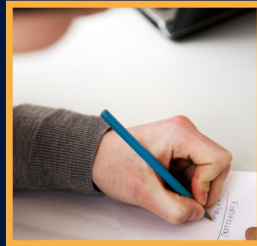
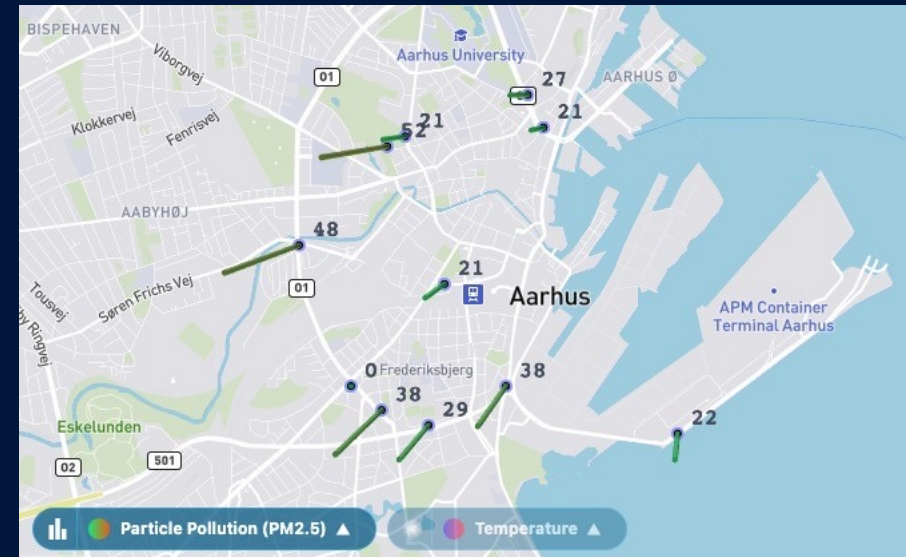
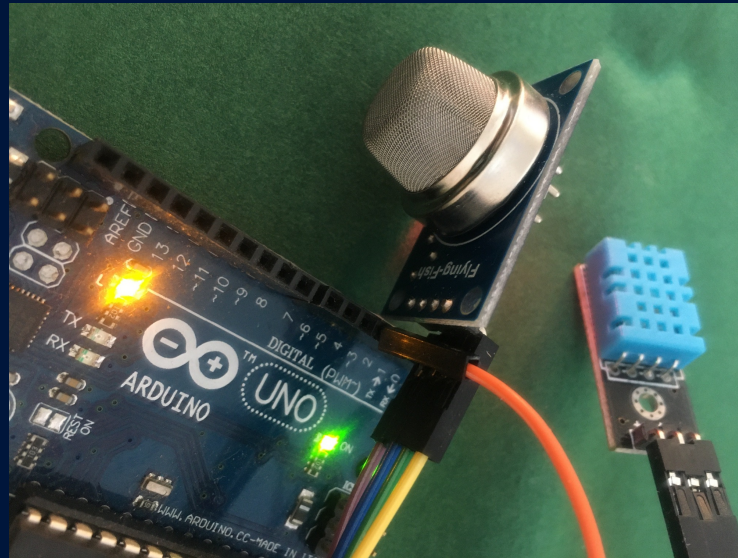


COPENHAGEN BUSINESS ACADEMY



DATA ENGINEERING



What is the Internet of Things

All computing devices that are **connected** to the internet

These devices are embedded with **environmental sensors** and other technologies that enable them to collect and exchange data **without human intervention**

Edge computing is about moving data processing from those data centers to devices at the network edge.

devices can be programmed to take their own actions or respond to events happening in their environments

Challenges to IoT Adoption

To make the IoT successful, organizations need to **integrate** various disparate applications, data, systems, people, and sensors.

Furthermore, once everything is connected, they need a way to **analyze and track** all the data coming from their devices

Capabilities needed to implement IoT

- Cybersecurity
- Integration
- Analytics
- Network and communications
- Data management
- device management
- App development

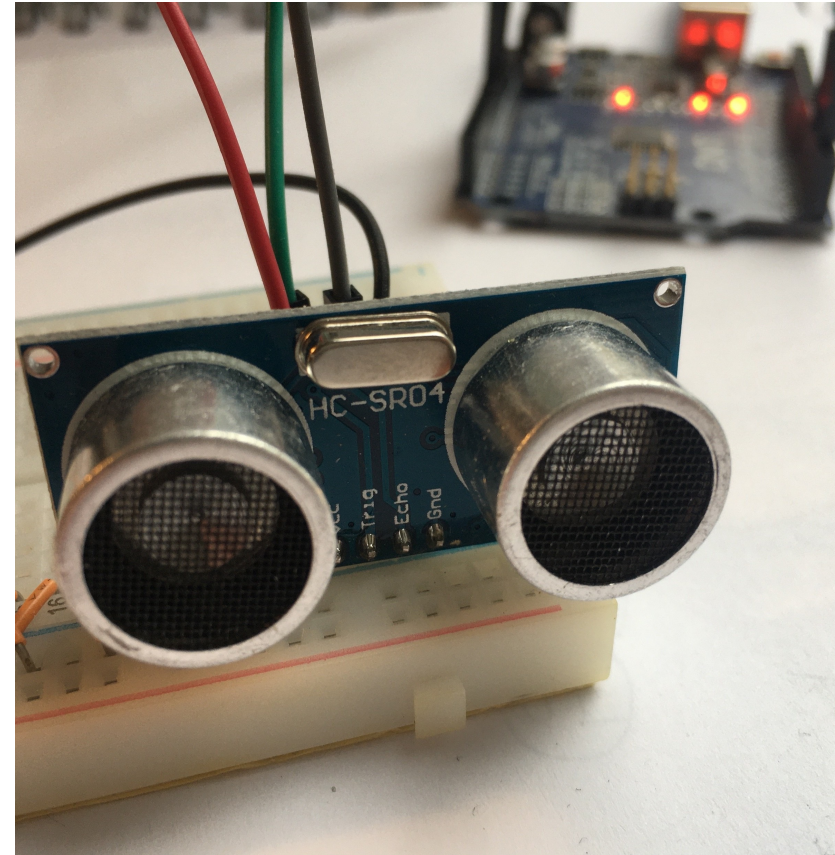
What is IoT Analytics?

A data analysis tool that assesses the wide range of data collected from IoT devices. IoT analytics assesses vast quantities of data and produces useful information from it.

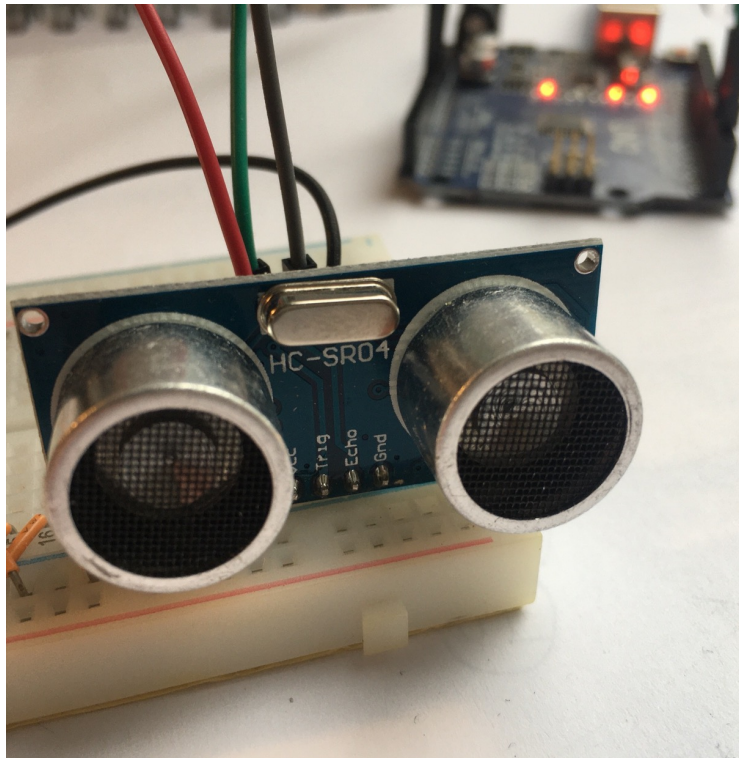
- 1.The first step is **to collect data aggregated** from a variety of sources, in a range of formats, and at multiple frequencies.
- 2.This data is then **processed** with a wide range of external sources.
- 3.The information is then **stored in a time-series** for analysis.
- 4.The **analysis** can be done in multiple ways--with custom analysis systems, with standard SQL queries, or with ML analysis techniques. The results can be used to make a wide range of predictions.
- 5.With the information received, organizations can build several systems and applications to ease business processes.

Den hvide bygning – Data-retrieval

- Design af IO-device
- Kodning af device
 - Arduino
 - R serial
- Analyse af data
- Design et API til forespørgslen: hvor mange er i bygningen <yyyy-mm-dd hh:mm>

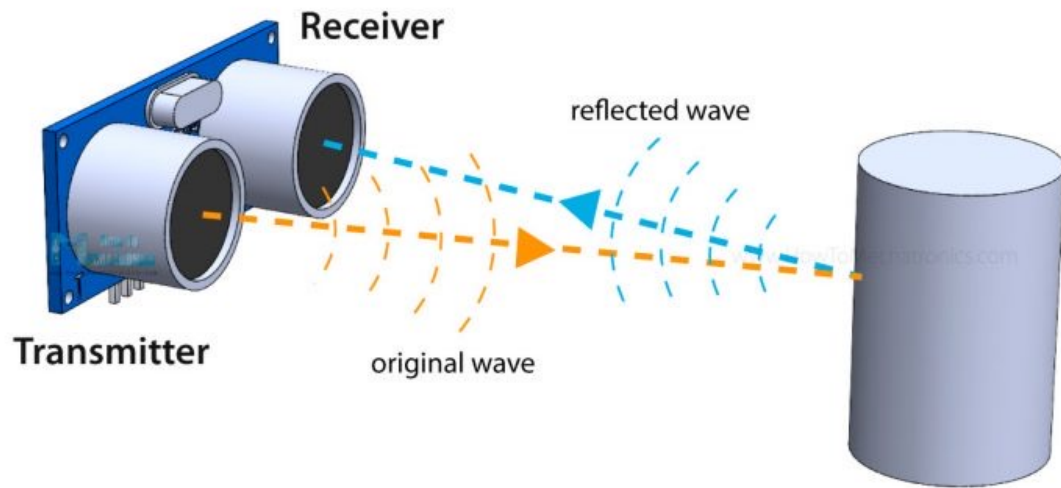


Den hvide bygning – Data-retrieval via USB

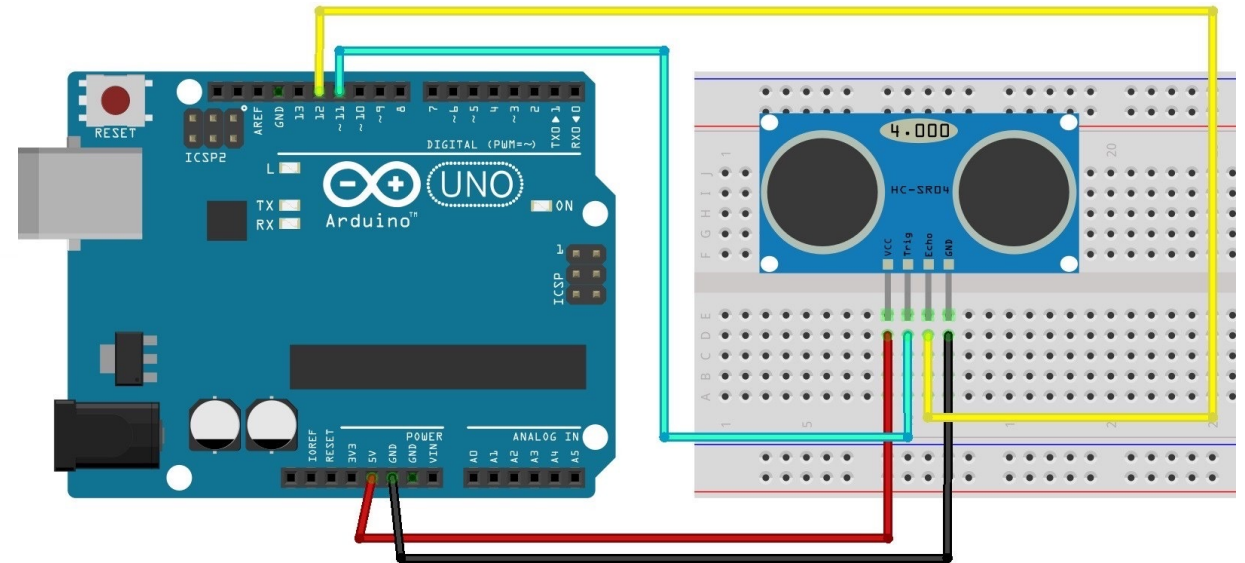


- ▶ Computer
- ▶ Disk drives
- ▶ Display adapters
- ▶ DVD/CD-ROM drives
- ▶ Human Interface Devices
- ▶ IDE ATA/ATAPI controllers
- ▶ Jungo
- ▶ Keyboards
- ▶ Mice and other pointing devices
- ▶ Monitors
- ▶ Network adapters
- ▶ **Other devices**
 - ▶ **Arduino Uno**
- ▶ Ports (COM & LPT)
- ▶ Processors
- ▶ Sound, video and game controllers

HC-SR04



- Plug device og ledninger
- Plug device til PC
- Åben Arduino
- Vælg board (Uno) og port



HC-SR04 - Arduino

sketch_dec1a.ino

```
1 // global vars
2 int tp = 2;
3 int ep = 3;
4
5
6 void setup() {
7   // put your setup code here, to run once:
8   pinMode(tp,OUTPUT);
9   pinMode(ep,INPUT);
10  Serial.begin(9600);
11
12 }
13
14 void loop() {
15   // put your main code here, to run repeatedly:
16
17 }
```


HC-SR04 - Arduino

```
14
15 void loop() {
16   // skriv til output-pin - sluk - vent - åben - vent - sluk
17   digitalWrite(tp, LOW);
18   delay(2);
19   digitalWrite(tp, HIGH);
20   delay(10);
21   digitalWrite(tp, LOW);
22
23   duration = pulseIn(ep, HIGH);
24
25
```

The screenshot shows the Arduino.cc website interface. At the top is a teal navigation bar with links for PROFESSIONAL, EDUCATION, and STORE. A search bar is on the right. Below this is another teal bar with the Arduino logo and links for HARDWARE, SOFTWARE, CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. A language selector indicates the page is available in 2 other languages besides English. The main content area shows the breadcrumb path: Reference > Language > Functions > Advanced io > Pulsein. The function name **pulseIn()** is prominently displayed, followed by the tag [Advanced I/O]. A left sidebar contains a table of contents with links to LANGUAGE, FUNCTIONS (which is highlighted), VARIABLES, and STRUCTURE, as well as LIBRARIES, IOT CLOUD API, and GLOSSARY. The main text area begins with a 'Description' section, explaining that the function reads a pulse (either HIGH or LOW) on a pin and returns its length in microseconds.

PROFESSIONAL EDUCATION STORE

Search on Arduino.cc

SIGN

HARDWARE SOFTWARE CLOUD DOCUMENTATION COMMUNITY BLOG ABOUT

This page is also available in **2 other languages** Change language English

LANGUAGE
FUNCTIONS
VARIABLES
STRUCTURE

LIBRARIES
IOT CLOUD API
GLOSSARY

Reference > Language > Functions > Advanced io > Pulsein

pulseIn()

[Advanced I/O]

Description

Reads a pulse (either HIGH or LOW) on a pin. For example, if value is HIGH, pulseIn() waits for the pin to go from LOW to HIGH, starts timing, then waits for the pin to go LOW and stops timing. Returns the length of the pulse in microseconds or gives up and returns 0 if no complete pulse was received within the timeout.

The Arduino Reference text is

HC-SR04 - Arduino

```
15 void loop() {  
16   // skriv til output-pin - sluk - vent - åben - vent - sluk  
17   digitalWrite(tp,LOW);  
18   delay(2);  
19   digitalWrite(tp,HIGH);  
20   delay(10);  
21   digitalWrite(tp,LOW);  
22  
23   duration = pulseIn(ep, HIGH);  
24   distance = (duration*0.034)/2;  
25  
26   Serial.println(distance);  
27   delay(100);  
28 }  
29
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on '/dev/cu.usbmodem146101')

.
9
12
16
13

HC-SR04 – Arduino & R

```
1 library("serial")
2
3 conw <- serialConnection(
4   name = "Arduino",
5   port = "cu.usbmodem146101",
6   mode = "9600,n,8,1",
7   newline = 1,
8   translation = "auto",
9   handshake = "xonxoff",
10  buffersize = 4096
11 )
12
13 open(conw)
14
15 stoptime=Sys.time()+10
16
17 while (Sys.time() < stoptime) {
18   tmp=read.serialConnection(conw)
19   print(tmp)
20   Sys.sleep(1)
21 }
22 close(conw)
23
```

22:1 (Top Level) ▾

Console Terminal x Background Jobs x

R 4.1.1 · ~/ ↗

```
3913813827812808278513828122801277827719162137
1205206205139205206206207206138928148136910987
193193119"
[1] "9101941959195194193193"
[1] "19319419388888"
[1] "987777777"
[1] "777777777"
```

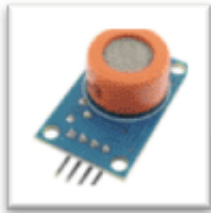
21:3 (Top Level) ▾

Console Terminal x Background Jobs x

R 4.1.1 · ~/ ↗

```
+ }
[1] "\n6\n6\n6\n6\n6\n6\n6\n6\n6\n6\n6\n6\n6"
[1] "\n6\n6\n7\n6\n6\n6\n6\n6"
[1] "\n6\n6\n6\n6\n6\n6\n6\n6\n6"
[1] "\n6\n6\n6\n6\n7\n6\n7\n6\n6"
[1] "\n6\n6\n6\n6\n6\n6\n6\n6\n6"
[1] "\n6\n6\n6\n6\n6\n6\n6\n7\n6"
[1] "\n6\n6\n6\n6\n6\n6\n6\n6\n6"
```

Sensors



Alcohol Sensor



Ultrasonic Sensor



IR optical Sensor



LDR Sensor



Gas Sensor



Gyroscope Sensor

Different types of Sensors



Rain Sensor



Sense Hat



Photo Diode



IR proximity
Sensor



Proximity Sensor



PIR Sensor