DIGITIZING PRODUCTS:
CREATING DEMONSTRATORS
FOR FUTURE EDUCATION

Snow depth sensor

Demonstrator classification and documentation

Kongsberg, October 2021

Co-funded by the
Erasmus+ Programme
of the European Union

Project consortium

Dissemination level

Code	Access granted to
PU	Public
PP	Restricted to other programme participants (including the Commission Services)
CO	Confidential, only for members of the consortium (including the Commission Services)

Legal Disclaimer

About the DigiDemo project

Environmental challenges and digital transformation are two of the main drivers changing the world and the way business will be is done in the future. Therefore, it is essential to enable future employees to address these drivers. The skills and competences needed to develop digitalized products and awareness of the environmental challenges are therefore crucial for the European workforce and industry to continue being competitive in a future green economy and to maintain jobs across Europe.

The DigiDemo project addresses these challenges by developing demonstrators especially for higher education allowing to improve mainly mechanical engineering studies by integrating skills and competences allowing them to understand, develop and commercialise connected products. The results will be publicly available and can be used by every institution interested in integrating this type of training in their cursus.

Content

Document authors

	First name Last name	Institution
Key author	Ole Martin Horne	FIV (FTO)
Further authors	Håkon Teigen Røed	FIV (FTO)

Revision history

Version	Date	Author(s)	Description
1.0	2021-09-05	Horne /Røed	Initial draft
1.1	2021-09-08	Horne /Røed	Design documentation
1.1	2021-09-17	Horne /Røed	Software and hardware documentation
1.2			
1.3			

Document status

Status description
Draft Version

Abbreviations

ESTA	ESTA Belfort (France)
FHV	Fachhochschule Vorarlberg (Austria)
FIV	Fagskolen i Viken (Norway)
UCN	University College Nordjylland (Danmark)
UEMR	Universitatea "Eftimie Murgu" din Resita (Romania)

List of figures

Har ikke fundet nogen tekst til listen over figurer.

List of tables

1 Elements of a Demonstrator

The purpose of this document is to describe the deliveries, scope, and responsibilities of developing the Snow Depth Sensor. The document will also describe the technical, functional, and architectural requirements.

The goal is to develop and build a physical demonstrator. The demonstrator will measure the snow depth and communicate the level to the user. It will be designed and built in such a way that it utilises a full IoT architecture enabling students to develop their skills and competencies.

The snow depth demonstrator is developed to measure current snow depth on cabin roofs by a laser distance sensor. The responsible owner of the cabin is notified by an app for smartphone.

1.1 Overview

The key properties of the focus project are:

Table 1: Specification of key properties of the focus project

Key Property	Value
EQF level	4 and 5 (Technician)
Year of study	2
Domain	Mechatronics
Objective	Hands-on
Workload	Depending on setup
Keywords	Internet of Things (IoT), C++/MCU programming, Microcontroller, WEB services

The demonstrator is suited for education on vocational college level. The demonstrator, shown below allows for teaching within the following subjects:

- IOT
- C++ programming
- Microcontrollers
- WEB services

Technical requirements

- Utilise a full IoT architecture including the edge, fog, and cloud layers.
- A microcontroller measures the bilge water level.
- The distance sensor is a Garmin Lidar-lite v3.
- A busser used as a bilge pump “simulator”
- Mobile communication.
- MQTT used as the communication protocol.
- Mosquitto broker used as the MQTT broker service.
- Node-Red receive the data and to prepare and display to the end user.

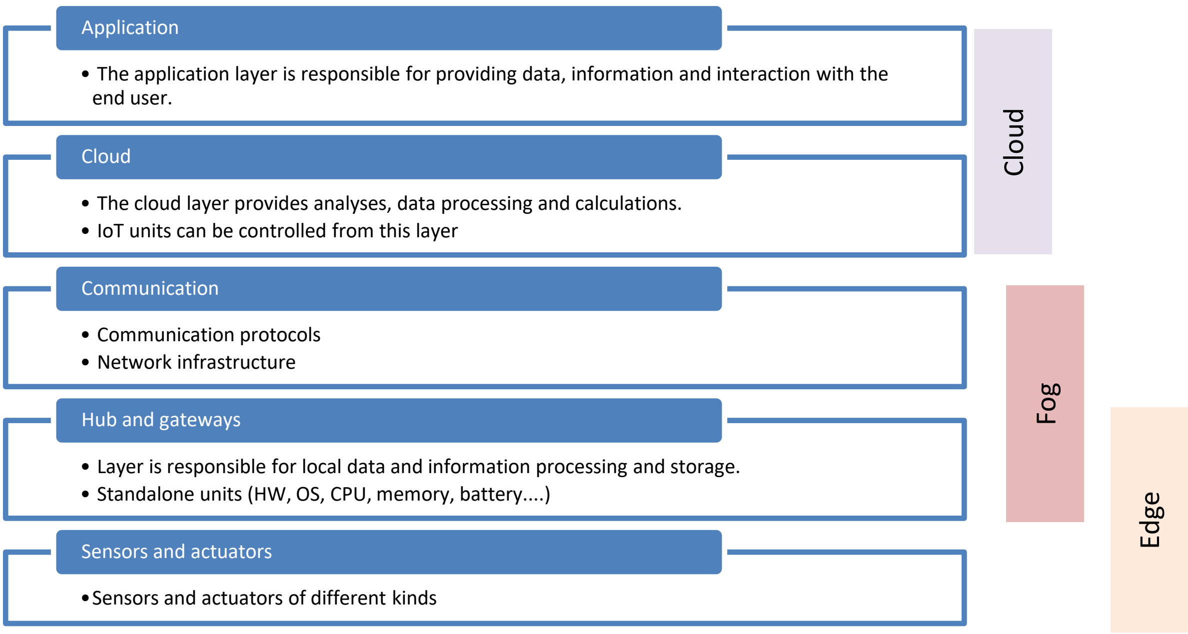
Functional requirements

- Surface level adjustable to simulate snow depth.
- Surface level presented on a web page to the end user.
- The demonstrator is easy to transport and store.
- Students will be able to do the wiring of the demonstrator
- Students will be able to do the programming of the demonstrator
- Students will be able to install and configure the infrastructure necessary to run the demonstrator.

The design includes all elements in the demonstrator architecture according to a generic IoT architecture.

IoT architecture

An IoT architecture can be viewed in several different ways and scenarios. Before we continue to discuss the architecture of the actual demonstrator it is necessary to understand the architecture from a principal point of view.



At the bottom we find the sensors and actuators. Basically, we can say a conversion between physical and electronical units are taking place here. As an example, we can measure movement, pressure, temperature, distance and so on. Or control motor and other types of actuators.

Hub and gateways are units that gathers information from the sensors or controls the actuators. They also enable communication between the sensor layer and the communication layer. Encryption and decryption are also a responsibility of these units to ensure a safe and secure communication.

The communication layer provides connectivity between the different hubs and gateways. In addition, it connects the units to the cloud layer. Here we find all the different communication technologies and protocols. Technologies and protocols can consist of Ethernet, WiFi, Bluetooth, 4G, 5G, Z-Wave, Zigbee, TCP/IP, MQTT and so on.

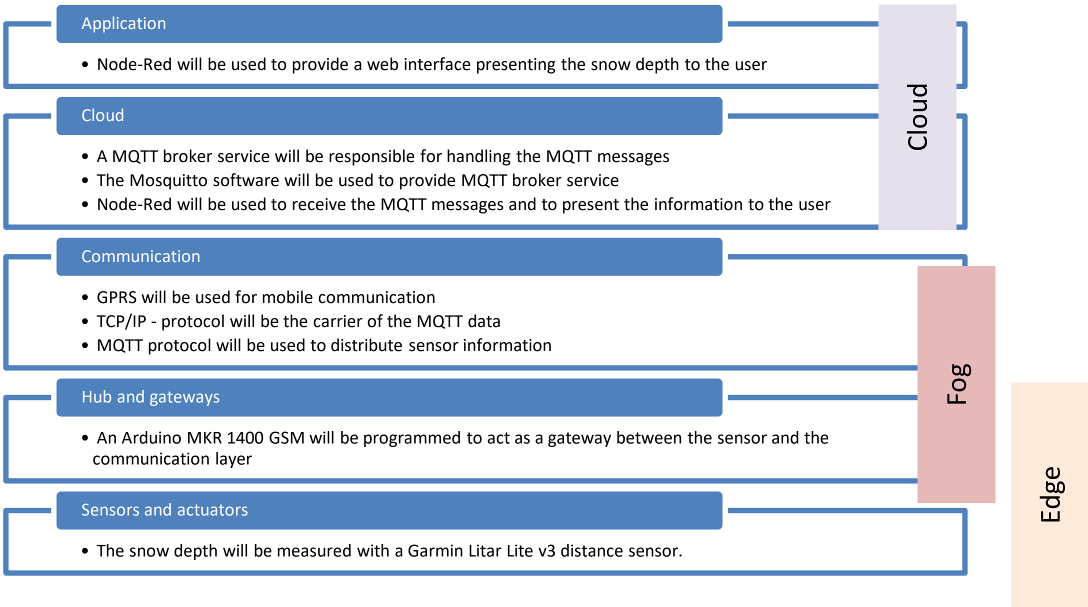
In the cloud layer we often find the large processing and storage capacities. Software services, data analyses, artificial intelligence, machine learning and so on can be found at this level.

On top we find the application layer. It is utilising the layers below to interact present information to the end users.

Demonstrator architecture

The demonstrator is designed according to the IoT architecture.

The table below show how each layer will be solved/delivered.

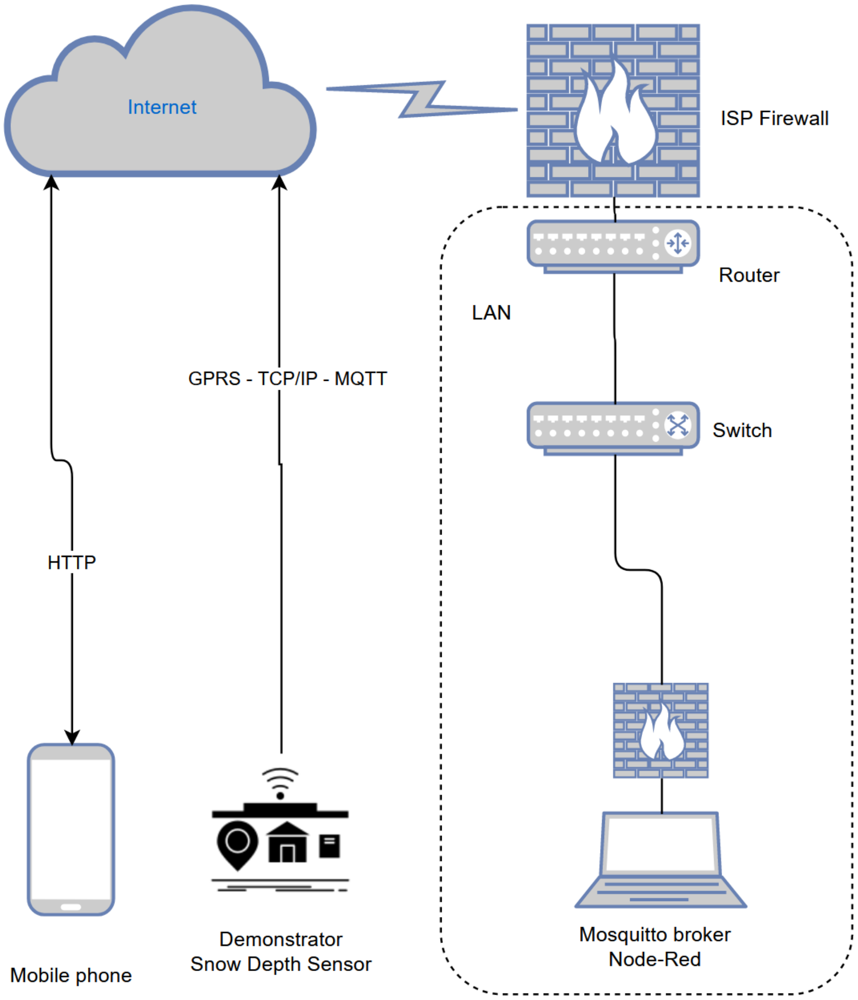


A laser distance sensor from Garmin (Lidar light v3) will be used to measure the snow depth. The sensor will be connected to an Arduino MKR 1400 GSM. The microcontroller will gather sensor data and distribute the measured values using the MQTT protocol. The program running on the microcontroller will be shared and documented to enable students to make their own adjustments and to understand how it works.

GPRS will be used on the microcontroller to provide connectivity. The MQTT protocol defines the message structure and format, and the data will be carried by the TCP/IP protocol.

In the cloud layer two different software packages will be used. The Mosquitto software will provide a MQTT broker service to manage the data messages and distribution, and Node Red will receive information, process it and display it to the user in the application layer.

Network design



The demonstrator will send the MQTT messages via the Internet and into the Mosquitto and Node-Red cloud services. MQTT is sent on IP port 1833 and Node-Red on port 1880. Hence it is necessary to open the ISP firewall for incoming traffic on these ports. The same applies to the computer firewall where the services are installed and running. The web interface for the end users can then be accessed from the LAN or externally on the Internet.

As an alternative the Mosquitto and Node-Red can be purchased as a cloud service from several vendors. Our approach, however, is to install locally and publish the services by ourselves on the Internet.

1.2 Description of fulfilment of keywords/characteristics

Table 2: Description of fulfilment of demonstrator characteristics for the focus project

Characteristic	Description
Teaching improvement	The demonstrator will improve teaching, by showing digitalization and value-adding of a well-known object.
Sustainability awareness	In this project the sustainability awareness is seen in a broader context, relating to World goal 4, Quality education.
Replicability	The demonstrator will be replicable, as most of the hardware will be based on readily available modules, and further instructions on system fitting will be provided.
Industry needs	
Interdisciplinarity	The demonstrator will be interdisciplinary, covering the following fields: • IoT architecture • Microcontroller GSM 1400 board • Antenna • SIM card with a data plan • Lidar-lite v3 • Solar panel • Battery

1.3 Classification according to the dimensions

Table 3: Classification of the focus project according to the dimensions

Dimension	Property	Value
Value chain	development	☑
	production	☑
	sales	☐
	after-sales-support	☐
	end-of-life	☐
Chain of technology	mechanical structure	☑
	sensors	☑
	electronic circuits	☑
	edge device	☑

Dimension	Property	Value
	data transmission	☑
	cloud	☑
Sustainability	energy reduction	☑
	material reduction	☐
	better materials	☐
	better production	☐
	reparability	☐
	recycling	☐
Physicality	physical setup	☑
	simulation	☐
Degree of student freedom	demonstrated	☐
	guided	☑
	coached	☑
	autonomous	☑
Transportability	fixed	☐
	transportable	☐
	portable	☑
Costs (implementation)	EUR	300 – 400
Costs (operation)	EUR	NA
Workload (implementation)	Hours	5-15
Workload (operation)	Hours	10-20
Size	m	< 1
Weight	kg	0,5-2
Special requests	no/yes, if yes: which	no

1.4 Educational information

Intended Learning Outcomes

The demonstrator is thought to be used over several teaching sessions with exercises. These sessions consist of 4 modules of 45 minutes each.

1.5 Organizational information

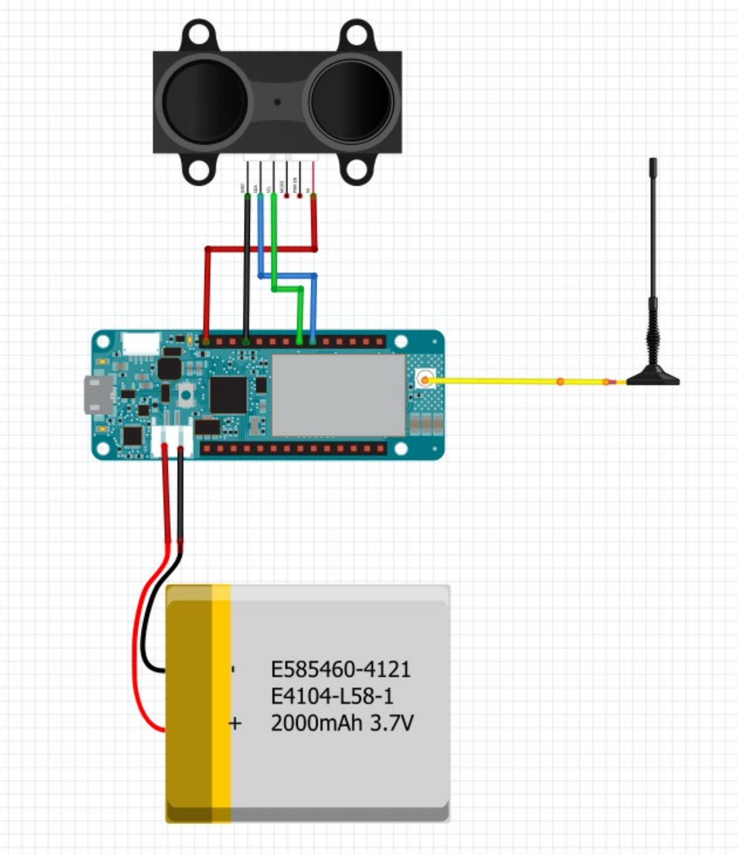
The following requirements apply.

- Project duration. *The optimal time frame for the project is four-five sessions depending on the student's level.*
- Team size. *The number of student team is three in each group.*

1.6 Description of the technology and the setup

The following describes the hardware and software needed to configure and run the snow depth sensor demonstrator. The presentation contains only universal user information, location specific information like firewall configuration and port forwarding is not described in detail.

Hardware



This demonstrator uses the following hardware:

Component	**Function**
MKR GSM 1400 board	Microcontroller with built inn battery management and GSM module.
Antenna	GSM antenna with micro UFL connector.
SIM card with a data plan	
Lidar-lite v3	Lidar module that runs on 5v and has I2C/PWM interface. I2C is used in this demonstrator.
Solar panel	Optional
Battery	Min 1500 mAh battery.

During cellular transmissions the peak current required by the board will exceed 500mA. This is more than what can be sourced by a standard USB port, so it is MANDATORY to have a 1500 mAh or higher LiPo battery plugged all the time, the current provided by the USB port will be supplemented by the battery.

Wire connections between LIDAR and Microcontroller:

- LIDAR-Lite 5 Vdc (red) to Arduino 5v
- LIDAR-Lite I2C SCL (green) to Arduino SCL
- LIDAR-Lite I2C SDA (blue) to Arduino SDA
- LIDAR-Lite Ground (black) to Arduino GND

Arduino software with libraries

Developer:	Arduino SRL
Website:	www.arduino.cc
Download link:	www.arduino.cc/en/software

Download and install Arduino IDE, the purpose of this application is to code and transfer the compilated software to the microcontroller board.

After the software installation some libraries need to be installed.

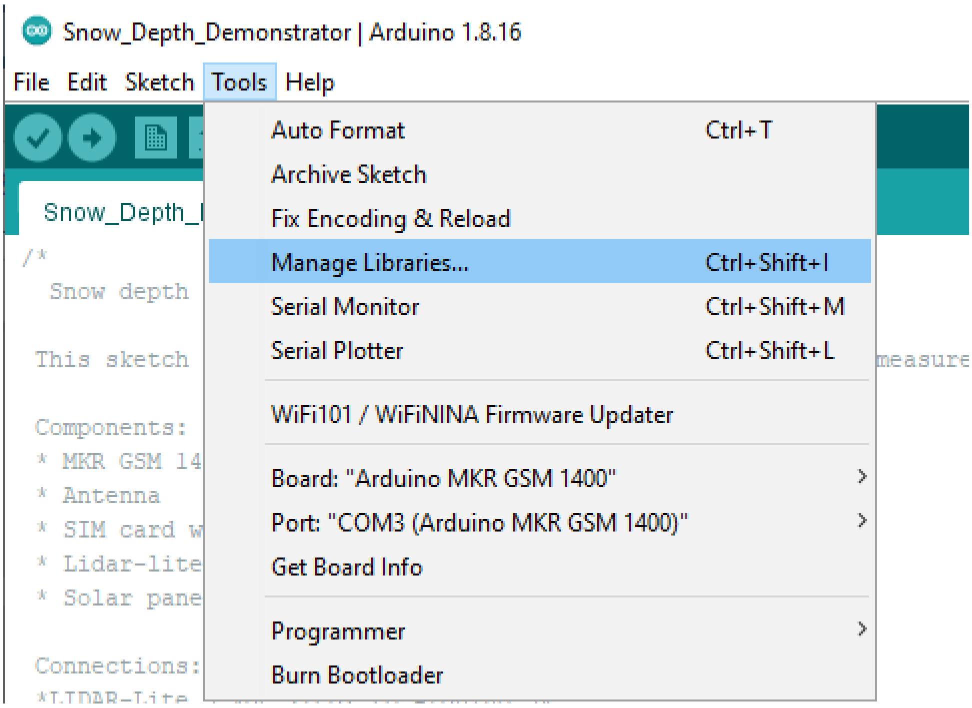


Figure 1 Library manager

Open the library manager and install the following libraries:

MKRGSM	Handles the GSM/GPRS connection
LIDAR-Lite	Controls the I2C communication to the Lidar sensor
MQTT	Handles the communication to the MQTT server

There is a bug in the LIDAR library, to fix this you need to edit the following file:

Documents\Arduino\libraries\LIDAR-Lite\src\LIDARLite_v3HP.cpp

After the #include section insert the following line: #define LEGACY_I2C

After all libraries are installed, you may open the Snow_Depth_Demonstrator.ino file and compile/flash the microcontroller.

Mosquitto MQTT broker

Developer:	Eclipse Foundation
Website:	https://mosquitto.org/
Download link:	https://mosquitto.org/download/

Download and install the Mosquitto MQTT server.

Remember to open port 1883 in firewall and/or router if applicable, no further configuration of the server is necessary.

Node red

Developer:	JS Foundation
Website:	https://nodered.org/
Download link:	https://nodered.org/docs/getting-started/windows

Download and install Node.js from nodejs.org.

Start a command prompt window and run the following command

```
npm install -g --unsafe-perm node-red
```

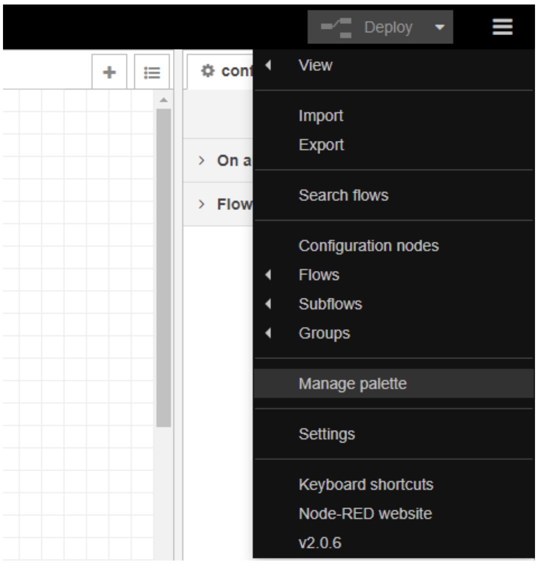
Once installed, the simple way to run Node-RED is to use the node-red command in a command prompt

```
C:>node-red
```

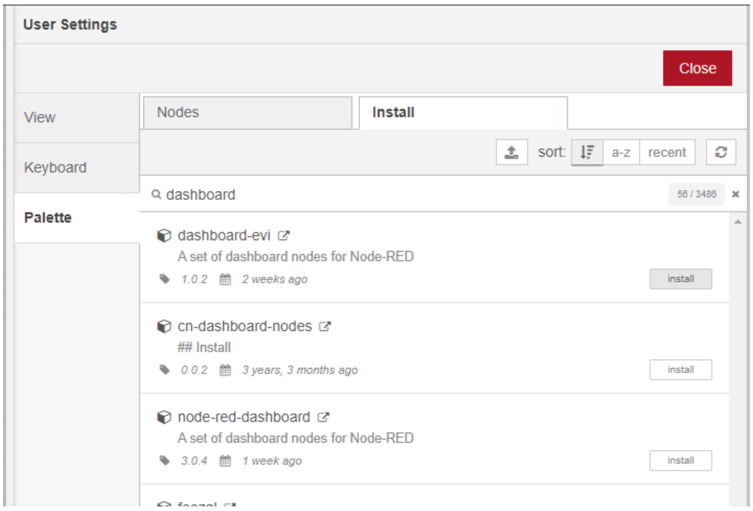
Remember to open port 1880 in firewall and/or router if applicable.

Type in http://localhost:1880/ in your standard webbrowser to open the Node Red workflow.

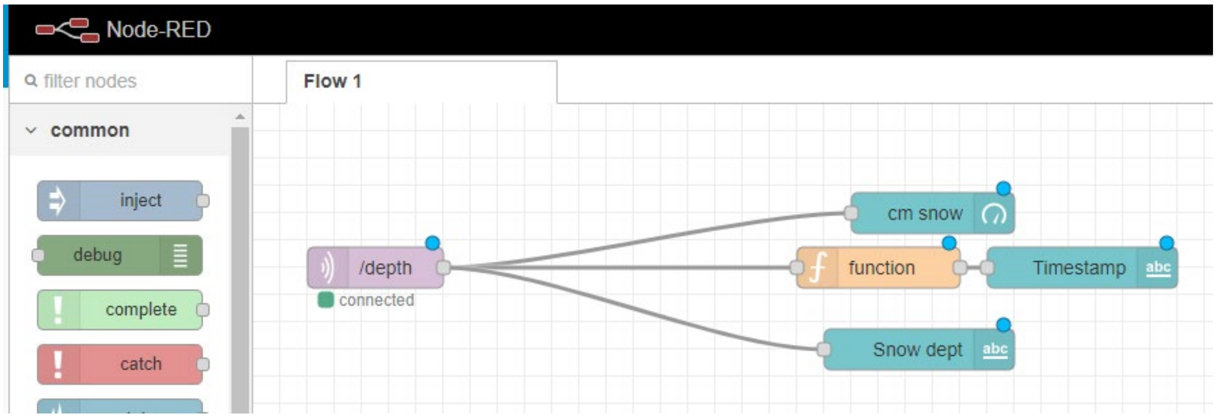
First we need to install an ekstra module with UI elements. On top right corner of screen select "Manage palette" from the drop down meny.



Select “Palette” and search for dashboard. Install the “dashboard-evi” module.



From the same dropdown menu select Import and import the Node_Red_Flow.json file. The main flow window will now have the project flow graph like below.



Select “Deploy” from top right corner and head over to http://localhost:1880/ui to see the flow graph running.

Bill of Materials

The following table will outline the materials needed to produce the demonstrator. Note this is the equipment for one group of SW developers and must be multiplied with the number of groups.

Table 4: Classification of the focus project according to the dimensions

	DIGIDEMO demonstrator: Snow depth monitoring	
System component	**Legend**	**Price*** €/pc
Snow depth sensor	SEN-14032 - LIDAR-Lite v3 avstandssensor, SparkFun Electronics	€200,00
Controller w/communication	ABX00018 + X000016 - Arduino MKR GSM 1400 + GSM Aerial	€75,00
Solar cell PS	PRT-16835 - Solcellepanel 10 W, SparkFun Electronics	€22,00
Charger/power supply	1944 - Lipo-lader med USB-forsterkning, Adafruit	€19,00
Battery	ICP622540PMT - Oppladbar batteripakke, Li-Po, 3.7V, 600mAh, Renata	€22,00
		€338,00
	*= Budget price incl. VAT (NOK)	

Cables for connection the sensors are not included in the embedded unit, as they are custom to the specific setup.

2 References