staden*. Inom spetsområdena kommer programvara också att spela en allt större roll i morgondagens applikationer och kommer att vara en av drivkrafterna i den tekniska utvecklingen.

Ett programvaruområde går under kategorin Artificiell intelligens (AI) och väntas få en betydande roll i framtiden, exempelvis i utvecklingen av smarta hem med produkter och system som smarta TV-apparater, smarta högtalare, smart belysning och samt säkerhet och hälsotillämpningar. Det lanseras komponenter som är skräddarsydda för sådana applikationer, exempelvis kretsen till höger för röststyrning och smarta högtalare.



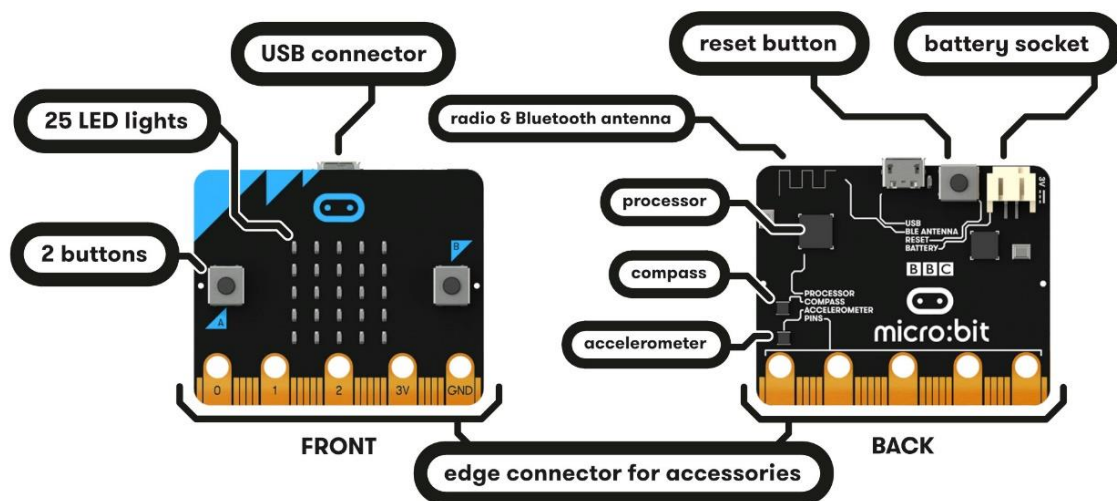
## Tema - Den smartare staden

Den smartare staden innehåller sex experimentella uppdrag och detta dokument avser att vara ett stöddokument med fakta och tillägg till uppdragen, främst för lärare. Mico:bit är plattformen vi arbetar med, och den är ett elektroniksystem som förutom processor innehåller ett antal tekniska funktioner för att kunna använda systemet och för att kunna sätta in elektronik och programmering in i ett sammanhang. I bilderna på kommande sidor ses microbit-kortets fram och baksida med de olika komponenterna och enheterna, samt bilder på den utrustning som ingår i de olika uppdragen i temat Den smartare staden (DSS). Oavsett version av Micro:bit fungerar uppdragen enligt handledning och beskrivningar. När det gäller version 2 av Micro:bit finns den beskriven i dokumentet, även med ett par exempel på hur den kan användas.

Förutom fakta om komponenter eller delsystemen finns även alternativa eller fördjupande exempel på programblock eller programavsnitt som kan användas i att veta mer kring uppdragen i temat. Vi återkommer som sagt till delarna i systemet i texter och bilder som följer.

## Micro:bit med enheter i systemet

### Version 1.5



### Version 2.0



### Micro:bit version 2 och våra uppdrag

Nyheter i hårdvara och funktioner:

- En mikrofon (syns på fram- och baksida)
- En högtalare (baksida)
- En touch-sensor (framsida)
- Antennen är vriden 45 grader av utrymmesskäl
- Det ska var enklare att fästa krokodilklämmor på kortet med urfasningarna nedtill
- Utvecklingsmiljön ser likadan ut och bland programmeringsblocken finns mjukvarustöd för att använda den nya hårdvaran, vilket för vår del ger utökade möjligheter till fördjupning etc.

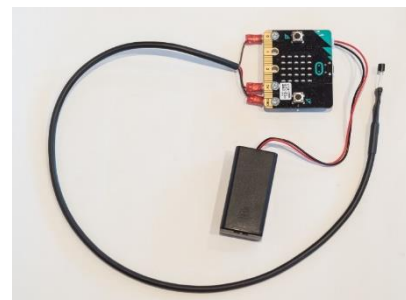
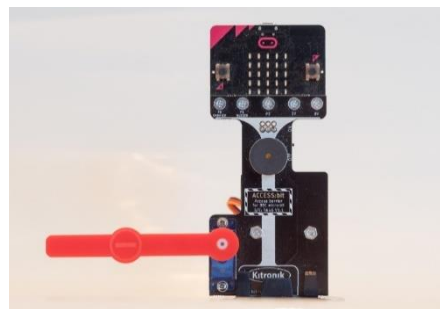
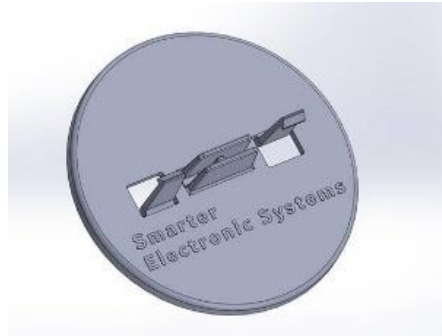
Enligt förhandsinformationen från Kitronik ska det inte vara några förändringar som kommer att påverka vår utrustning och våra uppdrag. Vi har testat igenom uppdragen med version 2, och allt ser ut att stämma (14 jan 2021). I dokumentet finns lite mer information om version 2 på sidan 14 samt mer omfattande med ett par applikationsexempel i ett separat kapitel/moment på sidan 47 - 48.

## Utrustning och tillbehör

Utrustning med tillbehör i temat Den smartare staden ses i bilderna nedan. Enheterna är dels utvecklade av företaget Kitronik, dels framtagna i ett samarbete mellan NTA Skolutveckling och Smartare elektroniksystem vid utvecklingen av temat. Till detta kommer annan kompletterande utrustning som behövs för att praktiskt kunna arbeta med uppdragen.

Grundutrustningen består av:

- Micro:bit med processor och andra enheter på kortet
- Belysning med ljussensor
- Trafikljus med LED
- Vägbovm med servomotor
- Temperatursensor med kablage
- Plastfot för att praktiskt kunna använda Kitroniks utrustning



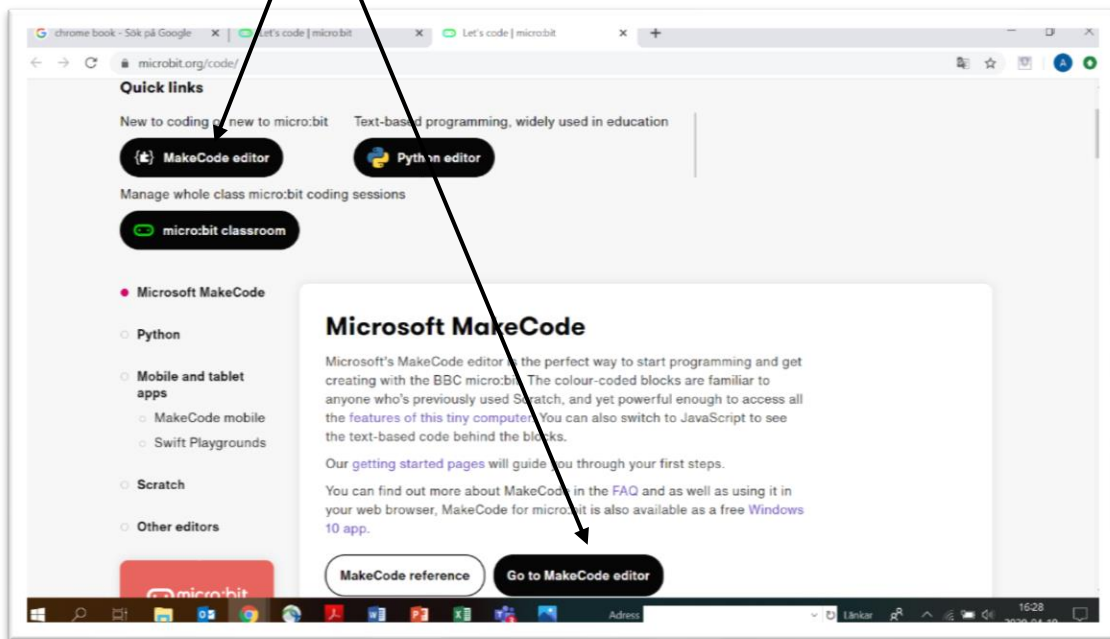
## Förberedelser med dator som programmeringsverktyg

Här följer några korta punkter i förberedelserna om du använder en dator eller Chromebook som hjälpmedel vid programmering, simulering och nedladdning av program till Micro:bit.

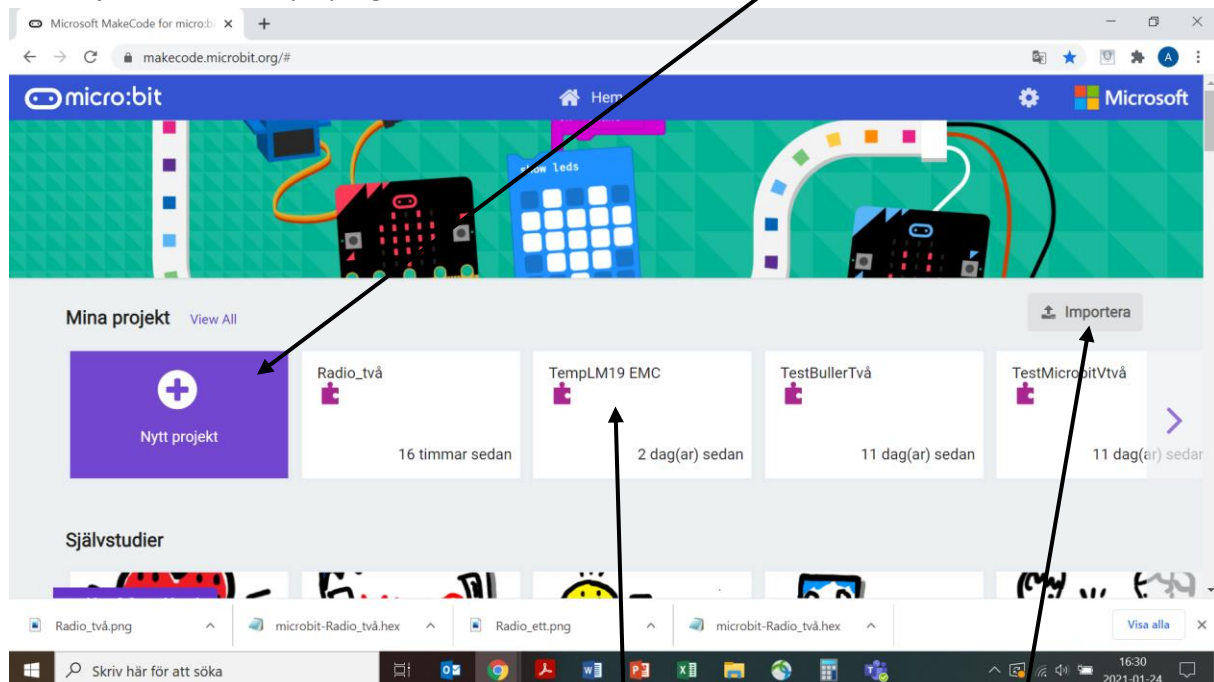
### Inledning

- Förbered din dator för programmering av en micro:bit
  - Koppla in micro:bit-kortet till ett USB-uttag på din dator
  - Skapa ett bibliotek på datorn där du vill spara dina filer, dina program
  - Datorn ska vara ansluten till Internet
  - Använd gärna webbläsaren Google Chrome, men även den nya Edge ska fungera bra.
- För att komma igång med att programmera micro:bit följer några steg på vägen
  - Gå till länken <https://microbit.org>
  - Välj **Lets code** (adress <https://microbit.org/code/>)

- I det nya fönstret som kommer upp då:
  - Välj knappen **MakeCode editor**, två knappar som ger samma resultat (fortsättning)



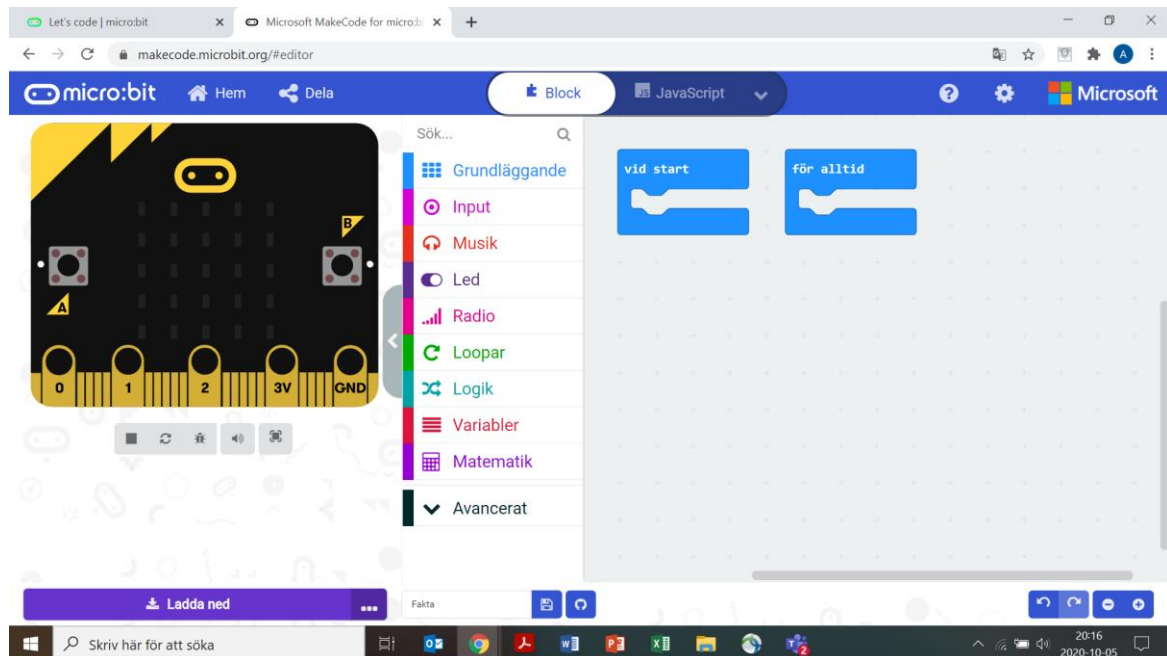
- Ännu ett nytt fönster där du kan välja att starta ett **Nytt Projekt (New Project)**, döpa det och börja utveckla ett nytt program.



- Alternativet är att välja något som du arbetat med tidigare (nyligen) eller att importera ett tidigare projekt från en separat katalog eller enhet.
- Tidigare projekt hittas som ett antal alternativ till höger om knappen New Projekt.

Observera att utvecklingsmiljön uppdateras emellanåt med tillägg etc. vilket gör att det som står i detta dokument bör justeras. Bevaka och tipsa gärna om det i så fall!

Slutligen kommer man till det fönster vi kallar för Micro:bits utvecklingsmiljö. Här utvecklas programmet i den sk editorn, dvs programmeringsytan till höger. I mitten finns rubrikerna för de block som används vid programmeringen och till vänster ses simulatören.



Programmet utvecklas som sagt på programmeringsytan till höger. När du känner dig klar med din kod har du ibland möjlighet att direkt **Simulera** ditt program, dvs se resultatet och verifiera programmet i simuleringsfönstret.

Simulatoren i programmeringsmiljön är ett bra verktyg för att analysera det program man arbetar med. Knapparna under simulatorfönstret har olika funktioner. Om du arbetat med temat Smartare Produkter har du förmodligen blivit bekant med simulatören, och den kommer även att användas i detta tema. Testa gärna om du vill repetera funktionerna.

I den senaste versionen av utvecklingsmiljön finns det möjlighet att *debugga* programmet man arbetar med. Se kommentaren i bilden till höger. Mer om den funktionen kommer vi tillbaka till i detta dokument, exempelvis under sidorna för uppdraget Kylkedjan.



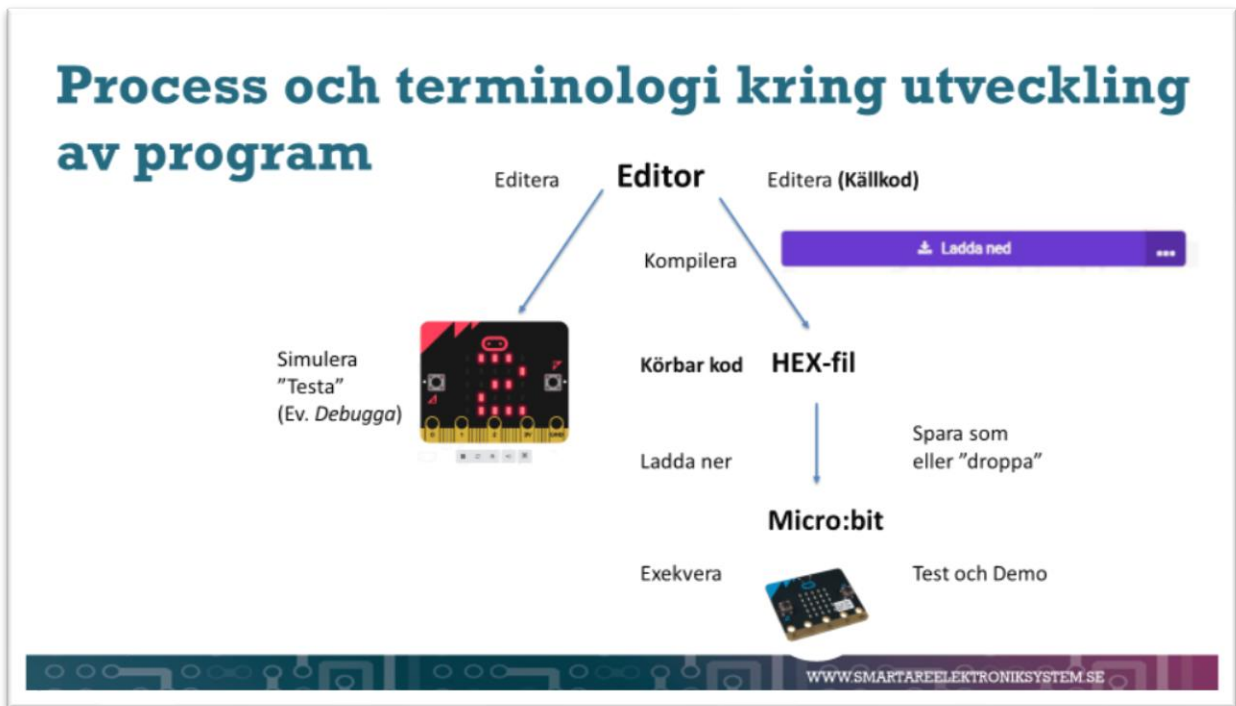
Huvudspåret i Den smartare staden (DSS) blir att programmet ska föras över till microbitkortet för att sedan exekvera och utföra den uppgift som uppdraget avser. När extern elektronik är inkopplad till microbiten finns det begränsningar i att använda simulatören och *Debug Mode* för programmet.

Att ladda ner programmet till microbit sker i ett par steg, vilka är lite olika beroende på vilken programmeringsutrustning som används. Med en dator sker nedladdningen av programmets hex-fil via USB-sladden till micro:bit. Med en surfplatta överförs programmet via trådlös kommunikation där kommunikationsteknologin kallas Bluetooth Low Energy (BLE). Se kommande sidor.

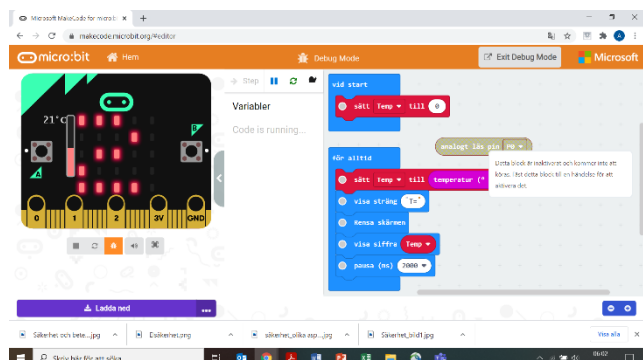
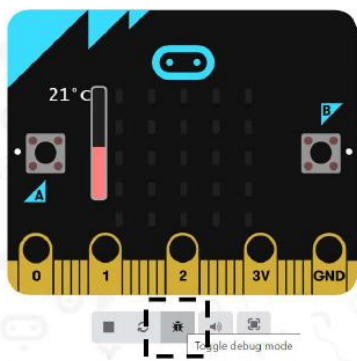
## Terminologi – en sammanfattning

I nedanstående bild ses processen samt viss terminologi kring arbetet med att utveckla, simulera och ladda ner applikationsprogram för Micro:bit. Bilden är generell och beskriver i princip de olika faser eller moment som en programvaruutvecklare befinner sig i sitt arbete.

Identifiera gärna var du/eleverna befinner sig under utvecklingen av uppdragen i temat, det underlättar i diskussionen kring den metod som används vid utvecklingen av ett program.



**Anm:** I simulatorn kan man välja att övergå till sk Debug Mode via ett klick på knappen med nyckelpigan. Se bilderna nedan samt vidare information i uppdragen Äldreomsorg eller Kylkedja.

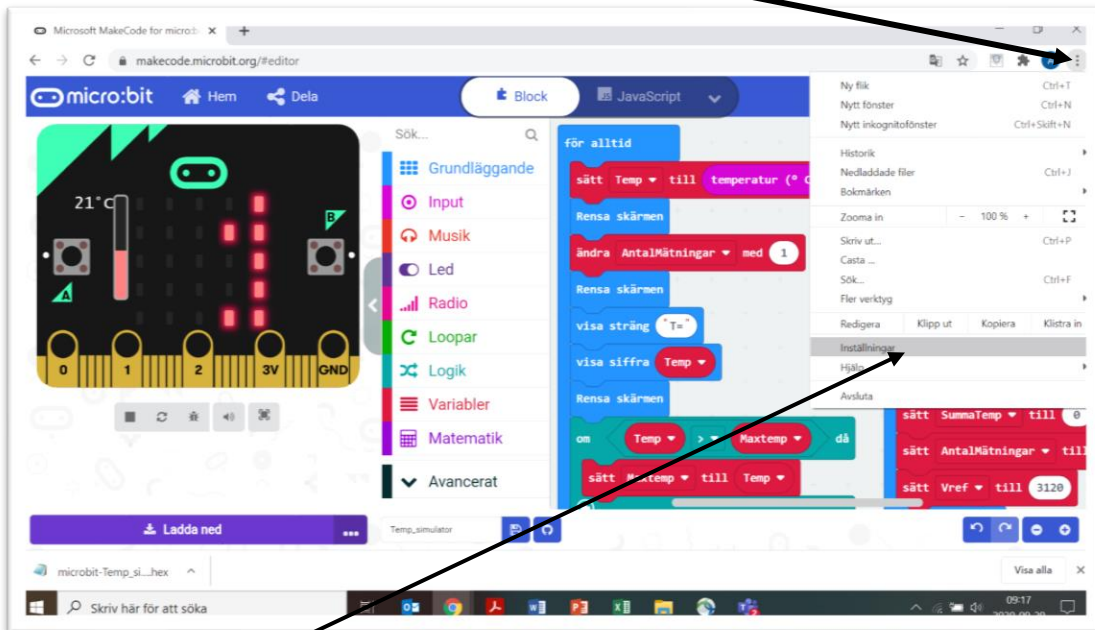


## Val av plats för fil vid nedladdning

Vid nedladdning av en Hex-fil, den fil som skapas med knappen **Ladda ner** (*Download*), kan den med fördel droppas från katalogen hämtade filer till Micro:biten. Det är pedagogiskt och bra, men i den senaste versionen av utvecklingsmiljön finns möjligheten att snabba på (kanske förenkla) detta moment. Det handlar om att välja plats, dvs katalog eller enhet, där den färdiga HEX-filen ska sparas.

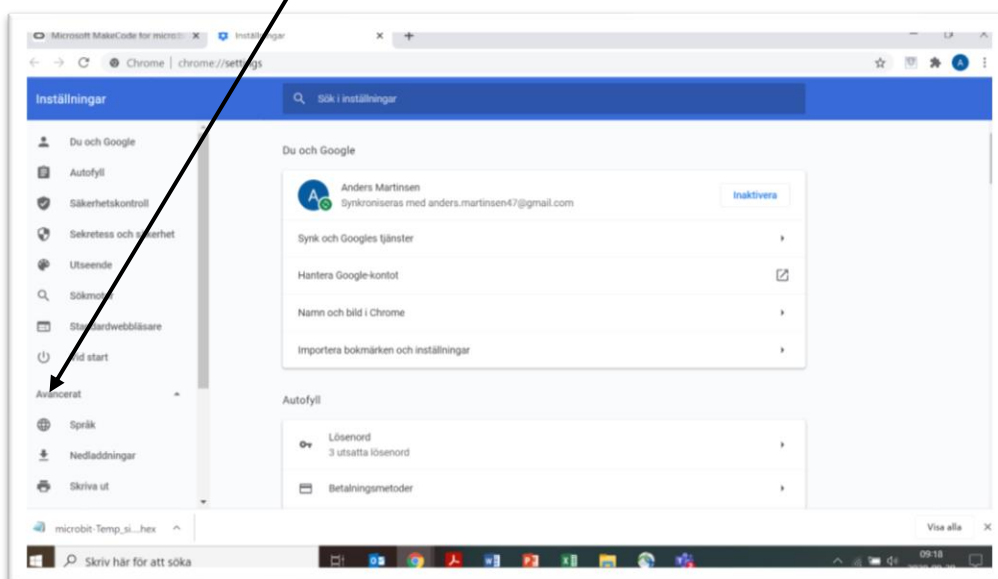
För att kunna göra detta i fallet med *webbläsaren Chrome* behövs det i så fall göras en justering av var HEX-filen ska sparas. Här följer en beskrivning hur man ställer in detta (Chrome).

Klicka på de tre punkterna (Uppdatera) uppe till höger på skärmen.

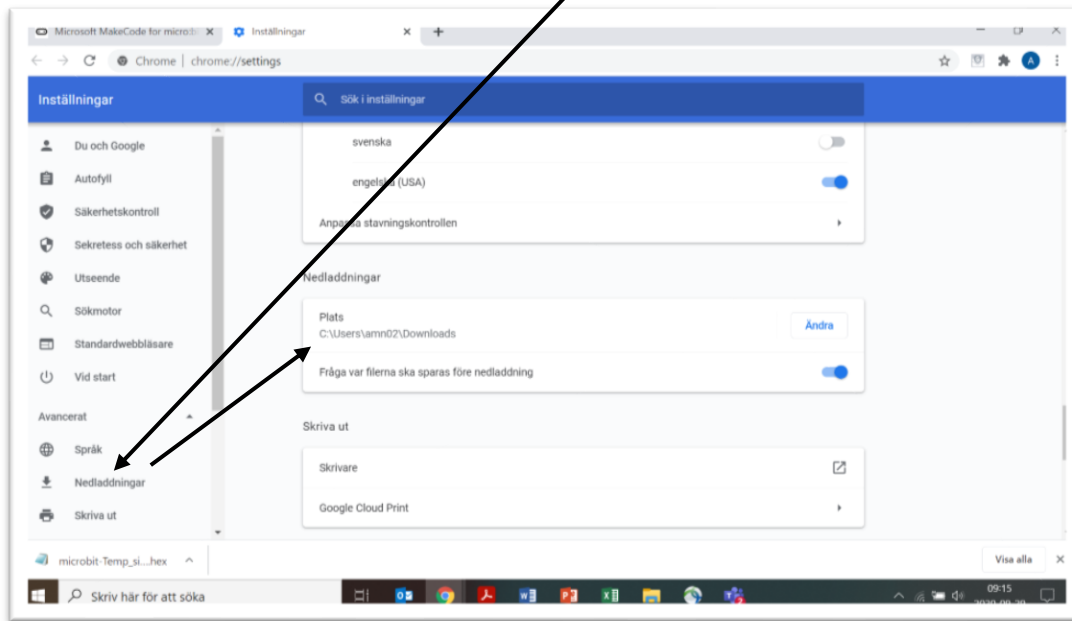


Välj raden **Inställningar** i rullisten

Ny skärmbild – välj **Avancerad**



Ytterligare nytt fönster där man skrollar ner till **Nedladdningar**



Under rubriken Nedladdningar kan man välja två alternativ;

1. att aktivera knappen vid frågan - "Fråga var filerna ska sparas före nedladdning" – för att i sparafönstret sedan välja enhet/plats att spara HEX-filen på, exempelvis D: Microbit.
2. eller om du vill att det ska gå ännu snabbare och slippa denna fråga, ändra *Platsen för nedladdningen* till D: Microbit, dvs enheten D.

**OBS! Glöm inte** att ev. återställa datorn för Hämtade filer om datorn är lånad och ditt pass är klart

#### Kommentar:

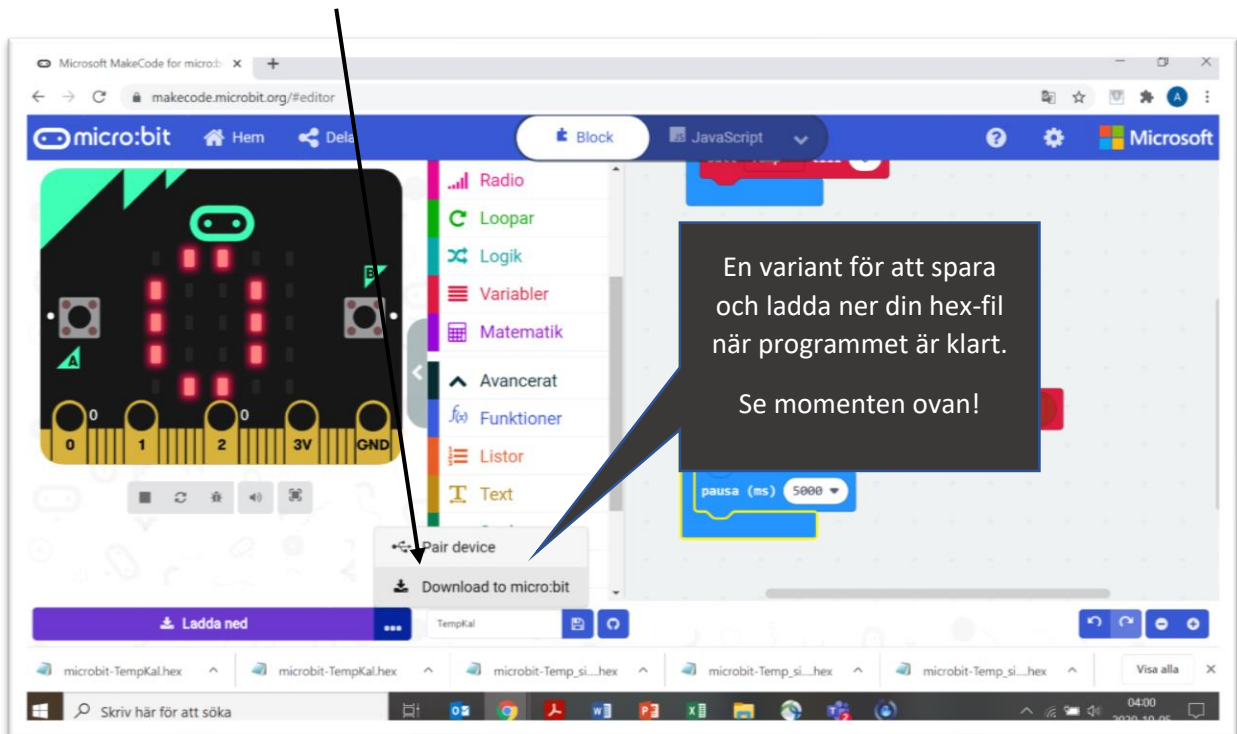
Här finns det (vad vi förstår) i flera fall en konflikt med vad man kan/får göra med datorer som ägs av skolan. Är det "min egen" dator är alternativet 1 ovan ett bra alternativ då man enkelt kan spara versioner i en separat katalog innan eller efter att de testats/laddats ner och då sparas med ett lämpligt filnamn.

Finns det någon försämring med att ändra? ... ja, de tidigare versionerna av HEX-fil försvinner, de finns inte kvar i exempelvis katalogen Hämtade filer (*Downloads*). Har man gjort förändringar i programmet som man ångrar finns den tidigare versionen inte kvar om man inte har sparat den först. Tänk igenom det här, med denna metod sparas inte hex-filen på elevens egna dator/moln och det kan ju vara en nackdel om man ska öppna ett sparat program vid en senare lektion.

Vilken version av Micro:bit som används har ingen betydelse när det gäller dator. När det gäller överföring med surfplatta har versionen av *Firmware* betydelse. Vid fel meddelas det i samband med överföringen i så fall.

Det som beskrivits i bild och text i ovanstående två sidor kan då användas ihop med den nya knapp med tre prickar som finns i den senaste versionen av utvecklingsmiljön nedan.

**Prova och testa några gånger** för att få in rutin på detta.




## Firmware

Det finns en programvara som "inte syns", men som används i bakgrunden vid användandet av systemet Micro:bit. Den programvaran kallas *Firmware*, ett slags enklare operativsystem, som hjälper till med kopplingen/kommunikationen mellan applikationsprogram och hårdvara. I bilden nedan ses en sammanställning av olika versioner av *Firmware*. För er som arbetar med surfplatta och nerladdning av program via Bluetooth har versionen av *Firmware* viss betydelse.

### Olika versioner av

- Micro:bit
- Firmware

Den/De version(er) av Micro:bit som delats ut av Smartare elektroniksystem är i huvudsak v 1.5, men version 1.3B kan förekomma.

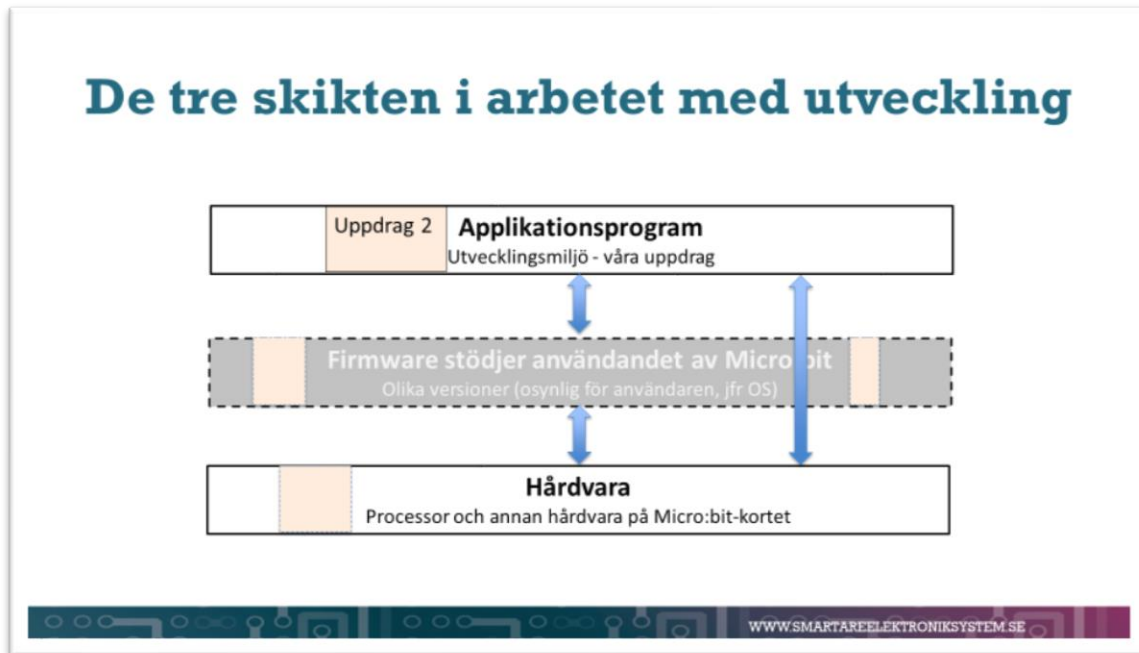


Board revision	Bootloader	Interface	Download
1.3	0234	0234	<a href="#">0234</a>
1.3b	0234	0241	<a href="#">0241</a>
1.5	0243	0249	<a href="#">0249</a>
2.0	0255	0255	<a href="#">0255</a>

WWW.SMARTAREELEKTRONIKSYSTEM.SE

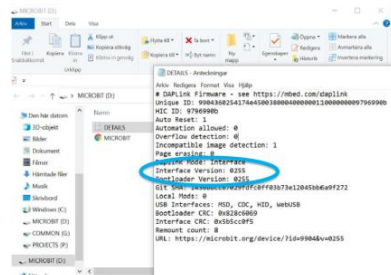
## Lite fakta kring Firmware

Kopplingen, kommunikationen, mellan applikationsprogrammet som utvecklas i varje uppdrag och hårdvaran sköts av skiktet *Firmware* i bilden, se bilden nedan. Den programvaran finns sparad i en av de integrerade kretsarna på microbitkortet och kan bytas ut då Micro:bit är i sk underhållsmod. Normalt tänker man inte på den programvaran, ”- det ska bara fungera!” och är osynlig i det normala arbetet. För er som arbetar med surfplattor kan det bli aktuellt att byta version av *Firmware*. Det kan bara göras med dator då Micro:bit är ansluten med en USB-sladd.



## Version av Firmware hos din Micro:bit

- Öppna utforskaren
- Klicka på enheten MICROBIT (D)
- TVÅ filer dyker upp
- Klicka på DETAILS
- I info-texten i DETAILS står den version av Firmware som din Micro:bit har



Hur vet jag vilken version av *Firmware* Micro:bit-kortet har?

Se bild med anvisningar till vänster!

WWW.SMARTAREELEKTRONIKSYSTEM.SE

<https://makecode.microbit.org/device/usb/webusb/troubleshoot> är en länk till hur det går till att uppdatera *Firmware*.

## Uppdatera Firmware



- Hur man uppdaterar Firmware  
- se länken <https://makecode.microbit.org/device/usb/webusb/troubleshoot>

WWW.SMARTAREELEKTRONIKSYSTEM.SE

## Praktiska tips och trix

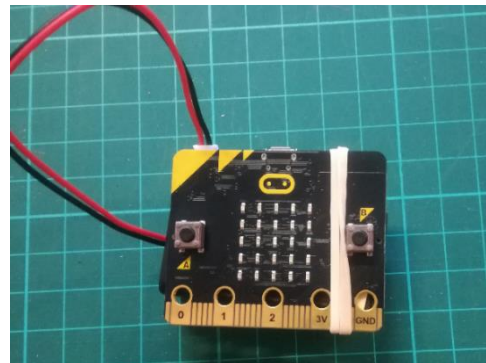
Under detta kapitel samlas en del praktiska tips i hanteringen av utrustningen kring micro:bit. Tips som kommit in och kan spara mycket tid och minska störningarna i klassrummet. Har du några fler – hör gärna av dig i så fall, vi fyller på!

### Före lektion

- Första gången en **ny** micro:bit kopplas till dator via USB ska det göras kalibrering. Kan vara bra att ha det gjort innan lektion med elever då det kan ta lång tid för alla elever att genomföra.
- Ladda gärna ner ett program till varje micro:bit innan första lektion. Då ser eleverna att de kopplat in batterier rätt.

### Under lektion

- Ta med gummisnoddar som eleven kan linda runt micro:bit och batterihållaren när övningar ska göras där eleverna ska röra sig i rummet med micro:bit. Detta för att undvika att kablar går av, batterier trillar ur hållare osv.



### Efter lektion

- När en elev satt in batterihållaren i micro:bit, låt dem inte dra ut den igen. Kablarna går väldigt lätt av. Då är det bättre att ta för vana att pocka ur batterierna ut batterihållaren och låta batterihållaren vara inkopplad när micro:bit läggs tillbaka i sin antistatiska påse.

## Micro:bit och surfplattor med trådlös överföring av program

- Ta med en dator och USB-sladd till micro:bit vid lektion. Om micro:bits programvara för Bluetooth inte är uppdaterad kommer en fil att behöva föras över till micro:bit. Detta kan bara göras via dator.
  - Den fil som ska laddas ner på micro:bit för ny programvara finns på: <https://microbit.org/guide/ble-ios/>
  - Se även länken på Youtube för mer information om hur det går till [https://www.youtube.com/watch?time\\_continue=65&v=k6OQ5XzIroY](https://www.youtube.com/watch?time_continue=65&v=k6OQ5XzIroY) .
- Om eleverna använder surfplattor. Märk micro:biten så att samma elev kan arbeta med samma micro:bit nästa gång. Detta spar tid gällande parkoppling.
- Om program laddas ner till micro:bit med USB kabel kommer den tidigare parkopplingarna till surfplattor att "glömmas" av micro:biten och elever behöver ta bort gamla kopplingar och parkoppla igen.

## Support – vad finns på nätet?

- Massor – men ett par länkar att börja med för att komma vidare kan vara
  - <https://support.microbit.org/support/home>
  - <https://microbit.org/guide/features/>

## Användbara länkar med information

### Allmänt

#### *Gör rent din Micro:bit*

Med tanke på smittspridning finns information på länken nedan om rengöring av utrustning. Viktigt om utrustningen används av flera olika personer samma dag!

<https://support.microbit.org/support/solutions/articles/19000112333-cleaning-the-micro-bit>

### Tekniska aspekter och möjligheter

Några användbara länkar för dig som lärare som kommit lite längre med Micro:bit eller som självstudier för duktiga elever.

#### *Firmware*

Programvaruversionen av det styrprogram (Firmware - ett slags operativsystem) som finns och hjälper till med att använda Micro:bit och dess hårdvara.

<https://microbit.org/get-started/user-guide/firmware/>

#### *Hårdvaran på Micro:bit*

En länk till en användarmanual för de olika enheterna på micro:bit-kortet.

<https://microbit.org/get-started/user-guide/features-in-depth/#processor>

#### *Specifikation av hårdvara*

En länk till djupare teknisk information av de olika enheterna på micro:bit.

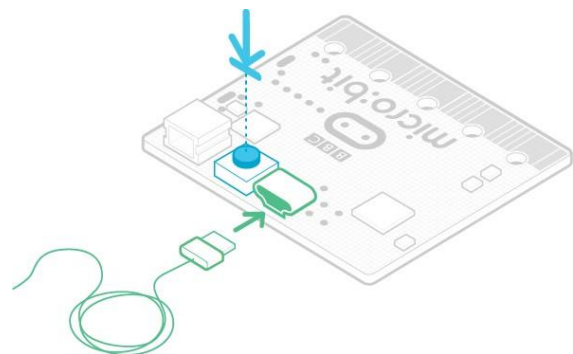
<https://tech.microbit.org/hardware/#nrf51-application-processor>

## Micro:bit och nedladdning av program

Kan något gå fel – en felkälla?

Det finns en resetknapp på baksidan av micro:bit-kortet.

- Kanske du eller eleven råkat trycka in resetknappen samtidigt som micro:bit ansluts till datorns USB?
- Micro:bit hamnar då i sk underhållsmod – *Maintenance mode*
- Inget applikationsprogram (HEX-fil) kan då laddas ner i micro:bitens Flash-minne.
- **Åtgärd:** Dra ur kabeln och anslut den igen **utan att trycka på Reset-knappen** i detta moment.



## Strömförsörjning

När man köper in Micro:bit finns alternativet att få med två batterier och en batterihållare. Hanteringen av batterier är ett litet praktiskt problem i sig i lektionssituationen och kanske du som lärare vill ha ett alternativ till. Om utrustningen ska kopplas bort från datorns USB-sladd och vara mobil är batterier eller möjligen en Powerbank alternativet.

Vid inköp av Micro:bit, fundera också på alternativet att få med en längre USB-kabel, den bör vara minst 30 cm för att överföring av program till vår utrustning ska fungera praktiskt och smidigt.



I bilden nedan skiljer vi på strömförsörjning och systemspänning – läs och fundera!

## Strömförsörjning och systemspänning - olika alternativ och begrepp

**Strömförsörjning** av Micro:bit-kortet

- USB-uttag på dator
  - Ger ca 5,0 V till Micro:bit
- Powerbank
  - Ger ca 5,0 V till Micro:bit då den är fulladdad. Har olika kapacitet i enheten mAh.
- AAA-batterier (alkaliska)
  - Två stycken nya ger 3,0 – 3,1 V och är det som levereras vid köp av Micro:bit.
- AAA-batterier (uppladdningsbara)
  - Finns olika fabrikat, exempelvis de från Kjell & Co ger 1,2 V per batteri. Då räcker det inte med två batterier. Fungerar kanske med tre eller fyra batterier med en batterihållare för detta.

**OBS! SE UPP** – inte att rekommendera då det finns andra uppladdningsbara batterier med högre spänning!

**Systemspänning** för elektroniken på Micro:bit samt annan inkopplad extern utrustning

- Strömförsörjningens spänning omvandlas till en systemspänning för processor och annan elektronik som finns på Micro:bit-kortet.
- **Systemspänningen är ca 3 V** och är den spänning vi använder oss av för våra externa enheter i temat Den Smartare staden.
- Andra elektronikenheter från exempelvis Kitronik kan kräva en högre systemspänning, men då har enheten en separat strömförsörjning för detta.
- Ett exempel och undantag i vårt fall är servomotorn till vägbommen som har tre AAA-batterier, dvs ca 4,5 V. Det beror på att servomotorn fordrar en större kapacitet hos strömförsörjningen för att fungera tillfredställande.

WWW.SMARTAREELEKTRONIKSYSTEM.SE

## ESD – en introduktion

ESD är ett viktigt fenomen att ta hänsyn till vid såväl tillverkning som användning av elektronik på exempelvis ett kretskort. ESD betyder **E**lectro**S**tatic **D**ischarge. Det innebär att man som användare vid experiment med elektronik kan ladda ur statisk elektricitet via elektronkort eller motsvarande och därmed skada eller förstöra en komponent.

Micro:bit och övrig elektronik är konstruerat för säker användning men ska hanteras med försiktighet i vissa avseenden. Det finns inget omgivande skydd, här några punkter att beakta:

### ESD

- Det är viktigt att undvika statisk elektricitet.
- Förvara alltid din micro:bit med dess övriga elektronikutrustning i antistatiska påsar när de inte används.
- Tag om möjligt i ett metallföremål som är anslutet till jord innan du börjar med de praktiska uppgifterna i temat. Det kan exempelvis vara en diskbänk av metall, ett rör eller ett element av metall som ingår i fastighetens värmesystem. Som lärare, diskutera med eleverna vad som kan vara ett lämpligt ställe att ta i.



Komponenter som skadats av ESD går oftast inte sönder helt, utan bara delvis. De kan få förändrade egenskaper och få förkortad livslängd. Detta kan vara orsaken till att oförklarliga fel uppstår i en elektronisk utrustning

### ESD-påsar

När utrustningen inte används ska den ligga i påsar av ESD-typ, exempelvis som påsen i bilden till höger. Om Micro:bit-kortet är monterat på respektive elektroniktillbehör är detta extra viktigt.

Micro:bit-kortet innehåller en typ av elektronik som är extra känslig för statisk elektricitet. Det handlar om de integrerade digitala kretsarna och att dras ingångar kan liknas vid en kondensator.

Att dessa laddas upp i kombination med de små avstånden som är mellan ledare i kretsen gör att ett överslag, en urladdning, kan ske redan vid ett par hundra volt. Det kan skada kretsen och därmed äventyra funktionen över tid.

**Anm:** Längst erfarenhet av utrustningen i systemet har vi av Micro:bit-kortet, dels innan våra teman startade, dels från undervisningen i temat Smartare Produkter. Till dags datum 2021-01-30 har vi inte fått in några rapporter om några trasiga Micro:bit-kort p g a ESD eller annan "elektrisk" påverkan. Det betyder inte att ni som användare ska bortse från rekommendationerna utan ta dessa rutiner på allvar och till en vana!

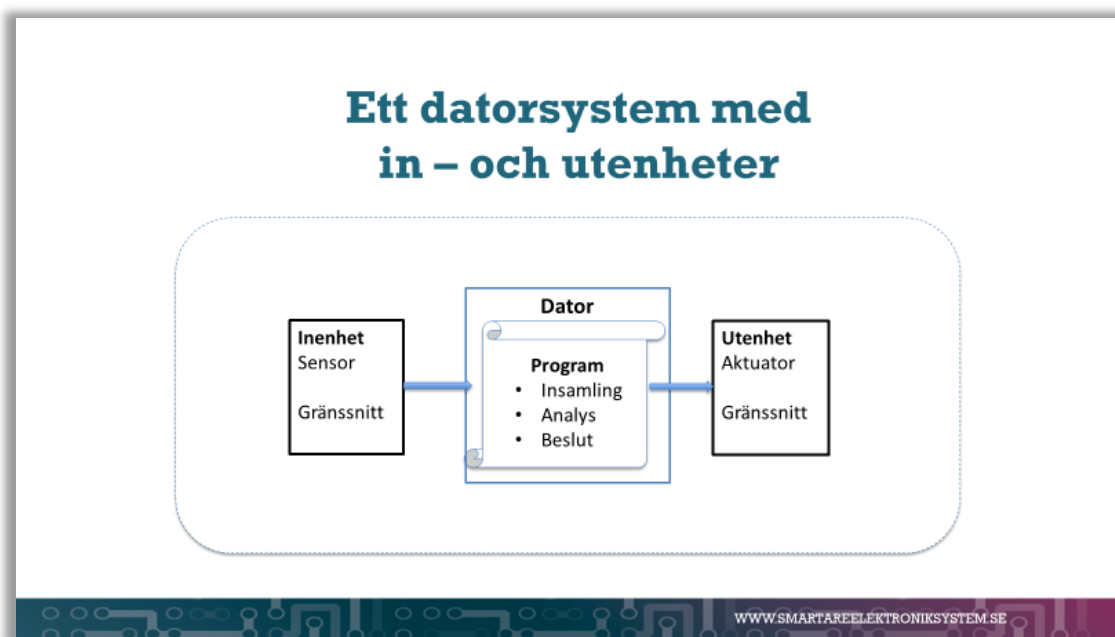


## Uppdragen i temat med fakta och kommentarer

Faktadokumentet kompletterar uppdragen i Den smartare staden (DSS) med mer information och kunskap kring utrustningen och dess programmering. Kunskap "mellan raderna" och viss fördjupning som kan komma till nytta i undervisningen i syfte att öka lärarens kompetens kring temat.

Till dokumentet finns även powerpointbilder från temautbildningarna, bilder som kan användas fritt för Temautbildare i den egna undervisningen.

I uppdragen kommer vi att arbeta med en utrustning som innefattas i de tre huvudenheterna i ett datorsystem, ett inbyggt system, med hård- och mjukvara. Figuren nedan är en beskrivning som man i samtliga uppdrag kan identifiera en enhets eller komponents roll i systemet, allt för att öka förståelsen.

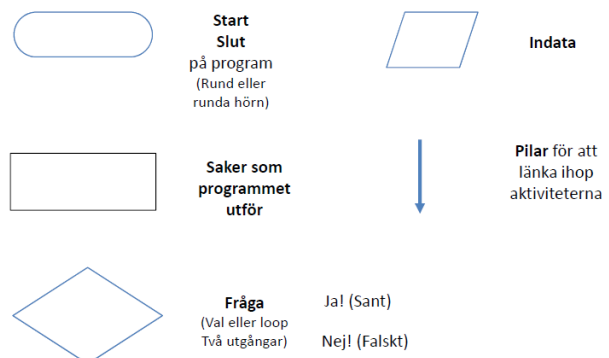


### Symboler flödesdiagram

När ett program planeras kan ett verktyg, en metod, som kallas flödesdiagram användas.

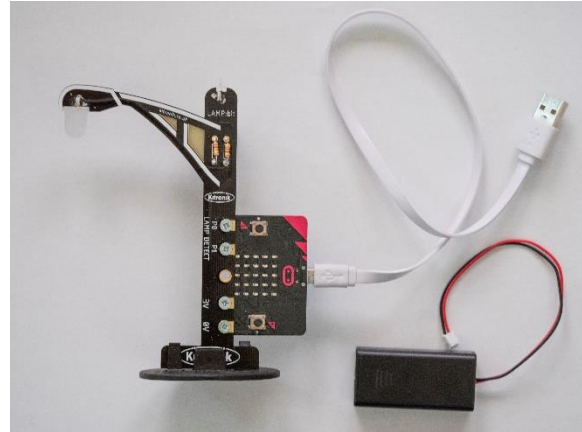
I bilden till höger ses några olika standardiserade symboler för detta.

I Den smartare staden används inte alla symbolerna i bilden. Om tid finns, öva gärna på dessa symboler i kommande uppdrag då program ska utvecklas eller analyseras.



## Uppdrag 2 – Smartare gatubelysning

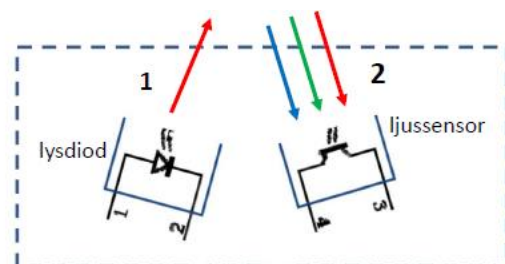
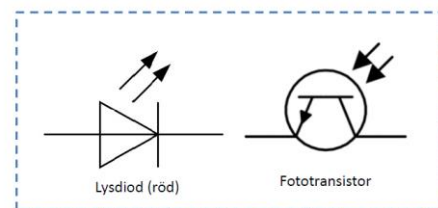
Inenhet i detta uppdrag är en ljussensor, antingen den som finns på micro:bit-kortet eller den externa som sitter högst upp på gatubelysningens stolpe.



Ljussensor på micro:bit-kortet.

**Fakta:** Varje lysdiod i Micro:bitkortets display innehåller två elektronikkomponenter, en lysdiod och en fototransistor. Det betyder att displayen även är sensor, och när funktionen **ljusnivå** anropas i programmet fås ett medelvärde som svar från dessa 25 sensorer.

Det sitter alltså 25 sådana kombinationer i lysdiodmatrisen. I matrisen nedan är två inkopierade.



**Fråga:** Vad består egentligen varje lysdiod av i displayen hos Micro:bit?

**Svar:** Varje "lysdiod" är egentligen två komponenter, den har två funktioner, en lysdiod och en fototransistor. Se bilderna ovan till höger. En diod som kan lysa och en fototransistor som kan mäta infallande ljus.

## Programmering med den interna ljussensorn.

Se programavsnittet till höger där den interna **funktionen ljusnivå** anropas och returnerar då ett värde. Variabeln vi kallar Ljus tilldelas detta värde som skrivs ut på displayen. Se bilderna ovan för fakta om displayens sensorer.

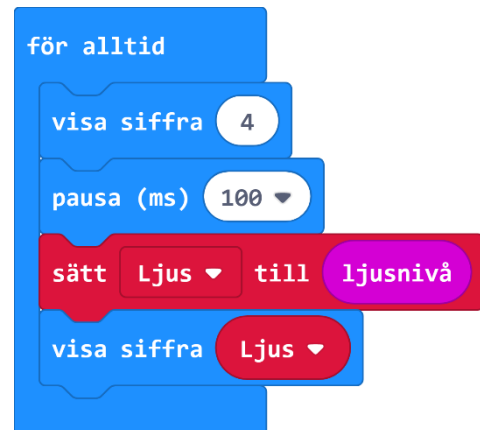
Funktionen **ljusnivå** returnerar ett värde i intervallet 0 – 255, vilket betyder att omvandlingen skett till ett åttabitsars binärt tal (en byte).

### Frågor till programexemplet:

- Var i programmet, i vilka block, används lysdioderna?
- Var i programmet, i vilka block, används fototransistorerna?

### Svar:

- Blocken **visa siffra 4** och **visa siffra Ljus**  
(Respektive block tänds lysdioder för att visa siffran 4 samt värdet som variabeln Ljus har)
- Blocket **sätt Ljus till ljusnivå**  
(Funktionen ljusnivå returnerar ett medelvärde från de 25 sensorelementen, dvs fototransistorerna).




### Sammanfattningsvis

## Inenhet – den inbyggda ljussensorn på Micro:bit


### Ljutmätning med Micro:bit

- Mäter med 25 sensorer
- Funktionen **ljusnivå** används bland **Inputblocken**
- ljusnivå** anger medelvärdet från de 25 sensorerna
- Mätvärdet** kan **tilldelas** en variabel
- Sensorn är känslig för skuggor i och med att matrisen har en stor yta
- 8 bitars talvärde (0-255)

### Egenskapstest



**Test:** Håll för ett antal lysdioder så minskar mätvärdet vid samma ljusstyrka

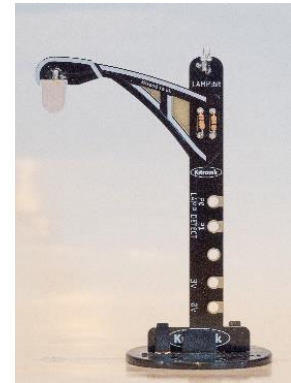


WWW.SMARTAREELEKTRONIKSYSTEM.SE

## Extern ljussensor

När extern elektronik kopplas in till micro:bitkortet kommer vi in på att använda blockinstruktioner i programmet som hanterar extern elektronik. Den externa ljussensorn sitter placerad högst upp på gatubelysningsstolpen. Signalen från den externa sensorn är en analog signal och är kopplad till pinne P1, vilken då ska vara en ingång.

Se grundprogram i bilden nedan. Programvarumässigt innebär blocket **analog läs pin P1** används för att få in det analoga värdet i programmet. **En variabel**, vi har kallat den **Ljus**, tilldelas detta värde i programmet. Blocket **analog läs pin P1** finns under **Avancerat i gruppen Pins**.



Blocket **analog läs pin P1** ställer "automatiskt" in pin P1 så att den blir en ingång, **en analog ingång!**

När ingången läses med blocket **analog läs pin P1** returneras ett tal mellan 0 – 1023 beroende på att inenheten som används hos micro:biten är en omvandlare, en AD-omvandlare (Analog till Digital omvandlare), och den använder 10 bitar för sitt talområde.

## Micro:bit med en extern ljussensor

### - programexempel

- Ett stift (en pin) i gränssnittet på Micro:bit kan vara en ingång eller en utgång
  - Vad bestämmer det?
- Gränssnittet kan också vara
  - Analogt
  - Digitalt
- I programexemplet
  - P1 är en **analog ingång**
  - en ljussensor ger data in
  - variabeln **Ljus** tilldelas detta värde
  - P0 är en **digital utgång**
  - en LED-lampa är kopplad som belysning



```

on start
  set Ljus to 0
  digital write pin P0 to 0

forever
  set Ljus to analog read pin P1
  if Ljus < 70 then
    digital write pin P0 to 1
  else
    digital write pin P0 to 0
          
```

WWW.SMARTARELEKTRONIKSYSTEM.SE

## Mer om hårdvaran i gatubelysning

Innan vi går in mer på programmeringen kommer här lite mer fakta om elektroniken, se första bilden nedan. I bilden nedan ses symbolen för en fototransistor, den komponent som sitter högst upp på belysningsstolpen. Fototransistorn är känslig för ljus, ett ökat ljus ger en ökad ström  $I$  genom transistorn. Strömmen går vidare genom motståndet  $R_1$ , vilket i sin tur ger en ökad spänning  $U = R_1 \times I$  enligt Ohms lag. Kopplingen med + 3 V och fototransistor i serie med resistorn  $R_1$  blir den elektriska kretsen och i detta fall gränssnittet mot micro:bit-kortet.

När spänningen  $U$  ansluts till microbitens pin P1 avläser programmet denna spänning med blocket **analog läs pin P1**, och resultatet blir ett binärt tal mellan 0 – 1023 i och med AD-omvandlaren.

### Gränssnitt extern hårdvara och Micro:bit

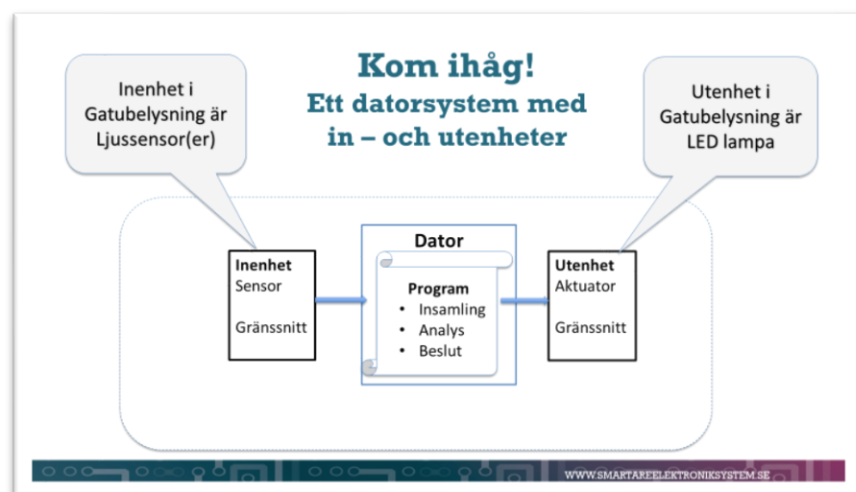
**Studera kretskortet för stolpen till gatubelysningen**

- Ljussensorn är kopplad till pin P1, en fototransistor i serie med en resistor (ett motstånd.)
- Ljussensorn ger en analog signal in, en spänning som varierar mellan 0 och + 3 V beroende på ljusnivån.  $U=R_1 \times I$
- Spänningen omvandlas av Micro:bitens AD-omvandlare till ett tal mellan 0 – 1023, **analog read pin P1**
- LED-lampan (utgången) tänds om **villkor=sant** med en "1:a" på pin P0, **digital write pin P0 ...**

WWW.SMARTAREELEKTRONIKSYSTEM.SE

**Sammanfattningsvis:** Fototransistorn är en sensor, och med resistorn  $R_1$  (10 k  $\Omega$ ) fås ett användbart elektriskt gränssnitt – en spänning till Micro:bit. Med instruktionen (blocket) **analog läs pin P1** aktiveras AD-omvandlaren som finns för ingången P1 och vi får ett värde på ljuset. Värdet tilldelas variabeln **Ljus** i programmet på föregående sida.

Vi vill också betona principen för de system vi arbetar med i temat, att det i varje uppdrag går att identifiera de tre enheterna i bilden, Inenhet – Datorenhet – Utenhet.



## Komponentfakta

Fototransistorn i vårt fall har beteckningen LTR-4206, se ett utdrag ut datablad nedan.

Allmänt

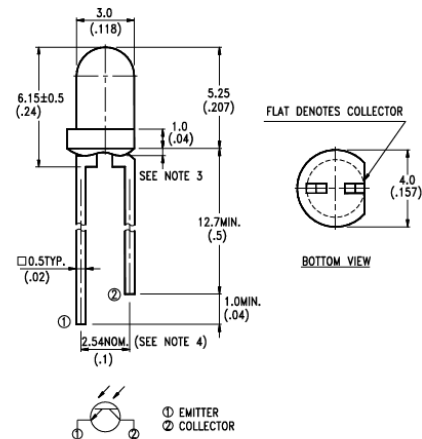
### Features

- Wide range of collector currents.
- Lens for high sensitivity.
- Low cost plastic package.

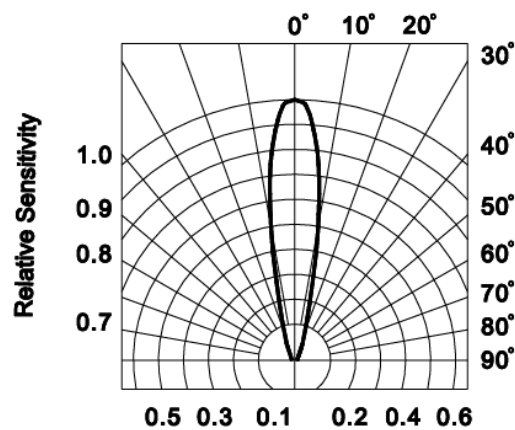
### Description

The LTR-4206 series consist of a NPN silicon phototransistor mounted in a lensed, clear plastic, end looking package. The lensing effect of the package allows an acceptance half angle of  $10^\circ$  measured from the optical axis to the half power point. This series is mechanically and spectrally matched to the LTE-4206 series of infrared emitting diodes. The LTR-4206E is a special dark plastic package that cut the visible light and suitable for the detectors of infrared application.

### Package Dimensions



### Känslighet



**FIG.5 SENSITIVITY DIAGRAM**

Det finns flera olika typer av fototransistorer på marknaden och de är känsliga för olika våglängder eller våglängdsområden hos ljuset. I vårt fall har fototransistorn kalibrerats för synligt ljus och passar för vår applikation med gatubelysning, men kan exempelvis också användas för nattlampor i bostäder.

I kapseln på LTR-4206 sitter en lins för att få ökad känslighet hos sensorn och inkapslingen i plast gör att den blir robust och tål en större yttre mekanisk påverkan.

## Talsystem

I temat Smartare produkter träffade vi på hur inenheter ger olika stora tal till Micro:bit vid programmeringen. Detta blir ännu mer aktuellt i detta tema!

- Ljussensorn på Micro:bit ger ett tal mellan 0 – 255 på ljuset
- Mikrofonen på Micro:bit v2 ger ett tal mellan 0 – 255 på ljudet
- Externa ljussensorn ger ett tal mellan 0 – 1023 på ljuset
- Accelerometer ger ett tal mellan 0 – 1023 på accelerationen
- Temperatursensorn ger ett tal mellan 0 – 1023 på temperaturen

Talen har sin grund i det binära talsystemet, ett talsystem som används av datorer vid beräkningar. Det är dags att veta mer om detta, speciellt i vårt tema Den smartare staden!

### Decimala talsystemet

Några fakta att utgå från är att decimala talsystemet har Basen 10. Exempelvis:

- Talet  $736_{10}$  som betyder  $7 \cdot 100 + 3 \cdot 10 + 6 \cdot 1$ , eller  $7 \cdot 10^2 + 3 \cdot 10^1 + 6 \cdot 10^0$  om vi analyserar med potenser och basen 10.
- Tillåtna tecken i varje position är 0 – 9, med tre positioner kan talen 0 ---  $999_{10}$  skrivas och med fyra positioner 0 ---  $9999_{10}$
- Exempel med fyra siffror  $8437_{10}$ 
  - Mest signifikant siffra är 8 ( $8 \cdot 10^3$ ) och Minst signifikant siffra är 7 ( $7 \cdot 10^0$ )

### Binära talsystemet

Nu är Basen 2. Exempelvis:

- Talet  $1101_2$  som betyder  $1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$  (=1310), eller  $1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$  om vi analyserar mer.
- Tillåtna tecken i varje position är 0 och 1, med tre positioner kan talen 000 ---  $111_2$ , ( $0 - 7_{10}$ ), skrivas och med fyra positioner 0000 ---  $1111_2$ , ( $0 - 15_{10}$ ).
  - ... osv, talen ovan består av 3 respektive 4 bitar
- Åtta bits (bitar) är en Byte
  - Med åtta bitar är talområdet 0 –  $255_{10}$ , exempelvis talet  $(1100\ 1110_2)$
  - **Mest Signifikant Bit (MSB)** är **1** ( $1 \cdot 2^7=128$ ) i detta exempel och **Minst Signifikant Bit (LSB)** är 0 ( $0 \cdot 2^0 = 0$ ), dvs sista 0:an till höger i talet
- De åtta positionerna i ett binärt tal betyder (har vikterna) 128 ... 64 ... 32 ... 16 ... 8 ... 4 ... 2 ... 1
- Hur gör jag om från decimalt till binärt?
  - En metod visas här!
  - Antag att talet 165 (decimalt) ska göras om till binärt tal.  
I talet 165 ryms **en 128**, 37 blir kvar alltså ingen 64 men **32 ryms**. Då blir det 5 kvar, dvs ingen 16, ingen 8 men **4 ryms**, sedan ingen 2 men **1 ryms**.
  - Det betyder 1010 0101 binärt om man går igenom tallistan ovan.  
Med ett mellanrum mellan biten för 16 och 8 blir det lite tydligare att läsa talet.
- Talområdet 0 – 1023 innebär att 10 bitar används.

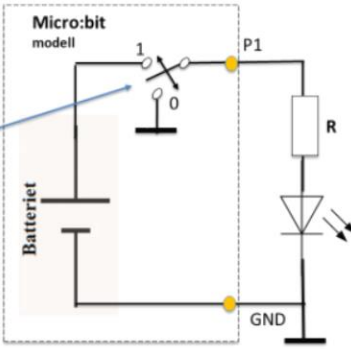
## Digital signal

I grundprogrammet för gatubelysningen finns blockinstruktionen **digital skriv pin P0 till 1**. Med detta block blir pin P0 en digital utgång och här finns bara två lägen, ett eller noll. En etta tänd lampen och en nolla släcker lampen. Blocket finns också här under blockkategorin **Avancerat/Pins**. Lampan är utenheten, den aktuator som aktiveras med ett eller noll på pin P0. Resistorn R i kopplingen är till för att skydda lysdioden mot för stor ström.

### Digital signal på utgång

- Den elektriska kretsen som illustrerar en digital signal
  - En "etta" (1)
  - En "nolla" (0)
- Mjukvarustyrd switch, med minne
 

digitalt skriv pin P1 till 1
- I vårt fall
  - En etta är + 3V
  - En nolla är 0V



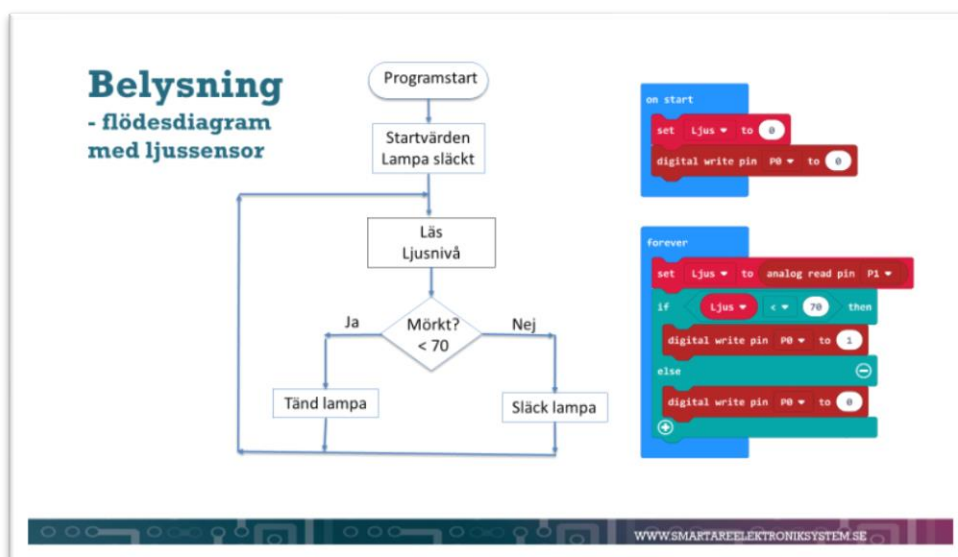
WWW.SMARTAREELEKTRONIKSYSTEM.SE

## Grundprogram för belysning med flödesschema

Grundprogrammet i detta uppdrag kan beskrivas i flödesschemat i figur nedan.

**Fråga:** Finns det någon svaghet i programmet?

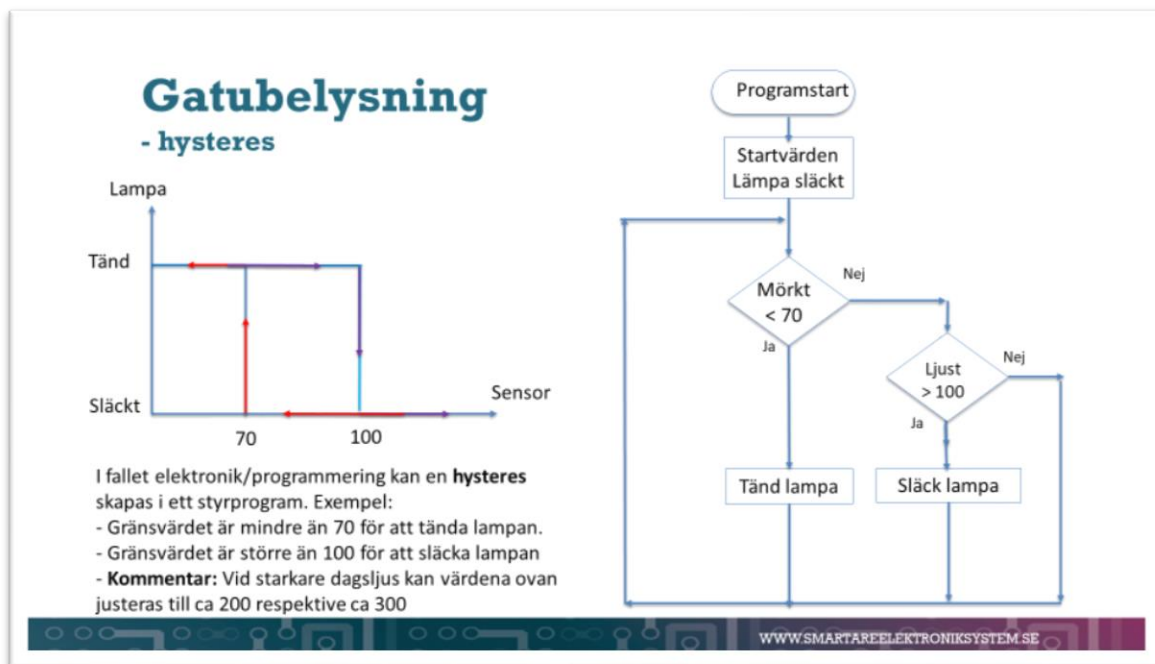
**Svar:** Ja, sensorvärdet kan hamna precis vid gränsen 70, vilket kan innebära att lampan kan börja att blinka. Det är då lätt att tro att det är fel på lampan, vilket inte är fallet.



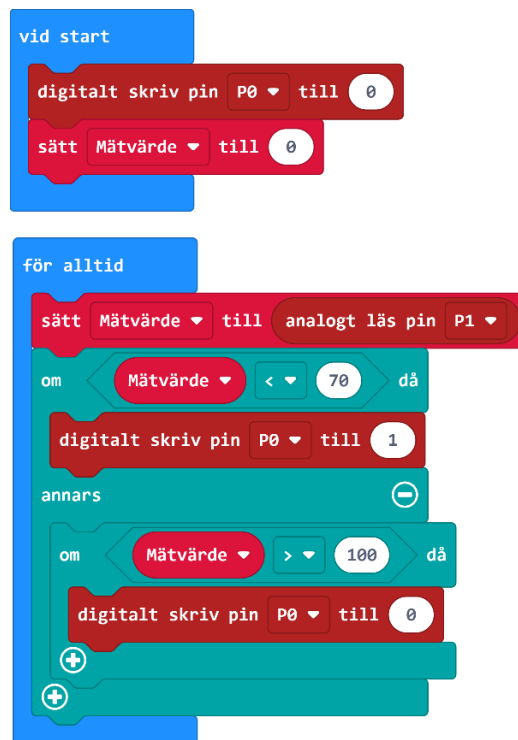
**Tips!** Prova med att hålla handen över sensorn för att få värdet kring 70 och demonstrera detta "fenomen".

## Hysteres

För att lösa problemet med **ett** omslagvärde kan man i styrprogrammet införa en sk hysteres i omslaget mellan ljust och mörkt. Se diagram tillsammans med flödesschemat i bilden nedan. Här tänds lampan om värdet är mindre än 70, men släcks först om värdet blir över 100. Hysteres är ett mycket vanligt sätt att få en funktion att bli bättre (stabilare) i ett styrprogram. I detta fall kommer lampan att sluta blinka vid ett värde som är nära gränsvärdet för tändning.



Programmet till flödesschemat med hysteres blir då enligt blockkoden nedan. Prova gärna med "olika stor hysteres" för att få en bra funktion!



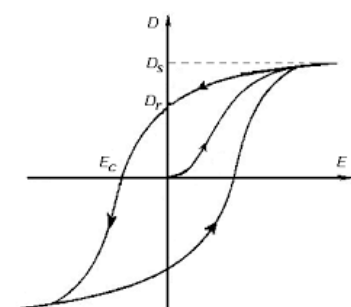
**Fråga:** Kanske en fråga, en diskussionspunkt, att ta upp med eleverna. Finns det fler applikationer där en hysteres kan tillämpas i ett vardagligt sammanhang?

I programmet nedan har hysteresfunktionen utvecklats ytterligare, med ett större spann i hysteresen OCH mätvärdet skrivs ut mellan mätningarna för att testa och visa på funktionen. En paus kan också läggas in för att mätningen med utskrift inte ska komma så tätt inpå varandra.



### Hysteres vid magnetisering

En bild på en hysteresfunktion i ett annat sammanhang, nämligen vid magnetisering av järn. Kurvan till höger visar detta. Om man magnetiserar en järnbit till "Nordpol" för att sedan ommagnetisera järnbiten till "Sydpol" och tillbaka igen krävs en "kraft" som ger olika omslagspunkter i den sekvensen. Det kallas som sagt hysteres och kurvans inneslutning är ett mått på energiförlusten i järnet i denna sekvens. Detta är viktigt i samband med motorer och transformatorer som man vill ska ha låga energiförluster. Då är det fråga om växelström (AC) genomlöps kurvan många ggr per sekund (50 Hz) och är anledningen till den adapter ni använder för laddning av en elektronikpryl blir varm, exempelvis laddaren för en mobiltelefon.

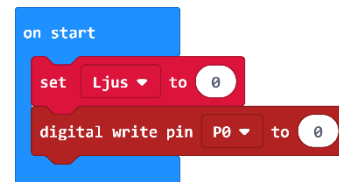


## Felsökning av gatubelysning (Lamp:bit)

**Problem:** Lampan lyser konstant när micro:biten har monterats.

### Felsökning - några punkter

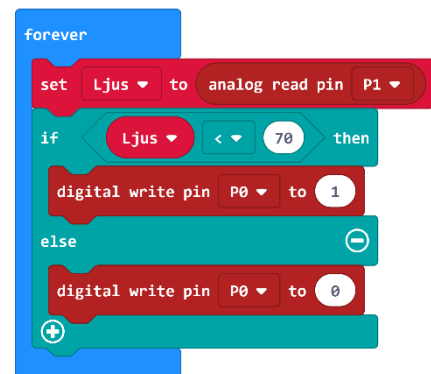
1. Micro:bit -kortet är felvänt. Ingen funktion mer än att lampan är släckt eller tänd permanent. Vi har testat detta, inget hände i vårt fall och inget gick sönder.
2. På vilken sida micro:bit sitter monterad på belysningsstolpen har ingen betydelse, den är identisk på båda sidor. MEN kanske det är bättre att micro:bit sitter på framsidan, se nedan.
3. Skruv/mutter sitter lite löst på ett par ställen, speciellt +3V och GND kan ge problem.
4. Skruv/mutter sitter för hårt åtdragna.
  - a. En lysdiod eller flera (trafikljuset) tänds utan förklaring.
  - b. Lossa lite på en skruv i taget och se om felet åtgärdas.
  - c. Förmodligen är det bättre att ha microbitkortet monterat på framsidan, det gäller för övrigt samtliga kort. Skruvskallen har en mindre diameter och bör inte kunna komma emot närliggande pinnar på microbit-kortet, vilket kan ge en felaktig funktion.
5. Programmet som vi använde i test och felsökning var det i fördjupningsdelen (elevblad) enligt bilden.



```

on start
  set Ljus to 0
  digital write pin P0 to 0
  
```

- a. Micro:bit är rätt monterad, men ett gammalt program gör att det finns en etta på pinne P0 som gör att lampan lyser konstant. Det var så vid vår testning. Ladda ner rätt program, ett enkelt testprogram.
- b. Ett "tänkbart fel" är att sensorvärdet i **if-satsen** ändrats och är satt alldeles för högt i programmet. Då är lampan tänd nästan hela tiden - bara solen kan släcka?!



```

forever
  set Ljus to analog read pin P1
  if Ljus < 70 then
    digital write pin P0 to 1
  else
    digital write pin P0 to 0
  
```

### Trasiga komponenter

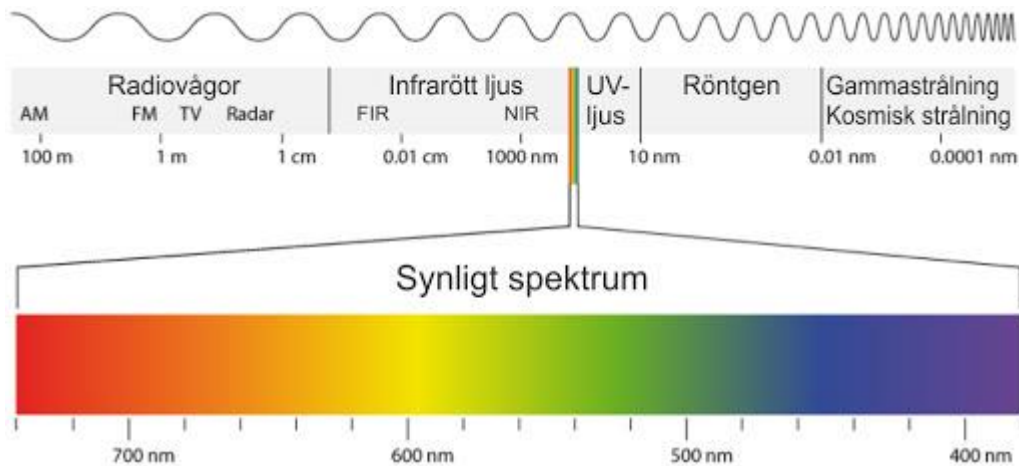
6. Ett annat tänkbart fel är att ljussensorn är trasig, två alternativ:
  - a. Ett avbrott, sensorn leder inte ström och ger då ett konstant lågt värde på ingång P1 (det blir nästan 0) oberoende av hur ljusst det är. Lampan kommer att vara tänd hela tiden med det program vi har.
  - b. Alternativt att sensorn kortsluter, och det blir konstant en etta på P1 från sensorn. Lampan kommer att vara släckt hela tiden med det program vi har ovan.
7. En trasig ljussensor kan vara att den fått en smäll och lossnat eller skadats vid montering eller användning.

**Anm:** Se även separata dokument för montering och test av utrustningen.

## Fördjupning – Fotonik

### Fysikaliska grunder

Synligt ljus är elektromagnetisk strålning liksom radio- och TV-signaler, mobiltelefonens signaler, röntgenapparaternas strålning på sjukhus osv. I bilden nedan ses det elektromagnetiska området indelat i de olika namngivna områdena vi har för olika tillämpningar.



Synliga ljusets våglängder	
Färg	Våglängd (nm)
 <b>Röd</b>	625–740
 <b>Orange</b>	590–625
 <b>Gul</b>	565–590
 <b>Grön</b>	520–565
 <b>Cyan *</b>	500–520
 <b>Blå *</b>	435–500
 <b>Violet</b>	380–435

Lysdioder (LED) finns för olika färger och har blivit mycket användbara då de förbrukar lite energi OCH kan styras och regleras med en elektronikenhet. Exempelvis i våra fordon används idag uteslutande lysdioder med olika färg, allt från strålkastare till indikering av olika funktioner i fordonet.

**En fråga till eleverna:** Vilka färger används i en bil till för olika typer av signaler?

## Övergången till teknik med nytta

Uppdragen med gatubelysning och trafikljus innehåller elektronikkomponenter med LED-teknik och ingår i ett teknikområde som kallas fotonik. Tillsammans med andra elektronikkomponenter som processorer etc. kommer vi också in på delområdet i fotoniken som kallas optonik (optoelektronik).

Med fotonik avses tekniken att generera och utnyttja ljus och annan strålningsenergi, att använda fotoner på samma sätt som elektroniken använder elektroner. Området är en av sex teknologier som av EU kommissionen pekats ut som en nyckelteknologi med mycket stor potential och bidra till ekonomisk tillväxt och jobbskapande.

- Ett exempel är inom belysning med LED-teknik. Nästan alla lampor är idag lysdioder, i hemmet, i trafiken etc. De kännetecknas av lång livslängd, hög verkningsgrad, låg energiförbrukning och kan styras med elektronik (optonik). I bilden ses en väst med lysdioder och den arbetande montören har båda händerna fria. Ett svenskt företag som utvecklat produkten, vilken även kan kopplas upp mot internet för styrning, se begreppet **IoT**.
- Ett annat exempel är infraröd teknik, IR-teknik, med värmekameror som ett specialområde. De kan användas för att detektera brister i en fastighets isolering mot kyla, förebyggande inspektion av elektriska ledningar för att upptäcka eventuella brister i kontaktpunkter eller i bilar för att upptäcka djur och människor i mörker.



- Ytterligare ett exempel på fotonik är laser vid materialbearbetning där mycket höga effekter används för att skära i plåt.



© Can Stock Photo

## Energimyndigheten testar

### Varannan LED-lampa lyser efter 10 000 timmar

Många LED-lampor har lång livslängd men det finns också lampor som slocknar i förtid. Det visar Energimyndighetens test av LED-lampor.

Testet är utfört enligt en provningsmetod där lamporna är tända hela tiden och där labbet gör kvalitets- och funktionstester var tusende timme.

Energimyndighetens Testlab har testat:

- hur mycket ljus lamporna ger.
- att de har förmåga att återge färger på föremål på ett rättvisande sätt
- att skenet från lamporna är fortsatt varmt eller kallt

Lampor av god kvalitet ska behålla sina egenskaper. För LED-lampor minskar ljusflödet med tiden. När ljusflödet gått ner under 70 procent av det som lampan hade från start räknas det som att lampan slocknat och slutat fungera.

De testade lamporna köptes in redan 2014 i olika EU-länder och några av lamporna fanns vid den tiden på den svenska marknaden.

– Ungefär hälften av de testade lamporna lyser fortfarande efter 10 000 timmar. Sedan testet startade har det kommit många nya LED-lampor på marknaden och erfarenheten är att kvaliteten ökar och priserna sjunker, berättar ansvarig för testet vid Energimyndighetens Testlab.

LED-lampor är känsliga för värme. Det är ofta kvaliteten på elektroniken i drivdonet som behövs för att lampan ska fungera och som avgör lampans livslängd.

Om ett år kommer det skärpta ekodesignkrav på LED-lampor och en ny uppdaterad energimärkning. Då kommer även en ny internationell provningsmetod för att mäta livslängd på lampor, en metod som Energimyndighetens Testlab har varit med att utveckla.

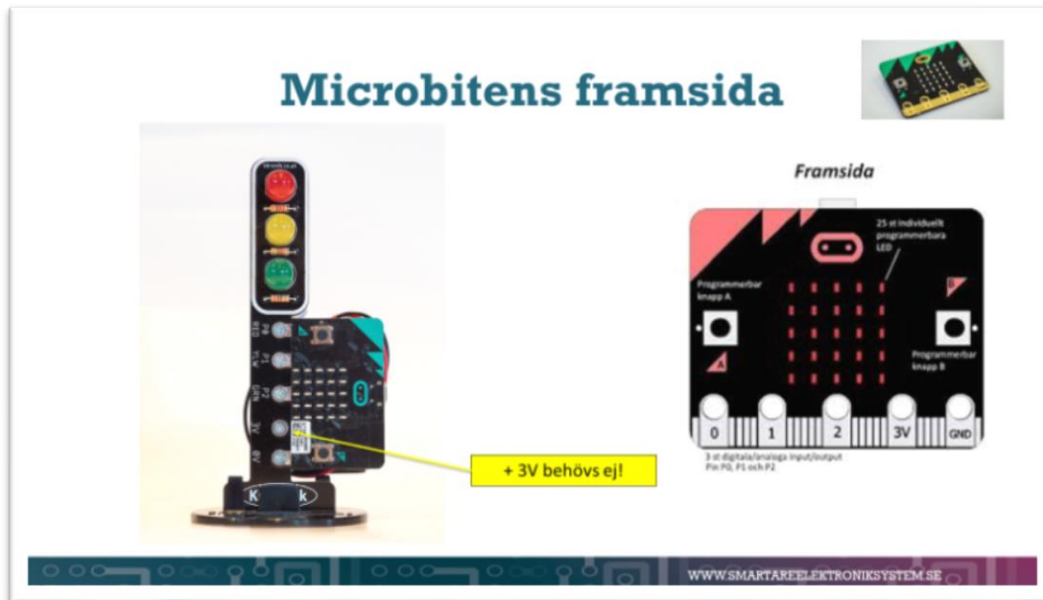
Det nya testet tar 3 600 timmar, det vill säga ungefär fem månader att utföra. I testet är lamporna tända i två och en halv timme och sedan får de svalna i 30 minuter innan de tänds på nytt. Max en av tio lampor får slockna under testet.

Källa: Elektroniktidningen den 25 september 2020.



## Uppdrag A – Smartare trafiksystem

Grundfunktionen i uppdraget följer ett traditionellt trafikljus, utan någon påverkan av sensorer eller andra inenheter. Lite senare i uppdraget kommer knapparna A och B att användas samt en trådlös signal från en annan Micro:bit. Utenhet är de tre lysdioderna, röd – gul – grön samt micro:bits display. Notera att +3V **inte** behöver kopplas in mellan trafikljus och micro:bit-kortet.



Först lite mer om en digital signal, den som tänds och släcker lysdioderna. Se bilden nedan.

Blocket **digital skriv pin P1 till 0 eller 1** i figur styr en "omkopplare" i micro:bit enligt bilden nedan. Med en etta tänds lysdioden då den elektriska kretsen sluts med den instruktionen.

### Elektronik/datateknik/digitalteknik

- Den elektriska kretsen som illustrerar digitala signaler
  - En "etta" (1) (sant)
  - En "nolla" (0) (falskt)

digitalt skriv pin P1 till 1

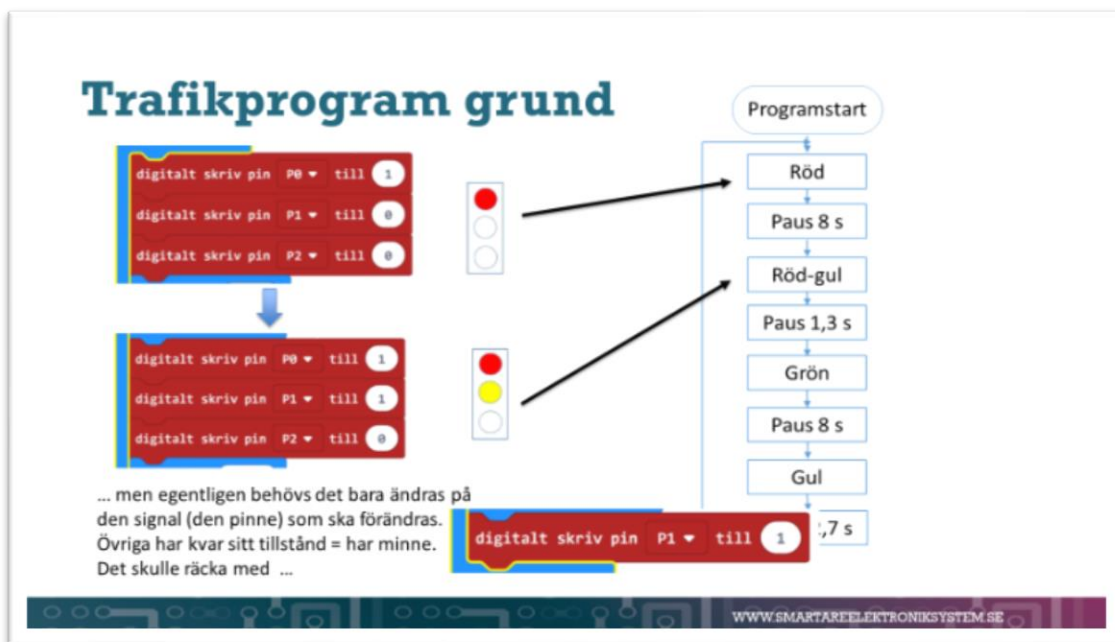
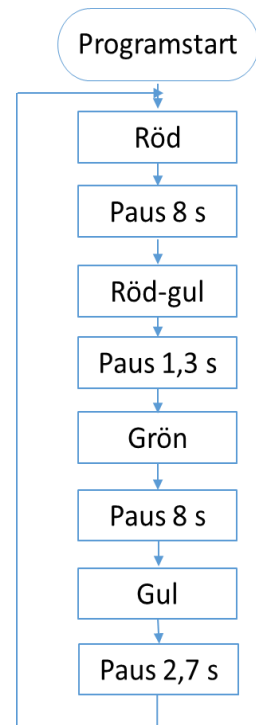
- I vårt fall
  - En etta är +3V
  - En nolla är 0V

WWW.SMARTAREELEKTRONIKSYSTEM.SE

## Grundprogram för trafikljuset

I flödesschemat ses det grundprogram vi utgår från i uppdraget. En ordinär funktion med tider för bilar i en trafik korsning. Med de digitala blockfunktionerna **digital skriv ...** tänds och släcks trafiksignalerna i den sekvens som flödesdiagrammet beskriver.

Varje signalkombination beskrivs med de tre blocken enligt figur nedan, men som visas i figur behövs det egentligen bara användas det block som ska ändra respektive lysdiod, exempelvis den gula i bilden nedan. Övriga pinnar har kvar sitt tillstånd, det finns alltså en minnesfunktion för varje pin så länge den inte ska ändra sitt värde. Detta är grunden till de olika minnen som finns i varje dator.



Vi skriver dock till samtliga *pins* i detta uppdrag av pedagogiska skäl!

## Händelser

När vi kommer vidare i Uppdrag A kommer trafikljusets sekvens och funktion att påverkas av att:

- en gångare vill gå över
- en kö uppstått så den riktningen ska ha längre grön tid
- en buss kommer till korsningen och ska prioriteras för grönt tidigare


De inenheter som används för att påverka trafikljusets funktion är Knapp A, Knapp B samt en radiosignal. Vi kallar det för händelse och när de används kallas vi det för händelsestyrning, dvs när respektive händelseenhet används påverkar det programmets grundsekvens vid korsningen. Blocken som används för respektive händelse ses i figur nedan.

### Händelsestyrning

- En händelse påverkar exekveringen av huvudloopen ( "för alltid" )

#### Prioriteringar vid trafikljuset

- Gångaren ska få grön gubbe!
- Ändra tiden för grönt vid kö!
- Prioritera en buss vid trafikljuset!



WWW.SMARTAREELEKTRONIKSYSTEM.SE

Nästa bild visar på ett par av dessa händelser och på vilket sätt de påverkar grundprogrammet i huvudloopen "för alltid".

### Vad händer då? - exempel

#### Grundprogram

```

graph TD
    Start([Programstart]) --> Röd[Röd]
    Röd --> Paus8[Paus 8 s]
    Paus8 --> RödGul[Röd-gul]
    RödGul --> Paus13[Paus 1,3 s]
    Paus13 --> Grön[Grön]
    Grön --> Paus8_2[Paus 8 s]
    Paus8_2 --> Gul[Gul]
    Gul --> Paus27[Paus 2,7 s]
    Paus27 --> Start
    
```



WWW.SMARTAREELEKTRONIKSYSTEM.SE

## Kitroniks tillägg

Till Kitroniks utrustning kommer det med ett antal färdiga funktioner för programmeringen, som exempelvis den för trafikljuset. Under Avancerad och valet Tillägg (1) kan du klicka och skriv sedan *Kitronik* i fältet vid (2). Sedan ny bild med bl a trafikljuset och välj enheten Stop:bit (3), vilket gör att dessa funktioner kommer med bland programblocken (4) i utvecklingsmiljön, se figur.



## Kitroniks funktioner

Ett exempel är de fyra funktionsvalen som finns i Kitroniks funktioner för våra signalkombinationer, se figur. Kanske en liten övning i engelska för att koppla respektive funktionsval till engelska begrepp?

## Lite mer om felsökning

Det kanske finns störningar i utrustningen, exempelvis i trafikljuset. I bilden nedan några synpunkter/tips på åtgärder. Se även tidigare avsnitt om felsökning under gatubelysning.

## Felsökning – tre kategorier

- Mekaniska / elektriska fel
  - Skruvar sitter
    - För löst
    - För hårt
- Fel i mjuk- eller hårdvara
  - Ha ett enkelt testprogram
    - Utesluta fel i hårdvaran
  - Simulera om det går
- Trasig komponent/kabel
  - Okulär besiktning av kretskort och anslutningar
  - Mäta med voltmeter/ohmmeter



[WWW.SMARTAREELEKTRONIKSYSTEM.SE](http://WWW.SMARTAREELEKTRONIKSYSTEM.SE)

**Uppgift testprogram:** Gör ett testprogram som i en sekvens tänder och släcker lysdioderna. Kanske något för en elev att göra om det finns tid över.

- alla samtidigt
- en i taget
- med knapp A tända en i taget i sekvens
- ...

**Anm:** Vad gäller testprogram finns ett särskilt dokument för detta, med programexempel!

## Radio

En funktion som kommer att användas i två av våra uppdrag är radiofunktionen. I trafikljuset gäller det en radiosignal för att kunna ge en buss förtur i en gatukorsning i fallet trafikljus.

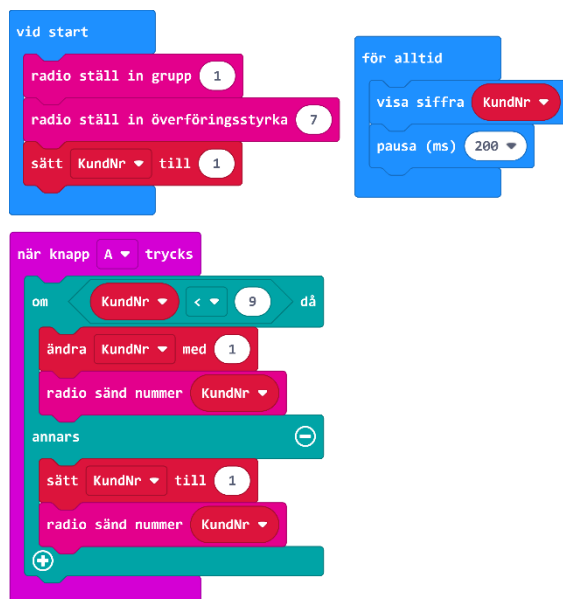
### Exempel

Nedan följer ett exempel med radio som tillämpning på könummer i ett apotek, en delikatesdisk eller ... . I exemplet skrivs kundnummret ut på två displayer, en som är direkt kopplad till apparaten som expediten hanterar (**sändaren**) samt en display som via radio får samma nummer (**mottagaren**).

Att genomföra exemplet kräver alltså två micro:bit.

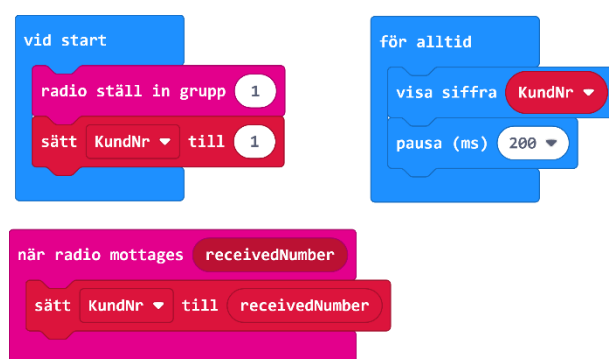
### Sändare med display

1. **Initiering** sändare (startrutin)
  - a. Ställ in radiogrupp (1 – 255)
  - b. Ställ in radiostyrka (1 – 7)
  - c. Sätt startvärde på variabler
2. **Med knapp A** sänds kundnummer.  
I programmet har vi av enkelhet valt att börja om från ett (1) när man kommer till kund nr 10. Knapp A är som sagt en händelse - vad ska hända?  
Svar: Bestäm kundnummer och sänd detta till andra displayen.
3. I huvudloopen **för alltid ...** skrivs aktuellt kundnummer ut på display



### Mottagare med display

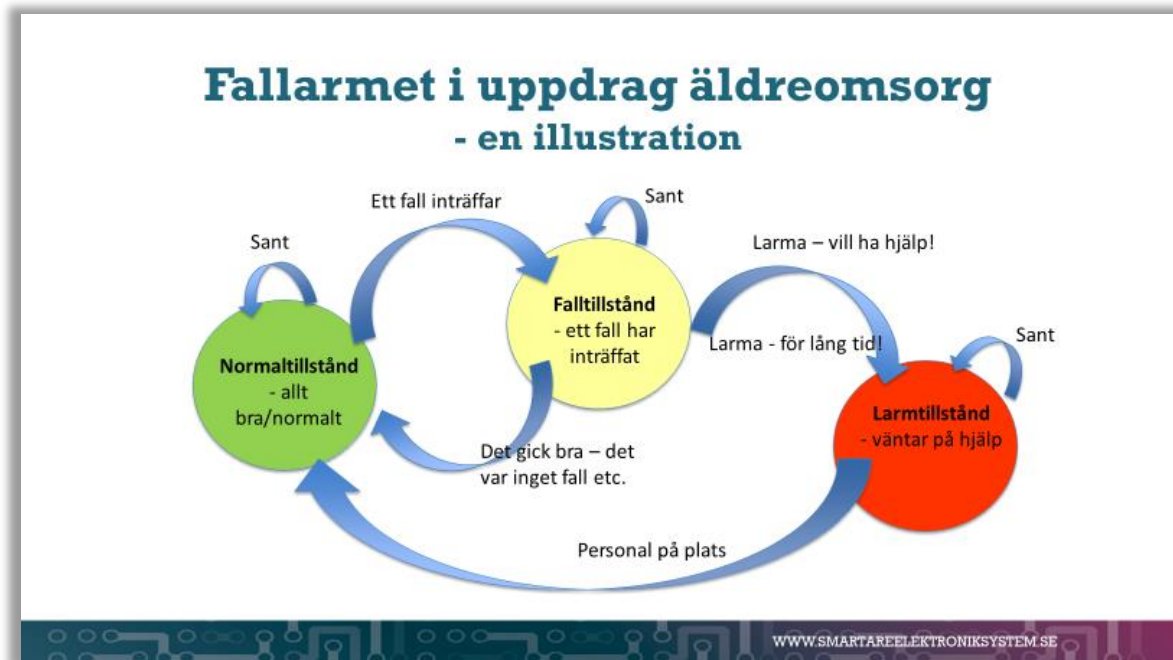
1. **Initiera** mottagare
  - a. Ställ in radiogrupp (1 – 255)
  - b. Sätt startvärde på variabler
2. När ett data kommer in via radio hanteras detta i händelsefunktionen **när radio mottages** och indata, dvs kundnummer, tilldelas en variabel.
3. I loopen **för alltid** skrivs kundnumret ut och är alltså en kopia av nummer på huvuddisplayen.



## Uppdrag B – Smartare äldreomsorg

### Inledning

Först en översiktlig illustration av hur larmet ska fungera – dess s k beteende!



I bilden ses de tre tillstånden hos larmet

- Normaltillstånd (grönt)
- Falltillstånd (gult blinkande)
- Larmtillstånd (rött)

Pilarna avser att beskriva vad som händer vid olika insignaler i systemet.

Illustrationen är en beskrivning på hur programmet ska fungera utifrån olika tillstånd och händelser, ett sätt att beskriva beteendet med s k tillståndsdigram. Metoden används i högre programmeringsutbildningar.

De hårdvaruenheter på micro:bit som ska användas i uppdraget förutom lysdioderna är

- Accelerometern för att registrera fall
- Knapp A för att återställa fall eller larm till normaltillstånd
- Knapp B för att larma

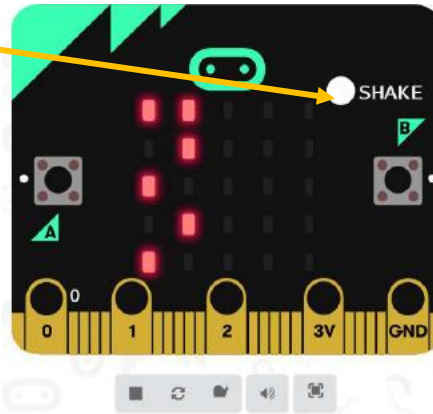
Programblock för dessa finns som händelser, vilket vi använder för att byta tillstånd.

För att stå kvar i ett tillstånd innan nästa används en ny loop i programmeringen, **medan-loopen** eller på engelska *while-loopen*. Se kommande sidor om detta.

## Utrustning och inledande val

Här ska ett **fall-larm utvecklas** och vi använder samma utrustning som i trafikljuset, men här är förutom knappar även micro:bits **accelerometer en inenhet** och har en central roll i första skedet av beteendet i uppdraget. Det innebär att **skaka**-blocket under **Input** ska användas i programmeringen. Här finns två användbara skaka-alternativen vid utvecklingen av prototypen.

- **valet skaka i simuleringsläge**



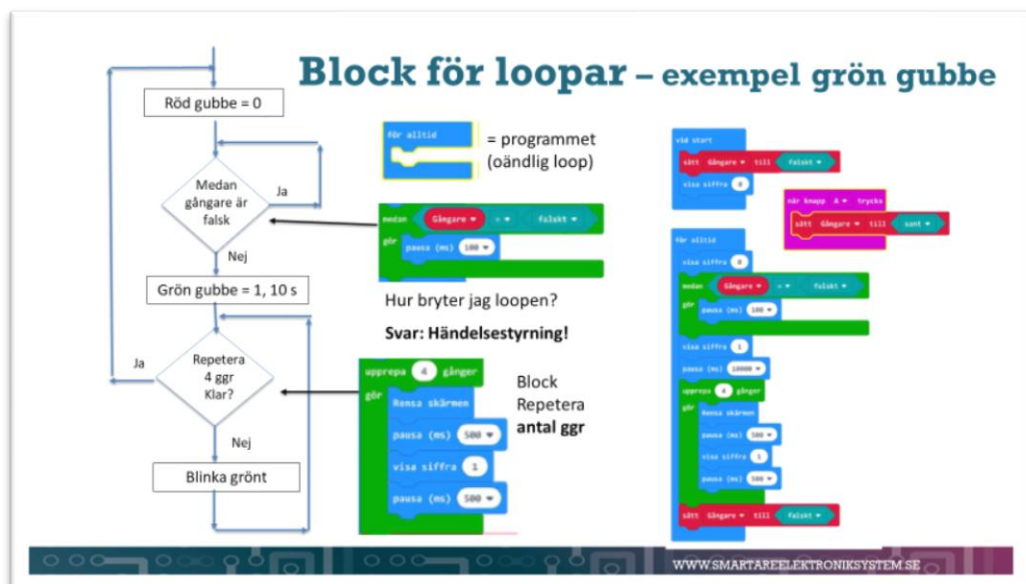
- **valet 3g för det nedladdade läget.** Valet 3g har gjorts för att accelerometern reagerar på mindre rörelser och stötar, vilket passar i sammanhanget. Utrustningen **ska inte** falla i golvet, i bänken eller motsvarande utan kan släppas från ena handen till den andra och "det fungerar som ett fall" för vårt behov!

## Programmering och loopar

Vid programmeringen kommer vi även här in på händelsestyrning i programmet. Att en person faller är en händelse som ska tas om hand av programmet för åtgärd!

Innan vi kommer dit ska vi titta på ett par exempel med loopar i programmering.

Vi går tillbaka till trafikljuset och tittar på exemplet att en gångare ska över gatukorsningen. Det blir inte grönt förrän gångaren trycker på knappen (en händelse). Vi analyserar programmet på nästa sida.



## Programmets flödesdiagram

Det börjar med Röd gubbe, vilket motsvarar en nolla på displayen

### Loop 1

**Så länge som (=medan)** en person inte har tryckt på knappen är det röd gubbe (=0).

En gångare trycker på knappen.

### Medan-loopen bryts då för att gångare blir sant

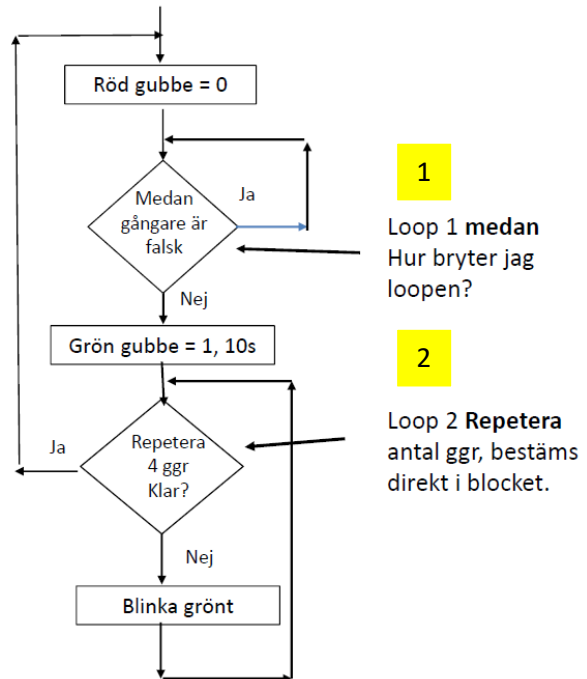
Knappen är händelsen som gör att Gångare blir sant, vilket kommer att bryta medan-loopen 1!

Det blir grön gubbe (=1) i 10 s.

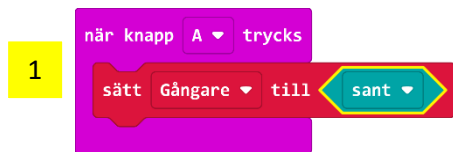
### Loop 2

Innan det blir röd gubbe igen blinkar den gröna några gånger, vi har 4 blinkar. Antalet upprepningar bestäms i ramen för loop - blocket.

Programmet börjar därefter om med röd gubbe i "för alltid"-loopen, programmets huvudloop.



## Djupare analys

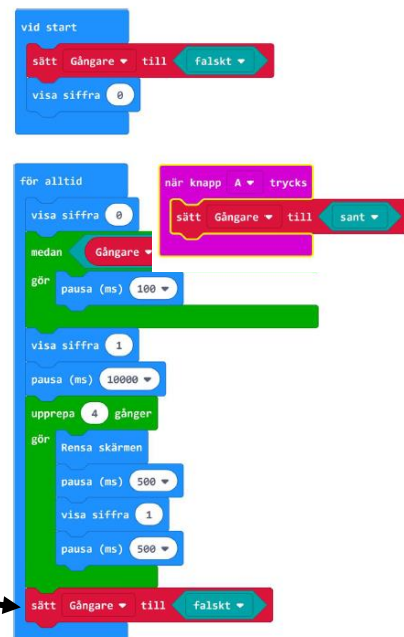


Med Knapp A meddelar gångaren att hen vill gå över. Variabeln Gångare blir sant, vilket **bryter** medan-loopen!

Paus-funktionen i medan-loopen är "bara en utfyllnad" för att få loopen att fungera.



I loopen **upprepa** talar man om från start av loopen att den ska upprepas 4 gånger, i detta fall blinka med grön gubbe.



... och hela programmet blir ==>

En **introduktion av framförallt medan-loopen**, som nu ska användas i uppdrag B med fall-larmet.

**Notera att** variabeln Gångare återställs till falskt innan loopen "för alltid" börjar om på nytt

## Fall-larmet

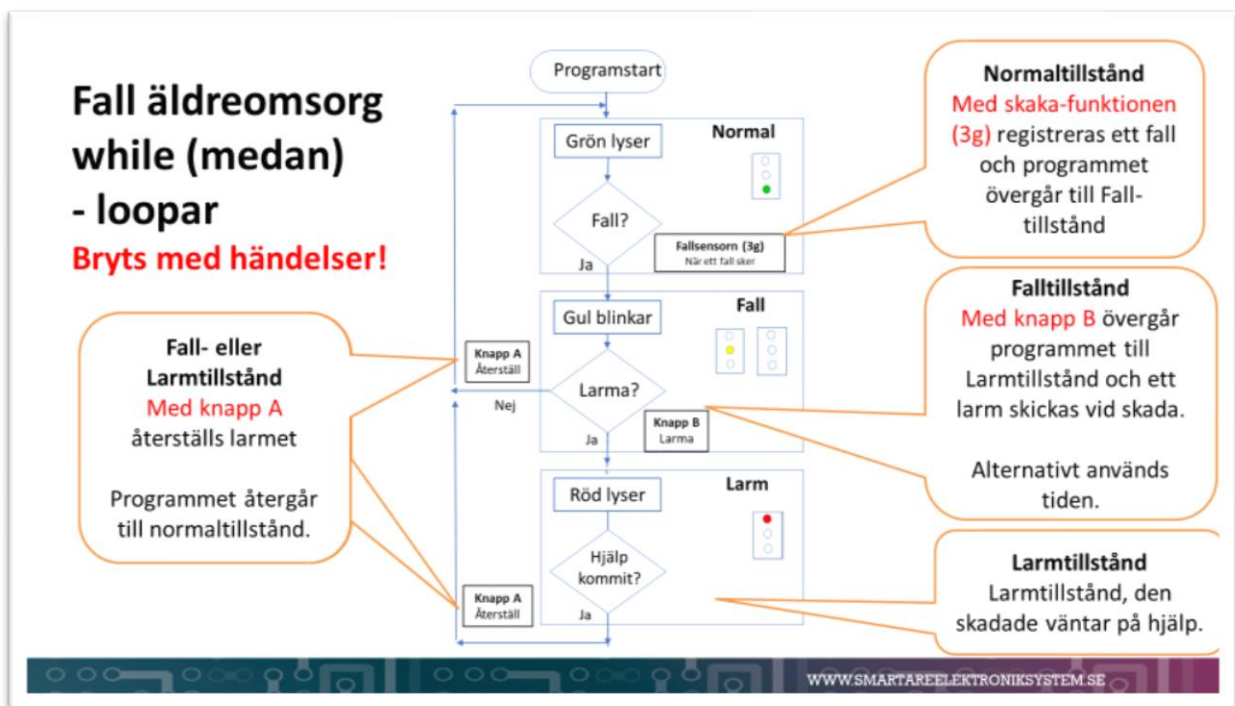
Nedan följer lite mer information i bild och text vad gäller det slutliga programmet i uppdraget med fall-larmet. Flödesdiagrammet nedan är lite förenklat bara för att få en översikt av larmets funktion.

Loopen **medan (while)** används i programmet för att ligga kvar i respektive tillstånd tills en händelse bryter loopen och programmet övergår till nästa tillstånd. En beskrivning som används i uppdraget och används i diskussionen med elever.

Händelserna som styr i detta program är;

- **Skaka**, ett fall har inträffat och den gula lampan ska blinka
- **Knapp A**, återställ till normaltillstånd och den gröna lampan ska lysa
- **Knapp B**, larma för en skada har inträffat och den röda lampan ska lysa
- **Tiden**, som inte syns här, men som utlöser ett larm efter 20 s i Falltillstånd

Nu till programmets slutliga uppbyggnad.



Diskutera eventuellt mer om tillstånden i figuren med eleverna:

- Vilket tillstånd borde vara det man befinner sig i längst tid?
- Hur länge befinner sig programmet i Normaltillstånd kontra Fall – respektive Larmtillstånd. Fundera kring tidsaspekten för respektive tillstånd.
- Mer om tidsaspekten i programmet, vilken längsta tid bör personen befinna sig i Falltillstånd innan ett larm utlöses?
- Vi har satt 20 s för att det ska vara praktiskt "att laborera med".

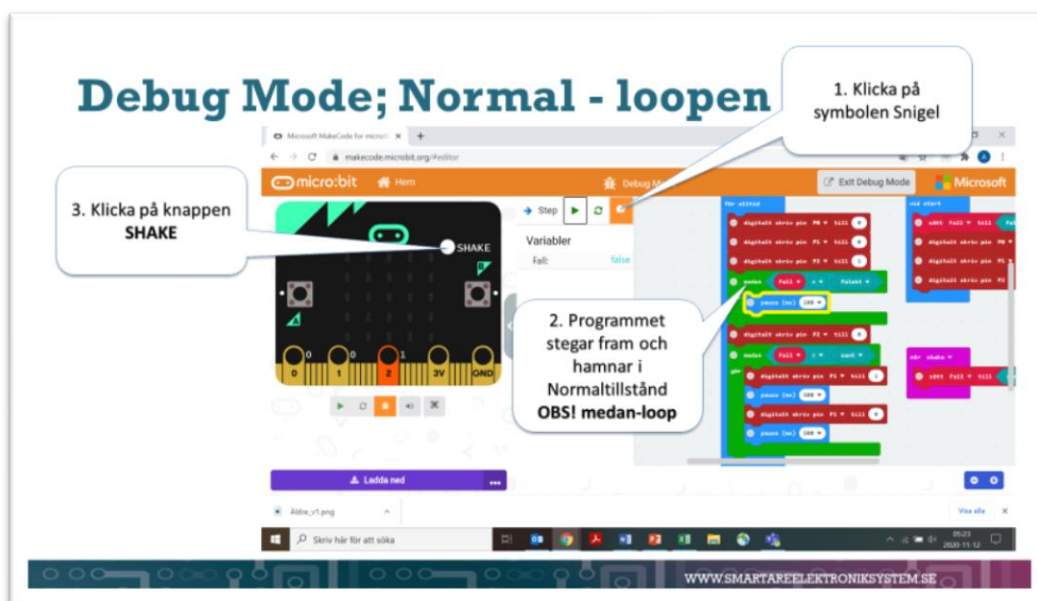
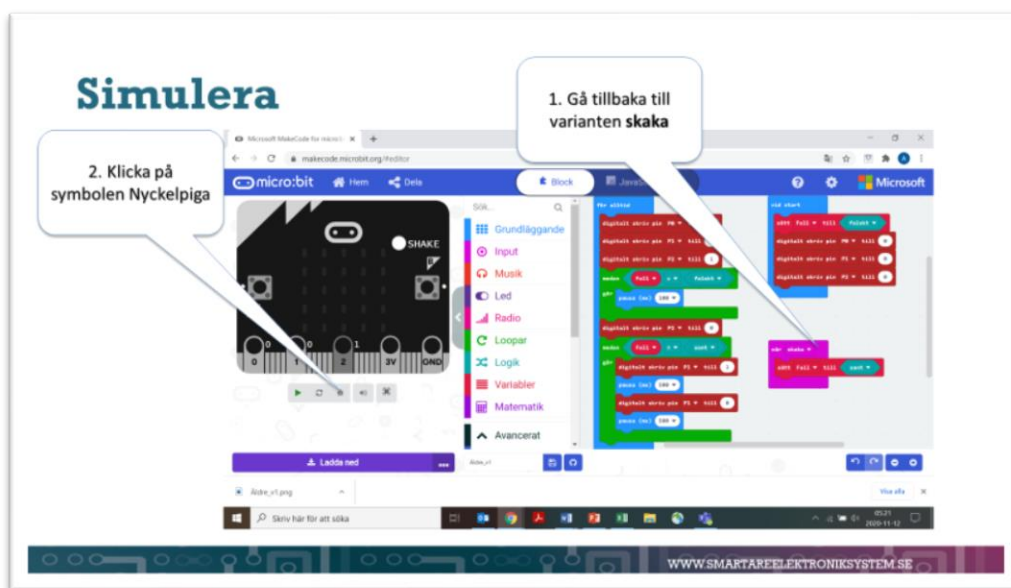
## Använda utvecklingsmiljön

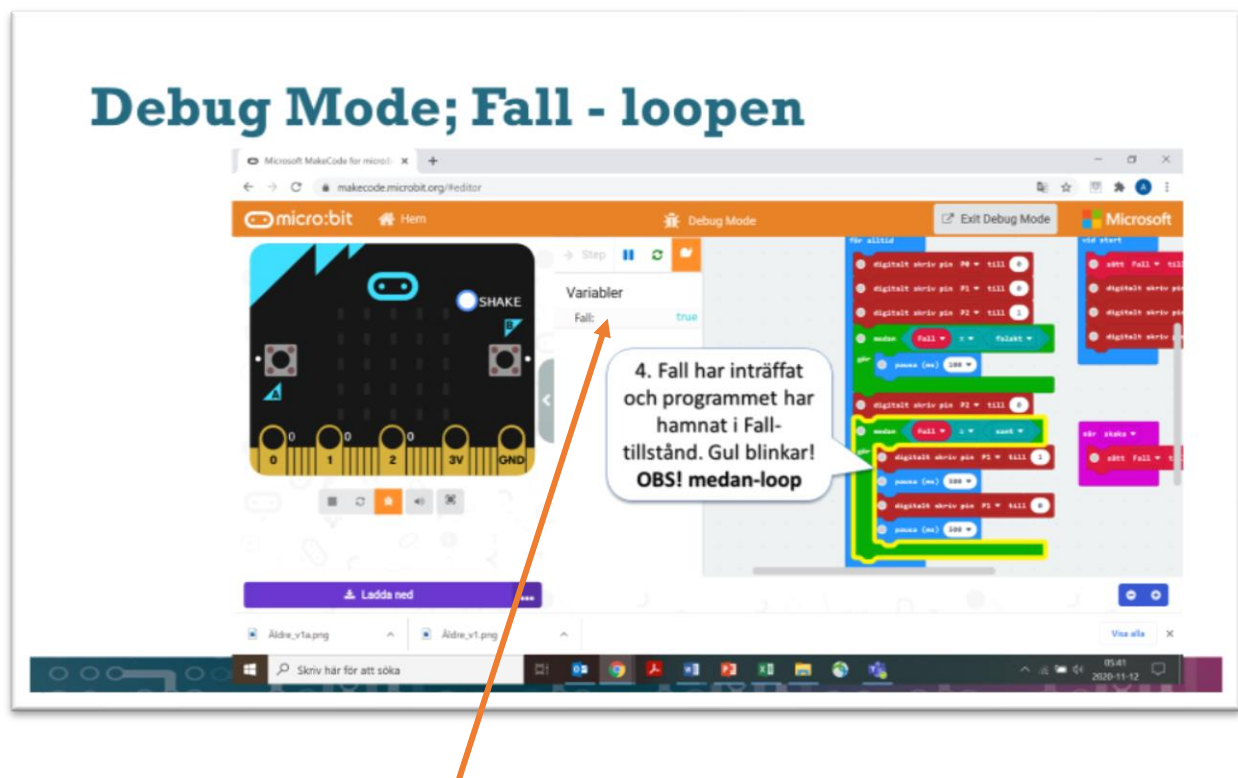
Nedan följer några bilder som beskriver hur utvecklingsmiljön kan användas i utvecklingen av programmet. Debug Mode har nämnts tidigare, ett nytt verktyg/tillbehör i miljön, och här kommer mer information om den möjligheten i att studera programmets exekvering samt hur variabler ändrar "värde" under exekveringen. Allt för att förstå programmet, dess beteende, eller att söka och diskutera vad som händer vid en felsökning.

## Simulera

I uppdraget med äldreomsorg väljs fallalternativet 3g för att enkelt känna av ett fall i praktiken. För att kunna simulera och undersöka programmet stödjer utvecklingsmiljön bara fallalternativet **skaka**. Ändra därför till detta alternativ i programmet när simulering och Debug ska användas..

Nu följer några bilder för fortsättningen. Följ anvisningarna i bilderna för att komma igång.





Här ser man också att variabeln Fall har värdet true. **Variabeln Fall är Boolesk** och kan bara ha två värden, false eller true ( falsk eller sann ).

**Debug Mode** är ett kraftfullt verktyg i att undersöka program och hur de beter sig under exekvering, ett verktyg som programvaruutvecklare/ingenjörer ofta använder.

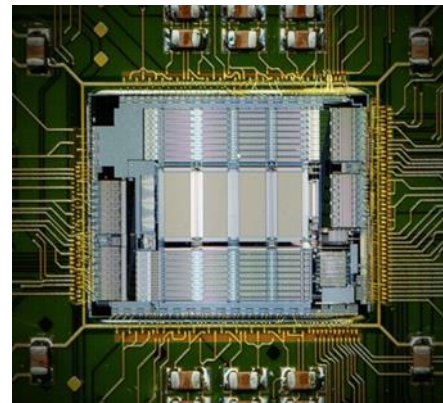
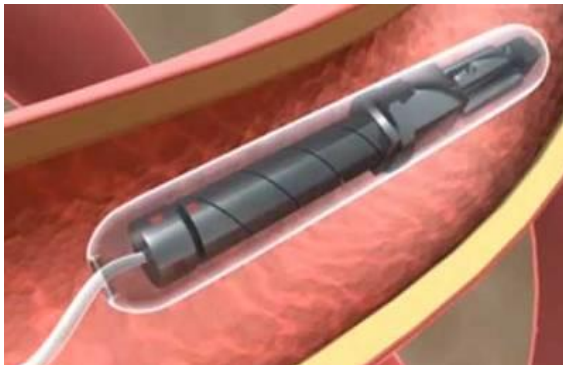
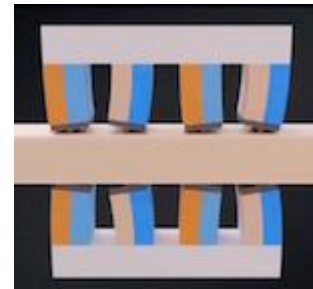
Efter att ha studerat programmet klicka på knappen **Exit Debug Mode**. Nu tillbaka till utvecklingsmiljön, ändra eventuella fel samt också **tillbaka till 3g** igen innan nerladdning.

## Mikro- och nanoelektronik

Hit räknas mikroelektromekaniska system in, s.k. MEMS. Det är mekaniska system med tillhörande elektronik, allt i mikrometerskala eller mindre. Accelerometern är en MEMS-komponent. Vi har i Sverige flera stora anläggningar för forskning och viss tillverkning kring tekniken, exempelvis i Kista, Uppsala, Göteborg och Lund.

Ett företag som har en världsledande MEMS-fabrik, en FAB, och heter Silex Microsystems. Företaget startade i Sverige i början 2000-talet och har sin anläggning i Järfälla utanför Stockholm.

- Kända produkter och system med MEMS-teknik som har utvecklats och tagits fram i Sverige är krocksensorer för bilar (företaget Autoliv). De innehåller sensorer med bl. a accelerometrar för att känna rörelseförändringar i tre dimensioner. Dessa komponenter finns numera även i mobiltelefoner för att exempelvis få en bild i rätt position (läge) när man ska titta på den.
- Andra sensorer med MEMS finns för kranskärlsundersökning med tillhörande operationer. Ett företag i Uppsala fick framgångar med detta. Företaget ägs numera av ett stort amerikanskt medicinteknikföretag.
- Ett annat Uppsalaföretag (Piezomotor) tillverkar motorer med Piezoelektriska egenskaper. Piezomotorer är kända för sin styrka baserade på piezoelektriska material. Motorerna kan röra sig linjärt som i bilden eller i rotation, **OCH** i nanometerskala. De används bl. a i kameror och i medicinsk utrustning för att få exakt positionering, exempelvis vid dosering av läkemedel. Tidigt under hösten 2020 avslöjade Piezomotor att företaget arbetar med utvecklingen av en motor för konsumentelektronik, exempelvis mobiltelefoner. Nu år 2021 förtydligar företaget att det är zoomfunktion i mobiltelefonkameror som intresserar flera mobiltelefonstillverkare. Under 2020 fick företaget förfrågningar från flera mobiltelefonstillverkare och deras underleverantörer, vilket lett till att företaget börjat utveckla en motorteknik för extremt små utrymmen i bärbart.  
**Bevaka området!**
- MEMS är ett högteknologiskt ämnesområde som kräver renrumsmiljö vid såväl utveckling som tillverkning!
- Nedan till vänster ses en produkt för att upptäcka förträngningar i blodkärl, en motor med sensor som kan föras in i blodkärl för mätning. De flesta större sjukhus i Sverige har sådan utrustning idag.



Sverige har idag inte kvar någon egen tillverkning av elektroniska kiselkomponenter och system som bilden till höger visar, men avancerad konstruktion av integrerade kretsar görs fortfarande och det är en kompetens som efterfrågas av allt fler företag.

## Micro:bit version 2

I december 2020 släpptes Micro:bit version 2, med en del nyheter. Som sagts tidigare har den också en ny *Firmware* som stödjer användandet av de nya elektroniska funktionerna i systemet. Det finns också ett antal nya block för programmering av de nya funktionerna. Se bilderna nedan, där det nya är synligt på såväl Micro:bits fram- som baksida samt i bilden från utvecklingsmiljön längst ner.

- En mikrofon (syns på fram- och baksida)
- En högtalare (baksida)
- En touch-sensor (framsida)
- Antennen är vriden 45 grader av utrymmesskäl
- Det ska var enklare att fästa krokodilklämmor på kortet med urfasningarna nedtill
- Utvecklingsmiljön ser likadan ut och bland programmeringsblocken finns mjukvarustöd för att använda den nya hårdvaran, vilket för vår del ger utökade möjligheter till fördjupning etc.

Enligt informationen från Kitronik ska det inte vara några förändringar som kommer att påverka vår utrustning och våra uppdrag, vilket våra tester bekräftar.



### Lite om nyheterna hos Micro:bit version 2

- Hårdvara
  - Touch-sensor
  - Mikrofon
- Mjukvara
  - Tillägg bland programmeringsblocken, se rubriken microbit (V2)
- Simulering
  - Exempelvis kan ljudnivå simuleras

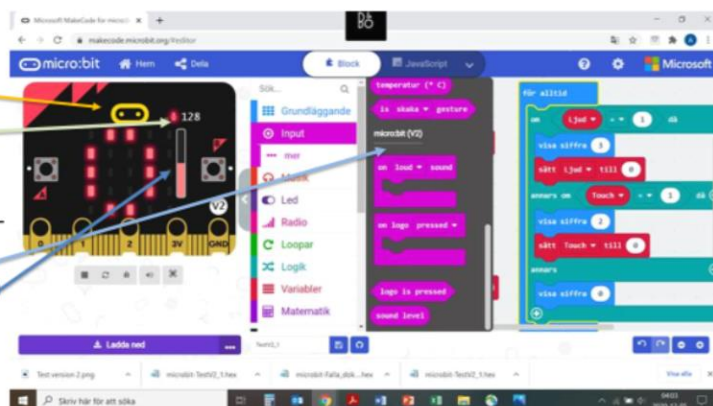


Bild från utvecklingsmiljön där det finns några nya tillägg vid simulering och debugging.

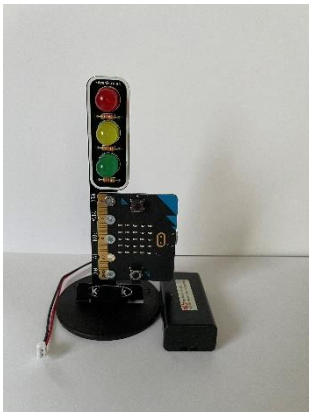
## Testprogram med de nya funktionerna i version 2

### Händelsestyrning med Touch-knapp och inkommande Ljud

Touch-knappen kan användas på samma sätt som Knapp A och B, men upplevs som känsligare i och med att den reagerar vid beröring, inte tryck.

Ljudet, ett nytt sätt för händelsestyrning som exempelvis kan användas för larm som glaskross etc. eller att kalla på hjälp i en nödsituation.

Som utrustning använder vi trafikljuset med Micro:bit version 2 inkopplad. Programmet nedan testat dessa två funktioner. Händelserna skriver ut en siffra på displayen när de har inträffat.



```

vid start
  sätt Ljud till 0
  sätt Touch till 0

on loud sound
  sätt Ljud till 1

on logo pressed
  sätt Touch till 1

för alltid
  om Ljud = 1 då
    visa siffra 3
    sätt Ljud till 0
  annars om Touch = 1 då
    visa siffra 2
    sätt Touch till 0
  annars
    visa siffra 0
  
```

### Bullermätare

En intressant applikation är att göra en bullermätare, som en jämförelse med en ljusmätare. Se programexempel på detta.

Bullret (ljudet) mäts med mikrofonen och man får ett värde mellan 0 – 255 precis som för ljusmätaren.

I programmet tänds grön signal på trafikljuset om det inte är något buller och den röda signalen tänds om det är ett buller. Gränsvärdet är satt till värdet 50 för att lätt kunna demonstrera.

Bullermätaren slås på och av med hjälp av touch-knappen. Displayen visar 1 om den är på och 0 om den är avstängd.

```

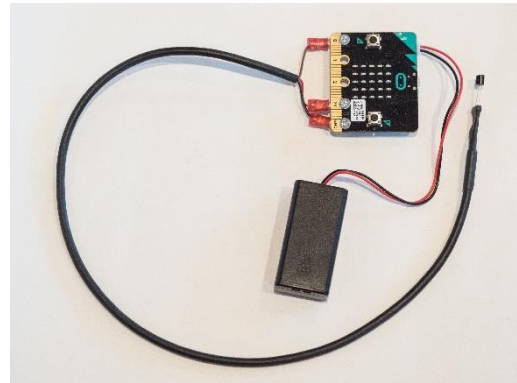
vid start
  sätt Ljudnivå till 0
  sätt Touch till 0
  digitalt skriv pin P2 till 0
  digitalt skriv pin P0 till 0
  visa siffra 0

on logo pressed
  om Touch = 0 då
    sätt Touch till 1
    visa siffra 1
  annars
    sätt Touch till 0
    digitalt skriv pin P2 till 0
    digitalt skriv pin P0 till 0
    visa siffra 0

för alltid
  medan Touch = 0
  gör
    pausa (ms) 100
  medan Touch = 1
  gör
    sätt Ljudnivå till sound level
    om Ljudnivå > 50 då
      digitalt skriv pin P0 till 1
      digitalt skriv pin P2 till 0
    annars
      digitalt skriv pin P0 till 0
      digitalt skriv pin P2 till 1
    pausa (ms) 100
  
```

## Uppdrag C – Kylkedjan

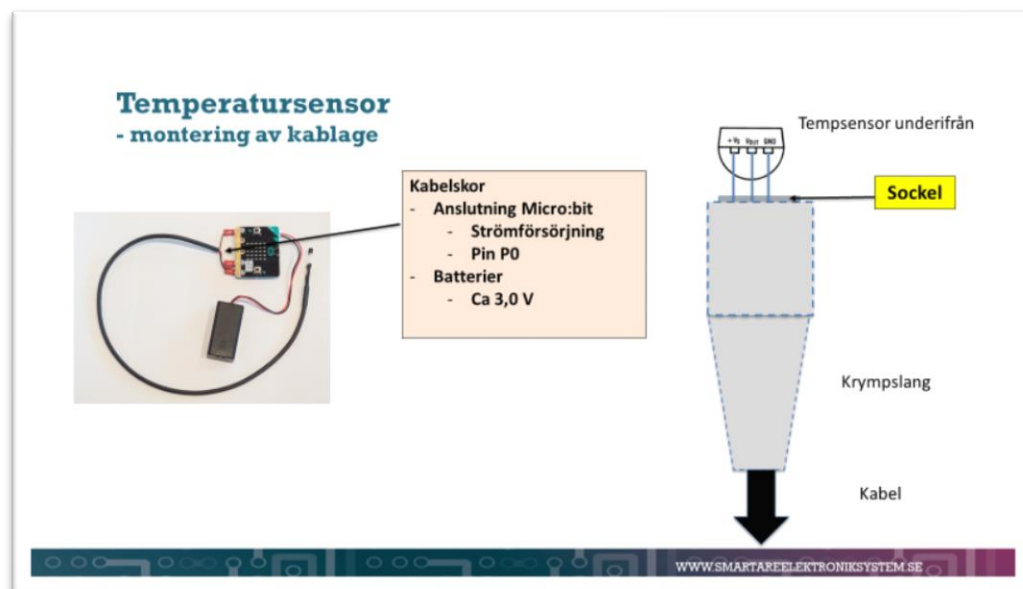
Här ska en temperatursensor användas. Utrustningen till denna är utvecklad i samarbete med Smartare elektroniksystem. Temperatursensorn är en elektronikkomponent, en integrerad elektronikkrets med sensorelement och förstärkare, vilken har kapslats in i en kapsel med tre ben (anslutningar). För att vi ska kunna använda komponenten för vårt ändamål ansluts den till micro:biten via en kabel med tre ledare.



För att ansluta kabelns tre ledare från temperatursensorn till micro:biten används sk kabelskor. På så sätt kan temperatursensorn anslutas till micro:bit-kortet på samma sätt som den övriga utrustningen i uppdraget, med skruv och mutter. De tre kabelskorna placeras så att

- Röd kabel ansluts till + 3V
- Svart kabel ansluts till 0V (GND)
- Brun kabel ansluts till pin P0

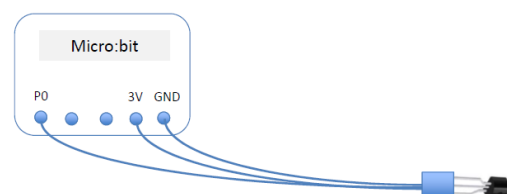
Tänk på att dra åt skruvarna, men som sagt, lagom hårt.



Temperatursensorn sitter monterad och är fastlödd i sockeln, stiftlisten. Tittar man på komponenten underifrån sitter GND ( 0V ) närmast det hål i stiftlisten som är igensatt (pluggad).

**Viktig kunskap för er som har prototypkablaget, där sensorn sitter utan lödning i sockeln.**

I dokumentationen för de experimentella delarna i uppdraget används bilden till höger, en förenklad bild som passar i det sammanhang som sensorn används.

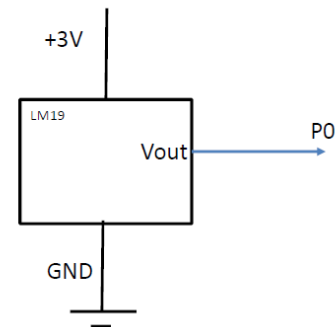


**Observera** att när programmet är nerladdat ska batterihållaren med de två AAA-batterierna vara den strömförsörjning (ca 3V) som används i detta uppdrag!

## Fakta om sensorn och program

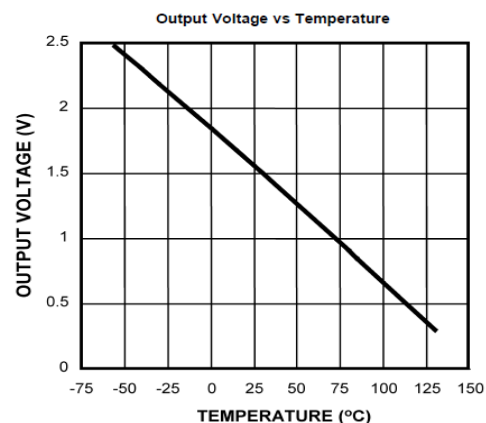
Till varje elektronikkomponent finns ett datablad, och gäller det en integrerad halvledarkrets kan databladet vara omfattande. I databladet beskrivs exempelvis hur komponenten ska anslutas, applikationsexempel osv, men den mesta informationen avser egenskaper hos komponenten i olika avseenden. Tabeller och diagram är vanliga, och en viktig information för den som ska välja komponent för att utveckla en elektronisk produkt eller att programmera med den i en tillämpning.

För oss är det viktigt hur vi ska använda komponenten och vilka egenskaper den har i grova drag. Det kan vara nog så svårt ändå! Vi kan se den som en svart box med tre anslutningar, där strömförsörjningen är två pinnar och signalen, dvs data, kommer på den tredje pinnen. Se figuren till höger.



Här följer några fakta:

- Temperatursensorn ger en analog signal, en spänning mellan 0 – 3 V beroende på temperatur. Den analoga signalens spänning är ett mått på temperaturen och är alltid densamma **oberoende** av batterispänningens värde.
- Med Micro:bit och dess ingång PO omvandlas spänningen från sensorn till ett tal mellan 0 – 1023 på samma sätt som för ljussensorn i uppdrag 2.
- Temperatursensorn har ett linjärt samband mellan temperatur och spänning, se diagram.
- I databladet får vi sambandet angivet, det står  $V_{out} = -0,01177 * T + 1,86$ . Enheten är Volt (V).
- Eller så kan sambandet tas fram genom att använda diagrammet i kombination med räta linjens ekvation  $y = k*x + m$ , där y är spänningen  $V_{out}$  och x är temperaturen T.
- Vi skriver om formeln till mV och då blir den  $V_{out} = -11,77 T + 1860$ , vilket passar bättre i och med att micro:bit gör om spänningen till ett tal, binärt mellan 0 – 1023.
- Eftersom det är temperaturen T vi är ute efter blir  $T = (V_{out} - 1860)/(-11,77)$ .
- Denna formel blir utgångspunkten till den algoritm vi ska använda för att få fram temperaturvärdet vid våra mätningar.
- Funktionen (algoritmen) kallar vi för Tempensor i programmet och den blir då enligt bilden till höger.
- Vi använder inte decimaler när temperaturen presenteras på displayen, därför görs resultatet för variabeln Temp bl a om till en **integer** i sista uträkningen, i divisionen.
- Funktionen anropas med **call Tempensor** i huvudprogrammet och variabeln Temp tilldelas temperaturvärdet för fortsatt bearbetning och användning i huvud-loopen.
- Temperaturen mäts kontinuerligt där Max- samt Medelvärde beräknas i det slutliga programmet.



## Simulering och Debug Mode

Programmet som utvecklas för temperaturmätning sker i två steg, som avser två varianter för anrop av temperaturfunktionen.

- Först det program som går att simulera för att testa dina tankar etc. i programmeringen.
  - Simulatoren är här ett alldeles utmärkt för detta och är ett bra hjälpmedel i utvecklingsarbetet. Precis som det ska gå till i ett arbete för en ingenjör/programmerare!
  - För att få simuleringen att fungera används micro:bits egen temperaturfunktion **temperatur (°C)** i programmet.
  - Med simulatoren **verifieras** programmets funktion.

Se fortsättningen med simulering i Debug Mode i nya fönstret på nästa sida.



I den nya versionen för utvecklingsmiljön kommer man hit, dvs fönstret nedan som kallas **Debug Mode** med möjligheter till simulering och analys av programmet i test- och felsökningsyfte.

3. Ändra temperaturen genom att röra på reglaget upp och ner.

5. Avsluta Debug Mode.

1. Nytt fönster – se rubriken i bilden. Vi är nu i Debug Mode.

2. Ett klick på Snigelsymbolen och programmet kan följas i *slowmotion* medan du ändrar temperaturen.

4. Programmets variabler och deras värden kan följas i mittenfältet.

Variabler	
Vref:	3120
Temp:	26
Maxtemp:	33
Medeltemp:	25
SummaTemp:	101

Följ anvisningarna i nummerföljd i bilden för att testa ditt program:

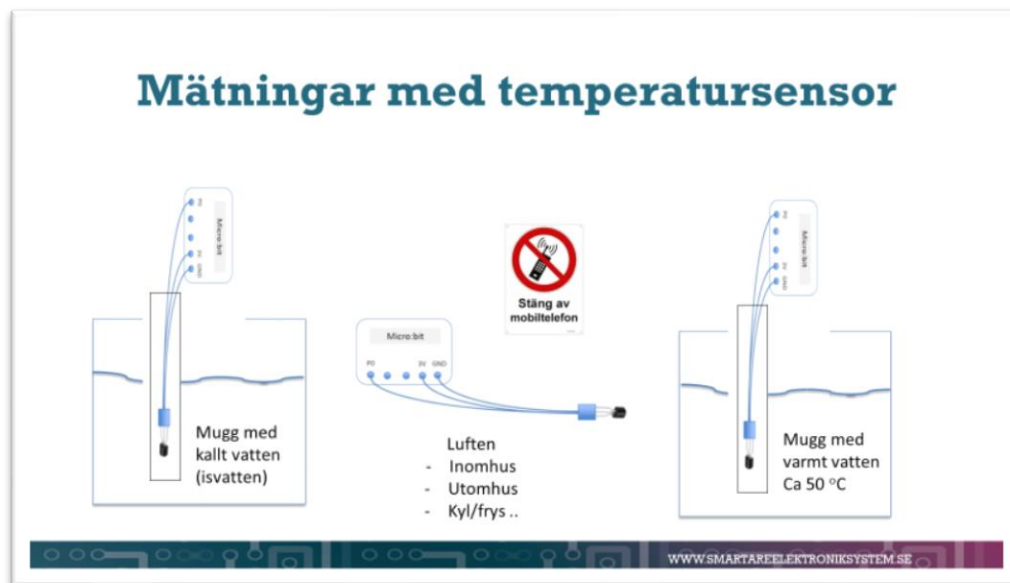
1. Du är nu i *Debug Mode*
2. För att analysera programmet i *slowmotion* - klicka på snigelsymbolen. Programmet exekverar här i *slowmotion* och man kan följa exekveringen via de gulmarkerade blocken i programmet.
3. Programmet ovan är ett temperaturprogram där temperaturen kan ändras med hjälp av temperaturstapeln.
4. I mitten av fönstret listas variablerna och deras värden vartefter programmet exekverar, dvs körs i *slowmotion*.
5. När du är nöjd, klicka på knappen Exit Debug Mode och man kommer tillbaka till editorn där programmet kan justeras, fortsätta att utvecklas osv.

### Fysiska mätningar

Temperaturen är den vanligaste parameter som vi människor är intresserade av i olika sammanhang. Det finns ett stort antal termometrar på marknaden för privat bruk, men också många exempel för yrkesbruk med olika behov. Ibland ställer det krav på stor noggrannhet och även på vilket sätt som temperaturen mäts.

Med vår temperatursensor kan vi mäta temperaturen i olika miljöer då kabeln är ca 50 cm lång.

I bilden ses några förslag på mätningar och i vår utrustning finns plaströr med för mätningar i vatten.



- I programmet med den verkliga temperatursensorn anropar vi i stället den egenutvecklade funktion som har sin grund hos vår temperatursensor och dess egenskaper.
  - Funktionen går inte att simulera i utvecklingsmiljön utan programmet måste laddas ner i micro:bit och exekveras för att testas i verkligheten.
  - När programmet exekverar kan vi både **verifiera och validera** program och funktioner.

I tester vi har gjort kan temperatursensorn på kabeln placeras inne i en kyl eller frys. Det fungerar bra om man är lite försiktig med sin mätuppställning. Andra exempel på mätningar finns i bilden på föregående sida.

**Resultat:** Systemet ska som sagts tidigare ha batterihållaren med två AAA-batterier på 1,5 V som strömförsörjning. USB-anslutningen ska alltså inte finnas kvar när mätningarna utförs. Batterierna kan ha lite olika kondition, vilket gör att eleverna får lite olika resultat i mätningarna. En grupp kan kanske få 24 grader i rumstemperatur, en annan 22 grader och en tredje 20 grader. För att få de rätta jämförelserna är det viktigt att samma batterier används i alla mätningar som den enskilda gruppen gör. Om man ska ta hänsyn till batteriernas kondition måste en kalibrering av "termometern" göras.

### Kalibrering av termometern i kylkedjeuppdraget

Nu kommer det ett lite svårare avsnitt när det gäller temperatursensorn och dess funktion.

Som vi sagt tidigare, " - Den analoga signalens spänning (från tempensorn) är ett mått på temperaturen och är alltid densamma **oberoende** av batterispänningens värde."

Microbit-kortets omvandlare på pinne P0 använder dock batterispänningen som referensspänning, vilket gör att det analoga omvandlingsvärdet av temperatursensorns värde kommer att variera med batterispänningen vid samma temperatur.

För att få rätt temperatur angiven kan termometern kalibreras. Detta går till på följande sätt:

- Mät upp spänningen mellan +3 V och GND på microbit-kortet med en voltmeter.
- **Observera att** mätningen ska ske då bara batterierna är inkopplade (USB är urkopplad) **OCH** då displayen ska vara släckt.
- Notera värdet i mV, exempelvis 3,045 V blir 3045 mV.
- Detta värde skrivs in i start-funktionen **sätter Vref till 3045**.

## Kalibrering

### - mätning med voltmeter

- I samband med att systemet (=termometern) ska kalibreras kan en voltmeter användas.
- Med voltmeter mäts batterispänningen. Voltmeter är inställd på V (DC).
- Denna spänning är referens-spänning för omvandlingen till det tal som sedan används i algoritmen för att räkna ut temperaturen.
- Spänningsvärdet (Vref) skrivs in i startrutinen **vid start** i programmet.
- Spänningen anges då i mV (millivolt), se bild 5.



WWW.SMARTAREELEKTRONIKSYSTEM.SE

**Ladda ner** koden och temperaturmätningen kan börja utan USB-inkoppling, och **gör en Reset** för att starta mätningen från början.

Termometern är nu kalibrerad och visar rätt temperatur så länge som batterierna har detta värde, rimligtvis under den lektion som mätningarna pågår.

I grundversionen utgår vi från att batterierna är någorlunda fräscha, vilket innebär att spänningen ungefär är 3,120 V. Variabeln Vref får värdet 3120 i start-funktionen enligt bild. Detta värde på Vref används sedan i funktionen Tempsensor för att justera (kalibrera) termometern så den visar rätt temperatur.

Temperaturen beräknas sedan utifrån det linjära uttrycket från databladet, dvs  $V_{out} = -11,77 T + 1860$ .

Funktionens implementering i blockkoden nedan är svår att förstå, och frestas inte att förklara denna för hela klassen. Möjligen för enstaka elev.

### Här följer en förklaring:

De två blocken i klammern räknar om temperatursensorns mätvärde i förhållande till referensspänningen Vref.

I funktionens fortsatta block hittar du de två siffervärdena som ses i den linjära funktionen (1860 och 11,77). Vi vill inte använda decimaler vid presentationen av temperatur, därav används multiplikationen med 10 samt **integer** i divisionen för att få ett heltal.

Se även bildspelet samt elev- och lärardokument till uppdraget.

### Sammanfattning av programmeringen i Kylkedjan

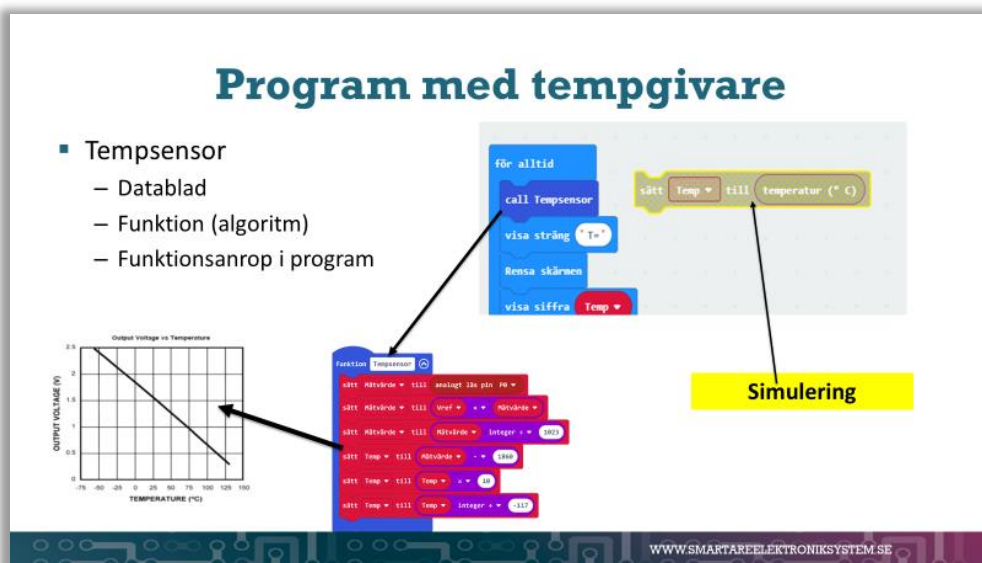
Sista bilden är en sammanfattning av utvecklingen - stegen i uppdragets temperaturmätning.

```

vid start
  sätt Temp till 0
  sätt MaxTemp till 0
  sätt Vref till 3120
  
```

```

funktion Tempsensor
  sätt Mätvärde till analogt läs pin P0
  sätt Mätvärde till Vref * Mätvärde
  sätt Mätvärde till Mätvärde / integer 1023
  sätt Temp till Mätvärde - 1860
  sätt Temp till Temp * 10
  sätt Temp till Temp / integer -117
  
```



**Program med tempgivare**

- Tempsensor
  - Datablad
  - Funktion (algorithm)
  - Funktionsanrop i program

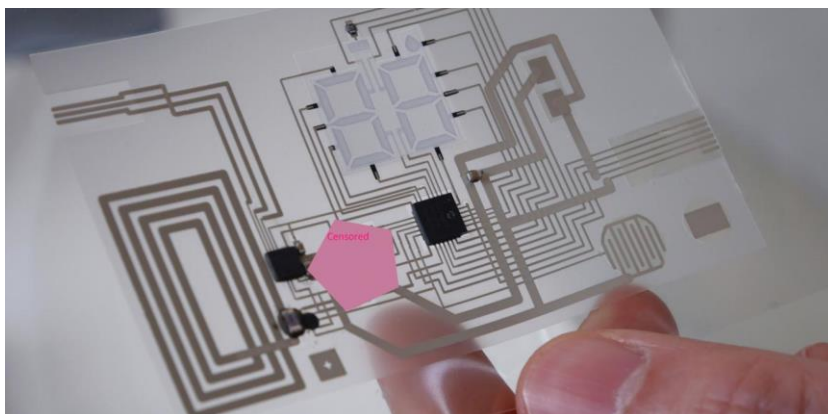
Simulering

WWW.SMARTAREELEKTRONIKSYSTEM.SE

## Tryckt elektronik

I uppdraget tas tryckt elektronik upp och som är ett förhållandevis nytt teknologiområde för elektronikbranschen. Området innebär en potentiell revolution för hur elektronik tillverkas. Detta har utvecklats de två senaste årtiondena. I Sverige utmärker sig verksamheten hos forskningsinstitutet RISE och Linköpings Universitet i Norrköping som världsledande. Det handlar om att kunna producera elektronik med konventionella metoder från den grafiska industrin.

- Det finns idag flera *start-up* - företag med produkter på marknaden men också flera stora företag som börjat använda teknologin i sina produkter och i sin produktion.
- Ett stort innovationsområde med många idéer till användning. Exempel på produkter som finns idag är fuktsensorer, temperatursensorer, solceller, blöjor för vuxna i äldreården som indikerar blöt blöja eller ett sårförband som indikerar när det är dags att lägga om sårförbandet.
- I bilden nedan ses ett hybridssystem med tryckta ledarbanor, antenner, displayer och några passiva komponenter tillsammans med några anslutna kapslade traditionella komponenter. Antenner för den trådlösa RFID-tekniken är ett nischområde i sig. Tryckt elektronik kännetecknas av produktion till en låg kostnad och hög effektivitet



Om du är intresserad av hur det går för den svenska utvecklingen av tryckt elektronik, gå till hemsidan <https://www.ri.se/sv/vad-vi-gor/expertiser/tryckt-elektronik>, här finns en hel del information kring området.

## Uppdrag D – Smartare säkerhetssystem

I uppdraget Smartare säkerhetssystem använder vi en vägbom som applikationsexempel på hur säkerhet kan utvecklas och belysas ur några olika aspekter. Det första man kanske tänker på när det gäller säkerhet är ett inbrottsskydd med larm etc. Men det finns fler begrepp inom området, som inte bara behandlar prylar för att förhindra inbrott utan även funktionssäkerhet och beteende!

Exempel på säkerhet och begrepp kan vara:

- Skalskydd, dvs en dörr eller fönster med elektroniskt lås
- Tillförlitlig/pålitlig, en grundfunktion fungerar även om en viss defekt i systemet har inträffat.
  - Exempel på detta är att en bil med ABS-bromsar ska kunna bromsas trots att ABS - funktionen inte fungerar som den ska
- Robusthet, en bro som har 5-faldig säkerhet i hållfasthet för den belastning den ska utsättas för
- Datasäkerhet, att informationen som sänds mellan olika enheter i systemet inte ska kunna avlyssnas eller påverkas
- Brandvägg, som har två betydelser
  - Att en vägg eller dörr i en byggnad som står emot brand betydligt längre än normalt
  - Att det är stopp för olaga intrång i ett datorsystem och dess programvara
- Här finns plats för diskussion kring säkerhet i olika situationer och tillämpningar!

Här finns också plats för engelsk terminologi, vilket eventuellt kan underlätta förståelsen.

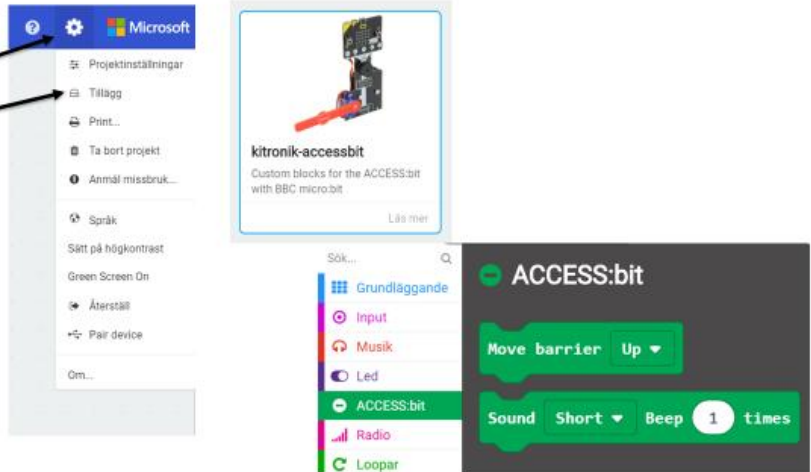
### Utrustning och Kitroniks funktioner

Till vägbommen från Kitronik finns liksom för trafikljus och belysning några funktioner att använda om man vill komma igång snabbt. Att hitta dessa kan gå till som i figuren nedan, med några steg (klick) och funktionerna finns med under ACCESS:bit bland blockkategorierna för programmeringen.

## Funktioner för styrning och reglering\_B

■ ACCESS:bit

1. Kugghjul
2. Tillägg
3. Skriv "Kitronik"
4. Välj accessbit
5. ACCESS:bit i listan



WWW.SMARTAREELEKTRONIKSYSTEM.SE

I uppdraget tas det i första programmen upp jämförelsen mellan Kitroniks funktioner kontra de standardfunktioner som kan användas från utvecklingsmiljön, se bild nedan.

## Standardfunktioner som alternativ

Move barrier Down ▾

=

servo skriv pin P0 ▾ till 90

Move barrier Up ▾

=

servo skriv pin P0 ▾ till 0

**Move barrier** - egentligen ett block som heter **servo skriv pin P0 till <grader>**.

© 2019 speediang! Copyright NTA Solutveckling WWW.SMARTAREELEKTRONIKSYSTEM.SE

Med de två varianterna kan bommen fällas ner och upp med två olika programblock, som vid en kortare anblick ser ganska olika ut. I bilden nedan används Kitroniks funktioner till vänster och till högra spinner vidare på det vi har som en röd tråd i uppdragen, att en händelse styr huvudprogrammet på ett mer generellt sätt.

## Program – ex. två alternativ

Högra programmet mer generellt och går bra att bygga på!

WWW.SMARTAREELEKTRONIKSYSTEM.SE

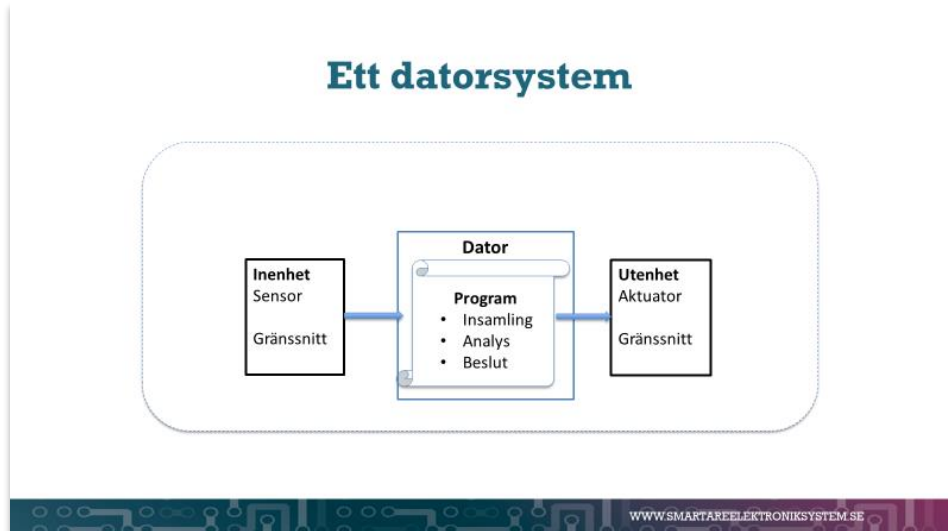
Med Micro:bit version 2 kan förmodligen ljud från den högtalaren användas som alternativ.

## Radio

I uppdraget med säkerhetssystem används lite senare i uppdraget radion hos Micro:bit för att öppna bommen. Under uppdraget Trafiksystem finns ett avsnitt om radion och hur den kan användas.

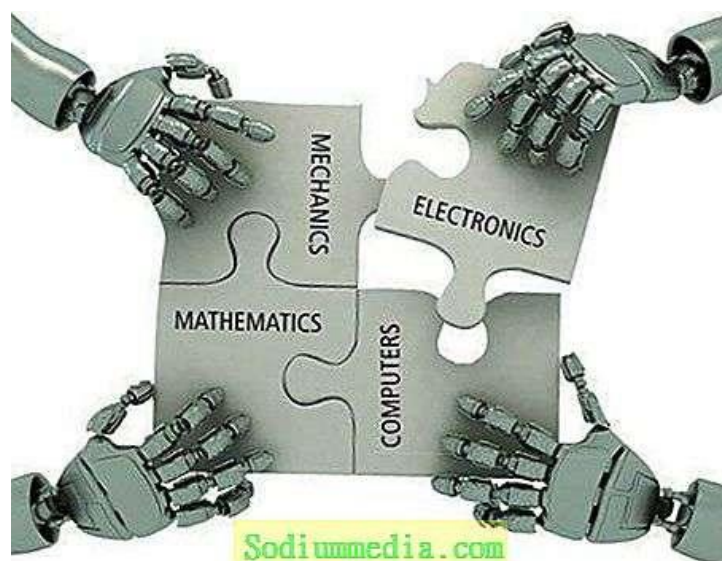
## Mekatronik

När en styrenhet påverkar en mekanisk funktion som exempelvis en rörelse har vi kommit in på ett begrepp som benämns Mekatronik. Det betyder att styrning och reglering av en mekanisk funktion görs med en elektronisk enhet. I fallet med vägbommen kommer vi tillbaka till bilden vi började med, det med **inenhet – analys/beslut – utenhet**. Som inenhet används en knapp inledningsvis för att sedan vara en krypterad information via radio.

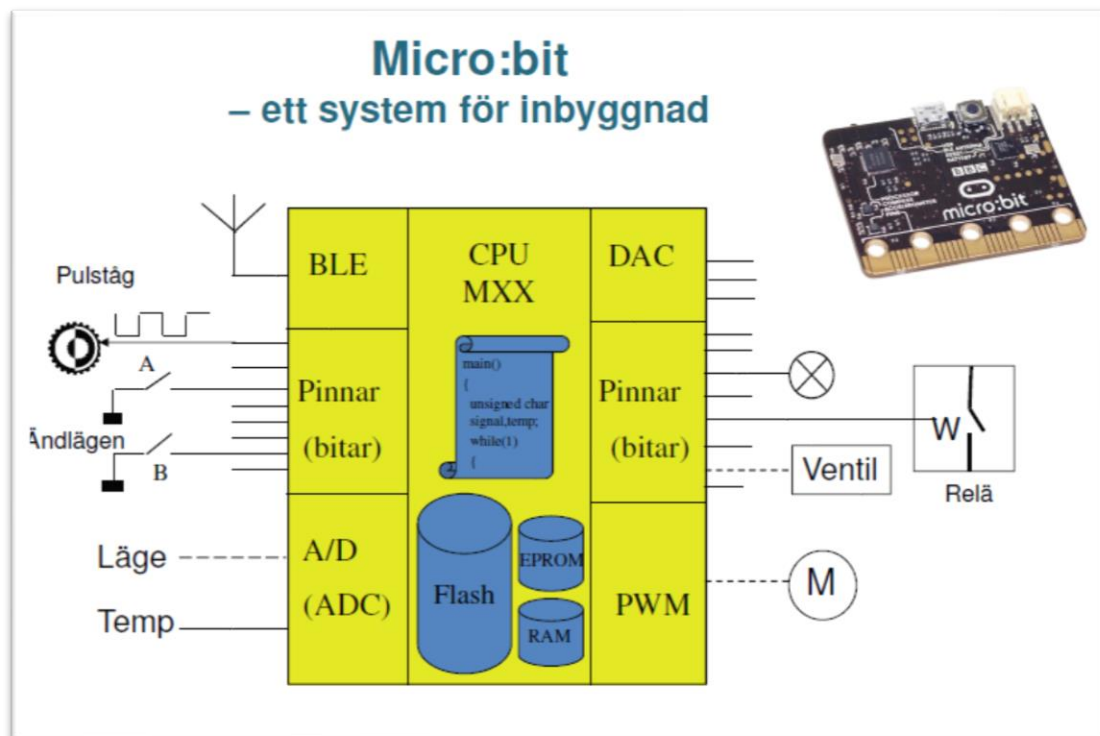


Utenheten är servomotorn i bommen, dvs aktuatoren. En servomotor är en motor som kan regleras med avseende på varvtal och position. Den har en egen elektronik och växellåda som påverkas av programmet som ska reglera motorn. Bland blocken under **Avancerat/pins** finns några block som kan användas för att köra en servomotor i ett program. Det finns exempelvis instruktionen **servo skriv P0 till 90** som vrider motoraxeln 90 grader i förhållande till en referens.

Mekatronik är en tvärvetenskaplig disciplin som innefattar flera kompetenser, se bilden nedan.



## Modell av en Micro:bit



Bilden är en illustration, en modell, på vilket sätt ett micro:bit-kort med sina olika enheter kan beskrivas. I bilden ingår dessutom några exempel på vad som kan vara en inenhet respektive en utenhet i sammanhanget. Processorn i systemet är en mikrokontroller, vilket innebär att den är gjord för att sitta inbyggd i produkter och hantera funktioner, som exempelvis i en tvättmaskin. Vi spinner vidare på detta med tvättmaskinen:

- Knapp A kan vara att ”- jag vill ha centrifugering på högre varvtal”
- Med en magnetsensor skapas det pulser när trumman snurrar, vilket kan avläsas och hastigheten hos trumman kan regleras till rätt varvtal hos motorn vid centrifugering.
- Temp, en sensor som finns med och reglerar inställd temperatur på vattnet vid tvättning
- Lampan talar om att tvättmaskinen är igång
- Ventilen styrs för att öppna och stänga för vatten till/från tvättmaskinen
- Med trådlös kommunikation (BLE) kan maskinen bli uppkopplad och startas på distans liksom att den talar om när tvätten är klar.
- Osv
- Grundprogrammet för maskinen finns i minnet EPROM. Variabler och tillfälliga värden under drift sparas i RAM och aktuellt tvättprogrammet med inställda parametrar finns i Flash-minnet.
  - EPROM betyder Erasable Programmable Read Only Memory, ett minne som är statiskt men som kan ändras under vissa förutsättningar och behov. Det görs i så fall av servicepersonal i fallet tvättmaskin, jfr underhållsmodet i vårt fall.
  - RAM betyder Random Access Memory, ett flyktigt minne som används av programmet för att tillfälligt spara variabler och adresser i program under drift, t ex vid händelser.
  - Flash är minnet för det aktuella applikationsprogrammet, det minne vi använder när ett program laddas ner via USB-kabel eller BLE i vårt fall.

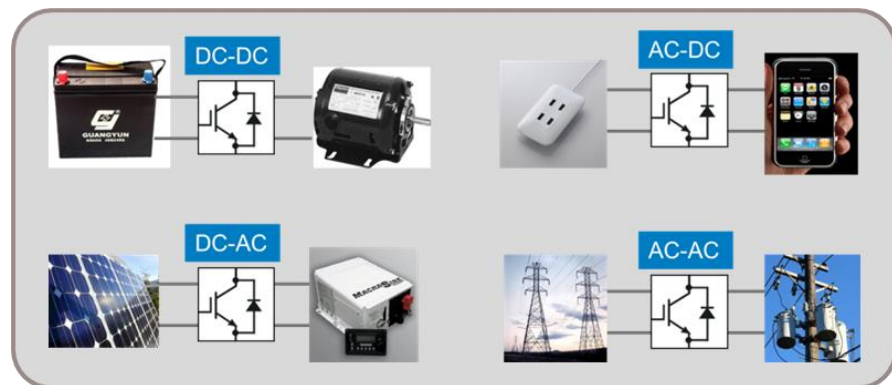
## Kraftelektronik

Mekatroniken omfattar exempelvis styrning och reglering av motorer för olika effekter. Vi kommer då in på kraftelektronikens område, en teknologi som exempelvis flera svenska företagen har i sina produkter och system. Behov som miniaturisering av elektronik samtidigt som man vill kunna hantera stora strömmar och effekter ställer stora krav på hur elektronik kan byggas in i en produkt samt hur programvara utformas i dessa tillämpningar.

Kraftelektronik är en nyckelteknologi för hållbar utveckling:

- Hållbar utveckling leder till ökad användning av elektrisk energi
- Ökande efterfrågan skapar krav på el-energibesparingar och effektivare användning av el-energi
- Stora besparingar kan fortfarande göras genom effektivare omvandling
- Material i halvledarna som kiselkarbid (SiC) och galliumnitrid (GaN) är två relativt nya material i elektroniksystem, som möjliggör en revolutionerande förbättring med lägre förluster och mer kompakta omvandlingssystem med hög verkningsgrad.
- Kraftelektronik handlar inte bara om att byta enskilda komponenter utan den stora vinsten uppstår när hela byggsättet angrips och utvecklas.

Bilden visar olika omvandlingar mellan t ex likström (DC) och växelströms (AC) – källor respektive belastningar.



Notera att en solpanel ger likström (DC), men i en fastighet är belastningen oftast förbrukare av olika typer

av växelströmsmaskiner (AC), som fläktar och vitvaror. Detta fodrar elektroniska omvandlare från DC till AC, en omvandling som exempelvis också används i fritidssammanhang i husvagnar och husbilar.

Användning och effektivisering av elektricitet är aktuellt i allt fler tillämpningar, exempelvis:

- Elektrifiering av transporter
- Industriella motordrifter
- Hissar och rulltrappor
- Transmissionstillämpningar
- Vitvaror
- IT och mobilkommunikation i basstationer, datacenter, laddare och trådlös laddning



Datorer och smart elektronik med inbyggd programvara för dessa ändamål utvecklas parallellt och blir ett allt vanligare verktyg i dessa sammanhang.

## Trådlös kommunikation

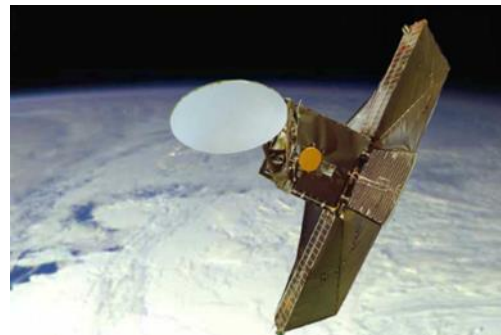
I ett par av våra uppdrag används trådlös kommunikation. Det kan vara att föra över programmet, hex-filen, mellan en iPad och ett microbit-kort. Det kan också vara att föra över information mellan två microbitkort. Det sistnämnda finns exempelvis med i uppdragen kring trafikljuset och vägbommen. Radioenheten använder samma antenn som vid överföring av hex-filer med Blåtand. Frekvensen är i båda fallen 2,4 GHz.

## Antenner för mikrovågs- och terahertzsystem (AMT)

Trådlös kommunikation och sensorer finns med i frekvenser från 0,3GHz till 300 GHz, därefter THz. Gigahertz (GHz) och Terahertz (THz) uttalas dessa frekvenser och antennerna för detta är utformade på olika sätt och storlek.

Några tillämpningar:

- Telekom: Basstationer, radiolänkar, multiantennsystem, satellitkommunikation
- Radarsensorer: Array-antennor för försvar och säkerhet
- Transport: Radar och fordonskommunikation
- Sensorsystem för inspektion, metrologi, vetenskap som kvantdatorer
- Medicinsk teknik
- Drivande av utvecklingen är idag kommunikationsteknologierna 5G (6G), IoT samt autonoma system med AI – programvara.



Svenska företag är världsledande i AMT-området för telekom, försvar och rymd. Många ny (små) företag har uppstått de senaste 10 åren.



## Inbyggda system och sensorer

Produkter med inbyggda system är ett flaggskepp för svensk ingenjörskonst. I Sverige finns det världsledande kunskap kring verktyg för utveckling av programvara till inbyggnadsprocessorer, så kallade *mikrokontroller*.

Ett exempel på en produkt med komplex funktionalitet är en robotgräsklippare. Data från många sensorer samlas in samt att hantera informationen från, för att sedan styra och reglera motorerna för drivningen. Dessutom trådlös kommunikation för att kunna köra och veta var den ska klippa, när den ska klippa, var den ska laddas osv.



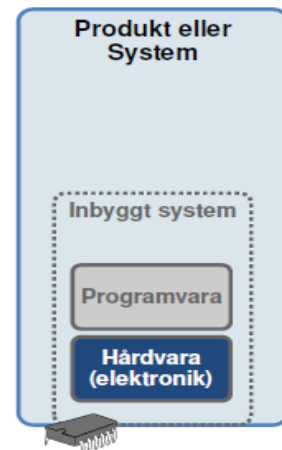
Ett inbyggt system innehåller typiskt en **mikrokontroller med programvara** som också kan **samverka** med andra system. Att samla in data från **sensorer, bearbeta och analysera data för att sedan** ta beslut om åtgärd är den traditionella systemuppbyggnaden. Resultatet kan vara att **aktivera** ett ställdon som en motor, kommunicera med andra inbyggda system på tråd eller trådlöst. Området är starkt kopplat till flera andra spetsområden inom den svenska elektronikbranschen och verksamheten i programmet Smartare elektroniksystem.

Processorn på micro:bit-kortet är en mikrokontroller och BBC i England har tillsammans med ingenjörer och programvaruutvecklare skapat elektronikkortet samt den utvecklingsmiljö som vi använder för programmering och simulering.

Det är många viktiga frågor att adressera runt inbyggda system:

- Tillförlitlighet och korrekt funktion
- Utvecklingsmetodik och verktyg
- Hantering av stora mängder data och information
- Olika typer av trådlös kommunikation
- Enkel hopkoppling av delkomponenter
- Energiförbrukning och laddning
- Systemdesign
- Sensorteknik
- Säkerhet
- Testning

Listan kan göras lång och innefattar ett tvärvetenskapligt kunnande!



## Sensorer och AI

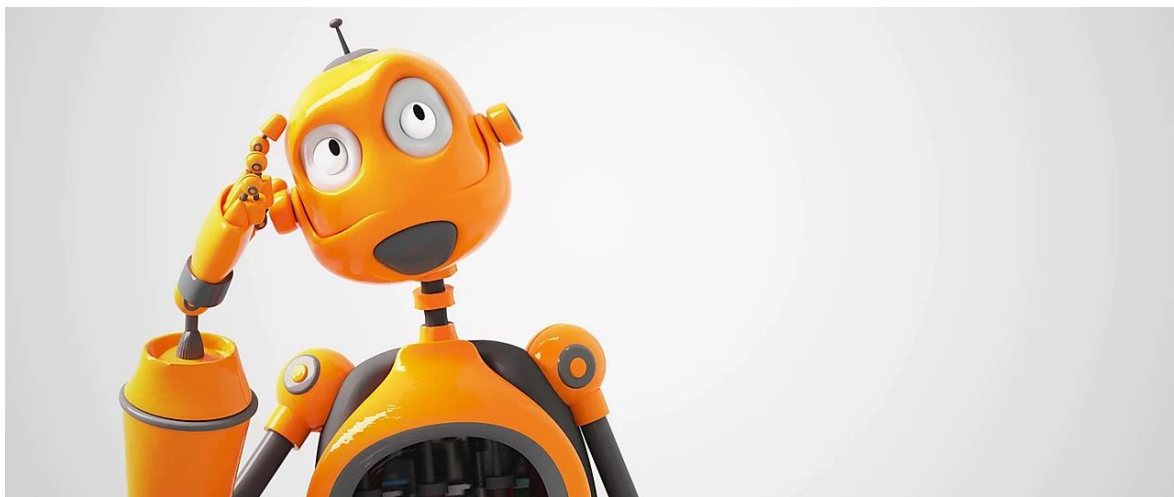
Antalet sensorer har fullkomligt exploderat och drivs bl.a. av mobiltelefonutvecklingen och teknologiområdet **Internet of Things (IoT)**. Med femte generationens (5G) mobilsystem kommer antalet uppkopplade enheter med sensorer att öka än mer då säkerheten och snabbheten i system även öppnar för industriella tillämpningar.

Alla sensorer och uppkopplade system ger tillgång till en oerhörd mängd data från produkter och processer. Denna digitalisering möjliggör en förflyttning från att sälja produkter till att sälja lösningar eller tjänster.

Insamling av och analys av stora datamängder öppnar också för det som kallas Deep learning, Machine learning och AI (Artificiell Intelligens).

Utvecklingen med allt fler sensorer och insamling av data från dessa påverkar inte bara tekniken utan det innebär förändring av affärsmodeller och hur vi människor

hanterar och förhåller oss till den nya tekniken. En viktig synpunkt är att dock att tekniken inte ska ta över beslut ur människans händer i en alltför omfattande utsträckning. Den ska vara ett verktyg, ett hjälpmedel!

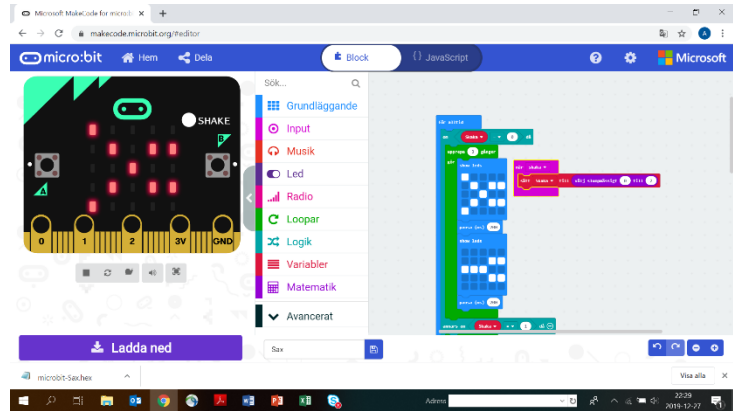


## Sammanfattning

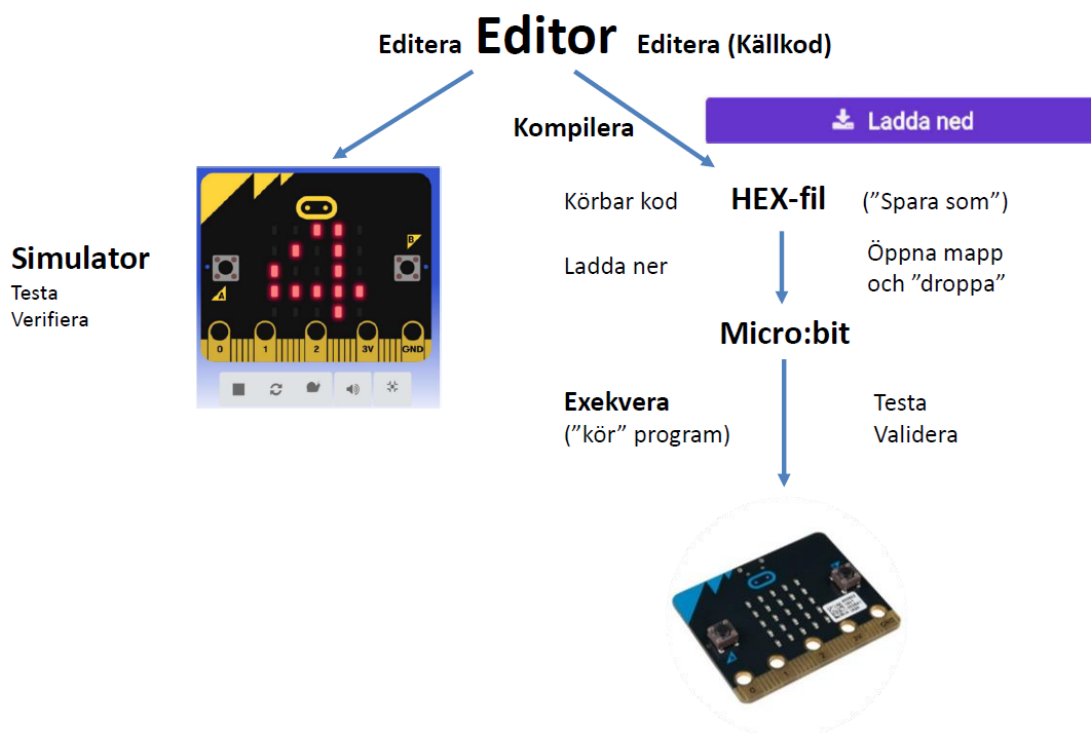
### Utvecklingsmiljön för Micro:bit

I temat Smartare Produkter arbetar eleverna med programvaruutveckling i en utvecklingsmiljö med blockprogrammering för en mikrodatator. Utvecklingsmiljön består av några olika verktyg med tillhörande möjligheter:

- **Editor** – den högra delen av fönstret där programmet har sin kod. I detta fall är koden i form av funktionsblock. Koden är den som programmeraren gör och ser och kallas ibland för källkod.
- **Blockkategorier** – rubriker i mittendelen av fönstret där programmeraren väljer block för programmet som ska utvecklas.
- **Simulator & Debugger** – i den vänstra delen av fönstret där koden kan testas (verifieras). Simulatoren har sina begränsningar när man kopplar in extern elektronik till Micro:bit. Via simuleringsfönstret kan man klicka på "nyckelpigan" för att komma till ett nytt fönster, Debug mode, där programmet/koden kan undersökas och testas ytterligare. Den fungerar i flera av de program som ska utvecklas under temat Den smartare staden och är ett mycket bra hjälpmedel i ett par av uppgifterna, exempelvis det med fallarm (äldreomsorg) samt uppgiften med kylkedjan.
- **Ladda ned** – knappen nere till vänster i fönstret, med vilken man startar processen för att få en exekverbar (=körbar) kod. Koden kompileras och en sk Hex-fil skapas. Hex-filen kan laddas ner ("droppas") till Micro:bits flashminne och är en kod som Micro:bits processor "förstår". Hex är förkortning av hexadecimal och är ett sätt att komprimera binär kod.



Se bilden nedan för process och terminologi – *Från källkod till exekverbar kod*



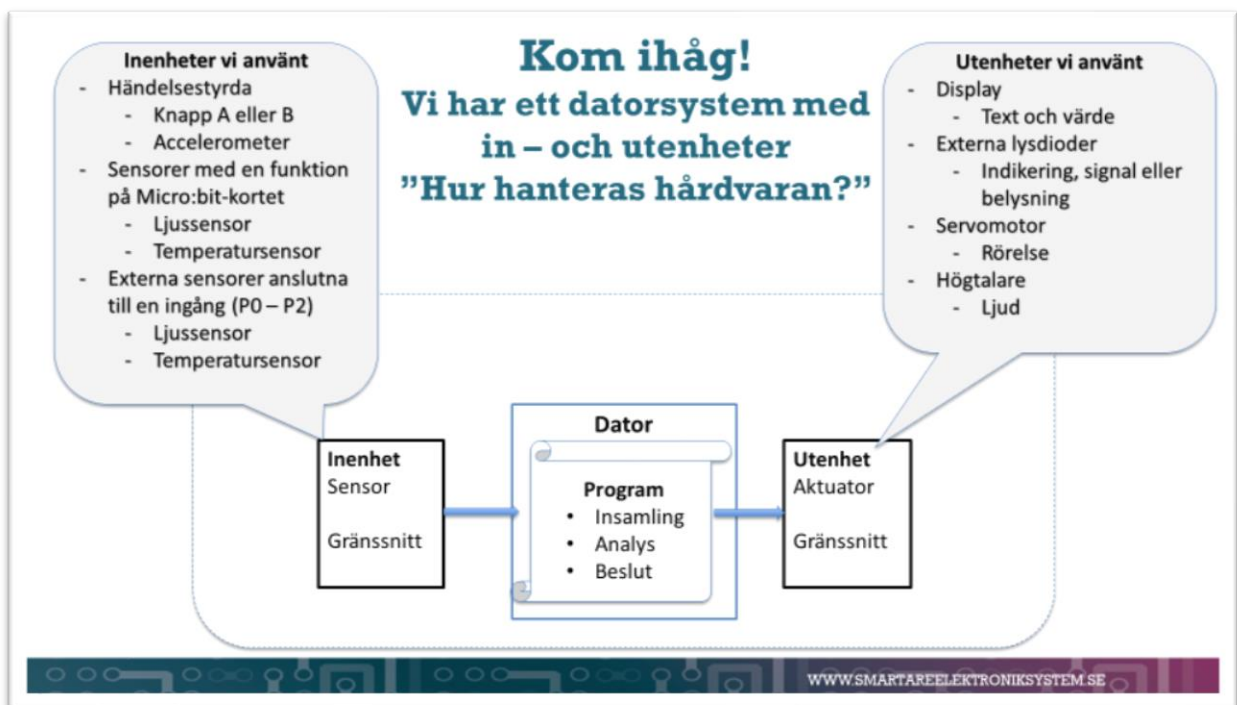
## Hårdvarunäva programmering

I uppdragen i Den smartare staden används det som kalla hårdvarunära programmering. För att ett applikationsprogram med hårdvara ska bli bra krävs det flera kompetenser av programmeraren.

- Dels är det bra om programmeraren förstår applikationen, vad ska programmet användas till och till vilken nytta?
- Dels val och egenskaper hos hårdvara som sensorer och aktuatorer samt deras gränssnitt mot processor/datorenhet.

Kunskaper i matematik, fysik används i såväl tekniken som i programmeringen och nyttan har ofta en tydlig samhällskoppling.

Vi kommer tillbaka till bilden av systemet och dess huvudblock. De inenheter (sensorer etc.) och utenheter (aktuatorer) vi använt i uppdragen ses sammanställda i bilden nedan.



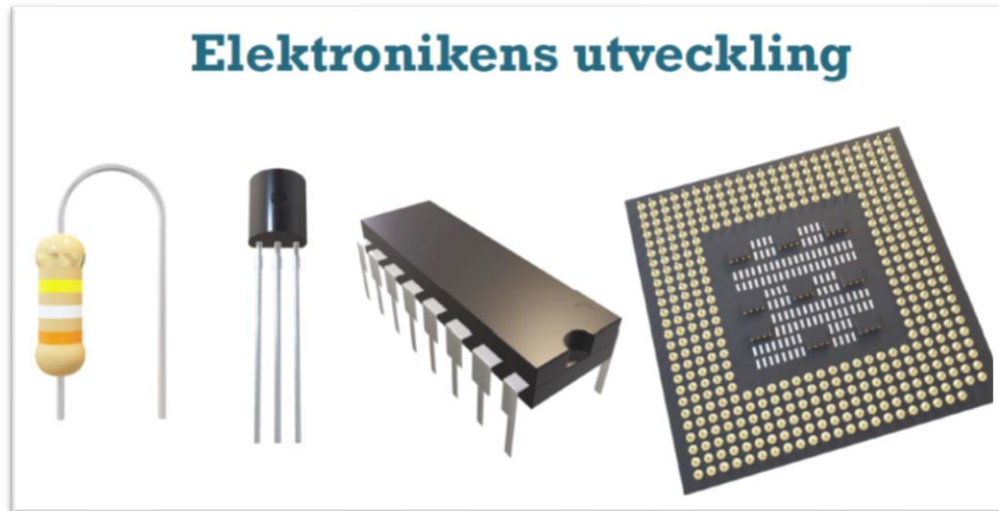
De två senaste sidorna i dokumentet visar på hur omfattande och komplext utvecklingen av elektroniksystem med dess programmering kan vara, men med Micro:bit som bas/plattform finns stora möjligheter att sätta sig in i området utan att det blir för svårt i första skedet.

## Extramaterial

Här följer lite extramaterial som tagits fram under utvecklingen av temat.

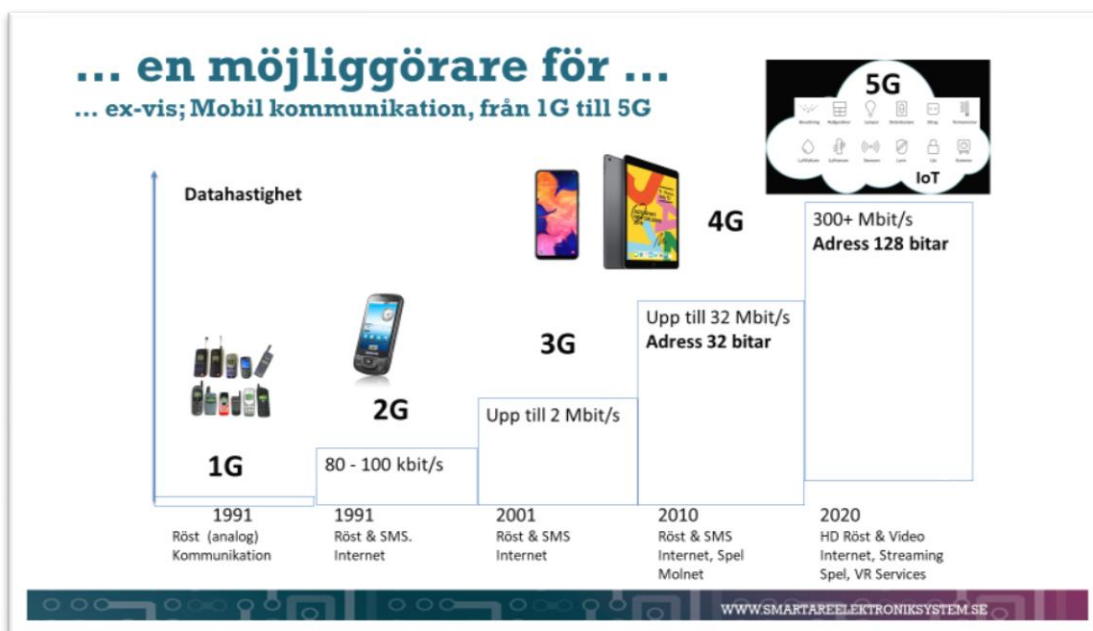
### Fakta om 5g och IoT

Sedan mitten av 50-talet har elektroniken utvecklats kraftigt, exempelvis från en transistor i en kapsel till flera miljarder transistorer i en integrerad krets.



Kom ihåg det som framhållits tidigare i temat, att elektronik är en möjliggörare att utveckla andra branscher med såväl i varor som i tjänster. Tillsammans med mjukvara är möjligheterna oanade. Detta har belyst tidigare och är som sagts på både gott och ont.

Exempelvis är elektroniken en möjliggörare för den snabba utvecklingen av internet och de parametrar som karakteriserar kommunikation nämligen hastighet, datamängd och adress till sändare/mottagare. I bilden nedan ses den mobila utvecklingen från teknologin 1G till 5G, vilket innebar att från att först kunna prata i en mobiltelefon kan vi idag överföra data för surf och streaming med hastigheter upp till 1 Gbit/s.



Tänk vilken inverkan internet har haft på våra liv. Världen gick på sätt och vis in i en ny tidsålder när vi blev uppkopplade. Det kan idag vara svårt att föreställa sig hur vår vardag hade fungerat utan möjligheten att skicka mejl, betala räkningar, följa nyheter, se play-TV och diskutera saker via internets olika sociala media.

### Internethastighet

För att kunna ta del av internet behöver vi någon form av internetanslutning. Idag finns fem vanliga sätt att ansluta till internet, nämligen via stadsnät, via telefonnät, via kabel-TV-nät, via fiber eller via mobilnätet. Det som skiljer mellan anslutningssätten är hur snabba uppkopplingar de kan leverera och vilken hårdvara som krävs för att ansluta. Utöver dessa sätt går det att ansluta till internet via uppringt modem och via satellit.

Enheten *b/s* eller *bps* (bits per second) används för att mäta internet-uppkopplingens hastighet. Den anger hur många bitar (ettor och nollor) som exempelvis en dator kan skicka eller ta emot per sekund. Eftersom antalet bit per sekund är väldigt snabbt anger vi hastigheterna i megabit per sekund eller kanske gigabit per sekund.

### Terminologi

Ett antal bits (b) kan räknas om till bytes (B), vilket är enheten som används för att mäta lagringskapacitet. 8 bitar motsvarar 1 byte (8 b = 1 B), och med en internethastighet på 24 Mb/s går det under optimala förhållanden att få en nedladdningshastighet på 3 MB/s. I tabellen ses hastighet och dess prefix.

Enhet	Innebörd
b/s	bitar per sekund
kb/s	tusentals bitar per sekund
Mb/s	miljontals bitar per sekund
Gb/s	miljardtals bitar per sekund

### Ny standard för IP-adresser på gång

Det råder ingen tvekan om att internet har vuxit till en storlek som ligger långt över de första förväntningarna. Denna tillväxt har skapat ett problem då det finns en begränsning i hur många enheter som samtidigt kan vara anslutna till internet. Varje "dator" ha en unik IP-adress, (IP betyder på svenska Internet Protokoll).

### IPv4

Idag använder vi IP version 4 (IPv4) som adresserar datorer med 32-bitars adresser. Varje adress består alltså av 32 stycken ettor och nollor, vilket räcker till drygt fyra miljarder adresser ( $2^{32}$  adresser). Det räckte gott och väl när protokollet skapades i början av 80-talet, men det är trots allt inte ens en IP-adress per person i världen.

## Rekonstruktion

Nu är en större rekonstruktion av hur internet är uppbyggt nödvändig. Iana (Internet Assigned Numbers Authority) som styr tilldelningen av IP-adresser globalt, reserverade i början av 2011 de sista uppsättningarna med IPV4-adresser. Det innebär inte att alla IP-adresser är slut, eftersom operatörerna och dess regionala tilldelare fortfarande har IP-adresser "i lager". När alla dessa är upptagna kommer det dock inte gå att få fram fler nya. Taket är nått.

### Ny standard - IPv6

IPV6 är ett helt omarbetat protokoll för kommunikation i nätverk. Det löser många problem, framförallt adresseringsproblematiken. Till skillnad från IPV4 använder IPV6 adressering med 128 bitar. Det ger möjlighet till  $2^{128}$  unika adresser, vilket idag skulle räcka till över 48 611 766 702 991 209 066 200 000 000 adresser per person i världen. Det bör räcka ett tag framöver.

### IoT - Internet of Things

Om alla prylar ska kunna kopplas till internet krävs det en kraftigt utökat antal bitar i adressering! Detta är ett av motiven och drivkrafterna i utvecklingen av 5G ... och senare 6G.

I bilden nedan ses ett antal exempel på vardagsprylar och situationer som kan tänkas vara uppkopplade. Här finns det ett antal frågeställningar att diskutera med eleverna, allt från tekniklösningar och säkerhet till problem och synpunkter kring etik och moral!

## Smartare system

Sakernas internet  
"Internet of Things (IoT)"



Ett exempel kan vara batteridrivna sensorer för kunder inom avfallshantering och återvinning. Sensorerna vid avfallsstationen kan hjälpa till med att samla information om antalet besökare, besökarnas beteendemönster, väder och behållarnas användning. Med hjälp av data kan återvinningscentralerna optimera sina processer och resurser, automatisera operationer och planera förebyggande underhåll.

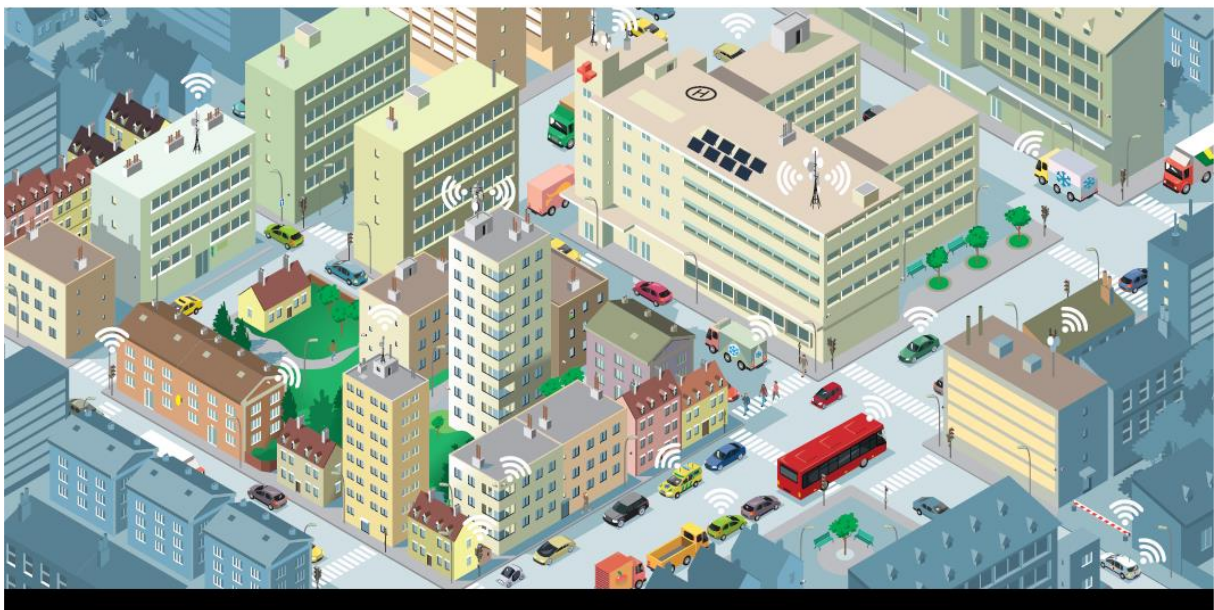
Idégivaren och entreprenören säger "– Enligt FNs mål för hållbar utveckling strävar vi efter att minimera slöseriet med resurser och hjälpa industri- och serviceföretag att fatta smarta och hållbara beslut. De industriella IoT-lösningarna och tjänsterna som vi tillhandahåller kan mäta många typer av data som kan användas för att analysera användning, beteendemönster och effektivitet i våra kunders verksamhet. Med denna information kan de optimera sina resurser, minimera miljöpåverkan och spara sina kostnader".

## Diskussionspunkter

Att hitta tekniska extrauppdrag eller projekt är inte så svårt när det gäller Micro:bit och dess programmering. Däremot kan det finnas andra aspekter att inventera där uppdragen är mer tvärvetenskapliga eller tvärdisciplinära. Det öppnar för samarbete med andra ämnen på skolan eller varför inte kontakt och samarbete med något annat land kring ett gemensamt intresse, exempelvis hållbar utveckling i uppbyggnaden av en samhällsfunktion.

### Tekniska system i staden

Vilka system kan finnas i Den smartare staden – se bilden för att hitta några ingångar till detta. Identifiera system för att sedan inventera behov, göra specifikationer osv hos dessa ur ett samhällsperspektiv.



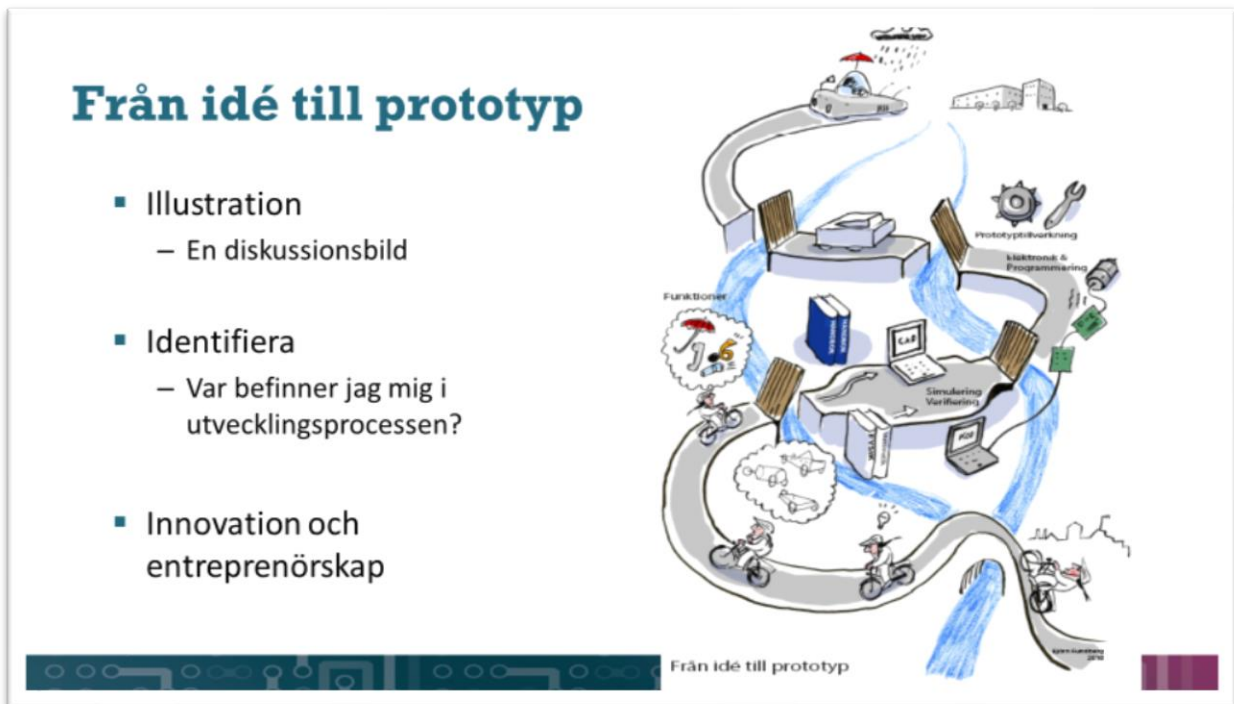
### Globala mål och intressen kontra Micro:bit

Användandet av Micro:bit och uppdrag kring denna är globalt. Nedan en bild från Venezuela med ungdomar och lärare som utbildar sig i programmering mm.

**Idé:** Möjlighet att få kontakt med andra länder / ungdomar och kanske stärka samarbetet med andra ämnen på egna skolan.



## Innovation och entreprenörskap



Bilden visar processen att gå från idé till prototyp. För att sätta en produkt på marknaden krävs oftast att man får fram en prototyp, eller varje fall en demo, som gör att man kan kommunicera med andra som kan ha intresse av att vilja delta och kanske hjälpa till.

Det räcker inte bara med en Powerpoint-presentation för att få upp detta intresse, man behöver kunna visa konkret på vad ny teknik kan innebära för mervärde i sammanhanget.

Utveckling av en produkt med elektroniksystem kräver en process att arbeta efter, med tydliga avstämningpunkter i en uppsatt projektplan.