

Implementace nové generace řídicí jednotky Fischertechnik TXT 4.0 do hardwarového simulačního modelu představujícího koncept Industry 4.0

Miroslav Malaga, Zdeněk Ulrych

Západočeská univerzita v Plzni, Fakulta strojní, Katedra průmyslového inženýrství a managementu

Univerzitní 8, Plzeň, 306 14, Česká republika

malaga@kp.v.zcu.cz

ulrychz@kp.v.zcu.cz

Anotace: Článek se zabývá přechodem z řídicí jednotky Fischertechnik Robotics TXT Controller na novou generaci Fischertechnik TXT 4.0 Controller se zachováním grafického způsobu programování a odesláním dat např. do aplikace pro sběr dat z digitálního dvojčete běžící na PC. Zachováním grafického způsobu programování odpadá složité programování v jazycích C/C++ a kompilace pro linuxové jádro běžící na TXT 4.0 Controlleru a jeho ARM procesor. Navržené řešení využívá Raspberry Pi Pico jako překladač mezi sériovým portem PC a rozhraním I²C řídicí jednotky Fischertechnik a oproti původnímu řešení odpadá nutnost převodníku napěťových úrovní mezi překladačem (nově RaspberryPi Pico místo Arduino UNO) a řídicí jednotkou Fischertechnik. Nové řešení tak zachovává možnost využít potenciál jednotky Fischertechnik TXT 4.0 Controller i studenty v oboru průmyslového inženýrství a průmyslovými inženýry všeobecně, kteří nejsou programátory.

1 Úvod

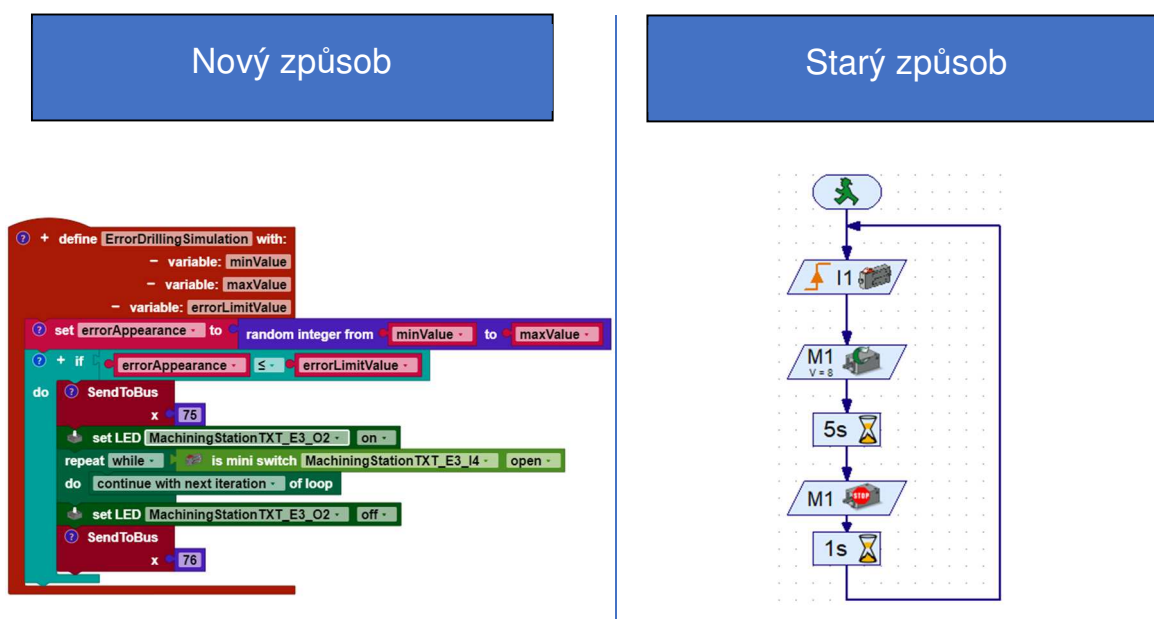
Cílem tohoto příspěvku je demonstrovat možnost nahrazení původní řídicí jednotky Fischertechnik Robotics TXT Controller novou jednotkou Fischertechnik TXT 4.0 Controller u hardwarových simulačních modelů při zachování největšího možného rozsahu řídicího kódu hardwarového modelu v grafickém programovacím jazyku.

Požadavek na maximální možné využití grafického programovacího jazyka vyplývá z účelu řízených modelů. Tyto modely jsou v našem prostředí využívány hlavně studenty průmyslového inženýrství a průmyslovými inženýry, kteří nejsou programátory, pouze disponují základními znalostmi z oblasti programování. Přesto ale potřebují vytvářet, nebo upravovat hardwarové modely sloužící např. ke sběru dat, tvorbě digitálních dvojčat, ověřování hypotéz, algoritmů a podobně. Pro tyto potřeby byly využívány řídicí jednotky Fischertechnik Robotics TXT Controller, kdy pro komunikaci s počítačem se používalo propojení jednotky Fischertechnik s Arduinem UNO s vloženým převodníkem napětí. Arduino pak předávalo data z řídicí jednotky Fischertechnik počítači na COM port. S příchodem nové generace řídicích

jednotek Fischertechnik TXT 4.0 Controller je potřeba původní propojení aktualizovat podle možností a způsobu programování nové řídicí jednotky.

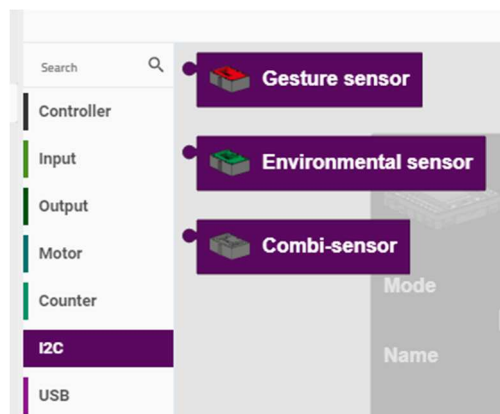
2 Programování nové řídicí jednotky Fischertechnik TXT 4.0

S novou generací řídicí jednotky Fischertechnik bylo představeno i nové vývojové prostředí ROBO Pro Coding, které nativně podporuje programování v grafickém programovacím jazyce (obdoba Scratch) anebo Pythonu. Ukázka původního způsobu grafického programování a nového způsobu grafického programování viz Obrázek 1. Pokud v grafickém programovacím jazyce chybí některé potřebné programovací bloky, lze přímo do grafického programovacího kódu vkládat pythonovský kód a obě podporované možnosti programování tak kombinovat. Novou řídicí jednotku nelze programovat pomocí starého vývojového prostředí a stejně tak starou řídicí jednotku nelze programovat pomocí nového vývojového prostředí. Protože i nová řídicí jednotka je samostatným počítačem, na kterém je nainstalovaný operační systém využívající linuxové jádro, není možné s ní komunikovat napřímo přes COM Port počítače stejně jako to nebylo možné u předchozí generace řídicí jednotky společnosti Fischertechnik.



Obrázek 1 - Ukázka ROBO Pro Coding vs ROBO Pro

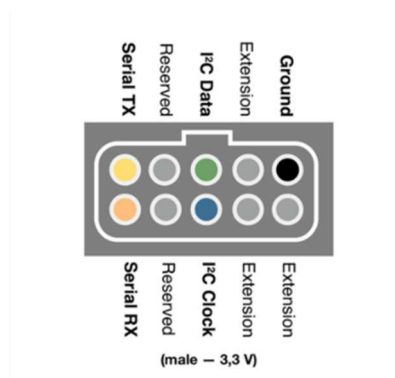
Původní řídicí jednotka s vývojovým prostředím přímo podporovala programovací grafické bloky pro komunikaci přes I²C sběrnici bez jakýchkoliv omezení. Nová řídicí jednotka s novým vývojovým prostředím obsahuje předpřipravené grafické programovací bloky pro komunikaci přes I²C sběrnici pouze pro vybrané aktivní prvky od společnosti Fischertechnik, konkrétně jejich Gesture sensor, Environmental sensor a Combi-sensor, viz Obrázek 2.



Obrázek 2 – Předpřipravené I²C prvky pro ROBO Pro Coding

Pro komunikaci s jiným zařízením přes I²C sběrnici je nutné využít v rámci grafického kódu vlastní Python script. Všeobecně lze u řídicí jednotky Fischertechnik komunikaci řešit přes zmíněnou I²C sběrnici, nebo bezdrátově přes Wi-Fi. Z důvodu komplikovaného povolení provozování vlastní Wi-Fi sítě v rámci naší univerzity a kvůli zpětné kompatibilitě s původní řídicí jednotkou Fischertechnik Robotics TXT byla prozatím nadále zvolena komunikace přes I²C sběrnici s pomocí překladače. Testování síťové komunikace pomocí Wi-Fi sítě bylo započato a plánuje se na něj v budoucnu přejít.

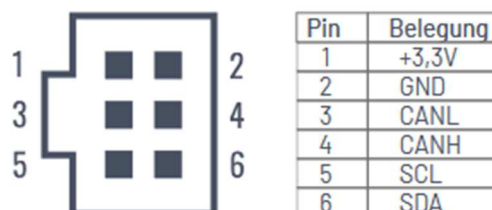
Poslední změna, která byla na řídicí jednotce provedena je změna EXT konektoru, tedy rozšiřujícího konektoru pomocí kterého lze propojit více řídicích jednotek dohromady a jsou na něm i vyvedeny piny sběrnice I²C pro připojení dalších zařízení. Popis konektoru EXT pro původní Fischertechnik Robotics TXT Controller, viz Obrázek 3.



Obrázek 3 - EXT konektor na Fischertechnik Robotics TXT Controller (stará generace) [1]

Nový konektor na řídicí jednotce Fischertechnik TXT 4.0 Controller je zmenšený o čtyři piny, přičemž možnost využití I²C sběrnice zůstala

zachována. Popis nového EXT konektoru je na následujícím obrázku, viz Obrázek 4.



Obrázek 4 - EXT konektor na Fischertechnik TXT 4.0 Controller (nová generace) [5]

3 Raspberry Pi Pico – překladač

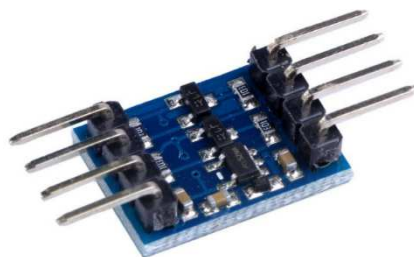
Při využití původního návrhu překladač dat s využitím Arduina UNO by nebylo potřeba do této části řešení zasahovat. S ohledem na nutnou aktualizaci minimálně softwarové části řešení s ohledem na nové řídicí jednotky Fischertechnik TXT 4.0 Controller se jeví jako vhodné upravit i tuto část. Původní návrh využívá Arduino UNO, které běží na napětí 5V, přičemž řídicí jednotky Fischertechnik běží na 3,3V. Z tohoto rozdílu pak vychází nutnost využití obousměrného převodníku napěťových úrovní mezi těmito dvěma jednotkami. Odstranění převodníku napětí lze poměrně jednoduše dosáhnout výměnou překladače za takový typ, který také využívá 3,3V. Toho lze docílit buď přechodem na jiný typ mikrokontroleru z rodiny Arduina, nebo na mikrokontroler od zcela jiného výrobce využívající napětí 3,3V. Po provedení průzkumu trhu bylo nakonec zvoleno Raspberry Pi Pico z těchto důvodů:

1. Cena – cena Raspberry Pi Pico osazené headery je na českém trhu 149 Kč [2], nebo 209 Kč v případě nově představené Wi-Fi varianty osazené headery [6]. V případě variant bez osazených headerů jsou ceny ještě nižší. Nejlevnější alternativa z rodiny Arduino běžící na 3,3V je aktuálně Arduino Nano 33 IoT za 599 Kč [3]. Pro porovnání aktuální cena Arduina UNO, který bylo využito v původním řešení, je 715 Kč [7].
2. Programovací jazyk – Raspberry Pi Pico lze programovat v C/C++ [9], Pythonu [9] i jazyce Wiring [10] který je určený pro mikrokontrolery z rodiny Arduino. Raspberry Pi Pico s čipem RP2040 má úplnou podporu ve vývojovém prostředí určeném pro řídicí jednotky Arduino a jimi využívaném programovacím jazyce Wiring [4]. Při testování bylo zjištěno, že Raspberry Pi Pico nemá v microPythonu možnost fungovat jako follower, ale pouze jako leader. Požadavek na tuto funkcionalitu je na GitHubu zadán a micropython by se to měl do budoucna naučit, aktuálně to ale není možné [8]. I to je důvodem, proč je v novém řešení využít programovací jazyk Wiring a Arduino IDE.

3. Provozní napětí 3,3V – je shodné s provozním napětím I²C sběrnice řídicí jednotky Fischertechnik TXT 4.0 Controller a v případě potřeby i jejího předchůdce Fischertechnik Robotics TXT Controller.
4. Velmi dobrá podpora výrobce a komunity.

4 Převodník napěťových úrovní

Kvůli rozdílným napěťovým hladinám I²C sběrnic v původním řešení využívající Arduino UNO (5V) a Fischertechnik TXT Controller (3,3V), bylo nutné použít převodník napěťových úrovní 3,3 V a 5 V viz Obrázek 5. Jednalo se o klíčový prvek komunikace I²C, a bez jeho použití by mohlo dojít k poškození jednotky Fischertechnik TXT Controller. Pro sběrnici I²C musel být použitý obousměrný převodník, jinak by původní řešení nefungovalo. Díky sjednocení napětí na 3,3V výměnou překladače (Arduino UNO za Raspberry Pi Pico) odpadla nutnost převodníku napětí a je možné tyto jednotky propojit napřímo.



Obrázek 5 – Obousměrný I²C převodník napěťových úrovní 3,3V-5V [11]

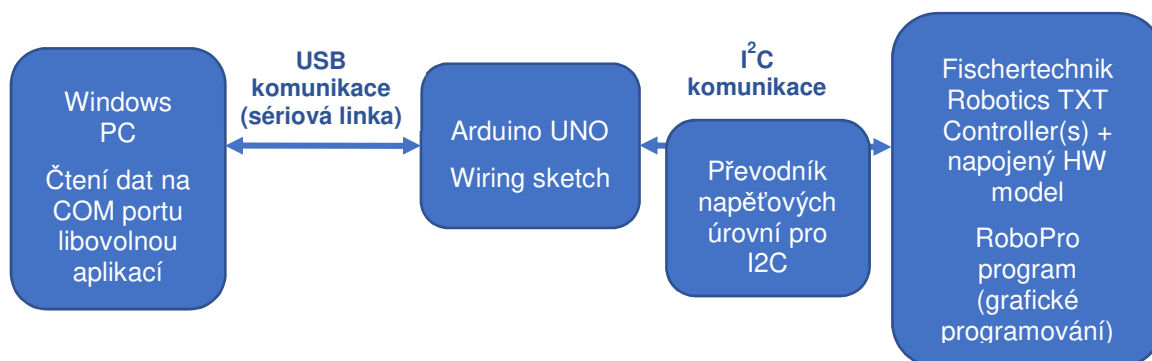
5 Návrh nového komunikačního rozhraní

Komunikační rozhraní je řešeno na dvou úrovních – hardwarové a programové (softwarové) úrovni.

5.1 Hardwarové propojení

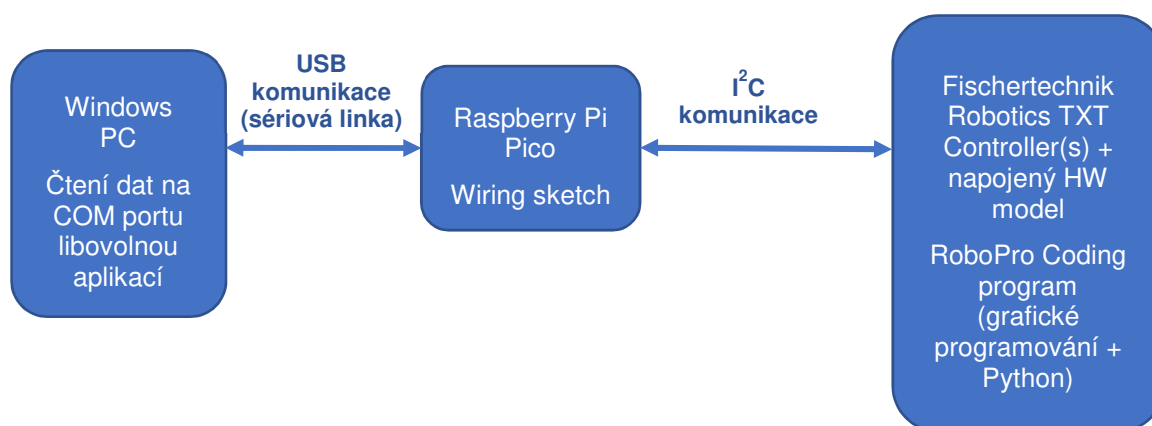
Původní i nové řešení rozšiřuje možnosti použití stavebnice Fischertechnik řízené řídicími jednotkami Fischertechnik TXT 4.0 Controller a Fischertechnik Robotics TXT Controller s počítačem a zároveň nabízí uživatelsky přívětivou možnost konfigurace tohoto propojení.

Schéma původního řešení využívající řídicí jednotku Fischertechnik Robotics TXT Controller, překladač Arduino UNO a nutný obousměrný převodník napěťových úrovní je ukázáno na následujícím obrázku, viz Obrázek 6.



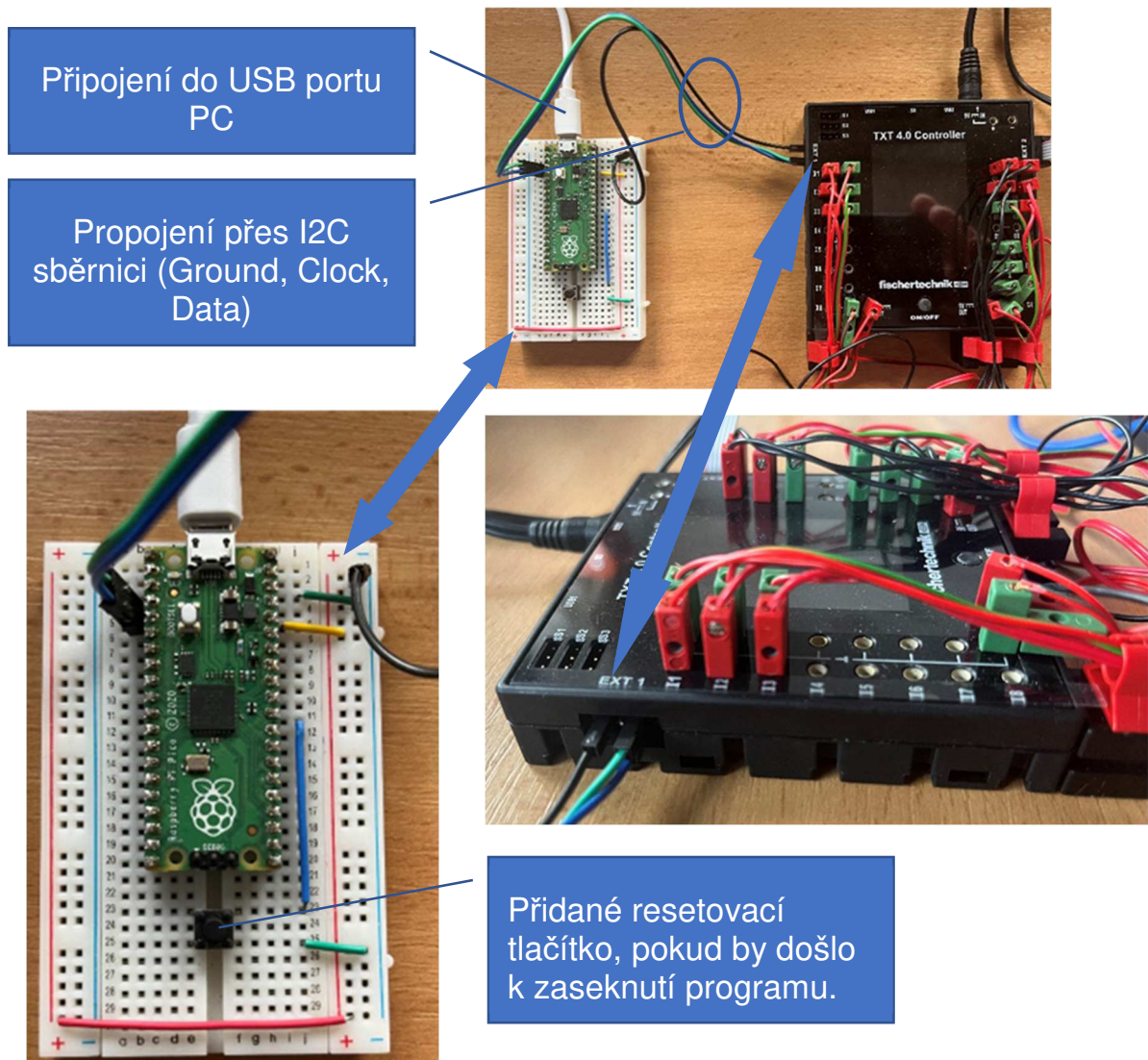
Obrázek 6 – Schéma původního HW propojení

Schéma nového řešení využívající řídicí jednotku Fischertechnik TXT 4.0 Controller s Raspberry Pi Pico jako překladačem je ukázáno na následujícím obrázku, viz Obrázek 7.



Obrázek 7 – Schéma nového HW propojení

Ukázka fyzického propojení realizovaného pro sběr dat z HW modelu výrobní linky využívaného pro prezentaci možností digitálního dvojčete a sběr dat je na následujícím obrázku, viz Obrázek 8.



Obrázek 8 – Ukázka využívaného reálného propojení

5.2 Softwarové propojení

Softwarové propojení se skládá z programu běžícího na řídicí jednotce Fischertechnik TXT 4.0 Controller, kde se kombinuje grafické programování s kódem v jazyce Python a programu běžícího na překladači Raspberry Pi Pico, kde se využívá programovací jazyk Wiring.

5.2.1 Řídicí program pro Fischertechnik TXT 4.0 Controller

Pro komunikaci přes I²C sběrnici je nutné využít možnosti vložení vlastního kódu v jazyce Python do řídicího kódu vytvářeného pomocí grafického způsobu programování. V tomto pythonovském kódu je potřeba použít balíček smbus2, která je součástí Pythonu v řídicí jednotce Fischertechnik TXT 4.0 Controller. Na následujícím obrázku, viz Obrázek 9, je ukázka vlastního skriptu v jazyce Python, který je pak v kódu volaný jako funkce s parametrem, který má přes I²C rozhraní předat překladači. Kvůli rychlosti a zpětné kompatibilitě

s původním řešením se předpokládá předávání dat v délce 1 Byte, tedy hodnoty 0-255. Tento rozsah je pro naše aktuální potřeby dostačující, jen je potřeba stanovit a implementovat převodní tabulku, která hodnotám přiřazuje určité stavy v rámci hardwarového modelu. S budoucím přechodem na komunikaci přes Wi-Fi se počítá i s předáváním celých stringů a strukturovaných dat např. pomocí JSON.

The image shows a Python code editor with the following code:

```
python imports
from smbus2 import SMBus

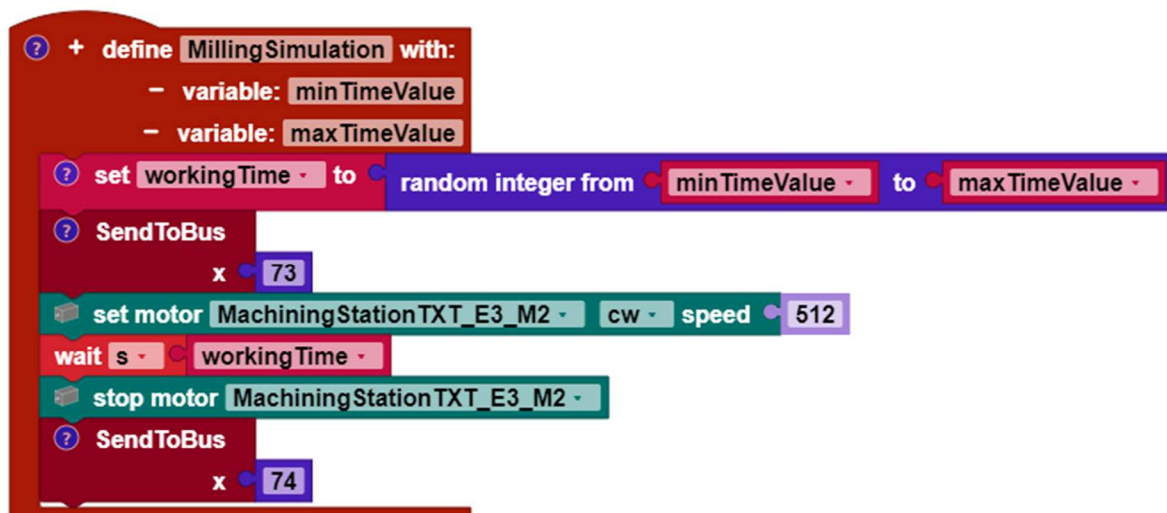
+ define SendToBus with:
  - variable: x
  wait s 0.2
  python code
  with SMBus(3) as bus:
    # Write a byte to address 0x27, offset 0
    bus.write_byte_data(0x27, 0, x)
  wait s 0.2
```

Importovaný balíček smbus2

Odeslání hodnoty, která byla funkci předána při jejím volání na adresu 0x27 (byte adresa překladače)

Obrázek 9 – Ukázka funkce pro odeslání dat přes I2C sběrnici

Ukázka volání vytvořené funkce v pythonu v rámci kódu pro frézování je na následujícím obrázku, viz Obrázek 10. Účelem této funkce je simulovat na hardwarovém modelu frézování o náhodné době trvání omezené minimální a maximální hodnotou. Při startu frézování má být odeslána informace, že frézování začalo (hodnota 73 – Začátek frézování) a po ukončení frézování má být odeslána informace, že frézování skončilo (hodnota 74 – Frézování ukončeno). V tuto chvíli přiřazujeme časy v aplikaci běžící na PC podle času obdržení informace o změně stavu v hardwarovém modelu z překladače. S přechodem na komunikaci přes Wi-Fi bude možné určit a přiřadit čas dané události přímo na straně řídicí jednotky Fischertechnik a s informací o změně stavu odeslat i čas, kdy k dané změně došlo.



Obrázek 10 – Ukázka využití funkce pro odesílání dat přes I2C sběrnici v kódu

5.2.2 Řídící program pro I²C/COM překladač Raspberry Pi Pico

Úkolem překladače je předávat data, která přijdou přes I²C sběrnici na sériovou linku (USB, přes které je překladač připojen k PC a kde lze přijatá data číst na COM portu), nebo obráceně data přijatá přes sériovou linku posílat na I²C sběrnici dané řídicí jednotce Fischertechnik. Aktuálně v rámci našeho modelu řešíme sběr dat pro řízení jejich prezentaci v mobilní aplikaci, nebo pro řízení digitálního dvojčete v simulačním nástroji Tecnomatix Plant Simulation. Proto následující kód řeší pouze překlad dat z I²C sběrnice na sériovou linku.

S ohledem na zjištění, že v jazyce micropython nelze aktuálně použít Raspberry Pi Pico v modu follower, je řídicí program vytvořený v jazyce Wiring s využitím Arduino IDE. Raspberry Pi Pico má plnou podporu ze strany Arduina, které dokonce nabízí vlastní klon Raspberry Pi Pica nazvaný Arduino NANO RP2040 využívající čip RP2040 od společnosti Raspberry Pi [4].

Ukázka okomentovaného řídicího kódu pro Raspberry Pi Pico viz Obrázek 11.

```

1 // University of West Bohemia
2 // Department of Industrial Engineering
3 //
4 // usage for communication between HW model and digital twin
5 // data translator between Fischertechnik model and Computer C# application
6 // the translator is RaspberryPi Pico as I2C follower, Fischertechnik TXT 4.0 is a Controller
7 //
8 // If there is a different voltage between Controller and Follower, the voltage converter must be used
9 //
10 // Miroslav Malaga, malaga@kpv.zcu.cz
11 // November 2021
12 // version n. 3
13 //
14 // tested and used with library: Arduino Mbed OS RP2040 Boards 2.6.1
15
16 #include <Wire.h> // using the Wire.h library to enable the i2c bus usage
17
18 int msg; // here will be stored a new message from Controller, message is a number
19 const byte f_addr = 0x27; //the I2C adress of the follower in hexadecimal system
20 MbedI2C MyWire(p2,p3); //using GP2(SDA) and GP3(SDA) pins for I2C communication
21
22 void setup()
23 {
24   pinMode(LED_BUILTIN, OUTPUT); // initialize digital pin for built in LED as an output
25   MyWire.begin(f_addr); // join i2c bus with address #39, address must be unique in used controllers
26   Serial.begin(9600); // start serial for output, the same baud rate must be on the computer side
27   MyWire.onReceive(receiveEvent); // register event
28 }
29
30 // function that executes whenever data is received from Controller, function is registered as an event, see setup()
31 void receiveEvent(int numBytes) // parameter tells how many bytes have been received, and consequently how many can be read with Wire.read()
32 {
33   digitalWrite(LED_BUILTIN, HIGH); // LED as visual info, that data transfer is in progress
34   MyWire.read(); // reading a first byte - the info that something is comming, dropping it...
35   msg = MyWire.read(); // reading a second byte - the new message
36   Serial.println(msg); // sending the message to serial
37   //delay(1000); // time delay to light up the LED if its necessary
38   digitalWrite(LED_BUILTIN, LOW); // LED as visual info, that data transfer is done
39 }
40
41 void loop()
42 {
43   //delay(100); //just to have some pause, protection against congestion, used only when testing
44 }

```

Obrázek 11 – Řídící kód pro překladač Raspberry Pi Pico

6 Závěr a hodnocení

Příspěvek se zabývá možnostmi náhrady řídicí jednotky Fischertechnik Robotics TXT Controller za jednotku nové generace Fischertechnik TXT 4.0 Controller pro řízení hardwarových modelů používaných např. pro vzdělávání, testování, ověřování v oblasti průmyslu 4.0. Nová řídicí jednotka má zcela nový způsob programování. Grafický způsob programování sestavováním vývojového diagramu je nahrazen grafickým způsobem typu Scratch, který je podstatně bližší klasickému způsobu programování. Tento způsob programování je navíc rozšířený o možnost kombinovat grafický způsob programování s programováním v Pythonu.

Souběžně s přechodem na novou generaci řídicích jednotek byla navržena a realizována úprava spočívající v nahrazení mikrokontroleru Arduino UNO za mikrokontroler Raspberry Pi Pico, které je podstatně levnější a využívá stejného provozního napětí 3,3V jako řídicí jednotky Fischertechnik. Tato úprava umožnila vypustit obousměrný převodník napěťových úrovní pro I²C sběrnici. Tím se celé řešení zjednodušilo a zvýšila se jeho robustnost. U nového typu překladače se stále využívá řídicí program v jazyce Wiring a Arduino IDE. V případě potřeby by tedy bylo možné vyměnit Raspberry Pi Pico za jiný 3,3V mikrokontroler z rodiny Arduino s minimálními změnami v řídicím programu.

Aktuální řešení využívající Raspberry Pi Pico je kompatibilní s původními řídicími jednotkami, což je aktuálně považováno za výhodu s ohledem na dostupnost starého typu řídicích jednotek. Do budoucna je ovšem plánovaný

u nových řídicích jednotek přechod na bezdrátový typ komunikace pomocí Wi-Fi sítě. U takové řešení se předpokládá nejen nárůst rychlost výměny dat a podstatné zjednodušení komunikace ve druhém směru (PC – model), ale i odstranění aktuálního limitu množství posílaných dat.

Výsledné řešení propojení využité pro propojení vlastní HW training factory s digitálním dvojčtem v nástroji Tecnomatix Plant Simulation vykazuje vysokou spolehlivost provozu. Během testování se projevilo minimum problémů, kdy se jednalo o problémy s komunikací na sběrnici I²C, které způsobovali minimální časové rozestupy mezi událostmi v HW modelu. Toto bylo vyřešeno úpravou řídicího programu HW training factory.

Navržené řešení představuje užitečné znalosti v oblasti průmyslového inženýrství, kdy umožňuje realizovat, nebo upravovat HW modely používané pro vzdělávání v oblasti Industry 4.0, tvorbě digitálních dvojčat a ukazuje i např. možnosti využití STEM produktů pro implementaci základních prvků konceptu Industry 4.0 do standardní výroby průmyslovými inženýry, kteří mají jen zcela základní znalosti v programování a elektronice.

Poděkování

Příspěvek byl vytvořen za podpory projektu SGS-2021-028 s názvem "Vývojové a tréninkové prostředky pro interakci člověka a kyber-fyzického výrobního systému" řešeného v rámci Interní grantové agentury Západočeské univerzity v Plzni.

Citovaná literatura

- [1] How to Connect an I²C Device to the Robotics TXT? *Rei Vilo's fischertechnik Corner* [online]. [vid. 2022-08-03]. Dostupné z: <https://reivilofischertechnik.weebly.com/how-to-connect-an-isup2c-device-to-the-robotics-txt.html>
- [2] Raspberry Pi Pico H. *RPishop.cz* [online]. [vid. 2022-08-03]. Dostupné z: <https://rpishop.cz/raspberry-pi/5069-raspberry-pi-pico-h-5056561803180.html>
- [3] Arduino Nano 33 IoT. *RPishop.cz* [online]. [vid. 2022-08-03]. Dostupné z: <https://rpishop.cz/arduino/2379-arduino-nano-33-iot.html>
- [4] *Nano RP2040 Connect | Arduino Documentation* [online]. [vid. 2022-08-03]. Dostupné z: <https://docs.arduino.cc/hardware/nano-rp2040-connect>
- [5] FISCHERTECHNIK GMBH. *Fischertechnik ROBOTICS TXT 4.0 CONTROLLER Operating manual* [online]. [vid. 2022-08-03]. Dostupné z: https://www.fischertechnik.de/-/media/fischertechnik/fite/service/elearning/lehren/txt40controller/fischertechnik-txt-operating-manual_en.ashx

- [6] Raspberry Pi Pico WH. *RPishop.cz* [online]. [vid. 2022-08-03]. Dostupné z: <https://rpishop.cz/raspberry-pi/5074-raspberry-pi-pico-wh-5056561800196.html>
- [7] Arduino Uno Rev3, originál. *RPishop.cz* [online]. [vid. 2022-08-03]. Dostupné z: <https://rpishop.cz/arduino/1356-arduino-uno-r3.html>
- [8] *Request I2C Slave-mode be added to micropython (#3935 follow-up) · Issue #8952 · micropython/micropython · GitHub* [online]. [vid. 2022-08-03]. Dostupné z: <https://github.com/micropython/micropython/issues/8952>
- [9] LTD, Raspberry Pi. Raspberry Pi Pico. *Raspberry Pi* [online]. [vid. 2022-08-03]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-pico/>
- [10] Using the Raspberry Pi Pico with Arduino. *Microcontroller Tutorials* [online]. 21. červen 2021 [vid. 2022-08-03]. Dostupné z: <https://www.teachmemicro.com/raspberry-pi-pico-with-arduino/>
- [11] GM ELECTRONIC, SPOL. S.R.O. *Převodník úrovní 5V - 3V | GM electronic, spol. s.r.o.* [online]. [vid. 2022-08-03]. Dostupné z: <https://www.gme.cz/prevodnik-urovni-5v-3-3v>