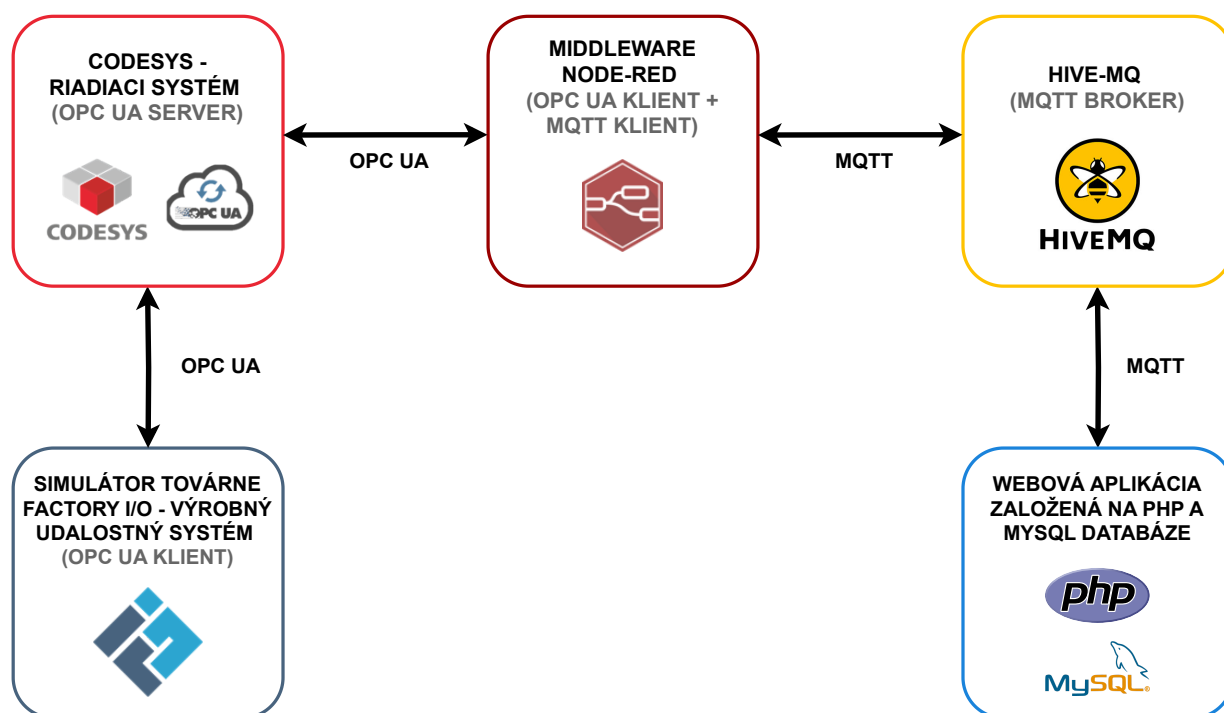


### 4.3 Tretia prípadová štúdia: Prepojenie PLC s webovou aplikáciou a databázou

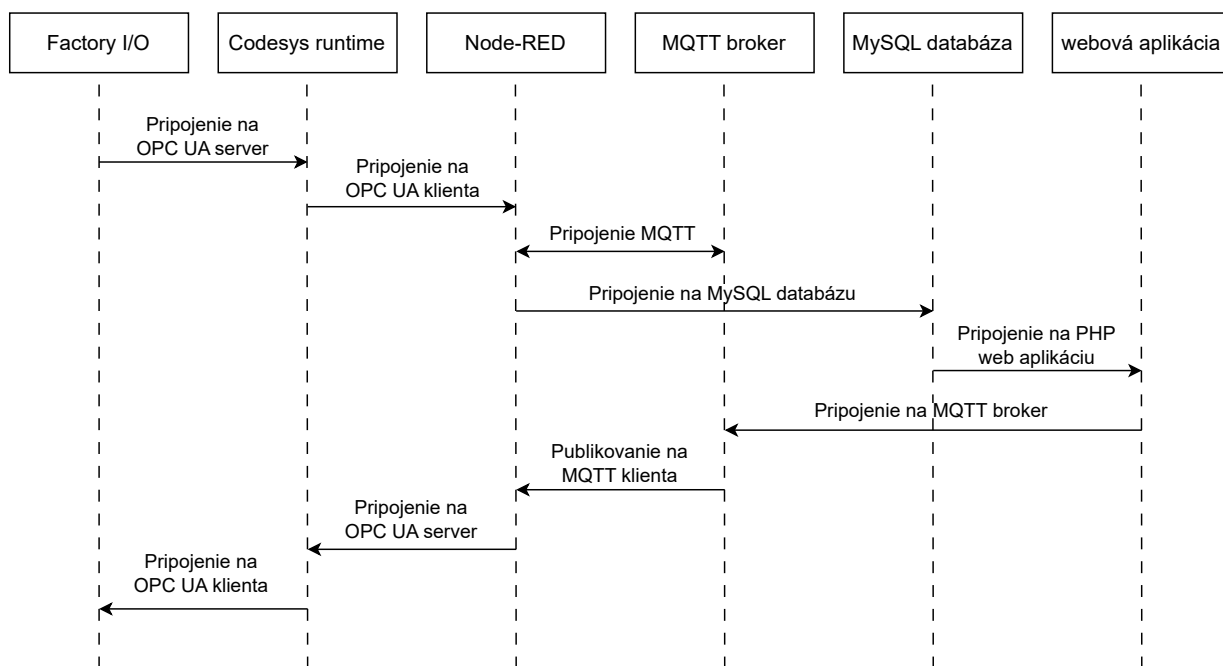
Aj cieľ tretej prípadovej štúdie bol z oblasti digitalizácie diskretných udalostných systémov a takisto bola vytýčená úloha návrhu riadenia pre virtuálny model výrobného systému realizovaného vo Factory I/O. Model je riadený prostredníctvom PLC engine Codesys, pričom monitorovanie udalostného systému je zabezpečené webovou aplikáciou s príslušnou databázou. Pomocou webovej aplikácie je možné do systému aj zasahovať (napr. núdzové zastavenie). Táto prípadová štúdia je opísaná s využitím diplomovej práce [98].



Obrázok č. 95: Schéma komunikácie jednotlivých častí prípadovej štúdie

V prípadovej štúdii bol prepojený Codesys a simulátor Factory I/O pomocou moderného komunikačného protokolu OPC UA. Codesys, ktorý zabezpečoval riadenie virtuálnej továrne, slúžil na naprogramovanie riadenia, ako PLC engine a tiež ako OPC UA server. Vo Factory I/O bol navrhnutý model diskretného udalostného systému. Konkrétne išlo o linku, ktorá triedila prichádzajúci materiál na základe nastavených kritérií a následne ho spracoval v obrábacom centre. Factory I/O figuroval ako OPC UA klient. Dáta z riadenia boli posielané do prostredníka Node-RED, ktorý tiež figuroval ako OPC UA klient. Node-RED zachytával dáta z OPC UA servera, spracoval ich a zaslal s konkrétnou témou na MQTT broker pomocou komunikačného protokolu MQTT. Na MQTT broker bola zaslaná požiadavka na prijímanie dát s našou témou. Prijaté dáta boli následne uložené do databázy, odkiaľ sú dopytované pomocou SQL dopytov v pro-

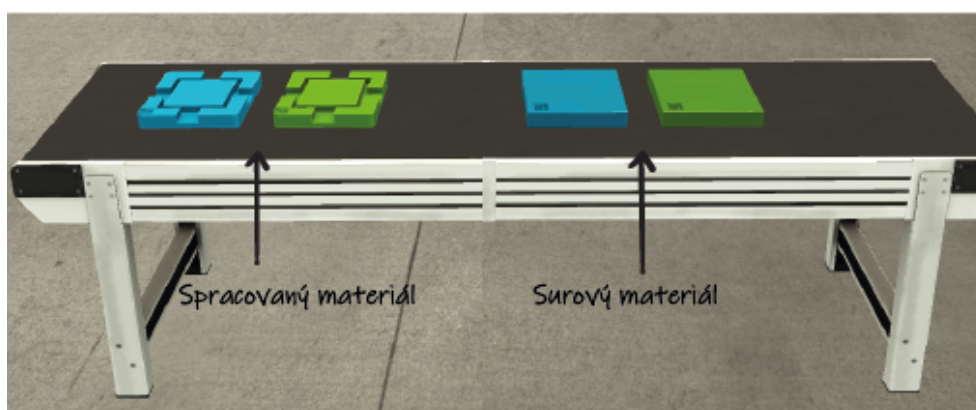
gramovacom jazyku PHP a zobrazené v PHP webovej aplikácii.



Obrázok č. 96: Priebeh komunikácie jednotlivých častí prípadovej štúdie

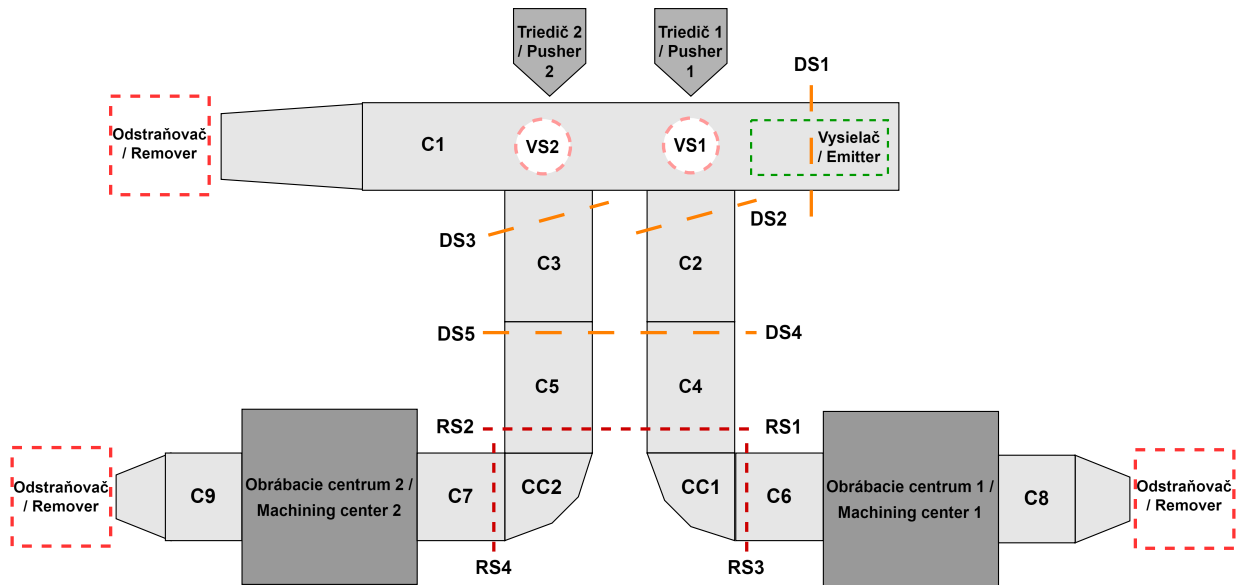
### 4.3.1 Špecifikácia a správanie diskretného udalostného systému

V prvom kroku bolo potrebné vo Factory I/O navrhnuť vhodný virtuálny model diskretného udalostného systému vo forme triediacej linky. Úlohou navrhnutého systému je vygenerovať dva rôzne materiály (teda polovýrobky), tieto roztriediť a poslať do správnej vetvy systému. Následne správne rozdelené materiály dopraviť k obrábaciemu centru, kde je materiál opracovaný a poslaný ďalej na expedíciu.



Obrázok č. 97: Ukážka materiálov [98]

Schému diskretného udalostného systému môžeme vidieť na obrázku č. 98.



Obrázok č. 98: Schéma udalostného systému

Legenda značiek z obrázka č. 98:

- C1-C9 -> dopravníky;
- CC1, CC2 -> zakrivené dopravníky;
- VS1, VS2 -> vizuálne senzory;
- DS1-DS5 -> difúzne senzory;
- RS1-RS3 -> retroreflektívne senzory;
- Pusher1, Pusher2 -> mechanické triediče;
- Machining Centre1, Machining Centre2 -> obrábacie centrá.

#### 4.3.2 Komponenty použité na tvorbu modelu linky

Najprv je potrebné uviesť charakteristiku prvkov Factory I/O, ktoré sú v tejto prípadovej štúdii nové a neboli využité v prvej a druhej prípadovej štúdii.

1. **Triedič (angl. Pusher)** — Pneumatický triedič v tvare piestu vybavený dvoma jazýčkovými snímačmi, ktoré indikujú prednú a zadnú hranicu. Obsahuje aj serwoventil, ktorý je možné využiť na nastavenie polohy piestu. Ovládanie sa môže vykonávať pomocou digitálnych alebo analógových hodnôt podľa zvolenej konfigurácie.



Obrázok č. 99: Pneumatický triedič [98]

2. **Difúzny senzor (angl. Diffuse sensor)** — Difúzny fotoelektrický senzor, ktorý dokáže rozpoznať akýkoľvek pevný objekt.



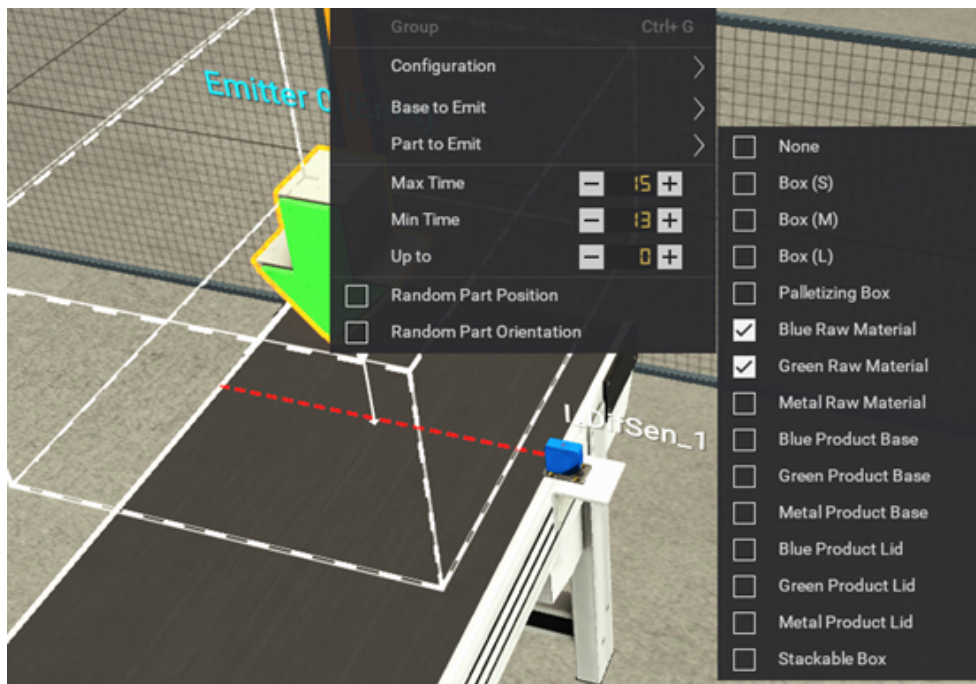
Obrázok č. 100: Difúzny senzor [98]

3. **Senzor videnia / vizuálny senzor (angl. Vision sensor)** — Vizuálny senzor rozpoznáva suroviny, viečka výrobkov a základne výrobkov a ich príslušné farby.



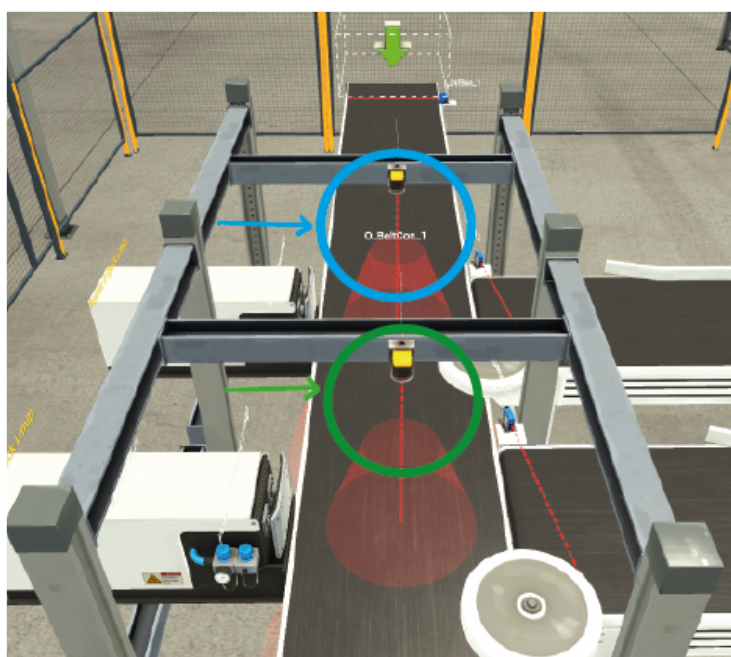
Obrázok č. 101: Vizuálny senzor [98]

Najprv boli vytvorené systémy pásových dopravníkov (C1-C9, CC1-CC2), po ktorých sa materiály budú prepravovať. Ďalej boli využité obrábacie centrá (*Machining centre* na obrázku č. 98) v oboch vetvách systému. Na generovanie materiálov sme použili vysielac (*vysielač / emitter* na obrázku č. 98). Na obrázku č. 102 je možné vidieť nastavenia, ktoré sme zvolili pre vysielac. Nastavené bolo generovanie dvoch typov materiálov – *Green Raw Material* a *Blue Raw Material* a tiež interval generovania. Pomocou *Max Time* a *Min Time* bolo určené, aby sa materiály generovali každých 13-15 sekúnd. *Up to* určuje, koľko materiálov má byť vygenerovaných. My sme nechali hodnotu 0, čo znamená, že nebola určená žiadna hranica.



Obrázok č. 102: Nastavenia pre vysielateľ (emitter) [98]

Ďalej bol navrhnutý subsystém na triedenie materiálu. To bude zabezpečovať obrazový senzor a triediť sa bude podľa farby neopracovaného materiálu. Každému senzoru sme nastavili, na ktorý druh materiálu má reagovať (obrázok č. 103). Po detekcii materiálu sa spustí pneumatický triedič (*pusher* na obrázku č. 98), ktorý zatlačí materiál na správny dopravníkový pás a ten ho dopraví k obrábaciemu centru.



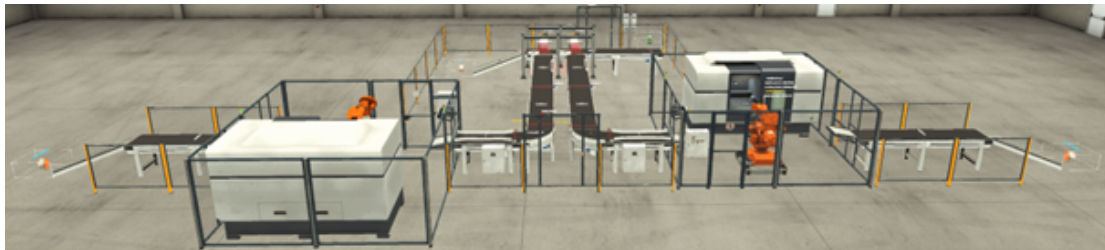
Obrázok č. 103: Sensory videnia [98]

Následne sme virtuálny systém doplnili o retroreflexné senzory a odrazky (podobne ako v prvej štúdii — obrázok č. 60). Tieto senzory pomáhajú riadiť chod systému. Vďaka nim vieme detegovať, kde sa materiál nachádza a spustiť dopravníky podľa potreby.

Okrem riadenia bolo potrebné realizovať aj zobrazovanie údajov o počte materiálov v systéme. Na zobrazovanie týchto údajov nám slúžia displeje priamo vo virtuálnom modeli. Jeden je umiestnený na začiatku systému, zvyšné dva pri obrábacích centrách.

Na záver každej vetvy bol umiestnený odstraňovač (*remover* na obrázku č. 98), ktorý zabezpečí odstránenie spracovaných materiálov (výsledných produktov), ktoré vyjdú z obrábacieho centra. Týmto modelujeme expedíciu tovaru zákazníkom.

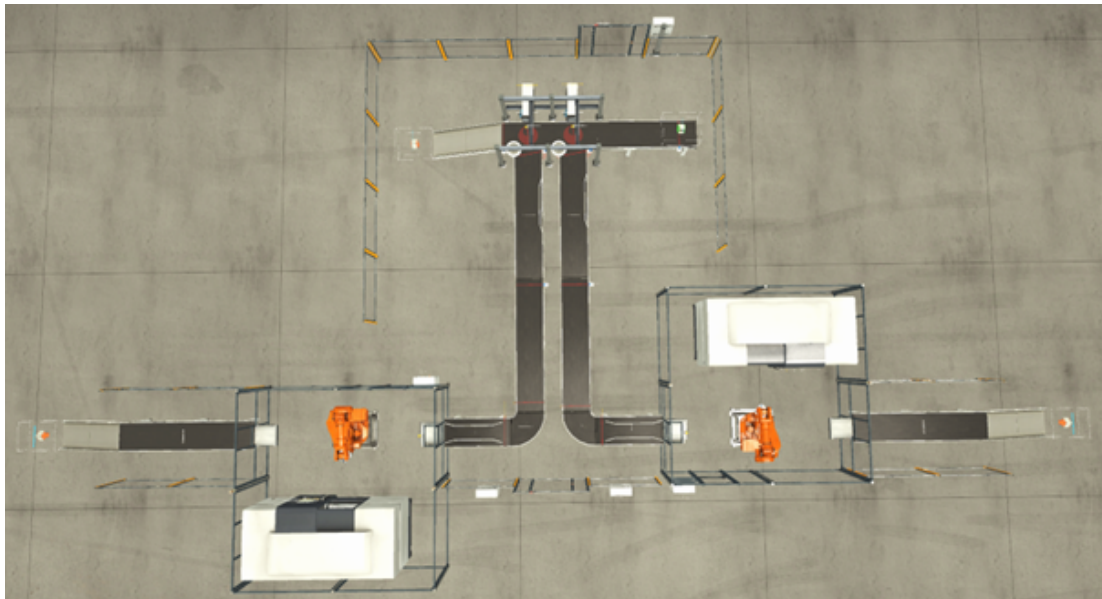
Na nasledujúcich obrázkoch č. 104 — 106 sú zobrazené rôzne pohľady na vytvorený kompletný model virtuálneho diskretného udalostného systému vo Factory I/O.



Obrázok č. 104: Pohľad na virtuálny výrobný systém (uhol č. 1) [98]



Obrázok č. 105: Pohľad na virtuálny výrobný systém (uhol č. 2) [98]



Obrázok č. 106: Pohľad na virtuálny výrobný systém (uhol č. 3) [98]

### 4.3.3 Riadenie diskrétného udalostného systému

Riadenie udalostného systému je navrhnuté v softvéri Codesys. Pre riadiaci program sme si zvolili PLC jazyk zvaný štruktúrovaný text (angl. Structured text). Pomocou tohto jazyka bol vytvorený riadiaci program, ktorý zabezpečuje chod celého virtuálneho výrobného systému.

Riadenie začína spustením systému a vygenerovaním materiálu vysielateľom. Akonáhle difúzny senzor umiestnený priamo pod vysielateľom deteguje materiál, dopravníkový pás (C1) sa pohne a počet vygenerovaných materiálov sa zvýši o hodnotu 1. Materiál putuje dopravníkovým pásom k obrazovému senzoru. V prípade, že senzor zaznamenal správny materiál, aktivuje sa mechanický triedič, ktorý posunie materiál na nasledujúci správny dopravníkový pás (C2 alebo C3). Prvý dopravník (C1) sa vypne a počet materiálov danej farby v systéme sa zvýši o 1.

V prípade, že by vysielateľ vygeneroval iný materiál, ako je určený pre túto časť linky, je popod obrazové senzory priamo dopravený na kĺzačku a pomocou nej k odstraňovaču.

Sieť dopravníkov (C2-C9, CC1, CC2), ktorá začína po dopravení materiálu triedičom, je riadená s využitím senzorov a informácií, či je obrábacie centrum práve vyťažené alebo čaká na materiál. Obrábacie centrum je počas celej simulácie zapnuté a čaká na informáciu, či bol materiál doručený. Na toto slúži obrazový senzor, ktorý je súčasťou obrábacieho centra.

Premenné a funkcie programu boli deklarované v hornej časti súboru PLC\_PRG. Pri premenných si je dôležité uvedomiť, že premenné budú posielané do Factory I/O, preto ich treba dôkladne pomenovať, aby ich vedel ľahko identifikovať a filtrovať. Typ premennej v Codesyse priradíme pomocou „:“.

Syntax: *nazov\_premennej* : *typ\_premennej*.

Každaj premennej bol zadefinovaný typ, prípadne aj hodnota. V riadiacom programe využívame premenné typu BOOL, UINT a funkciu CTU.

Dátový typ premennej BOOL pozná iba dve hodnoty — 0 a 1, prípadne TRUE a FALSE. Dátový typ UINT je neznamienkový integer, čo znamená, že je to premenná, do ktorej vieme uložiť hodnoty od 0 po 65 535.

Funkciu CTU poskytuje knižnica Standard Library. Táto funkcia zabezpečuje inkrementáciu. Vstupmi funkcie sú CU, RESET a PV. Výstupmi funkcie sú Q a CV.

Vstupy funkcie CTU:

- CU je typu BOOL, v prípade TRUE zvýši hodnotu uloženú v CV o 1.
- RESET je typu BOOL, v prípade TRUE sa vyresetuje hodnota uložená v CV.
- PV je typu WORD (rovnaká veľkosť ako UINT), určuje hornú hranicu pre inkrementáciu CV.

Výstupy funkcie CTU:

- Q je typu BOOL, defaultne FALSE, na TRUE sa nastaví, keď  $CV \geq PV$ .
- CV je typu WORD, ukladá sa tu hodnota po inkrementovaní.

Na obrázku č. 107 môžeme vidieť premenné, ktoré sme použili pri písaní riadiaceho programu. Na riadkoch 47-50 a 54 sa nachádzajú nielen zadefinované typy premenných ale aj priradené hodnoty danej premennej.



```
PLC_PRG x Symbol Configuration
1 PROGRAM PLC_PRG
2 VAR
3     // Outputs
4     O_BeltCon_1 : BOOL;
5     O_BeltCon_2 : BOOL;
6     O_BeltCon_3 : BOOL;
7     O_BeltCon_4 : BOOL;
8     O_BeltCon_5 : BOOL;
9     O_BeltCon_6 : BOOL;
10    O_BeltCon_7 : BOOL;
11    O_BeltCon_8 : BOOL;
12    O_BeltCon_9 : BOOL;
13    O_CurvedCon_1 : BOOL;
14    O_CurvedCon_2 : BOOL;
15    O_Pusher_1 : BOOL;
16    O_Pusher_2 : BOOL;
17
18    // Inputs
19    I_RetrorefSen_1 : BOOL;
20    I_RetrorefSen_2 : BOOL;
21    I_RetrorefSen_3 : BOOL;
22    I_RetrorefSen_4 : BOOL;
23    I_VisionSen_1 : BOOL;
24    I_VisionSen_2 : BOOL;
25    I_DifSen_1 : BOOL;
26    I_DifSen_2 : BOOL;
27    I_DifSen_3 : BOOL;
28    I_DifSen_4 : BOOL;
29    I_DifSen_5 : BOOL;
30    I_DifSen_6 : BOOL;
31
32    // Outputs
33    O_MachineOn_1 : BOOL; // Machining center on/off signal
34    // Inputs
35    I_MachineOpened_1 : BOOL;
36    I_MachineIsBusy_1 : BOOL;
37    // Outputs
38    O_MachineOn_2 : BOOL; // Machining center on/off signal
39    // Inputs
40    I_MachineOpened_2 : BOOL;
41    I_MachineIsBusy_2 : BOOL;
42
43    CTU_Counter_All : CTU;
44    CTU_Counter_Blue : CTU;
45    CTU_Counter_Green : CTU;
46    CTU_Counter_ErrorSorting : CTU;
47    MaterialCount_All : UINT := 0; // CounterAll variable
48    MaterialCount_Blue : UINT := 0; // CounterBlue variable
49    MaterialCount_Green : UINT := 0; // CounterGreen variable
50    MaterialCount_ErrorSorting : UINT := 0; // CounterErrorSorting variable
51
52    //other variables
53    O_Emitter : BOOL;
54    bSimulationEnabled : BOOL := FALSE;
55    run : BOOL;
56    pause : BOOL;
57
58 END_VAR
```

Obrázok č. 107: Premenné riadiaceho programu [98]

## RIADIACI PROGRAM REALIZOVANÝ V CODESYSE

Riadiaci program slúži na riadenie chodu virtuálneho modelu linky. Na tvorbu programu bol zvolený štruktúrovaný text. Na riadenie využívame funkcie, podmienky a cykly.

Hodnotu do premennej typu BOOL je možné zapísať dvomi spôsobmi, prvým je priradenie a druhým spôsobom je využitie SET a RESET. Na obrázku č. 108 môžeme vidieť obidva spôsoby zapísania premennej.

```
10 | O_BeltCon_8 S= TRUE;  
11 | O_BeltCon_9 S= TRUE;  
12 | O_Pusher_1 := FALSE;  
13 | O_Pusher_2 := FALSE;  
.. |
```

Obrázok č. 108: Ukážka zápisu hodnoty do premennej [98]

Rozdiel medzi týmito zápsmi je v tom, že pri využití „:=“ bude hodnota priradená len na dobu trvania podmienky. Pri použití „S=“ sa nastaví hodnota na dobu, pokiaľ túto hodnotu neupravíme pomocou „R=“. Na obrázku č. 109 je možné vidieť ukážku prepísania hodnoty pomocou SET a RESET. Útržok kódu slúži na riadenie dopravníka. Ak je premenná *O\_Pusher\_1* nastavená na hodnotu TRUE, tak sa premenná *O\_Belt\_Con\_2* nastaví na hodnotu TRUE. Toto nastavenie trvá, pokiaľ nie je premenná *I\_DifSen\_4* nastavená na TRUE. Ak je hodnota premennej *I\_DifSen\_4* TRUE, tak sa hodnota premennej *O\_BeltCon\_2* zmení na FALSE. Toto nastavenie trvá, pokiaľ nie je znovu splnená prvá podmienka. (*Riadime sa pravdivostnou tabuľkou pre SET a RESET.*)

```
38 | //run and stop O_BeltCon_2  
39 | IF O_Pusher_1 THEN  
40 |     O_BeltCon_2 S= TRUE;  
41 | END_IF  
42 | IF I_DifSen_4 THEN  
43 |     O_BeltCon_2 R= TRUE;  
44 | END_IF
```

Obrázok č. 109: Ukážka zápisu hodnoty do premennej [98]

Na nasledujúcom obrázku č. 110 je možné vidieť ukážku kódu, ktorý slúži na riadenie prvého dopravníka a následne na triedenie materiálu podľa farby.

```

13 //run O_BeltCon_1
14 IF I_DifSen_1 THEN
15     O_BeltCon_1 S=TRUE;
16 END_IF
17
18 //stop O_BeltCon_1
19 IF I_VisionSen_1 OR I_VisionSen_2 THEN
20     O_BeltCon_1 R= TRUE;
21 END_IF
22
23 //sorted blue
24 IF I_VisionSen_1 THEN
25     O_Pusher_1 := TRUE;
26 END_IF
27
28 //sorted green
29 IF I_VisionSen_2 THEN
30     O_Pusher_2 := TRUE;
31 END_IF

```

Obrázok č. 110: Ukážka kódu na riadenie dopravníka a triedenie [98]

Obrázok č. 111 je ukážkou kódu, ktorý slúži na riadenie dopravníkového systému na spracovanie modrého materiálu. Kód riadi dopravníky od triediča až po obrábacie centrum. Na riadenie dopravníkového systému so zeleným materiálom sme použili to-  
tožný kód s upravenými názvami premenných podľa potreby.

```

33 //MACHINING FOR BLUE BRANCH
34 //run and stop O_BeltCon_2
35 IF O_Pusher_1 THEN
36     O_BeltCon_2 S= TRUE;
37 END_IF
38
39 //run O_BeltCon_4
40 IF I_DifSen_2 THEN
41     O_BeltCon_4 S=TRUE;
42     O_BeltCon_2 S= TRUE;
43 END_IF
44
45 IF I_DifSen_4 THEN
46     O_BeltCon_2 R= TRUE;
47 END_IF
48
49 IF NOT I_RetrorefSen_1 AND I_MachineOpened_1 AND NOT I_MachineIsBusy_1 THEN
50     O_CurvedCon_1 S= TRUE;
51     O_BeltCon_6 S= TRUE;
52     O_BeltCon_4 S=TRUE;
53 END_IF
54
55 IF NOT I_RetrorefSen_1 AND I_MachineIsBusy_1 THEN
56     O_CurvedCon_1 R= TRUE;
57     O_BeltCon_6 R= TRUE;
58     O_BeltCon_4 R=TRUE;
59 END_IF
60

```

Obrázok č. 111: Riadiaci kód pre modrú vetvu [98]

Obrázok č. 112 obsahuje kód, ktorý sme použili na zaznamenávanie počtu mate-

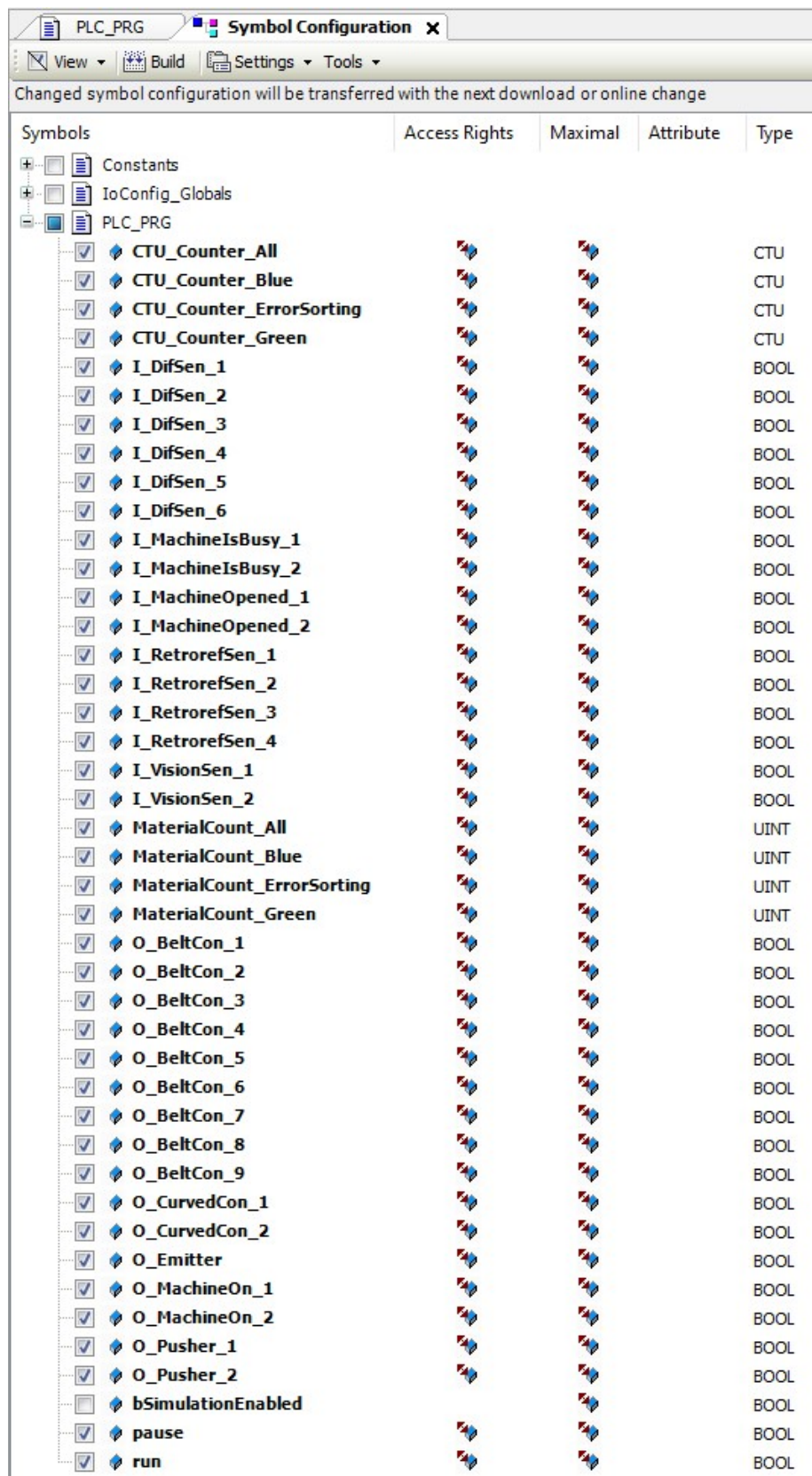
riálov v danom systéme. Hodnota je inkrementovaná pri zmene hodnoty konkrétneho senzora. Funkcia sleduje nábežnú hranu signálu, pretože pokiaľ by sa hodnota inkrementovala počas celej doby, kedy senzor deteguje materiál a má zmenenú hodnotu, tak by sa konečná hodnota nezvýšila o 1 ale o viac — potenciálne až nekonečno.

```
89 // Check if a material is detected and increment the counter
90 //Counter ALL
91 CTU_Counter_All(PV := 200, CU := NOT I_DifSen_1 );
92 IF I_DifSen_1 THEN
93     MaterialCount_All:=(CTU_Counter_All.CV);
94 END_IF
95
96 //Counter BLUE
97 CTU_Counter_Blue(PV := 100, CU := NOT I_VisionSen_1 );
98 IF I_VisionSen_1 THEN
99     MaterialCount_Blue:=(CTU_Counter_Blue.CV);
100 END_IF
101
102 //Counter GREEN
103 CTU_Counter_Green(PV := 100, CU := NOT I_VisionSen_2 );
104 IF I_VisionSen_2 THEN
105     MaterialCount_Green:=(CTU_Counter_Green.CV);
106 END_IF
107
108 //Counter error sorting
109 CTU_Counter_ErrorSorting(PV := 100, CU := NOT I_DifSen_6 );
110 IF I_DifSen_6 THEN
111     MaterialCount_ErrorSorting:=(CTU_Counter_ErrorSorting.CV);
112 END_IF
```

Obrázok č. 112: Ukážka kódu slúžiaceho na inkrementáciu premenných [98]

#### 4.3.4 Prepojenie Factory I/O s Codesys pomocou OPC UA

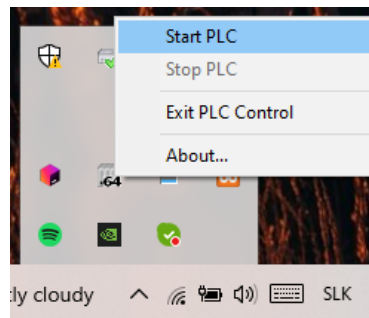
Po napísaní riadiaceho programu nasleduje jeho prepojenie s virtuálnym modelom systému vo Factory I/O, pričom toto bude prebiehať prostredníctvom komunikačného protokolu OPC UA. Na prepojenie premenných z Codesysu a Factory I/O je nutné do štruktúry Codesys projektu pridať objekt *Symbol Configuration*, kde treba označiť všetky premenné, ktoré chceme využiť vo Factory I/O. Následne realizujeme *Build* pre *Symbol Configuration* (v prípade neskoršej úpravy premenných je potrebné vždy znovu spustiť *Build* pre *Symbol Configuration*).



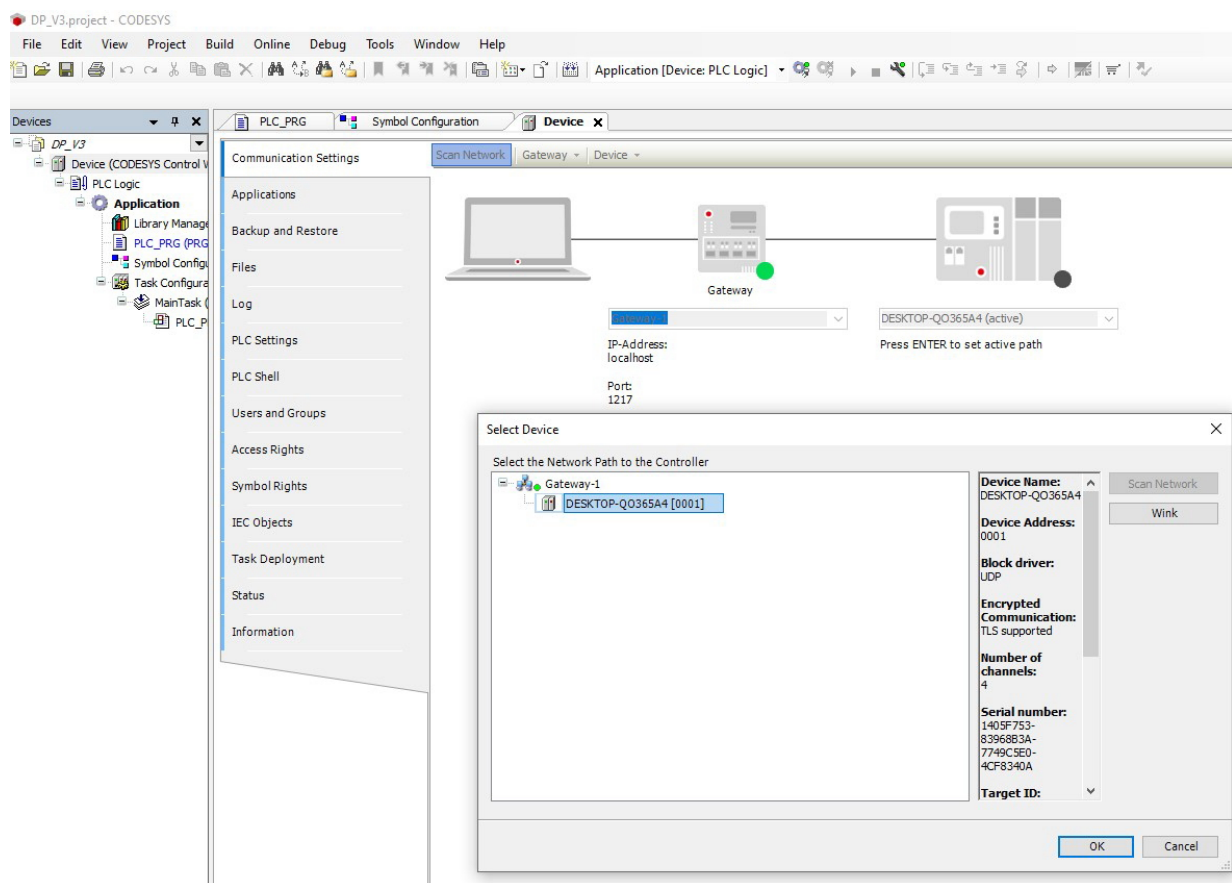
Obrázok č. 113: Objekt Symbol Configuration [98]

Po príprave premenných je potrebné urobiť build pre celý projekt. Je potrebné zap-

núť PLC server (obrázok č. 114), v možnosti *Device* zvolíť *Communication Settings*, spustiť *Scan Network* a vybrať zariadenie (obrázok č. 115). Následne je treba spustiť *Online->Login a Debug->Start*.



Obrázok č. 114: Zapnutie PLC servera [98]



Obrázok č. 115: Skenovanie siete a zvolenie zariadenia [98]

Po príprave Codesysu a spustení OPC servera je potrebné otvoriť Factory I/O, kde sme si zvolili scénu s naším pripraveným modelom diskretného udalostného systému. V konfigurácii je potrebné nastaviť ovládač OPC Client DA/UA, kde ako OPC UA Server zadáme `opc.tcp://localhost:4840` (v našom prípade sme použili `OPCUAServer@DESKTOP-`

QO365A4 (UA)). Ako filter pre premenné sme použili PLC. Po nastavení konfigurácie sme napárali premenné z riadiaceho programu a Factory I/O (obrázok č. 116).



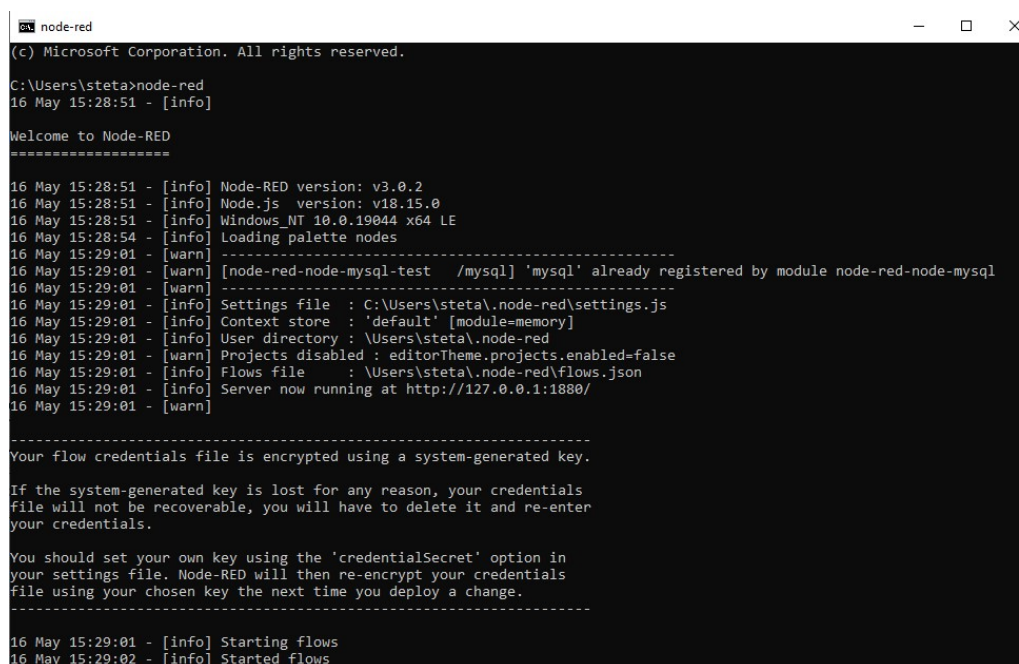
Obrázok č. 116: Párovanie premenných [98]

Po prepojení a správnom nakonfigurovaní sa riadiaci program nahral a je možné spustiť virtuálny model systému. Ako už bolo uvedené, komunikáciu zabezpečuje moderný komunikačný protokol OPC UA.

### 4.3.5 Node-RED a práca s dátami pomocou OPC UA a MQTT

Middleware Node-RED neslúži v tejto prípadovej štúdii na riadenie virtuálneho modelu systému, ale funguje ako prostredník medzi OPC UA serverom a webovou aplikáciou. Slúži na preposielanie dát, v tomto prípade do webovej aplikácie. Webová aplikácia totiž nevie v opísanom prípade priamo komunikovať s OPC UA serverom, nakoľko by tento OPC UA server musel figurovať na verejnej IP adrese, ktorú v tomto prípade nemáme k dispozícii.

Pre potreby tejto edukačnej prípadovej štúdie bol zvolený lokálne bežiaci Node-RED. Ten je potrebné nainštalovať do zariadenia. Prerekvizivitou lokálne bežiacieho Node-REDu je podporovaná verzia Node.JS, tú je teda tiež potrebné nainštalovať. Po splnení všetkých krokov a nainštalovaní je možné Node-RED spustiť. Spúšťame ho pomocou príkazu *node-red*, ktorý zadáme v konzole (cmd.exe) alebo PowerShell, nakoľko pracujeme s operačným systémom Windows. Na obrázku č. 117 sa nachádza výpis po spustení Node-RED.

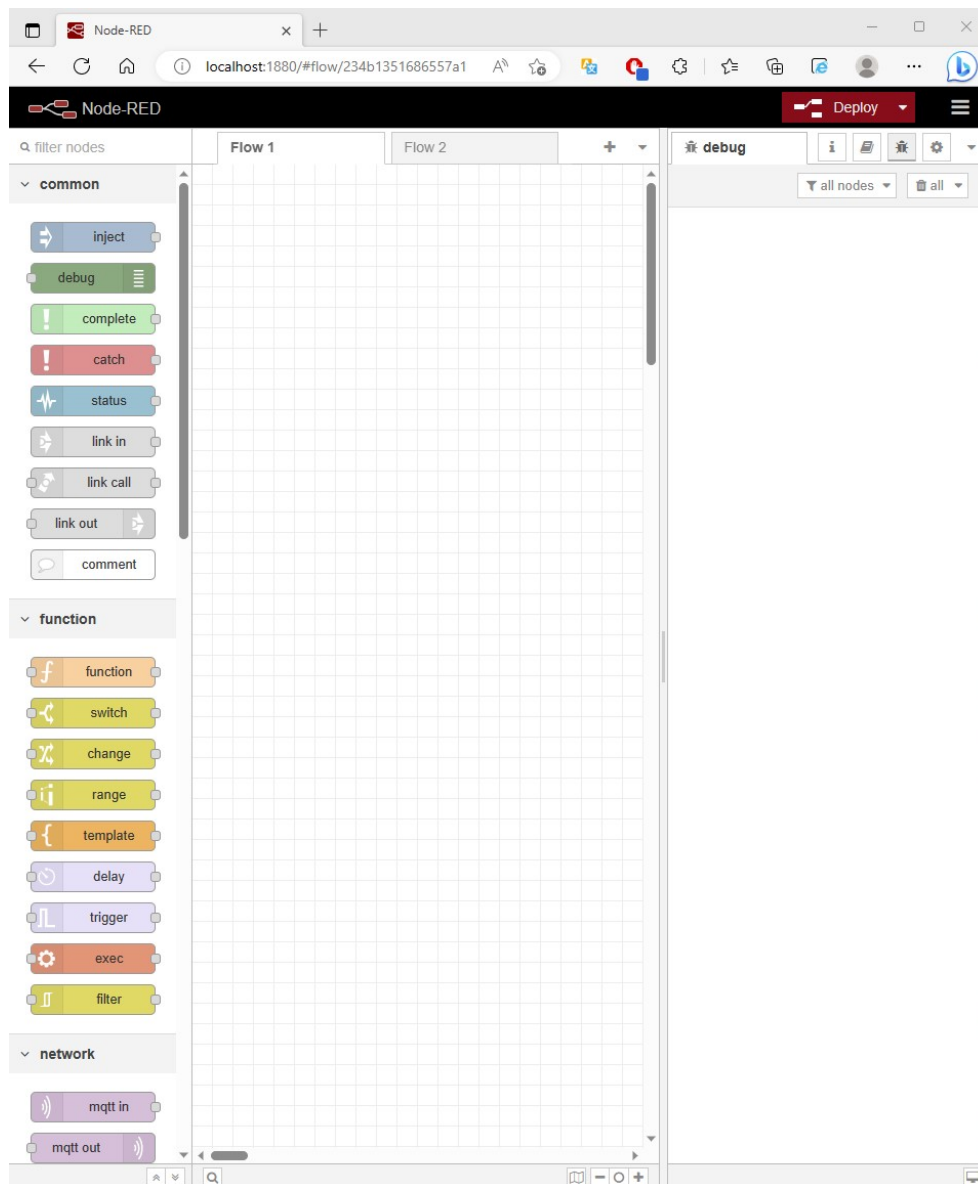


```
node-red
(c) Microsoft Corporation. All rights reserved.
C:\Users\steta>node-red
16 May 15:28:51 - [info]
Welcome to Node-RED
=====
16 May 15:28:51 - [info] Node-RED version: v3.0.2
16 May 15:28:51 - [info] Node.js version: v18.15.0
16 May 15:28:51 - [info] Windows_NT 10.0.19044 x64 LE
16 May 15:28:54 - [info] Loading palette nodes
-----
16 May 15:29:01 - [warn] [node-red-node-mysql-test /mysql] 'mysql' already registered by module node-red-node-mysql
16 May 15:29:01 - [warn]
-----
16 May 15:29:01 - [info] Settings file : C:\Users\steta\.node-red\settings.js
16 May 15:29:01 - [info] Context store : 'default' [module=memory]
16 May 15:29:01 - [info] User directory : \Users\steta\.node-red
16 May 15:29:01 - [warn] Projects disabled : editorTheme.projects.enabled=false
16 May 15:29:01 - [info] Flows file : \Users\steta\.node-red\flows.json
16 May 15:29:01 - [info] Server now running at http://127.0.0.1:1880/
16 May 15:29:01 - [warn]
-----
Your flow credentials file is encrypted using a system-generated key.
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
16 May 15:29:01 - [info] Starting flows
16 May 15:29:02 - [info] Started flows
```

Obrázok č. 117: Príkazový riadok po spustení lokálneho Node-RED [98]

Do editora Node-RED sa potom dostaneme tak, že v prehliadači otvoríme adresu <http://localhost:1880>.





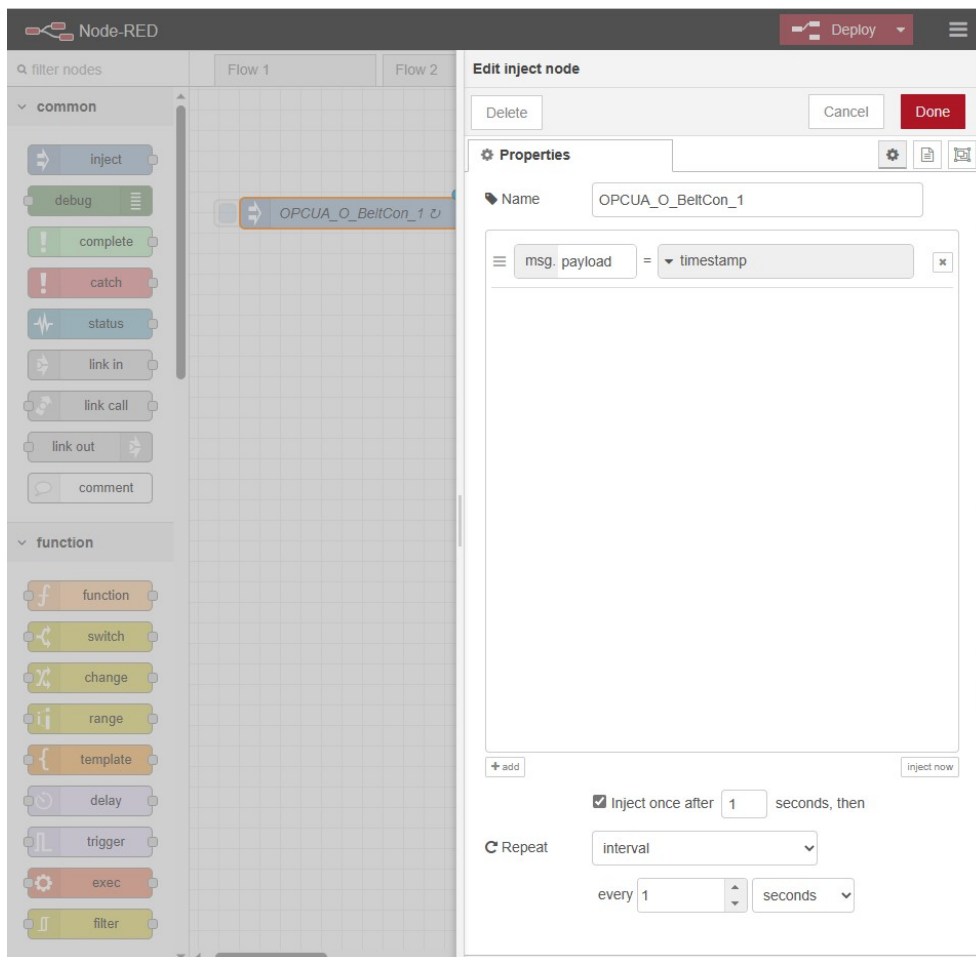
Obrázok č. 118: Prázdny Node-RED editor [98]

Programové toky v Node-REDe tvoríme prepájaním uzlov. Tvorbu tokov pre túto edukačnú prípadovú štúdiu môžeme rozdeliť do troch častí – čítanie dát z OPC UA servera a následné posielanie dát na MQTT broker, čítanie dát z MQTT brokera a ukladanie do databázy a treťou časťou je zasielanie dát späť na OPC UA server. Túto časť si však rozoberieme neskôr, pretože je to súčasť spätnej komunikácie z webovej aplikácie, ktorá slúži na manuálne (núdzové) riadenie nášho virtuálneho systému.

## ČÍTANIE DÁT Z OPC UA SERVERA A NÁSLEDNÉ POSIELANIE DÁT NA MQTT BROKER

Prvý uzol, ktorý bol použitý v toku, je uzol *Inject*. Slúži na vloženie správy do toku alebo (inak povedané) aktiváciu danej vetvy uzlov. Na vkladanie sme použili nastave-

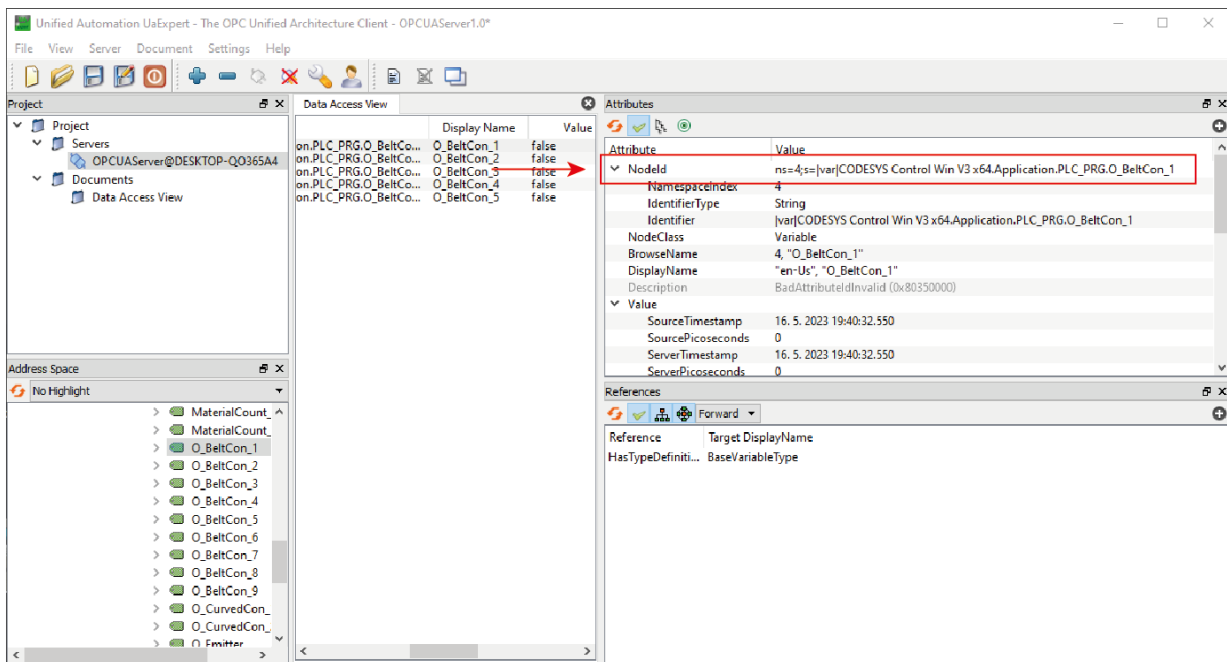
nie intervalu. Nastavenie uzla je možné vidieť na obrázku č. 119. Nastavený bol názov premennej do poľa *Name* a v dolnej časti bol nastavený interval na vkladanie správy. Prvé vloženie nastane jednu sekundu po pripojení a následne sa budú správy vkladat opakovane v intervale jednej sekundy. Toto nastavenie bolo použité v uzloch pre každú premennú, ktorej správu chceme prijímať z OPC UA servera.



Obrázok č. 119: Nastavenie uzla Inject [98]

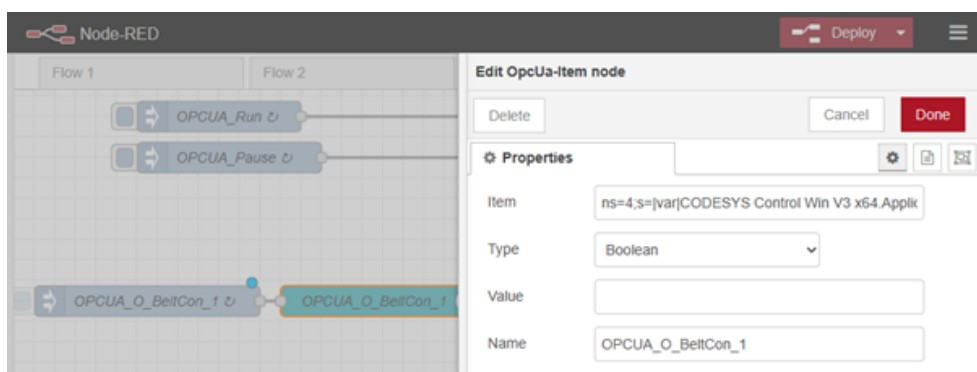
Nasledujúce dva uzly využívame z knižnice *node-opcua*, ktorá ponúka uzly na komunikáciu pomocou komunikačného protokolu OPC UA. Knižnicu nainštalujeme pomocou voľby *Manage Palette*, kde si vyhľadáme *node-red-contrib-opcua* a spustíme inštaláciu. Následne sa uzly pre komunikáciu OPC UA pridajú do palety editora.

Uzol *OpcUa-Item* sme využili na definovanie položky, názvu a typu premennej. Do poľa *Item* sme vložili charakteristiku premennej, ktorej správu chceme prijať. Na získanie tohto údaj (charakteristiky) sme podobne ako v prvej prípadovej štúdiu využili softvér (OPC UA klient) *UaExpert*, ktorým sme sa pripojili na OPC UA server vytvorený pomocou Codesysu a do stredného okna sme umiestnili všetky premenné, ktoré potrebujeme. Dôležitý údaj pre nás je *NodeId* každej premennej (obrázok č. 120).



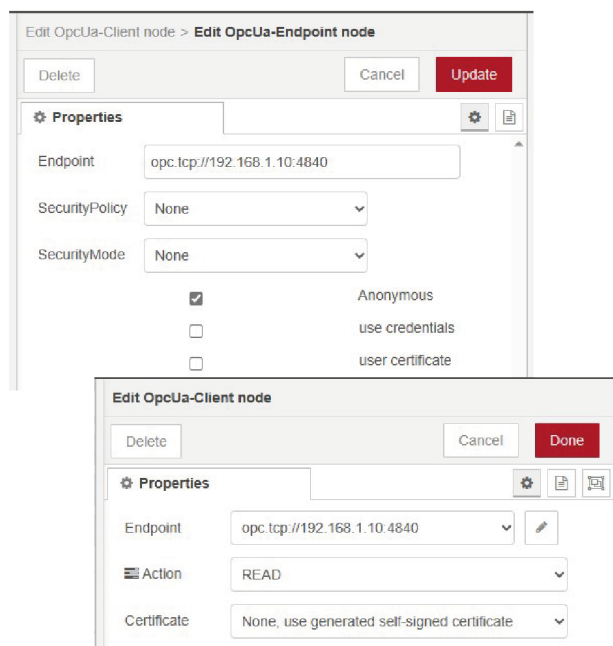
Obrázok č. 120: NodeId v programe UaExpert [98]

Následne je potrebné vyplniť typ premennej, ktorej správu budeme prijímať a jej názov. Nastavenia tohto uzla vidíme na obrázku č. 121.

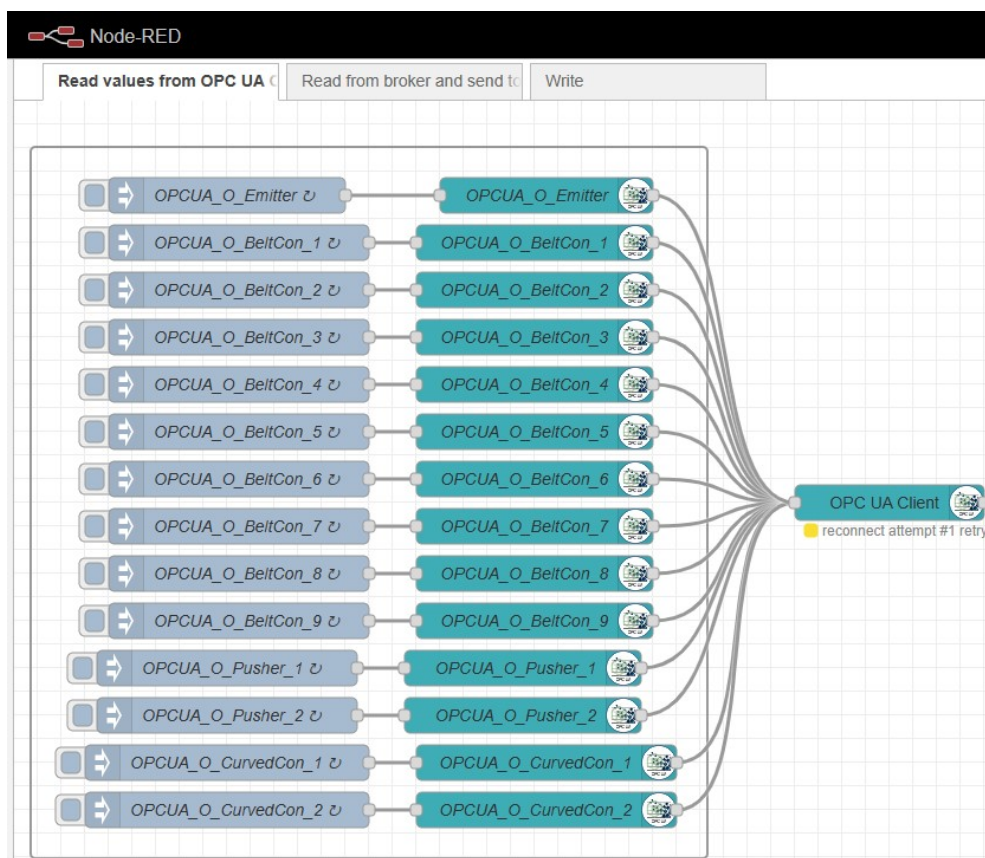


Obrázok č. 121: Uzol OpcUa Item s nastavením [98]

Uzol *OpcUa-Client* sme využili na pripojenie sa k OPC UA serveru. Ako endpoint sme zadali *opc.tcp://192.168.1.10:4840* – čo zodpovedá LAN IP zariadenia. V možnosti Action bol zvolaný READ, pretože chceme správy z OPC UA servera čítať (obrázok č. 122). Po správnej konfigurácii sme uzly prepojili.



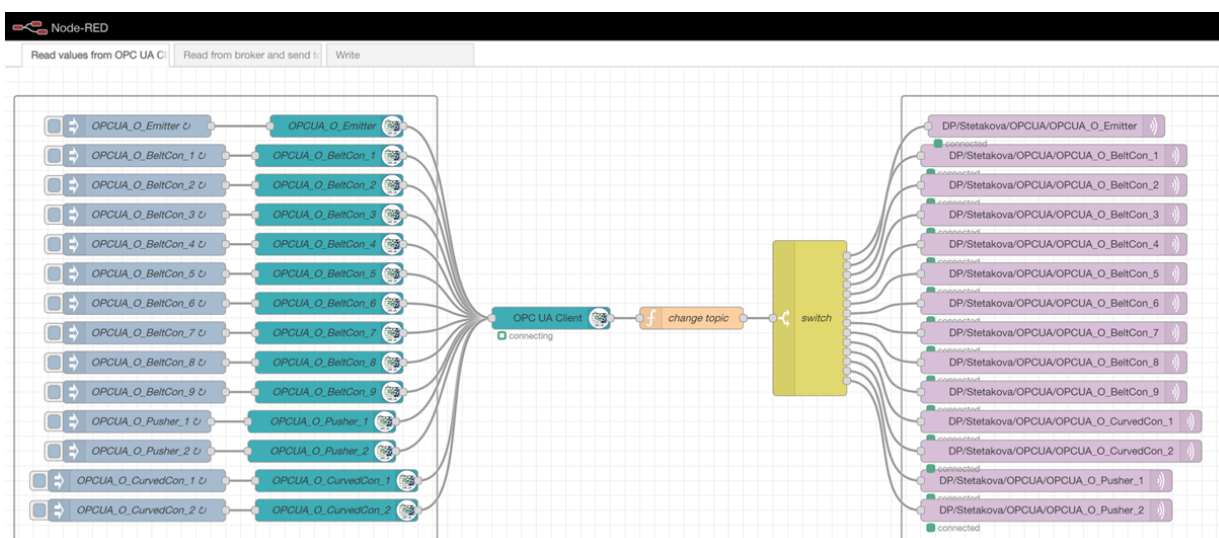
Obrázok č. 122: Nastavenie uzla OpcUa-Client a endpointu [98]



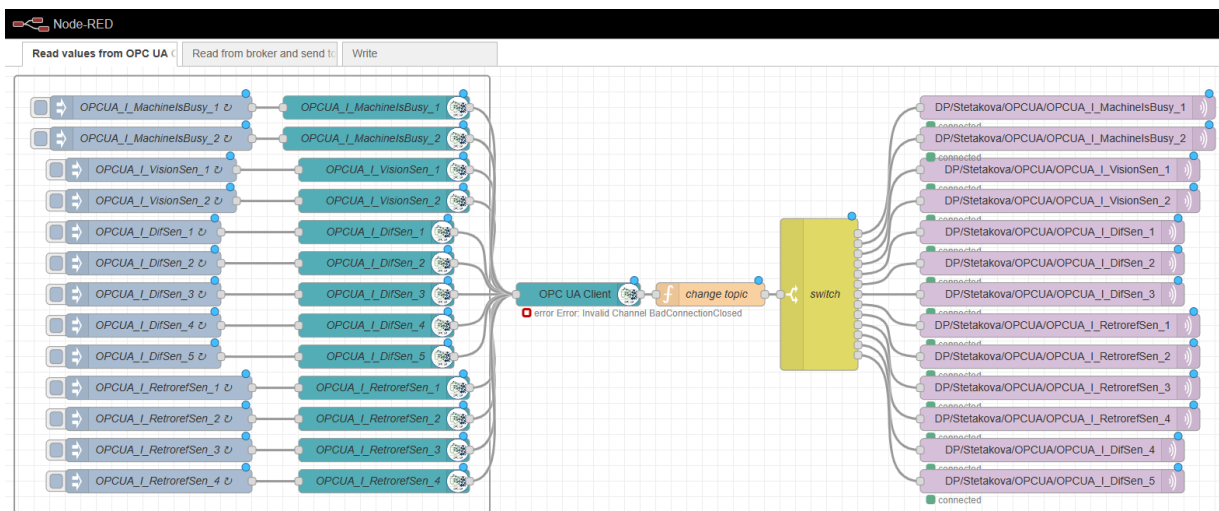
Obrázok č. 123: Prepojenie uzlov pre čítanie z OPC UA servera [98]

## POSIELANIE A PRIJÍMANIE DÁT POMOCOU MQTT

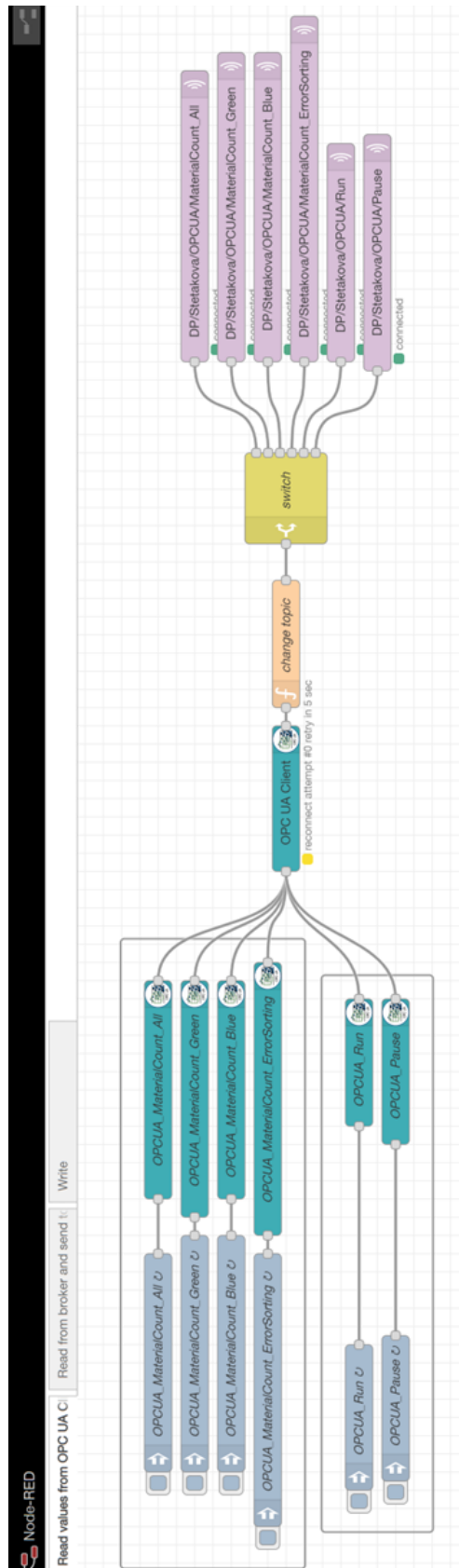
Nasleduje posielanie údajov na MQTT broker pomocou Node-RED uzla *mqtt out*. Avšak na to, aby sme ich mohli poselať so správnou témou (topic), sme využili na úpravu uzly *Function* a *Switch*. Uzol *Function* obsahuje kód napísaný v jazyku JavaScript, ktorý slúži na úpravu štruktúry správy a zmenu témy. Po úprave témy sme využili uzol *Switch*, ktorý zabezpečuje presmerovanie správy do *mqtt out* uzla so správnou témou. V uzle *mqtt out* sme ako server (broker) pre MQTT využili *broker.hivemq.com* a port 1883, čo je voľne dostupný broker aj pre testovacie účely. Po prepojení všetkých uzlov nám vznikla prvá časť tvorby toku dát. Pre lepší prehľad sme premenné rozdelili do skupín podľa vstupu, výstupu, premennej typu INT a riadiacich premenných.



Obrázok č. 124: Načítanie správ a ich posielanie na MQTT broker (výstupy) [98]



Obrázok č. 125: Načítanie premenných a ich posielanie na MQTT broker (vstupy) [98]



Obrázok č. 126: Načítanie premenných a ich posielanie na MQTT broker (INT a riadiace premenné) [98]

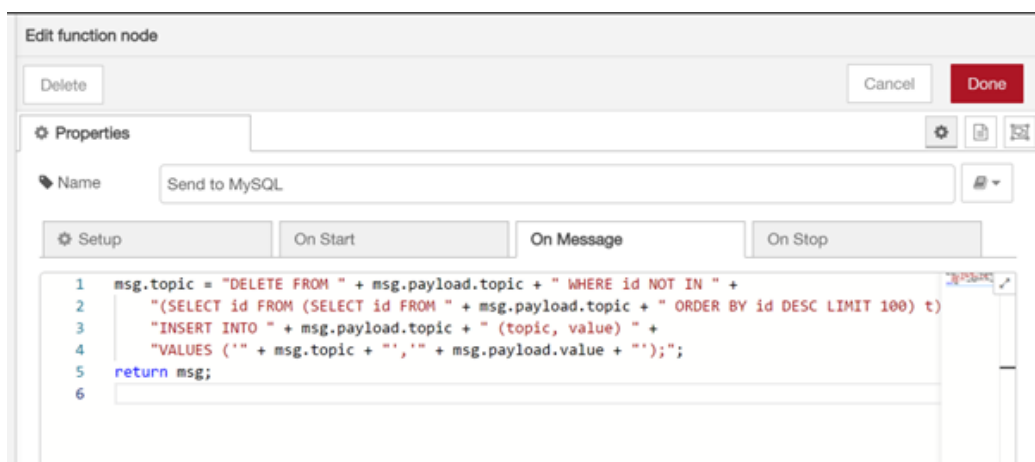
## ČÍTANIE DÁT Z MQTT BROKERA A UKLADANIE DO DATABÁZY

Druhá časť zabezpečuje zasielanie dopytu na MQTT broker s názvami tém, ktorých správy je žiadané prijímať. Následne sa tieto dáta spracujú a pomocou funkcie sa vytvorí dopyt (angl. query) na zápis údajov do databázy.

Na zasielanie dopytu a následné prijímanie správy konkrétnej témy bol využitý uzol *mqtt in*. Do nastavení bola zadaná adresa MQTT brokera, ktorého dáta prijímame a port. Nastavili sme dopytovanú tému a QoS (Quality of Service) level na 2. Toto znamená, že každý paket je doručený do cieľa presne jedenkrát.

Následne sme využili uzol *Function*. Napísali sme vlastný kód, ktorý do *msg.topic* vloží SQL dopyt (SQL query) pre MySQL databázu. Tento dopyt po odoslaní správy skontroluje počet záznamov a v prípade, že je počet záznamov viac ako 100, tak vyberie id prvých 100 záznamov z tabuľky a odstráni ich, čo zabezpečí zbytočné neplnenie databázy starými záznamami. Vždy teda zostane v tabuľke 100 najaktuálnejších záznamov. Po odstránení ďalej query zabezpečí vloženie nového záznamu do konkrétnej tabuľky s hodnotami správy, ktoré boli získané z MQTT brokera.

Na záver bol tok ukončený pripojením k uzlu *mysql*, ktorý zabezpečí pripojenie k databáze a vykonanie SQL dopytu.



Obrázok č. 127: Uzol Function s kódom na vytvorenie query [98]

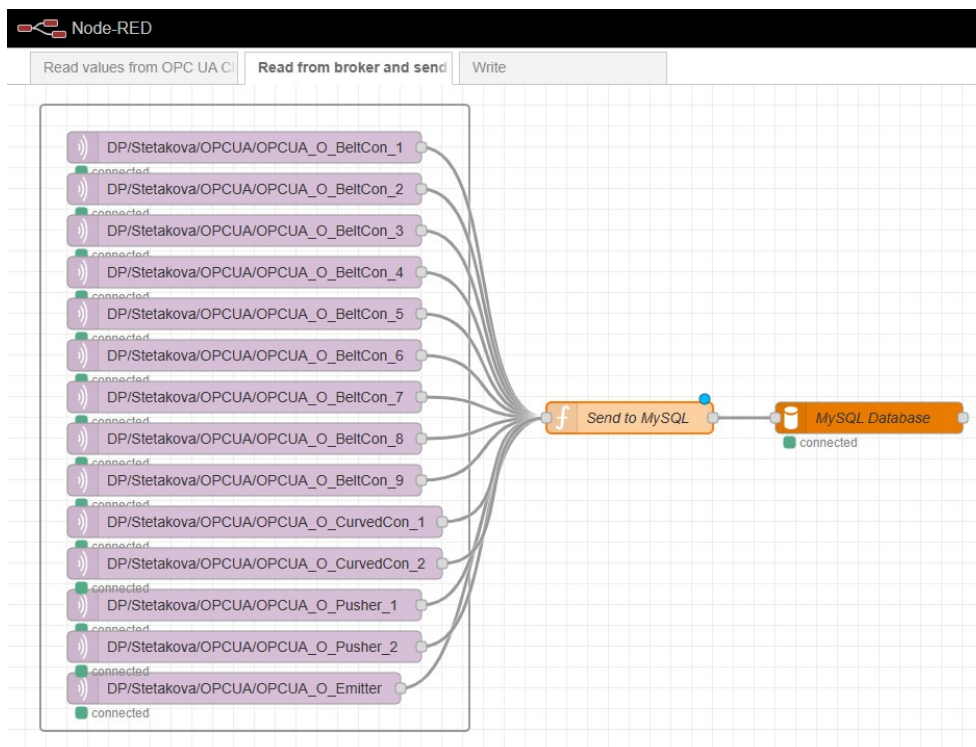
Pred spustením tejto časti je však potrebné vytvoriť si MySQL databázu a tabuľky v nej. V rámci prípadovej štúdie sme na vytváranie databázy a tabuliek využili PHP skript, vďaka ktorému sme sa vyhli manuálnemu vytváraniu každej tabuľky. Pre každú premennú bola vytvorená osobitná tabuľka so stĺpcami pre id, názov, hodnotu a časovú stopu (angl. timestamp).

Server: localhost Database: mynxqtdatabase Table: opcu\_a\_o\_beltcon\_1

Number of rows: 25

id	topic	value	timestamp
82798	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:45
82799	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:46
82800	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:47
82801	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:48
82802	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:49
82803	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:50
82804	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:51
82805	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:52
82806	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:53
82807	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:54
82808	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:55
82809	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:56
82810	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:57
82811	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:58
82812	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:25:59
82813	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:00
82814	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:01
82815	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:02
82816	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:03
82817	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:04
82818	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:05
82819	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:06
82820	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:07
82821	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:08
82822	DP/Stetakova/OPCUA/OPCUA_O_BeltCon_1	false	2023-05-17 17:26:09

Obrázok č. 128: Ukážka štruktúry databázy [98]



Obrázok č. 129: Tok pre načítanie dát a uloženie do databázy [98]



### 4.3.6 PHP webová aplikácia

Webová aplikácia slúži na zobrazenie posielaných údajov, ale je ju tiež možné použiť na manuálne ovládanie nášho virtuálneho modelu diskretného udalostného systému.

Vytvorili sme nový PHP projekt. Na vytvorenie PHP webovej aplikácie, ktorá dokáže prijímať a posilať dáta cez MQTT protokol je potrebné nainštalovať *php-mqtt/client*. Nainštalovali sme ďalej *composer*, ktorý slúži na správu knižníc a iných balíčkov pre PHP. Balíček *php-mqtt/client* je možné nainštalovať pomocou príkazu `composer require php-mqtt/client`, ktorý treba zadať do príkazového riadku s tým, aby ako pracovný priečinok bol nastavený priečinok s projektom. Po nainštalovaní je možné používať funkcie balíčka na odoberanie (`subscribe`) a publikovanie (`publish`) správ.

---

#### Programový modul č. 1 Kód na pripojenie k databáze [98]

---

```
1 <?php
2
3 $host = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "myMQTTdatabase";
7
8 // Create connection
9 $conn = mysqli_connect($host, $username, $password, $dbname);
10
11 // Check connection
12 if (!$conn) {
13     die("Connection failed: " . mysqli_connect_error());
14 }
15
16 ?>
```

---

Potom sme vytvorili PHP skript s kódom, ktorý zabezpečuje dopyt z databázy. Prpravili sme si prístupové údaje na pripojenie k databáze: názov servera, používateľské meno a heslo. Na pripojenie sme použili *mysqli* (programový modul č. 1). Následne sme vytvorili dopyt (programový modul č. 2), ktorým sme vybrali z tabuľky potrebné dáta a uložili sme ich do poľa.

Na ďalšie spracovanie týchto dát bol vytvorený skript v jazyku JavaScript (programový modul č. 3). Tu boli využité funkcie a AJAX, ktoré zabezpečili aktualizovanie prichádzajúcich údajov z databázy bez potreby znovunačítania webovej aplikácie. Po načítaní webovej aplikácie sa vygenerujú tabuľky aj s hodnotami z databáz. Následne

---

## Programový modul č. 2 Ukážka dopytu na získanie údajov z databázy [98]

---

```
1 <?php
2 require_once 'config.php';
3
4 // Query the data
5 $sql = "SELECT * FROM (SELECT * FROM OPCUA_O_Emitter ORDER BY timestamp DESC LIMIT 1) c1
6     UNION ALL
7     SELECT * FROM (SELECT * FROM OPCUA_I_DifSen_1 ORDER BY timestamp DESC LIMIT 1) c2
8     UNION ALL
9     SELECT * FROM (SELECT * FROM OPCUA_O_BetCon_1 ORDER BY timestamp DESC LIMIT 1) c3";
10
11 $result = mysqli_query($conn, $sql);
12
13
14 // Handle errors in the query
15 if (!$result) {
16     die('Query error: ' . mysqli_error($conn));
17 }
18
19 // Convert the data to an array
20 $data = array();
21
22 while ($row = mysqli_fetch_assoc($result)) {
23     $data[] = $row;
24 }
25 // Close the database connection
26 mysqli_close($conn);
27
28 // Return the data as JSON
29 header('Content-Type: application/json');
30 echo json_encode($data);
31
32 ?>
```

---

tiež funkcia zabezpečí aktualizáciu hodnoty prijatej z databázy každú sekundu.

---

### Programový modul č. 3 Funkcia v jazyku Javascript zabezpečujúca aktualizáciu údajov [98]

---

```
1 // Function to update the table data
2     function updateTableData() {
3         $.ajax({
4             url: 'getData/get_data_common.php',
5             type: 'post',
6             dataType: 'json',
7             success: function(response) {
8                 // Iterate over the response data and update values
9                 $.each(response, function(index, record) {
10                    const topic = record.topic;
11                    const myArray = topic.split("/");
12                    const variable = myArray[3];
13                    const value = record.value;
14
15                    // Update the circle with the new value
16                    updateCircle(variable, value);
17                });
18            }
19        });
20    }
21
22    // Update the table data periodically
23    setInterval(function(){
24        updateTableData();
25    }, 1000);
```

---

Na zobrazenie tabuliek sme vytvorili PHP súbor s HTML značkami, do ktorých sme vkladali vygenerované dáta na základe ID. Táto časť slúži na prijímanie a zobrazovanie údajov.

Následne bola vytvorená časť, ktorá nám zabezpečuje spätnú komunikáciu s MQTT brokerom. Na tieto účely sme z balíka *php-mqtt/client* využili funkciu *publish*. V jazyku JavaScript sme vytvorili funkcie *onClick*, ktoré prislúchajú k tlačidlám a ako odozva na kliknutie sa vytvorí správna správa a spustí sa PHP skript. Tento pomocou *publish* danú správu pošle na broker cez MQTT protokol. V programovom module č. 4 je možné vidieť súbor *publish.php* s pripojením na MQTT broker a následné odoslanie správy.

---

## Programový modul č. 4 Pripojenie na MQTT broker a odoslanie správy [98]

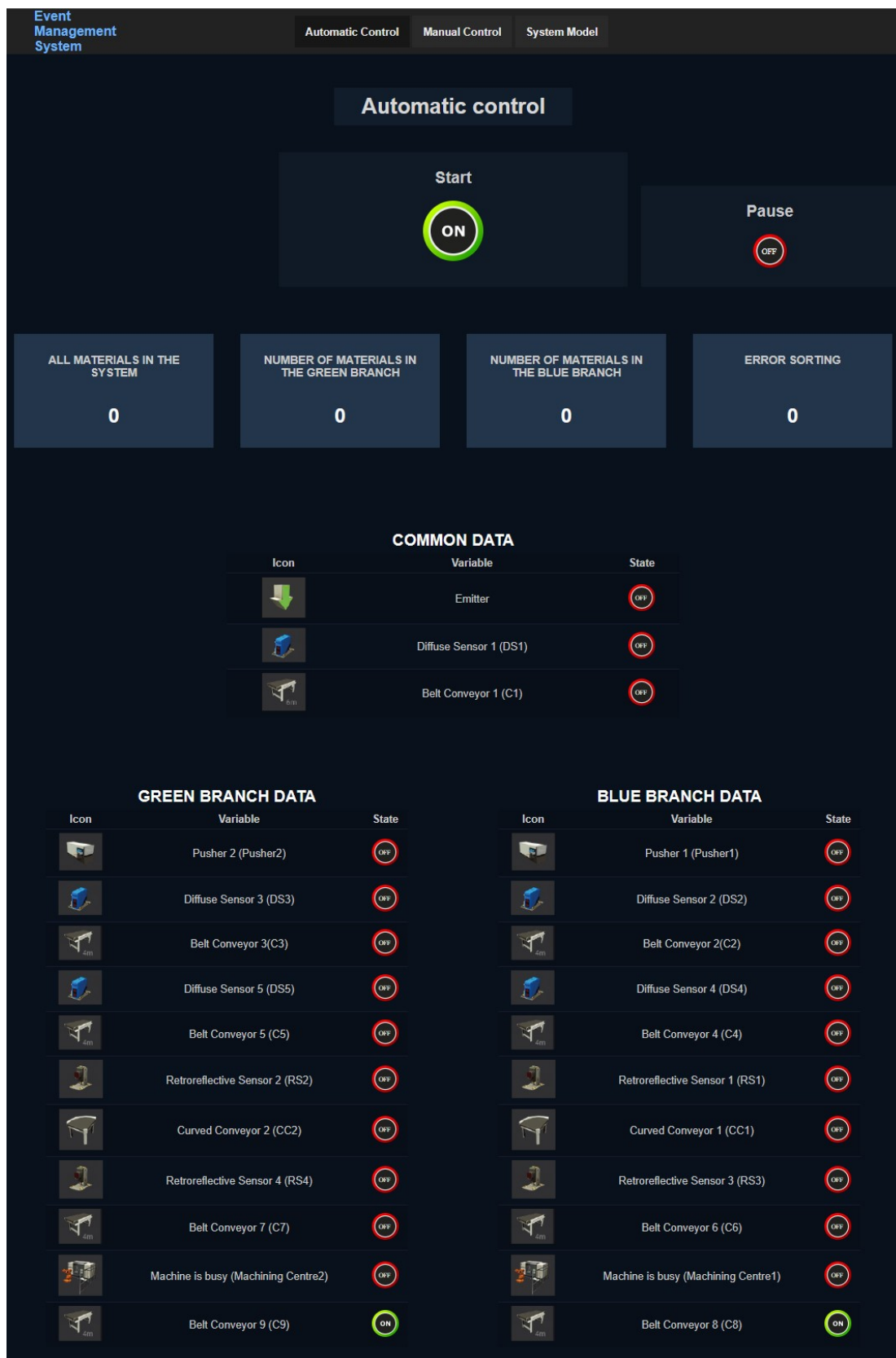
---

```
1 <?php
2
3 require_once 'vendor/autoload.php';
4
5 use PhpMqtt\Client\MqttClient;
6
7 // Get the JSON object sent in the POST request
8 $data = json_decode(file_get_contents('php://input'), true);
9
10 // Extract the topic and value from the JSON object
11 $topic = $data['topic'];
12 $value = $data['value'];
13
14 $client = new MqttClient('broker.hivemq.com', 1883);
15 $client->connect();
16 $client->publish($topic, $value);
17 $client->disconnect();
18
19 // Redirect back to the index page
20 header("Location: index.php");
21 die();
22 ?>
```

---

Na záver bola webová aplikácia naštýlovaná pomocou CSS súboru.

Finálna webová aplikácia obsahuje **tri podstránky**. Prvá podstránka **Automatic Control** (obrázok č. 130) slúži na zobrazenie stavov premenných. Vidíme tu tri tabuľky, karty s hodnotami určujúce počet materiálov v systéme. Ďalej sú tu dve osobitné karty. Jedna zobrazuje stav továrne, teda či je virtuálny model systému zapnutý alebo vypnutý. Druhá zobrazuje, či je továreň v stave *Pause*. Druhá podstránka **Manual Control** (obrázok č. 131) slúži užívateľovi na prípadné manuálne zásahy do systému.



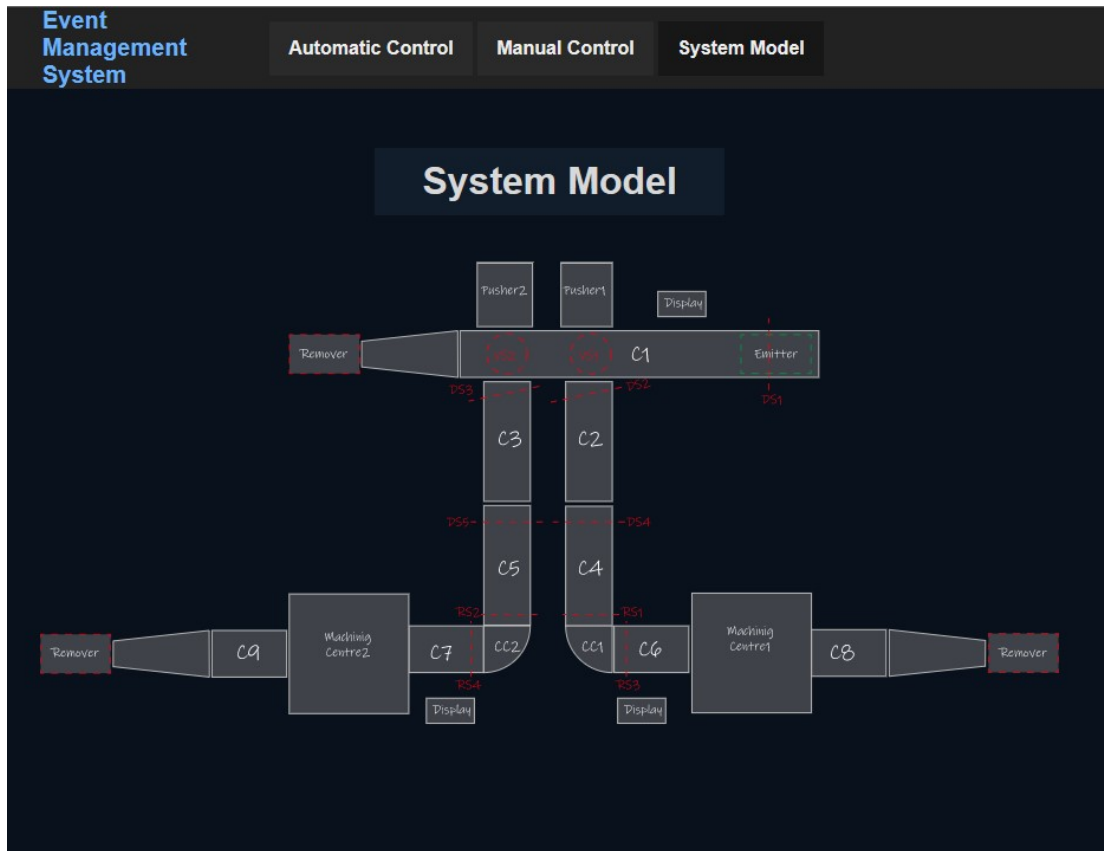
Obrázok č. 130: Ukážka webovej aplikácie — Automatic Control [98]



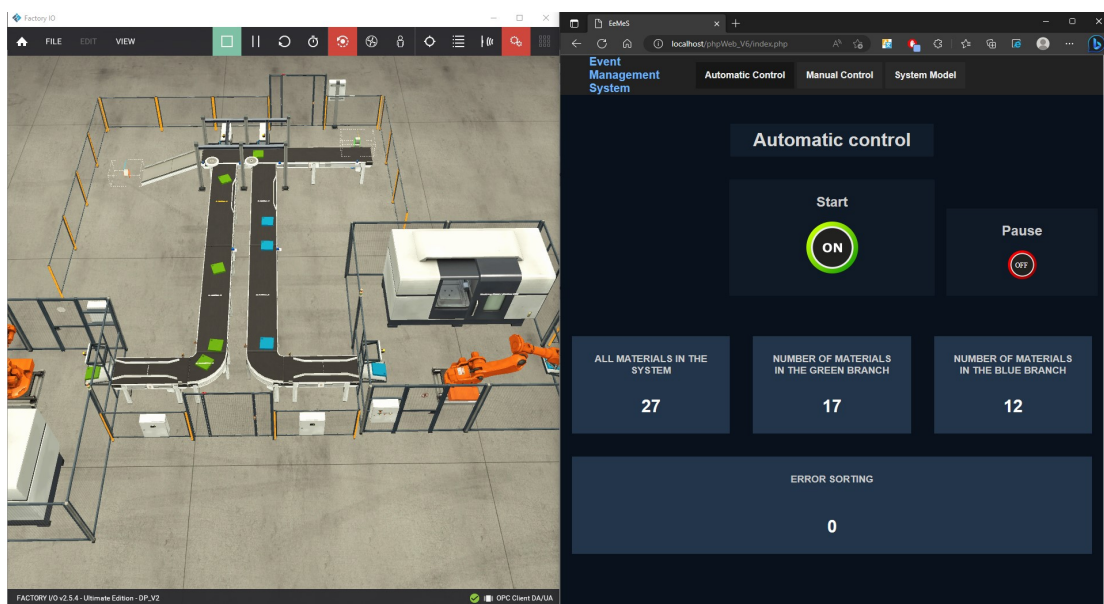
Obrázok č. 131: Ukážka webovej aplikácie — Manual Control [98]

Nachádzajú sa tu tiež dve osobitné karty, ktoré v tomto prípade slúžia nielen na zobrazenie stavu, ale tiež aj na riadenie spustenia alebo pozastavenia továrne. Okrem toho sa tu nachádzajú tri tabuľky, ktoré obsahujú zobrazenie stavu daného prvku a dve tlačidlá, ktoré zabezpečujú zapnutie alebo vypnutie. Toto riadenie je zabezpečené spätnou komunikáciou, ktorá bude vysvetlená v nasledujúcej podkapitole. Tretia pod-

stránka obsahuje schému modelu výrobného systému (obrázok č. 132). Toto môže slúžiť na lepšiu orientáciu používateľa.

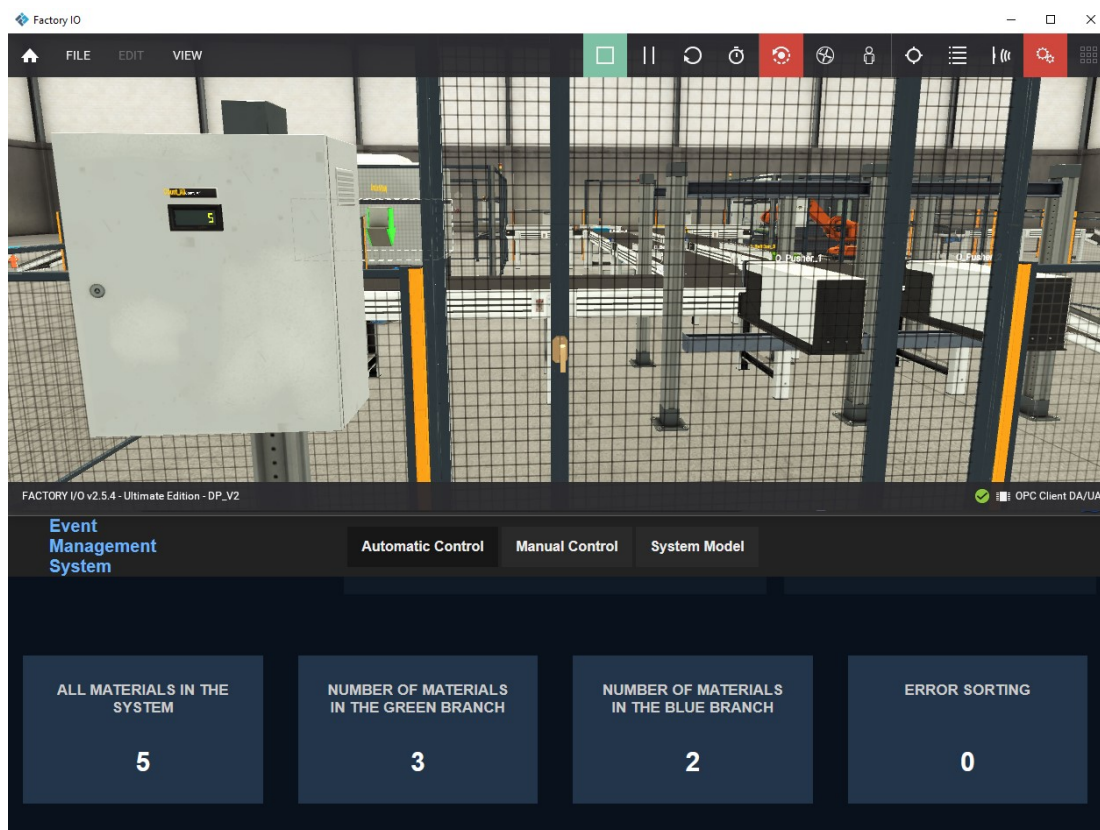


Obrázok č. 132: Ukážka webovej aplikácie — System Model [98]



Obrázok č. 133: Ukážka prepojenia webovej aplikácie s virtuálnou továrňou [98]



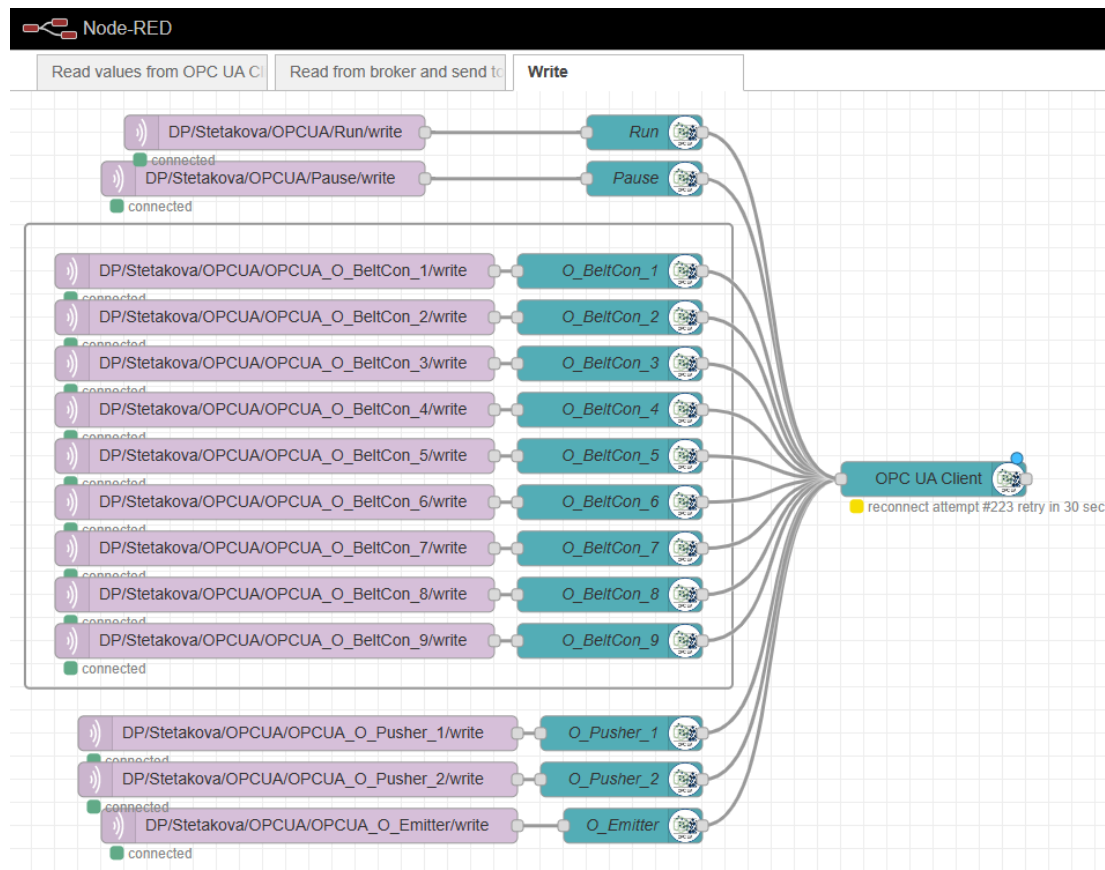


Obrázok č. 134: Ukážka prepojenia webovej aplikácie s virtuálnou továrňou [98]

### 4.3.7 Spätná komunikácia v Node-RED s OPC UA serverom

Spätná komunikácia zabezpečuje interakciu medzi webovou aplikáciou a OPC UA serverom prostredníctvom middlewaru Node-RED.

Prvú časť spätnej komunikácie zabezpečuje webová aplikácia pomocou balíka *php-mqtt/client*, z ktorého sme použili funkciu *publish*. Najprv sme sa pripojili k MQTT brokeru (v opisovanej štúdii ide o HiveMQ) a následne publikovali správu. Tieto publikované správy boli zaslané na MQTT broker, odkiaľ ich potrebujeme získať a poslať ďalej. Na zabezpečenie čítania z MQTT brokera sme znovu použili prostredník Node-RED, kde sme vytvorili nový programový tok (obrázok č. 135). Využili sme uzol *mqtt in*, ktorému sme určili, ktoré témy chceme odoberať. Následne bol vybraný uzol *OpcUa Item*, kde boli vyplnené údaje o premennej v riadiacom programe. Do Item sme vložili NodeId danej premennej, správne sme určili typ premennej a vyplnili sme názov — ten slúži len na označenie nášho uzla pre prehľadnosť. Potom sme využili uzol *OpcUa Client* a nakonfigurovali sme ho. Vyplnili sme endpoint (localhost sme nahradili LAN IP zariadenia) a ako Action sme zvolili WRITE. Vďaka vyplnenému NodeId si následne server priradí hodnoty pre správnu premennú.



Obrázok č. 135: Spätná komunikácia s OPC UA Serverom v Node-RED [98]

V poslednej fáze bolo riešenie otestované. Nosným prvkom systému je komunikácia, preto bola hlavným predmetom testovania práve celková komunikácia medzi jednotlivými zložkami celého systému. Po otestovaní komunikácie bola testovaná webová aplikácia vrátane jej responzivity na rôznych zariadeniach a rozmeroch obrazoviek. Výsledkom testovania bolo potvrdenie funkčnosti riadenia a komunikácie a aj responzivita webovej aplikácie. Z toho plynie, že návrh a následná implementácia riadiaceho a monitorovacieho systému boli funkčné.

Video k tejto edukačnej prípadovej štúdii je možné nájsť na tejto adrese:

- <https://youtu.be/k4RjiVXSW0>

Ďalšie informácie, návody a programové projekty je možné nájsť na e-learningovej webovej stránke <https://elearning.mechatronika.cool/?p=8348>.

# Zoznam použitej literatúry

- [1] ASAD, U., KHAN, M., KHALID, A., A LUGHMANI, W. A. Human-Centric Digital Twins in Industry: A Comprehensive Review of Enabling Technologies and Implementation Strategies. *Sensors* 23, 8 (2023).
- [2] ASHTON, K., ET AL. That 'Internet of Things' thing. *RFID journal* 22, 7 (2009), 97–114.
- [3] BENEŠOVÁ, A., A TUPA, J. Requirements for education and qualification of people in industry 4.0. *Procedia Manufacturing* 11 (2017), 2195 – 2202. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy. <https://doi.org/10.1016/j.promfg.2017.07.366>.
- [4] BEŇO, L. *Moderné metódy ovládania mechatronických zariadení s využitím hlasových povelov založené na technológii Edge computingy*. Dizertačná práca (Vedúci práce: Erik Kučera). Bratislava: FEI STU, 2022. 106 s.
- [5] BEŇO, L., PRIBIŠ, R., A DRAHOŠ, P. Edge Container for Speech Recognition. *Electronics* 10, 19 (2021). DOI: 10.3390/electronics10192420.
- [6] BOURKE, K. *What is CODESYS and Why is it Important?* Real Pars. [online]. [cit. 2023-04-26]. Dostupné z: <https://www.motioncontroltips.com/instruction-lists-ils-plc-programming/>.
- [7] BRECKO, A., KAJÁTI, E., A PAPCUN, P. Industry 5.0 – technológie: bio-inšpirované technológie a inteligentné materiály. *ATP Journal* 02/2022 (2022).
- [8] BRECKO, A., KAJÁTI, E., A ZOLOTOVÁ, I. Industry 5.0 – technológie: interakcie medzi človekom a strojom. *ATP Journal* 01/2022 (2022), 40–41.
- [9] BRIŠ, L. *Využitie formalizmov Petriho sietí v riadení laboratórnych systémov*. Diplomová práca (Vedúci práce: Alena Kozáková; Konzultant: Erik Kučera). Bratislava: FEI STU, 2016. 58 s.
- [10] BUCSAI, S. *Ovládanie a monitorovanie IoT zariadení s využitím zmiešanej reality*. Diplomová práca (Vedúci práce: Erik Kučera). Bratislava: FEI STU, 2019.
- [11] BUCSAI, S., KUČERA, E., HAFFNER, O., A DRAHOŠ, P. Control and Monitoring of IoT Devices Using Mixed Reality Developed by Unity Engine. In *2020 Cybernetics & Informatics (K&I)* (2020), pp. 1–8.

- [12] BURGER, A., KOZIOLEK, H., RÜCKERT, J., PLATENIUS-MOHR, M., A STOMBERG, G. Bottleneck Identification and Performance Modeling of OPC UA Communication Models. In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering* (Apr. 2019), ICPE '19, ACM. DOI: 10.1145/3297663.3309670.
- [13] CARTWRIGHT, A. *OPC-UA: the Flow of Data*. [online]. [cit. 2024-01-20]. Dostupné z: <https://medium.com/ai-build-techblog/opc-ua-the-flow-of-data-7c3e5c870a4c>.
- [14] CAVALIERI, S., SALAFIA, M. G., A SCROPPO, M. S. Integrating OPC UA with web technologies to enhance interoperability. *Computer Standards & Interfaces* 61 (Jan. 2019), 45–64. DOI: 10.1016/j.csi.2018.04.004.
- [15] CHAUDHRY, T., JUNEJA, A., A RASTOGI, S. AR Foundation for Augmented Reality in Unity. *International Journal of Advances in Engineering and Management* 3, 1 (2021), 1–7.
- [16] CIOLACU, M. I., SVASTA, P., BERG, W., A POPP, H. Education 4.0 for Tall Thin Engineer in a Data Driven Society. *2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)* (2017), 432–437.
- [17] COITO, T., MARTINS, M. S., VIEGAS, J. L., FIRME, B., FIGUEIREDO, J., VIEIRA, S. M., A SOUSA, J. M. A Middleware Platform for Intelligent Automation: An Industrial Prototype Implementation. *Computers in Industry* 123 (Dec. 2020), 103329. DOI: 10.1016/j.compind.2020.103329.
- [18] CROATTI, A., A RICCI, A. Towards the Web of Augmented Things. In *Software Architecture Workshops (ICSAW), 2017 IEEE International Conference on* (2017), IEEE, pp. 80–87.
- [19] DEAN, JAMES. *First-of-its-kind 3D-printed home blends concrete, wood*. [online]. [cit. 2024-01-27]. Dostupné z: <https://news.cornell.edu/stories/2022/09/first-its-kind-3d-printed-home-blends-concrete-wood>.
- [20] DRATH, R., MOSCH, C., HOPPE, S., FAATH, A., BARNSTEDT, E., FIEBIGER, B., A SCHLÖGL, W. Diskussionspapier–interoperabilität mit der verwaltungsschale, opc ua und automationml. *Technical Report. AutomationML eV and Industrial Digital Twin Association (IDTA) and OPC Foundation and VDMA* (2023).
- [21] DVOŘÁK, J. *Integrace měřičů spotřeby energií do SCADA systému, zpracování a vyhodnocení dat*. Diplomová práce (Vedúci práce: Jan Holub). Praha: ČVUT Fakulta elektrotechnická, 2017. 68 s.

- [22] EUROPEAN COMMISSION - DIRECTORATE GENERAL FOR RESEARCH AND INNOVATION. *Industry 5.0: towards a sustainable, human centric and resilient European industry*. Publications Office, 2021. DOI: 10.2777/308407.
- [23] FISCHERTECHNIK. *Indexed Line with two Machining Stations 24V - Simulation*. [online]. [cit. 2021-04-07]. Dostupné z: <https://www.fischertechnik.de/en/products/simulating/training-models/96790-sim-indexed-line-with-two-machining-stations-24v-simulation>.
- [24] FISCHERTECHNIK. *Punching Machine with Conveyor Belt 24v*. [online]. [cit. 2023-04-29]. Dostupné z: <https://www.fischertechnik.biz/punching-machine-with-conveyor-belt-24v>.
- [25] FORTUNATO, D., A BERNARDINO, J. Progressive Web Apps: An Alternative to the Native Mobile Apps. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI) (2018)*, IEEE, pp. 1–6.
- [26] GALLASCH, A. ThingWorx–Plattform zur Integration herausfordernder Anforderungen auf dem Shopfloor. *Produktions-und Verfügbarkeits-optimierung mit Smart Data Ansätzen (2018)*, 83–92.
- [27] GREŇČÍKOVÁ, A., KORDOŠ, M., A NAVICKAS, V. The Impact of Industry 4.0 on Education Contents. *Business: Theory and Practice* 22, 1 (Jan. 2021), 29–38. DOI: 10.3846/btp.2021.13166.
- [28] GUINARD, D., TRIFA, V., PHAM, T., A LIECHTI, O. Towards Physical Mashups in the Web of Things. In *Networked Sensing Systems (INSS), 2009 Sixth International Conference on (2009)*, IEEE, pp. 1–4.
- [29] HABIB, K., SAAD, M. H. M., HUSSAIN, A., SARKER, M. R., A ALAGHBARI, K. A. An Aggregated Data Integration Approach to the Web and Cloud Platforms through a Modular REST-Based OPC UA Middleware. *Sensors* 22, 5 (2022). DOI: 10.3390/s22051952.
- [30] HAFNER, O., A KUČERA, E. Multiplatform Mobile Application for Identification and Localization of Objects in Space. In *2020 Cybernetics & Informatics (K&I) (2020)*, pp. 1–9. DOI: 10.1109/KI48306.2020.9039849.
- [31] HALLIWELL, R. *Programmable Logic Controllers*. op-tec. [online]. [cit. 2023-05-17]. Dostupné z: <https://op-tec.co.uk/knowledge/using-programmable-logic-controllers/>.

- [32] HRÚZ, B., A MRAFKO, L. *Modelovanie a riadenie diskretných udalostných systémov: s využitím Petriho sietí a iných nástrojov*. Bratislava: Vydavateľstvo STU, 2003. ISBN: 80-227-1883-1.
- [33] HUBA, M., A KOZÁK, Š. From e-Learning to Industry 4.0. In *2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)* (Nov 2016), pp. 103–108. DOI: DOI: 10.1109/ICETA.2016.7802083.
- [34] IMTIAZ, J., A JASPERNEITE, J. Scalability of OPC-UA down to the chip level enables “Internet of Things”. In *2013 11th IEEE International Conference on Industrial Informatics (INDIN)* (July 2013), IEEE. DOI: 10.1109/indin.2013.6622935.
- [35] KAJÁTI, E., A ZOLOTOVÁ, I. Industry 5.0 – revolúcia alebo evolúcia? *ATP Journal* 12/2021 (2021), 36–37.
- [36] KOLEKTÍV ADVANTECH. *Quick Guide to OPC UA and Advantech IT/OT Convergent Solutions*. [online]. [cit. 2023-05-04]. Dostupné z: <https://page.advantech.com/en/global/industrial-automation/industrial-io/opc-ua>.
- [37] KOLEKTÍV AUTOMATIZÁCIA 365. *Ktoré PLC programovacie jazyky najviac používame?* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.automatizacia365.sk/2021/04/13/ktore-plc-programovacie-jazyky-najcastejsie-pouzivame>.
- [38] KOLEKTÍV AUTOROV WWW.OPTO22.COM. *Ako preklenúť medzeru medzi IT a OT*. *ATP Journal* 1/2017 (2017), 32–33.
- [39] KOLEKTÍV FIRMATA. *Firmata firmware for Arduino*. *GitHub*. 2016. [online]. [cit. 2020-08-20]. Dostupné z: <https://github.com/firmata/arduino>.
- [40] KOLEKTÍV HIVEMQ. *MQTT Essentials*. [online]. [cit. 2020-06-05]. Dostupné z: <https://www.hivemq.com/tags/mqtt-essentials/>.
- [41] KOLEKTÍV NODE-RED. *About Node-RED*. [online]. [cit. 2020-06-10]. Dostupné z: <https://nodered.org/about/>.
- [42] KOLEKTÍV NODE-RED. *Node-RED User Guide*. [online]. [cit. 2020-06-10]. Dostupné z: <https://nodered.org/docs/user-guide/>.
- [43] KOLEKTÍV OPENPLC PROJECT. *Modbus Address Mapping*. [online]. [cit. 2021-02-01]. Dostupné z: <https://www.openplcproject.com/reference/modbus/>.
- [44] KOLEKTÍV PRODUKTION 2030. *Ingenjör4.0*. [online]. [cit. 2021-02-07]. Dostupné z: <https://produktion2030.se/en/ingenjor-4-0>.

- [45] KOLEKTÍV RANDED.COM. *Information Technologies (IT) Vs Operational Technologies (OT)*. *Webranded*. [online]. [cit. 2020-08-06]. Dostupné z: <https://randed.com/information-technologies-it-vs-operational-technologies-ot/?lang=en>.
- [46] KOLEKTÍV REALITY REFLECTION. *Virtual, Augmented, Mixed-Reality; What is these all about?*. *Medium.com*. [online]. [cit. 2020-06-02]. Dostupné z: <https://medium.com/@realityreflection/vr-virtual-augmented-mixed-reality-what-is-these-all-about-25762bfda62a>.
- [47] KOLEKTÍV THE FOUNDRY. *VR? AR? MR? Sorry, I'm confused*. *The Foundry*. [online]. [cit. 2020-06-02]. Dostupné z: <https://www.foundry.com/insights/vr-ar-mr/vr-mr-ar-confused>.
- [48] KOLEKTÍV ÚRADU PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE INVESTÍCIE A INFORMATIZÁCIU. *Cloud computing*. *Informatizácia - Úrad podpredsedu vlády Slovenskej republiky pre investície a informatizáciu*. [online]. [cit. 2020-06-01]. Dostupné z: <http://www.informatizacia.sk/cloud-computing/22841s>.
- [49] KOSTOLÁNI, M., MURÍN, J., A KOZÁK, Š. Intelligent Predictive Maintenance Control Using Augmented Reality. In *2019 22nd International Conference on Process Control (PC19) (2019)*, pp. 131–135. DOI: 10.1109/PC.2019.8815042.
- [50] KOZÁK, Š., A HYPÍUSOVÁ, M. *Moderné metódy a algoritmy riadenia*. Bratislava: Slovenská chemická knižnica, 2016. ISBN: 978-80-89597-44-4.
- [51] KOZÁK, Š., RUŽICKÝ, E., ŠTEFANOVIČ, J., A SCHINDLER, F. Research and Education for Industry 4.0: Present Development. In *2018 Cybernetics & Informatics (K&I) (Feb 2018)*, pp. 1–7.
- [52] KUČERA, E., HAFFNER, O., DRAHOŠ, P., CIGÁNEK, J., LESKOVSKÝ, R., A ŠTEFANOVIČ, J. New Software Tool for Modeling and Control of Discrete-Event and Hybrid Systems Using Timed Interpreted Petri Nets. *Applied Sciences* 10, 15 (2020), 5027. DOI: 10.3390/app10155027.
- [53] KUNBUS GMBH. *Open Source IPC based on Raspberry Pi*. *Revolution Pi*. [online]. [cit. 2020-08-06]. Dostupné z: <https://revolution.kunbus.com/>.
- [54] KUČERA, E. *Využitie Petriho sietí na modelovanie a riadenie skladových systémov*. Diplomová práca (*Vedúci práce: Leo Mraško; Konzultant: Branislav Hrúz*). Bratislava: FEI STU, 2013. 81 s.

- [55] KUČERA, E. *Modelovanie a riadenie hybridných systémov s využitím Petriho sietí vyšších úrovní*. Dizertačná práca (Vedúci práce: Štefan Kozák). Bratislava: FEI STU, 2016. 121 s.
- [56] KUČERA, E. *Digitálne technológie a systémy pre Industry 4.0*. Habilitačná práca. Bratislava: FEI STU, 2020. 224 s.
- [57] KUČERA, E. *Internet of Things - slajdy z prednášok Virtuálna a zmiešaná realita pre Industry 4.0*. [online]. [cit. 2020-06-01]. Dostupné z: <http://vzri.mechatronika.cool/sites/default/files/Prednaska%2010%20IoT.pdf>.
- [58] KUČERA, E. *Prehľad cloudových služieb Microsoft Azure - slajdy z prednášok Virtuálna a zmiešaná realita pre Industry 4.0*. [online]. [cit. 2020-06-01]. Dostupné z: <http://vzri.mechatronika.cool/sites/default/files/Prednaska%208%20Azure.pdf>.
- [59] KUČERA, E. *Virtuálna, rozšírená a zmiešaná realita - slajdy z prednášok Virtuálna a zmiešaná realita pre Industry 4.0*. [online]. [cit. 2020-06-01]. Dostupné z: <http://vzri.mechatronika.cool/sites/default/files/Prednaska%207%20VR.pdf>.
- [60] KUČERA, E., HAFFNER, O., DRAHOŠ, P., A KOZÁKOVÁ, A. Modeling and Control of Discrete Event and Hybrid Systems Using Petri Nets and OPC Unified Architecture. *IEEE Access* 10 (2022), 120735–120751. DOI: 10.1109/ACCESS.2022.3222828.
- [61] KUČERA, E., HAFFNER, O., DRAHOŠ, P., LESKOVSKÝ, R., A CIGÁNEK, J. PetriNet Editor + PetriNet Engine: New Software Tool For Modelling and Control of Discrete Event Systems Using Petri Nets and Code Generation. *Applied Sciences* 10, 21 (2020). DOI: 10.3390/app10217662.
- [62] KÉPEŠIOVÁ, Z. *Monitorovanie a riadenie mechatronických systémov s využitím digitálneho dvojčata a zmiešanej reality*. Písomná práca k dizertačnej skúške (Vedúci práce: Danica Rosinová; Konzultant: Erik Kučera). Bratislava: FEI STU, 2019.
- [63] LESKOVSKÝ, R. *Moderné metódy ovládania a diagnostiky mechatronických zariadení s využitím IoT a zmiešanej reality*. Písomná práca k dizertačnej skúške (Vedúci práce: Danica Rosinová; Konzultant: Erik Kučera). Bratislava: FEI STU, 2020.
- [64] LEWICKI, P. *Controlling lights with the Hololens and Internet of Things*. [htmlfusion.com/jgfbso2](http://htmlfusion.com/jgfbso2). [online]. [cit. 2019-04-08]. Dostupné z: <https://tinyurl.com/jgfbso2>.
- [65] LIANG, Q., A LI, L. The Study of Soft PLC Running System. *Procedia Engineering* 15 (2011), 1234–1238. CEIS 2011. DOI: <https://doi.org/10.1016/j.proeng.2011.08.228>.



- [66] LOPATNIKOVÁ, S. *Modelovanie a riadenie udalostného systému s využitím PLC a cloudového riešenia*. Diplomová práca (Vedúci práce: Oto Haffner; Konzultant: Erik Kučera). Bratislava: FEI STU, 2022. 81 s.
- [67] MAHNKE, W., LEITNER, S.-H., A DAMM, M. *OPC Unified Architecture*. Springer Science & Business Media, 2009.
- [68] MARTON, M. *Riadenie udalostných systémov s využitím Petriho sietí a OPC UA*. Diplomová práca (Vedúci práce: Erik Kučera). Bratislava: FEI STU, 2022. 44 s.
- [69] MIKSAD, M. *Prepojenie open-source PLC s počítačom generovanou realitou*. Diplomová práca (Vedúci práce: Erik Kučera). Bratislava: FEI STU, 2023.
- [70] MILGRAM, P., A KISHINO, F. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems* 77, 12 (1994), 1321–1329.
- [71] OPENBCI. *Ultracortex "Mark IV" EEG Headset*. [online]. [cit. 2024-01-27]. Dostupné z: <https://shop.openbci.com/collections/frontpage/products/ultracortex-mark-iv>.
- [72] ORGANIZÁCIA SPOJENÝCH NÁRODOV (OSN). *17 cieľov udržateľného rozvoja*. [online]. [cit. 2024-01-27]. Dostupné z: [https://unis.unvienna.org/unis/sk/topics/sustainable\\_development\\_goals.html](https://unis.unvienna.org/unis/sk/topics/sustainable_development_goals.html).
- [73] PAJPACH, M., HAFFNER, O., KUČERA, E., A DRAHOŠ, P. Low-Cost Education Kit for Teaching Basic Skills for Industry 4.0 Using Deep-Learning in Quality Control Tasks. *Electronics* 11, 2 (2022). DOI: 10.3390/electronics11020230.
- [74] PAJPACH, M., PRIBIŠ, R., DRAHOŠ, P., KUČERA, E., A HAFFNER, O. Design of an Educational-development Platform for Digital Twins using the Interoperability of the OPC UA Standard and Industry 4.0 Components. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICEC-CME)* (2023), pp. 1–6. DOI: 10.1109/ICECCME57830.2023.10252941.
- [75] PAPCUN, P., MIČKO, K., A KAJÁTI, E. Industry 5.0 – technológie: bezpečný prenos, ukladanie a analýza údajov. *ATP Journal* 04/2022 (2022), 52–53.
- [76] PAVLOVIČ, D. *Riadenie udalostného systému s využitím PLC a mobilnej aplikácie*. Diplomová práca (Vedúci práce: Erik Kučera). Bratislava: FEI STU, 2023.
- [77] POHANKA, P. *Internet věcí*. [online]. [cit. 2020-06-01]. Dostupné z: <http://i2ot.eu/internet-of-things/>.

- [78] POPJÁKOVÁ, D., A MINTÁLOVÁ, T. Priemysel 4.0, čo mu predchádzalo a čo ho charakterizuje - geografické súvislosti. *Acta Geographica Universitatis Comenianae* 63, 2 (2019), 173–192.
- [79] PRIBIŠ, R., BEŇO, L., A DRAHOŠ, P. Asset Administration Shell Design Methodology Using Embedded OPC Unified Architecture Server. *Electronics* 10, 20 (2021). DOI: 10.3390/electronics10202520.
- [80] RAMBACH, J., PAGANI, A., STRICKER, D., ALEKSY, M., SCHMITT, J., LANGFINGER, M., SCHNEIDER, M., SCHOTTEN, H., MALIGNAGGI, A., KO, M., ET AL. Augmented Things: Enhancing AR Applications leveraging the Internet of Things and Universal 3D Object Tracking. In *IEEE International Conference on Industrial Technology (ICIT)* (2017), vol. 22, p. 25.
- [81] RIESZ, M. *PNEditor - a Petri Net editor*. *PNEditor*. 2014. [online]. [cit. 2016-04-24]. Dostupné z: <http://www.pneditor.org/>.
- [82] ROMERO, D., BERNUS, P., NORAN, O., STAHERE, J., A FAST-BERGLUND, Å. The Operator 4.0: Human Cyber-Physical Systems & Adaptive Automation Towards Human-Automation Symbiosis Work Systems. In *Advances in Production Management Systems. Initiatives for a Sustainable World* (Cham, 2016), I. Nääs, O. Vendrametto, J. Mendes Reis, R. F. Gonçalves, M. T. Silva, G. von Cieminski, a D. Kiritsis, Eds., Springer International Publishing, pp. 677–686.
- [83] STARK, E. *Moderné metódy ovládania a monitorovania mechatronických systémov s využitím počítačom generovanej reality*. Dizertačná práca (Vedúci práce: Peter Drahoš; Konzultant: Erik Kučera). Bratislava: FEI STU, 2019. 120 s.
- [84] STARK, E., KUČERA, E., HAFFNER, O., DRAHOŠ, P., A LESKOVSKÝ, R. Using Augmented Reality and Internet of Things for Control and Monitoring of Mechatronic Devices. *Electronics* 9, 8 (2020), 1272.
- [85] STERLING, I., A SWAROOP, P. *Control with your smart devices by staring and gesturing*. Arduino. [online]. [cit. 2019-04-08]. Dostupné z: <https://blog.arduino.cc/2016/07/26/control-with-your-smart-devices-by-staring-and-gesturing/>.
- [86] TIEGELKAMP, M., A JOHN, K.-H. *IEC 61131-3: Programming industrial automation systems*, vol. 166. Springer, 2010.
- [87] TRIFA, V., GUINARD, D., A CARRERA, D. *Web Thing Model*. W3C. [online]. [cit. 2020-04-08]. Dostupné z: <http://model.webofthings.io/>.

- [88] UNIPi.TECHNOLOGY. *Automatizace Jednoduše. Unipi*. [online]. [cit. 2020-08-06]. Dostupné z: <https://www.unipi.technology/cs/>.
- [89] UNITY TECHNOLOGIES. *Create a marker-based AR app*. [online]. [cit. 2023-05-06]. Dostupné z: <https://learn.unity.com/project/create-a-marker-based-ar-app>.
- [90] VIGANÒ, G. P. *M2MQTT for Unity*. GitHub. [online]. [cit. 2023-05-06]. Dostupné z: <https://github.com/gpvigano/M2MqttUnity>.
- [91] VORÁČOVÁ, V., PĚNIČKA, M., A VESELÝ, J. *Úvod do modelování procesů Petriho sítěmi*. Vyd. 1. Praha: Česká technika - nakladatelství ČVUT, 2008, 126 s. ISBN 978-80-01-03979-3.
- [92] WIPRO. *Edge Computing - Understanding the User Experience*. [online]. [cit. 2024-01-11]. Dostupné z: <https://www.wipro.com/infrastructure/edge-computing-understanding-the-user-experience/>.
- [93] ZHANG, Q., LIU, L., PU, C., DOU, Q., WU, L., A ZHOU, W. A Comparative Study of Containers and Virtual Machines in Big Data Environment. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (2018)*, pp. 178–185. DOI: 10.1109/CLOUD.2018.00030.
- [94] ZOLOTOVÁ, I., KAJÁTI, E., A POMŠÁR, L. Industry 5.0 – koncept, technológie, ciele. *ATP Journal 11/2021 (2021)*, 42–43.
- [95] ČESKÝ INSTITUT INFORMATIKY, ROBOTIKY A KYBERNETIKY ČVUT. *Testbed pro Průmysl 4.0. České vysoké učení technické v Praze*. [online]. [cit. 2020-06-03]. Dostupné z: <https://www.ciirc.cvut.cz/cs/teams-labs/testbed/>.
- [96] ČEŠEK, P. *Riadenie DEDES pomocou mikrokontrolérov s využitím Petriho sietí*. Diplomová práca (Vedúci práce: Alena Kozáková; Konzultant: Erik Kučera). Bratislava: FEI STU, 2016.
- [97] ČÍŽEK, J. *Počítačový master/slave je spoločensky nekorektní. Pripomína otroctví, a proto zmizí*. [online]. [cit. 2024-01-20]. Dostupné z: <https://www.zive.cz/clanky/pocitacovy-master/slave-je-spolecensky-nekorektni-pripomina-otroctvi-a-proto-zmizi/sc-3-a-204378/default.aspx>.
- [98] ŠTETÁKOVÁ, M. *Riadenie udalostného systému s využitím PLC a webovej aplikácie*. Diplomová práca (Vedúci práce: Erik Kučera). Bratislava: FEI STU, 2023.

- [99] ŠTĚPÁNEK, M. *Multiplatformová aplikácie pre získavanie dát o lokalitách s bezbariérovým prístupom*. Diplomová práca (Vedúci práce: Erik Kučera). Bratislava: FEI STU, 2018. 40 s.
- [100] ŽEMLA, F., CIGÁNEK, J., ROSINOVÁ, D., KUČERA, E., A HAFFNER, O. Smart Platform for Monitoring and Control of Discrete Event System in Industry 4.0 Concept. *Applied Sciences* 13, 19 (2023). DOI: 10.3390/app131910697.

doc. Ing. Erik Kučera, PhD.

## **DIGITÁLNE A INTELIGENTNÉ TECHNOLOGIE PRE INDUSTRY 4.0**

Vydala Slovenská technická univerzita v Bratislave vo Vydavateľstve SPEKTRUM  
STU, Bratislava, Vazovova 5, v roku 2024.

Edícia vysokoškolských učebníc

Rozsah 283 strán, 213 obrázkov, 7 tabuliek, 22,654 AH, 22,942 VH,  
1. vydanie, edičné číslo 6199.

85 – 217 – 2024

ISBN 978-80-227-5430-9

DOI: 10.61544/GYSJ2849