INNOWWIDE Call 1

Final Technical Report

Project ID: 2019-1120 INNOWWIDE

Project Acronym: CSF-Electric Statistic System

Organisation: CSF-ELECTRIC Kft.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement nº 822273. This document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

0. Publishable Summary

- A summary not exceeding 2500 characters with spaces, including description of the action context and objectives, of the work performed and the main results achieved. **This summary is publishable, so it must not contain confidential information or company secrets.**

PLEASE COMPLETE HERE (1 page)

The goal of this project was to create a user-friendly, easy to install, versatile PLC control, which is capable of using BAR codes for product tracking and updating warehouse inventory also. The main features of this development are the following:

- the system can be installed with our PLC control to any textile industry machine or machinery
- it can be installed to one machine and be synchronized to several machines later on
- it can make data analysis of daily/weekly/monthly production and based on these reports the production could be optimized thus making it possible to create more products in less time
- BAR code-based system which can facilitate the monitoring and update of the warehouse inventory
- VPN connection, which can provide remote communication for the users and CSF-Electric Kft. The managers and the people entitled can log into the system anywhere, anytime and with any smart device (mobile phone, tablet, laptop or PC). This is also useful for CSF-ELECTRIC Kft., since we are be able to identify any problems, solve them and implement requested modifications.
- certain machine parts or units can automatically turn off or enter sleeping mode whenever they are not used, thus reducing electricity usage
- we can install frequency inverters to the engines and fans in order to be able to accurately set the revolutions per minute (RPM) making them function in an optimal way based on the operation and energy use of the machine
- the language of the PLC control can be the given country’s language. So far, we installed English, German, French and Italian languages, but we are open to use different languages and the program is also able to use other ones.

- the system is be user-friendly and easy to learn. So far, our experience is that users are able to learn our current system very easily and use it without help only after a single day of usage

In the designing, implementation and testing phases we kept very close contact with Canadian Down and Feather company. Thanks to our good relationship and constant meetings we were able to reach every single goal we set in the beginning and create a user-friendly easy to install versatile PLC control system.

1. VAP fiche and updates

- includes list of activities:
    - activity number and short description
    - the activity name and description must be the same as it was described in the application form)
    - deliverables and subcontractor deliverables list (deliverable number, title, number of related activity)
- changes to activities – any changes can be reported, otherwise leave blank
- changes to timeline – any changes can be reported, otherwise leave blank

PLEASE COMPLETE HERE (1-2 pages)

WP1: Design and consultation

The first task was the selection of elements which are now part of the database. Important parameters and variables for future use are determined in our currently used equipment. The system store parameters needed for manufacturing, which could support the quality insurance system of the user. The user can manufacture products with manufacturing recipes, thus making the setting of manufacturing parameters easier.

The next group of data recorded consists of the parameters of the manufactured products. The specific manufacturing elements with which the product was created will be recorded. With this method the manufacturing process can be checked and controlled, which provides great help for the end user in terms of quality production; and in case of customer complaints involving manufacturing defects, the system provides opportunity to correct manufacturing parameters and settings. To be able to carry that out, the manufactured products can be given identification numbers, and to assign the used filling material and sheets to the product. The next group is the statistical data recorded during the manufacturing process, for example the operating hours of the machine units, the quantity of fillers used in the manufacturing process, and the number and manufacturing date of the products. With this the end user is be able to optimize the work process in terms of machine utilization and human resources. At the same time, the needs of the end users are identified, as well as the analysis and organization of their implementation.

We received the idea and referral from the Canadian Down and Feather company. Mr. Ashwin Aggarwal, the owner and leader of the company was very satisfied with the installed duvet and pillow filling machines and the feather mixing machine. As an innovative person, he started to wonder if we could create a statistical system from which he could collect data and analysis anywhere and anytime. We also liked this idea, since we could create such a system not only for machines but for a whole machinery too, and our existing business partners could also benefit from this. Designing was carried out by our engineers, comparing the needs of CDFC with our ideas and the abilities of the machine. What was certain that we needed a BAR code reader system, VPN connection for remote access, frequency changers on electric engines for optimal RPM settings, sleep mode for machine units when not used, and a user friendly and easy to use menu. Through the years we have created automatic control for more than 50 machines, including filling machines and feather processing machines. In this case we want to focus on the statistics. Based on our idea every new PLC control for which we can implement new statistical system would be customized for a person or a company

WP2: Software design and ongoing consultation

We needed to define databases for these data. The function of databases is the reliable and persistent storage of data, as well as to provide quick searches. The database capacity, which depends on the data quantity and frequency of recording, need to be optimized to function effectively. For this we needed to determine which data is recorded and stored at which event. For example, we store the settings made at the beginning of production, and these must be only stored if we made modifications to it. Following this we create the database with the proper database management system.

The next step is the creation of the software. The interface is designed to consider needs and feasibility. Since the interface design focuses on end user needs ongoing consultation is required for creation. It is also important to design the software in a way that it can be easily modified later in case there are new needs, or the original ones require change. Following the interface design, data recorded in the database was programmed. A user interface must also be created where users are identified and gain right to access to the

system. An administrator user will be needed who has access to the whole system and manage users (to create, delete, and change passwords and access rights).

WP3: Tests, R & D, removing errors, refining

Following the installation of the software the testing began. The system was placed into the manufacturing environment, so with real time monitoring we could refine errors and data. End user consultation about experience and future developments was also important here.

WP1

- Manufacturing recipes for mixing machine
- Manufacturing recipes for pillow filling machine
- Manufacturing recipes for duvet filling machine
- Parameters of the products manufactured by the mixer
- Parameters of the products manufactured by the pillow filling machine
- Parameters of the products manufactured by the duvet filling machine

WP2

- Recipe production management of mixing machine
- Recipe production management of pillow filling machine
- Recipe production management of duvet filling machine
- Store run-time data values of mixing machine in persistent log files
- Store run-time data values of pillow filling machine in persistent log files
- Store run-time data values of duvet filling machine in persistent log files
- Barcode management of raw materials and mixed products
- Mixing machine plc web pages programming
- Pillow filling machine web pages programming
- Duvet filling machine web pages programming

WP3

- Check the system, run tests

2. VAP Activities

- for each activity:

2.1 Title and description (The activity name and description must be the same as described in the application form)

2.2 Expected and achieved results (please mention % of achievement)

2.3 Deviations and corrective measures applied (in case of amendments please cite them)

2.4 Related deliverables (deliverable number and title)

PLEASE COMPLETE HERE (half page per activity)

2.1. Title and description

WP1: Design and consultation

The first task was the selection of elements which are now part of the database. Important parameters and variables for future use are determined in our currently used equipment. The system store parameters needed for manufacturing, which could support the quality insurance system of the user. The user can manufacture products with manufacturing recipes, thus making the setting of manufacturing parameters easier.

The next group of data recorded consists of the parameters of the manufactured products. The specific manufacturing elements with which the product was created will be recorded. With this method the manufacturing process can be checked and controlled, which provides great help for the end user in terms of quality production; and in case of customer complaints involving manufacturing defects, the system provides opportunity to correct manufacturing parameters and settings. To be able to carry that out, the manufactured products can be given identification numbers, and to assign the used filling material and sheets to the product. The next group is the statistical data recorded during the manufacturing process, for example the operating hours of the machine units, the quantity of fillers used in the manufacturing process, and the number and manufacturing date of the products. With this the end user is be able to optimize the work process in terms of machine utilization and human resources. At the same time, the needs of the end users are identified, as well as the analysis and organization of their implementation.

WP2: Software design and ongoing consultation

We needed to define databases for these data. The function of databases is the reliable and persistent storage of data, as well as to provide quick searches. The database capacity, which depends on the data quantity and frequency of recording, need to be optimized to function effectively. For this we needed to determine which data is recorded and stored at which event. For example, we store the settings made at the beginning of production, and these must be only stored if we made modifications to it. Following this we create the database with the proper database management system.

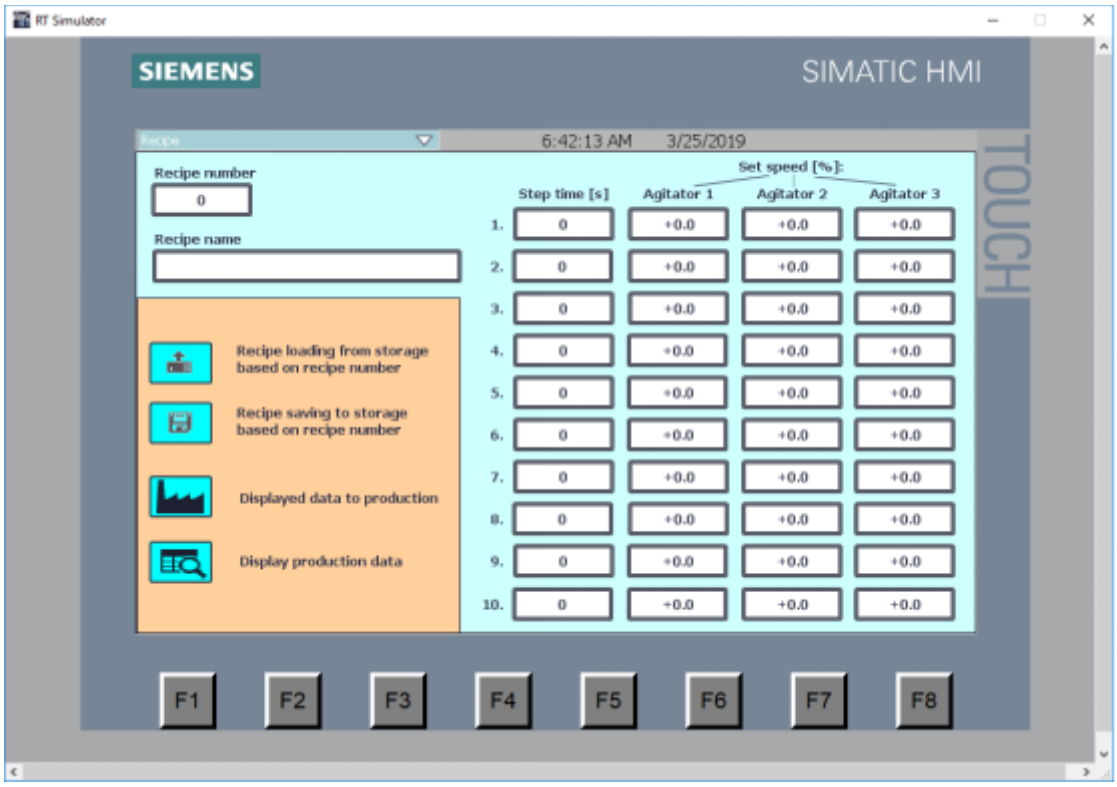
The next step is the creation of the software. The interface is designed to consider needs and feasibility. Since the interface design focuses on end user needs ongoing consultation is required for creation. It is also important to design the software in a way that it can be easily modified later in case there are new needs, or the original ones require change. Following the interface design, data recorded in the database was programmed. A user interface must also be created where users are identified and gain right to access to the system. An administrator user will be needed who has access to the whole system and manage users (to create, delete, and change passwords and access rights).

WP3: Tests, R & D, removing errors, refining

Following the installation of the software the testing began. The system was placed into the manufacturing environment, so with real time monitoring we could refine errors and data. End user consultation about experience and future developments was also important here.

2.2. Software design and ongoing consultation

WP1: Design and consultation



Manufacturing recipes for pillow filling machine

Pillow recipe data:

- Recipe number (1-1000)
- Recipe name
- down weight to be filled
- + weight tolerance for the set down weight
- - weight tolerance for the set down weight
- feather weight to be filled
- + weight tolerance for the set feather weight
- - weight tolerance for the set feather weight

Down recipe parameters:

- Recipe number (1-1000).
- Recipe name
- Filling fan speed (0 - 120%)
- Loading vacuum fan speed (0 - 100%)
- Down silo equalizing fan speed (0 - 100%)

- Down silo chain belt speed (0 - 100%)
- Down silo agitator speed (-100 - 100%)
- Vacuum cycle time (s)
- Vacuum time (s): weight bin emptying time
- Weighing time (s): this is a delay time after loading a weight bin to check its loaded weight
- Extra time of loading (s): if the loaded weight is less than the set weight the system will load more material inside the bin. This loading time is the extra time of loading. (Opening time of side slides)
- Time of emptying (s): if the bin is empty (measured weight is less than set weight) the control will continuously monitor the bin's weight for the time set here and stop emptying, if this condition is met
- Extra time of unloading: if the loaded weight is more than the set weight the system will take back material inside into the silo. This unloading time is the extra time of unloading.

Feather recipe parameters:

- Recipe number (1-1000).
- Recipe name: the saved name of the production recipe will be displayed
- Filling fan speed (0 - 120%) {+FS1-5M1}
- Loading vacuum fan speed (0 - 100%) {+S1-3M1}
- Feather silo equalizing fan speed (0 - 100%) {+S1-4M1}
- Feather silo chain belt speed (0 - 100%) {+S1-5M1}
- Feather silo agitator speed (-100 - 100%) {+S1-7M2}
- Vacuum cycle time (s)
- Vacuum time (s): weight bin emptying time
- Weighing time (s): this is a delay time after loading a weight bin to check its loaded weight
- Extra time of loading (s): if the loaded weight is less than the set weight the system will load more material inside the bin. This loading time is the extra time of loading. (Opening time of side slides
- Time of emptying (s): if the bin is empty (measured weight is less than set weight) the control will continuously monitor the bin's weight for the time set here and stop emptying, if this condition is met
- Extra time of unloading: if the loaded weight is more than the set weight the system will take back material inside into the silo. This unloading time is the extra time of unloading.

Manufacturing recipes for duvet filling machine:

- Comforter recipe name: name of comforter
- Length of the first, second, third etc. cell of the comforter can be set in mm

- Recipe number (1-1000): The previously set recipe number can be entered
- Complete length of comforter can be set in mm
- Comforter cell length can be adjusted in mm
- Number of duvets: Number of comforters being filled at the same time (1 or 2)
- Cell weights same: Sets all cell weight to the same gram
- Number of rows: The comforter's number of rows can be set
- Set weight + tolerance
- Set weight - tolerance
- Complete length of comforter
- Number of columns: The comforter's number of columns can be set
- Filling material recipe parameters:
- Recipe number (1-1000): The previously set recipe number can be entered
- Blower speed: (0 - 120%) You can set the speed of the fans on the filling stands. These fans will blow the filling material into the duvets
- Loading speed: (0 - 100%). These fans will generate the vacuum needed to load the weight bins from the silo
- Silo agitator speed (-100 - 100%). You can set the silo agitator speed
- Emptying time 1 (sec). When the weight bins are emptying after this time vacuuming will start inside the bins for loosening the down.
- Weighting time: After loading a bin after this time will the weighting process start
- Time of emptying: if the bin is empty (measured weight is less than set weight) the control will continuously monitor the bin's weight for the time set here and stop emptying, if this condition is met
- Equalizing speed: (0 - 100%) {+S1-4M1}. This fan will put back the filling material from the weight bins to into the silo
- Silo speed: (0 - 100%) {+S1-5M1}. You can set the speed of the silo's chain belt
- Vacuum time (sec). Time for vacuuming
- Extra time of loading: (sec). If the measured weight is less than the set weight (with – tolerance) the system will load more. You can set loading time here
- Extra time of unloading (sec). If the measured weight inside the bin is more than the set weight (with + tolerance) the system will transfer back some material in the silo. You can set this unloading time here.

Parameters of the products manufactured by the mixer

Mixing log file

Index	**Date**	**Material 1**	**M1 weight**	**...**	**Material 5**	**M5 weight**	**Keverék kódja**
Production number	Production date	Material code	Material weight		Material code	Material weight	Mixed material code

Bagging log file

Index	**Date**	**Material**	**Weight**
Production number	Production date	Material code	Material weight

Parameters of the products manufactured by the pillow filling machine

Record	1	Production number
Date	11/09/2019	Production date
UTC Time	12:53:27	Production time
Loc_time	2019-11-09 13:53:27.194	Local time
Name	Recept1	Recipe name
Mode	0	Production mode
Weight1	50,00	Weight of down
Weight 1+	2,00	Weight tolerance +
Weight1-	2,00	Weight tolerance -
Weight 2	1 000,00	Weight of feather
Weight 2+	5,00	Weight tolerance +
Weight 2-	5,00	Weight tolerance -

Measured_feather	1 001,62	Feather actual weight
Measured down 1	48,07	Down 1 actual weight
Measured down 2	49,68	Down 2 actual weight
User	3 Chamber	User name of operator
Code	0	Product code

Parameters of the products manufactured by the duvet filling machine

Record	3001	Production number
Time	2019-08-05 08:00:19.695	Production time
Name	C003521BLANC260	Recipe name
Nrof_rows	10	Number of rows
Nrof_columns	6	Number of columns
Cell_1	200,0	Cell 1 height
Cell_2	200,0	Cell 2 height
Cell_3	200,0	Cell 3 height
Cell_4	200,0	Cell 4 height
Cell_5	200,0	Cell 5 height
Cell_6	200,0	Cell 6 height
Cell_7	200,0	Cell 7 height
Cell_8	200,0	Cell 8 height
Cell_9	200,0	Cell 9 height
Cell_10	200,0	Cell 10 height
Pulling_corr	49,68	Correction of pulling
Cell_weight		Cell weight
Weight_t+		Weight tolerance +

Weight_t		Weight tolerance -
Size_length		Lenth of duvet
Bits		
Nrof_duvets		Producted duvet
User	3Chamber	User name of operator
Code	0	Product code

Recipe data management

The recipe data block uses an array of product recipe records. Each element of the recipe array represents a different recipe that is based on a common set of components.

We created a PLC data type that defines all the ingredients in a recipe record. This data type template is reused for all recipe records.

One of the recipes can be transferred at any time from the recipe data block (all recipes in memory) to the active recipe data block (one recipe in memory) using a function. After a recipe record is moved to memory, then our program logic can read the component values and begin a production run. This transfer minimizes the amount of CPU work memory that is required for recipe data.

If the active recipe component values are adjusted by an HMI device during a production run, we can write the modified values back to the recipe data block.

The complete set of recipe records can be generated as a CSV file using the RecipeExport instruction. Unused recipe records are also exported.

Once a recipe export operation is completed, then you can use the generated CSV file as a data structure template.

Recipe production management of mixing machine – for "Manufacturing recipes for mixing machine" use the "Recipe data management" description.

Recipe production management of pillow filling machine – for "Manufacturing recipes for pillow filling machine" use the "Recipe data management" description.

Recipe production management of duvet filling machine – for "Manufacturing recipes for duvet filling machine" use the "Recipe data management" description.

Data logs

Our control program can use the Data log instructions to store run-time data values in persistent log files. The CPU stores data log files in flash memory (memory card) in

standard CSV (Comma Separated Value) format. The CPU organizes the data records as a circular log file of a predetermined size.

Our program uses the Data log instructions in our program to create, open, write a record to, and close the log files. The CPU uses our data buffer as temporary storage for a new log record. Our control program moves new current values into the buffer during runtime. When the program has updated all current data values, it can then execute the DataLogWrite instruction to transfer data from the buffer to a data log record.

Store run-time data values of mixing machine in persistent log files – for (“Parameters of the products manufactured by the mixer” use the “Data logs” description

Store run-time data values of pillow filling machine in persistent log files – for the (“Parameters of the products manufactured by the pillow filling machine” use the “Data logs” description

Store run-time data values of duvet filling machine in persistent log files – for the (“Parameters of the products manufactured by the duvet filling machine” use the “Data logs” description

Barcode management of raw materials and mixed products

The barcodes of the raw materials are entered into the system by a barcode reader. Barcodes for mixed materials (which can be printed out by a printer) are generated by the system when the mixing process is complete. The code and weight of the materials filled in the bags are recorded by the program. The code and weight of the mixtures are also recorded and loaded directly into the production machines by the mixer.

Mixing machine plc web pages programming

The Web server for the PLC-s provides Web page access to data about your CPU and process data. The user can access the PLC Web pages from a PC or from a mobile device. For devices with small screens, the Web server supports a collection of basic pages. The users use a Web browser to access the IP address of the PLC CPU. The PLC-s supports multiple concurrent connections. The PLC web pages of the mixer are attached in the appendix.

Pillow filling machine web pages programming

The PLC web pages of the pillow filling machine are attached in the appendix.

Duvet filling machine web pages programming

The PLC web pages of the duvet filling machine are attached in the appendix.

3. VAP Conclusions

- VAP strategic, technical and commercial objectives achieved vs. expected

- VAP product-market development novelty verification at national and international scale conclusions after VAP implementation.

- Updated status of the innovative business plan and future collaborative RTDI that were described in the application.

PLEASE COMPLETE HERE (one page)

The results of the project are presented on our website. Photos and videos of the software are available and everyone who is interested can find results of the mechanical and control details. Videos were made about how the software works. These videos are uploaded on YouTube, where comments can be made, questions can be asked. We are showing user interfaces, navigational and setting possibilities, as well as system monitoring like the graphic display of data and parameters assisting operation maintenance.

- operator profile includes monitoring and interference
- system monitoring and management profile manages system settings before and during operation
- maintenance profile provides information and functions

The information mentioned above are shared on Facebook, google+, Instagram, and Twitter. We also plan machine demonstrations, inviting customers, even with video conferencing. We would like to rent a stand in the annual textile industry exhibition in Germany to present the prototype. This is an expensive method, but we believe it is one of the best ways, because the exhibition is visited by many potential customers.

CSF-ELECTRIC Kft. and the Canadian Down and Feather Company made an agreement to invite potential companies at their workplace who are interested in the new software developed in this INNOWWIDE call to share their experiences, results and successes with the visiting companies. They also undertake to share promotional materials.

These customers are mainly companies from Canada and the United States whose market is mainly North America. Since the project brought positive results, it is possible

to consult with stakeholders on further development options and possible corrections. From this not only our company but our project partner can benefit too, increasing their reputation and the number of customers. This opportunity is incredible, since there are a lot of potential, well-stocked feather processing and textile industry companies in Canada and in the United States.

Also advantages for our partner is that they could show their customers that the monitoring quality of the production process is high, guaranteeing high quality and excellent end products.

We made a list of textile industry companies to contact them via telephone, email or in person. The most effective would be a personal visit at their company or at our workshop in Makó, Hungary. We could present our company by showing pictures, videos, specifications and machines. After knowing their exact need, we would quote them. Mr. Hermann Pfrommer, who was the CEO of the German feather processing machine manufacturing company LORCH for nearly 50 years, would provide a great help in sales. We are in connection with him on a weekly basis. We have done more than 60 successful joint projects over the years.

We are members of the International Down and Feather Bureau (IDFB) and the European Down and Feather Association (EDFA). These organizations hold meetings every year, which is attended by small and big textile industry companies from all over the world. We get to meet our business partners and find new ones. We also visit Heimtextil trade fair in Frankfurt every yea. In 2020 was the sixth time we were there. The points mentioned above offer great opportunities to promote our products locally and internationally.

4. Annexes – Deliverables

- please attach any deliverables in annex

Attached separately.

INNOWIDE

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement nº 822273. This document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

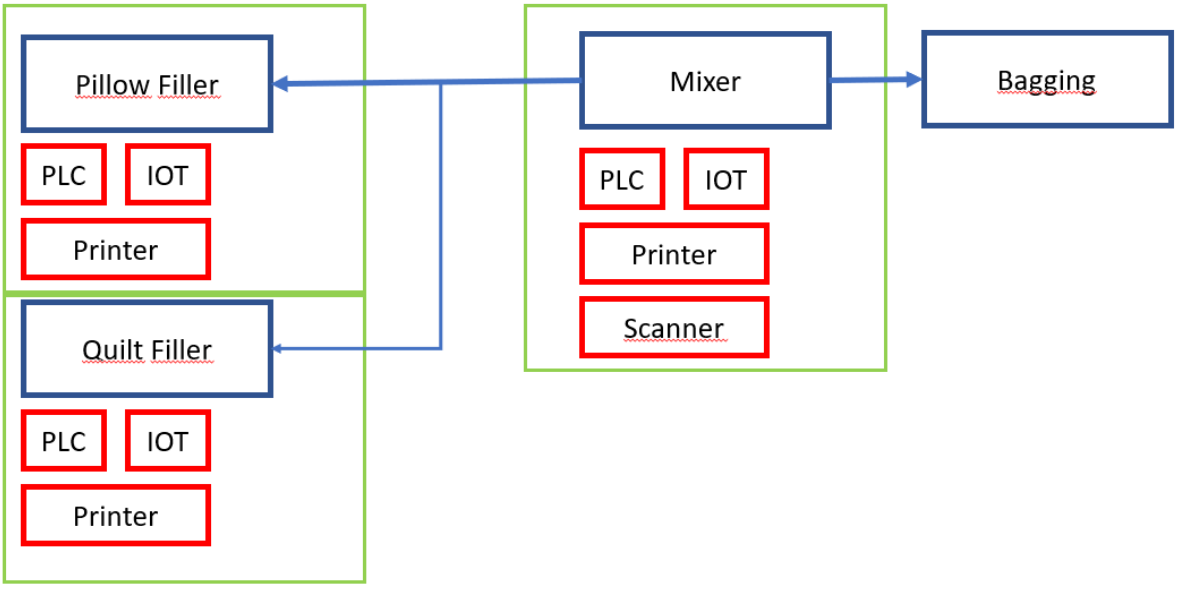
INNOWWIDE report 2019 November

In this month the following tasks have been completed:

1. The concept of the system is defined

Task 1:

The concept of the system can be seen of Figure 1.



1. Figure: The concept of the system

The proposed system is consisting of four main parts. The Pillow Filler, the Quilt Filler, the Mixer and the Manual Bagging. A PLC is connected to each station except the the Bagging station. The PLC can communicate with and IOT2040 Gateway. This gateway has the capability of handle a barcode scanner and a barcode printer.

The barcode scanner reads the barcode of the package. The IOT is convert the barcode to a material code between 000-999. After this the material code is sent to the PLC which creates the following code:

2. Figure: The created code

The weight is added by the Mixer PLC. For example if the weight of the base material is 23.4kg, PLC writes 234 into the Weight in Figure 2.

In the following months we work with the IOT2040 Gatewa to perform the following tasks:

3. The operating system for the IOT2040has been set up
4. The node-red software for data collection has been set upThe MySQL database server for data storage has been installed
5. Creating an application on the Siemens S7-1200 PLC to generate data for testing the node-red connection and database storage
6. Install and set up DBebaver
7. Creating a node-red application to connect the S7-1200 PLC and save data to the MySQL database
8. Creating a node-red application to connect the S7-1200 PLC and save data to the MySQL database and an external database via REST-API
9. Creating a node-red application to read barcodes from barcode reader, which connected to the IOT Gateway via USB (2 different solutions are shown below)
10. Creating a SIEMENS TIA Portal Project containing an S7-1200 PLC and a KTP 700 Comfort panel to read and store barcode

INNOWWIDE report 2019 December

In this month the following tasks have been completed:

1. The required hardware for the in-filed data collection and storage has been selected
2. The operating system for this hardware has been set up
3. The node-red software for data collection has been set up
4. The MySQL database server for data storage has been installed

Task 1.

The selected hardware for the in-field data collection and storage is the SIEMENS IoT2040 gateway which is designed especially for these kind of tasks by the SIEMENS. The IoT2040 can be seen on fig. 1.

Figure 1. The IoT2040 hardware

The main parameters of the IoT2040 hardware can be found on this link. This hardware was also equipped with a 32GB micro SD card to store the operating system, the node-red software, the MySQL database server and the collected data.

Task 2.

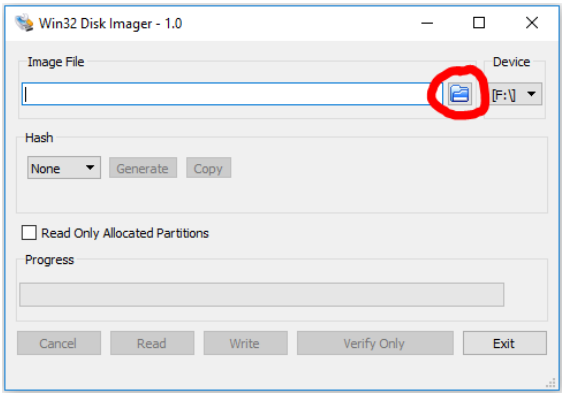
To install the operating system for the IoT2040 gateway the following software are required:

- **Micro-SD Card Example Image:**
  To use the full functionality of the SIMATIC IOT2040 a SD-Card Example Image with a Yocto Linux Operating System is necessary to be installed. This Image is provided through the Siemens Industry Online Support. It can be downloaded here.
- **PuTTY:**
  To get remote access to the SIMATIC IOT2040 the PuTTY software is required. With this software it is possible to establish a connection to different devices for example via Serial, SSH or Telnet. The PuTTY software can be downloaded here.
- **Win32 Disk Imager:**

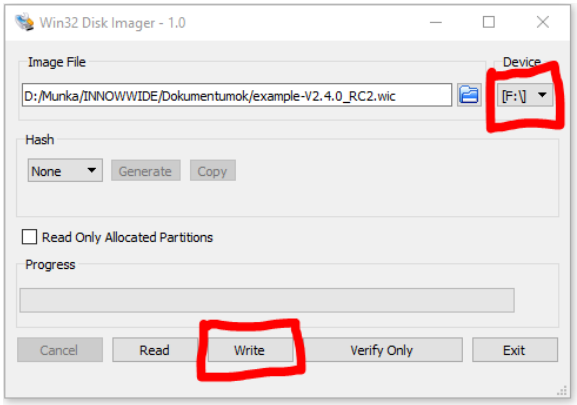
In order to put the SD Card image to the µSD Card the Win32 Disk Imager software is needed. The software can be downloaded here.

The first step is installing the SD-Card image to the IoT2040 gateway. For this the following steps are required:

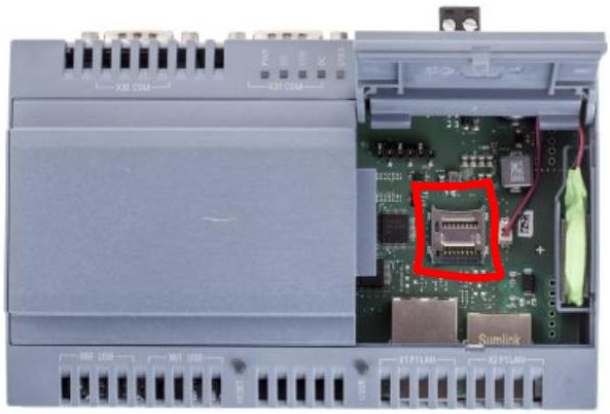
1. Insert the µSD-Card via SD-Card Adapter in the SD-Card Slot of the PC
2. Retrieve the downloaded SD Card image .zip-file
3. Install the downloaded “win32diskimager-1.0.0-install.exe”
4. Start the software and select the downloaded “example-V2.4.0_RC2.wic” image



5. Select the destination drive (the drive letter of the SD Card)



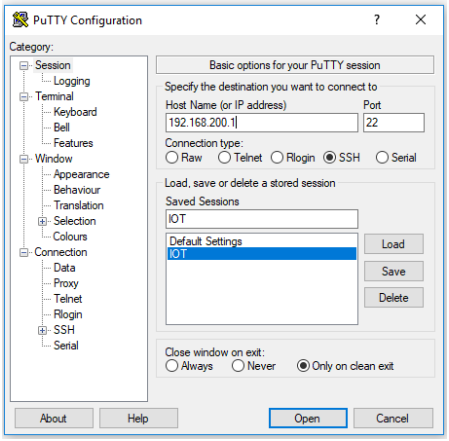
6. If the SD Card imaging ready insert it into the IoT2040



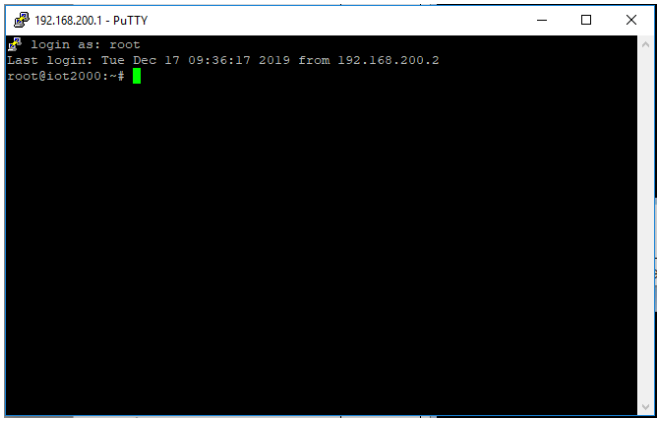
7. Connect the PC to the X1P1 Ethernet Port of the IoT2040 gateway using an Ethernet cable. Make sure the PC use the same subnet as the IoT2040 gateway. The IP address of the IoT2040 is 192.168.200.1 by default.
8. Connect the IoT2040 to a 24V power supply

The second step is to connect the IoT2040 gateway using the PuTTY

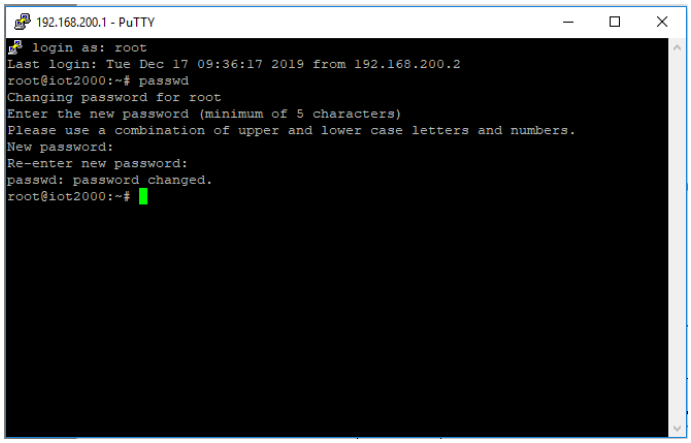
1. Start PuTTY
2. Choose the SSH connection type
3. Enter the IP address of the IoT2040 (192.168.200.1)
4. The port is 22 by default
5. You can save these setting
6. Click on Open



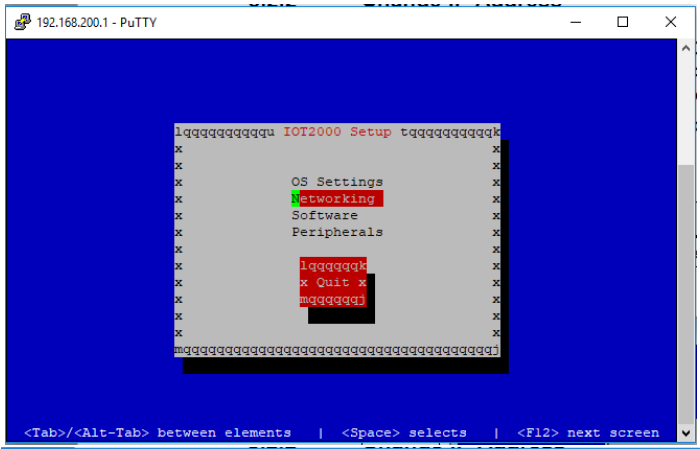
7. Type root and press Enter

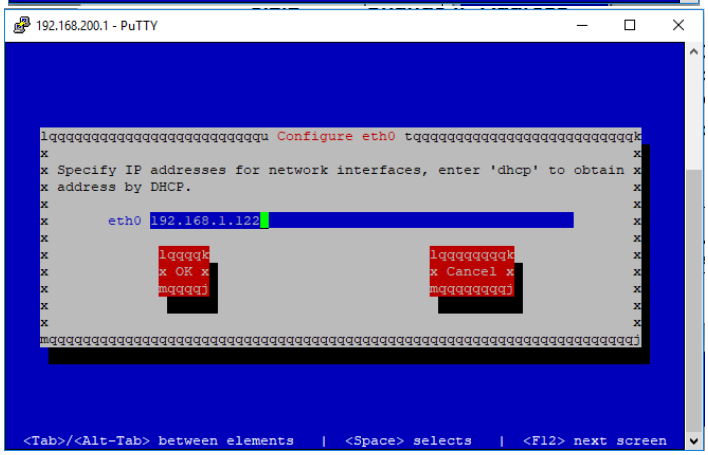


8. Set password for the root using the passwd command

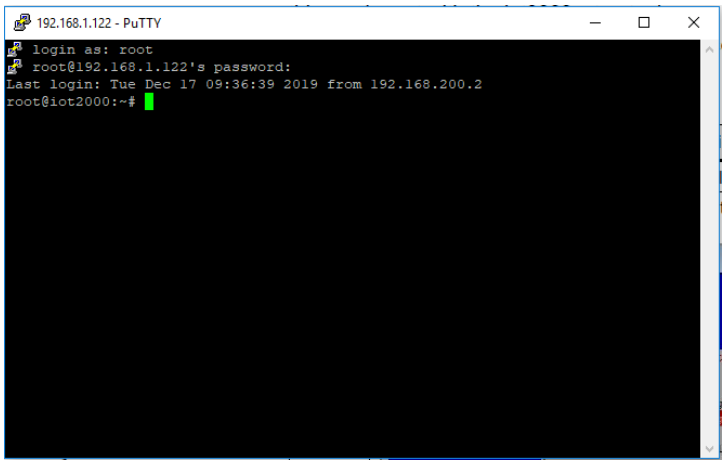


9. Change the IP address to 192.168.1.122 using iot2000setup command. Use this IP because the S7 PLC use the same subnet.





10. After changing the IP address reconnect to the IoT2040 using the PuTTY with the modified IP

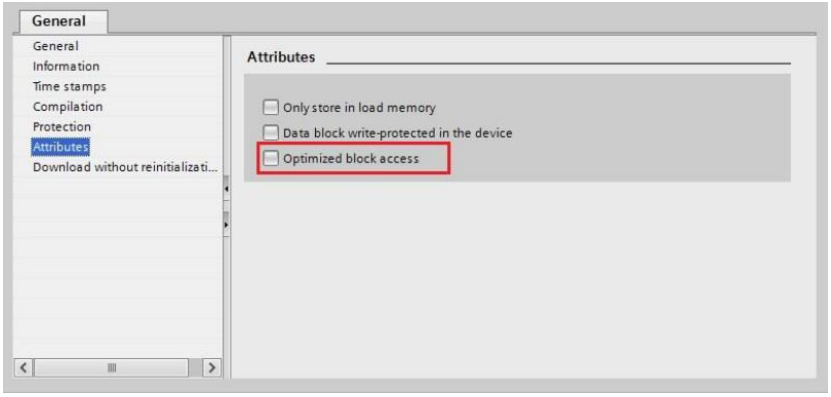


11. Now the IoT2040 is ready for the next task

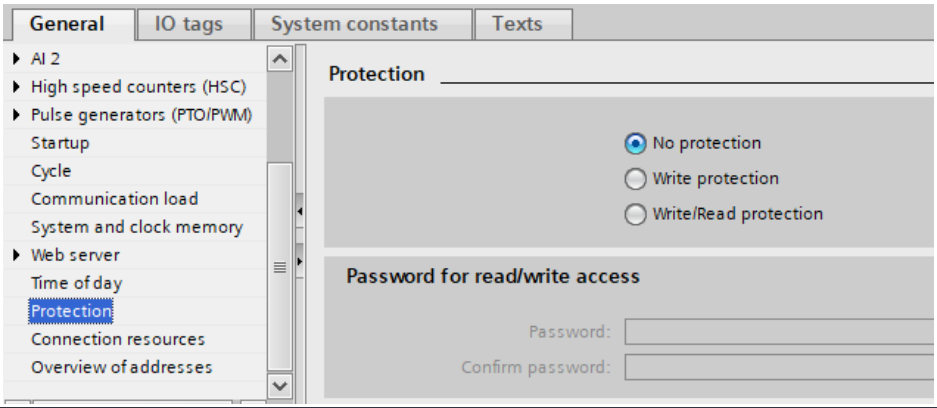
Task 3.

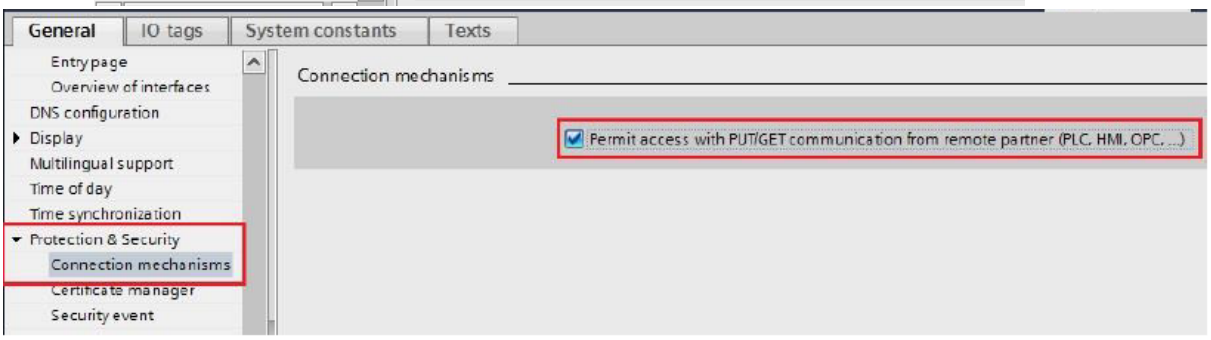
The first step is setting up the S7 PLC according to the IoT2040 gateway can communicate with is using the Node-red software.

1. Check the project and change the data block access to non-optimized



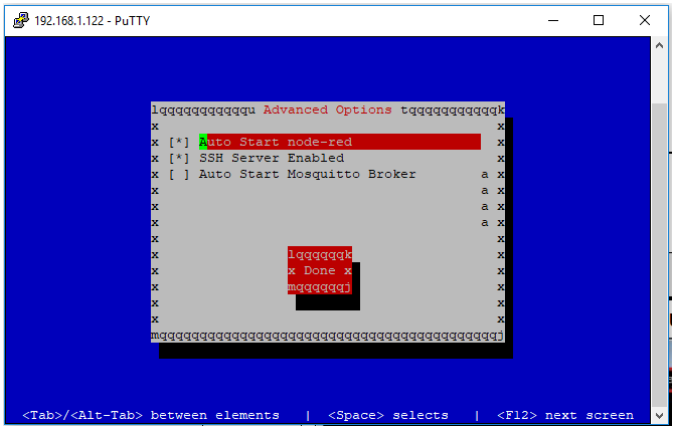
2. Enable PUT/GET communication in the S7 project. In S7-1200 and S7-1500 it can be done in a little bit different way



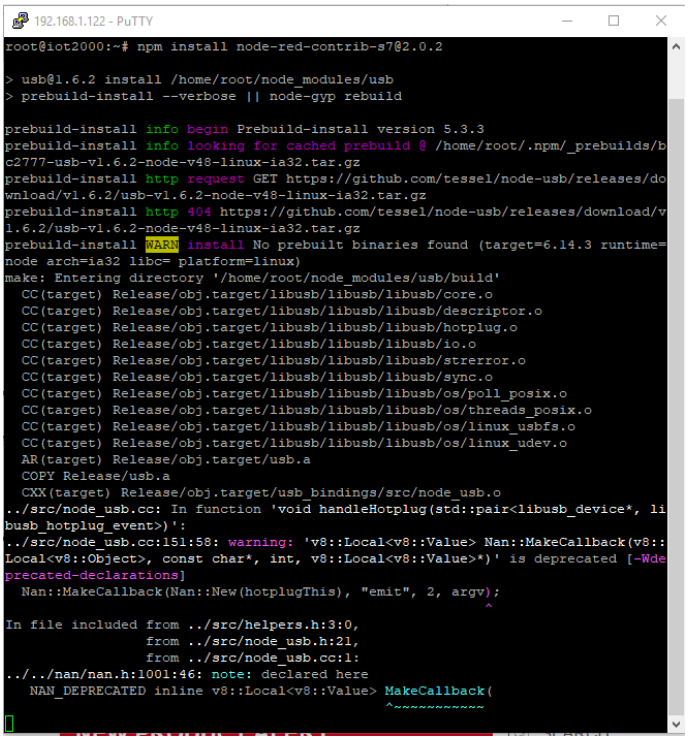


In the second step since the Node-red is preinstalled to the SD Card image it do not need to install. Only one package should be installed which is the node-red-contrib-s7 which is enables the Node-red to communicate with the S7 PLC.

1. Connect the X2P1 Ethernet Port of the IoT2040 gateway to WAN.
2. Enable the Node-red auto start after boot using the iot2000setup command
3. Software -> Manage Autostart Options -> Auto Start node-red-> Done



4. Restart IoT2040 gateway using reboot command
5. The Node-red can be manually start using node /usr/lib/nod_modules/node_red/red &
6. The start of the Node-red requires 4-5 minutes. Wait until the SD Card working LED stop flashing
7. Install the node-red-contrib-s7 using the npm install node-red-contrib-s7@2.0.2. The latest version of the node-red-contrib-s7 do not work with the preinstalled version of Node-red, so the 2.0.2 version of the package must be used.



8. Reboot the IoT2040 gateway
9. Start a browser and enter the 192.168.1.122:1880

Task 4

In the first step the MariaDB MySQL database software should be downloaded.

1. The MariaDB 5.5.66 can be downloaded from here

MariaDB 5.5.66 Stable 2019-11-05

Release Notes | Changelog

Affordable,enterprise class product support, professional services, and training for your MariaDB dat
Foundation's release sponsor, MariaDB Corporation. To learn more about them and their services fo
MariaDB Corporation at sales@mariadb.com.

File Name	Package Type	OS / CPU	Size	Meta
mariadb-5.5.66.tar.gz	source tar.gz file	Source	46.0 MB	Checksum Instructions
mariadb-5.5.66-winx64.msi	MSI Package	Windows x86_64	49.4 MB	Checksum Instructions
mariadb-5.5.66-winx64.zip	ZIP file	Windows x86_64	118.2 MB	Checksum Instructions
mariadb-5.5.66-win32.zip	ZIP file	Windows x86	120.4 MB	Checksum Instructions
mariadb-5.5.66-win32.msi	MSI Package	Windows x86	48.4 MB	Checksum Instructions
mariadb-5.5.66-linux-systemd-x86_64.tar.gz (for systems with systemd)	gzipped tar file	Linux x86_64	309.6 MB	Checksum Instructions
mariadb-5.5.66-linux-glibc_214-x86_64.tar.gz (requires GLIBC_2.14+)	gzipped tar file	Linux x86_64	306.1 MB	Checksum Instructions
mariadb-5.5.66-linux-x86_64.tar.gz	gzipped tar file	Linux x86_64	225.4 MB	Checksum Instructions
mariadb-5.5.66-linux-systemd-i686.tar.gz (for systems with systemd)	gzipped tar file	Linux x86	276.1 MB	Checksum Instructions
mariadb-5.5.66-linux-glibc_214-i686.tar.gz (requires GLIBC_2.14+)	gzipped tar file	Linux x86	273.7 MB	Checksum Instructions
mariadb-5.5.66-linux-i686.tar.gz	gzipped tar file	Linux x86	217.7 MB	Checksum Instructions
Red Hat, Fedora, and CentOS Packages	RPM Package	RedHat/CentOS/Fedora (x86, x86_64, ppc64, ppc64le)		Instructions

In the second step the MariaDB 5.5.66 should be installed.

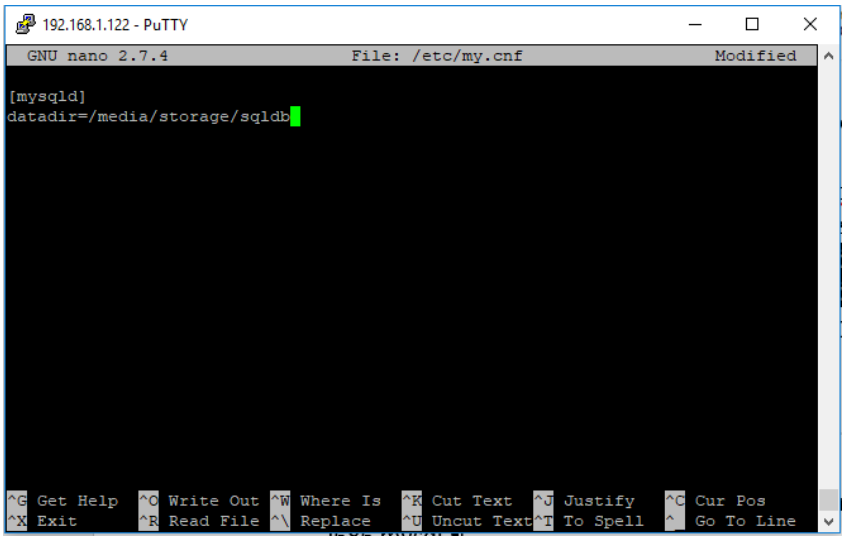
1. Connect to the IOT2040 via Putty
2. Create new user group using the groupadd mysql
3. Add new user to the previously created group using the useradd -g mysql mysql

```
root@iot2000:~# groupadd mysql
root@iot2000:~# useradd -g mysql mysql
```

4. Copy downloaded file to /home/root on IoT2040 using WinSCP
5. Create directory using the mkdir /usr/local
6. Navigate to /usr/local using cd /usr/local
7. Extract the downloaded MariaDB package using the tar -zxvpf /home/root/mariadb-5.5.66-linux-i686.tar.gz
8. Create a soft link to easily access MariaDB folder with the ln -s mariadb-5.5.66.linux-i686 mysql.

```
root@iot2000:/usr/local# ln -s mariadb-5.5.66-linux-i686 mysql
root@iot2000:/usr/local#
```

9. Use the mkdir /media/storage to create the required folder. Use the chown -R mysql /media/storage to add permission to mysql user to the media/storage/ folder. Enter nano /etc/my.cnf, and add the following lines.



10. Switch to mysql directory using cd mysql
11. Change the user to mysql using su mysql

12. Start the installation script for MariaDB using the ./scripts/mysql_install_db --datadir=/media/storage/sqldb

13. To start the server manually the ./bin/mysqld --user=mysql & command can be used

```
root@iot2000:/usr/local/mysql# ./bin/mysqld --user=mysql &
[1] 534
root@iot2000:/usr/local/mysql# 191217 11:45:23 [Note] ./bin/mysqld (mysqld 5.5.6
6-MariaDB) starting as process 534 ...
191217 11:45:25 InnoDB: The InnoDB memory heap is disabled
191217 11:45:25 InnoDB: Mutexes and rw_locks use InnoDB's own implementation
191217 11:45:25 InnoDB: Compressed tables use zlib 1.2.11
191217 11:45:25 InnoDB: Using Linux native AIO
191217 11:45:25 InnoDB: Initializing buffer pool, size = 128.0M
191217 11:45:26 InnoDB: Completed initialization of buffer pool
191217 11:45:26 InnoDB: highest supported file format is Barracuda.
191217 11:45:27  InnoDB: Waiting for the background threads to start
191217 11:45:28 Percona XtraDB (http://www.percona.com) 5.5.61-MariaDB-38.13 sta
rted; log sequence number 1597945
191217 11:45:28 [Note] Plugin 'FEEDBACK' is disabled.
191217 11:45:28 [Note] Server socket created on IP: '0.0.0.0'.
191217 11:45:29 [Note] Event Scheduler: Loaded 0 events
191217 11:45:29 [Note] ./bin/mysqld: ready for connections.
Version: '5.5.66-MariaDB'  socket: '/tmp/mysql.sock'  port: 3306  MariaDB Server
root@iot2000:/usr/local/mysql#
```

14. Test the connection with the ./bin/mysql command

```
root@iot2000:/usr/local/mysql# ./bin/mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 1
Server version: 5.5.66-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

15. Now use the exit command to exit the program

In the third step set up some security for the MariaDB server

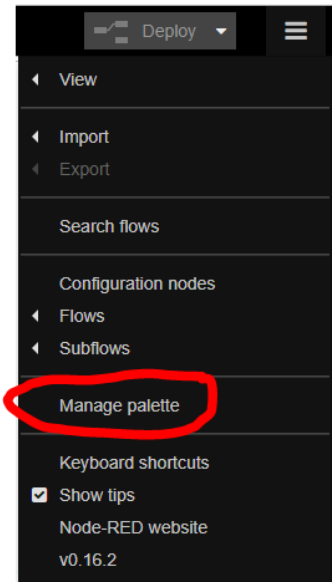
1. In root user change the directory using cd /usr/local/mysql
2. Run the ./bin/mysql_secure_installation command to secure the installation. In the enter current password part only an Enter is required.
3. Set the root password and accepting everything with (Y)

In the fourth step the MariaDB should be added to the autostart

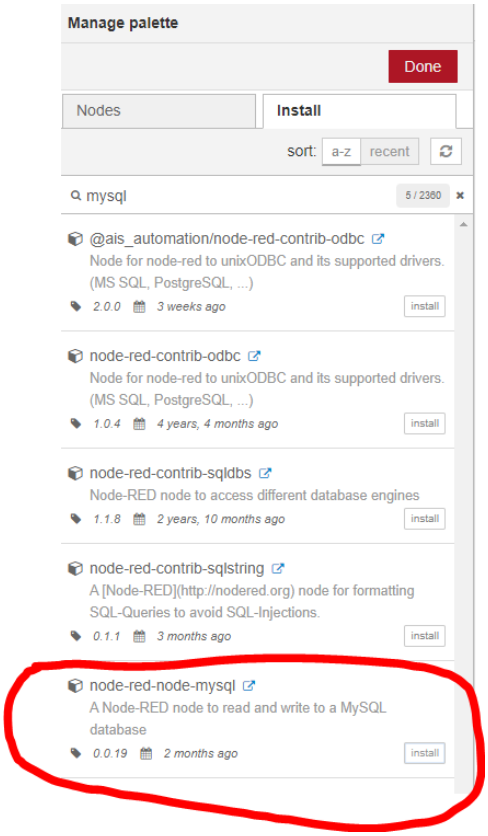
1. Copy startup script to the auto start folder: cp support-files/mysql.server /etc/init.d/mysql
2. Add copied file to the init.d system: update-rc.d mysql defaults

After these steps new database can be created. To reach the created database using the Node-red an additional module should be installed using the following steps

1. Open the 192.168.1.122:1880 IP address and port using a web browser
2. Open Manage palette



3. Search for mysql and install the node node-red-node-mysql



4. After installing this module, the MySQL database connection can be established

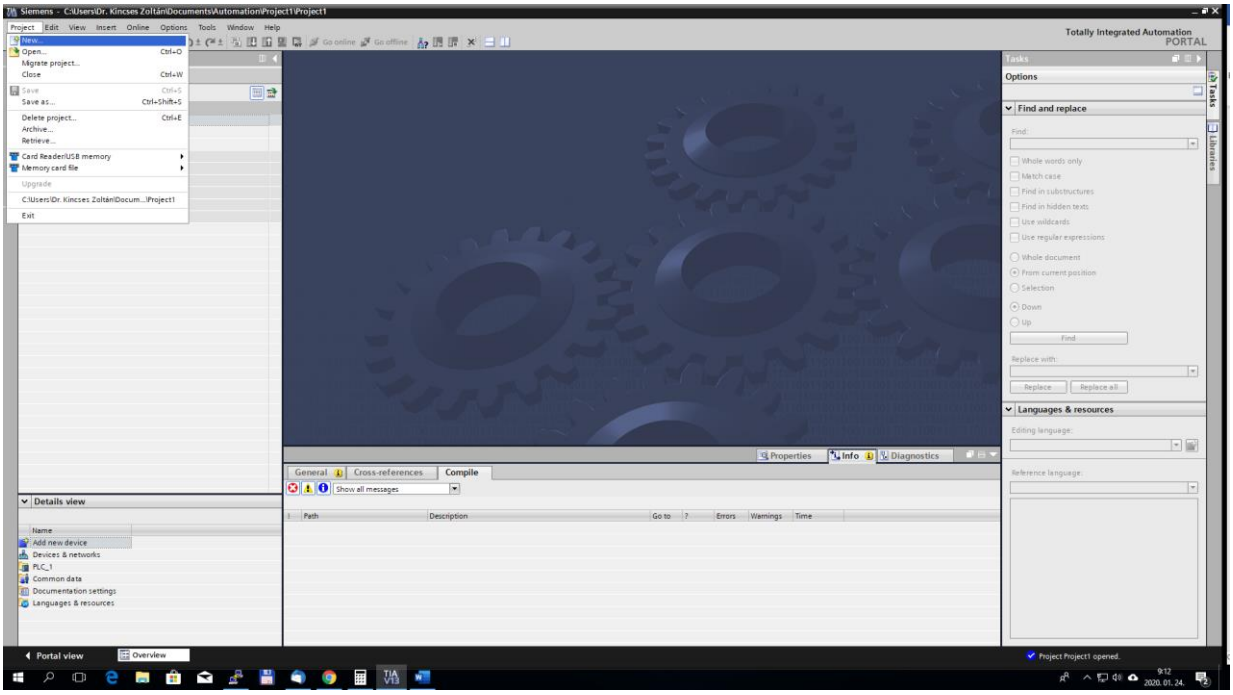
INNOWWIDE report 2020 January

In this month the following tasks have been completed:

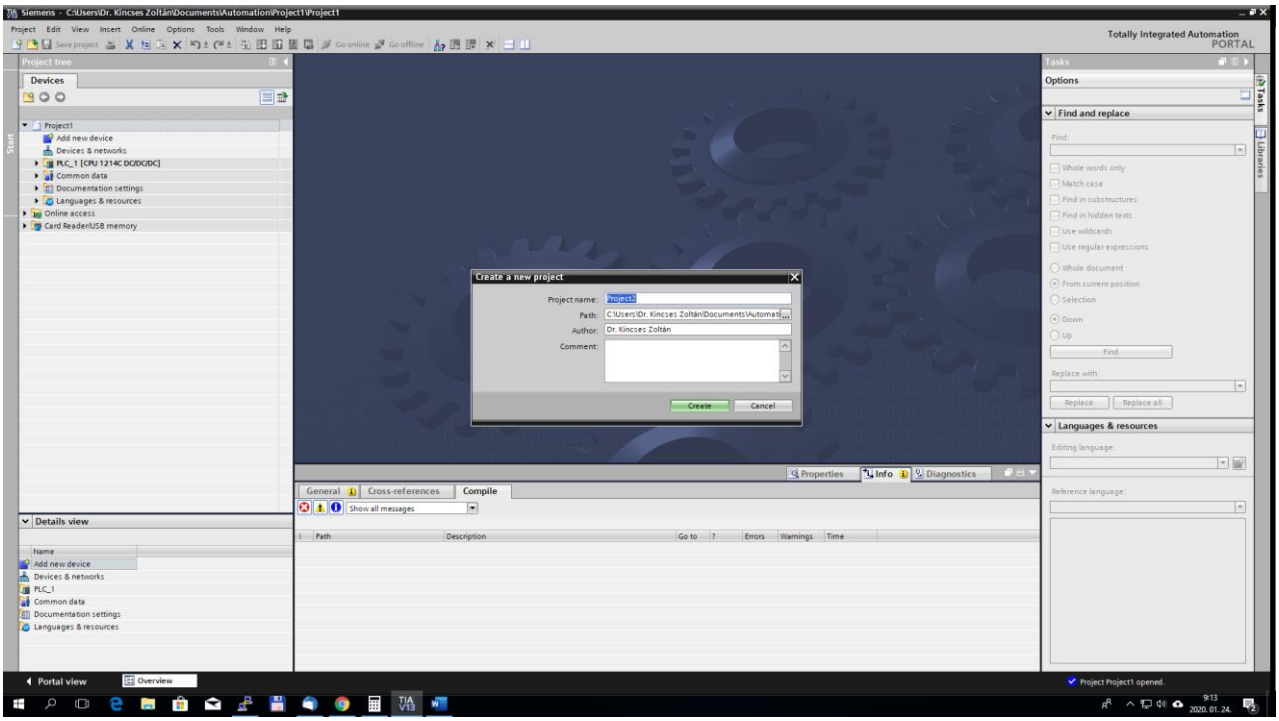
1. Creating an application on the Siemens S7-1200 PLC to generate data for testing the node-red connection and database storage
2. Install and set up DBebaver
3. Creating a node-red application to connect the S7-1200 PLC and save data to the MySQL database

Task 1.

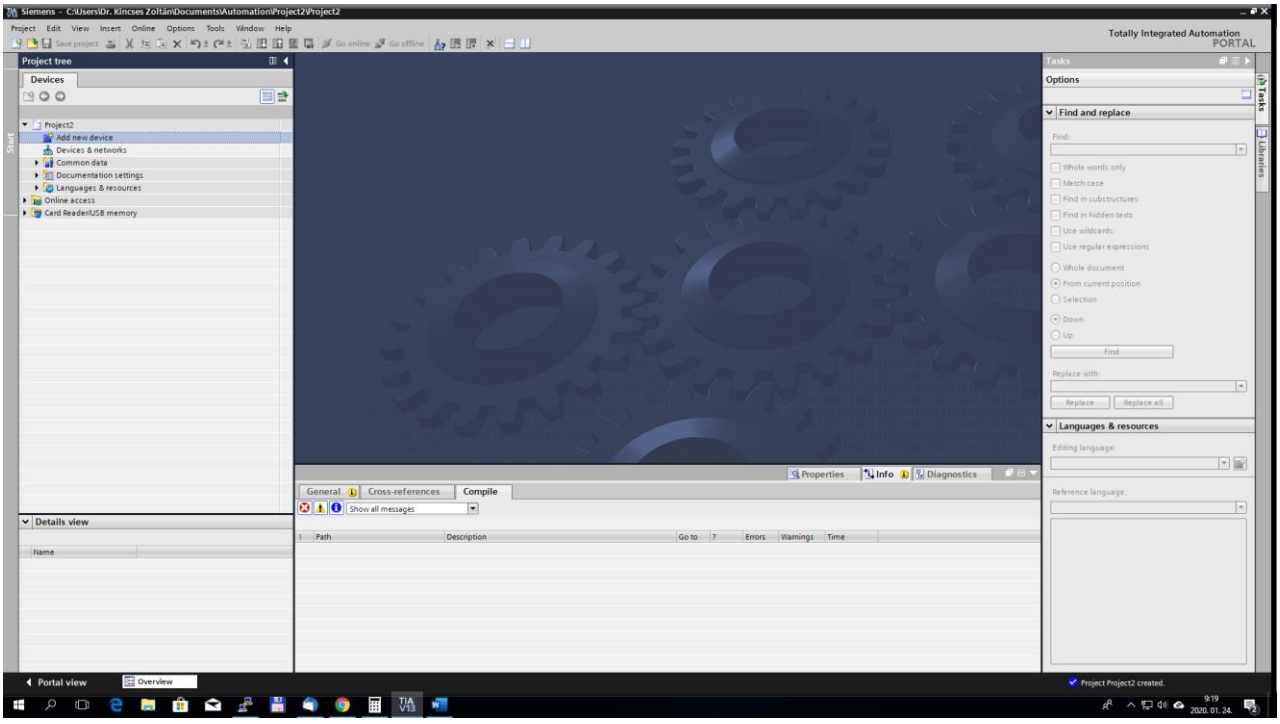
1. To create the test application first a new project should be created in TIA Portal.



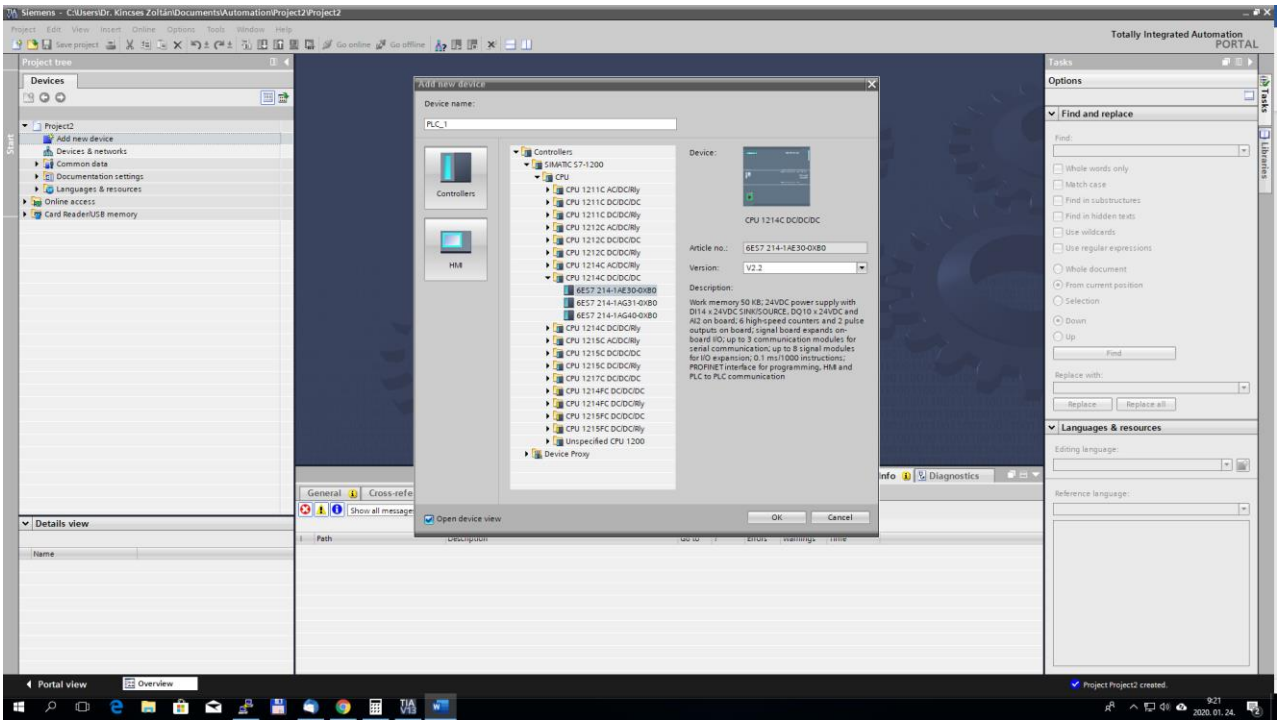
2. Give an appropriate name to the project and set the project path



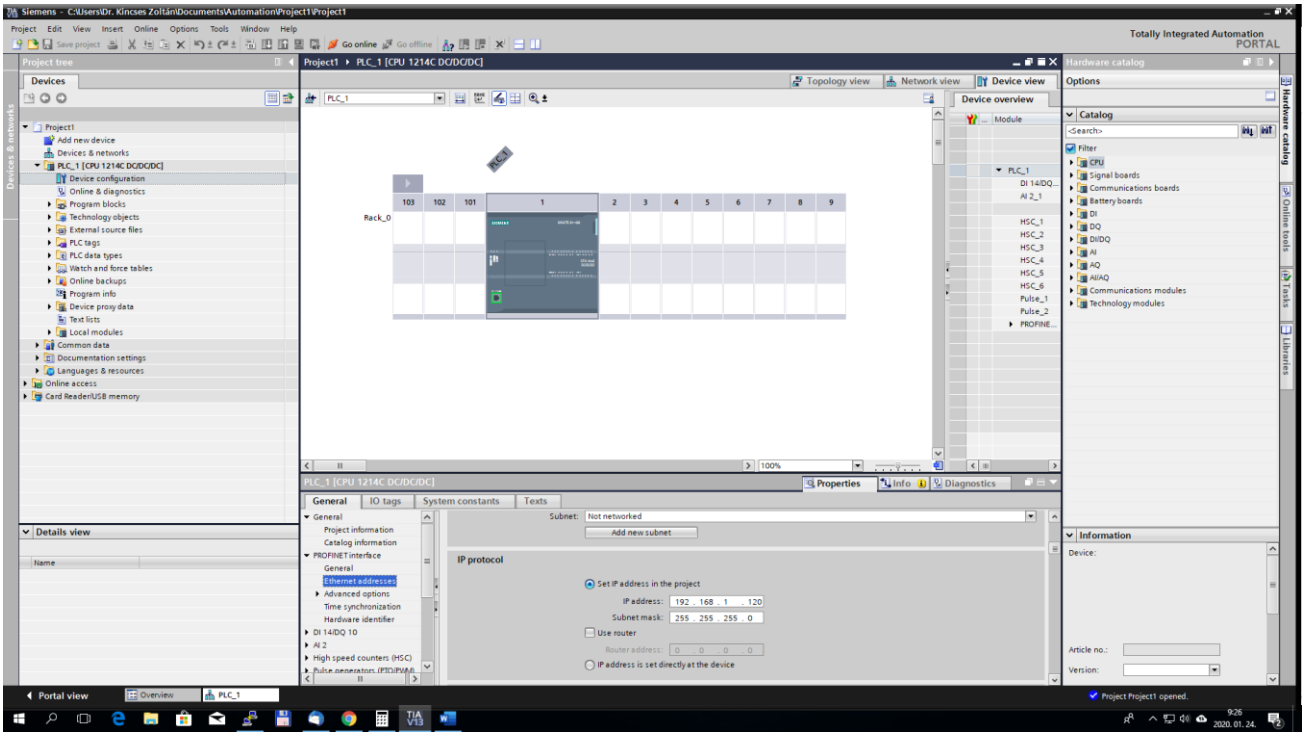
3. After the project created click on the Add new device tab



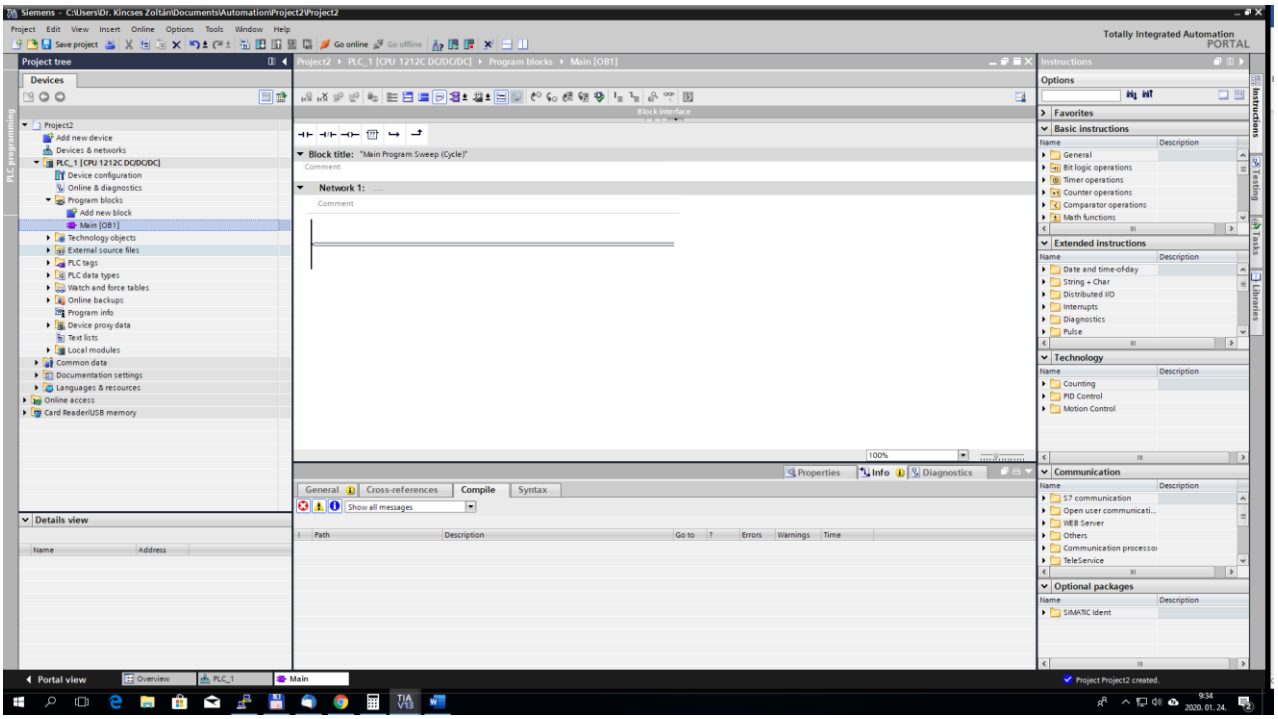
4. Select the appropriate PLC model and click on the OK button



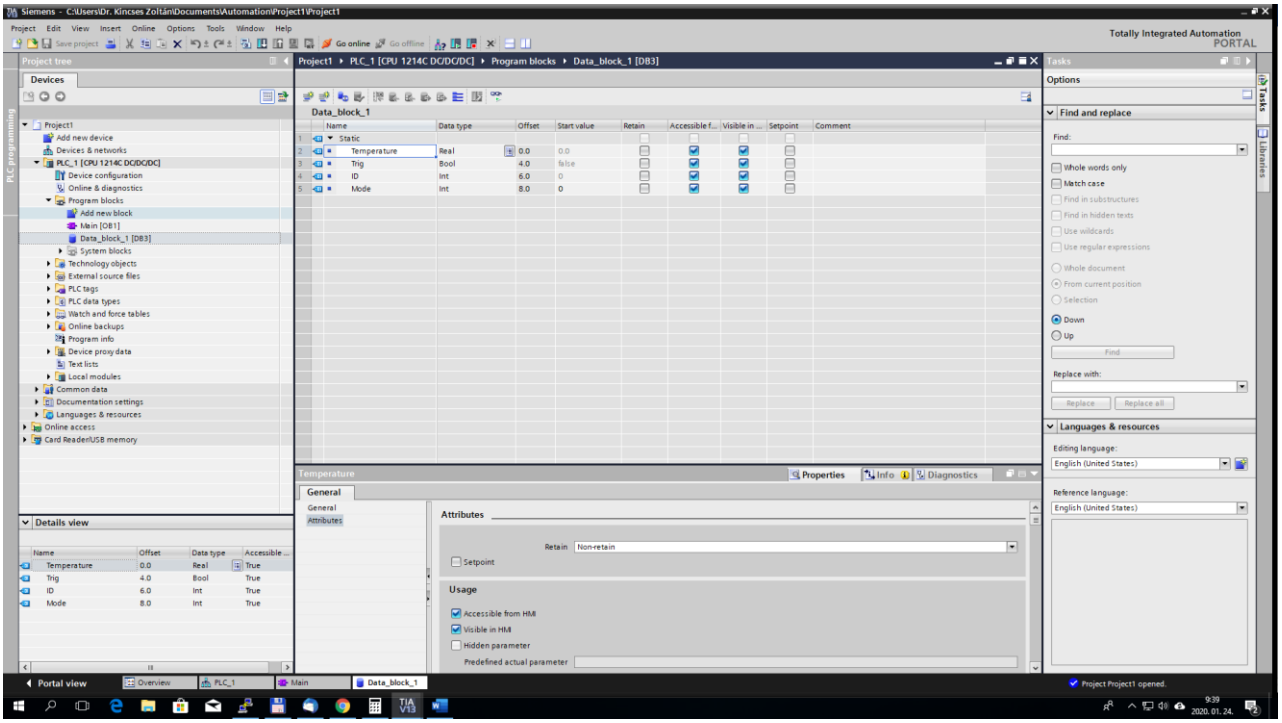
5. After adding the device in the Device configuration set the IP of the PLC to 192.168.1.x in order to make sure the PLC and the IOT gateway (192.168.1.122) is in the same subnet.



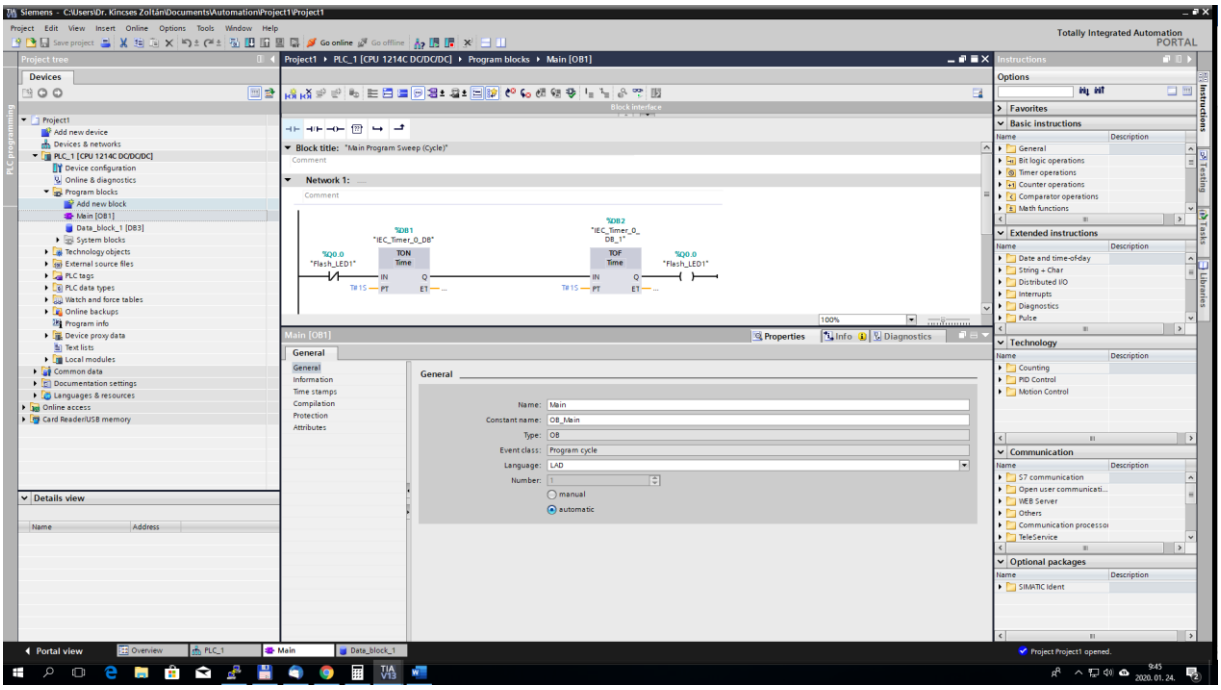
6. Open the Program block -> Main [OB1]



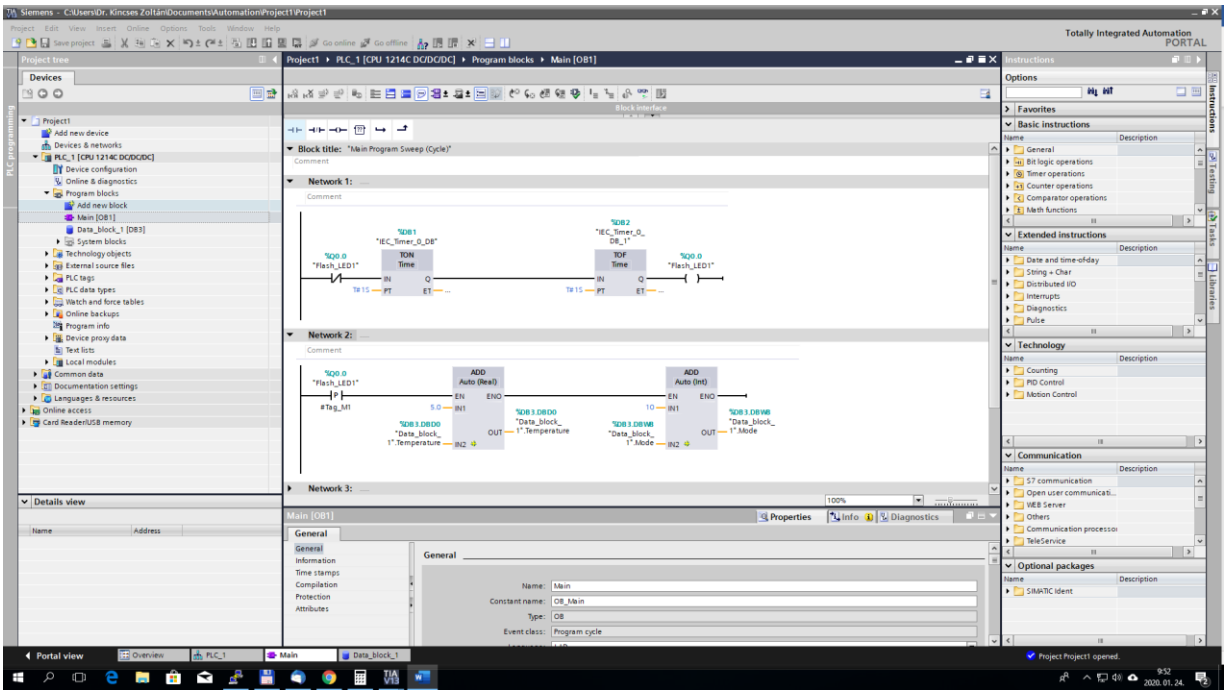
7. Create a data block and add some variables (Temperature, Trig, ID. Mode). The node-red can only read and write the variables created in a Data block.



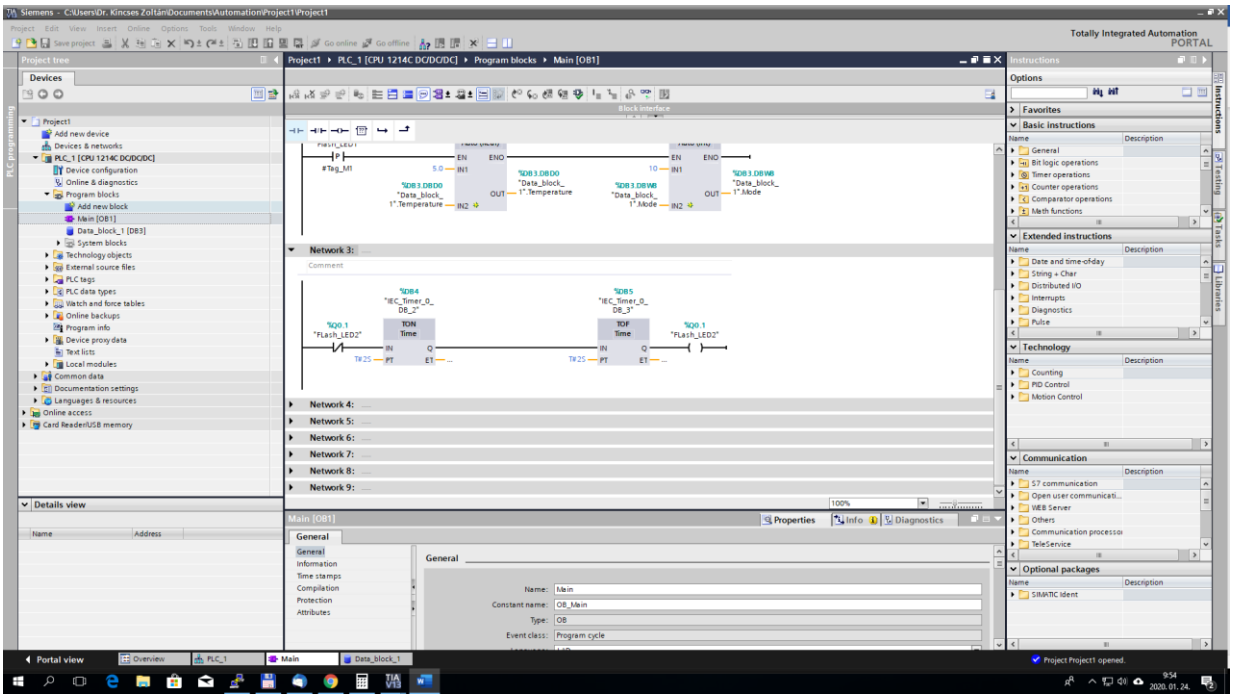
8. Create a 2 second fill factor control signal. The Flash_LED1 is required only for debugging purpose (it is used to flashing led on the S7-1200 PLC).



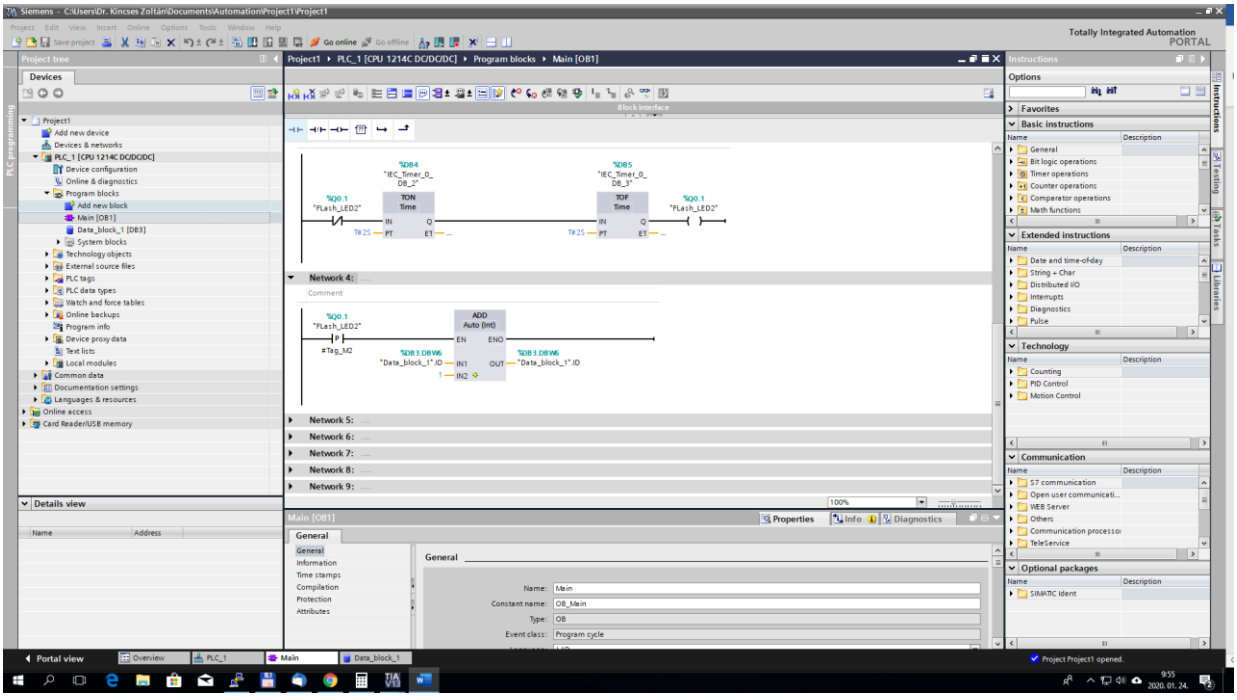
9. Using the control signal increase the Temperature and Mode variables with 5 and 10 respectively.



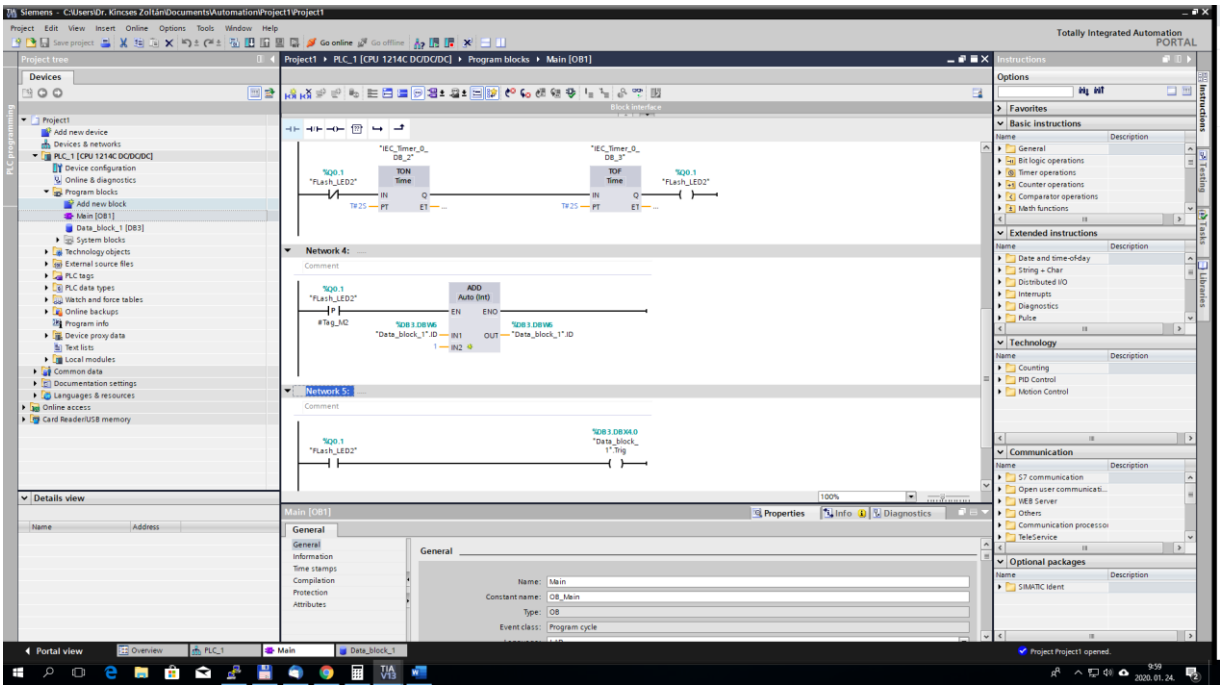
10. Add another 4 second fill factor control signal. Using this signal the slower signal changes can also be simulated.



11. Using the previously created control signal increase the ID variable with 1.



12. Change the Trig signal together with the led flashing signal.

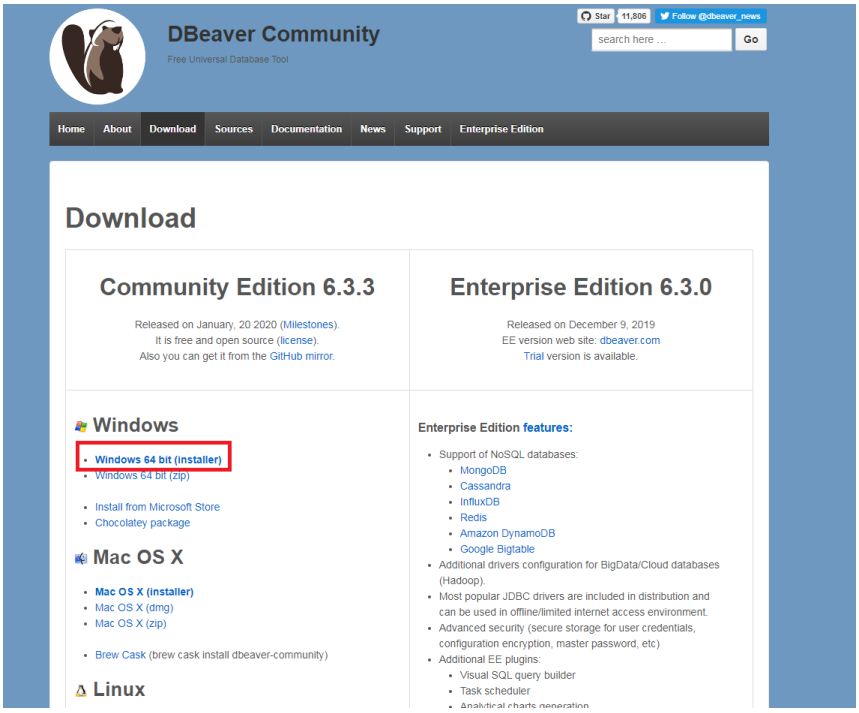


Optionally you can add some network to reset the variables using the switches connected to the S7-1200. Using this test program, the changes of the variables with different speeds can be simulated. The ID variable can be used to identify when new data is generated and need to be saved to the database on the IoT Gateway.

Task 2.

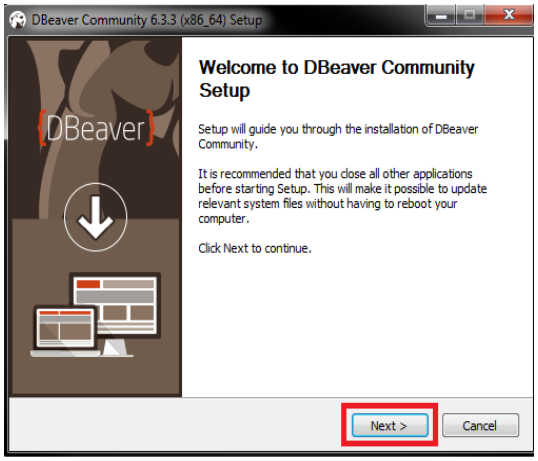
Before installing DBebaver on your local machine start the PLC and IoT Gateway.

1. Download DBeaver from https://dbeaver.io/download/.
   Choose the suitable installer for your operating system.

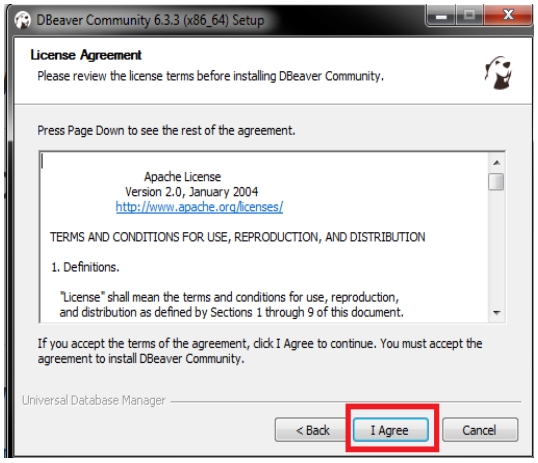


2. After DBevaer downloaded successfully start the installer exe file.

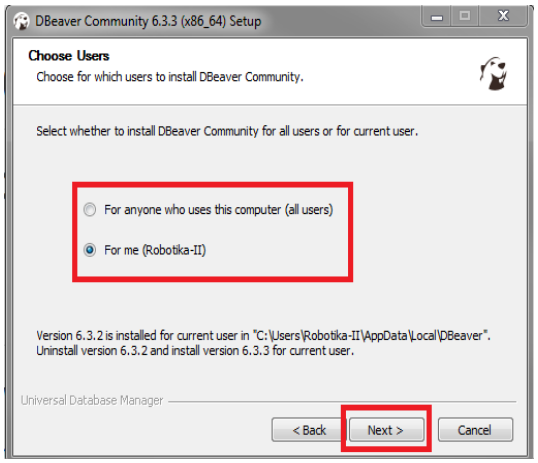
a. Select the language (default is English) and press the **OK** button
b. After we choose the language a welcome screen appear



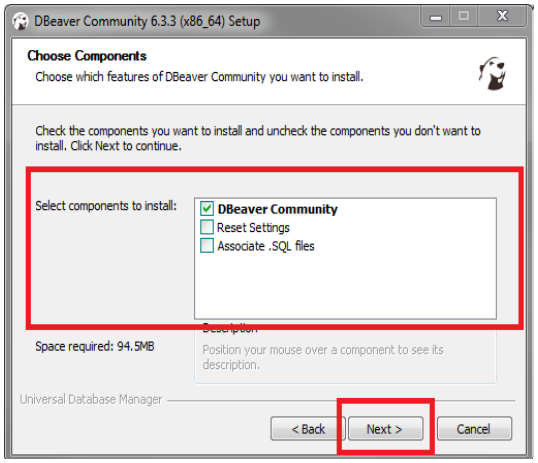
c. Agree the License Agreement



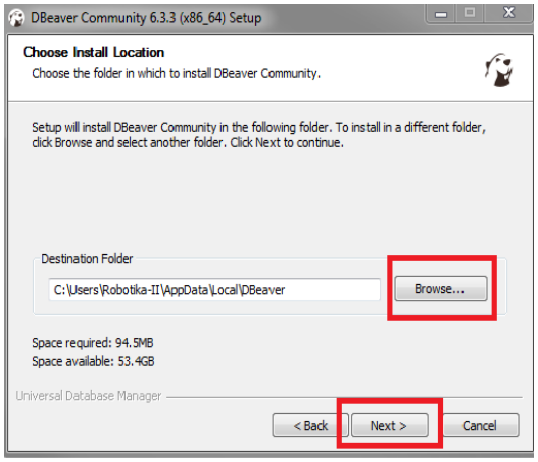
d. Choose for which users to install the Software and press next



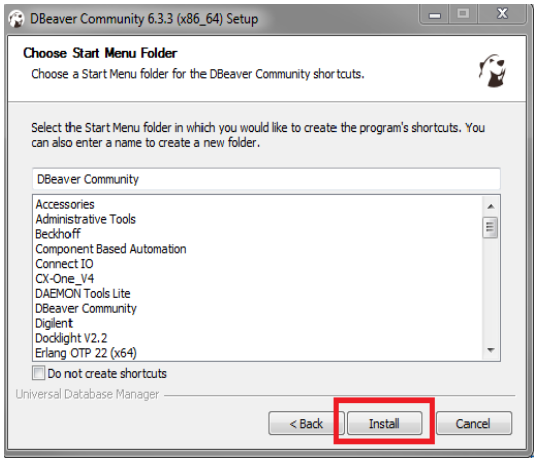
e. Choose which features of DBever you want to install. The default is only the software. It’s enough for us.



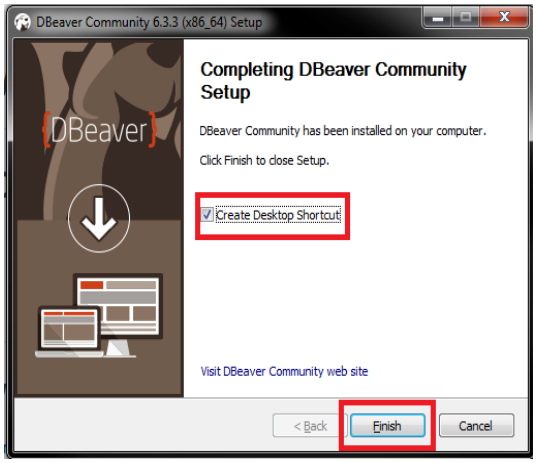
f. Select where you want to install DBeaver. For installing DBeaver 94.5MB free space is required



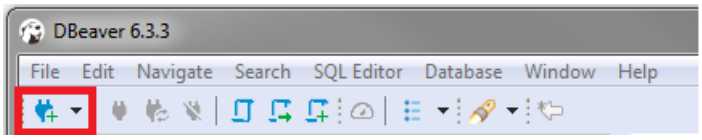
g. Finally press the **Install** button for start of installing



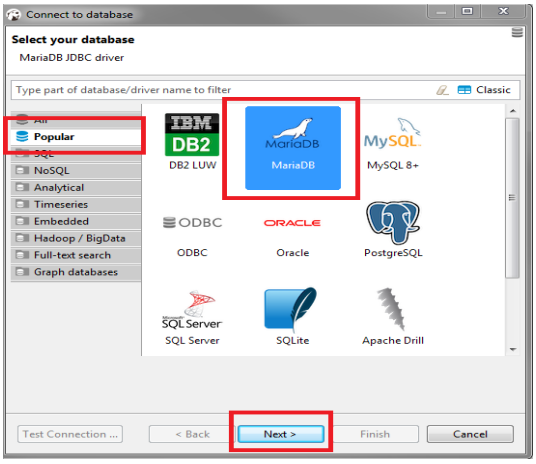
h. After the installing finished press the **Finish** button. (Recommended to Create Desktop Shortcut icon)



3. After DBeaver installed successfully start this program for reach the MYSQL server on the IOT Gateway
    a. Start the program
    b. First we need to setup the connection. For this we have to create a new Database Connection
        i.

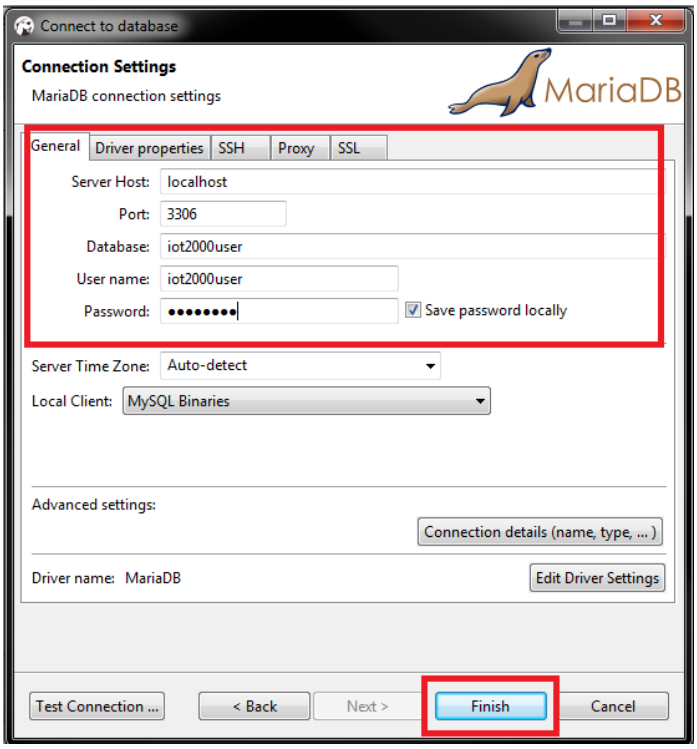


        ii. Choose MariaDB from **Popular** (Default) section. And press **Next** button.

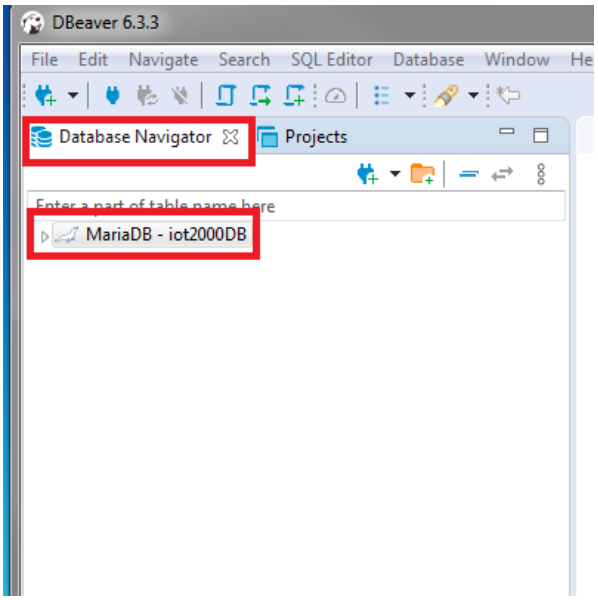


        iii. Now, we need to set up for our connection to MaiaDB server on IOT Gateway
        Server Host: *192.168.1.122*
        Port: *3306 (default of MariaDB)*
        Database: *which you choose when install mariadb to IOT Gateway (for us: iot2000DB)*
        User name: *which you choose when install mariadb to IOT Gateway (for us: iot2000user)*

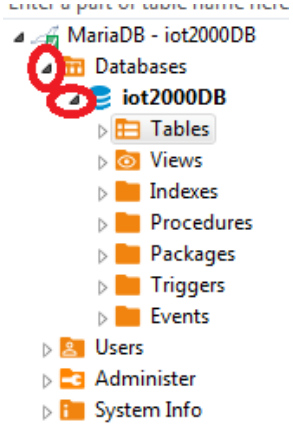
Password: which you choose when install mariadb to IOT Gateway



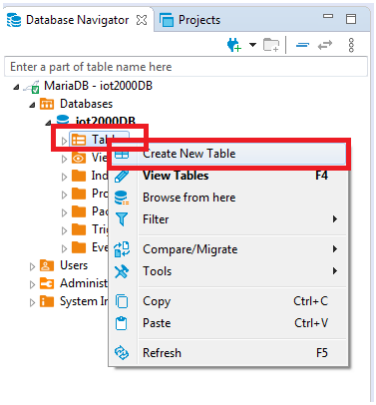
iv. If everything is okay a new connection added to Database Navigator. Double click on the connection and we can manage our database



v. Open down Databases and after the Database

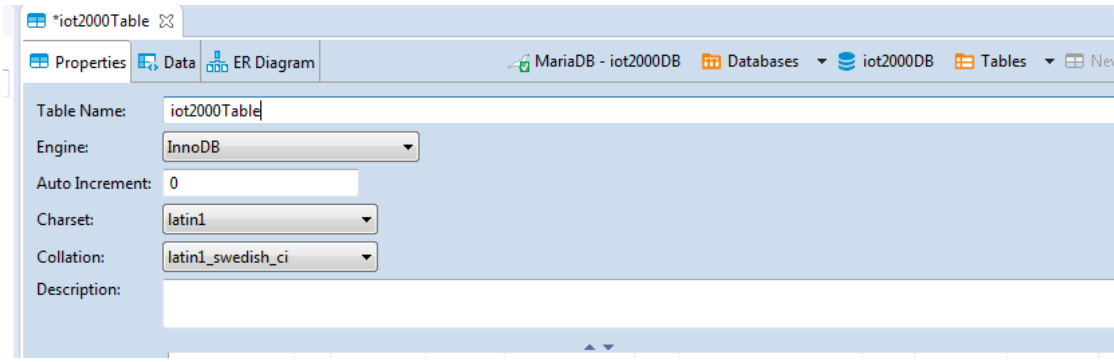


vi. Right click on Tables and select the **Create New Table option**



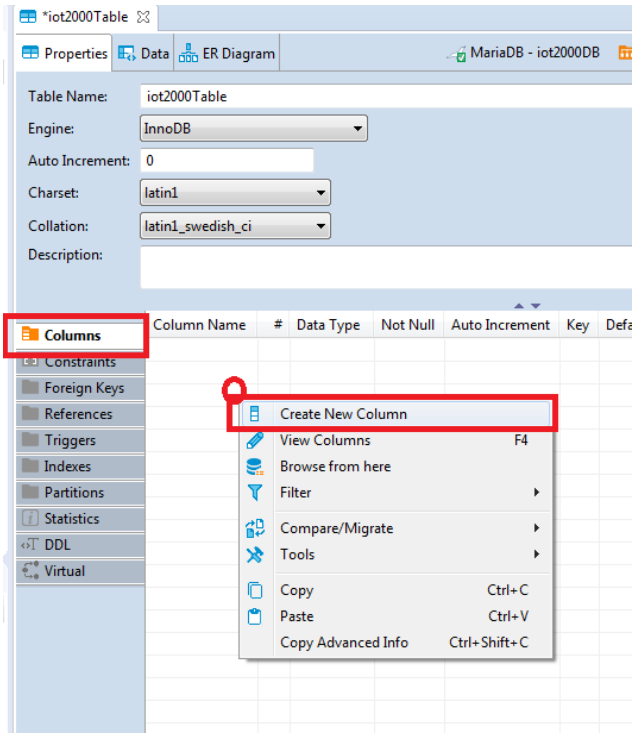
vii. First we set up our table's attributes

Table name: *Choose what you want (for us iot2000Table)*
Engine: *InnoDB which is the default*
Auto Increment: *0 which is the default*
Charset: *Choose which suitable for your data (for us latin1 (default))*
Collation: *Choose which suitable for your data (for us latin1_swedish_ci (default))*
Description: *Not required*

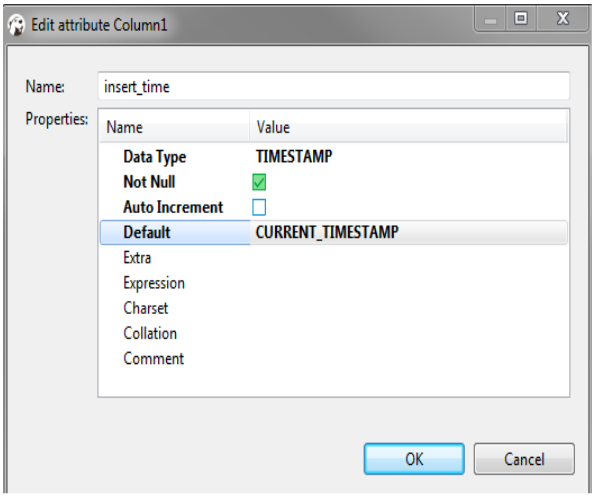


viii. Now we can add our columns to this table. For this we choose **Columns** from submenu. On the right section of this panel **right click**

and select the **Create New Column menu**.



ix. On the popup window we can set up our column attributes
First we add a time stamp column to our table
Name: *insert_time*
Data Type: *TIMESTAMP*
Not Null: *checked*
Auto Increment: *unchecked*
Default: *CURRENT_TIMESTAMP*
After we finished editing the attribute press the OK button



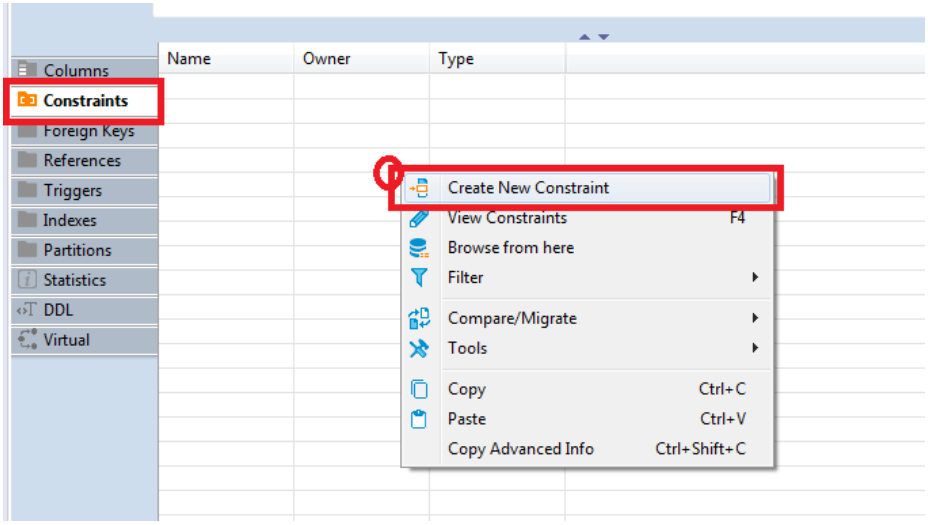
x. We **repeat viii-ix steps** with attributes below
    1. **Name:** *Temperature*
    Data Type: *float*
    Not Null: un*checked*
    Auto Increment: *unchecked*

2. **Name:** *Mode*
   Data Type: *INT*
   Not Null: un*checked*
   Auto Increment: *unchecked*
   Default:
3. **Name:** *Trig*
   Data Type: *BOOL*
   Not Null: u*nchecked*
   Auto Increment: *unchecked*
   Default:
4. **Name:** *ID*
   Data Type: *INT*
   Not Null: *checked*
   Auto Increment: *unchecked*
   Default:

xi. We need a Primary Key for our later work.
About Primary Key:

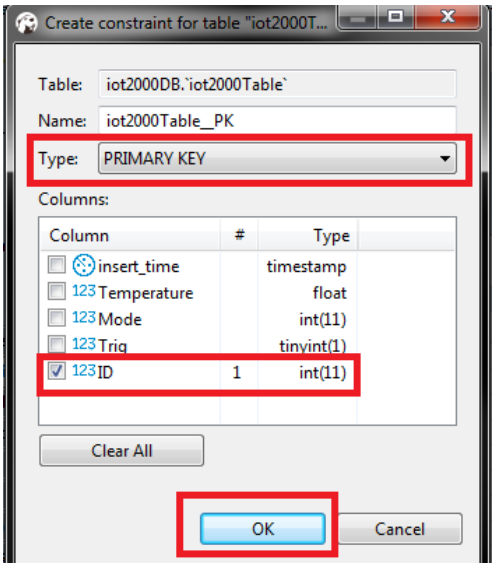
The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

We select ID column as our Primary Key
We add our primary key in Constrains sub menu. On the right section of this panel right click and select the Create New Constraint menu.



xii. On the popup window we can set up our constraint. Recommended to use defaults
Table: *Default is your table (for us iot2000DB.`iot2000Table`)*
Name: *Default suite for your table (for us iot2000Table__PK)*
Type: *PRIMARY_KEY (default)*
Columns: *Checked your column which you selected as primary key (for*

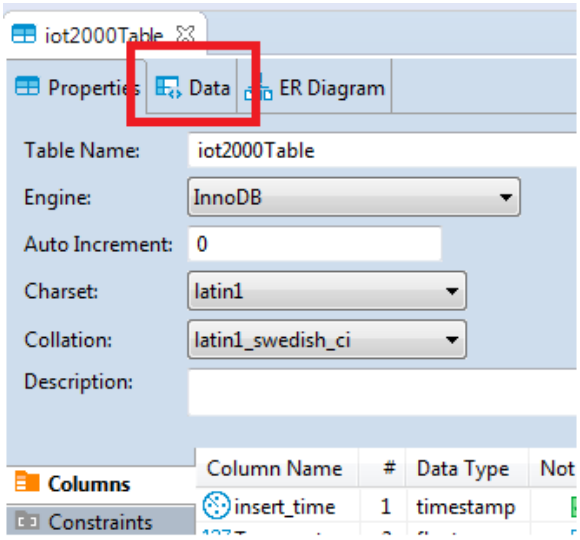
us ID)



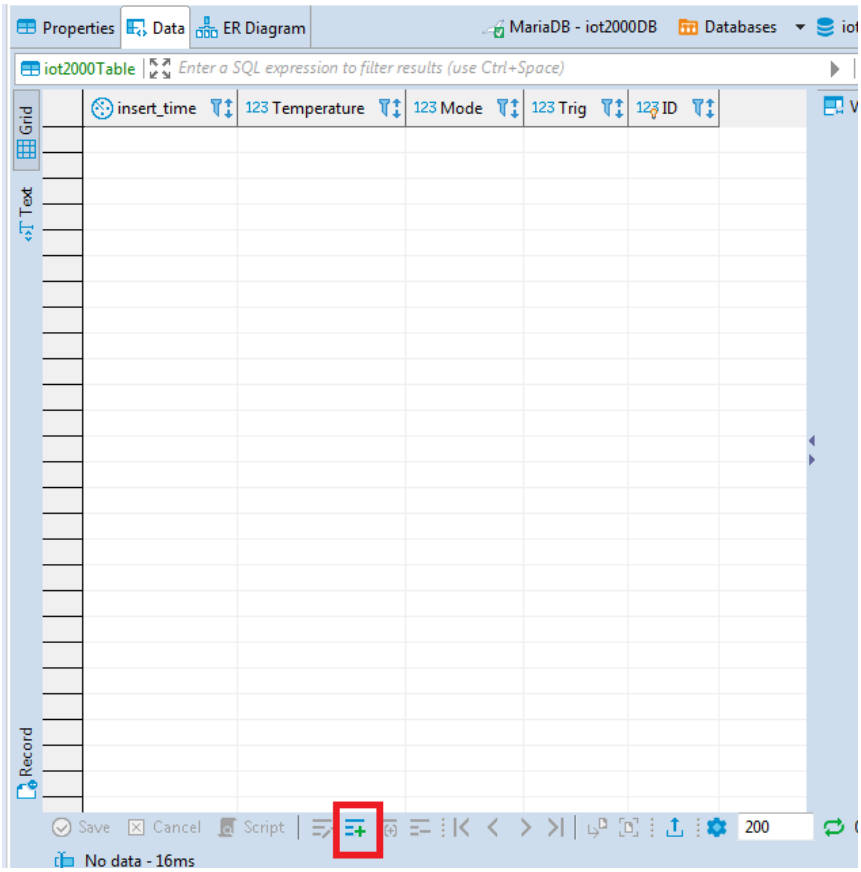
xiii. Press Ctrl+S for save our table. A popup window will be appear with the SQL command which is used for create the table. We don't need any editing in this just click the **Persist** button.

xiv. For our later work we add a record to our table.

1. Click the Data menu

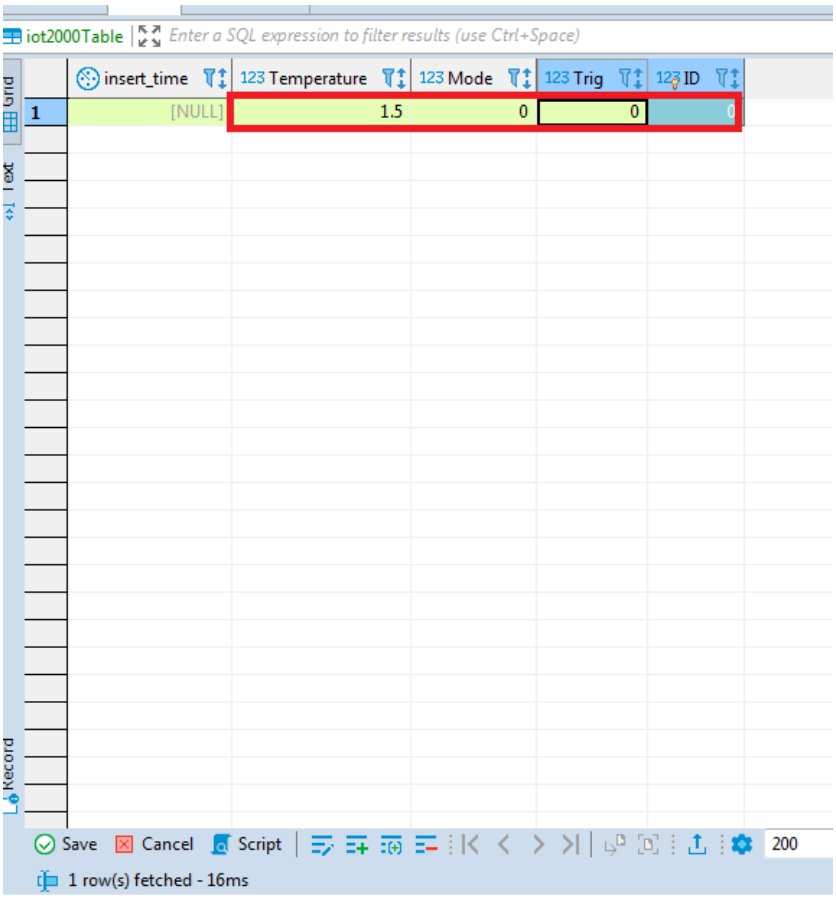


2. On the bottom of this panel we click **Add New Row** icon or press **Alt + Insert**



3. A new empty row will be added to the table. Just click to the NULL label and type what you want

The insert_time must be NULL (it's an auto generated value)



4. Press **Ctrl+S** to save our row.

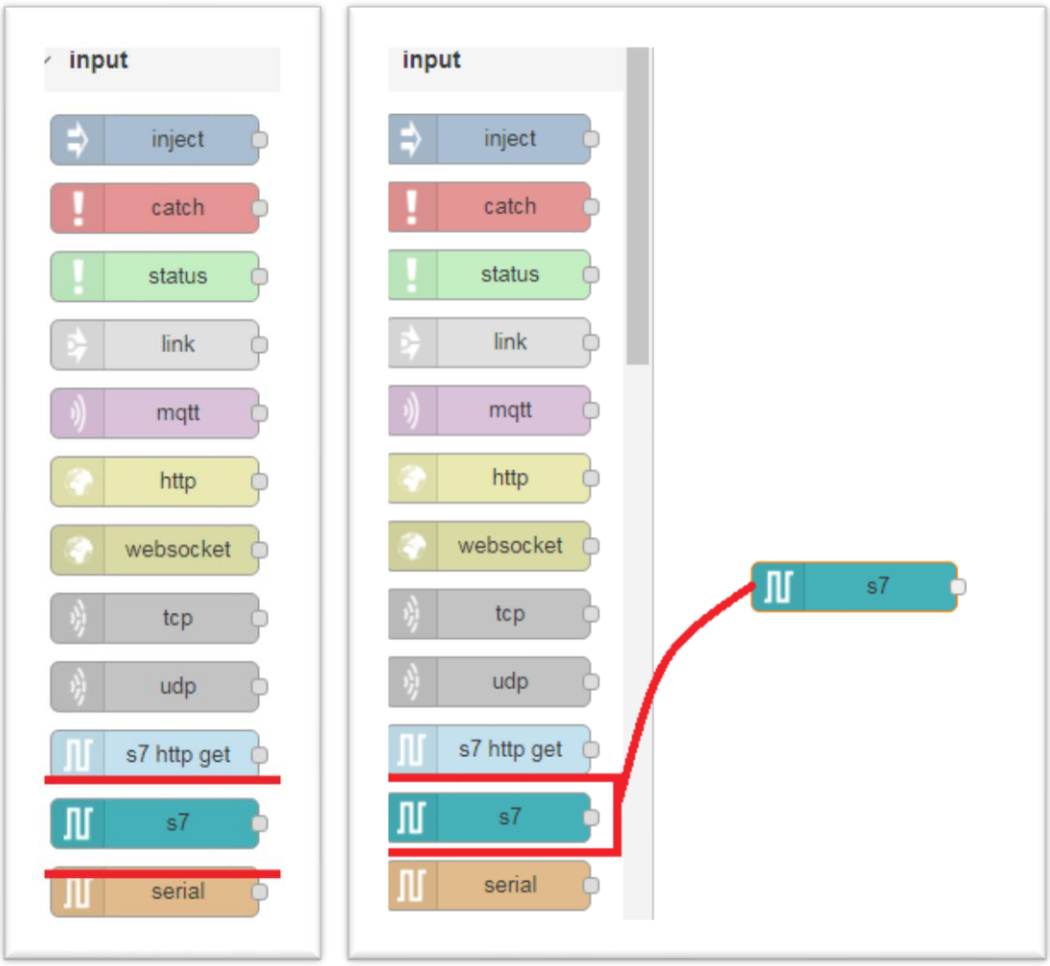
Task 3.

Turn on the PLC and IoT Gateway and wait for its booting. After PLC booted the Nod-Red software and MYSQL server will be started automatically. It takes 5-6 minutes. For this task we need to finished Task 2.

First step creating a node-red application to connect s7-1200 PLC

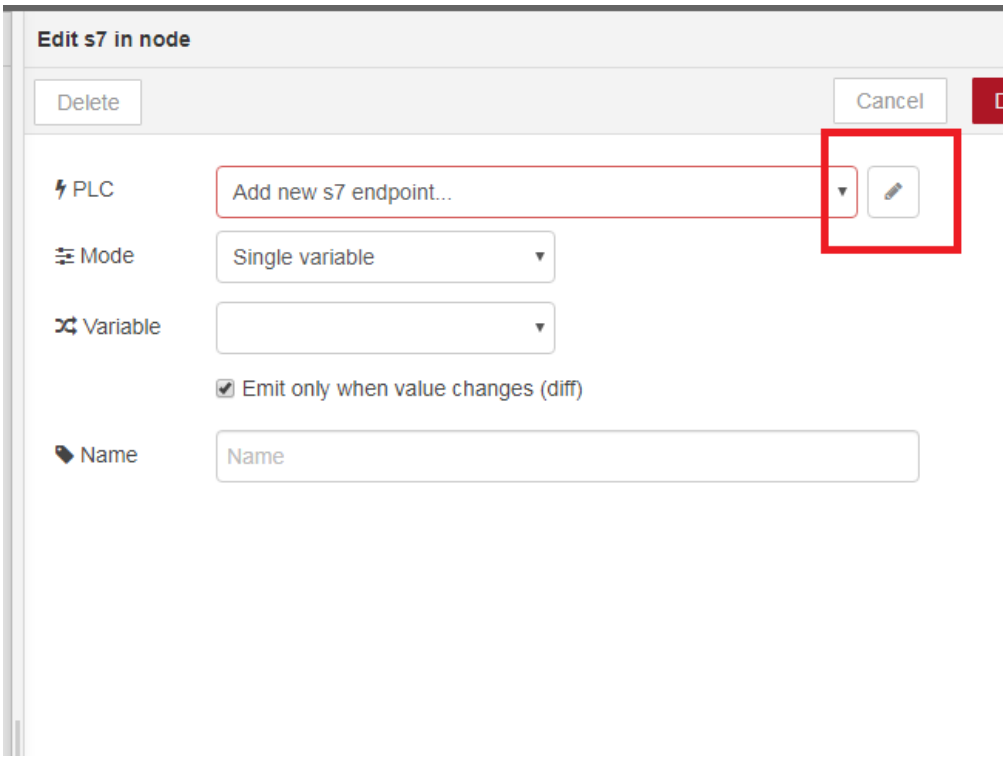
Open Node-Red Dashboard on http://192.168.1.122:1880/ link.

1. After Dashboard is loaded on left side from the input section choose s7 Node and Drag and Drop to the flow on right side

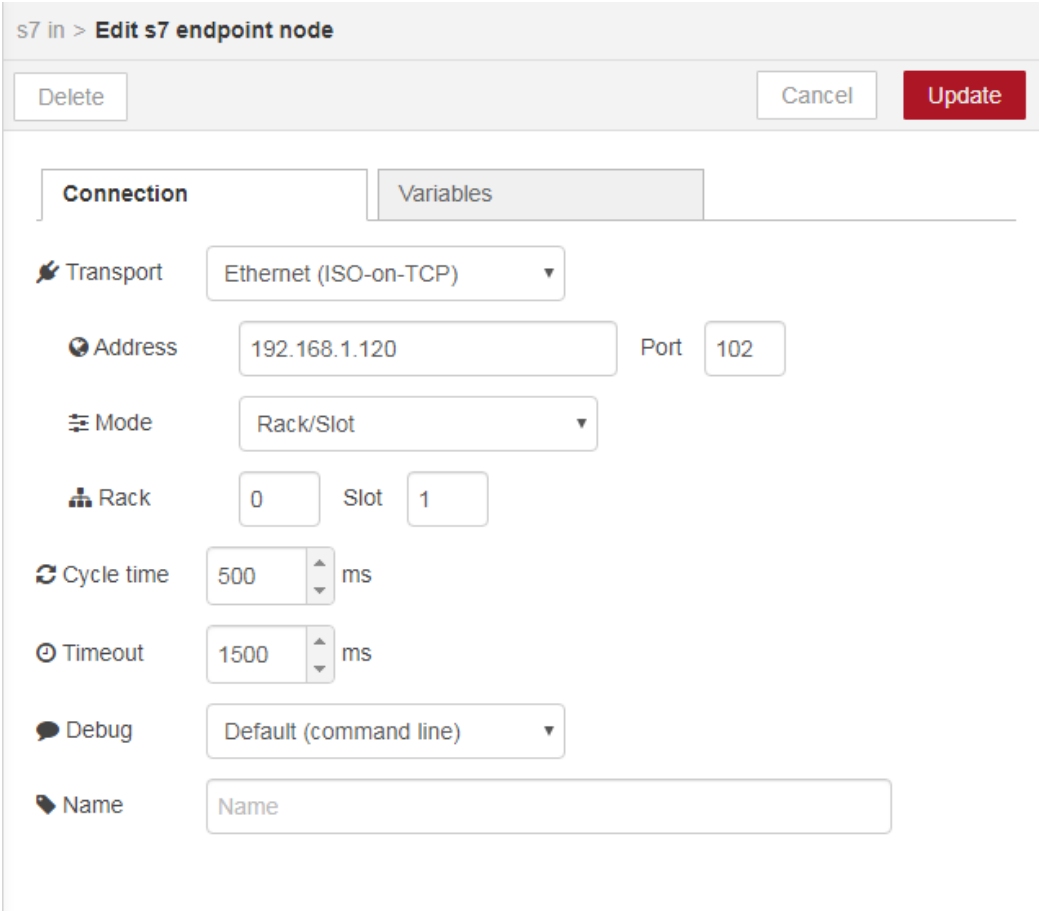


2. Double click on s7 node to set up our connection

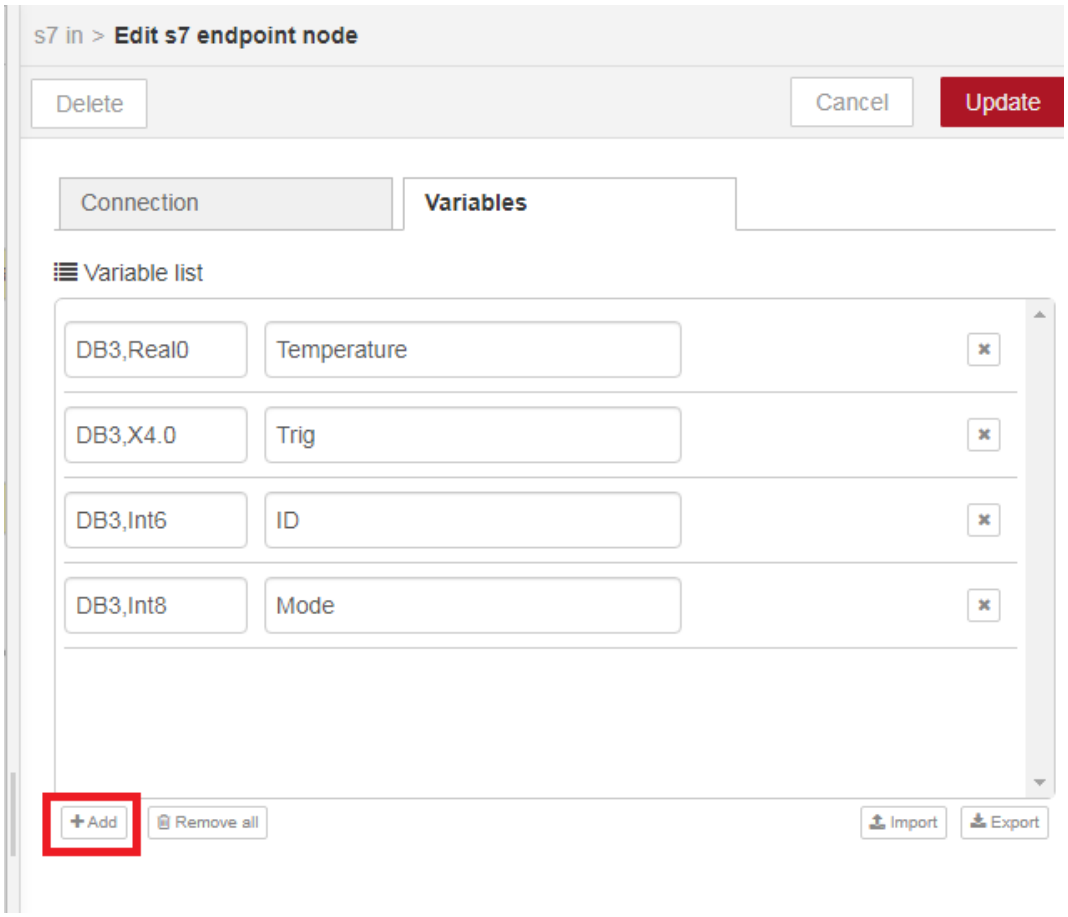
a. First we have to add a connection to our PLC



i. Set up the Connection on the **Connection** panel

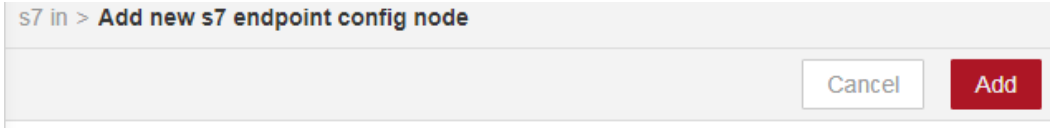


ii. Set up Variables on the **Variables** panel



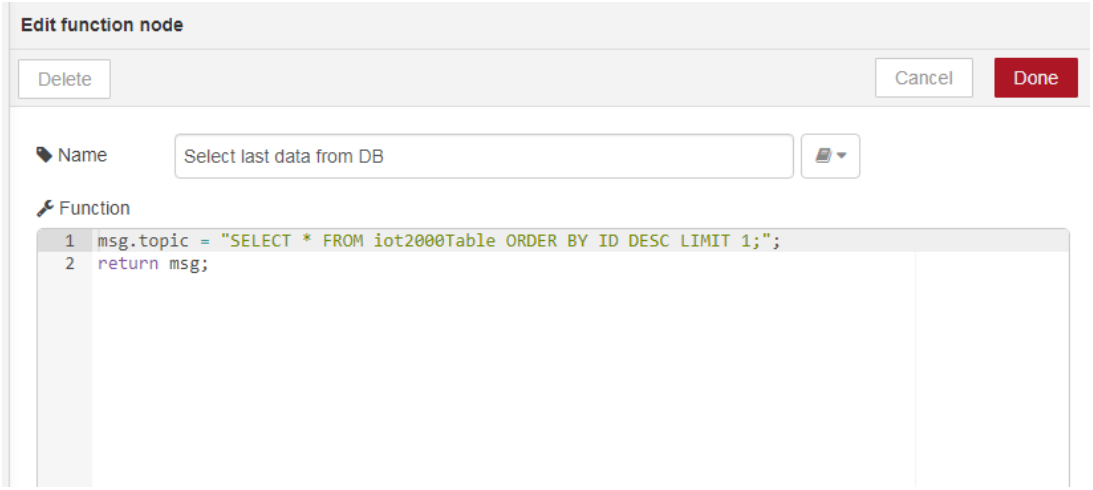
With **Add** button add a new Variable to Variable list

iii. Press **Add/Update** button



b. We add a function block which contains our SQL command. On left side from the function section choose function Node and Drag and Drop to the flow on right side.
This command selects the last element from the database (DB). It's important, because we insert a new value to DB if it is not already in.

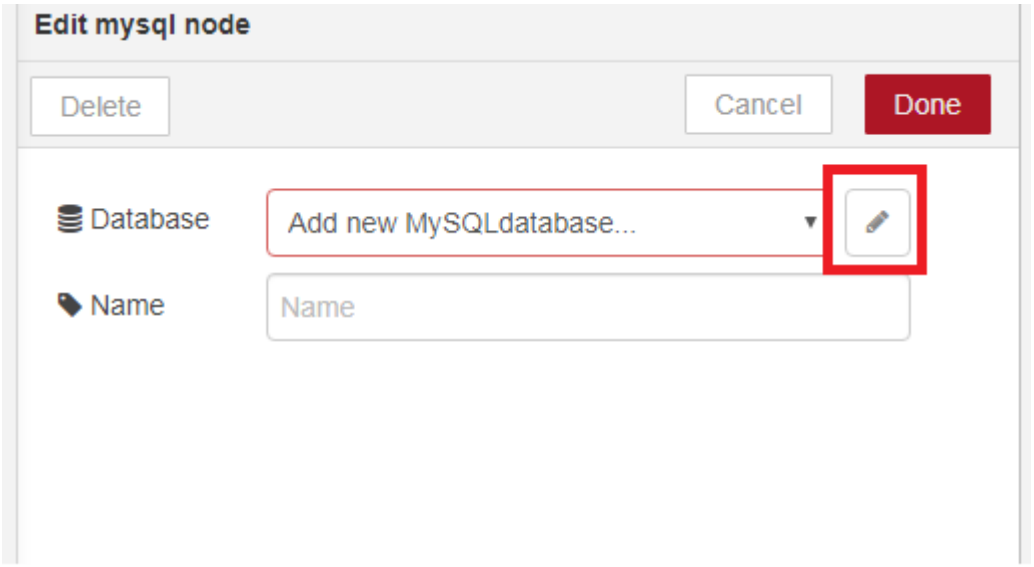
i. Double click on the new node of the flow and edit it



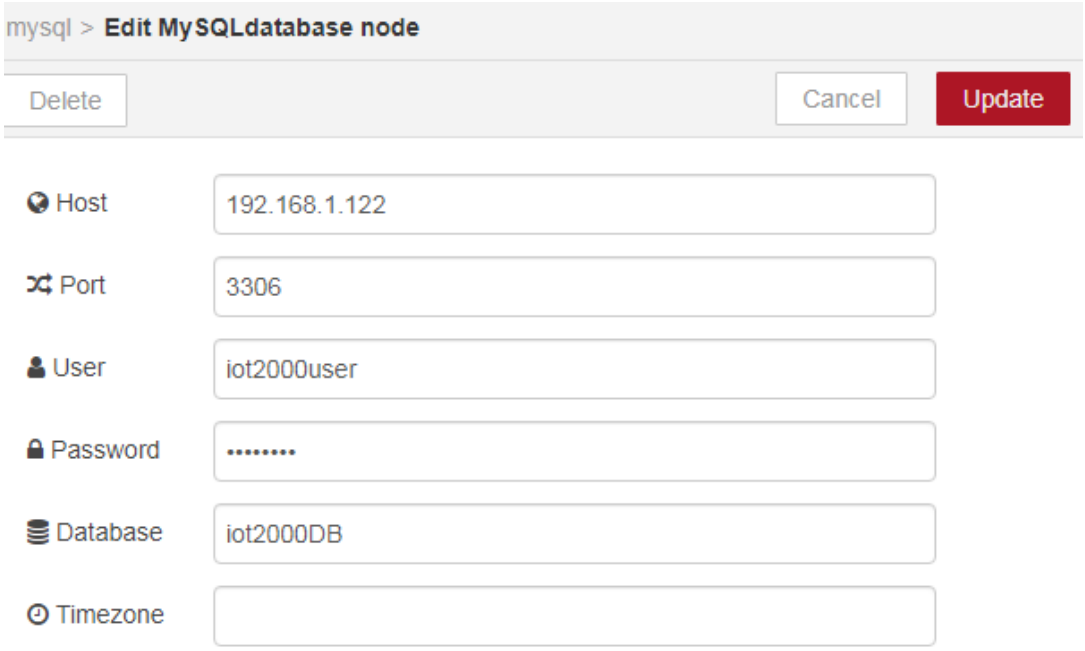
After press **Done** Button

c. Now we need a connection to our DB. For this on left side from the storage section choose mysql Node and Drag and Drop to the flow on right side.

i. Double click on the new Node and add a new connection

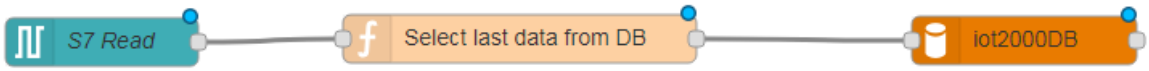


ii. Set up DB connection and press **Add/Update** button



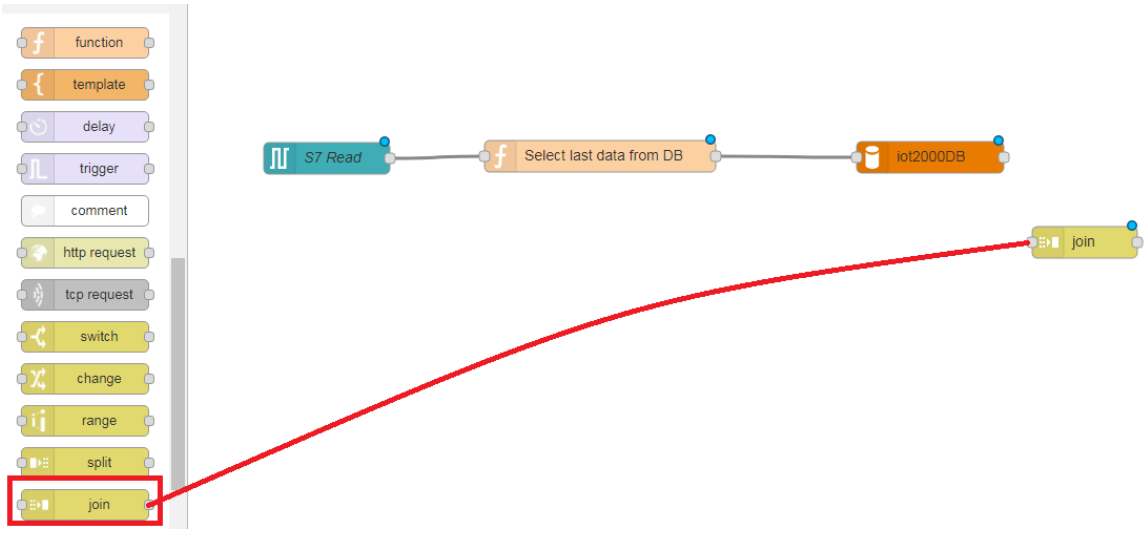
iii. Press the **Done** button

d. Now we need a connection between our 3 Nodes. For this you have to click the grey boy on right side of the node and move your mouth another node's grey box on left side. We need joins like this:

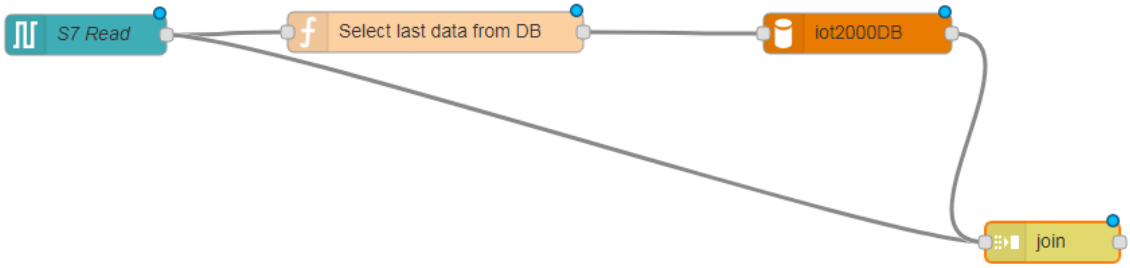


e. We need to join our Data from S7 and From Database. For this on left side from the function section choose join Node and Drag and Drop to the flow on right

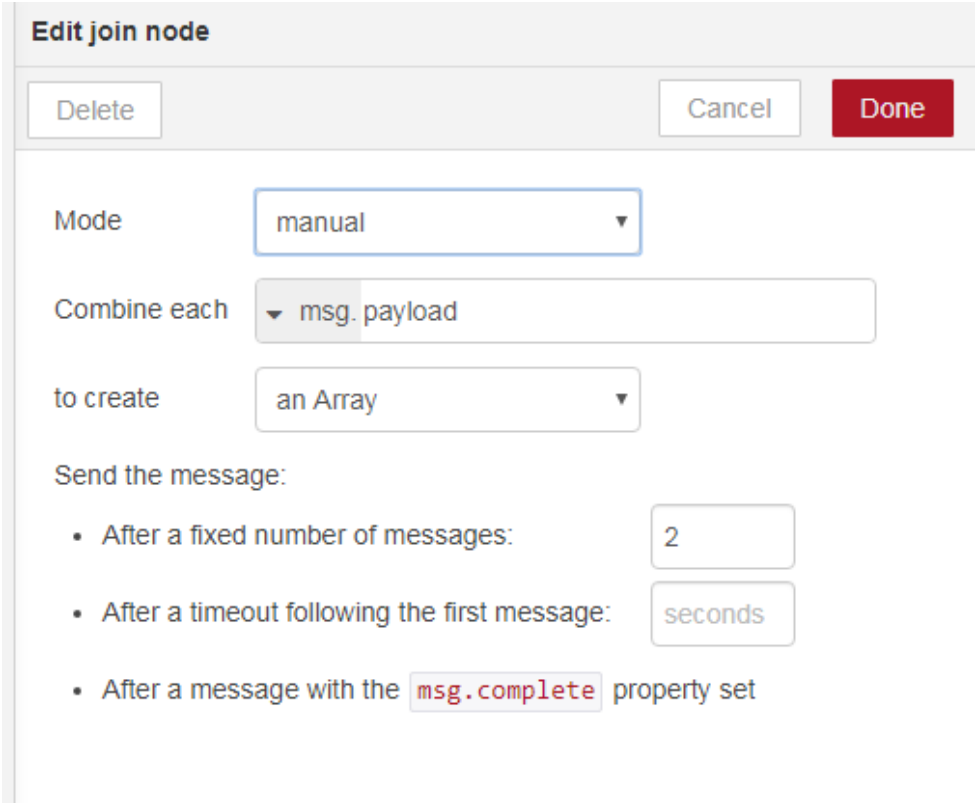
side.



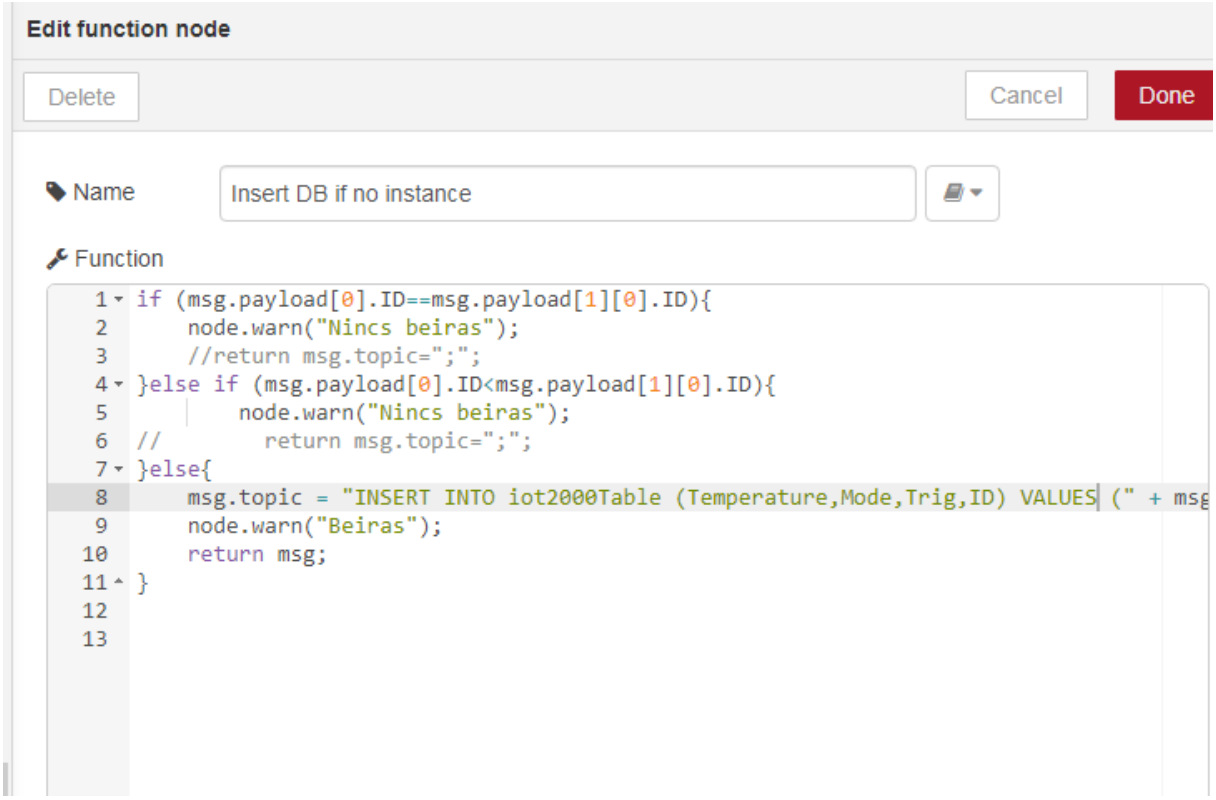
f. Make connections from S7 node and from MYSQL node to this join node like this:



g. Edit your join node (double click on it) as you can see above and press **Done** button



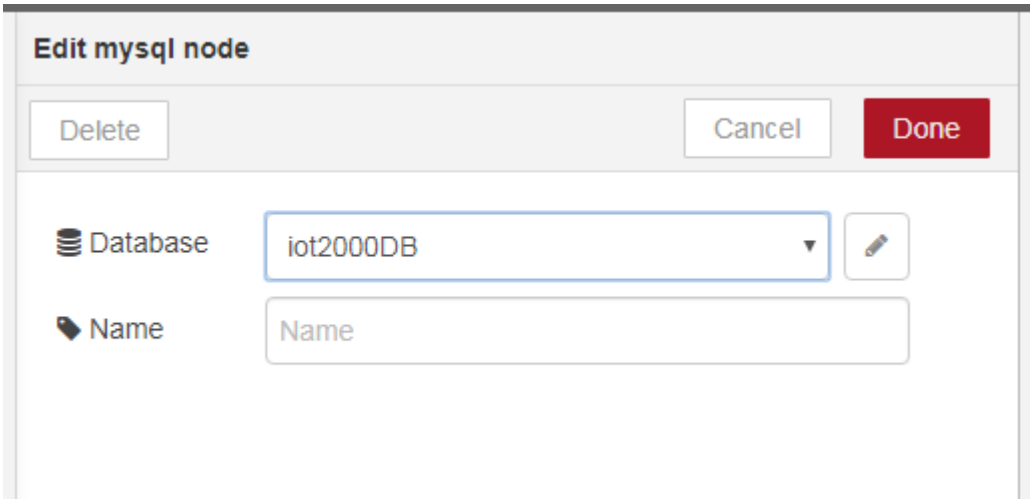
h. Now we need a new function block to add to our flow. This function block set up:



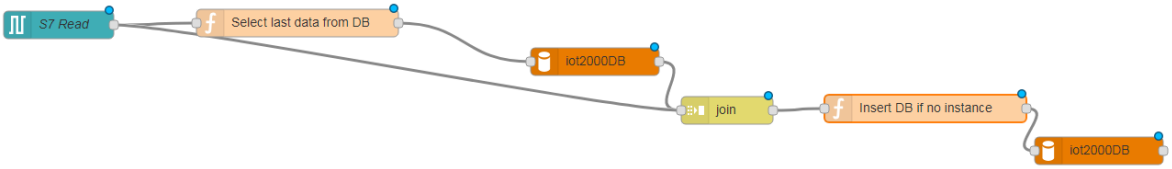
Code for this block:

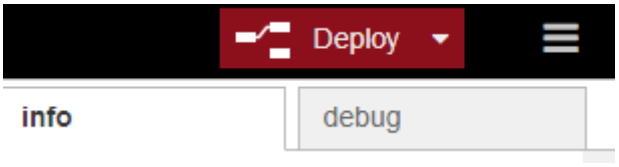
```
i
   if (msg.payload[0].ID==msg.payload[1][0].ID){
ii node.warn("Nincs beiras");
iii //return msg.topic=";";
iv }else if (msg.payload[0].ID<msg.payload[1][0].ID){
v node.warn("Nincs beiras");
vi //        return msg.topic=";";
vii }else{
viii msg.topic = "INSERT INTO iot2000Table (Temperature,Mode,Trig,ID) VALUES (
   " + msg.payload[0].Temperature + "," + msg.payload[0].Mode + "," + msg.pa
   yload[0].Trig + "," + msg.payload[0].ID + ");";
ix node.warn("Beiras");
x return msg;
xi }
```

i. Now we need a new MYSQL Node. Set up like below and press **Done** button:



j. Make connections between nods as below and press deploy button





Now our Node Red app save data from S7 to our database on IOT Gateway

INNOWWIDE report 2020 February

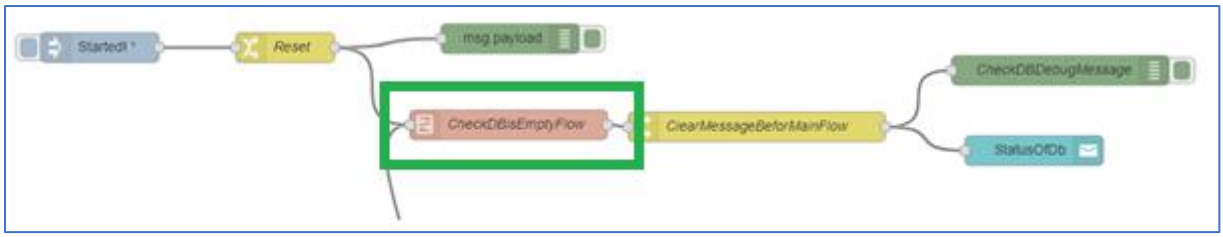
In this month the following tasks have been completed:

1. Creating a node-red application to connect the S7-1200 PLC and save data to the MySQL database and an external database via REST-API

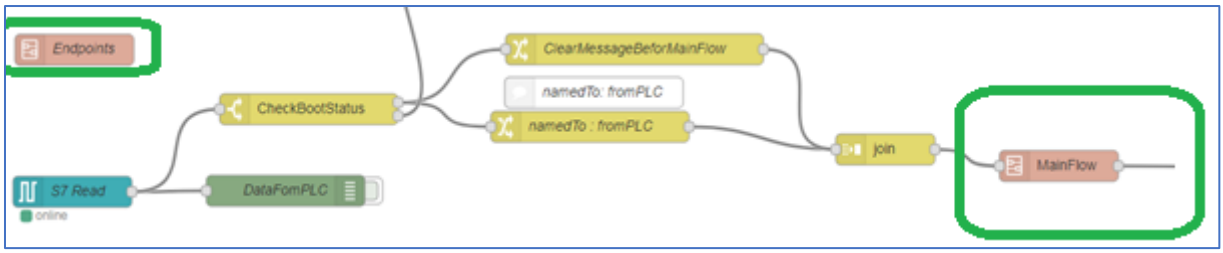
Task 1.

Turn on the PLC and wait for its booting. After PLC booted the Nod-Red software and MYSQL server will be started automatically. It takes 5-6 minutes. For this task we need to finished Task 2. First step creating a node-red application to connect s7-1200 PLC Open Node-Red Dashboard on http://192.168.1.122:1880/ link. Creating a Node-Red app which checks the status of the database, if its empty create a new empty line for the comparation of data from PLC and the Database. Save the new data to local mariaDB and if we have 5 or more data in our database, send it to an external database via API calls.

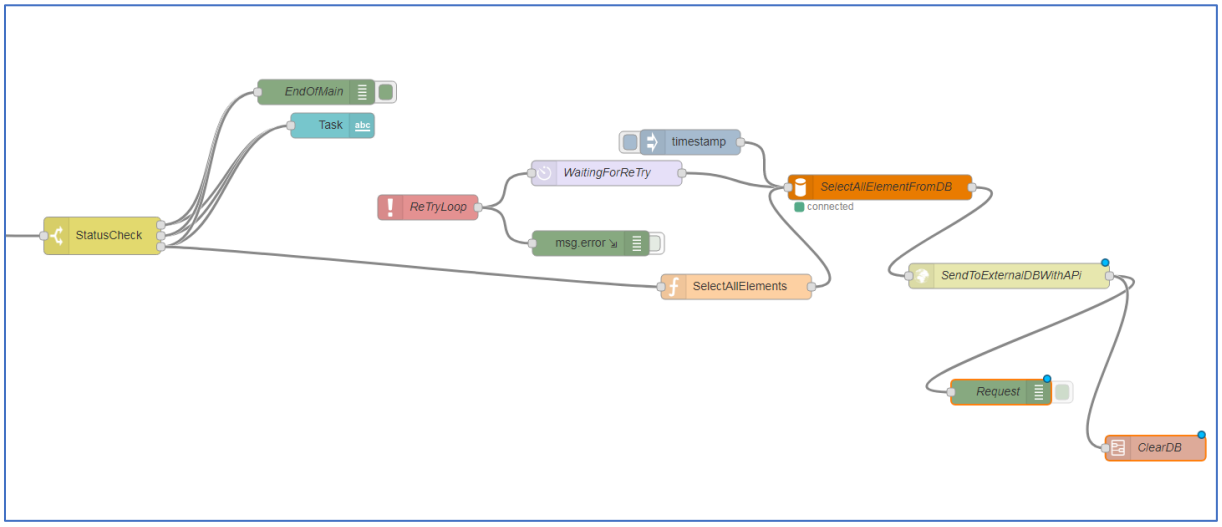
The full application shown on the following picture.



1. Figure: Full application (Program starter section)



2. Figure: Full application (Presetting's of main flow)



3. Figure: Full application (Sending data to external database)

On the pictures above we mark the subflows with green rectangle. The different subflows include the program logically unique sub programs.

The Program starter section.

When the node red started the first step is resetting of system Booted flag. This flag is true if the application start successfully means it connect to the local MariaDB and the **CheckDBisEmptyFlow** executed.

In the **CheckDBisEmptyFlow** we try to connect to the local database. We build a reconnect loop in the code. If the connection stabile, we select all rows from database. If we find at least one row we don't do anything. If the database is empty we insert a new empty line.

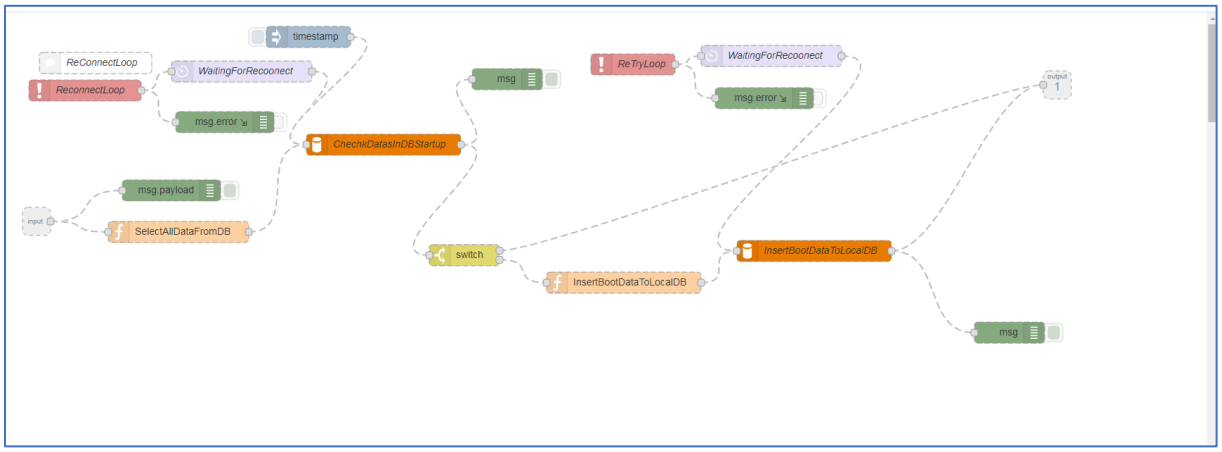
"INSERT INTO iot2000Table (Temperature,Mode,Trig,ID) VALUES (0,0,0,0);"

This flow important to the stabile working of our program, because we will compare the ID of last row with the ID from PLC data to decide, what will do with the data.

3 different ways:

- *NOINSERT* : in this case we don't do anything with the data from PLC
- *INSERTTODB*: in this case we save the data from PLC to local MariaDB
- *INSERTTOCLOUD*: we try to sand the existing data to the external database via api call.

The code of **CheckDBisEmptyFlow:**



4 Figure: CheckDBisEmptyFlow code

Presetting's of main flow

Before the main flow we do some data manipulation to make easier of the data processing in main flow.

The first thing if the PLC want to send a data on S7 port we check the status of *systemBooted* flag. If it's not true we jump to **CheckDBisEmptyFlow** detailed above. In that case if it's true we named the data from PLC to *fromPLC* and send it to **Main** flow. We check the boot status inside the **Main** flow again. For

this check we create a Boolean data named *fromBoot* and set it true (**ClearMessageBeforMainFlow**).

The endpoints subflow contains two API endpoints:

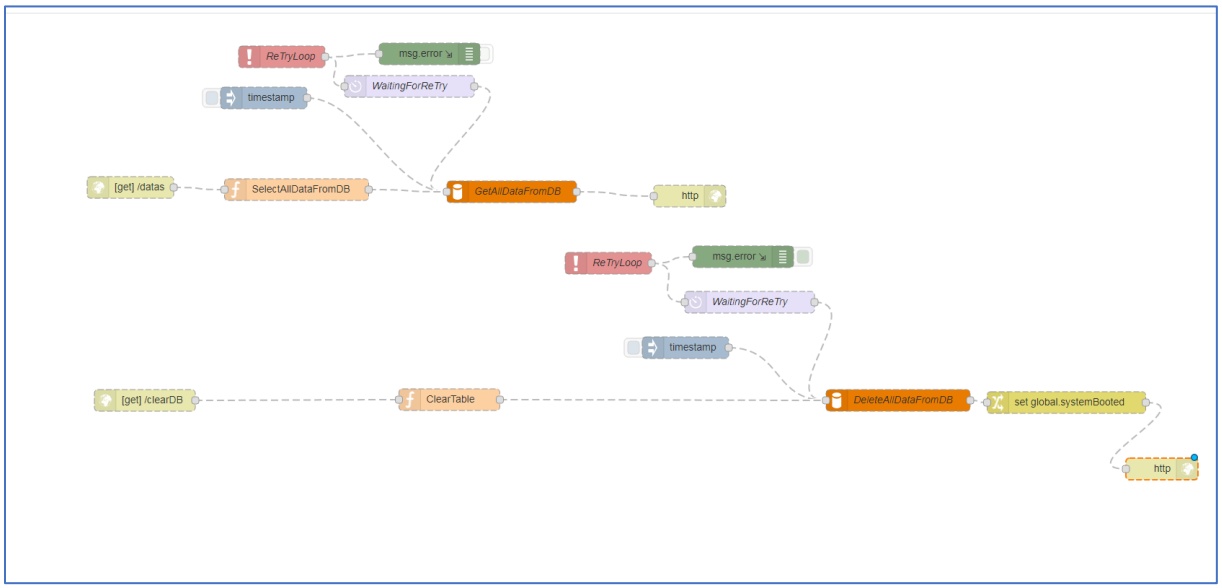
- http://192.168.1.122:1880/datas

  Get request: List all data from local database

- http://192.168.1.122:1880/clearDB

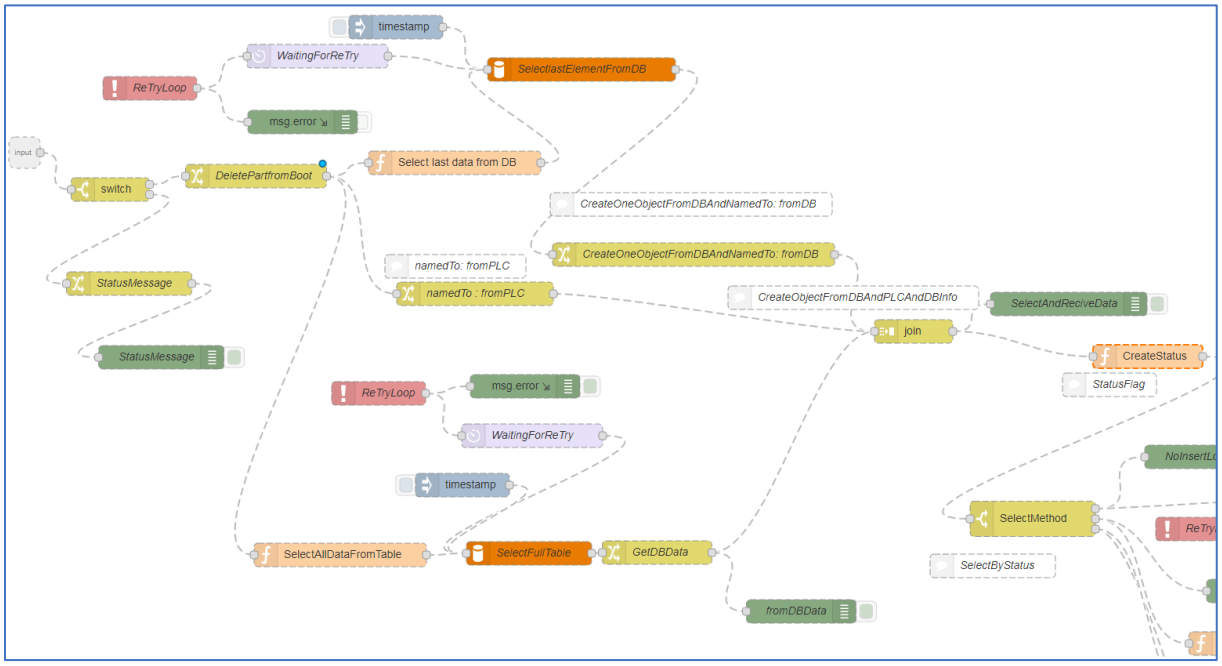
  Get request: Delete all data from local database

Use of these endpoints not recommended. It's just for testing the software.



5 Figure: Endpoint subflow

The main flow



5 Figure: Main flow

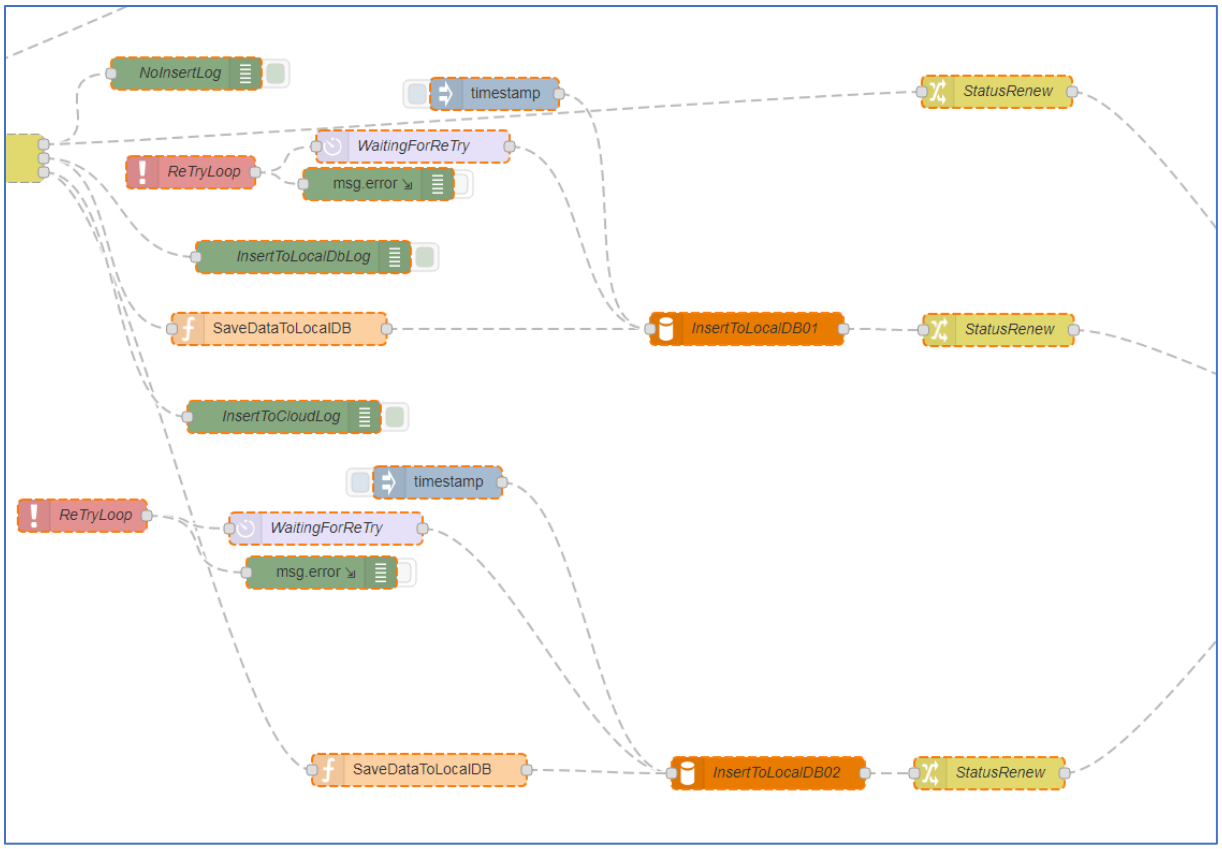
After the check of the value of *fromBoot*, we select the last row from database. The last row from data named to *fromDB*, after we create an object from array. In ***CreateObjectFromDBAndPLCAndDBInfo*** section we make one object from fromPLC and fromDB with the join function block.

In the ***CreateStatus*** function block we decide what will be do with our data with the following algorithm:

```
if(msg.payload.fromDB.ID > msg.payload.fromPLC.ID)
{
    msg.payload["status"] = "NOINSERT";
    msg.payload["message"] = "We don't save anywere";
}
else if((msg.payload.fromDB.ID <= msg.payload.fromPLC.ID) && msg.payload.fromDBData.length <= 5)
{
    msg.payload["status"] = "INSERTTODB";
    msg.payload["message"] = "We save to local DB";

}
else if((msg.payload.fromDB.ID <= msg.payload.fromPLC.ID) && msg.payload.fromDBData.length > 5)
{
    msg.payload["status"] = "INSERTTOCLOUD";
    msg.payload["message"] = "We try to save into external DB";

}
else
{
    node.error("Unexpected state!");
}

return msg;
```

6 Figure: CreateStatus function block algorithm

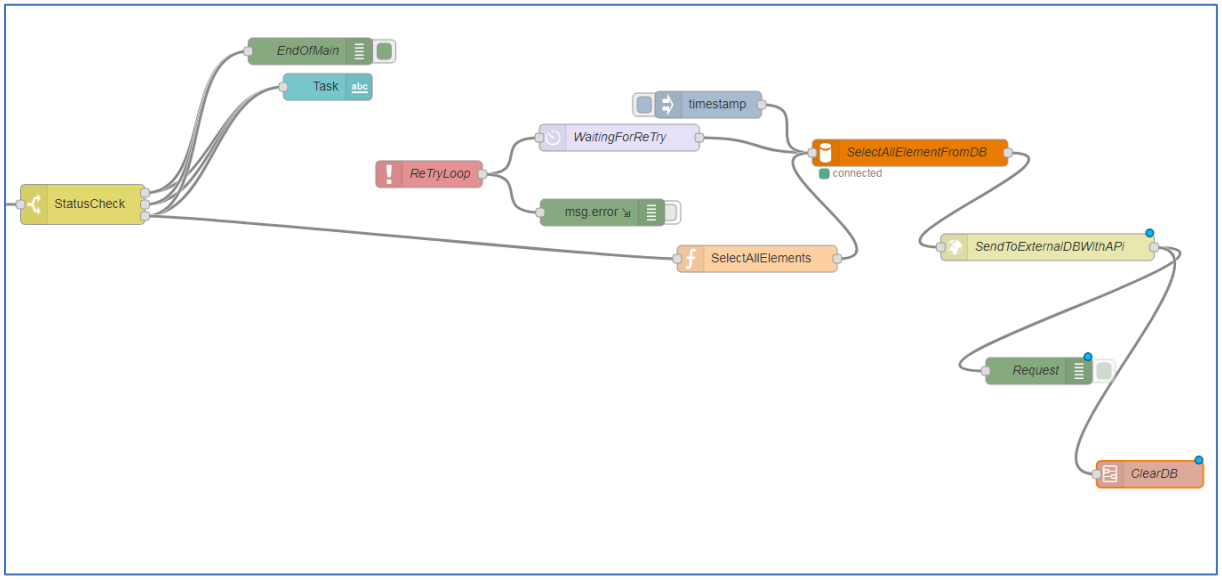
- *NOINSERT*: in this case we don't do anything with the data from PLC
- *INSERTTODB*: in this case we save the data from PLC to local MariaDB
- *INSERTTOCLOUD*: we try to sand the existing data to the external database via API call.



7 Figure: StatusRenew flow

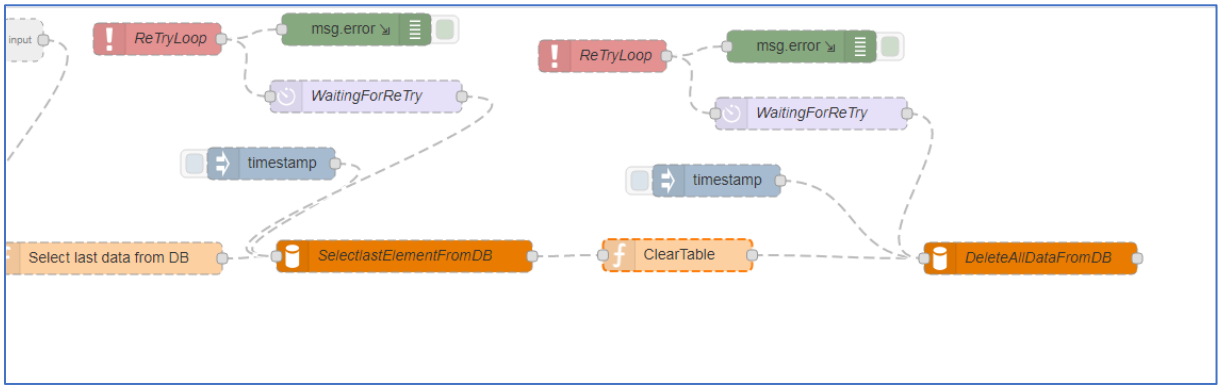
The **StatusRenew** just renew the status. We need for it because; the sending of data with API calls not in Main flow. This is in the main flow of the app.

Sending data to external database



8 Figure: Sending data to external database

The main flow give back what will we have to do with the data. If it's *INSERTTOCLOUD* we select all data from database, and we create a *HTTP POST* request to an API. If it's was successfully, we delete all row in the **ClearDB** subflow.



9 Figure: ClearDB subflow:

INNOWWIDE report 2020 March

In this month the following tasks have been completed:

1. Creating a node-red application to read barcodes from barcode reader, which connected to the IOT Gateway via USB (2 different solutions are shown below)
2. Creating a SIEMENS TIA Portal Project containing an S7-1200 PLC and a KTP 700 Comfort panel to read and store barcode

Task 1. (First solution with intermediate file)

The LS – 3000 1D Laser Barcode Scanner is used to read barcodes. The barcode reader was used in default mode. The barcode reader closes all read data with a new line character. The barcode reader is connected like a keyboard, so its output can be read from /dev/tty1. The data sent by the barcode reader can be read using the Linux ***cat*** command. During the tests it was realized that sometimes the barcodes are received by the IOT2040 in parts, without a newline character. The first solution of the problem was using an intermediate file to store the barcode parts until the whole code was arrived. Then the whole bare code can be read from this file.

The full Node-Red application

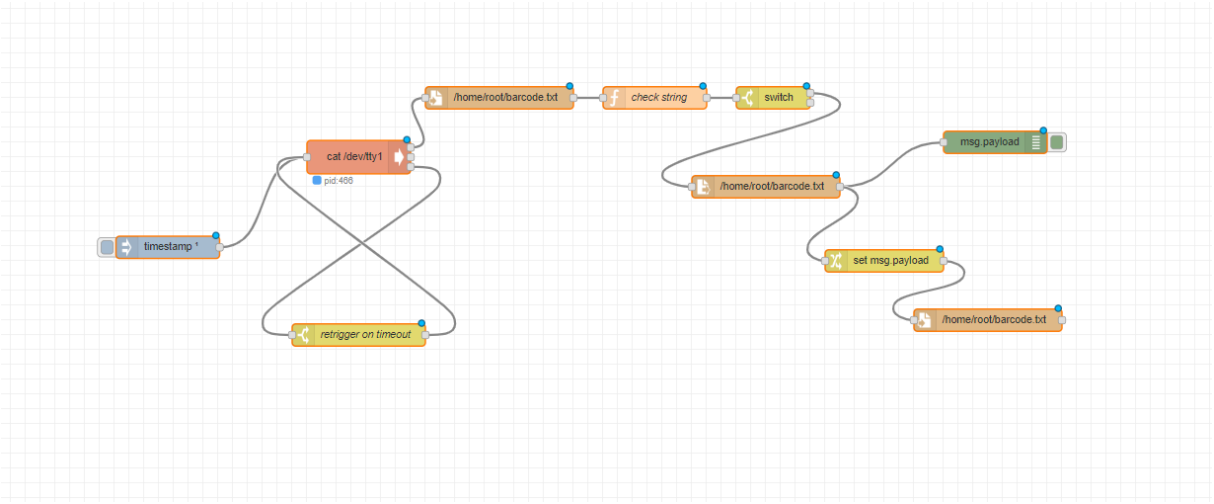


Figure 1.: The full Node-Red application

In the following section the detailed description of the overall system can be seen.

Run cat /dev/tty1 on IOT Gateway:

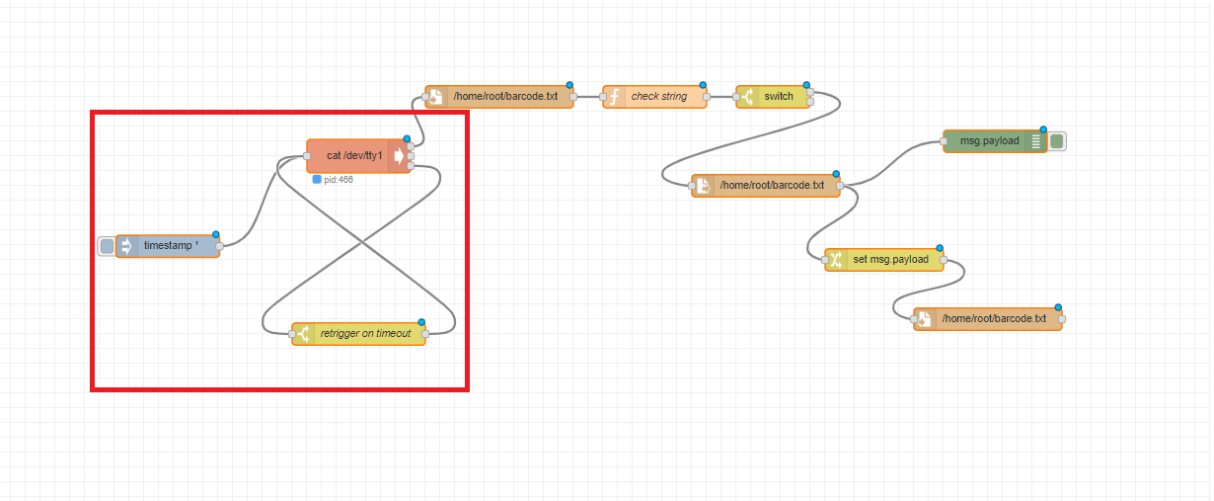


Figure 2.: The barcode reader part using the exec node

To run a Linux command on the IOT2040 gateway in Node-Red the **exec** node can be used (red node in the rectangle in Figure 2.). **Exec** node has 3 output and 1 input. The first output is the stdout. From this output we will get, the barcode parts. The second output which is the standard error output was not used in this solution, but it can be checked if it is necessary. The third output is the return code, if its false the exec block should be reconnected using the retrigger on timeout node.

The setup of nodes in the red rectangle can be seen in the following figures.

Exec block:

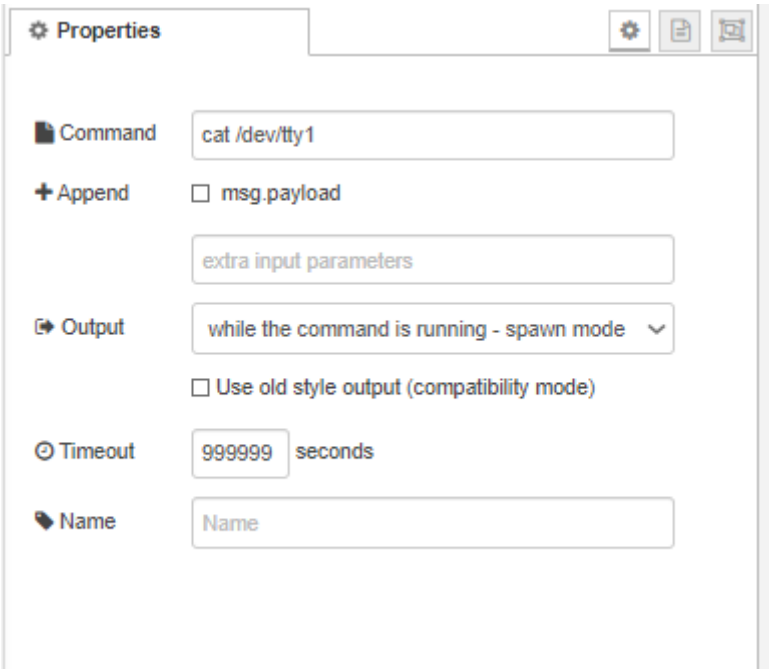


Figure 3.: The exec block setup

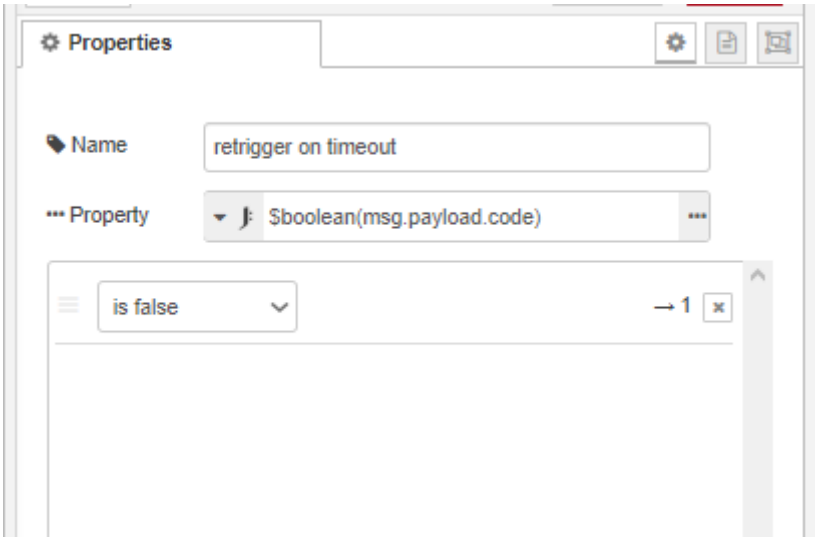


Figure 4.: Retrigger on timeout nod, which is a switch node

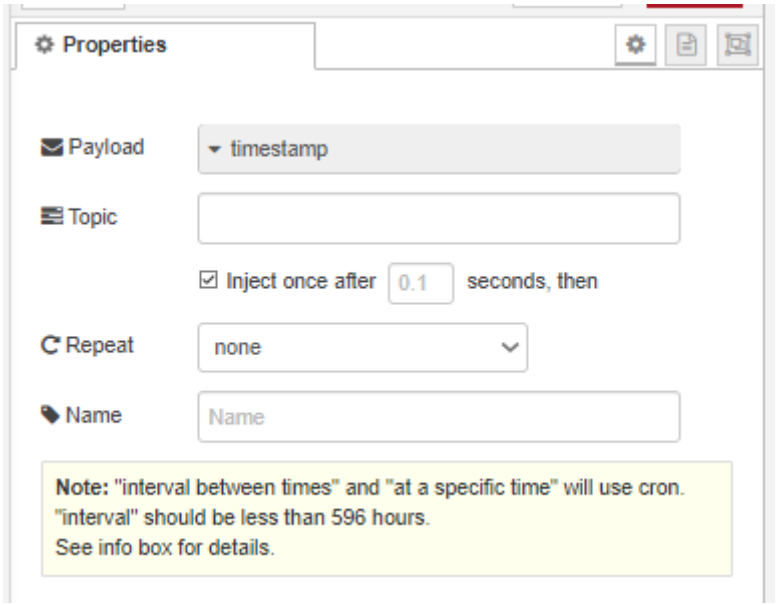


Figure 5.: Timestamp is an inject node

Write to file and check string

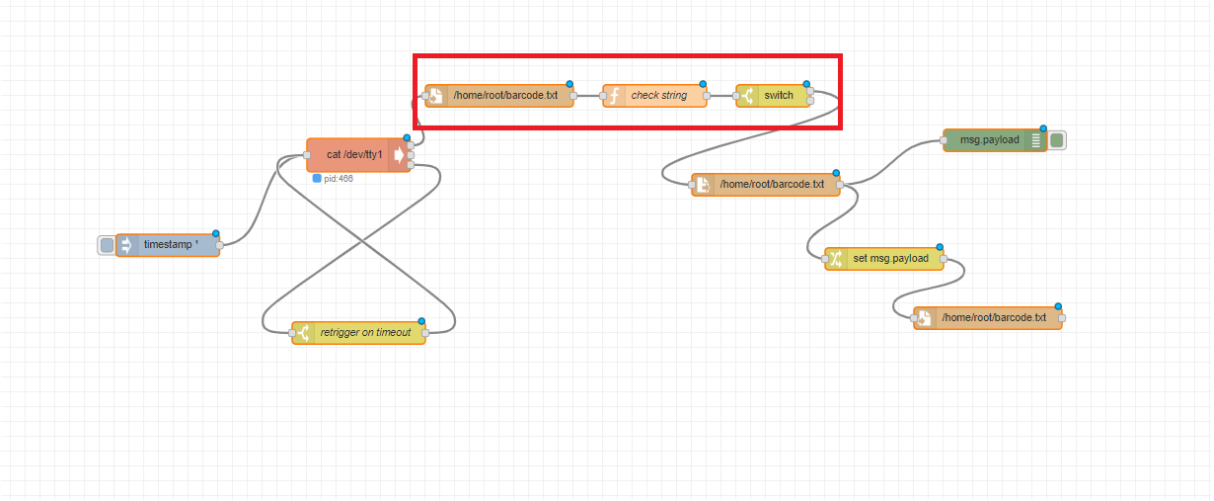


Figure 6.: Saving data to file and read back if it is a full barcode string

The data outputted by the **exec** block stdout output was written to */home/root/barcode.txt* file with append, but we don't add a new line character after it. The new line character will come from the barcode reader, if the barcode is full. The **check string** is a function node with the code can be seen in Figure 8. The file writer node return with the text which was written to the file. The **check string** node checks if the string written to the file contains a newline character. If it contains the msg.success will be true otherwise it will be false. The switch statement block in the flow check if msg.success is false or true. This can be used to decide the full barcode srting is read and the intermediate file can be deleted.

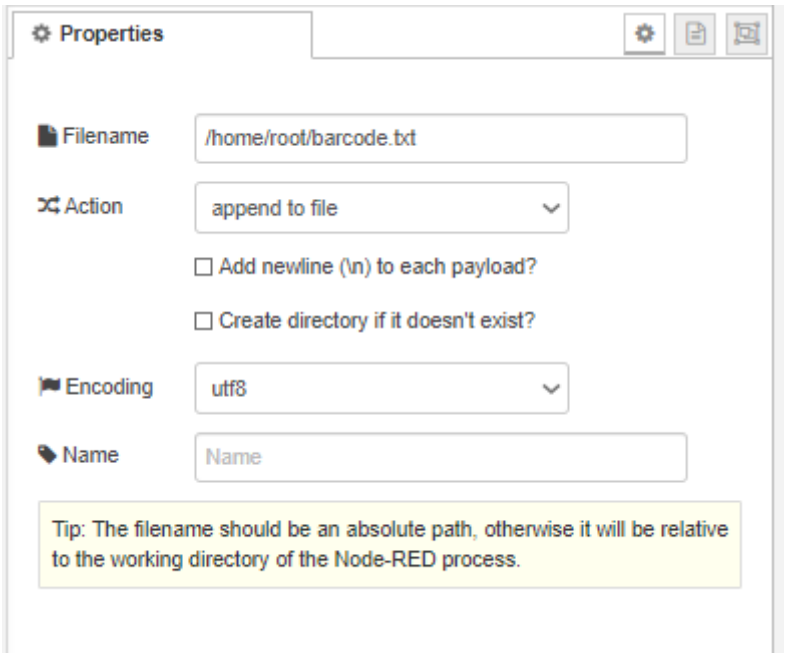


Figure 7.: File writer setup

Check string is a function node:

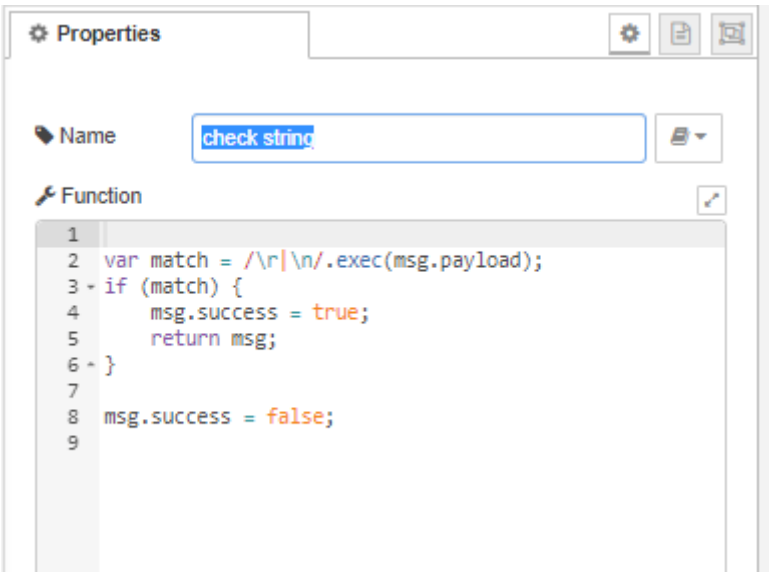


Figure 8.: Check string node is a function node with the following content

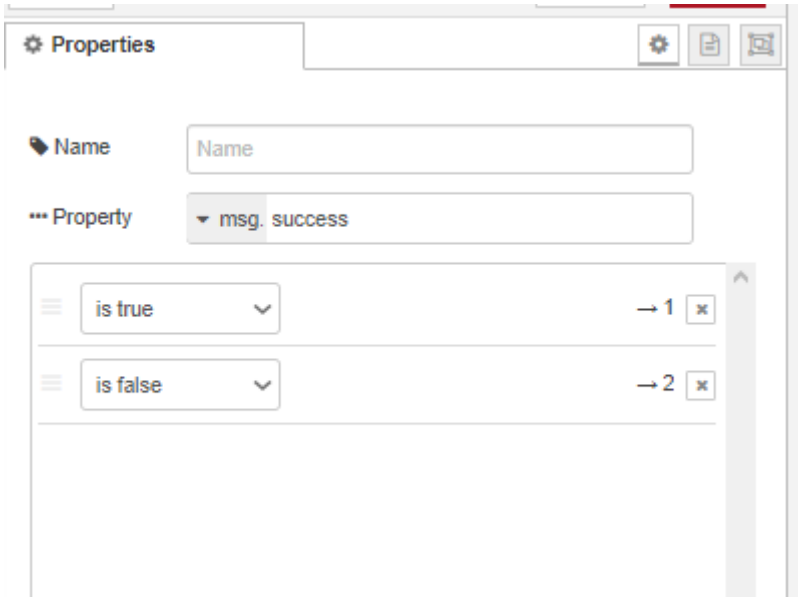


Figure 9.: Switch node

Read rom file and clear the file

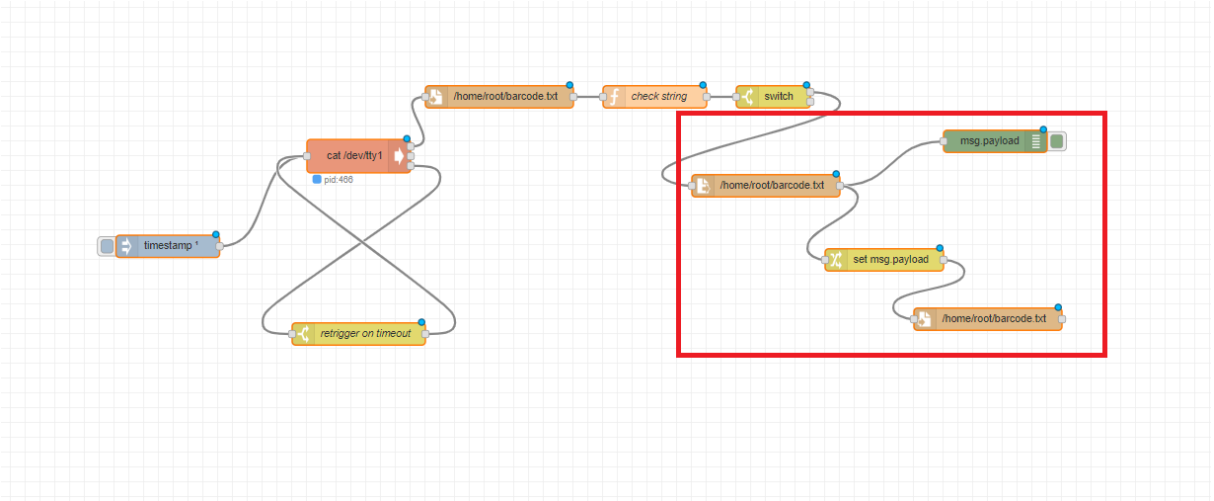


Figure 10.: Read from file and delete intermediate file

If the flow run over the switch it means the barcode in the file is full, so it can be read it out with file a **reader** node. The barcode is read out from the file and it is written to the debug console. After the barcode is read out the file can be overwritten with an empty string. To perform this msg.payload is needed to clear with a **change** node.

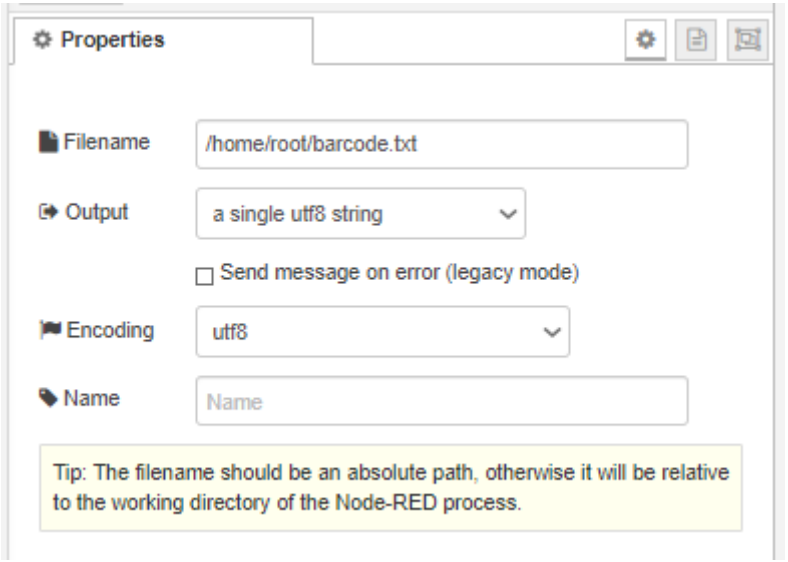


Figure 11.: File reader setup

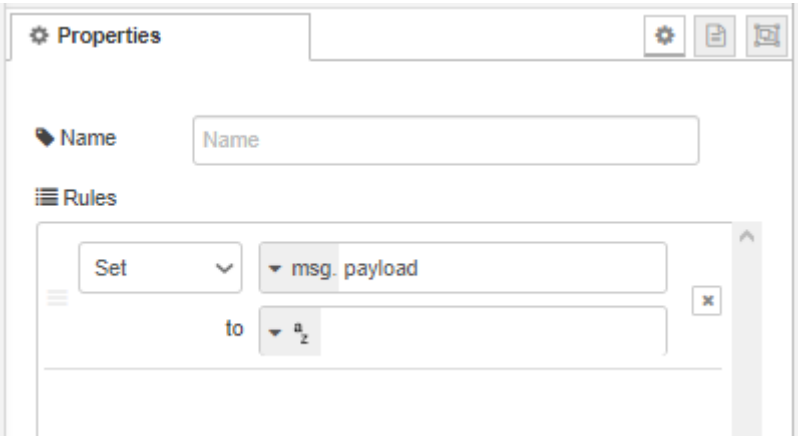


Figure 12.: Set msg.payload (change node)

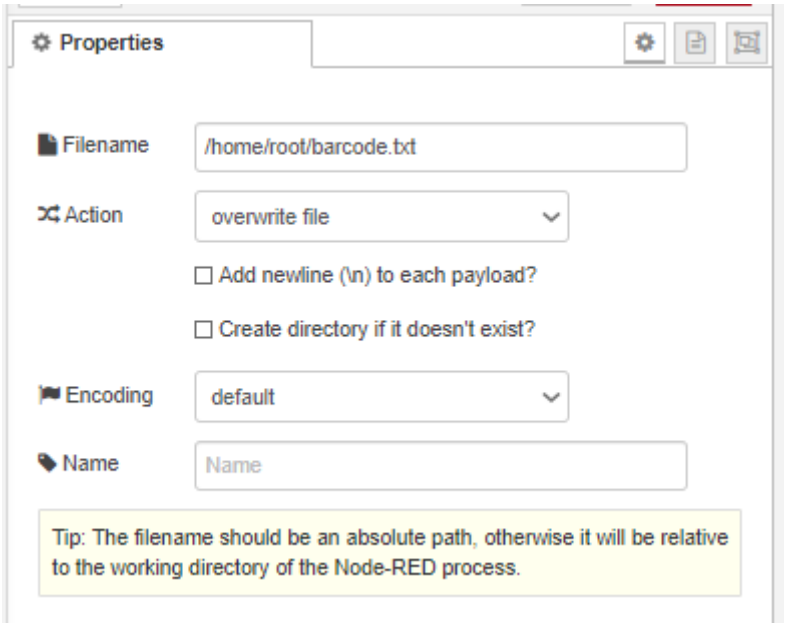


Figure 11.: File writer setup

Task 1 (Second solution)

First step: Via WinSCP or other ftp program with sftp compatibility connect to the IOT Gateway with IP of gateway and SSH Port, which is 22 on default.

Go the the folder /etc/ and open "inittab" file and add a # at the beginning of the last line.

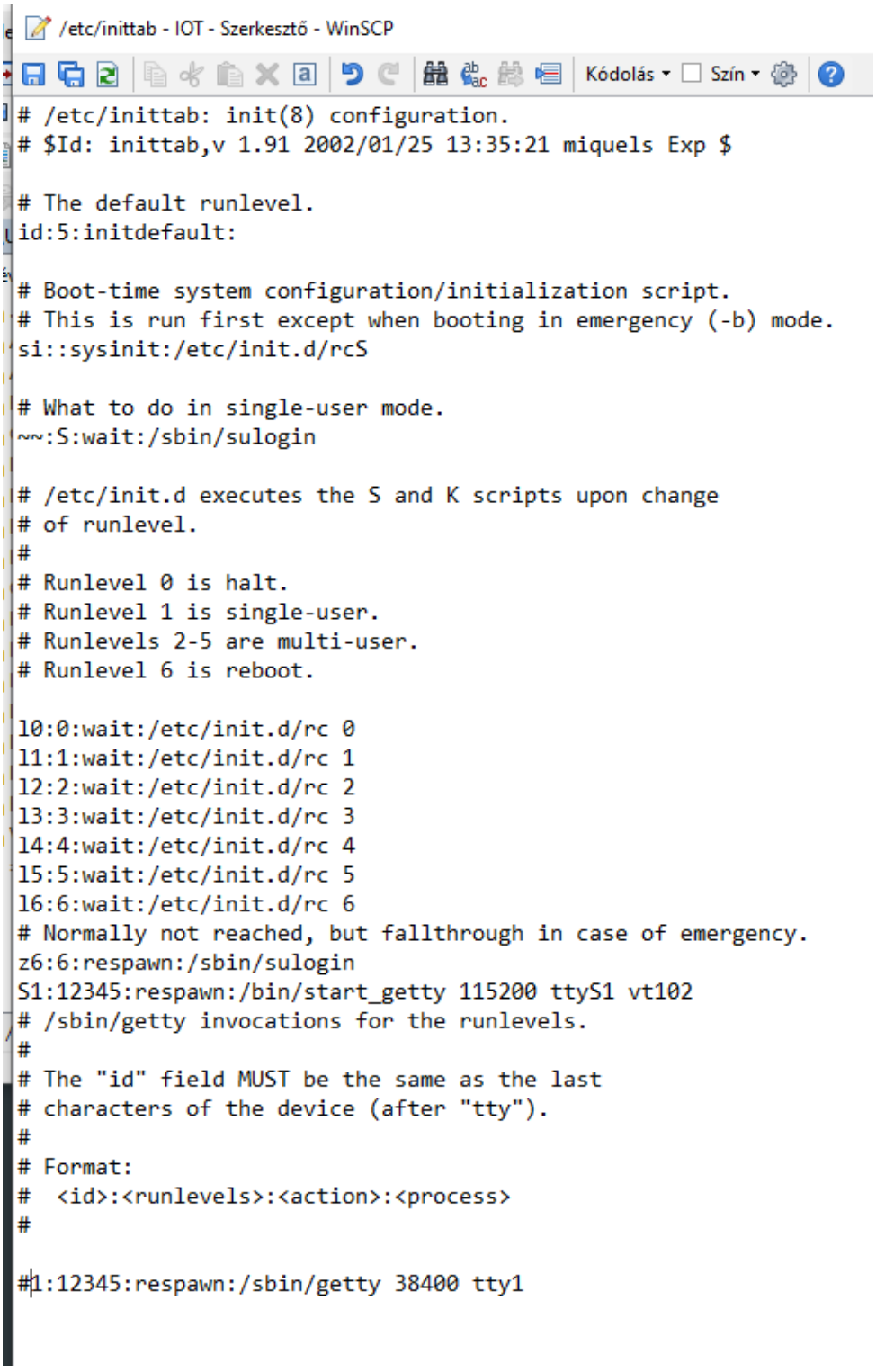


Figure 12.: /etc/inittab

Connect to IOT Gateway via PUTTY and use reboot command to reboot the IOT gateway. Wait some minutes to start Node-Red and create a new Node-Red application, which has less complexity than the first solution.

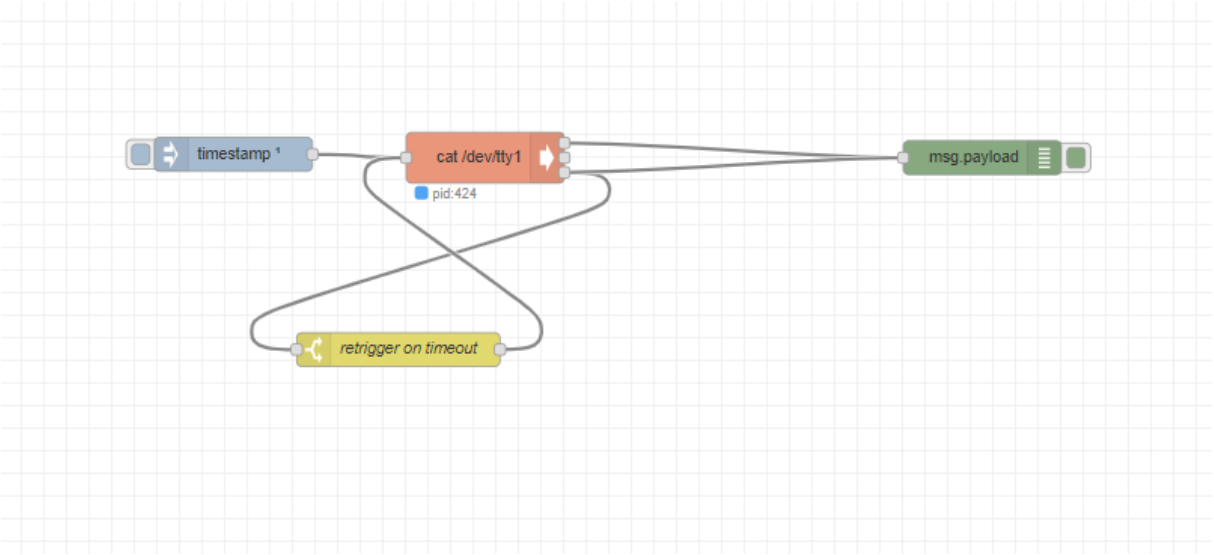


Figure 13.: The full Node-Red application

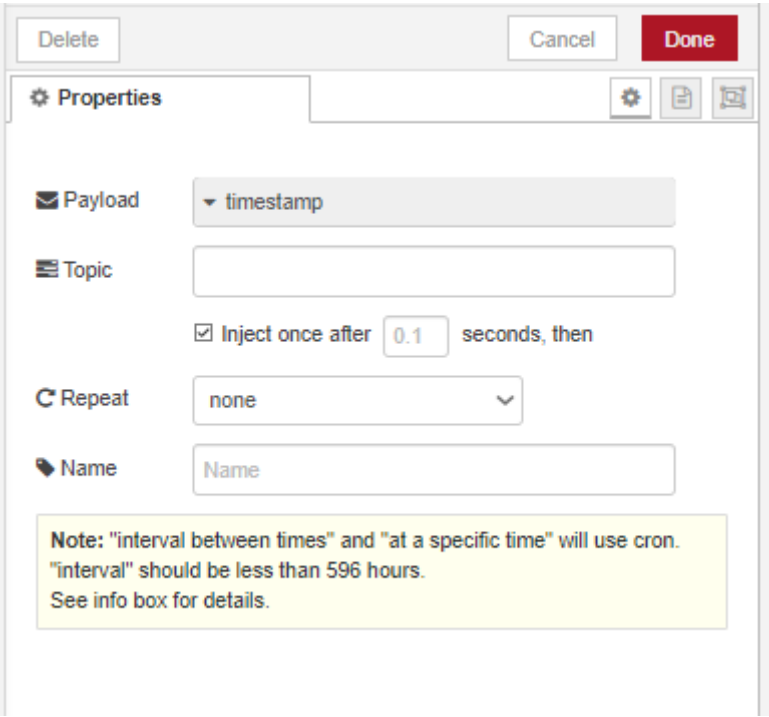


Figure 14.: Timestamp (Inject node)

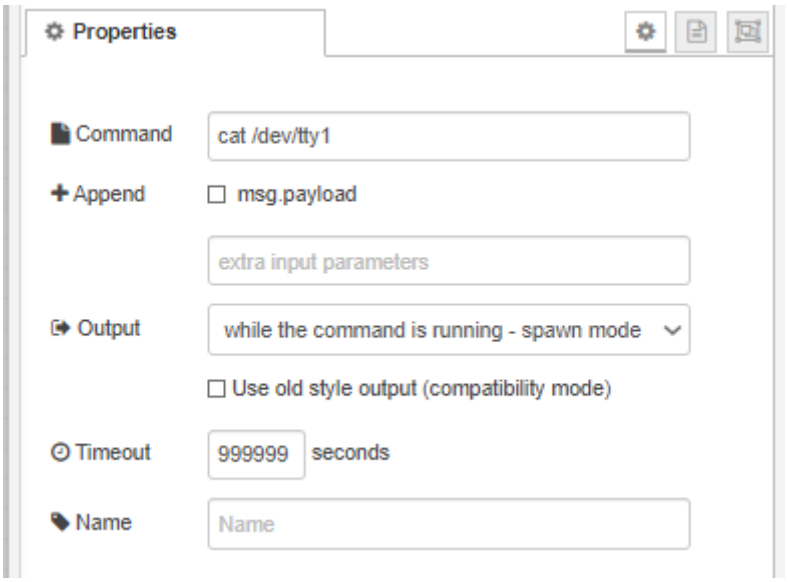


Figure 15.: cat /dev/tty1 (exec block)

Retrigger on timeout (switch node):

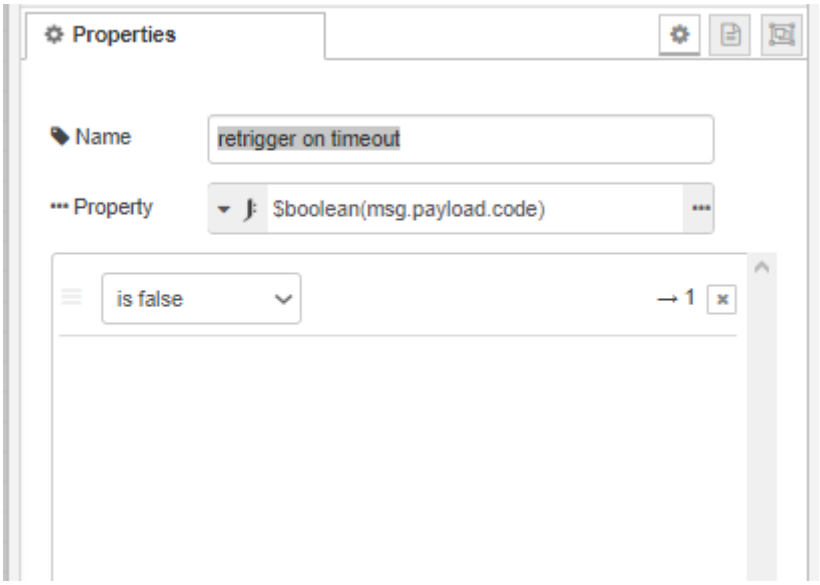


Figure 16.: Retrigger on timeout (switch node)

After the previous steps some other settings maybe needed:

- Connect your USB keyboard or barcode reader via PuTTY use the following command:
  - *ps auxww | grep tty1*
- Reboot again the gateway with reboot command.

After this setting the barcode reader will be work properly

Task 2 (PLC and HMI based solution)

To create the TIA Portal project the TIA Portal 15.1 was used. In the project setup phase the appropriate Siemens S7-1200 PLC and the appropriate KTP HMI panel (the panel must contain an USB port) should be added to the project as can be seen on Figure 17.

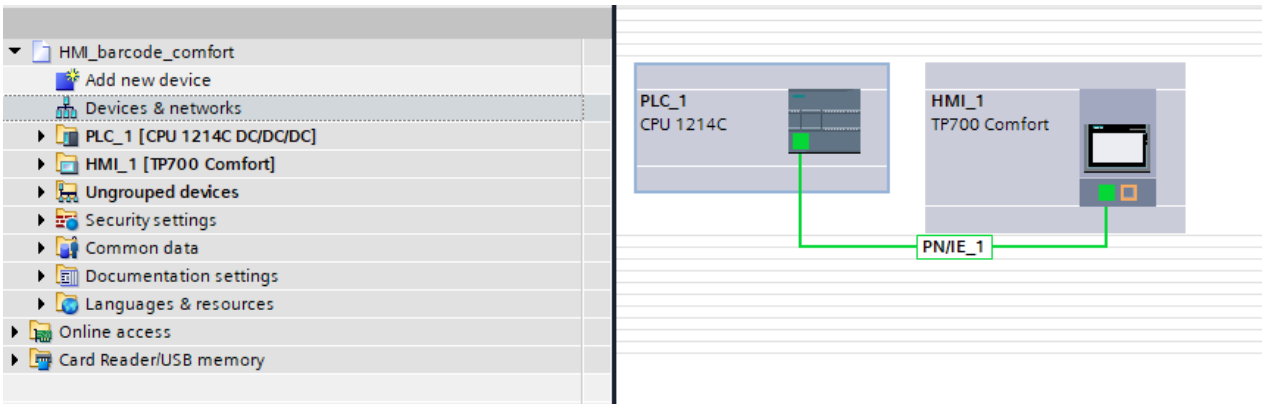


Figure 17.: The PLC and the HMI connected through PN/IE_1 connection

After setting up and connect the devices a new DB should be added to the PLC. Let the name of the DB be Bar_code. Add a new variable to the DB with String data type and named it to Bar_code or anything the user likes.

The DB and its data filed can be seen in Figure 18.

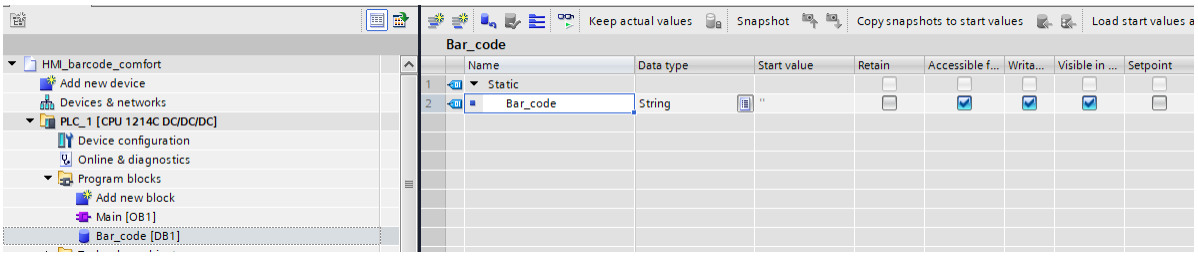


Figure 18.: The DB Bar_code created on the PLC

After creating a DB an I/O field should be added to the HMI root screen. This I/O field should be set up as input filed with String format as can be seen in Figure 19.

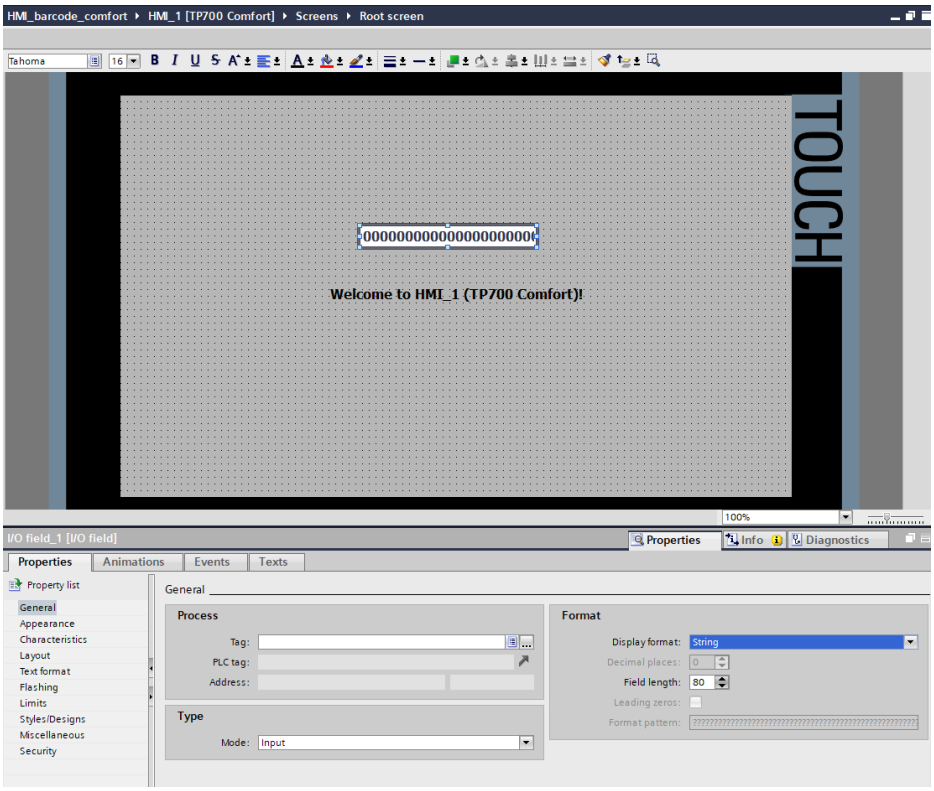


Figure 19.: The DB Bar_code created on the PLC

This I/O filed is connect to the Bar_code variable created in the Bar_code DB in the PLC as can be seen on Figure 20.

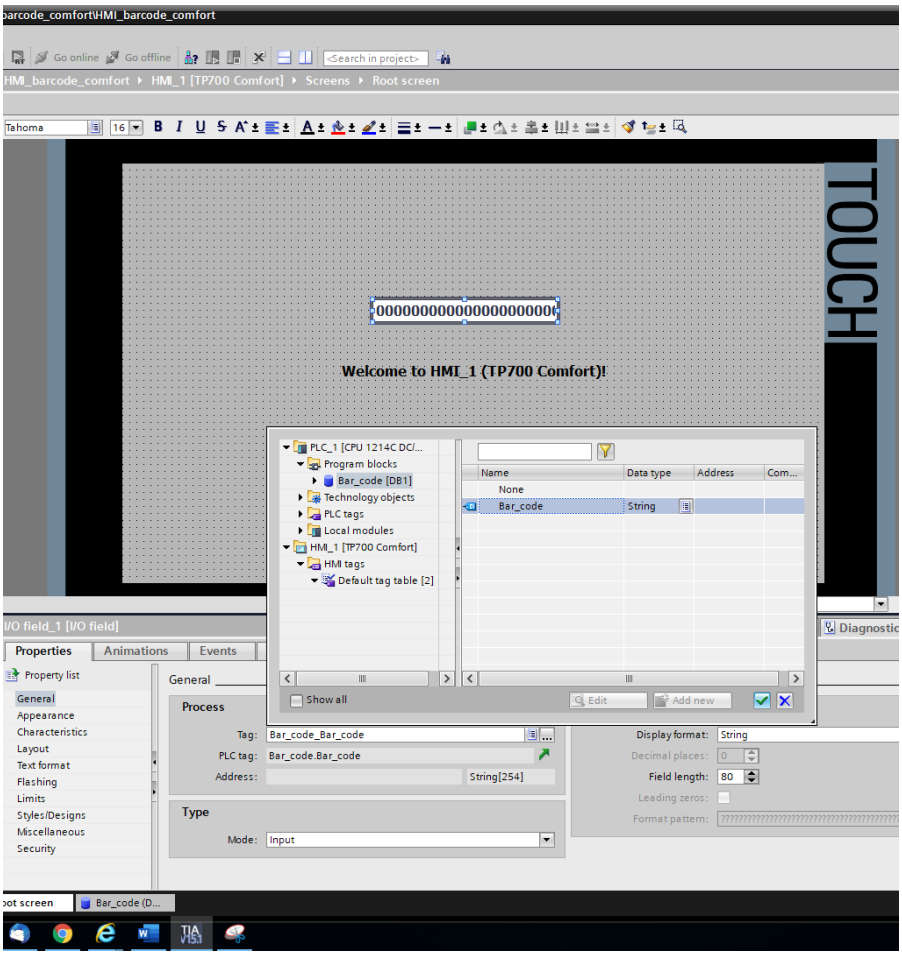


Figure 20.: The connection of the Bar_code variable and the I/O filed

Finally, the barcode reader should be connected to the HMI panel, and download the program of the PLC and the HMI screen.

As can be seen on Figure 21 after reading a bar code the bar code string is transferred to the Bar_code PLC tag.

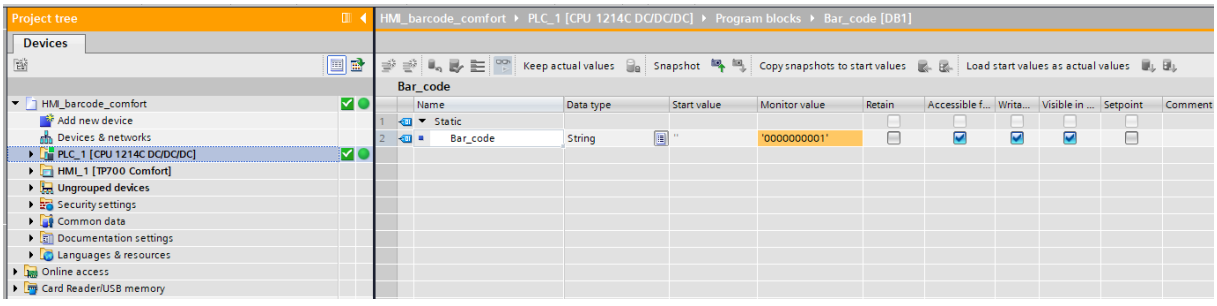


Figure 21.: The barcode string in the PLC read by the LS3000 barcode reader

INNOWWIDE report 2020 April

In this month, the following tasks have been completed:

1. Creating a Node-Red application to read barcodes and generate material codes using a csv file. The material codes will be sent to the main PLC via S7 connection.

Task 1.

First let us see the csv (codes.csv) file which contains the barcodes and the material codes. The file contains two columns named Barcode and Material Code as it can be seen on the following figure:

	A	B
1	Barcode	Material Code
2	0000000001	001
3	0000000002	002
4	0000000003	003
5	0000000004	004
6	0000000005	005
7	0000000006	006
8	0000000007	007
9	0000000008	008
10	0000000009	009
11	0000000010	010
12	0000000011	011
13	0000000012	012
14	0000000013	013
15	0000000014	014
16	0000000015	015
17	0000000016	016
18	0000000017	017
19	0000000018	018
20	0000000019	019
21	0000000020	020
22	0000000021	021
23	0000000022	022
24	0000000023	023
25	0000000024	024
26	0000000025	025
27	0000000026	026
28	0000000027	027
29	0000000028	028
30	0000000029	029
31	0000000030	030
32	0000000031	031
33	0000000032	032

Figure 1.: The content of the codes.csv

In the Barcodes column only example codes can be found. These codes can be arbitrarily modified according to the current barcodes. Please do not modify the Material Code column because these codes are used by the master PLC to identify the actual material filled in the Mixer unit.

After modifying the Barcode column the codes.csv file must be copied to /home/root/ folder on the IOT Gateway using the WinSCP application.

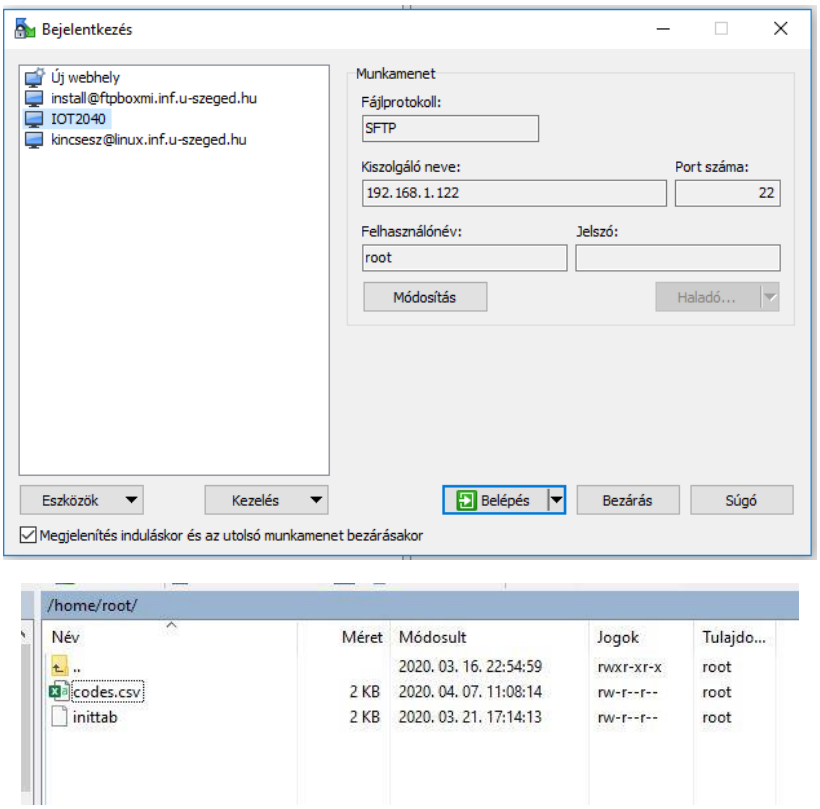


Figure 2.: The WinSCP application

After coping the codes-csv to the appropriate folder two methods can be used to reread the modified file:

1. Restart the IOT Gateway, because during the boot phase of the Node-Red application the codes.csv will be re-read automatically (see in Figure 5).
2. Push the READ CODES.CSV on the Node-Read UI (see in Figure 4)

The full Node-Red application can be seen on the following figure.

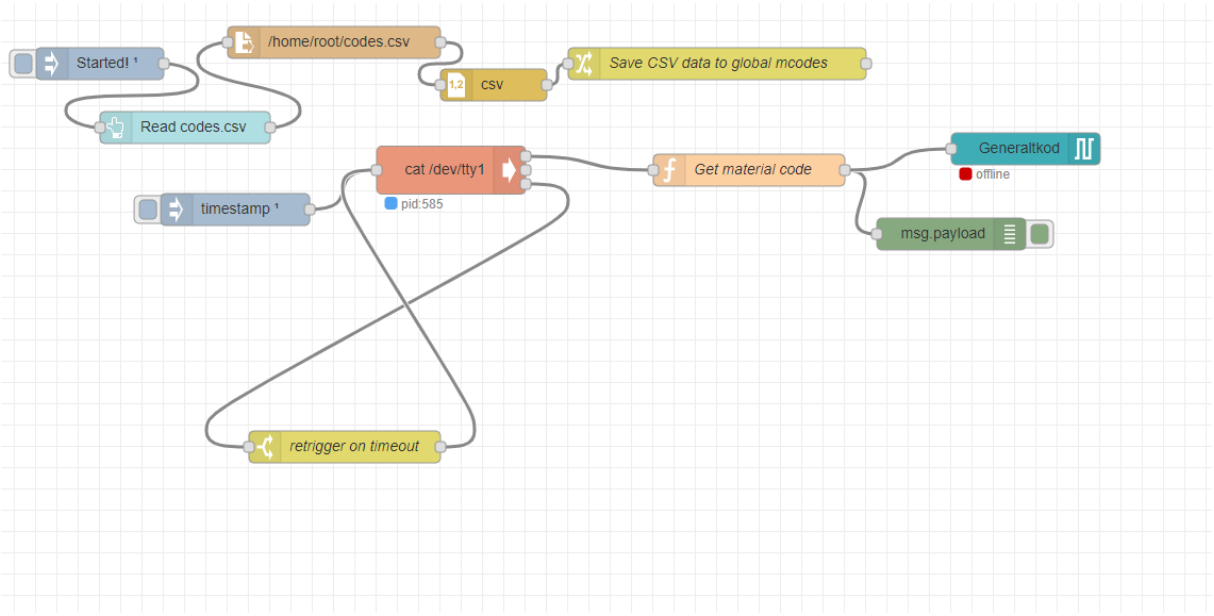


Figure 3.: The full Node-Red application

The Node-Red UI can be seen in the following Figure. This can be used to re-read the codes.csv after the appropriate modifications mentioned above.

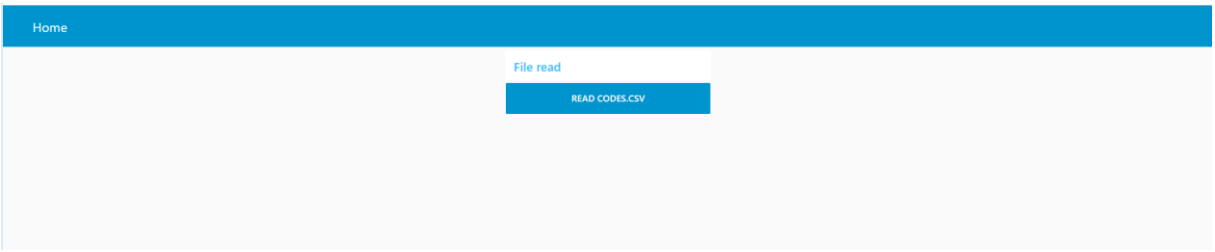


Figure 4.: The Node-Red UI

No lets see how it works. During the start-up of the Node-Red application the codes.csv file is read automatically from /home/root. The location of the file and the name must be the same. The codes.csv is also read if someone click on the READ CODES.CSV after the start-up on the Node-Red UI as can be seen on the following figure:

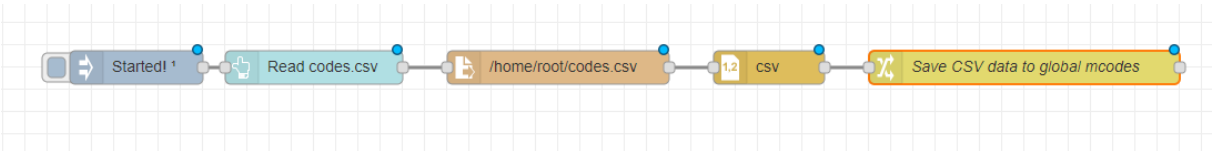


Figure 5.: The start-up code

The first node is the start-up injection with injection node as can be seen on the following figure:

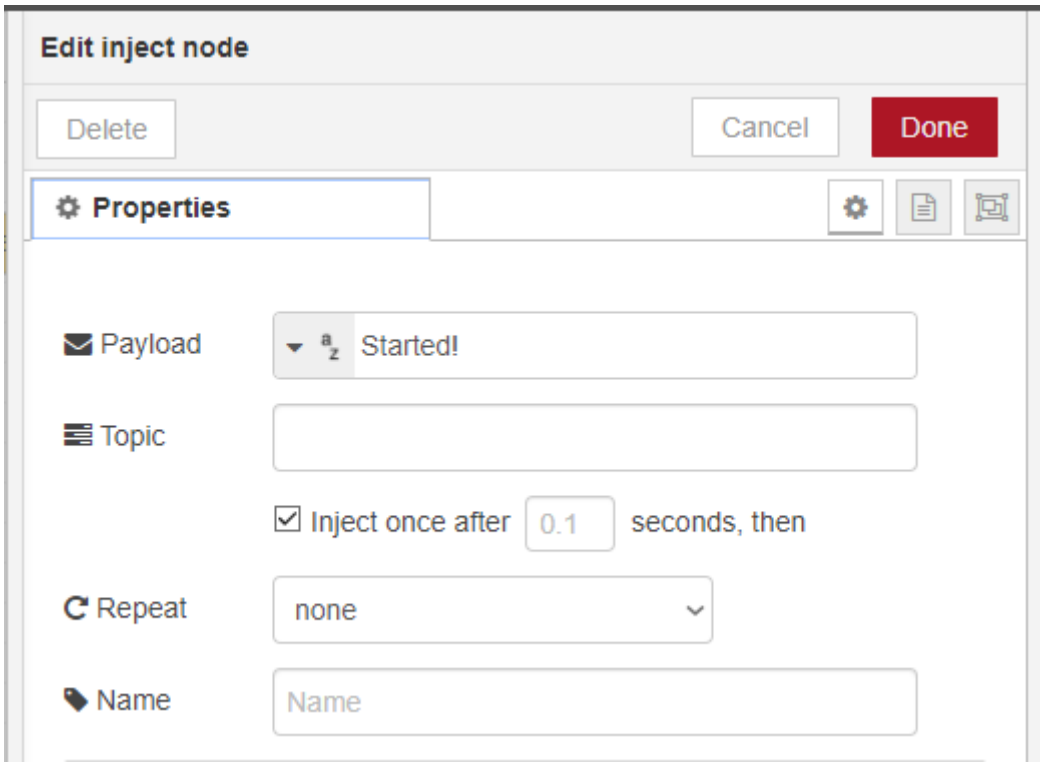


Figure 6.: The inject nod and its configuration

The next node is a button node named Read codes.csv. The configuration can be seen on the following figure:

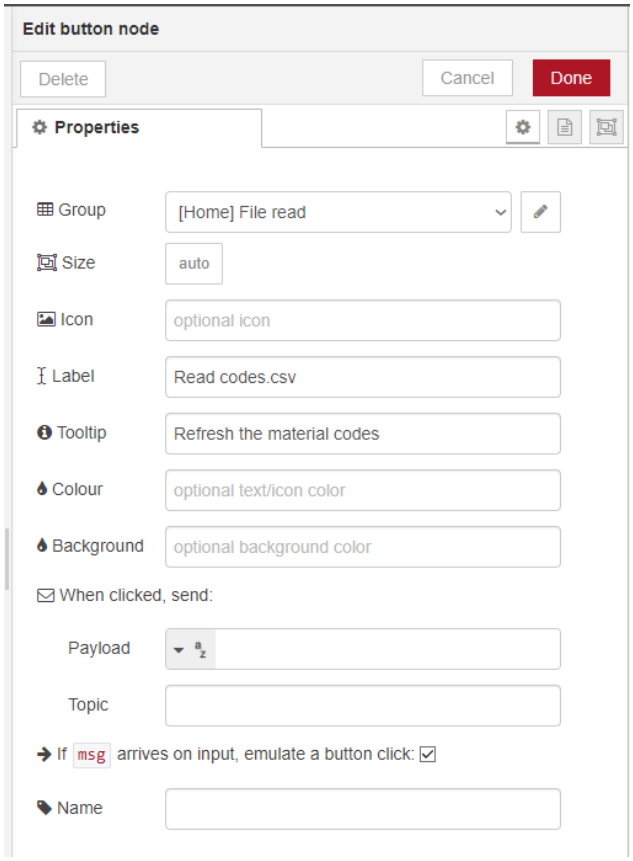


Figure 7.: The Read codes.csv node

The third node is an open file node to open the codes.csv:

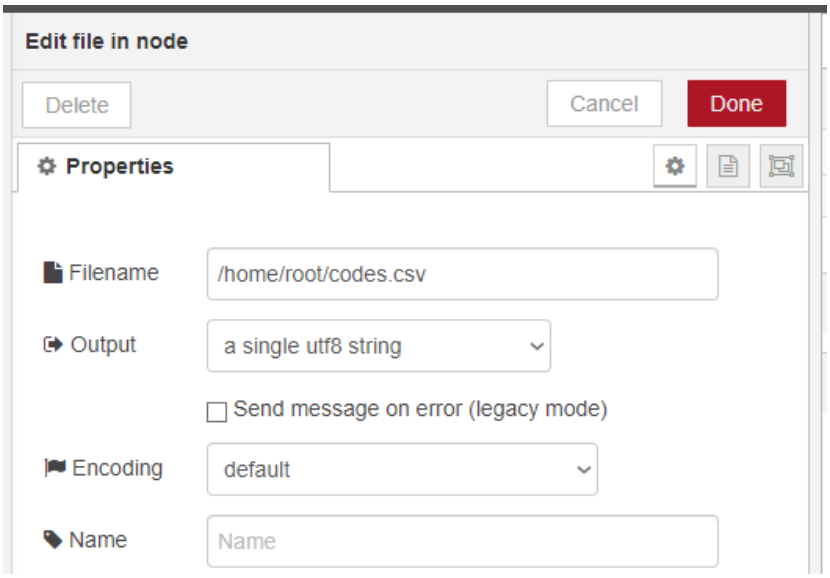


Figure 8.: The open file node

The 4. node is the Read from csv using the csv node:

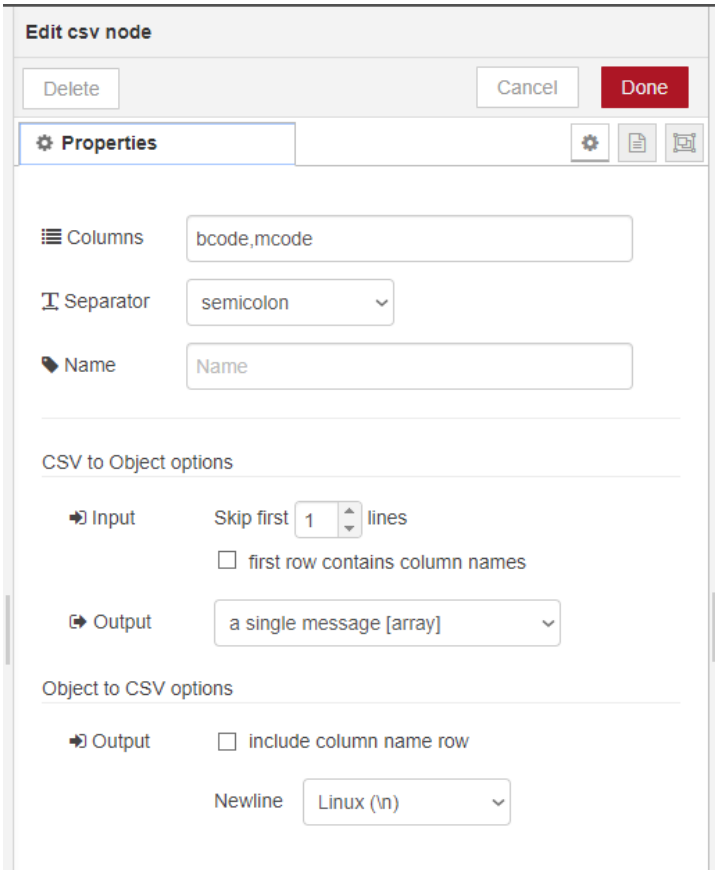


Figure 9.: The configuration of the open csv node

The 5. node is saving the csv data to a global variable using the change node:

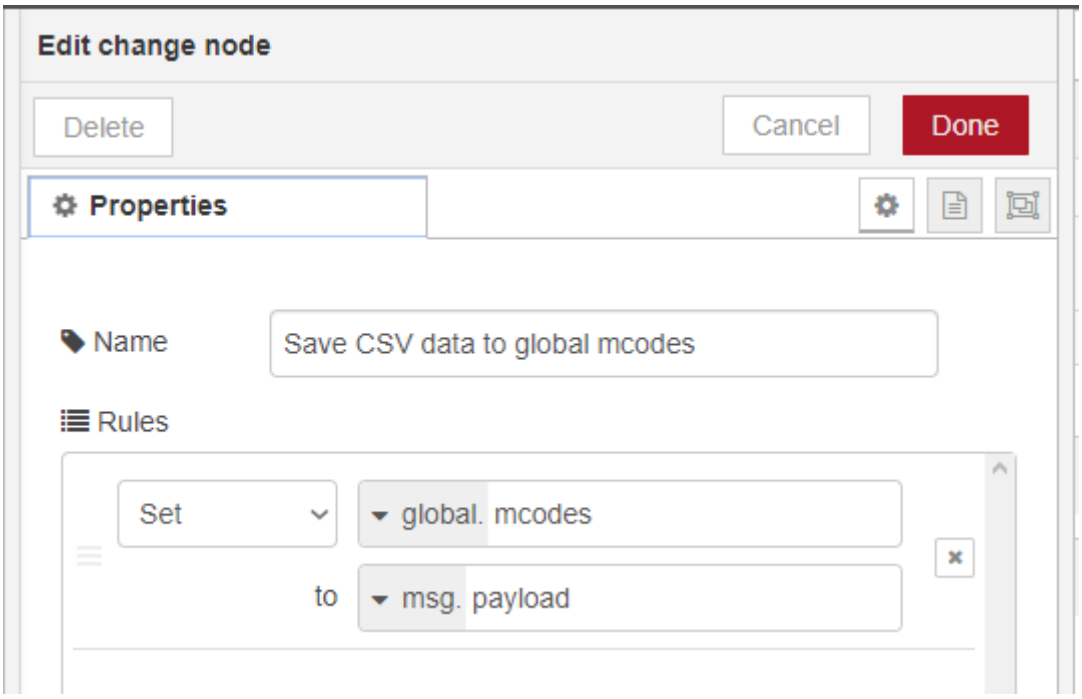


Figure 10.: The change node configuration

After reading and saving the content of the codes.csv let us see how the material codes are selected. In the following figure this flow can be seen.

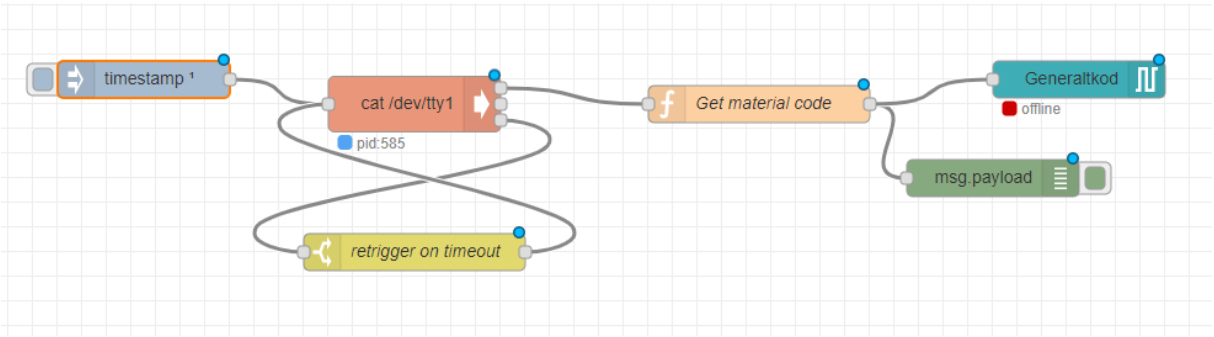
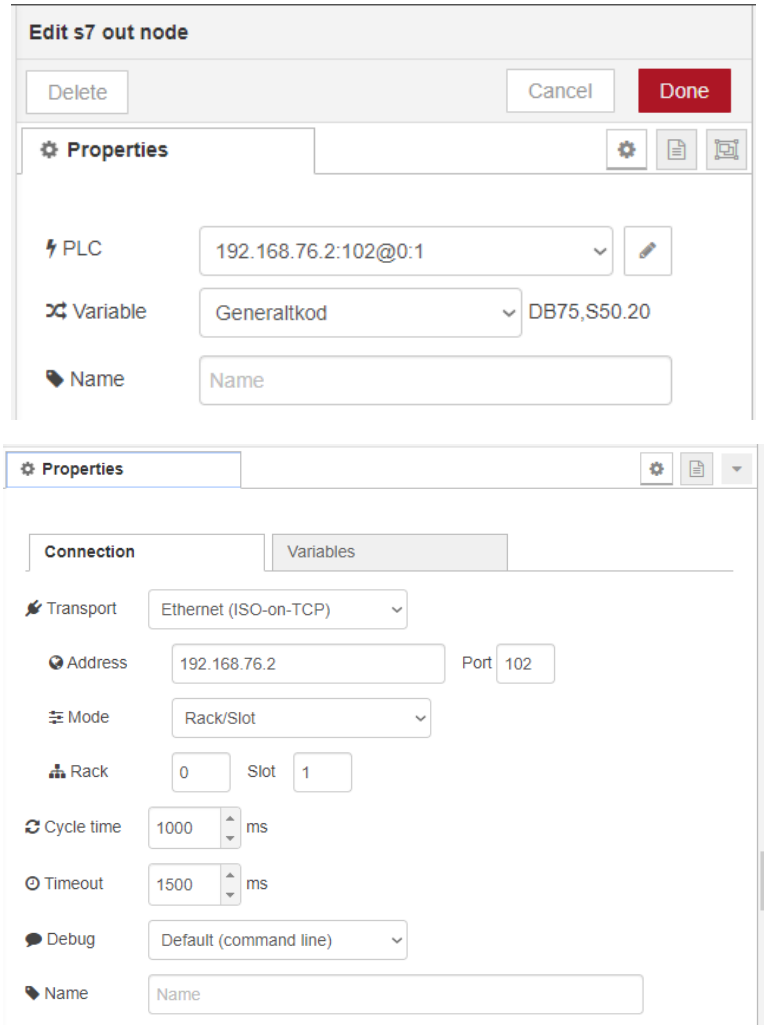


Figure 11.: The material code selection

The operation of the baracode reader part of the flow was describe in the previous report. After reading a barcode the function node named Get material code is started. The content and the configuration of this node can be seen on the following figure:

Figure 12.: The Get material code node

After selecting the material code it will be sent to the master PLC using S7 node:



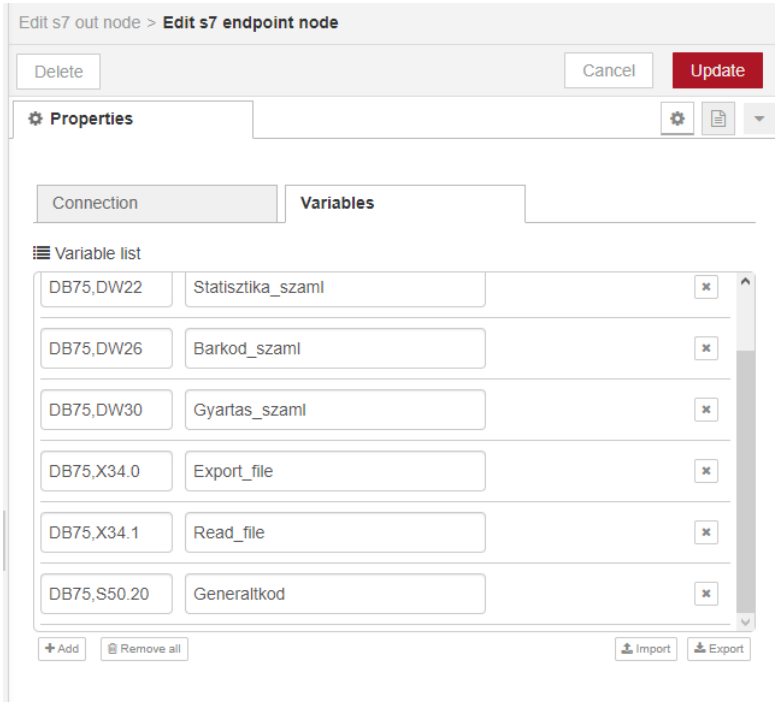


Figure 13.: S7 node and its configuration

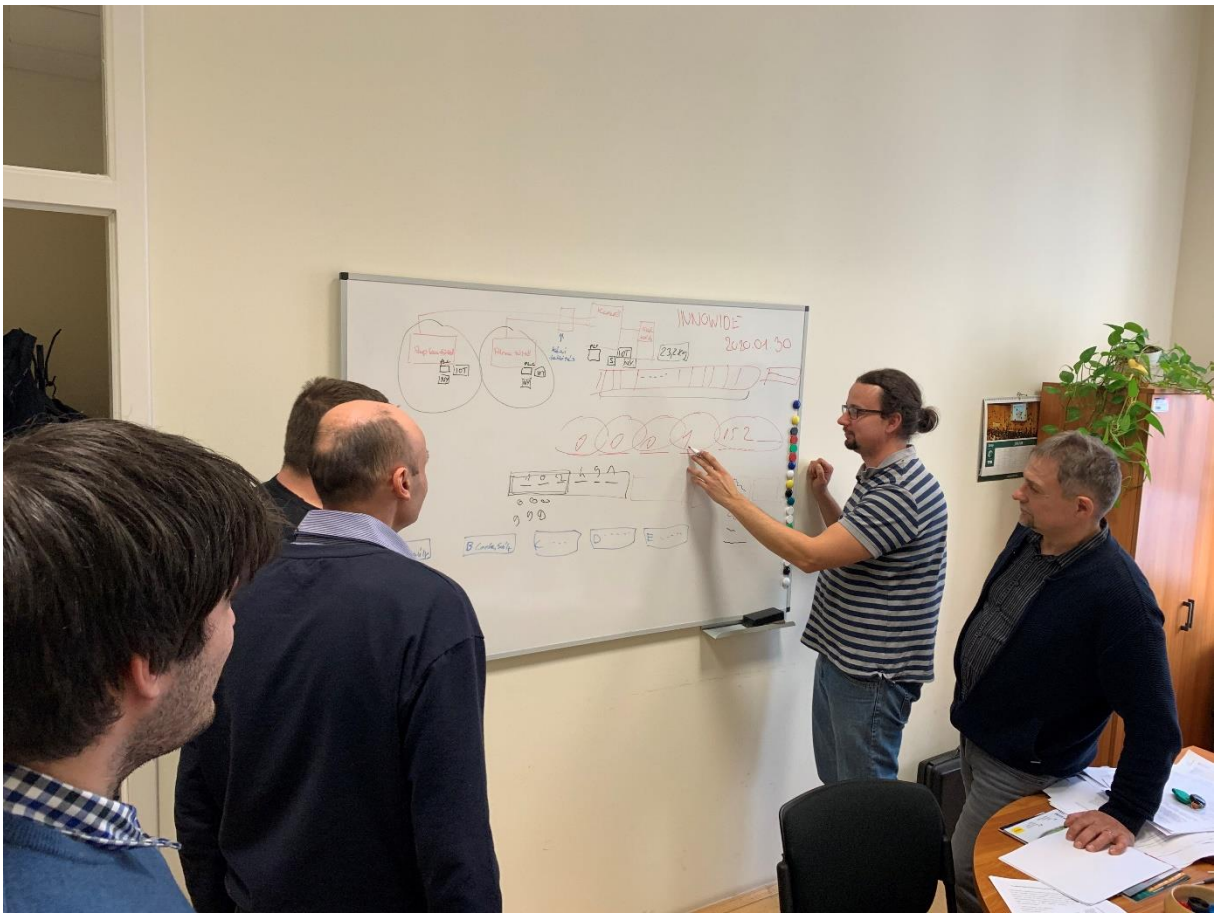
Meeting 2020 January 13.

Participants:

1. Balázs Fruzsa CSF Electric
2. András Császár CSF Electric
3. Szilveszter Pletl PhD. University of Szeged
4. Zoltán Kincses PhD. University of Szeged
5. Sándor Savanya University of Szeged


INNOWIDE
2020.01.30

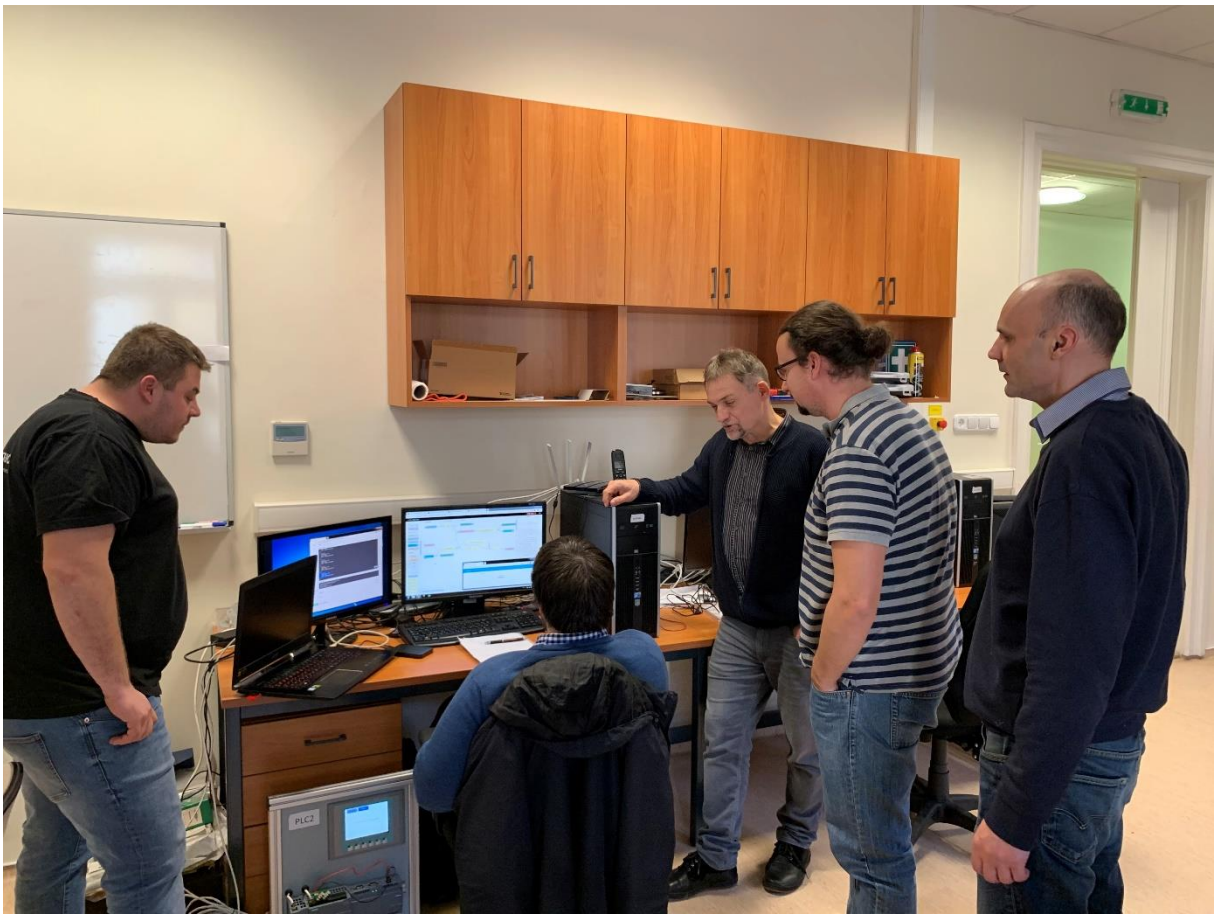

INNOWIDE
2020.01.30

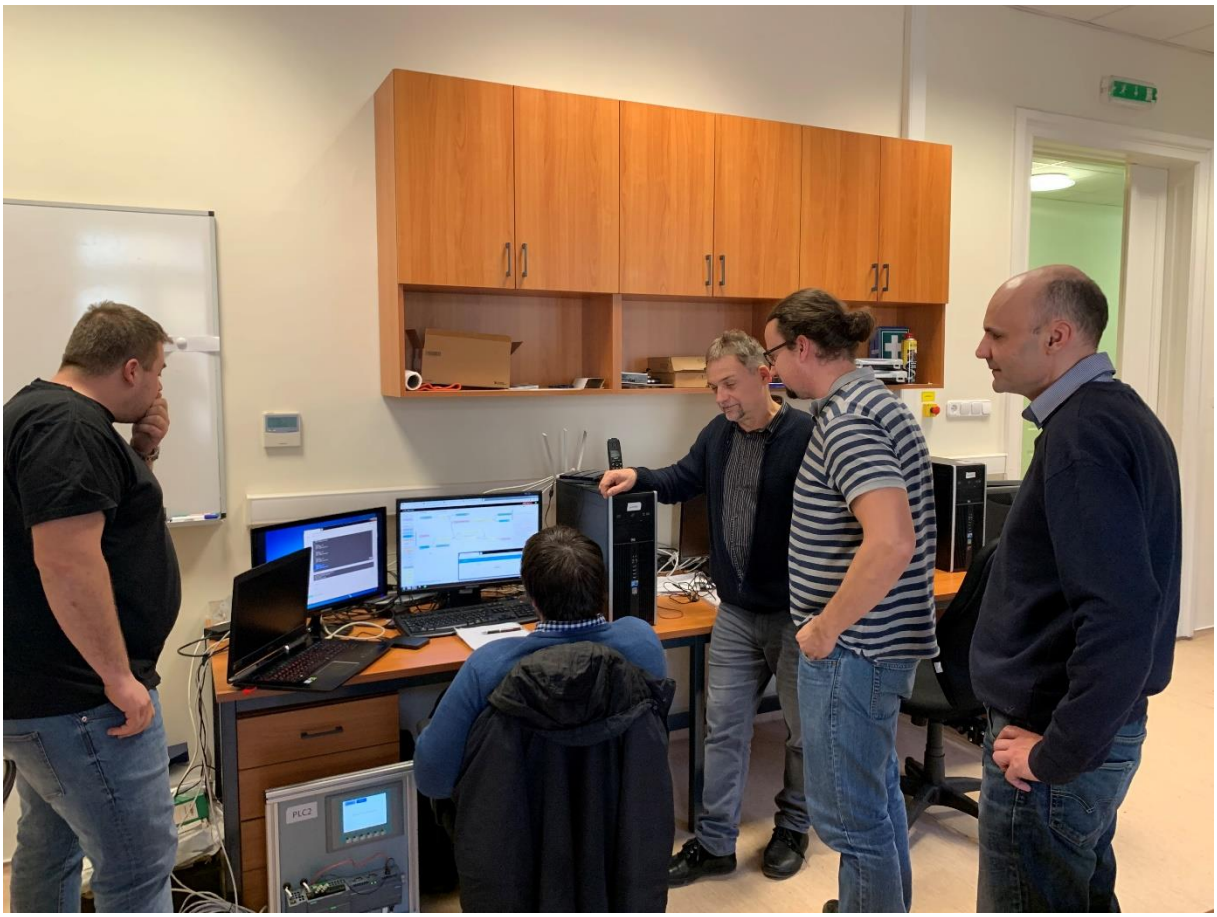

INNOWIDE

Meeting 2020 February 24.

Participants:

1. Balázs Fruzsa CSF Electric
2. András Császár CSF Electric
3. Szilveszter Pletl PhD. University of Szeged
4. Zoltán Kincses PhD. University of Szeged
5. Sándor Savanya University of Szeged




PLC2


CSF ELECTRIC
Feather Processing Machines